

# SUSE Linux

10.1

[www.novell.com](http://www.novell.com)

14. dubna 2006

Referenční příručka



## **Referenční příručka**

**Autoři:** Jörg Arndt, Stefan Behlert, Frank Bodammer, James Branam, Volker Buzek, Klara Cihlarova, Stefan Dirsch, Olaf Donjak, Roman Drahtmüller, Thorsten Dubiel, Torsten Duwe, Thomas Fehr, Stefan Fent, Werner Fink, Jakub Friedl, Kurt Garloff, Joachim Gleißner, Carsten Groß, Andreas Grünbacher, Berthold Gunreben, Franz Hassels, Andreas Jaeger, Jana Jaeger, Klaus Kämpf, Andi Kleen, Hubert Mantel, Lars Marowsky-Bree, Chris Mason, Johannes Meixner, Lars Müller, Matthias Nagorni, Anas Nashif, Siegfried Olschner, Edith Parzefall, Peter Pöml, Thomas Renninger, Hannes Reinecke, Thomas Rölz, Heiko Rommel, Marcus Schäfer, Thomas Schraitle, Klaus Singvogel, Hendrik Vogelsang, Klaus G. Wagner, Rebecca Walter, Christian Zoz

Toto dílo je duševním vlastnictvím společnosti SuSE CR, s.r.o a Novell Inc. Je možné ho kopírovat jako celek nebo jeho části při dodržení povinnosti uvést na každé kopii toto upozornění o autorských právech.

Všechny programy, obrázky a informace uvedené v těchto materiálech jsou pečlivě kontrolovány, ale ani tak není možné zcela vyloučit výskyt případných chyb. Z tohoto důvodu nejsme s to nést žádné záruky jakéhokoli druhu za případné vzniklé škody spojené s používáním této příručky. Autoři, překladatelé, ani SuSE CR, s.r.o., resp. SUSE Linux AG neposkytují žádné záruky a nenesou odpovědnost za případné škody vzniklé používáním těchto manuálů nebo programů zde uvedených uživateli samotným nebo třetím stranám.

Všechny názvy produktů jsou bez záruky volného používání a může se jednat o registrované obchodní značky. SuSE CR, s.r.o. se obecně řídí informacemi výrobce. Jiné, zde uvedené, produkty mohou být obchodními značkami stávajících výrobců.

Poznámky a komentáře směrujte na adresu [feedback@suse.cz](mailto:feedback@suse.cz) [mailto:feedback@suse.cz].

# Obsah

<b>O této příručce</b>	<b>xi</b>
<b>Část 1 Možnosti nasazení pro pokročilé</b>	<b>15</b>
<b>1 Vzdálená instalace</b>	<b>17</b>
1.1 Scénáře vzdálené instalace . . . . .	17
1.2 Nastavení serveru s instalačním zdrojem . . . . .	25
1.3 Příprava startu cílového systému . . . . .	33
1.4 Spuštění instalace na cílovém systému . . . . .	38
1.5 Instalační proces . . . . .	42
<b>2 Rozdělení disku pro experty</b>	<b>47</b>
2.1 Konfigurace LVM . . . . .	47
2.2 Konfigurace softwarového RAIDu . . . . .	53
<b>3 Aktualizace systému</b>	<b>59</b>
3.1 Aktualizace systému SUSE Linux . . . . .	59
3.2 Od verze k verzi . . . . .	61
<b>Část 2 Správa</b>	<b>77</b>
<b>4 Bezpečnost v Linuxu</b>	<b>79</b>
4.1 Firewall a maškaráda . . . . .	79
4.2 SSH: bezpečná práce v síti . . . . .	89
4.3 Šifrování diskových oddílů a souborů . . . . .	95

4.4	Bezpečnost a soukromí . . . . .	97
<b>5</b>	<b>ACLs v Linuxu</b>	<b>109</b>
5.1	Výhody ACLs . . . . .	109
5.2	Definice . . . . .	110
5.3	Používání ACLs . . . . .	110
5.4	Výhledy . . . . .	119
5.5	Další informace . . . . .	119
<b>6</b>	<b>Nástroje monitorování systému</b>	<b>121</b>
6.1	Seznam otevřených souborů: <code>lsdf</code> . . . . .	121
6.2	Přístup uživatelů k souborům: <code>fuser</code> . . . . .	123
6.3	Vlastnosti souboru: <code>stat</code> . . . . .	123
6.4	USB zařízení: <code>lsusb</code> . . . . .	124
6.5	SCSI zařízení: <code>scsiinfo</code> . . . . .	125
6.6	Procesy: <code>top</code> . . . . .	125
6.7	Seznam procesů: <code>ps</code> . . . . .	126
6.8	Strom procesů: <code>pstree</code> . . . . .	127
6.9	Kdo co dělá: <code>w</code> . . . . .	128
6.10	Využití paměti: <code>free</code> . . . . .	129
6.11	Systémové hlášení jádra: <code>dmesg</code> . . . . .	129
6.12	Souborový systém a jeho využití: <code>mount</code> , <code>df</code> a <code>du</code> . . . . .	130
6.13	Souborový systém <code>/proc</code> . . . . .	131
6.14	<code>vmstat</code> , <code>iostat</code> a <code>mpstat</code> . . . . .	133
6.15	<code>procinfo</code> . . . . .	133
6.16	PCI zdroje: <code>lspci</code> . . . . .	135
6.17	Systémová volání běžícího programu: <code>strace</code> . . . . .	136
6.18	Volání knihoven běžícím příkazem: <code>ltrace</code> . . . . .	137
6.19	Zjištění vyžadovaných knihoven: <code>ldd</code> . . . . .	137
6.20	Dodatečné informace o ELF binárních souborech . . . . .	138
6.21	Meziprocesová komunikace: <code>ipcs</code> . . . . .	138
6.22	Měření času: <code>time</code> . . . . .	139
<b>Část 3</b>	<b>System</b>	<b>141</b>
<b>7</b>	<b>32- a 64-bitové aplikace v 64-bitovém prostředí</b>	<b>143</b>
7.1	Podpora běhu aplikací . . . . .	143
7.2	Vývoj softwaru . . . . .	144
7.3	Kompilace softwaru pro jinou platformu . . . . .	144
7.4	Specifikace jádra . . . . .	146

<b>8</b>	<b>Startování</b>	<b>147</b>
8.1	Startovací proces v Linuxu . . . . .	147
8.2	Program init . . . . .	150
8.3	Úroveň běhu . . . . .	151
8.4	Změna úrovně běhu . . . . .	152
8.5	Init skripty . . . . .	153
8.6	Editor úrovní běhu . . . . .	157
8.7	SuSEconfig a /etc/sysconfig . . . . .	159
8.8	YaST sysconfig Editor . . . . .	160
<b>9</b>	<b>Starování systému a zavaděče</b>	<b>163</b>
9.1	Startování . . . . .	164
9.2	Výběr zavaděče . . . . .	165
9.3	Startování systému se zavaděčem GRUB . . . . .	165
9.4	Odstalace zavaděče LILO nebo GRUB . . . . .	175
9.5	Vytvoření startovacího CD . . . . .	178
9.6	Grafická konzole SUSE . . . . .	179
9.7	Řešení problémů . . . . .	179
9.8	Další informace . . . . .	181
<b>10</b>	<b>Zvláštní funkce systému SUSE Linux</b>	<b>183</b>
10.1	Nápověda k některým zvláštním balíčkům . . . . .	183
10.2	Virtuální konzole . . . . .	190
10.3	Mapování klávesnice . . . . .	190
10.4	Lokální přízpůsobení — I18N and L10N . . . . .	191
<b>11</b>	<b>Obsluha tisku</b>	<b>195</b>
11.1	Práce tiskového systému . . . . .	196
11.2	Způsoby a protokoly pro připojení tiskáren . . . . .	197
11.3	Instalace softwaru . . . . .	197
11.4	Konfigurace tiskárny . . . . .	198
11.5	Nastavení aplikací . . . . .	204
11.6	Zvláštní vlastnosti v systému SUSE Linux . . . . .	205
11.7	Řešení problémů . . . . .	210
<b>12</b>	<b>Dynamické uzly zařízení pomocí udev</b>	<b>217</b>
12.1	Tvorba pravidel . . . . .	218
12.2	Automatizace pomocí NAME a SYMLINK . . . . .	218
12.3	Regulární výrazy v klíčích . . . . .	219
12.4	Výběr klíčů . . . . .	219
12.5	Konzistentní pojmenování zařízení pro hromadné uchovávání dat . . . . .	220

<b>13</b>	<b>Souborové systémy</b>	<b>223</b>
13.1	Termíny	223
13.2	Hlavní souborové systémy Linuxu	224
13.3	Některé další podporované souborové systémy	229
13.4	Podpora souborů větších než 2 GB	231
13.5	Další informace	232
<b>14</b>	<b>Systém X Window</b>	<b>235</b>
14.1	Nstavení X11 pomocí SaX2	235
14.2	Optimalizace systému X Window	237
14.3	Instalace a konfigurace fontů	243
14.4	Konfigurace OpenGL – 3D	249
<b>15</b>	<b>FreeNX: vzdálené ovládaní plochy</b>	<b>253</b>
15.1	Úvod do NX	253
15.2	Možné problémy	255
15.3	Další informace	257
<b>16</b>	<b>Autentizace pomocí PAM</b>	<b>259</b>
16.1	Struktura PAM konfiguračního souboru	260
16.2	Konfigurace PAM pro sshd	261
16.3	Konfigurace PAM modulů	264
16.4	Další informace	266
<b>17</b>	<b>Virtualizace pomocí Xenu</b>	<b>267</b>
17.1	Instalace Xenu	268
17.2	Instalace domény	268
17.3	Konfigurace domény Xenu	271
17.4	Spuštění a správa Xen domén	272
17.5	Více informací	274
<b>Část 4</b>	<b>Služby</b>	<b>275</b>
<b>18</b>	<b>Základy síťování</b>	<b>277</b>
18.1	IP adresy a směrování	280
18.2	IPv6 – Internet další generace	283
18.3	Překlad jmen	291
18.4	Konfigurace síťového připojení pomocí YaST	292
18.5	Správa sítě s programem NetworkManager	303

18.6	Manuální konfigurace sítě . . . . .	306
18.7	smpppd jako pomocník s vytáčeným připojením . . . . .	316
<b>19</b>	<b>SLP služby v síti</b>	<b>319</b>
19.1	Registrace vlastních služeb . . . . .	319
19.2	SLP frontendy v systému SUSE Linux . . . . .	320
19.3	Aktivace SLP . . . . .	321
19.4	Další informace . . . . .	321
<b>20</b>	<b>DNS — Domain Name System</b>	<b>323</b>
20.1	Konfigurace pomocí YaST . . . . .	323
20.2	Spuštění nameserveru BIND . . . . .	329
20.3	Konfigurační soubor /etc/named.conf . . . . .	331
20.4	Nejdůležitější konfigurační volby v sekci options . . . . .	332
20.5	Konfigurace v sekci logging . . . . .	333
20.6	Struktura souboru odkazujícího na data pro zóny . . . . .	334
20.7	Struktura souboru s daty pro zónu . . . . .	335
20.8	Dynamická aktualizace údajů o zóně . . . . .	339
20.9	Bezpečné transakce . . . . .	339
20.10	DNSSEC . . . . .	341
20.11	Další informace . . . . .	341
<b>21</b>	<b>NIS — Network Information Service</b>	<b>343</b>
21.1	Konfigurace NIS serveru . . . . .	343
21.2	Konfigurace NIS klientů . . . . .	346
<b>22</b>	<b>NFS — sdílené souborové systémy</b>	<b>349</b>
22.1	Importování souborových systémů pomocí YaST2 . . . . .	349
22.2	Ruční import souborových systémů . . . . .	350
22.3	Exportování souborových systémů pomocí YaST . . . . .	350
22.4	Ruční export souborových systémů . . . . .	352
<b>23</b>	<b>DHCP</b>	<b>355</b>
23.1	DHCP protokol . . . . .	355
23.2	Konfigurace DHCP serveru pomocí nástroje YaST . . . . .	356
23.3	DHCP softwarové balíčky . . . . .	358
23.4	DHCP server dhcpd . . . . .	358
23.5	Další informace . . . . .	362

<b>24</b>	<b>Synchronizace času pomocí xntp</b>	<b>363</b>
24.1	Nastavení NTP klienta v programu YaST . . . . .	363
24.2	Nastavení xntp v síti . . . . .	366
24.3	Nastavení lokálních referenčních hodin . . . . .	367
<b>25</b>	<b>LDAP — adresářové služby</b>	<b>369</b>
25.1	LDAP versus NIS . . . . .	371
25.2	Struktura adresářového stromu LDAP . . . . .	371
25.3	Konfigurace LDAP serveru pomocí slapd.conf . . . . .	374
25.4	Správa dat v LDAP adresáři . . . . .	379
25.5	YaST LDAP klient . . . . .	382
25.6	Další informace . . . . .	389
<b>26</b>	<b>Webový server Apache</b>	<b>391</b>
26.1	Instalace . . . . .	391
26.2	Start serveru Apache . . . . .	392
26.3	Konfigurace webového serveru . . . . .	392
26.4	Používání Apache . . . . .	398
26.5	Aktivní obsah . . . . .	399
26.6	Moduly Apache . . . . .	401
26.7	Vlákna (threads) . . . . .	405
26.8	Bezpečnost . . . . .	406
26.9	Možné problémy . . . . .	407
26.10	Další dokumentace . . . . .	408
<b>27</b>	<b>Synchronizace souborů</b>	<b>411</b>
27.1	Programy pro datovou synchronizaci . . . . .	411
27.2	Výběr vhodného programu . . . . .	413
27.3	Úvod do Unison . . . . .	417
27.4	Úvod do programu CVS . . . . .	419
27.5	Úvod do Subversion . . . . .	422
27.6	Úvod do rsync . . . . .	425
27.7	Úvod do mailsync . . . . .	426
<b>28</b>	<b>Samba</b>	<b>431</b>
28.1	Nastavení serveru . . . . .	432
28.2	Samba jako přihlašovací server . . . . .	437
28.3	Konfigurace Samba serveru pomocí programu YaST . . . . .	438
28.4	Nastavení klienta . . . . .	440
28.5	Optimalizace . . . . .	440



<b>29 Proxy server Squid</b>	<b>443</b>
29.1 Informace o proxy-cache . . . . .	444
29.2 Systémové požadavky . . . . .	445
29.3 Spuštění squida . . . . .	447
29.4 Konfigurační soubor /etc/squid/squid.conf . . . . .	449
29.5 Konfigurace transparentní proxy . . . . .	454
29.6 cachemgr.cgi . . . . .	457
29.7 squidGuard . . . . .	459
29.8 Vytvoření protokolů programem Calamaris . . . . .	460
29.9 Další informace o Squidu . . . . .	461
<b>Část 5 Mobilita</b>	<b>463</b>
<b>30 Mobilita v Linuxu</b>	<b>465</b>
30.1 Notebooky . . . . .	465
30.2 Mobilní hardware . . . . .	471
30.3 Mobilní telefony a kapesní počítače . . . . .	472
30.4 Další informace . . . . .	473
<b>31 Linux a notebooky</b>	<b>475</b>
31.1 Hardware . . . . .	475
31.2 Software . . . . .	476
<b>32 Správa profilů</b>	<b>477</b>
32.1 Základní terminologie . . . . .	478
32.2 Nastavení SCPM . . . . .	478
32.3 Volba profilu při startu . . . . .	483
32.4 Problémy a jejich řešení . . . . .	483
32.5 Další informace . . . . .	484
<b>33 Správa napájení</b>	<b>487</b>
33.1 Funkce šetření spotřeby . . . . .	487
33.2 APM . . . . .	489
33.3 ACPI . . . . .	490
33.4 Zastavení disku . . . . .	496
33.5 Balík powersave . . . . .	497
33.6 Modul správy napájení programu YaST . . . . .	506

<b>34</b>	<b>Bezdrátová komunikace</b>	<b>511</b>
34.1	Bezdrátové sítě . . . . .	511
34.2	Bluetooth . . . . .	520
34.3	IrDA — Infrared Data Association . . . . .	530
	<b>Rejstřík</b>	<b>535</b>

# O této příručce

Tato příručka obsahuje podrobnější informace o systému SUSE Linux. Hodit se vám bude, pokud jste pokročilejší domácí uživatelé spravující svůj systém nebo jste systémoví administrátoři. Najdete zde nejen popis aplikací, které se hodí pro každodenní práci, ale také informace o možnostech nasazení systému SUSE Linux a expertní instalaci systému.

Možnosti nasazení pro pokročilé

Příklady nasazení systému SUSE Linux.

Správa

Naučte se, jak nastavit svůj systém SUSE Linux, aby byl co nejbezpečnější.

Systém

Popis komponent linuxového systému a jejich interakce.

Služby

Nastavení síťových i souborových služeb v systému SUSE Linux.

Mobilita

Úvod do mobilního světa se systémem SUSE Linux. Postup nastavení bezdrátového připojení, správy napájení a profilů.

## 1 Zpětná vazba

Velmi rádi uvítáme vaše připomínky k této příručce i k další dokumentaci, která je součástí tohoto produktu. Stačí, když kliknete na User Comments na každé stránce online dokumentace nebo použijete pro odeslání stránku <http://www.novell.com/documentation/feedback.html>.

## 2 Další dokumentace

K systému SUSE Linux najdete řadu velmi užitečné dokumentace na stránce <http://www.novell.com/documentation/> nebo přímo ve svém systému v adresáři `/usr/share/doc/manual/`:

### *Uživatelská příručka*

Příručka obsahující informace o instalaci a základní práci se systémem SUSE Linux. Online verzi najdete na stránce <http://www.novell.com/documentation/suse10/>.

### *Novell AppArmor Powered by Immunix 2.0 Installation and QuickStart Guide*

Instalace produktu *AppArmor*. Online verze je dostupná na stránce <http://www.novell.com/documentation/apparmor/>.

### *Novell AppArmor Powered by Immunix 2.0 Administration Guide*

Správa a používání produktu *AppArmor*. Online verze je dostupná na stránce <http://www.novell.com/documentation/apparmor/>.

## 3 Typografické konvence

V této knize se používají následující typografické konvence:

- `/etc/passwd`: soubory nebo adresáře.
- `Jmeno_uzivatele`: položku `Jmeno_uzivatele` nahradíte údajem platným ve svém systému.
- `PATH`: proměnné prostředí, zde `PATH`
- `ls --help`: příkaz a volba nebo parametr.
- `user`: uživatel.
- `[Alt]`, `[Alt] + [F1]`: klávesa, kombinace kláves.
- *Další*: tlačítka, položky nabídky atd.
- *Tančící tučňáci* (Tančící tučňáci, ↑*Referenční příručka*): odkaz na kapitolu v jiné příručce.

## 4 O vytváření tohoto manuálu

Pro vytvoření této knihy byl použit Novdoc, styl založený na DocBooku (viz <http://www.docbook.org>). Zdrojové XML byly validovány nástrojem `xmllint` a mezi formáty převáděny pomocí `xsltproc`, pro konverzi do XSL-FO byla použita upravené verze stylů Normana Walshe. Výsledné PDF bylo vytvořeno pomocí programu XEP od společnosti RenderX.

## 5 Poděkování

V první řadě děkujeme všem vývojářům, kteří se podílejí na vývoji operačního systému Linux. Děkujeme jim za jejich skvělou práci, bez nich by naše distribuce nemohla existovat. Také děkujeme Franku Zappovi, Pawar a Sněhulce.

A poslední a zároveň největší dík patří panu Linusi Torvaldsovi!

Have a lot of fun!

Váš SUSE Team



# **Část 1. Možnosti nasazení pro pokročilé**





# Vzdálená instalace

SUSE Linux lze nainstalovat různými způsoby. Tradiční způsob instalace z CD nebo DVD je popsán v kapitole „*Instalace pomocí nástroje YaST*“ (↑Uživatelská příručka). Mimo něj však můžete volit z celé řady síťových instalací.

Každá metoda je popsána z hlediska předpokladů pro její provedení a pak následuje základní jednoduchý postup pro nastavení procesu instalace. Pro všechny metody jsou pak popsány podobnější scénáře.

---

## Poznámka

O systému, na který má být nainstalován SUSE Linux zde budeme mluvit jako o *cílovém systému* nebo *cílu instalace*. Termín *instalační zdroj* je použit pro všechny typy instalačních zdrojů jako např. CD nebo DVD či síťové servery.

---

## 1.1 Scénáře vzdálené instalace

V této sekci najdete nejčastější scénáře vzdálené instalace. Každý scénář obsahuje předpoklady a postup. Podrobnější informace najdete v odkazech obsažených v textu.

---

## Důležité

Nastavení X Window není součástí vzdálené instalace. Po dokončení instalace se jako uživatel root přihlaste do systému, zadejte `telinit 3` a k nastavení

grafického prostředí použijte SaX2. Použití programu SaX2 je popsáno v kapitole 14.1 – „Nstavení X11 pomocí SaX2“ (strana 235).

---

## 1.1.1 Jednoduchá vzdálené instalace přes VNC — statická síť

U tohoto typu instalace je nutná počáteční fyzická přítomnost uživatele u cílového systému. Po spuštění je instalace prováděna vzdáleně z jiného počítače pomocí VNC. Instalační proces je kontrolován uživatelem z jiného počítače pomocí VNC prohlížeče nebo webového prohlížeče a postup je totožný s postupem uvedeným v kapitole „*Instalace pomocí nástroje YaST*“ (↑Uživatelská příručka).

Ujistěte se, že jsou splněny následující požadavky:

- Vzdálený instalační zdroj: NFS, HTTP, FTP nebo SMB s připojením do sítě
- Cílový systém s funkčním připojením k síti
- Externí systém s nainstalovaným VNC prohlížečem nebo webovým prohlížečem s podporou Javy (Firefox, Konqueror, Internet Explorer nebo Opera)
- Fyzické instalační médium (CD nebo DVD) pro spuštění instalace na sílovém systému
- Platná pevná IP adresa přidělená externímu systému a instalačnímu zdroji
- Cílový systém s platnou pevnou IP adresou

Při instalaci postupujte následujícím způsobem:

- 1 Nastavte instalační zdroj podle postupu v 1.2 – „Nastavení serveru s instalačním zdrojem“ (strana 25).
- 2 Spustěte instalaci cílového systému z CD nebo DVD.
- 3 Při startu nastavte parametry potřebné pro VNC instalaci. Podrobnosti najdete v části 1.4 – „Spuštění instalace na cílovém systému“ (strana 38).

Cílový systém se spustí do textového prostředí, kde vypíše svou adresu a display, na kterém je dostupné grafické prostředí. VNC instalace se použije ke svému ohlášení OpenSLP, takže bude dostupná např. v Konqueroru po zadání `service://` nebo `slp://` režimu.

- 4 Na externí stanici otevřete VNC prohlížeč nebo použijte pro ovládání instalace prohlížeč tak, jak je popsáno v 1.5.1 – „VNC instalace“ (strana 42).
- 5 Proveďte instalaci tak, jak je popsáno v „Instalace pomocí nástroje YaST“ (↑Uživatelská příručka).

Po restartu se musíte k systému znovu připojit, abyste instalaci dokončili.

- 6 Dokončete instalaci.

## 1.1.2 Jednoduchá vzdálená instalace přes VNC — dynamické nastavení sítě přes DHCP

U tohoto typu instalace je nutná počáteční fyzická přítomnost uživatele u cílového systému. Po spuštění je instalace prováděna vzdáleně z jiného počítače pomocí VNC. Instalační proces je kontrolován uživatelem z jiného počítače pomocí VNC prohlížeče nebo webového prohlížeče.

Ujistěte se, že jsou splněny následující požadavky:

- Vzdálený instalační zdroj: NFS, HTTP, FTP nebo SMB s funkčním síťovým připojením
- Cílový systém s funkčním připojením k síti
- Externí systém s nainstalovaným VNC prohlížečem nebo webovým prohlížečem s podporou Javy (Firefox, Konqueror, Internet Explorer nebo Opera)
- Fyzické instalační médium (CD nebo DVD) pro spuštění instalace na sílovém systému
- Běžící DHCP server poskytující adresy

Při instalaci postupujte následujícím způsobem:

- 1 Nastavte instalační zdroj podle postupu v [1.2 – „Nastavení serveru s instalačním zdrojem“](#) (strana 25). Zvolte NFS, HTTP nebo FTP server. Informace o SMB instalačním zdroji najdete v části [1.2.5 – „SMB instalační zdroj“](#) (strana 32).
- 2 Spustíte instalaci cílového systému z CD nebo DVD.
- 3 Při startu nastavte parametry potřebné pro VNC instalaci. Podrobnosti najdete v části [1.4 – „Spuštění instalace na cílovém systému“](#) (strana 38).

Cílový systém se spustí do textového prostředí, kde vypíše svou adresu a display, na kterém je dostupné grafické prostředí. VNC instalace se použije ke svému ohlášení OpenSLP, takže bude dostupná např. v Konqueroru po zadání `service://` nebo `slp://` režimu.

- 4 Na externím systému otevřete VNC prohlížeč nebo webový prohlížeč jako je popsáno v [1.5.1 – „VNC instalace“](#) (strana 42).
- 5 Provedte instalaci je jak popsáno v *„Instalace pomocí nástroje YaST“* (↑Uživatelská příručka).

Po restartu se musíte k systému znovu připojit, abyste instalaci dokončili.

- 6 Dokončete instalaci.

## 1.1.3 Vzdálená instalace přes VNC — PXE Boot a Wake on LAN

Tento typ instalace nevyžaduje při spuštění žádnou interakci. Cílový systém a instalace spustí automaticky. Uživatelský zásah je potřeba pouze u provedení instalace.

Ujistěte se, že jsou splněny následující požadavky:

- Vzdálený instalační zdroj: NFS, HTTP, FTP nebo SMB se síťovým připojením
- TFTP server
- Běžící DHCP server v lokální síti

- Cílový systém s podporou PXE bootu, sítě a Wake on LAN, zapojený do sítě a elektřiny
- Externí systém s nainstalovaným VNC prohlížečem nebo webovým prohlížečem s podporou Javy (Firefox, Konqueror, Internet Explorer nebo Opera)

Tento typ instalace provedete následujícím způsobem:

- 1 Nastavte instalační zdroj podle postupu v [1.2 – „Nastavení serveru s instalačním zdrojem“](#) (strana 25). Zvolte NFS, HTTP nebo FTP server nebo nastavte SMB zdroj podle postupu uvedeného v části [1.2.5 – „SMB instalační zdroj“](#) (strana 32).
- 2 Nastavte startovací obraz na TFTP serveru. Postup najdete v části [1.3.2 – „Nastavení TFTP serveru“](#) (strana 35).
- 3 Nastavte DHCP server a vložte do jeho nastavení také údaje o TFTP serveru. Postup je popsán v [1.3.1 – „Nastavení DHCP serveru“](#) (strana 34).
- 4 Nastavte na cílovém systému PXE boot. Postup je popsán v [1.3.3 – „PXE boot“](#) (strana 35).
- 5 Nastavte na cílovém systému Wake on LAN. Postup je popsán v [1.3.4 – „Ruční nastavení Wake on LAN“](#) (strana 38).
- 6 Na externím systému otevřete VNC prohlížeč nebo webový prohlížeč jako je popsáno v [1.5.1 – „VNC instalace“](#) (strana 42).
- 7 Projděte instalaci podle postupu uvedeného v *„Instalace pomocí nástroje YaST“* (↑Uživatelská příručka).

Abyste dokončili instalaci, musíte se po restartu systému znovu připojit.

- 8 Dokončete instalaci.

## 1.1.4 Jednoduchá vzdálená instalace přesSSH — statická síť

Tento typ instalace vyžaduje fyzickou přítomnost u cílového systému při startu instalace a zadání síťového nastavení. Samotná instalace je prováděna vzdáleně přes SSH. Insta-

laci provádí uživatel ze vzdáleného systému podle postupu uvedeného v „*Instalace pomocí nástroje YaST*“ (↑Uživatelská příručka).

Nutné předpoklady pro tento typ instalace:

- Vzdálený instalační zdroj: NFS, HTTP, FTP nebo SMB s funkčním síťovým připojením
- Cílový systém s funkčním připojením k síti
- Externí systém s připojením k síti a nainstalovaným SSH klientem
- Fyzické médium ke spuštění startu (CD, DVD, vlastní startovací disketa...) na cílovém systému
- Pevná IP adresa na externím systému a instalačním médiu
- Cílový systém s platnou pevnou IP adresou

Při instalaci postupujte následujícím způsobem:

- 1** Nastavte instalační zdroj podle postupu v [1.2 – „Nastavení serveru s instalačním zdrojem“](#) (strana 25).
- 2** Spustěte instalaci cílového systému z CD nebo DVD
- 3** V úvodní startovací obrazovce zadejte parametry pro nastavení sítě a povolení SSH. Postup je uveden v [1.4.3 – „Použití vlastních startovacích voleb“](#) (strana 40).

Cílový systém se spustí do textového prostředí a zobrazí svou IP adresu, na které je dosažitelný pro SSH klienty.

- 4** Na externí stanici otevřete okno terminálu a připojte se podle postupu uvedeného v „[Připojení k instalačnímu programu](#)“ (strana 44).
- 5** Projděte instalací jak je popsáno v části „*Instalace pomocí nástroje YaST*“ (↑Uživatelská příručka).

Abyste dokončili instalaci, musíte se po restartu systému znovu připojit k cílovému systému.

- 6** Dokončete instalaci.

## 1.1.5 Jednoduchá vzdálená instalace přes SSH — dynamické nastavení sítě přes DHCP

Tento typ instalace vyžaduje fyzickou přítomnost u cílového systému při startu instalace a zadání síťového nastavení. Samotná instalace je prováděna vzdáleně přes SSH. Instalaci provádí uživatel ze vzdáleného systému.

Pro tento typ instalace musíte splnit následující předpoklady:

- Vzdálený instalační zdroj: NFS, HTTP, FTP nebo SMB s funkčním síťovým připojením working network connection
- Cílový systém s funkčním připojením k síti
- Externí systém s připojením k síti a nainstalovaným SSH klientem
- Fyzické médium pro spuštění instalace (CD or DVD) na cílovém systému
- Běžící DHCP server poskytující IP adresy

Při instalaci postupujte následujícím způsobem:

- 1** Nastavte instalační zdroj podle postupu v [1.2 – „Nastavení serveru s instalačním zdrojem“](#) (strana 25). Zvolte NFS, HTTP nebo FTP server. Informace o SMB instalačním zdroji najdete v části [1.2.5 – „SMB instalační zdroj“](#) (strana 32).
- 2** Spustíte instalaci cílového systému z CD nebo DVD
- 3** V úvodní startovací obrazovce zadejte parametry potřebné pro SSH a nalezení instalačního zdroje. Podrobnosti o parametrech najdete v [1.4.3 – „Použití vlastních startovacích voleb“](#) (strana 40).

Cílový systém se spustí do textového režimu a zobrazí IP adresu, na kterou se může připojit SSH klient.

- 4** Na externím systému otevřete terminál a připojte se k cílovému systému jak je popsáno v [„Připojení k instalačnímu programu“](#) (strana 44).

- 5 Projděte instalaci jak je popsáno v části „*Instalace pomocí nástroje YaST*“ (↑Uživatelská příručka).

Abyste instalaci dokončili, musíte se po restartu znovu připojit k cílovému systému.

- 6 Dokončete instalaci.

## 1.1.6 Vzdálená instalace přes SSH — PXE Boot a Wake on LAN

Ke spuštění cílového systému a instalace není nutná fyzická přítomnost.

Ujistěte se, že jsou splněny následující požadavky:

- Vzdálený instalační zdroj: NFS, HTTP, FTP nebo SMB s funkčním síťovým připojením
- TFTP server
- Běžící DHCP server poskytující pevné IP
- Cílový systém s podporou PXE bootu, sítě a Wake on LAN, připojený do sítě
- Externí systém s funkčním připojením a nainstalovaným SSH klientem

Tento typ instalace provedete následujícím způsobem:

- 1 Nastavte instalační zdroj podle postupu v [1.2 – „Nastavení serveru s instalačním zdrojem“](#) (strana 25). Zvolte NFS, HTTP nebo FTP server. Informace o nastavení SMB najdete v části [1.2.5 – „SMB instalační zdroj“](#) (strana 32).
- 2 Nastavte startovací obraz na TFTP serveru. Postup je popsán v [1.3.2 – „Nastavení TFTP serveru“](#) (strana 35).
- 3 Nastavte DHCP server a do nastavení přidejte TFTP server. Postup je popsán v [1.3.1 – „Nastavení DHCP serveru“](#) (strana 34).
- 4 Nastavte na cílovém systému PXE boot. Postup je popsán v [1.3.3 – „PXE boot“](#) (strana 35).



- 5 Spustíte cílový systém přes Wake on LAN. Postup je popsán v [1.3.4 – „Ruční nastavení Wake on LAN“](#) (strana 38).
- 6 Na externím systému spustíte SSH a připojíte se k cílovému systému jak je popsáno v [1.5.2 – „SSH instalace“](#) (strana 44).
- 7 Projděte instalaci jak je popsáno v části *„Instalace pomocí nástroje YaST“* (↑Uživatelská příručka).

Abyste dokončili instalaci, musíte se po restartu znovu připojit k cílovému systému

- 8 Dokončete instalaci.

## 1.2 Nastavení serveru s instalačním zdrojem

Nastavení serveru se liší v závislosti na operačním systému, který běží na počítači užitém jako instalační zdroj. Nejsnadnější způsob je nastavení instalačního zdroje pomocí programu YaST v systému SUSE LINUX Enterprise Server 9 nebo SUSE Linux 9.3 či vyšším. Na starších verzích musíte zdroj nastavit ručně.

---

### Tip

Použít můžete také Microsoft Windows, viz [1.2.5 – „SMB instalační zdroj“](#) (strana 32).

---

### 1.2.1 Nastavení instalačního zdroje v systému YaST

YaST umožňuje nastavit instalační zdroj v přívětivém grafickém prostředí. Podporuje HTTP, FTP a NFS servery.

- 1 Na počítač, který má sloužit jako instalační server, se přihlaste jako uživatel `root`.
- 2 Zpusťte *YaST* → *Miscellaneous* → *Instalační server*.

### 3 Zvolte *Server Configuration*.

### 4 Vyberte typ serveru (HTTP, FTP nebo NFS).

Zvolená služba se bude automaticky spouštět při startu systému. Jestliže zvolená služba již běží a vy chcete provést nastavení ručně, deaktivujte automatické nastavení volbou *Do Not Configure Any Network Services*. V obou případech zadejte adresář pro instalační data.

### 5 Nastavte zvolený typ serveru.

Tento krok předpokládá automatické nastavení. Pokud automatické nastavení přeskočíte, je přeskočen. nastavte alias pro kořenový adresář FTP nebo HTTP serveru, kam se uloží instalační data. Instalační zdroj bude později dostupný v `ftp://Server-IP/Alias/Name` (FTP) nebo `http://Server-IP/Alias/Name` (HTTP). *Name* nahraďte názvem instalačního zdroje. V případě NFS zdroje zadejte zástupné znaky pro export souborů. NFS server bude dostupný pod `nfs://Server-IP/Name`. Podrobnosti o NFS a exportu najdete v kapitole [22 – „NFS — sdílené souborové systémy“](#) (strana 349).

### 6 Nastavte konfigurační zdroj

Před překopírováním instalačním médií zadejte název instalačního zdroje (nejvhodnější je zkratka produktu a verze). YaST umožňuje místo překopírování médií použít přímo jejich ISO obrazy. Pokud chcete použít ISO obrazy, zaškrtněte příslušnou volbu. Některé produkty mohou obsahovat další dodatečná CD, v takovém případě zvolte *Prompt for Additional CDs*. YaST vás požádá o média. Bay byl instalační server ohlášen přes OpenSLP, zaškrtněte volbu pro ohlašování přes SLP.

---

#### Tip

Pokud to vaše síť umožňuje, dejte přednost SLP oznamování. Ušetříte tím práci při zadávání nastavení instalačního serveru. Cílový systém jej automaticky při startu instalace vyhledá bez nutnosti ručního nastavení. Více informací najdete v [1.4 – „Spuštění instalace na cílovém systému“](#) (strana 38).

---

### 7 Nahrajte instalační data

Překopírování CD je časově nejnáročnější část vytvoření instalačního zdroje. Média zadávejte v pořadí požadovaném programem YaST. Po úspěšném překopírování ukončete konfiguraci kliknutím na tlačítko *Finish*.

Po provedení výše uvedených kroků je váš instalační server připraven na požadavky klientů a bude se automaticky spouštět při každém startu systému. Žádné další nastavení již není potřeba. Pokud jste zvolili ruční nastavení, nezapomeňte nastavit spuštění příslušné služby.

Instalační zdroj zrušíte v seznamu zdrojů jeho označením a kliknutím na *Change* a *Delete*. Tento postup data pouze deaktivuje, ale nesmaže data instalačního zdroje. Pokud chcete smazat i ty, proveďte smazání ručně.

Pokud chcete, aby server poskytoval zdroje pro různé produkty a verze, nastavte další instalační zdroje.

## 1.2.2 Ruční nastavení NFS instalačního zdroje

Nastavení NFS instalačního zdroje vyžaduje dva základní kroky. V prvním vytvoříte adresář a nainstalujete do něj instalační data, v druhém vyexportujete adresář přes NFS..

Adresář s daty vytvoříte následujícím způsobem:

**1** Přihlaste se jako uživatel `root`

**2** Vytvořte adresář pro instalační data např.:

```
mkdir install/product/productversion
```

```
cd install/product/productversion
```

Řetězec *product* nahraďte zkratkou jména produktu (např. SUSE Linux) a *productversion* verzí.

**3** Pro každé CD vykonajte:

**a** Překopírujte CD do adresáře:

```
cp -a /media/path_to_your_CD-ROM_drive .
```

Část `path_to_your_CD-ROM_drive` nahraďte aktuální CD nebo DVD. V závislosti na typu média může jít o `cdrom`, `cdrecorder`, `dvd` nebo `dvdrecorder`.

**b** Přejmenujte adresář podle pořadí CD:

```
mv path_to_your_CD-ROM_drive CDx
```

Písmeno *x* nahraďte číslem CD.

Instalační zdroj přes NFS pomocí programu YaST:

- 1 Přihlaste se jako `root`
- 2 Spustíte *YaST* → *Síťové služby* → *NFS server*.
- 3 Zvolte *Spustit NFS server a Open Port in Firewall* a klikněte na *Next*.
- 4 Zvolte *Přidat adresář* a zadejte cestu k adresáři s instalačními daty. V našem případě `/productversion`.
- 5 Zvolte *Add Host* a zadejte jména počítačům které budou mít přístup k datům. Mimo jmen můžete použít také zástupné znaky, rozsah IP adres nebo jen jméno své sítě. Zadejte volby exportu nebo je zanechte ve výchozím nastavení, které by mělo být dostačující pro většinu případů. Více informací o NFS najdete v manuálové stránce `exports`.
- 6 Klikněte na tlačítko *Finish*.

NFS server se bude automaticky spouštět při startu počítače.

Pokud dáváte přednost ručnímu nastavení místo nastavení pomocí YaST, postupujte následujícím způsobem:

- 1 Přihlaste se jako uživatel `root`
- 2 Otevřete soubor `/etc/exports` a vložte řádku:

```
/productversion *(ro,root_squash,sync)
```

Tím exportujete adresář `/productversion` pro všechny počítače ve vaší síti. Pokud chcete přístup omezit, nahraďte `*` maskou sítě nebo jmény počítačů.

Více informací o NFS najdete v manuálové stránce `exports`. Uložte soubor a zavřete konfigurační soubor.

- 3 Aby se NFS služba spouštěla automaticky při startu systému, zadejte příkaz:

```
insserv /etc/init.d/nfsserver
insserv /etc/init.d/portmap
```

- 4 NFS server spustíte příkazem:

```
rcnfsserver start
```

V případě pozdějších úprav nastavení restartujte NFS démona příkazem `rcnfsserver restart`.

Použití OpenSLP oznámí server všem klientům v síti.

- 1 Přihlaste se jako uživatel `root`
- 2 Přejděte do adresáře `/etc/slp.reg.d/`.
- 3 Vytvořte soubor `install.suse.nfs.reg` a vložte do něj:

```
# Register the NFS Installation Server
service:install.suse:nfs://$HOSTNAME/path_instsource/CD1,en,65535
description=NFS Installation Source
```

Řetězec `path_instsource` nahraďte aktuální cestou k adresáři s daty.

- 4 Uložte soubor a spusťte OPenSLP démona příkazem:

```
rcslpd start
```

Více informací o OpenSLP najdete v adresáři `/usr/share/doc/packages/openslp/` nebo v kapitole 19 – „SLP služby v síti“ (strana 319).

## 1.2.3 Ruční nastavení FTP instalačního zdroje

Vytvoření FTP instalačního zdroje je podobné konfiguraci NFS zdroje. Také FTP zdroj lze oznamovat přes OpenSLP .

**1** Vytvořte adresář jako bylo popsáno v [1.2.2 – „Ruční nastavení NFS instalačního zdroje“](#) (strana 27).

**2** Nastavte FTP server, aby distribuoval váš adresář:

**a** LPřihlaste se jako uživatel `root` a v správce balíčků programu YaST nainstalujte balíček `pure-ftpd` (FTP server) .

**b** Přejděte do kořenového adresáře FTP serveru:

```
cd/srv/ftp
```

**c** Vytvořte podadresář s instalačnímu zdroji:

```
mkdir instsource
```

Řetězec `instsource` nahraďte jménem produktu.

**d** Překopírujte obsah CD do kořenového adresáře FTP server (stejně jako v kroku in [1.2.2 – „Ruční nastavení NFS instalačního zdroje“](#) (strana 27), [Krok 3](#) (strana 27)).

Alternativně můžete do FTP serveru připojit již existující adresáře:

```
mount --bind path_to_instsource /srv/ftp/instsource
```

Řetězec `path_to_instsource` a `instsource` nahraďte hodnotami podle vašeho nastavení. Pokud potřebujete stálé připojení, vložte je do souboru `/etc/fstab`.

**e** Spust'ete `pure-ftpd`:

```
pure-ftpd &
```

**3** Oznamte službu přes OpenSLP:

**a** Vytvořte konfigurační soubor `install.suse.ftp.reg` v adresáři `/etc/slp/reg.d/`, s následujícím obsahem:

```
# Register the FTP Installation Server
service:install.suse:ftp://$HOSTNAME/srv/ftp/instsource/CD1,en,65535
description=FTP Installation Source
```

Řetězec *instsource* nahraďte jménem adresáře s instalačním zdrojem. Záznam by počínaje *service*: měl být vložen jako jedna řádka.

- b** Uložte konfiguraci a spusťte OpenSLP démona příkazem:

```
rcslpd start
```

## 1.2.4 Ruční nastavení HTTP instalačního serveru

Vytvoření HTTP instalačního zdroje je podobné jako v případě NFS. HTTP zdroj lze samozřejmě také oznámit přes OpenSLP.

- 1** Create a directory holding the installation sources as described in [1.2.2 – „Ruční nastavení NFS instalačního zdroje“](#) (strana 27).
- 2** Configure the HTTP server to distribute the contents of your installation directory:
  - a** Nainstalujte webový server Apache podle postupu uvedeného v kapitole [26.1 – „Instalace“](#) (strana 391).
  - b** Zadejte kořenový adresář HTTP serveru (*/srv/www/htdocs*) a vytvořte podadresář s instalačními zdroji příkazem:

```
mkdir instsource
```

Řetězec *instsource* nahraďte jménem produktu.

- c** Vytvořte symbolický odkaz z umístění zdroje do kořenového adresáře webového serveru (*/srv/www/htdocs*):

```
ln -s /path_instsource /srv/www/htdocs/instsource
```

- d** Změňte konfigurační soubor HTTP serveru (*/etc/apache2/default-server.conf*) tak, aby následoval odkaz. K tomu nahraďte řádku:

```
Options None
```

```
na
```

Options Indexes FollowSymLinks

- e Znovu zaveďte konfiguraci HTTP serveru příkazem `rcapache2 reload`.

### 3 Oznamte službu přes OpenSLP:

- a Vytvořte konfigurační soubor `install.suse.http.reg` v adresáři `/etc/slp/reg.d/` a vložte do něj:

```
# Register the HTTP Installation Server
service:install.suse:http://$HOSTNAME/srv/www/htdocs/instsource/CD1/,en,65535
description=HTTP Installation Source
```

Řetězec `path_to_instsource` nahraďte cestou k instalačnímu zdroji. Obsah od části `service:` by měl být jedna souvislá řádka.

- b Uložte konfigurační soubor a restartujte OpenSLP démona příkazem `rcslpd restart`.

## 1.2.5 SMB instalační zdroj

Pomocí Samby (SMB) můžete exportovat instalační zdroj ze serverů Microsoft Windows.

Sdílený adresář Windows pro systém SUSE Linux nastavíte následujícím způsobem:

- 1 Přihlaste se do systému Windows
- 2 Spustě správce souborů a vytvořte adresář, který bude obsahovat instalační zdroj, např. `INSTALL`
- 3 Vyexportujte sdílený adresář. Postup najdete v dokumentaci systému Windows
- 4 Vytvořte podadresář s názvem `product`. Řetězec `product` nahraďte jménem svého produktu (v našem případě SUSE Linux).
- 5 Každé instalační CD systému SUSE Linux překopírujte do vlastního adresáře, tj. `CD1`, `CD2`, `CD3` atd.



**6** Přejděte do hlavního sdíleného adresáře (INSTALL) a překopírujte do něj následující soubory a adresáře z *product/CD1*: *content*, *media.1*, *control.xml* a *boot*.

**7** V INSTALL vytvořte adresář *yast*.

V adresáři *yast* vytvořte soubory *order* a *instorder*.

**8** DO souboru *order* vložte řádku:

```
/NLD/CD1 smb://user:password@hostname/productCD1
```

Řetězec *user* nahraďte jménem, které používáte pro přihlášení ke sdílené složce. *password* nahraďte heslem pro přihlášení ke svazku. *hostname* nahraďte síťovým jménem systému Windows.

**9** Do souboru *instorder* vložte řádku:

```
/product/CD1
```

SMB zdroj použijte pro instalaci následujícím způsobem:

**1** Spust'ete cílový systém

**2** Ze startovací nabídky zvolte *Installation*.

**3** Stiskněte **F3** a **F4**.

**4** Zvolte SMB, zadajte jméno nebo IP systému Windows s instalačním zdrojem, jméno sdíleného adresáře (INSTALL), uživatelské jméno a heslo.

Stiskněte klávesu **Enter**, tím spustíte YaST a instalační proces.

## 1.3 Příprava startu cílového systému

V této části najdete postupy potřebné pro nastavení spuštění systému, jako je nastavení DHCP, PXE bootu, TFTP a Wake on LAN (WOL).

## 1.3.1 Nastavení DHCP serveru

Nastavení DHCP serveru v systému SUSE Linux provedete ručně tak, aby DHCP server poskytoval data potřebná pro TFTP, PXE a WOL.

### Ruční nastavení DHCP serveru

Mimo IP adresy bude DHCP server přidělovat také informace o umístění TFTP serveru.

**1** Přihlaste se jako `root` k systému, který bude sloužit jako DHCP server.

**2** Do souboru `/etc/dhcpd.conf` přidejte následující část:

```
group {
    # PXE related stuff
    #
    # "next server" defines the tftp server that will be used
    next server ip_tftp_server:
    #
    # "filename" specifies the pxelinux image on the tftp server
    # the server runs in chroot under /srv/tftpboot
    filename "pxelinux.0";
}
```

Řetězec `ip_of_the_tftp_server` nahraďte IP adresou TFTP serveru.

Podrobnosti o volbách v souboru `dhcpd.conf` najdete v manuálové stránce `dhcpd.conf`.

**3** Restartujte DHCP server příkazem `rcdhcpd restart`.

Jestliže chcete použít pro instalaci SSH a pro cílový systém PXE a WOL, přímo nastavte v souboru adresu pro cílový systém. Dosáhnete toho následující konfigurací:

```
group {
    # PXE related stuff
    #
    # "next server" defines the tftp server that will be used
    next server ip_tftp_server:
    #
    # "filename" specifies the pxelinux image on the tftp server
    # the server runs in chroot under /srv/tftpboot
    filename "pxelinux.0";
    host test { hardware ethernet mac_adresa;
                fixed-address ip_adresa; }
}
```

Aby se adresa spojila se správným počítačem, musíte k adrese přiřadit MAC adresu síťového rozhraní cílového systému, které je použito pro připojení do sítě. Řetězce *mac\_adresa* a *ip\_adresa* nahradíte hodnotami vašeho cílového systému.

Po restartu DHCP serveru můžete k cílovému systému přistupovat přes SSH.

## 1.3.2 Nastavení TFTP serveru

TFTP server nastavte pomocí programu YaST. TFTP server slouží k poskytování startovacích obrazů pro automatický start systému.

### Nastavení TFTP serveru pomocí programu YaST

- 1 Přihlaste se jako uživatel `root`
- 2 Zvolte *YaST* → *Síťové služby* → *TFTP Server* a doinstalujte požadované balíčky.
- 3 Povolte server. U starších systémů se o start postará automaticky `xinetd`. Od verze systému SUSE Linux 10.1 je nutné službu povolit v Editoru úrovní běhu.
- 4 Přístusnou volbou otevřete port pro server na firewallu.
- 5 Nastavte adresář se startovacím obrazem. Výchozí adresář `/tftpboot` je vytvořen a zvolen automaticky.
- 6 Ukončete nastavení serveru.

## 1.3.3 PXE boot

Technické pozadí a specifikaci PXE jsou k dispozici v Preboot Execution Environment (PXE) Specification (<ftp://download.intel.com/labs/manage/wfm/download/pxespec.pdf>).

- 1 Překopírujte `linux`, `initrd`, `message` a `memtest` z instalačního média do adresáře `/srv/tftpboot` příkazem:

```
cp -a boot/loader/linux boot/loader/initrd  
boot/loader/message boot/loader/memtest /srv/tftpboot
```

## 2 Nainstalujte balíček `syslinux`

### 3 Překopírujte soubor `/usr/share/syslinux/pxelinux.0` do adresáře `/srv/tftpboot` příkazem:

```
cp -a /usr/share/syslinux/pxelinux.0 /srv/tftpboot
```

### 4 Z instalačního média překopírujte `isolinux.cfg` do `/srv/tftpboot/pxelinux.cfg/default` příkazem:

```
cp -a boot/loader/isolinux.cfg /srv/tftpboot/pxelinux.cfg/default
```

### 5 Upravte soubor `/srv/tftpboot/pxelinux.cfg/default` tak, aby neobsahoval řádky začínající na `gfxboot`, `readinfo` a `framebuffer`.

### 6 K apendn řádkám položek `failsafe` a `apic` přidejte:

```
insmod=e100
```

Tím dosáhnete automatického zavedení modulu síťové karty Intel 100MBit/s na PXE klientech. Pokud potřebujete zavést jinou kartu, změňte jméno ovladače, např. `entry depends on the client's hardware and must be adapted` pro Broadcom GigaBit bude parametr vypadat takto: `insmod=bcm5700`.

```
netdevice=eth0
```

Touto položkou definujete síťové rozhraní pro instalaci. Tuto položku nemusíte zadávat, pokud máte jen jednu síťovou kartu.

```
install=nfs://ip_instserver/path_instsource/CD1
```

Nastavení NFS serveru s instalačním zdrojem. Řetězec

`ip_instserver` nahradíte IP adresou svého instalačního serveru.

`path_instsource` nahradíte adresářem instalačního zdroje na serveru.

HTTP, FTP a SMB se nastavují podobně, pouze změníte označení pro protokol `http`, `ftp` nebo `smb`.

---

### Důležité

Pokud potřebujete další parametry např. pro VNC nebo SSH instalaci, přidejte je k položce `install` Příklad je uveden v [1.4 – „Spuštění instalace na cílovém systému“](#) (strana 38).

---

Následuje příklad `/srv/tftpboot/pxelinux.cfg/default`. Nezapomeňte nastavit parametry pro VNC nebo SSH instalaci pomocí `vnc` a `vncpassword` nebo `ssh` a `sshpassword` v řádce `install`. Řádky ukončené `\` zadejte bez tohoto znaku jako jednu celou řádku spojené s následující řádkou.

```
default linux

# default
label linux
    kernel linux
        append initrd=initrd ramdisk_size=65536 insmod=e100 \
            install=nfs://ip_instserver/path_instsource/product

# failsafe
label failsafe
    kernel linux
        append initrd=initrd ramdisk_size=65536 ide=nodma apm=off acpi=off \
            insmod=e100 install=nfs://ip_instserver/path_instsource/product

# apic
label apic
    kernel linux
        append initrd=initrd ramdisk_size=65536 apic insmod=e100 \
            install=nfs://ip_instserver/path_instsource/product

# manual
label manual
    kernel linux
        append initrd=initrd ramdisk_size=65536 manual=1

# rescue
label rescue
    kernel linux
        append initrd=initrd ramdisk_size=65536 rescue=1

# memory test
label memtest
    kernel memtest

# hard disk
label harddisk
    kernel
        linux append SLX=0x202

implicit      0
display       message
prompt        1
timeout       100
```

Řetězce `ip_instserver` a `path_instsource` přizpůsobte svému nastavení.

V následující sekci najdete základní informace o PXELINUX volbách. Další informace najdete v dokumentaci balíčku `syslinux` v adresáři `/usr/share/doc/packages/syslinux/`.

## 1.3.4 Ruční nastavení Wake on LAN

- 1 Přihlaste se jako uživatel `root`
- 2 Pomocí programu YaST nainstalujte balíček `netdiag`.
- 3 V terminálu jako uživatel `root` zadejte příkaz:

```
ether-wakemac_of_target
```

Řetězec `mac_of_target` nahrad'te aktuální MAC adresou síťového rozhraní cílového systému.

## 1.4 Spuštění instalace na cílovém systému

Spuštění instalace můžete provést dvěma způsoby. Buď použijete standardní startovací volby na startovací obrazovce nebo použijete pro zadání voleb startovací prompt.

### 1.4.1 Použití výchozích startovacích voleb

Startovací volby jsou posány v „*Instalace pomocí nástroje YaST*“ (↑Uživatelská příručka).

Volbou nabídky *Installation* spustíte instalaci. V případě problémů zvolte *Installation—ACPI Disabled* nebo *Installation—Safe Settings*.

Více informací najdete v části „Problémy při instalaci“ (9 – „*Problémy a jejich řešení*“, ↑Uživatelská příručka).

## 1.4.2 Používání funkčních kláves

Úvodní obrazovka nabízí řadu nastavení parametrů startu dostupných přes funkční klávesy. Umožňují nastavení bez nutnosti znalosti přesné syntaxe parametrů (viz [1.4.3 – „Použití vlastních startovacích voleb“](#) (strana 40)).

Kompletní seznam je uveden v tabulce níže.

**Tabulka 1.1** *Funkční klávesy pro instalaci*

Klávesa	Účel	Dostupné volby	Výchozí hodnota
F1	Nápověda	-	-
F2	Volba jazyka instalace	Všechny podporované jazyky	Angličtina
F3	Změna rozlišení instalace	<ul style="list-style-type: none"><li>• Textový režim</li><li>• VESA</li><li>• rozlišení #1</li><li>• rozlišení #2</li><li>• ...</li></ul>	<ul style="list-style-type: none"><li>• Výchozí hodnota závisí na grafickém systému</li></ul>
F4	Zvolte instalační zdroj	<ul style="list-style-type: none"><li>• CD-ROM/DVD</li><li>• SLP</li><li>• FTP</li><li>• HTTP</li><li>• NFS</li><li>• SMB</li></ul>	CD-ROM/DVD

Klávesa	Účel	Dostupné volby	Výchozí hodnota
		• Pevný disk	
<b>F5</b>	Apply driver update disk	Ovladač	-

## 1.4.3 Použití vlastních startovacích voleb

řadu volbe můžete nastavit později pomocí linuxrc, ale použití startovacích parametrů je nejjednodušší cesta. V případě některých automatických nastavení lze volby předat přes soubory `initrd` nebo `info`.

### Tip

Více informací o volbách linuxrc najdete v souboru `/usr/share/doc/packages/linuxrc/linuxrc.html`.

Následující seznam obsahuje scénáře zde uvedených typů instalací a parametry potřebné pro jejich spuštění. Tyto parametry je potřeba pro správný průběh zvolené instalace předat při startu systému.

**Tabulka 1.2** *Instalační scénáře a jejich parametry*

Scénář instalace	Parametry potřebné pro start	Startovací parametry
„Instalace pomocí nástroje YaST“ (↑Uživatelská příručka)	Nic: systém startuje automaticky	Nepotřebné
<a href="#">1.1.1 – „Jednoduchá vzdálené instalace přes VNC — statická síť“</a> (strana 18)	<ul style="list-style-type: none"> <li>• Umístění instalačního serveru</li> <li>• Síťové zařízení</li> <li>• IP adresa</li> </ul>	<ul style="list-style-type: none"> <li>• <code>install=(nfs,http,ftp,smb)::/cesta_kintsmediu</code></li> </ul>



Scénář instalace	Parametry potřebné pro start	Startovací parametry
	<ul style="list-style-type: none"> <li>• Maska sítě</li> <li>• Brána</li> <li>• Povolení VNC</li> <li>• VNC heslo</li> </ul>	<ul style="list-style-type: none"> <li>• <code>netdevice=sitove_zarizeni</code> (pouze pokud je k dispozici více síťových zařízení)</li> <li>• <code>hostip=ip_adresa</code></li> <li>• <code>netmask=maska_site</code></li> <li>• <code>gateway=ip_gateway</code></li> <li>• <code>vnc=1</code></li> <li>• <code>vncpassword=nejake_heslo</code></li> </ul>
<p>1.1.2 – „Jednoduchá vzdálená instalace přes VNC — dynamické nastavení sítě přes DHCP“ (strana 19)</p>	<ul style="list-style-type: none"> <li>• Umístění instalačního serveru</li> <li>• Povolení VNC</li> <li>• VNC heslo</li> </ul>	<ul style="list-style-type: none"> <li>• <code>install=(nfs,http,ftp,smb)::/cesta_kintsmediu</code></li> <li>• <code>vnc=1</code></li> <li>• <code>vncpassword=nejake_heslo</code></li> </ul>
<p>1.1.3 – „Vzdálená instalace přes VNC — PXE Boot a Wake on LAN“ (strana 20)</p>	<ul style="list-style-type: none"> <li>• Umístění instalačního serveru</li> <li>• Umístění TFTP serveru</li> <li>• Povolení VNC</li> <li>• VNC heslo</li> </ul>	<p>Neaplikováno, postup přes PXE a DHCP</p>
<p>1.1.4 – „Jednoduchá vzdálená instalace přes SSH — statická síť“ (strana 21)</p>	<ul style="list-style-type: none"> <li>• Umístění instalačního serveru</li> <li>• Síťové zařízení</li> <li>• IP adresa</li> <li>• Maska sítě</li> <li>• Brána</li> <li>• Povolení SSH</li> <li>• SSH heslo</li> </ul>	<ul style="list-style-type: none"> <li>• <code>install=(nfs,http,ftp,smb)::/cesta_kintsmediu</code></li> <li>• <code>netdevice=sitove_zarizeni</code> (pouze v případě přítomnosti několika síťových rozhraní)</li> <li>• <code>hostip=ip_adresa</code></li> </ul>

Scénář instalace	Parametry potřebné pro start	Startovací parametry
		<ul style="list-style-type: none"> <li>• <code>netmask=maska_site</code></li> <li>• <code>gateway=ip_gateway</code></li> <li>• <code>usessh=1</code></li> <li>• <code>sshpassword=nejake_heslo</code></li> </ul>
<a href="#">1.1.5 – „Jednoduchá vzdálená instalace přes SSH — danymické nastavení sítě přes DHCP“</a> (strana 23)	<ul style="list-style-type: none"> <li>• Umístění instalačního serveru</li> <li>• Povolení SSH</li> <li>• SSH heslo</li> </ul>	<ul style="list-style-type: none"> <li>• <code>install=(nfs,http,ftp,smb):://cesta_kintsmediu</code></li> <li>• <code>usessh=1</code></li> <li>• <code>sshpassword=nejake_heslo</code></li> </ul>
<a href="#">1.1.6 – „Vzdálená instalace přes SSH — PXE Boot a Wake on LAN“</a> (strana 24)	<ul style="list-style-type: none"> <li>• Umístění instalačního serveru</li> <li>• Umístění TFTP serveru</li> <li>• Povolení SSH</li> <li>• SSH heslo</li> </ul>	Neaplikováno, postup přes PXE a DHCP

## 1.5 Instalační proces

Postup zobrazení instalačního procesu je závislý na nastavení parametrů startu. Sledovat můžete VNC nebo SSH instalaci..

### 1.5.1 VNC instalace

Pomocí VNC prohlížeče můžete nainstalovat systém SUSE Linux prakticky z libovolného operačního systému. V této části najdete návod pro VNC prohlížeč a webový prohlížeč.

## Příprava VNC instalace

Vše co potřebujete pro VNC instalaci je zadání startovacích parametrů při startu systému (viz 1.4.3 – „Použití vlastních startovacích voleb“ (strana 40)). Cílový systém se spustí do textového prostředí a bude čekat, dokud se k instalačnímu programu nepřipojí VNC klient.

Instalační program v textovém režimu vypíše IP adresu a display pro VNC instalaci. Zadejte data do VNC prohlížeče. Prohlížeč vás také požádá o VNC heslo.

Protože se instalační program oznamuje přes OpenSLP, můžete potřebné informace získat na libovolné stanici v síti také ze SLP prohlížeče:

- 1 Spustíte prohlížeč Konqueror.
- 2 Do pole adresy zadejte `service://yast.installation.suse`.

Cílový systém se objeví v seznamu nalezených služeb v okně prohlížeče. KDE VNC prohlížeč spustíte kliknutím na ikonu systému. Alternativně můžete nalezené údaje zadat do VNC prohlížeče.

## Připojení k instalačnímu programu

V zásadě máte dvě možnosti, jak se připojit k VNC serveru (v našem případě cílovému systému). Buď použijete VNC prohlížeč nebo se připojíte pomocí webového prohlížeče s podporou Javy.

V linuxovém systému můžete použít VNC prohlížeč `tightvncPort` aplikace TightVNC je k dispozici i pro systém Windows (<http://www.tightvnc.com/download.html>).

K instalačnímu programu se připojíte následujícím způsobem:

- 1 Spustíte VNC prohlížeč.
- 2 Zadejte IP a display získané ze SLP prohlížeče:

```
ip_address:display_number
```

Otevře se okno, které bude vypadat jako normální lokální instalace pomocí programu YaST.

Webový prohlížeč dělá instalaci zcela nezávislou na klientském softwaru a operačním systému. Použít můžete jakýkoliv prohlížeč s podporou Javy (Firefox, Internet Explorer, Konqueror, Opera atd.) .

VNC instalaci provedete následujícím způsobem:

**1** Spustíte webový prohlížeč

**2** Jako adresu zadejte:

```
http://ip_ciloveho_systemu:5801
```

**3** Zadejte VNC heslo. V okně prohlížeče se zobrazí standardní instalace pomocí programu YaST.

## 1.5.2 SSH instalace

Pomocí SSH můžete provádět vzdálenou instalaci z jakékoliv stanice s nainstalovaným SSH klientem.

### Příprava na SSH instalaci

Podle klientského programu (OpenSSH pro Linux nebo PuTTY pro Windows) nastavte příslušné startovací parametry na cílovém systému. Podrobnosti najdete v [1.4.3 – „Použití vlastních startovacích voleb“](#) (strana 40). OpenSSH je součástí standardní instalace systému SUSE Linux.

### Připojení k instalačnímu programu

**1** Získejte IP adresu cílového systému.

**2** V příkazové řádce zadejte příkaz:

```
ssh -X root@ip_ciloveho_systemu
```

Řetězec `ip_ciloveho_systemu` nahraďte IP adresou cílového systému.

**3** Při dotazu na uživatelské jméno zadejte `root` .

**4** Při výzvě k zadání hesla zadejte heslo uživatele `root`.

Po úspěšném přihlášení se objeví prompt pro instalaci.

**5** Instalační program spustíte příkazem `yast`.

Okno zobrazí standardní okno programu YaST jak je popsáno v části „*Instalace pomocí nástroje YaST*“ (↑Uživatelská příručka).



# Rozdělení disku pro experty

Specializované systémy často vyžadují zvláštní rozdělení disku. Abyte uchovali trvalá jména SCSI zařízení, musíte použít zvláštní skripty nebo provést nastavení udev. LVM (Logical Volume Management) je schéma rozdělení disku navržené tak, aby umožňovalo maximální flexibilitu. Díky možnosti vytváření obrazů umožňuje jednoduché a rychlé zálohování. RAID (Redundant Array of Independent Disks) umožňuje zvýšit bezpečnost dat, jejich integritu a výkon úložného systému.

## 2.1 Konfigurace LVM

Tato část krátce popisuje základy použití LVM a jeho nejdůležitější vlastnosti využitelné za mnoha různých okolností. V části [2.1.2 – „Konfigurace LVM pomocí nástroje YaST“](#) (strana 49) se dozvíte, jak nakonfigurovat LVM pomocí nástroje YaST.

---

### Varování

Použití LVM může představovat zvýšené riziko, např. ztráty dat. Může též zvýšit riziko pádu aplikací a dalších chybových stavů. Před nasazením LVM nebo změnou konfigurace svazků zazálohujte všechna důležitá data. Nikdy nepracujte bez zálohy.

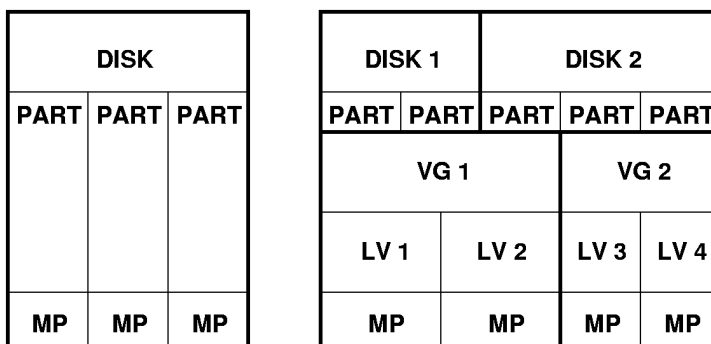
---

### 2.1.1 Správce logických svazků

Správce logických svazků (LVM) umožňuje flexibilní využití místa na pevných discích. LVM byl vyvinut, protože je často potřeba změnit rozdělení místa na pevných discích

až po instalaci systému, v době, kdy již byly vytvořeny diskové oddíly. Protože je obtížné měnit diskové oddíly na běžícím systému, poskytuje LVM tzv. *virtuální skupinu svazků* (VG, volume group), ze které se podle potřeby vyčleňují *logické svazky* (LV, logical volumes). Operační systém přistupuje k logickým svazkům místo fyzických oddílů. Virtuální skupiny svazků mohou být tvořeny více než jedním diskem, mohou se skládat z mnoha disků nebo jejich částí. LVM tak abstrahuje od fyzické podstaty disků a umožňuje tak flexibilnější a snadnější změny segmentace oproti fyzickému přerozdělování disků. Více informací o fyzickém rozdělování disků najdete v částech „Typy oddílů“ (1 – „*Instalace pomocí nástroje YaST*“, ↑Uživatelská příručka) a „Dělení disku“ (2 – „*Konfigurace pomocí YaST*“, ↑Uživatelská příručka).

**Obrázek 2.1** *Fyzické oddíly versus LVM*



Obrázek 2.1 – „*Fyzické oddíly versus LVM*“ (strana 48) srovnává fyzické rozdělování (vlevo) s použitím LVM (vpravo). Vlevo byl jeden fyzický disk rozdělen na tři fyzické oddíly (PART). Všechny tyto oddíly mají přiřazen bod připojení (MP), takže k nim může operační systém přistupovat. Vpravo byly dva disky rozděleny na dva a tři fyzické oddíly. Byly definovány dvě skupiny svazků (VG 1 a VG 2). VG 1 obsahuje dva oddíly z prvního disku a jeden z druhého disku. VG 2 obsahuje zbývající dva oddíly z druhého disku. V rámci LVM se fyzické oddíly zařazené do skupiny svazků nazývají *fyzické svazky* (PV). Ve skupinách svazků byly definovány čtyři logické svazky (LV 1 až LV 4), ke kterým může operační systém přistupovat pomocí asociovaných bodů připojení. Hranice mezi logickými svazky nemusí odpovídat hranici žádného fyzického svazku. Podívejte se například na hranici mezi LV 1 and LV 2 v našem příkladě.

Vlastnosti LVM:

- Několik pevných disků nebo oddílů lze sloučit do jednoho velkého logického svazku (LV).



- Nastane-li nedostatek volného místa v logickém svazku (např. `/usr`), lze ho při vhodné konfiguraci bez problémů rozšířit.
- Pomocí LVM lze dokonce přidat pevné disky nebo logické svazky za běhu systému. Podmínkou je ovšem hardware podporující tzv. hot swap.
- Logický svazek lze pomocí "stripping" režimu rozdělit mezi několik fyzických svazků. Pokud jsou tyto fyzické svazky na různých discích, lze dosáhnout zvýšení výkonu podobně jako v případě RAID 0.
- Funkce snapshot umožňuje vytvoření konzistentní zálohy běžících systémů (zejména serverů).

LVM se vyplatí používat na intenzivně využívaných domácích počítačích nebo malých serverech. Pokud máte rychle se rozšiřující množství dat, např. databáze či MP3 archívy, je LVM ideálním řešením. Umožňuje použití souborových systémů větších, než je velikost pevného disku. Další výhodou je skutečnost, že lze použít až 256 logických svazků. Mějte však na paměti, že se práce s LVM velmi liší od práce s běžnými oddíly. Instrukce a další informace o použití LVM jsou dostupné v oficiálním LVM HOWTO dokumentu na adrese <http://tldp.org/HOWTO/LVM-HOWTO/>.

V systému s jádrem 2.6 je možno používat LVM verze 2, který je zpětně kompatibilní s předcházející verzí a umožňuje správu dříve vytvořených logických svazků. Při vytváření nových svazků je však třeba rozhodnout, zda použít nový nebo starší, zpětně kompatibilní, formát. LVM verze 2 nevyžaduje žádné jaderné záplaty. Využívá mapovač zařízení (device mapper) integrovaný v jádře 2.6. Tato verze jádra podporuje pouze LVM verze 2. Proto, kdykoliv budeme mluvit o LVM, máme na mysli LVM verze 2.

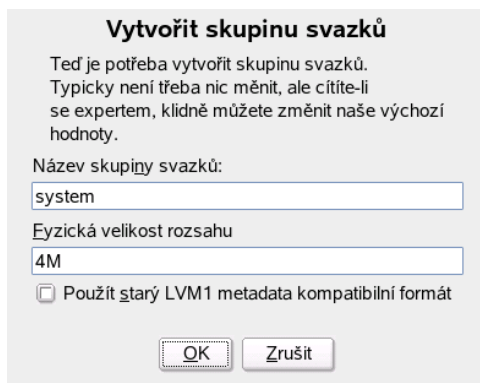
## 2.1.2 Konfigurace LVM pomocí nástroje YaST

YaST modul pro konfiguraci LVM je dostupný z expertního YaST modulu pro rozdělování disků (viz „Dělení disku“ (2 – „Konfigurace pomocí YaST“, ↑ Uživatelská příručka)). Tento profesionální nástroj pro rozdělování disku umožňuje vytvářet, upravovat a mazat oddíly pro použití v rámci LVM. Oddíl pro LVM v něm vytvoříte kliknutím na *Vytvořit* → *Neformátovat* a výběrem ID `0x8E Linux LVM`. Jakmile máte vytvořeny všechny potřebné oddíly pro LVM, klikněte na *LVM...* Tím se zahájí konfigurace LVM.

## Vytváření skupin svazků

Pokud na systému ještě neexistuje žádná skupina svazků, budete požádáni o její vytvoření (viz 2.2 – „Vytváření skupiny svazků“ (strana 50)). Další skupiny můžete vytvořit pomocí *Přidat skupinu*, obvykle ale stačí jedna. Pro skupinu svazků v systému SUSE Linux se doporučuje použít název `system`. *Fyzická velikost rozsahu* určuje velikost fyzického bloku ve skupině svazků. Veškerý prostor v rámci skupiny svazků se obhospodařuje v takto velkých blocích. Výchozí hodnota je 4 MB, což umožňuje fyzické a logické svazky o maximální velikosti 256 GB. Pokud potřebujete logické svazky větší, zvyšte hodnotu rozsahu na 8, 16 nebo 32 MB.

**Obrázek 2.2** Vytváření skupiny svazků



**Vytvořit skupinu svazků**

Teď je potřeba vytvořit skupinu svazků.  
Typicky není třeba nic měnit, ale cítili-li se expertem, klidně můžete změnit naše výchozí hodnoty.

Název skupiny svazků:

Fyzická velikost rozsahu

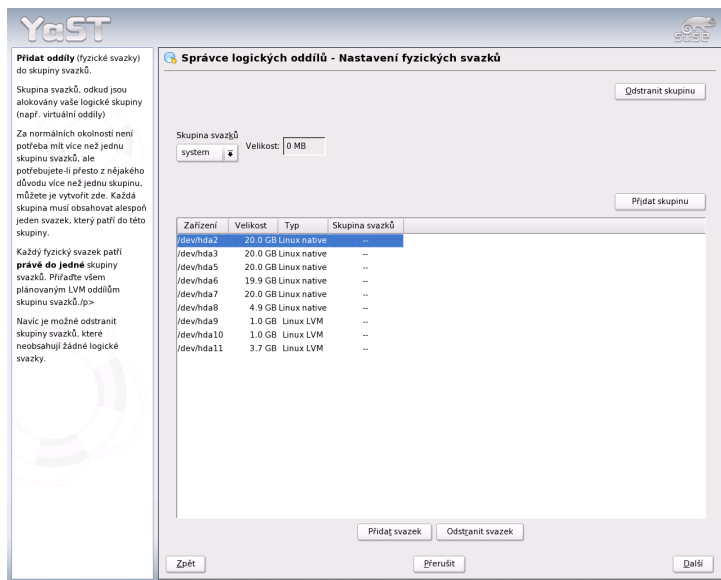
Použít starý LVM1 metadata kompatibilní formát

## Konfigurace fyzických svazků

Jakmile je vytvořena skupina svazků, objeví se seznam všech oddílů typu `Linux LVM` nebo `Linux native`. Nejsou zobrazeny odkládací ani DOS oddíly. Pokud je oddíl již zařazen do skupiny svazků, je v seznamu uvedeno její jméno. Nezařazené oddíly jsou označeny pomocí `--`.

Pokud je skupin svazků více, nastavte patřičnou skupinu v nabídce vlevo nahoře. Tlačítka vpravo nahoře umožňují vytvářet a mazat skupiny svazků. Smazat můžete pouze skupiny bez přiřazených oddílů. Oddíly zařazené do skupiny svazků se nazývají fyzické svazky (PV).

## Obrázek 2.3 Nastavení fyzického svazku

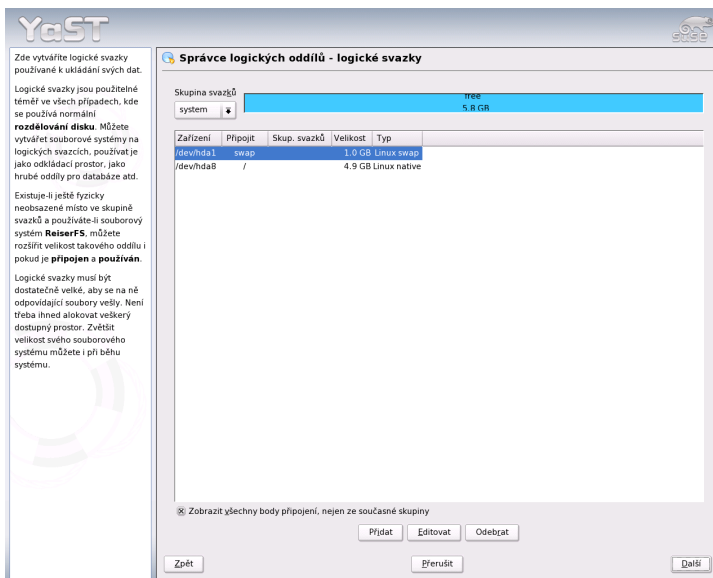


Dosud nezařazený oddíl do zvolené skupiny svazků přidáte jednoduše. Nejprve klikněte na oddíl v seznamu a pak na tlačítko *Přidat svazek*. V seznamu se v položce oddílu objeví název skupiny svazků. Zařaďte do skupiny všechny oddíly určené pro LVM. Jinak by místo na oddílu zůstalo nevyužito. Před opuštěním dialogu musíte přiřadit každé skupině svazků alespoň jeden fyzický svazek. Po přiřazení všech svazků klikněte na *Další*.

## Konfigurace logických svazků

Po naplnění skupiny svazků fyzickými svazky je třeba definovat logické svazky, které bude operační systém používat. V nabídce vlevo nahoře zvolte patřičnou skupinu svazků. Hned vedle je zobrazeno volné místo v aktuální skupině svazků. Seznam níže obsahuje všechny logické svazky v aktuální skupině. Jsou zde zobrazeny všechny běžné linuxové oddíly s bodem připojení, všechny odkládací oddíly a všechny existující logické svazky. Logické svazky můžete dle libosti přidávat, upravovat a odebírat pomocí tlačítek pod seznamem. V každé skupině svazků vytvořte alespoň jeden logický svazek.

**Obrázek 2.4** Správa logických svazků



Nový logický svazek vytvoříte kliknutím na *Přidat*. Otevře se dialog, ve kterém musíte zadat potřebné údaje. Zadejte velikost, typ souborového systému a bod připojení. Na logickém svazku se vytvoří souborový systém, např. ext2 nebo reiserfs, kterému je přidělen bod připojení. Soubory uložené na tomto logickém svazku jsou pak v systému dostupné v odpovídajícím adresáři. Je také možné rozmístit data v logickém svazku na několik fyzických svazků (striping). Pokud jsou tyto fyzické svazky umístěny na různých pevných discích, zvýší se výkon při čtení i zápisu (podobně jako u RAID 0). Striping LV s  $n$  "proužky" (stripes) lze ovšem správně vytvořit jen tehdy, pokud lze požadované místo rovnoměrně rozdělit mezi  $n$  fyzických svazků. Pokud jsou například k dispozici jen dva fyzické svazky, nelze vytvořit logický svazek se třemi "proužky".

---

### Varování: Striping

YaST v této fázi nemůže ověřit správnost vašeho nastavení ohledně stripingu. Chyby se projeví až později, když je LVM implementován na disk.

---

**Obrázek 2.5** Vytváření logických svazků

**Vytvořit logický svazek**

Název logického svazku  
[ ]  
(např. 'var', 'opt')

Velikost (např. 4.0 GB 210.0 MB)  
1.4 GB  
max. = 5.8 GB [max]

Stripes  
1

Velikost proužku  
64

Volby fstab

Bod připojení  
/home

Formátovat  
 Neformátovat  
 Formátovat  
Soub. systém  
Reiser  
[Volby]  
 Krypt. souborový systém

[OK] [Zrušit]

Pokud jste na systému LVM již nakonfigurovali, můžete zadat existující logické svazky. Před pokračováním jim přiřadíte body připojení. Kliknutím na *Další* se vrátíte do dialogu YaST *Rozdělování disku pro experty*.

## Přímá správa LVM

Pokud máte již LVM nakonfigurováno a jen chcete něco změnit, existuje jiná cesta. V nástroji YaST zvolte *System* → *LVM*. Otevře se dialog, který umožňuje veškeré nastavení zmíněné výše s výjimkou fyzického přerozdělování disku. Zobrazuje dva seznamy, jeden s existujícími fyzickými svazky, druhý s logickými svazky. Pracovat s nimi můžete dříve výše popsanými postupy.

## 2.2 Konfigurace softwarového RAIDu

Smyslem polí RAID (redundant array of inexpensive disks – pole nepříliš drahých disků s možností redundance) je zkombinovat více diskových oddílů do jednoho velkého virtuálního pevného disku s vyšším výkonem a lepším zabezpečením dat. Jedna z

těchto výhod je však při použití RAIDu uplatněna na úkor druhé. Většina řadičů RAID používá protokol SCSI, ten totiž umí adresovat velké množství disků efektivněji než řadiče IDE a je vhodnější pro paralelní zpracování příkazů. Nicméně existují i RAID řadiče podporující IDE nebo SATA disky. Více informací viz databázi hardwaru na adrese <http://cdb.suse.de>.

Podobné úkoly jako poměrně nákladný hardwarový RAID řadič dokáže plnit i RAID softwarový. SUSE Linux, s pomocí konfiguračního nástroje YaST, nabízí možnost spojit několik pevných disků do jednoho softwarového RAID pole – velmi výhodné alternativy k hardwarovému RAIDu. RAID umožňuje aplikovat různé strategie kombinace disků do RAID pole, každá má jiný cíl, výhody a charakteristiky. Tyto varianty jsou známé jako tzv. typ RAIDu (*RAID level*).

## 2.2.1 Běžné typy polí RAID

### RAID 0

Tento typ pole zlepšuje výkon při přístupu k datům rozložením datových bloků každého souboru na více pevných disků. Ve skutečnosti se nejedná o RAID v pravém slova smyslu, neboť neprobíhá žádné zabezpečování dat, nicméně se termín *RAID 0* pro tento režim ujal. RAID 0, spojuje dva nebo více pevných disků v jeden virtuální disk. Výkon je velmi vysoký, ale výpadek jediného disku znamená selhání celého pole a ztrátu dat.

### RAID 1

Tento typ pole poskytuje přiměřený stupeň ochrany dat, protože jsou kopírována na další disk v poměru 1:1. Metoda je též známá pod názvem *zrcadlení disku*. Pokud je některý disk zničen, kopie jeho obsahu je stále přístupná na dalším disku. Všechny disky kromě jednoho mohou být zničeny, aniž by byla data ohrožena. Výkon při zápisu dat je ve srovnání se samostatným pevným diskem kvůli kopírování dat o 10-20% nižší, ale čtení je naopak podstatně výkonnější, neboť se data načítají paralelně z více disků současně. Obecně se dá říci, že RAID 1 poskytuje zhruba dvojnásobný přenos při čtení a zhruba stejný při zápisu ve srovnání se samostatnými disky.

### RAID 2 a RAID 3

Nejedná se o typické implementace RAIDu. RAID 2 rozkládá data na úrovni bitů, nikoliv bloků. RAID 3 data rozkládá na úrovni bytů, má vyhrazený disk pro paritní data a nedokáže obsloužit více současných požadavků. Tyto typy se používají jen vzácně.

## RAID 4

RAID 4 pracuje na úrovni bloků, podobně jako RAID 0, ale je vybaven vyhrazeným diskem pro paritní data. V případě havárie disku jsou paritní data použita k jeho obnově. Paritní disk ale může znamenat ztrátu výkonu při zápisu dat. Nicméně se RAID 4 občas používá.

## RAID 5

RAID 5 je kompromisem mezi typy 0 a 1, co se týče výkonu i zabezpečení dat. Kapacita pole je rovná kapacitě všech použitých disků bez jednoho. Data jsou rozdělena na jednotlivých discích podobně jako v případě pole RAID 0, ale navíc se o bezpečnost dat starají tzv. *paritní bloky*, které jsou vytvořeny na jednom z diskových oddílů. Paritní bloky jsou navzájem spojeny pomocí logického XOR —, v případě výpadku jednoho z disků tak mohou být data obnovena z odpovídajících paritních bloků a ostatních dat. Při používání pole typu RAID 5 nesmí dojít k výpadku více než jednoho disku současně. V zájmu ochrany dat je tedy nutné vadný disk co nejrychleji nahradit.

## Další RAID typy

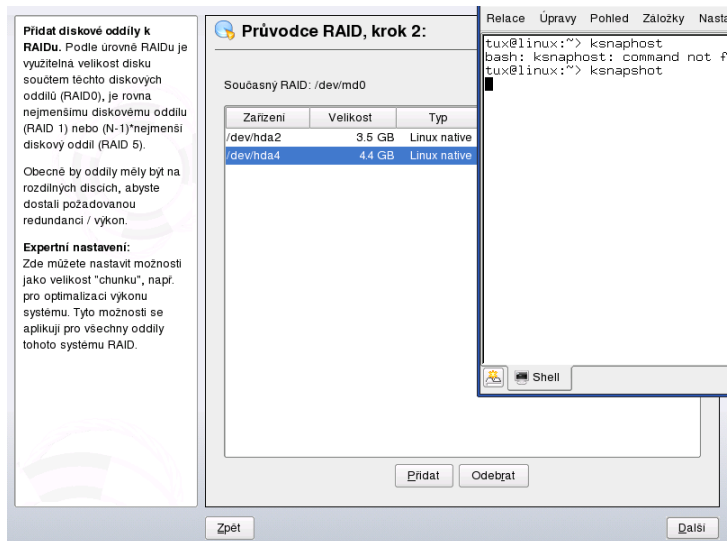
Existuje mnoho dalších typů RAIDu (RAIDn, RAID 10, RAID 0+1, RAID 30, RAID 50 atd.), některé z nich jsou proprietárními implementacemi výrobců hardware. Nejsou příliš rozšířené, a proto tu nejsou vysvětleny.

## 2.2.2 Konfigurace softwarového RAIDu pomocí YaST

Konfigurace softwarového RAIDu v YaSTu je přístupná z modulu *Rozdělování disku pro experty*, popsaného v části „Dělení disku“ (2 – „Konfigurace pomocí YaST“, ↑Uživatelská příručka). Tento profesionální nástroj pro rozdělování disku umožňuje upravovat a mazat existující oddíly a vytvářet nové pro použití v softwarovém RAIDu. RAID oddíl vytvoříte kliknutím na *Vytvořit* → *Neformátovat* a výběrem *0xFD Linux RAID* jako identifikátoru oddílu. Pro RAID 0 a RAID 1 jsou zapotřebí alespoň dva oddíly, pro RAID 1 se obvykle více než dva oddíly nepoužívají. Pokud chcete RAID 5, musíte použít alespoň tři oddíly, které by měly mít všechny stejnou velikost. Oddíly pro RAID by měly být umístěny na různých fyzických discích, aby se předešlo ztrátě dat v případě selhání disku (RAID 1 a 5) nebo dosáhlo vyššího výkonu RAID 0. Po vytvoření všech oddílů pro RAID klikněte na *RAID* → *Vytvořit RAID*. Zahájíte tak konfiguraci RAIDu.

V dalším dialogu vyberte mezi RAID levely 0, 1 nebo 5 (viz 2.2.1 – „Běžné typy polí RAID“ (strana 54)). Po kliknutí na *Další* se zobrazí dialog, který obsahuje přehled všech oddílů typu „Linux RAID“ nebo „Linux native“ (viz 2.6 – „Oddíly RAID“ (strana 56)). Odkládací (swap) ani DOS oddíly nejsou zobrazeny. Pokud je oddíl přiřazen do RAID svazku, je zobrazeno jméno RAID zařízení (např. /dev/md0). Nepřiřazené oddíly jsou označeny „--“.

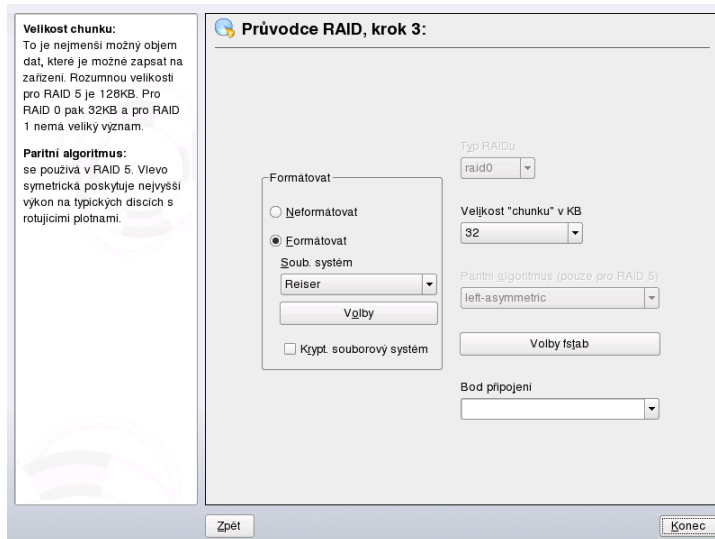
**Obrázek 2.6** Oddíly RAID



Chcete-li nepřiřazený oddíl přidat do RAID svazku, klikněte v seznamu na oddíl a pak na *Přidat*. Jméno RAID zařízení se zobrazí vedle vybraného oddílu. Přiřaďte všechny oddíly určené pro RAID. Jinak by místo na nich zůstalo nevyužité. Po přiřazení všech oddílů klikněte na *Další*, dostanete se tak do dialogu, ve kterém můžete vyladit výkon (viz 2.7 – „Nastavení souborového systému“ (strana 57)).



## Obrázek 2.7 Nastavení souborového systému



Stejně jako v případě konvenčních oddílů vyberte typ souborového systému, případně šifrování a bod připojení. Zaškrtnutí volby *Perzistentní superblok* zabezpečuje rozeznání RAID oddílů při startu systému. Po ukončení konfigurace tlačítkem *Konec* si prohlédněte vytvořené zařízení `/dev/md0`, případně další označená jako *RAID*, v expertním modulu pro rozdělování disku.

## 2.2.3 Řešení problémů

Zda byl některý z oddílů zapojených do RAIDu poškozen, zjistíte v souboru `/proc/mdstat`. Pokud nastala chyba, vypněte systém a vyměňte poškozený pevný disk za nový, obsahující stejné oddíly jako disk původní. Pak restartujte systém a zadejte příkaz `mdadm /dev/mdX --add /dev/sdX`. 'X' nahraďte patřičnými identifikátory zařízení. Tím bude nový disk automaticky zapojen do pole RAID a data budou obnovena.

## 2.2.4 Další informace

Pokyny ke konfiguraci a další informace o softwarovém RAIDu naleznete v následujícím HOWTO dokumentu:

- `/usr/share/doc/packages/raidtools/Software-RAID.HOWTO.html`
- <http://en.tldp.org/HOWTO/Software-RAID-HOWTO.html>

nebo v konferenci věnované linuxovému RAIDu: <http://www.mail-archive.com/linux-raid@vger.rutgers.edu>.

# Aktualizace systému

SUSE Linux nabízí možnost aktualizovat stávající systém, aniž by bylo nezbytné ho znovu instalovat. Přitom je třeba rozlišovat mezi *aktualizací jednotlivých balíků* a *celkovou aktualizací systému*. Balíčky lze také doinstalovat ručně pomocí RPM.

## 3.1 Aktualizace systému SUSE Linux

Existuje známý jev, že se software verze od verze rozrůstá. Proto je dobré podívat se *před* aktualizací příkazem `df`, jak jsou diskové oddíly zaplněny. Pokud máte dojem, že by na to jeho kapacita nestačila, zálohujte data a proveďte přerozdělení disku. Neexistuje žádná univerzální rada, kolik místa budete potřebovat, to závisí na způsobu stávající instalace, vybraném softwaru a na tom, z které verze aktualizujete.

### 3.1.1 Přípravy

Před začátkem aktualizace byste měli zálohovat konfigurační soubory na jiné médium (streamer, disketa, výměnný disk, ZIP mechanika, vypálit na CD). V první řadě se jedná o soubory v adresáři `/etc`, dále v adresáři `/var/lib` (např. News nebo XDM). Kromě toho zálohujte také soubory z domovských adresářů.

Než spustíte samotnou aktualizaci, poznamenejte si, jaký máte kořenový diskový oddíl `/`, což zjistíte příkazem `df /`

V příkladu výstupu je kořenovým oddílem `/dev/hda2`:

```
tux@linux:~>df -h
Filesystem Size Used Avail Use% Mounted on
/dev/hda1 1.9G 189M 1.7G 10% /dos
/dev/hda2 8.9G 7.1G 1.4G 84% /
/dev/hda5 9.5G 8.3G 829M 92% /home
```

## 3.1.2 Možné problémy

Po přechodu na novou verzi se můžete setkat s různými problémy. Zde najdete jejich popis.

### Kontrola passwd a group v /etc

Před aktualizací se ujistěte, že soubory `/etc/passwd` a `/etc/group` neobsahují žádné chyby v syntaxi. To provedete jako uživatel `root` pomocí ověřovacích nástrojů `pwck` a `grpck`. Zjištěné chyby opravte.

### PostgreSQL

Před aktualizací databáze PostgreSQL balík musíte vydumpovat databázi více v `pg_dump`. Tento postup je nutné dodržovat je v případě, že byla databáze PostgreSQL před aktualizací *používána*.

## 3.1.3 Aktualizace pomocí YaST

Postupujte jako u instalace podle postupu uvedeného v části [3.1.1 – „Přípravy“](#) (strana 59) a pak systém aktualizujte následujícím způsobem:

- 1 Spustíte instalaci podle postupu popsaného v části „Spouštění instalačního programu“ (1 – *„Instalace pomocí nástroje YaST“*, ↑Uživatelská příručka). V programu YaST, nastavte jazyk. Místo *Nová instalace* zvolte *Aktualizace stávajícího systému*.
- 2 YaST zjistí, zda se na disku nenachází více kořenových oddílů. Pokud ne, pokračuje dále. Pokud na disku máte více oddílů, musíte zvolit kořenový oddíl a potvrdit výběr stisknutím tlačítka *Další*. YaST načte starý `fstab` a pokusí se připojit zde uvedené oddíly.

- 3 Nyní můžete vytvořit zálohu systémových souborů. Pokud již nemáte vlastní zálohu, doporučujeme vám tuto volbu využít. Záloha může být později velmi užitečná.
- 4 Vyberte rozsah aktualizace systému (např. *Standardní systém*). Drobné nesrovnalosti můžete později upravit pomocí programu YaST.

## 3.2 Od verze k verzi

V následujících odstavcích bude popsáno, jaké detaily se změnilo od jedné verze k následující. V tomto přehledu bude např. uvedeno, zda se změnilo základní nastavení, zda došlo k přesunutí konfiguračních souborů na nové místo, nebo jestli se změnilo chování důležitých programů. Jsou zde uvedeny pouze věci, se kterými se uživatel resp. administrátor běžně setká. Tento seznam není v žádném případě úplný a vyčerpávající.

Problémy jednotlivých verzí jsou uveřejněny okamžitě po jejich odhalení. Důležité update jednotlivých balíčků najdete na stránce <http://www.novell.com/products/linuxprofessional/downloads/>. Jejich instalace provádí pomocí YaST Online Update (YOU)—viz „Online aktualizace“ (2 – „*Konfigurace pomocí YaST*“, ↑Uživatelská příručka).

### 3.2.1 Změny z 8.1 na 8.2

- 3D podpora karet s čipy nVidia (změna): `NVIDIA_GLX` a `NVIDIA_kernel` již nejsou součástí distribuce (včetně skriptů `switch2nvidia_glx`). Místo toho prosím použijte instalátor společnosti nVidia pro *Linux IA32*, který naleznete na <http://www.nvidia.com>. Následně pak použijte YaST pro aktivaci 3D podpory.
- Při nové instalaci bude použit místo `inetd` program `xinetd`. Konfigurační adresář je `/etc/xinetd.d`. Při aktualizaci zůstane zachován `inetd`.
- PostgreSQL je nyní k dispozici ve verzi 7.3. Při přechodu z verze 7.2.x doporučujeme `dump/restore` příkazem `pg_dump`. Pokud vaše aplikace přistupují k systémovým katalogům, pak je třeba provést ještě další úpravy, protože 7.3 již zavádí schémata. Podrobné informace naleznete na <http://www.ca.postgresql.org/docs/momjian/>

- PostgreSQL je nyní pouze ve verzi 7.3. pro přechod z verzí 7.2.x je určen `dump/restore` s příkazem `pg_dump`. Pokud vaše aplikace vyžaduje systémový katalog, musíte provést ještě další úpravy, kterými zavedete schéma verze 7.3. Více informací najdete na stránce [http://www.ca.postgresql.org/docs/momjian/upgrade\\_tips\\_7.3](http://www.ca.postgresql.org/docs/momjian/upgrade_tips_7.3)
- Verze 4 programu `stunnel` již nepodporuje na příkazové řádce žádné parametry. Je však poskytován spolu se skriptem `/usr/sbin/stunnel3_wrapper`, který parametry příkazové řádky pro `stunnel` dokáže konvertovat do konfiguračního souboru. Jeho použití je následující (položku `OPTIONS` nahraďte parametry):

```
/usr/sbin/stunnel3_wrapper stunnel OPTIONS
```

Konfigurační soubor se zároveň vypíše do standardního výstupu, aby bylo možné se seznámit se syntaxí pro zápis do trvalého konfiguračního souboru.

- `openjade` (`openjade`) je nyní DSSSL engine, který se používá místo `jade` (`jade_dsl`), když je spuštěn `db2x.sh` (`docbook-toys`). Z důvodů kompatibility jsou jednotlivé programy také bez předpony `o`.

Pokud je nějaká aplikace závislá na adresáři `jade_dsl` a tam umístěných souborech, pak je třeba buď ji přesměrovat na `/usr/share/sgml/openjade` nebo vytvořit odkaz (jako `root`):

```
cd /usr/share/sgml
rm jade_dsl
ln -s openjade jade_dsl
```

Abyste zabránili konfliktu s `rsz`, jmenuje se příkaz `sx` i nadále `s2x`, resp. `sgml2xml` nebo `osx`.

## 3.2.2 Změny z 8.2 na 9.0

Problémy a zvláštnosti:

- Došlo ke změně verze správce balíků RPM na verzi 4. Nové balíky se nyní vytvářejí příkazem `rpmbuild`. Příkaz `rpm` je nadále používán pro instalaci, aktualizaci a dotazy.
- Pro nastavení `tisku` přibyl balík `footmatic-filters`. Obsah byl získán z balíku `cups-drivers`, aby bylo možné filtry používat i v případě, že není nainstalován CUPS. Díky tomu nyní lze prostřednictvím programu YaST získat nastavení nezá-

vislé na tiskovém systému (CUPS, LPRng). Balík obsahuje konfigurační soubor `/etc/foomatic/filter.conf`.

- I při nasazení LPRng/lpdfiltru jsou nyní vyžadovány balíky `foomatic-filters` a `cups-drivers`.
- XML zdroje balíků jsou zpracovávány pomocí záznamů v `/etc/xml/suse-catalog.xml`. Tento soubor nesmí být změněn příkazem `xmlcatalog`, protože by mohlo dojít k přemazání komentářů nutných pro aktualizaci. Soubor `/etc/xml/suse-catalog.xml` je zpracován pomocí výrazu `nextCatalogv` `/etc/xml/catalog`, aby nástroje jako `xmllint` nebo `xsltproc` automaticky našli lokální zdroje.

### 3.2.3 Změny z 9.0 na 9.1

Problémy a zvláštnosti najdete popsané v článku v databázi instalační podpory na stránce <http://en.opensuse.org/SDB:SDB>.

- SUSE Linux používá jádro řady 2.6. Jádro řady 2.4 již není k dispozici a je možné, že pokud používáte programy, vyžadující starší jádro, tyto programy přestanou fungovat. Ze změnou jádra souvisí i následující změny:
  - Zavádění modulů se nyní nastavuje v souboru `/etc/modprobe.conf`. Soubor `/etc/modprobe.conf` se přestal používat. YaST dokáže do určité míry starý soubor převést (pomocí skriptu `/sbin/generate-modprobe.conf`).
  - Moduly mají nyní příponu `.ko`.
  - IDE vypalovačky již pro vypalování nepotřebují modul `ide-scsi`.
  - Z parametrů modulů ALSA byla odstraněna přímona `snd_`.
  - `/proc` byl nahrazen novým `sysfs`.
  - Správa napájení (především ACPI) lze nyní nastavit i prostřednictvím programu YaST.
- NGPT a `linuxthreads`

Programy linkované proti NGPT (*Next Generation POSIX Threading*) již s glibc 2.3.x nepoběží. Všechny takto postižené programy, které nejsou součástí distribuce SUSE Linux musí být kompilovány s podporou linuxthreads nebo NPTL (*Native POSIX Thread Library*).

Problémy s NPTL mohou nastat také na systémech se starší implementací linuxthreads, pokud nenastavíte následující proměnnou prostředí (*kernel-version* nahraďte příslušnou verzí jádra):

```
LD_ASSUME_KERNEL=kernel-version
```

Možné jsou tyto verze:

- 2.2.5 (i386, s390): linuxthreads bez Floating Stacks
- 2.4.1 (AMD64, i586, i686): linuxthread s Floating Stacks

Poznámky k jádru a linuxthreads s *Floating Stacks*:

Programy používající *errno*, *h\_errno* a *\_res*, potřebují hlavičkové soubory (*errno.h*, *netdb.h* a *resolv.h*). C++ programy s podporou multithread, potřebují ke správnému chodu nastavit proměnnou prostředí *LD\_ASSUME\_KERNEL=2.4.1*.

NPTL (*Native POSIX Thread Library*) je v systému obsažena jako balíček Thread. NPTL slouží k zajištění binární kompatibility se starší knihovnou linuxthreads.

- Jako výchozí kódování je pro systéme použít standard *UTF-8*. Při instalaci se zadá také národní kódování ve formátu *NarodniKodovani.UTF-8* (např. *cs\_CZ.UTF-8*).
- Nástroje z balíku *coreutils* jako *tail*, *chown*, *head*, *sort* se řídí POSIX standardem z roku 2001 (*Single UNIX Specification, version 3 == IEEE Std 1003.1-2001 == ISO/IEC 9945:2002*) ale již ne standardem z roku 1992. Staré nastavení můžete získat pomocí proměnné prostředí:

```
_POSIX2_VERSION=199209
```

(Nové nastavení je *200112* a je převzato z *\_POSIX2\_VERSION*.) SUSE standard je dostupný na stránce (zdarma po registraci) <http://www.unix.org/>

Současné nastavení:



**Tabulka 3.1** Srovnání POSIX 1992 a POSIX 2001

POSIX 1992	POSIX 2001
<code>chown tux.users</code>	<code>chown tux:users</code>
<code>tail +3</code>	<code>tail -n 3</code>
<code>head -1</code>	<code>head -n 1</code>
<code>sort +3</code>	<code>sort -k 4</code>
<code>nice -10</code>	<code>nice -n 10</code>
<code>split -10</code>	<code>split -l 10</code>

---

### Tip

Software třetích stran se novým standardem ještě nemusí řídit. V takovém případě nastavte proměnnou prostředí takto: `_POSIX2_VERSION=199209`.

---

- Soubor `/etc/gshadow` byl odstraněn. Důvody pro tento krok jsou tyto:
  - Nemá žádnou podporu v glibc.
  - Soubor nemá žádné oficiální rozhraní a propojení. Toto propojení nemá ani systém shadow.
  - Většina aplikací kontrolujících heslo skupiny ignoruje tento soubor z výše uvedených důvodů.
- Podle FHS jsou nyní XML zdroje (DTD, Stylesheety atd.) nainstalované v adresáři `/usr/share/xml`. Z tohoto důvodu již tyto soubory nenajdete v adresáři `/usr/share/sgml`. V případě problémů je nutné vytvořit případný skript, upravit Makefile nebo tzv. oficiální katalogy (především `/etc/xml/catalog` popř. `/etc/sgml/catalog`).

## 3.2.4 Změny z 9.1 na 9.2

Více informací získáte v anglickém článku *Known Problems and Special Features in SUSE LINUX 9.2* v databázi instalační podpory SUSE Support Database na stránce <http://en.opensuse.org/SDB:SDB> po zadání klíčových slov *special features*.

### Aktivace firewallu během instalace

Aby byla zvýšena bezpečnost systému, je na konci instalace v návrhu aktivován firewall SuSEFirewall2. Po spuštění firewallu jsou zavřeny všechny porty. Potřebné porty lze otevřít v dialogu návrhu.

V případě síťového přístupu během instalace příslušný modul programu YaST otevře potřebné TCP a UDP porty na interních i externích rozhraních. Pokud potřebujete jiné nastavení, proveďte je v modulu firewallu programu YaST po instalaci.

**Tabulka 3.2** *Porty důležitých služeb*

Služba	Port
HTTP server	Firewall je nastaven podle konfigurace (pouze TCP)
Mail (postfix)	smtp 25/TCP
Samba server	netbios-ns 137/TCP; netbios-dgm 138/TCP; netbios-ssn 139/TCP; microsoft-ds 445/TCP
DHCP server	bootpc 68/TCP
DNS server	domain 53/TCP; domain 53/UDP
- " -	Plus zvláštní podpora pro port mapper v aplikaci SuSEFirewall2
Port mapper	sunrpc 111/TCP; sunrpc 111/UDP
NFS server	nfs 2049/TCP
- " -	Plus port mapper

---

Služba	Port
NIS server	Aktivuje portmap
TFTP	tftp 69/TCP
CUPS (IPP)	ipp 631/TCP; ipp 631/UDP

---

## KDE a podpora IPv6

Ve výchozím nastavení KDE není podpora IPv6 povolena. Povolit ji můžete v modulu editor souborů `/etc/sysconfig` programu YaST. Důvod, proč není podpora IPv6 povolena, spočívá ve skutečnosti, že IPv6 adresy nejsou správně podporovány všemi poskytovateli internetového připojení. Chybná podpora může vést k chybovým hlášením a vysokým prodlevám při načítání stránek.

## YaST Online Update a "delta balíčky"

YaST Online Update nyní podporuje zvláštní druh RPM balíčků, které obsahují pouze odlišné části spustitelných souborů. Tato nová technologie výrazně zmenšuje velikost opravných balíčků a tím také délku jejich stahování. Nastavení potřebná k používání "delta balíčků" můžete provést v souboru `/etc/sysconfig/onlineupdate`. Podrobnější informace najdete v souboru `/usr/share/doc/packages/deltarpm/README`.

## Konfigurace tiskového systému

Na konci instalace (proposal dialog) je nutné na firewallu otevřít port pro tiskový systém. CUPS používá porty 631/TCP a 631/UDP. Pracovní stanice by měla mít tyto porty zavřené. V případě tisku přes LPD nebo SMB musí být otevřený také port 515/TCP (starý LPD protokol).

## Přechod na X.Org

Přechod z XFree86 na X.Org je zjednodušen kompatibilitou odkazy se starými jmény na nové důležité soubory a příkazy

**Tabulka 3.3** Příkazy

XFree86	X.Org
XFree86	Xorg
xf86config	xorgconfig
xf86cfg	xorgcfg

**Tabulka 3.4** Soubory s logy v adresáři /var/log

XFree86	X.Org
XFree86.0.log	Xorg.0.log
XFree86.0.log.old	Xorg.0.log.old

Při přechodu na X.Org bal samozřejmě balíček XFree86\* změněn na xorg-x11\*.

## Emulátory terminálu pro X11

Vyřadili jsme z distribuce řadu emulátorů, protože nejsou již spravované nebo jsou nefunkční ve výchozím prostředí např. nepodporují UTF-8. SUSE Linux nabízí standardní terminály jako xterm, terminály prostředí KDE a GNOME a mlterm (Multilingual Terminal Emulator for X), který může být náhradou za aterm a eterm.

## Změny v balíčku powersave

Došlo ke změně konfiguračních souborů v /etc/sysconfig/powersave:

**Tabulka 3.5** Splynutí konfiguračních souborů do /etc/sysconfig/powersave

Staré	Součástí
/etc/sysconfig/powersave/ common	common

Staré	Součásti
	cpufreq
	events
	battery
	sleep
	thermal

Soubor `/etc/powersave.conf` zastaral. Existující proměnné byly přesunuty do souborů v tabulce 3.5 – „[Splynutí konfiguračních souborů do /etc/sysconfig/powersave](#)“ (strana 68). Pokud jste měnili `events` proměnné v `/etc/powersave.conf`, musíte nyní provést v `/etc/sysconfig/powersave/events`.

Stavy uspání se změnilly z:

- `uspat` (ACPI S4, APM suspend)
- `standby` (ACPI S3, APM standby)

na:

- `uspat na disk` (ACPI S4, APM suspend)
- `uspat do ram` (ACPI S3, APM suspend)
- `standby` (ACPI S1, APM standby)

## OpenOffice.org (OOo)

Cesty:

OOo se nyní instaluje místo do adresáře `/opt/OpenOffice.org` do adresáře `/usr/lib/ooo-1.1`. Výchozí adresář pro uživatelská nastavení je `~/ .ooo-1.1` místo původního `~/OpenOffice.org1.1`.

Wrapper:

Některé OoO komponenty jsou spouštěny novými wrappery. Nová jména jsou uvedena v následující tabulce 3.6 – „Wrapper“ (strana 70).

**Tabulka 3.6** *Wrapper*

Starý	Nový
/usr/X11R6/bin/OoO-calc	/usr/bin/oocalc
/usr/X11R6/bin/OoO-draw	/usr/bin/oodraw
/usr/X11R6/bin/OoO-impress	/usr/bin/ooimpress
/usr/X11R6/bin/OoO-math	/usr/bin/oomath
/usr/X11R6/bin/OoO-padmin	/usr/sbin/oopadmin
/usr/X11R6/bin/OoO-setup	-
/usr/X11R6/bin/OoO-template	/usr/bin/oofromtemplate
/usr/X11R6/bin/OoO-web	/usr/bin/ooweb
/usr/X11R6/bin/OoO-writer	/usr/bin/oowriter
/usr/X11R6/bin/OoO	/usr/bin/ooffice
/usr/X11R6/bin/OoO-wrapper	/usr/bin/ooo-wrapper

Wrapper nyní podporuje volbu `--icons-set` pro přepnutí ikon mezi KDE a GNOME. Následující volby již nejsou podporovány:

`--default-configuration`, `--gui`, `--java-path`, `--skip-check`, `--lang` (jazyk je nastaven podle locales), `--messages-in-window` a `--quiet`.

Podpora KDE a GNOME:

Rozšíření pro KDE a GNOME jsou dostupné v balíčcích `OpenOffice_org-kde` a `OpenOffice_org-gnome`.

## Zvukový směšovač kmix

Jako výchozí zvukový směšovač je nastaven kmix. Pro high-end hardware jsou dostupné starší směšovače jako QAMix/KAMix, envy24control (pouze ICE1712) nebo hdspmixer (pouze RME Hammerfall).

### 3.2.5 Změny z 9.2 na 9.3

Více informací získáte v anglickém článku *Known Problems and Special Features in SUSE Linux 9.3* v databázi instalační podpory SUSE na stránce <http://en.opensuse.org/SDB:SDB> po zadání klíčových slov *special features*.

## Spuštění ruční instalace s promptu jádra

Instalační nabídka již neobsahuje položku *Manual Installation* (ruční instalace). Ruční instalaci v linuxrc spustíte zadáním parametru `manual=1`. Tento způsob instalace není ve většině již nutný. Instalační parametry jako např. instalační zdroj můžete zadat přímo jako parametr.

## Síťové ověřování a Kerberos

Místo aplikace `heimdal` je nyní výchozí pro síťové ověřování systém Kerberos. Existující `heimdal` konfiguraci nelze automaticky převést. Během aktualizace systému se vytvoří záložní soubory uvedené v tabulce 3.7 – „Záložní soubory“ (strana 71).

**Tabulka 3.7** Záložní soubory

Starý soubor	Záložní soubor
<code>/etc/krb5.conf</code>	<code>/etc/krb5.conf.heimdal</code>
<code>/etc/krb5.keytab</code>	<code>/etc/krb5.keytab.heimdal</code>

Klientská konfigurace (`/etc/krb5.conf`) je velmi podobná nastavení `heimdal`. Pokud jste neprováděli žádné zvláštní nastavení, stačí položku `kpasswd_server` nahradit za `admin_server`.

Data serveru (kdc/kadmind) převzít nelze. Po aktualizaci je heimdal databáze stále dostupná v `/var/heimdal`; MIT kerberos databáze se nachází v `/var/lib/kerberos/krb5kdc`.

## JFS: ukončení podpory

Z důvodů technických problémů s JFS není tento souborový systém již podporován. Jádro sice stále obsahuje příslušný modul, ale program YaST již neumožňuje vytvoření oddílu s JFS.

## AIDE jako náhrada Tripwire

K detekci průlomu použijte program AIDE (balíček `aide`), který je šířen pod licencí GPL. Tripwire již není součástí systému SUSE Linux.

## Konfigurační soubor X.Org Configuration File

SaX2, nástroj pro nastavení grafického prostředí, zapisuje konfiguraci X.Org do souboru `/etc/X11/xorg.conf`. Při čisté instalaci se již nevytvoří symbolický odkaz na `XF86Config`.

## Ukončená podpora XView a OpenLook

Balíčky `xview`, `xview-devel`, `xview-devel-examples`, `olvwm` a `xtoolpl` již nejsou součástí instalace. Knihovny XView nebudou proto po aktualizaci k dispozici. To může vést k problémům především u vlastních kompilovaných aplikací, které je vyžadují.

K dispozici již není ani OLVWM (OpenLook Virtual Window Manager). Zvolte jiného správce oken.

## Konfigurace PAM

*Nové konfigurační soubory (více informací najdete v komentářích):*

```
common-auth  
    výchozí nastavení PAM sekce auth
```



`common-account`  
výchozí nastavení PAM sekce `account`

`common-password`  
výchozí nastavení PAM pro změnu hesla

`common-session`  
výchozí nastavení PAM pro správu sezení

Protože je jednodušší tímto způsobem upravovat a spravovat nastavení, měli byste do těchto souborů přesunout také nastavení ze svých aplikací. Pokud pak některou z aplikací nainstalujete později, automaticky se aplikují již provedené změny a správce systému již nemusí nastavení provádět ručně.

Změny jsou jednoduché. Pokud máte následující konfigurační soubor (výchozí pro většinu aplikací):

```
##PAM-1.0
auth      required      pam_unix2.so
account   required      pam_unix2.so
password  required      pam_pwcheck.so
password  required      pam_unix2.so   use_first_pass use_authtok
#password required      pam_make.so     /var/yp
session   required      pam_unix2.so
```

změňte ho takto:

```
##PAM-1.0
auth      include      common-auth
account   include      common-account
password  include      common-password
session   include      common-session
```

## Striktnější syntaxe `tar`

Syntaxe příkazu `tar` je nyní striktnější. Parametry `tar` musí být zadány před před zadáním souborů a adresářů. Zápis parametrů jako např. `--atime-preserve` nebo `--numeric-owner` za souborem nebo adresářem povede k chybě. překontrolujte prosím své zálohovací skripty. Příkazy podobné následujícímu nebudou fungovat:

```
tar czf etc.tar.gz /etc --atime-preserve
```

Více informací najdete v info stránce příkazu `tar`.

## 3.2.6 Změny z 9.3 na 10.0

Více informací získáte v anglickém článku *Known Problems and Special Features in SUSE Linux 10.0* v databázi instalační podpory SUSE Support Database na stránce <http://en.opensuse.org/SDB:SDB> po zadání klíčových slov *special features*.

### Získání superuživatelských práv pomocí su

Po přihlášení na uživatele `root` pomocí příkazu `su` není nastavená proměnná `PATH`. Pokud chcete spustit prostředí s nastavenými cestami, použijte `su -` nebo v souboru `/etc/default/su` nastavte proměnnou `ALWAYS_SET_PATH` na `yes`.

### Proměnné Powersave

Kvůli zachování konzistence došlo ke změně proměnných Powersave, ale soubory `syconfig` zůstávají stejné. Více informací najdete v Referenční příručce.

### PCMCIA

PC karty již nejsou kontrolovány prostřednictvím `cardmgr`, ale přímo jednotlivými moduly jádra. Všechny potřebné akce provádí `hotplug`. Startovací skript `pcmcia` byl odstraněn a `cardctl` nahrazen `pccardctl`. Více informací najdete ve svém systému v souboru `/usr/share/doc/packages/pcmciautils/README.SUSE`.

### TEI XSL styly

TEI XSL styly (`tei-xsl-stylesheets`) byly přemístěny do `/usr/share/xml/tei/stylesheet/rahtz/current`. Pokud chcete např. získat HTML výstup, spouštějte je z tohoto umístění (`base/p4/html/tei.xsl`). Více informací najdete na stránce <http://www.tei-c.org/Stylesheets/teic/>

## Oznamování změn souborového systému pro GNOME aplikace

GNOME aplikace vyžadují podporu oznamování změny souborového systému. U lokálních souborů nainstalujte balíček gamin (preferováno) nebo spusťte démona FAM. U vzdálených souborových systémů spusťte na serveru i klientovi FAM a otevřete firewall pro jeho RCP volání.

GNOME (gnome-vfs2 a libgda) obsahují wrapper pro výběr gamin nebo fam:

- Jestliže FAM démon neběží, použijte se gamin. (Důvod: Inotify je podporován pouze gamin, který je ideální pro lokální souborové systémy).
- Pokud FAM démon běží, použijte se FAM (Důvod: V případě spuštění FAM pravděpodobně potřebujete vzdálené oznamování, které podporuje pouze FAM).

### 3.2.7 Změny z 10.0 na 10.1

Více informací získáte v anglickém článku *Known Problems and Special Features in SUSE Linux 10.1* v databázi instalační podpory SUSE Support Database na stránce <http://en.opensuse.org/SDB:SDB> po zadání klíčových slov *special features*.

## Apache 2.2

Apache verze 2.2, 26 – „*Webový server Apache*“ (strana 391), byl kompletně přepracován. Další informace najdete na stránce <http://httpd.apache.org/docs/2.2/upgrading.html> a v popisu nových funkcí na stránce [http://httpd.apache.org/docs/2.2/new\\_features\\_2\\_2.html](http://httpd.apache.org/docs/2.2/new_features_2_2.html).

## Spouštění FTP serveru (vsftpd)

Ve výchozím nastavení již xinetd nespouští vsftpd FTP server. Nyní jde o samostatného démona, kterého je nutné nastavit v Editoru úrovní běhu v programu YaST.

## Firefox 1.5: Příkaz otevření URL

V Firefoxu verze 1.5 došlo ke změně metody pro otevření nové instance nebo okna prohlížeče v aplikacích. Nová metoda byla již částečně dostupná v předešlých verzích skriptem wrapperu.

Pokud vaše aplikace nepoužívá `mozilla-xremote-client` nebo `firefox-remote`, nemusíte nic měnit. V opačném případě je nový příkaz pro otevření URL `firefox url` a je jedno, zda již Firefox běží. Pokud již běží, bude se řídit nastavením v nabídce *Open links from other applications in*.

Z příkazové řádky můžete ovlivnit chování prohlížeče Firefox příkazy `firefox -new-window url` nebo `firefox -new-tab url`.

## **Část 2. Správa**



# Bezpečnost v Linuxu

Ke kontrole a směrování datového provozu ve své síti můžete použít také další mechanismy např. maškarádu, firewally nebo Kerbera. Secure Shell (SSH) umožňuje šifrované připojení na vzdálený počítač. Šifrování a další nástroje chrání vaše choulostivá data před nepovolanými uživateli. Mimo čistě technických informací v této kapitole najdete také základní informace o bezpečnostních aspektech linuxových sítí.

## 4.1 Firewall a maškaráda

Linux v síťovém prostředí umožňuje takovou manipulaci s pakety, která udržuje oddělené vnější a vnitřní síťové oblasti. Linuxový systém *netfilter* poskytuje prostředky pro vybudování efektivního firewallu udržujícího jednotlivé sítě odděleny. S pomocí *iptables* — obecné tabulkové struktury pro definici pravidel — umožňuje přesnou kontrolu, kterým paketům je dovoleno přejít přes síťové rozhraní. Takový paketový filtr lze snadno nastavit pomocí SuSEfirewall2 a odpovídajícího modulu YaST.

### 4.1.1 Filtrování paketů pomocí iptables

Komponenty netfilter a iptables jsou zodpovědné za filtrování a manipulaci s palety a za překlad síťových adres (NAT). Filtrovací kritéria a všechny s nimi spojené akce jsou uloženy v řetězech (chains), se kterými jsou porovnávány všechny příchozí pakety. Řetězky jsou uloženy v tabulkách. Manipulaci s těmito tabulkami a sadami pravidel umožňuje příkaz *iptables*.

Linuxové jádro si udržuje tři tabulky, každou z nich pro jednu skupinu funkcí paketo-  
vého filtru:

#### filter

Tato tabulka obsahuje většinu filtrovacích pravidel, neboť implementuje *filtrování paketů* v užším slova smyslu. Určuje například, který paket může projít skrz (ACCEPT) a který je zahozen (DROP).

#### nat

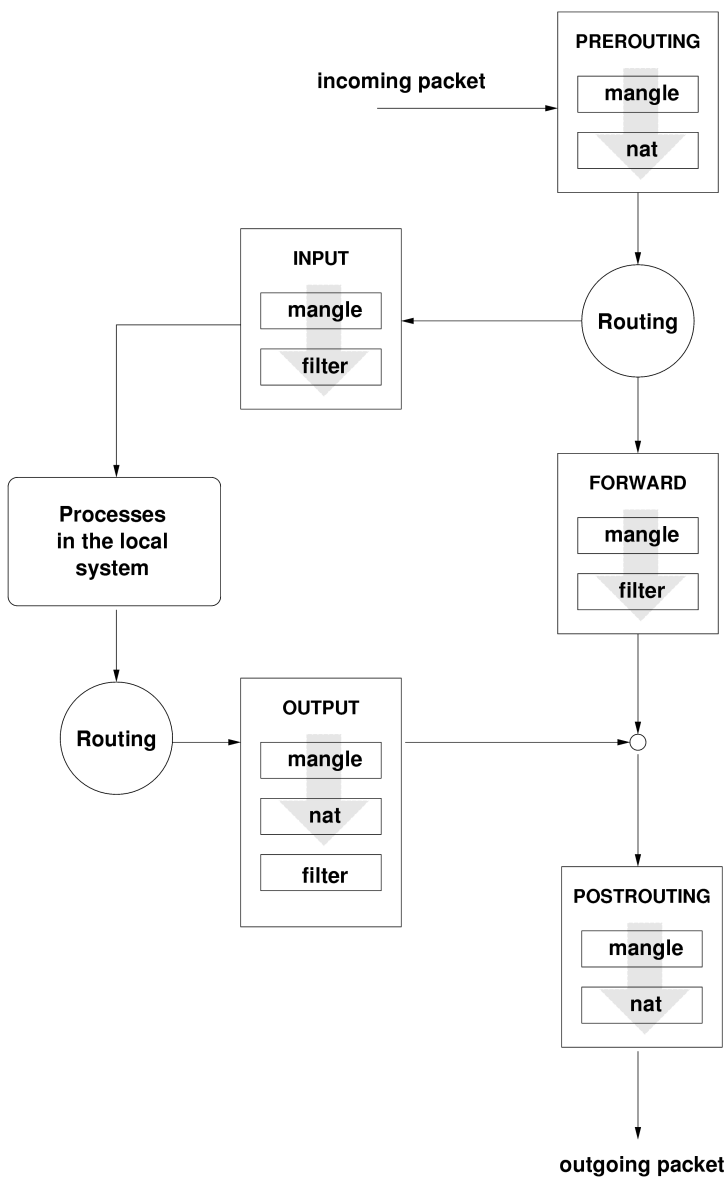
Tato tabulka určuje změny ve zdrojových a cílových adresách paketů. S její pomocí lze rovněž implementovat *maškarádu*, což je zvláštní případ NAT používaný pro propojení privátní sítě s Internetem.

#### mangle

Pravidla v této tabulce umožňují měnit hodnoty uložené v IP hlavičkách (např. typ služby).



**Obrázek 4.1** iptables: Možné cesty paketu



Výše zmíněné tabulky obsahují několik předdefinovaných řetězců (chains) pro porovnávání s pakety:

#### PREROUTING

Tento řetěz je aplikován na příchozí pakety.

#### VSTUP (input)

Tento řetěz je aplikován na pakety určené pro vnitřní systémové procesy.

#### FORWARD

Tento řetěz je aplikován na pakety, které jsou na systému pouze směrovány.

#### VÝSTUP (output)

Tento řetěz je aplikován na pakety, které pocházejí z vlastního systému.

#### POSTROUTING

Tento řetěz je aplikován na všechny odchozí pakety.

Obrázek 4.1 – „[iptables: Možné cesty paketu](#)“ (strana 81) znázorňuje cesty, po kterých se v systému může síťový paket pohybovat. Z důvodu jednoduchosti jsou tabulky zobrazeny jako části řetězů, ale ve skutečnosti jsou řetězy umístěny právě v tabulkách.

V nejjednodušším případě dorazí paket určený přímo pro systém na síťové rozhraní `eth0`. Paket je nejprve postoupen řetězu `PREROUTING` tabulky `mangle`, a pak řetězu `PREROUTING` tabulky `nat`. Následující krok určí, že cílem paketu je proces na vlastním systému. Po průchodu přes řetězy `INPUT` tabulek `mangle` a `filter` dosáhne paket konečně svého cíle, pokud ovšem odpovídá pravidlům v tabulce `filter`.

## 4.1.2 Základy maškarády

Maškaráda je linuxově specifická forma NAT (překlady síťových adres). Lze ji použít k propojení malé lokální sítě LAN (ve které počítače používají IP adresy z privátního rozsahu, viz 18.1.2 – „[Síťové masky a směrování](#)“ (strana 281)) s Internetem. Aby se mohl počítač z LAN připojit k Internetu, musí být jeho privátní adresa přeložena na veřejnou, používanou v Internetu. O to se stará router (směrovač), který slouží jako brána mezi LAN a Internetem. Princip je jednoduchý — router má více než jedno síťové rozhraní, obvykle síťovou kartu a zvláštní rozhraní pro připojení k Internetu. Zatímco druhé spojuje router s vnějším světem, první, nebo i více takových, spojuje router s počítači v síti LAN. Počítače v síti LAN tak mohou posílat pakety, které nejsou určeny pro lokální síť, na router.

---

## Důležité: Použití správné síťové masky

Při nastavení sítě se ujistěte, že oznamovací adresa a maska sítě je nastavena pro všechny počítače stejně. Pokud to tak není, síť nefunguje správně, protože pakety nemohou být správně směrovány.

---

Kdykoliv počítač v lokální síti LAN pošle paket určený pro internetovou adresu, je poslán na implicitní router. Router však musí být správně nakonfigurován, aby mohl pakety předávat dál. Z bezpečnostních důvodů to SUSE Linux neumožňuje v implicitní instalaci. Chcete-li předávání povolit, nastavte proměnnou `IP_FORWARD` v souboru `/etc/sysconfig/sysctl` na `IP_FORWARD=yes`.

Cílový počítač vidí váš router, ale neví nic o počítači ve vaší interní síti, ze kterého paket pochází. Proto se tato technika nazývá maškaráda. Díky překladu adres je router prvním cílem všech paketů zaslaných jako odpověď. Router musí tyto pakety rozpoznat a přeložit jejich cílovou adresu tak, aby mohly být předány správnému počítači v lokální síti.

Vzhledem k tomu, že směrování příchozích paketů závisí na maškarádové tabulce, neexistuje způsob, jak otevřít přímé spojení s počítačem v lokální síti zvenku. Pro takové spojení není v tabulce žádný zápis. Navíc každé již navázané spojení má v tabulce přiřazený stavový zápis, takže zápis nemůže být použit jiným spojením.

Důsledkem jsou možné problémy s některými komunikačními protokoly, např. ICQ, cucme, IRC (DCC, CTCP) a FTP (PORT režim). Mnoho FTP klientů používá režim PASV. Tento pasivní režim působí při používání maškarády a filtrování paketů podstatně méně problémů.

## 4.1.3 Základy firewallu

*Firewall* je běžně používaný termín pro mechanismus zajišťující propojení sítí a kontrolující přenos dat mezi nimi. Přesně řečeno, mechanismus popsany v této části se jmenuje *paketový filtr*. Paketový filtr řídí datový tok podle určitých kritérií, jako je komunikační protokol, porty a IP adresy. To umožňuje zablokovat pakety, které by, vzhledem ke svým adresám, neměly být do vaší sítě doručeny. Pokud chcete povolit veřejný přístup k vašemu webserveru, musíte explicitně otevřít příslušný port. Paketový filtr nicméně nezkontroluje obsah paketů s legitimními adresami, jako například paketů pro webserver. I když by příchozí pakety byly například zaslány za účelem nabourání CGI programu na webserveru, paketový filtr je nechá normálně projít.

Efektivnější ale složitější mechanismus je kombinace několika typů systémů, jako například paketový filtr spolupracující s aplikační bránou nebo proxy. V takovém případě paketový filtr odmítá všechny pakety určené pro zakázané porty. Přijaty jsou pouze pakety určené pro aplikační bránu. Tato brána nebo proxy předstírá, že je klientem. V jistém smyslu lze takovou proxy považovat za maškarádu na úrovni protokolu používaného aplikací. Takovou proxy je například Squid, HTTP proxy server. Aby prohlížeč mohl využít proxy, musí být patřičně nakonfigurován. Všechny HTTP požadavky jsou obsluhovány proxy s využitím cache (vyrovnávací paměti) a stránky, které se v cache nenalézají, jsou staženy proxy z Internetu. Dalším příkladem je SUSE proxy-suite (proxy-suite), která poskytuje proxy pro protokol FTP.

Následující část se zabývá paketovým filtrem dodávaným se systémem SUSE Linux. Další informace o filtrování paketů a firewallu naleznete v dokumentu Firewall HOWTO obsaženém v balíčku howto. Pokud je tento balíček nainstalovaný, můžete si dokument přečíst pomocí příkazu

```
less /usr/share/doc/howto/en/Firewall-HOWTO.gz.
```

## 4.1.4 SuSEfirewall2

Skript SuSEfirewall2 čte proměnné z `/etc/sysconfig/SuSEfirewall2` a generuje sadu iptables pravidel. SuSEfirewall2 definuje tři bezpečnostní zóny:

### Vnější síť

Protože neexistuje žádný způsob, jak kontrolovat dění ve vnější síti, musí být počítače proti ní chráněny. Ve většině případů je vnější síť Internet, ale může to být i jiná nezabezpečená síť, například WLAN.

### Vnitřní síť

Privátní síť, nejčastěji LAN. Pokud počítače v této síti používají IP adresy z privátního rozsahu (viz [18.1.2 – „Síťové masky a směrování“](#) (strana 281)), je pro přístup k vnější síti zapotřebí použít překlad síťových adres (NAT).

### Demilitarizovaná zóna (DMZ)

Ačkoliv jsou počítače v této zóně dosažitelné z vnitřní i vnější sítě, samy nemají do vnitřní sítě přístup. Tím se před vnitřní síti vytvoří obranný val navíc.

Jakýkoliv síťový provoz, který není explicitně povolen filtračním pravidlem, je pomocí iptables zakázán. Proto musí být každé síťové rozhraní umístěno do jedné ze tří zón. Pro každou zónu je třeba určit, které služby a protokoly jsou povoleny. Pravidla jsou

používána jen na pakety pocházející ze vzdálených počítačů. Lokálně generované pakety nejsou firewallem zachycovány.

Konfiguraci lze provést pomocí nástroje YaST (viz „[Konfigurace pomocí YaST](#)“ (strana 85)). Lze ji provést i ručně v dobře okomentovaném souboru `/etc/sysconfig/SuSEfirewall2`. Navíc je řada příkladů nastavení dostupná v `/usr/share/doc/packages/SuSEfirewall2/EXAMPLES`.

## Konfigurace pomocí YaST

---

### **Důležité: Automatické nastavení firewallu**

YaST automaticky spouští firewall na všech nakonfigurovaných rozhraních. Pokud je na systému nakonfigurován a aktivován server a v dialogích pro nastavení serveru použijete volbu *Na zvolených portech a rozhraních otevřít firewall* nebo *Na zvolených portech otevřít firewall*, YaST automaticky upraví konfiguraci firewallu. Dialogy některých serverových modulů mají tlačítko *Doladění firewallu* pomocí kterého lze aktivovat další služby a porty. Modul YaST pro firewall se používá pouze k aktivaci, deaktivaci či rekonfiguraci firewallu.

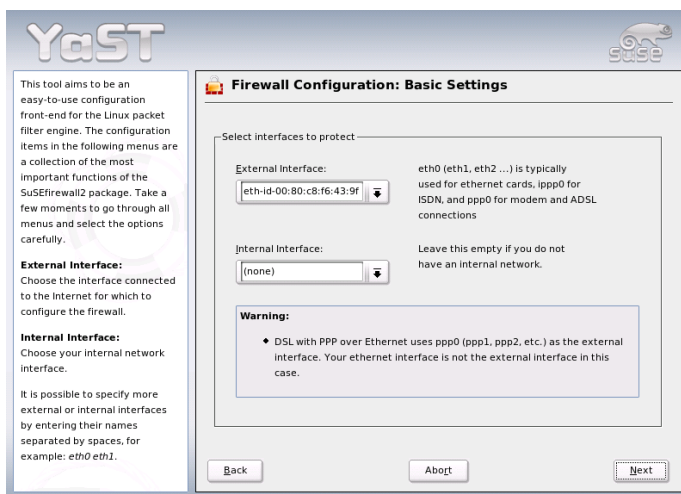
---

Dialogy pro nastavení firewallu v grafickém prostředí jsou dostupné v nástroji YaST zvolením položek *Bezpečnost a uživatelé* → *Firewall*. Konfigurace je rozdělena do sedmi částí, ke kterým lze přistupovat přímo stromové struktury vlevo.

### Začátek

V tomto dialogu můžete nastavit spouštění firewallu. Ve výchozím nastavení se SuSEfirewall2 spouští automaticky. Můžete ho ale spustit nebo zastavit v tomto dialogu ručně. Chcete-li použít svá nová nastavení, stiskněte *Uložit nastavení a restartovat firewall*.

## Obrázek 4.2 YaST: Konfigurace firewallu



### Rozhraní

Seznam v tomto dialogu obsahuje všechna známá rozhraní. Chcete-li rozhraní odebrat ze zóny, klikněte na něj, stiskněte tlačítko *Změnit* a vyberte *Není přiřazena žádná zóna*. Chcete-li rozhraní přiřadit zóně, stiskněte *Změnit* a vyberte některou z dostupných zón. Můžete také vytvořit zvláštní rozhraní s vlastním nastavením pomocí *Vlastní*.

### Povolené služby

Toto nastavení potřebujete pouze, pokud chcete aby systém nabízel služby dostupné ze zóny, proti které je chráněn. Ve výchozím nastavení je systém chráněn pouze proti vnějším zónám. Explicitně povolte služby, které mají být počítačům ve vnější síti dostupné. Nejprve v nabídce *Povolené služby pro vybranou zónu* zvolte zónu, pak přidejte služby, které pro ni mají být povoleny.

### Maškaráda

Maškaráda skrývá vnitřní síť před vnějšími sítěmi, jako je Internet, ale umožňuje počítačům z vnitřní sítě transparentně přistupovat k vnější síti. Požadavky z vnější do vnitřní sítě jsou zablokovány a požadavky z vnitřní sítě z vnějšku vypadají, jako by je vydával maškarádující server. Pokud mají být ve vnější síti dostupné služby stroje ve vnitřní síti, přidejte pro službu zvláštní přesměrovávací pravidlo.

## Broadcast

V tomto dialogu nastavte UDP porty, na kterých je povolen příjem broadcast paketů. K žádané zóně přidejte požadované porty nebo služby oddělené mezerami. Viz také `/etc/services`.

Lze zde také zapnout zaznamenávání nepřijatých broadcast paketů. To může působit problémy, neboť počítače s Windows generují velké množství broadcast paketů, kterým si o sobě dávají vědět. Jsou-li v síti takové počítače, mohou záznamy narůstat do velkých rozměrů.

## Podpora IPsec

V tomto dialogu nastavte, zda má být z vnější sítě dostupná služba IPsec. Po stisknutí tlačítka *Podrobnosti* můžete nastavit, jak se má IPsec paketům důvěřovat.

## Úroveň logování

Existují dvě pravidla pro logování: zaznamenávání přijatých a nepřijatých paketů. Pro každou z obou skupin můžete zvolit mezi *Zaznamenávat vše*, *Zaznamenávat pouze kritické* a *Nezaznamenávat nic*.

Po ukončení konfigurace pokračujte stisknutím tlačítka *Další*. Otevře se shrnutí nastavení konfigurace firewallu. V něm zkontrolujte všechna nastavení, služby, porty a protokoly, které byly povoleny. Chcete-li konfiguraci změnit, použijte tlačítko *Zpět*. Tlačítkem *Přijmout* konfiguraci uložíte.

# Ruční konfigurace

Následující odstavce popisují krok za krokem správný postup konfigurace. U každé konfigurační položky je uvedeno, zda je relevantní pro firewall nebo maškarádu. Nastavení týkající se DMZ (demilitarizované zóny) tu nejsou zmíněna. Jsou použitelná pouze ve složitějších sítích (obvykle podnikových), jejichž nastavení vyžaduje hlubokou znalost problematiky.

Nejprve pomocí editoru úrovní běhu YaST povolte `SuSEfirewall2` ve vámi používané úrovni (pravděpodobně 3 nebo 5). Tím se nastaví symbolické odkazy pro `SuSEfirewall2_*` skripty v adresářích `/etc/init.d/rc?.d/`.

## FW\_DEV\_EXT (firewall, maškaráda)

Zařízení připojené do Internetu. Pro modem vložte `ppp0`. Pro ISDN připojení použijte `ippp0`. Pro DSL připojení použijte `dsl0`. `auto` použijte pro rozhraní odpovídající výchozímu směrování.

FW\_DEV\_INT (firewall, maškaráda)

Zařízení připojené k vnitřní, privátní síti (např. eth0). Pokud firewall chrání jen počítač, na kterém běží, a nikoliv vnitřní síť, ponechte prázdné.

FW\_ROUTE (firewall, maškaráda)

Pokud chcete používat maškarádu, nastavte *yes*. Vnitřní počítače nebudou z vnější sítě viditelné, protože jejich privátní adresy (např. 192.168.x.x) nejsou v Internetu vůbec směrovány.

U firewallu bez maškarády nastavte *yes* pouze v případě, že chcete povolit přístup do vnitřní sítě. Pak ale musí mít počítače ve vnitřní síti platné IP adresy. V běžném případě byste neměli přístup zvenku povolovat!

FW\_MASQUERADE (maškaráda)

Pokud potřebujete maškarádu, uveďte zde *yes*. Tím vnitřní počítače získají v podstatě přímý přístup k Internetu. Uvědomte si, že přistupovat k Internetu je bezpečnější skrze proxy. Maškaráda není pro službu poskytovanou proxy serverem potřeba.

FW\_MASQ\_NETS (maškaráda)

Zde uveďte počítače či síť, které budou maškarádovány. Jednotlivé položky oddělujte mezerou, např.

```
FW_MASQ_NETS="192.168.0.0/24 192.168.10.1"
```

FW\_PROTECT\_FROM\_INT (firewall)

Nastavením *yes* zabezpečíte váš firewall před vnitřní sítí. Pak je třeba služby z interní sítě explicitně povolovat. Viz FW\_SERVICES\_INT\_TCP a FW\_SERVICES\_INT\_UDP.

FW\_SERVICES\_EXT\_TCP (firewall)

Zadejte TCP porty, které mají být přístupné. V případě domácí pracovní stanice, která nemá nabízet žádné služby, ponechte prázdné.

FW\_SERVICES\_EXT\_UDP (firewall)

Ponechte prázdné, pokud ovšem neprovádíte UDP službu, která by měla být z venku přístupná. Mezi takové služby patří DNS, IPSec, TFTP, DHCP a další. V takovém případě vložte potřebné UDP porty.



FW\_SERVICES\_INT\_TCP (firewall)

Tato proměnná určuje služby dostupné pro vnitřní síť. Zápis je stejný jako v případě FW\_SERVICES\_EXT\_TCP, jen je nastavení použito pro *vnitřní* síť. Proměnnou je potřeba nastavit, pouze pokud je FW\_PROTECT\_FROM\_INT nastavená na *yes*.

FW\_SERVICES\_INT\_UDP (firewall)

Viz FW\_SERVICES\_INT\_TCP.

Jakmile firewall nastavíte, otestujte ho. Sady pravidel vytvoříte jako uživatel `root` příkazem `SuSEfirewall2 start`. Pak se pokuste připojit telnetem z externího počítače, abyste ověřili, zda bude připojení skutečně odmítnuto. Následně si prohlédněte soubor `/var/log/messages`, ve kterém byste měli nalézt něco podobného:

```
Mar 15 13:21:38 linux kernel: SFW2-INext-DROP-DEFAULT IN=eth0
OUT= MAC=00:80:c8:94:c3:e7:00:a0:c9:4d:27:56:08:00 SRC=192.168.10.0
DST=192.168.10.1 LEN=60 TOS=0x10 PREC=0x00 TTL=64 ID=15330 DF PROTO=TCP
SPT=48091 DPT=23 WINDOW=5840 RES=0x00 SYN URGP=0
OPT (020405B40402080A061AFEBCC0000000001030300)
```

Mezi další balíčky, kterými můžete otestovat nastavení firewallu, patří `nmap` a `nessus`. Po jejich nainstalování k nim naleznete dokumentaci v adresářích `/usr/share/doc/packages/nmap` a `/usr/share/doc/packages/nessus-core`.

## 4.1.5 Další informace

Aktuální informace a další dokumentaci o balíčku `SuSEfirewall2` naleznete v `/usr/share/doc/packages/SuSEfirewall2`. Domovská stránka projektů `netfilter` a `iptables` na adrese <http://www.netfilter.org> poskytuje velké množství dokumentace v různých jazycích.

## 4.2 SSH: bezpečná práce v síti

V dnešní době, kdy je více a více počítačů instalovaných do prostředí sítě, je často nezbytné, aby se k nim dalo vzdáleně přistupovat. Obvykle to znamená, že se uživatel přihlásí – zašle přihlašovací jméno a heslo. Pokud jsou však tyto údaje zasilány přes síť jako prostý text, může se stát, že je cestou někdo odposlechne a získá přístup k účtu uživatele, aniž by o tom oprávněný uživatel věděl. Kromě toho, že útočník takto získá přístup k souborům uživatele, může se dostat i k účtu uživatele `root` nebo napadati další počítače. V minulosti se přihlašovalo na vzdálené počítače programem `telnet`,

který nenabízí žádné bezpečnostní mechanismy pro utajení přenášených údajů. Podobné chování mají i další často používané programy pro vzdálený přístup, např. ftp.

SSH naproti tomu nabízí ochranu přenášených informací. Šifruje jak přihlašovací údaje (login a heslo), tak i veškerou další komunikaci mezi počítači. Útočník stále může odposlouchávat, ale bez znalosti šifrovacího klíče nemůže získat původní obsah zasílaných dat. SSH tedy umožňuje bezpečně komunikovat se vzdálenými systémy přes nezabezpečenou síť, jako je např. Internet. Sada programů, které se v systému SUSE Linux starají o zabezpečení vzdáleného přístupu, se jmenuje OpenSSH.

## 4.2.1 Balíček OpenSSH

SUSE Linux instaluje balíček OpenSSH automaticky. Programy ssh, scp a sftp jsou pak dostupné jako alternativa programů telnet, rlogin, rsh, rcp a ftp. Ve výchozím nastavení je síťový přístup k systému možný jen pomocí OpenSSH nástrojů a pouze v případě, že je povolen na firewallu.

## 4.2.2 Program ssh

Program ssh vám umožní připojovat se na vzdálené stroje a interaktivně pracovat. Nahrazuje telnet i rlogin. Program slogin je jen symbolický odkaz na ssh. Například na vzdálený počítač sun se můžete přihlásit pomocí příkazu `ssh sun`. Vzdálený systém vás požádá o heslo (které máte nastavené na vzdáleném počítači sun).

Po úspěšném přihlášení můžete pracovat s příkazovým řádkem na vzdáleném stroji, nebo spouštět interaktivní aplikace, např. YaST. Pokud máte na vzdáleném počítači nastavené jiné přihlašovací jméno než na lokálním počítači, můžete se přihlásit s použitím jiného přihlašovacího jména příkazem `ssh -l augustynka sun` nebo `ssh augustynka@sun`.

Navíc můžete pomocí ssh spouštět příkazy na vzdáleném systému, stejně jako s programem rsh. Na následujícím příkladě si ukážeme, jak spustit příkaz `uptime` na počítači sun, a jak vytvořit adresář se jménem tmp. Výstup programů se zobrazí na terminálu lokálního počítače earth.

```
ssh slunce "uptime; mkdir tmp"
tux@slunce's password:
1:21pm up 2:17, 9 users, load average: 0.15, 0.04, 0.02
```

Uvozovky jsou nezbytné, aby byly obě instrukce zaslány jedním příkazem. Jen tak se druhý příkaz spustí na počítači sun.

### 4.2.3 Bezpečné kopírování pomocí scp

Program scp kopíruje soubory na vzdálený počítač. Je to bezpečná a šifrovaná náhrada za program rcp. Například příkaz `scp dopis.tex sun:` zkopíruje soubor `dopis.tex` z aktuálního adresáře lokálního počítače earth na počítač sun. Pokud máte na počítači sun jiné uživatelské jméno než na počítači earth, zadejte uživatelské jméno pro vzdálený počítač ve formátu `username@host`. Pro tento příkaz neexistuje volba `-l`.

Po zadání správného hesla začne scp přenášet soubor a zobrazuje při tom stav přenosu jako rostoucí řadu hvězdiček. Navíc zobrazuje i odhadovaný čas trvání přenosu. Tyto výstupy můžete vypnout použitím parametru `-q`.

Program scp také zvládá rekursivní kopírování celých adresářů. Příkaz `scp -r src/sun:backup/` zkopíruje obsah adresáře `src/` včetně jeho podadresářů do adresáře `backup/` na počítači sun. Pokud tento adresář neexistuje, scp ho automaticky vytvoří.

Parametrem `-p` řeknete scp, aby neměnil časové údaje u souborů. Volba `-C` zapne kompresi dat při přenosu, takže sníží velikost přenášených dat (zvyší se tím ale zatížení procesoru).

### 4.2.4 Bezpečný přenos souborů pomocí sftp

Program sftp lze použít místo scp pro bezpečný přenos souborů. Během sftp relace můžete používat některé z příkazů známých z ftp. Program sftp se hodí hlavně pro situace, kdy předem neznáte názvy souborů na vzdáleném počítači.

### 4.2.5 SSH démon (sshd) – strana serveru

Pro práci s SSH klienty ssh a scp musí v pozadí běžet SSH server (démon) naslouchající na TCP/IP portu 22. Démon při prvním spuštění generuje tři páry klíčů. Každý pár sestává ze soukromého a veřejného klíče. Proto se jedná o tzv. proceduru založenou na veřejném klíči. Aby byla zaručena bezpečnost komunikace pomocí SSH, musí mít přístup k soukromému klíči pouze administrátor systému. Ve standardní instalaci jsou přístupová práva k souborům podle toho nastavena. Soukromé klíče jsou potřebné pouze lo-

kálně pro démona SSH a nesmíte je nikomu poskytnout. Veřejné části klíče (soubory s příponovou `.pub`) jsou zasílány klientům požadujícím spojení; mohou je číst všichni uživatelé.

Spojení je vždy iniciováno klientem. Čekající démon si s klientem nejdříve vymění identifikační data (zjistí jakou verzi protokolu, případně jaký program a port, používá protější strana). Protože na požadavek odpovídá potomek hlavního procesu démona SSH, může současně běžet více různých SSH spojení.

Pro komunikaci mezi serverem a klientem podporuje program OpenSSH verzi 1 i 2 protokolu SSH. Nově instalovaný systém SUSE Linux používá standardně verzi 2. Pokud chcete u staršího systému po aktualizaci i nadále používat verzi 1, držte se instrukcí popsanych v souboru `/usr/share/doc/packages/openssh/README.SuSE`. V tomto dokumentu také najdete informace o tom, jak v několika krocích přejít z prostředí verze SSH 1 na verzi SSH 2.

Pokud používáte SSH verze 1, zaslá server svůj veřejný klíč stroje a klíč serveru, který je SSH démonem znovu vytvářen každou hodinu. Oba umožňují SSH klientovi zašifrovat libovolně zvolený klíč relace, který je zaslán SSH serveru. SSH klient také serveru oznámí, jaký šifrovací algoritmus používá.

Verze 2 protokolu SSH nevyžaduje klíč serveru. Obě strany používají pro výměnu klíčů algoritmus Diffie-Helman.

Pokud chcete rozšifrovat klíč relace, musíte znát soukromý klíč stroje i serveru, které nelze odvodit z veřejných klíčů. Pouze kontaktovaný SSH démon může rozšifrovat klíč relace pomocí svých soukromých klíčů (více viz `man /usr/share/doc/packages/openssh/RFC.nroff`). Počáteční fázi relace můžete podrobně sledovat, pokud zapnete u klienta SSH tzv. "užvaněný" režim volbou `-v`.

Výchozí je verze 2 SSH protokolu. Verzi 1 můžete vynutit přepínačem `-1`. Klient si po prvním kontaktu se serverem ukládá jeho veřejný klíč stroje do souboru `~/.ssh/known_hosts`. Tak se zabráni útokům cizích serverů s falešnými jmény a IP adresami (tzv. "man-in-the-middle" útok). Takový útok je odhalen buď díky klíči stroje nepřítomnému v `~/.ssh/known_hosts` nebo díky neschopnosti serveru rozšifrovat klíč relace kvůli tomu, že nemá odpovídající soukromé klíče.

Doporučujeme vám zálohovat na bezpečné místo veřejný i soukromý klíč (uloženy jsou v `/etc/ssh/`). Můžete tak odhalit manipulace s klíči, a pokud budete muset reinstalovat systém, můžete opět použít staré klíče. Tak ušetříte uživatele znepokojivých varo-

vání o změně klíče. Pokud se v případě varování o změně klíče ověří, že se skutečně jedná o správný SSH server, musí uživatel odstranit existující záznam o tomto serveru ze souboru `~/.ssh/known_hosts`.

## 4.2.6 Mechanismus ověřování pomocí SSH

Vlastní autentizace, v nejjednodušší formě, sestává z vložení hesla, jak bylo uvedeno výše. Cílem SSH bylo přinést snadno použitelný, ale bezpečný, software. Protože cílem je nahradit `rsh` a `rlogin`, SSH musí poskytovat autentizační metodu vhodnou pro každodenní použití. SSH toho dosahuje pomocí dalšího páru klíčů generovaného uživatelem. Balíček SSH k tomuto účelu obsahuje pomocný program `ssh-keygen`. Příkazem `ssh-keygen -t rsa` nebo `ssh-keygen -t dsa` se vygeneruje pár uživatelských klíčů a uživatel je dotázán, do jakého souboru se mají uložit.

Potvrďte standardní název a odpovězte na žádost o zadání hesla. I když vám program navrhně použít prázdné heslo, je lepší zadat netriviální heslo o délce 10 až 30 znaků. Potvrďte zopakováním hesla. Následně se uloží klíče do souborů, v našem příkladě do `id_rsa` (soukromý) a `id_rsa.pub` (veřejný) a program zobrazí celou cestu k souborům.

Pro změnu hesla u již vygenerovaných klíčů použijte (podle typu vašeho klíče) příkaz `ssh-keygen -p -t rsa` nebo `ssh-keygen -p -t dsa`. Nyní si na vzdáleném počítači, kam se chcete přihlašovat, uložte váš veřejný klíč (`id_rsa.pub`) do souboru `~/.ssh/authorized_keys`. Při přihlášení pak budete dotázáni na heslo ke klíči. Pokud se tak nestane, přezkontrolujte, zda jste vše správně uložili.

Tato procedura může vypadat složitěji, než samotné přihlašování pomocí přihlašovacího jména a hesla. SSH ale nabízí další nástroj, program `ssh-agent`, který si pamatuje privátní klíče během sezení. Celé sezení (X session) se musí spustit jako potomek programu `ssh-agent`. Nejjednodušší cestou je nastavit na začátku konfiguračního souboru `.xsession` proměnnou `usessh` na `yes` a přihlásit se přes KDM nebo XDM. Eventuálně spusťte X Window pomocí příkazu `ssh-agent startx`.

Nyní můžete používat `ssh` nebo `scp` jako obvykle. Pokud jste uložili na vzdálené počítače váš veřejný klíč, nebude po vás systém vyžadovat heslo. Nezapomeňte ale, pokud odejdete od počítače, zamknout váš desktop (např. pomocí `xlock`).

Veškeré změny SSH protokolu 2 oproti dřívější verzi jsou popsány v souboru `/usr/share/doc/packages/openssh/README.SuSE`.

## 4.2.7 X server, ověřování a přeposílací mechanismy

Kromě vylepšení bezpečnostních mechanismů popsaných výše, SSH také zjednodušuje používání vzdálených aplikací pro X server. Jestliže spustíte `ssh` s parametrem `-X`, proměnná `DISPLAY` se na vzdáleném stroji nastaví na hodnotu počítače, odkud se přihlašujete, a veškerý výstup X aplikací bude přeposílán na vzdálený počítač přes existující ssh spojení. Navíc tyto aplikace spuštěné vzdáleně a zobrazované lokálně nemohou být díky přenosu přes ssh odposlechnuty útočníkem.

Pokud při spouštění přidáte parametr `-A`, bude se ssh-agent autentizační mechanismus přenášet i na stroje, na které se připojíte. Můžete se tedy bez zadávání hesel přihlašovat na další počítače. Stačí abyste všude uložili váš veřejný klíč.

Oba tyto mechanismy jsou standardně vypnuty, ale lze je kdykoliv zapnout v systémovém souboru `/etc/ssh/sshd_config` nebo v uživatelském souboru `~/.ssh/config`.

Program `ssh` můžete také použít pro přesměrování TCP/IP spojení. V následujícím příkladě SSH přesměruje SMTP a POP3 port:

```
ssh -L 25:sun:25 earth
```

Tedy každé SMTP spojení, které půjde na port 25 (SMTP) počítače `earth`, je přes šifrovaný kanál přesměrováno na SMTP port počítače `sun`. To se může hodit, pokud nepoužíváte SMTP server s funkcemi SMTP-AUTH nebo POP-before-SMTP. Z jakéhokoliv místa připojeného k síti lze veškerý poštovní provoz přesměrovat na hlavní poštovní server. Stejně tak lze přesměrovat POP3 spojení (port 110) z počítače `earth` na počítač `sun` pomocí příkazu:

```
ssh -L 110:sun:110 earth
```

Oba dva příkazy musíte spustit jako superuživatel `root`, protože jde o přesměrování privilegovaných portů. Elektronická pošta je normálními uživateli odesílána a přijímána pomocí existujícího SSH spojení. SMTP a POP3 host musí být nastaven na `localhost`. Další informace naleznete v manuálových stránkách k jednotlivým programům a v adresáři `/usr/share/doc/packages/openssh`.

## 4.3 Šifrování diskových oddílů a souborů

### 4.3.1 Vhodné nasazení

Každý uživatel má data, u kterých si přeje, aby k nim neměl přístup nikdo jiný. Čím více mobilní jste, tím opatrnější byste měli být při práci s daty. Při přímém nebo síťovém přístupu třetí strany k vašim datům je vždy vhodné řešení šifrování souborů.

---

#### **Varování: Omezená bezpečnost šifrovaného média**

Po připojení šifrovaného média je jeho obsah přístupný všem uživatelům s příslušnými přístupovými právy. Šifrování má význam především v případě krádeže počítače nebo šifrovaného média.

---

V následující části najdete popis nastavení šifrování a jeho možné použití v různých situacích.

#### Notebooky

Pokud pracujete na cestách se svým notebookem nebo ho často převážíte z místa na místo, je vhodné šifrovat diskový oddíl s daty. V případě ztráty nebo krádeže notebooku jsou pak vaše data v bezpečí před nepovolanou osobou.

#### Vyměnitelná média

U USB flash disku nebo externího disku je pravděpodobnost ztráty nebo krádeže mnohem pravděpodobnější než u celého notebooku. V takovém případě šifrování souborů uchrání vaše data před čtením nepovolanými osobami.

#### Pracovní stanice

V případě počítače, ke kterému má přístup prakticky kdokolivěk, je rozumné šifrovat především soubory a datové diskové oddíly.

## 4.3.2 Nastavení šifrovaného souborového systému pomocí YaST

YaST nabízí možnost vytvoření šifrovaných souborů nebo diskových oddílů jak během instalace, tak na již nainstalovaném systému. Šifrované soubory lze bez problémů vytvářet bez ohledu na rozdělení disku. V případě šifrovaného diskového oddílu musíte nejdříve vytvořit příslušný diskový oddíl. Výchozí rozvržení rozdělení disku během instalace neobsahuje žádný šifrovaný diskový oddíl. Šifrovaný diskový oddíl je nutné vytvořit v rozdělování disku pro experty.

### Vytvoření šifrovaného oddílu při instalaci

---

#### Varování: Zadání hesla

Věnujte pozornost zprávám systému o bezpečnosti hesla při zadávání hesla pro šifrovaný diskový oddíl. Heslo si dobře zapamatujte, bez jeho zadání se nedostanete k datům na šifrovaném diskovém oddíle.

---

Vytvoření šifrovaného diskového oddílu najdete v dialogu rozdělování disku programu YaST popsáném v části „Rozdělování disku“ (2 – „*Konfigurace pomocí YaST*“, ↑Uživatelská příručka). Stejně jako při vytváření normálního diskového oddílu klikněte na tlačítko *Vytvořit*. Pak zadejte parametry nového diskového oddílu (formátování a bod připojení). Dále pokračujte kliknutím na *Krypt. souborový systém*. V následujícím dialogu zadejte heslo, které bude vyžadované před připojením šifrovaného diskového oddílu. Nastavení dokončíte kliknutím na tlačítko *OK*. Systém vás požádá před připojením oddílu o zadání hesla pro připojení oddílu.

Pokud nechcete, aby byl šifrovaný diskový oddíl připojený během startu systému, stiskněte místo zadání hesla klávesu . Stejně postupujte u dalších požadavků o zadání hesla pro připojení diskového oddílu. Šifrovaný diskový oddíl nebude připojen a systém bude pokračovat ve startu. Jde o jeden ze způsobů ochrany vašich dat, protože po připojení šifrovaného diskového oddílu je obsah tohoto oddílu přístupný všem uživatelům.

Pokud chcete souborový systém připojovat pouze v případě jeho potřeby, zvolte *Nepřipojovat při spuštění* v dialogu *Volby fstab*. Diskový oddíl pak nebude automaticky připojován během startu systému. Kdykoliv ho pak můžete připojit příkazem `mount jmeno_oddilu bod_pripojeni`. Zadejte heslo. Aby k datům nemohli



přístupovat další uživatelé, po ukončení práce odpojte diskový oddíl příkazem `umount jmeno_oddilu`.

## Vytvoření šifrovaného oddílu na běžícím systému

---

### Varování: Aktivace šifrování na běžícím systému

Šifrovaný diskový oddíl lze vytvořit také v již běžícím systému. Vytvoření šifrovaného oddílu na již existujícím oddílu povede ke ztrátě dat na zvoleném oddílu.

---

Na běžícím systému zvolte v ovládacím centru programu YaST *Systém* → *Rozdělování disku*. Výběr dialogu potvrďte kliknutím na tlačítko *Ano*. Místo tlačítka *Vytvořit* použitého v předcházejícím nastavení klikněte na tlačítko *Editovat*. Další postup je stejný.

## Šifrované soubory

Mimo šifrovaných oddílů je možné v dialogu rozdělování disku vytvářet šifrované soubory. Pod tabulkou diskových oddílů klikněte na tlačítko *Vytvořit šifrovaný soubor* a zvolte *Vytvořit šifrovaný soubor*. Zadejte cestu k souboru spolu s předpokládanou velikostí. Odsouhlaste výchozí nastavení pro formátování a typ souborového systému, zadejte bod připojení a nastavte, zda má být šifrovaný souborový systém připojen během startu systému.

### 4.3.3 Šifrování obsahu vyměnitelného média

Vyměnitelná média jako externí pevné disky a USB flash disky rozpoznává YaST jako normální pevný disk. Je tedy možné na nich šifrovat soubory nebo celé diskové oddíly stejným způsobem uvedeným výše. Protože k jejich připojení dochází obvykle pouze na omezenou dobu při práci, nenastavujte připojení těchto zařízení během startu systému.

## 4.4 Bezpečnost a soukromí

Jednou z hlavních vlastností linuxových a unixových systémů je schopnost obsluhovat více uživatelů najednou (víceuživatelský systém) a umožnit jim současně spouštět více úloh (multitasking). Navíc je tento operační systém síťově transparentní. Uživatelé

často neví, zda data či aplikace, které používají, jsou umístěny lokálně na jejich počítači, nebo v síti.

Multiuživatelská podstata systému vyžaduje možnost oddělení dat jednotlivých uživatelů. Je nutno zajistit soukromí a bezpečí. Bezpečnost dat byla důležitým problémem již před vznikem počítačových sítí. Stejně jako dnes bylo vždy nejdůležitější zajistit bezpečnost dat v případě havárie nebo ztráty paměťového média, např. pevného disku.

Tato část se zabývá především otázkami soukromí, ale je nutno si uvědomit, že každá kvalitní bezpečnostní politika musí pamatovat na pravidelné, funkční a ověřené zálohy dat. Bez nich budete mít problém obnovit data nejen v případě havárie hardwaru, ale také v případě podezření na nedovolenou manipulaci s nimi.

## 4.4.1 Lokální a síťová bezpečnost

Existuje řada způsobů přístupu k datům:

- osobní komunikace s lidmi, kteří mají požadované informace nebo přístup k počítači
- přímý fyzický přístup k počítači
- přes sériovou linku
- přes počítačovou síť

Ve všech těchto případech by se uživatel měl autentizovat dříve, než mu data budou zpřístupněna. Webový server nemusí být chráněn tak přísně, ale stále je nutné zajistit, aby neznámému uživateli neposkytl choulostivá data.

Ve výše uvedeném seznamu je první případ ten, který vyžaduje nejvíce komunikace mezi lidmi, jako např. tehdy, pokud kontaktujete zaměstnance banky a musíte ho přesvědčit, že bankovní účet je skutečně váš. Požádá vás o podpis, PIN nebo heslo, kterým si ověří vaši identitu. V některých případech se může podařit, na základě několika málo známých skutečností a psychologie, získat důvěru informované osoby a postupně z ní vymámit další a další potřebné informace, aniž by si to vůbec uvědomila. Hackeři tuto techniku nazývají *sociální inženýrství*. Proti této technice se můžete zabezpečit jedině vzděláváním a školením svých zaměstnanců v užívání jazyka a komunikaci s lidmi. Před vlastním útokem na počítačové systémy se útočníci často snaží získat zajímavé

informace od recepční, servisních techniků, nebo dokonce od rodinných příslušníků. V mnoha případech je útok založený na sociálním inženýrství odhalen příliš pozdě.

Útočník může použít i tradiční cestu a snažit se dostat přímo k vašemu hardwaru. Proto by počítače měly být chráněny proti nedovolené manipulaci, aby nikdo nepovolný nemohl odstraňovat, vyměňovat nebo poškozovat jejich součásti. To platí i pro zálohy dat, síťové a elektrické kabely. Zabezpečte také startování systému, protože existuje několik dobře známých klávesových kombinací schopných vyvolat neobvyklé chování. Chraňte se použitím hesel pro BIOS i zavaděč systému.

Na mnoha místech se stále používají sériové terminály připojené k sériovým portům. Na rozdíl od síťových rozhraní nezávisí jejich komunikace s počítačem na síťovém protokolu. Používají jednoduchý kabel nebo infračervený paprsek, který přenáší informace v podobě nezašifrovaných znaků. Kabel je nejslabším článkem systému: lze k němu připojit starší tiskárnu a nastavit ji tak, aby tiskla veškerou přenášenou komunikaci. Místo tiskárny lze samozřejmě použít i jiné metody útoku.

Lokální čtení souboru na počítači vyžaduje jiná přístupová pravidla než otevření síťového spojení se serverem. Je rozdíl mezi lokální a síťovou bezpečností. Hranice je tam, kde se data musí balit do paketů, aby byla zaslána na jiné místo.

## Lokální bezpečnost

Lokální bezpečnost závisí na fyzickém prostředí, ve kterém počítač běží. Umístěte stroj v prostředí, které bezpečnostním požadavkům odpovídá. Hlavním cílem lokální bezpečnosti je zajistit, aby byli uživatelé odděleni a nemohli navzájem zneužívat svá práva a identity. To je obecné pravidlo, které je třeba mít na pozoru, ale nejdůležitější je v případě uživatele `root`, který má nad systémem absolutní kontrolu. Může totiž používat identitu kteréhokoli dalšího uživatele, aniž by znal jeho heslo, a číst jakýkoliv lokálně uložený soubor.

## Hesla

Na Linuxu nejsou hesla ukládána jako text a uživatelem vložené heslo není jednoduše porovnáváno s heslem uloženým v systému. Kdyby tomu tak bylo, byly by všechny účty v počítači kompromitovány v okamžiku, kdy by někdo nepovolný získal přístup k patřičnému konfiguračnímu souboru. Místo toho je uložené heslo zašifrované a při každém vložení je zašifrováno znovu – porovnávají se pak dva zašifrované řetězce. V

případě, že zašifrovaná hesla nelze převést zpět do původního tvaru, to významně zvyšuje bezpečnost.

Používá se k tomu speciální jednosměrný algoritmus, tzv. *trapdoor algorithm*. Útočník, i když by získal zašifrované heslo, není schopný algoritmus otočit a získat nezašifrovanou podobu hesla. Musel by testovat všechny možné kombinace písmen a dalších znaků, dokud by nenašel kombinaci, která při zašifrování dává stejný výsledek jako původní heslo. Pokud jsou hesla tvořena osmi znaky, je takových kombinací velmi mnoho.

V sedmdesátých letech se věřilo, že je tato metoda bezpečnější díky relativní pomalosti použitého algoritmu, který k zašifrování jednoho hesla vyžadoval několik sekund. Počítače se však natolik zrychlily, že dnes zvládnou podobných operací za sekundu milióny. Proto zašifrovaná hesla nesmějí být běžným uživatelům viditelná (běžní uživatelé nesmí mít možnost číst soubor `/etc/shadow`). Je také velmi důležité zajistit, aby hesla nebyla snadno uhodnutelná, pro případ že by se tento soubor v důsledku chyby stal viditelným. Není také příliš užitečné měnit heslo typu „tantalize“ na „t@nt@1lz3“.

Záměna některých písmen za podobné znaky není dostatečně bezpečná, protože programy pro odhalování hesel používající slovníky umí provádět i podobné záměny. Lepší je použít slovo bez obecného významu, něco, co dává smysl jen vám osobně. Například první písmena slov nějaké věty nebo názvu knihy, například *Knihá Jméno růže, kterou napsal Umberto Eco* by vedla k bezpečnému heslu *KJrknUE8*. Hesla typu *cernakocka* nebo *zuzana76* může snadno uhádnout i někdo, kdo vás téměř nezná.

## Start systému

Systém nastavte tak, aby nemohl být spuštěn z diskety nebo CD. Buď mechaniky úplně odstraňte, nebo nastavte BIOS tak, aby spouštěl systém výhradně z pevného disku, a zajistěte BIOS heslem. Linux je obvykle spouštěn zavaděčem, který umožňuje jádru předávat různé parametry. Zakažte ostatním tyto parametry nastavením dalšího hesla v souboru `/boot/grub/menu.lst` (viz 9 – „*Starování systému a zavaděče*“ (strana 163)). Je to pro bezpečnost systému velmi důležité, protože jádro samotné běží s pravomocemi uživatele `root` a navíc je to právě jádro, kdo tyto pravomoci dále přiděluje.

## Souborová přístupová práva

Obecným pravidlem je pracovat vždy s nejpřísnějšími možnými nastaveními práv, které umožňují vykonat potřebný úkol. Například pro čtení a psaní pošty rozhodně

nejdou potřeba práva uživatele `root`. Pokud by v poštovním programu byla chyba, mohla by být zneužita k útoku, který by měl přesně ta práva, jako měl program při svém spuštění. Výše zmíněné pravidlo pomáhá minimalizovat škody v podobných případech.

Práva téměř čtvrt miliónu souborů obsažených v systému SUSE Linux jsou velmi pečlivě zvolena. Administrátor by při instalaci dodatečných souborů a programů měl dávat na nastavení práv velký pozor. Zkušení a bezpečnostních pravidel znalí administrátoři vždy používají spolu s příkazem `ls` volbu `-l`, což jim umožňuje okamžitě odhalit špatně nastavená přístupová práva. Špatně nastavená práva souboru mohou vést nejen ke změně či smazání souboru, ale mohou být spuštěny s právy superuživatele, nebo, v případě konfiguračních souborů, programy je mohou použít s právy superuživatele. To významně zvyšuje možnosti útočníka. Tento typ útoku se nazývá "kukaččí vejce", neboť je program spuštěn ("vysezen") jiným uživatelem ("ptákem"), podobně jako když kukačka oklame jiné ptáčky a donutí je tak starat se o svou snůšku.

SUSE Linux obsahuje v adresáři `/etc` soubory `permissions`, `permissions.easy`, `permissions.secure` a `permissions.paranoid`. Smyslem těchto souborů je definovat zvláštní práva, jako adresáře, do kterých může zapisovat kdokoli, nebo, v případě souborů, `setuser` ID bit (programy s nastaveným `setuser` ID bitem neběží pod uživatelem, který je spustil, nýbrž s právy vlastníka souboru, nejčastěji uživatele `root`). Administrátor může přidávat vlastní nastavení do souboru `/etc/permissions.local`.

Který z těchto souborů se bude používat konfiguračními programy nastavíte pomocí *Nastavení bezpečnosti* nástroje YaST. Více se dozvíte v komentářích souboru `/etc/permissions` nebo v manuálové stránce příkazu `chmod`.

## Přetečení zásobníku a chyby typu `format string`

Vždy, když program zpracovává data, která mohla být změněna uživatelem, je třeba být na pozoru. Je to však spíše problém programátorů než běžných uživatelů. Programátor musí zajistit, aby aplikace zpracovávala data správným způsobem, aniž by zapisovala do paměťových oblastí, které jsou pro data příliš malé. Program by měl také předávat data konzistentním způsobem přes k tomu určená rozhraní.

K *přetečení zásobníku* (`buffer overflow`) může dojít tehdy, pokud se při zápisu do paměťového zásobníku nevezme v úvahu jeho velikost. V určitých případech data (vytvořená uživatelem) zabírají více místa, než zásobník obsahuje. Důsledkem je, že jsou zapísána za hranici zásobníku. To může za určitých okolností znamenat vykonání instrukcí zadaných uživatelem (nikoliv programátorem) místo pouhého zpracování dat. Chyba

tohoto typu může mít velmi závažné následky, zvláště pokud je program spuštěn se zvláštními právy (viz „[Souborová přístupová práva](#)“ (strana 100)).

Chyby typu *format string* fungují trošku jinak, ale následky jsou podobné. Ve většině případů se tyto chyby zneužívají v programech, které běží se zvláštními právy (setuid a setgid), což ovšem také znamená, že se můžete chránit odebráním těchto práv. Nejlepší je aplikovat pravidlo o použití nejnižších možných oprávnění (viz „[Souborová přístupová práva](#)“ (strana 100)).

Protože se tyto chyby týkají zpracování uživatelských dat, lze je zneužívat bez přístupu k lokálnímu účtu. Často je lze zneužívat i po síti. Proto jsou důležité z hlediska místní i síťové bezpečnosti.

## Viry

Ačkoliv někteří lidé říkají opak, viry existují i na Linuxu. Nicméně známé linuxové viry jsou pouze pokusné laboratorní exempláře vyvinuté jako důkaz jejich možné existence. V divoké přírodě nikdo nikdy žádné linuxové viry nespasil.

Viry nemohou přežít a šířit se bez hostitele. Takovým hostitelem může být program nebo důležitý datový prostor, např. MBR disku, který musí být pro kód viru zapisovatelný. Vzhledem ke své multiuživatelské podstatě může Linux omezit práva zápisu k určitým souborům, zejména důležitým systémovým souborům. Proto zvyšujete pravděpodobnost napadení virem, pokud provádíte běžnou práci jako uživatel `root`. Naproti tomu, pokud používáte zmíněné pravidlo o nejnižších možných oprávněních, je pravděpodobnost infekce zanedbatelná.

Mimo to byste nikdy neměli bezhlavě spouštět program z neznámého internetového zdroje. SUSE RPM balíčky obsahují digitální podpis potvrzující jejich původ. Virová infekce je typickým příznakem administrátorů a uživatelů s nízkým povědomím o bezpečnosti. Takoví dokáží ohrozit i systém, který byl navržen jako vysoce bezpečný.

Nezaměňujte viry s červy. Červi jsou čistě síťové potvůrky, které nevyžadují pro své šíření hostitele.

## Síťová bezpečnost

Síťová bezpečnost je důležitá pro ochranu proti útokům pocházejícím z vnější. Běžná přihlašovací procedura zahrnující dotaz na uživatelské jméno a heslo je stále místní

bezpečnostní záležitost. V případě přihlašování po síti je nutno rozlišit mezi dvěma bezpečnostními aspekty. To, co se odehrává před vlastním přihlášením, je záležitost síťové bezpečnosti, to co se děje po vlastním přihlášení, je záležitost lokální bezpečnosti.

## X Window System a X autentizace

Jak bylo zmíněno na začátku, je síťová transparentnost jednou z hlavních charakteristik unixových systémů. X, okenní systém unixových systémů, toho umí využívat úžasným způsobem. Při použití X není problém přihlásit se na vzdálený stroj a spustit tam grafický program, jehož výstup je zaslán přes síť zpět k vám a zobrazen na vašem počítači.

Pokud se má X klient vzdáleně zobrazit, musí X server chránit zdroje (obrazovku) před neoprávněným přístupem. Klientská aplikace musí dostat určitá práva. Systém X Window to umí zařídit dvěma způsoby. První se nazývá kontrola přístupu na straně hosta (host-based access control), druhou je kontrola přístupu pomocí cookies (cookie-based access control). První spoléhá na IP adresu počítače, ze kterého běží klient, a je ovládána programem `xhost`. Program `xhost` uloží IP adresu klienta do malé databáze X serveru. Spoléhání na IP adresu však není nijak zvlášť bezpečné. Na počítači navíc může pracovat další uživatel, který může prvnímu uživateli ukrást přístup k X serveru. Z důvodů nízké bezpečnosti zde proto tuto metodu nebudeme popisovat. Pokud se s ní přesto chcete blíže seznámit, najdete informace v manuálové stránce `xhost`.

V případě kontroly pomocí cookies se generuje řetězec, který zná pouze X server a správný uživatel. Jde o něco jako občanský průkaz. Koláček (slovo se nevztahuje k obyčejným koláčkům, ale k čínským koláčkům pro štěstí, na kterých je epigram) je při přihlášení uložen v souboru `.Xauthority` v domovském adresáři uživatele a je dostupný všem X klientům vyžadujícím X server pro zobrazení okna. Soubor `.Xauthority` lze otestovat programem `xauth`. V případě přejmenování souboru `.Xauthority` nebo jeho smazání není možné otevřít žádné nové okno nebo X klienta. Více se o bezpečnostních mechanismech X Window dovíte v manuálové stránce `Xsecurity` (`man Xsecurity`).

SSH (secure shell) lze využít ke kompletnímu šifrování síťového připojení k X serveru, aniž by to uživatel pocítil. Tomuto přeposílání se říká X forwarding. Jde o simulaci X serveru na straně serveru a nastavení příslušné proměnné na straně vzdáleného klienta. Další podrobnosti o SSH najdete v části 4.2 – „SSH: bezpečná práce v síti“ (strana 89).

---

## Varování

Pokud si nejste jistí bezpečností počítače, na kterém pracujete, nepoužívejte X forwarding. Pokud ho přesto aktivujete, může případný útočník například zneužít vaše SSH připojení k napadení X serveru a odposlechu klávesnice.

---

## Přetečení zásobníku a chyby typu "format string"

Jak bylo vysvětleno v části „[Přetečení zásobníku a chyby typu format string](#)“ (strana 101), přetečení zásobníku i chyby typu "format string" jsou záležitostmi místní i síťové bezpečnosti. Stejně jako v případě lokálních útoků, i zde jsou tyto chyby nejčastěji zneužívány k získání pravomocí superuživatele. I když se nepodaří přímo toto, může útočník získat neprivilégovaný lokální účet a ten použít ke zneužívání dalších potenciálních bezpečnostních slabín systému.

Přetečení zásobníku a chyby typu "format string" zneužitelné po síti jsou nepochybně nejčastějším typem vzdálených útoků. Programy zneužívající nově nalezených chyb tohoto typu (tzv. exploitů) se často distribuují v konferencích věnovaných bezpečnosti. Lze je použít bez znalosti vlastního kódu. Během let se ukázalo, že jejich dostupnost vede k bezpečnějším systémům, prostě proto, že výrobci operačních systémů byli donuceni se bezpečností zabývat. V případě svobodného softwaru má ke zdrojovým kódům přístup kdokoli (SUSE Linux je dodáván s kompletními zdrojovými kódy) a tak může kdokoli, kdo našel bezpečnostní chybu, dodat patřičnou záplatu.

## Zahlčení (Denial of Service)

Smyslem útoků typu zahlčení (Denial of Service – DoS) je zablokovat serverový program nebo dokonce celý systém, čehož lze dosáhnout několika způsoby: přetížením serveru, jeho zaměstnáním nesmyslnými pakety nebo zneužitím přetečení zásobníku. DoS útok má často jediný účel – zlikvidovat určitou službu v síti. Zmizení služeb může znamenat další ohrožení, například útoky typu *man-in-the-middle* (odposlech, přebírání TCP spojení, předstírání adresy) či otravu DNS.

## Útoky typu "Man-in-the-Middle"

Každý vzdálený útok, při kterém se útočník vplete *mezi* dva komunikující počítače, se řadí mezi tzv. *man-in-the-middle* útoky. Většina útoků toho to typu má jedno společné – oběť vůbec netuší, že se děje něco zlého. Existuje mnoho různých variant těchto



útoků, útočník například může zachytit požadavek na spojení a přeposlat ho cílovému stroji. Oběť se tak spojí s nežádoucím protějškem, ale nic netuší, protože ten předstírá skutečný cíl spojení.

Nejjednodušší forma takového útoku je tzv. *sniffing*, při kterém útočník jen odposlouchává okolní síťový provoz. Složitější "man-in-the-middle" útok může znamenat převzetí již existujícího spojení (*hijacking*). Aby tak mohl útočník učinit, musí po nějakou dobu analyzovat pakety, aby mohl předpovědět TCP sekvenci daného spojení. V okamžiku, kdy útočník spojení převzme, si oběť problému všimne, protože se její spojení ukončí a dostane chybové hlášení. Skutečnost, že existují protokoly nezabezpečené šifrováním, útočníkům život jenom usnadňuje.

*Spoofing* je útok, při kterém jsou pakety pozměněny, aby obsahovaly falešná data, obvykle IP adresu. Většina aktivních útoků závisí na možnosti zaslání takových falešných paketů, což je něco, co na linuxovém stroji může udělat pouze superuživatel (`root`).

Mnoho zmíněných útoků se kombinuje s útoky typu DoS. Pokud má útočník možnost určitý počítač, byť na krátkou dobu, vyřadit z provozu, usnadňuje to aktivní útok, protože počítač nebude schopný s útokem po určitou dobu interferovat.

## Otrava DNS

Otrava DNS (poisoning) nastává, když útočník pozmění cache DNS serveru [pomocí podvržených DNS paketů], který pak předává informace oběti, která je vyžaduje. Mnoho serverů udržuje s dalšími počítači důvěrné vztahy na základě IP adres nebo jmen. Útočník těmto důvěrným vztahům musí dobře porozumět, aby se mohl vydávat za jeden z důvěryhodných strojů. Obvykle proto útočník analyzuje pakety ze serveru. Současně často musí použít dobře načasovaný DoS útok. Obranou je zde použití šifrovaných spojení a ověřování identity počítačů, s nimiž je navazováno spojení.

## Červi

Červi jsou často zaměňováni s viry, rozdíl mezi nimi je však jasný. Na rozdíl od virů není červí život závislý na hostiteli. Místo toho se specializují na co možná nejrychlejší šíření sítí. Červi, kteří se v minulosti objevili, jako Ramen, Lion či Adore, využívali dobře známých bezpečnostních děr v programech jako `bind8` nebo `lprNG`. Ochrana proti červům je poměrně snadná. Protože mezi objevem bezpečnostní díry a výskytem červů uplyne nějaký čas, může mít svědomitý administrátor dávno instalovány potřebné bezpečnostní opravy.

## 4.4.2 Bezpečnostní tipy a triky

Je velmi důležité být informován o novinkách na poli bezpečnosti a o nejnovějších bezpečnostních problémech. Jedním z nespolehlivějších způsobů ochrany systému je instalace všech aktualizací balíčků, které bezpečnostní zprávy doporučují. Bezpečnostní oznámení pro SUSE jsou publikována v poštovní konferenci, do které se můžete přihlásit na adrese <http://www.novell.com/linux/security/securitysupport.html>. Tato konference ([suse-security-announce@suse.de](mailto:suse-security-announce@suse.de)) je skvělým zdrojem informací o bezpečnostních aktualizacích a mezi její aktivní členy patří řada odborníků z bezpečnostního týmu SUSE.

Poštovní konference [suse-security@suse.de](mailto:suse-security@suse.de) je dobrým místem pro diskusi o bezpečnostních problémech, které vás zajímají. Můžete se do ní přihlásit na výše uvedené webové stránce.

[bugtraq@securityfocus.com](mailto:bugtraq@securityfocus.com) je jedna z nejznámějších bezpečnostních konferencí na světě. Doporučujeme její sledování. Denně je v jejím rámci publikováno přibližně 15 až 20 příspěvků. Více informací najdete na stránce <http://www.securityfocus.com>.

Následující základní bezpečnostní pravidla vám mohou přijít vhod:

- V souladu s pravidlem o použití nejpřísnějších omezení práv, jaká ještě umožňují provést požadovanou úlohu, neprovádějte svou běžnou činnost jako superuživatel `root`. Snížíte tak pravděpodobnost infekce virem nebo kukaččím vejcem. Také ochráníte sami sebe před vlastními chybami.
- Pokud je to možné, pracujte na vzdáleném stroji s využitím šifrovaného spojení. Používání `ssh` (secure shell) místo programů `telnet`, `ftp`, `rsh` a `rlogin` by mělo být samozřejmostí.
- Vyhněte se autentizačním metodám založeným pouze na IP adrese.
- Snažte se udržovat nejdůležitější balíčky spojené se sítí aktuální. Přihlaste se do konferencí, které vás budou informovat o potřebných aktualizacích takových programů (`bind`, `sendmail`, `ssh` atd.). Totéž platí pro programy ovlivňující lokální bezpečnost.
- Optimalizujte soubor `/etc/permissions` podle potřeb vašeho systému. Pokud odeberete programu `setuid` bit, může přestat správně pracovat. Na druhou stranu

přestane být potenciální bezpečnostní dírou do vašeho systému. Podobně je tomu se soubory a adresáři, do kterých může každý zapisovat.

- Zakažte všechny síťové služby, které na serveru nutně nepotřebujete. Tím zvýšíte bezpečnost systému. Porty, na kterých se naslouchá, lze vyhledat programem `netstat`. Používejte ho spolu s následujícími volbami: `netstat -ap` nebo `netstat -anp`. Volba `-p` zobrazí, který proces obsadil který port pod jakým jménem.

Výstup programu `netstat` porovnejte s pečlivým skenem portů provedeným z jiného počítače. Vynikající program k tomuto účelu je `nmap`, který nejen prověří porty na vašem stroji, ale dokáže odhadnout, jaké služby za nimi čekají. Skenování portů však může být považováno za nepřátelský akt, proto nikdy nic takového nedělejte na počítači bez výslovného souhlasu administrátora. Uvědomte si také, že je nutné oskenovat nejen TCP, ale i UDP porty (volby `-sS` a `-sU`).

- Program `tripwire` umožňuje monitorovat integritu souborů na vašem systému. Tento program je součástí distribuce SUSE Linux. Databázi tohoto programu zašifrujte, aby s ní nikdo nepovoláný nemohl provádět psí kusy. Navíc si její kopii uložte mimo počítač na externí datové médium, které není připojené přes síť.
- Mějte se na pozoru při instalaci softwaru třetích stran. Byly případy, kdy útočník vložil trojského koně do balíčku s bezpečnostní aktualizací programu. Naštěstí byl brzy odhalen. Pokud instalujete binární balíček, nesmíte mít nejmenší pochybnosti o jeho původu.

RPM balíčky distribuce SUSE jsou elektronicky podepsány (gpg). Klíč, který SUSE k podepisování používá, je následující:

```
ID:9C800ACA 2000-10-19 SUSE Package Signing Key <build@suse.de>
```

```
Key fingerprint = 79C1 79B2 E1C8 20C1 890F 9994 A84E DAE8 9C80 0ACA
```

Příkazem `rpm --checksig balicek.rpm` můžete zkontrolovat kontrolní součet a podpis nenainstalovaného balíčku. Klíč naleznete na prvním instalačním CD i na většině světových klíčových serverů.

- Pravidelně kontrolujte zálohy vašich uživatelských i systémových souborů. Pokud nemáte zálohy ověřené, mohou být nepoužitelné a bezcenné.

- Kontrolujte záznamy v protokolových souborech (logy). Pokud je to možné, napište si malý skript hledající podezřelé záznamy. Není to jednoduchá úloha, ale jen vy můžete vědět, které záznamy jsou ve vašem systému podezřelé.
- Pomocí `tcp_wrapper` omezte přístup ke službám běžícím na vašem stroji, takže získáte kontrolu nad tím, kterým IP adresám je povolen přístup ke službě. Více informací o tomto nástroji najdete v manuálových stránkách `tcpd` a `hosts_access` (`man 8 tcpd` a `man hosts_access`).
- Pro zvýšení bezpečnosti `tcpd` (`tcp_wrapper`) použijte `SUSEfirewall`.
- Bezpečnostní opatření by měla být vícenásobná: dvakrát zobrazená zpráva je lepší než žádná zpráva.

## 4.4.3 Ústřední adresa pro hlášení bezpečnostních problémů

Pokud objevíte bezpečnostní problém (nejprve prosím zkontrolujte dostupné aktualizace), napište e-mail na adresu [security@suse.de](mailto:security@suse.de). Přiložte prosím podrobný popis problému a číslo verze postiženého balíčku. SUSE vám odpoví co nejrychleji. Doporučujeme poštu šifrovat pomocí `pgp`. Klíč SUSE je:

```
ID:3D25D3D9 1999-03-06 SUSE Security Team <security@suse.de>  
Key fingerprint = 73 5F 2E 99 DF DB 94 C4 8F 5A A3 AE AF 22 F2 D5
```

Tento klíč si můžete stáhnout i ze stránky <http://www.novell.com/linux/security/securitysupport.html>.

# ACLs v Linuxu

V této kapitole je popsáno pozadí a funkce POSIX ACLs pro linuxové souborové systémy. Zároveň zde získáte informace o používání a výhodách ACLs (*Access Control Lists*).

## 5.1 Výhody ACLs

V tradičním linuxovém systému má každý objekt tři typy přístupových práv. Jde o práva ke čtení (r, zápisu w a vykonání x) pro každý ze tří typů uživatelů (vlastníka, skupinu a ostatní). Navíc lze nastavit *user id*, *group id* a *sticky* bit.

Toto pojetí je zcela dostačující v naprosté většině situací. Ve velmi rozsáhlých systémech a zvláštních typech aplikací však naráží na řadu limitů.

ACLs vznikly právě proto, aby tyto situace ošetřily rozšířením tradičního pojetí přístupových práv o další vlastnosti. Pomocí ACLs je možné nastavit přístupová práva pouze pro určité uživatele nebo skupiny, kteří nejsou vlastníky objektu ani nepatří do příslušné skupiny. Access Control Lists jsou součástí jádra a mají podporu v souborových systémech ReiserFS, Ext2, Ext3, JFS a XFS. Díky ACLs můžete nastavovat přístupová práva, aniž byste museli zároveň zasahovat do celého systému přístupových práv.

Výhody ACLs si uvědomíte především při náhradě serveru s Windows za server s Linuxem. Řada stanic v síti může pracovat se systémem Windows i po migraci a systém Linux bude těmto stanicím poskytovat tiskové a souborové služby pomocí Samby.

Díky podpoře ACLs v Sambě lze práva nastavit jak na linuxovém serveru tak na stanicích Windows (pouze Windows NT a vyšší). Pomocí programu winbindd lze nastavovat práva uživatelů, kteří existují pouze na straně Windows a na linuxovém serveru nemají účet. Access Control Lists je nastaven pomocí `getfacl` a `setfacl` pouze na straně serveru.

## 5.2 Definice

### Třídy uživatelů

Tradiční koncept POSIX používá v souborovém systému tři *třídy* přístupových práv. Vlastníka, skupinu vlastníka a ostatní. Pro každou z těchto tří tříd lze nastavit bity dávající práva ke čtení (r,zápisu w a vykonáníx).

### Přístupové ACLs

Přístupová práva skupin a uživatelů jsou pro všechny typy objektů souborového systému (soubory a adresáře) omezeny přístupovými ACLs.

### Výchozí ACL

Výchozí ACLs se nastavuje pouze u adresářů. Omezuje nastavení přístupových práv u nově vytvářených podadresářů a souborů.

### Položka ACL

Každý ACLs se skládá ze skupiny položek. ACLs položky se skládají z typu (viz. tabulka 5.1 – „[Typy ACL položek](#)“ (strana 111)), ukazatelem na skupinu nebo uživatele a nastavením práv. Pro některé typy položek musí být ukazatel na skupinu nebo uživatele prázdný.

## 5.3 Používání ACLs

V následující části si na příkladech ukážeme používání ACLs a jejich interakci s tradičním systémem přístupových práv. Popíšeme postup pro vytvoření vlastních ACLs a také syntaxi ACLs.

ACLs dělíme na dva základní typy. *Minimální* ACLs obsahují položku pro typ uživatele (owner), skupinu vlastníka (owner group) a ostatní (other) s konvenčními přístupovými bity pro soubory a adresáře. *Rozšířené* ACLs jde ještě dál. Musí obsahovat nastavení položky *mask* a musí obsahovat více položek pro typy *named user* a *named group*. V

tabulce 5.1 – „Typy ACL položek“ (strana 111) najdete přehled různých typů možných ACLs položek.

**Tabulka 5.1** *Typy ACL položek*

Typ	Zápis
owner	user::rwx
named user	user:name:rwx
owning group	group::rwx
named group	group:name:rwx
mask	mask::rwx
other	other::rwx

Práva definována v položce *owner* a *other* jsou vždy platná. S vyjímkou položky *mask* všechny ostatní položky (*named user*, *owning group*, a *named group*) mohou být neaktivní nebo maskované. Platné jsou v případě, že jsou součástí jak určité položky, tak masky. Pokud jsou pouze součástí masky, jsou neaktivní. Tento mechanismus je demonstrován v tabulce 5.2 – „Maskování práv“ (strana 111).

**Tabulka 5.2** *Maskování práv*

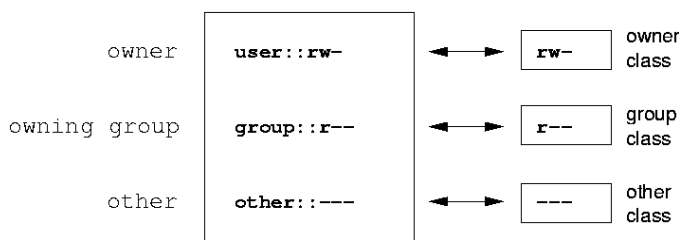
Typ položky	Zápis	Práva
named user	user:jane:r-x	r-x
mask	mask::rw-	rw-
	effective permissions:	r--

## 5.3.1 ACL položky a přístupové bity

V systému s ACLs existují minimální a rozšířené ACLs, první jsou znázorněny na obrázku 5.1 – „Minimální ACL: ACL úpoložky porovnávány podle přístupového bitu“ (strana 112), druhé na 5.2 – „Rozšířené ACL: ACL položky porovnávány podle přístupového bitu“ (strana 112). V následujících příkladech si ukážeme dva případy minimálních a rozšířených ACLs.

V obou případech jsou práva *třídy owner* mapována na ACL položky *owner*. Stejně tak jsou na příslušnou položku mapována také práva *třídu other*. V obou případech je však jiné mapování na *třidu group*.

**Obrázek 5.1** Minimální ACL: ACL úpoložky porovnávány podle přístupového bitu



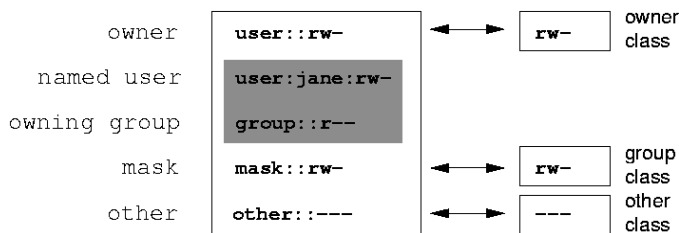
V případě minimálních ACLs *bez masky*

jsou práva *třídy group* mapována na ACLs položku *owning group*.

V případě rozšířených ACLs *s maskou*

jsou práva *třídy group* mapována na položku *mask*.

**Obrázek 5.2** Rozšířené ACL: ACL položky porovnávány podle přístupového bitu





Mapování zajišťuje hladký chod aplikací s podporou ACLs spolu s aplikacemi bez této podpory. Práva zde nezmíněná buď nejsou nastavena pomocí ACLs nebo jsou neaktivní. Pokud dojde ke změně přístupových bitů, dojde ke změně ACLs a vice versa.

## 5.3.2 Adresář s ACL přístupem

Princip přístupu ACLs je znázorněn v následujícím příkladě:

- Vytvoření objektu souborového systému (v našem případě adresáře)
  - Změna ACL
  - Maskování
1. Před vytvořením adresáře použijte příkaz `umask` k nastavení výchozích práv:

```
umask 027
```

Příkaz `umask 027` nastaví výchozí přístupová práva tak, že vlastníkoví dá všechna práva (0, skupině zakáže zápis 2 a ostatním nedá práva žádná 7). `umask` zároveň maskuje všechny přístupové bity a deaktivuje je. Více informací o tomto příkazu získáte z jeho manuálových stránek (`man umask`).

Zdejte příkaz `mkdir`. Výsledkem je vytvoření adresáře `mydir` s přístupovými právy nastavenými prostřednictvím `umask`. Následujícím příkazem překontrolujete, zda jsou práva nastavena správně:

```
ls -dl mydir
drwxr-x- ... tux project3 ... mydir
```

2. Zjistěte počáteční nastavení ACL a vložte nové hodnoty pro uživatele a skupiny.

```
getfacl mydir
user::rwx
group::r-x
other::---
```

Výstup příkazu `getfacl` velmi jasně ukazuje nastavení bitů a ACL položek popsaných v části 5.3.1 – „ACL položky a přístupové bity“ (strana 112). První tři řádky zobrazují jméno adresáře, vlastníka a jeho skupinu. Následující tři

řádky obsahují ACL položky *owner*, *owning group* a *other*. V tomto případě má adresář minimální ACL nastavení a pomocí příkazu `getfacl` jsme získali stejný výpis jako v případě použití prostého `ls`.

V první změně ACL přidáme práva pro čtení, zápis a vykonání pro dalšího uživatele se jménem `jane` a další skupiny `djungle`.

```
setfacl -m
user:jane:rwx,group:djungle:rwx mydir
```

Parametrem `-m` příkazu `setfacl` říkáme, že má změnit ACLs. Parametr je následován hodnotami (jednotlivé položky jsou odděleny dvojtečkami). Poslední částí příkazu je jméno adresáře, na který se mají změny aplikovat.

Příkazem `getfacl` si můžete nechat vypsát výsledné nastavení ACLs.

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:jane:rwx
group::r-x
group:djungle:rwx
mask::rwx
other:---
```

Jako další nastavení pro uživatele `jane` a skupinu `djungle` byla vytvořena položka *mask*. Tato položka automaticky redukuje všechny položky v *třídě group* na společný základ.

*Maska* definuje maximální efektivní přístupová práva pro všechny položky v *třídě group*. To obsahuje *named user*, *named group* a *owning group*. Přístupové bity *třídě group* lze zobrazit příkazem `ls -dl mydir`.

```
ls -dl mydir
drwxrwx- ... tux project3 ... mydir
```

První sloupec mimo obvyklého výstupu obsahuje také `+`, který indikuje existenci *rozšířených* ACLs.

- Podle výstupu příkazu `ls` obsahuje položka *mask* práva k zápisu. V tradičním pojetí by to znamenalo, že má *vlastnická skupina* (zde `project3`) také práva zápisu do adresáře `mydir`. Přístupová práva *vlastnické skupiny* však souhlasí s

nastavením v *mask*, které jsou v našem příkladě *r-x* (viz. tabulka 5.2 – „Maskování práv“ (strana 111)). Dodatečné nastavení tak nebude mít na dosavadní nastavení žádný vliv.

Editujte položku *mask* příkazem `setfacl` nebo `chmod`.

```
chmod g-w mydir
```

```
ls -dl mydir
```

```
drwxr-x---+ ... tux project3 ... mydir
```

```
getfacl mydir
```

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:jane:rwx          # effective: r-x
group::r-x
group:djungle:rwx     # effective: r-x
mask::r-x
other::---
```

Po vykonání příkazu `chmod` bude odstraněn bit pro zápis z *třídy group* a výstup příkazu `ls` ukazuje, že musí být změněn i bity masky. Práva zápisu jsou opět omezeny pouze na vlastníka adresáře `mydir`. Výstup příkazu `getfacl` tuto skutečnost potvrzuje. Výstup obsahuje komentář pro všechny položky, kde přístupové bity nesouhlasí s originálním nastavením, protože jsou filtrovány pomocí položky *mask*. Původní nastavení lze kdykoliv vrátit příkazem `chmod`:

```
chmod g+w mydir
```

```
ls -dl mydir
```

```
drwxrwx---+ ... tux project3 ... mydir
```

```
getfacl mydir
```

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:jane:rwx
group::r-x
group:djungle:rwx
mask::rwx
other::---
```

## 5.3.3 Adresář s výchozími ACL

Adresáře mohou mít zvláštní typ ACL tzv. výchozí ACL. Výchozí ACL nastavuje přístupová práva ke všem podřízeným adresářům s nastavenými výchozími ACL. Výchozí ACL se nastavuje přístupové ACL jak u adresářů tak v nich obsažených souborech.

### Vliv výchozích ACL

S výchozím ACL je pracováno různě podle toho, na jaký typ objektu je uplatňován:

- ACL podadresáře se skládá z výchozího ACL, jeho vlastního výchozího ACL a přístupového ACL adresáře.
- Přístupová práva souboru se skládají z jeho vlastních ACL a výchozího ACL.

Všechny objekty souborového systému používají při nastavení přístupových práv parametr `mode`, který definuje přístupová práva nově vytvářených objektů.

- Pokud rodičovský adresář nemá nastavené výchozí ACL, nastaví se přístupové bity podle hodnoty parametru `mode` příkazu `umask`.
- Pokud má rodičovský adresář nastavené výchozí ACL, nově vytvářený objekt převezme přístupová práva od parametru `mode` a z výchozího ACL. `Umask` je ignorován.

### Aplikace výchozích ACLs

Následující tři kroky ilustrují operace pro adresáře a výchozí ACLs:

- vytvoření výchozího ACL pro aktuální existující adresář
- Vytvoření podadresáře v adresáři s nastavených výchozím ACL
- Vytvoření souboru v adresáři s výchozím ACL

1. Vložení výchozí ACLs do existujícího adresáře `mydir`:

```
setfacl -d -m group:djungle:r-x mydir
```

Parametr `-d` příkazu `setfacl` zajistí změny (parametr `-m`) ve výchozím ACLs.

Podívejme se blíže na výstup příkazu `getfacl mydir`:

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:jane:rwx
group::r-x
group:djungle:rwx
mask::rwx
other:---
default:user::rwx
default:group::r-x
default:group:djungle:r-x
default:mask::r-x
default:other:---
```

`getfacl` vrátí jak přístupová ACL tak výchozí ACL. Výchozí ACL je tvořeno řádkami začínajícími na `default`. Po nastavení výchozího ACL příkazem `setfacl` pro skupinu `djungle` příkaz `setfacl` automaticky překopíruje všechny ostatní položky k nastavení platného výchozího ACL. Nastavení výchozího ACL nebude mít na existující objekty žádný okamžitý vliv. Ovlivňovat bude pouze nově vytvářené objekty po nastavení výchozího ACL. Tyto nové objekty budou mít přístupová práva skládající se pouze z výchozího ACL rodičovského adresáře.

2. Nyní použijte příkaz `mkdir` k vytvoření podadresáře v adresáři `mydir`, který bude mít stejné ACLs.

```
mkdir mydir/mysubdir
```

```
getfacl mydir/mysubdir
```

```
# file: mydir/mysubdir
# owner: tux
# group: project3
user::rwx
group::r-x
group:djungle:r-x
mask::r-x
other:---
default:user::rwx
default:group::r-x
default:group:djungle:r-x
default:mask::r-x
default:other:---
```

Jak jsme očekávali, nově vytvořený podadresář `mysubdir` má přístupová práva rodičovského adresáře. Nastavení přístupových práv `mysubdir` je stejné jako `mydir`.

3. Použití příkazu `touch` k vytvoření souboru v adresáři `mydir`:

```
touch mydir/myfile
```

```
ls -l mydir/myfile
```

```
-rw-r-----+ ... tux project3 ... mydir/myfile
```

```
getfacl mydir/myfile
```

```
# file: mydir/myfile
# owner: tux
# group: project3
user::rw-
group::r-x      # effective:r--
group:djungle:r-x # effective:r--
mask::r--
other::---
```

Důležitým je v tomto příkladě příkaz `touch` s režimem s hodnotou `0666`, což znamená, že nově vytvářené soubory mají nastaveno právo pro čtení a zápis pro všechny třídy uživatelů a *umask* ani ACLs nenastavují žádná další omezení (viz. „[Vliv výchozích ACL](#)“ (strana 116)).

V důsledku to znamená, že všechna přístupová práva neobsažená v režimu hodnoty jsou odstraněny z ACLs položky. Přestože nebyla z ACLs třídy *group* odstraněna žádná práva, položka *mask* byla změněna k maskování jiným způsobem než s nastaveným režimem.

Tato vlastnost zajišťuje bezchybnou funkci ACLs aplikací např. kompilátorů. Můžete tak vytvářet souboru s omezenými přístupovými právy a zároveň je označit jako vykonatelné. Pomocí *mask* mechanismu zajistí, že k nim budou mít práva pouze ti správní uživatelé a skupiny.

## 5.3.4 ACL kontrolní algoritmus

Všechny procesy a aplikace projdou před tím, než je jim povolen přístup k objektům chráněným ACLs kontrolním algoritmem. ACLs jsou testovány na následující sekvence:

*owner*, *named user*, *owning group* nebo *named group* a *other*. Přístup je pak řízen s nejlépeším výsledkem ve prospěch procesu. Sekvence nelze slučovat.

Tento algoritmus je samozřejmě mnohem komplikovanější, pokud objekt patří do více skupin s různými vlastnostmi. V takovém případě algoritmus náhodně vybere ze skupin, které mají požadované vlastnosti. Je jedno, jaká z položek bude vést k výsledku *access granted*. Pokud algoritmus nenajde žádnou vhodnou skupinu, výsledkem bude *access denied*.

## 5.4 Výhledy

Jak bylo napsáno výše, ACLs umožňuje mnohem podrobnější nastavení přístupových práv. ACLs lze v případě potřeb kombinovat se starým konceptem nastavení přístupových práv. Některé důležité aplikace však stále ACLs nepodporují. Mimo programu star například stále není k dispozici zálohovací program s plnou podporou ACLs.

Základní příkazy (`cp`, `mv`, `ls` atd.) ACLs podporují, ale řada editorů a správců souborů na (např. Konqueror). Při kopírování souborů v Konqueroru dojde ke ztrátě jejich ACLs. Při změně v editorech jsou někdy ACLs zachovány, jindy ne. Důvodem je různý zálohovací režim editorů. Možnosti jsou tyto:

- Pokud editor zapisuje změny do originálního souboru, jsou ACLs zachovány.
- Pokud editor vytváří nový soubor s pozměněným obsahem starého souboru a pak provádí přejmenování na původní jméno, dojde ke ztrátě ACLs bez ohledu na to, zda editor ACLs podporuje.

Aplikací s podporou ACL se objevuje stále více, takže se dá předpokládat, že Linux dokáže plně využít této funkce již v nejbližší době.

## 5.5 Další informace

Detailní informace o ACLs získáte na následujících stránkách [http://sdb.suse.de/en/sdb/html/81\\_acl.html](http://sdb.suse.de/en/sdb/html/81_acl.html), <http://acl.bestbits.at/> a v manálových stránkách příkazů `getfacl`, `acl(5)` a `setfacl(1)`





# Nástroje monitorování systému

Aktuální stav systému lze zjistit pomocí mnoha různých nástrojů. Najdete zde také nástroje potřebné pro každodenní práci včetně jejich nejdůležitějších parametrů.

U každého příkazu je současně uveden také příklad výstupu. Na první řádce příkladu je vždy příkaz (po znaku dolaru). Komentáře jsou uzavřeny v závorkách [ . . . ]. U dlouhých řádek, pokud je to potřeba, je zalomení. Zalomení dlouhých řádek se provádí pomocí znaku zpětného lomítka (\).

```
$ command -x -y
output line 1
output line 2
output line 3 is annoyingly long, so long that \
    we have to break it
output line 3
[...]
```

Popis každého z nástrojů je pouze stručný, aby bylo možné zmínit co největší množství užitečných příkazů. Podrobnější informace o každém příkazu najdete v jeho manuálové stránce. U většiny příkazů lze také použít krátkou nápovědu zadáním parametru `--help`.

## 6.1 Seznam otevřených souborů: lsof

Seznam všech souborů otevřených procesem s ID *PID* získáte zadáním parametru `-p`. Například všechny soubory otevřené aktuálním shellem zjistíte příkazem:

```

$ lsof -p $$
COMMAND PID USER  FD  TYPE DEVICE      SIZE      NODE NAME
zsh      4694  jj   cwd   DIR   0,18      144 25487368 /suse/jj/t
(totan:/real-home/jj)
zsh      4694  jj   rtd   DIR   3,2       608      2 /
zsh      4694  jj   txt   REG   3,2      441296   20414 /bin/zsh
zsh      4694  jj   mem   REG   3,2     104484   10882 /lib/ld-2.3.3.so
zsh      4694  jj   mem   REG   3,2     11648   20610
/usr/lib/zsh/4.2.0/zsh/rlimits.so
[...]
zsh      4694  jj   mem   REG   3,2     13647   10891 /lib/libdl.so.2
zsh      4694  jj   mem   REG   3,2     88036   10894 /lib/libnsl.so.1
zsh      4694  jj   mem   REG   3,2    316410  147725 /lib/libncurses.so.5.4
zsh      4694  jj   mem   REG   3,2    170563  10909 /lib/tls/libm.so.6
zsh      4694  jj   mem   REG   3,2   1349081  10908 /lib/tls/libc.so.6
zsh      4694  jj   mem   REG   3,2      56     12410
/usr/lib/locale/de_DE.utf8/LC_TELEPHONE
[...]
zsh      4694  jj   mem   REG   3,2      59     14393
/usr/lib/locale/en_US/LC_NUMERIC
zsh      4694  jj   mem   REG   3,2   178476   14565
/usr/lib/locale/en_US/LC_CTYPE
zsh      4694  jj   mem   REG   3,2   56444   20598
/usr/lib/zsh/4.2.0/zsh/computil.so
zsh      4694  jj   0u    CHR 136,48      50 /dev/pts/48
zsh      4694  jj   1u    CHR 136,48      50 /dev/pts/48
zsh      4694  jj   2u    CHR 136,48      50 /dev/pts/48
zsh      4694  jj   10u   CHR 136,48      50 /dev/pts/48

```

Ve výše uvedeném příkladu byla použita proměnná shellu \$\$, kde \$\$ vrací ID aktuálního shellu.

Bez parametru vypíše příkaz `lsof` všechny otevřené soubory. Obvykle jde o velmi velké množství souborů. Jejich počet zjistíte příkazem:

```

$ lsof | wc -l
3749

```

Seznam používaných znakových zařízení:

```

$ lsof | grep CHR
sshd     4685   root  mem   CHR   1,5      45833 /dev/zero
sshd     4685   root  mem   CHR   1,5      45833 /dev/zero
sshd     4693   jj    mem   CHR   1,5      45833 /dev/zero
sshd     4693   jj    mem   CHR   1,5      45833 /dev/zero
zsh      4694   jj    0u    CHR 136,48      50 /dev/pts/48
zsh      4694   jj    1u    CHR 136,48      50 /dev/pts/48
zsh      4694   jj    2u    CHR 136,48      50 /dev/pts/48
zsh      4694   jj    10u   CHR 136,48      50 /dev/pts/48
X        6476   root  mem   CHR   1,1      38042 /dev/mem
lsof     13478  jj    0u    CHR 136,48      50 /dev/pts/48
lsof     13478  jj    2u    CHR 136,48      50 /dev/pts/48

```

```
grep      13480      jj      1u      CHR 136,48      50 /dev/pts/48
grep      13480      jj      2u      CHR 136,48      50 /dev/pts/48
```

## 6.2 Přístup uživatelů k souborům: fuser

Předpokládejme, že chcete odpojit souborový systém připojený k /mnt:

```
$ mount -l | grep /mnt
/dev/sda on /mnt type ext2 (rw,noexec,nosuid,nodev,noatime,user=jj)
```

Pokus o odpojení selže:

```
$ umount /mnt
umount: /mnt: device is busy
```

Proces, který k adresáři /mnt přistupuje, zjistíte příkazem:

```
$ fuser -v /mnt/*

                USER          PID ACCESS COMMAND
/mnt/notes.txt
                jj            26597 f....  less
```

Po ukončení procesu less spuštěného z jiného terminálu půjde souborový systém bez problémů odpojit.

## 6.3 Vlastnosti souboru: stat

Příkazem stat zobrazíte vlastnosti souboru:

```
$ stat xml-doc.txt
  File: `xml-doc.txt'
  Size: 632          Blocks: 8          IO Block: 4096   regular file
Device: eh/14d Inode: 5938009      Links: 1
Access: (0644/-rw-r--r--)  Uid: (11994/      jj)   Gid: ( 50/      suse)
Access: 2004-04-27 20:08:58.000000000 +0200
Modify: 2003-06-03 15:29:34.000000000 +0200
Change: 2003-07-23 17:48:27.000000000 +0200
```

Pomocí parametru --filesystem získáte podrobnosti o souborovém systému, jehož je soubor součástí:

```
$ stat . --filesystem
File: "."
  ID: 0      Namelen: 255      Type: ext2/ext3
Blocks: Total: 19347388  Free: 17831731  Available: 16848938  Size: 4096
Inodes: Total: 9830400  Free: 9663967
```

Pokud používáte z shell (zsh), musíte zadat `/usr/bin/stat`, protože z shell obsahuje zabudovaný příkaz `stat` s jinými parametry a jiným typem výstupu:

```
% type stat
stat is a shell builtin
% stat .
device 769
inode 4554808
mode 16877
nlink 12
uid 11994
gid 50
rdev 0
size 4096
atime 1091536882
mtime 1091535740
ctime 1091535740
blksize 4096
blocks 8
link
```

## 6.4 USB zařízení: `lsusb`

Příkazem `lsusb` získáte výpis všech připojených USB zařízení. S parametrem `-v` bude výpis podrobnější. Program načítá informace z adresáře `/proc/bus/usb/`. V následujícím příkladu si můžete prohlédnout výpis příkazu `lsusb` po připojení flash disku. Zařízení je vypsáno na poslední řádce.

```
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
Bus 001 Device 018: ID 0402:5634 ALi Corp.
```

## 6.5 SCSI zařízení: `scsiinfo`

Příkazem `scsiinfo` můžete získat výpis všech připojených SCSI zařízení. Všechna SCSI zařízení vypíšete přidáním parametru `-l` (podobné informace můžete získat také s pomocí příkazu `lsscsi`). V následujícím příkladu výstupu `scsiinfo -i /dev/sda` můžete vidět informace o disku `/dev/sda`.

```
Inquiry command
-----
Relative Address          0
Wide bus 32              0
Wide bus 16              1
Synchronous neg.        1
Linked Commands          1
Command Queueing        1
SftRe                    0
Device Type              0
Peripheral Qualifier     0
Removable?              0
Device Type Modifier    0
ISO Version              0
ECMA Version             0
ANSI Version             3
AENC                    0
TrmIOP                  0
Response Data Format     2
Vendor:                  FUJITSU
Product:                 MAS3367NP
Revision level:         0104A0K7P43002BE
```

Podrobnější informace získáte zadáním parametru `-a`. Ve výstupu je pak vypsán také seznam chyb na disku, který obsahuje dvě tabulky chybných bloků: první je tabulka od výrobce (manufacturer table) a druhá obsahuje chyby, ke kterým došlo během používání disku (grown table). Pokud se počet položek v druhé tabulce zvyšuje, je čas uvažovat o výměně disku.

## 6.6 Procesy: `top`

Příkaz `top` zobrazí každé dvě sekundy obnovovaný seznam procesů. Program ukončíte stisknutím klávesy `q`. Pokud chcete program automaticky ukončit po zobrazení prvního seznamu, spusťte ho s parametrem `-n 1`:

```
$ top -n 1
top - 14:19:53 up 62 days,  3:35, 14 users,  load average: 0.01, 0.02, 0.00
Tasks: 102 total,   7 running, 93 sleeping,   0 stopped,   2 zombie
```

```
Cpu(s):  0.3% user,  0.1% system,  0.0% nice,  99.6% idle
Mem:    514736k total,  497232k used,    17504k free,    56024k buffers
Swap:   1794736k total,  104544k used,  1690192k free,   235872k cached
```

```

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM   TIME+  Command
 1426 root        15   0  116m  41m  18m  S   1.0   8.2  82:30.34 X
20836 jj          15   0   820   820  612  R   1.0   0.2   0:00.03 top
   1 root        15   0   100    96   72  S   0.0   0.0   0:08.43 init
   2 root        15   0     0     0     0  S   0.0   0.0   0:04.96 keventd
   3 root        34  19     0     0     0  S   0.0   0.0   0:00.99 ksoftirqd_CPU0
   4 root        15   0     0     0     0  S   0.0   0.0   0:33.63 kswapd
   5 root        15   0     0     0     0  S   0.0   0.0   0:00.71 bdflush
    [...]
 1362 root        15   0   488   452  404  S   0.0   0.1   0:00.02 nscd
 1363 root        15   0   488   452  404  S   0.0   0.1   0:00.04 nscd
 1377 root        17   0    56    4     4  S   0.0   0.0   0:00.00 mingetty
 1379 root        18   0    56    4     4  S   0.0   0.0   0:00.01 mingetty
 1380 root        18   0    56    4     4  S   0.0   0.0   0:00.01 mingetty

```

Stisknutí klávesy **F** během běhu příkazu `top` vstoupíte do nabídky umožňující změnu formátu výstupu.

Zadáním parametru `-U UID` a uživatelského jména, získáte seznam procesů zadaného uživatele. `UID` je ID uživatele. Následující příkaz vypíše `UID` uživatele zadaného uživatelského jména a jeho procesy:

```
$ top -U $(id -u UzivatelскеJmeno)
```

## 6.7 Seznam procesů: `ps`

Zadáním příkazu `ps` získáte seznam procesů. S parametrem `r` omezíte výpis pouze na aktuální procesy využívající počítačový čas:

```
$ ps r
  PID TTY          STAT TIME COMMAND
 22163 pts/7      R      0:01 -zsh
  3396 pts/3      R      0:03 emacs new-madeoc.txt
 20027 pts/7      R      0:25 emacs xml/common/utilities.xml
 20974 pts/7      R      0:01 emacs jj.xml
 27454 pts/7      R      0:00 ps r
```

Tento parametr se zadává *bez* minus před písmenem. Některé příkazy se někdy píšou s minus a někdy bez. Správný zápis obvykle najdete v manuálové stránce. Návod vypsany příkazem `ps --help` bývá obvykle velmi stručný.

Počet běžících příkazů např. `emacs` zjistíte příkazem:

```
$ ps x | grep emacs
1288 ? S 0:07 emacs
3396 pts/3 S 0:04 emacs new-makedoc.txt
3475 ? S 0:03 emacs .Xresources
20027 pts/7 S 0:40 emacs xml/common/utilities.xml
20974 pts/7 S 0:02 emacs jj.xml
```

```
$ pidof emacs
20974 20027 3475 3396 1288
```

Parametr `-p` seřadí procesy podle ID:

```
$ ps www -p $(pidof xterm)
  PID TTY          STAT       TIME COMMAND
  9025 ?            S          0:01 xterm -g 100x45+0+200
  9176 ?            S          0:00 xterm -g 100x45+0+200
29854 ?            S          0:21 xterm -g 100x75+20+0 -fn \
-B&H-LucidaTypewriter-Medium-R-Normal-Sans-12-120-75-75-M-70-iso10646-1
  4378 ?            S          0:01 xterm -bg MistyRose1 -T root -n root -e su -l
25543 ?            S          0:02 xterm -g 100x45+0+200
22161 ?            R          0:14 xterm -g 100x45+0+200
16832 ?            S          0:01 xterm -bg MistyRose1 -T root -n root -e su -l
16912 ?            S          0:00 xterm -g 100x45+0+200
17861 ?            S          0:00 xterm -bg DarkSeaGreen1 -g 120x45+40+300
19930 ?            S          0:13 xterm -bg LightCyan
21686 ?            S          0:04 xterm -g 100x45+0+200 -fn \
lucidasanstypewriter-12
23104 ?            S          0:00 xterm -g 100x45+0+200
26547 ?            S          0:00 xterm -g 100x45+0+200
```

Seznam procesů můžete naformátovat podle vlastních potřeb. Seznam všech možností získáte příkazem `-L`. Podle využití paměti procesy seřadíte příkazem:

```
$ ps ax --format pid,rss,cmd --sort rss
  PID  RSS CMD
    2     0 [ksoftirqd/0]
    3     0 [events/0]
   17     0 [kblockd/0]
[... ]
10164 5260 xterm
31110 5300 xterm
17010 5356 xterm
 3896 29292 /usr/X11R6/bin/X -nolisten tcp -br vt7 -auth
/var/lib/xdm/authdir/au
```

## 6.8 Strom procesů: `pstree`

Příkaz `pstree` zobrazí běžící procesy ve stromovém výpisu:

```

$ pstree
init--atd
  |-3*[automount]
  |-bdflush
  |-cron
  [...]
  |-usb-storage-1
  |-usb-storage-2
  |-10*[xterm---zsh]
  |-xterm---zsh---mutt
  |-2*[xterm---su---zsh]
  |-xterm---zsh---ssh
  |-xterm---zsh---pstree
  |-ypbind---ypbind---2*[ypbind]
  `--zsh---startx---xinit4--X
                                     `--ctwm--xclock
                                         |-xload
                                         `--xosview.bin

```

Parametrem `-p` získáte ke jménům procesů také jejich ID. S parametrem `-a` vypíše příkaz také parametry příkazů:

```

$ pstree -pa
init,1
  |-atd,1255
  [...]
  `--zsh,1404
      `--startx,1407 /usr/X11R6/bin/startx
          `--xinit4,1419 /suse/jj/.xinitrc [...]
              |-X,1426 :0 -auth /suse/jj/.Xauthority
              `--ctwm,1440
                  |-xclock,1449 -d -geometry -0+0 -bg grey
                  |-xload,1450 -scale 2
                  `--xosview.bin,1451 +net -bat +net

```

## 6.9 Kdo co dělá: w

Příkazem `w` zjistíte uživatele přihlášené na počítači a jejich činnosti. Například:

```

$ w
15:17:26 up 62 days,  4:33, 14 users,  load average: 0.00, 0.04, 0.01
USER      TTY      LOGIN@  IDLE   JCPU   PCPU   WHAT
jj        pts/0    30Mar04  4days 0.50s  0.54s xterm -e su -l
jj        pts/1    23Mar04  5days 0.20s  0.20s -zsh
jj        pts/2    23Mar04  5days 1.28s  1.28s -zsh
jj        pts/3    23Mar04  3:28m  3.21s  0.50s -zsh
[...]
jj        pts/7    07Apr04  0.00s  9.02s  0.01s w
jj        pts/9    25Mar04  3:24m  7.70s  7.38s mutt

```



```
[...]
jj      pts/14    12:49    37:34    0.20s  0.13s  ssh totan
```

Podle poslední řádky je uživatel `jj` k počítači `totan` připojen pomocí secure shellu (`ssh`). U vzdáleně připojených uživatelů a jiných systémů získáte informace o vzdáleném počítači parametrem `-f`.

## 6.10 Využití paměti: `free`

Nástrojem `free` zjistíte využití RAM. Zobrazeny jsou jak informace o využití paměti, tak o volné paměti (a swapu):

```
$ free
```

	total	used	free	shared	buffers	cached
Mem:	514736	273964	240772	0	35920	42328
-/+ buffers/cache:		195716	319020			
Swap:	1794736	104096	1690640			

Údaje v MB získáte zadáním parametru `-m`:

```
$ free -m
```

	total	used	free	shared	buffers	cached
Mem:	502	267	235	0	35	41
-/+ buffers/cache:		191	311			
Swap:	1752	101	1651			

Následující řádka obsahuje skutečně zajímavé informace:

```
-/+ buffers/cache:      191      311
```

Jde o paměť zásobníků a vyrovnávací paměti. Parametrem `-d n` zadáte, aby došlo k obnovení výpisu každých `n` sekund. Například `free -d 1.5` obnoví výpis každé 1,5 sekundy.

## 6.11 Systémové hlášení jádra: `dmesg`

Linuxové jádro uchovává systémová hlášení v paměti omezené velikosti (standardně 2 na 14 B). Tato hlášení zobrazíte příkazem `dmesg`:

```
$ dmesg
[...]
sdc : READ CAPACITY failed.
sdc : status = 1, message = 00, host = 0, driver = 08
Info fld=0xa00 (nonstd), Current sd00:00: sense key Not Ready
```

```

sdc : block size assumed to be 512 bytes, disk size 1GB.
sdc: test WP failed, assume Write Enabled
sdc: I/O error: dev 08:20, sector 0
I/O error: dev 08:20, sector 0
I/O error: dev 08:20, sector 2097144
I/O error: dev 08:20, sector 2097144
I/O error: dev 08:20, sector 0
I/O error: dev 08:20, sector 0
unable to read partition table
I/O error: dev 08:20, sector 0
nfs: server totan not responding, still trying
nfs: server totan OK

```

Poslední řádka indikuje dočasné problémy s NFS serverem totan. Řádky před ní jsou spojeny se zasunutím USB flash disku.

Starší události najdete v souborech /var/log/messages a /var/log/warn.

## 6.12 Souborový systém a jeho využití: mount, df a du

Příkaz mount souborový systém (zařízení a typ) a jeho body připojení:

```

$ mount
/dev/hdb2 on / type ext2 (rw)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/hda1 on /data type ext2 (rw)
shmfs on /dev/shm type shm (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
automount(pid1012) on /suse type autofs \
  (rw,fd=5,pgrp=1012,minproto=2,maxproto=3)
totan:/real-home/jj on /suse/jj type nfs \
  (rw,nosuid,rsize=8192,wsize=8192,hard,intr,nolock,addr=10.10.0.1)

```

Informaci o využití místa získáte příkazem df. S parametrem -h (nebo --human-readable) získáte výstup v uživatelsky přívětivém formátu.

```

$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hdb2       7.4G  5.1G  2.0G  73% /
/dev/hda1       74G   5.8G   65G   9% /data
shmfs           252M   0    252M   0% /dev/shm
totan:/real-home/jj 350G  324G   27G  93% /suse/jj

```

Uživatelé NFS serveru `totan` by měli neodkladně promazat své domovské adresáře. Celkovou velikost všech souborů a podadresářů vypisuje příkaz `du`. S parametrem `-s` vypíše pouze celkovou velikost bez dalších detailů. Parametr `-h` povede k uživatelsky přívětivému výstupu. Zadáním příkazu:

```
$ du -sh ~
361M    /suse/jj
```

získáte velikost svého domovského adresáře.

## 6.13 Souborový systém `/proc`

V adresáři `/proc` se nachází pseudo souborový systém, do kterého jádro ve formě virtuálních souborů ukládá důležité informace. Například k informacím o typu procesoru můžete přistoupit příkazem:

```
$ cat /proc/cpuinfo
processor       : 0
vendor_id     : AuthenticAMD
cpu family    : 6
model         : 8
model name    : AMD Athlon(tm) XP 2400+
stepping      : 1
cpu MHz       : 2009.343
cache size    : 256 KB
fdiv_bug      : no
[...]
```

Využití přerušení zjistíte příkazem:

```
$ cat /proc/interrupts
          CPU0
 0: 537544462          XT-PIC timer
 1:  820082           XT-PIC keyboard
 2:         0          XT-PIC cascade
 8:         2          XT-PIC rtc
 9:         0          XT-PIC acpi
10:    13970          XT-PIC usb-uhci, usb-uhci
11: 146467509          XT-PIC ehci_hcd, usb-uhci, eth0
12:  8061393          XT-PIC PS/2 Mouse
14: 2465743           XT-PIC ide0
15:   1355            XT-PIC ide1
NMI:         0
LOC:         0
ERR:         0
MIS:         0
```

Některé důležité soubory a jejich obsah:

/proc/devices  
dostupná zařízení

/proc/modules  
zavedené moduly jádra

/proc/cmdline  
příkazová řádka jádra

/proc/meminfo  
podrobné informace o využití paměti

/proc/config.gz  
gzip archiv s konfigurací běžícího jádra

Další informace najdete v souboru /usr/src/linux/Documentation/filesystems/proc.txt. Informace o běžících procesech najdete v adresáři /proc/*NNN*, kde *NNN* je ID (PID) příslušného procesu. Proces a jeho částečnou charakteristiku najdete v /proc/self/:

```
$ ls -l /proc/self
lrwxrwxrwx 1 root root 64 Apr 29 13:52 /proc/self -> 27585
```

```
$ ls -l /proc/self/
total 0
dr-xr-xr-x  2 jj suse 0 Apr 29 13:52 attr
-r-----  1 jj suse 0 Apr 29 13:52 auxv
-r--r--r--  1 jj suse 0 Apr 29 13:52 cmdline
lrwxrwxrwx  1 jj suse 0 Apr 29 13:52 cwd -> /suse/jj/t
-r--r--r--  1 jj suse 0 Apr 29 13:52 delay
-r-----  1 jj suse 0 Apr 29 13:52 environ
lrwxrwxrwx  1 jj suse 0 Apr 29 13:52 exe -> /bin/ls
dr-x-----  2 jj suse 0 Apr 29 13:52 fd
-rw-----  1 jj suse 0 Apr 29 13:52 mapped_base
-r--r--r--  1 jj suse 0 Apr 29 13:52 maps
-rw-----  1 jj suse 0 Apr 29 13:52 mem
-r--r--r--  1 jj suse 0 Apr 29 13:52 mounts
lrwxrwxrwx  1 jj suse 0 Apr 29 13:52 root -> /
-r--r--r--  1 jj suse 0 Apr 29 13:52 stat
-r--r--r--  1 jj suse 0 Apr 29 13:52 statm
-r--r--r--  1 jj suse 0 Apr 29 13:52 status
dr-xr-xr-x  3 jj suse 0 Apr 29 13:52 task
-r--r--r--  1 jj suse 0 Apr 29 13:52 wchan
```

Adresy spustitelných adres a knihoven jsou v souboru maps:

```

$ cat /proc/self/maps
08048000-0804c000 r-xp 00000000 03:02 22890      /bin/cat
0804c000-0804d000 rw-p 00003000 03:02 22890      /bin/cat
0804d000-0806e000 rwxp 0804d000 00:00 0
40000000-40016000 r-xp 00000000 03:02 10882     /lib/ld-2.3.3.so
40016000-40017000 rw-p 00015000 03:02 10882     /lib/ld-2.3.3.so
40017000-40018000 rw-p 40017000 00:00 0
4002b000-40135000 r-xp 00000000 03:02 10908     /lib/tls/libc.so.6
40135000-4013d000 rw-p 0010a000 03:02 10908     /lib/tls/libc.so.6
4013d000-40141000 rw-p 4013d000 00:00 0
bffffe000-c00000000 rw-p bffffe000 00:00 0
fffffe000-fffff000 ---p 00000000 00:00 0

```

## 6.14 vmstat, iostat a mpstat

Nástroje `vmstat` slouží ke zjištění informací o virtuální paměti. Údaje získávají ze souborů `/proc/meminfo`, `/proc/stat` a `/proc/*/stat`. Jde o velmi užitečné nástroje při zjišťování slabín ve výkonu počítače.

S pomocí příkazu `iostat` můžete získat informace o procesoru, I/O zařízeních a diskových oddílech. Údaje jsou čteny z `/proc/stat` a `/proc/partitions`. Výstup může být velmi užitečný např. při ladění zátěže vstupních a výstupních operací mezi disky. Příkaz `mpstat` vypisuje statistiky související s CPU.

## 6.15 procinfo

Souhrn všech důležitých informací a systému `/proc` získáte příkazem `procinfo`:

```

$ procinfo
Linux 2.6.4-54.5-default (geeko@buildhost) (gcc 3.3.3 ) #1 1CPU [roth.suse.de]

Memory:      Total      Used      Free      Shared    Buffers
Mem:         516696    513200    3496      0         43284
Swap:        530136    1352     528784

Bootup: Wed Jul 7 14:29:08 2004   Load average: 0.07 0.04 0.01 1/126 5302

user  :      2:42:28.08   1.3% page in :      0
nice  :      0:31:57.13   0.2% page out:      0
system:  0:38:32.23     0.3% swap in :      0
idle  :    3d 19:26:05.93 97.7% swap out:      0
uptime:  4d 0:22:25.84   context :207939498

```

```

irq 0: 776561217 timer                irq 8:          2 rtc
irq 1:  276048 i8042                  irq 9:        24300 VIA8233
irq 2:          0 cascade [4]         irq 11: 38610118 acpi, eth0, uhci_hcd
irq 3:          3                    irq 12:  3435071 i8042
irq 4:          3                    irq 14:  2236471 ide0
irq 6:          2                    irq 15:        251 ide1

```

Po zadání parametru `-a` vypíše příkaz všechny informace. S parametrem `-nN` bude výpis obnovován každých `N` sekund. program ukončíte stisknutím klávesy `Q`.

Ve výchozím nastavení jsou zobrazeny hodnoty kumulativně. Parametr `-d` povede k výpisu změněných hodnot. Příkazem `procinfo -dn5` získáte hodnoty změněné za posledních 5 sekund:

```

Memory:      Total      Used      Free      Shared      Buffers      Cached
Mem:         0          2        -2         0          0
0
Swap:         0          0          0

Bootup: Wed Feb 25 09:44:17 2004   Load average: 0.00 0.00 0.00 1/106 31902

user  :      0:00:00.02   0.4% page in :      0 disk 1:      0r
0w
nice  :      0:00:00.00   0.0% page out:      0 disk 2:      0r
0w
system: 0:00:00.00   0.0% swap in :      0 disk 3:      0r
0w
idle  :      0:00:04.99 99.6% swap out:      0 disk 4:      0r
0w
uptime: 64d 3:59:12.62      context :    1087

irq 0:      501 timer                irq 10:          0 usb-uhci, usb-uhci
irq 1:          1 keyboard          irq 11:         32 ehci_hcd, usb-uhci,
irq 2:          0 cascade [4]         irq 12:        132 PS/2 Mouse
irq 6:          0                    irq 14:          0 ide0
irq 8:          0 rtc                irq 15:          0 ide1
irq 9:          0 acpi

```

## 6.16 PCI zdroje: lspci

Příkaz `lspci` vypíše PCI zdroje:

```
$ lspci
00:00.0 Host bridge: VIA Technologies, Inc. \
  VT8366/A/7 [Apollo KT266/A/333]
00:01.0 PCI bridge: VIA Technologies, Inc. \
  VT8366/A/7 [Apollo KT266/A/333 AGP]
00:0b.0 Ethernet controller: Digital Equipment Corporation \
  DECchip 21140 [FasterNet] (rev 22)
00:10.0 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.1 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.2 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.3 USB Controller: VIA Technologies, Inc. USB 2.0 (rev 82)
00:11.0 ISA bridge: VIA Technologies, Inc. VT8235 ISA Bridge
00:11.1 IDE interface: VIA Technologies, Inc. VT82C586/B/686A/B \
  PIPC Bus Master IDE (rev 06)
00:11.5 Multimedia audio controller: VIA Technologies, Inc. \
  VT8233 AC97 Audio Controller (rev 50)
01:00.0 VGA compatible controller: Matrox Graphics, Inc. \
  MGA G550 AGP (rev 01)
```

Podrobnější výpis získáte zadáním parametru `-v`:

```
$ lspci -v
[...]
01:00.0 \
  VGA compatible controller: Matrox Graphics, Inc. MGA G550 AGP (rev 01) \
  (prog-if 00 [VGA])
  Subsystem: Matrox Graphics, Inc. Millennium G550 Dual Head DDR 32Mb
  Flags: bus master, medium devsel, latency 32, IRQ 10
  Memory at d8000000 (32-bit, prefetchable) [size=32M]
  Memory at da000000 (32-bit, non-prefetchable) [size=16K]
  Memory at db000000 (32-bit, non-prefetchable) [size=8M]
  Expansion ROM at <unassigned> [disabled] [size=128K]
  Capabilities: <available only to root>
```

Informace o jménech zařízení jsou uložena v souboru `/usr/share/pci.ids`. PCI ID neobsažené v tomto souboru jsou označena jako **Unknown device**.

Parametr `-vv` povede k vypísání všech dostupných informací. Čistě numerické hodnoty získáte zadáním parametru `-n`.

# 6.17 Systémová volání běžícího programu: strace

Nástroj `strace` umožňuje zjistit všechna systémová volání běžících procesů:

```
$ strace -e open ls

execve("/bin/ls", ["ls"], [/* 88 vars */]) = 0
uname({sys="Linux", node="edison", ...}) = 0
brk(0) = 0x805b000
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
\
    = 0x40017000
open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=76333, ...}) = 0
old_mmap(NULL, 76333, PROT_READ, MAP_PRIVATE, 3, 0) = 0x40018000
[...]
ioctl(1, SNDCTL_TMR_TIMEBASE or TCGETS, {B38400 opost isig icanon echo ...})
    = 0
ioctl(1, TIOCGWINSZ, {ws_row=53, ws_col=110, ws_xpixel=897, ws_ypixel=693})
    = 0
open(".", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 3
fstat64(3, {st_mode=S_IFDIR|0755, st_size=144, ...}) = 0
fcntl64(3, F_SETFD, FD_CLOEXEC) = 0
getdents64(3, /* 5 entries */, 4096) = 160
getdents64(3, /* 0 entries */, 4096) = 0
close(3) = 0
fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 48), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
    = 0x40018000
write(1, "ltrace-ls.txt myfile.txt strac"..., 41) = 41
munmap(0x40018000, 4096) = 0
exit_group(0) = ?
```

Pro vypísání všech pokusů o otevření určitého souboru (např. `myfile.txt`) stačí napsat:

```
$ strace -e open ls myfile.txt

open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
open("/lib/tls/librt.so.1", O_RDONLY) = 3
open("/lib/libacl.so.1", O_RDONLY) = 3
open("/lib/libselinux.so.1", O_RDONLY) = 3
open("/lib/tls/libc.so.6", O_RDONLY) = 3
open("/lib/tls/libpthread.so.0", O_RDONLY) = 3
open("/lib/libattr.so.1", O_RDONLY) = 3
open("/proc/mounts", O_RDONLY) = 3
[...]
```



```
open("/proc/filesystems", O_RDONLY) = 3
open("/proc/self/attr/current", O_RDONLY) = 4
```

K výpisu potomků určitého procesu použijte parametr `-f`. Chování i výstup příkazu lze ovlivnit. Podrobnější informace získáte v manuálové stránce `man strace`.

## 6.18 Volání knihoven běžícím příkazem: `ltrace`

Příkazem `ltrace` získáte výpis všech volání knihoven procesu. Příkaz je používán podobně jako `strace`. Zadáním parametru `-c` získáte počet a trvání volání knihoven:

```
$ ltrace -c find /usr/share/doc
% time      seconds  usecs/call   calls   errors syscall
-----
 86.27     1.071814      30    35327          write
 10.15     0.126092      38    3297          getdents64
  2.33     0.028931       3   10208          lstat64
  0.55     0.006861       2    3122          1 chdir
  0.39     0.004890       3    1567          2 open
[... ]
  0.00     0.000003       3         1          uname
  0.00     0.000001       1         1          time
-----
100.00     1.242403          58269          3 total
```

## 6.19 Zjištění vyžadovaných knihoven: `ldd`

Pomocí příkazu `ldd` zjistíte jaké dynamické knihovny vyžaduje určitá dynamicky linkovaná aplikace. Pro příkaz `ls` bude výstup vypadat takto:

```
$ ldd /bin/ls
linux-gate.so.1 => (0xffffe000)
librt.so.1 => /lib/tls/librt.so.1 (0x4002b000)
libacl.so.1 => /lib/libacl.so.1 (0x40033000)
libseline.so.1 => /lib/libselinux.so.1 (0x40039000)
libc.so.6 => /lib/tls/libc.so.6 (0x40048000)
libpthread.so.0 => /lib/tls/libpthread.so.0 (0x4015d000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
libattr.so.1 => /lib/libattr.so.1 (0x4016d000)
```

Statically linkované aplikace nevyžadují žádné dynamické knihovny:

```
$ ldd /bin/sash
        not a dynamic executable
$ file /bin/sash
/bin/sash: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), \
for GNU/Linux 2.2.5, statically linked, stripped
```

## 6.20 Dodatečné informace o ELF binárních souborech

Obsah spustitelných binárních souborů lze číst pomocí nástroje `readelf`. Funguje také pro ELF soubory vytvořené pro jinou hardwarovou architekturu:

```
$ readelf --file-header /bin/ls
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00
  Class:                   ELF32
  Data:                     2's complement, little endian
  Version:                  1 (current)
  OS/ABI:                   UNIX - System V
  ABI Version:              0
  Type:                     EXEC (Executable file)
  Machine:                  Intel 80386
  Version:                  0x1
  Entry point address:      0x8049b40
  Start of program headers: 52 (bytes into file)
  Start of section headers: 76192 (bytes into file)
  Flags:                    0x0
  Size of this header:      52 (bytes)
  Size of program headers:  32 (bytes)
  Number of program headers: 9
  Size of section headers:  40 (bytes)
  Number of section headers: 29
  Section header string table index: 26
```

## 6.21 Meziprocesová komunikace: `ipcs`

Příkazem `ipcs` získáte seznam používaných IPC zdrojů:

```

$ ipcs
----- Shared Memory Segments -----
key          shmid      owner      perms      bytes      nattch     status
0x000027d9  5734403   toms       660        64528      2
0x00000000  5767172   toms       666        37044      2
0x00000000  5799941   toms       666        37044      2

----- Semaphore Arrays -----
key          semid      owner      perms      nsems
0x000027d9  0          toms       660        1

----- Message Queues -----
key          msqid      owner      perms      used-bytes  messages

```

## 6.22 Měření času: `time`

Čas potřebný pro vykonání určitého příkazu lze zjistit pomocí příkazu `time`. Tento příkaz je dostupný ve dvou variantách buď jako zabudovaný příkaz shellu nebo jako program (`/usr/bin/time`).

```

$ time find . > /dev/null

real    0m4.051s
user    0m0.042s
sys     0m0.205s

```



## **Část 3. Systém**



# 32- a 64-bitové aplikace v 64-bitovém prostředí

SUSE Linux je dostupný pro několik 64-bitových platforem. To však nutně neznamená, že všechny v distribuci obsažené aplikace byly portovány na 64 bitů. SUSE Linux podporuje spouštění 32-bitových aplikací v 64-bitovém prostředí. Tato kapitola nabízí základní přehled o podpoře 32-bitových aplikací na 64-bitových SUSE Linux platformách. Vysvětluje, jak se 32-bitové aplikace spouští (podpora běhu) a jak by měly být 32-bitové aplikace kompilovány, aby mohly běžet ve 32- i 64-bitovém prostředí. Obsahuje také informace o API jádra a vysvětlení toho, jak mohou 32-bitové aplikace běžet pod 64-bitovým jádrem.

SUSE Linux pro 64-bitové platformy AMD64 a EM64T je navržen tak, že existující 32-bitové aplikace v 64-bitovém rozhraní běží bez problémů přímo po instalaci. Díky této podpoře můžete používat své oblíbené 32-bitové aplikace bez čekání na jejich 64-bitové verze.

## 7.1 Podpora běhu aplikací

---

### Důležité: Konflikty mezi verzemi aplikací

Pokud je aplikace dostupná pro 32- i 64-bitové prostředí, vede současná instalace obou verzí obvykle k problémům. Rozhodněte se pouze pro jednu verzi a tu nainstalujte a používejte.

---

Pro správné spuštění potřebují aplikace řadu různých knihoven. Bohužel jsou jména 32- i 64-bitových knihoven totožná. Musí být proto rozlišeny jinak než jménem.

Pro zachování kompatibility s 32-bitovou verzí jsou všechny knihovny uloženy na stejných místech jako ve verzi 32-bitové. 32-bitová verze knihovny `libc.so.6` je dostupná na cestě `/lib/libc.so.6` ve 32- i 64-bitovém prostředí.

Všechny 64-bitové knihovny a objektové soubory jsou uloženy v adresářích pojmenovaných `lib64`. 64-bitový objektový soubor, který byste normálně očekávali v adresářích `/lib`, `/usr/lib` a `/usr/X11R6/lib` naleznete v adresářích `/lib64`, `/usr/lib64` a `/usr/X11R6/lib64`. Pro 32-bitové knihovny tak zůstává místo v adresářích `/lib`, `/usr/lib` a `/usr/X11R6/lib`. Jména obou verzí knihoven tak mohou zůstat totožná.

Podadresáře objektových adresářů jejichž obsah není závislý na velikosti slova jsou stále na stejných místech. Například X11 fonty jsou stále na obvyklém místě v adresáři `/usr/X11R6/lib/X11/fonts`. To odpovídá standardům LSB (Linux Standards Base) a FHS (File System Hierarchy Standard).

## 7.2 Vývoj softwaru

Cross kompilační vývojářské nástroje umožňují vytvářet 32- i 64-bitové objekty. Výchozí je kompilace 64-bitových objektů. Pomocí zvláštních prepínačů lze kompilovat i 32-bitové objekty. Pro GCC se používá prepínač `-m32`.

Všechny hlavičkové soubory musí být napsány v podobě nezávislé na architektuře. Nainstalované 32- a 64-bitové knihovny musí mít API (programovací aplikační rozhraní) odpovídající nainstalovaným hlavičkovým souborům. Běžné SUSE prostředí tomuto požadavku odpovídá. Pokud jste ručně aktualizovali knihovny, vyřešte si problémy sami.

## 7.3 Kompilace softwaru pro jinou platformu

Pro vývoj binárních souborů pro jinou platformu je nutné nainstalovat příslušné knihovny pro tuto druhou platformu. Tyto balíčky se nazývají `jmenorpm-32bit`. Můžete také potřebovat hlavičky a knihovny z balíčků `jmenorpm-devel` a vývojové knihovny pro druhou platformu z `jmenorpm-devel-32bit`.



Většina opensource programů používá konfiguraci založenou na `autoconf`. Chcete-li použít `autoconf` pro konfiguraci programu pro druhou architekturu, přepište normální nastavení spuštěním skriptu `configure` s přidánými proměnnými prostředí.

Následující příklad se vztahuje k AMD64 či EM64T systému s x86 jako druhou architekturou:

1. Nastavte `autoconf` k použití 32-bitového kompilátoru:

```
CC="gcc -m32"
```

2. Příklad linkeru zpracovávat 32-bitové objekty:

```
LD="ld -m elf64_i386"
```

3. Nastavte assembler, aby vytvářel 32-bitové objekty:

```
AS="gcc -c -m32"
```

4. Určete, že knihovny pro `libtool` atd. jsou v `/usr/lib`:

```
LDFLAGS="-L/usr/lib"
```

5. Určete, že jsou knihovny uloženy v podadresáři `lib`:

```
--libdir=/usr/lib
```

6. Určete, že jsou používány 32-bitové X knihovny:

```
--x-libraries=/usr/X11R6/lib/
```

Ne všechny proměnné jsou potřeba pro každý program. Upravte je podle potřeby.

```
CC="gcc -m64"          \  
LDFLAGS="-L/usr/lib64;" \  
    .configure        \  
        --prefix=/usr  \  
        --libdir=/usr/lib64  
make  
make install
```

## 7.4 Specifikace jádra

64-bitová jádra pro AMD64 a EM64T poskytují 64- i 32-bitové jaderné ABI (binární aplikační rozhraní). To 32-bitové je identické s ABI odpovídajícího 32-bitového jádra. To znamená, že může 32-bitová aplikace komunikovat s 64-bitovým jádrem stejně, jako by komunikovala s 32-bitovým jádrem.

32-bitová emulace systémových volání pro 64-bitové jádro nepodporuje řadu API používaných systémovými programy. Závisí to na konkrétní platformě. Z tohoto důvodu musí být některé aplikace, jako např. `lspci` nebo programy pro administraci LVM, zkompileovány jako 64-bitové programy, aby správně fungovaly.

64-bitové jádro umí nahrát pouze 64-bitové jaderné moduly zkompileované přímo pro toto jádro. Nelze použít 32-bitové jaderné moduly.

---

### Tip

Některé aplikace vyžadují zvláštní moduly nahrávané jádrem. Pokud potřebujete použít takovou 32-bitovou aplikaci na 64-bitovém systému, kontaktujte výrobce aplikace a SUSE, abyste se ujistili, že je dostupná 64-bitová verze modulu a 32-bitová kompilovaná verze jaderného API pro tento modul.

---

# Startování

Startování a inicializace unixového systému bývají oříškem i pro zkušeného administrátora. Tato kapitola přináší stručný úvod do velmi komplexní. Najdete zde také informace o úrovních běhu a systémové konfiguraci v `sysconfig`.

## 8.1 Startovací proces v Linuxu

proces startování Linuxu se skládá z několika úrovní, ve kterých se spouští různé procesy. Následující část pojednává o startovacím procesu a nejdůležitějších komponentech.

### 1 BIOS

První věc, která se stane po zapnutí počítače je, že BIOS (Basic Input Output System) převezme řízení, nastaví obrazovku a klávesnici na počáteční hodnoty, a otestuje paměť. V této chvíli systém ještě neví o žádných ukládacích či externích zařízeních. Poté systém načte z paměti CMOS (kde je uloženo nastavení BIOSu) současný čas a datum, a informace o nejdůležitějších periferních zařízeních. Po načtení CMOS by měl BIOS rozeznat první pevný disk včetně informací o jeho geometrii. Poté může z tohoto disku začít zavádět operační systém (dále jen OS).

### 2 Zavaděč

Nejdříve se nahraje počátečních 512 bytů z prvního segmentu pevného disku do paměti a spustí se kód, který je uložen na začátku tohoto segmentu. Tento kód, zavaděč, začne nahrávat zbytek operačního systému. Proto se tomuto segmentu

disku obvykle říká `Master Boot Record (MBR)`. Více informací o zavaděči najdete v kapitole 9 – „[Starování systému a zavaděče](#)“ (strana 163).

### 3 *Jádro a initrd*

Po systémové kontrole zavaděč nahraje jádro a ramdisk (`initrd`) do paměti. Linuxové jádro umožňuje zavedení malého souborového systému do paměti, ve kterém se zajistí před připojením kořenového souboru spuštění několika programů. Pak dojde k rozbalení `initrd` a jeho připojení ve formě dočasněho kořenového souborového systému. `initrd` obsahuje minimální linuxový systém s programem `linuxrc`, který je spuštěn ještě před připojením skutečného kořenového systému. Po úspěšném dokončení běhu `linuxrc` jádro, pokud je to možné, uvolní paměť zabranou `initrd` a spustí `init`. Více informací o `initrd` najdete v části 8.1.1 – „[initrd](#)“ (strana 148).

### 4 *linuxrc*

tento program provádí všechny akce potřebné ke správnému připojení kořenového souborového systému jako např. zavedení správného modulu souborového systému a ovladačů pro diskové zařízení. Po úspěšném připojení kořenového souborového systému se `linuxrc` ukončí a jádro spouští program `init`. Více informací o `linuxrc` najdete v části 8.1.2 – „[linuxrc](#)“ (strana 149).

### 5 *init*

`init` se stará o proces spuštění na několika úrovních. Popis `init` najdete v části 8.2 – „[Program init](#)“ (strana 150).

## 8.1.1 `initrd`

`initrd` je malý (obvykle komprimovaný) souborový systém, který se zavádí do ramdisku jako dočasný kořenový systém. Obsahuje minimální linuxové prostředí umožňující vykonání programů před připojením skutečného kořenového systému. Zavádí se přímo do paměti a nemá jiné hardwarové nároky než dostatečnou velikost paměti. `initrd` vždy spouští `linuxrc`, který by měl být ukončen bez chybového návratového kódu.

Ještě před připojením kořenového souborového systému a spuštěním operačního systému potřebuje jádro zavést moduly. Může jít o moduly ovladačů diskových zařízení nebo třeba o moduly s podporou síťových souborových systémů (viz [Správa síťového nastavení](#) (strana 149)). Moduly potřebné pro připojení kořenového souborového systému

by také měl zavádět `linuxrc`. Aby vše proběhlo úspěšně, musí jádro obsahovat kód, který mu umožní číst souborový systém `initrd`.

Vytvořte `initrd` skriptem `mkinitrd`. V systému SUSE Linux se zaváděné `initrd` zadávají do proměnné `INITRD_MODULES` v souboru `/etc/sysconfig/kernel`. Po instalaci se proměnná automaticky nastaví na správnou hodnotu (`linuxrc` uloží instalační nastavení). Moduly se pak zavádějí v pořadí, v jakém jsou zadány v `INITRD_MODULES`. To je důležité především u systémů, kde je vyžadováno několik SCSI ovladačů současně. Změna pořadí modulů by vedla ke změně jmen disků. Nastaveny by tedy měly být pouze ovladače, které jsou potřebné pro připojení kořenového souborového systému. Při instalaci se však zapíší všechny použité SCSI ovladače, protože jejich pozdější ruční zavádění by mohlo působit problémy.

---

### **Důležité: Update initrd**

Zavaděč zavádí `initrd` současně s jádrem. Po reinstalaci `initrd` není GRUB nutné neinstalovat, protože GRUB vyhledává soubory v adresářích při startu.

---

## **8.1.2 linuxrc**

Hlavním účelem `linuxrc` je příprava pro připojení a přístupu ke kořenovému adresáři. V závislosti na nastavením vašeho systému je `linuxrc` také odpovědný za:

### Zavádění modulů jádra

V závislosti na vaší konfiguraci může být potřeba před zavést pro některé zařízení ovladače (nejdůležitější je obvykle pevný disk). Aby bylo možné např. správně přistupovat k disku, musí jádro zavést správný modul pro souborový systém.

### Správa nastavení RAID a LVM

Pokud konfiguruje kořenový souborový systém na RAID nebo LVM, `linuxrc` nastaví, aby bylo možné přistupovat k souborovému systému, LVM nebo RAID. Informace o RAIDu najdete v části [2.2 – „Konfigurace softwarového RAIDu“](#) (strana 53), informace o LVM v [2.1 – „Konfigurace LVM“](#) (strana 47).

### Správa síťového nastavení

Jestliže používáte síťový souborový systém, `linuxrc` zajišťuje zavedení správných síťových modulů.

Pokud je `linuxrc` volán při startu jako část instalace, úlohy se od uvedených výše liší:

## Vyhledání instalačního média

Po startu systému se z instalačního média zavádí jádro a zvláštní `initrd` s instalátorem YaST. Instalátor YaST běžící v RAM souborovém systému potřebuje k instalaci informace o aktuálním umístění instalačního média.

## Prvotní rozpoznání hardwaru a zavedení příslušných modulů

Jak bylo zmíněno v části 8.1.1 – „`initrd`“ (strana 148), spouštění systému je proces, kdy se zavádí minimum ovladačů, které mohou být použity s většinou hardwaru. `linuxrc` provádí první zjištění hardwaru, aby zjistil, jaké ovladače budou pro váš systém potřeba. Tyto moduly jsou pak zapsány do proměnné `INITRD_MODULES` v souboru `/etc/sysconfig/kernel` a používány při příštích startech systému. Tyto moduly také `linuxrc` zavede během instalačního procesu.

## Zavedení instalačního nebo záchranného systému

Po rozpoznání hardwaru a zavedení správných modulů spouští `linuxrc` instalační proces a to buď systém s instalátorem YaST nebo záchranný systém.

## Spuštění programu YaST

Na závěr `linuxrc` spustí samotná YaST, který začne s instalací balíčků a nastavením systému.

## 8.1.3 Informace i `initrd`

Další informace o `initrd` najdete v souborech `/usr/src/linux/Documentation/ramdisk.txt` a a v manuálových stránkách `initrd(4)` a `mkinitrd(8)`.

## 8.2 Program `init`

Program `init` inicializuje všechny další procesy, představuje tedy otce všech procesů. Mezi všemi programy má zvláštní roli: spouští ho přímo jádro a je imunní proti signálu 9, který normálně ukončí každý proces. Všechny další procesy pak program `init` spouští buď sám, nebo některý z jeho potomků.

Program `init` se konfiguruje centrálně v souboru `/etc/inittab`, kde se definují *úroveň běhu* angl. *runlevel* (více v 8.3 – „Úroveň běhu“ (strana 151)) a kde se určí, které služby a démoni mají být na jednotlivých úrovních k dispozici. Podle údajů v souboru `/etc/inittab` pak program `init` spouští různé skripty, které jsou z důvodu přehlednosti umístěny ve společném adresáři `/etc/init.d`.

Celý postup startu systému (a stejně tak i jeho zastavení) má tedy na starost program (a stejnojmenný proces) `init`. Z tohoto hlediska lze chápat činnost jádra jako proces na pozadí, jehož úlohou je udržovat všechny ostatní procesy a přidělovat hardware a čas CPU podle požadavků ostatních programů.

## 8.3 Úrovně běhu

V Linuxu existují různé *úrovně běhu*, které definují, v jakém stavu se nachází systém. Standardní úroveň běhu, které systém dosáhne po startu, je uvedena v souboru `/etc/inittab` v položce `initdefault`. Obvykle je to úroveň 3 nebo 5 (viz tabulka 8.1 – „Seznam platných úrovní běhu“ (strana 151)). Alternativou je zadat požadovanou úroveň běhu při startu (např. ze startovací výzvy LILO). Všechny parametry, které jádro samo nepoužije, totiž předá beze změny procesu `init`.

Aby šlo později úroveň běhu změnit, lze zavolat program `init` s udáním požadované úrovně běhu (což je dovoleno pouze superuživateli).

Například příkazem `init 1` přejde systém do *jednouživatelského režimu single user mode*, vhodného pro správu systému. Po ukončení této práce administrátor opět zadá `init 3`, čímž systém přejde opět na normální úroveň běhu, na které běží potřebné služby a kde se mohou přihlašovat uživatelé.

Tabulka níže podává přehled o dostupných úrovních běhu.

---

### **Důležité: Úroveň běhu 2 s oddílem `/usr/` připojeným přes NFS**

Nepoužívejte úroveň běhu 2, pokud je adresář `/usr` na oddílu připojeném přes NFS. Adresář `/usr` obsahuje programy důležité pro běh systému. Služba NFS není na úrovni běhu 2 aktivní (lokální víceuživatelský režim bez sítě) a systém by v důsledku neexistence adresáře `/usr` nefungoval korektně.

---

**Tabulka 8.1** Seznam platných úrovní běhu

---

Úroveň běhu	Význam
0	Stop <i>System halt</i>
S	Jednouživatelský režim, US klávesnice <i>Single user mode</i>

---

Úroveň běhu	Význam
1	Jednouživatelský režim <i>Single user mode</i>
2	Lokální víceživatelský režim bez sítě <i>Local multiuser without remote network (např. NFS)</i>
3	Plně víceživatelský režim se sítí <i>Full multiuser with network</i>
4	Nepoužito
5	Plně víceživatelský režim se sítí a KDM (standard), GDM nebo XDM <i>Full multiuser with network and xdm</i>
6	Restart systému <i>System reboot</i>

Ve standardní instalaci je jako výchozí nastavena úroveň 5. Uživatelé se tedy po startu systému pohybují v grafickém prostředí. Máte-li výchozí úroveň nastavenou na 3 a na počítači máte nainstalovaný systém X Window (kap. 14 – „*Systém X Window*“ (strana 235)) a přejete-li si, aby se uživatel přihlašoval přímo v grafickém prostředí, můžete nastavit standardní úroveň běhu pomocí programu YaST na hodnotu 5. Předtím si ovšem vyzkoušejte příkazem `init 5`, zda se systém bude chovat podle vašich představ.

#### Varování: Změna `/etc/inittab`

Doporučuje se velká opatrnost, chcete-li do souboru `/etc/inittab` zasahovat ručně. Jeho poškození totiž může vést k neschopnosti systému řádně nastartovat. Pokud se to stane, je zde ještě možnost z výzvy zavaděče zadat parametr `init=/bin/bash`, čímž se vám objeví přímo výzva příkazového procesoru:

```
boot:linux init=/bin/bash
```

## 8.4 Změna úrovně běhu

Při změně úrovně běhu se nejprve spustí tzv. *stop-skripty*, které ukončí činnost některých programů současně úrovně. Dále se spustí *start-skripty* nové úrovně, a tím se zpravidla spustí i řada programů.



Pro názornost zde ukážeme příklad změny úrovně běhu z hodnoty 3 na 5:

- Administrátor (uživatel `root`) sdělí procesu `init`, že se má změnit úroveň běhu:

```
init 5
```

- Podle konfiguračního souboru `/etc/inittab` `init` usoudí, že má spustit skript `/etc/init.d/rc` s novou úrovní běhu jakožto parametrem.
- Nyní volá program `rc` ty stop skripty současné úrovně běhu, jimž neodpovídají start-skripty v nové úrovni. V našem případě jsou to ty skripty, jež se nalézají v adresáři `/etc/init.d/rc3.d` (stará úroveň běhu byla 3) a začínají písmenem `K`. Jména stop skriptů začínají písmenem `K` *kill*, zatímco jména startovacích skriptů začínají písmenem `S` *start*. Po písmenu `K` následuje číslo, udávající pořadí, aby byly respektovány případné závislosti mezi programy.
- Nakonec se zavolají startovací skripty nové úrovně běhu, které v našem případě leží v adresáři `/etc/init.d/rc5.d` a začínají písmenem `S`. Rovněž zde se dodržuje pořadí.

Pokud se stane, že změníte úroveň běhu na úroveň právě běžící (tj. např. z úrovně 3 opět na úroveň 3), přečte program `init` pouze svůj konfigurační soubor `/etc/inittab` a zjistí, zda i v rámci téže úrovně nejsou nějaké změny. Pokud je najde, provede příslušné kroky (například spustí program `getty` pro další konzoli).

## 8.5 Init skripty

Skripty v adresáři `/etc/init.d` se dělí do dvou kategorií:

Skripty, které program `init` volá přímo

to je případ startu a korektního zastavení systému (např. klávesovou kombinací `Ctrl` + `Alt` + `Return`) Vykonání těchto skriptů je definováno v `/etc/inittab`.

Skripty, které program `init` volá nepřímo

to se stane při změně úrovně běhu. Spustí se skript `/etc/init.d/rc` volající správné skripty ve správném pořadí.

Skripty pro změnu úrovně běhu se rovněž nalézají v adresáři `/etc/init.d`, ale volají se pomocí symbolických odkazů z jednoho z adresářů počínaje `/etc/init.d/`

`rc0.d` až po `/etc/init.d/rc6.d`. To je velmi názorné a zabraňuje to duplicitě skriptů, použitých pro více úrovní běhu.

Každý z těchto skriptů se dá volat jako start-skript i stop-skript, rozlišují proto parametry `start` a `stop`.

Navíc rozlišují skripty parametry `restart`, `reload`, `force-reload` a `status`. Význam všech voleb je v následující tabulce.

**Tabulka 8.2** *Přehled voleb init skriptů*

Volba	Význam
<code>start</code>	Spustit službu.
<code>stop</code>	Ukončit službu.
<code>restart</code>	Pokud služba běží, ukončit ji a znovu spustit, pokud neběží, pouze spustit.
<code>reload</code>	Znovu načíst konfiguraci služby, aniž by se zastavovala a spouštěla.
<code>force-reload</code>	Totéž jako <code>reload</code> , pokud to služba podporuje, jinak jako <code>restart</code> .
<code>status</code>	Zobrazit aktuální status.

Příklad:

Při opuštění úrovně běhu 3 je skript `/etc/init.d/rc3.d/K40network` jedním ze spuštěných skriptů. Program `/etc/init.d/rc` volá skript `/etc/init.d/networks` s parametrem `stop`. Při vstupu do úrovně běhu 5 se spustí tentýž skript, ale s parametrem `start`.

Odkazy v podadresářích pro jednotlivé úrovně běhu slouží pouze k tomu, aby umožnily přiřadit skripty úrovním běhu.

Vytvoření a odstranění potřebných odkazů provádí program `insserv` při instalaci a deinstalaci balíků. Podrobnosti najdete v manuálové stránce tohoto programu.

V dalším odstavci najdete krátký popis startovacího a ukončovacího skriptu spolu s řídicím skriptem:

### *boot*

Spouští se při startu systému přímo z programu `init`. Je nezávislý na požadované výsledné úrovni běhu a provádí se pouze jednou. Spustí se démon jádra, který zajistí zavedení modulů jádra. Zkontrolují se souborové systémy, zruší se některé nadbytečné soubory v adresáři `/var/lock` a sít' se nakonfiguruje pro *loopback device* (pokud je to nastaveno v souboru `/etc/rc.config`). Dále se nastaví systémový a PnP hardware pomocí nástroje `isapnp`.

Pokud se stane chyba při automatické opravě souborového systému, má systémový administrátor možnost po zadání hesla zadat další informace přispívající k jejímu odstranění.

Dále se vykonají všechny skripty v adresáři `/etc/init.d/boot.d` začínající písmenem `S`. Je to proto vhodné místo pro vaše rozšíření o ty kroky, které by měl systém dělat pouze při startu.

Nakonec se spustí skript `boot.local`.

### *boot.local*

Zde můžete přidat další příkazy, které se mají provést při startu, než se začne zvyšovat úroveň běhu. Funkční obdobou v dosových systémech je soubor `AUTOEXEC.BAT`.

### *boot.setup*

Všeobecná nastavení při přechodu z jedinouživatelského režimu *single user mode* na libovolnou vyšší úroveň běhu, například rozložení kláves a konfigurace konzole.

### *halt*

Tento skript se spouští při přechodech na úroveň běhu 0 nebo 6. Proto se může zavolat jak pod jménem `halt`, tak i `reboot`, a podle předaného jména se systém znovu nastartuje nebo ukončí.

### *rc*

Řídicí skript pro změnu úroveň běhu. Spouští nejprve stop skripty současné úrovně a po nich start skripty nové úrovně.

Do této kostry můžete vhodně zasadit své vlastní skripty. Šablonu na to najdete v souboru `/etc/init.d/skeleton`. Pro konfiguraci spuštění vlastního skriptu v souboru

`/etc/rc.config` zde vytvořte proměnnou `START_služba`. Dodatečné parametry lze uvést v případě potřeby také do souboru `/etc/rc.config` (viz např. skript `/etc/init.d/gpm`).

---

## Varování

Při vytvoření vlastních skriptů zachovejte opatrnost. Chybný skript může způsobit nefunkčnost systému.

---

## 8.5.1 Vkládání skriptů

V Linuxu není problém vytvářet vlastní skripty a poměrně jednoduše je integrovat do stávajícího prostředí. Informace o způsobu pojmenování, formátu a organizaci vlastních skriptů najdete ve specifikaci LSB a manuálových stránkách `init`, `init.d` a `insserv`. Zajímavé informace najdete také v manuálových stránkách `startproc` a `killproc`.

---

### Varování: Vytváření vlastních init skriptů

Chyby v `init` skriptech mohou vést k zamrznutí počítače. Věnujte prosím editaci těchto skriptů maximální pozornost a pokud je to možné, otestujte je. Užitečné informace o `init` skriptech najdete v části [8.3 – „Úroveň běhu“](#) (strana 151).

---

- Jako šablonu pro svůj nový `init` skript použijte soubor `/etc/init.d/skeleton`. Kopii tohoto souboru uložte pod novým jménem a editujte důležité položky jako `program`, jména souborů, cesty a další detaily. Šablonu samozřejmě můžete rozšířit o vlastní části.
- Blok `INIT INFO` je povinnou částí skriptu a měly by v něm být provedeny příslušné změny:

```
### BEGIN INIT INFO
# Provides:          FOO
# Required-Start:    $syslog $remote_fs
# Required-Stop:     $syslog $remote_fs
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Description:       Start FOO to allow XY and provide YZ
### END INIT INFO
```

Na první řádce bloku `INFO` po řádce `Provides :`, uveďte jméno služby nebo programu kontrolovaného nově vytvářeným skriptem. V řádkách `Required-Start :` a `Required-Stop :` uveďte všechny služby, které je nutné spustit a zastavit před startem nebo spuštěním vaší nové služby.

Tyto informace budou později použity při generování jména a čísla skriptu v adresářích úrovní běhu. V `Default-Start` a `Default-Stop` uveďte úroveň běhu, kdy se služba má automaticky spustit nebo ukončit. Na konec do řádky `Description` napište krátký popis služby.

- Odkazy z `/etc/init.d/` do příslušného adresáře úrovně běhu (`/etc/init.d/rc?.d/`), vytvoříte zadáním příkazu `insserv jmeno_skriptu`. Program `insserv` používá hlavičku `INIT INFO` pro vytváření důležitých odkazů potřebných pro spuštění a zastavení skriptu v adresářích úrovní běhu (`/etc/init.d/rc?.d/`). Program se také stará o správné pořadí spuštění a zastavení v určených úrovních běhu. Pokud byste raději používali grafický nástroj, můžete použít editor úrovní běhu v programu YaST, popsany v sekci 8.6 – „Editor úrovní běhu“ (strana 157).

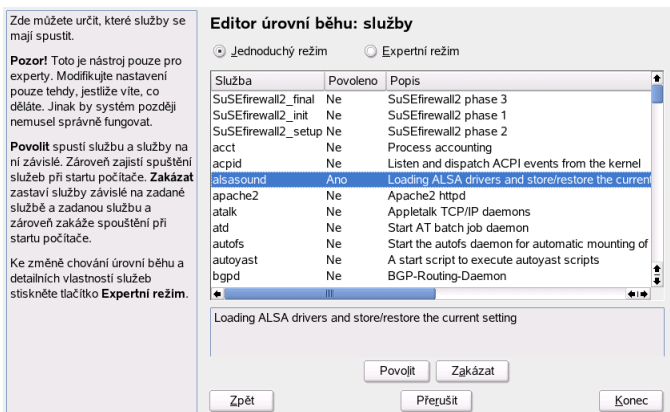
Pokud již skript v adresáři `/etc/init.d/` existuje, můžete ho do existujícího schématu úrovní běhu jednoduše integrovat pomocí programu `insserv` nebo povolením příslušné služby v programu YaST. Vámi provedené změny se projeví při následujícím restartu počítače, během kterého dojde k automatickému spuštění nové služby.

## 8.6 Editor úrovní běhu

Po spuštění tohoto modulu programu YaST se zobrazí seznam dostupných služeb a jejich stav (zda jsou povoleny či ne). Zvolit si můžete ze dvou režimů zobrazení *Jednoduchý režim* nebo *Expertní režim*. Jako výchozí je nastaven *Jednoduchý režim*, který je vhodný pro většinu situací.

V levém sloupci *Jednoduchého režimu* je jméno služby, v prostředním stav služby a v pravém sloupci krátký popis služby. U zvolené služby je detailnější popis dostupný v okně pod seznamem. Službu povolíte tak, že ji označíte a kliknete na *Povolit*. Pokud chcete službu zakázat, opět ji zvolte a klikněte na tlačítko *Zakázat*.

**Obrázek 8.1** Editor úrovní běhu



Pokud potřebujete o službách více informací a chtěli byste použít detailnější nastavení, vyberte *Expertní režim*. V tomto režimu získáte informace o nastavené výchozí úrovni nebo-li `initdefault`, která říká, do jaké úrovně se má systém spustit při startu. Jako výchozí je nastavena úroveň 5 (Plný víceuživatelský režim se sítí a xdm). Vhodnou náhradou obvykle bývá úroveň 3 (Plný víceuživatelský režim se sítí).

YaST umožňuje výběr nové výchozí úrovně běhu (viz tabulka 8.1 – „Seznam platných úrovní běhu“ (strana 151)). Zároveň nabízí tabulku, kde můžete povolit nebo zakázat běh určité služby. V tabulce najdete všechny dostupné služby a demony. Příslušnou úroveň nastavíte tak, že v řádce vybrané služby označíte příslušné pole úrovně běhu (*B*, *0*, *1*, *2*, *3*, *5*, *6* a *S*), ve které se má služba spustit. Úroveň 4 není definována a můžete si ji nastavit podle svých potřeb. Jako poslední najdete v tabulce krátký popis služby nebo démona.

Pomocí *Nastavit/Obnovit* můžete určit, co se má se zvolenou službou provést. Okamžitě můžete služby povolit či zakázat v *Spustit/Zastavit/Načíst znovu*. Pokud po změnách chcete zobrazit aktuální stav, zvolte v *Spustit/Zastavit/Načíst znovu* položku *Znovu načíst stav*. Kliknutím na tlačítko *Konec* uložíte změny.

---

### Varování: Změna úrovně běhu

Chybné nastavení úrovně běhu může vést k chybě systému. Před změnou úrovně běhu se prosím ujistěte, zda se tím neovlivní některá ze služeb důležitých pro váš systém.

---

## 8.7 SuSEconfig a /etc/sysconfig

Prakticky celá konfigurace systému SUSE Linux je otázkou centrálního konfiguračního adresáře `/etc/sysconfig`. Ve verzích starších než 8.0 byla konfigurace soustředěna do souboru `/etc/rc.config`. Tento soubor již není používán.

Každý ze skriptů v adresáři `/etc/init.d` načítá soubory z adresáře `/etc/sysconfig`, kde převezme platné hodnoty jednotlivých proměnných. Nastavení v `/etc/sysconfig` vede také k automatickému vytváření nebo změně některých dalších konfiguračních souborů skriptem `SuSEconfig`. Tak například po změnách v síťové konfiguraci se nově vytvoří soubor `/etc/host.conf`, protože na těchto změnách závisí.

Po ručních změnách v některém ze souborů v adresáři `/etc/sysconfig` musíte vždy zavolat program `SuSEconfig`, abyste tak zajistili, že se vaše změny rozšíří i do závislých konfiguračních souborů. Použijete-li na konfiguraci program `YaST`, nemusíte se o to starat, protože ten zavolá program `SuSEconfig` při korektním ukončení automaticky.

Tato koncepce vám umožní provést zásadní změny v konfiguraci, aniž byste museli restartovat počítač. Některé změny však jdou tak daleko, že je třeba restartovat alespoň některé jimi ovlivněné programy. To je typické například u konfigurace sítě, kde zadáním příkazů `rcnetwork stop` a `rcnetwork start` dosáhnete toho, že se změnou postížené programy restartují.

Doporučený postup změny systémového nastavení se skládá z následujících kroků:

1. Přejděte do jednovýživatelského režimu *single user mode* (úroveň běhu 1) pomocí příkazu `init 1`.
2. Změňte konfigurační soubory podle své potřeby. Použít můžete svůj oblíbený textový editor nebo editor v programu `YaST`.

---

### Důležité: Manuální změna systémové konfigurace

Pokud ke změně *nepoužíváte* `YaST`, ujistěte se že jsou prázdné proměnné a proměnné skládající se z více položek v souborech v adresáři `/etc/sysconfig` v uvozovkách (`KEYTABLE= " "`). Proměnné s jednou hodnotou není nutné uzavírat do uvozovek.

---

3. Aby se změny projevíly, spusťte `/sbin/SuSEconfig`. Pokud jste změny provedli pomocí programu YaST, spustí se SuSEconfig automaticky.
4. Vraťte se do původní úrovně běhu příkazem `init 3` (nahraďte 3 číslem vaší úrovně běhu).

Tento postup je nutné dodržovat při hlubších zásazích do systému, jako je například změna konfigurace sítě. V případě jednoduchých změn není zapotřebí přechod do *jed-nouživatelského režimu*, ale získáte tak jistotu, že u všech služeb došlo ke správnému spuštění.

---

### Tip

Automatickou konfiguraci programem SuSEconfig lze vypnout tak, že se proměnná `ENABLE_SUSECONFIG` v souboru `/etc/sysconfig/suseconfig` nastaví na hodnotu `no`. Je to ovšem i cesta, jak současně ztratit instalační podporu SUSE. Nevypínejte SuSEconfig, pokud chcete využít bezplatné instalační podpory. Autokonfiguraci je možné zakázat také pouze částečně.

---

## 8.8 YaST sysconfig Editor

Nejdůležitější konfigurační soubory systému SUSE Linux jsou uloženy v adresáři `/etc/sysconfig`. Sysconfig editor představuje způsob, jak zde uložená nastavení editovat s co nejvyšším pohodlím. Hodnoty lze měnit a v případě nutnosti také vkládat do vlastních konfiguračních souborů. Většinu nastavení není nutné nastavovat ručně. K nastavení dojde automaticky při instalaci příslušných balíčků.

---

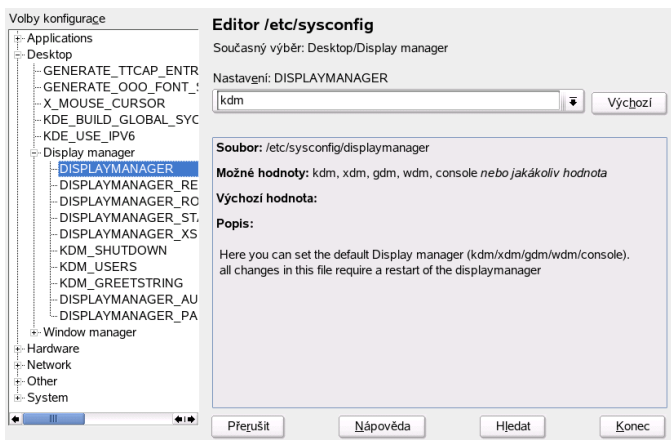
### Varování: Změna souborů v `/etc/sysconfig/`

Pokud nemáte se změnou konfiguračních souborů žádné zkušenosti, neměňte žádná nastavení v adresáři `/etc/sysconfig`. Chybný zásah do těchto souborů může vést k nefunkčnosti systému. Pokud je ruční editace nezbytná, věnujte pozornost komentářům u jednotlivých proměnných.

---



**Obrázek 8.2** Konfigurace systému pomocí editoru souborů `sysconfig`



Dialog YaST `sysconfig` editoru se skládá ze tří částí. V levé části jsou zobrazeny nastavitelné proměnné. Po volbě proměnné se v pravé části objeví aktuální nastavení zvolené proměnné. Pod tímto nastavením najdete krátký popis funkce proměnné, možné dosažitelné hodnoty, výchozí hodnotu a soubor, kde se tato proměnná nachází. Dialog také poskytuje informace o skriptech, které se po nastavení této proměnné spustí a službách, které se v důsledku nového nastavení mohou spustit. Po změně se YaST dotáže, zda si skutečně proměnnou přejete změnit. Nastavení uložíte kliknutím na *Dokončit*.



# Starování systému a zavaděče

Tato kapitola popisuje různé metody startování linuxového systému. Nejdříve jsou však vysvětleny některé technické detaily tohoto procesu. Poté následuje detailní popis programů GRUB (současný zavaděč používaný v systému SUSE Linux) a možnosti použití programu YaST. V této kapitole najdete také popis řešení některých problémů, které mohou u nastavení zavaděče GRUB nastat.

Tato kapitla se zaměřuje na konfiguraci zavaděče GRUB. Proces startování systému je popsán v kapitole 8 – „*Startování*“ (strana 147). Zavaděč je rozhraním mezi počítačem (BIOSem) a operačním systémem (SUSE Linux). Nastavení zavaděče ovlivňuje spouštění nainstalovaných operačních systémů a dostupnost parametrů, kterými můžete spouštění ovlivnit.

Nejdůležitější pojmy používané v této kapitole jsou:

## Master Boot Record

Struktura MBR je standardizována a není závislá na použitém operačním systému. Prvních 446 bytů je rezervováno pro kód startovacího programu. Následujících 64 bytů je určeno pro uložení tabulky diskových oddílů, která obsahuje informace o maximálně 4 oddílech. Bez této tabulky nemůže být na disku žádný souborový systém - disk je bez této tabulky nepoužitelný. Poslední 2 byty musí obsahovat speciální magické číslo (AA55). MBR, který na této pozici obsahuje jiné číslo, může být BIOSem, a některými operačními systémy, posouzen jako neplatný.

## Zavaděcí sektory

Zavaděcí sektory jsou uloženy na každém diskovém oddílu jako první. Výjimku tvoří pouze rozšířené diskové oddíly, které jsou pouze kontejnery pro další oddíly. Zavaděcí sektory jsou velké 512 bytů, a slouží k uložení kódu pro spuštění operač-

ního systému uloženého na tomto oddílu. Zaváděcí sektory na oddílech vytvořených z DOSu, OS/2, a Windows fungují přesně jak bylo popsáno (navíc obsahují některá základní data o struktuře souborového systému). V Linuxu, na rozdíl od jmenovaných OS, je tento sektor prázdný (i po vytvoření souborového systému), a Linuxový oddíl není schopen zavést sám sebe, i když oddíl obsahuje platný souborový systém s jádrem. Aby bylo možné zavést z tohoto oddílu Linux, musíme do tohoto sektoru uložit zaváděcí program. Zaváděcí sektor s platným zaváděcím kódem obsahuje na stejné pozici jako MBR (poslední 2 byty) shodné magické číslo (AA55).

## 9.1 Startování

V tom nejjednodušším případě, kdy se na počítači nachází pouze jeden operační systém, se zavaděč nainstaluje do MBR. V případě více operačních systémů však přicházejí ke slovu následující postupy:

### Spouštění dalšího systému z externího média

Jeden z nainstalovaných systémů je spouštěn z disku, druhý pomocí zavaděče uloženého na externím médiu (disketa, CD, USB flash disk). Obvykle tato metoda není nutná, protože GRUB umí spouštět i jiné operační systémy než Linux.

### Instalace zavaděče do zaváděcího sektoru

V případě, že zavaděč neumí spustit určitý operační systém, je možné ho nainstalovat do zaváděcího sektoru a k jeho spuštění použít zavaděč, který ho dokáže vyvolat. Tato volba je v systému SUSE Linux v případě instalace několika operačních systémů na jeden počítač, výchozí.

### Instalace zavaděče do MBR

Zavaděč umožňuje spouštění různých operačních systémů. Který bude spuštěn, si může vybrat ve startovací nabídce. Aby došlo ke spuštění jiného systému, musí být počítač restartován. Toto řešení je samozřejmě možné jen v případě, že je zavaděč kompatibilní se všemi operačními systémy, které chcete s jeho pomocí spouštět. GRUB, zavaděč systému SUSE Linux, je schopný spouštět většinu obvyklých operačních systémů. Do MBR se zavaděč v systému SUSE Linux automaticky nainstaluje pouze v případě, že na počítači není nainstalován žádný další operační systém.

## 9.1.1 Startování DOSu a Windows 9x

MBR DOSu na prvním pevném disku obsahuje informaci o tom, který oddíl je aktivní - tedy kde se má hledat kód pro zavedení operačního systému. Proto musí být DOS nainstalován na první pevný disk. Spustitelný kód v MBR (zavaděč prvního stupně) potom testuje, zda označený oddíl obsahuje platný zavaděcí sektor. Jestliže je vše v pořádku, spustí se odtud zavaděč druhého stupně. Odtud je možné nahrávat DOSové programy, a objeví se obvyklý DOSový prompt. V DOSu lze označit jako aktivní pouze primární diskové oddíly. Z toho důvodu nemůžete použít pro zavedení DOSu logické diskové oddíly, které jsou uvnitř rozšířených oddílů.

## 9.2 Výběr zavaděče

V systému SUSE Linux je jako výchozí zavaděč použit GRUB. V některých případech, kdy je použit zvláštní hardware ve spojení s určitým softwarem, však může být mnohem vhodnější použití zavaděče LILO.

Zavaděč LILO se automaticky nainstaluje v případě aktualizace ze staršího systému SUSE Linux, který používal jako výchozí zavaděč LILO. V nové instalaci se vždy nainstaluje zavaděč GRUB.

Informace o instalaci a nastavení zavaděče LILO najdete v databázi instalační podpory pod heslem LILO a v souboru `/usr/share/doc/packages/lilo`.

## 9.3 Startování systému se zavaděčem GRUB

GRUB (GRand Unified Boot loader) podobně jako LILO pracuje ve dvou fázích. V první fázi, `stage1`, se spustí kód velký pouze 512 bytů, který je zapsán v MBR, zavaděcím sektoru diskového oddílu nebo na disketě. Druhá fáze, `stage2`, spočívá ve spuštění většího programu vykonávajícího zavedení jako takové. Jedinou funkcí programu první fáze je zavést program fáze druhé.

Odsud již GRUB pracuje jinak než LILO, poněvadž program druhé fáze obsahuje kód pro čtení ze souborového systému. V současné době jsou podporovány tyto souborové

systemy: Ext2, Ext3, ReiserFS, JFS, XFS, Minix a DOS FAT používaný Windows. GRUB tedy může přistupovat na souborové systémy již před vlastním startováním systému. Číst lze z těch zařízení, která jsou dostupná přes BIOS (disketové mechaniky a pevné disky). Ve výsledku to znamená, že provedené změny v konfiguraci programu GRUB nemusíme po každé změně zapsat reinstalací zavaděče. Při zavádění GRUB načte svůj soubor s menu a odsud zjistí, na kterých oddílech leží jádro a výchozí RAM disk (`initrd`), a je sám schopen tyto soubory najít.

Konfigurace zavaděče GRUB se nalézá v následujících souborech:

```
/boot/grub/menu.lst
```

Informace o všech diskových oddílech a operačních systémech, které lze spustit pomocí zavaděče GRUB. Pokud není operační soubor zanesen v tomto souboru, nepůjde spustit pomocí zavaděče GRUB.

```
/boot/grub/device.map
```

Překlad jmen zařízení od zavaděče GRUB a BIOSu do linuxových jmen.

```
/etc/grub.conf
```

Parametry a volby zavaděče GRUB potřebné pro správnou instalaci zavaděče.

Výhodou programu GRUB je, že lze jednoduše měnit veškeré parametry startu systému před samotným startem (viz „[Změna položek v menu při startu](#)“ (strana 170)). Pokud při zavádění zjistíte, že soubor s menu obsahuje chyby, je stále možné opravit tyto chyby za chodu. V programu GRUB také můžete zadávat příkazy interaktivně na příkazový řádek, takže lze startovat i systém, jenž není uveden v konfiguračním souboru.

## 9.3.1 Startovací menu

GRUB zobrazuje zaváděcí menu na grafické titulní obrazovce nebo v rozhraní textového režimu. Co bude obsahem této obrazovky, lze nastavit v souboru s menu `/boot/grub/menu.lst`. V tomto souboru jsou popsány veškeré informace o diskových oddílech a operačních systémech, které lze zvolit z nabídky při zavádění.

GRUB nahraje menu přímo ze souborového systému při každém startu systému. Pokud chcete změnit nastavení zavaděče, upravíte pouze menu soubor pomocí programu YaST nebo vašim oblíbeným editorem.

Soubor s menu obsahuje příkazy spouštěné při zavádění a jeho skladba je jednoduchá na pochopení. Každý řádek sestává z příkazu, volitelně následovaného parametry. Ty

jsou odděleny mezerou stejně jako v shellu. Z historických důvodů lze u některých příkazů použít před jejich prvním parametrem =. Řádky začínající znakem hash # jsou považovány za komentáře.

Každý záznam, jenž se objeví v menu zavaděče, odpovídá jménu v menu souboru, které musí být uvozeno pomocí slova `title`. Jinými slovy: textový řetězec následující za `title` (včetně mezer) se zobrazí jako volitelná položka. Následující řádky až do další položky `title` pak reprezentují příkazy, které se provedou, pokud zvolíte tuto položku v menu.

Jednoduchý příklad takového příkazu je zřetěžené nahrání zavaděče jiného operačního systému. Příkaz se nazývá `chainloader` a jako parametr má obvykle zaváděcí blok jiného diskového oddílu. Zapsáno v notaci programu GRUB:

```
chainloader (hd0,3)+1
```

Jak GRUB pojmenovává zařízení je vysvětleno v sekci „[Konvence pojmenování pevných disků a oddílů](#)“ (strana 168). Příklad uvedený výše odkazuje na první blok čtvrtého oddílu prvního disku.

Příkaz pro určení obrazu jádra je `kernel`. První parametr je cesta k obrazu jádra na diskovém oddíle. Zbylé argumenty se během zavádění předají jádru jako parametry pro start Linuxu.

Pokud jádro nemá zabudované nezbytné ovladače pro souborový systém nebo disk (aby mohlo přistupovat na kořenový oddíl), připojte také příkaz `initrd`. Tento příkaz má pouze jeden parametr, a to cestu k souboru `initrd`. Příkaz `initrd` musí být umístěn bezprostředně po příkazu `kernel`, protože jádro (nyní již zavedené) očekává nějaký obraz `initrd` na konkrétní adrese v paměti.

Příkaz `root` zjednodušuje určení, kde se nachází obrazy jádra a `initrd`. `root` má jako jediný parametr označení zařízení nebo diskového oddílu (v notaci GRUB).

GRUB následně připojí na začátek všech cest k souborům (jádra, `initrd` nebo jiných souborů, které výslovně neurčují cestu nebo zařízení) hodnotu svého parametru. Toto připojování se děje do nalezení dalšího příkazu `root`. Tento příkaz není použit v souboru `menu.lst`, který je generován během instalace.

Příkaz `boot` je automaticky proveden jako poslední u každé položky menu. Nemusí se tedy zapisovat jako příkaz do souboru `s.menu`. Jestliže se však dostanete do situace, že musíte zadávat příkazy do příkazové řádky programu GRUB, nezapomeňte nakonec

zadat příkaz `boot`. Příkaz nemá parametry a pouze spustí zavádění obrazu jádra nebo zřetězený zavaděč (chain loader).

Jakmile máte vytvořen soubor s nabídkou položek odpovídajících jednotlivým OS, vyberte jednu jako implicitní pomocí příkazu `default`. Pokud nevyberete implicitní položku tímto příkazem, zavede se systém z první položky v menu (číslo 0). Lze také nastavit časovou prodlevu ve vteřinách, kdy můžete vybrat některou z položek. Řádky s příkazy `timeout` a `default` jsou obvykle umístěny před položky menu. Vzorový menu soubor je popsán v sekci „Vzorový soubor menu.lst“ (strana 169).

## Konvence pojmenování pevných disků a oddílů

GRUB pojmenovává disky a oddíly podle jiných konvencí, než jste zvyklí v Linuxu, a jaké byste nejspíš očekávali (např. `/dev/hda1`). První disk je vždy odkazován jako `hd0`. Disketová mechanika se nazývá `fd0`.

---

### Důležité: Výpočet čísla oddílu

GRUB počítá diskové oddíly od nuly. `hd0,0` tedy odkazuje na první oddíl prvního disku. Označení odpovídá typickému stolnímu počítači s jedním diskem připojeným jako primární master disk. V Linuxu bychom se na něj odkazovali pomocí `/dev/hda1`.

---

Čtyři primární oddíly (které lze na disku vytvořit) jsou číslovány od 0 do 3 a logické oddíly jsou číslovány od 4 výš.

```
(hd0,0)   první primární oddíl prvního disku
(hd0,1)   druhý primární oddíl prvního disku
(hd0,2)   třetí primární oddíl prvního disku
(hd0,3)   čtvrtý primární oddíl prvního disku
(hd0,4)   první logický oddíl
(hd0,5)   druhý logický oddíl
...
```

---

### Důležité: IDE, SCSI a RAID

GRUB nerozlišuje mezi IDE, SCSI nebo RAID zařízeními. Veškeré pevné disky detekované BIOSem nebo diskovým řadičem jsou číslovány podle pořadí zavádění nastaveném v BIOSu.

---



Fakt, že disky jsou jinak adresovány Linuxem a jinak BIOSem, je problém jak pro LILO, tak pro GRUB. Oba programy používají podobný algoritmus pro mapování. Nicméně GRUB ukládá výsledek tohoto algoritmu do souboru (`device.map`), který lze editovat. Více informací o souboru `device.map` najdete v 9.3.2 – „Soubor `device.map`“ (strana 172).

V programu GRUB musí být cesta uvedena jako jméno zařízení, uzavřené do kulatých závorek, následovaná jménem souboru včetně plné cesty na tomto zařízení nebo oddílu. Cesta musí vždy začínat lomítkem. Například v systému s jedním IDE diskem a Linuxem uloženým na prvním oddílu, se odkážete na jádro takto:

```
(hd0,0)/boot/vmlinuz
```

## Vzorový soubor `menu.lst`

Následující příklad ukazuje, jak funguje soubor `menu.lst`.

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8

title linux
    kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
    initrd (hd0,4)/initrd
title windows
    chainloader(hd0,0)+1
title floppy
    chainloader(fd0)+1
title failsafe
    kernel (hd0,4)/vmlinuz.shipped root=/dev/hda7 ide=nodma
    apm=off acpi=off vga=normal nosmp maxcpus=0 3
    initrd (hd0,4)/initrd.shipped
```

Tento fiktivní stroj má zavaděcí Linuxový oddíl na `/dev/hda5`, kořenový oddíl na `/dev/hda7`, a instalaci Windows na `/dev/hda1`.

První část souboru definuje nastavení titulní obrazovky a standardní chování:

```
gfxmenu (hd0,4)/message
    Obrázek zobrazený na pozadí je uložen na /dev/hda5 a jmenuje se message.
```

```
color
    Barevné schéma: bílá pro popředí, modrá jako pozadí, černá jako popředí pro vybranou položku a světle šedá pro pozadí zvolené položku. Definice barev neovlivní
```

titulní grafickou obrazovku definovanou pomocí `gfxmenu`, ale pouze standardní textové rozhraní programu GRUB. V systému SUSE Linux se můžete z grafického menu do textového přepnout stisknutím `[Esc]`.

`default 0`

Implicitně se zavede první položka `title linux`.

`timeout 8`

Časová prodleva 8 vteřin. Pokud uživatel nezvolí jinak, zavede se implicitní volba.

Obsáhlejší druhá část definuje zavádění jednotlivých operačních systémů:

- První položka (`title linux`) nastavuje zavádění systému SUSE Linux. Jádro (`vmlinux`) je uloženo na prvním disku na prvním logickém oddílu (v tomto případě zaváděcí oddíl). Následné parametry blíže určují kořenový oddíl a mód zobrazení při startování jádra. Kořenový oddíl je uveden podle Linuxové konvence, protože bude interpretován samotným jádrem (a ne programem GRUB). Obraz `initrd` je uložen na stejném logickém oddíle prvního disku.
- Druhá položka (`title windows`) je odpovědná za zavedení Windows, které jsou nainstalované na prvním oddíle prvního disku (`hd0 , 0`). Příkaz `chainloader +1` způsobí, že GRUB načte a spustí první sektor definovaného oddílu.
- Další záznam povoluje zavádění systému z disketové mechaniky bez zásahů do BIOSu.
- Položka `failsafe` zavádí jádro Linuxu s mnoha přesně specifikovanými parametry jádra, aby bylo možné zavést systém na problematickém hardwaru.

Konfigurační soubor s menu můžete kdykoliv změnit. GRUB automaticky při příštím restartu načte tyto změny ze souboru. Abyste provedli permanentní změny v nastavení zavádění systému, použijte odpovídající modul programu YaST, nebo váš oblíbený editor. Pokud chcete změnit pouze jednorázově chování programu GRUB při zavádění, využijte jeho příkazovou řádku.

## Změna položek v menu při startu

Grafické rozhraní dovoluje nejen zvolit položku pro zavedení systému (pomocí kurzorových kláves), ale umožňuje vám také zadat přídatné parametry pro jádro na příkazový řádek (pokud jste vybrali položku s Linuxem). Toto umí i LILO, avšak GRUB jde ještě

o krok dál. Pokud stisknete `[Esc]`, přepnete se do textového módu. Nyní stiskem `[E]` vstoupíte do editovacího režimu. Zde můžete přímo měnit nastavení vybrané položky, které bude platné pouze pro toto zavádění systému. Žádná změna se nezapiše do souboru.

---

### Důležité: Rozložení klávesnice během fáze zavádění

V době zavádění systému můžete použít pouze americké rozložení klávesnice. Dejte pozor na jiné umístění znaků.

---

Po zapnutí režimu editace použijte kurzorové klávesy pro výběr položky, kterou chcete upravit. Nyní stiskněte `[E]`. Upravte parametry (diskové oddíly, cesty k souborům), které mají chybné hodnoty a ovlivňují proces zavádění. Opusťte režim editace stiskem `[Enter]` a jděte zpět do menu, kde můžete spustit zavádění systému s upravenými parametry. GRUB zobrazuje v dolní části obrazovky rady ohledně dalších možných činností.

Aby byly změny trvalé, upravte soubor `menu.lst` jako uživatel `root`, a přidejte libovolné parametry jádra oddělené mezerou na konec existujícího řádku:

```
title linux
  kernel (hd0,0)/vmlinuz root=/dev/hda3 parametry_jadra
  initrd (hd0,0)/initrd
```

Při příštím startování systému GRUB použije tyto nové parametry. Další možností, jak předat jádru přídavné parametry, je pomocí modulu programu YaST. Veškeré argumenty napište na konec řádku, oddělené mezerou.

## Zástupné znaky a zadání jádra ke spuštění

Pokud se podílíte na vývoji jádra nebo používáte jádro vlastní, musíte, aby se systém správně spouštěl, buď změnit položky v `menu.lst` nebo zadat příslušné parametry do startovacího promptu. Nyní máte možnost se těmto procedurám vyhnout použitím *zástupných znaků*. S jejich pomocí se všechna jádra vyhovující kritériím, automaticky vloží do startovací nabídky.

Pro použití zástupných znaků stačí dodržovat pravidla při pojmenování obrazů jader a `initrd` a nová položka v souboru `menu.lst`. Předpokládejme, že máme systém s jádry a příslušnými `initrd`:

```
initrd-default
initrd-test
vmlinuz-default
vmlinuz-test
```

Abyste jak `linux-default`, tak `linux-test` vložili do souboru `menu.lst` musíte zadat:

```
title linux-*
  wildcard (hd0,4)/vmlinuz-*
  kernel (hd0,4)/vmlinuz-* root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd-*
```

V tomto příkladě GRUB vyhledá dostupná jádra na oddíle (hd0,4) a doplní do souboru `menu.lst`:

```
title linux-default
  wildcard (hd0,4)/vmlinuz-default
  kernel (hd0,4)/vmlinuz-default root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd-default
title linux-test
  wildcard (hd0,4)/vmlinuz-test
  kernel (hd0,4)/vmlinuz-test root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd-test
```

Problémy mohou nastat, pokud jste obrazy jader pojmenovali jiným než obvyklým způsobem, který nevyhovuje zadaným kritériím hledání, nebo pokud některý ze souborů neexistuje. Problémy s nastavením a používáním zástupných znaků nespádají do instalační podpory.

## 9.3.2 Soubor `device.map`

Výše zmíněný soubor `device.map` mapuje zařízení pojmenovaná podle notace programu GRUB na jména podle Linuxové notace. Pokud váš systém má jak IDE tak SCSI zařízení, GRUB zkouší určit pořadí zavádění podle určitého algoritmu. Bohužel GRUB není schopen získat tuto informaci z BIOSu. Ukládá proto pořadí zařízení, ze kterých se zavádí systém do souboru `/boot/GRUB/device.map`. Na systémech kde je BIOS nastaven tak, aby zaváděl OS z IDE disků a až poté z SCSI, by soubor vypadal takto:

```
(fd0) /dev/fd0
(hd0) /dev/hda
(hd1) /dev/hdb
(hd2) /dev/sda
(hd3) /dev/sdb
```

Jestliže GRUB zavádí systém podle `device.map` a narazí na problém, zkontrolujte pořadí zařízení v tomto souboru, a případně změňte jejich pořadí v GRUB shellu. Jakmile nastartujete systém, můžete změnit pořadí v modulu konfigurace zavaděče programu YaST, nebo ve vašem oblíbeném editoru.

Po změnách provedených v souboru `device.map` musíte aktualizovat instalaci zavaděče. To provedete následujícími příkazy:

```
GRUB -batch < /etc/GRUB.conf
```

### 9.3.3 Soubor `/etc/grub.conf`

Kromě souborů `menu.lst` a `device.map` GRUB používá pro uložení svého nastavení také soubor `GRUB.conf`. V tomto souboru jsou uložena data o místech, kam má příkaz GRUB uložit kód zavaděče:

```
root (hd0,4)
install /GRUB/stage1 d (hd0) /GRUB/stage2 0x8000
(hd0,4)/GRUB/menu.lst
quit
```

Druhá a první řádka jsou napsané v jedné řádce. Jednotlivé údaje mají následující význam:

```
root (hd0,4)
```

Tato položka říká programu GRUB, že veškeré následující příkazy se týkají prvního logického oddílu na prvním disku, na kterém jsou uloženy soubory pro zavádění.

```
install parametr
```

Zde se říká, že GRUB má spustit svůj interní příkaz `install` a určuje, kam uložit kód. Zavaděč prvního stupně zapsat do MBR prvního disku (`/GRUB/stage1 d (hd0)`), a na paměťovou adresu `0x8000` nahrát zavaděč druhé fáze (`/GRUB/stage2 0x8000`). Poslední parametr (`(hd0,4)/GRUB/menu.lst`) ukazuje, kde je uložen soubor s menu.

### 9.3.4 GRUB shell

GRUB sestává ze dvou částí: zavaděče a běžného Linuxového programu (`/usr/sbin/GRUB`). Tomuto programu se také říká `GRUB shell`. Program obsahuje interní příkazy pro zapsání kódu zavaděče na disk nebo disketu (`install` a `setup`). Jinými slovy, tyto vnitřní příkazy můžete spustit v rámci GRUB shellu na běžícím Linuxovém stroji. Nicméně tyto příkazy jsou také dostupné během zavádění pomocí programu GRUB - ještě před tím, než je nastartován Linux. Díky tomu je mnohem jednodušší opravit vadný systém.

Výše zmíněný algoritmus pro mapování zařízení se použije pouze tehdy, pokud GRUB spouští svůj shell. GRUB načte soubor `device.map` a namapuje jména používaná programem GRUB na Linuxová jména. Každé zařízení je na jednom řádku. Pokud máte potíže se zaváděním systému, zkontrolujte zda pořadí zařízení uvedených v `device.map` koresponduje s nastavením v BIOSu počítače. Soubor najdete v adresáři `/boot/GRUB/`. Chcete-li vědět o tomto tématu více, přečtěte si sekci [9.3.2 – „Soubor device.map“](#) (strana 172).

## 9.3.5 Nastavení hesla pro zavádění

Protože GRUB umí během zavádění přistupovat na různé souborové systémy, můžeme ho použít i pro čtení souborů, které by za normálních okolností nebyly přístupné - na běžícím systému by uživatel potřeboval mít oprávnění uživatele `root`. Abyste tomuto zamezili, nastavte si heslo pro zavaděč GRUB. Tímto můžete zabránit neautorizovaným osobám v přístupu k souborům během zavádění, a předejít zavedení jiného než implicitního operačního systému.

---

### Důležité: Startovací heslo a splash

Pokud pro GRUB nastavíte heslo, není při startu zobrazen standardní splash.

---

Heslo vytvoříte tak, že se přihlásíte jako `root` a provedete následující kroky:

1. Spustíte GRUB shell a zašifrujete heslo:

```
GRUB> md5crypt
Password: ****
Encrypted: $1$1S2dv/$JOYcdxIn7CJk9xShzzJVw/
```

2. Vložte zašifrovaný řetězec do globální sekce souboru `menu.lst`:

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8
password -md5 $1$1S2dv/$JOYcdxIn7CJk9xShzzJVw/
```

Od teď nelze spouštět příkazy programu GRUB při zavádění systému bez znalosti hesla. Oprávnění získáte po stisknutí **P** a zadání hesla. Uživatelé ale stále mohou zavádět libovolné nainstalované OS bez omezení.

3. Abyste zamezili zavedení některých operačních systémů, přidejte ke každé položce, kterou chcete mít chráněnou heslem, řádek `lock`. Jako v následujícím příkladě:

```
title linux
                kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
                initrd (hd0,4)/initrd
                lock
```

Po restartování počítače se při pokusu o zavedení OS z takto označené položky zobrazí chybová hláška:

```
Error 32: Must be authenticated
```

Česky tedy:

```
Chyba 32: Musíte zadat heslo
```

Vraťte se do menu stisknutím `Enter`. Zde stiskněte `P` a zadejte heslo. Vybraný OS (v našem případě Linux) se zavede po zadání hesla.

---

### **Důležité: Heslo pro zavádění a úvodní obrazovka**

Nastavení hesla vypne implicitní zobrazování grafické úvodní obrazovky (boot splash screen).

---

## **9.4 Odinstalace zavaděče LILO nebo GRUB**

Při odinstalaci programů GRUB a LILO se do zaváděcího sektoru (kde sídlí zavaděč) musí nahrát původní obsah. SUSE Linux uchovává platnou původní zálohu obsahu tohoto sektoru. YaST modul pro zavaděče lze použít pro vytvoření zálohy, integraci této zálohy do menu zavaděče a nebo pro obnovení standardního MBR. Tento modul je popsán v kapitole věnující se instalaci systému.

---

### **Varování**

Záloha zaváděcího sektoru se stane neplatnou, jestliže na oddíl kde leží zavaděcí sektor nainstalujeme nový souborový systém. Tabulka rozdělení diskových oddílů v záloze MBR je nepoužitelná, pokud jsme od doby vytvoření zálohy změ-

nili rozložení oddílů. Tyto staré zálohy jsou jako časovaná bomba. Je lepší je mazat hned jak změním rozložení disku.

---

## 9.4.1 Obnova MBR (DOS, Win9x/ME, OS/2)

Obnovit MBR DOSu, OS/2 nebo Windows je velice snadné. Pouze zadejte příkaz DOSu (který je dostupný od verze 5.0):

```
fdisk /MBR
```

nebo na OS/2:

```
fdisk /newmbr
```

Tyto příkazy zapíší do MBR pouze prvních 446 bytů (kód zaváděče) a ponechají tabulku rozdělení disků nedotčenou. Pokud však je MBR označen jako neplatný kvůli špatnému magickému číslu), nastaví se tabulka na hodnotu nula. Po obnově MBR zkontrolujte zda je požadovaný oddíl nastaven jako zaváděcí (znovu pomocí fdisk). Tento příznak požaduje kód startující DOS, Windows a OS/2.

## 9.4.2 Obnova MBR v Windows XP

Zaveďte systém z instalačního CD Windows XP a stiskněte během startu R pro spuštění konzole pro zotavení. Vyberte vaši instalaci Windows XP ze seznamu a zadejte heslo administrátora. Poté z příkazové řádky spusťte příkaz `FIXMBR` a poté potvrďte stiskem `y`. Nyní restartujte počítač pomocí příkazu `exit`.

## 9.4.3 Obnova MBR v Windows 2000

Zaveďte systém z instalačního CD Windows 2000 a stiskněte R a poté v dalším menu C. Zvolte ze seznamu vaši instalaci Windows 2000 a zadejte heslo pro administrátora. Do promptu zadejte příkaz `FIXMBR` a potvrďte tuto volbu pomocí `y`. Následně můžete restartovat počítač pomocí `exit`.



## 9.4.4 Zavedení systému Linux po obnovení MBR

Po obnovení standardního Windows MBR můžete nastavit jeden z Linuxových zavaděčů, abyste mohli dále používat instalovaný Linuxový systém.

### GRUB

I když je nainstalován v MBR, ukládá GRUB svá data pro zaváděcí fázi 1 na linuxový oddíl. Po obnovení MBR pomocí YaST nebo ve Windows s nástroji zmíněnými výše, musíte označit oddíl, kde leží GRUB, jako aktivní.

### LILO

Po obnovení MBR můžete znovu nainstalovat LILO, pokud máte uložený záložní soubor. Nejprve zkontrolujte jestli velikost souboru je přesně 512 bytů a poté obnovte sektor (nejdříve však provedeme zálohu do +jmeno-noveho-souboru). Pomocí příkazů:

- Jestliže LILO leží na oddíle yyyy (např. hda1, hda2,...):

```
dd if=/dev/yyyy of=jmeno-noveho-souboru bs=512 count=1
```

```
dd if=jmeno-souboru-se-zalohou of=/dev/yyyy
```

- Jestliže LILO leží v MBR na disku zzz (např., hda, sda):

```
dd if=/dev/zzz of=jmeno-noveho-souboru bs=512 count=1
```

```
dd if= of=jmeno-souboru-se-zalohou /dev/zzz bs=446  
count=1
```

Poslední příkaz je bezpečná verze předešlého - nepřepisuje tabulku oddílů. Nyní opět označte oddíl jako aktivní pomocí programu `fdisk`.

## 9.5 Vytvoření startovacího CD

V některých případech se může stát, že nelze systém spustit pomocí standardních zavaděčů LILO nebo GRUB na instalovaných do MBR disku. V takových případech obvykle nastupujete použití startovací diskety. U novějších jader je však vytvoření startovací diskety kvůli nedostatku místa na disketě často nemožné. Pokud máte k dispozici vypalovací mechaniku, můžete si místo startovací diskety vytvořit startovací CD.

K vytvoření startovacího CD se zavaděčem GRUB je potřeba zvláštní forma *stage2* nazývaná *stage2\_eltorito* a upravený soubor *menu.lst*. Klasické soubory *stage1* a *stage2* nejsou potřebné.

Vytvořte si adresář určený pro obsah ISO obrazu.

```
cd /tmp
mkdir iso
```

V adresáři */tmp* si vytvořte podadresář GRUB :

```
mkdir -p iso/boot/grub
```

Překopírujte soubor *stage2\_eltorito* do adresáře *grub* :

```
cp /usr/lib/grub/stage2_eltorito iso/boot/grub
```

Překopírujte jádro (*/boot/vmlinuz*), *initrd* (*/boot/initrd*) a soubor */boot/message* do adresáře *iso/boot/* :

```
cp /boot/vmlinuz iso/boot/
cp /boot/initrd iso/boot/
cp /boot/message iso/boot/
```

Aby byly tyto soubory dostupné pro GRUB, překopírujte soubor *menu.lst* do adresáře *iso/boot* a upravte jednotlivé položky tak, aby ukazovaly na CD mechaniku. To uděláte tak, že všechny odkazy na pevný disk (např. *(hd\*)*) zaměníte za jméno CD mechaniky (*(cd)*):

```
gfxmenu (cd)/boot/message
timeout 8
default 0

title Linux
    kernel (cd)/boot/vmlinuz root=/dev/hda5 vga=794 resume=/dev/hda1
    splash=verbose showopts
    initrd (cd)/boot/initrd
```

ISO můžete například vytvořit následujícím příkazem:

```
mkisofs -R -b boot/grub/stage2_eltorito -no-emul-boot \  
-boot-load-size 4 -boot-info-table -o grub.iso iso
```

Soubor `grub.iso` vypalte svým oblíbeným vypalovacím programem na CD.

## 9.6 Grafická konzole SUSE

Od verze SUSE Linux 7.2, má SUSE při nastavení parametru `vga=<value>` první konzoli grafickou. V případě instalace pomocí programu YaST se tento parametr nastaví automaticky. Grafickou konzoli lze vypnout třemi způsoby:

Vypnutí grafiky podle potřeby

V příkazové řádce zadejte příkaz `echo 0 >/proc/splash`. Opět ji aktivujete příkazem `echo 1 >/proc/splash`.

Vypnutí jako přednastavená možnost

Stačí přidat do konfigurace zavaděče parametr jádra `splash=0`. Více informací najdete v kapitole 9 – „*Starování systému a zavaděče*“ (strana 163). Pokud dáváte přednost textovém režimu z předchozích verzí, zadejte `vga=normal`.

Vypnutí úplně a na vždy

Přeložte si jádro a vypněte volbu *Use splash screen instead of boot logo* v menu *frame-buffer support*.

---

### Tip

Pokud si vypnete podporu pro framebuffer, pak se splash screen vypne automaticky. V případě, že si budete sami kompilovat jádro, nebudete pro takto upravené jádro moci využít instalační podporu.

---

## 9.7 Řešení problémů

V této části jsou popsány nejčastější problémy související s používáním zavaděče GRUB a jejich řešení. Řešení nejčastějších problémů najdete v databázi instalační podpory <http://en.opensuse.org/SDB:SDB>. Můžete použít také funkci hledání.

## GRUB a XFS

XFS neponechá na oddílu žádné místo pro `stage1`, proto nenastavujte XFS oddíl jako umístění zavaděče. Tento problém se dá vyřešit vytvořením zvláštního startovacího oddílu, který nebude naformátován na XFS.

## GRUB a JFS

Kombinace zavaděče GRUB a souborového systému JFS bývá problematická. Doporučujeme použít zvláštní startovací oddíl (`/boot`) a naformátovat jej např. na Ext2. Pak GRUB nainstalujte na tento oddíl.

## GRUB Hláška GRUB Geom Error

GRUB zjišťuje geometrii připojeného disku při startu systému. Občas BIOS vrátí nekorektní informace a GRUB nahlásí chybu `GRUB Geom Error`. V takovém případě použijte zavaděč LILO nebo proveďte update BIOSu. Podrobnější informace o tomto problému najdete v databázi instalační podpory pod klíčovým slovem LILO.

GRUB tuto chybu hlásí také v případě instalace linuxového systému na BIOSem neregistrovaném disku. `stage1` se zavede, ale `stage2` není nalezen. Tento problém vyřešíte registrací disku v BIOSu.

## Kombinovaný systém s IDE i SCSI nespustí

Během instalace může YaST špatně detekovat startovací sekvenci disků (a vy ji nemůžete opravit). Například GRUB může `/dev/hda` označit jako `hd0` a `/dev/sda` jako `hd1`, přestože je startovací sekvence v BIOSu nastavena jinak (SCSI před IDE).

V takovém případě použijte příkazovou řádku zavaděče GRUB. Trvalé změny provedete po spuštění systému editací souboru `device.map`. Pak překontrolujte jména zařízení v souborech `/boot/GRUB/menu.lst` a `/boot/GRUB/device.map` a přeinstalujte zavaděč příkazem:

```
grub -batch < /etc/grub.conf
```

## Start Windows z druhého disku

Některé operační systémy jako např. Windows umí startovat pouze z prvního disku. Pokud takový operační systém chcete nainstalovat na jiný než první disk, musíte změnit logické pořadí disků v konfiguračním souboru zavaděče.

```
...
title windows
  map (hd0) (hd1)
  map (hd1) (hd0)
```

```
chainloader(hd1,0)+1  
...
```

Ve výše uvedeném příkladu startuje Windows z druhého disku. Z tohoto důvodu je přenastaveno logické pořadí disků pomocí `map`. Tato změna nijak neovlivní soubor nabídku zavaděče GRUB, takže je nutné ještě provést zvláštní nastavení pro `chainloader`.

## 9.8 Další informace

Více informací o programu GRUB v angličtině, němčině a japonštině získáte na adrese <http://www.gnu.org/software/grub/>. Online manuál je pouze v angličtině. Můžete se také podívat na stránky podpory zákazníkům na adrese <http://en.opensuse.org/SDB:SDB> a vyhledávat informace podle klíčového slova GRUB.



# Zvláštní funkce systému SUSE Linux 10

Tato kapitola nabízí informace o různých balíčcích, virtuálních konzolích a rozložení klávesnice. Budeme zde probírat programy měnící se mezi jednotlivými verzemi jako `bash`, `cron` a `logrotate`. Kapitulu ukončíme krátkým pojednáním o jazycích a jazykově specifických nastaveních (I18N a L10N).

## 10.1 Náповěda k některým zvláštním balíčkům

Zde najdete důležité informace o balíčcích jako je například `bash` či `cron` a příkazům `ulimit`, `logrotate`, `locate` a `free`.

### 10.1.1 Balíček `bash` a `/etc/profile`

1. `/etc/profile`
2. `~/.profile`
3. `/etc/bash.bashrc`
4. `~/.bashrc`

Osobní nastavení si každý uživatel může zapsat do souboru `~/.profile` nebo do `~/.bashrc`. Aby bylo nastavení těchto souborů správné, je nezbytné zkopírovat zá-

kladní nastavení z `/etc/skel/.profile` nebo `/etc/skel/.bashrc` do domovského adresáře uživatele. Je doporučeno překopírovat `/etc/skel` ihned po updatu. Aby nedošlo k ztrátě osobních nastaveních, doporučuje se nejdříve provést následující příkazy:

```
mv ~/.bashrc ~/.bashrc.old
cp /etc/skel/.bashrc ~/.bashrc
mv ~/.profile ~/.profile.old
cp /etc/skel/.profile ~/.profile
```

Osobní nastavení je pak nutné překopírovat zpět z `*.old`.

## 10.1.2 Balíček cron

Tabulky programu cron se nyní nacházejí v `/var/cron/tabs`. `/etc/crontab` nyní slouží jako rozsáhlý konfigurační soubor systémové tabulky. Zde zadávejte jako uživatel `root` jméno počítače, který by měl v určitý čas spouštět některé příkazy podle časové tabulky. Tabulky specifické pro balíček, uložené v `/etc/cron.d`, mají stejný formát. Více v `man cron`.

Ukázka údajů v `/etc/crontab` uživatele `root`):

```
1-59/5 * * * * root test -x /usr/sbin/atrun && /usr/sbin/atrun
```

`/etc/crontab` nelze spustit příkazem `crontab -e`. Musíte jej nejdřív otevřít v editoru, pak změnit a uložit.

Mnoho balíčků instaluje skripty příkazového řádku do adresářů `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly`, a `/etc/cron.monthly`, instrukce jsou kontrolovány skriptem `/usr/lib/cron/run-crons`. `/usr/lib/cron/run-crons` je z hlavní systémové tabulky (`/etc/crontab`) spouštěn každých patnáct minut. To zajistí spuštění všech správných procesů včas.

úlohy, které jsou vykonávány denně, jsou z důvodů přehlednosti rozděleny do několika samostatných skriptů (`aaa_base`, `/etc/cron.daily` obsahují komponenty `backup-rpmdb`, `clean-tmp`, či `clean-vi`).

## 10.1.3 Soubory logů: logrotate a balíčky

V systému je spuštěno mnoho služeb (*démonů*), které pravidelně zaznamenávají stav systému a určitých událostí do záznamů (logovacích souborů). Tímto způsobem může



administrátor pravidelně zkontroluje stav systému v určitém časovém okamžiku, najít problémy a chyby funkcí, řešit a ladit je s velkou precizností. Záznamy - logy jsou uloženy v adresáři `/var/log`, přesně dle specifikace FHS a denně nabírají nové a nové záznamy. Balíček `logrotate` umožňuje zvýšení počtu těchto souborů a lepší kontrolu systému.

## Změny v logrotate

Tato stará nastavení budou změněna při updatu z verze starší než SUSE Linux 8.0:

- Položky `/etc/logfile` neasociované s některým balíčkem jsou přesunuty do `/etc/logrotate.d/aaa_base`.
- Proměnná `MAX_DAYS_FOR_LOG_FILES` ze souboru jako je `rc.config` je mapována v konfiguračním souboru jako `dateext` a `maxage`. Více v `man logrotate`.

## Nastavení

Nastavení `logrotate` je uloženo v souboru `/etc/logrotate.conf`. Položka `include` specifikuje další soubory pro čtení. SUSE Linux zajišťuje instalování jednotlivých balíčků do `/etc/logrotate.d` např. `apache2` (`/etc/logrotate.d/apache2`) `syslog` (`/etc/logrotate.d/syslog`).

Příklad z `/etc/logrotate.conf`:

```
# podívejte se na "man logrotate" pro další detaily
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own lastlog or wtmp - we'll rotate them here
#/var/log/wtmp {
```

```
# monthly
# create 0664 root utmp
# rotate 1
#}
```

```
# system-specific logs may be also be configured here.
```

logrotate je kontrolován programem cron a bývá volán denně. `/etc/cron.daily/logrotate`.

---

### Důležité

Volba `create` umožňuje načíst veškerá nastavení vytvořená administrátorem v souboru `/etc/permissions*`. Zajišťuje, že nedojde ke konfliktu žádných nastavení.

---

## 10.1.4 Příkaz locate

`locate`, příkaz pro rychlé vyhledávání souborů, není součástí standardní instalace. Pokud jej chcete používat, nainstalujte si balíček `find-locate`. Příkaz pro vytváření databáze `updatedb` se spouští automaticky každý den v noci nebo zhruba 10 minut po spuštění systému.

## 10.1.5 Manuálové stránky

Manuálové stránky GNU aplikací (jako je např. `tar`) již nejsou delší dobu spravovány a aktualizovány. Byly nahrazeny info stránkami. Pro zjištění základních příkazů programů, použijte parametr `--help`, který poskytuje rychlý přehled info stránek. Ty ovšem poskytují mnohem hlubší pohled na jednotlivé možnosti programů a vysvětlují příkazové instrukce. `info` je hypertextový systém vyvíjený v rámci projektu GNU. Úvod do info stránek zobrazíte jednoduchým vypsáním `info info` na příkazovou řádku. Info stránky lze prohlížet editorem Emacs, i přímo při jeho spuštění pomocí `emacs -f info`, nebo přímo v konzoli příkazem `info`. Programy jako `tinfo`, `xinfo`, lze jednoduše prohlížet pomocí nápovědy SUSE.

## 10.1.6 Příkaz ulimit

Díky příkazu `ulimit` (*user limits*) je možné nastavit využívání zdrojů systému a zároveň si je nechat zobrazit. `ulimit` je užitečný zvláště pro omezení paměti využívané aplikacemi. Můžete zabránit aplikaci v nadměrném čerpání zdrojů, aby nemohlo dojít k zamrznutí systému.

`ulimit` může být používán s mnoha volbami. Využívání paměti omezíte některou z voleb z tabulky [10.1 – „ulimit: Přidělení zdrojů uživateli“](#) (strana 187).

**Tabulka 10.1** *ulimit: Přidělení zdrojů uživateli*

---

-m	maximální velikost fyzické paměti
-v	maximální velikost virtuální paměti
-s	maximální velikost zásobníku
-c	maximální velikost core souborů
-a	zobrazení limitů

---

Nastavní platná pro celý systém zapisujte do `/etc/profile`. Zde musíte povolit vytváření core souborů, které jsou potřebné při *ladění*. Normální uživatelé hodnoty uvedené v `/etc/profile` měnit nemohou, mohou si ovšem vytvořit speciální nastavení ve vlastním `~/ .bashrc`.

Příklad omezení paměti v `~/ .bashrc`:

```
# Omezení fyzické paměti:  
ulimit -m 98304  
  
# Omezení virtuální paměti:  
ulimit -v 98304
```

Velikost paměti musí být zadána v KB. Více informací najdete v `man bash`.

---

## Důležité

Některé shelly příkaz `ulimit` nepodporují. V tom případě využijte PAM `pam_limits`, který nabízí podobné možnosti pro omezování přidělených prostředků.

---

### 10.1.7 Příkaz `free`

Pokud chcete zjistit, kolik paměti RAM je momentálně používáno, může být výstup programu `free` trochu matoucí. Podstatné informace naleznete v souboru `/proc/meminfo`. V moderních operačních systémech jako je Linux, se již uživatelé nedostatku paměti nemusí obávat. Koncepce *dostupné RAM* zdědil Linux z období řízení unifikovaného přístupu k paměti. Slogan *volná paměť je špatná paměť* padne Linuxu jako ulitý. Výsledkem je vlastnost systému, kdy je nesmyslné být i jen mluvit o volné či nepoužívané paměti.

Jádro v podstatě nemá přímé informace o aplikačních či uživatelských datech. Místo toho obsluhuje aplikace a uživatelská data pomocí *stránkování*. V případě nedostatku paměti budou načítány na odkládací oddíl nebo do souborů, ze kterých je možné je číst příkazem `mmap`. (viz `man mmap`).

Jádro obsahuje také jiné cache, např. *slab cache*, používanou pro uložení síťového přístupu. To může vést k situaci, kdy jsou informace v souboru `/proc/meminfo` odlišné od reality. K většině, ale ne ke všem, lze přistupovat přes `/proc/slabinfo`.

### 10.1.8 Soubor `/etc/resolv.conf`

Rozpoznávání doménových jmen je řešeno souborem `/etc/resolv.conf`. Více najdete v kapitole 20 – „DNS — Domain Name System“ (strana 323).

Soubor je aktualizován výlučně skriptem `/sbin/modify_resolvconf`, což znamená, že žádný jiný program nesmí soubor `/etc/resolv.conf` upravovat přímo. Přísnost tohoto pravidla zaručuje konzistentní stav konfigurace sítě.

## 10.1.9 Nastavení programu GNU Emacs

GNU Emacs je komplexním pracovním prostředím. Více informací je k dispozici na <http://www.gnu.org/software/emacs/>. Následující sekce popisují konfigurační soubory načítané při startu GNU Emacs.

Během startu načítá Emacs množství souborů s nastaveními uživatele, administrátora systému a distributora lokalizace a předkonfigurovaných vlastností. Inicializační soubor `~/ .emacs` se nainstaluje do home adresářů uživatelů z adresáře `/etc/skel/`. `.emacs` posléze čte ze souboru `/etc/skel/ .gnu-emacs`. Pro vlastní úpravy programu by si měl uživatel zkopírovat `.gnu-emacs` do svého domovského adresáře. Požadované změny by měl provést zde:

```
cp /etc/skel/.gnu-emacs ~/.gnu-emacs
```

`.gnu-emacs` definuje soubor `~/ .gnu-emacs-custom` jako `custom-file`. V případě, že uživatel dělá změny nastavení pomocí `customize` nastavení, ukládají se pak tyto změny do `~/ .gnu-emacs-custom`.

V systému SUSE Linux, emacs balíček instaluje soubor `site-start.el` do adresáře `/usr/share/emacs/site-lisp`. Soubor `site-start.el` je načítán ještě *předtím*, než je načten konfigurační soubor `~/ .emacs`. Mezi mnoha službami, které soubor `site-start.el` zajišťuje, je také automatické nahrávání speciálních konfiguračních souborů obsažených v přídatných balíčcích programu Emacs (to jsou balíčky jako např. `psgml`). Konfigurační soubory tohoto typu jsou také umístěny v `/usr/share/emacs/site-lisp`, a vždy začínají `suse-start-`. Místní administrátor může specifikovat nastavení velmi široce v souboru `default.el`.

Více informací o těchto souborech najdete v info souboru pod Emacs *Inicializačním souborem*: <info:/emacs/InitFile>. Na druhou stranu zde najdete taktéž informace o tom, jak v případě potřeby tyto soubory vypnout.

Komponenty programu Emacs jsou rozděleny do několika balíčků:

- Základní balík `emacs`.
- Obvykle by měl být instalován `emacs-x11`. Balíček obsahuje program s *podporou X11*.
- `emacs-nox` naopak podporu X11 *neobsahuje*.

- `emacs-info`: Jde o online dokumentaci v info formátu.
- `emacs-el` obsahuje nekompilevanou knihovnu souborů v jazyce Emacs Lisp. Vyžadovány pro běh programu nejsou tyto soubory nejsou.
- Množství přídatných balíčků, které instalujete v případě potřeby:

`emacs-auctex` (pro LaTeX)

`psgml` (pro SGML a XML)

`gnuserv` (pro fungování jako klient a server) a další.

## 10.2 Virtuální konzole

Linux je víceúlohový víceuživatelský systém. Tyto vlastnosti systému oceníte dokonce i na obyčejné uživatelské stanici. V textovém režimu je k dispozici šest virtuálních konzolí. Přepínání mezi nimi zajišťuje kombinace `Alt` + `F1` až `Alt` + `F6`. Sedmá konzole je rezervována pro X11. Počet konzolí je možné změnit v souboru `/etc/inittab`.

K přepnutí z X11 do konzole použijte kombinaci kláves `Ctrl` + `Alt` + `F1` až `Ctrl` + `Alt` + `F6`. Stisknutím kláves `Alt` + `F7` se vrátíte zpět do X11.

## 10.3 Mapování klávesnice

Standardizace mapování klávesnice si vynutila změny v následujících souborech:

```
/etc/inputrc
/usr/X11R6/lib/X11/Xmodmap
/etc/skel/.Xmodmap
/etc/skel/.exrc
/etc/skel/.less
/etc/skel/.lesskey
/etc/csh.cshrc
/etc/termcap
/usr/lib/terminfo/x/xterm
/usr/X11R6/lib/X11/app-defaults/XTerm
/usr/share/emacs/<VERSION>/site-lisp/term/*.el
```

Změny se týkají pouze aplikací, které používají `terminfo` nebo těch aplikací, jejichž konfigurační soubory se mění v systému přímo, jako je (`vi`, `less`, atd.). Ostatní aplikace ne od SUSE by měly být přizpůsobeny tomuto původnímu nastavení.

Pod systémem X může být ovládání pomocí (klávesových zkratk) zpřístupněno přes kombinaci kláves `Ctrl` + `Shift` (pravý). Podívejte se na příslušný příkaz v souboru `/usr/X11R6/lib/X11/Xmodmap`.

Další nastavení jsou možná přes "X rozšíření klávesnice" (XKB). Toto rozšíření používají také prostředí GNOME (`gswitchit`) a KDE (`kxkb`). Informace o XKB najdete v souboru `/etc/X11/xkb/README` a dalších dokumentech v adresáři.

Detailní informace o čínských, japonských a korejských (CJK) specifických klávesových zkratkách najdete na stránkách Mika Fabiana zde: <http://www.suse.de/~mfabian/suse-cjk/input.html>.

## 10.4 Lokální přizpůsobení — I18N and L10N

SUSE Linux je mezinárodní systém, který se dá velmi flexibilně přizpůsobit lokálním potřebám. Internacionální charakter (*I18N*) jinými slovy umožňuje specifický přístup k lokalizaci (*L10N*).

Lokální nastavení pro národní jazyky je zajištěno proměnnými `LC_` definovanými v souboru `/etc/sysconfig/language`. Nejde přitom pouze o určení jazyka pro komunikaci s jednotlivými aplikacemi a *prostředí programů v původním jazyce*, ale také o *zprávy systému, znakové sady, pořadí při abecedním třídění, formát časových údajů, desetinných čísel a peněžních částek*. Každou z těchto kategorií můžete definovat přímo její proměnnou, nebo nepřímo hlavní proměnnou v souboru `language` (podívejte se na manuálové stránky `man locale`).

1. `RC_LC_MESSAGES`, `RC_LC_CTYPE`, `RC_LC_COLLATE`, `RC_LC_TIME`, `RC_LC_NUMERIC`, `RC_LC_MONETARY`: Tyto proměnné se exportují do prostředí příkazového interpretu bez předpony `RC_` a určují jednotlivé z lokalizačních kategorií. Soubory, kterých se to týká najdete v seznamu níže. Nastavení proměnných zjistíte výpisem příkazu `locale`.

2. `RC_LC_ALL`: Pokud je nastavena tato proměnná, přepíše svou hodnotou výše uvedené proměnné.
3. `RC_LANG`: Pokud není nastavena žádná z výše uvedených proměnných, je výchozí hodnotou. Defaultně SUSE Linux nastavuje pouze `RC_LANG`. Tato vlastnost pomáhá uživatelům zavést své vlastní hodnoty.
4. `ROOT_USES_LANG`: V případě nastavení na `no`, `root` pracuje `root` v prostředí standardu POSIX.

Ostatní proměnné můžete nastavit YaSTem v editoru souborů `sysconfig`. Hodnota těchto proměnných obsahuje kód jazyka, země, kódování a modifikátor. Individuální komponenty jsou připojitelné pomocí speciálních znaků:

```
LANG=<language>[[<_<COUNTRY>].<Encoding>[@<Modifier>]]
```

## 10.4.1 Některé příklady

Nastavení jazyka a kódů země by mělo jít ruku v ruce. Jazyková nastavení jsou dle standardu ISO 639 (<http://www.evertype.com/standards/iso639/iso639-en.html> a <http://www.loc.gov/standards/iso639-2/>). Kódy zemí naleznete v ISO 3166, podívejte se na ([http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/en\\_listp1.html](http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/en_listp1.html)). Smysl má nastavit hodnoty, jejichž popis užití naleznete v `/usr/lib/locale`. Další soubory s popisy můžete vytvořit ze souborů v adresáři `/usr/share/i18n` použitím příkazu `localedef`. Soubor s popisem `cs_CZ.UTF-8` (pro naši krásnou zemi) vytvoříte takto:

```
localedef -i cs_CZ -f UTF-8 cs_CZ.UTF-8
```

```
LANG=cs_CZ.UTF-8
```

Toto je defaultní nastavení, když je v průběhu instalace vybrána čeština. Jestliže zvolíte jiný jazyk, bude tento jazyk také s kódováním UTF-8.

```
LANG=cs_CZ.ISO-8859-2
```

Takto nastavíme proměnnou na češtinu, zemi na Českou republiku a znakovou sadu na ISO-8859-2. Řetězec definující znakovou sadu, kterou je v našem případě ISO-8859-2 pak bude načítán programy jako je Emacs.

SuSEconfig čte proměnnou ze souboru `/etc/sysconfig/language` a zapisuje nezbytné změny do `/etc/SuSEconfig/profile` a do `/etc/SuSEconfig/`



`cs.cshrc`. Pak přečte `/etc/SuSEconfig/profile`, nebo data načte *ze zdroje*, kterým je `/etc/profile`. `/etc/SuSEconfig/csh.cshrc` hledá svůj zdroj v `/etc/csh.cshrc`. Toto uspořádání umožňuje široké spektrum nastavení i pro velký systém.

Uživatelé také mohou přepisovat původní hodnoty v systému editací `~/ .bashrc` ve svém home adresáři. Jako příklad je možné uvést zobrazování programových hlášek v češtině `cs_CZ` do španělštiny, což znamená použít `LC_MESSAGES=es_ES`.

## 10.4.2 Nastavení jazykové podpory

Dle pravidel pro kategorii *Messages* pak systém zprávy ukládá v příslušném jazykovém adresáři (v našem případě `cs`) jako zálohu. Jestliže nastavíte `LANG` na `cs_CZ` a soubor *zpráv* ukládáte do `/usr/share/locale/en_US/LC_MESSAGES`, který neexistuje, systém jej bude dále ukládat do souboru `/usr/share/locale/en/LC_MESSAGES`.

Řetěz zálohových souborů můžete nadefinovat např. pro slovenštinu a češtinu, či pro galštinu, španělštinu a portugalštinu:

```
LANGUAGE="cs_SK:cs_CZ"
```

```
LANGUAGE="gl_ES:es_ES:pt_PT"
```

Jestli toužíte po tradiční norštině *nynorsk* a *bokmål* namísto a s dodatečnou zálohou pro `no`), proveďte úpravu proměnné do tohoto tvaru:

```
LANG="nn_NO"
```

```
LANGUAGE="nn_NO:nb_NO:no"
```

or

```
LANG="nb_NO"
```

```
LANGUAGE="nb_NO:nn_NO:no"
```

Poznamenejme nakonec, že v norštině je odlišný i parametr `LC_TIME`.

## Vyskytující se problémy

Pro správnou práci s desetinnými čísly v češtině nestačí pouze nastavit proměnnou LANG na cs. Aby např. knihovna glibc našla správnou hodnotu v souboru `/usr/share/locale/en_US/LC_NUMERIC`, je třeba nastavit přímo proměnnou LC\_NUMERIC na hodnotu `cs_CZ`.

## Další informace

- *The GNU C Library Reference Manual*, Kapitola *Locales and Internationalization*, kterou najdeme v `glibc-info`.
- Markus Kuhn, *UTF-8 and Unicode FAQ for Unix/Linux*, na <http://www.cl.cam.ac.uk/~mgk25/unicode.html>.
- *Unicode-Howto*, autor Bruno Haible je v souboru `file:/usr/share/doc/howto/en/html/Unicode-HOWTO.html`.

# Obsluha tisku

CUPS je standardní tiskový systém v systému SUSE Linux a je vysoce uživatelsky orientovaný. V mnoha případech je kompatibilní s LPRng nebo ho je možno poměrně jednoduše přizpůsobit. LPRng je v systému obsažen pouze z důvodů kompatibility .

Tiskárny je možno rozlišovat na základě jejich rozhraní, jako např. USB tiskárny či síťové tiskárny, nebo podle tiskových jazyků. Při nákupu tiskárny se ujistěte, zda je tiskárna vybavena vhodným podporovaným rozhraním a tiskovým jazykem. Podle tiskového jazyka lze tiskárny rozdělit do následujících třech tříd:

## Postscriptové tiskárny

PostScript je tiskový jazyk, ve kterém se v Linuxu a Unixu zpracovává většina tiskových úloh a který je podporován interním tiskovým systémem. Je to jazyk poměrně starý a velmi efektivní. Pokud umí tiskárna zpracovat přímo postscriptové soubory a není nutné je převádět přes další meziformáty, velmi se snižuje riziko chyb. Protože jsou postscriptové tiskárny zatíženy vysokými licenčními poplatky, jsou obvykle o něco dražší než tiskárny bez podpory tohoto jazyka.

## Standardní tiskárny (jazyky typu PCL a ESC/P)

Ačkoliv i tyto jazyky jsou poměrně staré, stále se vyvíjejí, aby pokryly nové vlastnosti tiskáren. V případě známých jazyků může tiskový systém pomocí Ghostscriptu konvertovat postscriptové úlohy do patřičného jazyka. Tento proces se označuje jako interpretace. Nejznámější jazyky jsou PCL (užívaný zejména tiskárnami HP a jejich klony) a ESC/P (používaný tiskárnami Epson). Jsou obvykle v Linuxu podporovány a tiskový výstup je kvalitní. Linux nicméně nemusí podporovat některé nové a zvláštní vlastnosti tiskáren. S výjimkou ovladačů `hpijs` vyvíjených HP v současnosti žádní výrobci tiskáren nedodávají linuxové ovladače

dostupné pod opensource licencí. Cena těchto tiskáren se pohybuje ve střední kategorii.

Proprietární tiskárny (obvykle GDI tiskárny)

Pro proprietární tiskárny je obvykle k dispozici pouze ovladač pro operační systém Windows. Nepodporují žádný běžný tiskový jazyk a jazyky, které užívají, se mění s každým novým modelem tiskárny. Viz 11.7.1 – „Tiskárny bez podpory standardního tiskového jazyka“ (strana 210).

Před nákupem nové tiskárny si projděte následující informační zdroje a ověřte si, jak dobře je v Linuxu podporována.

- <http://cdb.suse.de/> nebo — databáze tiskáren pro SUSE Linux
- <http://www.linuxprinting.org/> — databáze tiskáren na LinuxPrinting.org
- <http://www.cs.wisc.edu/~ghost/> — stránky projektu Ghostscript
- `/usr/share/doc/packages/ghostscript/catalog.devices` — ovladače obsažené v systému

Online databáze obsahují vždy aktuální informace o podpoře jednotlivých tiskáren v Linuxu. Distribuce však může obsahovat pouze ovladače dostupné před jejím vydáním. Navíc tiskárny, které jsou dnes označeny jako *perfectly supported* (výborně podporované), nemusely takovou podporu mít v době vydání distribuce. Proto databáze nemusí vždy přesně odpovídat podpoře tiskáren v distribuci SUSE Linuxu.

## 11.1 Práce tiskového systému

Uživatel vytvoří tiskovou úlohu. Tisková úloha sestává z dat, která se mají vytisknout, a z informací pro spooler, jako je jméno tiskárny nebo tiskové fronty, a, volitelně, informací pro filtr, jako jsou volby specifické pro tiskárnu.

Každá tiskárna má vlastní tiskovou frontu. Spooler drží tiskovou úlohu ve frontě, dokud požadovaná tiskárna není připravená přijmout data. Jakmile je tiskárna připravená, pošle jí spooler data skrze filtr a backend.

Filtr zkonvertuje data, která chce uživatel vytisknout (ASCII, PostScript, PDF, JPEG atd.) do dat určených pro tiskárnu. (PostScript, PCL, ESC/P atd.). Vlastnosti tiskárny jsou popsány v PPD souborech. PPD soubor obsahuje volby a parametry specifické pro daný typ tiskárny. Filtr zajistí, aby byly volby vybrané uživatelem zapnuty.

Pokud používáte postscriptovou tiskárnu, zkonvertuje filtr data do PostScriptu specifického pro tiskárnu. To nevyžaduje tiskový ovladač. Pokud používáte nepostscriptovou tiskárnu, zkonvertuje filtr data do formátu specifického pro tiskárnu pomocí programu Ghostscript. To vyžaduje použití ghostscriptového tiskového ovladače vhodného pro vaši tiskárnu. Backend přijme data specifická pro tiskárnu a odešle je tiskárně.

## 11.2 Způsoby a protokoly pro připojení tiskáren

Existuje mnoho různých možností, jak připojit tiskárnu k počítači. Konfigurace systému CUPS nerozlišuje mezi lokálními a síťovými tiskárnami. Lokální tiskárny musí být připojeny tak, jak popisuje jejich výrobce v dodaném manuálu. CUPS podporuje připojení přes sériové, USB, paralelní a SCSI rozhraní. Více informací o připojování tiskáren naleznete v článku *CUPS in a Nutshell* v databázi podpory na adrese <http://en.opensuse.org/SDB:SDB>. Článek naleznete vyhledáním termínu *cups* ve vyhledávacím dialogu.

---

### Varování: Kabelové připojení k počítači

Při připojování tiskárny k počítači pamatujte na to, že pouze USB zařízení mohou být připojována či odpojována za provozu. Před změnou jiných typů připojení by měl být systém vypnut.

---

## 11.3 Instalace softwaru

PPD (PostScript Printer Description) je počítačový jazyk popisující vlastnosti postscriptových tiskáren, např. rozlišení a další možnosti, jako je duplexní jednotka. Pro využití různých vlastností tiskáren v systému CUPS je takový popis nutný. Bez souboru PPD by byla data odeslána tiskárně v nezpracovaném stavu, což je obvykle nežádoucí. Během

instalace systému SUSE Linux je předinstalováno množství PPD souborů, které umožňují použít i tiskárny bez podpory jazyka PostScript.

Nejlépeším způsobem konfigurace postscriptové tiskárny je získání patřičného PPD souboru. Mnoho jich je dostupných v balíčku `manufacturer-PPDs`, který je součástí standardní instalace (viz 11.6.4 – „PPD soubory v různých balíčcích“ (strana 208) a 11.7.2 – „Pro postscriptovou tiskárnu není k dispozici vhodný PPD soubor“ (strana 211)).

Nové PPD soubory lze ukládat do adresáře `/usr/share/cups/model/` nebo je přidat do tiskového systému pomocí nástroje YaST (viz „Ruční konfigurace“ (strana 199)). Pak je možné vybraný PPD soubor zvolit při instalaci tiskárny.

Pokud výrobce tiskárny chce instalovat celé softwarové balíčky, nikoliv pouze modifikovat konfigurační soubory, buďte velmi opatrní. Taková instalace znamená nejen ztrátu podpory poskytované SUSE, ale také může změnit funkci tiskových příkazů a způsobit nefunkčnost při práci se zařízeními jiných výrobců. proto takovou instalaci nedoporučujeme.

## 11.4 Konfigurace tiskárny

Po připojení tiskárny k počítači a instalaci softwaru musíte tiskárnu nainstalovat do systému. To by mělo být provedeno nástroji dodanými se systémem SUSE Linux. Protože SUSE Linux klade velký důraz na bezpečnost, mají nástroje třetích stran často potíže s bezpečnostními nastaveními a působí mnohdy více potíží než užitku.

### 11.4.1 Lokální tiskárny

Pokud je při vašem přihlášení rozpoznána nenakonfigurovaná lokální tiskárna, spustí se pro její konfiguraci YaST. Dialogy jsou stejné jako v následujícím popisu konfigurace.

Chcete-li nakonfigurovat tiskárnu, zvolte v nástroji YaST *Hardware* → *Tiskárna*. Tím se otevře hlavní okno pro konfiguraci tiskárny, v jehož horní části je zobrazen seznam rozpoznávaných zařízení. V dolní části jsou zobrazeny již nakonfigurované fronty. Pokud nebyla vaše tiskárna rozpoznána, nastavte ji ručně.

---

## Důležité

Pokud YaST neobsahuje položku *Tiskárna*, není zřejmě nainstalován balíček `yast2-printer`. Doinstalujte ho a restartujte YaST.

---

## Automatická konfigurace

Pokud lze tiskárnu automaticky rozpoznat, umí ji YaST automaticky nakonfigurovat. Je však zapotřebí, aby databáze tiskáren obsahovala ID tiskárny, kterou YaST rozpoznal. Pokud se ID liší, vyberte model tiskárny ručně.

Každá konfigurace by měla být otestována pomocí testovací funkce YaSTu. Vytisknutá testovací stránka obsahuje důležité informace o testované konfiguraci.

## Ruční konfigurace

Pokud vás automatická konfigurace z nějakého důvodu neuspokojuje, nastavte tiskárnu ručně.

Je nutné nastavit následující parametry:

### Způsob připojení (Port)

Konfigurace hardwarového připojení závisí na tom, zda byl YaST schopný tiskárnu automaticky rozpoznat. Pokud se tak stalo, dá se předpokládat, že připojení je na hardwarové úrovni v pořádku a není třeba ho dále nastavovat. Pokud YaST tiskárnu nerozpoznal, může to znamenat problém s hardwarovým připojením. Pak je nutné připojení upravit manuálně.

### Jméno fronty

Jméno fronty se používá při vydávání tiskových příkazů. Mělo by být relativně krátké a skládat se pouze z malých písmen a číslic.

### Model tiskárny a PPD soubor

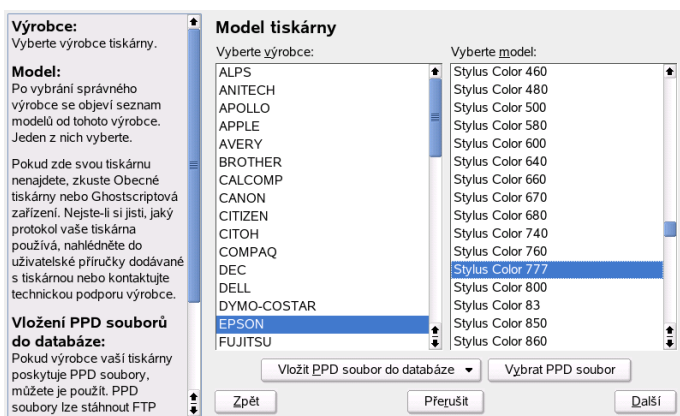
Všechny parametry specifické pro model tiskárny, jako typ používaného Ghostscript ovladače nebo filtrační parametry ovladače, jsou uloženy v PPD souboru (PostScript Printer Description). Viz [11.3 – „Instalace softwaru“](#) (strana 197).

Pro mnoho typů tiskáren je dostupných více PPD souborů, například tehdy, když s daným modelem funguje více Ghostscript ovladačů. Při výběru výrobce a modelu

tiskárny YaST sám zvolí vhodný PPD soubor. Pokud je pro tiskárnu k dispozici více PPD souborů, vybere YaST obvykle ten, který je označen jako doporučený (*recommended*). Tento výchozí PPD soubor můžete změnit po kliknutí na *Upravit*.

V případě nepostscriptových tiskáren jsou všechna data specifická pro tiskárnu vytvářena Ghostscript ovladačem. Proto je nastavení ovladače nejdůležitějším faktorem ovlivňujícím kvalitu tiskového výstupu. Tisk je ovlivněn jak druhem Ghostscript ovladače (PPD souboru), tak i pro něj nastavenými volbami. Pokud je to nutné, změňte další volby (dostupné díky PPD souboru) po kliknutí na *Upravit*.

**Obrázek 11.1** Výběr modelu tiskárny



Nastavení tisku vždy zkontrolujte vytištěním testovací stránky. Pokud je výstup špatný, například obsahuje několik prázdných stránek, zastavte tisk odstraněním papírů z tiskárny a následným přerušением tisku v YaSTu.

Pokud databáze tiskáren neobsahuje vaši tiskárnu, můžete přidat nový PPD soubor kliknutím na *Vložit PPD soubor do databáze* nebo použít některý z obecných PPD souborů a zprovoznit tiskárnu pomocí standardního tiskového jazyka. Učiníte tak volbou výrobce tiskárny *UNKNOWN MANUFACTURER* (neznámý výrobce).

### Pokročilé nastavení

Za běžných okolností není třeba do pokročilého nastavení zasahovat.



## Konfigurace tiskárny pomocí příkazové řádky

Chcete-li tiskárnu konfigurovat ručně pomocí nástrojů pro příkazovou řádku, které jsou popsány v části „[Konfigurace pomocí nástrojů pro příkazovou řádku](#)“ (strana 203), potřebujete URI (Uniform Resource Identifier) zařízení. To se skládá z backendu, například `usb`, a parametrů, jako `/dev/usb/lp0`. Plné URI může například být `parallel:/dev/lp0` (tiskárna na prvním paralelním portu) nebo `usb:/dev/usb/lp0` (první rozpoznaná tiskárna na USB portu).

### 11.4.2 Síťové tiskárny

Síťová tiskárna může podporovat různé protokoly, někdy dokonce více protokolů najednou. Přestože je většina protokolů standardizována, někteří výrobci protokoly modifikují, protože chtějí nabídnout funkce, které standard nepodporuje. Nabídnou k tiskárně ovladače pro několik málo systémů, na nichž tak odstraní problémy s protokolem. Bohužel, linuxové ovladače jsou dodávány jen zřídka. V současné době nelze předpokládat, že v Linuxu bude fungovat libovolný protokol. Proto je někdy k dosažení funkčnosti třeba experimentovat s nastavením.

CUPS podporuje protokoly `socket`, `LPD`, `IPP` a `smb`:

`socket`

*Socket* je připojení, během kterého jsou data posílána na TCP/IP soket bez předchozího navazování spojení (*handshaking*). Mezi běžně používané porty soketů se řadí 9100 a 35. Příklad URI zařízení je `socket://host-printer:9100/`.

`LPD (Line Printer Daemon)`

Spolehlivý protokol LPD je popsán v dokumentu RFC 1179. Při použití tohoto protokolu jsou některé údaje spojené s tiskovou úlohou (např. ID tiskové fronty) zasílány před vlastními tiskovými daty. Proto musí být při konfiguraci LPD protokolu pro datový přenos specifikována tisková fronta. Implementace různých výrobců jsou většinou natolik flexibilní, že je možné používat jakékoliv jméno fronty. V případě potřeby by správné jméno mělo být uvedeno v manuálu tiskárny. Obvykle se používají jména jako `LPT`, `LPT1`, `LP1` apod. LPD fronta může být samozřejmě nastavena v systému CUPS i na jiných linuxových či unixových počítačích. Číslo portu pro službu LPD je 515. Příklad URI je `lpd://host-printer/LPT1`.

### IPP (Internet Printing Protocol)

IPP je poměrně nový (1999) protokol založený na HTTP. Při použití IPP je přenášeno více dat spojených s úlohou než u jiných protokolů. CUPS používá protokol IPP pro vnitřní datové přenosy. Je to upřednostňovaný protokol pro předávací frontu mezi dvěma CUPS servery. Jméno tiskové fronty je nutno nastavit správně. Používaný port je 631. Příklad URI je `ipp://host-printer/ps` nebo `ipp://host-cupsserver/printers/ps`.

### SMB (Windows Share)

CUPS umožňuje tisk i na sdílených tiskárnách Windows. Používaný protokol je SMB. Používané porty jsou 137, 138 a 139. URI může být například `smb://Uzivatel:Heslo@PracovniSkupina/Server/Tiskarna`, `smb://Uzivatel:Heslo@Pocitac/Tiskarna` nebo `smb://Server/Tiskarna`.

Protokol, který tiskárna podporuje, musí být určen před vlastní konfigurací. Pokud výrobce potřebné informace neuvádí, lze protokol odhadnout příkazem `nmap` (balíček `nmap`). Program `nmap` hledá na tiskárně otevřené porty. Například:

```
nmap -p 35,137-139,515,631,9100-10000 IP_tiskarny
```

## 11.4.3 Konfigurace

Konfiguraci lze provést pomocí nástroje YaST nebo pomocí nástrojů pro příkazovou řádku.

### Konfigurace CUPS v síti pomocí YaST

Síťové tiskárny by měly být konfigurovány nástrojem YaST, který je nejlépe vybaven pro práci s bezpečnostními omezeními systému CUPS. (viz kapitola [11.6.2 – „Administrátor webového frontendu CUPS“](#) (strana 206)).

Více informací o instalaci CUPS v síti naleznete v článku *CUPS in a Nutshell* v databázi podpory na adrese <http://en.opensuse.org/SDB:SDB>.

# Konfigurace pomocí nástrojů pro příkazovou řádku

CUPS lze nakonfigurovat i přes příkazovou řádku nástroji jako `lpadmin` a `lpoptions`. Pokud jste již učinili přípravné práce (máte PPD soubor a znáte jméno zařízení), pokračujte následujícím způsobem:

```
lpadmin -p fronta -v URIza ízení \  
-P PPDsoubor -E
```

Volbu `-E` nepoužívejte jako první. U všech CUPS příkazů znamená `-E` jako první argument použití šifrovaného spojení. Pro zprovoznění tiskárny musí být argument `-E` použit tak jako v následujících příkladech:

```
lpadmin -p ps -v parallel:/dev/lp0 -P \  
/usr/share/cups/model/Postscript.ppd.gz -E
```

Příklad pro síťovou tiskárnu:

```
lpadmin -p ps -v socket://192.168.1.0:9100/ -P \  
/usr/share/cups/model/Postscript-levell.ppd.gz -E
```

## Úprava voleb

Během instalace systému jsou určité volby nastaveny jako výchozí. Volby lze pak pro jednotlivé tiskové úlohy měnit (v závislosti na tiskovém nástroji) nebo je měnit trvale, například pomocí YaST. Pomocí nástrojů pro příkazovou řádku toho dosáhnete následujícím způsobem:

### 1 Nejprve zobrazte všechny volby:

```
lpoptions -p fronta -l
```

Příklad:

```
Resolution/Output Resolution: 150dpi *300dpi 600dpi
```

Aktivovaná výchozí volba je označena hvězdičkou.

### 2 Změňte volbu příkazem `lpadmin`:

```
lpadmin -p fronta -o Resolution=600dpi
```

### 3 Zkontrolujte nové nastavení:

```
lptions -p fronta -l
```

```
Resolution/Output Resolution: 150dpi 300dpi *600dpi
```

## 11.5 Nastavení aplikací

Aplikace tisknou do tiskových front podobným způsobem jako příkazy z příkazové řádky. Pro tisk z aplikací není nutné přenastavovat tiskárnu, tisk bude prováděn pomocí již nastavených front.

### 11.5.1 Tisk z příkazové řádky

Pro tisk z příkazové řádky zadejte příkaz `lp -d jmeno_fronty jmeno_souboru`, kde *jmeno\_fronty* nahradíte jménem tiskové fronty, kterou chcete použít, a *jmeno\_souboru* nahradíte jménem souboru, který si přejete vytisknout.

### 11.5.2 Tisk z aplikací pomocí příkazů příkazové řádky

Některé aplikace používají pro tisk příkaz `lp`. V takovém případě do tiskového dialogu aplikace zadejte správný tiskový příkaz (obvykle bez jména *souboru*), např. `lp -d jmeno_fronty`. Aby tento postup fungoval také v programech z prostředí KDE, musíte v ovládacím centru KDE v nastavení tiskáren povolit *Tisk pomocí externího programu*. V opačném případě nelze příkaz zadat.

### 11.5.3 Použití tiskového systému CUPS

Nástroje jako `xpp` nebo `kprinter` z prostředí KDE poskytují grafické rozhraní pro výběr tiskových front, nastavení voleb systému CUPS a nastavení vlastností tiskáren pomocí PPD souboru. Aplikaci `kprinter` můžete použít jako standardní tiskové rozhraní také pro ostatní (ne z KDE) programy zadáním příkazu `kprinter` nebo `kprinter --stdin` jako tiskového příkazu v těchto aplikacích. Volba příkazu je závislá na chování programu. Pokud je nastaven správně, program spustí při každém tisku dialog aplikace `kprinter`, ve kterém můžete zvolit požadovanou frontu a další tis-

kové volby. Samozřejmě je nutné, aby nativní nastavení tisku aplikace s programem kprinter nekolidovalo a aby tiskové volby byly nastavované pouze přes kprinter.

## 11.6 Zvláštní vlastnosti v systému SUSE Linux

V SUSE Linuxu je v systému CUPS řada zajímavých vlastností. O těch nejdůležitějších se píše v následujícím textu:

### 11.6.1 CUPS server a firewall

Existuje několik možností, jak nastavit CUPS jako klienta síťového serveru.

- Ke každé frontě na síťovém serveru můžete nastavit lokální frontu, přes kterou lze přeposílat tiskové úlohy na správný server. Tento přístup nelze obecně doporučit, neboť v případě změny konfigurace na serveru je nutno přenastavit i všechny klienty.
- Tiskové úlohy je též možno přeposílat přímo na jeden síťový server. Při použití tohoto typu konfigurace nespouštějte lokálního démona CUPS. `lp` (a odpovídající knihovni volání dalších programů) umožňuje zasílat úlohy přímo na síťový server. Tuto konfiguraci však nelze použít, pokud chcete používat lokální tiskárnu.
- Démon CUPS může naslouchat oznamovacím IPP paketům vysílaným síťovými servery pro oznámení dostupných front. Je to nejlepší možná CUPS konfigurace pro tisk na vzdálených CUPS serverech. Existuje ovšem riziko, že útočník vyšle falešné IPP pakety a lokální démon pak zašle tisková data na podvrženou frontu. Při používání této konfigurace musí být port 631/UDP otevřen pro příchozí pakety.

YaST může použít dvě metody vyhledávání CUPS serverů. Může skenovat všechny počítače na síti a zjišťovat, zda nabízejí službu CUPS, nebo může naslouchat IPP paketům (metoda popsaná výše). Takto jsou také během instalace vyhledávány CUPS servery nabízející služby. Druhá metoda vyžaduje otevření portu 631/UDP pro příchozí pakety.

Výchozí nastavení firewallu zakazuje naslouchat IPP oznamovacím paketům na všech rozhraních. Proto nemůže fungovat druhá metoda vyhledávání vzdálených front ani

třetí metoda pro přístup ke vzdáleným frontám. Je tedy potřeba změnit nastavení firewallu. Je možné některé ze síťových rozhraní nastavit jako vnitřní (na kterém je port defaultně otevřen) nebo explicitně otevřít port na vnějším rozhraní. Z bezpečnostních důvodů není žádný z portů ve výchozím nastavení otevřen. Otevření portu pro konfiguraci vzdálených front druhou metodou může znamenat bezpečnostní riziko.

Nabídnuté nastavení firewallu je nutno změnit, aby mohl CUPS server během instalace detekovat vzdálené fronty. Jinou možností je oskenovat všechny lokální počítače nebo nakonfigurovat fronty ručně. Z důvodů zmíněných výše to však nedoporučujeme.

## 11.6.2 Administrátor webového frontendu CUPS

Pro administraci přes webový frontend (CUPS) nebo nástroj pro administraci tiskáren v KDE je nutné nastavit uživatele `root` jako CUPS administrátora, CUPS administrační skupinu `sys` a CUPS heslo. Učinit tak může uživatel `root` následujícím příkazem:

```
lppasswd -g sys -a root
```

Pokud toto nastavení neprovedete, nebude možná administrace přes webové rozhraní nebo administrační nástroj v KDE, protože autentizace bez nastavení CUPS administrátora selže. Jako CUPS administrátor může být nastaven i jakýkoliv jiný uživatel (viz [11.6.3 – „Změny v tiskové službě CUPS \(cupsd\)“](#) (strana 206)).

## 11.6.3 Změny v tiskové službě CUPS (cupsd)

Tyto změny byly poprvé provedeny v systému SUSE Linux 9.1.

### cupsd běží pod uživatelem lp

Při spuštění se program `cupsd` přepne z běhu pod uživatelem `root` na uživatele `lp`. Tím je dosaženo vyšší bezpečnosti, protože služba CUPS tak běží jen s potřebnými právy.

Nicméně autentizace (lépe řečeno kontrola hesla) nemůže být provedena přes `/etc/shadow`, protože uživatel `lp` k němu nemá přístup. Místo toho je použita autentizace specifická pro CUPS přes soubor `/etc/cups/passwd.md5`. Proto je do tohoto

souboru nutné vložit CUPS administrátora, CUPS administrační skupinu `sys` a heslo. Provést to může uživatel `root` následujícím příkazem:

```
lppasswd -g sys -a CUPS-administrátor
```

Pokud běží `cupsd` pod uživatelem `lp`, nemůže vygenerovat soubor `/etc/printcap`, neboť nemá právo zapisovat do adresáře `/etc/`. Místo toho `cupsd` vytvoří `/etc/cups/printcap`. Aby nebyla ohrožena funkce aplikací, které umí číst jména front pouze z `/etc/printcap`, je `/etc/printcap` symbolickým odkazem na `/etc/cups/printcap`.

Když `cupsd` běží pod uživatelem `lp`, nelze otevřít port 631. Proto nelze použít příkaz `rc cups reload`. Místo něj použijte `rc cups restart`.

## Obecná funkce `BrowseAllow` a `BrowseDeny`

Přístupová práva nastavená pro `BrowseAllow` a `BrowseDeny` platí pro všechny pakety zaslané na `cupsd`. Výchozí nastavení v souboru `/etc/cups/cupsd.conf` jsou následující:

```
BrowseAllow @LOCAL
BrowseDeny All
```

a

```
<Location />
  Order Deny,Allow
  Deny From All
  Allow From 127.0.0.1
  Allow From 127.0.0.2
  Allow From @LOCAL
</Location>
```

Při tomto nastavení mohou ke `cupsd` na CUPS serveru přistupovat pouze `LOCAL` počítače, tj. počítače, jejichž IP adresa náleží non-PPP rozhraní (přesněji rozhraní, jehož `IFF_POINTOPOINT` příznak není nastaven) a jejichž adresa náleží do stejné sítě jako CUPS server. Pakety z ostatních počítačů jsou okamžitě odmítnuty.

## `cupsd` je defaultně aktivní

Ve standardní instalaci je `cupsd` automaticky aktivní, což umožňuje pohodlný přístup ke CUPS frontám bez manuálního nastavování. Dvě předchozí vlastnosti (viz „[cupsd běží pod uživatelem lp](#)“ (strana 206) a „[Obecná funkce `BrowseAllow` a `BrowseDeny`](#)“

(strana 207)) jsou podmínkou k tomuto automatickému spuštění, neboť jinak by nebyla zajištěna dostatečná bezpečnost.

## 11.6.4 PPD soubory v různých balíčcích

V této části jsou popsány zdroje PPD souborů a jejich použití.

### Konfigurace tiskáren pouze pomocí PPD souborů

Modul pro konfiguraci tiskáren nástroje YaST nastavuje CUPS fronty pouze s využitím PPD souborů v `/usr/share/cups/model/`. Vhodný PPD soubor vybírá YaST porovnáním modelu tiskárny zjištěného během rozpoznávání hardwaru a modelů v PPD souborech v adresáři `/usr/share/cups/model/`. Za tímto účelem si YaST vytváří databázi modelů tiskáren získaných z PPD souborů. Když vyberete model ze seznamu výrobců a typů tiskáren, bude automaticky přiřazen vhodný PPD soubor.

Konfigurace s využitím pouze PPD souborů a žádných jiných informací má výhodu v tom, že je možné PPD soubory v adresáři `/usr/share/cups/model/` volně modifikovat. Modul YaST pro nastavení tiskáren si všímá všech změn a obnovuje svou databázi. Pokud například máte jen postscriptové tiskárny, nepotřebujete Foomatic PPD soubory z balíčku `cups-drivers` ani Gimp-Print PPD z balíčku `cups-drivers-stp`. Místo toho můžete prostě přkopírovat PPD soubory pro vaše postscriptové tiskárny přímo do adresáře `/usr/share/cups/model/` (pokud nejsou již součástí balíčku `manufacturer-PPDs`).

### PPD soubory v balíčku `cups`

Obecné PPD soubory v balíčku `cups` byly doplněny upravenými Foomatic PPD soubory pro tiskárny PostScript level 1 a 2:

- `/usr/share/cups/model/Postscript-level1.ppd.gz`
- `/usr/share/cups/model/Postscript-level2.ppd.gz`

### PPD soubory v balíčku `cups-drivers`

Normálně je pro nepostscriptové tiskárny používán Foomatic tiskový filtr `foomatic-rip` spolu s Ghostscriptem. Vhodné Foomatic PPD soubory s položkami



```
*NickName: ... Foomatic/Ghostscript driver a *cupsFilter:
... foomatic-rip jsou umístěny v balíčku cups-drivers.
```

YaST upřednostňuje Foomatic PPD soubory za následujících podmínek:

- Foomatic PPD soubor s položkou `*NickName: ... Foomatic ... (recommended)` odpovídá modelu tiskárny.
- Balíček `manufacturer-PPDs` neobsahuje vhodnější PPD soubor (viz níže).

## Gimp-Print PPD soubory v balíčku `cups-drivers-stp`

Místo `foomatic-rip` lze s mnoha nepostscriptovými tiskárnami použít CUPS filtr `rastertoprinter` z projektu Gimp-Print. Tento filtr a vhodné Gimp-Print PPD soubory jsou dostupné v balíčku `cups-drivers-stp`. Gimp-Print PPD soubory jsou umístěny v adresáři `/usr/share/cups/model/stp/` a mají položky `*NickName: ... CUPS+Gimp-Print a *cupsFilter: ... rastertoprinter`.

## PPD soubory od výrobců tiskáren v balíčku `manufacturer-PPDs`

Balíček `manufacturer-PPDs` obsahuje PPD soubory od výrobců tiskáren, pokud jsou uvolněny pod dostatečně volnou licenci. Postscriptové tiskárny by měly být nakonfigurovány s příslušným PPD souborem od výrobce, protože jsou tak dostupné všechny funkce tiskárny. YaST upřednostňuje PPD soubor z balíčku `manufacturer-PPDs` za následujících podmínek:

- Výrobce a model tiskárny zjištěný během detekce hardwaru odpovídá výrobcí a modelu tiskárny uvedeným v PPD souboru z balíčku `manufacturer-PPDs`.
- PPD soubor z balíčku `manufacturer-PPDs` je jediný vhodný PPD soubor pro danou tiskárnu nebo existuje Foomatic PPD soubor s položkou `*NickName: ... Foomatic/Postscript (recommended)`, který rovněž odpovídá dané tiskárně.

YaST nepoužije žádný soubor z balíčku `manufacturer-PPDs` v následujících případech:

- PPD soubor z balíčku `manufacturer-PPDs` neodpovídá výrobci a modelu tiskárny. To se může stát v případě, že balíček obsahuje jen jeden PPD soubor pro několik podobných tiskáren.
- Foomatic PostScript PPD soubor není *recommended* (doporučený). To může být v případě, kdy daná tiskárna nefunguje v postscriptovém režimu efektivně, například je v tomto režimu nespolehlivá pro nedostatek paměti či pomalá kvůli slabému procesoru. Dalším důvodem může být to, že tiskárna nepodporuje PostScript ve výchozí konfiguraci (je např. dostupný jako rozšiřující výbava).

Pokud je některý PPD soubor z balíčku `manufacturer-PPDs` pro postscriptovou tiskárnu vhodný, ale YaST ho nepoužije z výše zmíněných důvodů, zvolte vybraný model tiskárny v nástroji YaST ručně.

## 11.7 Řešení problémů

Následující odstavce se zabývají řešením nejčastějších hardwarových i softwarových problémů s tiskem.

### 11.7.1 Tiskárny bez podpory standardního tiskového jazyka

Tiskárny, které nepodporují žádný standardní tiskový jazyk, ale je s nimi možno komunikovat pouze pomocí speciálních kontrolních sekvencí, se nazývají *GDI tiskárny*. Takové tiskárny jsou funkční pouze s operačním systémem, ke kterému výrobce dodává ovladač. *GDI* je programovací rozhraní vyvinuté firmou Microsoft pro grafická zařízení. Problémem není programovací rozhraní jako takové, ale skutečnost, že pro komunikaci s *GDI* tiskárnami lze použít pouze proprietární jazyk specifický pro daný typ tiskárny.

Některé tiskárny lze používat v režimu *GDI* i v režimu standardního tiskového jazyka. Někteří výrobci dodávají ke *GDI* tiskárnám proprietární ovladače. Nevýhoda takových ovladačů ale spočívá v tom, že nemusí být vhodné pro všechny tiskové systémy či hardwarové platformy. Tiskárny podporující standardní tiskový jazyk jsou naopak na tiskovém systému či hardwarové platformě nezávislé.

Často může být výhodnější zakoupit podporovanou tiskárnu se standardním tiskovým jazykem, než trávit čas snahou zprovoznit proprietární linuxový ovladač. Problém s

ovladači se tak vyřeší jednou pro vždy a odstraní se nutnost instalovat a konfigurovat speciální ovládací software a shánět jeho nové verze v případě změn v tiskovém systému.

## 11.7.2 Pro postscriptovou tiskárnu není k dispozici vhodný PPD soubor

Pokud balíček `manufacturer-PPDs` neobsahuje pro vaši postscriptovou tiskárnu žádný vhodný PPD soubor, zkuste použít PPD soubor z CD s ovladači dodaného s tiskárnou nebo stáhněte soubor z webových stránek výrobce.

Pokud je PPD soubor k dispozici ve formě zip archívu (.zip) nebo samorozbalovacího zip archívu (.exe), rozbalte ho programem `unzip`. Přečtěte si licenční podmínky souboru a pomocí programu `cupstestppd` ověřte, zda odpovídá specifikaci *Adobe PostScript Printer Description File Format Specification, version 4.3*. Pokud program vrátí `FAIL`, jsou v PPD souboru závažné chyby, které mohou způsobit problémy. Proto by objevené chyby měly být odstraněny. Pokud je to nutné, požádejte výrobce tiskárny o vhodný PPD soubor.

## 11.7.3 Paralelní porty

Nejspolehlivější je připojit tiskárnu přímo k prvnímu paralelnímu portu a v BIOSu zvolit následující nastavení:

- I/O address: 378 (hexadecimal)
- Interrupt: irrelevant
- Mode: Normal, SPP nebo Output Only
- DMA: disabled

Pokud tiskárna na paralelním portu s tímto nastavením BIOSu nefunguje, explicitně vložte I/O adresu nastavenou v BIOSu do souboru `/etc/modprobe.conf` ve tvaru `0x378`. Pokud jsou paralelní porty dva a jejich I/O adresy jsou 378 a 278 (hexadecimálně), vložte je do souboru ve tvaru `0x378, 0x278`.

Pokud je volné přerušení 7, lze ho aktivovat zápisem nastavení uvedeným v příkladu 11.1 – „[/etc/modprobe.conf: Režim přerušení pro první paralelní port](#)“ (strana 212).

Před aktivací přerušeni zkontrolujte v souboru `/proc/interrupts`, jaká přerušeni se již používají. Jsou tam zobrazena jen právě používaná přerušeni, což závisí na právě aktivních hardwarových komponentách. Přerušeni pro paralelní port nesmí být používáno žádným jiným zařízením. Pokud si nejste jisti, použijte `irq=none`.

**Rovnice 11.1** */etc/modprobe.conf: Režim přerušeni pro první paralelní port*

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378 irq=7
```

## 11.7.4 Připojení síťových tiskáren

Identifikace síťových problémů

Připojte tiskárnu přímo k počítači. Nakonfigurujte ji pro účely testování jako lokální. Pokud funguje, problém je spojený se sítí.

Kontrola TCP/IP sítě

TCP/IP síť a převod jmen musí být funkční.

Kontrola vzdáleného `lpd`

Následujícím příkazem otestujte, zda je možné navázat TCP spojení s `lpd` (port 515) na vzdáleném počítači `host`:

```
netcat -z host; 515 && echo ok || echo selhalo
```

Pokud spojení s `lpd` nelze navázat, je možné, že `lpd` není aktivní, nebo, že jsou vážné problémy se sítí.

Jako uživatel `root` použijte následující příkaz k získání (možná velmi dlouhé) zprávy o stavu fronty `queue` na vzdáleném počítači `host`, za předpokladu, že je `lpd` aktivní a vzdálený počítač odpovídá na dotazy:

```
echo -e "\004queue" \  
| netcat -w 2 -p 722 host 515
```

Pokud `lpd` neodpovídá, může být neaktivní nebo může být problém se sítí. Pokud `lpd` odpoví, měla by odpověď ozřejmit, proč nelze na frontě `queue` na počítači `host` tisknout. Pokud dostanete odpověď jako v příkladu 11.2 – „Chybové hlášení programu `lpd`“ (strana 213), je problém způsobený vzdáleným `lpd`:

## Rovnice 11.2 Chybové hlášení programu lpd

```
lpd: your host does not have line printer access
lpd: queue does not exist
printer: spooling disabled
printer: printing disabled
```

### Kontrola vzdáleného cupsd

Ve výchozím nastavení by měl CUPS server oznamovat své fronty každých třicet sekund na UDP portu 631. Následující příkaz testuje, zda je na síti přítomný CUPS síťový server.

```
netcat -u -l -p 631 & PID=$! ; sleep 40 ; kill $PID
```

Pokud síťový CUPS server skutečně existuje, vrátí se za čtyřicet sekund zpráva zobrazená v příkladu 11.3 – „Oznámení síťového CUPS serveru“ (strana 213).

## Rovnice 11.3 Oznámení síťového CUPS serveru

```
ipp://pocitac.domena:631/printers/fronta
```

Následující příkaz lze použít k otestování možnosti navázání TCP spojení s cupsd (port 631) na vzdáleném počítači *host*:

```
netcat -z host 631 && echo ok || echo selhalo
```

Pokud nelze spojení navázat, je cupsd neaktivní nebo jsou závažné problémy se sítí. Příkaz `lpstat -h host -l -t` vrátí (možná velmi dlouhou) zprávu o stavu všech front na vzdáleném počítači *host*, pokud je cupsd aktivní a počítač odpovídá na dotazy.

Následující příkaz lze použít k otestování, zda fronta *queue* na počítači *host* přijme tiskovou úlohu sestávající z jednoho znaku carriage return (nový řádek). Vytiskeno by nemělo být nic, jen možná vysunut jeden prázdný list papíru.

```
echo -en "\r" \  
| lp -d queue -h host
```

### Řešení problémů se síťovou tiskárnou nebo zařízením *print server box*.

Při velkém množství tiskových úloh se občas objeví problémy se spoolery běžícími v zařízení *print server box*. Problém nelze řešit přímo, ale můžete spooler obejít adresováním tiskárny přímo přes TCP soket (viz 11.4.2 – „Síťové tiskárny“ (strana 201)).

Abyste mohli tuto metodu použít, musíte znát příslušný port na zařízení *print server box*. Když je tiskárna zapnuta a připojena k tomuto zařízení, lze TCP port určit

krátce po zapnutí zařízení pomocí programu `nmap`. Příkaz `nmap IP_adresa` může mít pro zařízení `print server box` následující výstup:

Port	State	Service
23/tcp	open	telnet
80/tcp	open	http
515/tcp	open	printer
631/tcp	open	cups
9100/tcp	open	jetdirect

Tento výstup značí, že tiskárnu připojenou k zařízení lze adresovat přes TCP socket na portu 9100. Ve výchozím nastavení kontroluje `nmap` jen běžně používané porty uvedené v `/usr/share/nmap/nmap-services`. Chcete-li zkontrolovat všechny možné porty, použijte příkaz `nmap -p od_portu-do_portu IP_adresa`. Může to ale trvat poměrně dlouho. Další informace naleznete v manuálové stránce `nmap`.

K otestování, zda lze tiskárnu na určitém portu adresovat, zašlete na příslušný port následujícím příkazem řetězce nebo soubory k tisku:

```
echo -en "\rAhoj\r\f" | netcat -w 1 IP_adresa port
cat soubor | netcat -w 1 IP_adresa port
```

## 11.7.5 Problém s tiskem bez chybového hlášení

Tiskový systém považuje úlohu za hotovou v okamžiku, kdy dokončí přenos dat příjemci (tiskárně). Pokud zpracování na tiskárně z nějakého důvodu selže (pokud například tiskárna nedokáže zpracovat data specifická pro určitou tiskárnu), tiskový systém se o tom nedozví. Pokud není tiskárna schopna vytisknout data specifická pro tiskárnu, použijte jiný, pro vaši tiskárnu vhodnější, PPD soubor.

## 11.7.6 Nepřístupné fronty

Pokud datový přenos k příjemci z nějakého důvodu i po několika pokusech selže, oznámí CUPS backend (např. `usb` nebo `socket`) tiskovému systému (přesněji `cupsd`) chybu. Backend rozhoduje o tom, kolik pokusů o přenos dat má smysl, a kdy prohlásí spojení za nemožné. Protože v takovém případě by další pokusy byly zbytečné, zablokuje `cupsd` na příslušné frontě tisk. Jakmile odstraníte zdroj problémů, musí systémový administrátor reaktivovat tisk na frontě příkazem `/usr/bin/enable`.

## 11.7.7 Rušení tiskových úloh

Pokud síťový CUPS server oznamuje fronty klientským počítačům přes prohlížení sítě a na klientovi je vhodně nastaven `cupsd`, přijímá od aplikací tiskové úlohy klientský `cupsd` a přeposílá je programu `cupsd` na serveru. Když `cupsd` tiskovou úlohu přijme, je jí přiřazeno nové číslo. Proto je číslo úlohy jiné na klientovi a jiné na serveru. Protože je tisková úloha obvykle přeposlána ihned, nelze ji zrušit pomocí čísla na klientovi. Klientský `cupsd` považuje tiskovou úlohu za dokončenou v okamžiku jejího přeposlání na server.

Chcete-li úlohu na serveru zrušit, použijte příkaz `lpstat -h tiskovyserver -o` ke zjištění čísla úlohy na serveru (za předpokladu, že server úlohu dosud nedokončil, tj. neposlal ji na tiskárnu). Pomocí takto získaného čísla můžete úlohu na serveru zrušit:

```
cancel -h tiskovyserver fronta-cisloulohy
```

## 11.7.8 Vadné tiskové úlohy a chyby v přenosu dat

Tiskové úlohy ve frontách zůstávají i když vypnete a zapnete tiskárnu nebo restartujete počítač během tisku. Vadné tiskové úlohy je nutno odstranit z fronty pomocí příkazu `cancel`.

Pokud je tisková úloha vadná nebo se objeví chyba v komunikaci mezi počítačem a tiskárnou, vytiskne tiskárna mnoho listů papíru s nečitelnými znaky, neboť není schopná data správně zpracovat. Vypořádat se s ní můžete následujícím způsobem:

1. Chcete-li tisk zastavit, vyjměte z inkoustových tiskáren papír nebo, u tiskáren laserových, otevřete zásobníky papíru. Kvalitní tiskárny mají pro zastavení tisku zvláštní tlačítko.
2. Tisková úloha může ve frontě přetrvávat, neboť úlohy jsou odstraňovány, až když jsou odeslány celé. Příkazem `lpstat -o` (nebo `lpstat -h tiskovy-server -o`) zjistíte, která fronta se právě tiskne. Úlohu pak odstraníte příkazem `cancel fronta-cislo-ulohy` (nebo `cancel -h tiskovy-server fronta-cislo-ulohy`).
3. Někdy je část dat tiskárně odesílána i v případě, že tisková úloha byla z fronty odstraněna. Ověřte si, zda pro frontu stále běží CUPS backend proces, a pokud

ano, ukončete ho. Například (v případě tiskárny na paralelním portu) lze použít příkaz `fuser -k /dev/lp0`, který ukončí všechny procesy přistupující k tiskárně (či přesněji k paralelnímu portu).

4. Tiskárnu resetujte jejím vypnutím. Po chvilce do ní vložte papír a zapněte ji.

## 11.7.9 Hledání problémů v tiskovém systému CUPS

Chcete-li identifikovat problém v tiskovém systému CUPS, použijte následující postup:

1. Nastavte `LogLevel debug` v souboru `/etc/cups/cupsd.conf`.
2. Zastavte `cupsd`.
3. Odstraňte `/var/log/cups/error_log*`, vyhněte se tak prohledávání příliš velkého protokolového souboru.
4. Spusťte `cupsd`.
5. Zopakujte činnost, která vedla k problému.
6. Zkontrolujte záznamy v souboru `/var/log/cups/error_log*`. Měly by vést k odhalení problému.

## 11.7.10 Další informace

Řešení mnoha specifických problémů najdete v Databázi podpory (<http://en.opensuse.org/SDB:SDB>) a v návodech na stánkách projektu openSUSE (<http://www.opensuse.org>).



# Dynamické uzly zařízení pomocí udev

# 12

Linuxové jádro 2.6 přineslo nové řešení v *uživatelském prostoru* umožňující používat v dynamickém adresáři `/dev` pro zařízení stálá a konzistentní označení: `udev`. Předchozí implementace `/dev` pomocí `devfs` již není podporována a byla nahrazena implementací založenou na `udev`.

Tradičně byly v linuxových systémech v adresáři `/dev` umístěny uzly (device nodes) pro všechny možné typy zařízení, bez ohledu na jejich skutečnou existenci. Adresář proto zabíral velké množství místa. Příkaz `devfs` přinesl významné zlepšení, neboť díky němu mají v adresáři `/dev` své uzly pouze ta zařízení, která skutečně existují.

Nový způsob vytváření uzlů přinesl příkaz `udev`. Ten porovná informace dostupné ze systému souborů `sysfs` s daty zadanými uživatelem ve formě pravidel. `sysfs` je nový souborový systém dostupný v jádře 2.6. Poskytuje základní informace o zařízeních připojených k systému. Souborový systém `sysfs` je připojený jako `/sys`.

Pravidla není nutno vytvářet. Pokud je k systému připojeno zařízení, je vytvořen příslušný uzel, Pravidla ovšem umožňují změnit jména uzlů. Lze tak nahradit nesrozumitelná jména jmény snadno zapamatovatelnými a dosáhnout konzistentních jmen zařízení, když je připojeno více zařízení stejného typu.

Dvě připojené tiskárny budou například, není-li určeno jinak, označeny jako `/dev/lp0` a `/dev/lp1`. Které tiskárně bude přiřazen který uzel závisí na pořadí, v jakém jsou zapnuty. Jiným příkladem jsou externí zařízení pro ukládání dat, jako USB disky. Příkaz `udev` umožňuje přesně zvolit cesty vkládané do `/etc/fstab`.

## 12.1 Tvorba pravidel

Pravidla načítá udev ze souboru `/etc/udev/udev.rules` ještě předtím, než vytvoří uzly v adresáři `/dev`. Pokud odpovídá více pravidel, použije první z nich. Komentáře jsou v souboru uvozeny znakem hash (`#`). Pravidla jsou zapisována v následujícím formátu:

```
klíč , [klíč , ...] NAME [, SYMLINK]
```

Každé pravidlo musí obsahovat alespoň jeden klíč, neboť pravidla jsou zařízením přiřazována právě pomocí těchto klíčů. Rovněž je nezbytné určit jméno (parametr `name`). To je totiž přiřazeno uzlu zařízení vytvořenému v adresáři `/dev`. Volitelný parametr `symlink` umožňuje vytvoření uzlů i na dalších místech. Pravidlo pro tiskárnu může vypadat například takto:

```
BUS="usb", SYSFS{serial}="12345", NAME="lp_hp", SYMLINK="printers/hp"
```

V příkladu jsou použity dva klíče — `BUS` a `SYSFS{serial}`. Tyto klíče říkají udev, aby porovnal sériové číslo obsažené v klíči se sériovým číslem zařízení připojeného na USB sběrnici. Pokud oba klíče souhlasí, přiřadí zařízení jméno `lp_hp` v adresáři `/dev`. Navíc na něj vytvoří symbolický odkaz `/dev/printers/hp`. Adresář `printers` se vytvoří automaticky. Tiskové úlohy bude možno posílat jak na `/dev/printers/hp`, tak i na `/dev/lp_hp`.

## 12.2 Automatizace pomocí NAME a SYMLINK

Parametry `NAME` a `SYMLINK` umožňují využití operátorů pro automatické přiřazení hodnot, které odkazují na informace jádra o příslušném zařízení. Následující jednoduchý příklad objasňuje princip:

```
BUS="usb", SYSFS{vendor}="abc", SYSFS{model}="xyz", NAME="kamera%n"
```

Operátor `%n` v parametru `name` bude nahrazen číslem zařízení `kamera`, např. `kamera0` nebo `kamera1`. Další užitečný operátor je `%k`, který je nahrazován standardním jménem zařízení v jádře, např. `hda1`. Seznam všech operátorů je k dispozici v manuálové stránce `udev`.

## 12.3 Regulární výrazy v klíčích

V interpretu příkazů lze používat regulární výrazy a zástupné znaky. Např. znak `*` lze použít místo libovolných znaků a znak `?` lze použít místo právě jednoho libovolného znaku.

```
KERNEL="ts*", NAME="input/%k"
```

Toto pravidlo přiřazuje standardní jméno ve standardním adresáři zařízení jehož označení začíná písmeny "ts". Podrobné informace o použití regulárních výrazů viz manuálová stránka `udev`.

## 12.4 Výběr klíčů

Důležité je pro každé `udev` pravidlo vybrat dobrý klíč. Následují příklady běžně používaných klíčů:

**BUS**

typ sběrnice

**KERNEL**

jméno zařízení používané jádrem

**ID**

číslo zařízení na sběrnici (např. ID na sběrnici PCI)

**PLACE**

fyzické místo připojení zařízení (např. USB)

Klíče `ID` a `PLACE` jsou užitečné, obvykle se ale používají klíče `BUS`, `KERNEL`, a `SYSFS { . . . }`. Konfigurace `udev` umožňuje použít i klíče volající externí skripty a vyhodnocující jejich výsledky. Další informace lze získat pomocí příkazu `man udev`.

Souborový systém `sysfs` obsahuje v adresářovém stromu malé soubory s informacemi o hardwaru. Každý soubor obvykle obsahuje jednu informační položku, jako je jméno zařízení, výrobce nebo sériové číslo. Každý z těchto souborů může být použit jako hodnota klíče. V jednom pravidlu však mohou být použity jako klíče pouze soubory nacházející se ve stejném adresáři.

Přítom může být užitečný příkaz `udevinfo`. Je potřeba nalézt podadresář `/sys`, který odpovídá příslušnému zařízení a obsahuje soubor `dev`. Ty se všechny nacházejí v adresáři `/sys/block` nebo `/sys/class`. Pokud pro zařízení již uzel existuje, může vám `udevinfo` ušetřit kus práce. Příkaz `udevinfo -q path -n /dev/sda` vypíše `/block/sda`. To znamená, že hledaný adresář je `/sys/block/sda`. Nyní zavolejte `udevinfo` příkazem `udevinfo -a -p /sys/block/sda`. Oba příkazy lze rovněž sloučit v jeden: `udevinfo -a -p `udevinfo -q path -n /dev/sda``. Toto je část výstupu:

```
BUS="scsi"  
ID="0:0:0:0"  
SYSFS{detach_state}="0"  
SYSFS{type}="0"  
SYSFS{max_sectors}="240"  
SYSFS{device_blocked}="0"  
SYSFS{queue_depth}="1"  
SYSFS{scsi_level}="3"  
SYSFS{vendor}=" "  
SYSFS{model}="USB 2.0M DSC      "  
SYSFS{rev}="1.00"  
SYSFS{online}="1"
```

Z výstupu příkazu si vyberte vhodné klíče, které se nebudou měnit. Pamatujte, že nelze použít klíče z různých adresářů.

## 12.5 Konzistentní pojmenování zařízení pro hromadné uchovávání dat

SUSE Linux obsahuje skripty, které pomáhají přiřadit pevným diskům a dalším úložným zařízením vždy stejná jména, `/sbin/udev.get_persistent_device_name .sh` je obalovací skript (wrapper). Nejprve zavolá `/sbin/udev.get_unique_hardware_path.sh`, který zjistí hardwarovou cestu k příslušnému zařízení. Skript `/sbin/udev.get_unique_drive_id.sh` zjistí sériové číslo. Oba výstupy jsou následně předány `udev`, který vytvoří symbolický odkaz na uzel zařízení v adresáři `/dev`. Obalovací skript lze rovněž přímo použít v `udev` pravidlech. Následující příklad pro SCSI může být zobecněn i pro USB nebo IDE (musí být zapsán na jedné řádce):

```
BUS="scsi",  
PROGRAM="/sbin/udev.get_persistent_device_name.sh",  
NAME="%k", SYMLINK="%c{1+}"
```

Jakmile je nahrán ovladač pro zařízení pro hromadné uchovávání dat, zaregistruje všechny dostupné pevné disky v jádře. Každý z nich spustí blokovou hotplug událost, která volá `udev`. `udev` nejdříve načte pravidla aby zjistil, zda je potřeba vytvořit symbolický odkaz.

Pokud je ovladač nahrán prostřednictvím `initrd`, hotplug události se ztratí. Nicméně jsou všechny informace uloženy v souborovém systému `sysfs`. Utilita `udevstart` vyhledá všechny zařízení v `/sys/block` a `/sys/class` a spustí `udev`.

Existuje také startovací skript `boot.udev`, který během startu systému znovu vytvoří všechny uzly zařízení. Tento startovací skript musí být aktivován pomocí editoru úrovní běhu YaST nebo příkazem `insserv boot.udev`.

---

### Tip

Mnoho programů a nástrojů spoléhá na skutečnost, že `/dev/sda` je SCSI pevný disk a `/dev/hda` je IDE pevný disk. Pokud tomu tak není, přestanou fungovat. YaST je na těchto nástrojích závislý, takže pracuje pouze jaderným označením zařízení.

---



# Souborové systémy

Linux podporuje řadu různých souborových systémů. V této kapitole najdete krátký přehled těch nejpobulárnějších včetně jejich popisu, výhod a příkladů vhodného nasazení. Zároveň se zde dočtete o podpoře LFS (*Large File Suppnebot*) v Linuxu.

## 13.1 Termíny

### metadata

Interní datová struktura souborového systému, která zajišťuje okamžité organizování a přístupnost dat na disku. Lze je nazvat také daty o datech. Prakticky všechny souborové systémy metadata používají a jejich struktura bývá jedním z důvodů odlišných výkonů.

### inod

Inody obsahují různé informace o souboru, včetně velikosti, počtu odkazů, data a času vytvoření, změny a posledního přístupu, stejně jako ukazatele na diskové bloky, kde je soubor skutečně uložen.

### žurnál

Žurnál je struktura na disku obsahující záznam o změnách metadat souborového systému. Žurnalování má významnou zásluhu na obnově souborového systému v případě poškození a kontrole konzistence při startu. Při kontrole jsou obnovovány pouze žurnály.

## 13.2 Hlavní souborové systémy Linuxu

Před několika lety byla volba souborového systému v Linuxu otázkou několika vteřin (buď Ext2 nebo ReiserFS). Jádra řad 2.4 a 2.6 nabízejí však mnohem víc.

Při volbě souborového systému je především v situacích, kdy je požadován maximální výkon, nutné uvážit, jaké aplikace hodláte používat. Každý souborový systém má své výhody i nevýhody, které je nutné přitom brát v úvahu. Ani ten nejlepší souborový systém však nedokáže nahradit rozumné zálohování.

Termíny integrita dat nebo konzistence dat používané v této kapitole, nemají nic společného s konzistencí uživatelských dat (dat zapisovaných aplikacemi do souborů). Zda jsou data pro aplikace konzistentní, si kontrolují přímo aplikace.

---

### **Důležité: Nastavení souborového systému**

Všechna zde uvedená nastavení lze snadno provést pomocí programu YaST.

---

### 13.2.1 Ext2

Historie Ext2 sahá až do počátečních dnů Linuxu. Jeho předchůdce Extended souborový systém byl implementován v dubnu roku 1992 v Linuxu 0.96c. Od té doby prošel Extended souborový systém celou řadou změn až k Ext2, nejoblíbenějšímu linuxovému souborovému systému. Z trůnu ho sesadil až příchod žurnálovacích souborů.

Ext2 neumožňuje dynamickou alokaci inodů. Znamená to, že datové bloky, do jsou data ukládána, jsou vždy stejně velké. Tato skutečnost může vést k nevhodnému využívání diskového prostoru.

Základní přehled vlastností Ext2 vám pomůže porozumět tomu, proč byl tento souborový systém (a v některých oblastech stále ještě je) nejoblíbenějším linuxovým souborovým systémem.

#### Spolehlivost

Od počátků svého vzniku Ext2 prošel celou řadou testů a zlepšení. To může být důvod, proč se jeví tak spolehlivým. Pokud systém není možné korektně odpojit, spustí se e2fsck, který začne kontrolovat data souborového systému. Metadata jsou spojována do konzistentního stavu a chybná nebo poškozená data nebo bloky dat



jsou zapisována do příslušného souboru (nazývaného `lost+found`). Na rozdíl od žurnálovacích souborových systémů `e2fsck` nekontrolujete jen pozměněná data, ale celý systém. To u dnešních disků samozřejmě zabere mnoho času. Protože však není nutné spravovat žurnály a používá mnohem méně paměti, je v některých případech rychlejší než ostatní souborové systémy.

#### Jednoduchý upgrade

Souborový systém Ext2 tvoří z velké části podklad pro souborový systém další generace Ext3. Jeho spolehlivost byla elegantně zkombinována s výhodami žurnálování.

## 13.2.2 Ext3

Ext3 navrhl Stephen Tweedie. Na rozdíl od všech ostatních novějších souborových systémů není Ext3 založen na zcela nových základech. Jeho vývoj byl založen na Ext2. Tyto dva souborové systémy tak k sobě mají velmi blízko. Není proto problém vystavět Ext3 na již existujícím systému Ext2. Největší rozdíl, který tyto dva systémy odlišuje, je především podpora žurnálování v Ext3.

Ext3 nabízí tyto nejvýznamnější výhody:

#### Jednoduchý upgrade z Ext2

Ext3 je založen na kódu Ext2 a sdílí s ním formát dat na disku. Z toho důvodu je přechod z Ext2 na Ext3 velmi jednoduchý. Obnova při poškození a kontrola tohoto systému je extrémně rychlá a bezpečná. Pokud z nějakého důvodu Ext3 nevyhovuje vašim požadavkům, není problém vrátit se zpět k Ext2. Downgrade je stejně jednoduchý jako upgrade. Stačí čistě odpojit souborový systém Ext3 a pak ho připojit jako Ext2.

#### Spolehlivost a výkon

Naprostá většina žurnálovacích souborů je metadata-only. To znamená, že metadata jsou vždy udržována v konzistentním stavu, což ale není vždy garancí konzistentnosti samotných dat souborového systému. Ext3 je navržen tak, aby se staral jak o metadata tak o samotná data. Stupeň této péče lze nastavit. Povolení Ext3 v režimu *data=journal* poskytuje maximální bezpečnost (integritu dat), ale žurnálování dat i metadat může vést k výraznému zpomalení systému. Jednou z novějších záležitostí je režim *data=ordered*, který zajišťuje integritu dat i metadat, ale žurnálování provádí pouze u metadat. Ovladač souborového systému sbírá všechny bloky dat, které náleží k určitému updatu metadat. Tyto bloky jsou seskupovány do transakcí

a ty jsou pak před updatem metadat zapsány na disk. Výsledkem je zajištění konzistence dat i metadat bez viditelného zvýšení zatížení systému. Třetí volbou je režim `data=writeback`, který umožňuje zapsat data po zapsání metadat do žurnálu. Tato volba vykazuje nejlepší hodnoty při měření výkonu. Zároveň dokáže zajistit obnovu dat při narušení integrity souborového systému. Pokud pro Ext3 nenastavíte žádný režim, použije se `data=ordered`.

Přechod z Ext2 na Ext3 na již existujícím systému se skládá ze dvou kroků:

### Žurnály

Přihlaste se jako `root` a zadejte příkaz `tune2fs -j`. Tak vytvoříte žurnál Ext3 s výchozími parametry. Pokud chcete nastavit délku žurnálu, zadejte místo předšlého příkazu příkaz `tune2fs -J` spolu s volbami `size=` a `device=`. Více informací o programu `tune2fs` najdete v jeho manuálové stránce (*man 8 tune2fs*).

### Nastavení typu souborového systému v `/etc/fstab`

Aby byl Ext3 správně rozpoznáván, je nutné ho uvést v souboru `/etc/fstab`. U položky diskového oddílu, u které jsme souborový systém změnili, musíte změnit typ souborového systému z `ext2` na `ext3`. Změna se projeví po restartu počítače.

## 13.2.3 ReiserFS

Ten souborový systém byl jednou z hlavních novinek jádra 2.4. Pro SUSE jádra předcházející řady 2.2.x byl dostupný jako jaderný patch od verze 6.4. ReiserFS vznikl díky Hansi Reiserovi a týmu vývojářů společnosti Namesys jako alternativa ke starému Ext2. Může se pochlubit lepším využitím disku, rychlejším přístupem a mnohem lepší a rychlejší opravou dat. Zaměřuje se však na péči o metadata, ale ne o samotná data. Následující verze vy již měly obsahovat také datové žurnalování (do žurnálu jsou zapisovány informace o metadatach i aktuálních datech).

Výhody souborového systému ReiserFS:

### Lepší využití disku

V ReiserFS jsou všechna data organizována ve strukturách nazývaných B\* stromy. Stromová struktura umožňuje lepší využití disku, protože malé soubory lze umístit přímo do listu stromu, místo rozmístění po celém disku a spravovat pak ukazatele na umístění dat. Data navíc nejsou umísťována do bloků s pevnou velikostí (obvykle 1 nebo 4 kB), ale do bloků potřebné velikosti. Další výhoda ReiserFS spočívá v

dynamickém alokování inodů. To umožňuje oproti starším systémům vyšší flexibilitu.

#### Vyšší diskový výkon

U malých souborů najdete informace o datech souboru a `stat_data` (inode) vedle sebe. Lze je přečíst jednou jednoduchou diskovou IO operací, což znamená, že je potřeba pouze jeden přístup na disk.

#### Rychlá obnova po poškození

V případě havárie počítače a poškození souborového systému lze souborový systém ve většině případů opravit během několika sekund. Žurnálování také urychluje pravidelné kontroly konzistence souborového systému.

#### Žurnálování

ReiserFS podporuje také žurnálování podobné tomu popsanému v části věnované Ext3 section, 13.2.2 – „Ext3“ (strana 225). Výchozí režim je `data=ordered`. tento režim zajišťuje jak integritu metadat, tak samotných dat, ale žurnálování používá pouze u metadat.

## 13.2.4 Reiser4

Krátce po vydání jádro 2.6 se rodina žurnálovacích souborových systémů rozšířilo o nového člena: Reiser4. Reiser4 je od svého předchůdce (version 3.6) zcela odlišný. Představuje koncept modulů vylepšujících funkčnost souborového systému a vylepšenou bezpečnost.

#### Bezpečnostní koncept

Při návrhu souborového systému Reiser4 věnovali vývojáři zvláštní pozornost funkcím spjatým s bezpečností. Reiser4 je proto vybaven řadou bezpečnostních modulů. Jedním z nejvýznamnějších jsou „položky“ souboru. V současnosti jsou ACLs definovány pro každý soubor. V systému s velkým počtem souborů každý soubor obsahuje potřebné informace o právech každého uživatele, skupiny či aplikace. V systému Reiser4 jsou tyto soubory rozděleny do menších jednotek („položky“). Přístupová práva mohou být pro každou položku a uživatele nastavena zvlášť, čímž je umožněna mnohem lepší správa přístupu. Jako příklad může posloužit soubor `/etc/passwd`. Práva zápisu do tohoto souboru má pouze uživatel `root`, ostatní uživatelé mají jen práva pro čtení. S využitím položek souborového systému Reiser4 můžete rozdělit soubor na řadu položek (jednu pro každého uživa-

tele), takže uživatel může editovat vlastní data, ale nesmí měnit data ostatních uživatelů. Tento koncept sebou přináší jak vyšší bezpečnost, tak také pružnost.

Rozšiřitelnost prostřednictvím modulů

V systému Reiser4 je řada běžných i rozšířených funkcí prováděna moduly. Lze je snadno přidávat, takže není nutné kvůli nové funkci kompilovat nové jádro nebo formátovat disk.

Lepší výkon souborového systému díky delayed alokaci

Stejně jako u XFS podporuje Reiser4 delayed alokace viz [13.2.6 – „XFS“](#) (strana 228).

## 13.2.5 JFS

*JFS Journaling file system* byl navržen společností IBM. První testovací verze JFS se v linuxové komunitě objevila na jaře roku 2000. Verze 1.0.0 vyšla roku 2001. JFS byl navržen pro výkonné servery a proto byl velký důraz kladen na jeho výkonnost. Jako plně 64 bitový souborový systém, JFS podporuje větší velikost souborů i oddílů.

Vlastnosti JFS:

Výkonné žurnálování

JFS klade stejně jako ReiserFS důraz pouze na metadata. Stejně jako ReiserFS při opravě kontroluje pouze změny v metadatach, což vede k vysoké úspoře času. Konkurenční operace vyžadují současně záznam lze spojit do jedné skupiny a tak vícenásobnými operacemi zápisu redukovat ztráty výkonu.

Vynikající organizace adresářů

JFS používá dva typy organizace adresářů. Pro malé adresáře umožňuje ukládání obsahu přímo v inodu. U větších adresářů používá B<sup>+</sup> stromy.

Lepší využití prostoru díky dynamické alokaci inodů

JFS šetří váš čas --- inody jsou alokovány automaticky.

## 13.2.6 XFS

Původně společnost SGI spustila vývoj tohoto systému na začátku roku 1990 pro svůj operační systém IRIX OS. XFS měl být výkonným 64-bitovým žurnálovacím souborovým systémem určeným pro ty nejnáročnější výpočetní úlohy. XFS dosahuje vynikajících

výsledků při práci s velkými soubory a špičkovým hardwarem. Stejně jako jiné žurnálovací systémy jako např. ReiserFS však kontroluje pouze integritu metadat.

Rychlý pohled na hlavní vlastnosti XFS ukáže, proč je tak dobrým souborovým systémem pro náročné výpočetní úlohy:

**Vysoká stabilita díky využití alokačních skupin**

Při vytvoření souborového systému XFS je souborový systém rozdělen do osmi nebo více lineárních částí stejné velikosti. Ty jsou označovány jako alokační skupiny. Na alokační skupiny lze pohlížet jako na souborový systém v souborovém systému. Jednotlivé alokační skupiny na sobě nejsou nijak závislé, takže jádro může současně adresovat několik skupin najednou. Tato funkce pak vede k vysokému výkonu souborového systému XFS.

**Vysoký výkon podpořený účinnou správou diskového prostoru**

Volný prostor a inody jsou spravovány  $B^+$  stromy vně alokačních skupin. Využívání  $B^+$  stromů zvyšuje výkon. S XFS je spojena funkce delayed alokace. XFS při alokaci dělí proces na dvě části. Transakce jsou uloženy v RAM a je pro ně rezervována předpokládaná velikost prostoru. XFS nerozhoduje, kde přesně budou data uložena (bloky souborového systému). Toto rozhodnutí je odloženo na poslední možnou chvíli. Některá data se tak vůbec nedostanou na disk, protože dřív než XFS rozhodne o jejich uložení, zastarají. Tímto způsobem je zvyšován výkon při zápisu a redukována fragmentace souborového systému. Vzhledem ke strategii delayed alokace je však XFS mnohem náchylnější ke ztrátám dat při pádu systému než jiné souborové systémy.

**Prelokace souborového systému jako prevence fragmentace**

Před zápisem dat do souborového systému, XFS rezervuje (prelokuje) volný prostor potřebný pro soubor. Tak je maximálně redukována fragmentace souborového systému. Zároveň dojde ke zvýšení výkonu, protože jednotlivé soubory nejsou rozmístěny po celém souborovém systému.

## 13.3 Některé další podporované souborové systémy

Následující tabulka shrnuje některé další souborové systémy podporované Linuxem. Jedná se především o takové souborové systémy, které jsou podporovány z důvodů kompatibility s jinými systémy nebo typy médií.

**Tabulka 13.1** Typy souborových systémů v Linuxu

---

cramfs	<i>Komprimovaný souborový systém ROM souborový systém: systém pouze ke čtení.</i>
hpfs	<i>High Performance souborový systém: IBM OS/2 standard souborový systém --- systém pouze ke čtení.</i>
iso9660	Standardní souborový systém na CD.
minix	První linuxový souborový systém používaný v Linuxu. Dnes se používá prakticky pouze pro diskety s ovladači.
msdos	<i>fat</i> , souborový systém používaný systémem DOS. Dnes je používán řadou dalších operačních systémů.
ncpfs	souborový systém pro připojení svazků Novellu přes síť.
nfs	<i>Síťový souborový systém: Síťový souborový systém umožňuje uložení dat na jednom počítači, na který pak mohou přes síť přistupovat uživatelé z jiných počítačů.</i>
smbfs	<i>Server Message Block: síťový souborový systém umožňující přístup po síti používaný systémy Windows.</i>
sysv	Používané systémy SCO UNIX, Xenix a komerční unixové systémy pro PC.
ufs	Používané systémy BSD, SunOS a NeXTstep. Podporuje pouze režim <i>read-only</i> .
umsdos	<i>UNIX na MSDOS: aplikovaný na normálním fat souborovém systému. Unixové funkčnosti (přístupová práva, odkazy, dlouhá jména souborů) dosahuje vytvářením zvláštních souborů.</i>
vfat	Virtual FAT: rozšíření souborového systému fat (podporuje dlouhá jména souborů).
ntfs	<i>Windows NT souborový systém, pouze ke čtení.</i>

---

## 13.4 Podpora souborů větších než 2 GB

Původně podporovaná maximální velikost linuxového souboru je 2 GB. Před příchodem multimediálních souborů a rozsáhlých databází se tato velikost zdála dostatečná. Především velmo rychlý rozmach digitálního zpracování médií sebou přinesl nutnost poupravit jádro a knihovnu C tak, aby bylo možné pracovat také se soubory většími než 2 GB. V současné době již LFS podporují prakticky všechny novější souborové systémy.

Následující tabulka poskytuje přehled současných omezení velikostí linuxových souborů a souborových systémů v jádrech řady 2.4.

**Tabulka 13.2** Maximální velikost souborových systémů

Souborový systém	Velikost souboru [Byte]	Velikost souborového systému [Byte]
Ext2 or Ext3 (velikost bloku 1 kB)	$2^{34}$ (16 GB)	$2^{41}$ (2 TB)
Ext2 or Ext3 (velikost bloku 2 kB)	$2^{38}$ (256 GB)	$2^{43}$ (8 TB)
Ext2 or Ext3 (velikost bloku 4 kB)	$2^{41}$ (2 TB)	$2^{44}$ (16 TB)
Ext2 or Ext3 (velikost bloku 8 kB) (systémy s 8 kB stránkami jako Alpha)	$2^{46}$ (64 TB)	$2^{45}$ (32 TB)
ReiserFS v3	$2^{46}$ (64 GB)	$2^{45}$ (32 TB)
XFS	$2^{63}$ (8 EB)	$2^{63}$ (8 EB)
JFS (velikost bloku 512 bytů)	$2^{63}$ (8 EB)	$2^{49}$ (512 TB)
JFS (velikost bloku 4 kB)	$2^{63}$ (8 EB)	$2^{52}$ (4 PB)
NFSv2 (na straně klienta)	$2^{31}$ (2 GB)	$2^{63}$ (8 EB)

---

Souborový systém	Velikost souboru [Byte]	Velikost souborového systému [Byte]
NFSv3 (na straně klienta)	$2^{63}$ (8 EB)	$2^{63}$ (8 EB)

---

### Důležité: Omezení linuxového jádra

Existují také omezení jádra:

V tabulce 13.2 – „Maximální velikost souborových systémů“ (strana 231) najdete omezení velikosti disku. Jádro 2.6 má následující vlastní omezení velikosti souborového systému a souborů:

Velikost souboru

Na 32 bitových systémech nemohou být soubory větší než 2 TB (241 bytů).

Velikost souborového systému

Souborové systémy mohou být veliké 2 na 73 73 bytů. Tohoto limitu v současné době ani reálně nelze kvůli omezením hardwaru dosáhnout.

---

## 13.5 Další informace

Každý z uvedených souborových systémů je spravován vlastním projektem, který má vlastní internetové stránky obsahující veškerou dostupnou dokumentaci a také emailovou konferenci.

- <http://e2fsprogs.sourceforge.net/>
- <http://www.zipworld.com.au/~akpm/linux/ext3/>
- <http://www.namesys.com/>
- <http://oss.software.ibm.com/developerworks/opensource/jfs/>
- <http://oss.sgi.com/projects/xfs/>



Srovnávací tutoriál linuxových souborových systémů najdete na stránkách *IBM developerWorks*:

<http://www-106.ibm.com/developerworks/library/l-fs.html>

Srovnání linuxových žurnálovacích souborových systémů najdete v článku od Juan I. Santos Florido uveřejněného v *Linuxgazette*:

<http://www.linuxgazette.com/issue55/flneboido.html>.

Pokud byste rádi získali další informace o LFS v Linuxu, doporučujeme vám stránky Andrease Jaegera: [http://www.suse.de/~aj/linux\\_lfs.html](http://www.suse.de/~aj/linux_lfs.html).



# Systém X Window

Systém X Window (X11 nebo X server) se stal prakticky standardem grafického uživatelského rozhraní v unixových systémech. Je to síťový systém, který umožňuje, aby se programy spuštěné na jednom počítači zobrazovaly na jiném počítači připojeném jakoukoli síťovou technologií, ať už v LAN nebo Internetu.

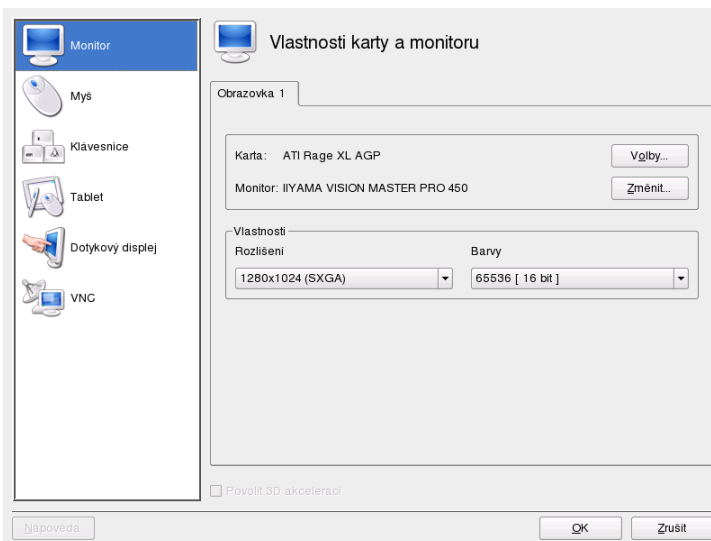
V této kapitole se pojednává o optimalizaci prostředí vašeho systému X Window, základech práce s fonty v systému SUSE Linux a o konfiguraci OpenGL a 3D. Konfigurace myši a klávesnice pomocí modulů YaST je popsána v *Uživatelské příručce*.

## 14.1 Nastavení X11 pomocí SaX2

X server se stará o komunikaci mezi hardwarem a softwarem. Pracovní prostředí (KDE nebo GNOME) a mnoho správců oken používá X server pro interakci s uživatelem.

Grafické prostředí se nastavuje během instalace. Chcete-li později změnit nastavení, spusťte SaX2. Aktuální nastavení je uloženo a můžete se k němu kdykoliv vrátit. Při konfiguraci se použijí jako výchozí aktuální hodnoty, které můžete změnit: rozlišení obrazovky, barevná hloubka, obnovovací frekvence a výrobce a typ monitoru, pokud byl rozpoznán.

**Obrázek 14.1** Hlavní okno SaX2



V levé liště je šest položek, každá z nich obsahuje konfigurační dialog z YaSTu. Jejich popis najdete v kapitole „*Konfigurace pomocí YaST*“ (↑Uživatelská příručka).

#### Monitor

Popis nastavení monitoru a grafické karty najdete v části „Vlastnosti karty a monitoru“ (2 – „*Konfigurace pomocí YaST*“, ↑Uživatelská příručka).

#### Myš

Popis nastavení myši v grafickém prostředí najdete v části „Vlastnosti myši“ (2 – „*Konfigurace pomocí YaST*“, ↑Uživatelská příručka).

#### Klávesnice

Popis nastavení klávesnice v grafickém prostředí najdete v části „Vlastnosti klávesnice“ (2 – „*Konfigurace pomocí YaST*“, ↑Uživatelská příručka).

#### Tablet

Popis nastavení grafického tabletu najdete v části „Vlastnosti tabletu“ (2 – „*Konfigurace pomocí YaST*“, ↑Uživatelská příručka).

Dotykový displej

Popis nastavení dotykového displeje najdete v části „Dotykový displej“ (2 – „*Konfigurace pomocí YaST*“, ↑Uživatelská příručka).

VNC

Popis nastavení VNC najdete v části „Vlastnosti vzdáleného přístupu“ (2 – „*Konfigurace pomocí YaST*“, ↑Uživatelská příručka).

## 14.2 Optimalizace systému X Window

X.Org je open source implementace X Window systému. Je vyvíjena "X.Org Foundation", která je také odpovědná za vývoj nových technologií a standardů X Window systému.

Abyste maximálně využili možností svého hardwaru (myš, grafická karta, monitor, klávesnice), můžete nastavení ručně optimalizovat. Podrobnější informace o nastavení X Window systému najdete v souborech v adresáři `/usr/share/doc/packages/Xorg` a manuálových stránkách, ke kterým můžete přistupovat například příkazem `man xorg.conf`.

Program SaX2 umožňuje i náročné zásahy do konfigurace X Window, nicméně abyste naplno využili schopnosti vašeho hardwaru jako jsou myš, grafická karta, monitor nebo klávesnice, může být nutná ruční editace konfiguračního souboru. Některé aspekty tohoto procesu budou vysvětleny v následujícím textu. Podrobnější informace o konfiguraci systému X Window získáte v manuálových stránkách - viz příkaz `man xorg.conf`, k užítku vám mohou být i soubory v adresáři `/usr/share/doc/packages/xf86`.

---

### Varování

Při konfiguraci systému X Window buďte opatrní. Nikdy X Window nespouštějte před dokončením jeho řádné konfigurace, protože chybná konfigurace může způsobit neodstranitelné škody na vašem hardwaru (to se vztahuje zejména na monitory s pevnou frekvencí, které se však dnes už téměř nepoužívají). Autoři této knihy a společnost SUSE Linux AG není za takovéto škody odpovědná. Následující informace byly pečlivě ověřovány, to ovšem nezaručuje, že všechny zde popsané postupy jsou správné a nemohou poškodit váš hardware.

---

V následujících odstavcích je popsána struktura konfiguračního souboru `/etc/X11/xorg.conf`. Tento soubor je členěn na sekce uvedené klíčovým slovem `Section` `<designation>` a ukončené klíčovým slovem `EndSection`. Níže naleznete stručný přehled nejdůležitějších sekcí.

Ve výchozím nastavení vytváří programy `SaX2` a `xf86config` konfigurační soubor `xorg.conf` v adresáři `/etc/X11`. To je hlavní konfigurační soubor systému X Window. Zde se nachází veškerá nastavení vaší grafické karty, myši a monitoru.

Každá sekce souboru `xorg.conf` popisuje určitou část konfigurace a má následující podobu:

```
Section název
    položka 1
    položka 2
    položka n
EndSection
```

Rozlišovány jsou následující typy sekcí:

**Tabulka 14.1** *Sekce `/etc/X11/xorg.conf`*

Typ sekce	Popis
<code>Files</code>	Tato sekce obsahuje cesty použité pro vyhledávání fontů a tabulku RGB.
<code>ServerFlags</code>	Zde se zadávají obecné volby pro X server.
<code>InputDevice</code>	Zde se konfiguruje vstupní zařízení jako klávesnice a speciální zařízení (touchpady, joysticky atd.). Důležitými položkami jsou: <code>Driver</code> a volby určující položky <code>Protocol</code> a <code>Device</code> .
<code>Monitor</code>	Popisuje použitý monitor: jméno, na které později odkazuje definice <code>Screen</code> , dále šířka pásma a obě mezní synchronizační frekvence ( <code>HorizSync</code> a <code>VertRefresh</code> ). Frekvence se zadávají v MHz, kHz, nebo Hz. Server obvykle odmítne jakékoli zobrazovací parametry, které neodpovídají specifikaci monitoru. Cílem je zabránit náhodnému nastavení monitoru na příliš vysokou řádkovou nebo snímkovou frekvenci.

Typ sekce	Popis
Modes	Zde jsou uloženy zobrazovací parametry pro různá rozlišení obrazovky. Jejich hodnoty jsou obvykle vypočteny programem SaX2 na základě údajů zadaných uživatelem a obvykle je není třeba měnit. Ruční zásah může být nutný např. při použití monitoru s pevnými frekvencemi. Podrobnější popis jednotlivých parametrů by byl nad rámec této knihy, ale najdete ho např. v dokumentu HOWTO <code>/usr/share/doc/howto/en/XFree86-Video-Timings-HOWTO.gz</code> .
Device	Zde je definována konkrétní grafická karta v systému, na kterou je prostřednictvím jejího názvu odkazováno v jiných sekcích konfiguračního souboru.
Screen	Zde je definován vztah mezi sekcemi Monitor a Device, jimiž je tvořena nezbytná konfigurace systému XFree. V podsekcích Display je určena barevná hloubka a škála rozlišení obrazovky použitelná pro danou hloubku.
ServerLayout	V této sekci je definována použitá kombinace vstupních zařízení ze sekce InputDevice a zobrazovacích zařízení (sekce Screen), ať už je v systému jedna grafická karta nebo se jedná o režim multihead (více karet provozovaných zároveň).

O sekcích Monitor, Device, a Screen se podrobněji dočtete dále. Informace o ostatních sekcích naleznete například v manuálových stránkách `XFree86 a xorg.conf`.

Konfigurační soubor `xorg.conf` může obsahovat více různých sekcí Monitor a Device. V souboru může existovat i více sekcí typu Screen. V sekci ServerLayout, která po nich následuje, je pak určeno, které sekce budou skutečně použity.

## 14.2.1 Sekce Screen

Nyní se pozastavíme u sekce Screen, která je styčným místem sekce Monitor a sekce Device a určuje, jaké kombinace barevné hloubky a rozlišení obrazovky budou použity.

Příklad sekce Screen:

```
Section "Screen"
    DefaultDepth 16
    SubSection "Display"
        Depth 16
        Modes "1152x864" "1024x768" "800x600"
        Virtual 1152x864
    EndSubSection
    SubSection "Display"
        Depth 24
        Modes "1280x1024"
    EndSubSection
    SubSection "Display"
        Depth 32
        Modes "640x480"
    EndSubSection
    SubSection "Display"
        Depth 8
        Modes "1280x1024"
    EndSubSection
    Device "Device[0]"
    Identifier "Screen[0]"
    Monitor "Monitor[0]"
EndSection
```

V řádce `Identifier` (zde `Screen[0]`) je dán jednoznačný název této sekce, na nějž je odkazováno v následující sekci `ServerLayout`. Řádky `Device` a `Monitor` určují kombinaci grafické karty a monitoru, pro které je tato sekce `Screen` platná a ve skutečnosti jsou to jen odkazy na odpovídající sekce `Device` a `Monitor` konfiguračního souboru. Těm se budeme více věnovat později.

Řádkou `DefaultDepth` nastavíte barevnou hloubku, se kterou se spustí X server, pokud nebude explicitně stanoveno jinak. Každé barevné hloubce odpovídá jedna podsekce `Display`. Na řádce `Depth` je této podsekci přiřazena konkrétní barevná hloubka, jejíž hodnoty mohou být 8, 15, 16, 24 a 32. Všechny moduly X serveru však nepodporují všechny hodnoty. Pro některé grafické karty znamenají hodnoty 24 a 32 totéž, zatímco u jiných udává hodnota 24 tzv. `packed-pixel 24 bpp` mód a 32 tzv. `padded-pixel 32 bpp` mód.

Nastavené barevné hloubce odpovídá seznam rozlišení obrazovky v sekci `Modes`. Tento seznam je zpracováván zleva doprava X serverem, který přiřadí danému rozlišení



příslušný řádek `Modeline` se zobrazovacími parametry. Jejich hodnoty jsou závislé na schopnostech grafické karty a monitoru. Výsledný řádek je tedy předurčen obsahem sekce `Monitor`.

První nalezené platné rozlišení je tzv. `Default mode` a X server se s ním pustí. Během jeho provozu se pak dá kombinací kláves `Ctrl` + `Alt` + `+` (na numerické klávesnici) přepínat mezi hodnotami v seznamu směrem doprava, zatímco kombinací kláves `Ctrl` + `Alt` + `-` procházíme seznam směrem vlevo. Tím se dá měnit rozlišení obrazovky i za běhu X serveru.

Poslední řádka podsekce `Display` s označením `Depth 16` udává barevnou hloubku a přímo ovlivňuje maximální velikost virtuální obrazovky. Ta je dále závislá na velikosti videopaměti, nikoli na maximálním rozlišení monitoru. Moderní grafické karty mají jsou osazeny pamětí o dostatečné velikosti, lze tedy používat velké virtuální obrazovky. Pokud má grafická karta videopaměť např. o 16 MB, lze při barevné hloubce 32 bitů vytvořit virtuální obrazovku o velikosti až 2048x248 bodů. Zejména u moderních akcelerovaných karet však není doporučeno použít veškerou dostupnou paměť na virtuální obrazovku, neboť jejich paměť slouží také jako vyrovnávací paměť pro uložení fontů a grafických objektů.

## 14.2.2 Sekce Device

Tato sekce popisuje konkrétní grafickou kartu. Soubor `xorg.conf` může obsahovat více těchto sekcí, které jsou odlišeny hodnotou řádku `Identifier`. Máte-li více grafických karet, sekce jsou očíslovány tak, že první karta bude `Device[0]`, druhá karta `Device[1]` atd. Následující výpis je příklad konfigurace sekce `Device` u počítače s jednou kartou Matrox Millennium PCI:

```
Section "Device"
    BoardName      "MGA2064W"
    BusID          "0:19:0"
    Driver         "mga"
    Identifier     "Device[0]"
    VendorName     "Matrox"
    Option        "sw_cursor"
EndSection
```

Při konfiguraci pomocí SaX2 bude vaše sekce `Device` vypadat podobně. Položky `Driver` a se liší podle hardwaru ve vašem počítači a `BusID` jsou zjištěny programem SaX2 automaticky. Hodnota `BusID` představuje pozici na sběrnici PCI nebo AGP, ve které je instalována grafická karta. Odpovídá hodnotě zjištěné příkazem `lspci` (nenechte

se nicméně zkrát tím, že X server zde používá dekadické hodnoty a program `lspci` hodnoty hexadecimální.

V sekci `Driver` přiřadíte grafické kartě ovladač. Máte-li např. kartu Matrox Millennium, nazývá se modul ovladače `mga`. X server pak hledá daný modul v podadresáři s ovladači uvedeném v položce `ModulePath` v sekci `Files`. Ve výchozím stavu po instalaci to je adresář `/usr/X11R6/lib/modules/drivers`. Pokud ke jménu modulu přidáte `_drv.o`, získáte jméno souboru s ovladačem, v případě modulu `mga` bude tedy zaveden soubor `mga_drv.o`.

Chování X server nebo ovladačů lze ovlivnit dalšími volbami. Příkladem je například volba `sw_cursor` ze sekce `Device`, která zakáže hardwarový kurzor myši a simuluje ho hardwarově. Různé ovladače mohou mít implementovány různé volby. Popis voleb dostupných u konkrétního ovladače najdete v adresáři `/usr/X11R6/lib/X11/doc` (máte-li nainstalován balík `XFree-doc`. Popis obecně platných voleb obsahují také manuálové stránky (`man xorg.conf` a `man XFree86`).

## 14.2.3 Sekce Monitor a Modes

Podobně jako každá sekce `Device` popisuje jednu grafickou kartu, popisují sekce `Monitor` a `Modes` jeden monitor. Konfigurační soubor může obsahovat libovolné množství těchto sekcí (lišících se minimálně v jejich symbolických jménech). V sekci `SystemLayout` je pak určeno, která ze sekcí `Monitor` je platná.

Nastavení monitoru by měli provádět pouze zkušební uživatelé. Nejdůležitějšími položkami sekcí `Monitor` jsou horizontální a vertikální frekvence monitoru pro dané rozlišení.

---

### Varování

Pokud nerozumíte principům spolupráce monitoru a grafické karty, hodnoty frekvencí neměňte, neboť to zejména u starších monitorů může vést až k jejich zničení.

---

Pokud si troufáte ručně měnit navrženou konfiguraci monitoru, měli byste věnovat pozornost dokumentaci `/usr/X11/lib/X11/doc`. Velký význam má zejména část popisující režimy monitoru, manipulaci s horizontální a vertikální frekvencí a funkci grafických komponent systému.

V dnešní době se s ručním nastavením frekvencí monitoru prakticky nesetkáte. Při použití moderního monitoru schopného přizpůsobit obraz libovolné frekvenci generované grafickou kartou v určitém rozsahu (dnes v tomto režimu pracuje naprostá většina monitorů), dokáže X server zpravidla zjistit rozsah frekvencí a optimální rozlišení pomocí DDC přímo od monitoru. Této možnosti využívá i konfigurační program SaX2. Pokud se to nepodaří, může využít i X serverem nabízené módy VESA, jež fungují prakticky pro jakékoli kombinace monitorů a grafických karet.

## 14.3 Instalace a konfigurace fontů

V systému SUSE Linux je instalace dalších fontů velmi jednoduchá. Stačí když fonty přepokopírujete do určité adresářové struktury X11, (viz odstavec „[Systém písem X11 Core](#)“ (strana 247)), tak aby je mohl používat nový systém pro zobrazování fontů - xft. Instalační adresář s fonty by tedy měl být podadresářem adresářů, jež jsou uvedeny v `/etc/fonts/fonts.conf` (viz odstavec „[Xft](#)“ (strana 244)).

Fonty můžete (jako uživatel `root`) přepokopírovat ručně do adresáře jako je např. `/usr/X11R6/lib/X11/fonts/truetype`. Instalaci fontů lze provést také pomocí Ovládacího centra KDE - položka Vzhled a motivy->Písma.

Místo kopírování fontů můžete vytvořit také symbolické odkazy na fonty, které jsou uloženy na připojeném diskovém oddílu se systémem Windows. Pak stačí spustit příkaz `SuSEconfig --module fonts`.

Příkaz `SuSEconfig --module fonts` spustí skript `/usr/sbin/fonts-config`, který zajistí instalaci fontů. Pokud vás zajímá, co přesně tento skript dělá, podívejte se do jeho manuálové stránky např. příkazem (`man fonts-config`).

Ať už se jedná o písma bitmapová, TrueType, OpenType nebo Type1 (Postskriptová), tento postup je stejný. Fonty všech těchto typů mohou být umístěny v jednom adresáři. Jedinou výjimkou jsou tzv. CID-keyed fonty (tyto fonty umožňují kombinovat znaky různých kódování a používají se pro japonštinu, čínštinu a podobné jazyky). U těchto písem se instalační postup poněkud liší, viz odstavec „[Písma s kódováním CID \(CID-Keyed\)](#)“ (strana 248).

## 14.3.1 Systémy písem

X.Org používá dva naprosto rozdílné systémy písem: původní *X11 Core-Font systém* a nově navržený *Xft/fontconfig*. V následující části si stručně popíšeme jejich charakteristiku.

### Xft

Při vývoji Xft byl od počátku kladen důraz na podporu škálovatelných písem včetně jejich vyhlazování. Na rozdíl od X11 Core písem nejsou písma spravována X serverem, ale jednotlivými aplikacemi. Jednotlivé programy získaly přímý přístup ke konfiguračním souborům písem a tím i kontrolu na interpretaci jednotlivých znaků. Zároveň je díky tomu zaručeno, že tisk z těchto programů bude vypadat přesně tak, jak vidíte na obrazovce.

V systému SUSE Linux obě velká grafická prostředí KDE a GNOME, program Mozilla i řada dalších aplikací již standardně Xft používá a tento systém je dnes používán více než tradiční X11-Core.

Systém Xft používá při vyhledávání písem a jejich interpretaci knihovnu fontconfig. Její chování lze ovlivnit globálním konfiguračním souborem `/etc/fonts/fonts.conf` a uživatelskými konfiguračními soubory `~/ .fonts.conf`. Každý konfigurační soubor musí začínat touto hlavičkou:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

a končit patičkou

```
</fontconfig>
```

Každý adresář s fonty je v konfiguračním souboru definován na samostatném řádku následujícím způsobem:

```
<dir>/usr/local/share/fonts/</dir>
```

Není však nutné přidávat do souboru nový záznam pro každý adresář. Jako výchozí uživatelský adresář s písmi je v `/etc/fonts/fonts.conf` nastaven adresář `~/ .fonts`. Chcete-li si tedy nainstalovat další písma, nakopírujte je do `~/ .fonts` ve svém domovském adresáři.

Můžete zde také definovat pravidla určující vzhled písem. Takto například vypnete vyhlazování pro všechna písma:

```
<match target="font">
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

Vyhlazování pro konkrétní písma pak povolíte např. takto:

```
<match target="font">
  <test name="family">
    <string>Luxi Mono</string>
    <string>Luxi Sans</string>
  </test>
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

Ve svém výchozím nastavení používá většina aplikací písma `sans-serif` (nebo jejich ekvivalent `sans`), `serif`, nebo `monospace`. Nejde o skutečné fonty, ale o aliasy, které podle jazykového nastavení teprve ukazují na konkrétní písma.

Uživatel si může vytvořit vlastní soubor `~/ .fonts.conf` a nasměrovat zde tyto aliasy na svá oblíbená písma:

```
<alias>
  <family>sans-serif</family>
  <prefer>
    <family>FreeSans</family>
  </prefer>
</alias>
<alias>
  <family>serif</family>
  <prefer>
    <family>FreeSerif</family>
  </prefer>
</alias>
<alias>
  <family>monospace</family>
  <prefer>
    <family>FreeMono</family>
  </prefer>
</alias>
```

Protože systém aliasů používají téměř všechny aplikace, ovlivní tyto změny celý systém. Máte tak možnost centrálně nastavit používání vašich oblíbených písem a nemusíte měnit konfiguraci v každé aplikaci zvlášť.

Příkazem `fc-list` získáte seznam nainstalovaných písem. Pokud vás zajímá pouze určitý typ písem, např. škálovatelný (`:outline=true`) s hebrejskými znaky (`:lang=he`), obsahující ve jméně slovo (`family`) a chcete znát jeho styl (`style`), řez (`weight`) a název souboru, v němž se písmo nachází, zadejte příkaz:

```
fc-list ":lang=he:outline=true" family style weight file
```

Výstup tohoto příkazů může vypadat např. takto:

```
/usr/X11R6/lib/X11/fonts/truetype/FreeSansBold.ttf:
FreeSans:style=Bold:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoBoldOblique.ttf:
FreeMono:style=BoldOblique:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSerif.ttf:
FreeSerif:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifBoldItalic.ttf:
FreeSerif:style=BoldItalic:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSansOblique.ttf:
FreeSans:style=Oblique:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifItalic.ttf:
FreeSerif:style=Italic:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoOblique.ttf:
FreeMono:style=Oblique:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeMono.ttf: FreeMono:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSans.ttf: FreeSans:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifBold.ttf:
FreeSerif:style=Bold:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSansBoldOblique.ttf:
FreeSans:style=BoldOblique:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoBold.ttf:
FreeMono:style=Bold:weight=200
```

Důležité parametry příkazu `fc-list` jsou:

#### **Tabulka 14.2** *Vybrané parametry příkazu `fc-list`*

<b>Parametr</b>	<b>Popis a možné hodnoty</b>
<code>family</code>	Název rodiny písma, např., <code>FreeSans</code> .
<code>foundry</code>	Výrobce písma, např., <code>urw</code> .
<code>style</code>	Styl písma, např. <code>Medium</code> , <code>Regular</code> , <code>Bold</code> , <code>Italic</code> , <code>Heavy</code> .

Parametr	Popis a možné hodnoty
lang	Jazyky, které písmo podporuje, např. <code>cs</code> pro češtinu, <code>ja</code> pro japonštinu, <code>zh-TW</code> pro tradiční čínštinu, <code>zh-CN</code> pro zjednodušenou čínštinu atd.
weight	Tloušťka písma, např., 80 pro normální, 200 pro tučné.
slant	Šikmost 0 pro normální písmo, 100 pro kurzívu.
file	Název souboru s písmem.
outline	<code>true</code> pokud se jedná o obrysová písma, <code>false</code> pro ostatní.
scalable	<code>true</code> pokud se jedná o škálovatelná písma, <code>false</code> pro ostatní.
bitmap	<code>true</code> u bitmapových písem, <code>false</code> u ostatních.
pixelsize	Velikost písma v pixelech. Má význam pouze u bitmapových písem.

## System písma X11 Core

System X11 Core byl navržen v roce 1987 pro zpracování monochromatických bitmapových písem v X11R1. Dnes podporuje kromě bitmapových písem i škálovatelná písma jako jsou fonty Type1, TrueType, OpenType a písma typu CID-keyed. Již velmi dlouho jsou podporována také uncodová písma. Zdaleka však nenabízí takové možnosti jako Xft/fontconfig.

Například u škálovatelných písem není implementována podpora antialiasingu. Zpracování fontů se znaky v mnoha jazycích může trvat déle. Také použití Unicodových písem vede ke zpomalení a vyžaduje více paměti.

System písma X11 Core zdědil několik slabín. Je zastaralý a nedá se rozumným způsobem rozšiřovat. Z důvodu zpětné kompatibility je stále zachovávan při životě, nicméně je vhodné ho nahradit moderním systémem Xft/fontconfig, pokud je to možné.

X server dokáže zpracovat pouze adresáře splňující jednu z následujících podmínek:

- Adresář je uveden v direktivě `FontPath` v části `Files` konfiguračního souboru `/etc/X11/XF86Config`.
- Adresář obsahuje platný soubor `font.dir` (vytvořený skriptem `SuSEconfig`).
- Adresář není za běhu X serveru vyřazen ze seznamu adresářů s fonty příkazem `xset -fp`.
- Adresář je zařazen za běhu X serveru do seznamu adresářů s fonty příkazem `xset +fp`.

Pokud X server už běží, lze nově nainstalované (tj. do příslušných adresářů nakopírované) fonty zpřístupnit příkazem `xset fp rehash`. Tento příkaz je spuštěn skriptem `SuSEconfig --module fonts`.

Příkaz `xset` potřebuje přímý přístup k běžícímu X serveru, skript `SuSEconfig --module fonts` tedy musí být spuštěn ze shellu, který k němu přístup má. Toto lze nejjednodušeji zajistit získáním administrátorských oprávnění, tj. zadáním příkazu `su and` hesla uživatele `root`. Příkaz `su` předá přístupová oprávnění uživatele, který spustil X server, administrátorskému shellu. Korektní instalaci písem a jejich dostupnost prostřednictvím systému X11 core fontů ověříte příkazem `xlsfonts`, jenž vrátí právě seznam všech dostupných písem.

SUSE Linux používá ve výchozím nastavení kódování UTF-8. Je tedy vhodné dávat přednost fontům typu Unicode, jež poznáte tak, že ve výstupu příkazu `xlsfonts` bude jméno fontu končit na `iso10646-1`. Seznam všech Unicodových písem nainstalovaných na vašem systému získáte příkazem `xlsfonts | grep iso10646-1`. Protože téměř všechna písmena typu Unicode ze systému SUSE Linux obsahují alespoň znaky evropských abeced, nahradilo kódování Unicode předchozí kódování `iso-8859-*`.

## Písmena s kódováním CID (CID-Keyed)

Narozdíl od jiných typů písem nelze písmena s kódováním CID umístěna v libovolném adresáři. Musíte je instalovat do adresáře `/usr/share/ghostscript/Resource/CIDFont`. Pro Xft/fontconfig nehraje sice umístění fontů žádnou roli, ale Ghostscript a systém fontů X11 Core vyžadují, aby se nacházela právě zde.



---

## Tip

Další informace o fontech v prostředí X11 najdete na stránce <http://www.xfree86.org/current/fonts.html>.

---

# 14.4 Konfigurace OpenGL – 3D

## 14.4.1 Podpora hardware

SUSE Linux používá pro 3D podporu několik OpenGL ovladačů. Jejich přehled se nachází v tabulce 14.3 – „Karty s podporou 3D“ (strana 249):

**Tabulka 14.3** *Karty s podporou 3D*

---

Ovladač OpenGL	Podporovaný hardware
nVidia	čipové sady nVidia: všechny kromě Riva 128(ZX)
DRI	3Dfx Voodoo Banshee, 3Dfx Voodoo-3/4/5, Intel i810/i815/i830M, Intel 845G/852GM/855GM/865G, Matrox G200/G400/G450/G550, ATI Rage 128(Pro)/Radeon

---

Při instalaci nové karty do systému pomocí programu YaST nebo již při prvotní konfiguraci systému lze aktivovat 3D podporu. Pokud YaST nerozpozná vaši kartu automaticky, můžete ji vybrat sami ze seznamu. Výjimkou jsou grafické čipy společnosti nVidia. Originální ovladač s 3D podporou pro tyto čipy není v distribuci z licenčních důvodů obsažen, a pokud vyžadujete podporu 3D, musíte si ho nejprve stáhnout a nainstalovat – nejnázne pomocí YOU (YaST Online Update).

Pokud neprovádíte novou instalaci, ale aktualizaci, nebo přidáváte akcelerátor Voodoo Graphics (či Voodoo-2), postup při nastavení 3D podpory se poněkud liší podle toho, jaký ovladač OpenGL použijete. Více informací najdete v následujících odstavcích.

## 14.4.2 Ovladače OpenGL

Prostřednictvím programu SaX2 lze OpenGL nVidia a DRI ovladače jednoduše konfigurovat. V případě karet společnosti nVidia je třeba nejprve stáhnout originální ovladač. Příkazem `3Ddiag` ověříte, zda byla instalace a konfigurace nVidia nebo DRI ovladače úspěšná.

Z bezpečnostních důvodů mají k 3D hardwaru přístup jen uživatelé patřící do skupiny `video`, proto se přesvědčete, že všichni lokální uživatelé jsou členy této skupiny. V opačném případě se ovladač OpenGL přepne do režimu tzv. softwarového renderingu (vykreslování obrazu má na starosti software a nikoli hardware), což se významně projeví na rychlosti aplikací využívajících OpenGL. Pokud příslušní uživatelé do skupiny `video` nepatří (což ověříte např. příkazem `id`), je vhodné je do skupiny přidat, např. programem YaST.

## 14.4.3 Diagnostický nástroj 3Ddiag

Diagnostický nástroj `3Ddiag` slouží v systému SUSE Linux ke kontrole konfigurace podpory pro 3D. Jedná se o program, který je nutno spouštět z příkazové řádky. Seznam voleb tohoto příkazu získáte zadáním `3Ddiag -h`.

Program zkontroluje, zda jsou nainstalovány balíky zajišťující 3D podporu a zda jsou použity správné knihovny OpenGL popř. rozšíření GLX. Pokud ve výstupu programu najdete hlášení *failed*, řiďte se jeho dalšími instrukcemi. Pokud je všechno v pořádku, objeví se pouze zpráva *done*.

## 14.4.4 Testování OpenGL

Funkčnost OpenGL můžete vyzkoušet programem `glxgears` popř. pomocí her `tuxracer` nebo `armagetron` (balíčky mají stejné názvy). Při aktivované podpoře 3D by měly být hry hratelné i na slabších počítačích, bez této podpory poběží hry pomalu – obraz bude trhaný. Dalším prostředkem, který ověří, zda má váš systém podporu

pro 3D, je příkaz `glxinfo | grep direct`, jehož výsledkem by měl být řádek `direct rendering: Yes`.

## 14.4.5 Řešení problémů

Pokud máte s OpenGL nějaké problémy (např. hry jsou trhané), zkontrolujte programem 3Ddiag konfiguraci OpenGL, a pokud se objeví hlášení *failed*, odstraňte daný problém podle instrukcí. Pokud opravný zásah nepomohl, popřípadě se ve výstupu 3Ddiag žádná závada neobjevila, přičemž váš problém s 3D přetrvává, nahlédněte do protokolových souborů X.Org.

Často zjistíte, že se v protokolovém souboru `X.Org /var/log/Xorg.0.log` objevuje hláška `DRI is disabled`, jejíž přesnou příčinu lze objevit zevrubným zkoumáním protokolového souboru. Je to však úkol pro zkušeného uživatele.

Pokud se s tím setkáte, většinou se o chybu v konfiguraci nejedná, neboť program 3Ddiag by ji již odhalil. Pak vám obvykle zbývá jediná možnost – používat DRI ovladač v režimu softwarového renderingu, tj. bez využití podpory 3D, kterou obsahuje váš hardware. Pokud dochází k chybám v zobrazení nebo jsou aplikace používající OpenGL nestabilní, bude lepší, když 3D podporu vypnete pomocí SaX2 úplně.

## 14.4.6 Instalační podpora

Pokud pomíneme režim softwarového renderingu v ovladači DRI, jsou všechny linuxové ovladače OpenGL ve vývojovém stádiu a jsou tedy považovány za experimentální. Ovladače byly však zařazeny do distribuce, protože poptávka po podpoře 3D v Linuxu je vysoká. Vzhledem ke stavu ovladačů však nejsme schopni zajistit instalační podporu uživatelům ohledně konfigurace hardwarové akcelerace 3D, ani řešení podobných problémů. Ve výchozím nastavení X serveru není hardwarová akcelerace zapnuta, a pokud se při jejím používání setkáváte s nějakými problémy, doporučujeme ji úplně vyřadit.

## 14.4.7 Dodatečná online dokumentace

Informace o DRI naleznete v souboru `/usr/X11R6/lib/X11/doc/README.DRI (xorg-x11-doc)`. Více informací o instalaci nVidia ovladačů naleznete na adrese

<http://ftp.suse.com/pub/suse/i386/supplementary/X/nvidia-installer-HOWTO.html>.

# FreeNX: vzdálené ovládaní plochy 15

FreeNX je GPL implementace NX serveru pro vzdálený přístup k pracovní ploše na jiném počítači. Umožňuje vzdálenou práci i přes nízkou průpropnost a rychlost sítě.

## 15.1 Úvod do NX

V následující výčtu najdete kroky potřebné pro nastavení funkčního NX serveru pro 10 klientů:

- 1 Na servery a klienty pomocí programu YaST nainstalujte následující software:

Server	Klient
<ul style="list-style-type: none"><li>• NX</li></ul>	<ul style="list-style-type: none"><li>• NX</li></ul>
<ul style="list-style-type: none"><li>• FreeNX</li></ul>	<ul style="list-style-type: none"><li>• knx (pro KDE sezení)</li><li>• NoMachine <code>nxclient</code> (pro ostatní prostředí)</li></ul>

- 2 Nastavte NX server jako uživatel `root` příkazem:

```
nxsetup --install --clean --purge --setup-nomachine-key
```

Server se spustí s výchozím nastavením ze souboru `/etc/nxserver/node.conf`. K serveru se mohou připojit všechny pracovní stanice. Pokud chcete přístup omezit, použijte pro klienty např. distribuované klíče.

- 3 Na NX server ve firewallu povolte NX připojení. V pokročilém nastavení povolte porty 22 (SSH), 5000 až 5009 a 7000 až 7009. Zápis pro *TCP porty* bude vypadat takto:

```
22 5000:5009 7000:7009
```

---

## Tip

Další informace o nastavení firewallu pro NX server najdete v souboru `/usr/share/doc/packages/FreeNX/NX-Firewall.txt`.

---

K serveru se ze vzdálené pracovní stanice připojíte následujícím způsobem:

- 1 Z nabídky spusťte KNX.
- 2 Při prvním přihlášení musíte vytvořit nové připojení, které vytvoříte takto:
  - a V *KNX Client Login* klikněte na *Connection Settings*.
  - b Zadejte jméno připojení a jméno serveru.
  - c Zadejte informace o počítači, číslo portu a rychlost vašeho připojení.
  - d Kde spustíte v *Sessiontype* volbou *UNIX/KDE*.
  - e Zvolte rozlišení monitoru.
  - f Klikněte na tlačítko *OK*.
- 3 Po přihlášení budete mít přístup k ploše vzdáleného počítače.

Pokud se chcete připojit z GNOME, postupujte takto:

- 1 Stáhněte a nainstalujte si balíček `nxclient` ze stránky NoMachine [http://www.nomachine.com/download\\_client\\_linux.php](http://www.nomachine.com/download_client_linux.php).

- 2 Z hlavní nabídky zvolte *NX Connection Wizard*.
- 3 Ve třech krocích zadejte jméno připojení, port a podrovnosti o počítači a zvolte typ sezení *Unix/Gnome*. Pokud chcete na ploše ikonu NX, zvolte příslušnou volbu. Po zadání uživatelského jména a hesla klikněte na tlačítko *OK*.

Nyní se můžete připojit ke vzdálené ploše.

## 15.2 Možné problémy

V této části najdete postupy spojené s odstraněním nejčastějších problémů.

### 15.2.1 knx selže při pokusu o navázání připojení

Snažíte se připojit k NX serveru přes knx. Při inicializaci selže ověření uživatele.

Abyste zjistili příčinu a provedli nápravu postupujte následujícím způsobem:

- 1 Překontrolujte, zda neběží na serveru Novell AppArmor, postup je uveden v části [15.2.2 – „Nelze se připojit k NX serveru“](#) (strana 256).
- 2 Pokuste se znovu připojit k serveru.
- 3 Překontrolujte, zda je na klientovi v nastavení firewallu v povolených službách uvedena také položka SSH. Pokud není, povolte SSH.
- 4 Překontrolujte v nastavení firewallu serveru, zda jsou otevřené porty pro SSH a NX uvedené v části [15.1 – „Úvod do NX“](#) (strana 253). Pokud některé z uvedených portů jsou zakázány, otevřete je.
- 5 `Retry establishing a connection between knx and the server.`
- 6 Přihlaste se jako `root` na server a postupujte následujícím způsobem:
  - a Přejděte do adresáře `/tmp` a vyhledejte soubory zámků NX serveru:

```
cd /  
ls -ltr .nx*
```

**b** Staré soubory zámků odstraňte

**c** Odhlaste se

**7** Znovu se pokuste připojit k serveru.

**8** Pomocí programu YaST na klientovi smažte a znovu nainstalujte knx klienta.

Nyní byste měli být schopni se připojit k serveru.

## 15.2.2 Nelze se připojit k NX serveru

Po spuštění knx a inicializaci připojení systém vypsal následující chybové hlášení:

```
Connection to NX server could not be established. Connection timed out.
```

Abyste odhalili příčinu problémů, postupujte takto:

**1** Přihlaste se k serveru jako uživatel `root`

**2** Ve výstupu příkazu `dmesg` vyhledejte hlášení podobné následujícímu příkladu:

```
SubDomain: REJECTING r access to  
/var/lib/nxserver/home/.ssh/authorized_keys2 (sshd(31247) profile  
/usr/sbin/sshd active /usr/sbin/sshd)
```

Podle této zprávy běží na serveru Novell AppArmor, který neumožňuje ssh démonovi přístup k některým NX souborům.

**3** Zakažte na počítači AppArmor

*nebo*

nastavte ssh do výukového režimu a nastavte správně práva přístupu k NX souborům v existujícím profilu- Více informací najdete v manuálu *Novell AppArmor Powered by Immunix 2.0 Administration Guide*.

**4** Připojte se k serveru.



## 15.2.3 Ověření uživatele proběhlo úspěšně, ale spojení není navázáno

Po spuštění knx se úspěšně ověříte, ale místo nového okna s novým sezením vypíše systém chybu podobnou následující:

```
Could not yet establish the connection to the remote proxy. Do you  
    want to terminate the current session?
```

Pravděpodobnou příčinou je neotevřený port pro NX sezení na vašem firewallu. Proveďte nastavení podle postupu popsaného v části [15.1 – „Úvod do NX“](#) (strana 253).

## 15.3 Další informace

Více informací o balíčku FreeNX najdete v README v souboru `/usr/share/doc/packages/FreeNX/README.SUSE`. Další informace získáte zadáním příkazu `nxserver --help`.



# Autentizace pomocí PAM

Linux používá PAM (Pluggable Authentication Modules – připojovatelné autentizační moduly) při procesu autentizace jako zprostředkující vrstvu mezi uživatelem a aplikací. PAM moduly jsou dostupné v celém systému, takže mohou být použity libovolnou aplikací. Tato kapitola se věnuje popisu funkce modulárního autentizačního mechanismu a jeho konfiguraci.

Systémoví administrátoři a programátoři často potřebují omezit přístup k určitým částem systému nebo použití určitých funkcí aplikace. Bez využití PAM by aplikace musely být upraveny, kdykoliv je zaveden nový autentizační mechanismus (jako LDAP nebo SAMBA). To je však časově náročný a k chybám náchylný proces. Problémům se lze vyhnout oddělením aplikací od autentizačního procesu a převedením autentizační funkce na centrálně spravované moduly. Kdykoliv je pak potřeba zavést nový autentizační mechanismus, stačí upravit nebo napsat příslušné PAM moduly.

Každý program závislý na mechanismu PAM má svůj vlastní konfigurační soubor v adresáři `/etc/pam.d/<jmenoprogramu>`. Tyto soubory určují, jaké PAM moduly mají být použity při autentizaci. Navíc pro většinu PAM modulů existují globální konfigurační soubory uložené v adresáři `/etc/security` (např. `pam_env.conf`, `pam_pwcheck.conf`, `pam_unix2.conf`, `time.conf`). Ty určují přesné chování modulů. Každá aplikace používající PAM modul ve skutečnosti volá sadu PAM funkcí, které následně zpracují údaje v různých konfiguračních souborech a vrátí výsledek volající aplikaci.

# 16.1 Struktura PAM konfiguračního souboru

Každý řádek PAM konfiguračního souboru obsahuje nejvýše čtyři sloupce:

```
<Typ modulu> <Kontrolní příznak> <Jméno modulu> <Parametry>
```

Moduly PAM jsou zpracovávány postupně za sebou. Různé moduly mají různé účely. Jeden modul například kontroluje správnost hesla, jiný ověřuje umístění, z kterého je k systému přistupováno, a další načítá uživatelsky specifická nastavení. PAM obsahuje čtyři různé typy modulů:

`auth`

Účelem modulu tohoto typu je autentizovat uživatele. Obvykle se tak činí ověřením hesla, ale lze toho dosáhnout i s pomocí čipových karet nebo biometrie (otisků prstů či rozpoznání oční duhovky).

`account`

Moduly tohoto typu ověřují, zda má uživatel obecné oprávnění využít příslušnou službu. Například lze s jejich pomocí zajistit, aby se k systému nemohl přihlásit nikdo pod uživatelským jménem, jehož účet vypršel.

`password`

Smyslem tohoto typu modulu je umožnit změnu autentizačního tokenu. Tímto tokenem je ve většině případů heslo.

`session`

Moduly tohoto typu jsou zodpovědné za správu a konfiguraci uživatelských relací. Jsou spuštěny před a po autentizaci, aby zaznamenaly pokusy o přihlášení do systémových logů a nakonfigurovaly uživatelsky specifické prostředí (poštovní účty, domovský adresář, systémová omezení atd.).

Druhý sloupec obsahuje kontrolní příznaky, které ovlivňují chování spuštěných modulů:

`required`

Modul s tímto příznakem musí být úspěšně zpracován dříve, než proběhne autentizace. Selže-li modul s příznakem `required`, musí být zpracovány všechny ostatní moduly se stejným příznakem dříve, než je uživatel informován o neúspěšnosti pokusu o autentizaci.

### requisite

Moduly s tímto příznakem musí být, stejně jako moduly s příznakem `required`, úspěšně zpracovány. Nicméně v případě selhání modulu s příznakem `requisite` je uživatel okamžitě informován a nejsou zpracovávány žádné další moduly. Pokud je zpracování úspěšné, jsou zpracovávány i další moduly, stejně jako v případě modulů s příznakem `required`. Příznak `requisite` lze použít jako základní filtr pro ověření podmínek nezbytných pro korektní autentizaci.

### sufficient

Pokud je úspěšně zpracován modul s tímto příznakem, dostane volající aplikace okamžitou zprávu o úspěšnosti autentizace a žádné další moduly nejsou zpracovávány. Platí to však jen tehdy, pokud již dříve nedošlo k selhání modulu s příznakem `required`. Selhání modulu s příznakem `sufficient` nemá žádné přímé důsledky, všechny další moduly jsou zpracovávány v běžném pořadí.

### optional

Úspěch ani selhání modulu s tímto příznakem nemá žádné přímé důsledky. Toho se využívá v případě modulů, jejichž jediným účelem je zobrazit zprávu (například oznámení o příchozí poště).

### include

Tento příznak slouží ke vložení souboru udaného jako argument.

Pokud se modul nachází v implicitním adresáři `/lib/security` (`/lib64/security` na 64-bitových platformách se systémem SUSE Linux), nemusí být cesta explicitně stanovena. Čtvrtý sloupec může obsahovat parametry předávané modulu, jako např. `debug` (umožňuje ladění programu) nebo `nullok` (dovoluje použití prázdných hesel).

## 16.2 Konfigurace PAM pro sshd

Následující praktický příklad ukazuje konfiguraci PAM pro sshd:

### **Rovnice 16.1** *Konfigurace PAM pro sshd*

```
##PAM-1.0
auth    include      common-auth
auth    required     pam_nologin.so
account include     common-account
password include    common-password
session include     common-session
# Enable the following line to get resmgr support for
# ssh sessions (see /usr/share/doc/packages/resmgr/README.SuSE)
#session optional   pam_resmgr.so fake_ttyname
```

Typická PAM konfigurace aplikace (v našem případě sshd) obsahuje čtyři vkládací příkazy (`include`) odkazující na konfigurační soubory čtyř typů modulů: `common-auth`, `common-account`, `common-password` a `common-session`. Tyto čtyři soubory obsahují výchozí konfiguraci pro každý typ modulů. Toto vkládání zajišťuje automatické použití aktuálního výchozího nastavení. Dříve bylo třeba všechny konfigurační soubory pro všechny aplikace upravit ručně, kdykolí došlo k aktualizaci PAM. Nyní existuje centrální konfigurace; jsou-li v ní provedeny změny, automaticky se dědí PAM konfiguracemi jednotlivých služeb.

První vkládaný soubor (`common-auth`) volá dva moduly typu `auth`: `pam_env` a `pam_unix2`. Viz 16.2 – „Výchozí konfigurace pro `auth` sekci“ (strana 262).

### **Rovnice 16.2** *Výchozí konfigurace pro `auth` sekci*

```
auth    required     pam_env.so
auth    required     pam_unix2.so
```

První z nich, `pam_env`, nahraje soubor `/etc/security/pam_env.conf` a nastaví proměnné prostředí specifikované v tomto souboru. To lze využít k nastavení proměnné `DISPLAY` na správnou hodnotu, neboť modul `pam_env` zná místo, ze kterého probíhá přihlašování. Druhý, `pam_unix2`, zkontroluje přihlašovací jméno a heslo podle `/etc/passwd` a `/etc/shadow`.

Po úspěšném zavolání modulů z `common-auth` zkontroluje třetí modul, `pam_nologin`, zda existuje soubor `/etc/nologin`. Pokud existuje, nesmí se přihlásit nikdo kromě superuživatele `root`. Všechny `auth` moduly jsou zpracovány dříve než sshd dostane informaci o výsledku přihlašování. Protože všechny `auth` moduly mají příznak `required`, musí být všechny úspěšně zpracovány před tím, než sshd dostane zprávu o výsledku autentizace. Pokud některý z modulů selže, stejně musí být zpracována celá sada, a teprve potom sshd dostane zprávu o negativním výsledku.

Jakmile jsou všechny auth moduly úspěšně zpracovány, přijde na řadu další vkládací (`include`) příkaz, tentokrát ten, který je uvedený v 16.3 – „Výchozí konfigurace pro account sekci“ (strana 263). Soubor `common-account` obsahuje jen jeden modul, `pam_unix2`. Pokud `pam_unix2` zjistí, že uživatel existuje, dostane `sshd` zprávu o úspěchu a je zpracována další sada modulů (`password`) – viz 16.4 – „Výchozí konfigurace pro password sekci“ (strana 263).

### **Rovnice 16.3** *Výchozí konfigurace pro account sekci*

```
account required          pam_unix2.so
```

### **Rovnice 16.4** *Výchozí konfigurace pro password sekci*

```
password required        pam_pwcheck.so  nullok
password required        pam_unix2.so   nullok use_first_pass use_authok
#password required       pam_make.so    /var/yp
```

PAM konfigurace `sshd` zahrnuje pouze vkládací (`include`) příkaz odkazující na výchozí konfiguraci `password` modulů v souboru `common-password`. Tyto moduly se musí úspěšně zpracovat (příznak `required`) kdykoliv aplikace vyžaduje změnu autentizačního tokenu. Změna hesla či jiného tokenu vyžaduje bezpečnostní kontrolu. Tu zajišťuje modul `pam_pwcheck`. Po něm použitý modul `pam_unix2` přenáší hesla z modulu `pam_pwcheck`, takže se uživatel nemusí znovu autentizovat. Také tím znemožňuje obejít kontroly prováděné modulem `pam_pwcheck`. Moduly typu `password` by měly být používány vždy, když jsou moduly `account` či `auth` nakonfigurovány tak, aby upozorňovaly na vypršení hesla.

### **Rovnice 16.5** *Výchozí konfigurace pro session sekci*

```
session required         pam_limits.so
session required         pam_unix2.so
```

Jako poslední krok jsou volány moduly typu `session` z `common-session`, jejichž úkolem je nastavit relaci pro konkrétního uživatele. Opětovné použití modulu `pam_unix2` nemá žádné praktické důsledky, neboť je volán s parametrem `none`, který je nastaven v konfiguračním souboru tohoto modulu (`pam_unix2.conf`). Modul `pam_limits` zpracovává soubor `/etc/security/limits.conf`, ve kterém mohou být definována omezení pro využívání určitých systémových zdrojů. Moduly typu `session` jsou volány podruhé při odhlášení uživatele.

## 16.3 Konfigurace PAM modulů

Některé PAM moduly jsou konfigurovatelné. Příslušné konfigurační soubory jsou umístěny v adresáři `/etc/security`. Tato kapitola stručně popisuje konfigurační soubory vztahující se k předchozímu příkladu s `sshd`, tj. `pam_unix2.conf`, `pam_env.conf`, `pam_pwcheck.conf` a `limits.conf`.

### 16.3.1 pam\_unix2.conf

Běžná autentizace založená na heslech je řízená PAM modulem `pam_unix2`. Ten může přistupovat k potřebným údajům v `/etc/passwd`, `/etc/shadow`, NIS mapách, NIS+ tabulkách nebo v LDAP databázi. Chování modulu lze ovlivnit individuálním nastavením PAM pro jednotlivé aplikace nebo globálně úpravou souboru `/etc/security/pam_unix2.conf`. Velmi jednoduchý konfigurační soubor pro tento modul ukazuje příklad 16.6 – „`pam_unix2.conf`“ (strana 264).

#### **Rovnice 16.6** *pam\_unix2.conf*

```
auth:    nullok
account:
password:    nullok
session:    none
```

Parametr `nullok` pro moduly `auth` a `password` znamená, že jsou povolena prázdná hesla. Uživatelé také mohou měnit hesla ke svým účtům. Parametr `none` modulu typu `session` znamená, že nebudou logovány žádné zprávy modulu (to je implicitní nastavení). Další konfigurační možnosti jsou popsány v komentářích v samotném souboru a v manuálové stránce `pam_unix2(8)`.

### 16.3.2 pam\_env.conf

Tento soubor lze použít k nastavení standardizovaného uživatelského prostředí, kdykoliv je zavolán modul `pam_env`. Proměnné prostředí lze nastavit pomocí následující syntaxe:

```
VARIABLE [DEFAULT=[hodnota]] [OVERRIDE=[hodnota]]
```

```
VARIABLE
```

Jméno proměnné prostředí, která má být nastavena.



```
[DEFAULT=[hodnota]]
```

Implicitní hodnota proměnné.

```
[OVERRIDE=[hodnota]]
```

Hodnota, na kterou se modul `pam_env` dotáže a kterou přepíše implicitní hodnotu.

Obvyklým příkladem implicitní hodnoty, jež má být modulem `pam_env` přepsána, je proměnná `DISPLAY`, která se mění při každém vzdáleném přihlášení. Viz příklad 16.7 – „`pam_env.conf`“ (strana 265).

### **Rovnice 16.7** `pam_env.conf`

```
REMOTEHOST      DEFAULT=localhost OVERRIDE=@{PAM_RHOST}
DISPLAY         DEFAULT=${REMOTEHOST}:0.0 OVERRIDE=${DISPLAY}
```

První řádka nastavuje proměnnou `REMOTEHOST` na hodnotu `localhost`. Tato hodnota je použita, pokud modul `pam_env` nemůže zjistit jinou hodnotu. Proměnná `DISPLAY` obsahuje hodnotu proměnné `REMOTEHOST`. Další informace lze získat z komentářů v souboru `/etc/security/pam_env.conf`.

## 16.3.3 `pam_pwcheck.conf`

Tento konfigurační soubor je určen pro modul `pam_pwcheck`, který z něj načítá nastavení pro všechny moduly typu `password`. Nastavení z tohoto souboru jsou načtena před PAM nastaveními pro jednotlivé aplikace. Pokud nemá aplikace nastavení definováno specificky, použije se toto globální nastavení. Příklad 16.8 – „`pam_pwcheck.conf`“ (strana 265) přikazuje modulu `pam_pwcheck` povolit prázdná hesla a jejich změnu. Více nastavení je zmíněno v souboru `/etc/security/pam_pwcheck.conf`.

### **Rovnice 16.8** `pam_pwcheck.conf`

```
password: nullok
```

## 16.3.4 `limits.conf`

V souboru `limits.conf`, který je načítán modulem `pam_limits`, lze nastavit systémová omezení pro jednotlivé uživatele nebo jejich skupiny. Umožňuje nastavit pevná omezení, která nelze v žádném případě překročit, a měkká omezení, která mohou být překročena dočasně. Syntaxe souboru a další možnosti nastavení jsou popsány v komentářích.

## 16.4 Další informace

V adresáři `/usr/share/doc/packages/pam` naleznete následující dokumentaci:

### Soubory README

V kořenu adresáře jsou obecně zaměřené README dokumenty. Podadresář `modules` obsahuje README dokumenty zabývající se dostupnými PAM moduly.

### The Linux-PAM System Administrators' Guide

Tento dokument obsahuje vše, co by měl systémový administrátor o PAM vědět. Zabývá se širokým okruhem témat, od syntaxe konfiguračních souborů, až po bezpečnostní aspekty. Dokument je dostupný ve formátech PDF, HTML a jako prostý text.

### The Linux-PAM Module Writers' Manual

Tento dokument shrnuje PAM moduly z pohledu vývojáře. Poskytuje informace o vývoji PAM modulů v souladu se standardy. Je dostupný ve formátech PDF, HTML a jako prostý text.

### The Linux-PAM Application Developers' Guide

Tato příručka obsahuje vše, co potřebuje znát vývojář aplikací používajících PAM knihovny. Je dostupný ve formátech PDF, HTML a jako prostý text.

Thorsten Kukuk napsal množství PAM modulů pro SUSE Linux a některé informace o nich zveřejnil na adrese: <http://www.suse.de/~kukuk/pam/>

## Virtualizace pomocí Xenu

Xen umožňuje běh několika linuxových systémů na jednom fyzickém stroji. Hardware poskytovaný jednotlivým systémům je virtuální. V této kapitole naleznete přehled možností, ale i omezení této technologie. Části o instalaci, konfiguraci a běhu Xenu tento úvod uzavírají.

Virtuální stroje obvykle emulují hardware vyžadovaný systémem. Nevýhodou takového přístupu je skutečnost, že emulovaný hardware je podstatně pomalejší než hardware skutečný. Xen používá jiný přístup. Omezuje emulaci na co nejmenší rozsah, používá tzv. *paravirtualizaci*. Tato technika vytváří virtuální stroje podobné, nikoliv však identické, nativnímu hardwaru. Hostitelský systém i hostované systémy jsou proto upraveny na úrovni jádra. Uživatelský prostor změněn není. Xen ovládá hardware pomocí hypervizoru a ovládacího hosta, tzv. domény-0 (domain-0). Ty poskytují všechna bloková a síťová zařízení potřebná k běhu systému a spojení s ostatními hosty či místní sítí. Pokud jsou na několika fyzických strojích s běžícím Xenem dostupná bloková a síťová zařízení, lze hostovaný systém přenést z jednoho fyzického stroje na druhý za běhu. Původně byl Xen vyvinut pro běh až sto hostovaných systémů na jednom počítači, ale maximální počet systémů závisí na jejich požadavcích, zejména spotřebě paměti.

K omezení využití procesoru nabízí Xen tři různé schedulery (plánovače). Scheduler je možno změnit i za běhu hostovaného systému a tak změnit jeho prioritu. Dostupný výkon procesoru lze změnit také přesunem hostovaného systému na jiný hardware.

Virtualizační systém Xenu má také některé nevýhody týkající se podpory hardwaru:

- Řada ovladačů s uzavřeným zdrojovým kódem, například nVidia či ATI ovladače, nefungují podle očekávání. V takových případech je nahraďte, pokud jsou dostupné,

opensource ovladači, i když nepodporují všechny funkce karet. Při použití Xenu také nejsou podporovány některé čipy WLAN a cardbus bridge.

- Xen verze 2 nepodporuje PAE (physical address extension), což znamená, že nepodporuje více než 4 GB paměti.
- Neexistuje podpora ACPI. Správa napájení a další funkce závislé na ACPI nefungují.

## 17.1 Instalace Xenu

Instalace Xenu zahrnuje vytvoření domény-0 a instalaci Xen klientů. Nejprve se ujistěte, že jsou nainstalovány potřebné balíčky: `python`, `bridge-utils`, `xen` a `kernel-xen`. Jsou-li použity SUSE balíčky, je Xen přidán do konfigurace grubu. V ostatních případech zapište do `boot/grub/menu.lst` podobnou položku:

```
title Xen2
  kernel (hd0,0)/boot/xen.gz dom0_mem=458752
  module (hd0,0)/boot/vmlinuz-xen <parametry>
  module (hd0,0)/boot/initrd-xen
```

(`hd0,0`) nahraďte oddílem obsahujícím váš adresář `/boot` (viz 9 – „*Starování systému a zavaděče*“ (strana 163)). Upravte množství paměti `dom0_mem` podle možností vašeho systému. Maximální možná hodnota je paměť vašeho stroje v kilobajtech mínus 65536. `<parametry>` nahraďte parametry, které normálně používáte při spouštění linuxového jádra. Pak restartujte do Xen režimu. Nastartuje se tax hypervisor Xenu a mírně upravené linuxové jádro jako doména-0 (domain-0), obsluhující většinu hardwaru. Kromě dříve zmíněných výjimek by vše mělo pracovat jako obvykle.

## 17.2 Instalace domény

Instalace a nastavení hostované domény zahrnuje několik operací. V následující části si ukážeme instalaci domény a různé další úlohy.

Nová doména potřebuje ke své existenci kořenový souborový systém na blokovém zařízení nebo obraz souborového systému. K přístupu k tomuto souborovému systému použijte terminál nepoužijte připojení přes síť k běžícímu virtuálnímu systému. Instalaci systému SUSE Linux do adresáře provádí program YaST. Hardwarové požadavky jsou stejné jako pro normální systém.

Domény umí sdílet souborové systémy připojené pouze pro čtení, např. /usr nebo /opt. Nikdy nesdílejte souborové systémy připojené s možností zápisu. Pokud potřebujete zapisovatelné sdílení, použijte NFS, některý z dalších síťových souborových systémů nebo clustrový souborový systém.

---

## Varování: Spuštění nové domény

Pokud spouštíte novou doménu, ujistěte se, že její souborový systém již není připojen instalátorem nebo doménou domain-0.

---

Nejdříve si vytvořte obraz souborového systému pro novou doménu.

- 1 Vytvořte prázdný soubor `guest1` v adresáři `/var/tmp/` o velikosti 4 GB příkazem:

```
dd if=/dev/zero of=/var/tmp/guest1 seek=1M bs=4096 count=1
```

- 2 Vytvořte v souboru souborový systém příkazem:

```
mkreiserfs -f /var/tmp/guest1
```

Příkaz `mkreiserfs` vrátí varování, že nejde o blokové zařízení a požádá vás o potvrzení provedení příkazu. Napište `Y` a stiskněte `Enter`.

- 3 Instalace se provádí do adresáře, takže je nutné soubor `/var/tmp/guest1`, např.:

```
mkdir -p /var/tmp/dirinstall  
mount -o loop /var/tmp/guest1 /var/tmp/dirinstall
```

---

## Důležité

Po dokončení instalace do adresáře odpojte soubor. YaST při instalaci připojí také `/proc`, který odpojíte příkazem:

---

```
umount /var/tmp/dirinstall/proc  
umount /var/tmp/dirinstall
```

## 17.2.1 Instalace domény do adresáře pomocí programu YaST

Spustíte YaST a zvolíte *Software* → *Instalace do adresáře*.

Modul programu YaST bude potřebovat zadat několik voleb:

- Zvolený adresář: `/var/tmp/dirinstall`

Nastavte cestu k obrazu souborového systému. Výchozí nastavení je obvykle vyhovující.

- Spouštění YaST a SuSEconfigu při prvním startu: *Ano*

Nastavte na *Ano*. Při prvním spuštění nové domény vás systém požádá o rotoovské heslo a prvního uživatele.

- Vytvoření obrazu: *Ne*

Volba vytváří tar archiv instalačního adresáře. Tato volba není v našem případě potřebná.

- *Software*

Nastavte typ instalace, výchozí nastavení je obvykle vyhovující.

Instalaci spustíte kliknutím na tlačítko *Next*. Po instalaci je nutné odstranit tls knihovnu:

```
mv /var/tmp/dirinstall/lib/tls /var/tmp/dirinstall/lib/tls.disabled
```

Xen používá jedno z jader nainstalovaných `domain-0` ke spuštění nové domény. Aby v nové hostující doméně fungovala síť, musí být v doméně dostupné stejné síťové moduly. To zajistíte příkazem:

```
cp -a /lib/modules/$(rpm -qf --qf %{VERSION}-%{RELEASE}-xen \  
/boot/vmlinuz-xen) /var/tmp/dirinstall/lib/modules
```

Abyste předešli systémovým chybám, musí být po instalaci odpojen obraz souborového systému:

```
umount /var/tmp/dirinstall/proc  
umount /var/tmp/dirinstall/
```

Je možné vytvořit specializovaná jádra pro domain-0 a další domény. Hlavní rozdíl by představovaly ovladače zařízení, které hostující systémy obvykle nepoužívají. SUSE obsahuje pouze jediné jádro, protože ovladače jsou modulární a zavádějí se jen tehdy, pokud je jich potřeba.

## 17.2.2 Nastavení záchranného systému jako domény

Nejrychlejší způsob spuštění systému je použít již existující kořenový systém, např. záchranný systém systému SUSE Linux. Změňte pro tento obraz jádra ovladače blokových zařízení a síťových zařízení. Aby to bylo co nejjednodušší, použijte skript `mk-xen-rescue-img.sh` ze souboru `/usr/share/doc/packages/xen/`.

Nevýhodou tohoto řešení je neexistence RPM databáze v takto vzniklém systému, takže nebudete moci snadno přidávat nové balíčky. Na druhou stranu jde o relativně malý systém, který obsahuje vše potřebné pro připojení do sítě.

Před spuštěním skriptu `mk-xen-rescue-img.sh` musíte mít připravený adresář s obrazem jádra záchranného systému a adresář, kam se má uložit výsledný obraz. Na startovacím médiu najdete obraz v adresáři `/boot`. Pro DVD tedy provedete následující příkaz:

```
cd /usr/share/doc/packages/xen
./mk-xen-rescue-img.sh /media/dvd/boot /usr/local/xen 64
```

První parametr je adresář obrazu, druhý jeho cílové umístění. Volitelné parametry jsou prosotorové požadavky nově generované domény a verze jádra, která se má použít.

Skript překopíruje obraz do nového umístění, nahradí moduly jádra a zakáže v systému `tls`. Posledním krokem je vytvoření konfiguračního souboru pro nově vytvořený obraz v adresáři `/etc/xen/`.

## 17.3 Konfigurace domény Xenu

Dokumentace o konfiguraci nové domény je poměrně stručná. Nejvíce užitečných informací najdete v příkladu konfiguračního souboru `/etc/xen/config`. Okomentovány jsou v něm všechny potřebné volby včetně výchozích hodnot. Pro instalaci popsanou

v 17.2.1 – „Instalace domény do adresáře pomocí programu YaST“ (strana 270) vytvořte soubor `/etc/xen/guest1` s následujícím obsahem:

```
kernel = "/boot/vmlinuz-xen" ❶
ramdisk = "/boot/initrd-xen" ❷
memory = 128 ❸
name = "guest1" ❹
nics = "1" ❺
vif = [ 'mac=aa:cc:00:00:00:ab, bridge=xen-br0' ] ❻
disk = [ 'file:/var/tmp/guest1,hda1,w' ] ❼
root = "/dev/hda1 ro" ❽
extra = "3" ❾
```

- ❶ Zadejte cestu k jádru Xenu pro domain-0.
- ❷ Zvolte správný ramdisk pro jádro Xenu obsahující potřebné ovladače. Bez ramdisku obvykle dojde ke `kernel panic`, protože nebude možné připojit kořenový systém.
- ❸ Nastavte pro nový systém velikost paměti. `guests`.
- ❹ Zadejte jméno systému.
- ❺ Zadejte čísla virtuálních síťových rozhraní nové domény.
- ❻ Nastavte virtuální síťová rozhraní.
- ❼ Nastavte dostupná bloková zařízení. Abyste mohli používat skutečná zařízení, vytvořte řádku podobnou této: `[ 'phy:sdb1,hda1,w', 'phy:system/swap1,hda2,w' ]`.
- ❽ Nastavte kořenové zařízení jádra. Musí jít o virtuální zařízení viditelné novým systémem.
- ❾ Zadejte potřebné parametry jádra. 3 například znamená, že se nový systém spustí do úrovně 3.

## 17.4 Spuštění a správa Xen domén

Před spuštěním nové domény musíte hypervisoru Xenu zajistit dostatek paměti pro nový systém. nejdřív přezkontrolujte volnou paměť:

```
xm list
Name           Id  Mem(MB)  CPU  State  Time(s)  Console
Domain-0       0    458      0  r----  181.8
```



Pokud je v příkladu počítač s pamětí o velikosti 512 MB, Xen hypervisor zabere 64 MB a Domain-0 zbytek. Pokud potřebujete paměť nastavit přesně, použijte příkaz `xm balloon`, např. chcete-li pro Domain-0 použít 330 MB, zadejte jako `root` příkaz:

```
xm balloon 0 330
```

V následujícím `xm list` bude paměť pro Domain-0 omezena na 330 MB. Nyní máte dostatek paměti pro systém s 128 MB. Příkazem `xm start guest1 -c` spustíte nový systém a propojíte ho s aktuálním terminálem. Pokud jde o první spuštění, dokončete instalaci pomocí programu YaST.

Nastavení konzole nebo terminálu můžete kdykoliv změnit. K odpojení použijte `Ctrl` + `]`. K přepojení si nejdříve zjistíte ID systému, který chcete přepojit příkazem `xm list` a k přepojení použijte příkaz `xm console ID`.

Nástroj `xm` má řadu voleb. Jejich seznam s krátkým vysvětlením získáte příkazem `xm help`. V tabulce 17.1 – „Příkazy `xm`“ (strana 273) najdete nejčastěji používané soubory.

**Tabulka 17.1** Příkazy `xm`

---

<code>xm help</code>	Vypíše seznam dostupných příkazů nástroje <code>xm</code> .
<code>xm console ID</code>	Připojí první konzoli ( <code>tty1</code> ) hosta s ID <code>ID</code> .
<code>xm balloon ID Mem</code>	Nastaví velikost paměti domény s ID <code>ID</code> na <code>Mem</code> (MB).
<code>xm create domname [-c]</code>	Spustí doménu s konfiguračním souborem <code>domname</code> . Volitelný parametr <code>-c</code> spojí aktuální terminál s první konzolí hosta.
<code>xm shutdown ID</code>	Běžným způsobem ukončí běh hosta s ID <code>ID</code> .
<code>xm destroy ID</code>	Okamžitě ukončí běh hosta s ID <code>ID</code> .
<code>xm list</code>	Vypíše seznam všech běžících domén včetně jejich ID, množství paměti a času procesoru.
<code>xm info</code>	Zobrazí informace o Xen hostiteli, včetně informací o procesoru a paměti.

---

## 17.5 Více informací

Více informací o Xenu naleznete na následujících stránkách (v angličtině):

- <file:///usr/share/doc/packages/xen/user/html/index.html>
  - oficiální informace pro uživatele Xenu. Vyžaduje instalaci balíčku `xen-doc-html`.
- <file:///usr/share/doc/packages/xen/interface/html/index.html>
  - technická dokumentace rozhraní. Vyžaduje instalaci balíčku `xen-doc-html`.
- <http://www.cl.cam.ac.uk/Research/SRG/netos/xen/index.html>
  - domácí stránka Xenu s řadou odkazů na dokumentaci.
- <http://lists.xensource.com/>
  - několik poštovních konferencí o Xenu.

## **Část 4. Služby**



## Základy síťování

Linux je dítě Internetu. Nabízí proto samozřejmě všechny potřebné funkce pro integraci do všech typů sítí. Linuxový protokol TCP/IP má řadu funkcí a poskytuje řadu služeb, které zde popisujeme. Přístup k síti pomocí síťové karty, modemu nebo jiného zařízení lze nakonfigurovat nástrojem YaST. Je možná i manuální konfigurace. V této kapitole jsou popsány pouze základní síťové mechanismy a konfigurace.

Linux a jiné unixové operační systémy používají především tzv. TCP/IP protokol. V tomto případě se nejedná o jeden, ale o celou skupinu síťových protokolů, která poskytuje různé služby. Protokoly uvedené v tabulce 18.1 – „Různé protokoly z rodiny TCP/IP“ (strana 277) slouží k výměně dat mezi dvěma stroji přes TCP/IP. TCP/IP síť tvoří navzájem provázanou celosvětovou síť známou pod jménem Internet.

RFC dokumenty (Request for comments) popisují různé internetové protokoly a související procedury operačního systému a aplikací. Pokud si tedy chcete prohloubit své znalosti o určitém protokolu, pak je pro vás odpovídající RFC dokument to pravé. RFC naleznete na internetové adrese <http://www.ietf.org/rfc.html>

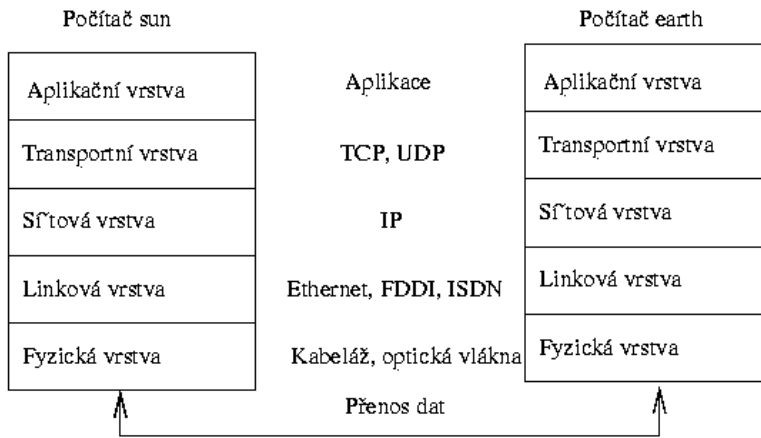
**Tabulka 18.1** Různé protokoly z rodiny TCP/IP

Protokol	Popis
TCP	(angl. <i>Transmission Control Protocol</i> ) Spojovací zabezpečený protokol. Přenášená data jsou aplikací odesílána jako datový tok a samotný operační systém je upravuje do formátu vhodného pro přenos. Data pak přichází cílové aplikaci opět ve formě datového toku tak, jak byla

Protokol	Popis
	odeslána. TCP zajišťuje, že se po cestě žádná data neztratí. TCP se používá tam, kde je důležité pořadí dat.
UDP	(angl. <i>User Datagram Protocol</i> ) Nezabezpečený protokol. Data jsou odesílána ve formě paketů. Není garantováno pořadí příchodu dat příjemci a stejně tak se může stát, že se některé pakety ztratí. UDP se hodí pro datově orientované aplikace (např. přenos multimédií) a nemá žádné prodlevy způsobené ověřováním tak, jak je tomu u TCP.
ICMP	(angl. <i>Internet Control Message Protocol</i> ) Jedná se o servisní protokol, který sděluje stav chyb a řídí chování počítačů při přenosu TCP/IP dat. Navíc podporuje ICMP echo režim, který používá program ping.
IGMP	(angl. <i>Internet Group Management Protocol</i> ) Tento protokol řídí chování počítačů při IP multicast. Naneštěstí IP multicast přesahuje rozsah této publikace.

Jak je vidět v tabulce [18.1 – „Zjednodušený model vrstev TCP/IP“](#) (strana 279), výměna dat probíhá v několika vrstvách. Vlastní síťová vrstva představuje nezabezpečený přenos dat pomocí IP (angl. *Internet Protocol*). Nad IP je TCP (angl. *Transmission Control Protocol*), který, do jisté míry, zajišťuje bezpečnost přenášených dat. IP sám je zase nadstavbou hardwarového protokolu, např. Ethernetu.

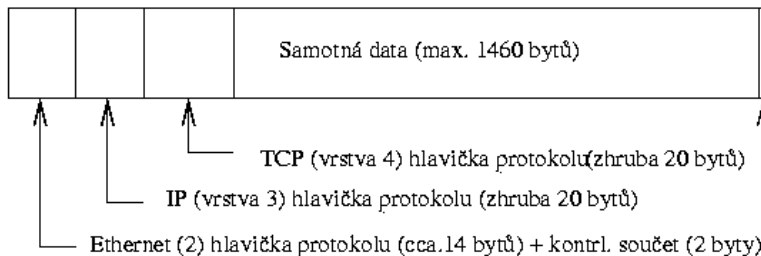
**Obrázek 18.1** Zjednodušený model vrstev TCP/IP



Takřka všechny hardwarové protokoly jsou paketově orientovány. Je tedy třeba přenášená data zabalit do malých paketů a není možné posílat vše v jednom. Proto také TCP/IP pracuje s menšími datovými jednotkami. Maximální velikost jednoho TCP/IP paketu je skoro 64 KB (kilobytů). Obvykle jsou tyto pakety značně menší, protože limitujícím faktorem je síťový hardware. Takže např. maximální velikost datových paketů v Ethernetu je zhruba 1500 bytů. Tomu také odpovídá velikost TCP/IP paketů, pokud jsou data posílána přes Ethernet. Pokud posíláte větší objem dat, musí je operační systém rozdělit do více paketů a ty pak poslat.

Aby mohla každá vrstva plnit přidělenou funkci, musí přidat doplňující informace do paketu. Ty jsou uloženy v *hlavičce* paketu. Každá vrstva připojí malý blok dat, tzv. hlavičku protokolu (angl. *protocol header*). Paket v ethernetové síti může vypadat jako na obrázku 18.2 – „TCP/IP paket v Ethernetu“ (strana 279). Kontrolní součet je umístěn na konci paketu, ne na začátku. To usnadňuje život hardwaru.

**Obrázek 18.2** TCP/IP paket v Ethernetu



Pokud chce nějaká aplikace posílat data přes síť, pak proběhnou data jednotlivými vrstvami, které jsou (s výjimkou hardwarové vrstvy) implementovány do linuxového jádra. Každá z vrstev upraví data tak, aby mohla být předána níže položené vrstvě. Nejnižší vrstva je pak zodpovědná za poslání dat. Při příjmu dat probíhá to samé, ale v opačném gardu. Paket je zde loupán jako cibule a v každé vrstvě jsou odstraňovány hlavičky protokolu. Čtvrtá vrstva pak připravuje data pro aplikaci na cílovém počítači. Přitom komunikuje každá vrstva pouze s vrstvou přímo nad, resp. pod ní. Aplikace se tedy nemusí starat o to, zda data půjdou přes 100 MB FDDI síť nebo 56 kbit vytáčenou linku. Stejně tak je např. transportní vrstvě jedno, zda jsou posílaná data správně zabalená.

## 18.1 IP adresy a směrování

Následující část je věnována protokolu IPv4. Informace o IPv6 naleznete v části [18.2 – „IPv6 – Internet další generace“](#) (strana 283).

### 18.1.1 IP adresa

Každý počítač v internetové síti má jednoznačnou 32bitovou (4 byty) adresu. Ta může vypadat jako v příkladu [18.1 – „Zápis IP adres“](#) (strana 280)

#### **Rovnice 18.1** Zápis IP adres

```
IP adresa (binárn ):  11000000 10101000 00000000 00010100
IP adresa (decimáln ):  192.    168.    0.    20
```

Tyto čtyři byty jsou v desítkové soustavě odděleny tečkou. IP adresa je přiřazena každému počítači, resp. každému síťovému rozhraní, takže už nemůže být použita v jakémkoliv jiném počítači na celém světě. Sice existují výjimky z tohoto pravidla, ale zde nehrají žádnou roli.

Také Ethernetové karty obsahují jednoznačnou adresu, tzv. *MAC* (angl. *Media Access Control*). Ta je 48 bitů dlouhá, celosvětově jedinečná a je výrobcem kartě jednoznačně přidělena. Má ale jeden obrovský nedostatek. MAC adresy netvoří hierarchický systém, ale jsou přidělovány víceméně náhodně. Není je proto možné používat pro adresování vzdálených počítačů. Rozhodující úlohu ale tyto adresy hrají při komunikaci počítačů v lokální síti (a jsou součástí hlavičky paketů pro druhou vrstvu).



A nyní zpět k IP adresám. Jak již napovídá výše uvedený text, tvoří IP adresy hierarchický systém. Do poloviny devadesátých let byly IP adresy pevně členěny do jednotlivých tříd. Tento systém se ukázal jako neflexibilní a proto se přestal používat. Používá se pouze směrování bez tříd (CIDR – Classless Inter Domain Routing).

## 18.1.2 Síťové masky a směrování

Protože počítač s IP adresou 192.168.0.0 nemůže vědět, kde se nachází počítač s IP adresou 192.168.0.20, byly zavedeny síťové masky. Zjednodušeně řečeno síťové masky sdělují počítači s IP adresou, co je uvnitř a co vně. Počítače, které se nacházejí uvnitř (ve stejné části počítačové sítě) spolu mohou komunikovat přímo. Při přístupu k počítačům nacházejícím se vně je třeba použít tzv. bránu (angl. *gateway*) nebo router. Protože má každé síťové rozhraní svou IP adresu, může to být poměrně komplikované.

Předtím, než se paket vydá na svou cestu, proběhne v počítači následující proces. Cílová adresa je se síťovou maskou binárně spojena pomocí operátoru AND. Také adresa odesílatele je spojena se síťovou maskou pomocí operátoru AND. Pokud je k dispozici více síťových rozhraní, pak jsou zpravidla ověřeny všechny adresy odesílatele. Výsledky spojení adres (AND) jsou pak porovnány. Pokud jsou tyto výsledky zcela shodné, nachází se cílový počítač ve stejné části sítě. V opačném případě je třeba použít bránu. To znamená, že čím více 1 bitů se nachází v síťové masce, tím méně počítačů je přímo dostupných. V následující tabulce je uvedeno několik příkladů:

### **Rovnice 18.2** Spojování IP adres se síťovou maskou

IP adresa	(192.168.0.20):	11000000	10101000	00000000	00010100
síťová maska	(255.255.255.0):	11111111	11111111	11111111	00000000
-----					
výsledek	(binární):	11000000	10101000	00000000	00000000
výsledek	(decimální):	192.	168.	0.	0
IP adresa	(213.95.15.200):	11010101	10111111	00001111	11001000
síťová maska	(255.255.255.0):	11111111	11111111	11111111	00000000
-----					
výsledek	(binární):	11010101	10111111	00001111	00000000
výsledek	(decimální):	213.	95.	15.	0

Síťová maska se zapisuje, tak jako IP adresa, ve formě decimálních čísel oddělených tečkami. Protože má síťová maska také velikost 32 bitů, jsou jednotlivá čísla psána za sebe. Které počítače jsou bránou nebo které oblasti adres jsou přístupné přes které síťové rozhraní, je třeba nakonfigurovat.

A následuje další příklad – všechny počítače připojené na jeden ethernetový kabel se nacházejí *ve stejné části sítě* a jsou přímo přístupné. I když je v Ethernetu rozdělují tzv. switche a bridge, je možné k počítačům přistupovat přímo.

Pokud chcete překlenout delší vzdálenost, není již možné použít Ethernet. Pak je třeba IP pakety převést na jiný hardware (např. FDDI nebo ISDN). Taková zařízení se nazývají routery, resp. brány. Linuxový počítač může plnit i tyto úlohy, tato volba se označuje jako `ip_forwarding`.

Pokud je nakonfigurována brána, je paket poslán na odpovídající gateway. Ta se pak pokusí paket přeposlat dále. To se opakuje na každém dalším počítači tak dlouho, než paket dosáhne cílový počítač nebo vyprší jeho *životnost* TTL (angl. *time to live*).

### **Tabulka 18.2** *Vyhrazené adresní prostory*

<b>Adresa</b>	<b>Popis</b>
Základní síťová adresa	Síťová maska spojená (AND) s libovolnou adresou v síti, tedy výsledek z tabulky 18.2 – „ <a href="#">Spojování IP adres se síťovou maskou</a> “ (strana 281). Tuto adresu nelze přiřadit žádnému počítači.
Oznamovací adresa	Ta říká: hovoř se všemi počítači v této části sítě. Získá se binární inverzí síťové masky a spojením výsledku se základní síťovou adresou pomocí operace OR. Náš příklad vede k výsledku 192.168.0.255. Ani tato adresa nemůže být přiřazena žádnému počítači.
Lokální počítač	Adresa 127.0.0.1 odkazuje na každém počítači na tzv. loopback device. Pomocí této adresy je možné navázat spojení s vlastním počítačem.

Protože je třeba, aby byly IP adresy jedinečné, nemůžete si zvolit libovolné adresy. Abyste i přesto mohli postavit síť na bázi IP adres, existují tři oblasti, které můžete ihned použít. S těmito adresami se ale bez překladu adres nemůžete připojit k Internetu. Tyto adresové oblasti jsou definovány v RFC 1597 a jejich seznam si můžete prohlédnout v tabulce 18.3 – „[Neveřejné adresní rozsahy](#)“ (strana 283).

**Tabulka 18.3** Neveřejné adresní rozsahy

sít' / síťová maska	oblast
10.0.0.0 / 255.0.0.0	10.x.x.x
172.16.0.0 / 255.240.0.0	172.16.x.x - 172.31.x.x
192.168.0.0 / 255.255.0.0	192.168.x.x

## 18.2 IPv6 – Internet další generace

Díky vynálezu WWW začal Internet, a tím i počet počítačů komunikujících pomocí TCP/IP, v posledních patnácti letech exponenciálně růst. Podle informací CERN (<http://public.web.cern.ch/>) vzrostl jejich počet z několika tisíc v roce 1990 na zhruba 100 000 000 v současnosti.

Jak již víte, má IP adresa pouze 32 bitů. Protože není z organizačních důvodů možné používat mnoho adres z 32 bitového adresního prostoru, je počet adres již nedostačující. Pouze pro připomenutí - Internet se skládá z podsítí, které jsou dále členěny. Ty se skládají vždy z mocniny dvou mínus 2 použitelných adres. Pokud tedy chcete připojit k Internetu 128 počítačů, pak potřebujete podsít' s 256 síťovými adresami, ze kterých můžete použít pouze 254 adres. Dvě adresy není možné použít, protože jedna je broadcast a druhá základní adresa sítě.

Aby se maximálně využívaly současné adresy v IPv4, používá se DHCP nebo NAT (angl. *Network Address Translation*). Tyto nástroje, spolu s veřejnými a neveřejnými adresními prostory, částečně řeší nedostatek adres. Nevýhodou těchto metod je náročnější konfigurace, protože pro korektní nastavení počítače v IPv4 sítích potřebujete množství informací, jako je vlastní IP adresa, síťová maska, adresa brány a podle potřeby také nameserver. Všechny tyto informace musíte *vědět*.

S IPv6 je omezený adresní prostor a komplikovaná konfigurace minulostí. V následujících odstavcích si přiblížíme základní přednosti IPv6 a způsob přechodu od starého k novému protokolu.

## 18.2.1 Přednosti IPv6

Největší výhodou nového protokolu je enormní rozšíření adresního prostoru, protože IPv6 obsahuje místo 32bitových adres 128bitové adresy.

IPv6 adresy se neliší od svých předchůdců pouze délkou, ale také vnitřní strukturou, která obsahuje informace o systému a síti. Více v části [18.2.2 – „Adresování v IPv6“](#) (strana 285).

Dalšími důležitými přednostmi nového protokolu jsou:

### Automatická konfigurace

IPv6 zavádí v síťování princip *Plug and Play*, protože nový systém se do lokální sítě integruje bez nutnosti manuální konfigurace. Autokonfigurační mechanismus zjistí vlastní adresu z informací, které obdrží prostřednictvím ND (*Neighbor Discovery*) protokolu ze sousedních routerů. Tento proces nevyžaduje žádný zásah ze strany správce sítě a oproti DHCP v IPv4 sítích má tu výhodu, že není nutné udržovat centrální server.

### Mobilita

IPv6 umožňuje, aby jednomu síťovému rozhraní bylo přiděleno více adres. Tím pádem budete mít jako uživatel systému jednoduše přístup k různým sítím. Tuto funkci je možné porovnat s roamingem u mobilních telefonů. Pokud se nacházíte se svým mobilem v zahraničí, připojí se telefon automaticky k cizí síti. Je zcela jedno, kde jste. Máte zaručenou dostupnost prostřednictvím běžného telefonního čísla a můžete telefonovat v cizích sítích, jako by to byly domovské sítě.

### Bezpečná komunikace

Zatímco v IPv4 patří zabezpečení komunikace pouze mezi doplňkové funkce, obsahuje IPv6 IPsec pro bezpečnou komunikaci.

### Zpětná kompatibilita

Rychlý přechod celého Internetu na IPv6 není realistický. Proto je důležité, že obě verze mohou koexistovat v jednom systému. Koexistence obou je možná díky používání kompatibilních adres (IPv4 lze převést na IPv6). Je také možné použít různé tunely (viz část [18.2.3 – „IPv4 versus IPv6 – cestování mezi světy“](#) (strana 289)). Prostřednictvím tzv. *Dual-Stack-IP* je možná podpora obou protokolů na jednom systému. Každý z obou protokolů používá vlastní síťový stack, takže nikdy nedojde ke kolizi.

## Multicasting

Zatímco v IPv4 sítích posílají některé služby (např. SMB) své pakety prostřednictvím všesměrového vysílání všem počítačům v lokální síti, je v IPv6 dostupný zcela jiný způsob. Pomocí multicastu je možné komunikovat se skupinou počítačů, tedy ne nutně se všemi jako v případě broadcast. Která skupina to bude, záleží na aplikaci. Existují však i určité předdefinované skupiny, jako jsou *všechny nameservery* (angl. *all nameservers multicast group*) nebo *všechny routery* (angl. *all routers multicast group*).

## 18.2.2 Adresování v IPv6

Jak již bylo uvedeno, má současný IP protokol dvě výrazné nevýhody. První je blížící se nedostatek IP adres a druhým složitá správa routování, jejíž složitost stále narůstá. První problém odstraňuje IPv6 rozšířením adresního prostoru na 128 bitů. Řešení druhého problému leží v hierarchické adresní kultuře, sofistikovaných mechanismech pro přiřazování adresy v síti a možnosti používání více adres pro jedno rozhraní, které zajišťuje přístup do různých sítí (tzv. multihoming).

Existují tři důležité typy IPv6 adres:

### Unicast

Adresy tohoto typu patří právě jednomu síťovému rozhraní. Pakety s adresou tohoto typu jsou směrovány přímo na příjemce. Unicast adresy se používají pro komunikaci s jednotlivými počítači v lokální síti nebo Internetu.

### Multicast

Adresy tohoto typu odkazují na skupinu rozhraní. Pakety s touto adresou jsou doručeny všem členům skupiny. Multicast používají především různé síťové služby, aby komunikovaly s určitou skupinou počítačů.

### Anycast

Adresy tohoto typu odkazují na skupinu rozhraní. Pakety s adresou tohoto typu jsou odeslány členu skupiny, který je podle směrovacích protokolů nejbližší odesílateli. Anycast adresy se používají v případě, kdy je vyhledáván server poskytující určité síťové služby. Všechny servery určitého typu obdrží stejnou anycast adresu. Pokud tedy terminál vyžaduje službu, odpoví ten server, který je podle směrovacího protokolu počítače nejbližší. Pokud tento server neodpovídá, je kontaktován další nejbližší.

IPv6 adresa sestává z osmi bloků po 16ti bitech, které jsou odděleny dvojtečkou a jsou v hexadecimálním zápise. Počáteční nulové byty (v rámci bloku) je možné vypustit, uprostřed nebo na konci musí být zachovány. Více než čtyři nulové byty za sebou je možné nahradit :: (tzv. *collapsing*). V každé adrese je však možné :: použít maximálně jednou. Příklad 18.3 – „Sample IPv6 Address“ (strana 286) obsahuje tři různé ekvivalentní zápisy.

**Rovnice 18.3** *Sample IPv6 Address*

```
fe80 : 0000 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4
fe80 :    0 :    0 :    0 :    0 : 10 : 1000 : 1a4
fe80 :                               : 10 : 1000 : 1a4
```

Každá část IPv6 adresy má definovaný význam. První byty tvoří prefix a vypovídají o typu adresy. Prostřední část adresuje síť nebo je bez významu. Konec adresy tvoří tzv. host část. Síťová maska se určuje v IPv6 délkou prefixu a zapisuje se za lomítko na konci adresy. Adresa zobrazená v příkladu 18.4 – „IPv6 adresa s vyznačenou délkou prefixu“ (strana 286) obsahuje informaci, že prvních 64 bitů tvoří síťovou část adresy a posledních 64 bitů část týkající se počítače. Jinými slovy, 64 značí, že je síťová maska tvořena 64 1-bitovými hodnotami z levé části. Stejně jako v případě IPv4 je IP adresa kombinována pomocí AND s hodnotami síťové masky, aby se zjistilo, zda jsou počítače ve stejné části sítě.

**Rovnice 18.4** *IPv6 adresa s vyznačenou délkou prefixu*

```
fe80::10:1000:1a4/64
```

IPv6 rozpoznává různé prefixy s definovaným významem (viz tabulka 18.4 – „Různé IPv6 prefixy“ (strana 286)).

**Tabulka 18.4** *Různé IPv6 prefixy*

Prefix (hexadecimálně.)	Definice
00	IPv4 adresy a IPv4 over IPv6 adresy. Jedná se o adresy zpětně kompatibilní s IPv4. Vhodný router musí ještě převést IPv6 paket na IPv4. Tento prefix používají i další speciální adresy, jako je loopback smyčka.
První číslice 2 nebo 3	<i>Aggregatable Global Unicast Address</i> – Stejně jako IPv4 lze síť IPv6 dělit na jednotlivé části. Aktuálně je možné použít

Prefix (hexadecimální.)	Definice
	následující adresní prostory: 2001::/16 ( <i>production quality address space</i> ) a 2002::/16 ( <i>6to4 address space</i> ).
fe80::/10	Tzv. <i>link-local</i> adresy. Adresy s tímto prefixem není možné routovat a jsou dostupné pouze v rámci podsítě.
fec0::/10	Tzv. <i>site-local</i> adresy. Tyto adresy je sice možné směřovat, ale pouze v rámci organizace. Tím tedy odpovídají tyto adresy současným <i>privátním</i> adresním prostorům (např. 10.x.x.x).
ff	<i>Multicast</i> IPv6 adresy.

Unicast adresy jsou vystavěny ze tří stupňů:

#### Public Topology

První část (která obsahuje také výše uvedený prefix) slouží pro směřování paketů v prostředí Internetu. Zde jsou obsaženy informace o poskytovateli nebo instituci, která zajišťuje připojení k Internetu.

#### Site Topology

Druhá část obsahuje směrovací informace o podsíti, ke které paket náleží.

#### Interface ID

Třetí díl pak jednoznačně určuje rozhraní, pro které je paket určen. To umožňuje použít MAC adresy jako součást adresy. Protože jsou celosvětově jedinečné a pevně přidělené výrobcem hardwaru, znamená to velké zjednodušení konfigurace. Ve skutečnosti se prvních 64 bitů skládá z tzv. EUI-64 tokenu, kde se odejme posledních 48 bitů MAC adresy a zbylých 24 bitů tvoří speciální informace, které vypovídají o typu tokenu. To také umožňuje přiřadit EUI-64 token zařízením bez MAC adresy, jako jsou PPP a ISDN spojení.

Na základě této struktury existuje 5 různých typů IPv6 unicast adres:

#### :: (unspecified)

Tuto adresu používá počítač jako zdrojovou adresu, když poprvé inicializuje síťové rozhraní a nemá ještě žádné informace o vlastní adrese.

: : 1 (loopback)

Adresa pro smyčku loopback.

Adresy kompatibilní s IPv4

IPv6 adresa sestává z IPv4 adresy a 96-bitového prefixu samých nul. Tento typ kompatibilních adres se používá při tunelování (viz odst. 18.2.3 – „IPv4 versus IPv6 – cestování mezi světy“ (strana 289)). IPv4/IPv6 počítače tak mohou komunikovat s ostatními počítači, které se nacházejí v čistě IPv4 síti.

IPv4 adresy mapované na IPv6

Tento typ specifikuje čistě IPv4 adresy v IPv6 zápisu.

Lokální adresy

Existují dva typy adres pro lokální používání:

link-local

Tento typ adres je vyhrazen pouze pro používání v lokálních částech sítě. Routery nesmí předávat pakety s touto zdrojovou nebo cílovou adresou do Internetu nebo jiné části sítě. Tyto adresy jsou označeny speciálním prefixem ( $f\epsilon 80 : : / 10$ ) a ID rozhraní síťové karty. Střední část adresy obsahuje nulové byty. Tento druh adres se používá autokonfiguračními programy, které komunikují s počítači ve stejném segmentu sítě.

site-local

Tento typ adres je možné směřovat mezi jednotlivými podsítěmi, ale pouze v rámci sítě, nesmí se použít v rámci Internetu. Takové adresy se používají pro intranet a jsou ekvivalentem pro privátní adresy v IPv4. Kromě definovaného prefixu ( $f\epsilon c0 : : / 10$ ) a ID rozhraní obsahují tyto adresy 16-bitové pole s informacemi o ID segmentu sítě. Zbytek je vyplněn nulovými byty.

Navíc obsahuje IPv6 další vynález a to možnost přiřadit jednomu síťovému rozhraní více síťových adres. To má tu výhodu, že je k dispozici více sítí. Jedna z nich může být nakonfigurována zcela automaticky pomocí MAC adresy a známého prefixu, výsledkem je dosažitelnost všech počítačů v IPv6 síti (pomocí link-local adresy) okamžitě po jejím zprovoznění. Pokud je součástí IP adresy MAC adresa, jsou jednotlivé IP adresy celosvětově unikátní. Jediné variabilní části adresy jsou ty, které určují topologii (*site topology* a *public topology*) v závislosti na síti, ve které se počítač právě nachází.

Pokud se počítač pohybuje mezi jednotlivými sítěmi, potřebuje minimálně dvě adresy. Jedna je jeho domovská adresa skládající se z ID rozhraní, informací o domovské síti a odpovídajícího prefixu. Domovská adresa je statická a neměnná. Všechny pakety,



kteřé jsou určeny pro tento počítač, mu budou doručeny, ať se fyzicky nachází kdekoli. To umožňují zcela nové funkce IPv6, tzv. *Stateless Autoconfiguration* a *Neighbor Discovery*. Přenosný počítač může tedy mít kromě domovské adresy jednu nebo více adres, které patří sítím, ve kterých se počítač právě nachází. Těmto adresám se říká *Care-of-Address*. V domácí síti mobilního počítače musí existovat instance, která bude komunikaci směrovanou na jeho domovskou adresu dále přeposílat, pokud se nalézá v jiné síti. Tuto funkci přebírá v IPv6 tzv. *Home Agent*. Ten pak vytvoří tunel, kterým posílá pakety. Pakety, které mají jako cílovou *Care-of-Address*, mohou putovat bez okliky přes Home agenta.

## 18.2.3 IPv4 versus IPv6 – cestování mezi světy

Přechod všech počítačů připojených k Internetu z IPv4 na IPv6 není možné provést okamžitě, spíš je pravděpodobné, že starý a nový protokol budou koexistovat dlouhou dobu. Sdílení na jednom počítači je řešeno pomocí *Dual Stack*, zůstává ale otázkou, jak bude komunikovat IPv6 počítač s IPv4 počítačem a jak přenášet IPv6 přes stávající IPv4 síť. Odpovědí na tyto otázky je tzv. tunelování a používání kompatibilních adres (viz 18.2.2 – „Adresování v IPv6“ (strana 285)).

Jednotlivé ostrůvky IPv6 v moři IPv4 sítí si vyměňují svá data pomocí tunelů. Při tunelování jsou IPv6 pakety zabaleny do IPv4 paketů, aby je bylo možné přenášet v IPv4 sítích. Tunel je definován jako spojení mezi dvěma IPv4 konci. Pakety musí obsahovat IPv6 cílovou adresu (nebo odpovídající prefix) a IPv4 adresu počítače na konci tunelu. V jednoduchých případech se konfiguruji takové tunely ručně a říká se jim *statické*.

Pokud není ruční vytváření tunelů reálné kvůli jejich vysokému počtu, existují tři různé způsoby pro vytváření *dynamických tunelů*:

### 6over4

IPv6 pakety jsou automaticky zabaleny do IPv4 paketů a posílány přes IPv4 síť, kde je aktivován multicasting. IPv6 se tedy zdá, že celý Internet je pouze velká LAN. Nevýhodou tohoto řešení je špatná škálovatelnost a také skutečnost, že IP multicasting není dostupný v celém Internetu. Toto řešení se hodí pro malé firmy a organizace, které mají možnost provádět IP multicasting. Více informací naleznete v RFC 2529.

6to4

Zde jsou IPv4 adresy automaticky generovány z IPv6 adres. Tak mohou jednotlivé ostrůvky IPv6 komunikovat prostřednictvím IPv4. Problém ale nastává při komunikaci s čistě IPv4 počítači. Více viz RFC 3056.

IPv6 Tunnel Broker

Tento postup se používá pro speciální servery, které vytvářejí uživatelům tunely automaticky a je popsán v RFC 3053.

---

### Důležité: Iniciativa 6Bone

Uprostřed starobylého Internetu existuje *6Bone* ([www.6bone.net](http://www.6bone.net)), což je celosvětová síť IPv6 podsítí, které jsou navzájem spojeny tunely. V rámci 6Bone sítě se testuje IPv6. Softwaroví vývojáři a poskytovatelé, kteří vyvíjí nebo poskytují IPv6 služby, mohou tyto segmenty použít pro testování, aby získali důležité zkušenosti s protokolem. Bližší informace naleznete na stránkách projektu 6Bone.

---

## 18.2.4 Konfigurace IPv6

Pokud chcete používat IPv6, není za běžných okolností třeba na pracovních stanicích provádět žádné změny. Musí však být zavedena podpora pro IPv6 v jádře. Jako uživatel `root` ji zavedete příkazem `modprobe ipv6`.

Protože se IPv6 z velké části konfiguruje samo, bude síťové kartě přiřazena adresa v *link-local* síti. Standardně není třeba mít na pracovní stanici směrovací tabulku. Pro směrování se používá *Router Advertisement Protocol*, pomocí kterého se pracovní stanice dotazují na prefix a brány, které mají být používány. K nastavení směrovače pro IPv6 slouží program `radvd`. Tento program pak sdělí pracovním stanicím prefixy pro IPv6 adresy a informace o směrování. Pro automatické nastavení adres a směrování lze také použít program `zebra`.

Informace o nastavení různých typů tunelů pomocí souborů `/etc/sysconfig/network` naleznete v manuálové stránce `ifup` (`man ifup`).

## 18.2.5 Další informace

Přehled v této kapitole neobsahoval všechny podrobnosti o IPv6. Pro hlubší studium můžete využít následující literaturu:

<http://www.ngnet.it/e/cosa-ipv6.php>

Série dokumentů, kde jsou velice dobře vysvětleny základy IPv6. Dobrý úvod do problematiky.

<http://www.bieringer.de/linux/IPv6/>

Dokument Linux-IPv6-HOWTO a mnoho odkazů.

<http://www.6bone.de/>

Připojení k IPv6 pomocí tunelů.

<http://www.ipv6.org/>

Vše o IPv6.

RFC 2640

Úvod do IPv6.

IPv6 Essentials

Kniha popisující všechny důležité aspekty IPv6. Silvia Hagen: *IPv6 Essentials*. O'Reilly & Associates, 2002 (ISBN 0-596-00125-8).

## 18.3 Překlad jmen

DNS se stará o to, abyste si nemuseli pamatovat žádné IP adresy. V Linuxu se o tento převod stará specializovaný software, který se nazývá bind. Počítač, na kterém se tento převod realizuje, je *nameserver* (jmenný server). Názvy tvoří také hierarchický systém, kde jsou jednotlivé části názvu oddělovány tečkou. Tato hierarchie je nezávislá na hierarchii IP adres.

Jako celé jméno můžeme použít např. `laurent.suse.de`. Jedná se o tzv. *fully qualified domain name (FQDN)*, plně kvalifikované doménové jméno. Je zapsáno ve formátu `název po íta e.doména`. Doména (v našem případě `suse.de`) obsahuje tzv. *TLD* (Top level domain) `de`.

Z historických důvodů je přiřazování TLD trochu zamotané. Proto jsou v USA používány domény první úrovně složené ze tří písmen, v ostatním světě pak národní ISO dvou písmenné domény. Od roku 2000 jsou k dispozici další TLD pro speciální oblasti, které se skládají i z více písmen (např. `.info`, `.name`, `.museum` atd.).

V kamenných dobách Internetu (před rokem 1990) se používal soubor `/etc/hosts`, kde byly uvedeny názvy všech počítačů, které existovaly na Internetu. To se ukázalo, při rychle rostoucím počtu připojených počítačů, jako nepraktické. Proto byla navržena distribuovaná databáze, která obsahuje názvy počítačů spolu s jejich IP adresami. Jelikož je databáze distribuovaná, nemusí znát všechny počítače, místo toho se zeptá jmeného serveru vyšší úrovně, zda náhodou počítač neznají. To ale neznamená, že nemůžete soubor použít pro překlad adres, např. v lokální podsíti.

Na vrcholu hierarchie nameserverů se nachází tzv. kořenový nameserver *root nameserver*. Tento nameserver spravuje top level domény a běží v tzv. *Network Information Centers*, zkráceně (NIC). Informace o českém správci domény naleznete na adrese <http://www.nic.cz>, případně obecnější informace na adrese <http://www.internic.net/>.

Pomocí DNS nemusíte převádět pouze názvy počítačů, DNS toho zvládne daleko více. Např. nameserver ví, který počítač přebírá pro celou doménu e-mailů, tzv. *Mail exchanger (MX)*.

Aby dokázal i váš počítač převádět IP adresy, musí mít přístup alespoň k jednomu nameserveru (a znát jeho IP adresu). Konfiguraci nameserveru můžete pohodlně provést pomocí YaST. Pokud používáte vytáčenou linku, pak se může stát, že nemusíte ručně konfigurovat žádný nameserver. Protokol používaný pro vytáčené linky vám poskytne adresu nameserveru při navazování spojení. Konfigurace přístupu k nameserveru je popsána v kapitole 20 – „DNS – Domain Name System“ (strana 323).

Těsně spojený s DNS je protokol *whois*. Se stejnojmenným programem *whois* máte možnost rychle zjistit, kdo je za určitou doménu odpovědný.

## 18.4 Konfigurace síťového připojení pomocí YaST

Počítač musí být vybaven podporovanou síťovou kartou. Většinou je síťová karta rozpoznána již při instalaci a je nahrán vhodný ovladač. Jestli je karta správně připojena,

zjistíte příkazem `ip address list eth0`. Pokud se zobrazí všechny informace o síťovém zařízení `eth0` a nikoliv chybové hlášení, je karta nainstalována správně.

Pokud máte jadernou podporu pro síť implementovanou jako modul, což je v jádře SUSE výchozí, musí být jméno modulu zadáno v souboru `/etc/sysconfig/hardware/hwcfg-*`. Pokud v něm není nic uvedeno, `hotplug` automaticky zvolí ovladač. `Hotplug` přiřadí ovladač pro vestavěnou i `hotplug` síťovou kartu.

## 18.4.1 Konfigurace síťové karty pomocí YaST

Po spuštění modulu zobrazí YaST obecný dialog pro nastavení sítě, ve kterém si můžete zvolit, zda chcete pro ovládání sítě použít `NetworkManager` nebo použít standardní konfiguraci pomocí `ifup`. Podrobnější informace o programu `NetworkManager` najdete v části 18.5 – „[Správa sítě s programem NetworkManager](#)“ (strana 303). Jestliže zvolíte standardní nastavení, v následujícím dialogu v horní části uvidíte seznam dosud nenakonfigurovaných síťových karet. Všechny správně automaticky rozeznané karty jsou v seznamu uvedené pod svým jménem. Nerozpoznaná zařízení jsou uvedena jako *Jiné (nerozpoznáno)*. Ve spodní části je zobrazen seznam již nakonfigurovaných zařízení spolu s typem sítě a adresou. Můžete nakonfigurovat novou kartu nebo změnit existující konfiguraci.

### Ruční konfigurace síťové karty

Konfigurace síťové karty, která nebyla automaticky rozpoznána (tedy je uvedena pod *Jiné (nerozpoznáno)*), sestává z následujících částí:

#### Konfigurace sítě

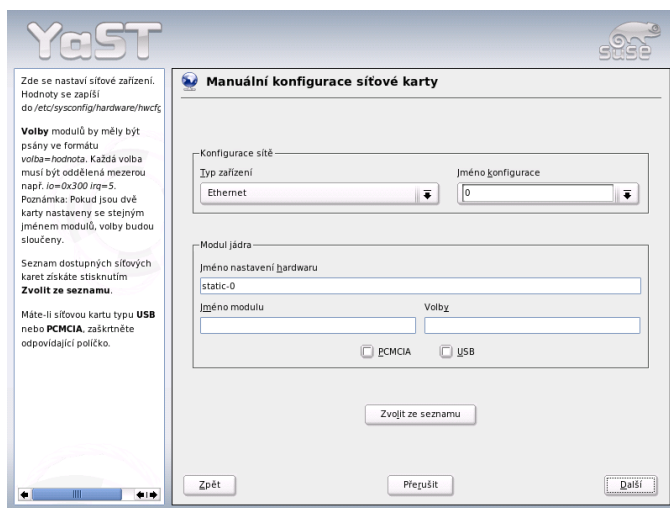
Nastavte typ zařízení rozhraní a jméno konfigurace. Typ zařízení vyberte z nabízených možností. Jméno konfigurace nastavte podle potřeby. Obvykle je možno použít výchozí hodnoty. Informace o konvencích používaných při pojmenování konfigurací naleznete v manuálové stránce `getcfg`.

#### Modul jádra

*Jméno nastavení hardwaru* specifikuje jméno souboru `/etc/sysconfig/hardware/hwcfg-*`, ve kterém je obsaženo hardwarové nastavení vaší síťové karty, např. jméno vhodného jaderného modulu. Pro PCMCIA a USB hardware obvykle YaST nabídne užitečná jména. Jméno nabízené pro ostatní hardware má obvykle smysl jen v případě, že je karta konfigurována pomocí `hwcfg-static-0`.

Pokud je síťová karta zařízení PCMCIA nebo USB, zaškrtněte příslušné políčko a opusťte dialog pomocí tlačítka *Další*. Pokud není, klikněte na *Zvolit ze seznamu* a vyberte správný typ karty. YaST automaticky vybere správný jaderný modul. Opusťte dialog pomocí tlačítka *Další*.

**Obrázek 18.3** Konfigurace síťové karty



## Nastavení síťové adresy

Vyberte z nabízených možností typ zařízení a jméno konfigurace podle svých potřeb. Obvykle lze použít výchozí hodnoty. V manuálové stránce `getcfg` naleznete informace o konvencích používaných při pojmenovávání konfigurací.

Pokud jste jako typ zařízení rozhraní vybrali *Bezdrátová technologie*, nastavte v následujícím dialogu (*Nastavení bezdrátové síťové karty*) operační režim, název sítě (ESSID) a údaje o šifrování. Kliknutím na *OK* konfiguraci dokončíte. Podrobný popis konfigurace WLAN karet naleznete v kapitole 34.1.3 – „Nastavení pomocí programu YaST“ (strana 515). V případě ostatních rozhraní pokračujte nastavením síťové adresy:

### *Automatické přidělení adresy (pomocí DHCP)*

Pokud na vaší síti běží DHCP server, můžete se na něj spolehnout a nechat nastavit síťovou adresu automaticky. Tato volba je vhodná také v případě, kdy jste připojeni přes DSL linku bez přidělené statické adresy. Pokud se rozhodnete použít DHCP, vyberte z nabídky *Rozšířené* položku *Nastavení DHCP klienta* a nastavte podrob-

nosti. Nastavte, zda má být požadována všesměrová odpověď a identifikátory, které se mají používat. Ve výchozím nastavení identifikují DHCP servery rozhraní podle hardwarové adresy síťové karty. Pokud ale různí virtuální klienti komunikují přes jedno rozhraní, je pro rozlišení nutné nastavit identifikátory.

### *Nastavení statické adresy*

Pokud máte statickou IP adresu, zaškrtněte příslušnou položku v dialogu a zadejte IP adresu a síťovou masku podsítě. Přednastavená maska by měla vyhovovat běžné domácí síti.

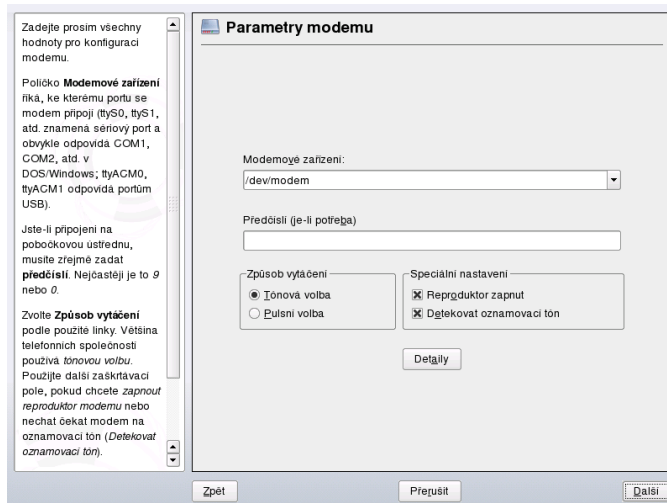
Dialog opusťte kliknutím na *Další* nebo pokračujte nastavením jména počítače, name-serveru a podrobností o směrování (viz části DNS a jméno počítače (↑Uživatelská příručka) a [18 – „Základy síťování“](#) (strana 277)).

*Rozšířené...* umožňuje nastavit podrobnosti. V položce *Detailní nastavení* zaškrtněte *Ovládání uživatelem*, pokud chcete, aby měl běžný uživatel kontrolu nad síťovou kartou (nikoliv pouze root). V případě mobilního použití to umožňuje uživateli flexibilně reagovat na změnu podmínek, neboť může sám aktivovat a deaktivovat rozhraní. Dále lze v tomto dialogu nastavit způsob *Aktivace zařízení* a MTU (Maximum Transmission Unit).

## **18.4.2 Modem**

V Řídícím středisku YaST, v sekci *Síťová zařízení*, zvolte modul *Modem*. Pokud nebyl modem rozpoznán automaticky, otevřete dialog pro ruční konfiguraci (*Konfigurovat...*) a v políčku *Modemové zařízení* zadejte rozhraní, ke kterému je modem připojen.

**Obrázek 18.4** Konfigurace modemu



Pokud jste připojeni přes pobočkovou ústřednu (PBX), může být nutné zadat volací předčísli. Obvykle je to nula. Podrobné informace naleznete v dokumentaci k vaší ústředně. Vyberte také, zda se má používat tónová nebo pulzní volba, zda má být zapnut reproduktor a zda má modem vyčkat, dokud nedetekuje oznamovací tón. Poslední z voleb by v případě připojení přes pobočkovou ústřednu neměla být zapnuta.

V dialogu, který se otevře po kliknutí na *Detaily*, nastavte přenosovou rychlost a inicializační řetězce pro modem. Nastavení měňte pouze tehdy, pokud modem nebyl automaticky rozpoznán nebo pokud vyžaduje pro funkci zvláštní nastavení. To obvykle nastává při použití ISDN terminálového adaptéru. Chcete-li umožnit kontrolu nad modemem (možnost aktivace a deaktivace) uživatelům bez pravomocí superuživatele, zaškrtněte *Ovládání uživatelem*. V položce *Regulární výraz vytáčeného předčísli* zadejte regulární výraz, kterému musí odpovídat hodnota zadaná uživatelem v položce *Vytáčené předčísli* programu KInternet. Pokud je pole pro regulární výraz ponecháno prázdné, uživatel bez administrátorských pravomocí nebude moci nastavit jiné předčísli. Dialog opusťte kliknutím na *OK*.

V dalším dialogu vyberte vašeho poskytovatele připojení k Internetu (ISP). Chcete-li poskytovatele vybrat z přednastaveného seznamu, vyberte položku *Země*. Druhou možností je kliknout na tlačítko *Nový* a zadat údaje o vašem poskytovateli ručně. Potřebné údaje zahrnují jméno poskytovatele, telefonní číslo a jméno a heslo, které vám



poskytovatel přidělil. Pokud chcete být před každým připojením dotazováni na heslo, zaškrtněte položku *Vždy se ptát na heslo*.

Poslední dialog umožňuje nastavit další volby pro spojení:

#### *Vytáčení na vyžádání*

Pokud povolíte vytáčení na vyžádání, nastavte alespoň jeden jmenný server (name-server).

#### *Modifikovat DNS po spojení*

Tato volba je implicitně zapnuta, což znamená, že je nameserver automaticky aktualizován při každém připojení na Internet.

#### *Automaticky obnovit DNS*

Pokud poskytovatel při navazování připojení nevysílá adresu jmenného serveru (DNS), zakažte *Automaticky obnovit DNS* a zadejte DNS ručně.

#### *Hloupý režim*

Hloupý režim vypne detekci všech výzev na straně dial-in serveru. Pokud je navázání spojení pomalé nebo vůbec nefunguje, zkuste tuto volbu.

#### *Vnější rozhraní firewallu*

Volbou *Vnější rozhraní firewallu* aktivujete firewall a nastavíte toto rozhraní jako externí. Vaše vytáčená připojení k Internetu tak budou chráněna před možnými útoky z vnější sítě.

#### *Čas nečinnosti (v sekundách)*

Tato volba určuje čas v sekundách, po kterém se spojení přeruší, nejsou-li přenášena žádná data (0 znamená nekonečno).

#### *Detaily IP*

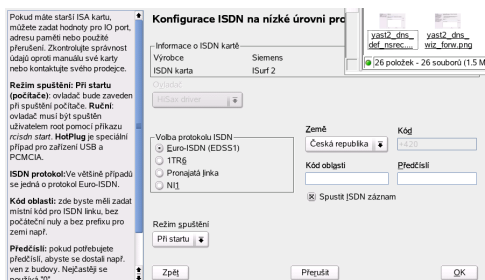
Kliknutím na tlačítko otevřete dialog pro nastavení IP adresy. Pokud váš poskytovatel připojení nepoužívá dynamické přidělování IP adres, zakažte volbu *Dynamická IP adresa* a vložte lokální IP adresu svého počítače a vzdálenou IP adresu (na adresy se zeptejte svého poskytovatele). Volbu *Výchozí směrování* ponechte zaškrtnutou a dialog ukončete kliknutím na *OK*.

Kliknutím na *Další* se vrátíte k původnímu dialogu, který zobrazuje souhrn konfigurace modemů. Dialog zavřete kliknutím na *Konec*.

## 18.4.3 ISDN

Tento modul použijte ke konfiguraci jedné nebo více ISDN karet. Pokud YaST kartu nedetekoval, vyberte ji ručně. Je možno nastavit více rozhraní, ale i jedno rozhraní může být nastaveno pro více ISP. V následujících dialogích nastavte volby ISDN nutné pro správnou funkci karty.

**Obrázek 18.5** Konfigurace ISDN

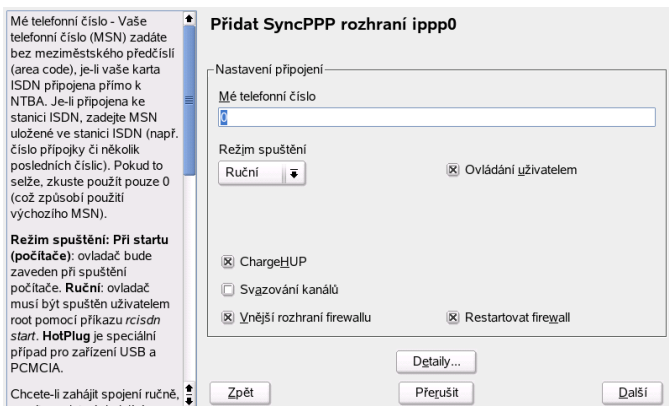


V dialogu zobrazeném na obrázku 18.5 – „Konfigurace ISDN“ (strana 298) vyberte požadovaný protokol. Implicitní je *Euro-ISDN (EDSS1)*, ale pro starší nebo větší ústředny použijte *1TR6*. Pokud se nacházíte v USA, vyberte *NI1*. V příslušném poli nastavte zemi. V sousedním poli se objeví příslušný kód. Zadejte *Kód oblasti* a (pokud potřebujete) *Předčísli*.

*Režim spuštění* určuje, jak je ISDN rozhraní spouštěno: *Při startu* znamená, že je ISDN ovladač zaváděn vždy při startu systému. Je-li zvoleno *Ručně*, musí být ovladač zaveden uživatelem `root` pomocí příkazu `rcisdn start`. *Hotplug* se používá pro zařízení PCMCIA nebo USB, ovladač se nahraje po připojení zařízení. Jste-li s nastavením hotovi, stiskněte *OK*.

V následujícím dialogu vyberte pro ISDN kartu rozhraní a k němu poskytovatele připojení. Rozhraní může být typu `SyncPPP` nebo `RawIP`, většina poskytovatelů však dnes používá níže popsaný `SyncPPP`.

## Obrázek 18.6 Konfigurace ISDN rozhraní



Číslo, které je třeba vložit do pole *Mé telefonní číslo*, závisí na konkrétní situaci:

### ISDN karta přímo připojena do telefonní zásuvky

Standardní ISDN linka poskytuje tři telefonní čísla (tzv. vícenásobné účastnické číslo, MSN). Pokud účastník požaduje čísel více, může jich být až deset. Jedno z těchto čísel je na tomto místě nutné vybrat a nastavit, ale bez kódu oblasti. Pokud vložíte nesprávné číslo, váš telefonní operátor automaticky použije první z čísel přidělených vaší ISDN lince.

### ISDN karta připojená k telefonní ústředně

Konfigurace opět závisí na instalovaném zařízení:

1. Menší ústředny určené k domácímu použití obvykle pro interní hovory používají protokol Euro-ISDN (EDSS1). Tyto ústředny mají vnitřní sběrnici S0 a pro připojená zařízení používají interní čísla.

Použijte jedno z interních čísel. Měli byste moci použít alespoň jedno z čísel ústředny, kterým je umožněno přímé volání ven. Pokud to nefunguje, zkuste jednu nulu. Další informace naleznete v dokumentaci dodané s vaší ústřednou.

2. Větší ústředny určené pro firmy obvykle pro vnitřní hovory používají protokol 1TR6. Jejich MSN (vícenásobné účastnické číslo) se nazývá EAZ a obvykle odpovídá přímému volacímu číslu. Pro nastavení v Linuxu by mělo stačit použít poslední číslici EAZ. Pokud to nefunguje, vyzkoušejte všechny číslice od 1 do 9.

Chcete-li spojení ukončovat těsně před započtením další tarifní jednotky (impulzu), zaškrtněte *ChargeHUP*. Nemusí však fungovat s každým poskytovatelem. Můžete také povolit *svazování kanálů* (multilink PPP). Zaškrtnutím volby *Vnější rozhraní firewallu* aktivujete SuSEfirewall2 a nastavíte toto rozhraní jako externí. Chcete-li povolit běžným uživatelům aktivaci a deaktivaci rozhraní, zaškrtněte volbu *Ovládání uživatelem*.

Výběrem *Detaily...* otevřete dialog s pokročilým nastavením, které není určeno pro běžné domácí uživatele. Pokračujte proto k dalšímu dialogu stisknutím tlačítka *Další*.

V dalším dialogu nastavte IP adresu. Pokud vám poskytovatel připojení nepřidělil pevnou IP adresu, zvolte *Dynamická IP adresa*. V opačném případě zadejte lokální IP adresu (adresa vašeho počítače) a vzdálenou IP adresu podle specifikace vašeho poskytovatele. Pokud má být toto rozhraní používáno jako výchozí pro směrování paketů, zaškrtněte volbu *Výchozí směrování*. Na každém počítači může být jako výchozí nastaveno pouze jedno rozhraní. Pokračujte stisknutím tlačítka *Další*.

Následující dialog umožňuje nastavit zemi, ve které se nacházíte, a poskytovatele připojení (ISP). V seznamu jsou pouze operátoři dostupní přes službu Call-by-Call (volba operátora předčíslení). Pokud v seznamu není váš poskytovatel, zvolte *Nový*. Tím se otevře dialog *Volby poskytovatele*, do kterého vložte příslušné údaje. Ujistěte se, že jste do telefonního čísla nevložiteli žádné mezery nebo čárky. Zadejte uživatelské jméno a heslo přidělené poskytovatelem a stiskněte *Další*.

Chcete-li na samostatné pracovní stanici používat *Vytáčení na vyžádání*, zadejte jmenný server (nameserver, DNS). Většina poskytovatelů podporuje dynamický DNS, což znamená, že adresa jmenného serveru je zaslána poskytovatelem vždy v okamžiku připojení. Na samostatné pracovní stanici je ovšem i v takovém případě uvést zástupnou adresu, např. 192 . 168 . 22 . 99. Pokud poskytovatel dynamický DNS nepodporuje, musíte zadat IP adresu jmenného serveru poskytovatele. Pokud chcete, můžete v položce *Čas nečinnosti (v sekundách)* zadat i dobu, po které se spojení automaticky přeruší, nejsou-li přenášena žádná data. Nastavení potvrďte zvolením *Další*. YaST zobrazí přehled nastavených rozhraní. Stisknutím *Konec* nastavení aktivujete.

## 18.4.4 Kabelový modem

V některých zemích (v Rakousku, USA, ale i u nás) je běžný přístup na Internet přes síť kabelové televize). Účastník sítě obvykle dostane modem, který je na jedné straně připojen k rozvodu kabelové televize a na druhé straně k síťové kartě počítače (pomocí kabelu 10Base-T kroucený pár).

V závislosti na instrukcích od vašeho poskytovatele připojení zvolte při konfiguraci síťové karty buď *Automatické přidělení adresy (pomocí DHCP)* nebo *Nastavení statické adresy*. Dnes většina poskytovatelů používá DHCP. Statická adresa je obvykle volitelnou doplňkovou službou.

## 18.4.5 DSL

Chcete-li nakonfigurovat zařízení DSL, zvolte modul *DSL* ze sekce *Síťová zařízení* nástroje YaST. Modul sestává z několika dialogů, v nichž je třeba nastavit parametry DSL linky založené na některém z následujících protokolů:

- PPP přes Ethernet (PPPoE)
- PPP přes ATM (PPPoATM)
- CAPI pro ADSL (Fritz karty)
- Point-to-Point Tunneling Protocol (PPTP) – Rakousko

Konfigurace DSL připojení založeného na PPPoE nebo PPTP vyžaduje předem správně nastavenou síťovou kartu. Pokud ještě karta není nastavena, nastavte ji volbou *Konfigurovat síťové karty* (viz 18.4.1 – „[Konfigurace síťové karty pomocí YaST](#)“ (strana 293)). V případě DSL připojení sice mohou být adresy automaticky přidělovány, ale nikoliv pomocí DHCP. Proto volbu *Automatické přidělení adresy (přes DHCP)* ponechte nezaškrtnutou. Místo toho zadejte statickou fiktivní adresu rozhraní, např. 192 . 168 . 22 . 1. V poli *Síťová maska podsítě* zadejte 255 . 255 . 255 . 0. Pokud nastavujete samostatnou pracovní stanici, ujistěte se, že je položka *Výchozí brána* (v dialogu *Směrování*) prázdná.

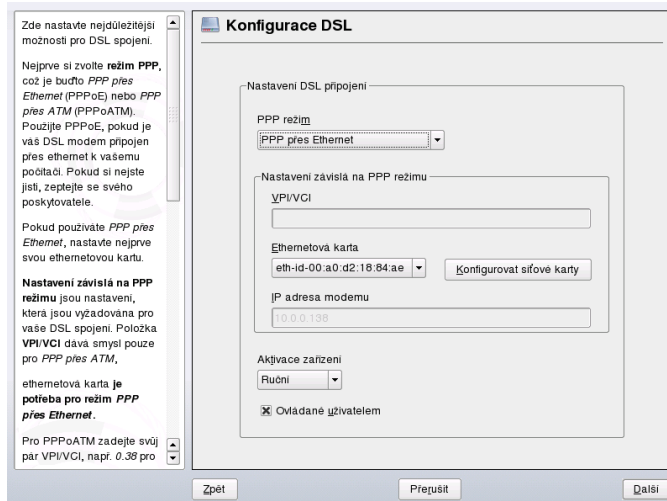
---

### Tip

Hodnoty *IP Adresa* a *Síťová maska podsítě* jsou pouze zástupné a nereprezentují DSL připojení jako takové. Slouží pouze k inicializaci síťové karty.

---

## Obrázek 18.7 Konfigurace DSL



Konfiguraci DSL (viz obrázek 18.7 – „Konfigurace DSL“ (strana 302)) začněte výběrem PPP režimu a ethernetové karty, ke které je modem připojen (obvykle je to eth0). Pak ze seznamu *Aktivace zařízení* zvolte způsob aktivace DSL připojení. Pokud chcete povolit běžným uživatelům aktivaci či deaktivaci rozhraní pomocí programu KInternet, zaškrtněte položku *Ovládání uživatelem*. V dalším dialogu zvolte zemi a poskytovatele připojení (ISP). Podrobnosti nastavení v dalších dialogích závisí na dosud provedeném nastavení, proto jsou v následujících odstavcích jen krátce zmíněny. Podrobnosti se dozvíte z nápovědy přímo v jednotlivých dialogích.

Chcete-li používat *Vytáčení na vyžádání* na samostatné pracovní stanici, zadejte adresu jmenného serveru (nameserver, DNS). Většina poskytovatelů podporuje dynamický DNS – IP adresa jmenného serveru je zaslána poskytovatelem při každém připojení. Pro samostatnou stanici však v takovém případě zadejte zástupnou adresu, např. 192.168.22.99. Pokud váš poskytovatel dynamický DNS nepodporuje, zadejte adresu, kterou vám dodal.

*Čas nečinnosti (v sekundách)* určuje dobu síťové neaktivity, po které bude spojení automaticky přerušeno. Vhodná je hodnota mezi 60 a 300 sekundami. Pokud je zakázáno *Vytáčení na vyžádání*, může být užitečné nastavit dobu nečinnosti rovnou nule, což znemožní automatické přerušování spojení.

Chcete-li nastavit T-DSL, postupujte stejně jako při nastavení DSL. Pouze při výběru poskytovatele připojení zvolte *T-Online*. YaST otevře dialog pro nastavení T-DSL, ve kterém vyplňte některé doplňující informace vyžadované T-DSL, jako ID linky, T-Online číslo, uživatelský kód a heslo. Všechny potřebné údaje jste dostali při přihlášení ke službě T-DSL.

## 18.5 Správa sítě s programem NetworkManager

NetworkManager je ideální řešení pro mobilní pracovní stanice. Pokud jej použijete, nemusíte se starat o nastavení síťového rozhraní a přepínání mezi sítěmi, když se přeusouváte. NetworkManager se umí sám automaticky připojit ke známé bezdrátové síti. Pokud se nabízí více možností připojení, zvolí pro vás tu nejrychlejší.

---

### **Poznámka: NetworkManager a SCPM**

Nepoužívejte NetworkManager dohromady se SCPM, pokud SCPM profily mění také nastavení sítě. Jestliže chcete NetworkManager a SCPM používat současně, odstraňte ze zdrojů SCPM nastavení sítě.

---

NetworkManager není vhodným řešením pro následující případy:

- Váš počítač používá pevnou IP adresu
- Chcete pro vytáčené připojení používat více než jednoho poskytovatele
- Chcete v bezdrátovém připojení používat šifrování WPA-EAP
- Váš počítač slouží jako router
- Váš počítač slouží jako server, např. DHCP nebo DNS

### 18.5.1 Ovládání NetworkManager

Abyste mohli NetworkManager používat, musíte jej nejprve povolit v programu YaST v modulu nastavení síťové karty. Protože NetworkManager nepotřebuje klasické síťové nastavení, tradiční nastavení se v programu YaST deaktivuje. NetworkManager se umí

automaticky připojit pouze do známých sítí, proto je nutné mu nejprve zdělit, které síť chcete používat a případně zadat ověřovací a šifrovací údaje. Pro první připojení použijte NetworkManager applet. Pokud zvolená síť bude vyžadovat dodatečné údaje, applet vás o ně požádá.

KDE i GNOME obsahují vlastní applety programu NetworkManager. Příslušný applet se spustí automaticky při přihlášení do prostředí a zobrazí se v systémovém panelu. Funkce obou appletů jsou velmi podobné, ale liší se v grafickém rozhraní. Jestliže používáte jiné grafické prostředí s podporou panelů, můžete je použít i v těchto prostředích.

## KNetworkManager — OptioKDE NetworkManager applet

KNetworkManager je KDE applet pro ovládání programu NetworkManager. Jestliže neběží, lze jej spustit příkazem `knetworkmanager`. Jestliže běží, uvidíte v panelu modrou ikonku zeměkoule. Nabídku appletu vyvoláte, když na ikonku kliknete pravým tlačítkem myši.

V nabídce najdete dostupné sítě, klasické i bezdrátové. Pokud nad síť najedete kurzorem, zobrazí se detaily. Aktuálně používané připojení je označené jako vybrané. Šifrované bezdrátové sítě jsou označeny modrou ikonkou zámku. K šifrované síti se připojíte kliknutím na její položku. Jestliže jde o vaše první připojení k této síti, budete vyzváni k zadání *Šifrování* a příslušného *Hesla* nebo *Klíče*.

K síti, která nevysílá své jméno (ESSID), se připojíte tak, že zvolíte z nabídky položku *Připojit k jiné bezdrátové síti*. V následujícím dialogu zadejte ESSID a údaje o šifrování.

Vytáčené připojení aktivujete volbou *Vytáčená připojení*. Jestliže již máte nadefinovaného poskytovatele, kliknutím na uvedenou položku rovnou spustíte vytáčení. Nové nastavení vytvoříte kliknutím na *Nastavit vytáčené připojení*.

Pokud se chcete odpojit ze sítě, zvolte *Možnosti* → *Přepnout do offline režimu*. Připojení opět povolíte volbou *Možnosti* → *Přepnout do online režimu*. Jestliže chcete zakázat pouze bezdrátové připojení, zvolte *Možnosti* → *Zakázat bezdrát*. Připojení opět povolíte volbou *Možnosti* → *Povolit bezdrát*. Ukončení a navazování spojení může trvat několik sekund.



## GNOME NetworkManager applet

Také GNOME má svůj applet pro NetworkManager. Pokud neběží, můžete jej spustit příkazem `nm-applet`. Jeho běh signalizuje ikona v systémovém panelu. Vzhled ikony je závislý na stavu připojení. Pokud si nejste jisti významem ikony, najed'te myši na ikonu. Tím zobrazíte nápovědu.

Kliknutím levým tlačítkem na ikonku appletu získáte seznam dostupných sítí. Aktuálně používané síť je označena. V nabídce je zobrazena také síla signálu u bezdrátových sítí. Šifrované sítě jsou označeny ikonou štítu. Do sítě se připojíte jejím vybráním ze seznamu. V následujícím dialogu zadejte *Šifrování* a *Heslo* nebo *Klíč*.

K síti, která nevysílá své jméno (ESSID), se připojíte tak, že zvolíte z nabídky položku *Připojit se k jiné bezdrátové síti*. V následujícím dialogu zadejte ESSID a údaje o šifrování.

Pokud chcete přejít do offline režimu, deaktivujte v nabídce položku *Povolit síť*. Pouze bezdrátovou síť vypnete, pokud deaktivujete položku *Povolit bezdrátové*.

Informace o aktuálním připojení (rozhraní, IP adresa, adresa sítě) zobrazíte kliknutím pravým tlačítkem myši a volbou *Informace o spojení*.

## 18.5.2 Další informace

Více informací NetworkManager a technologii d-bus najdete na stránkách a v adresářích:

- <http://www.gnome.org/projects/NetworkManager/> Stránka projektu NetworkManager
- <http://www.freedesktop.org/Software/dbus> Stránka projektu d-bus
- `/usr/share/doc/packages/NetworkManager`

## 18.6 Manuální konfigurace sítě

Manuální konfigurace sítě by měla být používána pouze jako nouzové řešení nebo ve speciálních případech. Jinak je lepší využít YaST. Zde uvedené informace o konfiguraci sítě ale mohou být užitečné i při práci s YaSTem.

Všechny vestavěné i hotplug (PCMCIA, USB, některé PCI) síťové karty jsou detekovány a konfigurovány pomocí hotplug systému. Systém chápe síťovou kartu dvěma různými způsoby: jako fyzické zařízení a jako rozhraní. Připojení nebo rozpoznání zařízení spustí hotplug událost, která zahájí inicializaci zařízení pomocí skriptu `hwup`. Pokud je síťová karta inicializována jako nové síťové rozhraní, jádro vyvolá další hotplug událost, která pomocí `ifup` rozhraní nastaví.

Jádro přiděluje jména rozhraní podle časového pořadí jejich registrace. O přidělených jménech rozhoduje inicializační sekvence. Když jedna z několika síťových karet selže, čísla všech následujících karet se posunou. V případě skutečných hotplug karet (připojitelných za běhu systému) rozhoduje okamžik (pořadí) připojení k systému.

Pro zvýšení flexibility byla oddělena konfigurace zařízení (hardware) a rozhraní; a přiřazování konfigurací k zařízením a rozhraním již není založeno na jménech rozhraní. Konfigurace zařízení jsou uloženy v souborech `/etc/sysconfig/hardware/hwcfg-*`, zatímco v souborech `/etc/sysconfig/network/ifcfg-*` jsou uloženy konfigurace rozhraní. Jména konfigurací jsou přiřazována tak, že popisují zařízení a rozhraní, s nimiž jsou spojeny. Protože dříve používané přiřazování ovladačů ke jménům rozhraní vyžadovalo stálá jména rozhraní, nelze přiřazování jmen nadále provádět v souboru `/etc/modprobe.conf`. Uvedení aliasu v tomto souboru může nyní mít nepříjemné vedlejší účinky.

Jména konfiguračních souborů (vše, co následuje po `hwcfg-` či `ifcfg-`) mohou na jednotlivá zařízení odkazovat pomocí použité sběrnice, ID zařízení nebo jména rozhraní. Například konfigurace PCI karty může být `bus-pci-0000:02:01.0` (sběrnice PCI) nebo `vpid-0x8086-0x1014-0x0549` (identifikační číslo produktu). Jméno příslušného rozhraní může být `bus-pci-0000:02:01.0` nebo `wlan-id-00:05:4e:42:31:7a` (MAC adresa).

Chcete-li přiřadit konfiguraci libovolné kartě určitého typu (pokud je v tu chvíli připojena jen jedna karta tohoto typu), místo konkrétní kartě, zvolte méně specifické jméno konfigurace. Například, konfigurace se jménem `bus-pcmcia` bude použita libovolnou

PCMCIA kartou. Chcete-li rozsah použití omezit, přidejte na začátek jména typ rozhraní, např. `wlan-bus-usb` bude přiřazeno všem WLAN kartám na USB portu.

System vždy použije tu konfiguraci, která zařízení nebo rozhraní nejlépe popisuje. Nejvhodnější konfiguraci vyhledává program `getcfg`. Výstup programu obsahuje veškeré informace použitelné pro popis zařízení. Podrobnosti o pravidlech tvorby jmen konfigurací naleznete v manuálové stránce `getcfg`.

Vzhledem k popsané metodě jsou síťová rozhraní vždy správně nakonfigurována bez ohledu na pořadí inicializace. Nicméně jméno rozhraní na pořadí inicializace stále závisí. Jsou dva způsoby, jak zajistit spolehlivý přístup k rozhraní určité síťové karty:

- `getcfg-interface jméno konfigurace` vrací jméno rozhraní asociovaného s danou konfigurací. V některých konfiguračních souborech tak lze místo nestálého jména rozhraní použít jméno konfigurace (např. `firewall`, `dhcpd`, směrování nebo různá virtuální síťová rozhraní, `tunely`).
- Rozhraním, jejichž konfigurace jméno rozhraní neobsahuje, můžete trvalé jméno přiřadit pomocí perzistentního (trvalého) jména v `/etc/udev/rules.d/30-net_persistent_names.rules`. Trvalá jména (*pname*) by ovšem neměla být stejná, jako jména automaticky přidělovaná jádrem. Proto nejsou povolena jména jako `eth*`, `tr*`, `wlan*` atd. Místo nich používejte `net*` nebo popisná jména jako `vnejsi`, `vnitrni` či `dmz`. Trvalá jména je možné přiřadit rozhraní pouze vzápětí po jeho registraci, což znamená, že je nutné znovu zavést ovladač síťové karty nebo spustit příkaz `hwup popis_zarizeni`. Příkaz `rcnetwork restart` není v tomto případě dostatečný.

---

### **Důležité: Použití trvalých jmen rozhraní**

Použití trvalých jmen zatím nebylo důkladně otestováno. Proto se může stát, že některé aplikace nebudou schopny s volně vybranými jmény rozhraní zacházet. Pokud na podobný problém narazíte, dejte nám vědět na adrese <http://www.suse.de/feedback>. Pokud upřednostňujete komunikaci v českém jazyce, napište nám na adresu [feedback@suse.cz](mailto:feedback@suse.cz)

---

`ifup` vyžaduje existenci rozhraní, protože neinicializuje hardware. Inicializaci hardwaru má na starost příkaz `hwup` (spouštěný pomocí `hotplug` nebo `coldplug`). Jakmile je zařízení inicializováno, je pomocí `hotplug` automaticky spuštěn `ifup`. Rozhraní je spuštěno, pokud je startovací režim nastaven na `onboot`, `hotplug` nebo `auto` a služba `network` je spuštěna. Dříve inicializaci hardwaru spouštěl příkaz

`ifup jmeno_zarizeni`. Nyní je postup opačný. Nejprve je inicializována hardwarová komponenta, pak následují ostatní akce. Tímto způsobem lze pomocí existující sady konfiguračních optimálně nakonfigurovat měnící se množství zařízení.

Tabulka 18.5 – „Skripty pro manuální síťovou konfiguraci“ (strana 308) shrnuje nejdůležitější skripty účastníci se síťové konfigurace. Tam kde je to možné, jsou rozlišeny podle toho, zda se týkají hardwaru nebo rozhraní:

**Tabulka 18.5** *Skripty pro manuální síťovou konfiguraci*

Fáze konfi- gurace	Příkaz	Funkce
Hardware	<code>hw{up,down,status}</code>	Skripty <code>hw*</code> jsou spouštěny systémem hotplug, aby inicializovaly zařízení, zrušily inicializaci nebo zjistily stav zařízení. Více informací naleznete v manuálové stránce <code>hwup</code> .
Rozhraní	<code>getcfg</code>	Skript <code>getcfg</code> lze použít ke zjištění jména rozhraní asociovaného s určitým jménem konfigurace nebo popisem zařízení. Více informací naleznete v manuálové stránce <code>getcfg</code> .
Rozhraní	<code>if{up,down,status}</code>	Skripty <code>if*</code> spouští existující síťová rozhraní nebo vrací stav určeného rozhraní. Více informací naleznete v manuálové stránce <code>ifup</code> .

Další informace o systému hotplug a trvalých jménech rozhraní naleznete v kapitole 12 – „*Dynamické uzly zařízení pomocí `udev`*“ (strana 217).

## 18.6.1 Konfigurační soubory

Zde je uveden přehled síťových konfiguračních souborů, jejich formátů a funkcí.

## **/etc/sysconfig/network/hwcfg-\***

Tyto soubory obsahují hardwarovou konfiguraci síťových karet a dalších zařízení. Obsahují potřebné parametry, jako je jaderný modul, režim spouštění a asociace se skripty. Více informací najdete v manuálové stránce `hwup`. Bez ohledu na existující hardware jsou při spuštění `coldplug` aplikovány konfigurační soubory `hwcfg-static-*`.

## **/etc/sysconfig/network/ifcfg-\***

Tyto soubory obsahují data pro jednotlivá síťová rozhraní. Obsahují např. režim spouštění a IP adresu. Možné parametry jsou popsány v manuálové stránce `ifup`). Navíc lze, pokud chcete obecné nastavení použít jen pro jedno rozhraní, používat v `ifcfg-*` souborech všechny proměnné ze souborů `dhcp`, `wireless`, a `config`.

## **/etc/sysconfig/network/config, dhcp, wireless**

Soubor `config` obsahuje obecné nastavení chování skriptů `ifup`, `ifdown` a `ifstatus`. Soubor `dhcp` obsahuje nastavení pro DHCP. Soubor `wireless` obsahuje nastavení pro bezdrátové síťové karty. Proměnné v těchto souborech jsou dobře komentovány. Všechny proměnné z těchto souborů je možné použít také v `ifcfg-*`, kde mají vyšší prioritu.

## **/etc/sysconfig/network/routes, ifroute-\***

Zde je nastaveno statické směrování TCP/IP paketů. Všechny statické směrovací záznamy vyžadované různými systémovými úlohami lze nastavit v souboru `/etc/sysconfig/network/routes`: pro směrování k počítači, skrze bránu nebo k síti. Pro všechna rozhraní, která potřebují individuální směrování, je možné vytvářet samostatné konfigurační soubory `/etc/sysconfig/network/ifroute-*` (hvězdičku nahraďte názvem rozhraní). Záznamy ve směrovacích konfiguračních souborech vypadají následovně:

# Destination	Dummy/Gateway	Netmask	Device
#			
127.0.0.0	0.0.0.0	255.255.255.0	lo
204.127.235.0	0.0.0.0	255.255.255.0	eth0
default	204.127.235.41	0.0.0.0	eth0

207.68.156.51	207.68.145.45	255.255.255.255	eth1
192.168.0.0	207.68.156.51	255.255.0.0	eth1

V prvním sloupci (DESTINATION) je uveden cíl směrovacího záznamu. Může zde být IP adresa sítě nebo počítače. Pokud je dostupný nameserver, pak také celý název sítě nebo počítače.

Druhý sloupec (GATEWAY) slouží pro uvedení výchozí brány nebo brány, skrze kterou se přistupuje k počítači, resp. síti.

Ve třetím sloupci se uvádějí síťové masky pro síť nebo počítače za bránou, např. 255.255.255.255.

Čtvrtý sloupec má smysl pro síť připojené k lokálnímu počítači, jako např. loopback, ethernet, ISDN, PPP či dummy zařízení. Musí v něm být zapsáno jméno zařízení.

Pátý (volitelný) sloupec lze použít k zadání typu směrování. Aby se předešlo případným chybám parseru, nevyplněné sloupce, které není třeba zadávat, by měly obsahovat znaménko mínus –Podrobnosti naleznete v manuálové stránce `routes(5)`.

## **`/etc/resolv.conf`**

V tomto souboru je specifikována doména, do které počítač patří (klíčové slovo `search`). Je uvedena též adresa nameserveru, ke kterému se má přistupovat (klíčové slovo `nameserver`). Lze uvést i více domén. Při převodu jména, které není plně kvalifikováno, se k němu postupně připojují jednotlivé položky `search`. Více nameserverů lze uvést zápisem více řádků začínajících klíčovým slovem `nameserver`. Komentáře jsou uvozeny znaky `#`. YaST zapisuje nastavení nameserveru do tohoto souboru. 18.5 – „`/etc/resolv.conf`“ (strana 310) ukazuje příklad skutečného souboru `/etc/resolv.conf`.

### ***Rovnice 18.5*** `/etc/resolv.conf`

```
# Our domain
search example.com
#
# We use sun (192.168.0.20) as nameserver
nameserver 192.168.0.20
```

Některé služby, jako `pppd` (`wvdial`), `ipppd` (`isdn`), `dhcp` (`dhcpcd` a `dhclient`), `pcmcia` a `hotplug`, modifikují soubor `/etc/resolv.conf` pomocí skriptu `modify_resolvconf`. Pokud byl soubor skriptem `/etc/resolv.conf` dočasně změněn,

obsahuje komentář informující o službě, která změnu provedla, místu, kde je uložena záloha původního souboru a o způsobu, jakým můžete zamezit automatickým změnám souboru. Pokud je soubor `/etc/resolv.conf` změněn vícekrát, obsahuje všechny změny ve vnořené podobě. Změny lze korektně vrátit i v jiném pořadí, než byly učiněny. Mezi služby, které toho využívají, patří `isdn`, `pcmcia` a `hotplug`.

Pokud se stane, že je služba ukončena nestandardním způsobem, lze k obnovení původního souboru použít `modify_resolvconf`. Při startu systému se rovněž kontroluje, zda není přítomen modifikovaný `resolv.conf` (např. po pádu systému), případně je původní nezměněný soubor `resolv.conf` obnoven.

YaST pomocí `modify_resolvconf` kontroluje, zda byl `resolv.conf` modifikován, a případně varuje uživatele, že se provedené změny po obnovení souboru ztratí. Navíc YaST sám `modify_resolvconf` nepoužívá, což znamená, že změna souboru `resolv.conf` provedená pomocí YaST má stejnou váhu jako manuální editace. V obou případech je změna trvalá, zatímco změny provedené výše zmíněnými službami jsou pouze dočasné.

## **`/etc/hosts`**

V tomto souboru (viz [18.6 – „/etc/hosts“](#) (strana 311)) se jménům počítačů přiřazují IP adresy. Pokud se nepoužívá nameserver, musíte zde uvést všechny počítače, na které chcete mít přístup pomocí jména. Každý počítač je na zvláštní řádce, sestávající postupně z IP adresy, plně kvalifikovaného jména počítače a jména počítače. IP adresa musí být uvedena na začátku řádky, položky musí být odděleny mezerami nebo tabulátory. Komentáře začínají znakem `#`.

### ***Rovnice 18.6*** `/etc/hosts`

```
127.0.0.1 localhost
192.168.0.20 sun.example.com sun
192.168.0.0 earth.example.com earth
```

## **`/etc/networks`**

V tomto souboru se nastavuje převod jmen sítí na síťové adresy. Formát je podobný jako u souboru `hosts`, pouze síťová jména jsou první a za nimi následují adresy. Viz [18.7 – „/etc/networks“](#) (strana 312).

### **Rovnice 18.7** */etc/networks*

```
loopback      127.0.0.0
localnet      192.168.0.0
```

## **/etc/host.conf**

Tento soubor kontroluje převod jmen pomocí *resolver* knihovny. Používá se pouze programy slinkovanými proti *libc4* nebo *libc5*. Novější *glibc* programy se nastavují v souboru */etc/nsswitch.conf*. Každý parametr je uveden na samostatném řádku a komentáře jsou uvozeny znakem *#*. Přípustné parametry jsou uvedeny v tabulce 18.6 – „Parametry pro */etc/host.conf*“ (strana 312). Ukázku souboru */etc/host.conf* si můžete prohlédnout v příkladu 18.8 – „*/etc/host.conf*“ (strana 313).

**Tabulka 18.6** *Parametry pro /etc/host.conf*

---

<i>order hosts, bind</i>	Stanoví, v jakém pořadí se volají služby pro převod jména počítače na IP adresu. Možné argumenty jsou (odděleny mezerami nebo čárkami):  <i>hosts</i> : prohledávat soubor <i>/etc/hosts</i>  <i>bind</i> : použít nameserver  <i>nis</i> : použít NIS
<i>multi on/off</i>	Stanoví, zda počítač, uvedený v <i>/etc/hosts</i> smí mít více IP adres.
<i>nospoof on</i> <i>spoofalert on/off</i>	Tyto parametry mají vliv pouze na <i>spoofing</i> nameserveru.
<i>trim</i> název domény	Zadané jméno domény se při převodu oddělí od jména počítače (pokud ovšem jméno počítače obsahovalo doménu). Tato volba se hodí, pokud jsou v souboru <i>/etc/hosts</i> jen jména z lokální domény, které by však měla být rozpoznatelná i s připojenou doménou.

---



### **Rovnice 18.8** */etc/host.conf*

```
# We have named running
order hosts bind
# Allow multiple addrs
multi on
```

## **/etc/nsswitch.conf**

S GNU C Library 2.0 můžete nyní využívat tzv. *Name Service Switch* (NSS). (Viz man 5 `nsswitch.conf` a manuál *The GNU C Library Reference Manual*.)

V souboru `/etc/nsswitch.conf` je uvedeno pořadí dotazů. Soubor `nsswitch.conf` si můžete prohlédnout v příkladu 18.9 – „`/etc/nsswitch.conf`“ (strana 313). Komentáře jsou uvozeny znaky `#`. V tomto příkladu uvedená položka `hosts` znamená, že po dotazu na `/etc/hosts` (`files`) je proveden dotaz pomocí DNS (viz kapitolu 20 – „*DNS — Domain Name System*“ (strana 323)).

### **Rovnice 18.9** */etc/nsswitch.conf*

```
passwd:      compat
group:       compat

hosts:       files dns
networks:    files dns

services:    db files
protocols:   db files

netgroup:    files
automount:   files nis
```

Databáze dosažitelné pomocí NSS jsou uvedeny v tabulce 18.7 – „[Databáze dosažitelné pomocí /etc/nsswitch.conf](#)“ (strana 313). V budoucnu se navíc počítá s parametry `automount`, `bootparams`, `netmasks` a `publickey`. Konfigurační volby pro databáze jsou uvedeny v tabulce 18.8 – „[Konfigurační možnosti NSS databází](#)“ (strana 314).

### **Tabulka 18.7** *Databáze dosažitelné pomocí /etc/nsswitch.conf*

---

<code>aliases</code>	Poštovní aliasy pro <code>sendmail</code> ; viz man 5 <code>aliases</code> .
<code>ethers</code>	Ethernetové adresy.

group	Uživatelské skupiny pro <code>getgrent</code> . Viz man 5 <code>group</code> .
hosts	Jména počítačů a IP adresy pro <code>gethostbyname</code> a podobné funkce.
netgroup	Platný seznam počítačů a uživatelů v síti pro účely kontroly přístupových práv, viz manuálová stránka <code>netgroup</code> (5).
networks	Jména a adresy sítí pro <code>getnetent</code> .
passwd	Uživatelská hesla pro <code>getpwent</code> ; viz manuálové stránka <code>passwd</code> (5).
protocols	Síťové protokoly pro <code>getprotoent</code> ; viz manuálová stránka <code>protocols</code> (5).
rpc	Jména a adresy <i>Remote procedure call</i> pro <code>getrpcbyname</code> a podobné funkce.
services	Síťové služby pro <code>getservent</code> .
shadow	Stínová hesla uživatelů pro <code>getspnam</code> ; viz manuálová stránka <code>shadow</code> (5).

---

**Tabulka 18.8** Konfigurační možnosti NSS databázi

files	Přímý přístup k souborům, například <code>/etc/aliases</code> .
db	Přístup přes databázi.
nis, nisplus	NIS, viz kapitola 21 – „ <i>NIS — Network Information Service</i> “ (strana 343).
dns	Lze použít pouze jako rozšíření <code>hosts</code> a <code>networks</code> .
compat	Lze použít pouze jako rozšíření <code>passwd</code> , <code>shadow</code> a <code>group</code> .

---

## **/etc/nscd.conf**

Pomocí tohoto souboru se konfiguruje program `nscd` (Name Service Cache Daemon), viz manuálové stránky `nscd` (8) a `nscd.conf` (5). Ve výchozím nastavení jsou položky `passwd` a `groups` programem `nscd` ukládány do vyrovnávací paměti. Je to důležité pro výkon adresářových služeb jako je NIS nebo LDAP, protože jinak by bylo nutné používat síťové spojení pro každý přístup ke jménům nebo skupinám. Položka `hosts` ukládána do vyrovnávací paměti není, protože používaný mechanismus zneumožňuje lokálním počítačům odpovědím na dotazy důvěřovat. Místo ukládání do vyrovnávací paměti programem `nscd` použijte DNS server s ukládáním do vyrovnávací paměti.

Je-li aktivována vyrovnávací paměť (cache) pro `passwd`, trvá zpravidla 15 sekund, než je systému znám nově založený lokální uživatel. Opětovným spuštěním programu `nscd` se tato doba čekání dá zkrátit. Slouží k tomu příkaz `rcnscd restart`.

## **/etc/HOSTNAME**

Tento soubor se čte různými skripty při startu systému. Smí obsahovat jedinou řádku se jménem počítače (bez domény).

## **18.6.2 Startovací skripty**

Kromě výše popsaných konfiguračních souborů existuje řada skriptů, které spouští síťové programy během startu systému. Jsou spuštěny v okamžiku, kdy systém přejde do některé *víceuživatelské úrovně běhu* (viz tabulka 18.9 – „[Některé startovací skripty pro síťové programy](#)“ (strana 315)).

**Tabulka 18.9** *Některé startovací skripty pro síťové programy*

---

<code>/etc/init.d/network</code>	Tento skript se stará o konfiguraci síťových rozhraní. Hardware musí být inicializováno předem pomocí <code>/etc/init.d/coldplug</code> (přes <code>hotplug</code> ). Pokud nebyla spuštěna služba <code>network</code> , nejsou implementována žádná síťová rozhraní.
----------------------------------	--

<code>/etc/init.d/inetd</code>	Spouští program xinetd. xinetd umožňuje na systému používat serverové služby. Například spouští vsftpd při každé inicializaci FTP spojení.
<code>/etc/init.d/portmap</code>	Spouští portmapper potřebný pro RPC server, např. NFS.
<code>/etc/init.d/nfsserver</code>	Spouští NFS server.
<code>/etc/init.d/sendmail</code>	Řídí proces sendmail.
<code>/etc/init.d/ypserv</code>	Spouští NIS server.
<code>/etc/init.d/ypbind</code>	Spouští klienta NIS.

---

## 18.7 smpppd jako pomocník s vytáčeným připojením

Většina uživatelů nemá pro internetové připojení vyhrazenou pevnou linku, ale používají vytáčené připojení. V závislosti na metodě vytáčení (ISDN nebo DSL) se o spojení stará program ippd nebo pppd. Všechno, co je potřeba pro připojení k Internetu, je správné spuštění těchto programů.

Pokud používáte paušální připojení, jednoduše spustíte příslušného démona. Stav připojení pak lze kontrolovat pomocí apletu v KDE nebo z příkazové řádky. Pokud je internetové připojení poskytováno jiným počítačem, tzv. bránou, můžete chtít připojení kontrolovat po síti.

Právě pro kontrolu vytáčeného připojení po síti je určen program smpppd. Tento program poskytuje jednotné rozhraní pro řadu programů a plní dvě funkce. První je volání programu pppd nebo ippd spolu s kontrolou vlastností vytáčeného připojení. Druhou je správa více poskytovatelů Internetu a přenos informací o aktuálním stavu připojení. Pokud používáte vytáčené připojení pro soukromou síť, můžete program smpppd ovládat také po síti.

## 18.7.1 Konfigurace smpppd

Připojení prostřednictvím smpppd je automaticky nakonfigurováno YaSTem. Programy pro vytáčení kinternet a cinternet jsou také předkonfigurovány. Manuální nastavení smpppd je potřeba pouze pro aktivaci zvláštních funkcí, jako je např. vzdálené ovládání po síti.

Konfigurační soubor smpppd je `/etc/smpppd.conf`. Ve výchozím nastavení není vzdálená kontrola povolena. Nejdůležitější volby v tomto souboru jsou:

`open-inet-socket = yes|no`

Ke kontrole smpppd po síti musí být nastavena na `yes`. Port, na kterém smpppd naslouchá, je 3185. Pokud je tento parametr nastaven na `yes`, musí být příslušně nastaveny také parametry `bind-address`, `host-range` a `password`.

`bind-address = ip`

Pokud má počítač více IP adres, nastavte zde adresu, na které má smpppd přijímat spojení.

`host-range = min ip max ip`

Parametr `host-range` se používá k nastavení rozsahu sítě. Přístup pomocí smpppd je povolen pouze počítačům z tohoto rozsahu.

`password = heslo`

Nastavením hesla omezíte přístup pouze pro autorizované uživatele. Pokud nenastavíte žádné heslo, mohou smpppd používat všichni klienti. Heslo je uloženo v textové podobě, nepřeceňujte proto jeho bezpečnost.

`slp-register = yes|no`

Tento parametr rozhoduje o zveřejňování smpppd služby v síti pomocí SLP.

Více informací o smpppd najdete v manuálových stránkách `man 8 smpppd` a `man 5 smpppd.conf`.

## 18.7.2 Programy kinternet, qinternet a cinternet a vzdálené použití

Programy kinternet, qinternet a cinternet lze používat pro ovládání lokálního i vzdáleného smpppd. Program cinternet je textová alternativa grafického programu kinternet. Program qinternet je v podstatě totéž jako kinternet, ale nepoužívá knihovny KDE, takže není na KDE závislý. Abyste mohli tyto programy používat se vzdáleným smpppd, upravte ručně nebo pomocí programu kinternet konfigurační soubor `/etc/smpppd-c.conf`. V tomto souboru jsou používány pouze tři volby:

`sites = seznam_mist`

Zde nastavte, kde mají frontendy hledat program smpppd. Frontendy testují volby v pořadí zde uvedeném. Volba `local` nařizuje připojení k lokálnímu smpppd. Volba `gateway` ukazuje na smpppd na bráně. Připojení lze nastavit ve volbě `server`. Volba `slp` nařizuje použití smpppd nalezeného přes SLP.

`server = server`

Zde nastavíte jméno počítače, na kterém běží smpppd.

`password = heslo`

Zde zadejte heslo pro smpppd.

Pokud je program smpppd aktivní, můžete otestovat přístup. To provedete příkazem `cinternet --verbose --interface-list`. Pokud narazíte na jakýkoliv problém, přečtěte si prosím manuálové stránky `cinternet` (8) a `smpppd-c.conf` (5).

## SLP služby v síti

SLP (*Service Location Protocol*) byl vyvinut pro zjednodušení konfigurace klientů v lokální síti. Taková konfigurace (včetně všech požadovaných služeb) vyžaduje detailní znalost serverů dostupných v síti. SLP informuje všechny klienty v síti o dostupnosti služeb. Aplikace, které SLP podporují, mohou tyto informace využít a provést automatickou konfiguraci.

SUSE Linux podporuje instalaci s využitím instalačních zdrojů dostupných pomocí SLP a obsahuje řadu systémových služeb s integrovanou podporou SLP. YaST i Konqueror poskytují pro SLP příslušné uživatelské rozhraní. SLP můžete využít k poskytování centrálně řízených služeb klientům, např. instalačního serveru, YOU serveru, souborového serveru nebo tiskového serveru.

### 19.1 Registrace vlastních služeb

Mnoho aplikací v systému SUSE Linux má podporu SLP integrovanou pomocí knihovny `libslp`. Pokud služba nebyla přeložena s podporou SLP a chcete, aby byla přes SLP dostupná, použijte jeden z následujících postupů:

Statická registrace pomocí `/etc/slp.reg.d`

Pro každou službu vytvořte zvláštní registrační soubor. Následující příklad ukazuje soubor pro registraci skenovací služby:

```
## Register a saned service on this system
## en means english language
## 65535 disables the timeout, so the service registration does
```

```
## not need refreshes
service:scanner.sane://$HOSTNAME:6566,en,65535
watch-port-tcp=6566
description=SANE scanner daemon
```

Nejdůležitější řádek souboru je řádek obsahující *URL služby*, který začíná řetězcem `service:`. Obsahuje typ služby (`scanner.sane`) a adresu, na které je služba na serveru dostupná. `$HOSTNAME` je automaticky nahrazeno úplným jménem počítače. Za dvojtečkou následuje číslo TCP portu, na kterém je služba dostupná. Následuje kód jazyka, ve kterém má být služba dostupná, a doba registrace v sekundách, obojí oddělené čárkou. Doba registrace zadávejte v rozmezí 0 až 65535. 0 registraci znemožňuje, 65535 ruší veškerá omezení.

Registrační soubor také obsahuje dvě proměnné: `watch-tcp-port` a `description`. První váže SLP oznámení služby na to, zda služba skutečně běží (slpd kontroluje stav služby). Druhá obsahuje přesnější popis služby pro zobrazení ve vhodných prohlížečích.

Statická registrace pomocí `/etc/slp.reg`

Jediným rozdílem oproti postupu popsánému výše je seskupení všech služeb v jednom centrálním souboru.

Dynamická registrace pomocí `slptool`

Pokud chcete zaregistrovat službu pro SLP z proprietárního skriptu, použijte příkaz `slptool` jako frontend.

## 19.2 SLP frontendy v systému SUSE Linux

SUSE Linux obsahuje několik frontendů, které umožňují kontrolovat a využívat SLP informace přes síť:

`slptool`

`slptool` je jednoduchý program pro příkazový řádek využitelný pro SLP dotazy v síti nebo pro oznamování proprietárních služeb. Příkaz `slptool --help` vypíše všechny dostupné volby a funkce programu. Příkaz `slptool` lze volat i ze skriptů, které zpracovávají SLP informace.



YaST SLP prohlížeč

YaST obsahuje samostatný SLP prohlížeč zobrazující stromový diagram se všemi službami oznámenými přes SLP v lokální síti. Je dostupný přes *Síťové služby* → *SLP prohlížeč*

Konqueror

Používáte-li Konqueror jako síťový prohlížeč, můžete zobrazit služby dostupné v lokální síti zadáním adresy `slp:/`. Kliknutím na ikony v hlavním okně získáte podrobné informace o příslušné službě. Pokud v Konqueroru zadáte adresu `service:/`, spojíte se kliknutím na ikonu s příslušnou službou.

## 19.3 Aktivace SLP

Pokud chcete nabízet služby, musí na systému běžet `slpd`. Pro pouhé dotazování na služby není nutné tohoto démona spouštět. Jako většina systémových služeb v SUSE Linuxu, je i `slpd` démon řízen samostatným init skriptem. Implicitně je démon neaktivní. Chcete-li démona aktivovat na dobu trvání relace, spusťte ho jako `root` příkazem `rcslpd start` nebo zastavte příkazem `rcslpd stop`. Volbami `restart` a `status` provedete restart a kontrolu stavu. Pokud chcete, aby byl `slpd` aktivní vždy po startu systému, spusťte jako `root` příkaz `insserv slpd`. Tím bude `slpd` automaticky zařazen mezi služby spouštěné při startu systému.

## 19.4 Další informace

O SLP jsou dostupné následující zdroje informací:

RFC 2608, 2609, 2610

RFC 2608 definuje SLP, RFC 2609 detailně popisuje URL služeb a RFC 2610 se zabývá DHCP přes SLP.

<http://www.openslp.com>

Domovská stránka projektu OpenSLP.

`file:/usr/share/doc/packages/openslp/*`

Tento adresář obsahuje všechnu dostupnou dokumentaci k SLP, včetně `README`. SUSE s detaily o systému SUSE Linux, výše zmíněných RFC a dvou úvodních HTML dokumentů. Programátoři, kteří mají zájem o využití služeb SLP, by si

měli nainstalovat balíček `openslp-devel`, ve kterém je programátorská příručka (*Programmers Guide*).

# DNS — Domain Name System

# 20

Síťová služba DNS (*Domain Name Service*) se používá k překladu doménových jmen a jmen počítačů na odpovídající IP adresy. Tím se například jménu počítače `earth` přiřadí IP adresa `192.168.0.0`. Před spuštěním vlastního nameserveru si nastudujte obecné informace o DNS v části [18.3 – „Překlad jmen“](#) (strana 291). Následující příklad konfigurace se týká nameserveru BIND.

## 20.1 Konfigurace pomocí YaST

DNS modul nástroje YaST lze použít ke konfiguraci DNS serveru pro lokální síť. Při prvním spuštění modulu se spustí průvodce základním nastavením serveru. Zodpovězením dotazů získáte jednoduchou ale funkční konfiguraci DNS serveru. V expertním režimu je možno nastavit pokročilejší volby.

### 20.1.1 Průvodce konfigurací

Průvodce nastavením sestává ze tří dialogů a umožňuje přechod do expertní konfigurace.

Instalace DNS serveru: nastavení forwarderů

Při prvním spuštění modulu spatříte dialog zobrazený na obrázku [20.1 – „Instalace DNS serveru: Nastavení forwarderů“](#) (strana 324). Umožňuje volbu mezi nastavením forwarderů pomocí PPP démona při vytáčeném spojení přes DSL nebo ISDN (*PPP démon nastaví forwardery*) a manuálním nastavením forwarderů (*Nastavit forwardery ručně*).

**Obrázek 20.1** Instalace DNS serveru: Nastavení forwarderů

**Forwardery**  
Pokud chcete povolit aktualizaci forwarderů PPP démonem, nastavte **PPP démon nastaví forwardery**. Pokud chcete aktualizovat forwardery pouze ručně, nastavte **Nastavit forwardery ručně**.

Pro přidání záznamu forwarder, zadejte **IP adresu** a klikněte na **Přidat**. Pro smazání použijte zvolte záznam forwarder a klikněte na **Smazat**.

**Instalace DNS serveru - Nastavení forwarderů**  
Zvolte nastavení pro forwarder

PPP démon nastaví Forwardery (používá se spolu s vytvářením spojení, pokud toto podporuje poskytovatel připojení)

Nastavit Forwardery ručně

Přidat IP adresu  
IP adresa

Seznam Forwarderů

## DNS zóny

Tento dialog sestává z několika částí a je zodpovědný za správu souborů zón popsaných v části [20.7 – „Struktura souboru s daty pro zónu“](#) (strana 335). Pro vytvoření nové zóny zadejte v položce *Jméno zóny* její jméno. Chcete-li přidat reverzní zónu, musí jméno končit řetězcem `.in-addr.arpa`. Dále specifikujte *Typ zóny* (master nebo slave) a klikněte na tlačítko *Přidat*. Viz obrázek [20.2 – „Instalace DNS serveru: DNS zóny“](#) (strana 324). Další nastavení zóny lze provést po kliknutí na tlačítko *Upravit*. Chcete-li zónu odstranit, použijte tlačítko *Smazat*.

**Obrázek 20.2** Instalace DNS serveru: DNS zóny

**Zóny DNS**  
V tomto dialogu můžete upravovat zóny DNS.

Pokud chcete přidat zónu, zadejte její **jméno, typ zóny** a klikněte na **Přidat**.

Pro přidání reverzní zóny, zadejte část reverzní IP adresy doplněné `.in-addr.arpa` (např. `0.168.192.in-addr.arpa` pro síť `192.168.0.0/24`).

Zónu odstraníte tak, že ji vyberete a kliknete na **Smazat zónu**.

Zónu upravíte kliknutím na **Upravit zónu...**

**Instalace DNS serveru - DNS zóny**  
Přidat novou zónu

Jméno zóny  Typ zóny

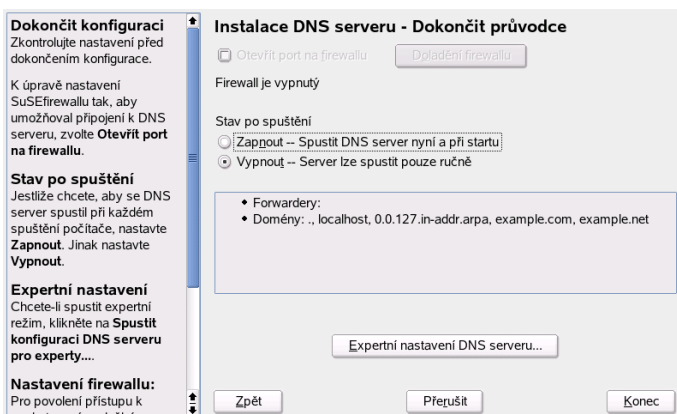
Konfigurované DNS zóny

Zóna	Typ
example.com	master
example.net	slave

Dokončit průvodce

V posledním dialogu můžete ve firewallu otevřít port pro DNS a rozhodnout, zda má být DNS server automaticky spouštěn po startu systému. Lze odsud také přejít do expertního režimu konfigurace. Viz obrázek 20.3 – „Instalace DNS serveru: Dokončit průvodce“ (strana 325)).

**Obrázek 20.3** Instalace DNS serveru: Dokončit průvodce



## 20.1.2 Expertní nastavení

V expertním režimu zobrazuje YaST okno s množstvím konfiguračních možností. Jejich nastavením získáte DNS server se všemi základními funkcemi:

**Spuštění**

V položce *Spuštění* nastavte, zda se má DNS server spouštět při startu systému automaticky nebo ručně. Chcete-li DNS server spustit okamžitě, stiskněte tlačítko *Spustit DNS server*. Chcete-li jej zastavit, stiskněte *Zastavit DNS server*. Chcete-li uložit nastavení, stiskněte *Uložit nastavení a restartovat DNS server*.

Port pro DNS můžete na firewallu otevřít zaškrtnutím *Otevřít port na firewallu*. Změnit nastavení firewallu lze po stisknutí tlačítka *Doladění firewallu*.

**Forwardery**

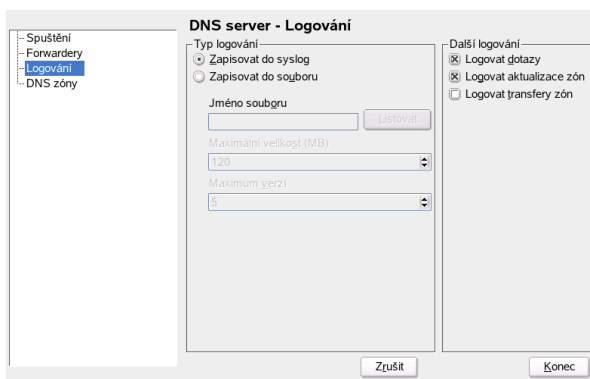
Jedná se o stejný dialog jako je ten, který se objeví po spuštění průvodce (viz kapitola [Instalace DNS serveru: nastavení forwarderů](#) (strana 323)).

## Logování

V této sekci můžete nastavit co a jak má DNS server zapisovat do logů (protokolových souborů). V položce *Typ logování* vyberte kam má DNS server logy zapisovat. Na výběr je mezi systémovým logem `/var/log/messages` (vyberte *Zapisovat do syslog*) a libovolným jiným souborem (vyberte *Zapisovat do souboru*, specifikujte jméno souboru, jeho maximální povolenou velikost a počet verzí souboru, který bude uchovávan).

V položce *Další logování* můžete zaškrtnout následující volby: *Logovat dotazy* zapisuje *veškeré* dotazy klientů, což může způsobit extrémní nárůst velikosti souboru. Proto aktivace této volby bývá rozumná pouze pro účely ladění. Volba *Logovat aktualizace zón* zapisuje datové přenosy při aktualizaci zón mezi DHCP a DNS servery. Chcete-li zapisovat přenosy mezi primárním a sekundárním serverem (master, slave), aktivujte volbu *Logovat transfery zón*. Viz obrázek 20.4 – „DNS server: Logování“ (strana 326).

**Obrázek 20.4** DNS server: Logování



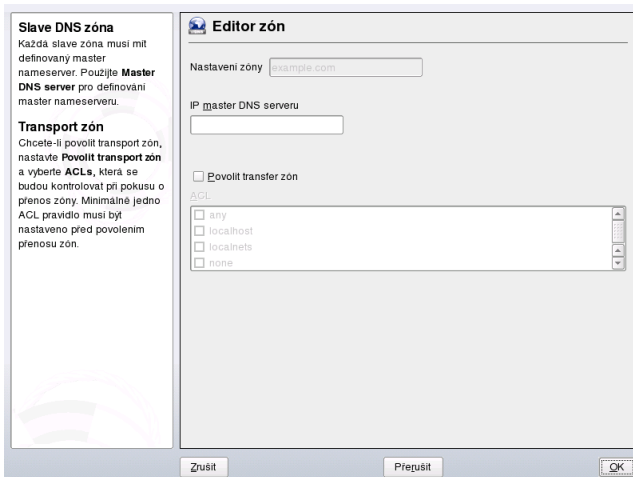
## DNS zóny

Tento dialog je popsán v části věnované průvodci konfigurací. Viz 20.1.1 – „Průvodce konfigurací“ (strana 323).

## Editor slave zón

Tento dialog se objeví, pokud v předchozím dialogu zvolíte možnost *Upravit* pro některou slave zónu. V položce *Master DNS server IP* nastavte IP adresu serveru, ze kterého má slave získávat data. Chcete-li povolit transport zón, zaškrtněte *Povolit transport zón*. Pro omezení přístupu k serveru vyberte ze seznamu ACL. Viz 20.5 – „DNS server: Editor slave zón“ (strana 327).

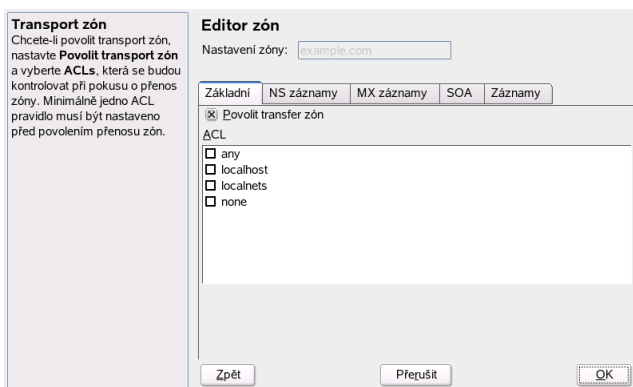
**Obrázek 20.5** DNS server: Editor slave zón



### Editor master zón

Tento dialog se objeví, pokud v dialogu popsaném v části [DNS zóny](#) (strana 326) zvolíte možnost *Upravit* pro některou master zónu. Skládá se z několika karet: *Základní* (ta je otevřená první), *NS záznamy*, *MX záznamy*, *SOA* a *Záznamy*.

**Obrázek 20.6** DNS server: Editor zón (Základní)



### Editor zón (NS záznamy)

V tomto dialogu můžete nastavit alternativní nameservery. Ujistěte se, že je v seznamu uveden i váš vlastní nameserver. Nový nameserver přidáte tak, že zadáte

adresu serveru do pole *Přidat nameserver* a kliknete na *Přidat*. Viz [20.7 – „DNS server: Editor zón \(NS záznamy\)“](#) (strana 328).

**Obrázek 20.7** DNS server: Editor zón (NS záznamy)

The screenshot shows the 'Editor zón' window with the 'NS záznamy' tab selected. On the left, a sidebar contains instructions: 'NS záznamy' and 'Nový nameserver přidáte tak, že zadáte adresu serveru a kliknete na Přidat. Odstranění provedete zvolením serveru, a kliknutím na Smazat.' The main area has a 'Nastavení zóny:' field with 'example.com'. Below are tabs for 'Základní', 'NS záznamy', 'MX záznamy', 'SOA', and 'Záznamy'. The 'Přidat nameserver' section includes a text input field and a 'Přidat' button. The 'Seznam nameserverů' section includes a list area and a 'Smazat' button. At the bottom are 'Zpět', 'Přerušit', and 'OK' buttons.

### Editor zón (MX záznamy)

Chcete-li pro zónu přidat poštovní server, zadejte do příslušných polí jeho adresu a prioritu. Potvrďte stisknutím tlačítka *Přidat*. Viz obrázek [20.8 – „DNS server: Editor zón \(MX záznamy\)“](#) (strana 328).

**Obrázek 20.8** DNS server: Editor zón (MX záznamy)

The screenshot shows the 'Editor zón' window with the 'MX záznamy' tab selected. The sidebar instructions are: 'MX záznamy' and 'Nový poštovní server přidáte tak, že zadáte adresu serveru a jeho prioritu, a kliknete na Přidat. Odstranění provedete zvolením serveru, a kliknutím na Smazat.' The main area has a 'Nastavení zóny:' field with 'example.com'. Below are tabs for 'Základní', 'NS záznamy', 'MX záznamy', 'SOA', and 'Záznamy'. The 'Přidat poštovní server' section includes 'Adresa' and 'Priorita' (a dropdown menu) fields and a 'Přidat' button. The 'Seznam mail relay' section includes a table with columns 'Poštovní server' and 'Priorita', and a 'Smazat' button. At the bottom are 'Zpět', 'Přerušit', and 'OK' buttons.

### Editor zón (SOA)

Na této kartě můžete vytvořit záznamy SOA (*Start Of Authority*). Vysvětlení jednotlivých voleb naleznete v části [20.6 – „Soubor /var/lib/named/world.zone“](#)



(strana 336). Změny SOA záznamů nejsou podporovány pro dynamické zóny spravované přes LDAP.

**Obrázek 20.9** DNS server: Editor zón (SOA)

The screenshot shows the 'Editor zón' window with the 'SOA' tab selected. The window title is 'Editor zón' and the subtitle is 'Nastavení zóny: example.com'. The interface is divided into several sections:

- Nastavení záznamů SOA:** A sidebar on the left with instructions for setting SOA parameters.
- SOA Parameters:** A grid of input fields for 'Série' (2005010602), 'TTL' (2), 'Obnovit' (3), 'Zkusit znovu' (1), 'Vypršení' (1), and 'Minimální' (1). Each field has a 'Jednotka' (Unit) dropdown menu.
- Buttons:** 'Zpět', 'Přerušit', and 'OK' at the bottom.

**SOA Parameter Details:**

Parameter	Value	Unit
Série	2005010602	Hodin
TTL	2	Dni
Obnovit	3	Hodin
Zkusit znovu	1	Hodin
Vypršení	1	Týdnů
Minimální	1	Dni

### Editor zón (Záznamy)

Na této kartě se nastavuje překlad jmen. V položce *Klíč záznamu* zadejte jméno, v rozbalovací nabídce vpravo vyberte jeho typ. *A-Překlad doménového jména* představuje hlavní záznam. Jeho hodnotou by měla být IP adresa. *CNAME* je alias pro doménové jméno. *NS* a *MX* použijte pro záznamy rozšiřující informace zadané na kartách *NS záznamy* a *MX záznamy*. Hodnotou pro poslední tři typy je existující A záznam. Typ *PTR* je určen pro reverzní zóny. Je opakem A záznamu.

## 20.2 Spuštění nameserveru BIND

Nameserver BIND (*Berkeley Internet Name Domain*) je v SUSE Linuxu již předkonfigurovaný, takže ho můžete spustit ihned po instalaci. Pokud máte fungující internetové připojení a do `/etc/resolv.conf` jako adresu nameserveru pro `localhost` vložíte `127.0.0.1`, máte k dispozici překlad jmen na IP adresy bez nutnosti znát IP adresu DNS serveru poskytovatele připojení. BIND tak ale provádí překlad jmen prostřednictvím root nameserveru, což je výrazně pomalejší. Výhodnější je uvést IP adresu DNS serveru poskytovatele do konfiguračního souboru `/etc/named.conf` v položce `forwarders`. Získáte tak efektivní a bezpečný překlad. Takto nastavený nameserver běží v tzv. *caching-only* režimu. Skutečným DNS serverem se stane v případě, že nastavíte příslušné zóny.

---

## Tip: Automatické přizpůsobení informací o nameserveru

Informace o nameserveru lze, v závislosti na typu internetového nebo síťového připojení, automaticky přizpůsobovat aktuální situaci. Tuto vlastnost aktivujete nastavením proměnné `MODIFY_NAMED_CONF_DYNAMICALY` v souboru `/etc/sysconfig/network/config` na `yes`.

---

Nezřizujte však oficiální domény, které nemáte řádně registrovány. Nečiňte tak ani pokud jste sice vlastníky domény, ale tu spravuje poskytovatel, protože BIND nebude forwardovat (přeposílat dále) dotazy na tuto doménu. Takže třeba webový server umístěný u poskytovatele nebude pro vlastní doménu přístupný.

Nameserver může spustit uživatel `root` příkazem `rcnamed start`. Pokud se vpravo zobrazí zeleně `done`, spustil se úspěšně proces nameserveru `named`. Na lokálním počítači je možné fungování nameserveru ihned vyzkoušet programy `host` nebo `dig`, které by jako výchozí server měly vrátit `localhost` s adresou `127.0.0.1`. Pokud tomu tak není, pak je pravděpodobně v `/etc/resolv.conf` uveden špatný nameserver nebo tento soubor vůbec neexistuje. Zkuste příkaz `host 127.0.0.1`, který by měl fungovat vždy. Pokud se zobrazí chybové hlášení, otestujte příkazem `rcnamed status`, zda `named` vůbec běží. Jestliže nameserver není spuštěn nebo vykazuje chybné chování, naleznete obvykle příčinu v protokolovém souboru `/var/log/messages`.

Chcete-li používat nameserver poskytovatele nebo vlastní nameserver běžící ve vlastní síti jako forwarder, pak je třeba v části `options` mezi `forwarders` uvést jeho/jejich IP adresy. Adresy uvedené v příkladu 20.1 – „Volby pro přeposílání v souboru `named.conf`“ (strana 330) jsou pouze ukázkové.

### **Rovnice 20.1** Volby pro přeposílání v souboru `named.conf`

```
options {
    directory "/var/lib/named";
    forwarders { 10.11.12.13; 10.11.12.14; };
    listen-on { 127.0.0.1; 192.168.0.99; };
    allow-query { 127/8; 192.168.0/24; };
    notify no;
};
```

Položka `options` je následována položkami pro jednotlivé zóny, `localhost`, `0.0.127.in-addr.arpa` a položkou `type hint` pod `.`, která by měla být vždy přítomná. Příslušné soubory není nutno měnit a měly by pracovat tak, jak jsou. Ujistěte se, že je každá položka ukončena znakem `;`, a že jsou správně umístěny složené závorky.

Změníte-li soubor `/etc/named.conf` nebo soubor zóny, přikážete programu BIND pomocí příkazu `rndc reload`, aby soubor znovu načetl. Dosáhnete toho také zastavením a novým spuštěním serveru příkazem `rndc restart`. Server můžete zastavit také příkazem `rndc stop`.

## 20.3 Konfigurační soubor `/etc/named.conf`

Všechna nastavení pro BIND se provádějí v souboru `/etc/named.conf`. Nicméně data pro zóny, jako názvy počítačů, IP adresy atd. jsou uloženy v separátních souborech v adresáři `/var/lib/named`. Bližší informace jsou uvedeny v následujícím textu.

Konfigurační soubor `/etc/named.conf` se dělí na dvě oblasti. Obecná nastavení jsou v části `options`, v části `zone` jsou položky pro jednotlivé domény. Kromě toho je zde volitelně také oblast `logging` a položky typu `acl` (Access Control List). Komentáře začínají znakem `#` či znaky `//`. Minimalistický `/etc/named.conf` je uveden v příkladu 20.2 – „Jednoduché nastavení souboru `/etc/named.conf`“ (strana 331).

### **Rovnice 20.2** *Jednoduché nastavení souboru `/etc/named.conf`*

```
options {
    directory "/var/lib/named";
    forwarders { 10.0.0.1; };
    notify no;
};

zone "localhost" in {
    type master;
    file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone";
};

zone "." in {
    type hint;
    file "root.hint";
};
```

## 20.4 Nejdůležitější konfigurační volby v sekci options

`directory "adresar";`

Udává adresář, ve kterém BIND hledá soubory s daty o jednotlivých zónách. Obvykle je to adresář `/var/lib/named`.

`forwarders { IP adresa; };`

Určuje IP adresy jednoho nebo více nameserverů (většinou nameserverů poskytovatele), na které jsou DNS dotazy přeposílány v případě, že je není možné zodpovědět přímo. Řetězec `IP adresa` nahraďte IP adresou, např. `10.0.0.1`.

`forward first;`

Tato volba způsobuje, že je DNS dotaz ihned, před dotazováním na root nameserveru, přeposílán. Místo `forward first` lze použít `forward only`, pak nebude root nameserver dotazován vůbec.

`listen-on port 53 { 127.0.0.1; IP adresa; };`

Tato položka sděluje BINDu, na kterém síťovém rozhraní a portu má poslouchat dotazy klientů. `port 53` je standardní a není třeba jej explicitně uvádět. Zadáním adresy `127.0.0.1` povolíte dotazy z počítače `localhost`. Pokud je tato položka zcela vynechána, jsou standardně použita všechna rozhraní.

`listen-on-v6 port 53 {any; };`

Tato položka sděluje BINDu, aby naslouchal klientským požadavkům přes protokol IPv6. Jedinou alternativou k `any` je `none` (nenaslouchat IPv6 požadavkům). Server akceptuje pouze IPv6 adresy typu `wild card`.

`query-source address * port 53;`

Tato volba se používá pokud firewall blokuje externí DNS dotazy. BIND pak komunikuje přes port 53 a ne přes porty vyšší než 1024.

`query-source-v6 address * port 53;`

Tato volba určuje, jaký port má být použit pro IPv6 dotazy.

`allow-query { 127.0.0.1; s1 ; };`

Tato volba určuje sítě, ze kterých mohou klienti posílat DNS dotazy. Řetězec `s1` nahraďte adresou, např. `192.168.1/24`. Číslo `/24` je zkrácený zápis síťové masky `255.255.255.0`.

allow-transfer { ! \*; };

Tato volba řídí, které počítače mohou požadovat transfer zóny. V uvedeném příkladu jsou takové požadavky zcela zakázány pomocí ! \*. Pokud by zde tato položka nebyla, bylo by možné provádět transfer zóny odkudkoliv a bez omezení.

statistics-interval 0;

Bez této položky generuje BIND každou hodinu několik řádků do protokolového souboru `/var/log/messages`. Nula potlačuje tento výstup, jinak je možné uvádět čas v minutách.

cleaning-interval 720;

Tato položka určuje, v jakém časovém odstupu bude BIND mazat svou cache (vyrovňovací paměť). Smazání cache vždy vygeneruje zápis do `/var/log/messages`. Čas se udává v minutách a výchozí hodnotou je 60 minut

interface-interval 0;

BIND pravidelně prohledává síťová rozhraní a hledá nová či odpojená rozhraní. Nula zamezí tomuto hledání a BIND bude pracovat pouze s rozhraními, která nalezne při startu. Čas se udává v minutách a výchozí hodnotou je 60 minut.

notify no;

Parametr `no` zabraňuje informování ostatních nameserverů při změně data pro zónu nebo restartu nameserveru.

## 20.5 Konfigurace v sekci logging

BIND má široké možnosti protokolování (logování) různých událostí. Výchozí nastavení by mělo vyhovovat ve většině případů. Příklad [20.3 – „Položka zakazující protokolování“](#) (strana 333) obsahuje nejjednodušší možnou formu nastavení a zakazuje logování zcela:

### **Rovnice 20.3** *Položka zakazující protokolování*

```
logging {  
    category default { null; };  
};
```

## 20.6 Struktura souboru odkazujícího na data pro zóny

### **Rovnice 20.4** *Data zóny moje-domena.cz*

```
zone "moje-domena.cz" in {  
    type master;  
    file "moje-domena.zone";  
    notify no;  
};
```

Za `zone` je uveden název spravované domény, zde tedy `moje-domena.cz`, následovaný `in` a složenými závorkami, které obsahují volby pro tuto zónu (viz 20.5 – „Data zóny jina-domena.cz“ (strana 334)). Chcete-li definovat sekundární (*slave zone*), změňte `type` na `slave` a uveďte `nameserver`, který spravuje zónu jako `master` (ale sám může být `slave` jiného serveru).

### **Rovnice 20.5** *Data zóny jina-domena.cz*

```
zone "jina-domena.cz" in {  
    type slave;  
    file "slave/jina-domena.zone";  
    masters { 10.0.0.1; };  
};
```

Volby pro nastavení zón:

`type master;`

Volba `master` určuje, že je zóna spravována lokálním `nameserverem`. To předpokládá správně vytvořený soubor pro zónu.

`type slave;`

Zóna je transferována z jiného `nameserveru`. Volba musí být použita společně s volbou `masters`.

`type hint;`

Zóna `type hint` se používá pro specifikaci `root nameserveru`. Můžete ponechat výchozí nastavení.

`file "moje-domena.zone" nebo "slave/jina-domena.zone";`

Tato volba specifikuje soubor, ve kterém jsou uložena data pro doménu. V případě zóny typu `slave` není potřeba, neboť potřebné údaje jsou získány z jiného `name-`

serveru. Aby byly primární (master) a sekundární (slave) soubory odlišeny, používá se pro sekundární soubory zvláštní adresář `slave`.

```
masters { IP adresa serveru; };
```

Tuto položku je třeba uvádět pouze u sekundárních (slave) zón. Specifikuje name-server, ze kterého jsou získávána data o zóně.

```
allow-update { ! *; };
```

Tato volba určuje práva externích uživatelů pro zápis do konfigurace. To je obvykle z bezpečnostních důvodů nevhodné. Chybí-li tato položka, nebo je-li použit zápis uvedený výše, je zápis zakázán.

## 20.7 Struktura souboru s daty pro zónu

Používají se dva druhy souborů s daty zóny. Jedny slouží pro přiřazení IP adresy počítačům a druhé pak pro reverzní převod, tedy pro přiřazení názvu počítače k IP adrese.

---

### Tip: Použití tečky v souborech s daty zóny

V souborech s daty zóny má velký význam tečka (.). Jsou-li názvy počítačů uvedeny bez tečky na konci, je vždy doplňována zóna. Proto je třeba již kompletní názvy počítačů uvedené i s doménou ukončit tečkou tak, aby nebyla doména uvedena dvakrát. Chybějící tečky nebo jejich špatné umístění jsou často příčinou chyb v konfiguraci nameserveru.

---

Ukážeme si soubor `world.zone` odpovědný za doménu `world.cosmos`:

## Rovnice 20.6 Soubor /var/lib/named/world.zone

```
$TTL 2D
world.cosmos. IN SOA      gateway root.world.cosmos. (
                    2003072441 ; serial
                    1D        ; refresh
                    2H        ; retry
                    1W        ; expiry
                    2D )      ; minimum

                    IN NS      gateway
                    IN MX      10 sun

gateway IN A      192.168.0.1
        IN A      192.168.1.1
sun     IN A      192.168.0.2
moon    IN A      192.168.0.3
earth   IN A      192.168.1.2
mars    IN A      192.168.1.3
www     IN CNAME   moon
```

### Řádek 1:

§TTL definuje standardní délku platnosti TTL (*Time To Live*), která platí pro všechny položky v tomto souboru. V našem případě jsou to dva dny (2D).

### Řádek 2:

Zde začíná SOA záznam:

- Na prvním místě je uveden název spravované domény `world.cosmos` ukončený tečkou (jinak by zóna byla přidána ještě jednou. Alternativním řešením je použití zavináče (@), který znamená použití zóny z `/etc/named.conf`).
- Za `IN SOA` je uveden název primárního (*master*) nameserveru pro danou zónu. Jméno `gateway` bude rozšířeno na `gateway.world.cosmos`, protože není ukončeno tečkou.
- Následuje e-mailová adresa osoby odpovědné za nameserver. Protože zavináč má v tomto souboru zvláštní význam, používá se místo něj tečka. Adresa `root@world.cosmos` se tedy zapíše jako `root.world.cosmos.` Na konci je opět nutné uvést tečku.
- Řádka končí levou závorkou (, která uzavírá, spolu s následující pravou závorkou ), řádky tvořící SOA záznam.



Řádek 3:

Obsahuje tzv. sériové číslo (*serial number*), které se má při každé změně v souboru zvýšit. Slouží sekundárním nameserverům pro porovnávání konfigurace s primárním nameserverem. Jako formát čísla se ujal YYYYMMDDNN.

Řádek 4:

Položka `refresh rate` udává časový interval, po jehož uplynutí sekundární server kontroluje `serial number` na primárním serveru. V našem případě jeden den (1D).

Řádek 5:

Položka `retry rate` udává časový interval, po jehož uplynutí se sekundární server opět pokusí kontaktovat primární server v případě, že se původní kontakt z důvodu chyby neuskutečnil. Zde dvě hodiny (2H).

Řádek 6:

Položka `expiration time` udává dobu, po jejímž uplynutí sekundární name-server smaže data z cache, pokud nemůže kontaktovat primární server. Zde jeden týden (1W).

Řádek 7:

Poslední SOA položka určuje tzv. `negative caching TTL`, čas po který mají ostatní servery uchovávat v cache negativně vyřízené dotazy.

Řádek 9:

Položka `IN NS` udává nameserver odpovědný za doménu. Také zde platí, že `gateway` expanduje na `gateway.world.cosmos`, protože je bez tečky na konci. Řádků podobných tomuto může být více, jeden pro primární a další pro sekundární nameservery. Pokud není `notify` v souboru `/etc/named.conf` nastaven na `no`, pak budou všechny zde uvedené nameservery informovány o změnách dat zóny.

Řádek 10:

`MX` záznam určuje poštovní server pro doménu `world.cosmos`. Tento server poštu přijímá a dále zpracovává, resp. přeposílá. V uvedeném příkladě to je server `sun.world.cosmos`. Kromě názvu serveru se uvádí preferenční hodnota (zde 10) — v případě většího počtu `MX` položek bude pošta zaslána serveru s nejnižším číslem a teprve při problémech s doručením bude použit server s vyšší hodnotou.

Řádky 12 až 17:

Zde jsou uvedeny vlastní adresní záznamy přiřazující jménům počítačů IP adresy. Názvy počítačů jsou uváděny bez tečky a budou tak rozšířeny o doménu. Více IP adres se používá u počítačů, které mají více síťových karet. Pokud je použita tradiční (IPv4) adresa, je záznam označen písmenem A. Záznamy s IPv6 adresou jsou označeny jako A6. (Dříve se IPv6 adresy označovaly jako AAAA, což je již zastaralé.)

Řádek 18:

Alias `www` je použit k adresování počítače `moon` (CNAME = *canonical name*).

Pro reverzní převod (*reverse lookup*) IP adres na názvy počítačů se používá pseudodomena `in-addr.arpa`. Je připojena k obrácenému zápisu adresy. Ze `192.168.1` se tak stane `1.168.192.in-addr.arpa`, viz příklad 20.7 – „Zpětný převod“ (strana 338).

### Rovnice 20.7 Zpětný převod

```
$TTL 2D
1.168.192.in-addr.arpa. IN SOA gateway.world.cosmos. root.world.cosmos. (
                                2003072441      ; serial
                                1D                ; refresh
                                2H                ; retry
                                1W                ; expiry
                                2D )              ; minimum

                                IN NS            gateway.world.cosmos.

1                               IN PTR        gateway.world.cosmos.
2                               IN PTR        earth.world.cosmos.
3                               IN PTR        mars.world.cosmos.
```

Řádek 1:

Položka `$TTL` definuje standardní délku platnosti TTL (*Time To Live*), která platí pro všechny položky v tomto souboru. V našem případě jsou to dva dny (2D).

Řádek 2:

Reverzní převod je nastaven pro síť `192.168.1.0`. Protože se zde zóna nazývá `1.168.192.in-addr.arpa`, nechceme ji připojovat za názvy počítačů, a proto je píšeme celé včetně domény a s tečkou na konci.

Řádek 3-7:

Viz předchozí příklad pro `world.cosmos`.

Řádek 9:

I zde je uveden nameserver, který odpovídá za zónu. Tentokrát je uveden včetně domény a s tečkou na konci.

Řádek 11-13:

Pointer záznamy, které uvádějí k IP adrese náležející názvy počítačů. Uvádí se pouze poslední pozice IP adresy bez tečky. Připojením zóny (bez `.in-addr.arpa`) vznikne kompletní IP adresa v obráceném pořadí.

Přenosy zón mezi různými verzemi BINDu by měly být bezproblémové.

## 20.8 Dynamická aktualizace údajů o zóně

Termín *dynamická aktualizace* se vztahuje na mechanismy, kterými jsou záznamy v souborech zón na primárním (master) serveru přidávány, měněny nebo mazány. Tyto mechanismy jsou popsány v dokumentu RFC 2136. Dynamická aktualizace je pro každou zónu nastavována individuálně přidáním volitelného pravidla `allow-update` nebo `update-policy`. Dynamicky aktualizované zóny by neměly být upravovány ručně.

Záznamy, které se mají na serveru aktualizovat, přenesete příkazem `nsupdate`. Přesná syntaxe je popsána v manuálové stránce (`man 8 nsupdate`). Z bezpečnostních důvodů by všechny aktualizace měly být prováděny s využitím TSIG klíčů popsaných v kapitole 20.9 – „Bezpečné transakce“ (strana 339).

## 20.9 Bezpečné transakce

Bezpečné transakce lze zajistit pomocí transakčních signatur (TSIG) založených na sdílených tajných klítech (TSIG klítech). V této sekci je popsáno, jak tyto klíče vytvořit a používat.

Bezpečné transakce jsou potřeba pro komunikaci mezi různými servery a pro dynamickou obnovu zónových dat. Kontrola pomocí klíčů je mnohem bezpečnější než pouhá kontrola pomocí IP adres.

TSIG klíč můžete vygenerovat následujícím příkazem (podrobnosti viz `man dnssec-keygen`):

```
dnssec-keygen -a hmac-md5 -b 128 -n HOST host1-host2
```

Tím se vytvoří dva soubory se jmény podobnými následujícím:

```
Khost1-host2.+157+34265.private Khost1-host2.+157+34265.key
```

Samotný klíč (např. řetězec `ejIkuCyyGJwwuN3xAteKgg==`) se nachází v obou souborech. Aby mohl být používán pro transakce, musí být druhý soubor (`Khost1-host2.+157+34265.key`) přenesen na vzdálený počítač (nejlépe bezpečnou cestou, např. pomocí `scp`). Na vzdáleném serveru musí být vložen do souboru `/etc/named.conf`, čímž se umožní bezpečná komunikace mezi oběma počítači (`host1` a `host2`):

```
key host1-host2. {
    algorithm hmac-md5;
    secret "ejIkuCyyGJwwuN3xAteKgg==";
};
```

---

### **Varování: Přístupová práva k `/etc/named.conf`**

Ujistěte se, že přístupová práva k souboru `/etc/named.conf` jsou správně nastavena (a omezena). Výchozí práva pro tento soubor jsou `0640`, vlastníkem souboru je `root` a skupina je `named`. Jinou možností je přesunout klíče do jiného souboru s patřičně nastavenými právy, který se pak do souboru `/etc/named.conf` vkládá.

---

Aby mohl server `host1` používat klíč pro `host2` (jehož adresa je `192.168.2.3`), musí soubor `/etc/named.conf` na serveru obsahovat následující pravidlo:

```
server 192.168.2.3 {
    keys { host1-host2. ;};
};
```

Obdobné nastavení je třeba učinit i v konfiguračních souborech na počítači `host2`.

Kromě seznamů správy přístupu (ACL, *Access Control Lists* — neplést s ACL souborového systému) definovaných pro jednotlivé IP adresy a rozsahy adres přidejte pro zvýšení bezpečnosti TSIG klíče. Příslušný záznam v konfiguraci by měl vypadat asi takto:

```
allow-update { key host1-host2. ;};
```

K tomuto tématu naleznete více informací v příručce *BIND Administrator Reference Manual* v části `update-policy`.

## 20.10 DNSSEC

DNSSEC, bezpečné DNS, je popsáno v RFC 2535. Nástroje pro práci s DNSSEC jsou probírány v BIND manuálu.

Bezpečná zóna musí mít přiřazen jeden nebo více zónových klíčů, generovaných pomocí `dnssec-keygen`, stejně jako klíče počítačů. V současnosti se pro tvorbu klíčů používá algoritmus DES. Veřejné klíče by měly být vloženy do příslušného zónového souboru pomocí pravidla `$INCLUDE`.

Příkazem `dnssec-makekeyset` jsou všechny klíče spojeny do jedné sady, která pak musí být bezpečným způsobem přenesena do rodičovské (nadřazené) zóny. Tam je sada podepsána pomocí `dnssec-signkey`. Soubory generované tímto příkazem jsou použity k podepsání zón pomocí `dnssec-signzone`, čímž jsou vytvořeny soubory, které se vloží do `/etc/named.conf` každé zóny.

## 20.11 Další informace

Další informace naleznete v příručce *BIND Administrator Reference Manual* nainstalované v adresáři `/usr/share/doc/packages/bind/`. Zvažte i studium RFC dokumentů zmiňovaných v tomto manuálu a příslušných manuálových stránek. Soubor `/usr/share/doc/packages/bind/README` . SuSE obsahuje aktuální informace o BINDu v systému SUSE Linux.



# NIS — Network Information Service **21**

Jakmile přistupuje v síti více unixových počítačů ke společným prostředkům, je třeba zajistit, aby bylo všude společné označení uživatelů a skupin. Síť musí být pro každého uživatele transparentní – ať pracuje na kterémkoli z těchto počítačů, vždy by měl najít stejné prostředí. To umožňují služby *NIS* a *NFS*. *NFS* slouží pro přístup k souborovým systémům přes síť a je popsán v kapitole 22 – „*NFS — sdílené souborové systémy*“ (strana 349).

*NIS* (Network Information Service) je databázová služba poskytující síťový přístup k obsahu souborů `/etc/passwd`, `/etc/shadow` a `/etc/group`. *NIS* lze použít i k dalším účelům (např. pro zpřístupnění souborů `/etc/hosts` nebo `/etc/services`), ale to je nad rámec tohoto textu. *NIS* se také často nazývá *YP* (žluté nebo zlaté stránky).

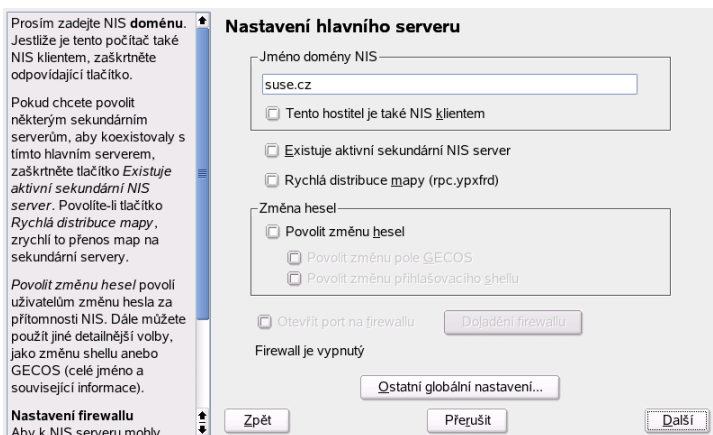
## 21.1 Konfigurace *NIS* serveru

Konfiguraci zahájíte výběrem YaST modulu *NIS server* v části *Síťové služby*. Pokud ve vaší síti dosud neexistuje žádný *NIS* server, zvolte v dialogu *Instalovat a nastavit NIS hlavní server*. Pokud již *NIS* server máte (hlavní server, master), můžete přidat sekundární (slave) *NIS* server (např. pro konfiguraci nové podsítě). Nejprve popíšeme konfiguraci hlavního serveru.

Pokud některé balíčky chybí, vložte instalační zdroj, doinstalují se automaticky. V horní části dialogu (viz 21.1 – „*Nástroj pro nastavení NIS serveru*“ (strana 344)) zadejte jméno domény. Zatřetí položky *Tento hostitel je také NIS klientem* zvolte, zda má

být server zároveň i NIS klientem (to umožňuje uživatelům přihlašovat se a přistupovat k datům z NIS serveru).

**Obrázek 21.1** Nástroj pro nastavení NIS serveru



Pokud chcete později v síti vytvořit sekundární NIS server (slave), nezapomeňte zaškrtnout tlačítko *Existuje aktivní sekundární NIS server*. Kromě toho byste měli zapnout i položku *Rychlá distribuce mapy*, která zajistí velmi rychlý přenos informací z hlavního (master) NIS serveru na sekundární (slave).

Jestliže chcete uživatelům v síti (lokálním i spravovaným pomocí NIS serveru) povolit změnu vlastních hesel uložených na NIS serveru (příkazem `yppasswd`), vyberte *Povolit změnu hesel*. *Povolit změnu pole GECOS* umožní uživatelům měnit i nastavení jména a adresy (příkazem `ypchfn`). *Povolit změnu přihlašovacího shellu* dovoluje uživatelům zvolit přihlašovací shell příkazem `ypchsh` (např. `sh` místo `bash`).

Tlačítkem *Ostatní globální nastavení* přejdete do dialogu *Nastavení detailů hlavního serveru NIS* (viz obrázek 21.2 – „Změna adresáře a synchronizace souborů NIS serveru“ (strana 345)), kde můžete změnit zdrojový adresář NIS serveru (výchozím adresářem je `/etc`). Aby byly synchronizovány soubory `/etc/passwd` a `/etc/shadow` nebo `/etc/group` a `/etc/gshadow`, zvolte *Ano*. Nastavit můžete i minimální ID uživatele a skupiny. Nastavení potvrdíte kliknutím na tlačítko *OK*. Vráťte se do původního dialogu, kde můžete pokračovat stisknutím tlačítka *Další*.



**Obrázek 21.2** Změna adresáře a synchronizace souborů NIS serveru

Můžete změnit zdrojový adresář NIS serveru (většinou `/etc`).

Vyberte pokud mají být sloučeny soubory `passwd` s `shadow` a `group` s `gshadow`. Možné pouze pokud soubory `shadow` nebo `gshadow` existují.

Můžete taktéž upravit minimální ID uživatele a skupiny.

**Nastavení detailů hlavního serveru NIS**

Zdrojový adresář YP  
/etc

Sloučit hesla  
 Ne  
 Ano

Sloučit skupiny  
 Ne  
 Ano

Minimální UID  
500

Minimální GID  
500

Zpět Přesuň OK

Pokud jste předtím aktivovali tlačítko *Existuje aktivní sekundární NIS server*, pak je třeba nyní uvést název či názvy počítačů, které budou fungovat jako sekundární servery. Pokračujte tlačítkem *Další*. Pokud sekundární servery nepoužíváte, je tento dialog vynechán.

V dalším dialogu můžete upravit mapy, které budou z NIS serveru přeneseny na klienty. Výchozí nastavení většinou není třeba měnit.

Stisknutím tlačítka *Další* se přenesete do dalšího dialogu (viz 21.3 – „Nastavení přístupových práv k NIS serveru“ (strana 346)). V něm nastavte, které sítě mohou přistupovat k NIS serveru. Obvykle je to vaše vnitřní síť. V takovém případě nastavte následující dvě položky:

```
255.0.0.0      127.0.0.0
0.0.0.0        0.0.0.0
```

První položka umožňuje přístup z vašeho počítače (NIS serveru). Druhá umožňuje přístup každému, kdo má přístup do lokální sítě.

**Obrázek 21.3** Nastavení přístupových práv k NIS serveru

Prosím zadejte, kterým hostitelům je povoleno dotazovat se NIS serveru.

Adresa bude povolena, pokud **síť** odpovídá bitovému **AND** **adresy hostitele** a **síťové masky**.

Musí existovat položka se **síťovou maskou** 255.0.0.0 a **síťí** 127.0.0.0, pokud mají být povolena spojení z localhost.

Pokud je zadána **síťová maska** 0.0.0.0 a **síť** 0.0.0.0, povolíte tím přístup všem hostitelům.

### Nastavení dotazů hostitelů na NIS servery

Síťová maska podsítě	Síť
255.0.0.0	127.0.0.0
0.0.0.0	0.0.0.0

Přidat   Editovat   Smazat

Zpět   Přeskočit   Konec

---

### Důležité: Automatické nastavení firewallu

Pokud máte aktivovaný firewall (SuSEfirewall2) a zvolili jste *Otevřít port na firewallu*, YaST upraví nastavení firewallu pro NIS server povolením `portmap` služby.

---

## 21.2 Konfigurace NIS klientů

Pro konfiguraci NIS klienta použijte YaST modul *Klient NIS*. Zvolíte-li používání NIS nebo, v závislosti na okolnostech, automounter, otevře se tento dialog. Zvolte, zda má stanice pevnou IP adresu nebo zda ji má získat z DHCP serveru. DHCP server nastaví také NIS doménu a NIS server. Více informací o DHCP najdete v části [23 – „DHCP“](#) (strana 355). V případě používání pevné IP adresy nastavte NIS doménu a NIS server ručně (viz [21.4 – „Nastavení domény a adresy NIS serveru“](#) (strana 347)). NIS server v síti můžete vyhledat pomocí volby *Najít*.

Zadat lze i více domén s tím, že jedna bude nastavena jako výchozí. K zadání dalšího serveru použijte tlačítko *Upravit*.

Aby nebylo možné z jiného počítače zjistit, který NIS server vaše stanice používá, za-  
kažte v expertním nastavení volbu *Odpovídat vzdáleným počítačům*. Pokud zvolíte

Poškozený server, může klient přijímat odpovědi serveru na neprivilegovaném portu. Další informace získáte v manuálové stránce `ypbind`.

**Obrázek 21.4** Nastavení domény a adresy NIS serveru

Zadejte svou NIS doménu (např. 'foo.com') a adresu NIS serveru (např. 'nis.foo.com' nebo 10.20.1.1).

Můžete určit více serverů, pokud oddělíte jejich adresy mezerami.

Volba **Broadcast** umožňuje hledat v lokální síti server poté, co zadané servery neodpověděly. Je to ovšem bezpečnostní riziko.

Pokud používáte **DHCP** a server poskytuje NIS doménu nebo servery, můžete zde povolit jejich použití. Samotné DHCP můžete nakonfigurovat v síťovém modulu.

Automounter je démon, který automaticky připojuje adresáře. Předpokládá se, že jeho konfigurační soubory

### Konfigurace klienta NIS

Nepoužívat NIS  
 Používat NIS

Klient NIS

Automatické nastavení (pomocí DHCP)  
 Statické nastavení

NIS doména  
suse.cz

Adresy NIS serverů  
10.20.0.2

Broadcast

Další NIS domény

Spustit automatické připojení



# NFS — sdílené souborové systémy **22**

Jak již bylo uvedeno v kapitole 21 – „*NIS — Network Information Service*“ (strana 343), NFS (spolu s NIS) zabezpečují transparentnost sítě pro uživatele. NFS umožňuje počítačům sdílet souborové systémy v síti – uživatel pak vidí stejné prostředí nezávisle na tom, odkud se přihlásí.

Podobně jako NIS, představuje i NFS nesymetrickou službu – rozlišuje se NFS server a NFS klient. Počítač může vykonávat obě tyto úlohy, tj. exportovat do sítě své vlastní souborové systémy a připojovat (mount) souborové systémy jiných počítačů. Centrální server NFS mívá obvykle velkou diskovou kapacitu. Jednotliví klienti si z něho připojují povolené adresářové stromy ke svému souborovému systému.

---

## **Důležité: Potřeba DNS**

Teoreticky lze export provádět pouze pomocí IP adres. Abyste zabránili prodlevám, potřebujete funkční DNS systém. Je to potřeba minimálně pro účely logování, neboť mountd démon provádí reverzní překlady.

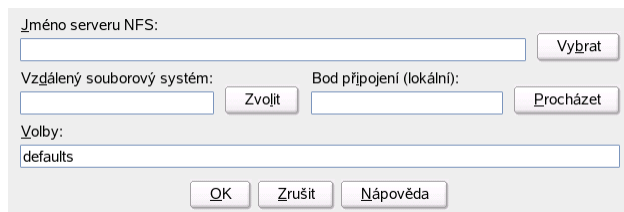
---

## **22.1 Importování souborových systémů pomocí YaST2**

Každý oprávněný uživatel může připojit NFS adresáře ke svému systému. Nejjednodušší je použít YaST modul *Klient NFS*. Zvolte *Přidat* a uveďte potřebné informace: jméno NFS serveru, adresář, který chcete importovat, a bod připojení (adresář), ve kterém se

importovaná data zobrazí. Viz [22.1 – „Nastavení NFS klienta v programu YaST“](#) (strana 350).

**Obrázek 22.1** *Nastavení NFS klienta v programu YaST*



The screenshot shows a graphical user interface for configuring an NFS client. It features several input fields and buttons:

- Jméno serveru NFS:** A text input field with a "Vybrat" (Select) button to its right.
- Vzdálený souborový systém:** A text input field with a "Zvolit" (Choose) button to its right.
- Bod připojení (lokální):** A text input field with a "Procházet" (Browse) button to its right.
- Volby:** A text input field containing the text "defaults".
- At the bottom, there are three buttons: "OK", "Zrušit" (Cancel), and "Nápověda" (Help).

## 22.2 Ruční import souborových systémů

Importovat systém souborů ze serveru NFS je snadné. Jediným předpokladem je, aby běžel RPC portmapper (může ho spustit uživatel `root` příkazem `rpcportmap start`). Je-li tento předpoklad splněn, lze vzdálené souborové systémy připojovat stejně snadno jako lokální souborové systémy příkazem `mount` s následující syntaxí:

```
mount jmeno-serveru:vzdalena-cesta lokalni-cesta
```

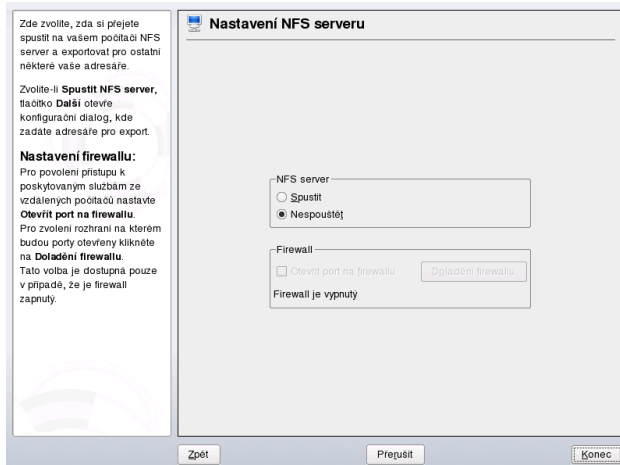
Uživatelské adresáře ze serveru `sun` se například importují následujícím příkazem:

```
mount sun:/home /home
```

## 22.3 Exportování souborových systémů pomocí YaST

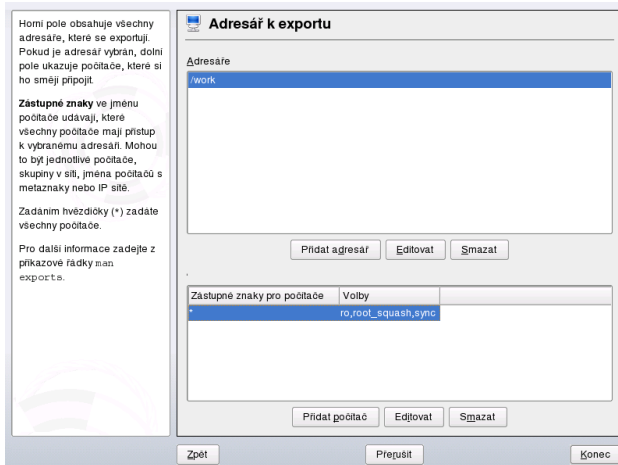
S pomocí programu YaST můžete svůj počítač proměnit v NFS server – server exportující adresáře a soubory na všechny ostatní počítače s povoleným přístupem. Lze tak poskytnout aplikace všem účastníkům v síti, aniž by bylo nutné tyto aplikace instalovat na jednotlivé pracovní stanice. Server nainstalujete tak, že spustíte YaST a zvolíte *Sítové služby* → *NFS server*. Objeví se dialog zobrazený na obrázku [22.2 – „Nástroj pro nastavení NFS serveru“](#) (strana 351)).

## Obrázek 22.2 Nástroj pro nastavení NFS serveru



V dialogu zvolte položku *Spustit NFS server* a klikněte na tlačítko *Další*. V horním poli se zadávají soubory a adresáře k exportu. Dolní pole je určeno pro seznam počítačů s povoleným přístupem. Dialog je zobrazen na obrázku 22.3 – „Nastavení NFS serveru v programu YaST“ (strana 352). Klientské počítače lze specifikovat čtyřmi způsoby: jako jednotlivý počítač, skupinu v síti, jméno počítače s metaznakem nebo IP síť. Podrobný popis najdete v manuálové stránce `exports`. Nastavení dokončíte kliknutím na *Konec*.

**Obrázek 22.3** Nastavení NFS serveru v programu YaST



---

### Důležité: Automatické nastavení firewallu

Pokud máte aktivovaný firewall (SuSEfirewall2) a zvolili jste *Otevřít port na firewallu* v prvním dialogu, YaST automaticky upraví nastavení firewallu a povolí službu `nfs`.

---

## 22.4 Ruční export souborových systémů

Pokud nechcete pro konfiguraci NFS serveru použít YaST, ujistěte se, že na NFS serveru běží následující systémy:

- RPC portmapper (`portmap`)
- RPC mount démon (`rpc.mountd`)
- RPC NFS démon (`rpc.nfsd`)

Aby se tyto služby spouštěly při startu systému automaticky pomocí skriptů `/etc/init.d/portmap` a `/etc/init.d/nfsserver`, zadejte příkazy `insserv /etc/init.d/nfsserver` a `insserv /etc/init.d/portmap`.



V konfiguračním souboru `/etc/exports` určete, které souborové systémy mají být exportovány kterým počítačům.

Pro každý exportovaný adresář je potřeba jeden řádek, na kterém jsou specifikovány počítače, kterým se má exportovat, a jejich oprávnění. Automaticky jsou exportovány i všechny podadresáře. Oprávněné počítače se obvykle zadávají plnými jmény, včetně domény. Také je možno použít zástupné znaky (wildcards) jako `*` a `?` (chovají stejně jako v `bash`). Pokud se nezadá žádný počítač, mohou adresář importovat všechny počítače, podle zadaných přístupových práv.

Přístupová práva se zadávají do závorek za jméno počítače. Nejdůležitější volby jsou ukázány v tabulce [22.1 – „Přístupová práva k exportovaným souborům“](#) (strana 353).

**Tabulka 22.1** *Přístupová práva k exportovaným souborům*

volba	význam
<code>ro</code>	Souborový systém se exportuje pouze pro čtení (výchozí).
<code>rw</code>	Souborový systém se exportuje pro čtení i zápis.
<code>root_squash</code>	Uživatel <code>root</code> daného počítače nemá rootovská práva pro tento souborový systém. Dosáhne se toho změnou <code>user-ID</code> 0 na <code>user-ID</code> 65534, a to se přiřadí uživateli <code>nobody</code> (výchozí volba).
<code>no_root_squash</code>	Zachovat rootovská práva (opak předchozího).
<code>link_relative</code>	Nahradit absolutní symbolické odkazy (začínající <code>/</code> ) odpovídající posloupností <code>../</code> . Tato volba má smysl jen tehdy, je-li připojen úplný systém souborů počítače (výchozí volba).
<code>link_absolute</code>	Symbolické odkazy zůstávají nezměněny.
<code>map_identity</code>	Na klientovi budou stejná uživatelská ID jako na serveru (výchozí volba)
<code>map_daemon</code>	Klient a server nemají odpovídající ID uživatelů. To se sdělí programu <code>nfsd</code> , aby vytvořil konverzní tabulku pro ID. Předpokladem je spuštění démona <b>ugidd</b>

Soubor `exports` může vypadat například tak, jak je uvedeno v příkladu [22.1 – „/etc/exports“](#) (strana 354). Soubor `/etc/exports` je používán démony `mountd` a `nfsd`. Pokud soubor změňte, `mountd` a `nfsd` restartujte příkazem `rcnfsserver restart`.

**Rovnice 22.1** */etc/exports*

```
#
# /etc/exports
#
/home                sun(rw)   venus(rw)
/usr/X11             sun(ro)   venus(ro)
/usr/lib/texmf       sun(ro)   venus(rw)
/                   earth(ro,root_squash)
/home/ftp            (ro)
# End of exports
```

# DHCP

## 23.1 DHCP protokol

Protokol DHCP (*Dynamic Host Configuration Protocol*) umožňuje centrální nastavení sítě na serveru místo individuální konfigurace jednotlivých stanic. Klient, který používá DHCP, nemá kontrolu nad svou statickou IP adresou, ta je mu automaticky přidělována DHCP serverem.

Jednotlivé klienty je možné identifikovat podle hardwarové adresy síťové karty, tzv. MAC adresy, a tak jim, kdykoliv se spojí se serverem, přiřadit stejné nastavení. I přes dynamické přidělování IP adres je tak možno pro jednotlivé počítače zachovat stále stejné IP adresy (i když se počítače připojí až po delší době). Nefunguje to ale v případě, kdy je v síti více počítačů než adres; tehdy jsou adresy přidělovány podle potřeby.

Použití DHCP přináší dvě výhody. Zaprvé je možné jednoduše provádět i velice rozsáhlé změny v síti a spravovat všechny konfigurační soubory centrálně bez nutnosti individuální konfigurace všech klientů. Druhou výhodou je možnost velice jednoduchého připojování nových počítačů k síti. Připojovaným počítačům je automaticky přidělena IP adresa z vyčleněného adresního prostoru. To je požehnání zejména pro notebooky, které se pravidelně připojují do různých sítí.

Kromě IP adres a síťových masek je možné spravovat také názvy počítačů a domén, používané brány a adresy nameserverů, které jsou pak sdělovány klientům. Navíc je možné centrálně konfigurovat i např. server pro synchronizaci času (ntp) nebo tiskový server.

## 23.2 Konfigurace DHCP serveru pomocí nástroje YaST

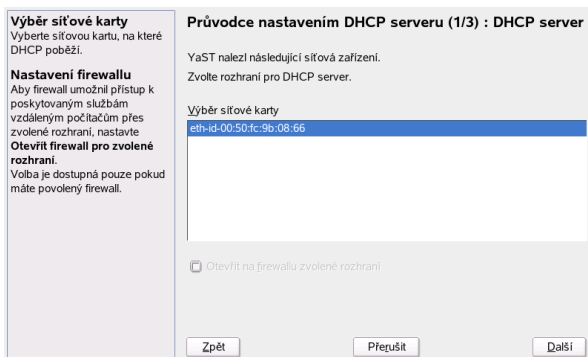
YaST DHCP modul umožňuje nastavit vlastní DHCP server pro lokální síť. Modul pracuje ve dvou různých režimech, jednoduchém a expertním:

Při prvním spuštění modulu vyvolá YaST průvodce, který vám pomůže provést základní konfiguraci DHCP serveru.

### Výběr síťové karty

V prvním kroku YaST zjistí, jaká jsou na vašem systému dostupná síťová rozhraní, a zobrazí jejich seznam. Ze seznamu vyberte rozhraní, na kterém má DHCP server naslouchat, a otevřete pro toto rozhraní firewall zaškrtnutím položky *Otevřít na firewallu zvolené rozhraní*. Viz 23.1 – „DHCP server: Výběr karty“ (strana 356).

**Obrázek 23.1** DHCP server: Výběr karty



### Obecná nastavení

V jednotlivých polích zadejte podrobnosti o klientech, které má DHCP server spravovat. Je třeba určit jméno domény, adresu časového serveru, adresu primárního a sekundárního DNS serveru, adresu tiskového serveru, WINS serveru (v případě smíšené sítě zahrnující počítače se systémem Linux i Windows), adresu výchozí brány a výchozí čas přidělení adresy. Viz 23.2 – „DHCP server: Obecná nastavení“ (strana 357).

**Obrázek 23.2** DHCP server: Obecná nastavení

The screenshot shows the 'Průvodce nastavením DHCP serveru (2/3) : DHCP server' dialog box. On the left is a sidebar with the title 'Obecná nastavení' and several sections: 'Zde můžete provést řadu DHCP nastavení.', 'Jméno domény' (with a text input field), 'IP primárního nameserveru a IP sekundárního nameserveru' (with two text input fields), 'Výchozí brána' (with a text input field), 'Časový server' (with a text input field), 'Tiskový server' (with a text input field), and 'WINS server' (with a text input field). The main area contains: 'Jméno domény' (text input), 'Časový server' (text input), 'IP primárního DNS serveru' (text input), 'Tiskový server' (text input), 'IP sekundárního DNS serveru' (text input), 'WINS server' (text input), 'Výchozí brána (router)' (text input), and 'Výchozí čas přidělení' (spinners for '4' and 'h'). At the bottom are 'Zpět', 'Přerušit', and 'Další' buttons.

## Dynamické DHCP

V tomto kroku nastavte, jak mají být klientům přiřazovány dynamické IP adresy. Určete rozsah, ze kterého budou adresy přidělovány. Všechny adresy musejí mít stejnou masku. Nastavte rovněž dobu přidělení adresy, po jejímž uplynutí musí počítač zažádat o prodloužení přidělení. Můžete také určit maximální dobu, po kterou je IP na serveru blokována pro klienta (*Max. čas přidělení*). Viz obrázek 23.3 – „DHCP server: Dynamické DHCP“ (strana 357)).

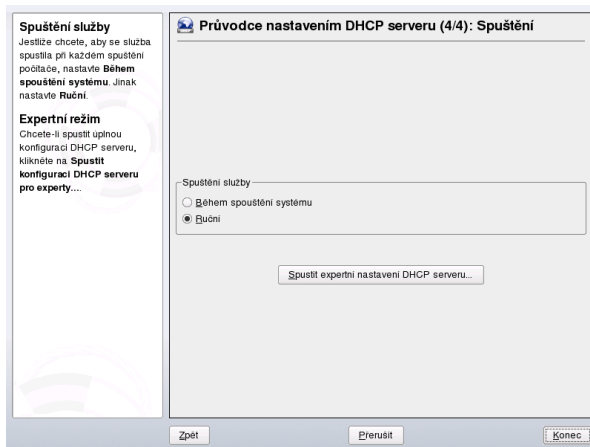
**Obrázek 23.3** DHCP server: Dynamické DHCP

The screenshot shows the 'Průvodce nastavením DHCP serveru (3/3) : DHCP server' dialog box. The left sidebar has the title 'Rozsah IP adres' and sections: 'Zde nastavte nejvyšší IP adresu a nejnižší IP adresu z rozsahu přidělovaného klientům. Tyto adresy musí mít stejnou masku. Například 192.168.1.1 a 192.168.1.64.', 'Přidělení' (with sub-sections for 'Zde můžete nastavit výchozí Čas přidělení' and 'Max. čas přidělení (volitelné)'), and 'Zde můžete nastavit výchozí Čas přidělení aktuálního rozsahu IP adres, kterým nastavíte optimální obnovování IP klientů.' The main area contains: 'Rozsah IP adres' (text input), 'Nejvyšší IP adresa:' (text input), 'Nejnižší IP adresa:' (text input), 'Přidělení' (text input), 'Čas přidělení' (spinners for '4' and 'h'), 'Max. čas přidělení' (spinners for '2' and 'Dni'), and 'Dni' (dropdown). At the bottom are 'Zpět', 'Přerušit', and 'Další' buttons.

## Ukončení konfigurace a nastavení režimu spouštění

V posledním dialogu konfiguračního průvodce zvolte, jak má být DHCP server spouštěn – automaticky při startu operačního systému nebo manuálně v případě potřeby (např. pro testovací účely). Klikněte na *Konec*, konfigurace DHCP serveru se tak dokončí. Viz obrázek 23.4 – „DHCP server: Spouštění systému“ (strana 358).

**Obrázek 23.4** DHCP server: Spouštění systému



## 23.3 DHCP softwarové balíčky

Pro systém SUSE Linux je k dispozici jak DHCP server, tak i klientský DHCP software. V systému SUSE Linux je DHCP server `dhcpcd` od konzorcia ISC (Internet Software Consortium). Na straně klienta lze použít program `dhclient` (rovněž od ISC) nebo klientského démona z balíčku `dhcpcd`.

SUSE Linux standardně používá `dhcpcd`, který je velmi snadno nastavitelný, spouští se automaticky při startu systému a okamžitě hledá DHCP server. Ke své práci nepotřebuje žádný konfigurační soubor a ve většině případů pracuje bez nutnosti jakéhokoliv zásahu. Pro složitější případy použijte ISC `dhclient`, který se nastavuje pomocí konfiguračního souboru `/etc/dhclient.conf`.

## 23.4 DHCP server `dhcpcd`

Srdcem každého DHCP systému je démon *Dynamic Host Configuration Protocol Daemon* (`dhcpcd`). Pronajímá adresy a kontroluje jejich používání tak, jak je nastaveno v konfiguračním souboru `/etc/dhcpcd.conf`. Změnou parametrů a hodnot uvedených v tomto souboru lze ovlivnit chování programu. Podívejte se na jednoduchý příklad

konfiguračního souboru `/etc/dhcpd.conf` v 23.1 – „Konfigurační soubor `/etc/dhcpd.conf`“ (strana 359):

### **Rovnice 23.1** Konfigurační soubor `/etc/dhcpd.conf`

```
default-lease-time 600;           # 10 minutes
max-lease-time 7200;             # 2 hours

option domain-name "cosmos.all";
option domain-name-servers 192.168.1.1, 192.168.1.2;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option subnet-mask 255.255.255.0;

subnet 192.168.1.0 netmask 255.255.255.0
{
    range 192.168.1.10 192.168.1.20;
    range 192.168.1.100 192.168.1.200;
}
```

Tento jednoduchý konfigurační soubor stačí k tomu, abyste prostřednictvím DHCP mohli přidělovat v síti IP adresy. Nezapomeňte na středníky na konci každé řádky, bez kterých není možné `dhcpd` spustit!

Jak je vidět z výše uvedeného příkladu, soubor je rozdělen do tří bloků. V první části je uvedeno, na kolik vteřin bude IP adresa standardně počítači přidělena (`default-lease-time`), nezažádá-li o jiný časový úsek. Po uplynutí této doby musí počítač požádat o prodloužení. Druhá položka určuje maximální dobu, o kterou si počítač může požádat (`max-lease-time`).

V druhé části jsou nastaveny některé obecné síťové parametry:

- Volbou `option domain-name` je definována výchozí doména sítě.
- `option domain-name-servers` může obsahovat až tři DNS servery, které slouží pro převod IP adres na názvy počítačů (a obráceně). V ideálním případě máte již v systému nebo v síti provozuschopný jmenný server (`nameserver`). Ten by měl pro každou dynamickou adresu definovat jméno počítače a naopak. Více informací o konfiguraci `nameserverů` viz 20 – „DNS — Domain Name System“ (strana 323).
- `option broadcast-address` určuje, jakou oznamovací (*broadcast*) adresu má použít dotazující se počítač.

- `option routers` určuje, kam mají být zasílány pakety, které nejsou určeny počítači v lokální síti (podle zdrojové a cílové adresy a masky podsítě). U malých sítí je tento směrovač obvykle bránou k Internetu.
- `option subnet-mask` určuje síťovou masku pro klienty.

Poslední část souboru definuje síť, včetně masek podsítě. Nakonec je zde uveden rozsah adres, které bude DHCP démon přiřazovat klientům. V našem příkladu může být klientům přiřazena libovolná adresa mezi 192.168.1.10 a 192.168.1.20 nebo mezi 192.168.1.100 a 192.168.1.200.

Pokud jste provedli tato nastavení, měli byste být schopni spustit DHCP démona příkazem `rcdhcpd start`. Démon tak bude okamžitě připraven k provozu. Pro kontrolu syntaxe konfiguračního souboru můžete použít příkaz `rcdhcpd check-syntax`. Pokud nastanou problémy a server skončí s chybou nebo nevrátí po startu `done`, podívejte se na systémová hlášení do protokolového souboru `/var/log/messages`, případně na desátou konzoli ( `Ctrl` + `Alt` + `F10` ).

Ve výchozím nastavení systému SUSE Linux se DHCP démon z bezpečnostních důvodů spouští ve chroot prostředí. Aby démon našel konfigurační soubory, musí být do chroot prostředí zkopírovány. Obvykle si s tím nemusíte lámat hlavu, protože příkaz `rcdhcpd start` soubory automaticky zkopíruje.

## 23.4.1 Počítač s pevnou IP adresou

Jak jsme zmínili výše, DHCP lze nastavit tak, aby určitý počítač dostal při každém požadavku přednastavenou statickou adresu. Explicitně určené adresy mají přednost před dynamickými adresami vybíranými z přiděleného rozsahu. Navíc statická adresa nikdy nevyprší, jak se to může stát s adresou dynamickou, například v případě, kdy je nedostatek adres a server je potřeby mezi počítači přerozdělit.

K identifikaci počítače, který má mít přidělovánu *statickou* adresu, používá `dhcpd` celosvětově unikátní hardwarovou adresu (MAC). Hardwarová adresa sestává z šesti párů šestnáctkových číslic (např. 00:00:45:12:EE:F4). Pokud jsou do konfiguračního souboru 23.1 – „Konfigurační soubor `/etc/dhcpd.conf`“ (strana 359) přidány řádky podobné těm z příkladu 23.2 – „Additions to the Configuration File“ (strana 361), bude danému počítači vždy přidělováno stejné nastavení.



## Rovnice 23.2 Additions to the Configuration File

```
host earth {  
hardware ethernet 00:00:45:12:EE:F4;  
fixed-address 192.168.1.21;  
}
```

Jméno počítače (host *jmenopocitace*, v našem příkladu *earth*) se vkládá na první řádek. Hardwarová (MAC) adresa se zapisuje na řádek druhý. Na linuxových strojích lze MAC adresu zjistit příkazem (v případě síťového zařízení *eth0*) `ifstatus eth0`. Pokud není karta aktivní, aktivujte ji příkazem `ifup eth0`. Výstup příkazu `ifstatus` by měl obsahovat řádek podobný následujícímu:

```
link/ether 00:00:45:12:EE:F4
```

Při nastavení uvedeném v příkladu výše bude počítači se síťovou kartou s MAC adresou `00:00:45:12:EE:F4` automaticky přiřazena IP adresa `192.168.1.21` a jméno *earth*. Na řádce s MAC adresou je zapsán i typ hardwaru, většinou *ethernet*. Je ale podporován i *token-ring* často se vyskytující v systémech IBM.

## 23.4.2 Zvláštnosti v systému SUSE Linux

Pro zvýšení bezpečnosti je SUSE verze ISC DHCP serveru opatřena *non-root/chroot* záplatou Ari Edelkinda. Server tak může běžet s uživatelským ID *nobody* ve *chroot* prostředí (`/var/lib/dhcp`). Aby to bylo skutečně možné, musí se konfigurační soubor `dhcpd.conf` nacházet v adresáři `/var/lib/dhcp/etc`. Startovací skript ho tam automaticky zkopíruje.

Tuto vlastnost lze nastavit v souboru `/etc/sysconfig/dhcpd`. Chcete-li spouštět `dhcpd` bez prostředí *chroot*, nastavte v něm proměnnou `DHCPD_RUN_CHROOTED` na `no`.

Chcete-li aby `dhcpd` překládal jména počítačů i z prostředí *chroot*, musí se zkopírovat i některé další soubory:

- `/etc/localtime`
- `/etc/host.conf`
- `/etc/hosts`
- `/etc/resolv.conf`

Tyto soubory jsou startovacím skriptem kopírovány do adresáře `/var/lib/dhcp/etc/`. Kopie je nutno brát v úvahu při dynamické modifikaci souborů skripty jako např. `/etc/ppp/ip-up`. Pokud však konfigurační soubor specifikuje pouze IP adresy (a nikoliv jména počítačů), nemusíte se tím zabývat.

Pokud ve vaší konfiguraci potřebujete do chroot prostředí kopírovat další soubory, nastavte je v proměnné `DHCPD_CONF_INCLUDE_FILES` v souboru `etc/sysconfig/dhcpd`. Aby mohl DHCP server v prostředí chroot zaznamenávat údaje do protokolových souborů i po restartu syslog démona, musíte do proměnné `SYSLOGD_PARAMS` v souboru `/etc/sysconfig/syslog` vložit volbu `"-a /var/lib/dhcp/dev/log"`.

## 23.5 Další informace

Více informací o DHCP najdete na stránkách *Internet Software Consortium* (<http://www.isc.org/products/DHCP/>). Řada důležitých informací je také v manuálových stránkách `dhcpd`, `dhcpd.conf`, `dhcpd.leases` a `dhcp-options`.

# Synchronizace času pomocí xntp 24

NTP (Network Time Protocol) je protokol pro synchronizaci systémového času po síti. Počítače mohou s jeho pomocí získávat informaci o času z přesných časových serverů. Takto seřízený počítač pak může poskytovat informaci o přesném čase dalším počítačům v síti. Cíle jsou dva – zajistit přesnou informaci o absolutním čase a synchronizovat čas všech strojů v síti.

Nastavení správného a jednotného času v síti je důležité v řadě situací. Počítače samozřejmě obsahují vlastní hardwarové hodiny. Jejich čas se však může u různých počítačů lišit. Takové časové rozdíly pak mohou způsobit řadu problémů např. při práci s databázemi. Také v síti je obvykle potřeba mít čas na jednotlivých strojích synchronizovaný. Lze ho nastavit ručně, ale to není dobrý přístup. Síťové řešení tohoto problému nabízí program xntp. Neustále upravuje systémový čas pomocí údajů ze spolehlivých časových serverů v síti. Navíc umožňuje spravovat lokální referenční hodiny, např. rádiem řízené.

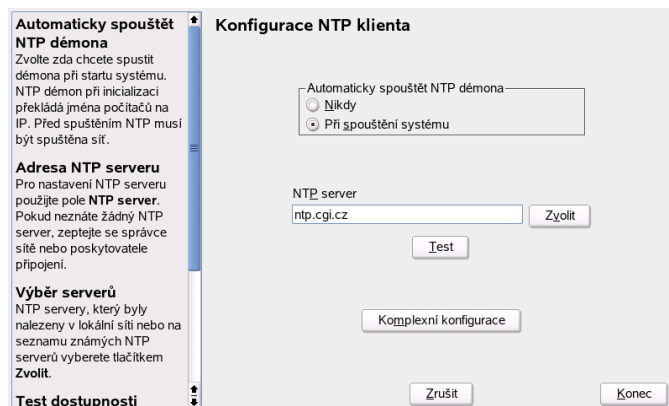
## 24.1 Nastavení NTP klienta v programu YaST

Nastavení NTP klienta můžete v systému SUSE Linux provést pomocí nástroje YaST v části *Síťové služby* v modulu *Klient NTP*. Na výběr máte z rychlé nebo komplexní konfigurace.

## 24.1.1 Rychlé nastavení NTP klienta

Rychlé nastavení NTP klienta se skládá ze dvou kroků. V prvním je nutné nastavit spuštění xntpd, ve druhém zadat NTP server. Chcete-li, aby se xntpd spouštěl automaticky při startu systému, vyberte *Během zavádění systému*. Pak klikněte na *Pokračovat*. Tím se otevře druhý dialog, ve kterém zadáte vhodný server.

**Obrázek 24.1** YaST: Konfigurace NTP klienta

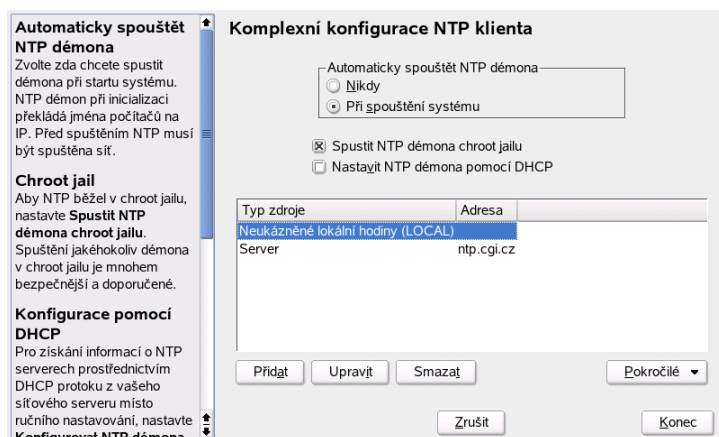


Po kliknutí na *vyberte* máte na výběr *Lokální síť* nebo *Veřejný NTP server*. Zvolte nejvhodnější server a otestujte nastavení tlačítkem *Test*. Pokud test dopadl dobře, potvrďte výběr tlačítkem *OK*.

## 24.1.2 Komplexní nastavení NTP klienta

Komplexní nastavení NTP klienta je dostupné v hlavním dialogu *Klient NTP* po nastavení spuštění kliknutím na tlačítko *Komplexní konfigurace* (viz 24.1 – „YaST: Konfigurace NTP klienta“ (strana 364).

## Obrázek 24.2 YaST: Komplexní konfigurace NTP klienta



V *Komplexní konfiguraci NTP klienta* lze nastavit, zda se má xntpd spouštět v chroot jail. Tímto nastavením výrazně zvýšíte bezpečnost systému, protože v případě napadení xntpd nebude mít útočník k dispozici přístup do systému. Volba *Nastavit NTP démona pomocí DHCP* zajistí získání NTP serverů pro NTP klienta přes DHCP.

Jednotlivé časové servery a další časové zdroje najdete v tabulce pod volbami. Můžete je *Přidat*, *Upravit* nebo *Smazat*.

Nový zdroj časových informací zadáte kliknutím na *Přidat*. Vyberte požadovaný typ zdroje a klikněte na tlačítko *Další*. Vybrat si můžete z následujících typů zdrojů:

### Server

Zvolíte-li tuto volbu, zadejte v následujícím dialogu NTP server (viz 24.1.1 – „*Rychlé nastavení NTP klienta*“ (strana 364)). Aktivujte *Použit pro počáteční synchronizaci*, pokud chcete provádět synchronizaci s tímto serverem při startu systému. V dalším poli můžete zadat dodatečné volby. Více informací najdete v adresáři `/usr/share/doc/packages/xntp-doc`.

### Rovnocenný

Zde můžete místo serveru zvolit jinou klientskou stanici, se kterou bude navázán symetrický vztah. Další dialog je podobný jako v případě volby *Server*.

### Radio hodiny

U radio hodin musíte v následujícím dialogu zadat typ hodin, číslo jednotky, jméno zařízení a další volby. Doladění provedete kliknutím na *Kalibrace ovladače*. Další

informace najdete v souboru `/usr/share/doc/packages/xntp-doc/html/refclock.htm`.

#### Odchozí všesměrové vysílání

Časové informace lze vysílat po síti. Pokud tak chcete činit, je v tomto dialogu nutné zadat adresu, na kterou mají být časové údaje vysílány. Nepoužívejte vysílání, pokud nemáte spolehlivý časový zdroj, např. rádiem řízené hodiny.

#### Příchozí všesměrové vysílání

Jestliže má klient přijímat vysílané pakety, zadejte v tomto poli adresu, ze které mají být přijímány pakety.

## 24.2 Nastavení xntp v síti

Výchozí nastavení xntp respektuje jako referenční čas lokální hodiny počítače. Použití těchto (BIOS) hodin je však pouze náhradní řešení pro případ, kdy není dostupný spolehlivější zdroj. Nejjednodušší způsob, jak přistupovat k časovému serveru, je zadat server do položky `server` v konfiguračním souboru `/etc/ntp.conf`. Např. pokud má být čas synchronizován podle serveru `ntp.example.com`, do souboru `/etc/ntp.conf` vložte řádek `server ntp.example.com`.

Chcete-li používat serverů více, vložte pro každý z nich samostatný řádek začínající klíčovým slovem `server`. Po spuštění xntpd příkazem `rcxntpd start` trvá asi hodinu, než se čas stabilizuje a vytvoří se *drift soubor* korigující lokální hardwarové hodiny. Pomocí drift souboru lze spočítat a opravit systematickou chybu hardwarových hodin okamžitě po spuštění počítače. Tím je zajištěna vysoká stabilita systémového času.

Jsou dva možné způsoby využití NTP na klientovi. Prvním je dotazování se na přesný čas na časovém serveru v pravidelných intervalech. Pokud je ale klientů hodně, může to pro server znamenat velkou zátěž. Druhou možností je čekat na vysílání časových údajů servery v síti. Nevýhodou je, že kvalita vysílajícího serveru není známá a server vysílající nesprávné časové údaje může způsobit vážné problémy.

Pokud je čas vysílán po síti, nepotřebujete znát jméno serveru. Stačí do souboru `/etc/ntp.conf` vložit řádek `broadcastclient`. Chcete-li používat pouze jeden nebo několik známých serverů, vložte jejich jména do řádky začínající slovem `servers`.

## 24.3 Nastavení lokálních referenčních hodin

Program `xntp` obsahuje také ovladač pro připojení lokálních referenčních hodin. Seznam podporovaných hodin najdete po nainstalování balíčku `xntp-doc` v souboru `file:/usr/share/doc/packages/xntp-doc/html/refclock.htm`. Každý ovladač je označen vlastním číslem. Konfigurace `xntp` se pak provádí pomocí pseudo IP adres. Údaje o hodinách se vloží do souboru `/etc/ntp.conf`, jako by šlo o standardní síťový časový server. Jsou jim přiřazeny speciální IP adresy ve formátu `127.127.t.u`. Hodnota `t` označuje typ hodin a určuje výběr použitého ovladače, zatímco `u` (unit) specifikuje použité rozhraní.

Jednotlivé ovladače mají specifické konfigurační parametry. Podrobnosti o jednotlivých typech hodin naleznete v souboru `/usr/share/doc/packages/xntp-doc/html/driverNN.htm` (kde `NN` je číslo ovladače). Například hodiny typu 8 (radiové hodiny připojené přes sériové rozhraní) vyžadují přídavný modul. Modul Conrad DCF77 má např. režim 5. Aby byly tyto hodiny používány jako primární referenční zdroj, je nutné použít klíčové slovo `prefer`. Kompletní položka pro nastavení modulu Conrad DCF77 v konfiguračním souboru se proto napíše takto:

```
server 127.127.8.0 mode 5 prefer
```

Ostatní hodiny se nastavují podobně. Příklady najdete v dokumentaci `xntp` v `/usr/share/doc/packages/xntp-doc/html`.





# LDAP — adresářové služby

# 25

LDAP (Lightweight Directory Access Protocol) je sada protokolů určených ke správě a přístupu k informačním adresářům. LDAP lze využít k mnoha účelům, jako je správa uživatelů a skupin, správa systémové konfigurace nebo správa adres. V této kapitole jsou popsány základy funkce LDAP a jeho konfigurace pomocí nástroje YaST.

V síťovém prostředí je velmi důležité uchovávat důležité informace na dostupném místě a v uspořádané podobě. To lze zajistit adresářovou službou, která, podobně jako zlaté stránky, poskytuje informace ve strukturované a přehledné formě s možností snadného vyhledávání.

V ideálním případě server všechna data uloží do adresáře a pomocí jednotného protokolu je pak distribuuje všem klientům. Data jsou strukturována tak, aby s nimi mohla pracovat celá řada různých aplikací. Není tak nutné, aby každá kalendářová aplikace či poštovní klient udržoval nezávislou databázi, stačí vytvořit jednu centrální. Tím se uspoří čas a náklady na údržbu několika databází. Použitím otevřeného a standardizovaného protokolu LDAP navíc zajistíte, že tato data budou dostupná pro různé typy aplikací a klientů.

Pojmem adresář v této souvislosti rozumíme databázi optimalizovanou pro rychlé a efektivní čtení a vyhledávání, která má tyto vlastnosti:

- Aby bylo umožněno vícenásobné čtení v maximálním objemu, je zápis omezen na aktualizace administrátorem databáze. Běžné typy databází jsou optimalizovány pro zápis maximálního množství dat v krátkém čase.
- Protože jsou možnosti zápisu značně omezeny, slouží adresářové služby především pro uchovávání neměnných *statických informací*. V normální databázi se naopak

data mění velmi často (dynamická data). Např. telefonní číslo společnosti se nemění tak často jako účetní údaje.

- Administrace statických dat vyžaduje jen výjimečné aktualizace a změny. Při práci s dynamickými daty, jako např. zůstatky na účtech, je kladen vysoký důraz na konzistenci dat. Pokud je například z jednoho účtu odečtena částka a připsána na jiný, musí obě operace proběhnout současně v rámci jedné transakce. Databáze takové transakce podporují, ale adresářové služby nikoliv. Krátkodobé nekonzistence nevedou u adresářové služby k žádným závažným problémům.

Adresářové služby jako LDAP nejsou navrženy pro podporu komplexní aktualizace a dotazovacího mechanismu. Přístup musí být rychlý a jednoduchý.

Řada adresářových služeb existovala a dosud existuje jak na platformě Unix, tak mimo ní. Několika příklady jsou Novell NDS, Microsoft ADS, Banyan Street Talk a OSI standard X.500. LDAP byl původně navržen jako verze DAP (Directory Access Protocol) navrženého pro přístup k X.500. Standard X.500 se zabývá hierarchickou organizací adresářové struktury.

LDAP je zjednodušená verze DAP, která neobsahuje některé funkce DAP, což umožňuje úspory zdrojů. Použití protokolu TCP/IP usnadňuje spojení aplikací se službou LDAP.

LDAP je dnes samostatným řešením pracujícím bez podpory X.500. LDAPv3 (verze protokolu v balíčku `openldap2`) podporuje tzv. *referrals*, které umožňují vytváření distribuovaných databází. Nové je také využití SASL (Simple Authentication and Security Layer).

LDAP není omezen na X.500 servery, jak bylo původně v plánu. Opensource server `slapd` dokáže ukládat objektové informace v lokální databázi. Díky rozšíření `slurpd` je možné LDAP servery replikovat.

Balíček `openldap2` obsahuje následující programy:

`slapd`

LDAPv3 server spravující informace v databázi typu BerkeleyDB.

`slurpd`

Program pro replikaci změn dat z lokálního serveru na ostatní LDAP servery v síti.

Další nástroje pro správu  
slapcat, slapadd, slapindex.

## 25.1 LDAP versus NIS

Unixoví administrátoři pro převod jmen a distribuci dat v síti tradičně používají službu NIS. Konfigurační data se nacházejí v souborech v adresáři `/etc:` `group`, `hosts`, `mail`, `netgroup`, `networks`, `passwd`, `printcap`, `protocols`, `rpc` a `services`, odkud jsou distribuována klientům v síti. Tyto soubory lze velmi jednoduše spravovat, protože jde o prosté textové soubory. Správa většího množství dat je ovšem náročnější vzhledem k neexistující strukturalizaci. Služba NIS je určena pouze pro unixové systémy, což znesnadňuje nasazení v heterogenních sítích.

Na rozdíl od NIS není služba LDAP omezená jen na čistě unixové sítě. LDAP podporují Windows servery (od verze 2000) a podpora nabízí také Novell.

LDAP je vhodný všude, kde je zapotřebí centrálně spravovat datovou strukturu, např.:

- Náhrada NIS.
- Směrování pošty (`postfix`, `sendmail`).
- Adresář pro poštovní klienty jako je Mozilla, Evolution či Outlook.
- Administrace popisů zón BIND9 name serveru.

Tento seznam by mohl být mnohem delší, protože LDAP je na rozdíl od NIS rozšiřitelný. Jasně definovaná hierarchická struktura dat usnadňuje administraci velkého množství dat.

## 25.2 Struktura adresářového stromu LDAP

LDAP adresář má stromovou strukturu. Všechny záznamy (zvané objekty) adresáře mají v hierarchii jasně definovanou pozici. Tato struktura je označována jako *informační adresářový strom* (DIT, directory information tree). Kompletní cesta k určité položce se nazývá *jedinečné jméno* nebo-li DN (distinguished name). Jednotlivé nody této cesty

se nazývají *relativní jedinečné jméno* nebo-li RDN (relative distinguished name). Objekty mohou být dvou typů:

kontejner

Tyto objekty mohou obsahovat další objekty. Mezi tyto objekty patří `root` (kořenový element adresářového stromu), `c` (country, země), `ou` (organizational unit, organizační jednotka) a `dc` (domain component, doménová komponenta).

list

Tyto objekty se nalézají na samém okraji větve a nemají žádné podobjekty. Jde např. o `person`, `InetOrgPerson` nebo `groupofNames`.

Na samém vrcholu adresářové struktury stojí objekt `root`. Ten obsahuje podobjekty `c` (country), `dc` (domain component) nebo `o` (organization). Vztahy mezi objekty v LDAP stromu jsou zřejmé z obrázku 25.1 – „Struktura LDAP adresáře“ (strana 372).

**Obrázek 25.1** Struktura LDAP adresáře

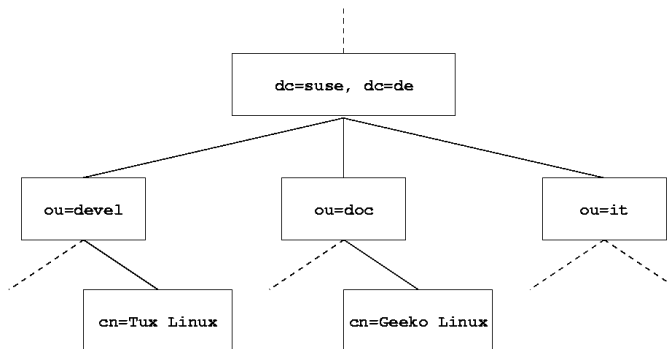


Diagram obsahuje fiktivní informační adresářový strom. Každý obdélník na obrázku představuje jeden záznam. Úplně validní *jedinečné jméno* (DN) smyšleného SUSE zaměstnance jménem Geeko Linux je v našem případě `cn=Geeko Linux, ou=doc, dc=suse, dc=de`. Je vytvořeno přidáním RDN `cn=Geeko Linux` k DN předcházejícího záznamu `ou=doc, dc=suse, dc=de`.

Obecná pravidla určující, jaké typy objektů mají být ukládány v DIT, jsou daná tzv. schématem (*scheme*). Typ objektu je určen *objektovou třídou*. Objektová třída určuje vlastnosti, které objekt *musí* nebo *může* mít. Schéma proto musí obsahovat definici všech objektových tříd a atributů. K dispozici je několik obecných schémat (viz RFC

2252 a 2256). Samozřejmě je možné vytvořit si schéma vlastní, které bude více vyhovovat vašim požadavkům.

Tabulka 25.1 – „Běžně používané objektové třídy a atributy“ (strana 373) nabízí krátký přehled tříd objektů ze schémat `core.schema` a `inetorgperson.schema` použitých v příkladu. Najdete zde také atributy a platné hodnoty těchto atributů.

**Tabulka 25.1** Běžně používané objektové třídy a atributy

Objektová třída	Význam	Příklad záznamu	Povinné atributy
<code>dcObject</code>	<i>domainComponent</i> (komponenta domény)	suse	dc
<code>organizationalUnit</code>	<i>organizationalUnit</i> (organizační jednotka)	doc	ou
<code>inetOrgPerson</code>	<i>inetOrgPerson</i> (osobní data pro intranet nebo internet)	Geeko Linux	sn a cn

Příklad 25.1 – „Výtah ze `schema.core` (řádky jsou dodatečně očíslovány)“ (strana 373) ukazuje výtah ze schématu s vysvětlením:

**Rovnice 25.1** Výtah ze `schema.core` (řádky jsou dodatečně očíslovány)

```
#1 attributetype ( 2.5.4.11 NAME ( 'ou' 'organizationalUnitName' )
#2         DESC 'RFC2256: organizational unit this object belongs to'
#3         SUP name )
...
#4 objectclass ( 2.5.6.5 NAME 'organizationalUnit'
#5         DESC 'RFC2256: an organizational unit'
#6         SUP top STRUCTURAL
#7         MUST ou
#8 MAY (userPassword $ searchGuide $ seeAlso $ businessCategory
        $ x121Address $ registeredAddress $ destinationIndicator
        $ preferredDeliveryMethod $ telexNumber
        $ teletexTerminalIdentifier $ telephoneNumber
        $ internationalISDNNumber $ facsimileTelephoneNumber
        $ street $ postOfficeBox $ postalCode $ postalAddress
        $ physicalDeliveryOfficeName
        $ st $ l $ description) )
```

Typ atributu `organizationalUnitName` a odpovídající objektová třída `organizationalUnit` zde slouží jako příklad. Řádka 1 obsahuje jméno atributu a jeho unikátní identifikátor OID (*object identifier*) (číselný údaj) a zkratku atributu.

Řádka 2 obsahuje krátký popis atributu (`DESC`). Je zde uveden i odkaz na příslušný RFC. `SUP` v řádce 3 uvádí nadřazený typ atributu, ke kterému tento atribut náleží.

Samotná definice objektové třídy `organizationalUnit` začíná na řádce 4. Stejně jako definice atributu obsahuje OID a jméno třídy. Na řádce 5 je krátký popis objektové třídy. Řádka 6 (`SUP top`) udává, že tato objektová třída není závislá na jiné objektové třídě. Řádka 7 začínající řetězcem `MUST` udává všechny atributy, které objekt typu `organizationalUnit` musí obsahovat. Řádka 8 začínající řetězcem `MAY` udává typy atributů, které *mohou* být s touto objektovou třídou používány.

Velmi hezký úvod do schémat najdete v dokumentaci OpenLDAP. Pokud je nainstalován, najdete ho v souboru `/usr/share/doc/packages/openldap2/admin-guide/index.html`.

## 25.3 Konfigurace LDAP serveru pomocí `slapd.conf`

Konfigurace LDAP serveru se nachází v souboru `/etc/openldap/slapd.conf`. Zde jsou popsány jednotlivé položky konfigurace. Položky začínající znakem `#` jsou zakomentované a tedy neaktivní. Pokud je chcete aktivovat, musíte znak smazat.

### 25.3.1 Globální nastavení v `slapd.conf`

**Rovnice 25.2** *slapd.conf: Include příkaz pro schéma*

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/inetorgperson.schema
```

První příkazy `slapd.conf` zobrazené v příkladu 25.2 – „[slapd.conf: Include příkaz pro schéma](#)“ (strana 374) určují schéma LDAP adresáře. K základnímu povinnému schématu (zde `core.schema`) lze přidávat i dodatečná schémata (v našem případě `inetorgperson.schema`). Další schémata naleznete v adresáři `/etc/openldap/schema`. Pro nahrazení služby NIS službou LDAP budete potřebovat dvě schémata –

`rfc2307.schema` a `cosine.schema`. Informace o této problematice najdete v dokumentaci OpenLDAP.

### **Rovnice 25.3** *slapd.conf: pidfile a argsfile*

```
pidfile /var/run/slapd/slapd.pid
argsfile /var/run/slapd/slapd.args
```

Tyto dva soubory obsahují PID (process ID) a některé argumenty, se kterými je spouštěn `slapd`. Žádné změny zde nejsou potřeba.

### **Rovnice 25.4** *slapd.conf: Kontrola přístupu*

```
# Sample Access Control
#     Allow read access of root DSE
# Allow self write access
#     Allow authenticated users read access
#     Allow anonymous users to authenticate
# access to dn="" by * read
#     access to * by self write
#         by users read
#         by anonymous auth
#
# if no access controls are present, the default is:
#     Allow read by all
#
# rootdn can always write!
```

**Příklad 25.4** – „`slapd.conf: Kontrola přístupu`“ (strana 375) ukazuje část souboru `slapd.conf`, která se týká nastavení přístupu k adresáři LDAP na serveru. Nastavení uvedená zde v globální sekci souboru `slapd.conf` jsou platná až do okamžiku vytvoření nastavení v části specifické pro databázi. Ta mají přednost před globálními nastaveními. V našem příkladě mají všichni uživatelé práva pro čtení, ale pouze administrátor (`rootdn`) může do této databáze zapisovat. Nastavení přístupových práv v LDAP je poměrně složité téma, nabízneme proto několik tipů:

- Každé pravidlo pro přístup má následující strukturu:

```
access to <what> by <who> <access>
```

- *what* nahraďte objektem nebo atributem, ke kterému se má přistupovat. Jednotlivé větve adresáře mohou být chráněny vlastními pravidly. Pokud chcete, můžete chránit části adresáře pomocí regulárních výrazů. Program `slapd` vyhodnocuje všechna pravidla v pořadí, v jakém jsou uvedena v konfiguračním souboru. Obecnější pravidla by měla být uvedena později – uplatněno je první platné pravidlo, ostatní jsou ignorována.

- *who* určuje, komu bude přiznán přístup do oblastí určených pomocí *what*. Lze použít i regulární výrazy. `slapd` opět ukončí vyhodnocování *who* po nalezení první shody, proto by obecnější pravidla měla být uvedena později. Možná jsou nastavení uvedená v tabulce 25.2 – „Uživatelské skupiny a jejich přístupová práva“ (strana 376)

**Tabulka 25.2** *Uživatelské skupiny a jejich přístupová práva*

Tag	Význam
*	všichni uživatelé bez výjimky
anonymous	neautentizovaní uživatelé
users	autentizovaní uživatelé
self	uživatelé spojení s cílovým objektem
dn.regex=<regex>	všichni uživatelé vyhovující regulárnímu výrazu

- *access* určuje typ přístupu. Možná nastavení najdete v tabulce 25.3 – „Typy přístupu“ (strana 376).

**Tabulka 25.3** *Typy přístupu*

Tag	Význam
none	bez přístupu
auth	spojení se serverem
compare	porovnávání
search	vyhledávání pomocí filtrů
read	čtení
write	zápis



slapd porovnává požadavky klientů s nastavením přístupových práv v souboru `slapd.conf`. Klientovi je přístup povolen jen v případě, že splňuje požadavky pro přístup (má požadovaná nebo vyšší práva). Pokud klient vyžaduje vyšší práva, než mu jsou přiřazena, je mu odmítnut přístup.

**Příklad 25.5** – „`slapd.conf`: Příklad nastavení přístupových práv“ (strana 377) ukazuje jednoduché nastavení přístupových práv pomocí regulárního výrazu:

**Rovnice 25.5** `slapd.conf`: Příklad nastavení přístupových práv

```
access to dn.regex="ou=(^[^,]+),dc=suse,dc=de"  
by dn.regex="cn=administrator,ou=$1,dc=suse,dc=de" write  
by user read  
by * none
```

V tomto příkladu má práva zápisu do záznamu `ou` pouze administrátor. Všichni ostatní autentizovaní uživatelé mají práva ke čtení. Ostatní uživatelé nemají žádný přístup.

---

### Tip: Vytvoření přístupových pravidel

Pokud chybí pravidlo `access to` nebo neexistuje vyhovující proměnná `by`, není přístup povolen. Jsou přiznána jen výslovně uvedená přístupová práva. Jestliže nezádáte vůbec žádné pravidlo, nastaví se výchozí přístupová práva, tj. právo zápisu pro administrátora a právo čtení pro všechny ostatní.

---

Podrobné informace a příklady nastavení přístupových práv k LDAP naleznete v dokumentaci balíčku `openldap2`.

Kromě nastavení přístupových práv v centrálním konfiguračním souboru (`slapd.conf`) je k dispozici také ACI (Access Control Information). ACI umožňuje ukládání informací o jednotlivých objektech LDAP stromu. Tento způsob kontroly přístupu je však stále ještě považován za experimentální. Viz <http://www.openldap.org/faq/data/cache/758.html>.

## 25.3.2 Nastavení specifická pro databázi v souboru slapd.conf

**Rovnice 25.6** *slapd.conf: Nastavení specifická pro databázi*

```
database ldbm
suffix "dc=suse,dc=de"
rootdn "cn=admin,dc=suse,dc=de"
# Cleartext passwords, especially for the rootdn, should
# be avoided. See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw secret
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory /var/lib/ldap
# Indices to maintain
index objectClass eq
```

Na prvním řádku této sekce (viz 25.6 – „[slapd.conf: Nastavení specifická pro databázi](#)“ (strana 378)) je určen typ databáze (v našem případě LDBM). Na druhé řádce (`suffix`) je určeno, za jakou část LDAP stromu server zodpovídá. Následující `rootdn` určuje administrátora serveru. Zde nastavený uživatel nepotřebuje mít LDAP záznam nebo existovat jako běžný uživatel. Heslo administrátora je nastaveno v položce `rootpw`. Místo `secret` můžete použít hash administrátorského hesla vytvořený pomocí programu `slappasswd`. Položka `directory` určuje adresář (v souborovém systému), ve kterém je uložena databáze. Poslední část, `index objectClass eq`, určuje, že index bude udržován pro všechny objektové třídy. Podle zkušenosti zde lze nastavit atributy, které uživatelé nejčastěji vyhledávají. `Access` pravidla nastavená v této sekci se použijí místo pravidel globálních.

## 25.3.3 Spuštění a zastavení serveru

Je-li server plně nakonfigurovaný a jsou-li vytvořeny všechny požadované záznamy, jak je popsáno v sekci 25.4 – „[Správa dat v LDAP adresáři](#)“ (strana 379), spusťte server jako uživatel `root` příkazem `rclldap start`. Ručně server zastavíte příkazem `rclldap stop`. Stav běžícího LDAP serveru zjistíte příkazem `rclldap status`.

Pokud chcete LDAP server spouštět automaticky při startu systému, použijte k nastavení editor úrovní běhu systému nástroje YaST (viz 8.6 – „[Editor úrovní běhu](#)“ (strana 157)). Automatické spuštění při startu systému můžete zajistit také pomocí příkazu `insserv` (viz 8.5 – „[Init skripty](#)“ (strana 153)).

## 25.4 Správa dat v LDAP adresáři

OpenLDAP nabízí pro správu dat v LDAP adresáři celou řadu nástrojů. Čtyři nejdůležitější nástroje pro vkládání, mazání, vyhledávání a úpravy dat jsou popsány dále.

### 25.4.1 Vkládání dat do LDAP adresáře

Pokud je LDAP server správně nakonfigurován, tedy pokud jsou v souboru `/etc/openldap/ldapd.conf` nastaveny položky `suffix`, `directory`, `rootdn`, `rootpw` a `index`, pokračujte vkládáním záznamů. K tomu OpenLDAP nabízí nástroj `ldapadd`. Objekty je z praktických důvodů vhodné vkládat po větších celcích. Vhodný je například LDIF formát (LDAP Data Interchange Format). LDIF je jednoduchý textový soubor obsahující páry atribut—hodnota. Dostupné objektové třídy a atributy jsou definované ve schématech uvedených v souboru `slapd.conf`. LDIF soubor k vytvoření hrubé kostry obrázku 25.1 – „Struktura LDAP adresáře“ (strana 372) by vypadal asi tak, jak je uvedeno v příkladu 25.7 – „Příklad LDIF souboru“ (strana 379):

#### **Rovnice 25.7** *Příklad LDIF souboru*

```
# The SUSE Organization
dn: dc=suse,dc=de
objectClass: dcObject
objectClass: organization
o: SUSE AG dc: suse

# The organizational unit development (devel)
dn: ou=devel,dc=suse,dc=de
objectClass: organizationalUnit
ou: devel

# The organizational unit documentation (doc)
dn: ou=doc,dc=suse,dc=de
objectClass: organizationalUnit
ou: doc

# The organizational unit internal IT (it)
dn: ou=it,dc=suse,dc=de
objectClass: organizationalUnit
ou: it
```

---

## Důležité: Kódování LDIF souborů

LDAP pracuje s UTF-8 (Unicode). Používejte proto editor s podporou UTF-8 (např. Kate nebo novější verze editorů Emacs či Vim). Jestliže použijete editor bez podpory UTF-8, budou se špatně zobrazovat znaky s českou diakritikou. Pokud potřebujete převést do UTF-8 již existující text, použijte program `recode`.

---

Soubor se ukládá s příponou `.ldif` a serveru se předává příkazem:

```
ldapadd -x -D <dn administrátora> -W -f <soubor>.ldif
```

První parametr, `-x`, vypíná ověřování pomocí SASL. Parametr `-D` specifikuje uživatele, který operaci volá. Za touto volbou musí následovat DN administrátora tak, jak je uvedeno v souboru `slapd.conf`. V našem případě jde o `cn=admin,dc=suse,dc=de`. Přepínač `-W` obejde zadávání hesla přímo na příkazovém řádku (v prostém textu) a zobrazí zvláštní výzvu k zadání hesla. Jde o heslo ze souboru `slapd.conf` (`rootpw`). Parametrem `-f` předáte jméno souboru. Ukázkou běhu programu `ldapadd` si můžete prohlédnout v příkladu [25.8 – „Použití ldapadd s example.ldif“](#) (strana 380).

### **Rovnice 25.8** *Použití ldapadd s example.ldif*

```
ldapadd -x -D cn=admin,dc=suse,dc=de -W -f example.ldif
```

```
Enter LDAP password:
adding new entry "dc=suse,dc=de"
adding new entry "ou=devel,dc=suse,dc=de"
adding new entry "ou=doc,dc=suse,dc=de"
adding new entry "ou=it,dc=suse,dc=de"
```

Data jednotlivých uživatelů lze připravit v oddělených LDIF souborech. Příklad [25.9 – „LDIF data uživatele Tux“](#) (strana 380) přidává do LDAP adresáře uživatele Tux:

### **Rovnice 25.9** *LDIF data uživatele Tux*

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
objectClass: inetOrgPerson
cn: Tux Linux
givenName: Tux
sn: Linux
mail: tux@suse.de
uid: tux
telephoneNumber: +49 1234 567-8
```

LDIF soubor může obsahovat libovolné množství objektů. Jednotlivé větve stromu je tak možné vložit do databáze najednou nebo po částech. Pokud se některé části mění častěji, je vhodné je oddělit zvlášť.

## 25.4.2 Úprava dat v LDAP adresáři

K úpravě dat se používá příkaz `ldapmodify`. Nejjednodušší způsob je změnit patřičný LDIF soubor a ten pak předat serveru. Pokud byste např. chtěli změnit telefonní číslo kolegy Tuxe z +49 1234 567-8 na +49 1234 567-10, změňte LDIF soubor tak, jak je uvedeno v příkladu 25.10 – „Upravený LDIF soubor `tux.ldif`“ (strana 381):

### **Rovnice 25.10** *Upravený LDIF soubor `tux.ldif`*

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
changetype: modify
replace: telephoneNumber
telephoneNumber: +49 1234 567-10
```

Upravený soubor importujete do adresáře na serveru příkazem:

```
ldapmodify -x -D cn=admin,dc=suse,dc=de -W -f tux.ldif
```

Vlastnosti lze měnit i přímo následujícím postupem:

- Spustíte příkaz `ldapmodify` a zadejte heslo:

```
ldapmodify -x -D cn=admin,dc=suse,dc=de -W
Enter LDAP password:
```

- Při zadání změn je nutné dodržovat syntaxi. Příkazy pro náš případ vypadají takto:

```
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
changetype: modify
replace: telephoneNumber
telephoneNumber: +49 1234 567-10
```

Více informací o `ldapmodify` a příslušné syntaxi najdete v jeho manuálové stránce (`ldapmodify(1)`).

## 25.4.3 Vyhledávání a čtení dat z LDAP adresáře

OpenLDAP poskytuje nástroj `ldapsearch` pro vyhledávání a čtení dat z LDAP adresáře. Jednoduchý dotaz má následující syntaxi:

```
ldapsearch -x -b dc=suse,dc=de "(objectClass=*)"
```

Parametrem `-b` nastavíte vyhledávací základnu (*search base*), tj. sekci stromu, která má být prohledána, v našem případě `dc=suse,dc=de`. Volba `-x` zapíná jednoduchou autentizaci. `(objectClass=*)` určuje, že budou čteny všechny objekty v adresáři. Tento příkaz je vhodný např. k ověření správnosti záznamů po vytvoření nového adresářového stromu. Více informací najdete v manuálové stránce `ldapsearch(1)`.

## 25.4.4 Mazání dat z LDAP adresáře

Nechtěné záznamy smažete pomocí příkazu `ldapdelete`. Syntaxe je podobná jako u příkazů uvedených výše. Např. celý záznam `Tux Linux` smažete příkazem:

```
ldapdelete -x -D cn=admin,dc=suse,dc=de -W cn=Tux \
Linux,ou=devel,dc=suse,dc=de
```

## 25.5 YaST LDAP klient

YaST obsahuje modul pro nastavení ověřování uživatelů pomocí LDAP. Pokud jste tuto vlastnost nepovolili během instalace systému, spusťte modul volbou *Sítové služby* → *Klient LDAP*. YaST automaticky povolí změny PAM a NSS vyžadované LDAP (jak je popsáno dále) a nainstaluje potřebné soubory.

### 25.5.1 Standardní procedura

Pro pochopení funkce YaST Klient LDAP modulu je nutné znát procedury probíhající na klientském počítači. Při aktivaci LDAP pro ověřování v síti nebo po spuštění YaST Klient LDAP modulu se nainstalují balíčky `pam_ldap` a `nss_ldap` a nastaví se dva související konfigurační soubory. `pam_ldap` je PAM modul odpovědný za přenos dat mezi přihlašovacím procesem a LDAP sloužícím jako zdroj autentizačních dat. Nain-

staluje se modul `pam_ldap`. `so` a přizpůsobí se PAM konfigurace (viz [25.11 – „pam\\_unix2.conf přizpůsobený pro LDAP“](#) (strana 383)).

**Rovnice 25.11** *pam\_unix2.conf přizpůsobený pro LDAP*

```
auth:      use_ldap nullok
account:   use_ldap
password:  use_ldap nullok
session:   none
```

Pokud nastavujete ručně další služby, aby používaly LDAP, vložte PAM LDAP modul do PAM konfiguračního souboru odpovídajícího dané službě v adresáři `/etc/pam.d`. Konfigurační soubory upravené pro jednotlivé služby lze nalézt v adresáři `/usr/share/doc/packages/pam_ldap/pam.d/`. Zkopírujte potřebné soubory do adresáře `/etc/pam.d`.

`glibc` rozpoznávání jmen mechanismem `nsswitch` se nasazení LDAP přizpůsobuje pomocí `nss_ldap`. V adresáři `/etc/` je při instalaci tohoto balíčku vytvořen nový přizpůsobený soubor `nsswitch.conf`. Více se o práci s `nsswitch.conf` dozvíte v části [18.6.1 – „Konfigurační soubory“](#) (strana 308). V souboru `nsswitch.conf` musí být řádky uvedené v příkladu [25.12 – „Přizpůsobení v souboru nsswitch.conf“](#) (strana 383).

**Rovnice 25.12** *Přizpůsobení v souboru nsswitch.conf*

```
passwd: compat
group: compat

passwd_compat: ldap
group_compat: ldap
```

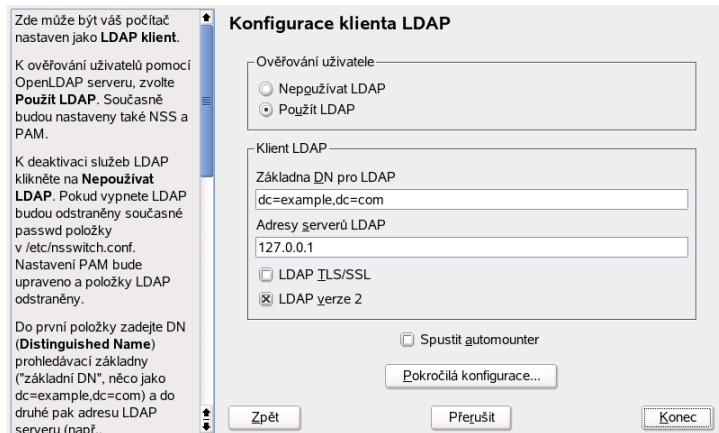
Tyto řádky příkazují resolver knihovně `glibc` nejprve vyhodnotit soubory v adresáři `/etc` a pak se připojit k LDAP serveru jako zdroji autentizačních a uživatelských dat. Mechanismus můžete otestovat přečtením uživatelské databáze příkazem `getent passwd`. Výsledek by měl obsahovat lokální uživatele vašeho systému i uživatele uložené na LDAP serveru.

Abyste zabránili běžným uživatelům spravovaným přes LDAP přihlásit se k serveru pomocí `ssh` nebo `login`, musí soubory `/etc/passwd` a `/etc/group` obsahovat následující řádek: `+:::/:sbin/nologin v /etc/passwd a +::: v /etc/group`.

## 25.5.2 Konfigurace LDAP klienta

Jakmile jsou `nss_ldap`, `pam_ldap`, `/etc/passwd` a `/etc/group` YaSTem upraveny, lze pokračovat v konfiguraci za pomoci prvního dialogu modulu YaST. Viz obrázek 25.2 – „YaST: Konfigurace LDAP klienta“ (strana 384).

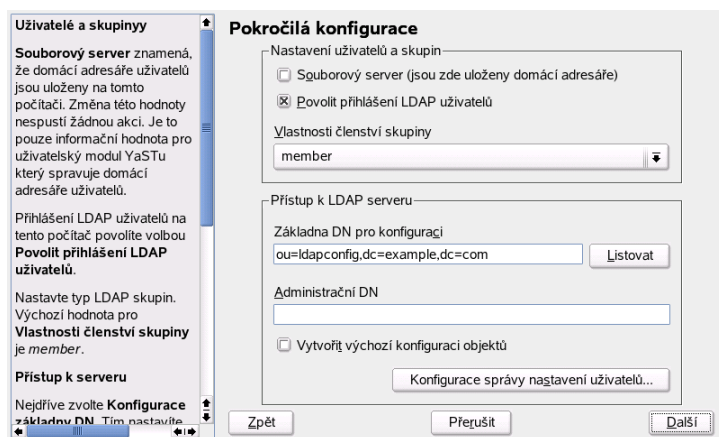
**Obrázek 25.2** YaST: Konfigurace LDAP klienta



V prvním dialogu aktivujte použití LDAP pro autentizaci uživatelů. V položce *Základna DN pro LDAP* zadejte prohledávací základnu, ve které jsou na serveru uložená data. IP adresu LDAP serveru zadejte v položce *Adresy serverů LDAP*. Můžete zadat více serverů oddělených mezerou. Chcete-li automaticky připojovat adresáře, zaškrtněte *Spustit automounter*. Chcete-li jako administrátor upravit data na serveru, klikněte na *Pokročilá konfigurace*. Viz obrázek 25.3 – „YaST: Pokročilá konfigurace“ (strana 385).



## Obrázek 25.3 YaST: Pokročilá konfigurace



Další dialog má dvě části: V horní části lze provést obecné nastavení uživatelů a skupin. V dolní části se nastavují data potřebná pro přístup k LDAP serveru. Nastavení uživatelů a skupin obsahuje následující položky:

### Souborový server

Pokud je aktuální systém souborový server pro uživatelské adresáře (`/home`), zvolte tuto volbu.

### Povolit přihlášení LDAP uživatelů

Povolením této volby umožníte uživatelům spravovaným přes LDAP přihlásit se do vašeho systému.

### Vlastnosti členství skupiny

Zde nastavte typ LDAP skupiny. Výchozí je *member*, další možností je *uniquemember*.

V dolní části nastavte údaje potřebné pro konfiguraci a přístup k LDAP serveru, tj. *Základna DN pro konfiguraci*, pod kterou jsou uloženy všechny konfigurační objekty, a *Administrační DN*.

Chcete-li editovat položky na serveru, klikněte na *Konfigurace správy nastavení uživatelů*. V dialogu, který se objeví, zadejte heslo pro autentizaci na serveru. Bude vám umožněn přístup ke konfiguračním modulům na serveru v souladu s ACL a ACI.

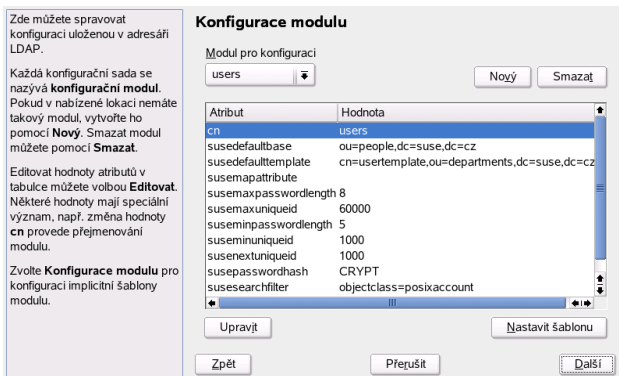
---

## Důležité: Použití YaST klienta

YaST LDAP klienta použijte k přizpůsobení YaST modulů pro správu uživatelů a skupin a k jejich případnému rozšíření. Navíc je možné definovat předlohy s výchozími hodnotami jednotlivých atributů pro usnadnění registrace údajů. Tato nastavení jsou sama uložena jako LDAP objekty v LDAP adresáři. Registrace uživatelských dat je stále prováděna pomocí běžných YaST formulářů. Údaje se ukládají jako objekty v LDAP adresáři.

---

**Obrázek 25.4** YaST: Konfigurace modulu



V dialogu pro konfiguraci modulu (25.4 – „YaST: Konfigurace modulu“ (strana 386)) lze vybírat a upravovat existující konfigurační moduly a vytvářet a upravovat šablony. Chcete-li upravit hodnotu v konfiguračním modulu nebo modul přejmenovat, vyberte příslušný modul v nabídce. Objeví se seznam všech jeho povolených atributů i s hodnotami. Obsahuje i atributy povolené schématem, ale nepoužité.

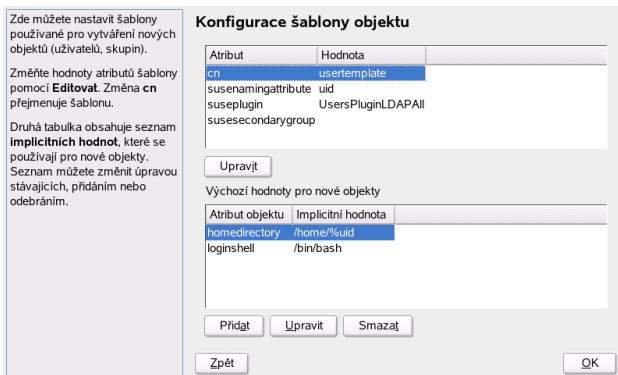
Chcete-li změnit hodnotu atributu, vyberte atribut ze seznamu a klikněte na *Upravit*. Provedené změny potvrdíte tlačítkem *OK*.

Chcete-li přidat nový modul, klikněte na *Nový*. Zadejte jméno a objektovou třídu nového modulu (buď `suseuserconfiguration` nebo `susegroupconfiguration`). Uzavřením dialogu tlačítkem *OK* přidáte nový modul do seznamu existujících modulů. Kliknutím na *Smazat* vybraný modul smažete.

Pokud byly předem definovány, obsahují YaST moduly pro správu uživatelů a skupin šablony se smysluplnými výchozími hodnotami. Chcete-li šablonu upravit, klikněte na *Nastavit šablonu*. Dialog pro nastavení šablon je rozdělen na dvě části. Horní část ob-

sahuje obecné atributy šablony. Upravte je podle potřeby a nebo nechte prázdné. Prázdné atributy budou na LDAP serveru smazány.

### Obrázek 25.5 YaST: Konfigurace šablony objektu



Druhá část (*Výchozí hodnoty pro nové objekty*) obsahuje všechny atributy odpovídajícího LDAP objektu (v tomto případě konfigurace uživatelů či skupin), pro které se definuje standardní hodnota. Lze přidávat nové a mazat již existující atributy a jejich standardní hodnoty, případně je měnit či mazat. Šablonu zkopírujete změnou hodnoty `cn`. Šablonu spojíte s modulem nastavením hodnoty atributu `suseDefaultTemplate` příslušného modulu na DN upravené šablony.

---

#### Tip

Výchozí hodnoty lze vytvářet z jiných atributů pomocí proměnných místo přímého zadání hodnoty. Například při vytváření nového uživatele lze použít `cn=%sn %givenName` a vytvářet tak automaticky hodnotu z `sn` a `givenName`.

---

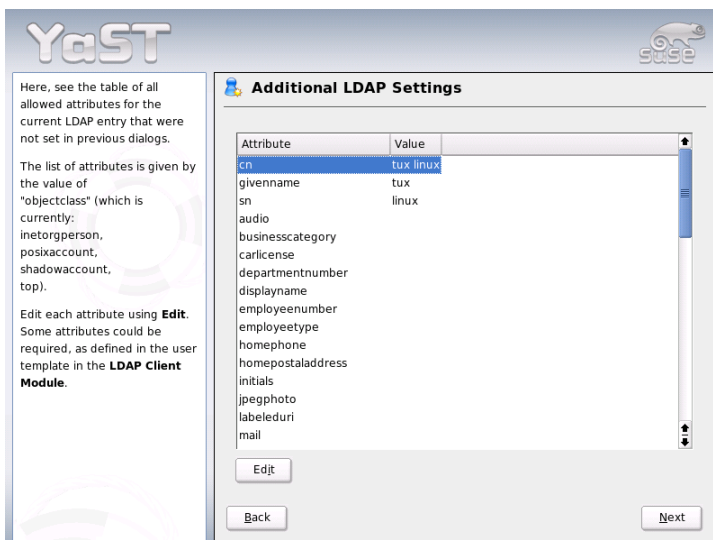
Jsou-li moduly a šablony správně nastaveny, můžete registrovat nové uživatele a skupiny běžným způsobem v nástroji YaST.

## 25.5.3 Uživatelé a skupiny — Konfigurace pomocí YaST

Registrace údajů o uživateli a skupinách se od postupu bez použití LDAP liší jen minimálně. Následující text se vztahuje k registraci uživatelů. Registrace skupin je analogická.

Spustíte YaST modul pro administraci uživatelů pomocí *Bezpečnost a uživatelé* → *Správce uživatelů*. Chcete-li prohlížet, přidávat či upravovat LDAP uživatele, klikněte na tlačítko *Nastavit filtr* vpravo dole a vyberte *LDAP uživatelé*. Při úpravě údajů o stávajícím uživateli nebo při zakládání nového uživatele pak máte v dialogu k dispozici kartu *Pluginy*. Kliknete-li v ní na *Upravit další vlastnosti LDAP uživatele* a pak na tlačítko *Spustit*, objeví se formulář pro zadání údajů specifických pro LDAP (25.6 – „YaST: Další LDAP nastavení“ (strana 388)). Vyberte atributy, jejichž hodnotu chcete upravit, a klikněte na *Upravit*. Završením dialogu, který se objeví po kliknutí na *Přijmout*, se vrátíte k hlavnímu dialogu správy uživatelů.

**Obrázek 25.6** YaST: Další LDAP nastavení



První dialog správy uživatelů obsahuje nabídku *LDAP volby*. Ta umožňuje použít vyhledávací LDAP filtry a nebo přejít do modulu pro konfiguraci LDAP uživatelů a skupin výběrem *Správa LDAP uživatelů a skupin*.

## 25.6 Další informace

Tato kapitola neobsahuje řadu témat, jako např. konfiguraci SASL nebo replikaci LDAP serveru, která umožňuje rozložit zatížení na několik strojů. Velmi vyčerpávajícím způsobem je toto nastavení popsáno v *OpenLDAP 2.1 Administrator's Guide* (viz níže).

Velmi rozsáhlou dokumentaci najdete přímo na stránkách projektu OpenLDAP:

### OpenLDAP FAQ-O-Matic

Sbírka otázek a odpovědí týkajících se instalace, konfigurace a správy OpenLDAP je dostupná na adrese <http://www.openldap.org/faq/data/cache/1.html>.

### Quick Start Guide

Jednoduchá instalační příručka LDAP serveru je dostupná na adrese <http://www.openldap.org/doc/admin21/quickstart.html> nebo přímo na vašem počítači v souboru `/usr/share/doc/packages/openldap2/admin-guide/quickstart.html`.

### OpenLDAP 2.2 Administrator's Guide

Detailní informace o konfiguraci LDAP včetně kontroly přístupu a šifrování. Příručka je dostupná na adrese <http://www.openldap.org/doc/admin22/> nebo přímo na vašem počítači v souboru `/usr/share/doc/packages/openldap2/admin-guide/index.html`

IBM vydalo o LDAP tyto červené knihy:

### Understanding LDAP

Základní principy LDAP. Kniha je dostupná na adrese <http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf>.

### LDAP Implementation Cookbook

Tato příručka je zaměřená především na administraci *IBM SecureWay Directory*. Obsahuje však také základní informace o LDAP. Naleznete ji na adrese <http://www.redbooks.ibm.com/redbooks/pdfs/sg245110.pdf>.

Tištěné knihy o LDAP:

- Howes, Smith, and Good: *Understanding and Deploying LDAP Directory Services*. Addison-Wesley, 2. Aufl., 2003. (ISBN 0-672-32316-8)

- Hodges: *LDAP System Administration*. O'Reilly & Associates, 2003. (ISBN 1-56592-491-6)

Vynikajícím referenčním manuálem pro LDAP jsou RFC dokumenty 2251–2256.

# Webový server Apache

Jedním z nejrozšířenějších webových serverů na všech platformách je Apache (zdroj: <http://www.netcraft.com>). Apache je často používán spolu s operačním systémem Linux, databází MySQL a programovacími jazyky PHP a Perl. Této kombinaci se často říká *LAMP*. V následující kapitole se vám pokusíme stručně přiblížit jeho principy, instalaci, základní konfiguraci a dostupné moduly. Jsou zmíněny i virtuální servery.

## 26.1 Instalace

Apache není součástí standardní instalace systému SUSE Linux. K instalaci použijte program YaST modul *Software* → *Správa softwaru*. Ve správě softwaru nastavte v části *Filtry* → *Výběry* a zvolte *Jednoduchý webový server Apache2*.

Apache se po tomto výběru nainstaluje s předdefinovaným nastavením, které umožňuje jeho okamžité spuštění bez nutnosti provádění složitějších nastavení. Instalace obsahuje modul `apache2-prefork` a PHP5 modul.

### 26.1.1 Moduly pro aktivní obsah

Abyste mohli používat aktivní obsah, musíte mít nainstalován modul s podporou příslušného jazyka, který se rozhodnete používat. K dispozici máte mimo PHP také např. `apache2-mod_perl` pro Perl, nebo `mod_python` pro Python. Použití těchto modulů je popsáno v části 26.6.2 – „Generování aktivního obsahu pomocí modulů“ (strana 402).

## 26.1.2 Další doporučené balíky

V některých případech je vhodné doinstalovat rozšířenou dokumentaci, kterou najdete v balíčku `apache2-doc`. Po instalaci balíčku a spuštění serveru lze k dokumentaci přistupovat přímo přes URL <http://localhost/manual>.

Pro vývoj nových modulů nebo jejich kompilaci potřebujete balíček `apache2-devel` a vývojové nástroje. Ty zahrnují `apxs` nástroje popsané v části [26.6.1 – „Instalace modulů pomocí apxs“](#) (strana 402).

## 26.2 Start serveru Apache

Aby se Apache spouštěl při startu systému, spusťte YaST a zvolte *Systém → Editor úrovní běhu*. Vyhledejte `apache2` a *povolte* službu. Webový server se spustí okamžitě. Po kliknutí na tlačítko *Ukončit* se bude webový server spouštět automaticky ve 3 a 5 úrovni běhu. Více informací o systému úrovní běhu v systému SUSE Linux a popis YaST Editoru úrovní běhu najdete v části [8.6 – „Editor úrovní běhu“](#) (strana 157).

Z příkazové řádky můžete Apache spustit okamžitě příkazem `rcapache2 start`. Aby se spouštěl automaticky při startu systému, zadejte příkaz `chkconfig -a apache2`.

Jestliže příkaz nevrátil žádné chybové hlášení, Apache je spuštěn. Testovací stránky si prohlédnete zadáním adresy <http://localhost/> do svého webového prohlížeče. Stránka bude začínat slovy: „If you can see this, it means that the installation of the Apache Web server software on this system was successful.“ Pokud váš prohlížeč žádnou stránku nezobrazil, prostudujte si kapitolu [26.9 – „Možné problémy“](#) (strana 407).

## 26.3 Konfigurace webového serveru

Pokud potřebujete zvláštní nastavení, proveďte je po instalaci Apache. V naprosté většině případů můžete Apache používat, jak je. Apache lze nastavit pomocí YaST a `SUSEconfig` nebo přímou editací souboru `/etc/apache2/httpd.conf`.



## 26.3.1 Konfigurace webového serveru pomocí YaST

Apache snadno nastavíte pomocí programu YaST. Nastavení vyžaduje alespoň základní znalosti o nastavení webového serveru. Po výběru *Síťové služby* → *HTTP server* vás může YaST před samotným nastavením webového serveru požádat o doinstalování potřebných balíčků. Po úspěšné instalaci se zobrazí konfigurační dialog.

Nejdřív povolte spuštění serveru zatrhnutím položky *Povoleno*. Zaškrtnutím *Na zvolených portech otevřít firewall* otevřete potřebné porty. Ve spodní části okna (*Nastavení/Shrnutí*) lze nastavit vlastnosti HTTP serveru: *Naslouchat na* (výchozí je `port 80`), *Moduly*, *Výchozí server* a *Servery*. Zvolenou položku změňte kliknutím na tlačítko *Upravit*.

Nejdřív překontrolujte nastavení položky *Výchozí server* a případně ji přizpůsobte svému serveru. Pak aktivujte potřebné moduly v položce *Moduly*. Dostupné jsou také další dialogy umožňující detailnější nastavení např. vytváření virtuálních serverů.

## 26.3.2 Ruční konfigurace webového serveru

Ruční konfigurace webového serveru představuje editaci textových souborů jako uživatel `root`.

### Konfigurační soubory

Konfigurační soubory Apache najdete na dvou místech:

- `/etc/sysconfig/apache2`
- `/etc/apache2/`

#### **`/etc/sysconfig/apache2`**

Soubor `/etc/sysconfig/apache2` obsahuje některá obecná nastavení webového serveru Apache jako zavádění modulů, přidávání souborů do konfigurace a příznaky pro spuštění serveru. Všechny volby jsou v souboru komentované. Přednastavené hodnoty jsou obvykle vhodné pro většinu běžných konfigurací.

---

## Důležité: SuSEconfig modul Apache

SuSEconfig modul Apache byl ze systému SUSE Linux odstraněn. Spuštění skriptu `SuSEconfig` po ručné změně `/etc/sysconfig/apache2` již není nutné.

---

### **/etc/apache2/**

Adresář `/etc/apache2/` obsahuje všechny konfigurační soubory Apache. V následujícím příkladu najdete popis jejich účelu. Každý soubor obsahuje konfigurační volby (nebo-li *příkazy*). Všechny příkazy jsou komentovány a proto je zde nebudeme zvlášť rozebírat.

Konfigurační soubory Apache jsou organizovány následujícím způsobem:

```
/etc/apache2/
|
| - charset.conv
| - conf.d/
|   |
|   |- *.conf
|
| - default-server.conf
| - errors.conf
| - httpd.conf
| - listen.conf
| - magic
| - mime.types
| - mod_*.conf
| - server-tuning.conf
| - ssl-global.conf
| - ssl.*
| - sysconfig.d
|   |
|   |- global.conf
|   |- include.conf
|   |- loadmodule.conf . .
|
| - uid.conf
| - vhosts.d
|   |- *.conf
```

### ***Konfigurační soubory Apache v /etc/apache2/***

`charset.conv`

nastavení znakových sad pro jednotlivé jazyky. Needitujte.

#### `conf.d/*.conf`

Konfigurační soubory vložené externími moduly. Tyto soubory lze začlenit do konfigurace virtuálního serveru. Příklad najdete v `vhosts.d/vhost.template`. Tímto způsobem můžete použít různé moduly pro různé virtuální servery.

#### `default-server.conf`

Obecná konfigurace virtuálních serverů. Místo změn hodnot voleb je přepíšete konfigurací virtuálního serveru.

#### `errors.conf`

Způsob odpovědi serveru Apache na chyby. Pokud chcete přizpůsobit chybová hlášení pro všechny virtuální servery, upravte zde obsažené soubory. Tato nastavení jsou přepsána nastavením jednotlivých virtuálních serverů.

#### `httpd.conf`

Hlavní konfigurační soubor serveru Apache. Zde můžete obecná nastavení Apache. Nastavení pro jednotlivé virtuální servery proveďte v konfiguraci jednotlivých virtuálních serverů.

#### `listen.conf`

Nastavení spojení IP adres a portů pro server. Také zde najdete nastavení virtuálních serverů založených na jméně (viz „[Virtuální server založený na jméně](#)“ (strana 396)).

#### `magic`

Data pro `mime_magic` modul, který umožňuje server Apache automaticky zjistit MIME typ neznámých souborů. Neměňte.

#### `mime.types`

Seznam známých MIME typů (odkaz na `/etc/mime.types`). Neměňte. Pokud potřebujete přidat nový MIME typ, vložte jej do souboru `mod_mime-defaults.conf`.

#### `mod_*.conf`

Konfigurační soubory pro výchozí moduly. Více informací o modulech najdete v části 26.6 – „[Moduly Apache](#)“ (strana 401). Konfigurace dodatečně dotaných souborů se nachází v adresáři `conf.d`.

#### `server-tuning.conf`

Konfigurace pro různé MPM a volby ovlivňující výkon Apache. Pokud zde provedete změny, nezapomeňte server řádně otestovat.

`ssl-global.conf` and `ssl.*`

Obecné nastavení SSL a data SSL certifikátu.

`sysconfig.d/*.conf`

Konfigurační soubory automaticky generované z `/etc/sysconfig/apache2`. Neměňte. Nevkládejte nové soubory. Pokud potřebujete změnit nastavení, změňte přímo `/etc/sysconfig/apache2`.

`uid.conf`

Nastavení ID uživatele a skupiny Apache. Neměňte.

`vhosts.d/*.conf`

Konfigurace virtuálních serverů. Adresář obsahuje šablonu pro virtuální server bez podpory SSL. Každý zde obsažený `.conf` soubor je automaticky připojen do nastavení serveru Apache. Podrobnosti o virtuálních serverech najdete v části „[Virtuální servery](#)“ (strana 396).

## Virtuální servery

Virtuální servery umožňují hostovat na jednom počítači více domén. Je to spolehlivý a ověřený způsob, jak ušetřit náklady na administraci zvláštního serveru pro každou doménu. Apache nabízí hned několik možností, jak virtuální servery nastavit:

- Virtuální server založený na jménu.
- Virtuální server založený na IP.
- Vícenásobné instance Apache na jednom počítači.

### Virtuální server založený na jménu

Virtuální server založený na jménu hostuje na jedné instanci Apache několik domén. Není nutné nastavovat žádné další IP adresy. Jedná se o nejjednodušší a nejčastěji používanou možnost. Důvody proti této konfiguraci najdete v dokumentaci Apache.

Konfigurace se provádí pomocí programu YaST nebo přímo v konfiguračním souboru `/etc/apache2/httpd.conf`. Abyste aktivovali virtuální server založený na jménu, musíte zadat `NameVirtualHost *`. Nastavení `*` způsobí, že bude Apache přijímat všechny příchozí požadavky. Pak nastavte jednotlivé servery:

```

<VirtualHost *>
    ServerName www.example.com
    DocumentRoot /srv/www/htdocs/example.com
    ServerAdmin webmaster@example.com
    ErrorLog /var/log/apache2/www.example.com-error_log
    CustomLog /var/log/apache2/www.example.com-access_log common
</VirtualHost>

<VirtualHost *>
    ServerName www.myothercompany.com
    DocumentRoot /srv/www/htdocs/myothercompany.com
    ServerAdmin webmaster@myothercompany.com
    ErrorLog /var/log/apache2/www.myothercompany.com-error_log
    CustomLog /var/log/apache2/www.myothercompany.com-access_log common

</VirtualHost>

```

`VirtualHost` musí být nastaven i pro originální doménu serveru ([www.example.com](http://www.example.com)). Originální doména i dodatečná doména ([www.myothercompany.com](http://www.myothercompany.com)) jsou v našem příkladě hostovány na stejném serveru.

Stejně jako v `NameVirtualHost` je v direktivách `VirtualHost` použita `*`. Apache používá pole "host" v HTTP hlavičce pro spojení požadavků s virtuálním serverem. Požadavek je doručen tomu virtuálnímu serveru, jehož nastavení v `ServerName` odpovídá údajům v HTTP hlavičce.

Pro direktivy `ErrorLog` a `CustomLog` nemusí záznamy obsahovat jméno domény. Použijte jméno podle vlastní volby.

`ServerAdmin` obsahuje e-mailovou adresu osoby, která má být kontaktována v případě problémů. Apache tuto adresu předává klientům v případě potíží.

## Virtuální server založený na IP

Alternativou serveru založeného na jménu je nastavení více IP adres pro jeden jediný počítač. V takovém případě jediná instance Apache hostuje více domén s různými IP adresami. V následujícím příkladu si ukážeme konfiguraci Apache používajícího vlastní IP adresu (192.168.1.10) plus další dvě dodatečné IP adresy (192.168.1.20 a 192.168.1.21). Tento konkrétní příklad funguje pouze na intranetu, protože se jedná o privátní adresy, které nejsou na Internetu směrovány.

Aby Apache mohl pracovat s více IP, musí počítač přijímat požadavky na více IP. Tomu se říká multi-IP hosting. Tato funkce vyžaduje podporu IP aliasingu v jádře. Tato podpora je v SUSE Linuxu výchozí.

Pokud je v jádře povolen IP aliasing, lze pomocí příkazů `ifconfig` a `route` nastavit další IP adresy počítače. Tyto příkazy musí vykonávat uživatel `root`. V následujícím příkladě budeme předpokládat, že počítač již má vlastní IP adresu (např. `192.168.1.10`), která je přiřazena zařízení `eth0`.

Příkazem `ifconfig` bez parametrů zjistíte IP adresu počítače. Další IP nastavíte příkazem:

```
ip addr add 192.168.1.20/24 dev eth0
```

Všechny IP adresy používají stejné síťové fyzické zařízení (`eth0`).

## Virtuální počítače s IP

Jakmile je na počítači nastaveno IP aliasování nebo má počítač více síťových karet, můžete nastavit virtuální servery Apache. Pro každý virtuální server musíte vložit vlastní blok `VirtualHost`:

```
<VirtualHost 192.168.1.20>
    ServerName www.myothercompany.com
    DocumentRoot /srv/www/htdocs/myothercompany.com
    ServerAdmin webmaster@myothercompany.com
    ErrorLog /var/log/apache2/www.myothercompany.com-error_log
    CustomLog /var/log/apache2/www.myothercompany.com-access_log common
</VirtualHost>

<VirtualHost 192.168.1.21>
    ServerName www.anothercompany.com
    DocumentRoot /srv/www/htdocs/anothercompany.com
    ServerAdmin webmaster@anothercompany.com
    ErrorLog /var/log/apache2/www.anothercompany.com-error_log
    CustomLog /var/log/apache2/www.anothercompany.com-access_log common
</VirtualHost>
```

Proměnná `VirtualHost` se používá pouze pro dodatečné domény. Výchozí doména ([www.example.com](http://www.example.com)) je nastavena zvlášť v `DocumentRoot` mimo bloky `VirtualHost`.

## 26.4 Používání Apache

Abyste zobrazili statické webové stránky, stačí je umístit do správného adresáře. V systému SUSE Linux jde o adresář `/srv/www/htdocs`. Několik pokusných stránek

je zde již nainstalováno. Tak si můžete ověřit, zda Apache běží správně. Tyto soubory můžete přepsat nebo smazat. Pro běh Apache nejsou nutné. CGI skripty jsou instalovány do `/srv/www/cgi-bin`.

Během svého běhu Apache zapisuje zprávy do souborů `/var/log/httpd/access_log` nebo `/var/log/apache2/access_log`. V těchto zprávách je uvedeno, jaké zdroje byly žádány, jaké doručeny, v jakém čase a jakou metodou (GET, POST atd.). Chybové zprávy jsou zapisovány do souboru `/var/log/apache2`.

## 26.5 Aktivní obsah

Apache nabízí několik způsobů, jak klientovi doručit aktivní obsah. Aktivní obsah HTML stránek je generován v závislosti na datech získaných od klienta. Např. vyhledávače poskytují seznam stránek na základě dotazu uživatele.

Apache generuje aktivní obsah třemi způsoby:

### SSI (Server Side Includes)

Jde o příkazy přímo v HTML stránce zapsané jako speciální komentáře. Apache komentáře interpretuje, vytvoří příslušný obsah a výsledek pošle jako část HTML stránky.

### CGI (Common Gateway Interface)

Programy v určitém adresáři. Apache jim předá parametry obdržené od klienta a klientovi vrátí výstup těchto programů. To je poměrně jednoduchý způsob, neboť lze snadno přizpůsobit mnoho existujících programů pro příkazovou řádku, aby takto spolupracovaly s Apachem.

### Moduly

Apache nabízí rozhraní pro vykonání jakéhokoliv modulu. Moduly jsou programy pracující s informacemi získanými od Apache. Apache umožňuje modulům přístup k důležitým informacím jako HTTP hlavičkám. Moduly lze použít kromě generování aktivních stránek také k jiným funkcím (například ověřování uživatele). Jejich výhodou je vysoký výkon a možnosti překonávající SSI i CGI. Podrobnější informace o modulech najdete v části [26.6 – „Moduly Apache“](#) (strana 401)

Normálně jsou CGI skripty vykonávány přímo serverem Apache pod uživatelským ID jejich vlastníka. Naopak moduly jsou kontrolovány interpretem, který je v serveru

Apache obsažen. Není tak nutné pro každý dotaz spouštět a ukončovat samostatný proces (což zvyšuje zátěž). Skript je interpretem spuštěn pod ID webserveru.

Toto řešení má i své chyby. CGI skripty jsou totiž oproti modulům velmi robustní. Při jejich použití nemají chyby při správě zdrojů a paměti tak ničivé následky jako u modulů, neboť dojde k ukončení programu po vyřízení požadavku. Při použití modulů může dojít ke kumulaci chyb. Pokud server běží bez restartu delší dobu, mohou se chyby hromadit a vést k nestabilitě systému.

## 26.5.1 SSI

Server-side includes jsou příkazy ve zvláštních komentářích vykonávané Apachem. Výsledek je zahrnut ve výstupu. Například aktuální datum lze zahrnout pomocí `<!--#echo var="DATE_LOCAL" -->`. Znak # na konci otevírací značky (`<!--`) říká indiánovi, že se jedná o SSI direktivu a nikoliv o obyčejný komentář.

SSI lze aktivovat několika způsoby. Nejjednodušší je vyhledat SSI ve všech spustitelných souborech. Jiná možnost je určit, ve kterých souborech se má SSI hledat.

## 26.5.2 CGI

CGI je zkratka z anglického *Common Gateway Interface*. Díky CGI je server schopný zasílat mimo klasických statických stránek také dynamicky generované stránky. Tak je možné vytvářet stránky, které jsou výsledkem výpočtu nebo hledání v databázi. V závislosti na obdržené proměnné je server schopný vytvářet na každý dotaz zvláštní stránky lišící se obsahem.

Hlavní výhodou technologie CGI je jednoduchost. Programy jsou obvykle uloženy v určitém adresáři a spouštěny serverem jako jakékoliv jiné programy v systému. Server pak zašle výstup programu ze standardního výstupu (`stdout`) klientovi.

Teoreticky mohou být CGI napsány v libovolném programovacím jazyce. Obvykle jsou k tomuto účelu používány skriptovací jazyky jako Perl nebo PHP. Pokud je rychlost kritická, může být vhodnější C/C++.

V nejjednodušším případě hledá indián tyto programy ve zvláštním adresáři (`cgi-bin`). Ten lze nastavit v konfiguračním souboru (viz 26.3 – „[Konfigurace webového serveru](#)“ (strana 392)). Pokud je potřeba, mohou být nastaveny další takové adresáře. Je však



nebezpečné umožnit Apache spouštět programy uživatele. Pokud jsou CGI omezeny na adresář `cgi-bin`, může administrátor lépe kontrolovat jejich obsah.

## 26.6 Moduly Apache

Pomocí modulů lze Apache rozšířit o řadu funkcí např. o schopnost pracovat s CGI skripty v různých jazycích. Mimo tradičních jazyků jako Perl a PHP jsou k dispozici také jazyky Python a Ruby. Použít lze mimo jiné i moduly pro bezpečný přenos dat (secure sockets layer - SSL), ověřování uživatelů, rozšířené logování a mnoho dalších.

S dostatkem know-how můžete Apache pomocí vlastních modulů přizpůsobit libovolným požadavkům. Více informací najdete v části [26.10.5 – „Další zdroje“](#) (strana 409).

Modularizace Apache dospěla tak daleko, že je moduly řešeno v podstatě vše kromě nejjednodušších úkolů. Dospělo to tak daleko, že dokonce samotné HTTP je zpracováváno moduly. Apache proto vůbec nemusí fungovat jako webserver. S patřičnými moduly může sloužit úplně jiným účelům. Byl například nasazen jako poštovní server (POP3).

Moduly Apache podporují řadu dalších užitečných funkcí:

### Virtuální servery

Podpora funkce virtuálního serveru znamená, že na jednom počítači s jednou instancí Apache lze provozovat více webů, které se návštěvníkům jeví jako samostatné servery. Virtuální servery mohou používat různé IP adresy nebo jména. Tak ušetříte výdaje za další hardware a software.

### Flexibilní přepis URL

Apache nabízí řadu možností, jak manipulovat a přepisovat URL. Více informací najdete v dokumentaci Apache.

### Content Negotiation

Apache umí klientovi (prohlížeči) doručit stránku ve stavu, který odpovídá jeho zobrazovacím schopnostem. Například starým prohlížečům nepodporujícím rámce pošle stránku bez rámců. Pokud jste ochotni připravit JavaScript zvlášť pro každý typ prohlížeče, můžete takto obejít případné nekompatibility v jeho implementaci.

Flexibilní nakládání s chybami

Apache na chybu, například chybějící stránku, dokáže reagovat flexibilně a odpovídajícím způsobem. Odpověď je možno generovat i dynamicky, například pomocí CGI.

## 26.6.1 Instalace modulů pomocí apxs

Příkaz `apxs2` je důležitý nástroj pro vývojáře modulů. Díky tomuto příkazu je možné jedním příkazem překompilovat i nainstalovat požadovaný nový modul (včetně provedení potřebných změn v konfiguračních souborech). Tímto příkazem lze instalovat také moduly dostupné jako objektové soubory (koncovka `.o`) nebo statické knihovny (koncovka `.a`). Ze zdrojového kódu příkaz `apxs2` vytvoří DSO (Dynamic Shared Object), který může Apache používat jako modul.

Instalaci modulu ze zdrojového kódu lze provést příkazem jako `apxs2 -c -i -a mod_foo.c`. Další volby tohoto příkazu jsou popsány v manuálové stránce.

`apxs2` je dostupný v několika verzích: `apxs2`, `apxs2-prefork` a `apxs2-worker`. `apxs2` instaluje moduly tak, aby je mohly používat všechny MPM. Ostatní programy instalují moduly tak, že mohou být používány pouze příslušnými MPM. `apxs2` instaluje moduly do `/usr/lib/apache2`. `apxs2-prefork` instaluje moduly do `/usr/lib/apache2-prefork`.

## 26.6.2 Generování aktivního obsahu pomocí modulů

Pro webový server Apache je dostupných mnoho různých modulů. Termín *modul* je zde používán ve dvou různých významech. První představuje moduly integrované přímo do Apache a ošetřující zvláštní funkce, jako je podpora programovacích jazyků.

Druhý význam je spojen s programovacími jazyky. Moduly zde odkazují na nezávislou skupinu funkcí, tříd a proměnných. Tyto moduly jsou integrovány do programu a poskytují různé funkce, jako např. CGI moduly pro skriptovací jazyky. Tyto moduly umožňují CGI programování poskytováním různých funkcí, jako jsou metody čtení parametrů dotazů a metody pro HTML výstup.

## 26.6.3 mod\_perl

Perl je populární a prověřený skriptovací jazyk. Existuje pro něj řada modulů a knihoven včetně knihovny pro rozšíření konfiguračního souboru Apache. Domovská stránka Perlu se nachází na adrese <http://www.cpan.org/>. Řada knihoven je dostupná v Comprehensive Perl Archive Network (CPAN) na adrese <http://www.cpan.org/>.

### Nastavení mod\_perl

Modul `mod_perl` nastavíte instalací příslušného balíčku (viz 26.1 – „Instalace“ (strana 391)). Po instalaci se v konfiguračním souboru automaticky objeví všechny důležité položky (viz `/etc/apache2/mod_perl-startup.pl`). Informace o `mod_perl` jsou dostupné na stránce <http://perl.apache.org/>.

### mod\_perl versus CGI

V nejjednodušším případě spustíte předešlý CGI skript jako `mod_perl` skript dotazem z jiné adresy. Konfigurační soubor obsahuje aliasy, které odkazují na stejný adresář a vykonají každý zde obsažený skript prostřednictvím buď CGI nebo `mod_perl`. Všechny položky již v konfiguračním souboru existují. Alias pro CGI je:

```
ScriptAlias /cgi-bin/ "/srv/www/cgi-bin/"
```

Položky pro `mod_perl` jsou:

```
<IfModule mod_perl.c>
    # Provide two aliases to the same cgi-bin directory,
    # to see the effects of the 2 different mod_perl modes.
    # for Apache::Registry Mode
    ScriptAlias /perl/          "/srv/www/cgi-bin/"
    # for Apache::Perlrun Mode
    ScriptAlias /cgi-perl/     "/srv/www/cgi-bin/"
</IfModule>
```

Pro `mod_perl` jsou potřebné také následující položky. Tyto položky se již v konfiguračním souboru nacházejí.

```
#
# If mod_perl is activated, load configuration information
#
<IfModule mod_perl.c>
PerlRequire /usr/include/apache/modules/perl/startup.perl
PerlModule Apache::Registry
```

```

#
# set Apache::Registry Mode for /perl Alias
#
<Location /perl>
SetHandler perl-script
PerlHandler Apache::Registry
Options ExecCGI
PerlSendHeader On
</Location>

#
# set Apache::PerlRun Mode for /cgi-perl Alias
#
<Location /cgi-perl>
SetHandler perl-script
PerlHandler Apache::PerlRun
Options ExecCGI
PerlSendHeader On
</Location>

</IfModule>

```

Tyto položky vytvoří aliasy pro režimy *Apache::Registry* a *Apache::PerlRun*. Rozdíly mezi těmito režimy jsou následující:

#### *Apache::Registry*

Všechny skripty jsou překompilovány a uloženy do vyrovnávací paměti. Každý skript je pak používán jako obsah subrutiny. Přestože tak získáte vysoký výkon, jsou zde i nevýhody. Skript je nutné napsat s extrémní opatrností, protože proměnné a subrutiny mezi jednotlivými požadavky přetrvávají. Znamená to, že vždy musíte každou proměnnou ošetřit tak, aby se před použitím rutiny dalším dotazem vynulovala. Například pokud ve skriptu uložíte jako proměnnou číslo bankovní karty, bez vynulování se může stát, že se číslo karty použije i u dalšího zákazníka.

#### *Apache::PerlRun*

Skripty jsou pro každý požadavek recompileovány. Všechny proměnné mezi požadavky mizí. Proto *Apache::PerlRun* nevyžaduje tak pečlivé programování, ale je pomalejší než *Apache::Registry*. Stále je však mnohem rychlejší než CGI (navzdory podobnostem), protože není spouštěn zvláštní proces pro interpret.

## 26.6.4 mod\_php4

PHP je jazyk vyvinutý speciálně pro webové servery. Na rozdíl od jiných jazyků, které využívají pro své příkazy samostatné soubory (skripty), PHP lze vložit přímo do HTML

stránky (podobně jako SSI). PHP interpret zpracuje vložené PHP příkazy a vygeneruje výsledek do webové stránky.

Domovskou stránku PHP najdete na adrese <http://www.php.net/>. Pro použití PHP musíte nainstalovat balíčky `mod_php4-core` a `apache2-mod_php4`.

## 26.6.5 mod\_python

Python je objektově orientovaný jazyk s velmi jasnou a čitelnou syntaxí. Neobvyklou ale velmi užitečnou vlastností je struktura programu závislá na odsazení. Jednotlivé bloky od sebe nejsou odděleny složenými závkami (jako v C a Perlu) ani jinými oddělovači (jako `begin` a `end`), ale stupněm odsazení. Pro podporu hada potřebujete balíček `apache2-mod_python`.

Více informací o tomto jazyce najdete na stránce <http://www.python.org/>. Informace o `mod_python` jsou dostupné na <http://www.modpython.org/>.

## 26.6.6 mod\_ruby

Ruby je poměrně nový objektově orientovaný jazyk s prvky Perlu a Pythonu. Stejně jako Python má jasnou a transparentní syntaxi. Na druhou stranu obsahuje zkratky jako `$.r` pro číslo poslední řádky načtené ze vstupního souboru, což je vlastnost, kterou někteří programátoři vítají a jiní nenávidí. Koncept Ruby částečně převzal ze Smalltalku.

Domovskou stránku Ruby najdete na adrese <http://www.ruby-lang.org/>. Apache modul má domovskou stránku <http://www.modruby.net/>.

## 26.7 Vlákna (threads)

Vlákno je jednoduchý proces. Výhoda vláken leží v nižší spotřebě zdrojů, čímž se zvyšuje výkon. Nevýhodou je, že aplikace musí být tzv. thread-safe. To znamená:

- Funkce (nebo metody v objektově orientovaných aplikacích) musí být reentrantní (vícenásobně přístupné) – funkce se stejným vstupem vždy vrátí stejný výstup, i když je současně vykonávána jiným vláknem. Funkce tedy musí být navrženy tak, aby mohly být vykonávány současně více vlákny.

- Přístup ke zdrojům (obvykle proměnným) musí být řízen tak, aby současně běžící vlákna nepřicházela do konfliktu.

Apache 2 přistupuje k dotazům jako odděleným procesům, nebo, ve smíšeném režimu, jako kombinaci procesů a vláken. Za zpracování dotazů jako procesů zodpovídá MPM *prefork*, za zpracování jako vláken MPM *worker*. Výběr MPM můžete provést při instalaci (viz 26.1 – „Instalace“ (strana 391)). Třetí režim – *perchild* – není zatím vyzrálý a není proto v naší distribuci dostupný.

## 26.8 Bezpečnost

Pokud Apache nepotřebujete, deaktivujte jeho spouštění v editoru úrovní běhu nebo ho oddinstalujte. Pokud chcete bezpečnostní rizika minimalizovat úplně, vypněte i další serverové služby. Platí to zejména pro počítače používané jako firewally. Na těch pokud možno nespouštějte žádné služby.

### 26.8.1 Přístupová práva

Jako výchozí vlastník adresáře *DocumentRoot* (`/srv/www/htdocs`) a adresáře CGI je nastaven uživatel `root`. Pokud je adresář zapisovatelný pro všechny, může do něj umístit soubory jakýkoliv uživatel. Tyto soubory pak budou vykonány Apachem pod uživatelem `wwwrun`. Apache by neměl mít práva zápisu do adresářů s daty a skripty, které dodává. Proto by neměl být vlastníkem těchto adresářů uživatel `wwwrun`, ale jiný uživatel (např. `root`).

Aby mohli do adresáře s dokumenty umístit své soubory také jiní uživatelé, musí mít práva k zápisu. Takové řešení však není bezpečné. Pokud máte možnost, vytvořte raději nový adresář, kam budou mít práva zápisu všichni (např. `/srv/www/htdocs/miscellaneous`).

### 26.8.2 Publikování dokumentů z domovských adresářů

Jiný způsob, jak zajistit, aby uživatelé mohli publikovat své stránky, je určení jednoho přesného jména adresáře v domovském adresáři, kam se mají stránky určené k publikaci

ukládat. Jméno tohoto podadresáře je obvykle `~/public_html`. To je výchozí nastavení v systému SUSE Linux.

Webové stránky pak můžete zobrazit zadáním jména uživatele v URL, pomocí části `~uživatel`. K zobrazení obsahu adresáře `public_html` uživatele `tux` zadejte do prohlížeče adresu <http://localhost/~tux>.

## 26.8.3 Aktualizace

Pokud provozujete webový server, který je veřejně přístupný, nezanedbávejte pravidelnou aktualizaci. Snažte se pravidelně získávat informace o bezpečnostních chybách a problémech. Zdroje, které vám v tom pomohou, najdete v části [26.10.4 – „Bezpečnost“](#) (strana 409).

## 26.9 Možné problémy

Pokud se Apache nespustí, stránky se nezobrazují nebo se uživatel nemůže připojit k serveru, je nutné nejdřív najít příčinu problémů. Zde si ukážeme postupy hledání a nejčastější chyby.

Příkaz `rcapache2` umí vypisovat chybová hlášení dejte mu proto přednost před přímým spouštěním serveru příkazem `/usr/sbin/httpd2`. Spolu s chybovým hlášením tak získáte také tipy na jejich odstranění.

Nepodceňujte význam logů. Právě v nich najdete nejdůležitější informace o stavu a problémech webového serveru. Stupeň logování navíc můžete ovlivnit příkazem `LogLevel`. Výchozím souborem s logy je `/var/log/apache2/error_log`.

---

### Tip: Jednoduchý test

Hlášení přímo při běhu serveru můžete sledovat příkazem `tail -F /var/log/apache2/error_log`. Stačí zadat tento příkaz a v jiném terminálu spustit server příkaz. Funkci si můžete prověřit tak, že si necháte zobrazit stránku poskytovanou serverem ve webovém prohlížeči.

---

Častým problémem je nepovolení portu Apache ve firewallu. Jestliže konfigurujete Apache pomocí programu YaST, můžete použít zvláštní volbu určenou pro otevření

portu na firewallu. Pokud provádíte konfiguraci ručně, použijte pro otevření potřebných portů modul pro nastavení firewallu v programu YaST.

Podívejte se také do databáze chyb na stránce <http://bugs.apache.org/>. Přihlaste se do uživatelské konference Apache dostupné na adrese <http://httpd.apache.org/userslist.html>. Doporučujeme také novinky (newsgroup) <comp.infosystems.www.servers.unix>.

## 26.10 Další dokumentace

Apache je velmi rozšířený webservice. Proto existuje mnoho dokumentace a mnoho webových stránek nabízí nápovědu a podporu.

### 26.10.1 Apache

Apache je dodáván s velmi obsáhlou dokumentací. Instalace dokumentace je popsána v části 26.1 – „Instalace“ (strana 391). Po instalaci můžete k dokumentaci přistupovat prostřednictvím svého prohlížeče na adrese <http://localhost/manual>. Nejnovější dokumentaci najdete na domovské stránce Apache <http://httpd.apache.org>.

### 26.10.2 CGI

Více informací CGI získáte z těchto stránek:

- <http://apache.perl.org/>
- <http://perl.apache.org/>
- <http://www.modperl.com/>
- <http://www.modperlcookbook.org/>
- <http://www.fastcgi.com/>
- <http://www.boutell.com/cgiic/>



## 26.10.3 Moduly

Více informací o externích modulech Apache získáte na následujících stránkách:

FastCGI

<http://www.fastcgi.com/>

mod\_perl

<http://perl.apache.org/>

mod\_php5

<http://www.php.net/manual/en/install.unix.apache2.php>

mod\_python

<http://www.modpython.org/>

mod\_ruby

<http://www.modruby.net/>

## 26.10.4 Bezpečnost

Poslední opravy pro balíčky SUSE najdete na stránce <http://www.novell.com/linux/security/securitysupport.html>. Navštěvujte tuto adresu v pravidelných intervalech. Zde se také můžete přihlásit do e-mailové konference o bezpečnosti, v rámci které vám budou zasílána upozornění o bezpečnostních chybách a opravách.

Apache tým zcela otevřeně informuje o všech chybách. Oznamuje nejnověji objevené chyby a snaží se co nejdřív vydat příslušnou opravu na stránce [http://httpd.apache.org/security\\_report.html](http://httpd.apache.org/security_report.html). Pokud objevíte bezpečnostní chybu (předtím překontrolujte výše zmíněné stránky, zda již nebyla hlášena), pošlete nám prosím hlášení na email [feedback@suse.cz](mailto:feedback@suse.cz) nebo přímo (anglicky) na [security@apache.org](mailto:security@apache.org).

## 26.10.5 Další zdroje

V případě problémů navštivte Databázi instalační podpory na stránce <http://en.opensuse.org/SDB:SDB/>. Novinky o webovém serveru Apache najdete na stránce <http://www.apacheweek.com/>.

Historie Apache je popsána v dokumentu [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html). Zde najdete i důvod pro pojmenování *Apache*.

Informace o aktualizaci z 1.3 na 2.0 najdete na stránce <http://httpd.apache.org/docs-2.0/en/upgrading.html>.

# Synchronizace souborů

Řada lidí používá více počítačů najednou — jeden počítač doma, jeden nebo více počítačů v práci a laptop nebo PDA na cestách. Dříve či později budete potřebovat upravovat určitý soubor na všech počítačích, ale současně mít všude k dispozici aktuální verzi bez nutnosti ručního kopírování souborů.

## 27.1 Programy pro datovou synchronizaci

Pro počítače trvale připojené do rychlé sítě není synchronizace dat žádným problémem. V takovém případě je nejjednodušší cestou nasazení síťového souborového systému, jako je NFS, který umožňuje ukládat všechna data na serveru a přistupovat k nim z klientských stanic v síti. Toto řešení je však vyloučené v případě pomalejší nebo dočasné sítě. I na laptopu potřebujete lokální kopii všech důležitých souborů. Tehdy přichází na řadu synchronizace souborů. Ta zajistí, že pokud je soubor na jakémkoliv počítači změněn, dojde k aktualizaci souboru na všech ostatních počítačích. Automaticky lze synchronizaci provádět pomocí programů scp nebo rsync. Ne vždy je však tento způsob žádoucí, protože může dojít např. k přepisu novější verze starší.

---

### Varování: Riziko ztráty dat

Dřív než začnete používat systém k synchronizaci dat, seznamte se s funkcemi zvoleného programu a proveďte několik testů. U zvláště důležitých dat proveďte zálohu.

---

Ruční synchronizace je vysoce časově náročná a náchylná k chybám. Tomu lze předejít automatizací. Zde vám některé z programů, které takovou automatizaci umožňují, krátce představíme. Pokud se pro některý z nich rozhodnete, nezapomeňte si pročit jeho dokumentaci.

## 27.1.1 Unison

Unison není síťový souborový systém. Soubory jsou jednoduše ukládány a upravovány lokálně. Program Unison pak po ručním spuštění provede synchronizaci dat. Při první synchronizaci se na obou počítačích vytvoří databáze obsahující kontrolní součty, časová razítka a informace o přístupových právech jednotlivých zvolených souborů. Při dalším spuštění již program Unison rozpozná, které soubory se mají synchronizovat, a navrhne přenos na jiný počítač. Obvykle lze všechny návrhy akceptovat.

## 27.1.2 CVS

CVS je nejčastěji používán pro správu verzí zdrojových kódů programů. Nabízí možnost udržování kopie souborů na řadě počítačů. Použitelný je samozřejmě také pro synchronizaci dat. CVS spravuje centrální sklad dat na serveru. Neukládají se jen samotné soubory, ale také záznamy o změnách. Změny se provádějí lokálně a odesílají se do centrálního skladu odkud mohou být stahovány ostatními uživateli. Odeslání i stažení změn vyžaduje aktivní účast uživatele.

CVS je odolný proti chybám, které nastanou v případě současného odesílání ze dvou různých počítačů. Všechny změny spojuje, ale pokud ke změnám dojde současně na jedné řádce, nahlásí konflikt. Databáze zůstává i v případě konfliktu v konzistentním stavu. Konflikty jsou viditelné a řešitelné pouze na klientských stanicích.

## 27.1.3 subversion

Na rozdíl od CVS, které se vyvinulo živelně, je subversion pečlivě navržený projekt, technicky zdokonalený následník CVS.

Program subversion byl zdokonalen v mnoha směrech. CVS umí z historických důvodů pracovat jen se soubory a nikoliv s adresáři, zatímco subversion udržuje i historii adresářů, které lze kopírovat a přejmenovávat stejně jako soubory. Ke každému adresáři i souboru lze navíc přiřadit metadata, pro která je taktéž udržována historie verzí. Na

rozdíl od CVS podporuje subversion transparentní přístup přes speciální síťové protokoly, např. WebDAV (Web-based Distributed Authoring and Versioning). WebDAV rozšiřuje funkčnost HTTP protokolu o zápis do souborů na vzdálených webových serverech s možností spolupráce.

Při vývoji subversion byly využity již existující programy. Proto je společně se subversion vždy používán webserver apache a rozšíření WebDAV.

## 27.1.4 mailsync

Program mailsync se používá pouze k synchronizaci elektronické pošty ve schránkách na různých serverech. Synchronizovat lze jak lokální schránky, tak schránky IMAP.

Zprávy jsou synchronizovány či mazány v závislosti na ID zprávy obsaženém v hlavičce. Synchronizace je možná mezi jednotlivými schránkami nebo skupinami schránek.

## 27.1.5 rsync

Pokud není potřeba správa verzí, ale je potřeba synchronizovat rozsáhlé adresářové struktury přes pomalou síť, je vhodné použít nástroj rsync, který nabízí dobrý mechanismus pro přenos změn v souborech, a to nejen textových, ale i binárních. Aby rsync zjistil změny v souborech, rozdělí je na jednotlivé bloky, ze kterých spočítá kontrolní součty.

Zjišťování změn je poměrně náročná činnost. Systémy, na kterých se má synchronizace provádět, by měly být náležitě vybaveny. Důležitý je zejména dostatek operační paměti.

## 27.2 Výběr vhodného programu

Při výběru vhodného programu byste měli zvážit následující hlediska:

### 27.2.1 Klient-Server vs. Peer-to-Peer

Pro distribuci dat se používají dva odlišné modely. V prvním modelu všichni klienti synchronizují data s centrálním serverem, který musí být alespoň čas od času pro klienty dostupný. Tento model používá subversion, CVS a WebDAV.

Druhou možností je synchronizace dat mezi klienty navzájem. Tak pracuje např. unison. Program rsync obvykle pracuje v klientském režimu, ale každý klient může fungovat i jako server.

## 27.2.2 Přenositelnost

CVS, subversion a unison jsou dostupné také ve verzích pro jiné operační systémy včetně Unixu a Windows.

## 27.2.3 Interaktivní vs. automatický

V programech subversion, CVS, WebDAV a unison synchronizaci spouští uživatelé ručně. Mají nad ní tak větší kontrolu. Pokud však uživatelé synchronizují v příliš dlouhých intervalech, zvyšuje se pravděpodobnost konfliktu.

## 27.2.4 Konflikty: výskyt a řešení

Konflikty jsou v CVS a subversion vzácné i v případě spolupráce velkého množství lidí na rozsáhlém projektu. Je to díky tomu, že změny v souborech jsou slučovány po jednotlivých řádcích. Když konflikt přeci jen nastane, je postižen pouze jeden klient. Konflikty se v CVS i subversion dají obvykle snadno řešit.

Unison oznamuje konflikty a umožňuje vyjmout postižené soubory ze synchronizace. Slučování změn je však obtížnější než v aplikacích subversion a CVS.

Na rozdíl od subversion či CVS, ve kterých lze přijmout změny v případě konfliktu alespoň částečně, přijme WebDAV změny pouze pokud je vše v pořádku.

Aplikace rsync se o konflikty vůbec nestará. Uživatel je zodpovědný za ruční řešení veškerých konfliktů a za to, aby omylem nepřepsal žádné soubory. Na druhou stranu lze dodatečně zapojit systém správy verzí, jako např. RCS.

## 27.2.5 Výběr a vkládání souborů

Ve standardní konfiguraci synchronizuje unison celý adresářový strom. Nové soubory přidávané do adresářového stromu jsou automaticky synchronizovány.

V subversion nebo CVS musí být nové soubory explicitně přidány příkazem `svn add` či `cv add`. Znamená to větší uživatelskou kontrolu nad synchronizací, ale na druhou stranu se nové soubory často přehlédnou, zejména v případě, kdy je souborů mnoho a otazníky ve výstupu příkazů `svn update` a `svn status` nebo `cv update` nejsou uživatelem zpozorovány.

## 27.2.6 Historie

Další funkcí subversion a CVS je možnost rekonstrukce starých verzí. Ke každé změně je možno doplnit krátkou poznámku. Vývoj všech souborů lze později snadno vysledovat na základě záznamů o změně obsahu a poznámek. To je neocenitelná pomoc zejména v případě vědeckých prací a zdrojových programových kódů.

## 27.2.7 Objem dat a požadavky na diskový prostor

Při synchronizaci je nutné mít na všech klientech dostatek místa pro data. V případě subversion a CVS budete navíc potřebovat místo na serveru pro repositář. Historie souborů je také uložena na serveru a vyžaduje další prostor. U textových souborů se ukládají pouze pozměněné řádky. Binární soubory se ukládají celé, pro uložení každé změny tedy vyžadují tolik místa, kolik zabírá celý soubor.

## 27.2.8 GUI

Unison nabízí pro zobrazení navrhovaného postupu synchronizace grafické uživatelské prostředí. Můžete v něm návrh přijmout či vyjmout jednotlivé soubory ze synchronizace. V textovém režimu lze interaktivně přijímat jednotlivé procedury.

Zkušení uživatelé obvykle pracují se subversion či CVS přes příkazovou řádku. Pro Linux však k těmto programům existují i grafická prostředí, jako např. `cervisia`. V jiných operačních systémech existují podobné programy, např. `wincvs`. Mnoho vývojářských nástrojů, jako např. `kdevelop`, a textových editorů, jako např. `emacs`, podporuje CVS či subversion. Řešení konfliktů je s těmito nástroji obvykle o poznání jednodušší.

## 27.2.9 Uživatelská přívětivost

Programy unison a rsync se používají poměrně snadno a jsou vhodné pro začátečníky. CVS a subversion jsou poněkud obtížnější. Vyžadují, aby uživatel pochopil vztah mezi repositářem a lokálně umístěnými daty. Změny by nejprve měly být sloučeny s repositářem lokálně pomocí příkazu `cvsv update` nebo `svn update`. Pak musí být data odeslána zpět do repositáře příkazem `cvsv commit` nebo `svn commit`. Pokud uživatel pochopí tento princip, bude pro něj i použití CVS či subversion snadné.

## 27.2.10 Bezpečnost

Data by během přenosu měla být chráněna proti nedovolené manipulaci. Unison, subversion, CVS i rsync lze používat spolu s ssh (Secure Shell). Pokud chcete svým datům zajistit maximální bezpečnost, vyhněte se používání rsh (Remote Shell). V nedůvěryhodných nebo otevřených sítích nepoužívejte s CVS *pserver*. Program subversion použitý spolu se serverem apache již obsahuje bezpečnostní mechanismy.

## 27.2.11 Ochrana proti ztrátě dat

CVS je vývojáři používán velmi dlouho a je extrémně stabilní. Protože ukládá historii projektu, je CVS chráněn i proti chybám uživatelů jako je např. nechtěné smazání souboru. Ačkoliv není subversion tak rozšířená jako CVS, je již běžně nasazována do produkčního prostředí, například sama při svém vývoji.

Unison patří k novějším programům, ale vyznačuje se vysokou stabilitou. Je však mnohem citlivější na chyby uživatelů. Např. smazaný soubor nelze po synchronizaci obnovit.

**Tabulka 27.1** *Funkce synchronizačních nástrojů: -- = velmi nízká, - = nízká nebo žádná, o = střední, + = dobrá, ++ = výborná, x = dostupná*

	unison	CVS/subv.	rsync	mailsync
Klient/server	rovnocenné	C-S/C-S	C-S	rovnocenné
Přenositelnost	Lin,Un*x,Win	Lin,Un*x,Win	Lin,Un*x,Win	Lin,Un*x



	<b>unison</b>	<b>CVS/subv.</b>	<b>rsync</b>	<b>mailsync</b>
Interaktivita	x	x/x	x	-
Rychlost	-	o/+	+	+
Konflikty	o	++/++	o	+
výběr soub.	adresář	výběr/soub., adr.	adresář	mailbox
Historie	-	x/x	-	-
Místo na disku	o	--	o	+
GUI	+	o/o	-	-
Obtížnost	+	o/o	+	o
Útoky	+(ssh)	++(ssh)	+(ssh)	+(SSL)
Ztráta dat	+	++/++	+	+

## 27.3 Úvod do Unison

Unison je vynikající řešení pro synchronizaci a přenos adresářového stromu. Synchronizace je prováděna v obou směrech a lze ji kontrolovat pomocí přehledného grafického rozhraní. V případě potřeby je k dispozici ovládání přes příkazovou řádku. Synchronizaci lze automatizovat tak, že není potřebný žádný zásah uživatele. Takové nastavení již vyžaduje určité zkušenosti.

### 27.3.1 Požadavky

Unison je nutné nainstalovat na server i na klienty. *Serverem* se zde rozumí vzdálený počítač (na rozdíl od CVS, viz 27.1.2 – „CVS“ (strana 412)).

V následujících příkladech je Unison používán spolu s ssh. ssh klient musí být nainstalován na klientovi a ssh server na serveru.

## 27.3.2 Používání Unison

Podstatou práce Unison je asociace dvou adresářů (*kořeny, roots*). Tato asociace je symbolická — nejde o online spojení. V našem příkladu je asociace následující:

---

Klient:	/home/tux/dir1
Server:	/home/geeko/dir2

---

Synchronizovat se budou dva výše uvedené adresáře. Uživatel má na klientovi uživatelské jméno tux a na serveru geeko. Před zahájením práce je vhodné otestovat komunikaci klient—server příkazem:

```
unison -testserver /home/tux/dir1 ssh://geeko@server//homes/geeko/dir2
```

Problémy, které mohou nastat:

- Nekompatibilita verzí Unison na klientu a serveru.
- Server nepovoluje SSH připojení.
- Některá z uvedených cest neexistuje.

Pokud vše funguje, vynechejte volbu `-testserver`. Během první synchronizace Unison nezná vztahy mezi adresáři a navrhne směr přenosu jednotlivých souborů a adresářů. Šipka ve sloupci *Action* indikuje směr přenosu. Otazník znamená, že Unison nedokáže určit směr přenosu, protože obě verze byly změněny nebo jsou nové.

Kurzorovými klávesami (šipkami) můžete nastavit směr přenosu jednotlivých položek. Pokud jsou nastaveny správné směry pro všechny položky, potvrďte nastavení kliknutím na *Go*.

Vlastnosti Unison (například, zda má v jasných případech provést synchronizaci automaticky) lze nastavit při spuštění programu v příkazové řádce parametry. Seznam parametrů získáte příkazem: `unison --help`.

### **Rovnice 27.1** Soubor `~/unison/example.prefs`

```
root=/home/tux/dir1
root=ssh://wilber@server//homes/wilber/dir2
batch=true
```

Pro každou dvojici se vytváří záznam (log) v uživatelském adresáři `~/unison`. Konfigurace se také ukládá v tomto adresáři (např. `~/unison/example.prefs`). Při startu synchronizace zadejte na příkazovém řádku soubor s konfigurací jako parametr: `unison example.prefs`.

## **27.3.3 Další informace**

Velmi užitečná je oficiální dokumentace Unison. Kompletní manuál najdete na stránce <http://www.cis.upenn.edu/~bcpierce/unison/> a v SUSE balíčku `unison`.

## **27.4 Úvod do programu CVS**

CVS je velmi užitečný v případě časté editace textových souborů velkým počtem uživatelů. CVS lze použít i pro netextová data, ale za cenu velkých požadavků na prostor na serveru, protože budou ukládány všechny verze souborů celé. Navíc v takových případech není dostupná řada užitečných funkcí. Synchronizace pomocí CVS vyžaduje na rozdíl od Unison existenci jednoho centrálního serveru, ke kterému se mohou připojit všichni klienti.

### **27.4.1 Konfigurace CVS serveru**

*Server* je místo, kde jsou uloženy všechny platné soubory včetně nejnovějších verzí. Jako server lze používat libovolnou pracovní stanici. Pokud je to možné, měli byste provádět pravidelné zálohování tohoto serveru.

Při konfiguraci serveru je vhodné nastavit přístup pro uživatele přes SSH. Pokud je uživatel serveru znám např. jako `tux` a CVS je nainstalován jak na klientovi, tak na serveru, je nutné na straně klienta nastavit následující proměnné prostředí:

```
CVS_RSH=ssh CVS_ROOT=tux@server:/serverdir
```

Příkazem `cvs init` lze inicializovat CVS server ze strany klienta. Tento příkaz je třeba provést pouze jednou.

Nakonec musí být synchronizaci přiřazeno jméno. Na klientovi vytvořte adresář, který bude obsahovat soubory spravované pomocí CVS. Jméno adresáře bude také jméno synchronizace. V našem případě používáme adresář pojmenovaný *synchome*. Jméno synchronizace nastavíme v tomto adresáři příkazem:

```
cvs import synchome tux wilber
```

Řada CVS příkazů vyžaduje komentář. Pro tento účel CVS spouští editor (definovaný proměnnou prostředí *\$EDITOR* nebo *vi*, pokud jste žádný editor nenastavili). V editoru můžete doplnit komentář jako v následujícím příkladě:

```
cvs import -m 'toto je test' synchome tux wilber
```

## 27.4.2 Používání CVS

Od tohoto okamžiku lze k repositáři přistupovat ze všech klientů a stahovat jeho obsah pomocí příkazu `cvs co synchome`. Voláním tohoto příkazu se vytvoří na klientském počítači podadresář *synchome*. Změny provedené v tomto adresáři (tento adresář nebo některý z jeho podadresářů musí být aktuálním adresářem) odešlete do repositáře příkazem `cvs commit`.

Implicitně jsou na server zasílány všechny soubory včetně podadresářů. Chcete-li zaslat pouze jednotlivé soubory nebo adresáře, určete je příkazem `cvs commit soubor1 adresar1`. Nové soubory a adresáře musí být do repositáře vloženy příkazem `cvs add soubor1 adresar1` dříve, než jsou zaslány na server příkazem `cvs commit soubor1 adresar1`.

Pokud přejdete k jiné pracovní stanici, proveďte checkout synchronizačního repositáře, pokud jste tak neučinili na této stanici již dříve (viz výše).

Synchronizaci se serverem zahájíte příkazem `cvs update`. Jednotlivé soubory a adresáře synchronizujete příkazem `cvs update soubor1 adresar1`. Rozdíly mezi aktuálními lokálními soubory a soubory na serveru získáte příkazem `cvs diff` nebo `cvs diff soubor1 adresar1`. Příkaz `cvs -nq update` použijte, pokud chcete zjistit, jaké soubory budou synchronizací ovlivněny.

Během synchronizace jsou používány následující stavové symboly:

U

Lokální verze byla aktualizována verzí ze serveru. To se týká všech souborů, které jsou na serveru, ale na lokálním systému chyběly.

M

Lokální verze souboru obsahuje oproti serveru změny. Pokud byly změny i na serveru, bylo je možné sloučit s lokálními změnami. Nedošlo ke konfliktu.

P

Byla aktualizována lokální verze. Nepřenesl se celý soubor, ale byl použit tzv. patch (záplata).

C

Lokální verze je v konfliktu s verzí na serveru.

?

Soubor v CVS repositáři neexistuje.

Stav označený písmenem M upozorňuje na lokálně změněný soubor. Buď nahrajte lokální soubor na server nebo lokální soubor odstraňte a proveďte znovu update – chybějící soubor bude nahrán ze serveru. Pokud budete nahrávat lokálně změněný soubor, který byl mezitím změněn ve stejné řádce i na serveru, může dojít ke konfliktu označenému písmenem C.

V takovém případě v souboru vyhledejte konfliktní značky a rozhodněte se mezi verzemi. Je to poměrně nepříjemná práce, takže někdy může být lepší rezignovat na své změny, lokální soubor smazat a pomocí příkazu `cvcs up` nahrát aktuální verzi ze serveru.

## 27.4.3 Další informace

Zde jsme vám poskytli pouze krátký úvod do možností CVS. Rozsáhlou dokumentaci naleznete na následujících adresách:

<http://www.cvshome.org/>

<http://www.gnu.org/manual/>

## 27.5 Úvod do Subversion

Subversion je svobodný opensource systém pro správu verzí, který je často považován za nástupce staršího systému CVS. To znamená, že funkce známé z CVS jsou běžně dostupné i v subversion, avšak bez nutnosti potýkat se s omezeními a nevýhodami CVS. O některých vlastnostech jsme psali již v kapitole [27.1.3 – „subversion“](#) (strana 412).

### 27.5.1 Instalace Subversion serveru

Instalace skladovací databáze na serveru je poměrně snadná. Subversion k tomuto účelu nabízí speciální administrační nástroj. Chcete-li vytvořit nový repositář (skladovací databázi), použijte příkaz:

```
svnadmin create /cesta/k/repositari
```

Další možnosti lze zjistit pomocí příkazu `svnadmin help`. Na rozdíl od CVS není subversion založená na RCS, nýbrž na Berkeley databázi. Proto se ujistěte, že repositář neinstalujete na vzdálené souborové systémy (např. NFS, AFS, Windows SMB). Databáze totiž vyžaduje POSIX kompatibilní zamykací mechanismy, které nejsou na těchto souborových systémech podporovány.

Příkaz `svnlook info` poskytuje informace o stávajícím repositáři.

```
svnlook info /cesta/k/repositari
```

Server musí být nastaven tak, aby umožnil uživatelům přístup k repositáři. Použijte k tomu buď Apache webserver s WebDAV nebo `svnserve`, což je server dodávaný spolu se subversion. Jakmile je `svnserve` spuštěn, je repositář přístupný na příslušné URL přes protokol `svn://` nebo `svn+ssh://`. Uživatelé, kteří se musejí při použití `svn` autentizovat, lze nastavit v souboru `/etc/svnserve.conf`.

Výběr mezi servery Apache a `svnserve` záleží na mnoha faktorech. Doporučujeme proto nastudovat si příručku k subversion. Více se o ní dozvíte v části [27.5.3 – „Další informace“](#) (strana 424).

## 27.5.2 Použití a provoz

K přístupu do repozitáře použijte příkaz `svn` (podobně jako příkaz `cv`s). Obsah poskytovaný správně nastaveným serverem s odpovídajícím repozitářem je přístupný jakýmkoliv klientem jedním z následujících příkazů:

```
svn list http://svn.example.com/cesta/k/projektu
```

nebo

```
svn list svn://svn.example.com/cesta/k/projektu
```

Uložit existující projekt do aktuálního adresáře (`check out`) lze příkazem `svn checkout`:

```
svn checkout http://svn.example.com/cesta/k/projektu jmenoprojektu
```

`Checkout` vytvoří na klientovi nový podadresář `jmenoprojektu`. V něm lze následně provádět operace se soubory (přidávání, kopírování, přejmenovávání, mazání):

```
svn add soubor
svn copy starysoubor novysoubor
svn move starysoubor novysoubor
svn delete soubor
```

Tyto příkazy lze rovněž použít na adresáře. Program `subversion` navíc umí zaznamenat vlastnosti souboru či adresáře:

```
svn propset license GPL foo.txt
```

Předchozí příklad nastaví hodnotu `GPL` vlastnosti `license`. Vlastnosti lze zobrazit příkazem `svn proplist`:

```
svn proplist --verbose foo.txt
Properties on 'foo.txt':
license : GPL
```

Změny lze na server uložit příkazem `svn commit`. Ostatní uživatelé se mohou synchronizovat příkazem `svn update`.

Na rozdíl od `CVS` lze stav pracovního adresáře zobrazit bez přístupu k repozitáři pomocí `svn status`. Lokální změny jsou zobrazeny v pěti sloupcích, z nichž nejdůležitější je první:

```
"
```

```
    Žádné změny.
```

'A'	Objekt bude přidán.
'D'	Objekt bude smazán.
'M'	Objekt byl změněn.
'C'	Objekt je v konfliktu.
'I'	Objekt byl ignorován.
'?'	Objekt není verzovacím systémem spravován.
'!'	Objekt chybí. Tento příznak značí, že byl objekt smazán či přesunut bez použití příslušného příkazu <code>svn</code> .
'~'	Objekt je spravován jako soubor, ale byl nahrazen adresářem, nebo naopak.

Druhý sloupec zobrazuje stav vlastností. Význam všech sloupců je popsán v příručce k `subversion`.

Příkaz `svn help` použijte, pokud chcete získat popis parametrů příkazu:

```
svn help proplist
proplist (plist, pl): List all properties on files, dirs, or revisions.
usage: 1. proplist [PATH...]
       2. proplist --revprop -r REV [URL]
```

1. Lists versioned props in working copy.
2. Lists unversioned remote props on repos revision.

## 27.5.3 Další informace

Prvním místem, kde hledat další informace, je domovská stránka projektu `subversion` na adrese <http://subversion.tigris.org/>. Velmi doporučujeme také příruč-



ku, která je dostupná online na adrese <http://svnbook.red-bean.com/svnbook/index.html> nebo po instalaci balíčku `subversion-doc` v souboru `file:///usr/share/doc/packages/subversion/html/book.html`.

## 27.6 Úvod do rsync

Program `rsync` je užitečný, pokud je potřeba pravidelně přenášet velké množství dat, která se příliš nemění. To je často případ záloh nebo staging serverů. Tyto servery obsahují kompletní adresářové stromy webserverů, které jsou pravidelně zrcadleny na webserver v demilitarizované zóně.

### 27.6.1 Konfigurace a provoz

Program `rsync` lze provozovat ve dvou různých režimech. Může být používán k archivování nebo kopírování dat. K tomu je na cílovém systému potřeba pouze vzdálený interpret příkazů, např. `ssh`. Program `rsync` lze ale používat také jako démon, který poskytuje adresáře na síti.

Základní provozní režim `rsync` nevyžaduje žádné zvláštní nastavení. Program `rsync` umožňuje přímo zrcadlit celé adresáře na jiný systém. Následující příkaz například vytvoří zálohu domovského adresáře uživatele `tux` na záložním serveru `sun`:

```
rsync -baz -e ssh /home/tux/ tux@sun:backup
```

A tímto příkazem se adresář nahraje zpět:

```
rsync -az -e ssh tux@sun:backup /home/tux/
```

Použití se příliš neliší od běžného kopírovacího nástroje, jako např. `scp`.

Program `rsync` by ale měl být používán v režimu `rsyncd`, který umožňuje plně využívat všechny jeho funkce. Lze tak učinit spuštěním démona `rsyncd` na jednom ze systémů. Démon se konfiguruje v souboru `/etc/rsyncd.conf`. Pokud například chcete aby byl adresář `/srv/ftp` dostupný přes `rsync`, použijte následující konfiguraci:

```
gid = nobody
uid = nobody
read only = true
use chroot = no
transfer logging = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log
```

```
[FTP]
path = /srv/ftp
comment = An Example
```

Po provedení konfigurace spusťte rsyncd příkazem `rcrsyncd start`. Program rsyncd může být spouštěn i automaticky během startu systému. To nastavíte v editoru úrovní běhu pomocí nástroje YaST nebo ručně příkazem `insserv rsyncd`. Program rsyncd může být také spuštěn pomocí `xinetd`, je to však doporučeno jen na serverech, které rsyncd používají jen výjimečně.

Konfigurace v použitém příkladu rovněž vytváří protokolový soubor `/var/log/rsyncd.log`, ve kterém jsou zaznamenávána všechna spojení.

Přenos z klientského systému lze otestovat příkazem:

```
rsync -avz sun::FTP
```

Tento příkaz vypíše všechny soubory v adresáři `/srv/ftp` na serveru. Požadavek je zaznamenán v souboru `/var/log/rsyncd.log`. Pro zahájení skutečného přenosu specifikujte cílový adresář. Aktuální adresář zapište jako `..`. Například:

```
rsync -avz sun::FTP .
```

Implicitně se při synchronizaci pomocí `rsync` nemažou žádné soubory. Pokud si chcete smazání souborů vynutit, musíte použít parametr `--delete`. Pokud si chcete být jistí, že nebudou smazány žádné novější soubory, použijte parametr `--update`. Veškeré konflikty je nutné řešit manuálně.

## 27.6.2 Další informace

Důležité informace o `rsync` naleznete v manuálových stránkách (`man rsync` a `man rsyncd.conf`). Technický popis funkce `rsync` naleznete v souboru `/usr/share/doc/packages/rsync/tech_report.ps`. Novinky o `rsync` najdete na webové stránce projektu na adrese <http://rsync.samba.org/>.

## 27.7 Úvod do mailsync

Program `mailsync` se používá zejména pro tři úlohy:

- Synchronizace lokálně uložených poštovních zpráv se zprávami uloženými na serveru.
- Přenos schránek na jiný server nebo převod do jiného formátu.
- Kontrola integrity schránky a vyhledávání duplikátů.

## 27.7.1 Konfigurace a použití

mailsync rozlišuje mezi samotnými schránkami (store) a kanály mezi schránkami (channel). Definice schránek a kanálů jsou uloženy v `~/mailsync`. Následující odstavce vysvětlují použití schránek (store) na několika příkladech.

Jednoduchá definice může vypadat takto:

```
store saved-messages {
    pat      Mail/saved-messages
    prefix  Mail/
}
```

`Mail/` je podadresář v domovském adresáři uživatele, který obsahuje zprávy včetně složky `saved-messages`. Pokud program mailsync spustíte příkazem `mailsync -m saved-messages`, vypíše seznam zpráv ve složce `saved-messages`.

Při nastavení:

```
store localdir {
    pat      Mail/*
    prefix  Mail/
}
```

vypíše příkaz `mailsync -m localdir` všechny zprávy ve složce `Mail/`. Příkaz `mailsync localdir` naopak vypíše jména složek.

Příklad specifikace pro IMAP server:

```
store imapinbox {
    server {mail.edu.harvard.com/user=gulliver}
    ref    {mail.edu.harvard.com}
    pat    INBOX
}
```

Uvedený příklad specifikuje pouze hlavní složku na IMAP serveru. Pro podsložky bude vypadat takto:

```
store imapdir {
server {mail.edu.harvard.com/user=gulliver}
ref {mail.edu.harvard.com}
pat INBOX.*
prefix INBOX.
}
```

Pokud IMAP server podporuje šifrované připojení, měla by jeho specifikace vypadat takto:

```
server {mail.edu.harvard.com/ssl/user=gulliver}
```

nebo, pokud je certifikát neznámý:

```
server {mail.edu.harvard.com/ssl/novalidate-cert/user=gulliver}
```

Nyní je možné složky v `Mail/` připojit k podadresářům na IMAP serveru:

```
channel folder localdir imapdir {
msinfo .mailsync.info
}
```

Program `mailsync` používá soubor `msinfo` k zaznamenávání již synchronizovaných zpráv.

Příkaz `mailsync folder` provede následující:

- Expanduje schéma schránky na obě strany.
- Ze získaných jmen složek odstraní předponu.
- V párech synchronizuje složky (pokud neexistují, vytvoří je).

Složka `INBOX.sent-mail` na IMAP serveru je synchronizována s lokální složkou `Mail/sent-mail` (pokud existují definice uvedené výše). Synchronizace mezi jednotlivými složkami se provádí následovně:

- Pokud zpráva existuje na obou stranách, nic se neděje.
- Pokud zpráva existuje jen na jedné straně a je nová (není uvedena v souboru `msinfo`), je přenesena.
- Pokud zpráva existuje jen na jedné straně a je stará (je již uvedena v souboru `msinfo`), je smazána (neboť byla očividně na jedné straně úmyslně smazána).

Pokud chcete s předstihem vědět, které zprávy budou během synchronizace přeneseny a které smazány, spusťte mailsync pomocí `mailsync folder localdir`. Tímto příkazem získáte seznam všech zpráv, které jsou na lokálním počítači nové, a seznam všech zpráv, které budou na IMAP serveru během synchronizace smazány. Podobně příkazem `mailsync folder imapdir` získáte seznam všech zpráv, které jsou nové na straně IMAP serveru, a zpráv, které budou během synchronizace smazány na lokálním počítači.

## 27.7.2 Možné problémy

V případě ztráty dat je nejbezpečnější metodou smazat příslušný soubor se záznamy `msinfo`. Tak budou všechny soubory existující na jedné straně považovány za nové a přeneseny během další synchronizace.

Synchronizace zahrnuje pouze zprávy s ID. Zprávy, které ID nemají, jsou ignorovány, tzn. nejsou ani přenášeny ani mazány. Chybějící ID je většinou důsledkem chyby programu při vytváření nebo odesílání zprávy.

Na některých IMAP serverech je hlavní složka adresována pomocí `INBOX` a podsložky pomocí náhodně zvoleného jména (na rozdíl od `INBOX` a `INBOX.jmeno`). Proto pro takové IMAP servery nelze nastavit vzorec jen pro podsložky.

Po úspěšném přenosu zpráv na IMAP server nastaví ovladače schránky (c-client) používané programem mailsync zvláštní příznak. Z tohoto důvodu nejsou některé programy, jako např. mutt, schopny rozpoznat tyto zprávy jako nové. Nastavení tohoto příkazu lze zakázat volbou `-n`.

## 27.7.3 Další informace

Další informace najdete po instalaci balíčku `mailsync` v souboru *README* v adresáři `/usr/share/doc/packages/mailsync/`. V této souvislosti věnujte také pozornost RFC 2076 *Common Internet Message Headers*.



# Samba

Pomocí balíku Samba lze doplnit libovolný unixový počítač o funkce výkonného souborového a tiskového serveru pro DOS, OS/2 a Windows počítače. Postupem doby se Samba vyvinula ve složitý a komplexní produkt. V této kapitole najdete popis základního nastavení Samba a konfigurace pomocí modulu programu YaST.

Podrobné informace jsou dostupné v digitální podobě. Příkazem `apropos samba` zobrazíte dostupné manuálové stránky. Pokud je Samba nainstalována, najdete další dokumentaci a příklady v adresáři `/usr/share/doc/packages/samba`. V adresáři `examples` najdete okomentovaný příklad konfigurace (`smb.conf.SuSE`).

Balíček `samba` verze 3 obsahuje řadu novinek a zlepšení, z nichž nejvýznamnější jsou:

- Podpora Active Directory.
- Výrazně vylepšená podpora Unicode.
- Přepracovaný interní autentizační mechanismus.
- Vylepšená podpora tiskového systému pro Windows 200x/XP.
- Možnost nastavení jako serveru domény Active-Directory.
- Možnost migrace z NT4 domény na Samba doménu.

---

### Tip: Migrace na Sambu verze 3

Pokud chcete migrovat ze Samby 2.x na Sambu 3, musíte být maximálně opatrní. Aby nedošlo k chybě, věnujte prosím pozornost dokumentu *Samba-HOWTO-Collection*. Najdete ho po instalaci balíčku `samba-doc` v souboru `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf`.

---

Samba používá SMB protokol (server message block) založený na službách NetBIOSu. Díky tlaku společnosti IBM Microsoft tento protokol uveřejnil, a tak je možné připojit se do domén sítě Microsoft. Protože Samba pracuje na základě TCP/IP protokolu, musí být tento protokol nainstalován na všech klientech.

NetBIOS je softwarové rozhraní (API) pro komunikaci mezi počítači poskytující tzv. *name service* umožňujícím počítačům připojeným k síti rezervovat si pro sebe jména, sloužící k oboustranné identifikaci. Pro přidělování nebo kontrolu jmen zde není žádná centrální autorita. Každý počítač v síti smí mít libovolný počet jmen, pokud se tato jména již nepoužívají jiným počítačem. Rozhraní NetBIOS lze implementovat v různých síťových architekturách. Jedna z implementací, která je těsně svázána se síťovým hardwarem, se nazývá NetBEUI (bývá však často zaměňována za NetBIOS). Síťové protokoly implementované v NetBIOSu pocházejí z IPX od společnosti Novell (NetBIOS via TCP/IP) a TCP/IP.

Všechny běžné operační systémy, jako Mac OS X, Windows nebo OS/2, podporují protokol SMB. NA všech počítačích musí být nainstalovaný TCP/IP protokol. Samba poskytuje klienta pro různé UNIXové systémy. Pro Linux existuje jaderný modul umožňující integraci SMB zdrojů na systémové úrovni.

SMB servery poskytují hardwarové místo klientům ve formě sdílení (shares). Sdílení zahrnuje adresář na serveru včetně podadresářů. Je exportováno pod zadaným jménem. Jako jméno sdílení lze nastavit jakékoliv jméno, nemusí to být jméno sdíleného adresáře. Tiskárna má také přiděleno jméno. Klienti pak k tiskárně přes její jméno přistupují.

## 28.1 Nastavení serveru

Nejdříve je třeba nainstalovat balíček `samba` Ručně pak můžete spustit službu příkazem `rcnmb start && rcsmb start a pomocí rcsmb stop && rcnmb stop`, ji opět ukončit.



Hlavní konfigurační soubor Samby je `/etc/samba/smb.conf`. Skládá se ze dvou logických částí. V části `[global]` jsou obecná a centrální nastavení. V části `[share]` se nastavují individuální sdílení souborů a tiskáren. Rozdělení mezi tyto dvě sekce zvyšuje přehlednost konfiguračního souboru.

## 28.1.1 Sekce `[global]`

Aby ostatní počítače s Windows mohly přistupovat prostřednictvím SMB k vašemu Samba serveru, vyžadují následující parametry ze sekce `[global]` určité úpravy v závislosti na nastavení sítě.

`workgroup = TUX-NET`

Samba serveru je pomocí této řádky přiřazena pracovní skupina. `TUX-NET` nahraďte správným jménem skupiny ve vašem síťovém prostředí. Samba server se objeví pod svým DNS jménem, pokud ovšem není používáno jiným strojem v síti. Pokud DNS jméno není dostupné, nastavte jméno serveru pomocí `netbiosname=MYNAME` (viz `mansmb.conf`).

`os level = 2`

Podle tohoto parametru se bude Samba server rozhodovat, zda se stane *LMB (Local Master Browser)* pro svou pracovní skupinu. Nízká hodnota zajistí, že existující windowsová síť nebude rušena špatně nakonfigurovanou Sambou. Bližší informace k této volbě naleznete v souborech `BROWSING.txt` a `BROWSING-Config.txt`, které najdete v podadresáři `textdocs` dokumentace balíku.

Pokud ještě neprovozujete SMB server (např. ve Windows NT, 2000, XP) a sambový server by měl v lokální síti udržovat informace o jménech dostupných systémů, zvýšte `os level` na vyšší hodnotu (např. 65). Váš Samba server se tak stane LMB.

Při změnách této hodnoty byste měli být obzvláště opatrní, protože můžete rušit komunikaci ve stávající síti. Nejprve si nastavení otestujte v izolované síti nebo o víkend.

`wins support` a `wins server`

Pokud chcete integrovat Sambu do windowsové sítě, kde již běží *WINS* server, tak položku `wins server` odkomentujte a uveďte IP adresu *WINS* serveru.

Pokud jsou windowsové systémy provozovány v oddělených podsítích a měly by se přesto vidět, potřebujete *WINS* server. Sambu proměníte na takový *WINS* server

nastavením volby `wins support = yes`. Pozor na to, abyste tuto položku aktivovali pouze na jednom serveru. Volby `wins server` a `wins support` nesmí být v souboru `smb.conf` nikdy povoleny současně.

## 28.1.2 Sdílení

V následujících příkladech si ukážeme, jak sdílet CD mechaniku a domovské adresáře uživatelů.

[`cdrom`]

Aby nedošlo ke zneužití CD mechaniky, je ve výchozím nastavení deaktivována pomocí komentáře (zde středník). Odstraněním středníků v prvním sloupci můžete CD-ROM sdílet.

### **Rovnice 28.1** *Sdílení CD-ROM*

```
;cdrom
;      comment = Linux CD-ROM
;      path = /media/cdrom
;      locking = No
```

[`cdrom`] a `comment`

Položka [`cdrom`] je jméno, které bude vidět na SMB klientech. Pomocí `comment` můžete sdílení podrobněji popsat.

```
path = /media/cdrom
```

Exportuje adresář `/media/cdrom`.

Vzhledem k velmi přísné implicitní konfiguraci je tento způsob exportování omezen na lokální uživatele. Ostatním umožníte přístup volbou `guest ok = yes`. Protože tato volba umožňuje přístup ke čtení všem, je potřeba s ní zacházet velice opatrně. Hlavně při jejím používání v sekci [`global`].

[`homes`]

Zvláštní postavení má export domovských adresářů. Pokud má uživatel na linuxovém souborovém serveru platný účet a vlastní domovský adresář, pak se může jeho klient po zadání platného uživatelského jména a hesla připojit

## Rovnice 28.2 *Sdílení domovských adresářů*

```
[homes]
comment = Home Directories
valid users = %S
browseable = No
read only = No
create mask = 0640
directory mask = 0750
```

[homes]

Při připojení uživatele k SMB serveru je automaticky vytvořeno sdílení pomocí direktivy [homes]. Výsledné jméno sdílení je shodné s uživatelským jménem a vytvoří se pouze, pokud již neexistuje sdílení se stejným jménem.

```
valid users = %S
```

%S je po úspěšném spojení nahrazen konkrétním jménem sdílení. V případě sdílení [homes] je to vždy jméno uživatele. Důsledkem je omezení používání home pouze na jeho vlastníka.

```
browseable = No
```

Toto nastavení činí sdílení neviditelným v síťovém prostředí.

```
read only = No
```

Samba má přenastaven zápis u exportovaných dat na `read only = Yes`. Pokud má být adresář přístupný pro zápis, pak je třeba nastavit `read only = No`, což je totéž jako `writeable = Yes`.

```
create mask = 0640
```

Systemy nezaložené na MS Windows NT nedokáží pracovat s UNIXovými přístupovými právy a tím pádem ani nastavit tato práva při vytváření souborů. Parametr `create mask` nastavuje přístupová práva všech nově vytvořených souborů. Toto nastavení se týká pouze těch sdílení, do kterých mají uživatelé právo zápisu. Výše uvedená hodnota nastavuje právo pro čtení a zápis vlastníka souboru a práva pro čtení pro všechny uživatele z vlastníkovy skupiny. Nastavením `valid users = %S` zamezíte ostatním členům skupiny přístupu ke čtení i v případě, že to práva povolují. Aby měla celá skupina práva ke čtení či zápisu, je nutné řádku `valid users = %S` zakomentovat.

## 28.1.3 Bezpečnostní úrovně

SMB protokol vychází z prostředí DOS/Windows a bere ohledy na problematiku bezpečnosti. Proto je možné přístup ke každému exportovanému adresáři ochránit heslem. SMB rozlišuje tři různé způsoby:

Share Level Security (security = share):

Heslo je stejné pro všechny uživatele, je vázáno na sdílení. Každý, kdo toto heslo zná, má ke sdílení přístup.

User Level Security (security = user):

Každý uživatel má vlastní heslo. Po registraci server přiděluje uživateli přístup jen k jemu povoleným sdílením.

Server Level Security (security = server):

Samba před klienty předstírá práci v uživatelském režimu. Nicméně předává všechna hesla k ověření jinému serveru v uživatelském režimu. Toto nastavení vyžaduje další parametr (`password server=`).

Uvedená nastavení jsou aplikována na celý server. Není možné nastavit individuální sdílení s různými bezpečnostními stupni. Můžete však pro každou IP adresu nastavenou na systému spustit vlastní Samba server.

Více informací o této problematice najdete v Samba HOWTO Collection. U vícenásobného serveru na jednom počítači věnujte pozornost volbám `interfaces` a `bind interfaces only`.

---

### Tip

Pro jednoduchou správu Samba serverů existuje program `swat`. Ten používá pro konfiguraci Samba serveru jednoduché webové rozhraní. Po spuštění prohlížeče ho najdete na adrese <http://localhost:901>, kde se přihlaste jako uživatel `root`. Nezapomeňte, že `swat` je také potřeba aktivovat v souborech `/etc/xinetd.d/samba` a `/etc/services`. K tomu musíte v souboru `/etc/xinetd.d/samba` nastavit parametr `disable` na hodnotu `no` (`disable = no`). Další informace o `swat` najdete v jeho manuálové stránce.

---

## 28.2 Samba jako přihlašovací server

V sítích, kde je převaha windowsových klientů, je často žádoucí, aby se směl uživatel přihlásit pouze s platným účtem a heslem. Toto je možné zajistit pomocí Samba serveru. V čistě windowsové síti je to úloha NT serveru, který je konfigurován jako Primary Domain Controller (PDC). Proto je třeba provést změny v obecné *globals* části konfiguračního souboru `smb.conf` uvedené v příkladu 28.3 – „Globální sekce `smb.conf`“ (strana 437).

### Rovnice 28.3 Globální sekce `smb.conf`

```
[global]
workgroup = TUX-NET
domain logons = Yes
domain master = Yes
```

Pokud se pro verifikaci používají šifrovaná hesla, musí si s tím Samba umět poradit. To umožňuje položka `encrypt passwords = yes` v části `[globals]` (v Sambě 3 je to výchozí nastavení). Kromě toho je třeba převést uživatelské účty a hesla do šifrovaného formátu vhodného pro Windows. To provedete příkazem `smbpasswd -a name`. Protože v doménové koncepci Windows NT potřebují i samotné počítače doménový účet, vytvořte ho následujícími příkazy:

### Rovnice 28.4 Nastavení účtu počítače

```
useradd hostname\$_
smbpasswd -a -m hostname
```

Příkazem `useradd` je přidán znak dolaru. Příkaz `smbpasswd` ho vkládá automaticky, pokud je použit parametr `-m`. Komentovanou ukázkovou konfiguraci včetně automatizace výše uvedených činností najdete v souboru `/usr/share/doc/packages/Samba/examples/smb.conf.SuSE`.

### Rovnice 28.5 Automatizované nastavení účtu počítače

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \
-s /bin/false %m\$_
```

Aby mohla Samba tento skript vykonat, zvolte Samba uživatele s požadovanými administrátorskými právy. Vyberte jednoho uživatele a přidejte ho do skupiny `ntadmin`. Pak můžete všechny uživatele patřící do této linuxové skupiny obdařit statutem `Domain Admins` pomocí příkazu:

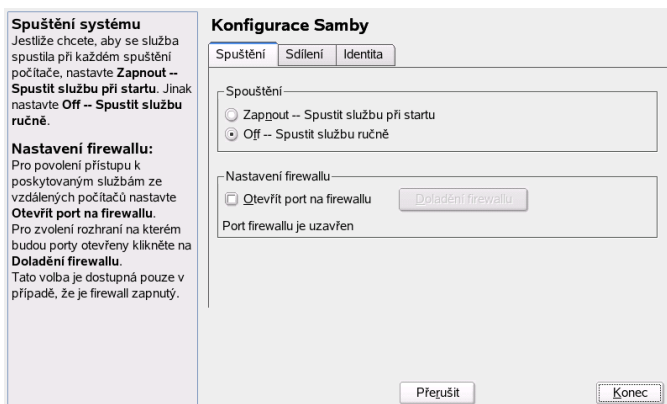
```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

Více informací naleznete ve dvanácté kapitole Samba-HOWTO-Collection v souboru `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf`.

## 28.3 Konfigurace Samba serveru pomocí programu YaST

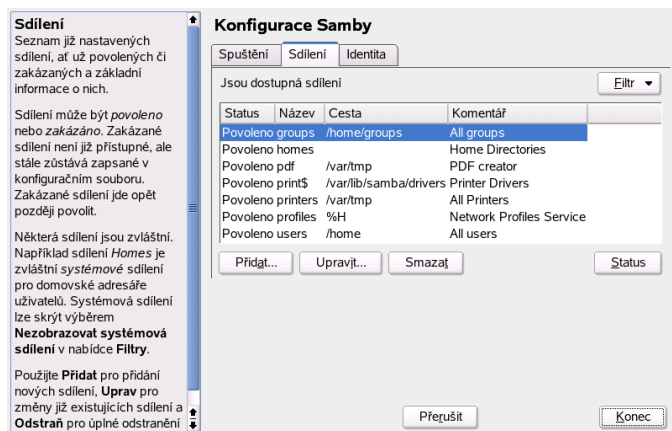
Na začátku nastavení Samba serveru zvolte doménu nebo pracovní skupinu, kterou bude server spravovat. V položce *Pracovní skupina nebo jméno domény* můžete zadat existující nebo zcela novou doménu či skupinu. V dalším kroku nastavte, zda má server plnit úlohu PDC (Primary Domain Controller) nebo BDC (Backup Domain Controller).

**Obrázek 28.1** Konfigurace Samby — start



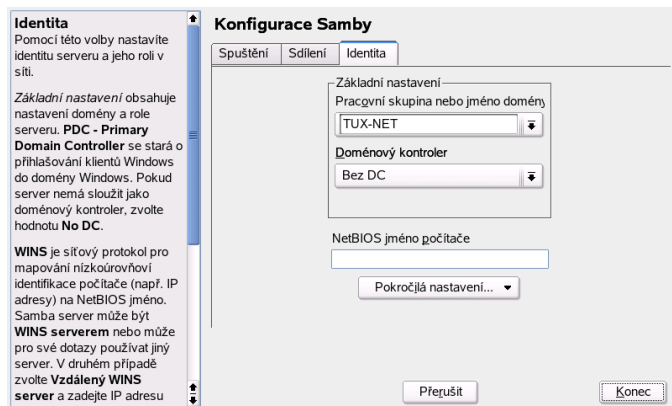
Na kartě *Spuštění* spusťte Sambu (viz 28.1 – „Konfigurace Samby — start“ (strana 438)) a v části *Nastavení firewallu* aktivujte *Otevřít port na firewallu*. Na všech rozhraních tak dojde k otevření portů pro služby `netbios-ns`, `netbios-dgm`, `netbios-ssn` a `microsoft-ds`. Pokud potřebujete upřesnit nastavení, klikněte na tlačítko *Doladění firewallu*.

## Obrázek 28.2 Konfigurace Samby — sdílení



Na kartě *Sdílení* (viz 28.2 – „Konfigurace Samby — sdílení“ (strana 439)) nastavte sdílení Samby. U jednotlivých položek lze tlačítkem *Změnit stav* přepínat mezi stavem *Zakázáno* a *Povoleno*. Nové sdílení zadáte kliknutím na *Přidat*.

## Obrázek 28.3 Konfigurace Samby — identita



Na kartě *Identita* (viz 28.3 – „Konfigurace Samby — identita“ (strana 439)) lze nastavit doménu počítače (*Základní nastavení*) a jméno v SMB síti (*NetBIOS jméno počítače*).

## 28.4 Nastavení klienta

Upozorňujeme, že server Samba je dosažitelný pro klienta pouze prostřednictvím protokolu TCP/IP. NetBEUI ani IPX nejsou pro Sambu v současnosti použitelné.

### 28.4.1 Nastavení Samba klienta pomocí YaST

Samba klienta nastavíte pro přístup ke zdrojům Samba serveru (soubory nebo tiskárny) následovně. V dialogu *Příslušnost k doméně Windows* zadejte doménu nebo pracovní skupinu. Všechny dostupné domény a skupiny zjistíte kliknutím na tlačítko *Procházet*. Skupinu vyberete označením myši. Pokud zvolíte *Použít SMB informace také pro autentizaci v Linuxu*, budou uživatelé ověřováni přes Samba server. Nastavení aktivujete kliknutím na tlačítko *Konec*.

### 28.4.2 Windows 9x a ME

Windows 9x a ME již sice podporu TCP/IP obsahují, avšak dosud nikoli jako výchozí nastavení. Proto pro přidání protokolu TCP/IP klikněte na *Ovládací panel*, dále *Systém* a vyberte *Přidat, Protokoly*, z nich vyberte *Microsoft* → *TCP/IP*. Po restartu počítače s Windows najdete Samba server dvojitým poklepáním na ikonu *Sít'* na pracovní ploše Windows.

Abyste mohli použít tiskárnu na Samba serveru, stačí nainstalovat standardní ovladač tiskárny (popřípadě ovladač Apple-PostScript) pro odpovídající verzi Windows. Nejlepší je provázat ho s tiskovou frontou, která přijímá úlohy ve formátu PostScript.

## 28.5 Optimalizace

Optimalizaci nabízí `socket options`. Přednastavení, která jsou součástí příkladové konfigurace se zaměřují především na lokální ethernetovou síť. Další podrobnosti naleznete v příslušné části manuálových stránek `smb.conf` a v manuálové stránce `socket(7)`. Další informace naleznete v Samba HOWTO Collection v kapitole věnované ladění výkonu.



Standardní konfigurace v `/etc/samba/smb.conf` není samozřejmě vhodná pro všechny sítě a způsoby nasazení, proto je třeba ji ještě upravit podle místních podmínek. Protože je ale tato optimalizace závislá na mnoha faktorech, neexistuje žádné univerzální řešení. Komentovaný příklad konfiguračního souboru `examples/smb.conf` .SuSE obsahuje užitečné informace pro přizpůsobení místním podmínkám.

Samba HOWTO Collection obsahuje návod pro řešení nejčastějších problémů. V části V (Part V) pak najdete podrobný návod, který vás krok za krokem provede kontrolou konfigurace.



## Proxy server Squid

Squid je na linuxových/unixových platformách nejrozšířenější proxy cache. Zde si popíšeme, jak ho konfigurovat, řekneme si, jaké má systémové požadavky a mnoho dalšího. Stranou nezůstane ani konfigurace transparentní proxy, zpracování statistik programy calamaris a cachemgr a filtrování internetových stránek pomocí squidGuard.

Squid funguje jako burzián. Přijímá požadavky od klientů (v tomto případě internetových prohlížečů) a ty pak předává dál odpovídajícím serverům poskytovatele. Když se požadovaný objekt vrátí, nechá si pro sebe jednu kopii, kterou uloží v diskové cache a druhou doručí zpět klientovi. Výhoda se projeví v okamžiku, kdy bude druhý uživatel požadovat stejný objekt — v tom případě není třeba stránku stahovat znovu, ale nahraje z cache. Výsledkem je nepoměrně rychlejší vyřízení požadavku a navíc dochází k úspoře kapacity linky.

Squid nabízí velké spektrum funkcí, např. hierarchické dělení proxy serveru, které rozkládá zátěž systému, vytváření pravidel pro přístup klientů, správu přístupových práv k jednotlivým stránkám a také statistiky nejčastěji používaných internetových stránek, chování uživatelů při surfování apod. Squid není generickou proxy. Standardně pouze zprostředkovává HTTP spojení. Kromě toho podporuje protokoly FTP, Gopher, SSL a WAIS, ale žádné další internetové protokoly typu Real Audio, News nebo video-konference. UDP protokol používá pouze pro podporu komunikace mezi různými cache. Z tohoto důvodu nejsou podporovány ani žádné další programy postavené na tomto protokolu.

# 29.1 Informace o proxy-cache

Proxy cache Squid lze využít různými způsoby. Spolu s firewallem může zlepšit bezpečnost. Lze použít více proxy společně. Umí také určit, jaké objekty se vyplatí cachovat a na jak dlouho.

## 29.1.1 Squid a bezpečnost

Squid můžete provozovat spolu s firewallem a zabezpečit vnitřní síť před vnější sítí pomocí proxy. Firewall odmítne všechny přístupy ke službám z vnějšku kromě přístupu ke Squid. Všechna webová spojení musí být zprostředkována proxy.

Pokud konfigurace firewallu obsahuje DMZ, měla by proxy pracovat v této zóně. V takovém případě je důležité, aby všechny počítače v DMZ zasílaly logy počítačům ve vnitřní síti. Možnost implementace tzv. *transparentní* proxy je popsána v části 29.5 – „[Konfigurace transparentní proxy](#)“ (strana 454).

## 29.1.2 Vícenásobná cache

Můžete nakonfigurovat více cache, které si vyměňují objekty. Snižuje se tak zátěž systému a zvyšuje pravděpodobnost nalezení objektu již v lokální síti. Můžete také vytvořit hierarchicky uspořádané cache, takže je cache schopná předat požadavek na objekt jiné cache na stejné úrovni nebo cache nadřazené – která pak vyřídí požadavek prostřednictvím jiné cache nebo stáhne objekt přímo ze zdroje.

Volba správné topologie je velice důležitá, protože by nemělo dojít ke zvýšení celkového síťového provozu. U velké sítě je možné nakonfigurovat proxy server pro každou podsíť a tu pak spojit s nadřazenou cache, která je opět napojena na proxy ISP (poskytovatele).

Tato komunikace je řízena prostřednictvím ICP (*Internet Cache Protocol*), který je vystavěn nad UDP. Výměna dat mezi jednotlivými cache se provádí prostřednictvím HTTP (*Hyper Text Transmission Protocol*) založeném na TCP.

Aby byl nalezen nejlepší server pro požadované objekty, posílá cache všem proxy stejné hierarchie tzv. ICP dotaz. Ostatní proxy pak odpoví prostřednictvím ICP buď *HIT* v případě, že objekt našly, nebo *MISS* v případě, že ho nenašly. V případě nálezu

více HITů se proxy rozhodne, ze které cache bude stahovat. Toto rozhodování se provádí na základě rychlosti odpovědi. Když všechny cache ohlásí MISS, pak bude dotaz předán nadřazené cache.

Abyste zabránili vícenásobnému ukládání objektů v různých cache lokální sítě – používají se jiné ICP protokoly, jako je např. *CARP Cache Array Routing Protocol* nebo *HTCP Hyper-Text Cache Protocol*. Čím více objektů je v síti udržováno, tím větší je pravděpodobnost nálezů požadovaného.

### 29.1.3 Přechovávání objektů z Internetu

Ne všechny objekty v síti jsou statické. Existuje velké množství dynamicky generovaných CGI stránek, počítadel a SSL dokumentů, které nejsou ukládány v cache, protože jsou měněny při každém přístupu.

A u všech ostatních objektů je třeba zvážit, jak dlouho by měly zůstat v cache. Kvůli tomu mají objekty v cache přiřazeny různé stavy. V hlavičkách pak obsahují informace jako *Last modified* (datum poslední změny) nebo *Expires* (datum expirace), případně informaci o zákazu cachování objektu. Objekty v cache jsou odstraňovány převážně kvůli nedostatku místa, kde se používají algoritmy jako je LRU (*Least Recently Used*). Ten v podstatě maže nejdéle nepoužité objekty.

## 29.2 Systémové požadavky

Nejdříve by měla být určena zátěž systému. Je třeba věnovat zvláštní pozornost špičkám, které mohou být i 4x vyšší, než je denní průměr. Pokud si nejste jisti, pak je lepší nadhodnotit systémové požadavky, protože nevhodný hardware pro Squid může vést k výraznému poklesu výkonu. V následujícím textu jsou jednotlivé části seřazeny podle důležitosti.

### 29.2.1 Pevný disk

Při ukládání do meziskladu (cache) hraje rychlost zápisu velkou roli. Proto byste měli tomuto faktoru věnovat velkou pozornost. U pevných disků je nejdůležitější doba přístupu (náhodného), která je udávána v milisekundách. Protože bloky dat se kterými Squid pracuje jsou poměrně malé, je přístupová doba disku důležitější než jeho datová

průchodnost. Pro účely proxy jsou lepší disky s vysokými otáčkami, neboť umožňují rychlejší pozicování hlavičky. Rychlost systému lze zvýšit využitím více disků současně, případně použitím RAID.

## 29.2.2 Velikost diskové cache

Pokud máte malou cache, pak je pravděpodobnost HITu velmi nízká, protože cache se velice rychle zaplní, a pak jsou starší objekty přepisovány novějšími. Pokud ale máte *1 GB* pro cache a uživatel potřebuje každý den pouze *10 MB*, pak máte minimálně sto dní, než se vám cache zaplní.

Nejjednodušší je určit velikost cache podle rychlosti připojení. Pokud máte 1 Mbit/s linku, pak bude maximální přenosová rychlost 128 KB/s. Za předpokladu, že veškerý datový přenos skončí v cache, máte za jednu hodinu uloženo více než 450 MB. Pokud bychom pokračovali a řekli bychom, že pracovní den má 8 hodin a pořád by byla linka plně využita, pak je to za jeden den 3,6 GB. Protože však nebývá linka vytížena na 100%, budou stačit zhruba 2 GB.

## 29.2.3 RAM

Velikost potřebné paměti pro Squid je závislá na počtu objektů, které se nachází v cache. Squid ukládá cachovací odkazy a často používané stránky v paměti tak, aby mohly být požadavky rychleji vyřizovány. Protože RAM je mnohem rychlejší než pevný disk.

Squid má v paměti také další data, např. tabulku se všemi použitými IP adresami, s nejčastěji používanými zásobníky, objekty a pak také seznamy s informacemi o přístupu a mnoho dalšího.

Proto je důležité, aby měl Squid dostatek operační paměti. Pokud by musel začít swapovat, tj. odkládat méně často používané části operační paměti do vyhrazeného diskového oddílu, dramaticky by klesl výkon. Pro správu cache v paměti můžete využít `cachemgr.cgi`, který je popsán v části [29.6 – „cachemgr.cgi“](#) (strana 457).

## 29.2.4 CPU

Proxy nepotřebuje příliš výkonný procesor. Pouze během kontroly obsahu cache se zvyšuje zatížení procesoru. Pokud byste chtěli použít víceprocesorové stroje, pak nedosáhnete zvýšení výkonu Squidu. Lepší je přidat disky a operační paměť.

## 29.3 Spuštění squidů

Program Squid má SUSE Linux již předkonfigurovaný, takže ho můžete spustit hned po instalaci. Předpokladem bezproblémového startu je správně nastavená síť – tj. aby byl nastaven alespoň nameserver a bylo možné pingnout. Problémy se mohou objevit v okamžiku, kdy používáte dynamickou DNS konfiguraci. V tom případě by alespoň nameserver měl mít platný zápis, protože pokud Squid nenajde v `/etc/resolv.conf` DNS server – tak se vůbec nespustí.

### 29.3.1 Příkazy pro spuštění squidů

Pro spuštění se přihlaste jako uživatel `root` a zadejte příkaz `rcsquid start` Při prvním spuštění se vytvoří adresářová struktura v `/var/squid/cache`, což provádí automaticky spouštěcí skript `/etc/init.d/squid` a může to trvat řádově několik vteřin až minut. Pokud se zobrazí zelené `done`, byla proxy úspěšně spuštěna. Na lokálním systému můžete funkčnost squidů ihned v prohlížeči nastavením proxy na `localhost` a portu na `3128`.

Abyste zpřístupnili Squid všem uživatelům, bude potřeba upravit konfigurační soubor `/etc/squid/squid.conf` tak, že změníte položku `http_access deny all` na `http_access allow all`. Mějte ale na mysli, že tím otevřete proxy všem, proto byste měli nastavit ACL. Bližší informace naleznete v části [29.4.2 – „Volby pro kontrolu přístupu“](#) (strana 452).

Změny v konfiguračním souboru `/etc/squid/squid.conf`, je potřeba načíst příkazem `rcsquid reload`. Nebo můžete Squid úplně restartovat příkazem `rcsquid restart`.

Příkaz `rcsquid status` slouží ke zjištění, zda proxy běží. Zastavit ji můžete příkazem `rcsquid stop`. Může to chvíli trvat, protože Squid čeká až půl minuty (volba

`shutdown_lifetime` v souboru `/etc/squid/squid.conf`), než přeruší spojení s klienty a zapíše data na disk.

---

## Varování

Pokud ukončíte squid tak, že ho zabijete příkazem `kill` nebo `killall`, může dojít k poškození cache, kterou je pak potřeba smazat, aby bylo možné squid znovu spustit.[indexterm](#)>

---

Pokud Squid zemře krátce po úspěšném spuštění, zkontrolujte, zda není špatně nastaven `nameserver` či zda nechybí soubor `/etc/resolv.conf`. Squid zaznamenává důvod selhání spuštění do protokolového souboru `/var/squid/logs/cache.log`. Pokud se má Squid spouštět při startu systému automaticky, použijte editor úrovní běhu YaST a aktivujte Squid v požadovaných úrovních, viz „Editor úrovní běhu“ (2 – „*Konfigurace pomocí YaST*“, ↑Uživatelská příručka).

Při odinstalování proxy se neodstraní ani cache, ani protokolové soubory. Je potřeba ručně smazat adresář `/var/cache/squid`.

## 29.3.2 Lokální DNS server

Lokální DNS server je výhodný i v případě, že nespravuje vlastní doménu. Stačí, když funguje pouze jako caching-only DNS a umí bez zvláštní konfigurace zpracovat DNS dotazy, resp. je předat root nameserveru (viz [20.2 – „Spuštění nameserveru BIND“](#) (strana 329)). Jak toho dosáhnete, záleží na tom, zda jste zvolili dynamické DNS při konfiguraci připojení k Internetu.

### Dynamické DNS

Za běžných okolností, při použití dynamického DNS, je DNS server nastaven poskytovatelem během navazování spojení. Lokální soubor `/etc/resolv.conf` je upraven automaticky. Toto chování je způsobeno nastavením `sysconfig` proměnné `MODIFY_RESOLV_CONF_DYNAMICALLY` na `YES`. Nastavte ji YaST `sysconfig` editorem (viz [8.8 – „YaST sysconfig Editor“](#) (strana 160)) na `NO`. Pak zadejte lokální DNS server do souboru `/etc/resolv.conf`: IP adresu `127.0.0.1` pro `localhost`. Tak Squid při startu vždy nalezne lokální DNS server.

Aby byl přístupný nameserver poskytovatele, zadejte ho v konfiguračním souboru `/etc/named.conf` spolu s jeho IP adresou do položky `forwarders`. Při po-



užití dynamického DNS to lze automatizovat nastavením proměnné `MODIFY_NAMED_CONF_DYNAMICALLY` na `YES`.

### Statické DNS

Při použití statického DNS nedochází během navazování spojení k žádným úpravám DNS, takže není třeba upravovat žádné `sysconfig` proměnné. Musíte ovšem do souboru `/etc/resolv.conf` zadat lokální DNS server, jak je výše popsáno. Navíc musíte ručně zadat statický DNS server poskytovatele (s IP adresou) do souboru `/etc/named.conf` (položka `forwarders`).

---

#### Tip: DNS a firewall

Pokud máte spuštěný firewall, ujistěte se, že skrze něj mohou DNS požadavky projít.

---

## 29.4 Konfigurační soubor `/etc/squid/squid.conf`

Všechna nastavení Squid proxy serveru jsou zapsána v souboru `/etc/squid/squid.conf`. Pro první spuštění Squida není třeba v tomto souboru provádět žádné změny, ale externím klientům bude zamítnut přístup. Proxy bude dostupná pouze pro `localhost`. Výchozí port je 3128. Předinstalovaný soubor `/etc/squid/squid.conf` obsahuje podrobné komentáře s popisy voleb a mnoho příkladů. Téměř všechny položky začínají znakem `#` (komentář) a obsahují podrobné informace. Zadané hodnoty jsou téměř vždy shodné s výchozími, takže odstranění komentáře bez změny hodnoty má pětštinou minimální vliv. Lepší je ale příklady nechat beze změny a zadat volby se změněnými parametry na nový řádek pod příklad. Tak budete mít přehled o výchozích hodnotách a vámi provedených změnách.

---

#### Tip: Přizpůsobení konfiguračního souboru po aktualizaci

Pokud jste aktualizovali Squid ze starší verze, doporučuje se upravit nový `/etc/squid/squid.conf` a jen do něj zadat změny provedené ve starším souboru. Pokud byste použili starší konfigurační soubor přímo, riskujete, že nebude správně fungovat, protože některé volby se mezi verzemi mění.

---

## 29.4.1 Základní nastavení

`http_port 3128`

Toto je port, na kterém poslouchá Squid požadavky klientů. Přednastaven je na `3128`, ale běžný je také port `8080`. Další porty můžete přidat (odděluje je mezerou).

`cache_peer hostname type proxy-port icp-port`

Zde uveďte nadřazenou proxy, např. když chcete využívat proxy poskytovatele. Jako *hostname* uveďte jméno a IP adresu používané proxy. Jako *type* zadejte *parent*. Jako číslo portu poskytovatele (*proxy-port*) se nejčastěji používá `8080`. *icp-port* můžete nastavit na `7` nebo `0`, pokud neznáte ICP port nadřazené proxy a její používání není dohodnuto s poskytovatelem. Navíc byste za čísla portů měli zapsat volby `default` a `no-query`, čímž zamezíte používání ICP protokolu. Squid se pak vůči proxy poskytovatele chová jako obyčejný webový prohlížeč.

`cache_mem 8 MB`

Tato položka stanoví, kolik operační paměti bude Squid pro cache používat. Přednastaveno je `8 MB`.

`cache_dir ufs /var/cache/squid 100 16 256`

Položka *cache\_dir* určuje adresář, do kterého budou na disku ukládány jednotlivé objekty. Čísla za cestou k adresáři znamenají: maximální velikost cache v MB; počet podadresářů; a počet podadresářů podadresářů. Parametr `ufs` by měl zůstat beze změny. Přednastavenými hodnotami pro velikost cache jsou 100 MB diskového prostoru v adresáři `/var/cache/squid`, kde bude vytvořeno 16 adresářů, každý z nich se 256 podadresářů. Při vyčleňování místa na disku byste si měli nechat dostatek rezerv, rozumné je vytvářet cache o velikosti 50 až 80 procent volného místa. Kromě toho byste měli poslední dvě čísla (počty adresářů) zvětšovat velice opatrně, protože režie adresářových struktur může snížit výkon systému. Pokud máte pro cache více disků, můžete vytvořit odpovídající množství řádků s definicí *cache\_dir*.

`cache_access_log /var/log/squid/access.log`

Cesta k protokolovému souboru.

`cache_log /var/log/squid/cache.log`

Cesta k protokolovému souboru.

`cache_store_log /var/log/squid/store.log`

Cesta k protokolovému souboru.

Tyto tři volby definují cesty k protokolovým souborům a není třeba je měnit. Pouze v případě, že je cache velice často dotazována, se může hodit přesunout protokolové soubory na jiný disk.

`emulate_httpd_log off`

Změnou na *on* získáte čitelné protokolové soubory, se kterými si ale neporadí některé programy, které mají na starosti vyhodnocování.

`client_netmask 255.255.255.255`

Touto položkou můžete maskovat IP adresy zapisované do logů a skrýt tak identitu klientů. Pokud zde napíšete např. `255 . 255 . 255 . 0`, tak bude poslední pozice IP adresy vynulována.

`ftp_user Squid@`

Zde nastavíte heslo, které bude Squid používat pro anonymní FTP login. Může mít smysl uvést zde platnou e-mailovou adresu, protože některé FTP servery její platnost kontrolují.

`cache_mgr webmaster`

Tato volba slouží pro uvedení e-mailové adresy, na kterou se pošle zpráva v případě neočekávaného pádu. Přednastaveno je *webmaster*.

`logfile_rotate 0`

Squid umí také rotovat uložené protokolové soubory, pokud ho spustíte s volbou `squid -k rotate`. Soubory jsou číslovány, jakmile se dojde k nastavené hodnotě, přepíše se nejstarší soubor. Výchozí nastavení je 0, protože pro archivaci a mazání protokolových souborů používá SUSE Linux cron úlohu nastavenou v `/etc/logrotate/squid`.

`append_domain domain`

Volbou *append\_domain* můžete určit, která doména bude automaticky připojena v případě, že není žádná uvedena. Nejčastěji se zde uvádí vlastní doména, takže stačí v prohlížeči uvést *www* a dostanete se na vlastní webserver.

`forwarded_for on`

Když nastavíte na *off*, odstraní Squid IP adresu a jméno počítače klienta z HTTP dotazu.

`negative_ttl 5 minutes; negative_dns_ttl 5 minutes`

Ve standardním případě není třeba toto nastavení upravovat. Pokud ale máte vytáčenou linku, pak se může stát, že Internet nebude po nějakou dobu přístupný. To

je tím, že si Squid poznamenává neúspěšné dotazy a brání se znovu dotazovat, i když je již spojení s Internetem obnoveno. V tom případě změňte *minutes* na *seconds* a nechte znovu načíst stránku v prohlížeči.

`never_direct allow acl_name`

Pokud chcete zabránit tomu, aby Squid vyřizoval požadavky přímo z Internetu, pak použijte tuto volbu. V tom případě je ale potřeba, aby existovala ještě další proxy, které bude Squid své požadavky zasílat. Tu je třeba nastavit ve volbě *cache\_peer*. Pokud zadáte jako *acl\_name* `all`, pak zajistíte, že všechny požadavky budou předávány *nadřazené* proxy. To je třeba např. tehdy, když poskytovatel striktně trvá na využívání jeho proxy, nebo když je firewall nastaven tak, že nepovoluje přímý přístup k Internetu.

## 29.4.2 Volby pro kontrolu přístupu

Squid obsahuje velice sofistikovaný systém pro řízení přístupu k proxy. Pomocí ACL je velice dobře a jednoduše konfigurovatelný. V zásadě se jedná o seznam pravidel, která jsou jedno po druhém zpracovávána. ACL je třeba definovat předtím, než budou použita. Některá jsou již definována, jako je *all* a *localhost*. Ale pouhým vytvořením ACL ještě nic neprovedete. Teprve, když ho použijete např. spolu s *http\_access*, tak se změny projeví.

`acl acl_name type data`

ACL potřebuje pro svou definici minimálně tři parametry. Název *acl\_name* může být libovolný. U *type* můžete zvolit z celé řady různých možností, které jsou uvedeny v části *ACCESS CONTROLS* souboru `/etc/squid/squid.conf`. Jaká *data* uvést, záleží na typu ACL. Lze je také načíst ze souboru, například, přes jméno počítače, IP adresu nebo URL. Následují krátké příklady:

```
acl mujnet srcdomain .ma-domena.cz
acl ucitele src 192.168.1.0/255.255.255.0
acl studenti src 192.168.7.0-192.168.9.0/255.255.255.0
acl obed time MTWHF 12:00-15:00
```

`http_access allow acl_name`

Volbou *http\_access* určíte, kdo může proxy používat a k čemu může na Internetu přistupovat. Zde využijete výše definovaná ACL nebo použijete ta přednastavená, tj. *localhost* a *all*. Přístup může být povolen nebo zakázán pomocí hodnot *deny* či *allow*. Můžete vytvořit celý seznam položek *http\_access*, které budou zpracovávány odshora dolů a podle toho, co se načte jako první bude přístup povolen nebo zakázán.

Jako poslední položka by měl být vždy *http\_access deny all*. V následujícím příkladu povolíme přístup všem lokálním uživatelům, zatímco všem ostatním ho zakážeme.

```
http_access allow localhost
http_access deny all
```

V dalším příkladu (s využitím vlastních ACL) mají učitelé povolen stálý přístup k Internetu, zatímco studenti k němu mají přístup pouze od pondělí do pátku v čase oběda.

```
http_access deny localhost
http_access allow ucitele
http_access allow studenti obed time
http_access deny all
```

Volby *http\_access* byste, kvůli přehlednosti, měli psát pouze na jedno, předem určené, místo v souboru */etc/squid/squid.conf*. A to mezi řádky:

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR
# CLIENTS
```

a uzavírající text:

```
http_access deny all
```

`redirect_program /usr/bin/squidGuard`

Tato volba slouží pro tzv. přesměrování, kdy jsou dotazy předávány externímu programu, v našem případě squidGuard, který dokáže zakázat přístup k určeným URL. Spolu s proxy autentizací a vhodnými ACL tak můžete velice precizně řídit přístup k Internetu pro různé skupiny. squidGuard je v separátním balíku a musí se tedy nainstalovat zvlášť.

`authenticate_program /usr/sbin/pam_auth`

Pokud je třeba autentizovat uživatele při přístupu k proxy, můžete použít program `pam_auth`. Při prvním přihlášení uživatele se spustí přihlašovací dialog, kde musí uživatel vložit uživatelské jméno a heslo. Navíc se stále vyžaduje ACL, připojit se mohou pouze klienti s platným loginem:

```
acl password proxy_auth REQUIRED
```

```
http_access allow password
http_access deny all
```

Klíčové slovo *REQUIRED* za *proxy\_auth* můžete nahradit seznamem povolených jmen uživatelů nebo cestou k takovému seznamu.

`ident_lookup_access allow acl_name`

Tato volba zajistí, že za všechny klienty definované v ACL je proveden identifikační dotaz, který prověří identitu uživatele. Když nastavíte `acl_name` na `all`, bude se provádět dotazování pro všechny klienty. Na klientech však musí běžet identifikační démon. V Linuxu můžete nainstalovat program `pidentd`, pro Windows existuje volně dostupný software, který si můžete stáhnout z Internetu. Aby byli připuštěni pouze klienti s úspěšným identifikačním dotazem `ident_lookup`, je potřeba opět definovat vhodný ACL.

```
acl idenhosts ident REQUIRED
```

```
http_access allow idenhosts  
http_access deny all
```

Také zde je možné nahradit `REQUIRED` seznamem povolených jmen uživatelů. Používání `ident` může přístup výrazně zpomalit, protože kontrola se provádí při každém dotazu.

## 29.5 Konfigurace transparentní proxy

Standardně posílá prohlížeč na určitý port proxy serveru dotazy a proxy mu odpovídající objekty poskytuje, ať už se v cache nacházejí nebo ne. V praxi pak mohou nastat různé situace:

- Z bezpečnostních důvodů je lepší, když proxy používají všichni klienti.
- Je třeba, aby uživatelé používali proxy, i když o ní neví.
- Proxy se v síti přesunula, ale klienti by si měli i nadále zachovat svou starou konfiguraci.

V každém z těchto případů je vhodné nasadit transparentní proxy. Princip je přitom velice jednoduchý. Internetový prohlížeč pošle svůj požadavek. Na cestě sedí proxy, která tento požadavek zpracuje a odpověď odešle zpět prohlížeči, který vůbec netuší, že komunikuje s proxy a ne přímo se zdrojem. Celý proces je zcela transparentní.

## 29.5.1 Konfigurace jádra

Nejprve se ujistěte, že jádro proxy serveru podporuje transparentní proxy. Jádro systému SUSE Linux tuto podmínku splňuje. Pokud tomu tak není, recompilejte jádro s podporou transparentní proxy.

## 29.5.2 Možnosti konfigurace v `/etc/squid/squid.conf`

Volby v souboru `/etc/squid/squid.conf` potřebné pro aktivaci transparentní proxy jsou následující:

- `httpd_accel_host virtual`
- `httpd_accel_port 80`  
Číslo portu, na kterém běží HTTP server.
- `httpd_accel_with_proxy on`
- `httpd_accel_uses_host_header on`

## 29.5.3 Konfigurace firewallu pomocí SuSEfirewall2

Všechny příchozí dotazy musí být pomocí firewallu přeměřovány na port Squida. K tomu můžete použít nástroj SuSEfirewall2. Jeho konfigurace se nachází v souboru `/etc/sysconfig/SuSEfirewall2`. Soubor je dobře komentovaný. I když chcete nastavit pouze transparentní proxy, je potřeba provést určitá nastavení ve firewallu:

- Rozhraní pro přístup k Internetu: `FW_DEV_EXT="eth1"`
- Rozhraní pro přístup k vnitřní síti: `FW_DEV_INT="eth0"`

Když jste definovali rozhraní pro přístup k jednotlivým sítím, je potřeba povolit porty a služby, které budou přístupné z vnější sítě. V našem příkladu jsou vně nabízeny jen webové služby:

```
FW_SERVICES_EXT_TCP="www"
```

Pak je třeba povolit porty a služby dostupné z vnitřní (bezpečné) sítě přes TCP i UDP:

```
FW_SERVICES_INT_TCP="domain www 3128"  
FW_SERVICES_INT_UDP="domain"
```

Tím jsou povoleny webové služby a Squid, který běží standardně na portu 3128. Navíc je povolena služba DNS (*domain*). Pokud DNS povolovat nechcete, smažte ho z nastavení výše a nastavte následující volbu na no:

```
FW_SERVICE_DNS="yes"
```

Nejdůležitější je volba číslo 15:

### **Rovnice 29.1** Konfigurace firewallu: Volba 15

```
#  
# 15.)  
# Which accesses to services should be redirected to a local port  
# on the firewall machine?  
#  
# This can be used to force all internal users to surf via your  
# Squid proxy, or transparently redirect incoming Web traffic to  
# a secure Web server.  
#  
# Choice: leave empty or use the following explained syntax of  
# redirecting rules, separated with spaces.  
# A redirecting rule consists of 1) source IP/net,  
# 2) destination IP/net, 3) original destination port and  
# 4) local port to redirect the traffic to, separated by a colon,  
# e.g. "10.0.0.0/8,0/0,80,3128 0/0,172.20.1.1,80,8080"  
#
```

Komentáře popisují syntaxi. Nejdřív se vezme IP adresa a síťová maska interní sítě, ze které se bude přistupovat k proxy firewallu. Pak zadejte adresu a masku cíle, tj. kam jsou požadavky klientů posílány. V případě webového prohlížeče zvolte síť 0/0, což značí přístup kamkoliv. Pak nastavte originální port a port, na který jsou požadavky přesměrovávány. Protože Squid podporuje kromě HTTP i další protokoly, přeměrujte na proxy požadavky i z dalších portů, jako např. FTP (port 21), HTTPS nebo SSL (port 443). V našem příkladě jsou webové služby (port 80) přeměrovány na Squid proxy (port 3128). Pokud je sítí nebo služeb více, musí být v položce odděleny mezerou.



```
FW_REDIRECT_TCP="192.168.0.0/16,0/0,80,3128 192.168.0.0/16,0/0,21,3128"  
FW_REDIRECT_UDP="192.168.0.0/16,0/0,80,3128 192.168.0.0/16,0/0,21,3128"
```

Firewall s novou konfigurací spustíte nastavením proměnné `START_FW` v souboru `/etc/sysconfig/SuSEfirewall2` na hodnotu `"yes"`.

Pak spustíte Squid tak, jak je uvedeno v části 29.3 – „Spuštění squid“ (strana 447). Zda vše funguje právně se můžete přesvědčit v protokolovém souboru `/var/log/squid/access.log`.

Zda jsou všechny porty nastaveny dobře zjistíte tak, že použijete z libovolného místa mimo vaši síť portscan, tj. že se pokusíte zjistit, které porty jsou otevřené. V našem případě by měl být otevřen pouze port 80. Ke skenování použijte např. program `nmap` (`nmap -O IP_adresa`).

## 29.6 cachemgr.cgi

Cache manager (`cachemgr.cgi`) je CGI program pro zpracování statistik spotřeby paměti běžící proxy Squid. Je to také pohodlný způsob správy cache.

### 29.6.1 Nastavení

Nejprve je třeba mít v systému běžící webový server. Zda server běží zjistíte jako uživatel `root` příkazem `rcache status`. Pokud se zobrazí hlášení:

```
Checking for service httpd: OK  
Server uptime: 1 day 5 hours 23 minutes 17 seconds
```

tak Apache běží. V opačném případě je třeba webový server spustit příkazem `rcache start`. Jako poslední krok je třeba zkopírovat `cachemgr.cgi` do adresáře `cgi-bin` Apache příkazem:

```
cp /usr/share/doc/packages/squid/scripts/cachemgr.cgi /srv/www/cgi-bin/
```

## 29.6.2 ACL cache manageru v /etc/squid/squid.conf

V originálním souboru jsou výchozí nastavení potřebná pro cache manager. První ACL je nejdůležitější, protože se cache manager snaží se Squidem komunikovat pomocí cache\_object protokolu.

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
```

Měla by být nastavena i následující pravidla:

```
http_access allow manager localhost
http_access deny manager
```

Následující pravidla předpokládají, že web server a Squid běží na stejném počítači. Pokud komunikace mezi cache managerem a Squidem vychází ze strany web serveru na jiném počítači, nastavte další ACL, jak je uvedeno v příkladu [29.2 – „Přístupová pravidla“](#) (strana 458).

### **Rovnice 29.2** Přístupová pravidla

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl webserver src 192.168.1.7/255.255.255.255 # webserver IP
```

Pak přidejte pravidla z příkladu [29.3 – „Přístupová pravidla“](#) (strana 458).

### **Rovnice 29.3** Přístupová pravidla

```
http_access allow manager localhost
http_access allow manager webserver
http_access deny manager
```

Nastavte heslo pro správce cache nutné pro přístup k rozšířeným volbám, jako vzdálenému zavření cache nebo zobrazení podrobných informací o cache. K tomu slouží položka `cachemgr_passwd` s heslem a seznam voleb, které budou zobrazeny po uvedení hesla. Tento seznam je uveden v komentáři v `/etc/squid/squid.conf`.

Pokaždé, když se změní konfigurace Squidu, je potřeba ho restartovat `rcsquid reload`.

## 29.6.3 Prohlížení statistik

Podívejte se na [http://vas\\_server/cgi-bin/cachemgr.cgi](http://vas_server/cgi-bin/cachemgr.cgi). Stiskněte *continue* a nechte si zobrazit různé statistiky. Bližší informace o jednotlivých volbách naleznete v často kladených dotazech k programu Squid (FAQ) na adrese <http://www.squid-cache.org/Doc/FAQ/FAQ-9.html>

## 29.7 squidGuard

Tato kapitola by měla být úvodem do konfigurace squidGuard a měla by vám představit možnosti jeho použití. Pro podrobné popisy jemných nuancí zde nebude dostatek místa. Hlubší informace naleznete na internetových stránkách <http://www.squidguard.org>.

squidGuard je svobodný, flexibilní a velice rychlý filtr pro Squida. Podporuje definování množství pravidel pro přístup s různými omezeními pro různé skupiny. Pro přesměrování používá squidGuard standardní rozhraní Squidu.

squidGuard můžete použít k následujícím úkolům:

- Omezení přístupu určitých uživatelů pouze k definovaným serverům nebo URL.
- Zakázání přístupu určitých uživatelů k určitým serverům nebo URL.
- Zamezení přístupu určitých uživatelů na základě regulárních výrazů nebo slov.
- Přesměrování ze zakázané URL na inteligentní CGI stránku.
- Přesměrování nepřihlášeného uživatele na registrační formulář.
- Náhrada reklamních banerů prázdným GIFem.
- Rozdílná pravidla přístupu v závislosti na čase, dni v týdnu a datu.
- Rozdílná pravidla pro jednotlivé skupiny uživatelů.

Ani squidGuard nebo Squid ale neumí:

- Filtrovat, cenzurovat nebo upravovat text v dokumentech.

- Filtrovat, cenzurovat nebo upravovat skriptovací jazyky (např. JavaScript nebo VBscript), které jsou součástí HTML.

Pro použití programu squidGuard musíte nejprve nainstalovat balíček squidGuard a pak upravit konfigurační soubor `/etc/squidguard.conf`. Pokud hledáte příkladové konfigurace, podívejte se na <http://www.squidguard.org/config/>. Později můžete zkusit složitější konfigurace.

Pak vytvořte stránku *Přístup odmítnut* nebo CGI stránku, na kterou bude klient přesměrován v případě, že přistoupí na zakázanou stránku. I zde doporučujeme používat Apache.

Nyní musíte squidovi říct, aby používal squidGuard. Stačí použít následující zápis v `/etc/squid/squid.conf`:

```
redirect_program /usr/bin/squidGuard
```

Další volba `redirect_children` nastavuje počet přesměrovacích procesů (squidGuardu), které na stroji poběží. Standardně dokáže squidGuard zpracovat 100000 požadavků za 10 vteřin na 500MHz Pentiu s 5900 doménami a 7880 URL. Proto se nedoporučuje nastavovat více než 4 procesy, protože pak zabírají pouze místo v paměti.

```
redirect_children 4
```

Nakonec necháte Squida znovu načíst konfiguraci příkazem `rcsquid reload`. Přišel čas otestovat nastavení v prohlížeči.

## 29.8 Vytvoření protokolů programem Calamaris

Calamaris je perlový skript, který vytváří hlášení o aktivitě cache. Tyto reporty jsou dostupné buď v ASCII nebo HTML. Calamaris využívá při sestavování protokolových souborů Squidu. Domovskou stránku projektu naleznete na <http://Calamaris.Cord.de/>.

Program se používá velice jednoduše. Přihlaste se jako uživatel `root` a použijte příkaz `cat access.log.soubory | calamaris volby > vystupnisoubor`. V případě, že zpracováváte více protokolových souborů, je důležité seřadit je chronologicky, nejstarší soubor první. Použitelné volby jsou následující:

-a  
Výstupem budou všechna dostupná hlášení.

-w  
Výstupem je protokol ve formátu HTML.

-l  
Nadpis nebo logo v záhlaví.

Další informace o různých volbách obsahuje manuálová stránka [calamaris](http://calamaris.org).

Typickým příkladem použití je:

```
cat access.log.2 access.log.1 access.log | calamaris -a -w \  
> /usr/local/httpd/htdocs/Squid/squidreport.html
```

Hlášení se tak uloží do adresáře webservru.

Dalším nástrojem, který můžete použít pro generování hlášení o stavu cache, je SARG (Squid Analysis Report Generator). Další informace naleznete na stránkách <http://web.onda.com.br/orso/>.

## 29.9 Další informace o Squidu

Navštivte domovskou stránku <http://www.squid-cache.org/>. Naleznete tam uživatelskou příručku a rozsáhlý seznam často kladených dotazů (FAQ).

Navíc máte k dispozici HOWTO, které naleznete po nainstalování balíčku `howtoen` v adresáři `/usr/share/doc/howto/en/mini/TransparentProxy.gz`. Využít můžete i konferenci [squid-users@squid-cache.org](mailto:squid-users@squid-cache.org) nebo její archiv na adrese <http://www.squid-cache.org/mail-archive/squid-users/>.



# **Část 5. Mobilita**





## Mobilita v Linuxu

Tato kapitola pojednává o používání Linuxu ve světě mobilních počítačů. Krátce si představíme různé oblasti a dostupná zařízení, najdete část o potřebných aplikacích i informace o možnostech minimalizace spotřeby. Na konci najdete odkazy na nejdůležitější zdroje informací.

Většina lidí si při slově mobilita představí notebooky, kapesní počítače a mobilní telefony. Tato kapitola se však zaměřuje také na další zařízení jako jsou externí disky, flash disky nebo digitální fotoaparáty, které můžete připojovat jak k notebookům, tak k pracovním stanicím.

### 30.1 Notebooky

#### 30.1.1 Zvláštní hardwarové vlastnosti notebooků

Z důvodů důrazu na mobilitu, minimální prostorové nároky a spotřebu energie se hardware notebooků od obyčejných stolních počítačů v mnoha ohledech odlišuje. Výrobci mobilních zařízení vyvinuli standard PCMCIA (*Personal Computer Memory Card International Association*), který pokrývá oblast paměťových karet, síťových rozhraní jako síťové karty a modemy a externích disků. Informace o implementaci tohoto standardu v Linuxu, potřebných nastaveních, dostupných aplikacích a řešení možných problémů najdete v kapitole [31 – „Linux a notebooky“](#) (strana 475).

## 30.1.2 Snížení spotřeby energie

Řada komponent je již od výrobce navržena a optimalizována tak, aby měla v případě napájení z baterií co nejnižší spotřebu energie. Podíl takto upravených komponent na úspoře energie je přinejmenším stejně tak důležitý jako schopnosti operačního systému. SUSE Linux řadu metod úspory spotřeby energie při napájení z baterie. Následující seznam možných způsobů snížení spotřeby je seřazen podle významu dopadu na spotřebu:

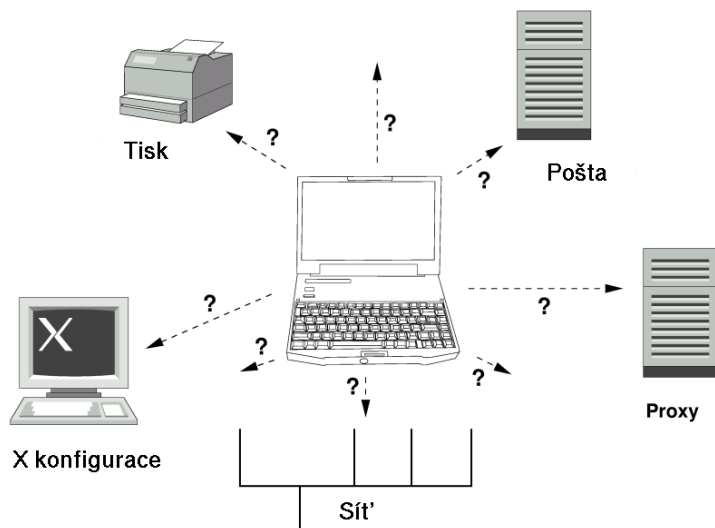
- Zpomalení rychlosti CPU
- Vypnutí monitoru během nečinnosti
- Ruční nastavení parametrů monitoru
- Odpojení nepoužívaných zařízení (USB CD-ROM, externí myš, nepoužívané PCMCIA karty, atd..)
- Zastavení disku při nečinnosti

Podrobnější informace o správě napájení v systému SUSE Linux a používání modulu správy napájení programu YaST najdete v kapitole [33 – „Správa napájení“](#) (strana 487).

## 30.1.3 Změny nastavení systému

V mobilním prostředí se systém často potřebuje přizpůsobovat novým podmínkám. Mnoho služeb závisí na pracovním prostředí a při změnách je nutné přenastavit jejich klienty. SUSE Linux dokáže obstarat i takové situace.

**Obrázek 30.1** Integrace notebooku do sítě



Služby měněné přenášením mezi domácí a podnikovou sítí mohou být následující:

#### Nastavení sítě

Nastavení sítě obsahuje IP adresu, jmenné služby, připojení k internetu a připojení k dalším sítím.

#### Tisk

V závislosti na síti, do které je notebook nastaven, musí být správně nastavená databáze tiskáren a příslušný tiskový server.

#### Email a proxy

Musí být nastaven správný seznam serverů.

#### Nastavení grafického prostředí

Pokud např. v zaměstnání připojujete notebook k externímu monitoru, musí být dostupné příslušné nastavení v grafickém prostředí.

SUSE Linux nabízí dvě možnosti, které lze kombinovat, jak notebook přizpůsobit aktuálnímu prostředí.

## SCPM

SCPM (*system configuration profile management*) umožňuje jednotlivá nastavení obsahující konfigurační soubory ukládat do tzv. *profilů*. Profily lze vytvářet pro různé situace. Jsou užitečné při potřebě změn prostředí (domácí síť, podniková síť). Mezi profily se lze jednoduše přepínat. Informace o SCPM najdete v kapitole 32 – „*Správa profilů*“ (strana 477). Přepínání mezi profily v KDE umožňuje applet Profile Chooser. Aplikace vyžaduje před přepnutím profilu zadání hesla uživatele root.

## SLP

SLP (*service location protocol*) zjednodušuje připojení notebooku do existující sítě. Bez SLP je obvykle potřeba znát pro nastavení řadu údajů. V případě SLP jsou všechny potřebné informace vysílány po síti a aplikace si vše nastaví samy automaticky. SLP lze používat také pro instalaci systému. Podrobnější informace o SLP najdete v části 19 – „*SLP služby v síti*“ (strana 319).

Význam SCPM spočívá v povolení a správě snadno reprodukovatelných systémových podmínek. SLP významně usnadňuje síťové nastavení.

## 30.1.4 Software

V oblasti mobilních zařízení je řada oblastí, které vyžadují zvláštní aplikace: monitorování systému (především stav baterií), synchronizace dat, bezdrátová komunikace v perifériemi nebo bezdrátové připojení k internetu. V této sekci najdete informace o nejdůležitějších aplikacích.

### Monitorování systému

V systému SUSE Linux najdete dva monitorovací nástroje prostředí KDE. Stav nabití baterií a status napájení zobrazuje applet KPowersave na hlavním panelu. Komplexní systém monitorování poskytuje KSysguard. Pokud používáte prostředí GNOME, budete používat GNOME ACPI (jako applet) a Monitor systému.

#### KPowersave

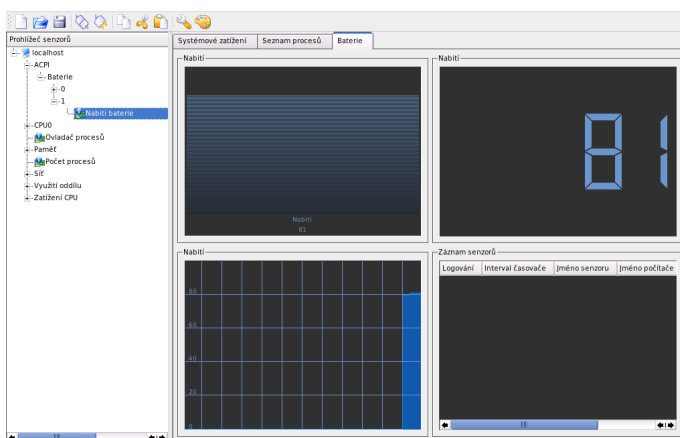
KPowersave je applet, který zobrazuje stav baterií a status napájení na hlavním panelu v prostředí KDE. V případě připojení do sítě je zobrazena malá zástrčka. Po přechodu na napájení z baterie se objeví ikonka baterie. Z kontextové nabídky aplikace lze po zadání hesla uživatele root otevřít modul správy napájení programu YaST. V tomto modulu můžete nastavit chování správy napájení. Informace o

modulu správy napájení programu YaST najdete v kapitole 33 – „*Správa napájení*“ (strana 487).

## KSysguard

KSysguard je nezávislá aplikace pro monitorování systému. Monitoruje ACPI (stav baterie), zatížení procesoru, síťový provoz, rozdělení disku a využití paměti. Může monitorovat a zobrazovat libovolné systémové procesy. Způsob zobrazení a filtrování lze upravit. Lze monitorovat různé parametry v několika stránkách nebo přes síť sbírat data z několika počítačů současně. KSysguard může běžet jako démon na počítači bez prostředí KDE. Více informací o tomto programu najdete v nápovědě.

**Obrázek 30.2** Monitorování stavu baterií pomocí KSysguard



## Synchronizace dat

Pokud střídavě pracujete na notebooku bez síťového připojení a na pracovní stanici v síti, je nezbytně nutné zajistit, abyste na obou počítačích měli všechna aktuální data. To zahrnuje poštovní složky, adresáře i jednotlivé soubory. Řešením je synchronizace dat, kterou můžete provádět následujícími způsoby:

### Synchronizace emailů

Používejte pro ukládání zpráv v podnikové síti IMAP účty. Ke zprávám lze přistupovat libovolným klientem, který umí pracovat také s odpojeným IMAP účtem jako např. Mozilla Thunderbird Mail, Evolution nebo KMail. Klienta je nutné nastavit tak, aby byla vždy použita shodná složka `Odeslané`. Tím zajistíte, že synchronizace proběhne bez problémů, a budete mít vždy aktuální data a zprávy budou

mít správný status. Abyste vždy měli přehled o neodeslaných zprávách, používejte místo systémových MTA jako postfix nebo sendmail SMTP služby implementované ve svém poštovním klientovi.

#### Synchronizace souborů a adresářů

Pro synchronizaci dat mezi pracovní stanicí a notebookem je k dispozici celá řada aplikací. Podrobnější informace najdete v kapitole [27 – „Synchronizace souborů“](#) (strana 411).

## Bezdrátová komunikace

Stejně jako doma nebo v kanceláři lze zapojit počítač do klasické sítě, lze notebooky propojit s ostatními notebooky, perifériemi, mobilními telefony nebo kapesními počítači pomocí bezdrátové technologie. Linux tří typy bezdrátové komunikace:

#### WLAN

WLAN je jako bezdrátová technologie s největším dosahem jediná vhodná volba pro budování rozsáhlých sítí. Lze ji použít k propojování nezávislých stanic nebo k připojení k internetu. Zařízení nazývané přístupový bod může hrát úlohu základní stanice sítě a zprostředkovávat přístup do internetu. Mobilní uživatel se může mezi přístupovými body přepínat a přistupovat do sítě přes bod, který mu umožňuje nejkvalitnější přístup. Stejně jako u mobilních telefonů je možný přístup kdykoliv. Podrobnější informace najdete v části [34.1 – „Bezdrátové sítě“](#) (strana 511).

#### Bluetooth

Bluetooth je bezdrátová technologie s kratším dosahem. Obvykle je používána pro komunikaci mezi počítači a kapesními počítači nebo také místo IrDA pro komunikaci s mobilními telefony. Touto technologií lze také propojovat více počítačů bez nutnosti dohledu na jednotlivá zařízení. Bluetooth je také používána u bezdrátových myši a klávesnic. Bližší informace o Bluetooth najdete v části [34.2 – „Bluetooth“](#) (strana 520).

#### IrDA

IrDA je bezdrátová technologie s nejkratším dosahem. Obě komunikační strany musí být v dohledu. Překážky jako zdi vedou k nefunkčnosti spojení. Jedním z využití IrDA je přenos souborů z mobilního telefonu do notebooku a naopak. Propojena pomocí IrDA je pouze část mezi notebookem a telefonem. Přenos na delší vzdálenosti je již veden mobilní sítí. Dalším obvyklým využitím IrDA je bezdrátové odesílání tiskových úloh na tiskárnu. Více informací o IrDA najdete v části [34.3 – „IrDA — Infrared Data Association“](#) (strana 530).

## 30.1.5 Ochrana dat

V ideálním případě by měla být data na notebooku chráněna několika způsoby. Možné oblasti zajištění jsou následující:

### Ochrana proti krádeži

Pokud je to možné, můžete počítač zajistit fyzicky. V obchodech je dnes k dispozici řada různých typů zabezpečení.

### Bezpečnost dat v systému

Důležitá data by neměla být šifrovaná jen během přenosu, ale také na disku. Tím zajistíte, že v případě krádeže nedojde k jejich zneužití. Popis vytváření šifrovaného souborového systému najdete v části 4.3 – „Šifrování diskových oddílů a souborů“ (strana 95).

### Síťová bezpečnost

Každý přenos dat by měl být bezpečný. Základní informace o Linuxu a sítích najdete v části 4.4 – „Bezpečnost a soukromí“ (strana 97). O bezpečnosti v bezdrátových sítích pojednává kapitola 34 – „Bezdrátová komunikace“ (strana 511).

## 30.2 Mobilní hardware

SUSE Linux podporuje automatickou detekci mobilních disků připojených přes firewire (IEEE 1394) nebo USB. Termín mobilní disky zde zahrnuje všechny typy firewire nebo USB disků, flash disků a digitálních kamer. Všechna tato zařízení jsou po připojení automaticky detekována systémem hotplug. subfs a submount zajišťují automatické připojení zařízení do souborového systému. Ruční připojování a odpojování zařízení již není používáno. Po ukončení programu, který přistupovat k zařízení, stačí disk jednoduše odpojit od počítače.

### Externí disky (USB a Firewire)

Po rozpoznání systémem jsou externí disky dostupné v seznamu připojených zařízení po kliknutí na ikonu *Můj počítač* (KDE) nebo *Počítač* (GNOME). Na externím disku můžete libovolně vytvářet, přejmenovávat a mazat adresáře i soubory. Disk lze přejmenovat kliknutím na ikonu disku pravým tlačítkem a volbou příslušné *Přejmenovat*. Nové jméno bude dostupné pouze ve správci souborů, skutečné jméno zařízení nastavené systémem jako např. `/media/usb-xxx` nebo `/media/ieee1394-xxx` zůstane nezměněno.

## USB flash disky

K flash diskům systém přistupuje jako k externím diskům. Přejmenovat je lze ve správci souborů.

## Digitální fotoaparáty (USB a Firewire)

Digitální fotoaparáty rozpoznané systémem jsou často ve správci souborů zobrazeny jako externí disky. KDE umožňuje přístup k obrázkům uloženým ve fotoaparátu zadáním URL `camera: /`. Obrázky lze upravovat například pomocí programu digikam nebo GIMP. V prostředí GNOME lze použít Nautilus. Jednoduchý nástroj pro správu a úpravu obrázků je GThumb. Pro pokročilé úpravy je určen GIMP. Programy digikam a GIMP a Nautilus jsou popsány v uživatelské příručce, kde je digitální fotografii věnována celá kapitola.

---

### **Důležité: Bezpečnost mobilních diskových zařízení**

Výměnné pevné disky a flash disky jsou stejně jako notebooky častým cílem zlodějů. Aby nedošlo k jejich zneužití, doporučujeme na nich vytvořit šifrovaný souborový systém viz. [4.3 – „Šifrování diskových oddílů a souborů“](#) (strana 95)

---

## 30.3 Mobilní telefony a kapesní počítače

Pracovní stanice a notebooky mohou komunikovat s mobilními telefony pomocí IrDA nebo Bluetooth. Některé modely podporují oba protokoly, jiné pouze jeden. Použití těchto protokolů je popsáno v [„Bezdrátová komunikace“](#) (strana 470). Nastavení nutná na straně mobilního telefonu najdete v manuálu svého telefonu. Nastavení na straně Linuxu je popsáno v částech [34.2 – „Bluetooth“](#) (strana 520) a [34.3 – „IrDA — Infrared Data Association“](#) (strana 530).

Podporu pro synchronizaci s kapesními počítači Palm obsahují programy Evolution a Kontact. Připojení zařízení je v obou případech prováděno pomocí průvodce. Po nastavení Palm Pilota je nutné zadat typ synchronizovaných dat /adresy, schůzky, atd.). Obě aplikace jsou popsány v uživatelské příručce.

Program KPilot je součástí aplikace Kontact nebo jako nezávislý nástroj. Pro synchronizaci kontaktů lze použít také program KitchenSync.



Další informace o aplikacích Evolution a Kontakt najdete v uživatelské příručce.

## 30.4 Další informace

Hlavní zdroj informací i Linuxu na mobilních zařízeních najdete na stránce <http://tuxmobil.org/>. Podrobnosti o notebookách, kapesních počítačích, mobilních telefonech a dalších zařízeních jsou roztríděné do jednotlivých podsekcí.

Podobnou stránku jako <http://tuxmobil.org/> věnovanou pouze notebookům a kapesním počítačům najdete na adrese <http://www.linux-on-laptops.com/>.

SUSE spravuje emailovou konferenci věnovanou notebookům. Základní informace najdete na stránce <http://lists.suse.com/archive/suse-laptop/>. V této konferenci uživatelé a vývojáři probírají problematiku systému SUSE Linux a mobilních počítačů. Konference je vedena v německém jazyce, ale běžně jsou zodpovědány také dotazy v angličtině.

V případě problémů se správou napájení na notebooku se systémem SUSE Linux doporučujeme nejdříve prostudovat soubor README v adresáři `/usr/share/doc/packages/powersave`. Tento soubor obsahuje nejnovější informace vývojářů a testerů, které již nebylo možné zařadit do oficiální dokumentace.



## Linux a notebooky

U notebooků se setkáváme s řadou hardwarových zvláštností, jako je řízení spotřeby infračervený port (IrDA), karty PCMCIA a Bluetooth. Tyto komponenty nacházíme příležitostně i u stolních počítačů a protože se funkčně neliší od provedení v notebooku, bude jejich použití a konfigurace popsána společně v této kapitole.

### 31.1 Hardware

Zkratka PCMCIA znamená *Personal Computer Memory Card International Association* a používá se všeobecně pro hardware a odpovídající software tzv. karet PCMCIA, u kterých rozlišujeme dva základní typy:

Klasické karty PCMCIA (též PC-karty):

To je zatím nejběžnější typ, kde se používá 16 bitová sběrnice. Jsou dnes již cenově dostupné a obvykle fungují bez problémů a mají stabilní podporu.

CardBus karty:

Jedná se o nový standard. Používají 32 bitovou sběrnici a jsou proto rychlejší, také ovšem dražší. Protože je však přenos dat často omezen i druhou stranou spojení, nemusí se náklady na ně vyplatit. Existuje zatím několik ovladačů na tyto karty, v závislosti na použitém řadiči PCMCIA však dosud nemusí být zcela stabilní.

Další důležitou komponentou je řadič PCMCIA, nazývaný též PCMCIA/CardBus--bridge. Ten vytváří spojení mezi kartou a sběrnici PCI, ve starších počítačích sběrnici ISA. Tyto řadiče jsou téměř vždy kompatibilní s čipem Intel i82365. Typ řadiče lze

zjistit příkazem `pcic_probe`. Jedná-li se o zařízení PCI, podá nám zajímavé informace i příkaz `lspci -vt`.

## 31.2 Software

O správnou funkci PCMCIA můstků a PCMCIA karet se stará systém hotplug. Jde o události `pcmcia_socket` a `pcmcia`. `udev` zavádí všechny potřebné moduly a volá aplikace potřebné k nastavení zařízení. Akce jsou definovány v adresáři `/etc/udev/rules.d/`.

Pro nastavení zdrojů je určen soubor `/etc/pcmcia/config.opts`. Potřebný ovladač je zadán v tabulce zařízení ovladače. Informace o stavu soketů je možné získat z `/sys/class/pcmcia_socket/` a prostřednictvím `pccardctl`.

Z důvodů velmi rychlých změn není dokumentace PCMCIA systému zatím kompletní. Přehled najdete v souboru `/usr/share/doc/packages/pcmciautils/README.SUSE`.

## Správa profilů

V této kapitole je popsán SCPM (system configuration profile management). S pomocí SCPM můžete svůj počítač přizpůsobit různým pracovním prostředím nebo odlišným hardwarovým konfiguracím. SCPM spravuje pro různé situace skupinu systémových souborů. Díky tomu umožňuje rychlé přepnutí mezi systémovými profily bez nutnosti jejich ručního přenastavení.

Jsou situace, kde je nezbytné změnit systémovou konfiguraci. Pokud často provozujete svůj počítač v prostředích, kde potřebujete různá nastavení systému, možná by se vám hodilo uložit si tato nastavení a obnovit je později, kdykoliv je to potřeba. To to je typická situace například pro uživatele notebooků, kteří pracují na různých místech. Také si lze představit stolní počítač, který chcete dočasně provozovat s jinou konfigurací. V takových případech byste rádi měli záložní mechanismus, který uloží současná systémová konfigurační data a uloží je do profilu. Tímto způsobem lze potom kdykoliv tuto konfiguraci obnovit.

Hlavní doménou SCPM je nastavit síť na notebookech. Předpokládejme tedy, že máte notebook a chcete jej připojit ke své domácí i firemní síti a používat jej nezávisle, když jste na cestách. Toto obvykle vyžaduje nakonfigurovat systém tak, aby zapadl do různých sítí. Například potřebujete DHCP klienta v kanceláři a pevnou IP adresu doma. Dále máte třeba v kanceláři spuštěné služby jako xntpd, NIS klienta, ale doma pouze auto-mounter, ale žádná z těchto služeb není potřeba, pokud cestujete. Pro tyto případy vám SCPM pomůže zvládnout rozdílné konfigurace a jednoduše se mezi nimi přepínat.

SCPM toho ale umí daleko víc. Je velmi konfigurovatelný; zvládne skoro všechny možné scénáře, kdy je potřeba uložit a obnovit data v různých verzích. Dokonce jej lze použít pro spouštění skriptů v závislosti na profilech, mezi kterými je přepínáno. Více informací najdete v příslušných info stránkách.

## 32.1 Základní terminologie

Dřív než začnete používat SCPM, seznamte se prosím se základními pojmy používanými v modulu programu YaST.

- Pod *systemovou konfigurací nebo nastavením* rozumíme souhrn nastavení počítače. Všechna důležitá nastavení jako např. připojení disků, nastavení sítě, časové zóny nebo rozložení klávesnice.
- *Profil* nebo také *konfigurační profil* je nastavení systému, které bylo uloženo pod určitým jménem.
- *Aktivním profilem* rozumíme profil, který je zrovna používán. Neznamená to však, že je systém nastaven právě podle tohoto profilu, protože každý uživatel má možnost si svůj systém z určité části poupravit.
- *Zdroje* jsou v pojetí SCPM všechny části spravované systemovou konfigurací. Může jít o soubory nebo odkazy. Pojem zahrnuje také systemové služby, které v jednom profilu běží a v jiném jsou vypnuté.
- Zdroje jsou organizovány do *Skupiny zdrojů*. Tyto skupiny jsou sestaveny podle určitých logických kritérií. Znamená to, že s určitou službou obsahují také její konfigurační soubory. To umožňuje spravovat zdroje bez znalosti konfiguračních souborů jednotlivých služeb.

## 32.2 Nastavení SCPM

V zásadě jsou dostupné dvě rozhraní pro nastavení SCPM. Balíček `s-cpm` obsahuje rozhraní pro příkazovou řádku. *Správce profilů* programu YaST je určen pro grafické prostředí. Obě rozhraní mají stejnou funkčnost, ale znalost rozhraní příkazové řádky vám výrazně usnadní pochopení modulu programu YaST. Následující popis bude proto zaměřen především na textové prostředí.

## 32.2.1 Spuštění SCPM a definice skupin zdrojů

SCPM musíte nejdřív aktivovat. To provedete příkazem `scpm enable`. Při prvním spuštění dochází k inicializaci SCPM. Inicializace je časově náročnější a může zabrat několik sekund. SCPM deaktivujete a tím zabráníte nechtěnému přepnutí profilů příkazem `scpm disable`.

Standardně SCPM obsahuje nastavení pro síť, tisk a grafické prostředí. Před použitím odpovídajícího nastavení musíte nejdřív aktivovat příslušné skupiny zdrojů. Dostupné skupiny zobrazíte příkazem:

```
scpm list_groups
```

Pokud si chcete nechat vypsát pouze aktivní skupiny, zadejte příkaz:

```
scpm list_groups -a
```

Uvedené příkazy musíte vykonávat jako uživatel `root`.

```
scpm list_groups -a
```

```
nis                Network Information Service client
mail               Mail subsystem
ntpd               Network Time Protocol daemon
xf86               X-Server settings
autofs             Automounter service
network            Basic network settings
printer            Printer settings
```

Skupiny aktivujete popř. deaktivujete příkazem:

```
scpm activate_group JMENO
```

popř.

```
scpm deactivate_group JMENO.
```

Část `JMENO` nahraďte jménem zvolené skupiny. Skupiny lze spravovat také prostřednictvím správce profilů programu YaST.

## 32.2.2 Vytváření a přepínání profilů

Po aktivaci SCPM se spustí profil `default`. Seznam všech dostupných profilů získáte příkazem `scpm list`. Pouze jeden ze všech dostupných profilů může být aktivní. Jméno aktivního profilu získáte příkazem `scpm active`. Profil `default` je základní profil, ze kterého jsou všechny ostatní odvozeny. Před spuštěním správy profilů proto nastavte všechna nastavení, která chcete mít v profilech dostupná. Příkazem `scpm reload` uložíte všechny změny na systému do aktivního profilu. Profil `default` si pak můžete ponechat nebo ho smazat.

Jsou dvě možnosti, jak vytvořit nový profil. Nový profil (zde `work`) např. odvozený od profilu např. `default` vytvoříte příkazem `scpm copy default work`. Příkazem `scpm switch work` se do nového profilu můžete přepnout a provést další nastavení. V některých případech je však výhodné vytvořit profil z již existujícího právě používaného nastavení. To provedete pomocí příkazu `scpm add work`. Po zadání tohoto příkazu budete mít aktuální nastavení systému uložené v profilu `work` a ten bude označen jako aktivní; `dasheisst` že příkaz `scpm reload` uloží změny do profilu `work`.

Profily lze samozřejmě také přejmenovávat a mazat. K tomu použijte příkazy `scpm rename x y` a `scpm delete x`. K přejmenování např. `work` na `prace` použijte příkaz `scpm rename work prace`. Aktivní profil nelze smazat.

Další příkazy:

```
scpm list  
zobrazení seznamu dostupných profilů
```

```
scpm active  
zobrazení aktivního profilu
```

```
scpm add Jmeno  
uložení aktuálního nastavení systému do profilu a nastavení tohoto profilu jako  
aktivního
```

```
scpm copy Jmeno NoveJmeno  
kopírování profilu
```

```
scpm rename Jmeno NoveJmeno  
přejmenování profilu
```



```
scpm delete Jmeno  
smazání profilu
```

Poznámka k modulu programu YaST: Při prvním spuštění máte k dispozici pouze nabídku *Volby*. Až po spuštění správy profilů, získáte možnost vybrat si jeden z předdefinovaných profilů, který se uloží jako profil `default`. Až pak získáte další možnosti úpravy.

## 32.2.3 Přepínání mezi profily

Pokud se chcete přepnout do jiného profilu použijte příkaz (zde `work`):

```
scpm switch work
```

Tímto příkazem vypnete aktivní profil a nastavíte nový. Před nastavením nového profilu můžete také právě aktivní profil zcela deaktivovat.

Při této změně SCPM porovná aktuální nastavení s novým profilem. Pak musí určit, které služby se budou restartovat a jaké konfigurační souboru bude potřeba načíst. Následně se spustí akce, která se jeví jako částečný systémový restart, kdy se restartují všechny měněné služby, ale zbytek systému funguje dál.

Nyní se spustí tyto akce:

- Systémové služby budou zastaveny.
- Zápis všech změněných zdrojů (např. konfigurační soubory).
- Systémové služby se (znovu) spustí.

## 32.2.4 Rozšířené nastavení

Ke každému profilu lze napsat krátký popis, který se zobrazí po zadání příkazu `scpm list`. Pro aktivní profil nastavíte popis příkazem:

```
scpm set description "text"
```

Pro neaktivní profil musíte zadat ještě jméno profilu, takže pro profil `work` bude příkaz vypadat takto:

```
scpm set description "text" work
```

Někdy je při vypínání či zapínání profilu nutné vykonat akce ještě po ukončení služeb či před jejich spuštěním. Pro každý profil jsou proto dostupné čtyři programy nebo skripty, které se vykonávají v různých fázích při přepnutí. Tyto body jsou následující:

prestop

    před zastavením služby při ukončení profilu

poststop

    po zastavení služby při ukončení profilu

prestart

    před spuštěním služby při aktivaci profilu

poststart

    po spuštění služby při aktivaci profilu

Přepnutí z profilu `work` na `home` funguje takto:

- Prestop akce profilu `work`
- Zastavení služeb
- Poststop akce profilu `work`
- Změna nastavení
- Prestart akce profilu `home`
- Spuštění služeb
- Poststart akce profilu `home`

Tyto akce lze vykonat příkazem `set`. Použití je takové:

```
scpm set prestop JmenoSouboru
```

```
scpm set poststop JmenoSouboru
```

```
scpm set prestart JmenoSouboru
```

nebo

```
scpm set poststart JmenoSouboru
```

Všechny tyto příkazy vykonává uživatel root.

---

### Varování

Protože tyto skripty mohou obsahovat citlivé informace o systému, měly by být čitelné pouze pro administrátora systému. Nejvhodnější je tedy nastavit souboru práva na `-rwx----- root root`. (`chmod 700 JmenoSouboru` a `chown root.root JmenoSouboru`).

---

Všechna nastavení provedená pomocí `set` lze získat příkazem `get`. Například příkaz `scpm get poststart` vypíše jméno poštovního programu nebo krátkou informaci, pokud není nic nastaveno.

Příkazy `set` a `get` lze aplikovat také na profil. K tomu účelu musíte zadat jméno profilu. Například:

```
scpm get prestop JmenoSouboru work
```

nebo

```
scpm get prestop work.
```

## 32.3 Volba profilu při startu

Profil při startu systému zvolíte tak, že během zobrazení startovacího seznamu stisknete klávesu `F4` a ze seznamu zvolíte požadovaný profil. Po seznamu se lze pohybovat pomocí šipek. Start do zvoleného profilu spustíte stisknutím klávesy `Enter`. Zvolený profil je pak použit jako startovací parametr.

## 32.4 Problémy a jejich řešení

SCPM není v současné době stále ještě možné aktualizovat spolu se systémem. problém spočívá ve skutečnosti, že se konfigurační soubory nacházejí na celé řadě míst, kam mechanismus aktualizace nemůže zasahovat. SCPM je však schopné aktualizaci rozpoznat a po jejím provedení vám nahlásí:

Vaše instalace se změní nebo je neznámá

V takovém případě stačí SCPM reinicializovat příkazem:

```
scpm -f enbale
```

Některé profily však mohou být při aktualizaci zcela ztraceny. V takovém případě není jiná cesta, než je znovu vytvořit.

Za určitých okolností se může stát, že SCPM při pokusu o přepnutí profilu přestane pracovat. K tomuto stavu může dojít např. při nenadálém vypnutí systému. Při spuštění SCPM obdržíte hlášení, že je SCPM zamčen. Tato služba chrání data v databázi SCPM v případě, že dojde k problémům se systémem. V takovém případě smažte soubor příkazem:

```
rm /var/lib/scpm/#LOCK
```

a obnovte SCPM zadáním:

```
scpm -s reload.
```

Pak již budete moci bez problémů pracovat.

## 32.4.1 Změna nastavení skupiny zdrojů

Změna v nastavení skupiny v již inicializovaném SCPM nepředstavuje v zásadě žádný problém. Po změně nebo smazání skupiny pouze musíte zadat příkaz:

```
scpm rebuild
```

Tento příkaz zavede do skupiny nové zdroje a smaže ty, které jste se rozhodli odstranit. Pokud provádíte změny pomocí programu YaST, není výše uvedený příkaz nutný. Programem YaST provedete všechna nutná nastavení a příkazy automaticky.

## 32.5 Další informace

Nejnovější dokumentace je dostupná na infostránkách SCPM, které si můžete prohlédnout např. pomocí programu Konqueror nebo Emacs (`konqueror info:scpm`). Na

příkazové řádce pomocí příkazu `info` nebo `pinfo`. Informace od vývojářů jsou dostupné v souboru `/usr/share/doc/packages/scpm`.



# Správa napájení

V této kapitole najdete stručný úvod do správy napájení v systému Linux. Popsány jsou oba v současné době používané standardy APM (Advanced Power Management) a ACPI (Advanced Configuration and Power Interface).

Na rozdíl od APM používaného pouze pro správu napájení, je ACPI nástroj umožňující získávání informací o hardwaru a jeho nastavení. V moderních počítačích je tak například možné nastavit frekvenci procesoru podle situace a dosáhnout tím významné úspory energie, což je velmi užitečné především u mobilních zařízení napájených z baterií.

Všechny technologie správy napájení vyžadují podporu v BIOSu a vhodný hardware. Řada moderních notebooků, pracovních stanic a serverů tyto podmínky splňuje. APM je dnes již používáno jen na starších počítačích. Protože se skládá především z funkcí implementovaných v BIOSu, je závislý na hardwaru. To platí také o ACPI, který je však mnohem komplexnější. Z toho důvodu je nemožné upřednostnit jednu technologii před druhou. Jednoduše otestujte potřebné funkce obou technologií na svém počítači a zvolte tu nejlepší.

---

## **Důležité: Správa napájení procesorů AMD64**

U procesorů AMD64 a 64 bitového jádra je podporován pouze ACPI.

---

## 33.1 Funkce šetření spotřeby

Celá řada funkcí, které správa napájení poskytuje, má největší uplatnění v oblasti mobilních počítačů. Nejdůležitější jsou tyto:

### Pohotovost (*standby*)

V tomto režimu se pouze vypne displej a u novějších počítačů se sníží příkon procesoru.

### Uspání do paměti (*suspend to memory*)

V tomto režimu se stav systému uloží *do paměti* a počítač (kromě této paměti) přestane pracovat. Spotřeba je pak nepatrná, takže pak počítač (podle typu) vydrží v tomto režimu pracovat na baterii 12 hodin až několik dní. Tento režim má oproti vypnutí tu výhodu, že je opět pohotovými po několika sekundách přesně v tom místě, kde skončil, aniž by bylo potřeba znovu startovat a zavádět potřebné programy. U Linuxu, který *nepotřebuje* být čas od času restartován z důvodu obnovy stability -- jako některé nejmenované systémy -- je tato možnost zvláště zajímavá. U moderních notebooků stačí jen zaklapnout víko, aby přešly do suspendovaného režimu. Opětvým odklopením víka notebook opět ožije.

### Uspání na disk (*hibernation, suspend to disk*)

V tomto režimu počítač doslova přezimuje období své nečinnosti. Současný stav se nejprve uloží *na disk* a počítač se pak sám vypne. Zpětné probuzení ze zimního spánku do stavu před usmáním pak ovšem trvá mezi 30 až 90 sekundami. Systém se spustí do původního stavu před usmáním. Někteří výrobci nabízejí různé hybridní varianty (např. RediSafe v IBM Thinkpadech). Odpovídající ACPI režim je S4. V Linuxu je *uspání na disk* prováděno rutinami nezávislými na APM a ACPI.

### Kontrola stavu baterií

APM i ACPI kontrolují stav baterií a při dosažení kritického stavu spouštějí zadané operace.

### Automatické vypnutí po zastavení systému

Hodí se i pro stolní počítače. Po zastavení systému *shutdown* se počítač (elektricky) vypne. Důležité např. v případě vybití baterií, kdy je nutné počítač korektně vypnout.

### Vypínání komponent

Šetří významně spotřebu a např. u hlučných disků i vaše nervy. U disků ovšem třeba brát ohled na editory, které v pravidelných intervalech nemilosrdně budí disk na záložní kopie.

### Kontrola výkonu procesoru

V případě CPU můžete energii spořit třemi způsoby: změnou napětí a frekvence (PowerNow! nebo Speedstep), přiškrcením a usmáním procesoru (C stavu). V závislosti na operačním režimu počítače lze tyto metody také kombinovat.



Některé z těchto funkcí podporuje již samotný BIOS. Úsporný režim *pohotovost a uspání do RAM* realizují notebooky klávesovou kombinací nebo detekcí zaklapnutí víka. Tyto funkce jsou nezávislé na operačním systému, při vhodném jádru a nainstalovaných balících je však můžeme navíc volat i pomocí linuxových příkazů.

## 33.2 APM

Některé funkce již obsahuje APM BIOS. Uspání a probuzení dokáže aktivovat mnoho notebooků pomocí klávesové kombinace nebo uzavřením víka. K tomu nejsou zapotřebí žádné funkce poskytované operačním systémem.

Podpora APM je přímo součástí standardního jádra a je automaticky aktivována v případě, že při startu je nalezen APM-BIOS a deaktivována podpora ACPI parametrem `acpi=off`. Když chcete vypnout podporu APM při startu, můžete to udělat parametrem `apm=off`. Zda je APM aktivováno, zjistíte velice jednoduše příkazem `cat /proc/apm`. Pokud se zobrazí řádek s různými čísly, pak je vše v pořádku.

Protože se některé implementace BIOSu nedrží platných standardů, dochází k zajímavému chování. Něco je možné obejít parametry při startu systému. Můžete použít např.:

on nebo off

Zapnout/vypnout podporu APM

(no-)allow-ints

Povolit během spouštění funkcí BIOSu přerušení

(no-)broken-psr

BIOS má vadnou funkci `GetPowerStatus`

(no-)realmode-power-off

Processor se přepne před ukončením chodu do reálného režimu

(no-)debug

Hlášení APM jsou protokolována v syslogu

(no-)power-off

Po zadání shutdown se počítač vypne

bounce-interval=*n*

Čas v setinách sekundy, kdy po přijetí výsledku uspání budou další požadavky ignorovány

idle-period=*n*

Čas v setinách vteřiny po kterém bude sdělena (ne)aktivita systému.

APM démon (apmd) již není používán. Jeho funkce jsou součástí nového démona powersaved, který podporuje také ACPI a nastavení frekvence CPU.

## 33.3 ACPI

ACPI je zkratka z *Advanced Configuration and Power Interface*. ACPI umožňuje operačnímu systému nastavit a kontrolovat spotřebu jednotlivých hardwarových součástí. Svou funkcí nahrazuje jak PnP tak APM. Část ACPI zodpovědná za inicializaci hardwaru není v této kapitole popsána.

BIOS poskytuje tabulku obsahující informace o jednotlivých komponentech a metodách přístupu. Tyto informace pak použijte operační systém např. k přiřazení přerušeni či aktivaci nebo deaktivaci tohoto zařízení. Jaké operace může operační systém provést, záleží na implementaci BIOSu. Záznamy ACPI o nalezení a použití tabulky najdete v souboru `/var/log/boot.msg`. Detekované a zavedené ACPI tabulky jsou zapsány do `/var/log/boot.msg`. Více o této problematice najdete v části [33.3.4 – „Možné problémy“](#) (strana 495).

### 33.3.1 ACPI v praxi

Když jádro detekuje při startu ACPI BIOS, ACPI se automaticky aktivuje (a APM deaktivuje). Některé starší počítače važdují pro spuštění ACPI zadání parametru jádra `acpi=on`. Počítač musí podporovat ACPI 2.0 nebo vyšší. Zda se ACPI aktivovalo, zjistíte ze záznamu jádra v souboru `/var/log/boot.msg`.

Zavádění modulů obstarává startovací skript ACPI démona. Pokud se při zavádění některého modulu objeví problémy, je možné ho vyřadit zápisem v souboru `/etc/sysconfig/powersave/common`.

Hlášení modulů, která vám umožní zjistit detekované komponenty, najdete v systémovém záznamu (`/var/log/messages`).

`/proc/acpi` nyní obsahuje řadu souborů s informacemi o stavu systému a možných změnách. Některé funkce se stále vyvíjejí a nejsou stále plně funkční. Podpora řady dalších funkcí je závislá na implementaci výrobce.

Všechny soubory (kromě `dsdt` a `fadt`) lze číst pomocí příkazu `cat`. V řadě souborů lze nastavení měnit, použít můžete např. příkaz `echo`. U nastavení vhodných hodnot pro X Window bude příkaz vypadat takto:

```
echo X ><soubor>.
```

K přístupu k těmto informacím vždy používejte příkaz `powersave`. Nejdůležitější soubory s nastaveními správy napájení jsou:

```
/proc/acpi/info
```

Základní informace o ACPI

```
/proc/acpi/alarm
```

Doba, kdy má dojít k probuzení. Nastavení je bezpředmětné v případě, že probuzení nefunguje. V současné době není tato funkce plně podporována.

```
/proc/acpi/sleep
```

Poskytuje informace o možných stavech uspaní. V současné době jsou funkční pouze S1 (standby) a S5 (vypnout, neuklízet): `echo 1 > /proc/acpi/sleep`.

```
/proc/acpi/event
```

Zde jsou ukládány záznamy o všech událostech. Ty jsou vykonávány démony 'acpid' nebo 'ospm'd'. Pokud k souboru nepřistupuje žádný démon, události lze číst příkazem `cat /proc/acpi/event` (ukončení stisknutím `Ctrl` + `C`).

```
/proc/acpi/dsdt a /proc/acpi/fadt
```

tento soubor obsahuje ACPI tabulky DSDT a FADT. Soubor lze číst pomocí `acpidmp`, `acpidisasm` a `dmdecode`.

Příklad: `acpidmp DSDT | acpidisasm`.

```
/proc/acpi/ac_adapter/AC/state
```

Je připojen AC adaptér?

```
/proc/acpi/battery/BAT*/{alarm,info,state}
```

Detailní informace o stavu baterií.

`/proc/acpi/button`

Tento adresář obsahuje informace o přepínačích.

`/proc/acpi/fan/FAN/state`

Ukazuje aktivitu větráčku. Lze ho také manuálně vypnout/spustit zapsáním 0 (zapnutý) nebo 3 (vypnutý) do tohoto souboru. V případě vysoké teploty může jádro toto nastavení přepsat.

`/proc/acpi/processor`

Adresář s podadresáři pro každý procesor ve vašem systému.

`/proc/acpi/processor/*/info`

Informace o úsporách energie procesoru.

`/proc/acpi/processor/*/power`

Informace o stavu procesoru.

`/proc/acpi/processor/*/throttling`

Zde se dá povolit lineární přiškrcení procesoru.

`/proc/acpi/processor/*/limit`

Nastavení limitů při použití omezení výkonu a přiškrcení procesoru. Nacházejí se zde jak systémové tak uživatelské limity. Příkazem `echo 1:5 >`

`/proc/acpi/processor/*/limit` předejdete použití stavů P0 nebo T0--T4.

`/proc/acpi/thermal_zone/`

Podadresáře pro jednotlivé teplotní zóny. termální zóna je oblast s určitými teplotními vlastnostmi, číslem a jménem určeným výrobcem zařízení. Velká část funkcí bohužel není implementována. Nejvhodnější ovládání je stále přímo prostřednictvím BIOSu. Některé z následujících nastavení mohou být pouze teoretické.

`/proc/acpi/thermal_zone/*/temperature`

Současná teplota teplotní zóny.

`/proc/acpi/thermal_zone/*/state`

Stav může být ok, aktivní nebo pasivní chlazení. Vše je ok v případě ovládání větráčku nezávisle na ACPI.

`/proc/acpi/thermal_zone/*/cooling_mode`

Volba výchozího chlazení v případě nasazení kontroly ACPI. Může být aktivní (méně úsporné, ale výkonnější) nebo pasivní (méně výkonné, ale úsporné).

`/proc/acpi/thermal_zone/*/trip_points`

Nastavení teploty pro pasivní nebo aktivní chlazení, usnutí nebo bezpečnostní vypnutí.

`/proc/acpi/thermal_zone/*/polling_frequency`

Hodnota v `temperature` není automaticky obnovována se změnou teploty, přepněte na `polling mode`. Příkaz `echo X >`

`/proc/acpi/thermal_zone/*/polling_frequency` zapíše aktuální hodnotu každých `X` second. Nastavením `X=0` polling deaktivujete.

Žádné z těchto nastavení není nutné provádět ručně. Použít můžete buď přímo Powersave démona (`powersaved`) nebo některou z aplikací jako `powersave`, `kpowersave` nebo `wmpowersave`. Více informací najdete v části 33.3.3 – „Nástroje ACPI“ (strana 494). Protože `powersaved` obsahuje všechny funkce staršího démona `acpid`, není již démon `acpid` potřebný.

## 33.3.2 Nastavení výkonu CPU

V případě procesoru lze snížit spotřebu energie třemi různými způsoby a v závislosti na operačním režimu lze tyto metody kombinovat. Nižší spotřeba vede k nižšímu zahřívání procesoru a méně častému spouštění větráčků.

Frekvence a napětí

Technologie nastavení frekvence a napětí PowerNow! a Speedstep byly navrženy společností AMD a Intel. Tyto technologie jsou implementovány také v procesorech jiných výrobců. Současné snížení frekvence a napětí vede k více jak lineárním úsporám energie, což znamená, že při snížení frekvence na polovinu, je spotřeba energie méně než poloviční. Technologie jsou závislé na APM nebo ACPI a vyžadují pro nastavení frekvence příslušného démona. Nastavení lze provést v adresáři `/sys/devices/system/cpu/*/cpufreq/`.

Přiškrcení

Pomocí této technologie lze přenastavit procento signálů časovače pro CPU. V případě 25% přiškrcení je vynechán každý čtvrtý impuls a k procesoru se dostane pouze 87.5% obvyklých signálů. Úspora energie je však menší než lineární. Obvykle

se přiškrcování používá, pokud není dostupná změna frekvence nebo je nutné dosáhnout maximální úspory energie. Tato technologie vyžaduje kontrolu zvláštním procesem. Systémové rozhraní je v `/proc/acpi/processor/*/throttling`.

### Uspání procesoru

Operační systém v případě nečinnosti procesor uspí zasláním příkazu `halt`. Uspání má stavy C1, C2 a C3. V neekonomičtějším stavu C3 je zastavena také synchronizace vyrovnávací paměti procesoru a operační paměti. Tento stav je tedy možné nastavit pouze v případě, že žádné zařízení nepřistupuje k operační paměti a nemění její obsah. Některé ovladače vylučují uvedení do stavu C3. Aktuální stav můžete zjistit v souboru `/proc/acpi/processor/*power`.

Změna frekvence a přiškrcování jsou učiněné pouze při velkém zatížení procesoru, protože u nevytíženého procesoru je automaticky aplikován ekonomický režim C. V případě pracujícího procesoru je doporučená metoda spoření energie změna frekvence. Ve většině případů totiž není procesor zcela vytížen a může bez problémů pracovat i na nižší frekvenci. Obvykle je nejvhodnější dynamická změna frekvence. Statické nastavení má význam pouze pokud stálá nižší frekvence vede k významným úsporám energie nebo pokud je potřeba, aby byl počítač dobře chlazený a tichý.

Přiškrcování je metoda poslední volby, např. v případě potřeby maximální vydrže baterií. Některé systémy nemusí při větším přiškrcení běžet korektně. Přiškrcení nemá žádný smysl, pokud je procesor málo vytížen.

V systému SUSE Linux jsou tyto technologie kontrolovány pomocí démona Powersave. Nastavení je popsáno v části [33.5 – „Balík powersave“](#) (strana 497).

## 33.3.3 Nástroje ACPI

K dispozici je řada více či méně komplexních ACPI nástrojů pro zobrazení informací jako např. stav baterií nebo teplota (`acpi`, `klaptopdaemon`, `wmacpimon` atd.), nástrojů umožňujících přístup ke struktuře `/proc/acpi` nebo pomáhajících monitorovat změny (`akpi`, `acpiw`, `gtkacpiw`) a také nástroje pro editaci ACPI tabulek v BIOSu (balíček `pmtools`).

## 33.3.4 Možné problémy

V zásadě se můžete setkat se dvěma základními typy problémů. V prvním případě může jít o selhání podpory ACPI v jádře. V takovém případě, hned jak bude k dispozici oprava, můžete problém vyřešit stažením a instalací novějšího typu jádra. Druhý typ problému je spojen s BIOSem počítače. Ne všichni výrobci bohužel správně dodržují ACPI specifikaci. Jejich zařízení pak nefungují správně. Zařízení s chybnou implementací ACPI jsou zařazeny na černou listinu linuxového jádra. Jádro pak pro tato zařízení ACPI nepoužije.

První krok, který byste při řešení problému s ACPI měli udělat, je update BIOSu. Tím můžete vyřešit mnoho problémů. Pokud se počítač nespouští správně, můžete použít jeden z parametrů jádra:

`pci=noacpi`

Nepoužívat ACPI pro nastavená PCI zařízení.

`acpi=oldboot`

Provést jen základní nastavení. Nepoužívat ACPI k ničemu jinému.

`acpi=off`

Vypnout ACPI.

V dalším kroku pečlivě prostudujte startovací záznamy. To můžete udělat např. příkazem `dmesg | grep -2i acpi` (nebo si nechte zobrazit všechny záznamy, protože chyba může být zapříčiněna něčím jiným). Pokud při parsování ACPI tabulky dojde k chybě, lze přepsat nejdůležitější tabulku – DSDT. To způsobí, že DSDT BIOSu bude ignorována. Jde však o značně složitý úkol, který by měl provádět pouze expert. Pro některé počítače jsou opravené DSDT tabulky dostupné na Internetu.

Při nastavení jádra máte možnost nastavit vytváření ladicích zpráv ACPI. Pokud jste překompilovali a nainstalovali jádro s ACPI laděním, mohou být výpisy jádra cennými informacemi při hledání chyby.

V případě problémů s BIOSem nebo hardwarem je vždy užitečné kontaktovat výrobce zařízení. Ne všichni výrobci jsou sice schopní poskytnout pomoc v případě podpory Linuxu, ale vždy je dobré je o svém problému informovat. Pokud se výrobce setká s větším počtem stížností na funkci svého výrobku, je větší pravděpodobnost, že chybu opraví. Pokud chcete, můžete také informovat výrobce svého hardwaru, že vám na něm Linux funguje bez jakýchkoliv problémů.

## Další informace

Dodatečnou dokumentaci najdete na následujících stránkách:

- <http://www.cpqlinux.com/acpi-howto.html> (podrobné ACPI HOWTO a DSDT opravy)
- <http://www.intel.com/technology/iapc/acpi/faq.htm> (ACPI FAQ @Intel)
- <http://acpi.sourceforge.net/> (ACPI4Linux projekt)
- <http://www.poupinou.org/acpi/> (DSDT opravy od Bruna Ducrota)

## 33.4 Zastavení disku

Pokud se disk nepoužívá, lze ho pod Linuxem zastavit. Slouží k tomu program `hdparm`, se kterým lze nastavit i další funkce disku. Volbou `-y` se disk okamžitě suspenduje, volbou `-Y` se úplně vypne. Příkazem `hdparm -S 6` se disk vypne po 30 sekundách nečinnosti. (Číslo 6 znamená počet intervalů po 5 sekundách, tj.  $6 \cdot 5 = 30$  sekund. Hodnota 0 zastavování disku zruší. U větších hodnot je větší multiplikátor, přesněji viz manuálovou stránku.)

Interní spořicí volby pro disk lze nastavit volbou `-B`. Zvolte hodnotu od 0 do 255 (úspora — výkon). Výsledek je závislý na použitém disku a není možné jej snadno odhadnout. Disk ztišíte pomocí volby `-M`. Zadejte hodnotu od 128 do 254 (tichý — hlasitý)

Často se stává, že zastavování disku je nepraktické, protože mnoho programů na něj ukládá dočasná data nebo záložní kopie, například editory. V některých případech to lze řešit, například, jak již bylo popsáno, použitím příkazu `tailf LogSoubor` při zobrazování narůstajícího výpisu.

Uvedení disku do klidu však vůbec není tak jednoduché, jak se z popisu výše může zdát. V Linuxu neustále probíhá celá řada procesů, které zapisují nebo ukládají na disk. Všechna data se před zápisem nejdříve shromažďují v zásobníku paměti. Tento zásobník spravuje Kernel Update Daemon (`kupdated`). Jakmile jsou data v zásobníku určitou dobu, dojde k vyprázdnění zásobníku zápisem na disk. Velikost zásobníku je dynamická a závisí na velikosti operační paměti. Aby byla zajištěna co největší bezpečnost dat,



stará se kupdated o tom, aby byla data na disk zapisována v pravidelných krátkých intervalech. Každých 5 sekund kontroluje zásobník a volá bdfush, pokud zásobník obsahuje data starší než 30 sekund nebo je zaplněn více než z 30 procent. Pokud máte stabilní systém, můžete toto nastavení změnit.

---

### **Důležité: Bezpečnost dat**

Změna nastavení Kernel Update démona může vést k ohrožení bezpečnosti dat. Pokud si nejste jistí, jaké důsledky budou změny mít, raději je neprovádějte.

---

Nastavení timeoutu disku a intervalu démona kupdated s hodnotami zaplnění zásobníku nastavíte v souboru `/etc/sysconfig/powermanagement`. Nastavení provedete dvakrát. Jednou pro provoz s baterií a jednou pro provoz s připojením do sítě. Další informace o tomto tématu najdete v souboru `/usr/share/doc/packages/powersave`.

Pomocí bdfush zapisují na disk metadata také žurnálovací souborové systémy jako ReiserFS nebo Ext3. Pro ošetření tohoto zápisu existuje podpora v jádře. Tato podpora byla vyvinuta především pro mobilní zařízení. Podrobnější popis této problematiky najdete v souboru `/usr/src/linux/Documentation/laptop-mode.txt`.

Další zápis na disk mohou provádět také aplikace, se kterými právě pracujete. Například naprostá většina textových editorů si vytváří bezpečnostní kopie právě editovaného textu. Pokud by došlo k pádu programu, můžete tak obnovit editovaný soubor. Toto ukládání se však provádí během editace textu a neustále aktivuje disk. Na druhou stranu, pokud deaktivujete ukládání bezpečnostní kopii, riskujete bezpečnost souboru.

Zvláštní nastavení vhodné pro situace, kdy potřebujete mít disk co nejvíce v klidu, má také démon postfix. Jde o proměnnou `POSTFIX_LAPTOP`. Pokud tuto proměnnou nastavíte na hodnotu `yes`, maximálně se omezí přístup postfix k disku. Aktivace tohoto parametru však nemá větší význam, pokud prodloužíte interval pro kupdated.

## **33.5 Balík powersave**

`powersave` je jedním z nejužitečnějších balíčků určených především pro notebooky, kde je velmi důležité kontrolovat stav baterií a proces napájení systému. Řada funkcí je užitečná i pro běžnou pracovní stanici (např. uspání/pohotovost, funkce ACPI a možnost zastavení IDE disků).

Balíček slučuje všechny funkce správy napájení. Podporuje hardware, který využívá technologie ACPI, APM, PowerNow! a např. i technologii SpeedStep. Obsahuje funkce balíčků:

- `apmd`
- `acpid`
- `ospm`
- `cpufreqd`
- `cpuspeed`
- `powersave`

Z toho důvodu není možné, pokud chcete používat `powersave`, spouštět zároveň demony obsažené ve výše jmenovaných balíčcích.

Doporučujeme vám používat `powersave` i v případě, že hardware nepodporuje všechny uvedené technologie. Případné změny hardwaru démon rozpozná automaticky.

---

### **Důležité: Informace o powersave**

Mimo této kapitoly najdete velmi užitečné informace o `powersave` také v souboru `/usr/share/doc/packages/powersave/README_POWERSAVE`.

---

## **33.5.1 Konfigurace powersave**

Nastavení `powersave` je rozděleno do několika souborů:

`/etc/sysconfig/powersave/common`

Soubor ze základním nastavením démona `powersave`. V tomto souboru lze například významně zkrátit zápis démona do záznamů (do souboru `/var/log/messages`) nastavením nižší hodnoty proměnné `DEBUG`.

`/etc/sysconfig/powersave/events`

Soubor potřebný pro zpracování systémových událostí. Každé události lze přiřadit externí akci nebo akce nebo akce vykonávané přímo démonem. V případě externích

akcí se démon snaží spustit některý ze skriptů uložený v adresáři `/usr/lib/powersave/scripts/`. Předdefinované interní akce jsou:

- `ignore`
- `throttle`
- `dethrottle`
- `suspend_to_disk`
- `suspend_to_ram`
- `standby`
- `do_suspend_to_disk`
- `do_suspend_to_ram`
- `do_standby`

`throttle` přiškrcuje procesor na hodnotu zadanou v proměnné `MAX_THROTTLING`. Tato proměnná je závislá na aktuálním schématu. `dethrottle` nastavuje procesor na plný výkon. `suspend_to_disk`, `suspend_to_ram` a `standby` zachycují systémové události režimu uspání. Tyto tři akce jsou odpovědné především za uspávání, ale vždy by měly být asociovány se zvláštními systémovými událostmi.

Adresář `/usr/lib/powersave/scripts` obsahuje skripty pro následující akce:

`notify`

Upozornění o události na textové konzoli, v grafickém prostředí nebo zvukovým signálem.

`screen_saver`

Aktivace spořiče obrazovky.

`switch_vt`

Užitečná akce v případě, že se po probuzení nebo standby režimu nechová korektně obrazovka.

## wm\_logout

Uložení všech nastavení a logy z GNOME, KDE nebo jiného grafického prostředí a provede odhlášení.

## wm\_shutdown

Uložení nastavení GNOME nebo KDE a vypnutí systému.

V případě nastavení proměnné

```
EVENT_GLOBAL_SUSPEND2DISK="prepare_suspend_to_disk  
do_suspend_to_disk",
```

provedou se při uspání na disk dva skripty nebo akce v zadaném pořadí. Démon powersaved spustí externí skript `/usr/lib/powersave/scripts/prepare_suspend_to_disk`. Po úspěšném vykonání tohoto skriptu provedete démon interní akci `do_suspend_to_disk` a poté, co skript odstraní kritické moduly, počítač uspí.

Akci tlačítka `uspání` lze pozměnit v proměnné

```
EVENT_BUTTON_SLEEP="notify_suspend_to_disk".
```

V takovém případě budou uživatelé o uspání informováni externím skriptem `notify`. Následně je generována událost `EVENT_GLOBAL_SUSPEND2DISK` vedoucí k akcím popsaným výše a bezpečnému uspání systému. Skript `notify` lze upravit pomocí proměnné `NOTIFY_METHOD` v souboru `/etc/sysconfig/powersave/common`.

```
/etc/sysconfig/powersave/cpufreq
```

Soubor obsahuje proměnné pro nastavení optimalizace dynamického nastavení frekvence procesoru.

```
/etc/sysconfig/powersave/battery
```

Omezení baterie a další pro specifická nastavení baterie.

```
/etc/sysconfig/powersave/sleep
```

V tomto souboru se aktivuje uspávání, nastavují kritické moduly, které je nutné při uspání odstranit ze systému, a určují služby, jež je nutné před uspáním nebo před režimem standby zastavit. Po probuzení počítače jsou zadané moduly opět zavedeny do systému a služby spuštěny. Proces uspání lze z důvodů bezpečného uložení souborů odložit. Výchozí nastavení se ve většině případů týká USB a PCMCIA modulů. Selhání uspání nebo režimu standby je obvykle zapříčiněno některým z modulů. Více informací o zjišťování příčin selhání najdete v části [33.5.3 – „Možné problémy“](#) (strana 503).

```
/etc/sysconfig/powersave/thermal
```

Aktivace chlazení a kontroly teploty. Podrobnosti o tomto tématu najdete v souboru `/usr/share/doc/packages/powersave/README.thermal`.

```
/etc/sysconfig/powersave/scheme_*
```

Různá schémata správy napájení závislá na situaci nasazení počítače. Mimo již přednastavených schémat se zde ukládají také vlastní schémata.

## 33.5.2 Konfigurace APM a ACPI

U APM a ACPI můžete provádět nastavení uspávání, vlastní hodnoty pro sledování stavu baterií a samozřejmě různé režimy práce lišící se např. spotřebou energie nebo hlučností.

### Uspání a probuzení

Protože režim uspání na některých počítačích stále nefunguje, je ve výchozím nastavení vypnutý. Dostupné jsou tři typy ACPI uspání a dva typy APM uspání:

Uspání na disk (ACPI S4, APM suspend)

Uložení obsahu paměti na disk. Počítač se zcela vypne a nespotřebovává elektrickou energii.

Uspání do RAM (ACPI S3, APM suspend)

Uložení stavu všech zařízení do operační paměti. Počítač potřebuje elektrickou energii pouze pro operační paměť.

Standby (ACPI S1, APM standby)

Vypnutí některých zařízení (funkce závislá na výrobci).

Aby uspání, standby a probuzení proběhly bez problémů, ujistěte se, že máte v souboru `/etc/sysconfig/powersave/events` následující nastavení (výchozí nastavení systému SUSE Linux):

```
EVENT_GLOBAL_SUSPEND2DISK=  
    "prepare_suspend_to_disk do_suspend_to_disk"  
EVENT_GLOBAL_SUSPEND2RAM=  
    "prepare_suspend_to_ram do_suspend_to_ram"  
EVENT_GLOBAL_STANDBY=  
    "prepare_standby do_standby"  
EVENT_GLOBAL_RESUME_SUSPEND2DISK=
```

```
"restore_after_suspend_to_disk"
EVENT_GLOBAL_RESUME_SUSPEND2RAM=
"restore_after_suspend_to_ram"
EVENT_GLOBAL_RESUME_STANDBY=
"restore_after_standby"
```

## Uživatелеm definovaný stav baterie

V souboru `/etc/powersave.conf` můžete nastavit tři hodnoty týkající se kapacity baterií. Jde o stavy v procentech, při jejichž dosažení buď dojde k hlášení o stavu baterií nebo se spustí nějaká akce.

```
BATTERY_WARNING=20
BATTERY_LOW=10
BATTERY_CRITICAL=5
```

Jaké akce se spustí, lze nastavit v souboru `/etc/powersave.conf`. Typy akcí nastavíte v souboru `/etc/sysconfig/powersave/common`:

```
EVENT_BATTERY_NORMAL="ignore"
EVENT_BATTERY_WARNING="notify"
EVENT_BATTERY_LOW="notify"
EVENT_BATTERY_CRITICAL="suspend"
```

Další možnosti nastavení najdete v komentářích konfiguračního souboru.

## Nastavení spotřeby na různé režimy práce

Svůj systém můžete nastavit tak, aby se při různých způsobech napájení, choval jiným způsobem. Tak můžete dočasně z důvodů šetření energie snížit výkon svého systému, a po připojení do sítě ho pak zase zvýšit. Konkrétními příklady změn nastavení jsou frekvence procesoru, aktivita disku, spořicí funkce a další vlastnosti.

V souboru `/etc/powersave.conf` můžete prostřednictvím `powersave_proxy` nastavit různé spořicí kroky. V souboru `/etc/sysconfig/powersave/common` k nim můžete nastavit různé scénáře (nazývané *schéma* nebo *profil*):

```
AC_SCHEME="performance"
BATTERY_SCHEME="powersave"
```

*Schémat* jsou uložena do jednotlivých souborů v adresáři `/etc/sysconfig/powersave`. Jméno se vždy skládá z částí: `scheme_FJmenoSchemata`. V našem případě máme dvě schémata `scheme_performance` a `scheme_powersave`. předkonfigurována jsou schémata `performance`, `powersave` a `acoustic`. Již

existující schémata můžete kdykoliv měnit pomocí programu YaST. Pomocí programu YaST můžete také schémata vytvářet a mazat.

## Další funkce ACPI

Pokud používáte ACPI, můžete si nastavit *ACPI tlačítka* (*Power*, *Sleep* a *Otevření, Zavření*). Příslušné akce pro powersave\_proxy lze nastavit v souboru `/etc/powersave.conf`. Jednotlivé akce jsou nastavené v souboru `/etc/sysconfig/powersave/common`. Více informace o nastavení najdete v komentářích těchto konfiguračních souborů.

```
EVENT_BUTTON_POWER="wm_shutdown"
```

Po stisknutí klávesy *Power* se ukončí nastavený správce oken (KDE, GNOME, fvwm...).

```
EVENT_BUTTON_SLEEP="suspend"
```

Po stisknutí klávesy *Sleep* dojde k usnutí notebooku.

```
EVENT_BUTTON_LID_OPEN="ignore"
```

Při otevření notebooku nedojde k žádné akci.

```
EVENT_BUTTON_LID_CLOSED="screen_saver"
```

Při zavření notebooku se aktivuje spořič obrazovky.

Nastavení procesoru můžete provést prostřednictvím proměnných `CPU_LOW_LIMIT` a `CPU_IDLE_TIMEOUT`. V případě, že je procesor zaneprázdněn i po vypršení timeoutu, spustí se událost nastavená v `EVENT_PROCESSOR_IDLE`. V případě dalšího zaneprázdnění procesoru se vykoná `EVENT_PROCESSOR_BUSY`.

## 33.5.3 Možné problémy

V následující části najdete nejčastější dotazy a problémy související s používáním powersave.

### Obecný postup určení příčiny problémů

Nejdřív se podívejte do souboru `/var/log/messages`. Do tohoto souboru se zapisuje řada chybových hlášení systému. Pokud v tomto souboru nic nenajdete, nastavte

v souboru `/etc/sysconfig/powersave/common` proměnnou `DEBUG` na hodnotu 7 nebo 15. Pak restartujte démona. Všechna chybová hlášení `powersave` se pak budou zapisovat do souboru `/var/log/messages`.

## ACPI je aktivován, ale klávesy ani stav baterie nereaguje podle nastavení

Zda se jedná o problémy související s ACPI zjistíte pomocí příkazu `dmesg` zadáním:

```
dmesg | grep -i acpi
```

Jestliže najdete nějaká chybová hlášení, aktualizujte BIOS. Novou verzi BIOSu najdete na stránkách výrobce své základní desky.

V případě, že chyba přetrvává i po updatu BIOSu, vyhledejte na stránkách pro svůj systém také aktuální tabulku DSDT a nahraďte jí tabulku v BIOSu:

- Ze stránky <http://acpi.sourceforge.net/dsdt/tables> si stáhněte DSDT tabulku. Ujistěte se, že jde o správný a překompilovaný soubor (obsahuje příponu `.aml` (ACPI Machine Language)). Pokud jste pro svůj systém našli takový soubor, pokračujte krokem 3.
- Pokud jste našli tabulku s příponou `.asl` (ACPI Source Language), musíte ji nejdřív pomocí `iasl` z balíčku `pmtools` překompilovat. Zadejte příkaz:

```
iasl -sa JmenoSouboru.asl
```

Nejnovější verzi programu `iasl` (Intel ACPI Compiler) najdete na stránce <http://developer.intel.com/technology/iapc/acpi/>.

- Překopírujte soubor `DSDT.aml` do systému (v našem případě `/etc/DSDT.aml`). Editujte soubor `/etc/sysconfig/kernel` a zadejte zde cestu k DSDT souboru. Spusťte příkaz:

```
mkinitrd
```

Tímto příkazem zajistíte, že se tabulka zavede ještě před startem jádra.



---

## Důležité

Náhrada DSDT tabulky vyžaduje pokročilejší znalosti správy počítače. Při nesprávném postupu může dojít k nefunkčnosti systému.

---

## Nefunguje nastavení CPU frekvence.

Překontrolujte v dokumentaci, zda je u vašeho procesoru tato funkce podporována a zda jsou zavedeny všechny potřebné moduly a nastavené správné parametry těchto modulů. Všechny potřebné informace najdete v souboru `/usr/src/linux/Documentation/cpu-freq/*`. Pokud je potřeba nastavit určité parametry, proveďte změny v souboru `/etc/sysconfig/powersave/common` pomocí proměnných `CPUFREQD_MODULE` a `CPUFREQD_MODULE_OPTS`.

## Nelze uspávat a budit počítač

V současné době je známo několik problémů s uspáváním a probouzením na systémech používajících ACPI:

- Systémy s více jak 1 GB RAM nemají v současné době podporu uspání.
- Víceprocesorové systémy nebo systémy s procesorem P4 nemají v současné době podporu uspání.

Problém může spočívat také v chybné implementaci DSDT. V takovém případě nahraďte novou DSDT podle postupu uvedeného v *Aktivovala jsem ACPI, ale klávesy ani stav baterie nereaguje podle nastavení?*

Pro APM i ACPI systémy:

Při pokusu o zavedení problémového modulu zamrzne proxy a nedojde k pokynu k uspání. To samé může nastat v okamžiku, kdy službu nebo modul nejde zastavit. V obou případech se můžete pokusit najít problémový modul pomocí úprav v souboru `/etc/sysconfig/powersave/common`:

```
UNLOAD_MODULES_BEFORE_SUSPEND=" "  
UNLOAD_MODULES_BEFORE_STANDBY=" "  
SUSPEN_RESTART_SERVICES=" "  
STANDBY_RESTART_SERVICES=" "
```

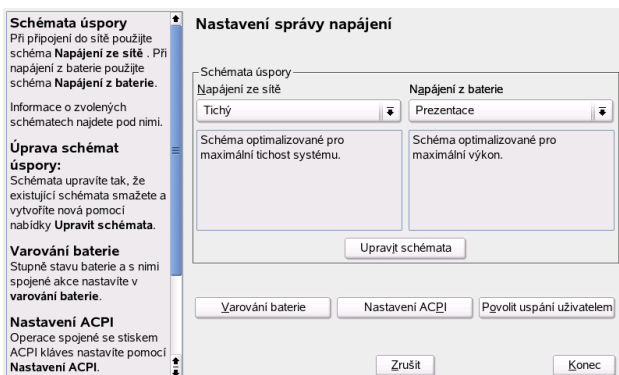
## Powersave u ACPI nesprávně rozpoznává stav baterií.

Při používání ACPI systém získává informace o stavu baterie od BIOSu. Výhoda tohoto řešení spočívá v tom, že stav baterií není nutné načítat nepřetržitě a tak je snížena zátěž systému a tím i jeho spotřeba. Může se však stát, že k přenosu informací mezi BIOSem a systémem nedochází. V takovém případě nastavte v souboru `/etc/powersave.conf` proměnnou `FORCE_BATTERY_POLLING` na hodnotu `yes`.

## 33.6 Modul správy napájení programu YaST

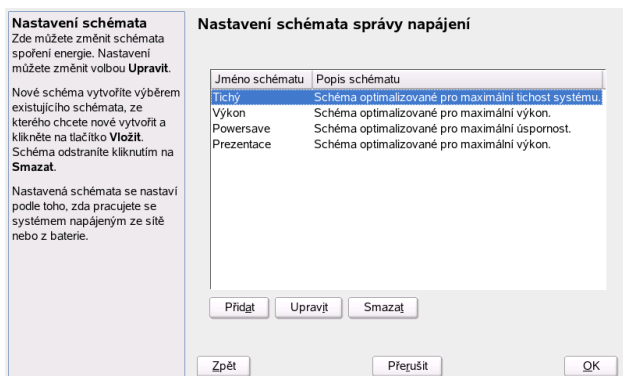
S modulem správy napájení programu YaST lze provést všechna výše zmíněná nastavení. Spustíte jej volbou *Systém* → *Správa napájení*. Modul správy napájení je zobrazen na obrázku 33.1 – „Výběr schéma“ (strana 506).

**Obrázek 33.1** Výběr schéma



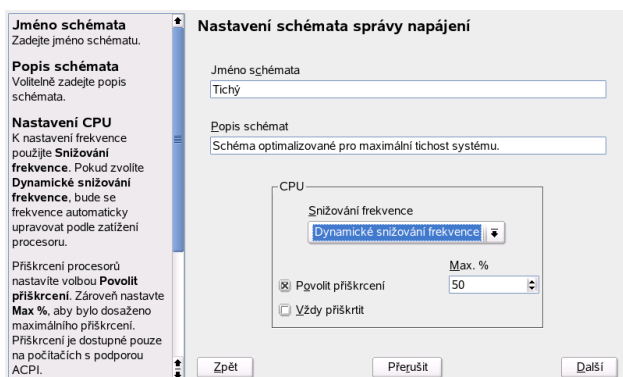
V dialogu správy napájení zvolte schéma, které chcete používat. Pokud chcete přidat nové schéma nebo upravit stávající, klikněte na tlačítko *Upravit schéma*. Otevře se dialog podobný obrázku 33.2 – „Přehled existujících schémat“ (strana 507).

## Obrázek 33.2 Přehled existujících schémat



V seznamu schémat vyberte to, které chcete upravit a klikněte na tlačítko *Upravit*. Nové schéma přidáte kliknutím na tlačítko *Přidat*. Dialog, který se otevře, můžete vidět na obrázku 33.3 – „Přidání schéma“ (strana 507).

## Obrázek 33.3 Přidání schéma

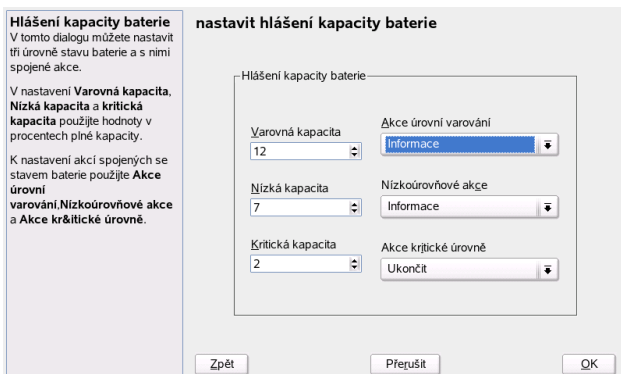


Nejdřív u upravovaného nebo nového schématu zadejte jméno a popis. Definujte ovládání výkonu procesoru. Nastavit můžete změnu frekvence CPU a příškrvcování. V následujícím dialogu nastavte politiku disku a chlazení. Některé metody politiky chlazení nemusí být podporovány BIOSem. Přesnější informace o používání větráčků a pasivním chlazení najdete v souboru `/usr/share/doc/packages/powersave/README.thermal`. Po nastavení požadovaných hodnot klikněte na tlačítko *Další*. V následujícím dialogu nastavte spojení monitoru. Po nastavení všech hodnot se vraťte

do úvodního dialogu kliknutím na tlačítko *OK*. Nově vytvořené schéma aktivujete a modul ukončíte kliknutím na tlačítko *OK*.

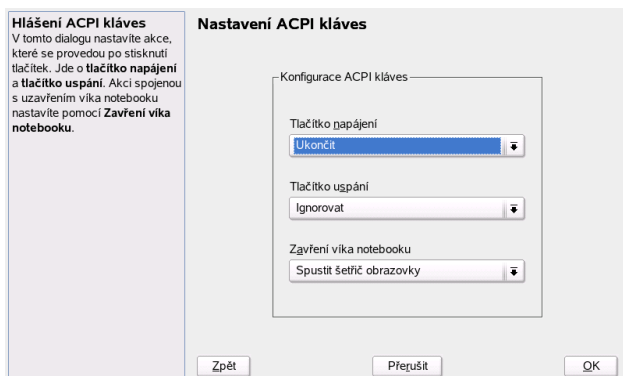
Obecná nastavení správy napájení lze provést také z dialogu *Varování baterie, Nastavení ACPI* nebo *Povolit uspání uživatelem*. Kliknutím na *varování baterie* se otevře dialog zobrazen na obrázku 33.4 – „Nabíjení baterie“ (strana 508).

**Obrázek 33.4** Nabíjení baterie



Po překročení určené kapacity napájení BIOS varuje operační systém. V tomto dialogu můžete nastavit tři různé typy limitů: *Varovná kapacita*, *Nízká kapacita* a *Kritická kapacita*. Po překročení těchto limitů se provedou k nim přidružené akce. U prvních dvou se obvykle jedná o varování. Třetí limit vede k vypnutí počítače, protože není možné nadále napájet systém. Po nastavení limitů a jejich akcí se vraťte do úvodního dialogu kliknutím na tlačítko *OK*.

## Obrázek 33.5 Nastavení ACPI



ACPI tlačítka nastavíte v dialogu dostupném po kliknutí na *Nastavení ACPI*. Dialog je znázorněn na obrázku 33.5 – „Nastavení ACPI“ (strana 509). Nastavení ACPI tlačítek určuje, jak bude systém reagovat na stisknutí určitých tlačítek jako tlačítko uspávání nebo také zavření víka notebooku. Po nastavení limitů a jejich akcí se vraťte do úvodního dialogu kliknutím na tlačítko *OK*.

Kliknutím na tlačítko *Povolit uspání uživatelem* vyvoláte dialog, ve kterém můžete nastavit možnosti uživatelů používat funkce uspání a probouzení. Po nastavení limitů a jejich akcí se vraťte do úvodního dialogu kliknutím na tlačítko *OK*. Dalším kliknutím na tlačítko *Konec* aktivujete všechny změny ve správě napájení.



## Bezdrátová komunikace

V linuxovém systému si můžete zvolit, jakým způsobem bude váš notebook komunikovat s ostatními počítači, mobilem nebo periferními zařízeními. Pro připojení počítače do sítě nejspíš zvolíte WLAN (*Wireless LAN*). Bluetooth slouží nejčastěji k připojení jednotlivých periférií (myš, klávesnice), mobilů, PDA a propojení počítačů. IrDA je nejčastěji používána při komunikaci s PDA nebo mobilním telefonem. V této kapitole najdete informace o základním nastavení všech tří možností.

### 34.1 Bezdrátové sítě

Bezdrátové sítě jsou významnou součástí mobilní výpočetní techniky. V současné době má velká část notebooků integrovanou WLAN kartu. Standard 802.11 bezdrátové komunikace WLAN karet byl připraven organizací IEEE. Původně umožňovat maximální rychlost 2 Mb/s. Prošel však řadou změn, které umožnily rychlost zvýšit. Tyto změny definují podrobnosti jako modulaci, přenosový výstup a rychlosti:

**Tabulka 34.1** Přehled různých WLAN standardů

Jméno	Pásmo (GHz)	Max. přenosová rychlost (Mb/s)	Poznámka
802.11	2.4	2	Zastaralý
802.11b	2.4	11	nejrozšířenější
802.11a	5	54	Méně obvyklý

Jméno	Pásmo (GHz)	Max. přenosová rychlost (Mb/s)	Poznámka
802.11g	2.4	54	Zpětně kompatibilní s 11b

Dostupné jsou také proprietární variace 802.11b např. od společnosti Texas Instruments s maximální přenosovou rychlostí 22 Mb/s (standard někdy označovaný jako 802.11b+). Rozšíření těchto karet však není velké.

## 34.1.1 Hardware

Karty 802.11 nejsou systémem SUSE Linux podporovány. Podporována je ale většina karet používajících protokoly 802.11a, 802.11b a 802.11g. Nově karty obvykle podporují standard 802.11g, ale dostupné jsou také karty s podporou 802.11b. Podporovány jsou karty obsahující následující čipové sady:

- Aironet 4500, 4800
- Atheros 5210, 5211, 5212
- Atmel at76c502, at76c503, at76c504, at76c506
- Intel PRO/Wireless 2100, 2200BG, 2915ABG
- Intersil Prism2/2.5/3
- Intersil PrismGT
- Lucent/Agere Hermes
- Ralink RT2400, RT2500
- Texas Instruments ACX100, ACX111
- ZyDAS zd1201

Podporována je také řada již nevyráběných starších karet. Vyčerpávající seznam WLAN karet a čipových sad je dostupný na stránce *AbsoluteValue Systems*: [http://www.linux-wlan.org/docs/wlan\\_adapters.html.gz](http://www.linux-wlan.org/docs/wlan_adapters.html.gz). Seznam různých WLAN



čipových sad najdete na stránce <http://wiki.uni-konstanz.de/wiki/bin/view/Wireless/ListeChipsatz>.

Některé karty vyžadují při zavádění ovladače nahrání obrazu s firmwarem. To je případ karet Intel PRO/Wireless 2100 (Centrino), Intersil PrismGT, Atmel a ACX100. Firmware lze snadno doinstalovat pomocí YaST online updatu. Více informací o této problematice najdete v souboru `/usr/share/doc/packages/wireless-tools/README.firmware`.

## 34.1.2 Funkce

Tato část popisuje základní aspekty bezdrátového síťování, operační režimy a způsoby ověřování a šifrování.

### Operační režimy

Bezdrátové sítě lze označit jako spravované (*managed*) nebo ad-hoc sítě. Spravované sítě mají kontrolní bod označovaný obvykle jako přístupový bod. V tomto režimu (také označovaném jako infrastructure), jsou všechny stanice připojené přes přístupový bod, který zároveň slouží jako připojení k Ethernetu. Ad-hoc sítě žádný přístupový bod nemají. jednotlivé stanice komunikují přímo mezi sebou. Protože je v ad-hoc sítích výrazně omezený rozsah vysílání a počet stanic, je přístupový bod vhodnějším řešením. Jako přístupový bod lze použít naprostou většinu WLAN karet.

Protože je bezdrátovou sítí snadnější odposlouchávat a kompromitovat než sítí klasickou, řada standardů obsahuje ověřovací a šifrovací metody. V původní verzi standardu IEEE 802.11 jsou popsány pod termínem WEP. WEP však nebyl dostatečně bezpečný (viz. „Bezpečnost“ (strana 519)) a tak WLAN výrobci (sdružení do skupiny známé jako *Wi-Fi Alliance*) definovali nové rozšíření WPA, které mělo odstranit slabiny WEP. Pozdější standard IEEE 802.11i (také nazývaný WPA2, protože WPA je založeno na 802.11i) obsahoval nejen WPA, ale také řadu dalších ověřovacích a šifrovacích metod.

### Ověřování

Aby bylo zajištěno, že dojde pouze k ověřeným připojením, obsahují spravované sítě několik ověřovacích mechanismů:

## Otevřený

Otevřený (anglicky *open*) systém nevyžaduje ověření. Do sítě se může připojit každá stanice, ale může být použito WEP šifrování (viz. „Šifrování“ (strana 515)).

## Sdílený klíč (podle IEEE 802.11)

Při této proceduře je používán pro ověření WEP klíč. Tento postup však není doporučován, protože je poměrně náchylný na útoky zvenčí. Vše, co potenciální útočník potřebuje k úspěšnému průniku, je naslouchat komunikaci. Během ověřovacího procesu si obě strany vyměňují stejné informace. Jednou v šifrované a jednou v nešifrované formě. Tak je poměrně jednoduché s pomocí příslušných nástrojů rekonstruovat použitý klíč. Vzhledem k použití klíče pro ověřování i šifrování není tato metoda zvýšení bezpečnosti sítě. Stanice se správným WEP klíčem se může přihlásit do sítě a šifrovat a dešifrovat provoz. Stanice bez klíče nemůže dešifrovat příchozí pakety ani komunikovat.

## WPA-PSK (podle IEEE 802.1x)

WPA-PSK (PSK je zkratka z *Pre-Shared Key*) pracuje podobně jako sdílený klíč. Stanice i přístupový bod používají jeden klíč. Klíč má 256 bitů a obvykle je zadáván jako heslo. tento systém nepotřebuje komplexní správu klíčů jako WPA-EAP a je vhodný pro běžné domácí používání. Proto se někdy o WPA-PSK mluví jako o WPA *home* nebo-li *domácím* WPA.

## WPA-EAP (podle IEEE 802.1x)

WPA-EAP ve skutečnosti není ověřovací systém ale protokol transportu ověřovacích informací. WPA-EAP je používán v podnikovém prostředí. V domácím prostředí je používán zřídka. Z toho důvodu se o WPA-EAP mluví jako o WPA *Enterprise* nebo-li *podnikovém* WPA.

WPA-EAP vyžaduje k ověřování uživatelů Radius server. EAP pak nabízí tři různé způsoby připojení a ověření k tomuto serveru: TLS (Transport Layer Security), TTLS (Tunneled Transport Layer Security), a PEAP (Protected Extensible Authentication Protocol). Jejich princip je následující:

### EAP-TLS

TLS ověřování je založeno na výměně certifikátů mezi klientem a serverem. TSL ověřování vyžaduje funkční správu certifikátů v síti a jen jen zřídka k vidění v malých domácích sítích.

## EAP-TTLS a PEAP

TTLS a PEAP jsou dvouúrovňové protokoly. V první úrovni je navázáno bezpečné připojení a v druhé dochází k výměně ověřovacích dat. Tento způsob ověřování je na rozdíl od TLS jen minimálně zatěžován potřebou správy certifikátů.

## Šifrování

Existuje řada šifrovacích metod, které se používají k zamezení čtení datových paketů neautorizovanými osobami a přístupu do sítě. Nejdůležitější jsou tyto:

### WEP (definován v IEEE 802.11)

Tento standard používá šifrovací mechanismus RC4 původně s délkou klíče 40 bitů, později s 104 bity. Zda je délka deklarována jako 64 bitů nebo 128 bitů často závisí na tom, zda je zahrnut také 24 bitový inicializační vektor. Tento standard má řadu slabín a klíče mohou být cílem případného útoku. Přesto je WEP vždy lepší než žádné šifrování.

### TKIP (definován v WPA/IEEE 802.11i)

Tento protokol správy klíčů definovaný v standardu WEP používá stejný šifrovací algoritmus jako WEP, ale neobsahuje jeho chyby. Nový klíč je generován pro každý datový paket, což výrazně snižuje pravděpodobnost úspěšného útoku. TKIP se používá současně s WPA-PSK.

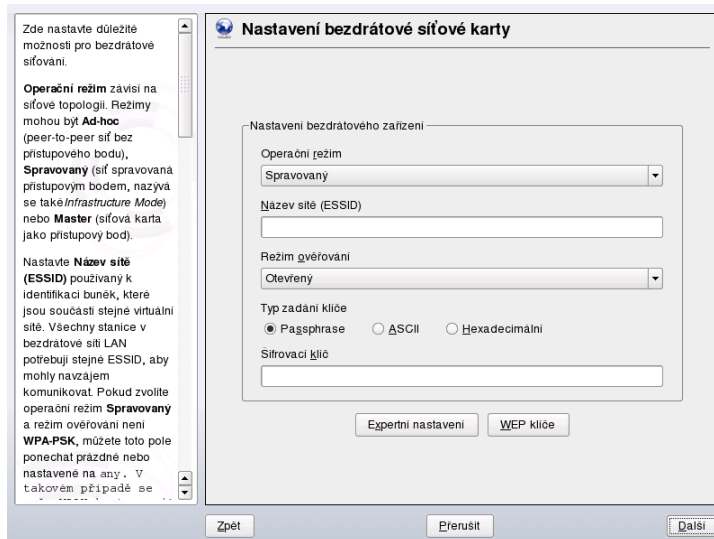
### CCMP (definován v IEEE 802.11i)

CCMP popisuje správu klíčů. Obvykle je používán současně s WPA-EAP, ale lze jej používat také s WPA-PSK. Šifrování se řídí podle AES a je silnější než RC4 nebo WEP standard.

## 34.1.3 Nastavení pomocí programu YaST

Bezdrátovou síťovou kartu nastavíte pomocí programu YaST v nabídce *Síťová zařízení* → *Síťová karta*. V části *Konfigurace sítě*, nastavte typ zařízení na *Bezdrátová technologie* a klikněte na tlačítko *Další*.

**Obrázek 34.1** YaST: nastavení bezdrátové síťové karty



V dialogu *nastavení bezdrátové síťové karty* na obr. 34.1 – „YaST: nastavení bezdrátové síťové karty“ (strana 516) provedete základní nastavení:

### Operační režim

Stanici lze zařadit do sítě ve třech různých režimech. Zvolený režim je závislý na typu sítě: *Ad-hoc* (peer-to-peer bez přístupového bodu), *Spravované* (spravovaná síť s přístupovým bodem) nebo *Master* (karta je používána jako přístupový bod).

### Jméno sítě (ESSID)

Aby mohly stanice v jedné síti spolu komunikovat, musí používat stejné ESSID. Pokud žádné nezvolí, karta automaticky nastaví některé z dostupných, to však nemusí být to, které chcete používat.

### Režim ověřování

Zvolte vhodný režim ověřování pro svou síť: *Otevřený*, *Sdílený klíč*, nebo *WPA-PSK*. Pokud zvolíte *WPA-PSK*, musíte nastavit jméno sítě.

### Expertní nastavení

Stisknutím tohoto tlačítka otevřete dialog expertního nastavení, ve kterém můžete provést podrobnější nastavení. Popis tohoto dialogu najdete níže.

Po provedení základního nastavení je síť připravená pro připojení do WLAN.

---

## Důležité: Bezpečnost v bezdrátových sítích

Ujistěte se, že svou síť chráníte některých ověřovacím a šifrovacím mechanismem. Nešifrované WLAN připojení umožňuje třetím stranám zachytit vaše data. I slabá ochrana (WEP) je lepší než žádná. Více najdete v částech „Šifrování“ (strana 515) a „Bezpečnost“ (strana 519).

---

V závislosti na zvoleném režimu ověřování umožňuje YaST nastavení doladit. U režimu *Otevřený* nelze nic dalšího nastavit, jedná se o nešifrovaný provoz bez ověřování.

### WEP klíče

Nastavte typ vstupu klíče. Na výběr máte z *Passphrase*, *ASCII* nebo *Hexadecimal*. Kliknutím na *Vícenásobné klíče* můžete nastavit až čtyři klíče. Délka klíče může být *128 bitů* nebo *64 bitů*. Výchozí nastavení je *128 bitů*. Jeden ze čtyř klíčů v seznamu můžete označit a kliknutím na tlačítko *Nastavit jako výchozí* nastavit jako výchozí. Pokud žádný klíč jako výchozí nenastavíte, bude jako výchozí použit první vložený klíč v seznamu. Pokud výchozí klíč smažete, musíte jako výchozí označit jiný klíč. Kliknutím na tlačítko *Upravit* lze měnit již existující klíče nebo vytvářet nové. V dialogu úpravy budete mít k dispozici všechny typy zadání klíče (*Passphrase*, *ASCII* nebo *Hexadecimal*). Při výběru *Passphrase* zadejte slovo nebo řetězec znaků, ze kterých se má klíč vytvořit. U *ASCII* je vyžadováno zadání pěti znaků pro 64 bitový klíč, 13 znaků pro 64 bitový nebo 26 znaků pro 128 bitový. U *Hexadecimal* zadejte deset znaků pro 64 bitový klíč nebo 26 pro 128 bitový.

### WPA-PSK

Pro WPA-PSK klíč zvolte vstupní metodu *Passphrase* nebo *Hexadecimal*. U režimu *Passphrase* zadejte 8 až 63 znaků, u režimu *Hexadecimal* 64 znaků.

### WPA-EAP

Zadejte své přihlašovací údaje. U TLS zadejte *Certifikát klienta* a *Certifikát serveru*. TTLS a PEAP vyžadují *Identitu* a *Heslo*. *Certifikát serveru* je volitelný. YaST hledá certifikáty v adresáři `/etc/cert`, uložte proto své certifikáty zde a omezte přístupová práva k souborům na 0600 (čtení a zápis pouze pro vlastníka).

Základní nastavení opustíte kliknutím na *Expertní nastavení*. Volby expertního nastavení jsou následující:

### Kanál

Nastavení kanálu WLAN karty je nutné pouze v režimech *Ad-hoc* a *Master*. Ve *spravovaném* režimu karta dostupné kanály automaticky vyhledá. V *Master* režimu

nastavte, který kanál bude nabízet služby přístupového bodu. Výchozí nastavení je *Automatický*.

#### Přenosová rychlost

Podle výkonnosti vaší sítě můžete nastavit přenosovou rychlost mezi body. Ve výchozím nastavení *Auto* se systém pokusí použít nejvyšší možnou rychlost. Některé WLAN karty změnu přenosové rychlosti nepodporují.

#### Přístupový bod

V prostředí s více přístupovými body lze jeden zvolit zadáním MAC adresy.

#### Použití správy napájení

Pokud jste na cestách, je zvýšíte výdrž baterií použitím správy napájení. Více informací o správě napájení najdete v kapitole 33 – „*Správa napájení*“ (strana 487).

## 34.1.4 Dostupné programy

hostap (balíček `hostap`) je používán k nastavení WLAN karty jako přístupového bodu. Více informací o tomto programu najdete na domovské stránce jeho projektu (<http://hostap.epitest.fi/>).

kismet (balíček `kismet`) je nástroj pro analýzu WLAN provozu. Tento nástroj vám může pomoci také při odhalování pokusů o průnik do sítě. Více informací najdete na stránce <http://www.kismetwireless.net/> a v manuálové stránce.

## 34.1.5 Tipy a triky nastavení WLAN

Při nastavování bezdrátové sítě se vám může hodit některý z následujících tipů:

### Stabilita a rychlost

Výkon a rychlost bezdrátové sítě závisí na čistotě signálu. překážky jako např. zdi výrazně snižují kvalitu signálu. Se slábnutím signálu se snižuje přenosová rychlost. Sílu signálu můžete překontrolovat pomocí nástroje `iwconfig` na příkazové řádce nebo pomocí `kwifimanager` v prostředí KDE. Pokud máte s kvalitou signálu problémy, proveďte nastavení na jiné zařízení nebo se pokuste nasměrovat anténu vašeho přístupového bodu. Přídavné antény lze připojit k řadě PCMCIA WLAN karet. Přenosová rychlost specifi-

kovaná výrobcem (např. 54 Mb/s) je maximální teoretická hodnota. V praxi obvykle získáte něco přes polovinu této hodnoty.

## Bezpečnost

Pokud nastavujete bezdrátovou síť, uvědomte si, že každý v dosahu vysílání může, pokud nepoužíváte šifrování, bez problémů zachytit váš signál. Všechny karty a přístupové body podporují WEP šifrování. Tato metoda ochrany však není naprosto bezpečná a obsahuje možná slabá místa připravená pro potencionální útočníky. WEP je obvykle dostatečná metoda ochrany pro běžné domácí používání. Mnohem bezpečnější je metoda WPA-PSK, která však není dostupná na přístupových bodech a routerech. Na některých zařízeních jí lze použít po updatu firmwaru, nicméně řada zařízení WPA v Linuxu vůbec nepodporuje. Během psaní tohoto článku bylo WPA možné používat pouze s kartami založenými na čípech Atheros nebo Prism2/2.5/3. WPA pracovalo pouze s ovladačem hostap (viz. „[Problémy s kartami Prism2](#)“ (strana 519)). Pokud není WPA k dispozici, je WEP lepší než žádné šifrování. V podnikové sféře s vysokými nároky na bezpečnost by bezdrátová síť měla používat WPA.

### 34.1.6 Možné problémy

Pokud WLAN karta neodpovídá, překontrolujte, zda máte potřebný firmware. Více o této problematice najdete v části [34.1.1 – „Hardware“](#) (strana 512).

### Více síťových zařízení

Moderní notebooky mívají síťovou i wlan kartu. Pokud obě zařízení nastavíte na DHCP (automatické přiřazení adresy), může dojít k problémům při přiřazování výchozí brány a resolvování jmen. Problém s resolvováním odhalíte snadno tak, že sice můžete poslat ping na adresu routeru, ale nemůžete brouzdat po internetu.

### Problémy s kartami Prism2

Pro zařízení s čipovou sadou Prism2 je k dispozici několik ovladačů. Kombinace různých ovladačů a různých karet vedou k různé kvalitě příjmu. WPA je dostupné pouze s ovladačem hostap. Pokud vaše karta nepracuje správně nebo chcete používat WPA, přečtěte si `/usr/share/doc/packages/wireless-tools/README.prism2`.

## WPA

Podpora WPA byla poprvé implementována v systému SUSE Linux. Protože je linuxová podpora WPA stále ve vývoji, podporuje YaST pouze nastavení WPA-PSK. S řadou karet WPA stále nepracuje. Některé karty potřebují pro podporu WPA update firmwaru. Pokud chcete WPA používat, prostudujte si `/usr/share/doc/packages/wireless-tools/README.wpa`.

### 34.1.7 Další informace

Řadu informací o bezdrátových sítích najdete na stránce Jeana Tourrilhes, který vytvořil linuxové nástroje pro práci s bezdrátovými sítěmi (*Wireless Tools*), na adrese [http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Wireless.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html).

## 34.2 Bluetooth

Bluetooth je technologie, která umožňuje propojovat různá zařízení jako mobilní telefony, PDA, notebooky nebo připojovat periférie (např. myši a klávesnice). Své jméno tato technologie získala podle dánského krále Haralda Modrozubého (Bluetooth). Logo Bluetooth je odvozeno od run pro písmena „H“ (podobá se hvězdě) a „B“.

Na rozdíl od IrDA není nutné, aby na sebe zařízení *viděla* a lze propojovat navzájem více zařízení. Pomocí této technologie je možné dosáhnout přenosové rychlosti 720 Kb/s (v aktuální verzi 1.2). Čistě teoreticky lze tento způsob připojení používat i v případě takových překážek, jakou je zeď. V praxi samozřejmě záleží na tloušťce a materiálu, ze kterého je zeď postavena, a třídě zařízení. Maximální dosah této technologie je podle třídy 10 až 100 metrů.

### 34.2.1 Základy

V následující části najdete informace o principech Bluetooth. Seznámíte se s potřebným softwarem a způsobem komunikace Bluetooth rozhraní s vaším systémem a samozřejmě i s Bluetooth profily.



## Software

Abyste mohli využívat Bluetooth, potřebujete Bluetooth adaptér (nejčastěji je integrovaný přímo v zařízení), ovladač a *Bluetooth Protocol Stack*.

Linuxové jádro již základní podporu Bluetooth obsahuje. Jako *Protocol Stack* slouží Bluez systém. Balíčky potřebné k používání Bluetooth:

- `bluez-libs`
- `bluez-bluefw`
- `bluez-pan`
- `bluez-sdp`
- `bluez-utils`

## Základní informace

Systém Bluetooth se skládá ze čtyř propojených vrstev:

### Hardware

Adaptér a příslušný ovladač v linuxovém jádře.

### Konfigurační soubory

Používané pro nastavení Bluetooth systému.

### Démoni

Služby nastavené v konfiguračním souborech a poskytující služby.

### Aplikace

Aplikace využívající služby démonů a ovládané uživateli.

Po vložení Bluetooth adaptéru systém hotplug zavede odpovídající ovladač. Po zavedení ovladače systém překontroluje konfigurační soubory, zda může Bluetooth spustit. Pokud ano, dojde ke spuštění služby a s ní spojených démonů. Z bezpečnostních důvodů je ve výchozím nastavení služba Bluetooth vypnuta.

## Profily

V Bluetooth jsou služby definovány pomocí profilů jako např.. transportní profil nebo základní tiskový profil. Aby zařízení mohlo používat službu jiného zařízení, musí rozumět stejnému profilu — informace, která často chybí v balíčku zařízení a v manuálu. někteří výrobci se však nedrží definic profilů, což vede k tomu, že je komunikace mezi jednotlivými zařízeními často velmi problematická.

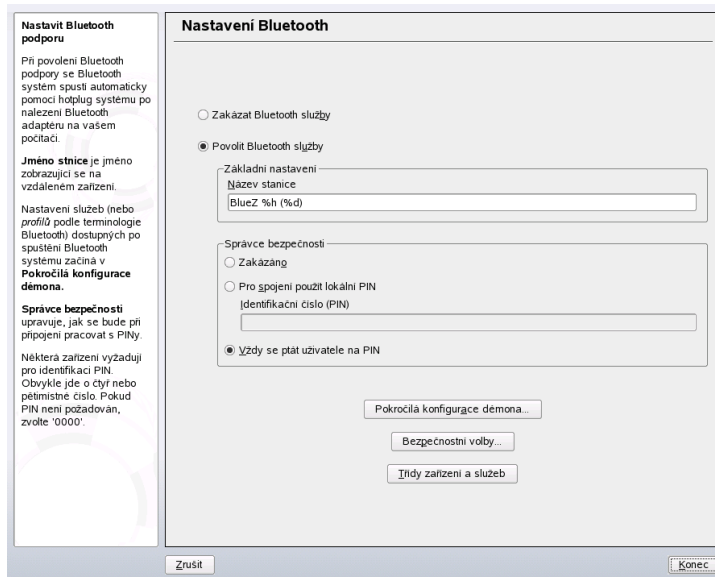
## 34.2.2 Nastavení

V následující části se dozvíte, jak nastavit Bluetooth na vašem počítači.

### Nastavení Bluetooth pomocí programu YaST

Podporu Bluetooth nastavíte pomocí Bluetooth modulu programu YaST viz. obr. 34.2 – „YaST konfigurace Bluetooth“ (strana 522). Pokud je pak systémem hotplug detekován Bluetooth adaptér, je Bluetooth automaticky spuštěn s nastaveními provedenými v tomto modulu.

**Obrázek 34.2** *YaST konfigurace Bluetooth*



První krok nastavení v programu YaST představuje povolení spuštění služby Bluetooth. Po povolení služby Bluetooth na nutné provést dvě nastavení. První nastavení se týká položky *Jméno stanice*. Jde o jméno, které se zobrazí při připojení k systému (počítači) na zařízení. Použít můžete dvě proměnné—%h pro jméno počítače (užitečné, např. pokud je jméno přiřazováno dynamicky přes DHCP) a %d vracející číslo rozhraní (užitečné, pokud připojujete více Bluetooth zařízení najednou). Pokud do tohoto pole nastavíte `Laptop %h` a přes DHCP získá stanice jméno `unit123`, budou k počítači všechna vzdálená zařízení přistupovat jako k `Laptop unit123`.

Další povinnou částí je *Správce bezpečnosti*, kde nastavujete chování svého systému při připojení vzdáleného zařízení. Správce bezpečnosti můžete zakázat, povolit lokální PIN nebo vyžadovat PIN vždy. Rozdíl mezi posledními dvěma položkami spočívá v tom, že v prvním případě se systém sice na PIN zeptá, ale pokud zařízení PIN neodešle nebo jen PIN chybný, spojení se přesto uskuteční. V druhém případě je PIN vyžadován vždy a spojení bez PINu nebo se zadáním chybného PINu se neprovedou. Z bezpečnostních důvodů vám doporučujeme použít třetí možnost, která navíc umožňuje používat pro různá zařízení různé PINy.

Dostupné služby (v Bluetooth nazývané *profily*) nastavíte v dialogu *Pokročilá konfigurace démona*. Služby lze povolit kliknutím na tlačítko *Povolit* a zakázat kliknutím na tlačítko *Zakázat*. V případě potřeby přenastavení služby jí upravíte jejím výběrem ze seznamu a kliknutím na tlačítko *Upravit*. Pokud nejste se službou blíže seznámeni, nemějte nastavení. Po provedení všech nastavení ukončete dialog kliknutím na tlačítko *OK*.

Dialog bezpečnostních nastavení, kde můžete nastavit šifrování, ověřování a nastavení skenování, vyvoláte v hlavním dialogu kliknutím na tlačítko *Bezpečnostní volby*. Zpět do hlavního dialogu se po nastavení vrátíte kliknutím na tlačítko *OK*. Všechna nastavení aktivujete kliknutím na tlačítko *Konec*.

Z hlavního dialogu je dostupný také dialog *Zařízení a třídy služeb*. Bluetooth zařízení jsou rozdělena do různých „tříd zařízení“. Zvolte pro svůj počítač správné zařazení jako „pracovní stanice“ nebo „notebook“. Nastavení třídy zařízení není tak důležité jako nastavení „třídy služeb“. Některá zařízení, např. mobilní telefony, totiž při špatně zvolené třídě služeb neumožňují využít všechny služby. Zvolit můžete několi tříd zároveň. Obvykle není vhodné předvolit všechny třídy současně. Ve většině případů je dostačující výchozí nastavení.

Pokud chcete nastavit síť, aktivujte v nabídce *Pokročilá konfigurace démona* profil *PAND* a nastavte režim služby pomocí tlačítka *Upravit*. Aby byla síť funkční, je nutné,

aby na jednom počítači byl profil pand nastaven na *naslouchací* režim a na druhém počítači na *vyhledávací*. Výchozí nastavení je *Listen*. Upravte pand podle své potřeby. Dále nastavte rozhraní bnePX (X je číslo pořadí zařízení v systému) v modulu *Síťová zařízení* → *Síťová karta*.

## Ruční konfigurace

Konfigurační soubory jednotlivých komponent Bluez systému se nacházejí v adresáři `/etc/bluetooth`. Výjimku představuje soubor `/etc/sysconfig/bluetooth` s nastaveními pro start komponent, který je upravován programem YaST module.

Konfiguraci popsanou v následujícím odstavci můžete provádět pouze jako uživatel `root`. V současné době zatím neexistuje žádný grafický konfigurační nástroj. Veškerá nastavení se provádějí pomocí editace textových souborů.

Při prvním spojení se nabídne zabezpečení pomocí PIN. PIN je číslo, které slouží např. u mobilních telefonů jako základní ochrana před nepovolanou manipulací s telefonem. Abyste mohli ovládat dva přístroje současně, musí mít oba stejný PIN. Na straně počítače PIN nastavíte v souboru `/etc/bluetooth/pin`. Bez ohledu na nainstalovaný počet externích zařízení umí Linux v současné době pracovat pouze s jedním PINem. Ovládání několika zařízení s různými PINy najednou není v současné době podporováno. Pokud tedy chcete ovládat více zařízení najednou, musí tato zařízení mít všechna stejný PIN, nebo vypnete ověřování pomocí čísla PIN.

---

### Důležité: Bezpečnost Bluetooth spojení

Bez ohledu na to, zda používáte PIN nebo ne, není spojení pomocí Bluetooth naprosto bezpečné!

---

V konfiguračním souboru `/etc/bluetooth/hcid.conf` lze provést řadu různých nastavení (např. jméno zařízení nebo režim bezpečnosti). Výchozí nastavení však obvykle není nutné měnit. Soubor obsahuje také popis jednotlivých voleb.

Aktivaci Bluetooth provedete v souboru `/etc/bluetooth/hcid.conf`. Zde můžete také změnit různá nastavení jako jméno zařízení či bezpečnostní režim. Soubor obsahuje u každé proměnné vysvětlující komentář.

Důležitou proměnnou je `security auto`. Pomocí této proměnné nastavujete použití PINu. V případě problémů se u tohoto nastavení použití PINu samo vypne. Pokud nechcete PIN používat vůbec, nastavte proměnnou na `none`. Z bezpečnostních důvodů

by výchozí nastavení mělo být `user`. Uživatel pak bude při každém připojení požádán o PIN.

Zajímavé jsou také proměnné vázající se k zařízení. Pomocí těchto proměnných můžete zadat, pod jakým jménem bude zařízení připojeno k počítači. Dále jsou zde definována také jednotlivé třídy jako `notebook`, `server` atd. včetně ověřování a připojení.

## 34.2.3 Systémové komponenty a programy pro práci s Bluetooth

Bluetooth je možné používat pouze ve spojení s různými službami. Ke spuštění potřebujete minimálně dva démony:

`hcid` (*Host Controller Interface*)

-- k vytvoření a rušení spojení.

`sdpd` (*Service Discovery Protocol*)

-- k zjištění dostupných služeb.

Pokud nejsou démoni spuštěni automaticky při startu systému, lze je oba aktivovat příkazem `rcbluetooth start`. Tento příkaz musí být vykonán s právy uživatele `root`.

Následující text obsahuje stručný popis nejdůležitějších příkazů pro práci s Bluetooth. Ačkoliv je v současnosti pro ovládání Bluetooth dostupná řada grafických programů, může se vám znalost programů příkazové řádky hodit.

Některé příkazy lze vykonat pouze jako uživatel `root`. Jde například o příkaz `l2ping <adresa_zarizeni>`, kterou se testuje připojení vzdáleného zařízení.

### hcitool

Prostřednictvím `hcitool` lze jednoduše určit, zda jde o lokální nebo vzdálené zařízení. Zařízení zobrazíte příkazem:

```
hcitool dev
```

Příkaz vypíše na každou řádku jedno zařízení ve formátu `JmenoRozhrani AdresaZarizeni`.

Příkazem `hcitool AdresaZarizeni` zjistíte jméno zařízení vzdáleného zařízení. Může jít například o další počítač, který má potřebné informace o třídě a jménu zařízení uložené v `/etc/bluetooth/hcid.conf`. V případě lokálních zařízení vám tento příkaz vrátí chybové hlášení.

Vzdálené zařízení se vyhledává pomocí příkazu `hcitool inq`. U každého zařízení získáte tři údaje: adresu zařízení, offset hodin a třídu zařízení. Adresa je důležitá, protože ji ostatní příkazy používají pro identifikaci cílového zařízení. Offset hodin slouží pouze k technickým účelům. Třída určuje typ zařízení a typ služby ve formě hexadecimálního čísla.

Příkaz `hcitool jmeno<adresa-zarizeni>` se používá k určení jména vzdáleného zařízení. V případě vzdáleného počítače je jméno stejné s informacemi v `/etc/bluetooth/hcid.conf`. Zadání lokální adresy povede k chybě výstupu.

## hciconfig

Příkazem `/usr/sbin/hciconfig` získáte informace o lokálních zařízeních. Bez argumentů příkaz zobrazí informace o zařízení jako jméno (`hciX`), fyzickou adresu (dvanácti místné číslo ve formátu `00:12:34:56:78`) a informace o přenesených datech.

`hciconfig hci0 jmeno` zobrazí jméno vrácené systémem po dotazu na vzdálené zařízení. Změnu nastavení lze provést s pomocí údajů získaných příkazem `hciconfig`. například `hciconfig hci0 name TEST` nastaví jméno na `TEST`.

## sdptool

Informace o tom, jaká služba je pro určité zařízení dostupná, získáte pomocí `sdptool`.

Příkaz `sdptool browse AdresaZarizeni` předá všechny služby jednomu zařízení se zadanou adresou.

Naproti tomu příkaz `sdptool search Sluzba` vyhledá jednu určitou službu.

Příkaz se dotáže na všechna dostupná zařízení a vypíše jejich služby spolu s krátkým popisem těchto služeb. Seznam všech dostupných služeb získáte zadáním příkazu `sdptool` bez parametrů.

## 34.2.4 Grafické aplikace

V prohlížeči Konqueror získáte seznam lokálních a vzdálených Bluetooth zařízení zadáním URL `sdp: /`. Dvojklikem na zařízení zobrazíte informace o zařízení. Pokud na zařízení umístíte kurzor, zobrazí se na stavové liště prohlížeče informace o profilu služby. Kliknutím na službu vyvoláte dialog nabízející uložení, použití služby (zařízení musí být aktivováno) nebo zrušení akce. Pokud nechcete, aby se dialog příště opět objevil a došlo přímo k vykonání služby, zatrhněte nabídku příště dialog nezobrazovat. Některá zařízení nejsou doposud podporována. Jiná vyžadují doinstalování dodatečných balíčků.

## 34.2.5 Příklady

Abyste si udělali přehled, co všechno je možné s Bluetooth dělat, připravili jsme pro vás několik příkladů.

### Propojení počítačů R1 a R2

V prvním příkladě si ukážeme, jak se nastavuje připojení mezi dvěma počítači. Potřeba k tomu budeme *pand* (*Personal Area Networking*). Všechny příkazy z tohoto příkladu je nutné zadávat jako uživatel `root`. K nastavení síťového připojení bude potřebný také příkaz (`ip`).

Na jednom z počítačů spustíte *pand* (v našem případě označen jako R1) příkazem:

```
pand -s
```

Na druhém počítači R2 získáte adresu pomocí příkazu:

```
hcitool ing
```

Spojení pak navážete zadáním příkazu:

```
pand -c AdresaZarizeni
```

Zjistíte jaké zařízení systém nastavil pro připojení příkazem:

```
ip link show
```

získáte výstup v následujícím formátu:

```
bnep0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
    link/ether 00:12:34:56:89:90 brd ff:ff:ff:ff:ff:ff
```

Zařízení bnep0 byste měli přiřadit IP adresu.

To uděláte např. pomocí následujících příkazů (na R1):

```
ip addr add 192.168.1.3/24 dev bnep0
ip link set bnep0 up
```

a na R2:

```
ip addr add 192.168.1.4/24 dev bnep0
ip link set bnep0 up
```

R1 je z R2 viditelný na adrese IP 192.168.1.3. Na počítač R2 se z počítače R1 můžete přihlásit příkazem:

```
ssh 192.168.1.4.
```

Příkaz ssh bude fungovat i pod normálním uživatelem.

## Datový transfer z mobilního telefonu na počítač

V dalším příkladě se ukážeme, jak překopírovat obrázek z fotoaparátu mobilního telefonu (bez dodatečných nákladů např. za MMS) na disk počítače. Prosím uvědomte si, že každý typ telefonu má jinou strukturu nabídky, ale v základech je postup podobný na všech typech telefonů. Aby bylo možné z telefonu na počítač přistupovat, na počítači musí být aktivována služba Obex-Push. O to se stará démon `opd` z `bluez-utils`. Službu spustíte příkazem:

```
opd --mode OBEX --channel 10 --daemonize --path /tmp --sdp
```

Důležité jsou zde dva parametry. Parametr `--sdp` aktivuje `sdpd`. Parametr `--path /tmp` říká, kam budou data ukládána, v našem příkladu do adresáře `/tmp`. Samozřejmě si můžete zvolit jiný adresář, do kterého máte práva zápisu.

Nyní je potřebné spustit na telefonu Bluetooth připojení. Postup najdete v manuálu vašeho mobilního telefonu. Nezapomeňte nastavit na počítači v souboru `/etc/bluetooth/pin` PIN. Po úspěšném připojení pošlete pomocí Bluetooth obrázky na počítač. Postup zasílání obrázků najdete opět v manuálu mobilního telefonu. Mimo obrázků můžete samozřejmě přenášet také např. hudební soubory.



## 34.2.6 Řešení možných problémů

Pokud máte s nastavením Bluetooth problémy, projděte nejdřív následující seznam postupů. pamatujte, že k chybě může docházet jak na straně počítače, tak na straně zařízení. Pokud máte možnost, otestujte funkčnost zařízení s jiným adaptérem.

Je ve výstupu příkazu `hcitool dev` uvedeno lokální zařízení?

Pokud ve výstupu není lokální zařízení uvedeno, nespustil se `hcid` nebo nebylo rozpoznáno Bluetooth zařízení. Příčin může být vícero, zařízení může být porouchané nebo chybí správný ovladač. Notebooky s integrovaným Bluetooth adaptérem mají často pro bezdrátová zařízení vypínač. Zda je nutné zařízení nejdřív fyzicky zapnout zjistíte v manuálu svého notebooku. Restartujte Bluetooth příkazem `rcbluetooth restart` a podívejte se do souboru `/var/log/messages`, zda systém nevypisuje chyby.

Nepotřebuje Bluetooth adaptér soubor s firmwarem?

Pokud ano, nainstalujte balíček `bluez-bluefw` a restartujte Bluetooth příkazem `rcbluetooth restart`.

Vrací příkaz `hcitool inq` jiná zařízení?

Proveďte tento test vícrát než jednou. Může docházet k interferenci s jiným zařízením používajícím stejnou frekvenci.

Souhlasí PIN?

Překontrolujte, zda zadaný PIN (v souboru `/etc/bluetooth/pin`) souhlasí se zařízením.

„Vidí“ vzdálené zařízení počítač?

Pokuste se navázat spojení ze vzdáleného zařízení. Překontrolujte, zda zařízení vidí počítač.

Nezdaří se síťové propojení počítačů z příkladu 1. (viz „[Propojení počítačů R1 a R2](#)“ (strana 527))

Příčin může být několik. Jedním může být skutečnost, že jeden nebo oba počítače nerozumí protokolu SSH. Otestujte, zda na sebe počítače vidí příkazy:

```
ping 192.168.1.3
```

a

```
ping 192.168.1.4
```

Pokud proběhnou příkazy bez problémů, ujistěte se, že běží sshd.

Další příčina může spočívat v tom, že jste nastavili jiné adresy, než jsou uvedeny v příkladu nebo jste pro oba počítače nastavili stejnou IP adresu. Změňte IP adresy.

Nedošlo k rozpoznání počítače jako cíle z propojení počítače a mobilního telefonu z příkladu 2.

Ujistěte se, že mobil rozpoznal službu Obex-Push na počítači. V nabídce mobilu je obvykle pro takové účely položka, která zobrazuje dostupné služby. Návod najdete v manuálu svého mobilního telefonu. Pokud není služba Obex-Push zobrazena, je problém na straně počítače u programu opd. Ujistěte se, že je opd spuštěn a že máte práva zápisu do zadaného adresáře.

Je možné kopírovat také z počítače na mobilní telefon?

Ano, kopírování je možné, pokud nainstalujete program obexftp a použijete příkaz:

```
obexftp -b AdresaZarizeni -B 10 -p Obrazek.
```

Tento postup byl testován na telefonech Siemens a Sony Ericsson a u jiných typů nemusí být funkční.

## 34.2.7 Další informace

Obsáhlý přehled různých návodů na používání a nastavení Bluetooth najdete na stránce <http://www.holtmann.org/linux/bluetooth/>.

- Oficiální howto integrace Bluetooth protokolu do jádra: <http://bluez.sourceforge.net/howto/index.html>
- Připojení k PDA PalmOS: <http://www.cs.ucl.ac.uk/staff/s.zachariadis/btpalmlinux.html>

## 34.3 IrDA — Infrared Data Association

IrDA (Infrared Data Association) je průmyslový standard pro bezdrátovou komunikaci v infračerveném spektru. Řada dnešních laptopů obsahuje IrDA kompatibilní vysílač

a přijímač, umožňující spojení s dalšími zařízeními, jako jsou tiskárny, modemy, LAN nebo jiné laptopy. Přenosová rychlost sahá od 2400 b/s až do 4 Mb/s.

IrDA má dva operační režimy. Standardní režim, SIR, přistupuje k zařízení přes sériové rozhraní. Tento režim pracuje na naprosté většině systémů a je dostačující pro většinu požadavků. Rychlejší režim, FIR, vyžaduje pro IrDA čip zvláštní ovladač. Z důvodů neexistence ovladače nejsou ve FIR režimu podporovány všechny čipy. Režim nastavíte v BIOSu svého počítače. V BIOSu také zjistíte, které sériové zařízení bude v SIR režimu používáno.

Informace o IrDA najdete v IrDA HOWTO Wernera Heusera na stránce <http://tuxmobil.org/Infrared-HOWTO/Infrared-HOWTO.html>. Další odkazy jsou dostupné na stránkách Linux IrDA projektu <http://irda.sourceforge.net/>.

## 34.3.1 Software

Všechny potřebné moduly jsou již obsaženy v jádře. Nezbytné aplikace pro podporu infračerveného portu a protokolu IrDA jsou součástí balíčku `irda`. Po instalaci balíku naleznete dokumentaci v souboru `/usr/share/doc/packages/irda/README`.

## 34.3.2 Konfigurace

IrDA systém se automaticky nespouští při startu systému. Ke změně tohoto nastavení použijte editor úrovní běhu v programu YaST, případně příkaz `chkconfig`. Každých několik sekund vysílá IrDA "průzkumný paket", kterým vyhledává periferní zařízení ve svém okolí. Tento proces je náročný na spotřebu energie a snižuje výdrž baterií. Z tohoto důvodu je ve výchozím nastavení podpora IrDA vypnuta a měla by být spouštěna pouze v případě potřeby. Ručně ji spustíte příkazem `rcirda start` a vypnete příkazem `rcirda stop`. Všechny potřebné moduly se zavedou automaticky.

Soubor `/etc/sysconfig/irda` obsahuje pouze jedinou proměnnou `IRDA_PORT`, pomocí které je nastaveno zařízení rozhraní v SIR režimu. Tuto proměnnou nastavuje skript `/etc/irda/drivers`.

## 34.3.3 Použití

K tisku přes infračervený port pošlete data do souboru zařízení `/dev/ir1pt0`. Tento soubor se chová stejně jako normální tiskový port `/dev/lp0`, jediný rozdíl je bezdrátový přenos.

Tiskárnu na tomto portu můžete konfigurovat pomocí YaST stejně jako na paralelním nebo sériovém portu. Při tisku dbejte na to, aby byla vždy zachována přímá viditelnost mezi počítačem a tiskárnou a aby byla aktivována podpora IrDA.

Pro komunikaci s jinými počítači, mobilními telefony a dalšími zařízeními použijte soubor zařízení `/dev/ircomm0`. Například s mobilním telefonem Siemens S25 můžete použít program `wdial` a mít tak bezdrátové spojení na Internet.

Bez dalších nastavení lze přistupovat pouze k zařízením podporující tiskový nebo IrCOMM protokol. Zařízení s podporou protokolu IROBEX (např. 3Com Palm Pilot) vyžadují zvláštní aplikace jako `irobexpalm` a `irobexreceive`. Více informací o této problematice najdete v IR-HOWTO. Podporovaný protokol zařízení najdete ve výstupu příkazu `irdadump` v závorkách za jménem příslušného zařízení. Podpora IrLAN protokolu je stále ve vývoji a není stabilní.

## 34.3.4 Možné potíže

Pokud zařízení nereagují na IrDA, přihlaste se jako `root` a příkazem `irdadump` se přesvědčte, zda váš počítač zařízení rozpoznal:

```
irdadump
```

V případě tiskárny Canon BJC-80 v dosahu počítače se objeví v pravidelných intervalech zprávy, které ukazují výstup na obrazovku:

```
21:41:38.435239 xid:cmd 5b62bed5 > ffffffff S=6 s=0 (14)
21:41:38.525167 xid:cmd 5b62bed5 > ffffffff S=6 s=1 (14)
21:41:38.615159 xid:cmd 5b62bed5 > ffffffff S=6 s=2 (14)
21:41:38.705178 xid:cmd 5b62bed5 > ffffffff S=6 s=3 (14)
21:41:38.795198 xid:cmd 5b62bed5 > ffffffff S=6 s=4 (14)
21:41:38.885163 xid:cmd 5b62bed5 > ffffffff S=6 s=5 (14)
21:41:38.965133 xid:rsp 5b62bed5 < 6cac38dc S=6 s=5 BJC-80
                    hint=8804 [ Printer IrCOMM ] (23)
21:41:38.975176 xid:cmd 5b62bed5 > ffffffff S=6 s=* erde
                    hint=0500 [ PnP Computer ] (21)
```

Pokud se výstup neobjeví nebo zařízení neodpovídá, prověřte konfiguraci IrDA. Používáte správný port? Někdy se infraport najde jako `/dev/ttyS2` nebo `/dev/ttyS3` nebo je použito jiné přerušení než 3, což se většinou dá nastavit v BIOSu konfigurovaného notebooku.

Dále je důležité si uvědomit, že IrDA komunikuje pouze se zařízeními, podporujícími protokoly `Printer` nebo `IrCOMM`. Na podporu protokolu `IROBEX` potřebujete ještě programy `irobex_palm3` a `irobex_receive` a pak můžete komunikovat například s 3Com Palm Pilot. Všechny protokoly podporované zařízením se zobrazí ve výstupu z příkazu `irdadump` za jménem zařízení v hranatých závorkách. Podpora protokolu `IrLAN` je zatím ve vývoji a očekává se v budoucích verzích Linuxu.

Pokud potřebujete zkontrolovat, zda IrDA port vysílá infračervené záření, můžete k tomu použít některou z běžných videokamer, které bývají na rozdíl od lidských očí citlivé i v infračervené oblasti.



# Rejstřík

## Symboly

- 64-bitový Linux, 143
  - podpora běhu aplikací, 143
  - specifikace jádra, 146
  - vývoj softwaru, 144
- úroveň běhu (Viz runlevel)
- šifrování
  - oddíly, 95
  - soubory, 95

## A

- ACLs, 109–119
  - definice, 110
  - podpora, 119
  - používání, 110
  - přístupové bity, 112
- ACLs
  - kontrolní algoritmus, 118
  - masky, 114
  - přístup, 113
  - struktura, 110
  - výchozí, 116
- adresa
  - IP, 280
  - MAC, 280
- aktualizace, 59–63
  - passwd a group, 60
  - YaST, 60
  - zvukové směšovače, 71
  - zálohování, 59
- Apache, 391–410
  - apxs, 402
  - bezpečnost, 406–407
  - CGI, 400
  - content negotiation, 401
  - instalace, 391

- konfigurace, 392
  - ruční, 393
  - soubory, 393
- logování, 399
- moduly, 401
  - mod\_perl, 403
  - mod\_php4, 404
  - mod\_python, 405
  - mod\_ruby, 405
- problémy, 407
- práva, 406
- Squid, 457
- SSI, 400
- virtuální servery, 396, 401

- aplikace
  - síťové
    - vzdáleně, 253
  - vzdáleně
    - FreeNX, 253
- autentizace
  - PAM, 259–266

## B

- Bash
  - .bashrc, 183
  - .profile, 183
  - profil, 183
- bezpečnost, 97–108
  - chyby, 101, 104
  - deteke průlomu, 72
  - DNS, 105
  - firewall, 79
  - hesla, 99–100
  - hlášení problémů, 108
  - inženýrství, 98
  - lokální, 99–102
  - práva, 100–101
  - RPM podpisy, 107
  - Samba, 436

- Squid, 444
- SSH, 89–94
- startování, 99
- sériové terminály, 99
- síť, 102–105
- tcpd, 108
- tipy a triky, 106
- viry, 102
- X, 103
- útoky, 104–105
- červi, 105
- šifrovaný souborový systém, 471

**BIND**, 329–339

**Bluetooth**, 470, 520–530

- bluez, 521
- hciconfig, 526
- síť, 523
- YaST, 522

## **C**

**CardBus** (Viz hardware, CardBus)

**CJK**, 191

**cron**, 184

**cryptofs**, 95

**CVS**, 412, 419–421

## **D**

**DHCP**, 355–362

- balíčky, 358
- dhcpcd, 358–360
- konfigurace pomocí YaST, 356
- server, 358–360
- statické přiřazování adres, 360

**digitální fotoaparáty**, 471

**disk**

- hdparm, 496
- správa napájení, 496

**diskové oddíly**

- fdisk, 176

- tabulka diskových oddílů, 163
- šifrování, 95

**DNS**, 291

- bezpečnost, 105
- BIND**, 329–339
- domény, 310
- konfigurace, 323
- logování, 333
- Mail Exchanger, 292
- nameservery, 310
- NIC**, 292
- options, 332
- přeposílání, 330
- spouštění, 330
- squid, 448
- top level domain, 291
- volby, 332
- zóny, 334
- soubory, 335
- řešení problémů, 330

**Domain Name System** (Viz DNS)

**DOS**

- sdílení souborů, 431

**drift soubor**, 366

## **E**

**editor úrovní běhu**, 157

**editory**

- Emacs, 189–190

**Emacs**, 189–190

- .emacs, 189
- default.el, 189

**Evolution**, 472

## **F**

**FHS**

- SGML, 65
- XML, 65

**filtrování paketů** (Viz firewall)



- Firefox
  - příkaz otevření URL, 76
- firewall, 79
  - filtrování paketů, 79, 83
  - Squid, 455
  - SuSEfirewall2, 79, 84
- Firewire (IEEE1394)
  - disky, 471
- flash disky, 471
- FreeNX, 253–257

## G

- grafické karty
  - 3D, 249–252
    - instalační podpora, 251
    - ovladače, 249
    - podpora, 249
    - ovladače, 242
- grafika
  - 3D, 249–252
    - 3Ddiag, 251
    - diagnostika, 250
    - problémy, 251
    - SaX, 250
    - testování, 250
  - GLIDE, 249–252
  - OpenGL, 249–252
    - ovladače, 249
    - testování, 250
- GRUB, 163–181
  - /etc/grub.conf, 173
  - device.map, 166
  - GRUB Geom Error, 180
  - GRUB shell, 173
  - grub.conf, 166
  - heslo pro zavedení, 174
  - informace, 181
  - JFS a GRUB, 180
  - jména oddílů, 168

- jména zařízení, 168
- menu, 166
- menu.lst, 166
- odinstalace, 175
- omezení, 165
- parametry jádra, 171
- správa spouštění, 164
- start z kombinovaného IDE/SCSI systému, 180
- zástupné znaky, 171
- řešení problémů, 179

## H

- hardware
  - CardBus, 475
  - ISDN, 298
  - karta PCMCIA, 475

## I

- I18N, 191
- inetd, 61
- info stránky, 186
- init, 150–151
  - skripty, 153–157
  - vkládání skriptů, 156
- instalace
  - GRUB, 165
  - ruční, 71
- instalační podpora
  - 3D grafické karty, 251
- Internet
  - cinternet, 318
  - dial-up, 316–318
  - DSL, 301
  - ISDN, 298
  - kinternet, 318
  - qinternet, 318
  - smpppd, 316–318
  - T-DSL, 303

- webový server (Viz Apache)
- IP adresa, 280
  - třídy adres, 281
- IP adresy
  - dynamické přidělování, 355
  - IPv6
    - konfigurace, 290
    - maškaráda, 82
    - privátní, 282
  - IrDA, 470, 530–533
    - konfigurace, 531
    - spuštění, 531
    - zastavení, 531

## J

- jade (Viz SGML, openjade)
- jmenný server
  - DNS, 323
- jádro
  - 2.6, 63
  - cache, 188
  - moduly
    - síťové karty, 293
  - omezení, 232

## K

- karta PCMCIA, 475
- karty
  - grafické
    - ovladače, 242
  - síť, 293
- klávesnice
  - mapování, 190
    - kombinace kláves, 191
    - skládání, 191
  - rozložení, 190
  - X rozšíření klávesnice, 191
  - XKB, 191
- konfigurace, 159

- Apache, 392
- DNS, 323
- DSL, 301
- GRUB, 165
- IPv6, 290
- IrDA, 531
- ISDN, 298
- kabelového modemu, 300
- modemu, 295
- routing, 309
- Samba, 432–436
  - klient, 440
- směrování, 309
- Squid, 449
- SSH, 89
- síť, 293
  - manuální, 306–316
- T-DSL, 303
- tisk, 198–200
- konfigurační soubory, 308
  - .bashrc, 183, 187
  - .emacs, 189
  - .mailsync, 427
  - .profile, 183
  - .xsession, 93
  - /boot/GRUB/menu.lst, 166
  - /etc/grub.conf, 173
  - /etc/gshadow, 65
  - /etc/inittab, 150
  - /etc/powersave.conf, 69
  - acpi, 491
  - config, 309
  - crontab, 184
  - csh.cshrc, 193
  - dhclient.conf, 358
  - dhcp, 309
  - dhcpd.conf, 358
  - exports, 352, 354
  - group, 60
  - host.conf, 312

- alert, 312
  - multi, 312
  - nospoof, 312
  - order, 312
  - trim, 312
- HOSTNAME, 315
- hosts, 292, 311
- httpd.conf, 392
- ifcfg-\*, 309
- inittab, 150, 190
- inputrc, 190
- irda, 531
- jazyk, 191, 193
- kernel, 149
- logrotate.conf, 185
- named.conf, 329, 331–339, 448
- networks, 311
- nscd.conf, 315
- nsswitch.conf, 313, 383
- ntp.conf, 366
- pam\_unix2.conf, 383
- passwd, 60
- powersave, 490
- profil, 183
- profile, 187
- profily, 193
- práva, 106
- resolv.conf, 188, 310, 329, 447
- routes, 309
- samba, 436
- services, 436, 455
- slapd.conf, 374
- smb.conf, 431, 433
- smppd.conf, 317
- smpppd-c.conf, 318
- squid.conf, 447, 449, 452, 455, 458, 460
- squidguard.conf, 460
- sshd\_config, 94
- sysconfig, 159–160

- termcap, 190
- wireless, 309
- XF86Config (Viz konfigurační soubory, xorg.conf)
- xorg.conf, 72, 237
  - Device, 241
  - Monitor, 242
  - obrazovka, 240
- Kontakt, 472
- konzole
  - grafická
    - vypnutí, 179
  - počte, 190
  - přepínání, 190
- KPilot, 472
- KPowersave, 468
- KSysguard, 469
- kódování
  - UTF-8, 64
  - výchozí, 64

## L

- L10N, 191
- LAMP, 391
- LDAP, 369–390
  - ACL, 375
  - administrace
    - skupin, 388
    - uživatelů, 388
  - adresářový strom, 371
  - konfigurace serveru, 374
  - kontrola přístupu, 377
  - ldapadd, 379
  - ldapdelete, 382
  - ldapmodify, 381
  - ldapsearch, 382
  - mazání dat, 382
  - vkládání dat, 379
  - vyhledávání dat, 382

- YaST
  - moduly, 384
  - YaST LDAP klient, 382
  - úprava dat, 381
- LFS soubory
  - velikost, 231
- Lightweight Directory Access Protocol (Viz LDAP)
- LILO, 165
  - odinstalace, 175
- Linux
  - odinstalace, 175
  - sdílení souborů s jiným OS, 431
  - sítě, 277
- linuxrc
  - ruční instalace, 71
- locale
  - UTF-8, 64
- locate, 186
- logování, 333
  - logrotate, 185
  - nastavení, 185
- logrotate, 184
- logy
  - apache2, 399
  - boot.msg, 490
  - httpd, 399
  - Squid, 448, 450, 457
  - Unison, 419
  - X.org, 251
  - zprávy, 330
- LVM
  - YaST, 47

## M

- manuálové stránky, 186
- Master Boot Record (Viz MBR)
- maškaráda, 82
  - konfigurace pomocí SuSEfirewall2, 84

- MBR, 163
  - obnova, 176
- mobilita, 465–473
  - digitální fotoaparáty, 471
  - externí disky, 471
  - Firewire (IEEE1394), 471
  - kapesní počítače, 472
  - mobily, 472
  - notebooky, 465
  - ochrana dat, 471
  - PDA, 472
  - USB, 471
- mobily, 472
- modemy
  - kabelové, 300
  - YaST, 295
- monitorování systému, 468
  - KPowersave, 468
  - KSysguard, 469
- mountd, 352, 354

## N

- named, 330
- nameserver (Viz DNS)
  - BIND, 329
- nastavení
  - PAM, 72
  - Poweresave, 74
  - SSH, 89
  - sítě, 293
  - tisk, 198
- NAT (Viz maškaráda)
- NetBIOS, 432
- Network Information Service (Viz NIS)
- NetworkManager, 303
  - GNOME applet, 305
  - KDE applet, 304
- NFS, 349
  - export, 352

- export souborů, 350
- firewall, 346, 352
- import souborů, 350
- mount, 350
- oprávnění, 353
- připojení, 350
- server, 350
- nfsd, 352, 354
- NIS, 343–347
  - klient, 346
  - master, 343–346
  - slave, 343–346
- notebooky, 465–471, 475
  - hardware, 465
  - IrDA, 530–533
  - PCMCIA, 465
  - SCPM, 466, 477
  - SLP, 468
  - správa napájení, 466, 487–497
  - správa profilů, 477
- NSS, 313
  - databáze, 313
- nVidia, 61
- nápověda
  - info stránky, 186
  - manuálové stránky, 186
  - X11, 242

## O

- obrazovka
  - rozlišení, 241
- ochrana dat, 471
- odinstalace
  - GRUB, 175
  - LILO, 175
  - Linuxu, 175
- OpenSSH (Viz SSH)
- OS/2
  - sdílení souborů, 431

- ověřování
  - Kerberos, 71

## P

- PAM, 259–266
  - nastavení, 72
- paměť
  - RAM, 188
- parametry jádra, 170–171
- PATH, 74
- PCMCIA, 74, 465, 475
  - IrDA, 530–533
  - software, 476
- PCMCIA karty (Viz hardware, karta PCMCIA)
- PDA, 472
- portmap, 352
- porty
  - 53, 332
  - skenování, 457
- power management (Viz správa napájení)
- powersave, 497
  - konfigurace, 498
  - probuzení, 501
  - standby, 501
  - suspend, 501
  - uspání, 501
- pošta
  - soubory, 413
    - mailsync, 426–429
    - synchronizace, 469
- proměnné
  - PATH, 74
  - prostředí, 191
- protokolové soubory
  - Unison, 419
- protokoly
  - ICMP, 278
  - IGMP, 278

- IPv6, 283
  - LDAP, 369
  - SLP, 319
  - SMB, 432
  - TCP/IP, 277
  - UDP, 278
  - proxy, 443–444 (Viz Squid)
    - transparentní, 444, 454
    - výhody, 443
  - písma, 244
    - CID-keyed, 248
    - X11 core, 247
    - Xft, 244
  - připojovatelné autentizační moduly (Viz PAM)
  - příkazy
    - chown, 64
    - fdisk, 176
    - fonty-konfigurace, 243
    - free, 188
    - hciconfig, 526
    - hdparm, 496
    - head, 64
    - ldapadd, 380
    - ldapdelete, 382
    - ldapmodify, 381
    - ldapsearch, 382
    - lp, 204
    - nice, 64
    - rpmbuild, 62
    - scp, 91
    - sftp, 91
    - slptool, 320
    - smbpasswd, 437
    - sort, 64
    - ssh, 90
    - ssh-agent, 93
    - ssh-keygen, 93
    - su, 74
    - sx, 62
    - tail, 64
    - udev, 217
  - přístupová práva
    - ACLs, 109–119
    - přístupová práva k souborům, 186
    - Samba, 436
- ## R
- ### RAID
- softwarový, 53
  - YaST, 53
- ### reverzní převod, 338
- ### RFC, 277
- ### routing, 309–310
- ### routování (Viz směrování)
- ### RPC mount démon, 352
- ### RPC NFS démon, 352
- ### RPC portmapper, 352
- ### RPM
- bezpečnost, 107
  - verze 4, 62
  - vytváření, 62
- ### rsync, 413, 425
- ### runlevel, 151
- přechod, 151, 158
  - typy, 152
  - YaST, 157
  - změna, 152
- ## S
- ### Samba, 431–441
- bezpečnost, 436
  - instalace, 432
  - jména, 432
  - klient, 432, 440
  - konfigurace, 432–436
  - NetBIOS, 432
  - nápověda, 441
  - optimalizace, 440

- práva, 436
- přihlášení, 437
- přístupová práva, 436
- sdílení, 432, 434
- server, 432–436
- SMB, 432
- spuštění, 432
- swat, 436
- TCP/IP, 432
- tisk, 440
- tiskárny, 432
- ukončení, 432
- SCPM, 477
  - nastavení, 478
  - notebooky, 466
  - přepínání profilů, 480
  - spuštění, 479
  - zdroje, 479
- scripty
  - init.d
    - nfsserver, 352
    - portmap, 352
    - squid, 447
- security
  - startování, 100
- security level, 436
- server
  - CUPS, 206
  - LDAP, 369
  - NFS, 352
  - NIS, 343
  - proxy, 443
  - Samba, 431
  - souborový, 431
  - tiskový, 205
  - webový, 391
  - X, 235
- Service Location Protocol (Viz SLP)
- SGML
  - openjade, 62
- skripty
  - boot.udev, 221
  - init.d, 156, 315
    - network, 315
    - nfsserver, 316
    - portmap, 316
    - sendmail, 316
    - xinetd, 316
    - ybind, 316
    - ypserv, 316
  - irda, 531
  - mkinitrd, 149
  - modify\_resolvconf, 188, 310
  - SuSEconfig, 159–160
- SLP, 319, 468
  - Konqueror, 321
  - prohlížeč, 321
  - registrace služeb, 319
  - slptool, 320
- SMB (Viz Samba)
- směrování, 280, 309
  - maškaráda, 82
  - statické, 309
  - síťová maska, 281
- souborové systémy, 223–233
  - access control lists, 109–119
  - Ext2, 224–225
  - Ext3, 225–226
  - JFS, 228
  - limity, 231
  - podporované, 229–230
  - Reiser4, 227–228
  - ReiserFS v3, 226–227
  - termíny, 223
  - výběr, 224
  - XFS, 228–229
  - šifrování, 95
- soubory
  - hledání, 186
  - jádra, 187

- logy, 184
- synchronizace
  - CVS, 412, 419
  - mailsync, 413, 426
  - rsync, 413
  - subversion, 412
  - Unison, 412, 417
- velikost, 231–232
- větší než 2 GB, 231
- šifrování, 95
- spindown, 496
- správa
  - profilů, 477
- správa napájení, 466, 487–506
  - ACPI, 487, 490–496
  - APM, 487, 489
  - disk, 496
  - frekvence CPU, 497
  - powersave, 497
  - rychlost CPU, 497
  - YaST, 506
- Správce logických svazků (Viz LVM)
- Squid, 443
  - adresáře, 447
  - Apache, 457
  - bezpečnost, 444
  - cache, 444
    - poškozená, 448
    - velikost, 446
    - vícenásobná, 444
  - cachemgr.cgi, 457, 459
  - Calamaris, 460
  - CPU, 447
  - firewall, 455
  - konfigurace, 449
  - kontrola přístupu, 452, 458
  - logy, 448, 450, 457
  - odinstalování, 448
  - operační paměť, 446
  - pevný disk, 445
  - problémy, 448
  - proxy cache, 443
  - práva, 447, 452
  - RAM, 446
  - reporty, 460
  - spuštění, 447
  - squidGuard, 459
  - statistiky, 457, 459
  - stav objektů, 445
  - transparentní proxy, 454, 457
  - ukládání, 445
  - vlastnosti, 443
  - zastavení, 447
- SSH, 89–94
  - autentizační mechanismy, 93
  - démon, 91
  - páry klíčů, 92–93
  - scp, 91
  - server, 91
  - sftp, 91
  - ssh, 90
  - ssh-agent, 93–94
  - ssh-keygen, 93
  - sshd, 91
  - X, 94
- startovací disketa, 164
- startování, 147
  - CD, 164
  - disketa, 164
  - DOS, 165
  - grafické
    - vypnutí, 179
  - graphic, 179
  - GRUB, 165–181
  - initrd
    - vytváření, 149
  - správa, 164
  - USB, 164
  - Windows, 165
  - zavaděče, 164



- zaváděcí sektory, 163
- Subversion, 412, 422
- synchronizace
  - pošta, 413
  - soubory, 411–429
    - CVS, 412, 419–421
    - mailsync, 413, 426–429
    - rsync, 413
    - subversion, 412
    - Unison, 412, 417–419
- synchronizace dat, 470
  - e-mail, 469
  - Evolution, 472
  - Kontakt, 472
  - KPilot, 472
- synchronizace času, 363
  - konfigurace, 366
  - xntp, 363
- systém
  - aktualizace, 59–63
  - lokalizace, 191
  - využívání omezených zdrojů, 187
  - X Window (Viz X)
- systémy písma, 244
  - písma s kódováním CID, 248
  - písma X11 core, 247
  - Xft, 244
- sítě, 277 (Viz TCP/IP)
  - bezdrátové, 470
  - Bluetooth, 470, 523
  - DHCP, 355
  - DNS, 291
  - IP adresa, 280
  - IrDA, 470
  - konfigurace, 292–303, 306–316
    - IPv6, 290
  - konfigurační soubory, 308–315
  - localhost, 282
  - nastavení, 292
  - oznamovací adresa, 282

- reverzní převod, 338
- SLP, 319
- směrování, 280–281
- síťové masky, 281
- WLAN, 470
- YaST, 293
- základní síťová adresa, 282
- síťování, 277
- síťové adresy
  - IPv4, 280
  - IPv6, 283
  - překlad jmen, 291
- síťový souborový systém (Viz NFS)

## T

- TCP/IP, 277
  - ICMP, 278
  - IGMP, 278
  - pakety, 279
  - přenosový model, 278
  - TCP, 277
  - UDP, 278
- TEI XSL styly
  - umístění, 74
- telefonní ústředna, 299
- tisk, 195, 198–200
  - CUPS, 204
  - footmatic filtry, 62
  - fronty, 199
  - GDI tiskárny, 210
  - Ghostscript ovladač, 199
  - konfigurace pomocí YaST, 198
  - kprinter, 204
  - LPRng, 63
  - ovladače, 199
  - port, 199
  - PPD soubor, 199
  - připojení, 199
  - příkazová řádka, 204

- Samba, 432
- sít'
  - řešení problémů, 212
- testovací stránka, 199
- xpp, 204
- z aplikace, 204
- řešení problémů
  - sít', 212
- Tripwire
  - nahrazen AIDE, 72
- TrueType (Viz X, TrueType fonty)

## U

- udev, 217
  - automatizace, 218
  - klíče, 219
  - mass storage, 220
  - pravidla, 218
  - regulární výrazy, 219
  - startovací skript, 221
  - sysfs, 219
  - udevinfo, 219
  - YaST, 221
  - zástupné znaky, 219
- ulimit, 187
  - nastavení, 187
- USB
  - disky, 471
  - flash disky, 471
- UTF-8, 64
- uzly zařízení
  - udev, 217
- uživatelé
  - /etc/passwd, 262, 383

## V

- vstupní metody
  - CJK, 191
- vzdálená práce

- FreeNX, 253–257

## W

- webový server
  - Apache (Viz Apache)
- whois, 292
- Windows
  - sdílení souborů, 431
- WLAN, 470

## X

- X, 235
  - bezpečnost, 103
  - fonty, 243
  - fonty TrueType, 243
  - nápověda, 242
  - optimalizace, 237–243
  - ovladače, 242
  - písma s kódováním CID, 248
  - písma X11 core, 247
  - SaX2, 238
  - SSH, 94
  - systémy písem, 244
  - virtuální obrazovka, 241
  - xf86config, 238
  - xft, 243
  - Xft, 244
  - znakové sady, 243
- X rozšíření klávesnice (Viz klávesnice, X rozšíření klávesnice)
- X11 (Viz X)
- Xen, 267
  - přehled, 267
- Xft, 244
- xinetd, 61
- XKB (Viz klávesnice, X rozšíření klávesnice)
- XML
  - Katalog, 63

- openjade, 62
- xorg.conf
  - barevná hloubka, 241
  - Cesty k fontům, 238
  - Depth, 240
  - Display, 240
  - Modeline, 241
  - Modes, 241
  - Monitor, 238, 240
  - parametry zobrazení, 238
  - sekce Device, 240
  - sekce InputDevice, 238
  - Sekce Modes, 239
  - sekce ServerFlags, 238

## Y

### YaST

- 3D, 249
- aktualizace, 60
- Bluetooth, 522
- DHCP, 356
- DSL, 301
- Editor úrovní běhu, 157
- ISDN, 298
- kabelový modem, 300
- LDAP klient, 382
- LVM, 47
- modem, 295
- NIS klient, 346
- RAID, 53
- Samba
  - klient, 440
- SLP prohlížeč, 321
- správa napájení, 506
- sysconfig editor, 160
- síťová karta, 293
- T-DSL, 303
- tisk, 198–200
- update, 60

YP (Viz NIS)

## Z

- zavaděče, 163
  - GRUB, 165
  - LILO, 165
- zavádění systému
  - MBR, 163
- zvukové směšovače, 71
- zálohování
  - aktualizace, 59
- záznamy
  - Unison, 419
  - zprávy, 89
- zóny, 334

