

Gentoo Handbook

Tomáš Vondra

Gentoo Handbook

Tomáš Vondra

Chief Architect: Daniel Robbins

Author: Sven Vermeulen, Chris Houser, Jerry Alexandratos

Gentoo x86 Developer: Seemant Kulleen

Gentoo Alpha Developer: Tavis Ormandy, Aron Griffis

Gentoo AMD64 Developer: Brad House

Gentoo HPPA developer: Guy Martin

Gentoo PPC developer: Pieter Van den Abeele

Gentoo SPARC developer: Joe Kallar

Editor: John P. Davis, Pierre-Henri Jondot, Eric Stockbridge, Rajiv Manglani, Jungmin Seo, Stoyan Zhekov, Jared Hudson, Colin Morey, Jorge Paulo, Carl Anderson, Jon Portnoy, Zack Gilburd, Jack Morgan, Benny Chuang, Erwin, Joshua Kinard

Reviewer: Grant Goodyear, Gerald J. Normandin Jr., Donnie Berkholz, Ken Nowack

Contributor: Lars Weiler

Obsah tohoto dokumentu podléhá licenci Creative Commons - Attribution / Share Alike
[<http://creativecommons.org/licenses/by-sa/1.0>].

Obsah

I. Instalujeme Gentoo	1
1. O instalaci Gentoo Linuxu	7
1.1. Úvod	7
1.1.1. Vítejte!	7
1.1.2. Jak je instalace strukturována?	7
1.1.3. Jaké jsou možnosti?	8
1.1.4. Problémy?	8
2. Volba správného způsobu instalace	9
2.1. Hardwarové požadavky	9
2.1.1. Úvod	9
2.1.2. Architektury	9
2.2. Rozhodněte se	10
2.2.1. Úvod	10
2.2.2. Tři etapy (stages)	10
2.2.3. Gentoo LiveCD	12
2.2.4. Gentoo Basic LiveCD	12
2.2.5. Gentoo 2-CD LiveCD sada	13
2.2.6. KDE/Gnome LiveCD	13
2.2.7. Knoppix	14
2.2.8. Existující distribuce	14
2.2.9. Net Boot (instalace ze sítě)	14
2.2.10. Volba způsobu instalace	14
2.3. Výchozí: Stáhněte, vypalte a spusťte Gentoo LiveCD	15
2.3.1. Stažení a vypálení LiveCD	15
2.3.2. Bootování LiveCD na x86 nebo AMD64	16
2.3.3. Bootování LiveCD na HPPA	18
2.3.4. Bootování LiveCD na SPARC	21
2.3.5. Bootování LiveCD na PPC	22
2.3.6. Konfigurace dalšího hardware	23
2.3.7. Nepovinné: Ladění výkonu vašeho pevného disku	23
2.3.8. Nepovinné: Uživatelské účty	23
2.3.9. Nepovinné: Spuštění SSH Démona	24
2.4. Náhradní: Stáhněte, vypalte a spusťte Knoppix	25
2.5. Náhradní: Start přes síť	26
2.5.1. Start přes síť s MIPS systémem	27
3. Konfigurace vaší sítě	29
3.1. Můžete instalovat i bez sítě, ale	29
3.1.1. Kdo může instalovat bez sítě?	29
3.1.2. Proč potřebuji připojení k síti?	29
3.1.3. Nepovinné: konfigurace proxy	29
3.1.4. Připojení k síti z jiných (non-Gentoo) instalačních médií	30
3.2. Automatická detekce sítě	30
3.2.1. Třeba to prostě funguje?	30
3.2.2. Testování sítě	31

3.3. Automatická konfigurace sítě	31
3.3.1. Výchozí: Použití skriptu net-setup	32
3.3.2. Náhradní: Použití RP-PPPoE	32
3.3.3. Náhradní: Použití PPTP	32
3.4. Manuální konfigurace sítě	33
3.4.1. Načtení potřebných modulů do jádra	33
3.4.2. Používáme DHCP	34
3.4.3. Rozumíme síťové terminologii	35
3.4.4. Používáme ifconfig a route	36
4. Příprava disků	38
4.1. Úvod do blokových zařízení	38
4.1.1. Bloková zařízení	38
4.1.2. Diskové oddíly a řezy (slices)	38
4.1.3. Diskové oddíly	38
4.2. Rozvržení diskových oddílů	39
4.2.1. Standardní rozvržení disku	39
4.2.2. Kolik a jak velkých?	40
4.3. Použití programu fdisk na architektuře x86 nebo amd64 k rozdělení vašeho disku	41
4.3.1. Změna aktuálního rozdělení disku	42
4.3.2. Odstranění všech diskových oddílů	43
4.3.3. Vytvoření startovacího oddílu	43
4.3.4. Vytvoření odkládacího oddílu	44
4.3.5. Vytvoření kořenového adresáře	45
4.3.6. Uložení rozdělení disku	45
4.4. Použití programu fdisk na architektuře Alpha k rozdělení vašeho disku	45
4.4.1. Identifikace dostupných disků	46
4.4.2. Smazání všech diskových oddílů	46
4.4.3. Vytvoření diskového oddílu	47
4.4.4. Vytvoření kořenového adresáře	48
4.4.5. Uložení rozdělení disku	49
4.5. Použití programu fdisk na architektuře SPARC k rozdělení vašeho disku	49
4.5.1. Spuštění programu fdisk	50
4.5.2. Smazání existujících diskových oddílů	50
4.5.3. Vytvoření diskového oddílu Sun Disk Label	51
4.5.4. Vytvoření startovacího diskového oddílu	52
4.5.5. Vytvoření odkládacího diskového oddílu	53
4.5.6. Vytvoření kořenového diskového oddílu	53
4.5.7. Uložení rozdělení disku	53
4.6. Použití programu mac-fdisk na architektuře PPC k rozdělení vašeho disku	54
4.7. Použití programu fdisk na architektuře HPPA k rozdělení vašeho disku	55
4.8. Použití programu fdisk na architektuře MIPS k rozdělení vašeho disku	55
4.8.1. Vytváření diskového oddílu SGI Disk Label	55
4.8.2. Nastavení té správné velikosti pro SGI Volume Header	57
4.8.3. Dokončení rozdělení disku	57
4.9. Vytváření souborových systémů	58
4.9.1. Úvod	58
4.9.2. Souborové systémy	58
4.9.3. Umístění souborového systému na diskový oddíl	59

4.9.4. Aktivace odkládacího oddílu	59
4.10. Připojování	60
5. Instalace Gentoo instalačních souborů	62
5.1. Volba správné etapy	62
5.2. Výchozí: Stažení z Internetu	62
5.2.1. Stažení etapového balíčku z Internetu	62
5.2.2. Rozbalení etapového balíčku	63
5.3. Náhradní: Použití etapového balíčku z LiveCD	64
5.3.1. Instalace obrazu Portage a zdrojového kódu	65
5.3.2. Volitelně: Příprava GRP	65
5.4. Konfigurace kompilátoru	66
5.4.1. Úvod	66
5.4.2. CHOST	66
5.4.3. CFLAGS a CXXFLAGS	67
5.4.4. MAKEOPTS	68
5.4.5. Připravít, Pozor, Ted'!	68
6. Instalace základního systému Gentoo	69
6.1. Chrootování	69
6.1.1. Nepovinné: Volba mirrorů	69
6.1.2. Kopírování DNS informací	69
6.1.3. Nepovinné: Připojení /dev	69
6.1.4. Vstup do nového prostředí	70
6.1.5. Nepovinné: aktualizace Portage	70
6.1.6. Konfigurace USE proměnné	71
6.1.7. Nepovinné: Použití distribuované Kompilace	72
6.2. Rozdíly mezi stage1, stage2 a stage3	72
6.3. Přejít od stage1 ke stage2	72
6.3.1. Úvod do Bootstrappingu	72
6.3.2. Nepovinné: Minimalizace času potřebného ke kompilaci	73
6.3.3. Nepovinné: Stažení zdrojových kódů nejdříve	73
6.3.4. Bootstrapping systému	73
6.4. Přejít od stage2 ke stage3	74
6.4.1. Úvod	74
6.4.2. Volitelně: Zobrazení následujících kroků	74
6.4.3. Volitelně: Stažení zdrojových kódů	74
6.4.4. Sestavení systému	75
7. Konfigurace jádra	76
7.1. Časové pásmo	76
7.2. Instalace zdrojových kódů	76
7.3. Výchozí: Manuální konfigurace	77
7.3.1. Úvod	78
7.3.2. Aktivace povinných voleb	78
7.3.3. Aktivace voleb doporučených pro architekturu x86	80
7.3.4. Aktivace voleb doporučených pro architekturu Alpha	80
7.3.5. Aktivace voleb doporučených pro architekturu HPPA	81
7.3.6. Aktivace voleb doporučených pro architekturu PPC	82
7.3.7. Aktivace voleb doporučených pro architekturu SPARC	83
7.3.8. Aktivace voleb doporučených pro architekturu MIPS	85
7.3.9. Aktivace voleb doporučených pro architekturu AMD64	87

7.3.10. Kompilace a instalace	88
7.4. Alternativa: Použití programu genkernel	89
7.5. Instalace samostatných modulů jádra	90
7.5.1. Instalace doplňujících modulů	90
7.5.2. Konfigurace modulů	92
8. Konfigurace systému	93
8.1. Informace o souborovém systému	93
8.1.1. Co je to fstab?	93
8.1.2. Vytvoření souboru /etc/fstab	93
8.2. Informace o nastavení sítě	95
8.2.1. Hostname, Domainname, atd.	95
8.2.2. Konfigurace sítě	96
8.2.3. Automatické zapnutí sítě při startu	97
8.2.4. Zápis informací o síti	98
8.2.5. Volitelně: Zprovoznění PCMCIA	99
8.3. Informace o systému	99
9. Konfigurace zavaděče	100
9.1. Volba zavaděče	100
9.1.1. Úvod	100
9.1.2. Volitelně: Framebuffer	100
9.2. Výchozí: GRUB	101
9.2.1. Rozumíme terminologii GRUBu	101
9.2.2. Instalace zavaděče GRUB	101
9.3. Volitelně: LILO	105
9.3.1. Instalace LILO	105
9.3.2. Konfigurace LILO	105
9.4. Volitelně: aBoot	107
9.5. Volitelně: MILO	108
9.6. Volitelně: SILO	109
9.7. Volitelně: PALO	111
9.8. Volitelně: yaBoot	111
9.8.1. Úvod	111
9.8.2. Výchozí: Použití yabootconfig	112
9.8.3. Alternativa: Ruční konfigurace yaBoot	112
9.9. Volitelně: BootX	114
9.10. Volitelně: MIPS ROM	114
10. Instalace potřebných systémových nástrojů	116
10.1. Systémový logger	116
10.2. Volitelně: Cronový démon	116
10.3. Nástroje souborového systému	117
10.4. Volitelně: Nástroje pro síť	117
11. Dokončení vaší instalace Gentoo	119
11.1. Správa uživatelů	119
11.1.1. Nastavení hesla uživatele root	119
11.1.2. Přidání uživatele pro běžné používání	119
11.2. Restartujte a kochejte se	120
11.2.1. Restart	120
11.2.2. Volitelně: Konfigurace BootX	120
11.2.3. Volitelně: Zprovoznění Gentoo/MIPS	121

11.2.4. Volitelně: Instalace dalších balíčků	122
12. Kam se vydat nyní?	124
12.1. Dokumentace	124
12.2. Gentoo Online	124
II. Práce s Gentoo	125
13. USE příznaky	129
13.1. Co jsou USE příznaky?	129
13.1.1. Myšlenka na které stojí USE příznaky	129
13.1.2. Definování USE příznaků	129
13.1.3. Jaké USE příznaky existují?	129
13.2. Použití USE příznaků	130
13.2.1. Deklarace trvalých USE příznaků	130
13.2.2. Deklarace dočasných USE příznaků	131
13.2.3. Dědění USE příznaků	131
13.2.4. Priorita příznaků	131
13.3. USE příznaky specifické pro balíčky	132
13.3.1. Zobrazení dostupných USE příznaků	132
14. Portage a software	134
14.1. Získávání informací o balíčcích	134
14.1.1. Pán všech nástrojů: emerge	134
14.1.2. Portage Tree	134
14.1.3. Hledání balíčku	136
14.1.4. Zobrazení ChangeLogu	138
14.2. Aktualizace Portage	138
14.2.1. Úvod	138
14.2.2. Volba mirroru pro rsync	138
14.2.3. Aktualizace Portage	139
14.3. Údržba software	139
14.3.1. Kompilované nebo předkompilované	139
14.3.2. Instalace programů ze zdrojového kódu	140
14.3.3. Instalace předkompilovaných balíčků	141
14.3.4. Práce se závislostmi	142
14.3.5. Aktualizace systému	143
14.3.6. Odstraňování software	145
14.4. Dostupnost software	145
14.4.1. ARCH nebo ne?	146
14.4.2. Skryté balíčky	146
14.4.3. Blokované balíčky	147
15. Vlastnosti Portage	148
15.1. DistCC	148
15.1.1. Co je distCC?	148
15.1.2. Instalace DistCC	148
15.1.3. Aktivace	148
15.2. ccache	150
15.2.1. Co je to ccache?	150
15.2.2. Instalace ccache	150
15.2.3. Aktivace podpory v Portage	150
15.3. Binární balíčky	151
15.3.1. Vytváření binárních balíčků	151

15.4. Vlastnosti související s bezpečností	152
15.4.1. Sandbox	152
15.4.2. Uživatelská práva	152
15.4.3. Aktivace sandboxu a/nebo uživatelských práv	152
15.4.4. Striktní kontroly	152
15.4.5. Smart File System práva	153
15.5. Další vlastnosti	153
15.5.1. Portage náповěda	153
16. Kontrola chování Portage	154
16.1. Nastavení ochrany souborů	154
16.1.1. Ochrana?	154
16.1.2. Deklarace CONFIG_PROTECT	154
16.1.3. etc-update	155
16.2. Síťová nastavení	156
16.2.1. Mirrory	156
16.2.2. Mirrorselect	157
16.2.3. Stahování	158
16.2.4. Nastavení programu rsync	158
16.3. Umístění adresářů	159
16.3.1. Úvod	159
16.3.2. Portage Tree	159
16.3.3. Distfiles	160
16.3.4. Balíčky a RPM	160
16.3.5. Dočasné soubory Portage	161
16.3.6. Logování	161
16.4. Další nastavení Portage	161
16.4.1. Nice Value	161
16.4.2. SLOTované balíčky a automatické čištění	162
16.4.3. Kompilace souvisejících balíčků	162
16.4.4. Formátování výstupu	163
16.5. Zdroje	164
16.5.1. Manuálové stránky	164
16.5.2. Komentované příklady	164
17. Inicializační skripty	165
17.1. Runlevely	165
17.1.1. Start systému	165
17.1.2. Inicializační skripty	165
17.1.3. Jak to funguje	165
17.1.4. Co je to runlevel?	167
17.1.5. Práce s inicializačními skripty	167
17.2. Práce s rc-update	169
17.2.1. Co je rc-update?	169
17.2.2. Přidávání a odebírání služeb	169
17.3. Konfigurace služeb	170
17.3.1. Proč je potřeba další konfigurace?	170
17.3.2. Adresář /etc/conf.d	170
17.4. Psaní init skriptů	170
17.4.1. Musím?	170
17.4.2. Schéma	171

17.4.3. Závislosti	171
17.4.4. Určování pořadí	172
17.4.5. Standardní funkce	173
17.4.6. Přidávání vlastních voleb	173
17.4.7. Proměnné pro konfiguraci služby	174
18. Proměnné prostředí	175
18.1. Proměnné prostředí?	175
18.1.1. Co jsou zač	175
18.1.2. Důležité příklady	175
18.2. Definice globálních proměnných	176
18.2.1. Adresář /etc/env.d	176
18.2.2. Skript env-update	177
18.3. Definice lokálních proměnných	178
18.3.1. Dle sezení (session)	178

Seznam obrázků

2.1. Start jádra s volbami dopcmcia a cdcache	18
2.2. Zpráva při startu HPPA systému	18
2.3. Hledání SCSI ID	19
2.4. Dostupná startovací média	19
2.5. Start z CD-ROM mechaniky	19
2.6. Interaktivní mód PALO	20
2.7. Doplnění startovací volby cdcache	20
2.8. Start jádra	20
2.9. Start z CD-ROM	21
2.10. Start gentoo jádra	21
2.11. Přihlášení do LiveCD	21
2.12. Přihlášení do LiveCD	22
2.13. Zobrazení seznamu map klávesnice	22
2.14. Zobrazení seznamu map klávesnice	22
2.15. Načtení jaderných modulů	23
2.16. Testování výkonu pevného disku	23
2.17. Ladění výkonu pevného disku	23
2.18. Změna hesla uživatele root	24
2.19. Vytváření uživatelského účtu	24
2.20. Změna uživatelského účtu	24
2.21. Spuštění SSH démona	25
2.22. Změna rootovského hesla pro Knoppix	26
2.23. Změna domovského adresáře pro uživatele root	26
2.24. Načtení provedených změn	26
2.25. Několik drobných úprav pro SGI stroje, aby TFTP pracovalo správně	27
2.26. Start přes síť pro MIPS systém	27
2.27. Triky které se vám při startu přes síť možná budou hodit	28
3.1. Definice proxy serveru	30
3.2. Doplnění jména/hesla do nastavení proxy serveru	30
3.3. Proxy server s autentizací	30
3.4. /sbin/ifconfig pro fungující síťovou kartu	31
3.5. Další testování sítě	31
3.6. Spuštění skriptu net-setup	32
3.7. Použití rp-pppoe	32
3.8. Úprava souboru /etc/ppp/chap-secrets	33
3.9. Úprava souboru /etc/ppp/options.pptp	33
3.10. Připojení k serveru přes telefonní linku	33
3.11. Hledání dostupných modulů	33
3.12. Testování dostupnosti vaší síťové karty, úspěšné	34
3.13. Testování dostupnosti vaší síťové karty, neúspěšné	34
3.14. Použití dhcpd	34
3.15. Příklad IP adresy	35
3.16. Rozdělení IP adresy na síťovou a uživatelskou část	35
3.17. Broadcast (všesměrová) adresa	36
3.18. Použití příkazu ifconfig	37

3.19. Použití příkazu route	37
3.20. Vytvoření souboru /etc/resolv.conf	37
3.21. Šablona souboru /etc/resolv.conf	37
4.1. Příklad rozdělení disku	40
4.2. Spuštění fdisku	42
4.3. Příkazová řádka programu fdisk	42
4.4. Příklad rozdělení disku	42
4.5. Mazání diskového oddílu	43
4.6. Prázdná tabulka rozdělení disku	43
4.7. Vytvoření startovacího oddílu	43
4.8. Vytvořený startovací oddíl	44
4.9. Tabulka rozdělení po vytvoření odkládacího oddílu	44
4.10. Výpis rozdělení disku po vytvoření kořenového adresáře	45
4.11. Uložení rozdělení a ukončení programu fdisk	45
4.12. Identifikace dostupných disků	46
4.13. Spuštění programu fdisk	46
4.14. Mazání diskového oddílu	47
4.15. Prázdné schéma	47
4.16. Vytváření diskového oddílu	47
4.17. Rozdělení disku po vytvoření odkládacího oddílu	48
4.18. Vytváření kořenového diskového oddílu	48
4.19. Zobrazení rozdělení disku	49
4.20. Uložení rozdělení a ukončení programu fdisk	49
4.21. Spuštění programu fdisk	50
4.22. Příkazová řádka programu fdisk	50
4.23. Zobrazení existujících diskových oddílů	50
4.24. Smazání diskového oddílu	51
4.25. Prázdný seznam diskových oddílů	51
4.26. Kroky potřebné pro vytvoření diskového oddílu Sun Disk Label	51
4.27. Zobrazení rozdělení disku	52
4.28. Vytvoření startovacího disku	52
4.29. Výpis rozdělení disku	52
4.30. Výpis rozdělení disku	53
4.31. Výpis rozdělení disku	53
4.32. Uložení rozdělení disku a ukončení programu fdisk	54
4.33. Spuštění programu mac-fdisk	54
4.34. Dělení pevného disku	55
4.35. Vytváření diskového oddílu SGI Disk Label	56
4.36. Změna velikosti diskového oddílu SGI Volume Header	57
4.37. Vytváření souborových systémů na našich diskových oddílech	59
4.38. Vytváření odkládacího oddílu (swap signature)	60
4.39. Aktivace odkládacího oddílu	60
4.40. Připojování diskových oddílů	60
4.41. Vytváření připojovacího místa /mnt/gentoo/proc	61
4.42. Připojování diskového oddílu proc	61
4.43. Bind-mount proc	61
5.1. Přesun do adresáře /mnt/gentoo	62
5.2. Zobrazení seznamu mirrorů programem lynx	63
5.3. Zobrazení seznamu mirrorů programem links2	63

5.4. Kontrola integrity etapového balíčku	63
5.5. Rozbalení etapového balíčku	63
5.6. Zobrazení seznamu dostupných obrazů	64
5.7. Připojení CD	64
5.8. Změna adresáře na /mnt/gentoo	64
5.9. Rozbalení etapového balíčku	64
5.10. Zobrazení obsahu /mnt/cdrom/snapshots	65
5.11. Rozbalení obrazu Portage	65
5.12. Zkopírování zdrojových kódů	65
5.13. Zkopírování předkompilovaných balíčků	65
5.14. GCC volba march	67
5.15. GCC volba O	67
5.16. Definice proměnných CFLAGS a CXXFLAGS	68
5.17. Definice proměnné MAKEOPTS	68
6.1. Hledání rychlých mirrorů	69
6.2. Kopírování DNS informací	69
6.3. Připojení /dev	70
6.4. Chrotnutí do nového prostředí	70
6.5. Aktualizace Portage	70
6.6. Zobrazení dostupných USE příznaků	71
6.7. Otevření souboru /etc/make.conf	71
6.8. Nastavení USE příznaků	72
6.9. Vypnutí podpory Javy	73
6.10. Zrušení proměnné USE	73
6.11. Stažení potřebných zdrojových kódů	73
6.12. Bootstrapping systému	74
6.13. Reinicializace proměnných prostředí	74
6.14. Zobrazení co "emerge system" provede	74
6.15. Stahování zdrojových kódů	75
6.16. Spuštění sestavování systému	75
7.1. Nastavení informací o časovém pásmu	76
7.2. Instalace zdrojových kódů jádra	77
7.3. Zobrazení symlinku na zdrojové kódy jádra	77
7.4. Změna symlinku	77
7.5. Spuštění nástroje menuconfig	78
7.6. Zapnutí podpory vývojových a experimentálních voleb a ovladačů	78
7.7. Volba potřebných souborových systémů	78
7.8. Konfigurace PPPoE voleb	79
7.9. Zapnutí podpory SMP	80
7.10. Volby doporučované pro architekturu Alpha	80
7.11. Zapnutí podpory HIL	81
7.12. Základní HIL podpora	82
7.13. Zapnutí plné podpory HIL	82
7.14. Vypnutí podpory ADB raw keycodes	82
7.15. Zapnutí podpory korektního RTC	82
7.16. Zapnutí podpory HFS	83
7.17. Zapnutí podpory SBUS/UPA	83
7.18. Zapnutí podpory OBP	83
7.19. Zapnutí podpory SCSI	84

7.20. Aktivace podpory síťových karet	84
7.21. Kontrola velikosti jádra	84
7.22. Stripping jádra	84
7.23. Zapnutí podpory pro SGI IP22	85
7.24. Zahnutí podpory binárních souborů Irix	85
7.25. Zapnutí podpory SGI Indigo2 pro ISA/EISA	85
7.26. Zapnutí podpory SGI paralelního portu	85
7.27. Zapnutí podpory PnP	85
7.28. Zapnutí ovladače SGI WD93C93	86
7.29. Zapnutí podpory SGI Seeq	86
7.30. Zapnutí podpory pro SGI Zilog85C30	86
7.31. Podpora Watchdog a SGI DS1286 RTC	86
7.32. Zapnutí podpory pro SGI diskové oddíly	87
7.33. Zapnutí podpory pro SGI Newport Card	87
7.34. Zapnutí podpory zvuku pro Indy/Indigo2	87
7.35. Zapnutí podpory SMP	87
7.36. Kompilace jádra	88
7.37. Instalace jádra	88
7.38. Záloha vaší konfigurace jádra	88
7.39. Emerging genkernel	89
7.40. Spuštění skriptu genkernel	89
7.41. Stažení a zapnutí hotplug	90
7.42. Zobrazení úplného seznamu instalovaných balíčků	91
7.43. Zobrazení použitých USE příznaků	91
7.44. Stažení emu10k1 se všemi USE příznaky odstraněnými	92
7.45. Zobrazení všech dostupných modulů	92
7.46. /etc/modules.autoload.d/kernel-2.4 or kernel-2.6	92
7.47. Spuštění modules-update	92
8.1. Otevření /etc/fstab	94
8.2. Příklad konfigurační řádky pro /boot v /etc/fstab	94
8.3. Vylepšená konfigurační řádka pro /boot v /etc/fstab	94
8.4. Tři řádky v /etc/fstab	94
8.5. Kompletní konfigurace /etc/fstab	94
8.6. Doplnění souborového systému openprom do /etc/fstab	95
8.7. Doplnění souborového systému usbfs do /etc/fstab	95
8.8. Nastavení hostname	95
8.9. Nastavení domainname	96
8.10. Nastavení jména NIS domény	96
8.11. Zařazení skriptu domainname do implicitního runlevelu	96
8.12. Otevření /etc/conf.d/net pro editaci	96
8.13. Syntaxe iface_eth0	97
8.14. Příklady pro /etc/conf.d/net	97
8.15. Zařazení net.eth0 do implicitního runlevelu	97
8.16. Tvorba dalších inicializačních skriptů	98
8.17. Otevření /etc/hosts	98
8.18. Zadání informací o síti	98
8.19. /etc/hosts pro samostatný a plně integrovaný počítač	98
8.20. Instalace balíčku pcmcia-cs	99
8.21. Zařazení skriptu pcmcia do implicitního runlevelu	99

8.22. Otevření <code>/etc/rc.conf</code>	99
9.1. Instalace zavaděče GRUB	101
9.2. Spuštění prostředí programu GRUB	102
9.3. Instalace GRUBu do MBR	102
9.4. Vytváření souboru <code>/boot/grub/grub.conf</code>	103
9.5. <code>grub.conf</code> pro ty kdo nepoužili skript <code>genkernel</code>	103
9.6. <code>grub.conf</code> pro ty kdo požili skript <code>genkernel</code> z LiveCD (resp. z GRP)	103
9.7. <code>grub.conf</code> pro ty kdo použili stažený skript <code>genkernel</code>	104
9.8. Doplnění volby <code>vga</code> do příkazu <code>kernel</code>	104
9.9. Instalace zavaděče LILO	105
9.10. Vytvoření <code>/etc/lilo.conf</code>	105
9.11. Vzorový <code>/etc/lilo.conf</code>	106
9.12. Použití <code>append</code> pro přidání voleb jádra	107
9.13. Ukončení instalace zavaděče LILO	107
9.14. Nainstalování zavaděče <code>aboot</code>	107
9.15. Označení startovacího disku jako bootovatelného	107
9.16. Vylepšení startu Gentoo	108
9.17. Povolení přihlášení na sériovém portu	108
9.18. Instalace zavaděče MILO	109
9.19. Instalace zavaděče MILO na startovací disketu	109
9.20. Instalace zavaděče SILO	109
9.21. Vytvoření souboru <code>/etc/silo.conf</code>	110
9.22. Příklad souboru <code>/etc/silo.conf</code>	110
9.23. Pouze pokud máte <code>/boot</code> na samostatném diskovém oddíle	110
9.24. Spuštění <code>silo</code>	110
9.25. Vzorový soubor <code>/etc/palo.conf</code>	111
9.26. Aplikace konfigurace PALO	111
9.27. Instalace <code>yaboot</code>	112
9.28. Instalace <code>yaboot</code>	112
9.29. <code>/etc/yaboot.conf</code>	113
9.30. Instalace nástroje <code>dvhtool</code>	114
9.31. Kopírování jádra do diskového oddílu SGI Volume Header	114
9.32. Zobrazení všech položek v SGI Volume Header	114
10.1. Instalace systémového loggeru	116
10.2. Instalace cronového démona	117
10.3. Instalace <code>rp-pppoe</code> s použitím GRP	117
10.4. Instalace <code>rp-pppoe</code> pro běžné uživatele	118
11.1. Nastavení hesla uživatele <code>root</code>	119
11.2. Přidání <code>ttyS0</code> do <code>/etc/securetty</code>	119
11.3. Přidání uživatele pro každodenní používání	119
11.4. Restart systému	120
11.5. Konfigurace MIPS PROM pro start do Gentoo	121
11.6. Získání rootovských práv	122
11.7. Připojení CD-2	122
11.8. Překopírování předkompilovaných balíčků	122
11.9. Instalace KDE	123
13.1. Krátký úryvek použitelných USE příznaků	130
13.2. <code>/etc/make.profile/make.defaults</code> USE proměnná	130
13.3. Příklad nastavení proměnné USE v souboru <code>/etc/make.conf</code>	131

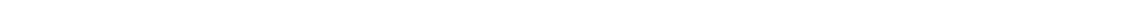
13.4. Použití USE jako proměnné prostředí	131
13.5. Úryvek z /etc/make.profile/use.defaults:	131
13.6. Spuštění příkazu emerge info	132
13.7. Zobrazení používaných USE příznaků	132
13.8. Instalace balíčku gentoolkit	133
13.9. Použití příkazu etcat pro zobrazení používaných USE příznaků	133
14.1. Zobrazení nápovědy pro emerge	134
14.2. Zobrazení Portage Tree	134
14.3. Zobrazení kategorie	135
14.4. Zobrazení balíčku	136
14.5. Zobrazení dalších souborů	136
14.6. Zobrazení informací o mozilla	137
14.7. Vyhledávání i v popisu	137
14.8. Zobrazení položek ChangeLogu pro gnumeric	138
14.9. Instalace nástroje mirrorselect	139
14.10. Spuštění nástroje mirrorselect	139
14.11. Aktualizace Portage Tree pomocí emerge rsync	139
14.12. Aktualizace Portage Tree pomocí emerge-websync	139
14.13. Kompilace gnumeric	140
14.14. Předstírání kompilace gnumeric	140
14.15. Stažení zdrojových kódů gnumeric	140
14.16. Zobrazení URL pro zdrojové kódy gnumeric	141
14.17. Instalace konkrétní verze gnumeric	141
14.18. Instalace předkompilovaného balíčku gnumeric	141
14.19. Instalace předkompilovaného balíčku gnumeric	141
14.20. Stažení a instalace předkompilovaného balíčku gnumeric	142
14.21. Předstírané stažení a instalace předkompilovaného balíčku gnumeric	142
14.22. Instalace předkompilovaného balíčku gnumeric bez použití lokální informace	142
14.23. Instalace konkrétní verze programu gnumeric	142
14.24. Zobrazení všech závislostí programu gnumeric	143
14.25. Instalace balíčku gnumeric bez ohledu na závislosti	143
14.26. Instalace pouze závislostí programu gnumeric	143
14.27. Aktualizace systémových balíčků	143
14.28. Aktualizace celého systému	144
14.29. Předstírání aktualizace celého systému	144
14.30. Upgrade celého systému	144
14.31. Odinstalování balíčků	145
14.32. Předstírání odinstalování balíčků	145
14.33. Zpráva o skrytých balíčcích	146
14.34. Dočasné povolení ~ARCH balíčků	146
14.35. Blokování balíčku	147
14.36. Řešení situace s blokem balíčku	147
15.1. Instalace DistCC	148
15.2. Možné nastavení proměnné MAKEOPTS v /etc/make.conf	149
15.3. Na konec souboru /etc/make.conf doplňte následující proměnnou specifickou pro distcc	149
15.4. Konfigurace distcc tak aby používal tři dostupné DistCC servery	149
15.5. Spuštění distccd	149
15.6. Instalace ccache	150
15.7. Úprava proměnné FEATURES v /etc/make.conf	150

15.8. Úprava proměnné CCACHE_SIZE v /etc/make.conf	150
15.9. Zobrazení statistik ccache	151
15.10. Použité quickpkg	151
15.11. Instalace programu gnumeric a současné vytvoření binárního balíčku	151
15.12. Automatické vytváření binárních balíčků	151
15.13. Vytvoření binárního balíčku pro gnumeric	152
15.14. Aktivace uživatelských práv a sandboxu	152
15.15. Aktivace striktních kontrol	153
15.16. Aktivace Smart File System práv	153
15.17. Více informací týkajících se Portage	153
16.1. Příklad nastavení proměnné CONFIG_PROTECT	154
16.2. Příklad nastavení CONFIG_PROTECT_MASK	154
16.3. Zobrazení informací o nastavení ochrany souborů	155
16.4. Spuštění etc-update	155
16.5. Volby etc-update	155
16.6. Aktualizace konkrétního konfiguračního souboru	155
16.7. Příkazy použitelné při interaktivním slučování	156
16.8. Definice SYNC v /etc/make.conf	157
16.9. Definice GENTOO_MIRRORS v /etc/make.conf	157
16.10. Definice PORTAGE_BINHOST v /etc/make.conf	157
16.11. Instalace nástroje mirrorselect	157
16.12. Spuštění nástroje mirrorselect	158
16.13. Výchozí nastavení proměnných FETCHCOMMAND a RESUMECOMMAND	158
16.14. Proměnná RSYNC_EXCLUDEFROM	159
16.15. Proměnná RSYNC_RETRIES	159
16.16. Proměnná RSYNC_TIMEOUT	159
16.17. Proměnná PORTDIR	159
16.18. Proměnná PORTDIR_OVERLAY	160
16.19. Proměnná DISTDIR	160
16.20. Proměnná PKGDIR	160
16.21. Proměnná	160
16.22. Proměnná PORTAGE_TMPDIR	161
16.23. Proměnná BUILD_PREFIX	161
16.24. Proměnná PORT_LOGDIR	161
16.25. Proměnná PORTAGE_NICENESS	162
16.26. Proměnná CLEAN_DELAY	162
16.27. Proměnná AUTOCLEAN	162
16.28. Zobrazení informací o CHOST, CFLAGS and CXXFLAGS	163
16.29. Výchozí nastavení proměnné DEBUGBUILD	163
16.30. Proměnná MAKEOPTS	163
16.31. Proměnná ROOT	163
16.32. Proměnná NOCOLOR	164
16.33. Zobrazení informací v make.conf	164
17.1. Inicializace systému v /etc/inittab	166
17.2. Pokračování inicializace systému	166
17.3. Řádek initdefault	166
17.4. Definice runlevelů	166
17.5. Definice virtuálních konzolí	167
17.6. Spuštění postfixu	168

17.7. Zastavení postfixu, ale závislé služby stále běží	168
17.8. Informace o stavu pro postfix	168
17.9. Resetování stavové informace pro postfix	168
17.10. Zobrazení seznamu služeb vyžadovaných pro běh postfixu	168
17.11. Zobrazení seznamu služeb které pro svůj běh vyžadují postfix	169
17.12. Zobrazení seznamu závislostí které postfixu chybí	169
17.13. Odstranění postfixu z runlevelu "default"	169
17.14. Zobrazení seznamu dostupných init skriptů	170
17.15. Proměnné definované v /etc/conf.d/apache2	170
17.16. Základní schéma init skriptu	171
17.17. Informace o závislostech postfixu	171
17.18. Funkce depend() služby portmap	172
17.19. Spouštění služby jako první v celém runlevelu	172
17.20. Příklad funkce start()	173
17.21. Zobrazení manuálové stránky pro start-stop-daemon	173
17.22. Doplnění podpory pro proměnnou restartdelay	173
18.1. Vzorové definice	176
18.2. /etc/env.d/05gcc	177
18.3. /etc/env.d/99local	177
18.4. Pořadí aktualizace používané skriptem env-update	177
18.5. Aktualizace prostředí	178
18.6. Aktualizace proměnné PATH pro lokální použití v ~/.bashrc	178
18.7. Změna proměnné PATH pouze pro aktuální session	178

Část I. Instalujeme Gentoo

V této části se naučíte instalovat Gentoo na svůj počítač.



Obsah

1. O instalaci Gentoo Linuxu	7
1.1. Úvod	7
1.1.1. Vítejte!	7
1.1.2. Jak je instalace strukturována?	7
1.1.3. Jaké jsou možnosti?	8
1.1.4. Problémy?	8
2. Volba správného způsobu instalace	9
2.1. Hardwarové požadavky	9
2.1.1. Úvod	9
2.1.2. Architektury	9
2.2. Rozhodněte se	10
2.2.1. Úvod	10
2.2.2. Tři etapy (stages)	10
2.2.3. Gentoo LiveCD	12
2.2.4. Gentoo Basic LiveCD	12
2.2.5. Gentoo 2-CD LiveCD sada	13
2.2.6. KDE/Gnome LiveCD	13
2.2.7. Knoppix	14
2.2.8. Existující distribuce	14
2.2.9. Net Boot (instalace ze sítě)	14
2.2.10. Volba způsobu instalace	14
2.3. Výchozí: Stáhněte, vypalte a spusťte Gentoo LiveCD	15
2.3.1. Stažení a vypálení LiveCD	15
2.3.2. Bootování LiveCD na x86 nebo AMD64	16
2.3.3. Bootování LiveCD na HPPA	18
2.3.4. Bootování LiveCD na SPARC	21
2.3.5. Bootování LiveCD na PPC	22
2.3.6. Konfigurace dalšího hardware	23
2.3.7. Nepovinné: Ladění výkonu vašeho pevného disku	23
2.3.8. Nepovinné: Uživatelské účty	23
2.3.9. Nepovinné: Spuštění SSH Démona	24
2.4. Náhradní: Stáhněte, vypalte a spusťte Knoppix	25
2.5. Náhradní: Start přes síť	26
2.5.1. Start přes síť s MIPS systémem	27
3. Konfigurace vaší sítě	29
3.1. Můžete instalovat i bez sítě, ale	29
3.1.1. Kdo může instalovat bez sítě?	29
3.1.2. Proč potřebuji připojení k síti?	29
3.1.3. Nepovinné: konfigurace proxy	29
3.1.4. Připojení k síti z jiných (non-Gentoo) instalačních médií	30
3.2. Automatická detekce sítě	30
3.2.1. Třeba to prostě funguje?	30
3.2.2. Testování sítě	31
3.3. Automatická konfigurace sítě	31

3.3.1. Výchozí: Použití skriptu net-setup	32
3.3.2. Náhradní: Použití RP-PPPoE	32
3.3.3. Náhradní: Použití PPTP	32
3.4. Manuální konfigurace sítě	33
3.4.1. Načtení potřebných modulů do jádra	33
3.4.2. Používáme DHCP	34
3.4.3. Rozumíme síťové terminologii	35
3.4.4. Používáme ifconfig a route	36
4. Příprava disků	38
4.1. Úvod do blokových zařízení	38
4.1.1. Bloková zařízení	38
4.1.2. Diskové oddíly a řezy (slices)	38
4.1.3. Diskové oddíly	38
4.2. Rozvržení diskových oddílů	39
4.2.1. Standardní rozvržení disku	39
4.2.2. Kolik a jak velkých?	40
4.3. Použití programu fdisk na architektuře x86 nebo amd64 k rozdělení vašeho disku	41
4.3.1. Změna aktuálního rozdělení disku	42
4.3.2. Odstranění všech diskových oddílů	43
4.3.3. Vytvoření startovacího oddílu	43
4.3.4. Vytvoření odkládacího oddílu	44
4.3.5. Vytvoření kořenového adresáře	45
4.3.6. Uložení rozdělení disku	45
4.4. Použití programu fdisk na architektuře Alpha k rozdělení vašeho disku	45
4.4.1. Identifikace dostupných disků	46
4.4.2. Smazání všech diskových oddílů	46
4.4.3. Vytvoření diskového oddílu	47
4.4.4. Vytvoření kořenového adresáře	48
4.4.5. Uložení rozdělení disku	49
4.5. Použití programu fdisk na architektuře SPARC k rozdělení vašeho disku	49
4.5.1. Spuštění programu fdisk	50
4.5.2. Smazání existujících diskových oddílů	50
4.5.3. Vytvoření diskového oddílu Sun Disk Label	51
4.5.4. Vytvoření startovacího diskového oddílu	52
4.5.5. Vytvoření odkládacího diskového oddílu	53
4.5.6. Vytvoření kořenového diskového oddílu	53
4.5.7. Uložení rozdělení disku	53
4.6. Použití programu mac-fdisk na architektuře PPC k rozdělení vašeho disku	54
4.7. Použití programu fdisk na architektuře HPPA k rozdělení vašeho disku	55
4.8. Použití programu fdisk na architektuře MIPS k rozdělení vašeho disku	55
4.8.1. Vytváření diskového oddílu SGI Disk Label	55
4.8.2. Nastavení té správné velikosti pro SGI Volume Header	57
4.8.3. Dokončení rozdělení disku	57
4.9. Vytváření souborových systémů	58
4.9.1. Úvod	58
4.9.2. Souborové systémy	58
4.9.3. Umístění souborového systému na diskový oddíl	59
4.9.4. Aktivace odkládacího oddílu	59

4.10. Připojování	60
5. Instalace Gentoo instalačních souborů	62
5.1. Volba správné etapy	62
5.2. Výchozí: Stažení z Internetu	62
5.2.1. Stažení etapového balíčku z Internetu	62
5.2.2. Rozbalení etapového balíčku	63
5.3. Náhradní: Použití etapového balíčku z LiveCD	64
5.3.1. Instalace obrazu Portage a zdrojového kódu	65
5.3.2. Volitelně: Příprava GRP	65
5.4. Konfigurace kompilátoru	66
5.4.1. Úvod	66
5.4.2. CHOST	66
5.4.3. CFLAGS a CXXFLAGS	67
5.4.4. MAKEOPTS	68
5.4.5. Připravit, Pozor, Ted'!	68
6. Instalace základního systému Gentoo	69
6.1. Chrootování	69
6.1.1. Nepovinné: Volba mirrorů	69
6.1.2. Kopírování DNS informací	69
6.1.3. Nepovinné: Připojení /dev	69
6.1.4. Vstup do nového prostředí	70
6.1.5. Nepovinné: aktualizace Portage	70
6.1.6. Konfigurace USE proměnné	71
6.1.7. Nepovinné: Použití distribuované Kompilace	72
6.2. Rozdíly mezi stage1, stage2 a stage3	72
6.3. Přechod od stage1 ke stage2	72
6.3.1. Úvod do Bootstrappingu	72
6.3.2. Nepovinné: Minimalizace času potřebného ke kompilaci	73
6.3.3. Nepovinné: Stažení zdrojových kódů nejdříve	73
6.3.4. Bootstrapping systému	73
6.4. Přechod od stage2 ke stage3	74
6.4.1. Úvod	74
6.4.2. Volitelně: Zobrazení následujících kroků	74
6.4.3. Volitelně: Stažení zdrojových kódů	74
6.4.4. Sestavení systému	75
7. Konfigurace jádra	76
7.1. Časové pásmo	76
7.2. Instalace zdrojových kódů	76
7.3. Výchozí: Manuální konfigurace	77
7.3.1. Úvod	78
7.3.2. Aktivace povinných voleb	78
7.3.3. Aktivace voleb doporučených pro architekturu x86	80
7.3.4. Aktivace voleb doporučených pro architekturu Alpha	80
7.3.5. Aktivace voleb doporučených pro architekturu HPPA	81
7.3.6. Aktivace voleb doporučených pro architekturu PPC	82
7.3.7. Aktivace voleb doporučených pro architekturu SPARC	83
7.3.8. Aktivace voleb doporučených pro architekturu MIPS	85
7.3.9. Aktivace voleb doporučených pro architekturu AMD64	87
7.3.10. Kompilace a instalace	88

7.4. Alternativa: Použití programu genkernel	89
7.5. Instalace samostatných modulů jádra	90
7.5.1. Instalace doplňujících modulů	90
7.5.2. Konfigurace modulů	92
8. Konfigurace systému	93
8.1. Informace o souborovém systému	93
8.1.1. Co je to fstab?	93
8.1.2. Vytvoření souboru /etc/fstab	93
8.2. Informace o nastavení sítě	95
8.2.1. Hostname, Domainname, atd.	95
8.2.2. Konfigurace sítě	96
8.2.3. Automatické zapnutí sítě při startu	97
8.2.4. Zápis informací o síti	98
8.2.5. Volitelně: Zprovoznění PCMCIA	99
8.3. Informace o systému	99
9. Konfigurace zavaděče	100
9.1. Volba zavaděče	100
9.1.1. Úvod	100
9.1.2. Volitelně: Framebuffer	100
9.2. Výchozí: GRUB	101
9.2.1. Rozumíme terminologii GRUBu	101
9.2.2. Instalace zavaděče GRUB	101
9.3. Volitelně: LILO	105
9.3.1. Instalace LILO	105
9.3.2. Konfigurace LILO	105
9.4. Volitelně: aBoot	107
9.5. Volitelně: MILO	108
9.6. Volitelně: SILO	109
9.7. Volitelně: PALO	111
9.8. Volitelně: yaBoot	111
9.8.1. Úvod	111
9.8.2. Výchozí: Použití yabootconfig	112
9.8.3. Alternativa: Ruční konfigurace yaBoot	112
9.9. Volitelně: BootX	114
9.10. Volitelně: MIPS ROM	114
10. Instalace potřebných systémových nástrojů	116
10.1. Systémový logger	116
10.2. Volitelně: Cronový démon	116
10.3. Nástroje souborového systému	117
10.4. Volitelně: Nástroje pro síť	117
11. Dokončení vaší instalace Gentoo	119
11.1. Správa uživatelů	119
11.1.1. Nastavení hesla uživatele root	119
11.1.2. Přidání uživatele pro běžné používání	119
11.2. Restartujte a kochejte se	120
11.2.1. Restart	120
11.2.2. Volitelně: Konfigurace BootX	120
11.2.3. Volitelně: Zprovoznění Gentoo/MIPS	121
11.2.4. Volitelně: Instalace dalších balíčků	122

12. Kam se vydat nyní?	124
12.1. Dokumentace	124
12.2. Gentoo Online	124

Kapitola 1. O instalaci Gentoo Linuxu

Abstrakt

Uživatelé, kteří Gentoo neznají často neví že Gentoo je především o možnosti volby.

1.1. Úvod

1.1.1. Vítejte!

Nejdříve ze všeho Vás vítáme v Gentoo. Chystáte se vstoupit do světa voleb a výkonu. Podstatou Gentoo je možnost volby. Při instalaci Gentoo, a na to budete několikrát jasně upozorněni - si můžete vybrat jak velkou část systému si chcete kompilovat sami, jak chcete Gentoo instalovat, který systémový logger chcete, atd.

Gentoo je rychlá a moderní metadistribuce s čistým a flexibilním designem. Gentoo je postaveno na svobodném softwaru a neskrývá před svými uživateli co se skrývá pod povrchem. Portage, systém správy balíčků používaný Gentoo, je napsán v Pythonu, což znamená že se můžete jednoduše podívat do zdrojového kódu a upravovat ho. Balíčkovací systém Gentoo používá zdrojový kód (ačkoliv obsahuje i podporu pro předkompilované balíčky) a konfigurace Gentoo je prováděna přes obyčejné textové soubory. Jinými slovy, otevřenost je všude.

Je velice důležité abyste pochopili že možnost volby je to co dělá systém Gentoo funkčním. Snažíme se vás nenutit do ničeho co se vám nelíbí. Pokud máte dojem že vás do něčeho nutíme, pošlete nám prosím bugreport (<http://bugs.gentoo.org> [<http://bugs.gentoo.org/>]).

1.1.2. Jak je instalace strukturována?

Na instalaci systému Gentoo se můžete dívat jako na proceduru o 10 krocích, které odpovídají kapitolám 2 - 11 této části. Každý krok končí v určitém stavu:

- Po kroku 1 jste v prostředí připraveném k instalaci systému Gentoo
- Po kroku 2 je vaše internetové připojení připraveno k instalaci Gentoo (tento krok je v některých případech nepovinný)
- Po kroku 3 jsou vaše pevné disky připraveny k nainstalování Gentoo
- Po kroku 4 je vaše instalační prostředí připraveno a chroot je nastaven do tohoto nového prostředí
- Po kroku 5 jsou nainstalovány hlavní balíčky společné pro všechny instalace Gentoo
- Po kroku 6 máte zkompileováno své Linuxové jádro
- Po kroku 7 máte napsánu většinu konfiguračních souborů Gentoo
- Po kroku 8 byl nainstalován a nakonfigurován zvolený zavaděč

- Po kroku 9 jsou nainstalovány potřebné systémové nástroje (které si můžete vybrat z pěkně dlouhého seznamu)
- Po kroku 10 jste přihlášení do své nové instalace Gentoo

Pokud budete stát před nějakou volbou, budeme se vám snažit vysvětlit jaká jsou pro a proti. Poté budeme pokračovat s výchozí volbou, označenou slovem "Default: " v názvu. Další možné volby jsou označeny slovem "Náhradní:". Nevykládejte si to tak že vám tuto volbu jakkoliv vnucujeme. Pouze předpokládáme že většina uživatelů zvolí právě ji.

Někdy můžete provést nepovinný krok. Takové kroky jsou označeny jako "Optional: " a nejsou potřeba pro nainstalování systému Gentoo. Nicméně některé nepovinné kroky závisí na vašich předchozích rozhodnutích. Až k takovému kroku dojdete upozorníme vás na něj, a to již ve chvíli kdy činíte toto rozhodnutí a stejně tak na místě kde je tento krok popisován.

1.1.3. Jaké jsou možnosti?

Gentoo můžete instalovat mnoha různými způsoby. Můžete si stáhnout naše LiveCD a instalovat z něj (instalační CD), instalovat z již existující distribuce, z bootovatelného CD (jako je například Knoppix), přes síťové spojení, atd.

Máte několik možností - můžete si celý systém zkompilovat od samého začátku (from scratch) nebo instalovat předkompilované balíčky abyste ušetřili čas a měli plně funkční systém co nejdříve. A samozřejmě máte možnosti "mezi tím" ve kterých neinstalujete vše ale začínáte s jakýmsi polopřipraveným systémem.

Pokud chcete Gentoo instalovat z předkompilovaných balíčků nemáte jinou možnost než použít LiveCD obsahující předkompilované balíčky. Novější vydání Gentoo budou tyto balíčky poskytovat i přes Internet.

1.1.4. Problémy?

Pokud při instalaci (nebo v instalační dokumentaci) narazíte na problém, navštivte prosím náš systém správy chyb a podívejte se zda je tato chyba již oznámena. Pokud není, vytvořte hlášení o chybě abychom ji mohli napravit. Nebojte se vývojářů kterým byla vaše chyba přiřazena - většinou lidi nežerou.

Pokud si nejste jisti zda jde o chybu uživatelskou (i když čtete dokumentaci pečlivě můžete udělat chybu) nebo softwarovou (i když instalaci/dokumentaci testujeme pečlivě může nám něco uniknout) můžete se připojit k #gentoo na irc.freenode.net. Samozřejmě, vítání jste i kdykoliv jindy.

Pokud máte otázku vztahující se k systému Gentoo, podívejte se do seznamu Frequently Asked Questions (FAQ, Často Kladené Dotazy), dostupného v Gentoo Dokumentaci. Můžete se také podívat do seznamů FAQ v našich diskusních forech. Pokud zde nebudete moci najít odpověď na svou otázku, zeptejte se na #gentoo, našem IRC kanále na irc.freenode.net. Ano, několik z nás jsou blázni co vysedávají na IRC ;-)

Kapitola 2. Volba správného způsobu instalace

Abstrakt

Gentoo můžete instalovat mnoha způsoby. Můžete použít naše LiveCD, existující distribuci atd.

2.1. Hardwarové požadavky

2.1.1. Úvod

Ještě než začneme, uvedeme seznam hardwarových požadavků které musí váš počítač splňovat, abyste na něj mohli úspěšně nainstalovat Gentoo. Tyto požadavky samozřejmě závisí i na architektuře vašeho počítače.

2.1.2. Architektury

Gentoo je oficiálně dostupné pro sedm architektur a má experimentální podporu pro jednu další. Oficiálně podporované architektury jsou *x86* (včetně všech subarchitektur, jako jsou Pentium, Athlon, atd.), *sparc* (a to jak Sparc32 tak i Sparc64), *ppc* (PowerPC), *hppa*, *alpha*, *mips* a *amd64*. Experimentální podpora je dostupná pro architekturu *ia64*.

Za předpokladu že znáte svou architekturu, zkontrolujte následující požadavky před pokračováním v instalaci Gentoo:

- Potřebujete alespoň 1 GB volného prostoru na disku
- Pokud nebudete používat předkompilované balíčky, potřebujete alespoň 300 MB paměti (RAM + swap)
- Pro architekturu *x86* potřebujete alespoň procesor 486 a alespoň 64 MB paměti
- Pro architekturu *Alpha* byste měli zkontrolovat Alpha/Linux FAQ [<http://www.alphalinux.org/faq/FAQ-5.html>]
- Pro architekturu *hppa* byste měli zkontrolovat PA Team Website [<http://www.pateam.org/>]
- Pro architekturu *PowerPC* potřebujete alespoň PowerPC (alespon G3 nebo G4 jako je iMac, iBook, PowerBook, etc.)
- Pro architekturu *SPARC* byste měli zkontrolovat UltraLinux FAQ [http://www.ultralinux.org/faq.html#s_2]
- Pro architekturu *MIPS* byste měli zkontrolovat dokument MIPS Hardwarové Požadavky [<http://www.gentoo.org/doc/en/mips-requirements.xml>]

- Pro architekturu *AMD64* byste měli zkontrolovat *AMD Tech Notes* [http://dev.gentoo.org/~brad_mssw/amd64-tech-notes.html]

2.2. Rozhodněte se

2.2.1. Úvod

Pořád ještě chcete zkusit Gentoo? Dobrá, v tom případě je ta správná chvíle zvolit si způsob instalace který chcete použít. Ano, máte možnost volby, ne, možnosti nejsou rovnocenné, a ano, výsledek je vždy stejný - základní systém Gentoo.

Popisované způsoby instalace jsou:

- Gentoo LiveCD
- Knoppix
- Jiná distribuce
- Net Boot (instalace přes síť)

Každý způsob instalace má své výhody a nevýhody. Uvedeme pro a proti každého způsobu takže budete mít všechny informace potřebné k rozhodnutí. Ale než budeme pokračovat, vysvětlíme si co je to třítapová instalace.

2.2.2. Tři etapy (stages)

Gentoo Linux může být instalován za použití jednoho ze tří etapových archivů. To který si zvolíte závisí na tom jakou část systému si sami chcete zkompilovat. Archiv *stage1* je používán pokud chcete pouze nabootovat a sestavit si celý systém sami (from scratch). Archiv *stage2* je používán pro sestavení celého systému z "polozkompilovaného" stavu. Archiv *stage3* již obsahuje základní Gentoo Linux systém který pro vás byl sestaven. Jak vysvětlíme později, můžete Gentoo nainstalovat bez toho abyste cokoliv kompilovali (kromě vlastního jádra). Pokud chcete právě toto, musíte použít archiv *stage3* a Gentoo LiveCD obsahující předkompilované balíčky.

A teď kterou etapu byste si měli zvolit?

Pokud použijete archiv *stage1*, budete mít plnou kontrolu nad nastavením procesu optimalizace a voleb během sestavování systému. Díky tomu je archiv *stage1* vhodný pro zkušené uživatele kteří vědí co dělají. Je to také skvělý způsob instalace pro ty kdo vědí jak Gentoo Linux vlastně funguje.

Stage1	Pro a proti
+	Poskytuje vám absolutní kontrolu nad nastavením optimalizace a základních voleb během sestavování systému.
+	Vhodné pro zkušené uživatele kteří vědí co dělají.

Stage1	Pro a proti
+	Umožňuje vám naučit se více o tom jak Gentoo funguje uvnitř.
-	Dokončení instalace zabere mnoho času.
-	Pokud nemáte v úmyslu měnit nastavení, je to pravděpodobně ztráta času.

Archiv *stage2* vám umožňuje přeskočit první kroky (bootstrap process) což je v pořádku pokud jste spokojeni s nastavením optimalizace které jsme zvolili pro tento náš balíček.

Stage2	Pro a proti
+	Nepotřebujete bootstrap.
+	Rychlejší než při použití <i>stage1</i> .
+	Stále ještě můžete měnit některá nastavení.
-	Nastavení která můžete měnit není tolik jako při použití <i>stage1</i> .
-	Není to nejrychlejší cesta jak nainstalovat Gentoo.
-	Musíte souhlasit s nastaveními která jsme udělali při vytváření bootstrapu.

Volba archivu *stage3* umožňuje nejrychlejší instalaci Gentoo Linuxu, ale znamená také že základní systém bude obsahovat optimalizační volby které jsme pro vás zvolili (což jsou upřímně řečeno dobrá nastavení která byla pečlivě vybírána tak aby zlepšila výkon a přitom neohrozila stabilitu). Balíček *stage3* je také vyžadován pro instalaci Gentoo Linuxu za použití předkompilovaných balíčků.

Stage3	Pro a proti
+	Nejrychlejší způsob instalace základního systému Gentoo.
-	Nemůžete konfigurovat základní systém - ten je již zcela vytvořen.
-	Nemůžete se chvástat tím že jste použili <i>stage1</i> nebo <i>stage2</i> .

Zapište si (nebo zapamatujte) který archiv chcete použít. Budete to potřebovat později až se budete rozhodovat které LiveCD (nebo jiný způsob instalace) použijete. Mohlo by vás zajímat že pokud se rozhodnete použít jiné optimalizační volby až po nainstalování Gentoo, budete moci překompilovat celý systém s novým nastavením.

Nyní se podívejme na seznam dostupných způsobů instalace:

- Gentoo LiveCD

- Gentoo basic LiveCD
 - Gentoo 2-CD LiveCD sada
 - Gentoo KDE/Gnome LiveCD
-
- Knoppix
 - Existující distribuce
 - Net Boot (start ze sítě)

2.2.3. Gentoo LiveCD

Gentoo LiveCD jsou bootovatelná CD obsahující plně samostatné prostředí Gentoo a díky tomu vám umožňují nainstalovat Linux z CD. Během bootování je rozpoznáván váš hardware a jsou načítány příslušné ovladače. Tato CD jsou udržována vývojáři Gentoo a jsou dostupná pro všechny podporované platformy (x86, alpha, sparc, ppc, hppa). Pro některé platformy je dokonce dostupných několik různých LiveCD.

Všechna LiveCD vám umožňují nainstalovat systém, nastavit síťové připojení a spustit instalaci Gentoo přes Internet. Avšak některá LiveCD obsahují také potřebné zdrojové kódy nebo dokonce předkompilované balíčky takže můžete Gentoo instalovat i bez síťového spojení.

Ted' určitě vykřikujete "Cože? Předkompilované balíčky?" Ano, ovšem, předkompilované balíčky. Protože Gentoo je o možnosti volby, chceme uživatelům poskytnout i možnost rychle nainstalovat Gentoo. Sbíрка předkompilovaných balíčků se nazývá Gentoo Reference Platform (GRP). O GRP se budeme hlouběji bavit dále během instalace.

Poskytujeme několik různých typů LiveCD. Následující tabulka uvádí která LiveCD jsou dostupná pro vaši architekturu.

Architektura	Basic LiveCD	2-CD LiveCD sada	KDE/Gnome LiveCD
x86	+	+	
sparc	+	+	
ppc	+	+	+
hppa	+		
alpha	+		
amd64	+		

Co tato LiveCD obsahují?

2.2.4. Gentoo Basic LiveCD

Toto je jednoduché bootovatelné CD, jehož jediným účelem je nainstalovat systém, připravit síťové spoje-

ní a posléze pokračovat v instalaci Gentoo Linuxu. Neobsahuje žádné etapové balíky (i když v některých případech může obsahovat *stage1* balík), zdrojový kód nebo předkompilované balíčky.

Basic LiveCD	Pro a proti
+	Nejmenší download
+	Vhodné pro celou architekturu
+	Můžete použít <i>stage1</i> , <i>stage2</i> nebo <i>stage3</i> po stažení balíku přes síť.
-	Neobsahuje žádné etapové balíky, žádný obraz portage, žádné GRP balíčky a není tedy vhodné pro instalaci bez sítě.

2.2.5. Gentoo 2-CD LiveCD sada

První CD je bootovatelné a obsahuje všechny etapové balíčky pro specifickou architekturu (jako jsou Athlon XP, G4, atd.) současně se zdrojovými kódy a předkompilovanými balíčky. Druhé CD obsahuje pouze předkompilované balíčky a může být použito pro instalaci dalších balíčků po ukončení instalace Gentoo. Pro instalaci systému potřebujete pouze CD-1, ale pokud chcete OpenOffice, Mozillu, KDE, GNOME atd. a nechcete si to všechno kompilovat sami, potřebujete také CD-2.

Poskytujeme jak "implicitní" LiveCD sadu, bootovatelnou na všech subarchitekturách specifické architektury, tak i optimalizované LiveCD sady pro jednotlivé subarchitektury (jako je například Athlon XP, G4, atd.)

2-CD LiveCD sada	Pro a proti
+	Optimalizováno pro vaši architekturu a subarchitekturu
+	K dispozici jsou předkompilované balíčky pro rychlou instalaci Gentoo
+	Obsahují všechno co potřebujete, takže můžete instalovat bez síťového připojení.
-	Velký objem dat ke stažení.

2.2.6. KDE/Gnome LiveCD

Toto CD obsahuje kompletní funkční prostředí Gentoo obsahující desktopové prostředí KDE a Gnome, Apache webserver, několik webových prohlížečů, irc klienty, prostředky pro firewall/router, nástroje pro audit sítě, mnoho CLI nástrojů ... Toto CD může být použito například k opravě poškozeného filesystému. KDE/Gnome LiveCD obsahuje kompletní sadu nástrojů gcc (se zapnutými distcc, ccache), stačí jenom nainstalovat všechny systémy v síti z tohoto CD a zažijete sílu distribuované kompilace. Toto CD může být použito pro instalaci Gentoo, ale neobsahuje GRP, zdrojové kódy ani portage snapshot nebo etapové balíčky.

¹CLI - Command Line Interface

KDE/Gnome LiveCD	Pro a proti
+	Můžete pracovat s Gentoo aniž byste ho instalovali na svůj pevný disk
+	Můžete z něj nainstalovat Gentoo :)
-	Velký download, ale neobsahuje portage snapshot, etapové balíčky, předkompilované balíčky ani zdrojové kódy.

2.2.7. Knoppix

Knoppix je dobře známé bootovatelné CD s plně funkčním desktopovým prostředím. Umožňuje vám na-bootovat z CD a, bez přístupu k vašemu pevnému disku, spouštět kancelářské aplikace nebo systémové nástroje. V našem případě budeme používat systém a síťové nástroje pro nainstalování Gentoo.

Pokud plánujete použít Knoppix, můžete ho stáhnout z domovské stránky Knoppixu [<http://www.knoppix.com/>].

2.2.8. Existující distribuce

Pokud začnete instalovat z jiné již nainstalované Linuxové distribuce, můžete využívat všechny nástroje již dostupné ve vaší Linuxové distribuci. Pokud chcete instalovat Gentoo z existující distribuce, můžete přeskóčit zbytek této kapitoly a pokračovat s kapitolou Konfigurace vaší sítě.

2.2.9. Net Boot (instalace ze sítě)

V některých případech nemusí být možné nastartovat systém z CD nebo použít k instalaci Gentoo nějakou již existující distribuci. V případě že má váš systém síťovou kartu a BIOS/ROM schopný netbootu (PXE), potom můžete při startu systému natáhnout malý systém přes síť a posléze instalovat z tohoto systému.

Pokud chcete použít tuto metodu (uživatelé MIPS nemají jinou možnost) měli byste mít funkční DHCP a TFTP server. Informace jak nastavit DHCP server [http://www.gentoo.org/doc/en/diskless-howto.xml#doc_chap3] a TFTP server [http://www.gentoo.org/doc/en/diskless-howto.xml#doc_chap4] jsou dostupné v Diskless-HOWTO [<http://www.gentoo.org/doc/en/diskless-howto.xml>].

V závislosti na vašem nastavení budete možná muset vytvořit nebo stáhnout kernel image pro svůj systém.

2.2.10. Volba způsobu instalace

Nyní se rozhodněte pro jeden ze způsobů instalace Gentoo a pokračujte v příslušném oddíle. Protože toto je Gentoo Handbook, budeme za výchozí volbu považovat Gentoo LiveCD, ale v příslušných oddílech najdete potřebné informace i o ostatních způsobech instalace.

- Výchozí: Stáhněte, vypalte a spust'te Gentoo LiveCD
- Náhradní: Stáhněte, vypalte a spust'te Knoppix
- Náhradní: Start přes síť

2.3. Výchozí: Stáhněte, vypalte a spust'te Gentoo LiveCD

2.3.1. Stažení a vypálení LiveCD

Vybrali jste si použití Gentoo LiveCD (pokud ne, čtete špatnou část). Nejdříve musíme stáhnout a vypálit zvolené LiveCD. O možných druzích LiveCD už jsme se bavili, ale kde je můžete stáhnout?

Následující tabulka obsahuje seznam relativních cest na kterých můžete najít LiveCD ISO obrazy (obrazy CD). Cesta je relativní vzhledem k hlavnímu adresáři Gentoo na našich mirrorrech [<http://www.gentoo.org/main/en/mirrors.xml>].

Architektura	Cesta
x86	releases/x86/1.4/livecd
sparc	experimental/sparc/1.4/livecd
ppc	releases/ppc/1.4/livecd
hppa	experimental/hppa/1.4/livecd
alpha	experimental/alpha/1.4/livecd
amd64	experimental/amd64/1.4/livecd

Navštivte jeden z našich mirrorů [<http://www.gentoo.org/main/en/mirrors.xml>] a přesuňte se do adresáře kde jsou umístěna vaše LiveCD. V tomto adresáři se nachází tzv. ISO-soubory. To jsou úplné obrazy CD, které můžete vypálit na CD-R. Pokud narazíte na několik ISO-souborů s různými date tagy (jako je například `livecd-x86-cd1-20031029.iso`) vyberte to nejnovější.

V případě že máte podezření že vámi stažený soubor je poškozený a chcete to ověřit, můžete zkontrolovat jeho kontrolní součet algoritmem MD5 a srovnat ho se součtem který poskytujeme (jako je například `livecd-x86-cd1-20031029.iso.md5`). Kontrolní součet můžete zkontrolovat nástrojem `md5sum` pod Linuxem/Unixem nebo `md5summer` pod Windows.

Poznámka

Přečtěte si prosím popisy v Gentoo Store [<http://store.gentoo.org/>], abyste zjistili které LiveCD (nebo etapový balíček) byste měli použít pro váš procesor. Obvyklou chybou je volba LiveCD nebo etapového balíčku pro novější procesor než ve skutečnosti máte (například volba "athlon-xp" pro běžný AMD Athlon).

Pro vypálení staženého ISO souboru musíte zvolit raw-burning mód. Jak to udělat závisí na programu

který používáte k palení CD. Podíváme se jak to provést pro několik populárních nástrojů.

- Pokud používáte program EasyCD Creator™, zvolte položku File, Record CD from CD image. Poté změňte položku Files of type na *ISO image file*. Najděte stažený ISO soubor a klikněte na Open. Až kliknete na tlačítko Start burning bude ISO obraz vypálen přímo na vložený CD-R disk.
- S programem Nero Burning ROM™ zvolte položku File, Burn CD image. Nastavte typ souborů na *. * a vyberte ISO soubor. Starší verze programu Nero™ vám oznámí že nerozpoznali formát - tento dialog potvrďte, protože ho sice zná ale ještě o tom neví :) V dalším dialogu zadejte následující parametry:
 - Type of image: Data Mode 1
 - Block size: 2048 bytes
 - File precursor and length of the image trailer: 0 bytes
 - Scrambled: no
 - Swapped: no

Nyní klikněte na OK a poté na Burn (CD-R disk).

- S programem cdrecord™ jednoduše zadejte **cdrecord dev=/dev/hdc** (cestu /dev/hdc zaměňte za cestu ke své CD-RW mechanice) a doplňte cestu k ISO souboru :)
- S Mac OS X Panther™ spusťte Disk Utility z Applications/Utilities, zvolte položku Open z menu Images, zvolte připojený obraz disku v hlavním okně a poté zvolte Burn v menu Images.
- S Mac OS X Jaguar™ spusťte Disk Copy z Applications/Utilities, zvolte položku Burn Image z menu Files, zvolte ISO soubor a klikněte na tlačítko Burn.

Následující podkapitoly vysvětlují jak nabootovat LiveCD specifická pro jednotlivé kapitoly. Ujistěte se že jste si vybrali subverzi určenou pro vaši architekturu!

- Bootování LiveCD na x86 nebo AMD64
- Bootování LiveCD na Alpha
- Bootování LiveCD na HPPA
- Bootování LiveCD na SPARC
- Bootování LiveCD na PPC

2.3.2. Bootování LiveCD na x86 nebo AMD64

Poznámka

Tato část je určena *pouze* pro architektury x86 a amd64. Přečtěte si ji celou než budete pokračovat dále, protože ne vždy budete mít dostatek času podívat se do manuálu.

Pokud už máte vypálena instalační CD, přišel čas nastartovat z nich systém. Restartujte počítač a vstupte do BIOSu. Toho obvykle docílíte stiskem kláves **DEL**, **F1** nebo **ESC** (která klávesa to je závisí na BIOSu který je ve vašem počítači použit). V BIOSu změňte pořadí zařízení ze kterých se má bootovat tak aby byla CD-ROM mechanika před pevným diskem. Toto nastavení se často nachází v submenu "CMOS Setup". Pokud toto neuděláte, nastartuje váš systém přímo z pevného disku a mechaniku CD-ROM bude ignorovat.

Nyní vložte instalační CD do CD-ROM mechaniky (duh) a restartujte počítač. Po chvíli byste měli vidět povedenou bootovací obrazovku s logem Gentoo Linuxu. Nyní můžete stisknout **Enter** čímž spustíte bootovací proces s přednastavenými hodnotami, nebo můžete nabootovat LiveCD s libovolnými volbami pokud uvedete volbu kernelu následovanou volbami a **Enter** stisknete až poté.

Uvedení volby kernelu? Ano, na našich LiveCD k dispozici vám dáváme několik kernelů. Výchozí volba je `gentoo`. Další volby jsou `smp`, což je jádro s podporou víceprocesorových systémů a `memtest`, který otestuje vaši RAM paměť (ale z tohoto jádra nelze Gentoo nainstalovat).

V následující tabulce najdete krátký přehled dostupných jader.

Kernel	Popis
gentoo	Výchozí jádro s podporou pro framebuffer
smp	Jádro s podporou víceprocesorových systémů
memtest	Utilita pro testování paměti

Poznámka

Některá LiveCD poskytují další jádra, nebo neposkytují jádra uvedená v tomto dokumentu. Pokud chcete vypsát seznam podporovaných jader, zvolte po zobrazení bootovací obrazovky klávesu **F1**.

Jak už bylo uvedeno, můžete také používat různá nastavení kernelu. Jedná se o nepovinné volby které můžete zapnout či vypnout dle vlastního úsudku. Následující tabulka vysvětluje dostupné volby.

Volba	Popis
acpi	zapnutí podpory ACPI
doataraid	zapnutí podpory pro ATA RAID zařízení
dofirewire	zapnutí podpory pro FireWire zařízení
dokeymap	uživatel bude dotázán na nastavení klávesnice (výchozí: <code>us</code>)
dopcmcia	zapnutí podpory pro PCMCIA
doscsi	zapnutí podpory pro SCSI
noapm	vypnutí podpory pro APM

Volba	Popis
nodetect	vypnutí automatické detekce hardware (kudzu/hotplug)
nodhcp	nepoužívat DHCP pro získání IP adresy
noevms	vypnutí podpory pro EVMS
nofb	vypnutí podpory pro framebuffer
nohotplug	vypnutí hotplug (kernel loading program)
nousb	vypnutí podpory USB
ide=nodma	vypnutí podpory DMA
cdcache	načtení potřebné části CD do paměti, což umožňuje odpojit CD a použít během instalace i další CD

Poznámka

Některá LiveCD poskytují další bootovací volby nebo neposkytují některé z těch právě uvedených. Seznam dostupných voleb získáte stisknutím klávesy **F2** na bootovací obrazovce.

Nyní nastartujte z CD, vyberte jádro (pokud vám nevyhovuje výchozí jádro `gentoo`) a bootovací volby. Jako příklad vám ukážeme jak nastartovat jádro `gentoo` s volbami `dopcmcia` a `cdcache`:

Obrázek 2.1. Start jádra s volbami `dopcmcia` a `cdcache`

```
boot: gentoo dopcmcia cdcache
```

Poté se objeví další obrazovka s progressbarem (znázorněním postupu instalace). Jakmile startovací proces skončí, budete automaticky přihlášení do "Live" Gentoo Linuxu jako "root", čili super user. Na aktuální konzoli by se měla zobrazit rootovská příkazová řádka ("`#`") a měli byste mít možnost přepnout se i do jiných konzol stiskem **Alt-F2**, **Alt-F3** a **Alt-F4**. Vraťte se do první konzole stiskem **Alt-F1**.

Nyní pokračujte v části Konfigurace dalšího hardware.

2.3.3. Bootování LiveCD na HPPA

Poznámka

Tato část je určena *pouze* pro architekturu hppa. Přečtěte si ji celou než budete pokračovat dále, protože ne vždy budete mít dostatek času podívat se do manuálu.

Nastartuje svůj HPPA systém, a během bootovacího procesu uvidíte na obrazovce zprávu podobnou té následující:

Obrázek 2.2. Zpráva při startu HPPA systému

```
Searching for Potential Boot Devices.  
To terminate search, press and hold the ESCAPE key.
```

Až se tato zpráva objeví, stiskněte klávesu Esc a držte ji dokud se neobjeví menu s volbami. Chvilku to trvá, tak buďte trpěliví. V nejčastějším případě byste měli vstoupit do BOOT_ADMIN konzole. Pokud se objevíte v menu s volbami, zvolte položku Enter Boot Administration mode pro vstup do BOOT_ADMIN konzole. Měl by se objevit příkazová řádka '>'.

Vložte Gentoo LiveCD do CD-ROM mechaniky. Pokud neznáte SCSI ID vaší CD-ROM mechaniky, vaše PA-RISC stanice ji začne hledat po zadání následujícího příkazu.

Obrázek 2.3. Hledání SCSI ID

```
> search  
Searching for Devices with Bootable Media.  
To terminate search, please press and hold the ESCAPE key.
```

Poté vaše PA-RISC stanice zobrazí seznam dostupných startovacích médií. Zde je příklad výsledku předchozího příkazu:

Obrázek 2.4. Dostupná startovací média

Device Selection	Device Path	Device Type and Utilities
P0	scsi.5.0	TOSHIBA CD-ROM XM-3301TA IPL
P1	scsi.2.0	COMPAQ ST32550N IPL
P2	lan.0010a7-06d1b6.3.6	server IPL

Pro start z CD-ROM mechaniky potřebujete znát zobrazenou hodnotu Device Path. Například, pokud budeme chtít nastartovat z CD-ROM mechaniky Toshiba, zadáme následující příkaz:

Obrázek 2.5. Start z CD-ROM mechaniky

```
> boot scsi.5.0 ip1  
  
Trying scsi.5.0
```

Klíčové slovo *ipl* (Initial Program Loader) znamená pro palo (PA-RISC boot LOader) že má vstoupit do interaktivního módu. To vám umožňuje například změnit startovací parametry.

Po úspěšném bootu bude palo spuštěn v interaktivním módu:

Obrázek 2.6. Interaktivní mód PALO

```
Boot path initialized.
Attempting to load IPL.

Hard booted.
palo ipl 1.2 root@b1801.da-kot Tue Apr  8 12:43:07 CEST 2003

Boot image contains:
    0/vmlinux32 4028015 bytes @ 0x1520000
    0/ramdisk 834748 bytes @ 0xf800
Current command line:
0/vmlinux initrd=initrd.gz TERM=linux console=tty root=/dev/ram0 init=/linuxrc
0: 0/vmlinux
1: initrd=initrd.gz
2: TERM=linux
3: console=tty
4: root=/dev/ram0
5: init=/linuxrc

Edit which field?
(or 'b' to boot with this command line)?
```

Tyto parametry ve většině případů postačují.

Pokud potřebujete další vlastnosti, musíte doplnit potřebná slova na konec příkazové řádky. Pro přidání slova stačí upravit poslední položku, doplnit mezeru a zadat nové klíčové slovo. Jediné další aktuálně implementované klíčové slovo je *cdcache* které LiveCD říká že se jeho obsah má načíst do paměti, což vám umožní CD odpojit.

Obrázek 2.7. Doplnění startovací volby cdcache

```
(or 'b' to boot with this command line)? 5
init=/linuxrc cdcache
```

Nyní když už jste upravili startovací parametry, nainstalujte.

Obrázek 2.8. Start jádra

```
(or 'b' to boot with this command line)? b
```

Na aktuální konzoli by se měla zobrazit rootovská příkazová řádka ("**#**") a měli byste mít možnost přepnout se i do jiných konzol stiskem **Alt-F2**, **Alt-F3** a **Alt-F4**. Vraťte se do první konzole stiskem **Alt-F1**.

Nyní pokračujte v části Konfigurace dalšího hardware.

2.3.4. Bootování LiveCD na SPARC

Poznámka

Tato část je určena *pouze* pro architekturu sparc. Přečtěte si ji celou než budete pokračovat dále, protože ne vždy budete mít dostatek času podívat se do manuálu.

Vložte Gentoo LiveCD do mechaniky a spusťte systém. Během startu stiskněte Stop-A pro vstup do OpenBootPROM (OBP). Jakmile jste v OBP, nastartujte z CD-ROM mechaniky.

Obrázek 2.9. Start z CD-ROM

```
ok boot cdrom
```

Privítá vás SILO boot manager (na LiveCD). Zadejte **gentoo** (pro jednoprocessorový stroj) nebo **smp** (pro víceprocesorový stroj) a stiskněte **Enter** pro pokračování startu. V následujícím příkladu nastartujeme **gentoo** jádro.

Obrázek 2.10. Start gentoo jádra

```
boot: gentoo
```

Jakmile LiveCD nastartuje, přivítá vás přihlašovací řádek. Přihlašte se jako **root**. Není potřeba žádné heslo, takže až o něj budete požádáni stiskněte **Enter**.

Obrázek 2.11. Přihlášení do LiveCD

```
login: root  
password: (Press Enter here)
```

Na aktuální konzoli by se měla zobrazit rootovská příkazová řádka ("**#**") a měli byste mít možnost pře-

pnout se i do jiných konzol stiskem **Alt-F2**, **Alt-F3** a **Alt-F4**. Vraťte se do první konzole stiskem **Alt-F1**.

Nyní pokračujte v části Konfigurace dalšího hardware.

2.3.5. Bootování LiveCD na PPC

Poznámka

Tato část je určena *pouze* pro architekturu ppc. Přečtěte si ji celou než budete pokračovat dále, protože ne vždy budete mít dostatek času podívat se do manuálu.

Vložte LiveCD do mechaniky CD-ROM a restartujte systém. Během startu podržte klávesu **C** (nebo spustíte OldWorld zavaděč jako jsou například BootX nebo quik). Objeví se uvítání a ve spodní části obrazovky příkazová řádka *boot*:

V tuto chvíli stiskněte **Enter**, a z CD disku se načte kompletní prostředí Gentoo Linux. Pokud se během startu objeví problémy, zadejte při startu volbu `-safe`. Tato volba předá jádru následující parametry: `append="video=ofonly nol3 init=/linuxrc"`.

Až LiveCD nastartuje, objeví se přivítání a přihlašovací řádek. Přihlašte se jako `root` a ponechte heslo prázdné (tj. stiskněte **Enter**).

Obrázek 2.12. Přihlášení do LiveCD

```
login: root
password: (Press Enter here)
```

Na aktuální konzoli by se měla zobrazit rootovská příkazová řádka ("`#`") a měli byste mít možnost přepnout se i do jiných konzol stiskem **Alt-F2**, **Alt-F3** a **Alt-F4**. Vraťte se do první konzole stiskem **Alt-F1**.

Pokud instalujete Gentoo na systém s neamerickou (non-US) klávesnicí, použijte příkaz **loadkeys** pro načtení mapy vaší klávesnice. Pro zobrazení dostupných map použijte příkaz **ls /usr/share/keymaps**.

Obrázek 2.13. Zobrazení seznamu map klávesnice

```
# ls /usr/share/keymaps
```

Pro načtení mapy použijte příkaz **loadkeys**.

Obrázek 2.14. Zobrazení seznamu map klávesnice

```
# loadkeys be2-latin1
```

Nyní pokračujte v části Konfigurace dalšího hardware.

2.3.6. Konfigurace dalšího hardware

Poté co LiveCD nastartuje, pokusí se rozpoznat hardwarová zařízení nainstalovaná ve vašem počítači a načíst příslušné moduly jádra. V naprosté drtivé většině případů odvede velice dobrou práci. Avšak v některých případech se nemusí povést automaticky načíst potřebné moduly. Pokud PCI auto-detekce něco nerozpozná, budete muset příslušné moduly do jádra načíst ručně.

V dalším příkladě se pokusíme načíst modul 8139too (podporující některé druhy síťových rozhraní).

Obrázek 2.15. Načtení jaderných modulů

```
# modprobe 8139too
```

2.3.7. Nepovinné: Ladění výkonu vašeho pevného disku

Pokud jste zkušený uživatel, možná si chcete pohrát s nastavením pevného IDE disku za použití programu **hdparm**. Za použití volby `-tT` můžete otestovat výkon vašeho disku (spusťte jej několikrát po sobě pro přesnější výsledky):

Obrázek 2.16. Testování výkonu pevného disku

```
# hdparm -tT /dev/hda
```

Při ladění se můžete nechat inspirovat následujícími příklady (nebo experimentovat) které používají zařízení `/dev/hda` (nahraďte jej vaším zařízením).

Obrázek 2.17. Ladění výkonu pevného disku

```
Zapnutí DMA: # hdparm -d 1 /dev/hda
Zapnutí DMA + Bezpečné volby zvyšující výkon: # hdparm -d 1 -A 1 -m 16 -u 1 -a 64 /dev/hda
```

2.3.8. Nepovinné: Uživatelské účty

Pokud plánujete že umožníte dalším lidem přístup k instalačnímu prostředí nebo pokud chcete používat `irssi` chat bez rootovských práv (z bezpečnostních důvodů), potřebujete vytvořit příslušné uživatelské účty a změnit heslo uživatele `root`.

Pro změnu hesla uživatele root spustíte příkaz **passwd**:

Obrázek 2.18. Změna hesla uživatele root

```
# passwd
New password: (Enter your new password)
Re-enter password: (Re-enter your password)
```

Pokud chcete vytvořit nový uživatelský účet, zadejte nejdříve osobní informace následované heslem. Pro tyto účely používáme příkazy **useradd** a **passwd**. V dalším příkladu vytvoříme uživatele s uživatelským jménem "john".

Obrázek 2.19. Vytváření uživatelského účtu

```
# useradd john
# passwd john
New password: (Enter john's password)
Re-enter password: (Re-enter john's password)
```

Z rootovského účtu se do účtu nově vytvořeného uživatele přepnete pomocí příkazu **su**:

Obrázek 2.20. Změna uživatelského účtu

```
# su john -
```

2.3.9. Nepovinné: Spuštění SSH Démona

Poznámka

Pouze architektury x86, amd64, alpha, sparc a ppc poskytují možnost nastartovat ssh démona během instalace.

Pokud chcete dalším uživatelům umožnit přístup k vašemu počítači během instalace Gentoo Linuxu (například pokud vám tyto uživatelé budou pomáhat s instalací nebo ji pro vás dokonce budou provádět samostatně), potřebujete pro ně vytvořit uživatelský účet, nebo jim poskytnout svoje rootovské heslo (to ale dělejte pouze pokud těmto osobám plně důvěřujete).

Pro nastartování SSH démona spustíte následující příkaz:

Obrázek 2.21. Spuštění SSH démona

```
# /etc/init.d/sshd start
```

Nyní pokračujte v kapitole Konfigurace vaší sítě.

2.4. Náhradní: Stáhněte, vypalte a spust'te Knoppix

Poznámka

Knoppix lze používat pouze na systémech s architekturou x86.

Knoppix si můžete stáhnout z domácí stránky Knoppixu [<http://www.knoppix.com/>], kde je dostupný ve formě ISO souborů. Až ho stáhnete, vypalte ho na CD-R, přičemž musíte použít "raw" pálení. Nyní vám vysvětlíme jak to udělat s několika velmi populárními nástroji.

- Pokud používáte program EasyCD Creator™, zvolte položku File, Record CD from CD image. Poté změňte položku Files of type na *ISO image file*. Najděte stažený ISO soubor a klikněte na Open. Až kliknete na tlačítko Start burning bude ISO obraz vypálen přímo na vložený CD-R disk.
- S programem Nero Burning ROM™ zvolte položku File, Burn CD image. Nastavte typ souborů na *. * a vyberte ISO soubor. Starší verze programu Nero™ vám oznámí že nerozpoznali formát - tento dialog potvrďte, protože ho sice zná ale ještě o tom neví :) V dalším dialogu zadejte následující parametry:
 - Type of image: Data Mode 1
 - Block size: 2048 bytes
 - File precursor and length of the image trailer: 0 bytes
 - Scrambled: no
 - Swapped: no

Nyní klikněte na OK a poté na Burn (CD-R disk).

- S programem cdrecord™ jednoduše zadejte **cdrecord dev=/dev/hdc** (cestu /dev/hdc zaměňte za cestu ke své CD-RW mechanice) a doplňte cestu k ISO souboru :)

Knoppix implicitně startuje do desktopového prostředí KDE 3.0. První věc kterou budete muset udělat je nastavení rootovského hesla pro Knoppix. Otevřete konzoli a zadejte následující příkaz:

Obrázek 2.22. Změna rootovského hesla pro Knoppix

```
$ sudo passwd root
Password: (Enter a password)
Re-enter password: (Re-enter the password)
```

Dále se pomocí příkazu **su** přepněte na uživatele root a změnit domovský adresář na `/root`. Pokud toto neuděláte, budete posléze překvapeni varováními o tom že `"/home/root/ not found"`.

Obrázek 2.23. Změna domovského adresáře pro uživatele root

```
$ su -
Password: (Enter the root password created above)
# usermod -d /root -m root
```

Nyní opusťte aktuální shell zadáním příkazu **exit** a znovu se pomocí příkazu **su** přepněte na roota. Tím načtete provedené změny.

Obrázek 2.24. Načtení provedených změn

```
# exit
$ su -
Password: (Enter the root password)
```

Nyní pokračujte v kapitole Konfigurace vaší sítě.

2.5. Náhradní: Start přes síť

Abyste mohli Gentoo nainstalovat, musí být v image staženém přes TFTP server dostupné potřebné nástroje pro vytváření souborových systémů, vytváření a připojování diskových oddílů, práci s balíčky vytvořenými programem **tar** a program **chroot**. Existující net boot image, obsahující vše potřebné, si můžete stáhnout ...

V tuto chvíli víme o jediné architektuře pro kterou jsou net boot image dostupné, a to je MIPS architektura.

- Start přes síť s MIPS systémem

2.5.1. Start přes síť s MIPS systémem

Nejdříve si stáhněte dostupné net boot image z adresy <http://dev.gentoo.org/~kumba/mips/netboot/>

Nyní nakonfigurujte svůj DHCP server tak aby tento soubor zaslal startujícímu klientovi. SGI stroje nicméně potřebují nějaké malé úpravy na straně serveru aby TFTP pracovalo správně:

Obrázek 2.25. Několik drobných úprav pro SGI stroje, aby TFTP pracovalo správně

```
(Disable "Path Maximum Transfer Unit", otherwise SGI Prom won't find the kernel)
# echo 1 > /proc/sys/net/ipv4/ip_no_pmtu_disc
(Set the port range usable by the SGI Prom)
# echo "2048 32767" > /proc/sys/net/ipv4/ip_local_port_range
```

Nyní zapněte svůj stroj, přejděte do PROM monitoru a zadejte příkaz pro nastartování jádra přes síť:

Obrázek 2.26. Start přes síť pro MIPS systém

```
Running power-on diagnostics

System Maintenance Menu

1) Start System
2) Install System Software
3) Run Diagnostics
4) Recover System
5) Enter Command Monitor

Option? 5
Command Monitor. Type "exit" to return to the menu.
>> bootp(): root=/dev/ram0
```

Poznámka

Možná budete muset stisknout klávesu Esc abyste se dostali do menu uvedeného výše, jinak se systém pokusí nabootovat jako obvykle přímo.

Někdy je start přes síť trnitá cesta. Pokud narazíte na problémy mohly by vám následující PROM příkazy pomoci, ale zaručit to nelze. Pokud vaše stanice odmítne přes síť nastartovat, dvakrát zkontrolujte nastavení TFTP serveru abyste se ujistili že:

- dhcpd přiděluje SGI stanici IP adresu

- přístupová práva jsou ve vaší tftp složce (typicky /tftpboot) nastavena správně
- zkontrolujte systémové logy abyste zjistili co tftp server hlásí (pravděpodobně nějaké chyby)
- modlete se k plyšákovi Tux (toto může ale nemusí fungovat, a nejedná se o oficiálně podporovanou techniku lokalizace a odstranování závad)

Obrázek 2.27. Triky které se vám při startu přes síť možná budou hodit

```
>> resetenv  
>> unsetenv netaddr  
>> unsetenv dlserver  
>> init  
>> bootp(): root=/dev/ram0
```

Pokud se vše podařilo, jste teď v ash příkazové řádce. V tom případě můžete pokračovat v kapitole Konfigurace vaší sítě.

Kapitola 3. Konfigurace vaší sítě

Abstrakt

Pokud chcete Gentoo instalovat přes Internet, potřebujete správně nastavenou síť.

3.1. Můžete instalovat i bez sítě, ale ...

3.1.1. Kdo může instalovat bez sítě?

V závislosti na tom jaké instalační médium jste si pro instalaci Gentoo Linuxu vybrali můžete nebo nemůžete pokračovat bez funkčního připojení k síti (a Internetu).

Obecně vzato budete potřebovat nastavit připojení k síti (a Internetu). Ale Gentoo také poskytuje možnost instalace bez síťového připojení. To je však možné pouze s následujícími instalačními médii:

- Gentoo 2-CD LiveCD sada pro architekturu x86 (včetně architektur x86, i686, pentium3, pentium4 a athlon-xp)
- Gentoo 2-CD LiveCD sada pro architekturu PPC

3.1.2. Proč potřebuji připojení k síti?

Výsledkem instalace Gentoo Linuxu přes Internet je plně aktualizovaný Gentoo systém. Máte instalaci založenou na nejnovějším Portage stromě (což je kolekce balíčků které poskytujeme společně s nástroji na správu vašeho software). To je také důvod proč je síťová instalace preferována. Avšak někteří lidé nemohou nebo nechťejí instalovat Gentoo na systém s fungujícím připojením na Internet.

Pokud jste v takové situaci potom budete potřebovat sadu 2 CD. Na těchto sadách je dostupný zdrojový kód, Portage strom a nástroje potřebné k instalaci Gentoo základního systému, a tak dále. Cenou za tuto volbu je že nebudete mít nejnovější verze programů, ačkoliv rozdíly budou minimální.

Pokud tedy chcete využít možnost instalace bez sítě, musíte použít tuto 2-CD sadu, přeskóčit zbytek této kapitoly a pokračovat v kapitole Příprava disků. V opačném případě pokračujte dále ve čtení této kapitoly.

3.1.3. Nepovinné: konfigurace proxy

Pokud k Internetu přistupujete přes proxy server, budete možná potřebovat zadat informace o něm během instalace. Je velice jednoduché nadefinovat proxy - potřebujete totiž pouze nastavit proměnnou obsahující tyto informace.

Ve většině případů stačí pouze nadefinovat proměnné za použití jména proxy serveru. Předpokládejme například, že proxy se jmenuje *proxy.gentoo.org* a číslo portu je *8080*.

Obrázek 3.1. Definice proxy serveru

```
(If the proxy filters HTTP traffic)
# export http_proxy="http://proxy.gentoo.org:8080"
(If the proxy filters FTP traffic)
# export ftp_proxy="ftp://proxy.gentoo.org:8080"
(If the proxy filters RSYNC traffic)
# export rsync_proxy="rsync://proxy.gentoo.org:8080"
```

Pokud váš proxy server vyžaduje uživatelské jméno a heslo, měli byste pro proměnné použít následující syntaxi:

Obrázek 3.2. Doplnění jména/hesla do nastavení proxy serveru

```
http://username:password@server
```

Například pro nastavení našeho HTTP proxy serveru s uživatelským jménem "john" a heslem "f00b_r" bychom použili:

Obrázek 3.3. Proxy server s autentizací

```
# export http_proxy="http://john:f00b_r@proxy.gentoo.org:8080"
```

3.1.4. Připojení k síti z jiných (non-Gentoo) instalačních médií

Většina informací uvedená dále v této kapitole je určena uživatelům kteří nainstalovali z Gentoo LiveCD. Nicméně Gentoo Linux můžete instalovat několika dalšími způsoby. Pokud jste v takové situaci, ujistěte se použitý způsob má fungující Internetové připojení (informace dostupné v dokumentech Používáme DHCP nebo Rozumíme síťové terminologii se vám mohou hodit) a pokračujte v kapitole Příprava disků.

3.2. Automatická detekce sítě

3.2.1. Třeba to prostě funguje?

Pokud je váš systém připojen do Ethernetové sítě s DHCP serverem, je velice pravděpodobné že vaše síťová konfigurace již byla automaticky nastavena. Pokud je tomu tak, měli byste být schopni používat

mnoho síťových nástrojů na LiveCD jako jsou mimo jiné ssh, scp, ping, irssi, wget a links.

Pokud pro vás byla síť nastavena, potom by příkaz `/sbin/ifconfig` měl kromě `lo` vypsat i některá další síťová rozhraní, jako je například `eth0`:

Obrázek 3.4. `/sbin/ifconfig` pro fungující síťovou kartu

```
# /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:BA:8F:61:7A
          inet addr:192.168.0.2  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::50:ba8f:617a/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1498792 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1284980 errors:0 dropped:0 overruns:0 carrier:0
          collisions:1984 txqueuelen:100
          RX bytes:485691215 (463.1 Mb)  TX bytes:123951388 (118.2 Mb)
          Interrupt:11 Base address:0xe800
```

3.2.2. Testování sítě

Možná budete chtít pingnout na DNS server svého providera (který se nachází v `/etc/resolv.conf`) a libovolný web server, pouze abyste se ujistili že vaše pakety dorazí do sítě, DNS funguje správně, atd. ...

Obrázek 3.5. Další testování sítě

```
# ping -c 3 www.yahoo.com
```

Můžete nyní používat svou síť? Pokud ano, přeskočte zbytek této kapitoly a pokračujte v kapitole Příprava disků. Pokud je síť nefunkční, potom máte smůlu a musíte pokračovat ještě dále.

3.3. Automatická konfigurace sítě

Pokud síť nefunguje ihned po startu, umožňují některá instalační média použití programu **net-setup** (pro klasické sítě), **adsl-setup** (pro uživatele ADSL) nebo **pptp** (pro uživatele PPTP).

Pokud vaše instalační médium tyto nástroje neobsahuje, pokračujte v podkapitole Manuální konfigurace sítě.

- uživatelé klasické Ethernetové sítě by měli pokračovat v odstavci Výchozí: Použití skriptu net-setup
- uživatelé ADSL by měli pokračovat v odstavci Náhradní: Použití RP-PPPoE
- uživatelé PPTP by měli pokračovat v odstavci Náhradní: Použití PPTP

3.3.1. Výchozí: Použití skriptu net-setup

Nejjednodušší způsob k nastavení sítě pokud není nastavena automaticky je spustit skript net-setup.

Obrázek 3.6. Spuštění skriptu net-setup

```
# net-setup eth0
```

Skript net-setup vám položí několik otázek o vašem síťovém prostředí, a po jeho skončení byste měli mít k dispozici funkční připojení k síti. Otestujte připojení jak bylo předvedeno výše, a pokud je test úspěšný tak gratulujeme! Nyní jste konečně připraveni na instalaci Gentoo. Přeskočte zbytek této kapitoly a pokračujte v kapitole Příprava disků.

Pokud vaše síťové spojení stále nefunguje jak by mělo, pokračujte v podkapitole Manuální konfigurace sítě.

3.3.2. Náhradní: Použití RP-PPPoE

Za předpokladu že potřebujete PPPoE pro připojení k Internetu, LiveCD (libovolná verze) vám celý proces zprovoznění sítě usnadňuje protože obsahuje **rp-pppoe**. Použijte skript **adsl-setup** ke konfiguraci vašeho spojení. Budete dotázáni na ethernetové zařízení připojené k vašemu adsl modemu, vaše jméno a heslo, IP adresy vašich DNS serverů a na to zda potřebujete základní firewall nebo ne.

Obrázek 3.7. Použití rp-pppoe

```
# adsl-setup  
# adsl-start
```

Pokud něco nefunguje, zkontrolujte dvakrát zda jste zadali správné jméno a heslo (podívejte se do /etc/ppp/pap-secrets nebo /etc/ppp/chap-secrets) a ujistěte se že jste zadali správné zařízení. Pokud vaše internetové zařízení neexistuje, budete do jádra muset zavést příslušné moduly. V tom případě byste měli pokračovat v podkapitole Manuální konfigurace sítě tam vysvětlujeme jak do jádra potřebné moduly zavést.

Pokud vše funguje, pokračujte v kapitole Příprava disků.

3.3.3. Náhradní: Použití PPTP

Pokud potřebujete podporu pro PPTP, můžete použít skript pptpclient, který je k dispozici na našich LiveCD. Ale nejdříve se ujistíme že vaše konfigurace je správná. Upravte soubor /etc/ppp/pap-secrets nebo /etc/ppp/chap-secrets tak aby obsahoval správnou kombinaci uživatelského jména a hesla:

Obrázek 3.8. Úprava souboru `/etc/ppp/chap-secrets`

```
# nano -w /etc/ppp/chap-secrets
```

Poté upravte `/etc/ppp/options.pptp` pokud je to nutné:

Obrázek 3.9. Úprava souboru `/etc/ppp/options.pptp`

```
# nano -w /etc/ppp/options.pptp
```

Až toto uděláte, spusťte jednoduše **pptp** (společně s volbami které jste nemohli nastavit v `options.pptp`) pro připojení k serveru:

Obrázek 3.10. Připojení k serveru přes telefonní linku

```
# pptp <server ip>
```

Nyní pokračujte v kapitole Příprava disků.

3.4. Manuální konfigurace sítě

3.4.1. Načtení potřebných modulů do jádra

Až LiveCD nastartuje, pokusí se určit veškeré vaše hardwarové vybavení a načíst potřebné moduly (ovladače) do jádra tak aby nalezený hardware fungoval. V naprosté většině případů systém odvede výbornou práci. Avšak v některých případech nemusí být všechny potřebné moduly automaticky načteny.

Pokud některý ze skriptů **net-setup** nebo **adsl-setup** selhal, je téměř jisté že vaše síťová karta nebyla automaticky rozpoznána. To znamená že budete muset požadované moduly do jádra zavést ručně.

Abyste zjistili jaké moduly naše jádro nabízí použijte příkaz **ls**:

Obrázek 3.11. Hledání dostupných modulů

```
# ls /lib/modules/`uname -r`/kernel/drivers/net
```

Pokud najdete modul (ovladač) pro vaši síťovou kartu, použijte pro jeho zavedení do jádra příkaz **modprobe**:

Obrázek 3.12. Testování dostupnosti vaší síťové karty, úspěšné

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr FE:FD:00:00:00:00
          BROADCAST NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Avšak pokud narazíte na následující chybu, nebyla síťová karta nalezena:

Obrázek 3.13. Testování dostupnosti vaší síťové karty, neúspěšné

```
# ifconfig eth0
eth0: error fetching interface information: Device not found
```

Za předpokladu že byla vaše síťová karta nalezena, můžete znovu spustit skript net-setup nebo adsl-setup (který by nyní měl fungovat), ale pro drsnáky teď vysvětlíme jak nakonfigurovat síťovou kartu ručně.

V zásadě existují dvě možnosti. Bud použijete DHCP (automatické získání IP adresy), nebo ručně nastavíte síťové parametry pomocí příkazů **ifconfig** a **route**.

3.4.2. Používáme DHCP

DHCP (Dynamic Host Configuration Protocol) umožňuje automaticky přijímat informace o konfiguraci sítě (IP adresu, síťovou masku, broadcast adresu, gateway, nameservery, atd.). To funguje pouze pokud máte ve své síti DHCP server DHCP (nebo pokud má DHCP server váš provider). Aby vaše síťová karta přijímala tyto informace automaticky, použijte **dhcpcd**.

Obrázek 3.14. Použití dhcpcd

```
# dhcpcd eth0
```

Pokud toto funguje (zkuste pingnout na některý internetový server, například na Google), potom je nastavení hotovo a jste připraveni pokračovat v instalaci kapitolou Příprava disků.

3.4.3. Rozumíme síťové terminologii

Poznámka

Pokud znáte svou IP adresu, broadcast adresu, síťovou masku a nameservery, potom můžete přeskočit tuto část a pokračovat v Používáme ifconfig a route.

Pokud všechny výše uvedené postupy selžou, budete muset nastavit vaši síť ručně. Nebojte se, zdaleka to není těžké. Ale nejdříve vám musíme vysvětlit určité množství síťových pojmů které budete potřebovat pro uspokojivou konfiguraci vaší sítě. Až toto dočtete, budete vědět co je to gateway, k čemu je síťová maska, jak je vytvářena broadcast adresa a proč potřebujete nameservery.

Počítače jsou v síti identifikováni svými IP adresami (Internet Protocol adresami). Tyto adresy jsou vlastně kombinacemi čtyř číslic mezi 0 a 255, Tedy, tak to alespoň vnímáme my. Ve skutečnosti se IP adresa skládá z 32 bitů (jedniček a nul). Podívejme se na názorný příklad:

Obrázek 3.15. Příklad IP adresy

IP Address (numbers):	192.168.0.2
IP Address (bits):	11000000 10101000 00000000 00000010

	192 168 0 2

Taková IP adresa je unikátní pro každý počítač pokud uvažujeme dostupnou síť (tj. všechny počítače dostupné ve vaší síti musí mít unikátní IP adresy). Abychom byli schopni rozlišovat mezi uživateli uvnitř lokální sítě a mimo ni, je IP adresa rozdělena na dvě části - na síťovou (network) a uživatelskou (host) část.

Toto rozdělení je zapsáno ve formě síťové masky (netmask), posloupnosti jedniček následované posloupností nul. Část IP adresy která odpovídá jedničkám je síťová část, část odpovídající nulám je uživatelská část. Síťová maska je obvykle zapsána podobným způsobem jako IP adresa:

Obrázek 3.16. Rozdělení IP adresy na síťovou a uživatelskou část

IP-address:	192	168	0	2
	11000000	10101000	00000000	00000010
Netmask:	11111111	11111111	11111111	00000000
	255	255	255	0
	+-----+-----+			
	Network			Host

Jinými slovy, IP adresa 192.168.0.14 je stále ještě částí naší sítě, ale 192.168.1.2 již nikoliv.

Broadcast (všesměrová) adresa je IP adresa se stejnou síťovou částí jako má naše síť, ale pouze s jedničkami na místě uživatelské části. Každý uživatel v naší síti na této adrese poslouchá. Jejím skutečným účelem je všesměrové vysílání (broadcast) paketů.

Obrázek 3.17. Broadcast (všesměrová) adresa

IP-address:	192	168	0	2
	11000000	10101000	00000000	00000010
Broadcast:	11000000	10101000	00000000	11111111
	192	168	0	255
	+-----+-----+			+
	Network			Host

Abyste byli schopni surfovat po internetu, musíte vědět s kým sdílíte spojení k webu. Tento počítač se jmenuje gateway. Protože se jedná o obyčejný počítač, má obyčejnou IP adresu (například 192.168.0.1).

Jak už jsme dříve uvedli, každý počítač má vlastní IP adresu. Abychom se k počítači dostali pomocí jména (namísto IP adresy), potřebujeme službu která takové jméno (jako je například dev.gentoo.org) přeloží na IP adresu (jako je například 64.5.63.82). Tato služba se jmenuje name service (jmenná služba). Abyste tuto službu mohli využívat, musíte v souboru `/etc/resolv.conf` zadat potřebné jmenné servery.

V některých případech slouží vaše gateway současně i jako nameserver (jmenný server). Jinak budete muset zadat jmenné servery poskytované vaším ISP.

Shrňme to - než budete pokračovat, budete potřebovat následující informace:

Síťová položka	Příklad
IP adresa	192.168.0.2
Síťová maska	255.255.255.0
Všesměrová adresa (broadcast)	192.168.0.255
Gateway	192.168.0.1
Jmenné servery (nameservy)	195.130.130.5, 195.130.130.133

3.4.4. Používáme ifconfig a route

Nastavení sítě spočívá ve třech krocích. Nejdříve si pomocí příkazu **ifconfig** přiřadíme IP adresu. Poté nastavíme routování na gateway pomocí příkazu **route**. Nakonec umístíme IP adresy jmenných serverů do `/etc/resolv.conf`.

Pro přiřazení IP adresy budete potřebovat IP adresu, všesměrovou (broadcast) adresu a síťovou masku (netmask). Poté spusťte následující příkaz, přičemž nahradíte `${IP_ADDR}` za IP adresu, `${BROADCAST}` za všesměrovou adresu a `${NETMASK}` za síťovou masku:

Obrázek 3.18. Použití příkazu `ifconfig`

```
# ifconfig eth0 ${IP_ADDR} broadcast ${BROADCAST} netmask ${NETMASK} up
```

Nyní nastavte routování pomocí příkazu `route`. Zaměňte `${GATEWAY}` za IP adresu vaší brány (gateway):

Obrázek 3.19. Použití příkazu `route`

```
# route add default gw ${GATEWAY}
```

Nyní otevřete soubor `/etc/resolv.conf` svým oblíbeným editorem (v našem příkladě je použit editor **nano**):

Obrázek 3.20. Vytvoření souboru `/etc/resolv.conf`

```
# nano -w /etc/resolv.conf
```

Doplňte IP adresy jmenných serverů, přičemž následující příklad použijte jako šablonu. Samozřejmě nahraďte `${NAMESERVER1}` a `${NAMESERVER2}` za příslušné IP adresy jmenných serverů:

Obrázek 3.21. Šablona souboru `/etc/resolv.conf`

```
nameserver ${NAMESERVER1}
nameserver ${NAMESERVER2}
```

Hotovo. Nyní otestujte svou síť například pingnutím na některý z Internetových serverů (například Google). Pokud to funguje, gratulujeme. Nyní jste připraveni k instalaci Gentoo. Pokračujte kapitolou Příprava disků.

Kapitola 4. Příprava disků

Abstrakt

Abyste vůbec byli schopni Gentoo nainstalovat, musíte vytvořit potřebné diskové oddíly. Tato část popisuje postup jak před dalším postupem rozdělit disk.

4.1. Úvod do blokových zařízení

4.1.1. Bloková zařízení

Pořádně se podíváme na vlastnosti Gentoo Linuxu a obecně Linuxu související s disky, včetně Linuxových souborových systémů, diskových oddílů a blokových zařízení. Poté, jakmile budete rozumět vstupům a výstupům disků a souborových systémů, vás provedeme postupem vytvoření diskových oddílů a souborových systémů pro vaši instalaci Gentoo Linuxu.

Začneme představením blokových zařízení. Nejslavnější z nich je pravděpodobně to které reprezentuje první IDE disk v Linuxovém systému, čili `/dev/hda`. V případě že váš systém používá SCSI disky je první disk označen `/dev/sda`.

Bloková zařízení uvedená výše reprezentují abstraktní rozhraní k diskům. Uživatelské programy mohou tato bloková zařízení používat k práci s vaším diskem aniž by se musela starat zda se jedná o IDE disk, SCSI disk nebo cokoli jiného. Program se může na disk obrátit jednoduše jako na hromadu souvislých, náhodně přístupných 512-bytových bloků.

4.1.2. Diskové oddíly a řezy (slices)

Ačkoliv je teoreticky možné použít pro váš Linuxový systém celý pevný disk, téměř nikdy se to nedělá. Namísto toho je celý disk rozdělen na menší, lépe spravovatelná, bloková zařízení která se na většině systémů nazývají diskové oddíly (partitions). Jiné architektury používají podobnou techniku, nazývanou řezy (slices).

4.1.3. Diskové oddíly

Diskové oddíly lze rozdělit na tři základní typy: primární, rozšířené a logické.

Primární diskový oddíl je takový diskový oddíl o kterém jsou informace uloženy v MBR (master boot record). Protože MBR je velice malý (512 bytů), lze vytvořit nejvýše čtyři primární diskové oddíly (například `/dev/hda1` až `/dev/hda4`).

Rozšířený diskový oddíl je zvláštním druhem primárního diskového oddílu (což znamená že rozšířený diskový oddíl musí být jedním ze čtyř možných primárních diskových oddílů) obsahující další diskové oddíly. Tento druh diskového oddílu původně neexistoval, ale protože maximální počet čtyř diskových oddílů byl příliš nízký, objevilo se toto řešení rozšiřující zaběhané schéma bez ztráty zpětné kompatibility.

Logický diskový oddíl je diskový oddíl uvnitř rozšířeného diskového oddílu. Jejich definice není ukládána do MBR, ale je uchovávána v rozšířeném diskovém oddíle.

4.2. Rozvržení diskových oddílů

4.2.1. Standardní rozvržení disku

Pokud si pro svůj systém nechcete navrhnout vlastní rozdělení disku, můžete použít (ne-LVM) schéma které používáme ve zbytku této knihy:

Pro x86 nebo amd64:

Diskový oddíl	Souborový systém	Velikost	Popis
/dev/hda1	ext2	32M	Startovací oddíl
/dev/hda2	(swap)	512M	Odkládací (swapovací) oddíl
/dev/hda3	ext3	zbytek disku	Kořenový adresář

Pro ppc:

Diskový oddíl	Partition dWorld	Ol-	Souborový systém	Velikost	Popis
/dev/hda1	(není třeba)		(bootstrap)	800k	Apple_bootstrap
/dev/hda2	/dev/hda1		(swap)	512M	Odkládací (swapovací) oddíl
/dev/hda3	/dev/hda2		ext3	zbytek disku	Kořenový adresář

Pro Sparc:

Diskový oddíl	Souborový systém	Velikost	Popis
/dev/hda1	ext2	32M	Startovací oddíl
/dev/hda2	(swap)	512M	Odkládací (swapovací) oddíl
/dev/hda3	(nic)	celý disk	Sund Disc Label (nutné)
/dev/hda4	ext3	zbytek disku	Kořenový adresář

Pokud chcete instalovat Gentoo z existující distribuce, měli byste nejdříve změnit velikost aktuálních diskových oddílů (pokud nemáte žádné místo volné) abyste mohli Gentoo nainstalovat. K tomu můžete použít například program GNU/Parted [<http://www.gnu.org/software/parted>].

Pokud vás zajímá jak velký má který diskový oddíl (nebo logický celek) být, nebo kolik diskových oddílů potřebujete, čtěte dále. Jinak pokračujte přímo v rozdělování disku:

- Použití programu fdisk na architektuře x86 nebo amd64 k rozdělení vašeho disku
- Použití programu fdisk na architektuře Alpha k rozdělení vašeho disku
- Použití programu fdisk na architektuře SPARC k rozdělení vašeho disku
- Použití programu mac-fdisk na architektuře PPC k rozdělení vašeho disku
- Použití programu fdisk na architektuře HPPA k rozdělení vašeho disku
- Použití programu fdisk na architektuře MIPS k rozdělení vašeho disku

4.2.2. Kolik a jak velkých?

Počet diskových oddílů silně závisí na vašem prostředí. Například, pokud máte velký počet uživatelů, budete zřejmě chtít umístit `/home` na oddělený disk abyste zvýšili bezpečnost a zjednodušili zálohování. Pokud instalujete gentoo jako mailserver, potom by zřejmě měl být umístěn adresář `/var` protože veškeré e-maily jsou ukládány právě zde. Dobré rozvržení souborového systému poté maximalizuje výkon. Gameservery budou mít samostatný adresář `/opt` protože právě sem jsou většinou umístovány herní servery. Důvody jsou stejné jako v případě `/home` - bezpečnost a zálohy.

Jak je zřejmé, velice závisí na tom čeho chcete dosáhnout. Oddělené diskové oddíly nebo disky mají následující výhody:

- Můžete si zvolit nejvýkonnejší souborový systém pro každý diskový oddíl nebo disk.
- Celému vašemu systému nemůže dojít volné místo pokud jeden špatně fungující program zaplní veškeré volné místo ve svém diskovém oddíle
- V případě nutnosti jsou kontroly integrity souborového systému redukovány, protože několik kontrol lze provádět paralelně (ačkoliv tato výhoda je zřejmější s několika oddělenými disky než diskovými oddíly)
- Bezpečnost lze zvýšit připojením některých diskových oddílů pouze ke čtení (read-only), s ignorováním setuid bitů (noisetuid), bez možnosti spouštění programů (noexec), atd.

Avšak rozdělení systému na více diskových oddílů má i své nevýhody: pokud není systém nastaven správně, skončíte se systémem který má na některých discích spoustu volného místa a na některých vůbec žádné.

Jako příklad rozdělení disku předvedeme jeden z našich 20 GB disků, použitých v předváděcím laptopu (obsahující webserver, mailserver, gnome, ...):

Obrázek 4.1. Příklad rozdělení disku

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/hda5	ext3	509M	132M	351M	28%	/
/dev/hda2	ext3	5.0G	3.0G	1.8G	63%	/home

```

/dev/hda7    ext3    7.9G  6.2G  1.3G  83% /usr
/dev/hda8    ext3   1011M  483M  477M  51% /opt
/dev/hda9    ext3    2.0G  607M  1.3G  32% /var
/dev/hda1    ext2     51M   17M   31M  36% /boot
/dev/hda6    swap    516M   12M  504M   2% <not mounted>
(Unpartitioned space for future usage: 2 Gb)

```

Diskový oddíl připojený jako `/usr` je relativně zaplněný, ale jakmile je jednou software nainstalován, `/usr` už nemá tendenci příliš růst. Mohli byste si myslet že pro `/var` je přiřazeno příliš mnoho místa. Avšak Gentoo kompiluje veškeré programy v `/var/tmp/portage`, takže pokud chcete kompilovat velké programy měli byste mít ve `/var` volný alespoň 1G, a pokud vám nečiní potíže kompilovat současně KDE i OpenOffice.org tak alespoň 3G.

Nyní rozdělte svůj pevný disk za použití následujících příkladů uvedených pro vaši architekturu:

- Použití programu `fdisk` na architektuře x86 nebo amd64 k rozdělení vašeho disku
- Použití programu `fdisk` na architektuře Alpha k rozdělení vašeho disku
- Použití programu `fdisk` na architektuře SPARC k rozdělení vašeho disku
- Použití programu `mac-fdisk` na architektuře PPC k rozdělení vašeho disku
- Použití programu `fdisk` na architektuře HPPA k rozdělení vašeho disku
- Použití programu `fdisk` na architektuře MIPS k rozdělení vašeho disku

4.3. Použití programu `fdisk` na architektuře x86 nebo amd64 k rozdělení vašeho disku

Poznámka

Tato část je určena pouze systémy s architekturou x86 nebo amd64.

Následující text vysvětluje jak dosáhnout ukázkového rozdělení disku popsaného výše, konkrétně:

Diskový oddíl	Popis
<code>/dev/hda1</code>	Startovací oddíl
<code>/dev/hda2</code>	Odkládací (swapovací) oddíl
<code>/dev/hda3</code>	Kořenový adresář

Samozřejmě toto rozdělení přizpůsobte svým představám.

4.3.1. Změna aktuálního rozdělení disku

Program **fdisk** je populární a mocný nástroj na dělení disku na jednotlivé diskové oddíly. Spustíte **fdisk** na svůj disk (v našem případě používáme `/dev/hda`):

Obrázek 4.2. Spuštění fdisk

```
# fdisk /dev/hda
```

Jakmile se ocitnete v prostředí programu **fdisk**, objeví se obrazovka podobná té následující:

Obrázek 4.3. Příkazová řádka programu fdisk

```
Command (m for help):
```

Zadejte **p**, čímž zobrazíte aktuální rozdělení disku:

Obrázek 4.4. Příklad rozdělení disku

```
Command (m for help): p

Disk /dev/hda: 240 heads, 63 sectors, 2184 cylinders
Units = cylinders of 15120 * 512 bytes

Device Boot      Start         End      Blocks   Id  System
/dev/hda1             1           14        105808+  83  Linux
/dev/hda2            15           49        264600    82  Linux swap
/dev/hda3            50           70        158760    83  Linux
/dev/hda4            71          2184       15981840     5  Extended
/dev/hda5            71           209        1050808+  83  Linux
/dev/hda6           210           348        1050808+  83  Linux
/dev/hda7           349           626       2101648+  83  Linux
/dev/hda8           627           904       2101648+  83  Linux
/dev/hda9           905          2184       9676768+  83  Linux

Command (m for help):
```

Tento konkrétní disk je nakonfigurován pro sedm různých Linuxových souborových systémů (každý odpovídá diskovému oddílu s popisem "Linux") a současně odkládacímu diskovému oddílu (ten je popsán jako "Linux swap").

4.3.2. Odstranění všech diskových oddílů

Nejdříve odstraníme všechny existující diskové oddíly z disku. Zadejte **d** pro smazání diskového oddílu. Například pro smazání oddílu `/dev/hda1`:

Obrázek 4.5. Mazání diskového oddílu

```
Command (m for help): d
Partition number (1-4): 1
```

Diskový oddíl byl označen pro smazání. Nebude se dále zobrazovat pokud stisknete **p**, ale nebude smazán dokud změny neuložíte. Pokud uděláte chybu a chcete program ukončit bez uložení změn, stiskněte klávesu **q** a poté **Enter**, a vaše diskové oddíly nebudou smazány.

Předpokládejme že jste chtěli smazat všechny Linuxové diskové oddíly na vašem disku, opakovaně jste mačkali **p** pro zobrazování seznamu diskových oddílů a poté jste mačkali **d** a číslo diskového oddílu které chcete smazat. To znamená že jste skončili s tabulkou neobsahující žádné položky:

Obrázek 4.6. Prázdná tabulka rozdělení disku

```
Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot      Start          End      Blocks   Id  System

Command (m for help):
```

Nyní když je tabulka uložená v paměti prázdná, jste připraveni vytvářet nové diskové oddíly. Použijeme výchozí schéma rozdělení popsané již dříve. Samozřejmě, nenásledujte tyto instrukce pokud disk stejným způsobem nechcete!

4.3.3. Vytvoření startovacího oddílu

Nejdříve vytvoříme malý startovací oddíl. Stiskněte **n** pro vytvoření nového diskového oddílu, poté **p** pro výběr primárního diskového oddílu a poté **1** pro výběr prvního primárního oddílu. Až budete dotázáni na číslo prvního cylindru, stiskněte **Enter**. Až budete dotázáni na číslo posledního cylindru, zadejte **+32M** pro vytvoření diskového oddílu o velikosti 32 MB:

Obrázek 4.7. Vytvoření startovacího oddílu

```
Command (m for help): n
```

```

Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-3876, default 1): (Hit Enter)
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-3876, default 3876): +32M

```

Nyní byste po stisku **p** měli vidět následující tabulku rozdělení disku:

Obrázek 4.8. Vytvořený startovací oddíl

```

Command (m for help): p

Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot      Start         End      Blocks   Id  System
/dev/hda1          1           14     105808+   83   Linux

```

Potřebujeme ještě tento diskový oddíl označit jako startovatelný (bootable). Zadejte **a** pro přepnutí startovacího flagu na tomto diskovém oddíle. Pokud poté znovu stisknete **p**, uvidíte že ve sloupci "Boot" se objevila hvězdička.

4.3.4. Vytvoření odkládacího oddílu

Vytvořme nyní odkládací oddíl. Stiskněte tedy **n** pro vytvoření nového diskového oddílu a **p** abyste programu **fdisk** sdělili že chcete primární diskový oddíl. Poté zadejte **2** pro vytvoření druhého primárního diskového oddílu, v našem případě `/dev/hda2`. Až budete dotázáni na číslo prvního válce, stiskněte **Enter** a na otázku na číslo posledního cylindru odpovězte `+512M` čímž vytvoříte diskový oddíl o velikosti 512MB. Až toto uděláte, zadejte **t** pro nastavení typu diskového oddílu, zadejte **2** pro výběr právě vytvořeného diskového oddílu a poté zadejte **82** čímž typ diskového oddílu nastavíte na "Linux swap". Až toto uděláte, zadejte **p** a zobrazená tabulka rozdělení disku by měla vypadat zhruba takto:

Obrázek 4.9. Tabulka rozdělení po vytvoření odkládacího oddílu

```

Command (m for help): p

Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	14	105808+	83	Linux
/dev/hda2		15	81	506520	82	Linux swap

4.3.5. Vytvoření kořenového adresáře

A nyní se konečně pustíme do vytváření kořenového diskového oddílu. Stiskněte proto **n** pro vytvoření nového diskového oddílu, poté **p** abyste programu **fdisk** oznámili že se bude jednat o primární diskový oddíl, v našem případě /dev/hda3. Až budete dotázáni na první cylindr, stiskněte **Enter** a až budete dotázáni na poslední cylindr tak opět stiskněte **Enter** což způsobí že tento nový diskový oddíl zabere všechno volné místo na disku. Po provedení těchto kroků a stisku **p** by se měla zobrazit zhruba následující tabulka rozdělení disku:

Obrázek 4.10. Výpis rozdělení disku po vytvoření kořenového adresáře

```
Command (m for help): p

Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	14	105808+	83	Linux
/dev/hda2		15	81	506520	82	Linux swap
/dev/hda3		82	3876	28690200	83	Linux

4.3.6. Uložení rozdělení disku

Pro uložení rozdělení disku a ukončení programu **fdisk** stiskněte **w**.

Obrázek 4.11. Uložení rozdělení a ukončení programu fdisk

```
Command (m for help): w
```

Nyní když máte vytvořeny potřebné diskové oddíly, můžete pokračovat v části Vytváření souborových systémů.

4.4. Použití programu fdisk na architektuře Alpha k rozdělení vašeho disku

Poznámka

Tato část je určena pouze pro systémy s architekturou Alpha.

Následující odstavce vysvětlují jak vytvořit vzorové rozdělení disku popsany výše, konkrétně:

Diskový oddíl	Popis
/dev/sdaa	Odkládací (swapovací) oddíl
/dev/sdab	Kořenový oddíl
/dev/sdac	Celý disk (nutné)

Toto rozvržení disku si samozřejmě můžete přizpůsobit svým požadavkům.

4.4.1. Identifikace dostupných disků

Ke zjištění jaké disky máte v počítači nainstalovány použijte následující příkazy:

Obrázek 4.12. Identifikace dostupných disků

```
(For IDE disks)      # dmesg | grep 'drive$'
(For SCSI disks)    # dmesg | grep 'scsi'
```

Z výstupu tohoto příkazu byste měli být schopni určit které disky jsou ve vašem systému nainstalovány a jim odpovídající položky v adresáři /dev. V následujícím textu předpokládáme že disk je SCSI disk odpovídající /dev/sda.

Nyní spusťte program **fdisk**:

Obrázek 4.13. Spuštění programu fdisk

```
# fdisk /dev/sda
```

4.4.2. Smazání všech diskových oddílů

Začneme smazáním všech diskových oddílů kromě oddílu označeného "c". Následující text předvádí postup jak smazat diskový oddíl (v našem příkladu používáme 'a'). Opakujete tento postup pro všechny ostatní diskové oddíly (pochopitelně kromě oddílu 'c').

Pro zobrazení všech existujících diskových oddílů použijte příkaz **p**. Příkaz **d** je používám pro mazání diskového oddílu.

Obrázek 4.14. Mazání diskového oddílu

```

BSD disklabel command (m for help): p

8 partitions:
#      start      end      size      fstype      [fsize bsize  cpg]
a:         1      235*     234*     4.2BSD      1024  8192   16
b:       235*     469*     234*      swap
c:         1     5290*    5289*    unused         0     0
d:       469*    2076*    1607*    unused         0     0
e:      2076*    3683*    1607*    unused         0     0
f:      3683*    5290*    1607*    unused         0     0
g:       469*    1749*    1280     4.2BSD      1024  8192   16
h:      1749*    5290*    3541*    unused         0     0

BSD disklabel command (m for help): d
Partition (a-h): a

```

Po opakování tohoto postupu pro všechny diskové oddíly dostanete něco jako:

Obrázek 4.15. Prázdné schéma

```

BSD disklabel command (m for help): p

3 partitions:
#      start      end      size      fstype      [fsize bsize  cpg]
c:         1     5290*    5289*    unused         0     0

```

4.4.3. Vytvoření diskového oddílu

Na Alpha systémech nepotřebujete oddělený startovací diskový oddíl. Avšak první cylindr nesmí být použit, protože tzv. boot image bude umístěn právě zde.

Vytvoříme proto odkládací oddíl začínající na třetím cylindru a o celkové velikosti 1 GB. Použijte **n** pro vytvoření nového diskového oddílu. Po vytvoření oddílu změňte jeho typ na 1, což znamená odkládací oddíl.

Obrázek 4.16. Vytváření diskového oddílu

```

BSD disklabel command (m for help): n
Partition (a-p): a
First cylinder (1-5290, default 1): 3
Last cylinder or +size or +sizeM or +sizeK (3-5290, default 5290): +1024M

```

```
BSD disklabel command (m for help): t
Partition (a-c): a
Hex code (type L to list codes): 1
```

Po provedení těchto kroků byste měli vidět něco jako:

Obrázek 4.17. Rozdělení disku po vytvoření odkládacího oddílu

```
BSD disklabel command (m for help): p

3 partitions:
#      start      end      size      fstype    [fsize bsize  cpg]
a:         3      1003      1001       swap
c:         1     5290*     5289*    unused          0      0
```

4.4.4. Vytvoření kořenového adresáře

Nyní vytvoříme kořenový diskový oddíl, který začíná hned na prvním cylindru za odkládacím oddílem. Použijte příkaz `p` pro zjištění na kterém cylindru odkládací oddíl končí. Ten náš končí na 1003, takže kořenový diskový oddíl začíná na 1004.

Další problém je že v programu `fdisk` je chyba díky které si myslí že dostupných cylindrů je o jeden více než je tomu ve skutečnosti. Jinými slovy až budete požádáni o zadání posledního cylindru, odečtete od čísla cylindru (v našem případě 5290) jedna.

Po vytvoření diskového oddílu změňte jeho typ na 8, což je `ext2`.

Obrázek 4.18. Vytváření kořenového diskového oddílu

```
D disklabel command (m for help): n
Partition (a-p): b
First cylinder (1-5290, default 1): 1004
Last cylinder or +size or +sizeM or +sizeK (1004-5290, default 5290): 5289

BSD disklabel command (m for help): t
Partition (a-c): b
Hex code (type L to list codes): 8
```

Vaše rozdělení disku by se nyní mělo podobat tomuto:

Obrázek 4.19. Zobrazení rozdělení disku

```

BSD disklabel command (m for help): p

3 partitions:
#      start      end      size      fstype      [fsize bsize  cpg]
a:         3       1003      1001        swap
b:       1004       5289      4286        ext2
c:         1       5290*     5289*       unused          0      0

```

4.4.5. Uložení rozdělení disku

Pro uložení rozdělení disku a ukončení programu **fdisk** stiskněte **w**.

Obrázek 4.20. Uložení rozdělení a ukončení programu fdisk

```

Command (m for help): w

```

Nyní když máte vytvořeny potřebné diskové oddíly, můžete pokračovat v části Vytváření souborových systémů.

4.5. Použití programu fdisk na architektuře SPARC k rozdělení vašeho disku

Poznámka

Tato část je určena výhradně pro systémy s architekturou SPARC.

Následující text vysvětluje jak vytvořit vzorové rozdělení disku uvedené výše, konkrétně:

Diskový oddíl	Popis
/dev/hda1	Startovací diskový oddíl
/dev/hda2	Odkládací (swapovací) oddíl
/dev/hda3	Sun Disk Label (vyžadováno)
/dev/hda4	Kořenový oddíl

Toto rozvržení disku si samozřejmě upravte dle svých potřeb.

4.5.1. Spuštění programu fdisk

Spustěte program **fdisk** s diskem uvedeným jako argument:

Obrázek 4.21. Spuštění programu fdisk

```
# fdisk /dev/hda
```

Objevit by se měla zhruba následující obrazovka:

Obrázek 4.22. Příkazová řádka programu fdisk

```
Command (m for help):
```

Pro zobrazení existujících diskových oddílů stiskněte klávesu **p**:

Obrázek 4.23. Zobrazení existujících diskových oddílů

```
Command (m for help): p

Disk /dev/hda (Sun disk label): 240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 bytes

Device Flag      Start      End      Blocks   Id  System
/dev/hda1      *           1         14      105808+  83  Linux
/dev/hda2                15         81      506520  82  Linux swap
/dev/hda3                 0      3876  29302528   5  Whole Disk
/dev/hda4                82      3876  28690200  83  Linux
```

Všimněte si textu *Sun disk label* ve výstupu. Pokud tento text chybí, znamená to že na disku je použito DOS rozdělení, nikoliv Sun rozdělení. V tom případě použijte klávesu **s** abyste se ujistili že na disku bude použita Sun tabulka rozdělení.

4.5.2. Smazání existujících diskových oddílů

Nyní nastal čas na smazání všech existujících diskových oddílů. Stiskněte tedy **d** a poté **Enter**. Budete dotázáni na číslo diskového oddílu který chcete smazat. Pro smazání existujícího diskového oddílu tedy stačí zadat:

Obrázek 4.24. Smazání diskového oddílu

```
Command (m for help): d
Partition number (1-4): 1
```

Pokud chcete smazat veškeré diskové oddíly použijte **p** pro zobrazení seznamu všech dostupných diskových oddílů a **d** pro jejich mazání jeden po druhém. Pokud máte pocit že jste udělali chybu, stiskněte okamžitě **q** - fdisk totiž nezapisuje provedené změny na disk okamžitě ale ponechává si změny v paměti dokud je nepotvrdíte. Na disk je zapíše pouze pokud stisknete **w**.

Po smazání všech diskových oddílů byste měli dosáhnout zhruba následujícího seznamu diskových oddílů:

Obrázek 4.25. Prázdný seznam diskových oddílů

```
Command (m for help): p

Disk /dev/hda (Sun disk label): 240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 bytes

Device Flag      Start      End      Blocks   Id  System
```

4.5.3. Vytvoření diskového oddílu Sun Disk Label

Nyní když je v paměti uložena prázdná tabulka rozdělení disku, jsme připraveni na vytvoření Sun Disk Label diskového oddílu. Stiskněte tedy **n** pro vytvoření nového diskového oddílu, a poté stiskněte **3** pro vytvoření nového třetího primárního diskového oddílu. Až budete dotázáni na první cylindr stiskněte **Enter**. Stejně tak stiskněte **Enter** až budete dotázáni na poslední cylindr a poté stiskněte **t** pro změnu typu diskového oddílu a **5** čímž typ nastavíte na "Whole disk" (Celý Disk).

Obrázek 4.26. Kroky potřebné pro vytvoření diskového oddílu Sun Disk Label

```
Command (m for help): n
Partition number (1-4): 3
First cylinder (1-3876, default 0): 0
Last cylinder or +size or +sizeM or +sizeK (1-3876, default 3876): (Press Enter)
Using default value 3876

Command (m for help): t
Partition number (1-8): 3
Hex code (type L to list codes): 5
```

Po provedení těchto kroků a po stisku **p** by se měla objevit zhruba následující tabulka rozdělení disku:

Obrázek 4.27. Zobrazení rozdělení disku

```
Command (m for help): p

Disk /dev/hda (Sun disk lable): 240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 bytes

Device Flag      Start      End      Blocks   Id  System
/dev/hda3         0        3876   29302528    5  Whole disk
```

4.5.4. Vytvoření startovacího diskového oddílu

Nyní jsme připraveni na vytvoření startovacího oddílu. Stiskněte tedy **n** pro vytvoření nového diskového oddílu a poté **1** pro vytvoření prvního diskového oddílu. Až budete dotázáni na první cylindr stiskněte Enter a při dotazu na poslední cylindr zadejte +32M čímž vytvoříte diskový oddíl o velikosti 32 MB. Ujistěte se že celý startovací oddíl leží v prvních 2 GB disku. Výsledek těchto kroků můžete vidět níže:

Obrázek 4.28. Vytvoření startovacího disku

```
Command (m for help): n
Partition number (1-4): 1
First cylinder (1-3876, default 1): (Press Enter)
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-3876, default 3876): +32M
```

Nyní byste po stisku **p** měli vidět výpis podobný tomu následujícímu:

Obrázek 4.29. Výpis rozdělení disku

```
Command (m for help): p

Disk /dev/hda (Sun disk label): 240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 bytes

Device Flag      Start      End      Blocks   Id  System
/dev/hda1         1         14    105808+   83  Linux
/dev/hda3         0        3876   29302528    5  Whole disk
```

4.5.5. Vytvoření odkládacího diskového oddílu

Jako další vytvoříme odkládací diskový oddíl. Stiskněte tedy **n** pro vytvoření nového diskového oddílu, poté **2** pro vytvoření druhého diskového oddílu, v našem případě `/dev/hda2`. Až budete dotázáni na první cylindr stiskněte **Enter**. Až budete dotázáni na poslední cylindr zadejte `+512MB` čímž vytvoříte diskový oddíl o velikosti 512 MB. Poté stiskněte **t** pro změnu typu nově vytvořeného diskového oddílu, nastavte typ na hodnotu 82 tj. na "Linux swap". Po provedení těchto kroků stiskněte **p** a měla by se zobrazit zhruba následující tabulka:

Obrázek 4.30. Výpis rozdělení disku

```
Command (m for help): p

Disk /dev/hda (Sun disk label): 240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 bytes

Device Flag      Start      End      Blocks    Id  System
/dev/hda1         1         14     105808+   83  Linux
/dev/hda2        15         81     506520   82  Linux swap
/dev/hda3         0        3876    29302528   5  Whole disk
```

4.5.6. Vytvoření kořenového diskového oddílu

A nakonec vytvoříme kořenový diskový oddíl. Zadejte proto **n** pro vytvoření nového diskového oddílu a zvolte **4** pro vytvoření čtvrtého diskového oddílu, `/dev/hda4` v našem případě. Až budete dotázáni na první cylindr stiskněte **Enter**, a stejně tak až budete dotázáni na poslední cylindr takže nově vytvořený diskový oddíl zabere celý volný zbytek na vašem disku. Po dokončení těchto kroků a stisku **p** by se měla objevit zhruba následující tabulka:

Obrázek 4.31. Výpis rozdělení disku

```
Command (m for help): p

Disk /dev/hda (Sun disk label): 240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 bytes

Device Flag      Start      End      Blocks    Id  System
/dev/hda1         1         14     105808+   83  Linux
/dev/hda2        15         81     506520   82  Linux swap
/dev/hda3         0        3876    29302528   5  Whole disk
/dev/hda4        82        3876    28690200   83  Linux
```

4.5.7. Uložení rozdělení disku

Pro uložení rozdělení disku a ukončení programu **fdisk** zadejte **w**:

Obrázek 4.32. Uložení rozdělení disku a ukončení programu **fdisk**

```
Command (m for help): w
```

Nyní když máte vytvořeny potřebné diskové oddíly, můžete pokračovat v části Vytváření souborových systémů.

4.6. Použití programu **mac-fdisk** na architektuře PPC k rozdělení vašeho disku

Diskové oddíly budeme vytvářet pomocí programu **mac-fdisk**:

Obrázek 4.33. Spuštění programu **mac-fdisk**

```
# mac-fdisk /dev/hda
```

Nejdříve vymažte diskové oddíly které jste uvolnili pro svoje budoucí Linuxové diskové oddíly. V programu **mac-fdisk** použijte volbu **d** pro smazání těchto diskových oddílů. Program se vás bude ptát na čísla diskových oddílů které chcete smazat.

Zadruhé, vytvořte diskový oddíl **Apple_Bootstrap** pomocí volby **b**. Program se vás zeptá na kterém bloku chcete začít. Pokud jste již zvolili **3** jako číslo diskového oddílu, zadejte **3p**.

Nyní vytvořte odkládací oddíl stiskem **c**. Program **mac-fdisk** se vás opět zeptá na kterém bloku má tento diskový oddíl začínat. Protože jsme při vytváření diskového oddílu **Apple_Bootstrap** zvolili **3**, zadáme nyní **4p**. Až budete dotázáni na velikost, zadejte 512MB (nebo cokoliv chcete - nicméně 512 MB je doporučená velikost). Až budete požádáni o zadání jména diskového oddílu, použijte **swap** (povinně).

Pro vytvoření kořenového adresáře zadejte **c** následované **5p** pro výběr od kterého bloku by měl diskový oddíl začínat. Až budete dotázáni na velikost, zadejte opět **5p**. Program **mac-fdisk** bude tuto volbu interpretovat jako "Použít všechnen volný prostor." Až budete požádáni o zadání jména diskového oddílu zadejte **root** (povinně).

Pro ukončení programu **mac-fdisk** a zapsání rozdělení disku použijte **w** a následně **q**.

Nyní když máte vytvořeny potřebné diskové oddíly, můžete pokračovat v části Vytváření souborových systémů.

4.7. Použití programu **fdisk** na architektuře HP-

PA k rozdělení vašeho disku

Pro vytvoření požadovaných diskových oddílů použijte program **fdisk**:

Obrázek 4.34. Dělení pevného disku

```
# fdisk /dev/sda
```

PALO potřebuje pro svou činnost speciální diskový oddíl. Musíte proto na začátku svého pevného disku vytvořit diskový oddíl o velikosti alespoň 16 MB. Tento diskový oddíl musí být typu **f0** (Linux/PA-RISC boot).

Poznámka

Důležité: pokud toto budete ignorovat a budete pokračovat bez zvláštního PALO diskového oddílu, váš systém vás přestane mít rád a odmítne nastartovat.

Také, pokud je váš disk větší než 2 GB, se ujistěte že startovací diskový oddíl je v prvních 2 GB na vašem disku. PALO je neschopné číst jádro za 2 GB limitem.

Nyní když máte vytvořeny potřebné diskové oddíly, můžete pokračovat v části Vytváření souborových systémů.

4.8. Použití programu fdisk na architektuře MIPS k rozdělení vašeho disku

4.8.1. Vytváření diskového oddílu SGI Disk Label

Všechny disky v SGI Systému vyžadují SGI Disk Label které mají podobnou funkci jako Sun & MS-DOS disklabely - ukládají informaci o diskovém oddílu. Vytvořením diskového oddílu SGI Disk Label vzniknou na disku dva nové zvláštní diskové oddíly:

- SGI Volume Header (9. diskový oddíl): Tento diskový oddíl je důležitý, protože sem budou umístovány obrazy jádra. Aby he sem bylo možné uložit budete muset použít nástroj známý jako dvhtool pro zkopírování obrazů jádra do tohoto diskového oddílu. Poté budete moci nastartovat jádro z tohoto diskového oddílu za použití SGI PROM Monitor.
- SGI Volume (11. diskový oddíl): Tento diskový oddíl má podobný účel jako třetí diskový oddíl systém SGI Disk Label, tj. "Whole Disk" (celý disk). Zabírá tedy celý pevný disk a měl by zůstat nedotčený. Neslouží žádnému zvláštnímu účelu kromě toho že nedokumentovaným způsobem spolupracuje s PROM Monitor (nebo je nějakým způsobem používán systémem IRIX).

Poznámka

Diskový oddíl SGI Volume Header musí začínat na cylindru 0. Pokud toto nedodržíte, PROM Monitor nebude moci nastartovat systém z disku.

Následující výpis je příkladem použití programu **fdisk**. Přečtěte si ho a přizpůsobte svým potřebám:

Obrázek 4.35. Vytváření diskového oddílu SGI Disk Label

```
# fdisk /dev/sda

Command (m for help): x

Expert command (m for help): m
Command action
    b   move beginning of data in a partition
    c   change number of cylinders
    d   print the raw data in the partition table
    e   list extended partitions
    f   fix partition order
    g   create an IRIX (SGI) partition table
    h   change number of heads
    m   print this menu
    p   print the partition table
    q   quit without saving changes
    r   return to main menu
    s   change number of sectors/track
    v   verify the partition table
    w   write table to disk and exit

Expert command (m for help): g
Building a new SGI disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content will be unrecoverably lost.

Expert command (m for help): r

Command (m for help): p

Disk /dev/sda (SGI disk label): 64 heads, 32 sectors, 17482 cylinders
Units = cylinders of 2048 * 512 bytes

----- partitions -----
Pt#      Device  Info      Start      End    Sectors  Id  System
   9:    /dev/sda1              0          4     10240   0   SGI volhdr
  11:    /dev/sda2              0     17481  35803136   6   SGI volume
----- Bootinfo -----
Bootfile: /unix
----- Directory Entries -----

Command (m for help):
```

Poznámka

Pokud už na vašem disku SGI Disk Label existuje, potom **fdisk** vytvoření dalšího nedovolí. Existují dva způsoby jak to obejít. Prvním je vytvořit Sun nebo MS-DOS disklabel, zapsat změny na disk a znovu spustit **fdisk**. Druhým je přepsat tabulku rozdělení disku nulami (null data) pomocí následujícího příkazu: **dd if=/dev/zero of=/dev/sda bs=512 count=1**.

4.8.2. Nastavení té správné velikosti pro SGI Volume Header

Nyní když máte SGI Disklabel vytvořen můžete definovat i jednotlivé diskové oddíly. V příkladu uvedeném výše již existují dva nadefinované diskové oddíly. Jsou to speciální diskové oddíly zmíněné výše a za normálních okolností by neměly být měněny. Avšak pro nainstalování Gentoo Linuxu budeme potřebovat načíst několik obrazů jádra přímo do diskového oddílu SGI Volume Header, protože v Portage dosud není žádný bootloader podporovaný SGI. Do tohoto diskového oddílu se vejde až osm obrazů jader libovolné velikosti přičemž každý obraz může mít přiřazeno jméno o osmi písmenech.

Proces zvětšení diskového oddílu SGI Volume Header není zcela přímočarý - je potřeba malý trik. Tento diskový oddíl nelze jednoduše smazat a znovu vytvořit kvůli zvláštnímu chování fdisku. V následujícím příkladu vytvoříme 50 MB diskový oddíl SGI Volume Header společně s 50 MB diskovým oddílem /boot. Konkrétní rozdělení disku se může lišit, takže toto je pouze pro ilustraci.

Obrázek 4.36. Změna velikosti diskového oddílu SGI Volume Header

```
Command (m for help): n
Partition number (1-16): 1
First cylinder (5-8682, default 5): 51
Last cylinder (51-8682, default 8682): 101
(Notice how fdisk only allows Partition #1 to be re-created starting at a minimum of cylinder 51.
(Had you attempted to delete & re-create the SGI Volume Header this way, this is the same error
you would have encountered.)
(In our example, we want /boot to be 50MB, so we start it at cylinder 51 (the Volume Header
start at cylinder 0, remember?), and set its ending cylinder to 101, which will roughly double the size.)

Command (m for help): d
Partition number (1-16): 9
(Delete Partition #9 (SGI Volume Header))

Command (m for help): n
Partition number (1-16): 9
First cylinder (0-50, default 0): 0
Last cylinder (0-50, default 50): 50
(Re-Create Partition #9, ending just before Partition #1)
```

4.8.3. Dokončení rozdělení disku

Jakmile toto máte hotovo, můžete klidně dokončit dělení disku a vytváření potřebných diskových oddílů. Poté co máte všechny potřebné diskové oddíly vytvořeny se ujistěte že jste ID odkládacího diskového oddílu nastavili na 82, což je Linux Swap. Implicitně to bude 83, tj. Linux Native.

Nyní když máte vytvořeny potřebné diskové oddíly, můžete pokračovat v části Vytváření souborových systémů.

4.9. Vytváření souborových systémů

4.9.1. Úvod

Nyní když máte vytvořeny diskové oddíly, nadešel čas umístit na ně souborové systémy. Pokud vás příliš netrápí který souborový systém použít a spokojíte se s výchozími volbami použitými v této příručce, pokračujte v části Umístění souborového systému na diskový oddíl. V opačném případě čtěte dále abyste se dozvěděli o zajímavých souborových systémech...

4.9.2. Souborové systémy

Je dostupných několik souborových systémů. Některé z nich jsou stabilní na všech architekturách, některé pouze na několika. Následující tabulka vyjmenovává seznam dostupných souborových systémů a architektury na kterých je ověřena jejich funkčnost. Pokud je architektura uvedena v závorkách, potom by souborový systém měl fungovat ale není to otestováno.

Souborový systém	Žurnálovací	Architektury
ext2	no	všechny architektury
ext3	yes	všechny architektury
reiserfs	yes	x86, hppa, alpha, (mips), (ppc), (amd64)
xfs	yes	x86, alpha, amd64, (ppc)
jfs	yes	x86, alpha, (mips), (amd64)

ext2 je odzkoušený a osvědčený Linuxový souborový systém ale neobsahuje žurnálování metadat, což znamená že rutinní kontroly *ext2* filesystemu při startu mohou být dosti časově náročné. V současnosti již existuje poměrně slušný výběr žurnálovacích souborových systémů nové generace jejichž konzistenci lze zkontrolovat velice rychle a jsou proto všeobecně upřednostňovány před těmi nežurnálovacími. Použití žurnálovacích souborových systémů zabráňuje dlouhým prodlevám při startu systému v nekonzistentním stavu.

ext3 je žurnálovací verze souborového systému *ext2*, poskytující žurnálování metadat pro rychlé zotavení navíc k dalším vyspělým módům žurnálování jako jsou "full data" a "ordered data". V současnosti je *ext3* velice použitelný souborový systém nabízející velice slušný výkon za většiny podmínek. Protože v designu nejsou příliš důsledně používány "stromy", nemá příliš dobrou škálovatelnost, což znamená že se nejedná o ideální volbu pro velmi velké souborové systémy nebo situace kdy musíte pracovat s velice velkými soubory nebo velkým množstvím souborů v jediném adresáři. Ale pokud je používán v rámci parametrů designu, je *ext3* skvělý souborový systém.

ReiserFS je souborový systém založený na B*-stromech poskytující skvělý celkový výkon vysoce převyšující jak *ext2* tak i *ext3* při práci s malými soubory (méně než 4k) a to často i 10x až 15x. *ReiserFS* je také extrémně dobře škálovatelný a používá žurnálování metadat. Od jádra 2.4.18+ je *ReiserFS* skvěle stabilní a je všeobecně doporučován pro použití jak jako běžný souborový systém tak i pro extrémní případy jako je například vytváření velkých souborových systémů, používání velkého počtu malých souborů a adresářů obsahujících desetitisíce souborů.

XFS je souborový systém se žurnálováním metadat který je plně podporován Gentoo Linux xfs jádrem. Je dodáván s bohatou množinou vlastností a je optimalizován pro škálovatelnost. Tento systém doporučujeme používat pouze na Linuxovém systému se špičkovým SCSI nebo fibre channel zařízením a nepřerušitelným zdrojem. Protože *XFS* agresivně kešuje pracovní data v paměti RAM, nevhodně navržené programy (ty které nejsou obezřetné při zápisu souborů na disk, a těch je spousta) mohou ztratit velkou část dat pokud systém neočekávaně havaruje.

JFS je vysoce výkonný žurnálovací souborový systém firmy IBM. Za production-ready byl uznán teprve nedávno a zatím tedy není dostatek informací pro posouzení jeho obecné stability.

4.9.3. Umístění souborového systému na diskový oddíl

Pro vytváření souborových systémů na diskovém oddíle existují pro každý typ souborového systému samostatné nástroje:

Souborový systém	Nástroj na vytváření
ext2	mke2fs
ext3	mke2fs -j
reiserfs	mkreiserfs
xfs	mkfs.xfs
jfs	mkfs.jfs

Například abychom na startovacím diskovém oddíle (v našem příkladě `/dev/hda1`) vytvořili souborový systém *ext2* a na kořenovém oddíle (v našem příkladě `/dev/hda3`) vytvořili souborový systém *ext3*, bychom použili:

Obrázek 4.37. Vytváření souborových systémů na našich diskových oddílech

```
# mke2fs /dev/hda1
# mke2fs -j /dev/hda3
```

Nyní vytvořte souborové systémy na vašich nově vytvořených diskových oddílech (nebo logických oddílech).

4.9.4. Aktivace odkládacího oddílu

Poznámka

Ti kdo používají Knoppix a již na svém systému mají nějaký odkládací oddíl mohou tento krok přeskočit, protože Knoppix odkládací oddíly aktivuje automaticky.

Pro inicializaci odkládacího oddílu je používán příkaz **mkswap**:

Obrázek 4.38. Vytváření odkládacího oddílu (swap signature)

```
# mkswap /dev/hda2
```

Pro aktivaci odkládacího oddílu je používán příkaz **swapon**:

Obrázek 4.39. Aktivace odkládacího oddílu

```
# swapon /dev/hda2
```

Nyní vytvořte a aktivujte odkládací oddíl.

4.10. Připojování

Nyní když máte vytvořeny diskové oddíly a vytvořili jste na nich souborové systémy, je čas tyto diskové oddíly připojit. Použijte příkaz **mount**. A nezapomeňte vytvořit potřebné adresáře do kterých mají být připojeny.

Obrázek 4.40. Připojování diskových oddílů

```
# mount /dev/hda3 /mnt/gentoo
# mkdir /mnt/gentoo/boot
# mount /dev/hda1 /mnt/gentoo/boot
```

Poznámka

Pokud chcete adresář `/tmp` umístit na samostatný diskový oddíl, nezapomeňte po připojení změnit přístupová práva: **chmod 1777 /mnt/gentoo/tmp**. To platí také pro `/var/tmp`.

Nyní potřebujeme připojit ještě souborový systém `proc` (virtuální rozhraní s jádrem) na `/proc`. Nejdříve vytvoříme připojovací místo `/mnt/gentoo/proc`.

Obrázek 4.41. Vytváření připojovacího místa /mnt/gentoo/proc

```
# mkdir /mnt/gentoo/proc
```

Pokud instalujete Gentoo z LiveCD, potom stačí obyčejný mount proc:

Obrázek 4.42. Připojování diskového oddílu proc

```
# mount -t proc none /mnt/gentoo/proc
```

Pokud Gentoo neinstalujete z LiveCD, potřebujete proc připojit jako bind-mount:

Obrázek 4.43. Bind-mount proc

```
# mount -o bind /proc /mnt/gentoo/proc
```

Nyní pokračujte v kapitole Instalace Gentoo instalačních souborů.

Kapitola 5. Instalace Gentoo

instalačních souborů

Abstrakt

Gentoo Linux je instalován pomocí takzvaných stage-files. V tomto odstavci je popsáno jak získat stage-file a nakonfigurovat Portage.

5.1. Volba správné etapy

Když jsme po vás chtěli abyste si zvolili způsob instalace (LiveCD, existující distribuce atd.) vysvětlili jsme vám jaké mají jednotlivé možnosti pro a proti. Jedním z nich byla možnost volby etapového balíčku - chcete vše kompilovat sami (stage1), přeskočit bootstrapping (stage2) nebo začít z předkompilované instalace (stage3)?

Podle toho jaký jste si zvolili způsob instalace se můžete stále ještě rozhodovat. Někteří se rozhodli již na začátku instalace. Nyní je třeba se rozhodnout který etapový balíček zvolíte.

Následující dva odstavce vysvětlují jak nainstalovat váš zvolený etapový balíček. Implicitní možnost je stáhnout si zvolený etapový balíček z Internetu, avšak některá LiveCD mají tyto etapové balíčky dostupné na CD.

Pokud máte fungující připojení k Internetu, doporučujeme vám použít implicitní možnost. Avšak pokud připojení k Internetu nemáte, nebo pokud chcete instalovat Gentoo za použití GRP (z předkompilovaných balíčků), potom si musíte zvolit alternativní způsob.

- Výchozí: Stažení z Internetu
- Náhradní: Použití etapového balíčku z LiveCD

5.2. Výchozí: Stažení z Internetu

5.2.1. Stažení etapového balíčku z Internetu

Přesuňte se do adresáře do kterého jste připojili vytvořené souborové systémy (pravděpodobně /mnt/gentoo):

Obrázek 5.1. Přesun do adresáře /mnt/gentoo

```
cd /mnt/gentoo
```

V závislosti na způsobu instalace máte dostupných několik nástrojů pro stažení etapového balíčku. Pokud máte k dispozici **lynx**, potom můžete okamžitě zobrazit seznam mirrorů Gentoo [<http://www.gentoo.org/main/en/mirrors.xml>] a zvolit si některý blízko vás. Poté zvolte adresář `releases/`, následovaný vaší architekturou (například `x86/`), poté verzí Gentoo (`1.4/`) a konečně adresář `stages/`. Nyní byste měli vidět všechny dostupné etapové balíčky pro svou architekturu. Zvolte si jeden a stiskněte **D** pro stažení. Až se balíček stáhne, stiskněte **Q** pro ukončení prohlížeče.

Obrázek 5.2. Zobrazení seznamu mirrorů programem lynx

```
# lynx http://www.gentoo.org/main/en/mirrors.xml
```

Pokud nemáte **lynx**, měli byste mít k dispozici **links2**. Program `links2` je mocnější než `lynx`, ale má několik nevýhod. Jednou z nich je že nerespektuje proměnné pro proxy server které jsme nastavili dříve. Pokud potřebujete použít proxy server, použijte **links2 -http-proxy proxy.server.com:8080**. Od této chvíle následujte stejný postup jako pro **lynx**, protože jsou stejné.

Obrázek 5.3. Zobrazení seznamu mirrorů programem links2

```
(Without proxy:) # links2 http://www.gentoo.org/main/en/mirrors.xml
(With proxy:)   # links2 -http-proxy proxy.server.com:8080 http://www.gentoo.org/main/en
```

Pokud chcete zkontrolovat integritu staženého balíčku, použijte příkaz **md5sum** a srovnejte výstup s MD5 kontrolním součtem poskytovaným na našem mirroru.

Obrázek 5.4. Kontrola integrity etapového balíčku

```
# md5sum stage1-x86-20030910.tar.bz2
6cdalcc745ba882731ac07fbae0dd973 stage1-x86-20030910.tar.bz2
```

5.2.2. Rozbalení etapového balíčku

Nyní rozbalte stažený balíček na svůj systém. Použijeme GNU nástroj **tar** protože se jedná o nejjednodušší metodu:

Obrázek 5.5. Rozbalení etapového balíčku

```
# tar -xvjp stage?-*.tar.bz2
```

Ujistěte se že používáte stejnou sadu voleb (`-xvjp`). Volba `x` znamená "Extract" (Rozbalit), `v` znamená "Verbose" (Ukecaný, a ano - tato volba je nepovinná), `j` znamená "Decompress with bzip2" (Rozbalit programem bzip2), `p` znamená "Preserve permissions" (Zachovej nastavení práv) a `f` znamená že chceme rozbalovat data ze souboru a nikoliv ze standardního vstupu.

hotovo? OK, teď jsme připraveni postoupit do další části kterou je Konfigurace kompilátoru.

5.3. Náhradní: Použití etapového balíčku z LiveCD

Na LiveCD jsou etapové balíčky umístěny v adresáři `/mnt/cdrom/stages`. Pro zobrazení jejich seznamu použijte příkaz `ls`:

Obrázek 5.6. Zobrazení seznamu dostupných obrazů

```
# ls /mnt/cdrom/stages
```

Pokud vám systém odpoví chybovou hláškou, musíte CD nejdříve připojit:

Obrázek 5.7. Připojení CD

```
# ls /mnt/cdrom/stages
ls: /mnt/cdrom/stages: No such file or directory
# mount /dev/cdroms/cdrom0 /mnt/cdrom
# ls /mnt/cdrom/stages
```

Nyní se přesuňte do adresáře kam jste připojili souborový systém (obvykle `/mnt/gentoo`):

Obrázek 5.8. Změna adresáře na /mnt/gentoo

```
# cd /mnt/gentoo
```

Nyní rozbalíme zvolený etapový balíček, a použijeme k tomu GNU nástroj `tar`. Ujistěte se že používáte stejnou sadu voleb (`-xvjp`)! V následujícím příkladě rozbalíme etapový balíček `stage3-20031011.tar.bz2`. Toto jméno samozřejmě musíte nahradit za jméno svého etapového balíčku.

Obrázek 5.9. Rozbalení etapového balíčku

```
# tar -xvjpf /mnt/cdrom/stages/stage3-20031011.tar.bz2
```

5.3.1. Instalace obrazu Portage a zdrojového kódu

Na některých LiveCD je k dispozici obraz Portage. Protože čtete tento odstavec, můžeme předpokládat že právě takové LiveCD máte. Stačí se podívat do adresáře `/mnt/cdrom/snapshots` a zjistíte jaké obrazy máte k dispozici:

Obrázek 5.10. Zobrazení obsahu `/mnt/cdrom/snapshots`

```
# ls /mnt/cdrom/snapshots
```

Nyní rozbalte obraz následujícím příkazem. Ujistěte že pro tar používáte správnou sadu voleb, přičemž mějte na paměti že `-C` je velké písmeno `C` a nikoliv `c`. V dalším příkladě používáme obraz `portage-20031011.tar.bz2` jako jméno souboru, ale vy samozřejmě použijte jméno svého balíčku.

Obrázek 5.11. Rozbalení obrazu Portage

```
# tar -xvjf /mnt/cdrom/snapshots/portage-20031011.tar.bz2 -C /mnt/gentoo/usr
```

Potřebujete také zkopírovat zdrojové kódy z CD.

Obrázek 5.12. Zkopírování zdrojových kódů

```
# mkdir /mnt/gentoo/usr/portage/distfiles
# cp /mnt/cdrom/distfiles/* /mnt/gentoo/usr/portage/distfiles/
```

Pokud chcete použít GRP (předkompilované balíčky), pokračujte ve čtení. V opačném případě přeskočte až na část Konfigurace kompilátoru.

5.3.2. Volitelně: Příprava GRP

Pokud chcete nainstalovat Gentoo GRP (předkompilované balíčky), potřebujete zkopírovat všechny balíčky do svého souborového systému aby je Portage mohl používat.

Obrázek 5.13. Zkopírování předkompilovaných balíčků

```
# mkdir -p /mnt/gentoo/usr/portage/packages/All
# cp /mnt/cdrom/packages/All/* /mnt/gentoo/usr/portage/packages/All/
```

Nyní pokračujte v části Konfigurace kompilátoru.

5.4. Konfigurace kompilátoru

5.4.1. Úvod

Pro optimalizaci Gentoo můžete nastavit několik proměnných které ovlivňují chování Portage. Všechny tyto volby lze nastavit jako proměnné prostředí (za použití příkazu **export**), ale toto nastavení samozřejmě není trvalé. Pro uložení vašich nastavení je vám k dispozici soubor `/etc/make.conf`, což je konfigurační soubor pro Portage. Právě tento soubor nyní budeme upravovat.

Poznámka

V souboru `/mnt/gentoo/etc/make.conf(.example)` lze najít komentovaný seznam všech možných proměnných. Pro úspěšnou instalaci Gentoo Linuxu potřebujete pouze proměnné které jsou uvedeny níže.

Spusťte svůj oblíbený editor (v naší příručce používáme nano) abychom mohli upravit optimalizační proměnné diskutované dále.

Jak už jste si pravděpodobně všimli, soubor `make.conf(.example)` je strukturován obvyklým způsobem: řádky s komentáři začínají `#`, ostatní řádky definují proměnné pomocí syntaxe `PROMĚNNÁ="obsah"`. Některé z těchto proměnných jsou diskutovány dále.

5.4.2. CHOST

Poznámka

Ačkoliv by to mohlo být zajímavé i pro uživatele jiných etapových balíčků než je `stage1`, neměli by měnit nastavení proměnné `CHOST` v souboru `make.conf`. Změna této hodnoty by mohla učinit systém neuživatelným. Znovu: tuto proměnnou měňte pouze pokud používáte instalaci z etapového balíčku `stage1`.

Hodnota proměnné `CHOST` definuje pro kterou architekturu má kompilátor `gcc` překládat. Možnosti jsou:

Architektura	Subarchitektura	Nastavení CHOST
x86	i386	i386-pc-linux-gnu
x86	i486	i486-pc-linux-gnu
x86	i586	i586-pc-linux-gnu
x86	i686 a výše (včetně athlona)	i686-pc-linux-gnu

Architektura	Subarchitektura	Nastavení CHOST
alpha		alpha-unknown-linux-gnu
ppc		powerpc-unknown-linux-gnu
sparc		sparc-unknown-linux-gnu
hppa	(obecná)	hppa-unknown-linux-gnu
hppa	pa7000	hppa1.1-unknown-linux-gnu
hppa	pa8000 a výše	hppa2.0-unknown-linux-gnu
mips		mips-unknown-linux-gnu
amd64		x86_64-pc-linux-gnu

5.4.3. CFLAGS a CXXFLAGS

Proměnné `CFLAGS` a `CXXFLAGS` definují optimalizační přepínače pro gcc C a C++ překladač. Ačkoliv jsou obecně nastaveny, maximálního výkonu dosáhnete pouze pokud optimalizujete tyto přepínače pro každý program samostatně. Důvodem je rozdílnost každého programu.

V souboru `make.conf` byste měli definovat optimalizační přepínače které způsobí že váš systém bude obecně rychlý. Nepoužívejte experimentální nastavení - příliš mnoho optimalizací může způsobit chybné chování programů (pády nebo v horším případě chybnou funkci).

Nebudeme zde vysvětlovat všechny možnosti nastavení optimalizace. Pokud je chcete znát všechny, přečtěte si GNU Online Manuály nebo info stránku překladače gcc (info gcc -- funguje pouze na plně funkčním Linuxovém systému). Soubor `make.conf` sám o sobě obsahuje také mnoho příkladů a informací, nezapomeňte si je také přečíst.

První nastavení je `-march=` přepínač, které specifikuje jméno cílové architektury. Možné volby jsou popsány v souboru `make.conf` (jako komentáře). Například pro x86 Athlon XP architekturu:

Obrázek 5.14. GCC volba march

```
-march=athlon-xp
```

Další je přepínač `-O`, který určuje gcc optimalizaci. Možné třídy optimalizace jsou s (optimalizace velikosti), 0 (žádné optimalizace), 1, 2 nebo 3 pro optimalizaci rychlosti (každá třída má ta nastavení jako předchozí plus něco navíc). Například pro optimalizační třídu 2:

Obrázek 5.15. GCC volba O

```
-O2
```

Další populární optimalizační přepínače jsou `-pipe` (použití rour namísto dočasných souborů pro komunikaci mezi různými etapami kompilaci) a `-fomit-frame-pointer` (který způsobí že frame pointer se nebude udržovat v registrech pro ty funkce které to nepotřebují).

Při definování proměnných `CFLAGS` a `CXXFLAGS`, byste měli kombinovat několik optimalizačních přepínačů, jako například:

Obrázek 5.16. Definice proměnných `CFLAGS` a `CXXFLAGS`

```
CFLAGS="-march=athlon-xp -pipe -O2"
CXXFLAGS="${CFLAGS}"                # Use the same settings for both variables
```

5.4.4. MAKEOPTS

Pomocí proměnné `MAKEOPTS` můžete definovat kolik paralelních kompilací by mělo probíhat při instalaci balíčku. Doporučený počet je počet CPU ve vašem systému plus jedna.

Obrázek 5.17. Definice proměnné `MAKEOPTS`

```
MAKEOPTS="-j2"
```

5.4.5. Připravit, Pozor, Ted'!

Upravte soubor `/mnt/gentoo/etc/make.conf` tak aby odpovídal vašim představám a uložte ho. Nyní jste připraveni pokračovat v kapitole Instalace základního systému Gentoo.

Kapitola 6. Instalace základního systému Gentoo

Abstrakt

Ať již v předchozí části zvolíte jakýkoliv stage-file, výsledkem bude plně provozuschopný základní Gentoo systém. Tato kapitola popisuje jak se k tomuto stavu propracovat.

6.1. Chrootování

6.1.1. Nepovinné: Volba mirrorů

Pokud jste nainstalovali z Gentoo LiveCD, potom jste schopni použít program **mirrorselect** pro update / `etc/make.conf` pro rychlé mirrorry použitelné pro Portage i zdrojový kód:

Obrázek 6.1. Hledání rychlých mirrorů

```
# mirrorselect -a -s4 -o >> /mnt/gentoo/etc/make.conf
```

Pokud z nějakého důvodu program **mirrorselect** selže, napanikařte. Tento krok je zcela nepovinný. Pokud **mirrorselect** selže, postačí přednastavené hodnoty.

6.1.2. Kopírování DNS informací

Jednu věc ale musíme před přepnutím do nového prostředí udělat, a to je zkopírování DNS informací uložených v souboru `/etc/resolv.conf`. To musíte udělat abyste si zajistili že síť bude fungovat i po přechodu do nového prostředí. `/etc/resolv.conf` obsahuje jmenné servery pro vaši síť.

Obrázek 6.2. Kopírování DNS informací

```
# cp /etc/resolv.conf /mnt/gentoo/etc/resolv.conf
```

6.1.3. Nepovinné: Připojení /dev

Uživatelé Knoppixu (a další uživatelé kteří instalují Gentoo z instalačního média které nepoužívá DevFS) by nyní měli připojit (bind-mount) strukturu `/dev`. Pokud používáte některé z našich LiveCD můžete tento krok přeskočit.

Obrázek 6.3. Připojení /dev

```
# mkdir /mnt/gentoo/dev
# mount -o bind /dev /mnt/gentoo/dev
```

6.1.4. Vstup do nového prostředí

Nyní když jsou vaše diskové oddíly inicializovány a základní prostředí je nainstalováno, je čas přepnout se do nově nainstalovaného prostředí tím že se do něj chrootneme. To znamená že se přepneme z aktuálního instalačního prostředí (LiveCD nebo jiné médium) do instalovaného systému (konkrétně do inicializovaného diskového oddílu).

Toto chrootování se provádí ve třech krocích. V prvním změníme kořenový adresář z / (na instalačním médiu) na /mnt/gentoo (na vašem diskovém oddílu) za použití příkazu **chroot**. Poté pomocí příkazu **env-update** vytvoříme nové prostředí, což v podstatě znamená že vytvoříme proměnné prostředí. A konečně tyto proměnné načteme do paměti pomocí příkazu **source**.

Obrázek 6.4. Chrootnutí do nového prostředí

```
# chroot /mnt/gentoo /bin/bash
# env-update
Regenerating /etc/ld.so.cache...
# source /etc/profile
```

Gratulujeme! Nyní jste ve svém novém prostředí Gentoo Linuxu. Samozřejmě ještě zdaleka nejsme hotoví, což znamená že instalace ještě chvíli potrvá :-)

6.1.5. Nepovinné: aktualizace Portage

Pokud nepoužíváte GRP, musíte si stáhnout nový obraz Portage z Internetu. Program emerge sync to udělá za vás. Uživatelé GRP by měli tento krok přeskočit a pokračovat v odstavci Konfigurace USE proměnné.

Obrázek 6.5. Aktualizace Portage

```
# emerge sync
(In case you are unable to use rsync, use "emerge-webrsync" which
downloads and installs a portage snapshot for you)
# emerge-webrsync
```

Pokud se vám dostane varování že je dostupná nová verze programu Portage a že byste ji měli aktualizovat, můžete to klidně ignorovat. Program Portage pro vás bude aktualizován později během instalace.

6.1.6. Konfigurace USE proměnné

USE je jedna z nejmocnějších proměnných kterou Gentoo nabízí svým uživatelům. Několik programů může být zkompileováno s nebo bez podpory určitých položek. Například, některé programy mohou být zkompileovány s podporou GTK nebo QT. Jiné mohou být zkompileovány s podporou SSL nebo bez ní. Některé programy mohou být dokonce zkompileovány s podporou framebufferu (svglib) namísto podpory X11 (X-Server).

Většina distribucí kompiluje svoje balíčky s podporou všeho dostupného, což ale zvětšuje velikost programů a čas spouštění, o enormním nárůstu závislosti ani nemluvě. S Gentoo můžete definovat se kterými vlastnostmi by balíčky měly být kompilovány. A právě zde vstupuje do hry USE.

V proměnné USE definujete klíčová slova mapovaná na konfiguraci kompilace (compile-options). Například `ssl` zkompileuje podporu SSL do programu které ji podporují. `-X` odstraní podporu X-Serveru (všimněte si znaku mínus na začátku). Volba `gnome gtk -kde -qt` zkompileje program s podporou GNOME (a GTK) a nikoliv s podporou KDE (a QT), čili váš systém bude plně vyladěný pro GNOME.

Výchozí nastavení USE je uloženo v souboru `/etc/make.profile/make.defaults` a to co umístíte do souboru `/etc/make.conf` je s tímto výchozím nastavením porovnáno. Pokud k nastavení USE něco přidáte je to k implicitnímu seznamu přidáno, pokud něco z nastavení USE odeberete (tím že před klíčovým slovem použijete znaménko mínus) potom je to z implicitního seznamu odstraněno (pokud to na něm vůbec bylo). Nikdy nic neměňte přímo v adresáři `/etc/make.profile`, protože informace zde uložené jsou přepisovány při aktualizaci Portage!

Detailní popis USE můžete najít v druhé části Gentoo Handbook, v Kapitole 1: USE příznaky. Detailní popis USE příznaků lze najít také na vašem systému v souboru `/usr/portage/profiles/use.desc`.

Obrázek 6.6. Zobrazení dostupných USE příznaků

```
# less /usr/portage/profiles/use.desc
```

Jako příklad předvedeme jak nastavit USE pro KDE systém s DVD, ALSOU a podporou vypalování CD:

Obrázek 6.7. Otevření souboru `/etc/make.conf`

```
# nano -w /etc/make.conf
```

Obrázek 6.8. Nastavení USE příznaků

```
USE="-gtk -gnome qt kde dvd alsa cdr"
```

6.1.7. Nepovinné: Použití distribuované Kompilace

Pokud se zajímáte o využití skupiny počítačů ke kompilaci vašeho systému, potom byste se měli podívat na DistCC Guide [<http://www.gentoo.org/doc/en/distcc.xml>]. Díky programu **distcc** můžete během instalace využít výpočetní sílu několika systémů.

6.2. Rozdíly mezi stage1, stage2 a stage3

Posad'te se a přemýšlejte o svých posledních krocích. Měli jste si vybrat mezi etapovými balíčky stage1, stage2 a stage3 a byli jste upozorněni že toto rozhodnutí je důležité pro další instalační kroky. Takže, toto je první místo kde vaše předchozí rozhodnutí určuje další postup.

- Pokud jste si zvolili stage1, potom musíte provést oba následující kroky v této kapitole (začněte s částí Přechod od stage1 ke stage2)
- Pokud jste si zvolili stage2, potom můžete první krok přeskočit a začít rovnou s druhým (tj. v části Přechod od stage2 ke stage3)
- Pokud jste si zvolili stage3 (at již s GRP nebo bez GRP) poté můžete přeskočit oba následující kroky a pokračovat v následující kapitole: Konfigurace jádra

6.3. Přechod od stage1 ke stage2

6.3.1. Úvod do Bootstrappingu

Takže si chcete všechno zkompilevat od začátku? No tak dobrá :-)

V tomto kroku bootstrapneme váš Gentoo system. Bude to trvat dlouho, ale výsledkem bude systém optimalizovaný od samého základu pro váš konkrétní stroj a potřeby.

Bootstrapping znamená sestavení GNU C knihovny, GNU Compiler Collection a několika dalších systémových nástrojů. GNU Compiler Collection musí být přeložen dokonce dvakrát: poprvé s "obecným" překladačem dodaný námi, a podruhé právě s překladačem který jste právě sestavili.

Než s bootstrappingem začneme, podíváme se na seznam možností které možná chcete a možná ne. Pokud toto číst nechcete, pokračujte v části Bootstrapping systému.

6.3.2. Nepovinné: Minimalizace času potřebného ke kom-

pilaci

Pokud chcete urychlit bootstrapping, můžete dočasně vypnout podporu Javy. To znamená že GNU Compiler Collection a GNU C knihovna budou zkompileovány bez podpory Javy (což významným způsobem snižuje čas potřebný ke kompilaci). Ačkoliv toto znamená že nebudete mít GNU Java Compiler (gcj) neznamená to že by váš systém nebyl schopen spouštět Java Applety nebo další Javový materiál.

Pro dočasné vypnutí podpory Java definujte `USE="-java"` přes spuštěním bootstrapovacího skriptu.

Obrázek 6.9. Vypnutí podpory Javy

```
# export USE="-java"
```

A po skončení bootstrapping nezapomeňte tuto proměnnou zrušit:

Obrázek 6.10. Zrušení proměnné USE

```
# unset USE
```

6.3.3. Nepovinné: Stažení zdrojových kódů nejdříve

Pokud jste v některém z předchozích kroků nezkopírovali veškerý zdrojový kód, potom bootstrapovací skript stáhne všechny potřebné soubory. Není snad třeba říkat že toto funguje pouze pokud máte fungující síťové spojení :-). Pokud chcete zdrojové kódy stáhnout v předstihu a později provést bootstrapping systému (například pokud nechcete nechat spojení zapnuté i během instalace) použijte volbu `-f` bootstrapovacího skriptu, který si stáhne (anglicky *fetch*, proto písmeno *f*) všechny zdrojové kódy.

Obrázek 6.11. Stažení potřebných zdrojových kódů

```
# cd /usr/portage
# scripts/bootstrap.sh -f
```

6.3.4. Bootstrapping systému

Tak dobrá, čapněte klávesnici a nabušte do ní následující příkazy spouštějící bootstrapping. Pak se jděte nějak zabavit (například obtěžujte Gentoo vývojáře na kanálu `#gentoo`), protože tento krok bude chvíli trvat.

Obrázek 6.12. Bootstrapping systému

```
# cd /usr/portage
# scripts/bootstrap.sh
```

Pokud jste změnilí nastavení proměnné `CHOST` v souboru `/etc/make.conf`, potom musíte reinicializovat některé proměnné aby program `gcc` pracoval rychle:

Obrázek 6.13. Reinicializace proměnných prostředí

```
# source /etc/profile
```

Nyní pokračujte dalším krokem, a to Přejchod od stage2 ke stage3.

6.4. Přechod od stage2 ke stage3

6.4.1. Úvod

Pokud čtete tento odstavec, potom zřejmě máte bootstrapovaný systém (buď proto že jste již bootstrap provedli sami nebo jste použili etapový balíček stage2). V každém případě je právě ten správný okamžik sestavit všechny systémové balíčky.

Všechny systémové balíčky? Ne, ve skutečnosti ne. V tomto kroku sestavíte ty systémové balíčky u kterých není možné volit z více možností. U některých systémových balíčků (například u systémových loggerů) je možno volit z několika alternativ a protože Gentoo je právě o možnosti volby nechceme vás jakkoliv nutit k jedné konkrétní možnosti.

6.4.2. Volitelně: Zobrazení následujících kroků

Pokud chcete vědět které balíčky budou nainstalovány, spusťte příkaz **emerge --pretend system**, který vypíše seznam všech balíčků které budou kompilovány. Protože tento seznam je poměrně dlouhý, měli byste použít také stránkování například pomocí příkazů **less** nebo **more** pro listování v něm.

Obrázek 6.14. Zobrazení co "emerge system" provede

```
# emerge --pretend system | less
```

6.4.3. Volitelně: Stažení zdrojových kódů

Pokud chcete aby příkaz **emerge** nejdříve stáhnul veškeré potřebné zdrojové kódy (například pokud nechcete během kompilace nechávat připojení k Internetu běžet) potom můžete použít volbu **--fetchonly** příkazu emerge, která pro vás veškeré potřebné zdrojové kódy stáhne.

Obrázek 6.15. Stahování zdrojových kódů

```
# emerge --fetchonly system
```

6.4.4. Sestavení systému

Pro začátek sestavování systému spusťte příkaz **emerge system** a poté se jděte něčím zabavit, protože provedení tohoto kroku potrvá dlouho.

Obrázek 6.16. Spuštění sestavování systému

```
# emerge system
```

Pokud sestavování systému skončilo, můžete přejít ke kapitole Konfigurace jádra.

Kapitola 7. Konfigurace jádra

Abstrakt

Linuxové jádro je srdcem každé distribuce. Tato kapitola vysvětluje jak svoje jádro správně nakonfigurovat.

7.1. Časové pásmo

Nejdříve potřebujete zvolit své časové pásmo aby systém věděl kde se nachází. Najděte své časové pásmo v `/usr/share/zoneinfo` a udělejte odpovídající symlink do `/usr/localtime` pomocí příkazu `ln`:

Obrázek 7.1. Nastavení informací o časovém pásmu

```
# ls /usr/share/zoneinfo
(Suppose you want to use GMT:)
# ln -sf /usr/share/zoneinfo/GMT /etc/localtime
```

7.2. Instalace zdrojových kódů

Srdcem okolo kterého je vystavěna každá distribuce je Linuxové jádro. Je to vrstva mezi uživatelskými programy a vašim systémovým hardwarem. Gentoo poskytuje svým uživatelům několik možných zdrojů jádra. Úplný seznam s popisem je možno najít v Gentoo Kernel Guide [<http://www.gentoo.org/doc/en/gentoo-kernel.xml>].

Pro systémy založené na architektuře x86 máme, kromě jiných jader, `vanilla-sources` (základní jádro vyvíjené přímo vývojáři jádra), `gentoo-sources` (jádro opatchované s ohledem na rychlost), `gentoo-dev-sources` (jádro řady 2.6 opatchované s ohledem na rychlost a stabilitu), `xfs-sources` (jádro v nejnovější podporou XFS), `gs-sources` (opatchované jádro pro serverové použití), `gaming-sources` (jádro opatchované pro optimální herní výkon), ...

Pro systémy založené na architektuře Alpha máme `vanilla-sources` (základní jádro vyvíjené přímo vývojáři jádra), `alpha-sources` (jádro optimalizované pro uživatele architektury Alpha) a `compaq-sources` (jádro používané firmou RedHat pro architekturu Alpha, udržované firmou Compaq).

Pro architekturu Sparc máme `vanilla-sources` (základní jádro vyvíjené přímo vývojáři jádra) a `sparc-sources` (jádro optimalizované pro uživatele architektury Sparc).

V případě architektury MIPS si lze vybrat z jader `mips-sources` (výchozí jádro pro architekturu MIPS) a `mips-prepatch-sources` (prerelease jaderný strom).

Pro architekturu AMD64 máme jádra `gentoo-dev-sources` (jádro řady 2.6 s patchi specifickými pro amd64 zaměřenými na stabilitu, výkon a podporu hardware).

Další architektury by měly používat jádro optimalizované speciálně pro ně: `hppa-sources` (HPPA) nebo `ppc-sources` (PowerPC).

Zvolte si zdrojové kódy jádra a nainstalujte je pomocí programu **emerge**. Od této chvíle budeme používat příkaz **emerge --usepkg** který nainstaluje předkompilovaný balíček pokud je dostupný (pro uživatele GRP) a v opačném případě ho stáhne. Jinými slovy, pokud nepoužíváte předkompilované balíčky, můžete vypustit volbu `--usepkg`, ale nemusíte. Navíc můžete volbu `--usepkg` zkrátit na pouhé `-k`.

V následujícím příkladě nainstalujeme jádro `vanilla-sources` (protože jádro `gentoo-sources` není na naší architektuře dostupné). Samozřejmě si do příkazu dosadíte jádro dle své volby:

Obrázek 7.2. Instalace zdrojových kódů jádra

```
# emerge --usepkg vanilla-sources
```

Když se nyní podíváte do adresáře `/usr/src`, měli byste najít symlink pojmenovaný `linux` a ukazující na zdrojové kódy vašeho jádra.

Obrázek 7.3. Zobrazení symlinku na zdrojové kódy jádra

```
# ls -l /usr/src/linux
lrwxrwxrwx    1 root    root              12 Oct 13 11:04 /usr/src/linux -> linux-2.4.22
```

Pokud to neodpovídá (tj. pokud symlink ukazuje na jiný zdroj jádra) změňte symlink než budete pokračovat dále.

Obrázek 7.4. Změna symlinku

```
# rm /usr/src/linux && ln -s /usr/src/linux-2.4.22 /usr/src/linux
```

Nyní nadešel čas ke konfiguraci a kompilaci zdrojových kódů jádra. Na architektuře x86 lze k tomuto účelu použít příkaz **genkernel**, který sestaví běžné jádro použité například na LiveCD. Vysvětlíme však také manuální konfiguraci, protože to je nejlepší způsob k optimalizaci vašeho prostředí.

Pokud chcete své jádro nakonfigurovat ručně, pokračujte v následující části Výchozí: Manuální konfigurace. Pokud jste uživatel systému s architekturou x86 a chcete použít program **genkernel**, měli byste přeskóčit až na část Náhradní: Použití programu genkernel.

7.3. Výchozí: Manuální konfigurace

7.3.1. Úvod

Manuální konfigurace jádra je často považována za nejsložitější zkoušku kterou musí každý uživatel Linuxu projít. Nic není tak nepravdivé - po nakonfigurování několika jader vás ani nenapadne že by to bylo složité ;)

Avšak jedna věc pravdivá je: musíte znát svůj systém když jádro začínáte konfigurovat ručně. Mnoho informací lze získat nahlédnutím do `/proc/pci` (nebo použitím příkazu **lspci** pokud je dostupný). Můžete také spustit **lsmmod** abyste zjistili jaké moduly LiveCD používá (což vám může napovědět co je potřeba v jádře povolit).

Nyní se přesuňte do adresáře se zdrojovými kódy jádra a spusťte příkaz **make menuconfig**. Tím spustíte konzolové (používající knihovnu ncurses) konfigurační menu.

Obrázek 7.5. Spuštění nástroje menuconfig

```
# cd /usr/src/linux
# make menuconfig
```

Objeví se nabídka s několika konfiguračními sekcemi. Nejdříve vyjmenujeme některé volby které musíte zapnout (jinak Gentoo nebude fungovat nebo nebude fungovat bez dalších úprav).

7.3.2. Aktivace povinných voleb

Nejdříve ze všeho zapněte podporu vývojových a experimentálních voleb a ovladačů. To potřebujete, jinak se některé velice důležité volby a ovladače vůbec nezobrazí:

Obrázek 7.6. Zapnutí podpory vývojových a experimentálních voleb a ovladačů

```
Code maturity level options --->
[*] Prompt for development and/or incomplete code/drivers
```

Nyní otevřete sekci `File Systems` (souborové systémy) a zvolte podporu souborových systémů které používáte. Nekompilujte je jako moduly, jinak Gentoo nebude schopno připojit vaše diskové oddíly. Vyberte také položky `Virtual memory`, `/proc file system` a `Automatically mount at boot`.

Obrázek 7.7. Volba potřebných souborových systémů

```
File systems --->
[*] Virtual memory file system support (former shm fs)
[*] /proc file system support
```

```
[*] /dev file system support (EXPERIMENTAL)
[*]   Automatically mount at boot

(Deselect the following unless you have a 2.6 kernel)
[ ] /dev/pts file system for Unix98 PTYs

(Select one or more of the following options as needed by your system)
<*> Reiserfs support
<*> Ext3 journalling file system support
<*> JFS filesystem support
<*> Second extended fs support
<*> XFS filesystem support
```

Poznámka

Uživatelé jader řady 2.6 najdou některé ze zmíněných voleb v sekci `Pseudo filesystems` což je podsekce sekce `File systems`.

Pokud k připojení k Internetu používáte protokol PPoE, budete potřebovat následující volby v jádře:

Obrázek 7.8. Konfigurace PPPoE voleb

```
Network device support --->
<*> PPP (point-to-point protocol) support
<*>   PPP support for async serial ports
<*>   PPP support for sync tty ports
```

Poznámka

Uživatelé jader řady 2.6 najdou některé ze zmíněných voleb v sekci `Networking support` což je podsekce sekce `Device drivers`.

Dvě volby týkající se komprese ničemu neuškodí ale nejsou nutné, a nutná není ani volba `PPP over Ethernet` protože je používána pouze programem **rp-pppoe** nakonfigurovaným pro "kernel mode PP-PoE".

Poznámka

Uživatelé jader řady 2.6 najdou tyto volby v sekci `Device drivers`.

Pokud ho potřebujete, nezapomeňte v jádře zapnout podporu pro svou Ethernetovou síťovou kartu.

Nyní, podle toho jakou používáte architekturu, byste mohli potřebovat následující volby:

- Aktivace voleb doporučených pro architekturu x86

- Aktivace voleb doporučených pro architekturu Alpha
- Aktivace voleb doporučených pro architekturu HPPA
- Aktivace voleb doporučených pro architekturu PPC
- Aktivace voleb doporučených pro architekturu SPARC
- Aktivace voleb doporučených pro architekturu MIPS
- Aktivace voleb doporučených pro architekturu AMD64

7.3.3. Aktivace voleb doporučených pro architekturu x86

Pokud máte procesor firmy Intel podporující technologii HyperThreading (tm), nebo pokud máte více-procesorový systém, měli byste aktivovat volbu "Symmetric multi-processing support":

Obrázek 7.9. Zapnutí podpory SMP

```
Processor type and features --->
[*] Symmetric multi-processing support
```

Pokud jste skončili s konfigurací jádra, pokračujte v odstavci Kompilace a Instalace.

7.3.4. Aktivace voleb doporučených pro architekturu Alpha

Pro uživatele architektury Alpha jsou doporučovány následující volby:

Obrázek 7.10. Volby doporučované pro architekturu Alpha

```
General setup --->
<*> SRM environment through procfs
<*> Configure uac policy via sysctl

Plug and Play configuration --->
<*> Plug and Play support
<M>   ISA Plug and Play support

SCSI support --->
  SCSI low-level drivers --->
    <*> SYM53C8XX Version 2 SCSI support (NEW)
    <*> Qlogic ISP SCSI support

Network device support --->
```

```
Ethernet (10 or 100 Mbit) --->
  <M> DECchip Tulip (dc21x4x) PCI support
  <M> Generic DECchip & DIGITAL EtherWORKS PCI/EISA
  <M> EtherExpressPro/100 support (eepro100)
  <M> EtherExpressPro/100 support (e100)
Ethernet (1000 Mbit) --->
  <M> Alteon AceNIC
  [*] Omit support for old Tigon I
  <M> Broadcom Tigon3
[*] FDDI driver support
<M> Digital DEFEA and DEFPA
<*> PPP support
  <*> PPP Deflate compression

Character devices --->
  [*] Support for console on serial port
  [*] Direct Rendering Manager

File systems --->
  <*> Kernel automounter version 4 support
Network File Systems --->
  <*> NFS
  [*] NFSv3 client
  <*> NFS server
  [*] NFSv3 server
Partition Types --->
  [*] Advanced partition selection
  [*] Alpha OSF partition support
Native Language Support
  <*> NLS ISO 8859-1

Sound --->
  <M> Sound card support
  <M> OSS sound modules
  [*] Verbose initialisation
  [*] Persistent DMA buffers
  <M> 100% Sound Blaster compatibles
```

Pokud jste skončili s konfigurací jádra, pokračujte v odstavci Kompilace a Instalace.

7.3.5. Aktivace voleb doporučených pro architekturu HP-PA

Pokud máte HIL myš nebo klávesnici, nezapomeňte pro ně zakompilovat podporu:

Obrázek 7.11. Zapnutí podpory HIL

```
Input core support --->
```

```
[*] Keyboard support
[*] Mouse support
[*] Event interface support
```

Pokud nemáte myš na HIL portu, potom postačí pouze základní podpora:

Obrázek 7.12. Základní HIL podpora

```
HIL support --->
[*] HIL Keyboard (basic) support
```

Ale pokud chcete plnou podporu HIL, potom zapněte následující volby:

Obrázek 7.13. Zapnutí plné podpory HIL

```
HIL support --->
[*] HP System Device Controller i8042 Support
[*] HIL MLC Support
[*] HIL Keyboard (full) support
[*] HIL Mouse & Pointer support
```

Pokud jste skončili s konfigurací jádra, pokračujte v odstavci Kompilace a Instalace.

7.3.6. Aktivace voleb doporučených pro architekturu PPC

Nejdříve ze všeho vypněte podporu "ADB raw keycodes":

Obrázek 7.14. Vypnutí podpory ADB raw keycodes

```
Macintosh Device Drivers --->
[ ] Support for ADB raw keycodes
```

Vyberte také podporu korektního RTC (odznačte volbu "Enhanced RTC"):

Obrázek 7.15. Zapnutí podpory korektního RTC

```
Character devices --->
[ ] Enhanced RTC

General setup --->
[*] Support for /dev/rtc
```

Uživatelé OldWorld systémů potřebují zapnout podporu HFS aby mohli zkopírovat zkompilevané jádro na MacOS diskový oddíl.

Obrázek 7.16. Zapnutí podpory HFS

```
File Systems --->
[*] HFS Support
```

Pokud jste skončili s konfigurací jádra, pokračujte v odstavci Kompilace a Instalace.

7.3.7. Aktivace voleb doporučených pro architekturu SPARC

Nejdříve zapněte správnou podporu sběrnice:

Obrázek 7.17. Zapnutí podpory SBUS/UPA

```
Console drivers --->
Frame-buffer support --->
  [*] SBUS and UPA framebuffers
  [*] Creator/Creator3D support      (Only for UPA slot adapter used in many Ultras)
  [*] CGsix (GX,TurboGX) support    (Only for SBUS slot adapter used in many SPARCStation)
```

Samozřejmě chcete také podporu pro OBP:

Obrázek 7.18. Zapnutí podpory OBP

```
Misc Linux/SPARC drivers --->
[*] /dev/openprom device support
```

Potřebovat budete také podporu SCSI:

Obrázek 7.19. Zapnutí podpory SCSI

```
SCSI support --->
  SCSI low-level drivers --->
    <*> Sparc ESP Scsi Driver          (Only for SPARC ESP on-board SCSI adapter)
    <*> PTI Qlogic, ISP Driver          (Only for SBUS SCSI controllers from PTI or QLogi
    <*> SYM53C8XX Version 2 SCSI support (Only for Ultra 60 on-board SCSI adapter)
```

Pro podporu své síťové karty zapněte následující:

Obrázek 7.20. Aktivace podpory síťových karet

```
Network device support --->
  Ethernet (10 or 100Mbit) --->
    <*> Sun LANCE support                (Only for SPARCStation, older Ultra systems, an
    <*> Sun Happy Meal 10/100baseT support (Only for Ultra; also supports "qfe" quad-ether
```

Až skončíte s konfigurací jádra, pokračujte v části Kompilace a Instalace. Nicméně až kompilace skončí, zkontrolujte jeho velikost:

Obrázek 7.21. Kontrola velikosti jádra

```
# ls -lh vmlinux
-rw-r--r--  1 root    root      2.4M Oct 25 14:38 vmlinux
```

Pokud je (nekomprimovaná) velikost vyšší než 2.5 MB (pro Sparc32) nebo 3.5 MB (pro Sparc64), pře-konfigurujte jádro tak aby příslušný limit nepřekračovalo. Jednou z možností jak toho dosáhnout je zkompileovat potřebnou část ovladačů jako moduly. Ignorování tohoto limitu může vést až k nenastartování jádra.

Také, pokud je vaše jádro pouze o chlup větší, můžete zkusit použít příkaz strip:

Obrázek 7.22. Stripping jádra

```
# strip -R .comment -R .note vmlinux
```

7.3.8. Aktivace voleb doporučených pro architekturu

MIPS

Pokud používáte systém založený na Indy/Indigo2, musíte pro něj zapnout podporu.

Obrázek 7.23. Zapnutí podpory pro SGI IP22

```
Machine selection --->
[*] Support for SGI IP22 (Indy/Indigo2)
```

Pokud chcete spouštět binární soubory určené pro Irix, zapněte následující volbu:

Obrázek 7.24. Zahrnutí podpory binárních souborů Irix

```
General setup --->
[*] Include IRIX binary compatibility
```

Pokud ve své stanici SGI Indigo2 máte ISA/EISA karty, zapněte pro ně podporu.

Obrázek 7.25. Zapnutí podpory SGI Indigo2 pro ISA/EISA

```
General setup --->
[*] Indigo-2 (IP22) EISA bus support
[*]   ISA bus support
```

Pokud má vaše SGI stanice paralelní port, můžete pro něj zapnout podporu. Pokud máte ISA paralelní port měli byste namísto toho zapnout "PC-type hardware".

Obrázek 7.26. Zapnutí podpory SGI paralelního portu

```
Parallel port support --->
<*> Parallel port support
<*>   SGI Indy/Indigo2 hardware (EXPERIMENTAL) (NEW)
<*>   IEEE 1284 transfer modes (NEW)
```

Pokud chcete použít ISA sloty ve stanici Indigo2, zapněte podporu plug a play.

Obrázek 7.27. Zapnutí podpory PnP

```
Plug and Play configuration --->
<*> Plug and Play support
<*>   ISA Plug and Play support
```

Nezapomeňte zapnout podporu SCSI a zvolit ovladač pro SGI WD93C93:

Obrázek 7.28. Zapnutí ovladače SGI WD93C93

```
SCSI low-level drivers --->
<*> SGI WD93C93 SCSI Driver
```

Pro síťové karty pravděpodobně budete potřebovat zapnout podporu pro ethernetový řadič SGI Seeq :

Obrázek 7.29. Zapnutí podpory SGI Seeq

```
Network device support --->
  Ethernet (10 or 100Mbit) --->
    [*] Ethernet (10 or 100Mbit)
    [*]   SGI Seeq ethernet controller support
```

Nezapomeňte zapnout podporu sériové konzole a podporu pro SGI Zilog85C30:

Obrázek 7.30. Zapnutí podpory pro SGI Zilog85C30

```
Character devices --->
  [*] Non-standard serial port support
  [*]   SGI Zilog85C30 serial support
```

Nezapomeňte také zapnout podporu pro Indy/I2 Watchdog ani na podporu SGI DS1286 RTC:

Obrázek 7.31. Podpora Watchdog a SGI DS1286 RTC

```
Character Devices --->
  [*] SGI DS1286 RTC support
  Watchdog Cards --->
    [*] Watchdog Timer Support
    <*>   Indy/I2 Hardware Watchdog
```

Měli byste také zapnout podporu pro SGI diskové oddíly :)

Obrázek 7.32. Zapnutí podpory pro SGI diskové oddíly

```
File Systems --->
Partition Types --->
  [*] Advanced partition selection
  [*]   SGI partition support
```

Pokud máte kartu SGI Newport (XL Gfx) Card a chcete ji používat, potom byste zřejmě měli povolit její podporu:

Obrázek 7.33. Zapnutí podpory pro SGI Newport Card

```
Console drivers --->
<*> SGI Newport Console support (NEW)
```

Pokud chcete plnou podporu zvuku pro Indy/Indigo2, zapněte pro ji:

Obrázek 7.34. Zapnutí podpory zvuku pro Indy/Indigo2

```
Sound --->
<*> Sound card support
<*>   SGI HAL2 sound (EXPERIMENTAL)
```

Pokud jste skončili s konfigurací jádra, pokračujte v odstavci Kompilace a Instalace.

7.3.9. Aktivace voleb doporučených pro architekturu AMD64

Pokud máte systém s více procesory Opteron, měli byste zapnout podporu "Symmetric multi-processing support":

Obrázek 7.35. Zapnutí podpory SMP

```
Processor type and features --->
  [*] Symmetric multi-processing support
```

Pokud jste skončili s konfigurací jádra, pokračujte v odstavci **Kompilace a Instalace**.

7.3.10. Kompilace a instalace

Nyní když máte jádro nakonfigurováno, je čas jej zkompileovat a nainstalovat. Opusťte konfigurační menu a spusťte příkaz **make dep && make bzImage modules modules_install**:

Obrázek 7.36. Kompilace jádra

```
(For x86-based systems, 2.4 kernel)
# make dep && make bzImage modules modules_install

(For other systems, 2.4 kernel)
# make dep && make vmlinux modules modules_install

(For amd64-based systems, 2.6 kernel)
# make bzImage modules modules_install

(For other systems, 2.6 kernel)
# make && make modules_install
```

Až kompilace jádra skončí, zkopírujte obraz do adresáře `/boot`. V následujícím příkladě předpokládáme že jste nakonfigurovali a zkompilevali jádro `vanilla-sources-2.4.22` (což ale nemusí být to správné jádro pro vaši architekturu).

Obrázek 7.37. Instalace jádra

```
(For x86-based systems)
# cp arch/i386/boot/bzImage /boot/kernel-2.4.22
# cp System.map /boot/System.map-2.4.22

(For amd64-based systems)
# cp arch/x86_64/boot/bzImage /boot/kernel-2.4.22

(For other systems)
# cp vmlinux /boot/kernel-2.4.22
# cp System.map /boot/System.map-2.4.22
```

Je také rozumné zkopírovat si konfigurační soubor do adresáře `/boot`, pro případ nouze :)

Obrázek 7.38. Záloha vaší konfigurace jádra

```
# cp .config /boot/config-2.4.22
```

Pokud jste uživatelé systému MIPS a pokud váš systém nepodporuje start ELF jádra, zkompilujte jádro pomocí **make vmlinux.ecoff** namísto **make vmlinux**. Obraz jádra bude uložen jako `arch/mips/boot/vmlinux.ecoff` namísto `vmlinux`.

Nyní pokračujte v části Instalace samostatných modulů jádra.

7.4. Alternativa: Použití programu genkernel

Pokud čtete tento odstavec, zvolili jste si použití skriptu **genkernel** který zkompiluje jádro za vás. Avšak to také znamená že máte systém s architekturou x86. Jiné architektury nejsou skriptem **genkernel** podporovány.

Nyní máte nainstalován strom zdrojového kódu, a nadešel čas jej zkompilovat pomocí skriptu **genkernel**. Skript **genkernel** funguje tak že nakonfiguruje kernel téměř stejně jako jádro na našem LiveCD. To znamená že když použijete **genkernel** pro sestavení jádra, váš systém obecně najde veškerý váš hardware při startu, stejně jako LiveCD. Protože skript **genkernel** nepotřebuje žádnou ruční konfiguraci, je to ideální řešení kdo se necítí na to nakonfigurovat si vlastní jádro.

Podívejme se jak použít skript **genkernel**. Zprvė spustěte `emerge` na `genkernel ebuild`:

Obrázek 7.39. Emerging genkernel

```
# emerge --usepkg genkernel
```

Nyní zkompilujte zdrojové kódy jádra. Pokud používáte **genkernel** nainstalovaný přes GRP, spustěte **genkernel**. Pokud máte čerstvý balíček **genkernel** (staženou verzi), spustěte **genkernel all**. Ale pozor, protože **genkernel** zkompiluje jádro podporující téměř všechny hardware, bude kompilace trvat dosti dlouho!

Obrázek 7.40. Spuštění skriptu genkernel

```
(This example is for the GRP genkernel)
# genkernel
Gentoo Linux genkernel, version 1.4
Copyright 2003 Gentoo Technologies, Inc., Bob Johnson, Daniel Robbins
Distributed under the GNU General Public License version 2

Settings:
compile optimization: 1 processor(s)
source tree: /usr/src/linux-2.4.22
config: gentoo (customized)
config loc: /etc/kernels/config-2.4.22
initrd config: (default) /etc/kernels/settings

* Running "make oldconfig"...
* Logging to /var/log/genkernel.log...
```

```
* Starting 2.4.22 build...
* Running "make dep"...
* Running "make bzImage"...
* Running "make modules"...
* Running "make modules_install"...
* Moving bzImage to /boot/kernel-2.4.22...
* Building busybox...
* Creating initrd...

* Build completed successfully!

* Please specify /boot/kernel-2.4.22 and /boot/initrd-2.4.22
* when customizing your boot loader configuration files.
```

Jakmile skript **genkernel** skončí, budou vytvořeny jádro, plná sada modulů a "initial root disk" (initrd). Jádro a initrd použijeme dále v tomto dokumentu při konfiguraci zavaděče. Zapište si jméno jádra a initrd protože je budete potřebovat při konfiguraci zavaděče. Soubor initrd bude spuštěn okamžitě po nastartování aby mohl provést autodetekci hardware (přesně jako LiveCD) ještě před tím než nastartuje "skutečný" systém.

Nyní provedme ještě jeden krok k tomu aby byl náš systém více jako LiveCD - **emerge hotplug**. Zatímco initrd provádí autodetekci hardware potřebného pro start systému, **hotplug** provádí autodetekci všeho ostatního. Pro stažení a zapnutí balíčku **hotplug** zadejte následující:

Obrázek 7.41. Stažení a zapnutí hotplug

```
# emerge --usepkg hotplug
# rc-update add hotplug default
```

7.5. Instalace samostatných modulů jádra

7.5.1. Instalace doplňujících modulů

Pokud je to potřeba, měli byste si stáhnout moduly pro další hardware ve svém systému. Zde je seznam s jádrem souvisejících balíčků které si můžete stáhnout pomocí příkazu emerge:

Ebuild	Účel	Příkaz
nvidia-kernel	Akcelerované ovladače grafických karet s čipem NVIDIA pro XFree86	emerge --usepkg nvidia-kernel
nforce-net	Ovladač pro síťovou kartu umístěnou na Motherboardech s čipem NForce(2)	emerge nforce-net

Ebuild	Účel	Příkaz
nforce-audio	Ovladač pro zvukovou kartu umístěnou na Motherboardech s čipem NForce(2)	emerge nforce-audio
e100	Intel e100 Fast Ethernet Adapters	emerge e100
e1000	Intel e1000 Gigabit Ethernet Adapters	emerge e1000
emu10k1	Podpora Creative Sound Blaster Live!/Audigy	emerge emu10k1
ati-drivers	Akcelerované ovladače grafických karet ATI Radeon 8500+/FireGL pro XFree86	emerge ati-drivers
xfree-drm	Akcelerované ovladače grafických karet ATI Radeon až do 9200, Rage128, Matrox, Voodoo a dalších karet pro XFree86. Přečtete si proměnnou IUSE_VIDEO_CARDS v /usr/portage/x11-base/xfree-drm ebuilds abyste zjistili co je potřeba doplnit na místo yourcard	VIDEO_CARDS="yourcard" emerge xfree-drm

Avšak pozor, protože některé z těchto balíčků mohou mít rozsáhlé závislosti. Abyste ověřil které balíčky budou nainstalovány stažením určitého balíčku, použijte příkaz **emerge --pretend**. Například pro balíček emu10k1:

Obrázek 7.42. Zobrazení úplného seznamu instalovaných balíčků

```
# emerge --pretend emu10k1
```

Pokud se vám nelíbí seznam balíčků které by se nainstalovaly, použijte příkaz **emerge --pretend -verbose** kterým zjistíte jaké USE příznaky byly zvoleny při rozhodování o závislostech:

Obrázek 7.43. Zobrazení použitých USE příznaků

```
# emerge --pretend --verbose emu10k1
...
[ebuild N ] media-sound/aumix-2.8 +gpm +nls +gtk +gnome +alsa -gtk2
```

V předchozím příkladě je vidět že jedna ze závislostí balíčku emu10k1 (aumix) používá USE příznaky gtk a gnome, což znamená že gtk (závisející na XFree) bude kompilováno s ním.

Pokud nechcete aby tomu tak bylo, odstraňte všechny USE příznaky, například:

Obrázek 7.44. Stažení emu10k1 se všemi USE příznaky odstraněnými

```
# USE="-gpm -nls -gtk -gnome -alsa" emerge --pretend emu10k1
```

Pokud jste nyní s výsledkem spokojeni, odstraňte z příkazu volbu **--pretend** a balíček emu10k1 bude nainstalován.

7.5.2. Konfigurace modulů

Pokud nepoužíváte hotplug, měli byste v souboru `/etc/modules.autoload.d/kernel-2.4` (nebo `kernel-2.6`) uvést seznam modulů které chcete automaticky načíst. K modulům můžete přidat i některé zvláštní volby, pokud chcete.

Pro zobrazení seznamu všech dostupných modulů spusťte následující příkaz **find**. Nezapomeňte za "`<kernel version>`" dosadit verzi jádra kterou jste právě zkompilevali:

Obrázek 7.45. Zobrazení všech dostupných modulů

```
# find /lib/modules/<kernel version>/ -type f -iname '*.o' -or -iname '*.ko'
```

Například pro automatické spuštění modulu 3c59x.o:

Obrázek 7.46. /etc/modules.autoload.d/kernel-2.4 or kernel-2.6

```
3c59x
```

Nyní spusťte příkaz `modules-update` pro potvrzení změn v souboru `/etc/modules.conf`:

Obrázek 7.47. Spuštění modules-update

```
# modules-update
```

Pokračujte v instalaci dle kapitoly Konfigurace systému.

Kapitola 8. Konfigurace systému

Abstrakt

Nyní musíte upravit několik důležitých konfiguračních souborů. V této kapitole vám předložíme přehled těchto souborů a návod jak postupovat.

8.1. Informace o souborovém systému

8.1.1. Co je to fstab?

V Linuxu musí být všechny diskové oddíly používané systémem uvedeny v souboru `/etc/fstab`. Tento soubor obsahuje místa připojení (mountpoints) těchto diskových oddílů (tj. místa kde se objeví v adresářové struktuře), jakým způsobem by měly být připojeny (zvláštní nastavení) a kdy (automaticky či manuálně, zda je mohou připojit uživatelé, atd.).

8.1.2. Vytvoření souboru `/etc/fstab`

Soubor `/etc/fstab` má zvláštní syntaxi. Každá řádka se skládá ze šesti položek, oddělených bílými znaky (mezerami, tabulátory nebo jejich kombinací). Každá položka má svůj vlastní význam:

- První položkou je příslušný *diskový oddíl* (cesta k souboru zařízení).
- Druhou položkou je *přípojně místo* (tzv. mountpoint) ke kterému by diskový oddíl měl být připojen.
- Třetí položka označuje *souborový systém* použitý na diskovém oddíle.
- Čtvrtá položka uvádí *zvláštní nastavení* používaná příkazem **mount** při připojování diskového oddílu. Protože každý souborový systém má svá vlastní zvláštní nastavení, doporučujeme vám přečíst si manuálovou stránku (**man mount**) pro jejich kompletní seznam. Pokud chcete uvést několik zvláštních nastavení, oddělte je čárkou.
- Pátá položka je používána příkazem **dump** pro určení zda diskový oddíl má být zpracován nebo ne. Obecně zde můžete nechat hodnotu 0 (nula).
- Šestá položka je používána příkazem **fsck** pro určení v jakém pořadí mají být souborové systémy kontrolovány při nesprávném vypnutí systému. Kořenový adresář by zde měl mít nastaveno 1 zatímco ostatní diskové oddíly by zde měly mít nastaveno 2 (nebo 0 pokud není kontrola nutná).

Takže spusťme **nano2** (nebo váš oblíbený editor) a vytvořte `/etc/fstab`:

²Poznámka překladatele: Editor **nano** (v závislosti na konfiguraci a na parametrech kompilace) automaticky zalamuje řádky takovým způsobem, že pátá a šestá položka se mohou objevit na samostatném řádku (což je špatně). V tom případě použijte při spuštění přepínač `-w`, kterým toto chování vypnete.

Obrázek 8.1. Otevření /etc/fstab

```
# nano -w /etc/fstab
```

Podívejme se napsat volby pro diskový oddíl `/boot`. Jedná se pouze o příklad, takže pokud vaše architektura nevyžaduje diskový oddíl `/boot`, nekopírujte to doslovně.

V našem vzorovém x86 rozdělení disku je `/boot` na diskovém oddílu `/dev/hda1` a jako souborový systém je použit `ext2`. Tento souborový systém by neměl být připojován automaticky (`noauto`) ale měl by být kontrolován. Takže konfigurace by mohla být:

Obrázek 8.2. Příklad konfigurační řádky pro /boot v /etc/fstab

```
/dev/hda1 /boot ext2 noauto 1 2
```

Nyní, by většina uživatelů pro zvýšení výkonnosti přidala volbu `noatime`, což zrychlí systém protože nebudou zaznamenávány časy přístupu (které stejně většinou nepotřebujete):

Obrázek 8.3. Vylepšená konfigurační řádka pro /boot v /etc/fstab

```
/dev/hda1 /boot ext2 noauto,noatime 1 2
```

Kdybychom takhle pokračovali dále, skončili bychom s následujícími třemi linkami (pro `/boot`, `/` a odkládací oddíl):

Obrázek 8.4. Tři řádky v /etc/fstab

```
/dev/hda1 /boot ext2 noauto,noatime 1 2
/dev/hda2 none swap sw 0 0
/dev/hda3 / ext3 noatime 0 1
```

Nakonec bychom měli doplnit konfiguraci pro `/proc`, `tmpfs` (povinně) a pro vaše CD-ROM mechaniky (a pochopitelně i pro další diskové oddíly, pokud nějaké máte):

Obrázek 8.5. Kompletní konfigurace /etc/fstab

/dev/hda1	/boot	ext2	noauto,noatime	1 2
/dev/hda2	none	swap	sw	0 0
/dev/hda3	/	ext3	noatime	0 1
none	/proc	proc	defaults	0 0
none	/dev/shm	tmpfs	defaults	0 0
/dev/cdroms/cdrom0	/mnt/cdrom	auto	noauto,user	0 0

Volba `auto` umožňuje hádat souborový systém (doporučeno pro výměnná média protože ta mohou být vytvářena s jedním z mnoha souborových systémů) a volba `user` umožňuje připojení CD disků i běžným uživatelům (a nikoliv jen rootovi).

Nyní použijte právě uvedené příklady pro vytvoření vlastního souboru `/etc/fstab`. Pokud používáte systém s architekturou Sparc, měli byste do `/etc/fstab` přidat také následující řádek:

Obrázek 8.6. Doplnění souborového systému openprom do /etc/fstab

none	/proc/openprom	openpromfs	defaults	0 0
------	----------------	------------	----------	-----

Pokud potřebujete souborový systém `usbfs`, přidejte do `/etc/fstab` také následující řádek:

Obrázek 8.7. Doplnění souborového systému usbfs do /etc/fstab

none	/proc/bus/usb	usbfs	defaults	0 0
------	---------------	-------	----------	-----

Ještě jednou si pro jistotu soubor `/etc/fstab` přečtěte, uložte jej a ukončete editor.

8.2. Informace o nastavení sítě

8.2.1. Hostname, Domainname, atd.

Jedno z rozhodnutí která musí uživatel učinit je pojmenování svého PC. Vypadá to jako jednoduchý úkol, ale mnoho uživatelů má s nalezením vhodného jména pro své Linuxové PC problémy. Stačí si však uvědomit že zvolené jméno můžete kdykoliv později změnit. Takže nám bude stačit jméno `tux` a jméno domény `homenetwork`.

Tyto hodnoty používáme v následujících příkladech. Nejdříve nastavíme `hostname`:

Obrázek 8.8. Nastavení hostname

```
# echo tux > /etc/hostname
```

A potom nastavíme domainname:

Obrázek 8.9. Nastavení domainname

```
# echo homenetwork > /etc/dnsdomainname
```

Pokud máte NIS doménu (pokud nevíte co to je tak nic takového nemáte), potřebujete nastavit také ji:

Obrázek 8.10. Nastavení jména NIS domény

```
# echo nis.homenetwork > /etc/nisdomainname
```

Nyní doplňte skript **domainname** do základního runlevelu:

Obrázek 8.11. Zařazení skriptu domainname do implicitního runlevelu

```
# rc-update add domainname default
```

8.2.2. Konfigurace sítě

Než vás přepadne pocit "Hele, tohle už jsme ale udělali", měli byste si uvědomit že nastavení sítě prováděné hned na začátku instalace bylo určeno jen a pouze pro instalaci. Teď budeme konfigurovat síť pro váš stálý Gentoo systém.

Veškeré nastavení sítě je ukládáno v `/etc/conf.d/net`. Používá přímou i když ne zcela intuitivní syntaxi pro ty kteří neumí síťové připojení nastavit ručně. Ale nebojte se, všechno vám vysvětlíme :)

Nejdříve tedy otevřete soubor `/etc/conf.d/net` ve svém oblíbeném editoru (v našich příkladech je používán **nano**):

Obrázek 8.12. Otevření /etc/conf.d/net pro editaci

```
# nano -w /etc/conf.d/net
```

První proměnná na kterou narazíte je `iface_eth0`, která používá následující syntaxi:

Obrázek 8.13. Syntaxe `iface_eth0`

```
iface_eth0="<your ip address> broadcast <your broadcast address> netmask <your netmask>"
```

Pokud používáte DHCP (automatické získávání IP adresy), měli byste hodnotu `iface_eth0` nastavit na `dhcp`. Pokud používáte `rp-pppoe` (čili například ADSL), použijte hodnotu `up`. Pokud potřebujete síť nastavit ručně a výše uvedené pojmy vám nic neříkají, přečtěte si prosím část Rozumíme síťové terminologii.

Podívejme se na následující tři příklady; první používá DHCP, druhý statickou IP adresu (192.168.0.2) se síťovou maskou 255.255.255.0, všesměrovou adresou 192.168.0.255 a bránou 192.168.0.1 zatímco třetí pouze aktivuje rozhraní pro použití pomocí `rp-pppoe`:

Obrázek 8.14. Příklady pro `/etc/conf.d/net`

```
(For DHCP:)
iface_eth0="dhcp"

(For static IP:)
iface_eth0="192.168.0.2 broadcast 192.168.0.255 netmask 255.255.255.0"
gateway="eth0/192.168.0.1"

(For rp-pppoe:)
iface_eth0="up"
```

Pokud máte několik síťových rozhraní, vytvořte další `iface_eth` proměnné (tj. samostatné řádky začínající `iface_eth`), jako například `iface_eth1`, `iface_eth2` atd. Brána byste neměli opakovat, protože pro každý počítač můžete uvést pouze jednu.

Nyní uložte konfiguraci a ukončete editor.

8.2.3. Automatické zapnutí sítě při startu

Aby vaše síťová rozhraní byla aktivována při startu, potřebujete je doplnit do implicitního runlevelu. Pokud máte PCMCIA rozhraní měli byste tuto akci přeskočit protože PCMCIA rozhraní jsou aktivována v PCMCIA init skriptu.

Obrázek 8.15. Zařazení `net.eth0` do implicitního runlevelu

```
# rc-update add net.eth0 default
```

Pokud máte několik síťových rozhraní, potřebujete pro ně vytvořit příslušné inicializační skripty `net.eth1`, `net.eth2` atd. K tomu můžete použít příkaz **ln**:

Obrázek 8.16. Tvorba dalších inicializačních skriptů

```
# cd /etc/init.d
# ln -s net.eth0 net.eth1
# rc-update add net.eth1 default
```

8.2.4. Zápis informací o síti

Nyní potřebujete svůj Linuxový systém informovat o své síti. Tyto informace jsou definovány v `/etc/hosts` a pomáhají na IP adresy překládat hostnames (jména počítačů) které nejsou uvedeny na jmenovém serveru (nameserver). Například pokud se vaše lokální síť skládá ze tří PC pojmenovaných jenny (192.168.0.5), benny (192.168.0.6) a tux (192.168.0.7 - váš systém) otevřeli byste soubor `/etc/hosts` a vložili byste do něj následující hodnoty:

Obrázek 8.17. Otevření `/etc/hosts`

```
# nano -w /etc/hosts
```

Obrázek 8.18. Zadání informací o síti

```
127.0.0.1    localhost
192.168.0.5  jenny.homenetwork jenny
192.168.0.6  benny.homenetwork benny
192.168.0.7  tux.homenetwork tux
```

Pokud je váš systém samostatný (nebo pokud jmenné servery ošetří všechna jména správně) postačí pouze jediná řádka:

Obrázek 8.19. `/etc/hosts` pro samostatný a plně integrovaný počítač

```
127.0.0.1    localhost    tux
```

Uložte soubor a ukončete editor.

Pokud nemáte PCMCIA, můžete nyní pokračovat v části Informace o systému. Uživatelé PCMCIA systému by si měli přečíst následující odstavec o PCMCIA.

8.2.5. Volitelně: Zprovoznění PCMCIA

Uživatelé PCMCIA periférií by si nejdříve ze všeho měli nainstalovat balíček **pcmcia-cs**:

Obrázek 8.20. Instalace balíčku pcmcia-cs

```
# emerge --usepkg pcmcia-cs
```

Když ho máte nainstalovaný, přidejte **pcmcia** do implicitního runlevelu:

Obrázek 8.21. Zařazení skriptu pcmcia do implicitního runlevelu

```
# rc-update add pcmcia default
```

8.3. Informace o systému

Gentoo používá soubor `/etc/rc.conf` pro obecné, systémové nastavení. Otevřete `/etc/rc.conf` a vychutnejte si komentáře v tomto souboru :)

Obrázek 8.22. Otevření `/etc/rc.conf`

```
# nano -w /etc/rc.conf
```

Jak vidíte, tento soubor je dobře okomentován aby vám pomohl nastavit potřebné konfigurační hodnoty. Dávejte pozor zejména na nastavení KEYMAP: pokud zvolíte chybnou hodnotu dostane se vám velmi podivných výsledků psaní na klávesnici.

Poznámka

Uživatelé USB systémů architektury Sparc a klonů by měli namísto "sunkeymap" zvolit i386 keymapu (jako například "us").

Pokud jste dokonfigurovali soubor `/etc/rc.conf`, uložte jej a ukončete editor. Nyní můžete přejít k následující kapitole Konfigurace zavaděče.

Kapitola 9. Konfigurace zavaděče

Abstrakt

Existuje několik zavaděčů operačního systému a každý se konfiguruje jinak. V této části vám popíšeme všechny možnosti a provedeme vás konfigurací zavaděče tak aby vyhovoval vašim nárokům.

9.1. Volba zavaděče

9.1.1. Úvod

Nyní když máte jádro nakonfigurováno a zkompileováno a potřebné konfigurační soubory jsou správně vyplněny, nadešel okamžik instalace programu který při startu systému načte vámi zkompileované jádro. Takový program se jmenuje zavaděč (bootloader). Ale než začneme, zvažte možnosti ...

Na Linuxu je dostupných několik zavaděčů. Avšak tyto zavaděče fungují pouze na malé množině architektur, takže si musíte vybrat mezi zavaděči které podporují tu vaši ...

Následující tabulka obsahuje seznam architektur a zavaděčů které je podporují. Zvolte si zavaděč který podporuje vaši architekturu. Například pokud máte Pentium IV, potom vaše architektura je x86 a musíte si tedy vybrat mezi zavaděčem GRUB (doporučovaná volba) a LILO.

Architektura	Doporučovaný zavaděč	Alternativy
x86	GRUB	LILO
alpha	aBoot	MILO
sparc	SILO	
ppc	yaBoot (NewWorld), BootX (OldWorld)	
hppa	PALO	
mips	MIPS ROM	
amd64	GRUB	

9.1.2. Volitelně: Framebuffer

Poznámka

Tato část platí pouze pro systémy s architekturou x86, amd64 a ppc které mají ve zkompileovaném jádře zahrnutu podporu framebufferu. To zahrnuje i ty kdo pro konfiguraci jádra použili skript **genkernel**.

Pokud jste jádro nakonfigurovali s podporou framebufferu a pokud vyžadujete podporu framebufferu, musíte do konfigurace zavaděče doplnit řádek s konfigurací vga. Následující tabulka obsahuje výčet

možných hodnot které při této konfiguraci můžete použít. Ve vzorové konfiguraci používáme mód 800x600@16bpp, čili hodnotu 788.

	640x480	800x600	1024x768	1280x1024
8bpp	769	771	773	775
16bpp	785	788	791	794
24bpp	786	789	792	795

Nezapomeňte si někde zapsat zvolenou hodnotu, protože ji už za chvíli budete potřebovat.

Nyní si z výše uvedeného výčtu zvolte zavaděč.

9.2. Výchozí: GRUB

9.2.1. Rozumíme terminologii GRUBu

Poznámka

GRUB lze použít pouze pro systémy s architekturou x86 nebo amd64!

Nejkritičtější na pochopení toho jak funguje GRUB je sžítí se s tím jak se GRUB odkazuje na diskové oddíly. Váš diskový oddíl `/dev/hda1` je v GRUBu označován jako `(hd0,0)`. Všimněte si závorek okolo `hd0,0` - jsou povinné.

Pevné disky jsou počítány od nuly namísto od "a" a diskové oddíly začínají na nule namísto na jedničce. Dejte si také pozor na to že jsou počítány pouze pevné disky a nikoliv atapi zařízení jako například CD přehrávače a vypalovačky. To samé platí pro SCSI mechaniky. (Obvykle dostanou vyšší čísla než IDE disky pokud není BIOS nakonfigurován tak aby startoval ze SCSI zařízení.)

Předpokládejme že máte pevný disk na `/dev/hda`, CD-ROM přehrávač na `/dev/hdb`, vypalovačku na `/dev/hdc` a druhý pevný disk na `/dev/hdd` a žádný SCSI pevný disk. V tom případě je diskový oddíl `/dev/hdd7` přeložen jako `(hd1,6)`. Může se to zdát ošidné a asi i je, ale jak uvidíme tak GRUB nabízí automatické doplňování pro ty z vás kteří mají spoustu pevných disků a diskových oddílů a kteří se v číslování GRUBu poněkud ztrácejí.

Pokud máte pocit že jste se s tím trochu sžili, je čas nainstalovat GRUB.

9.2.2. Instalace zavaděče GRUB

Pro nainstalování programu GRUB stačí použít **emerge**. Uživatelé systémů s architekturou x86 musí nainstalovat `grub`, uživatelé systémů s architekturou AMD64 musí instalovat `grub-static`:

Obrázek 9.1. Instalace zavaděče GRUB

(For the x86 architecture:)

```
# emerge --usepkg grub

(For the AMD64 architecture:)
# emerge --usepkg grub-static
# cp -Rpv /usr/share/grub/i386-pc/* /boot/grub
```

Pokud chcete začít konfigurovat GRUB, stačí pouze zadat příkaz **grub**. Objeví se příkazová řádka *grub>* tj. příkazová řádka programu GRUB. Teď potřebujete zadat ty správné příkazy pro nainstalování startovacího záznamu programu GRUB na váš pevný disk.

Obrázek 9.2. Spuštění prostředí programu GRUB

```
# grub
```

Ve vzorové konfiguraci budeme chtít program GRUB nainstalovat tak aby četl informaci ze startovacího diskového oddílu `/dev/hda1` a aby umístil startovací záznam programu GRUB do MBR (master boot record) našeho pevného disku takže první věc která se objeví po zapnutí počítače bude příkazová řádka programu GRUB. Samozřejmě pokud jste se nedrželi přesně naší vzorové konfigurace upravte odpovídajícím způsobem použité příkazy.

Automatické doplňování GRUBU lze použít z jeho příkazové řádky. Například pokud zadáte "root (" a stisknete TAB (tj. tabulátor), objeví se seznam zařízení (jako je například `hd0`). Pokud zadáte "root (`hd0`," a opět stisknete TAB, objeví se seznam dostupných diskových oddílů na daném zařízení (jako je například `hd0,0`).

S použitím automatického doplňování by konfigurace zavaděče GRUB neměla být příliš těžká. Tak do toho, nakonfigurujeme GRUB, ne? :-)

Obrázek 9.3. Instalace GRUBu do MBR

```
grub> root (hd0,0)          (Specify where your /boot partition resides)
grub> setup (hd0)           (Install GRUB in the MBR)
grub> quit                  (Exit the GRUB shell)
```

Poznámka

Pokud chcete GRUB nainstalovat přímo do některého diskového oddílu a nikoliv do MBR, musíte změnit příkaz **setup** tak aby odkazoval na správný diskový oddíl. Například pokud chcete GRUB nainstalovat do diskového oddílu `/dev/hda3`, potom použijte **setup (hd0,2)**. Ale to bude chtít jenom několik málo uživatelů.

Ačkoliv GRUB už je nyní nainstalován, ještě pro něj musíme napsat konfigurační soubor, aby GRUB

mohl automaticky zavést nově vytvořené jádro. Vytvořte soubor `/boot/grub/grub.conf` pomocí editoru **nano** (nebo kterýmkoliv jiným editorem):

Obrázek 9.4. Vytváření souboru `/boot/grub/grub.conf`

```
# nano -w /boot/grub/grub.conf
```

Nyní napíšeme `grub.conf`. Níže najdete tři varianty souboru `grub.conf` pro rozdělení disku použité v tomto návodu a pro jádro `kernel-2.4.22`. Řádně okomentována je pouze první varianta.

- První `grub.conf` je pro ty kdo nepoužili skript **genkernel** pro sestavení jádra
- Druhý `grub.conf` je pro ty kdo pro sestavení jádra použili skript **genkernel** z LiveCD (tj. skript **genkernel** nainstalovaný přes GRP)
- Třetí varianta souboru `grub.conf` je pro ty kdo použili stažený skript **genkernel**

Obrázek 9.5. `grub.conf` pro ty kdo nepoužili skript **genkernel**

```
# Which listing to boot as default. 0 is the first, 1 the second etc.
default 0
# How many seconds to wait before the default listing is booted.
timeout 30
# Nice, fat splash-image to spice things up :)
splashimage=(hd0,0)/grub/splash.xpm.gz

title=Gentoo Linux 2.4.22
# Partition where the kernel image (or operating system) is located
root (hd0,0)
kernel (hd0,0)/kernel-2.4.22 root=/dev/hda3

# The next three lines are only if you dualboot with a Windows system.
# In this case, Windows is hosted on /dev/hda6.
title=Windows XP
root (hd0,5)
chainloader +1
```

Obrázek 9.6. `grub.conf` pro ty kdo použili skript **genkernel** z LiveCD (resp. z GRP)

```
default 0
timeout 30
splashimage=(hd0,0)/grub/splash.xpm.gz
```

```
title=Gentoo Linux 2.4.22
root (hd0,0)
kernel (hd0,0)/kernel-2.4.22 root=/dev/hda3
initrd (hd0,0)/initrd-2.4.22

# Only in case you want to dual-boot
title=Windows XP
root (hd0,5)
chainloader +1
```

Code listing 7: grub.conf for no

Obrázek 9.7. grub.conf pro ty kdo použili stažený skript genkernel

```
default 0
timeout 30
splashimage=(hd0,0)/grub/splash.xpm.gz

title=Gentoo Linux 2.4.22
root (hd0,0)
kernel (hd0,0)/kernel-2.4.22 root=/dev/ram0 init=/linuxrc real_root=/dev/hda3
initrd (hd0,0)/initrd-2.4.22

# Only in case you want to dual-boot
title=Windows XP
root (hd0,5)
chainloader +1
```

Poznámka

Pokud používáte jiné rozdělení disku a/nebo obraz jádra, upravte odpovídajícím způsobem tyto příklady. Nicméně se ujistěte že cokoliv následující za GRUB zařízením (jako je například (hd0,0)) je uvedeno relativně k místu připojení, nikoliv ke kořenovému adresáři. Jinými slovy, (hd0,0)/grub/splash.xpm.gz je ve skutečnosti /boot/grub/splash.xpm.gz protože (hd0,0) je /boot.

Pokud potřebujete předávat jakékoliv další volby jádru, jednoduše je přidejte na konec příkazu **kernel**. Jednu volbu už vlastně používáme (root=/dev/hda3 nebo real_root=/dev/hda3), ale můžete samozřejmě přidat ještě další. Jako příklad používáme již dříve diskutovaný příkaz vga pro framebuffer:

Obrázek 9.8. Doplnění volby vga do příkazu kernel

```
title=Gentoo Linux
```

```
root (hd0,0)
kernel (hd0,0)/kernel-2.4.22 root=/dev/hda3 vga=788
```

Uživatelé kteří použili skript **genkernel** by si měli uvědomit že jejich jádro používá stejné volby jako použili pro LiveCD. Například pokud máte SCSI zařízení, měli byste doplnit volbu `doscsi`.

Nyní uložte soubor `grub.conf` a ukončete program. Od této chvíle je GRUB plně nakonfigurován a můžete pokračovat v kapitole Instalace potřebných systémových nástrojů.

9.3. Volitelně: LILO

9.3.1. Instalace LILO

LILO, tj. LInux LOader, je odzkoušená volba a mezi Linuxovými zavaděči je to skutečně dříč. Avšak chybí mu některé vlastnosti které má GRUB (což je také důvod proč je v poslední době GRUB stále populárnější). Důvod proč je LILO stále ještě používán je že na některých systémech GRUB nefunguje a LILO ano. Samozřejmě, je používán také proto že někteří uživatelé znají LILO a nechtějí se ho vzdát. Ať už je to jak chce, Gentoo podporuje oboje a vy jste si zřejmě vybrali LILO.

Instalace LILO je lehounké, stačí jenom použít **emerge**:

Obrázek 9.9. Instalace zavaděče LILO

```
# emerge --usepkg lilo
```

9.3.2. Konfigurace LILO

Pro nakonfigurování LILO musíte vytvořit soubor `/etc/lilo.conf`. Spusťte svůj oblíbený editor (v této příručce je z důvodu konzistence používán **nano**) a vytvořte tento soubor.

Obrázek 9.10. Vytvoření `/etc/lilo.conf`

```
# nano -w /etc/lilo.conf
```

O několik kroků zpět jsme po vás chtěli abyste si zapamatovali jméno obrazu jádra který jste vytvořili. V následujícím příkladu konfiguračního souboru `lilo.conf` předpokládáme že toto jméno je `kernel-2.4.22`. V tomto příkladu také používáme vzorové rozdělení disku. V konfiguračním souboru `lilo.conf` jsou tři oddělené části:

- První část je pro ty kdo nepoužili skript **genkernel** pro sestavení jádra
- Druhá část je pro ty kdo pro sestavení jádra použili skript **genkernel** z LiveCD (tj. skript **genkernel** nainstalovaný přes GRP)
- Třetí část je pro ty kdo použili nejnovější (stažený) skript **genkernel**

Obrázek 9.11. Vzorový `/etc/lilo.conf`

```
boot=/dev/hda          # Install LILO in the MBR
prompt                 # Give the user the chance to select another section
delay=50               # Wait 5 (five) seconds before booting the default section
default=gentoo         # When the timeout has passed, boot the "gentoo" section
# Only if you use framebuffer. Otherwise remove the following line:
vga=788                # Framebuffer setting. Adjust to your own will

# For non-genkernel users
image=/boot/kernel-2.4.22
    label=gentoo        # Name we give to this section
    read-only          # Start with a read-only root. Do not alter!
    root=/dev/hda3      # Location of the root filesystem

# For GRP genkernel users
image=/boot/kernel-2.4.22
    label=gentoo
    read-only
    root=/dev/hda3
    initrd=/boot/initrd-2.4.22

# For recent genkernel users
image=/boot/kernel-2.4.22
    label=gentoo
    read-only
    root=/dev/ram0
    append="init=/linuxrc real_root=/dev/hda3"
    initrd=/boot/initrd-2.4.22

# The next two lines are only if you dualboot with a Windows system.
# In this case, Windows is hosted on /dev/hda6.
other=/dev/hda6
    label=windows
```

Poznámka

Pokud používáte odlišné rozdělní disku a/nebo jméno jádra tuto vzorovou konfiguraci odpovídajícím způsobem upravte.

Pokud potřebujete jádru předat jakékoliv další volby, použijte příkaz `append`. Jako příklad jsme použili

volbu `acpi=off` která vypne podporu ACPI:

Obrázek 9.12. Použití `append` pro přidání voleb jádra

```
image=/boot/kernel-2.4.22
label=gentoo
read-only
root=/dev/hda3
append="acpi=off"
```

Uživatelé kteří použili skript **genkernel** by si měli uvědomit že jejich jádro používá stejné volby jako použili pro LiveCD. Například pokud máte SCSI zařízení, měli byste doplnit volbu `doscsi`.

Pro dokončení instalace musíte spustit příkaz `/sbin/lilo` aby LILO mohlo aplikovat konfiguraci ze souboru `/etc/lilo.conf` na váš systém (tj. nainstalovat se na disk).

Obrázek 9.13. Ukončení instalace zavaděče LILO

```
# /sbin/lilo
```

Nyní můžete pokračovat v kapitole Instalace potřebných systémových nástrojů.

9.4. Volitelně: aBoot

Poznámka

Zavaděč aBoot lze používat pouze na systémech s architekturou Alpha!

Nejdříve ze všeho aboot nainstalujeme na náš systém a samozřejmě k tomu použijeme **emerge**:

Obrázek 9.14. Nainstalování zavaděče aboot

```
# emerge --usepkg aboot
```

V následujícím kroku označíme náš bootdisk jako (startovatelný) bootovatelný. To způsobí že se po startování počítače spustí aboot. K označení disku jako bootovatelný stačí zapsat aboot na začátek disku.

Obrázek 9.15. Označení startovacího disku jako bootovatelného

```
# swriteboot -f3 /dev/sda /boot/bootlx
# abootconf /dev/sda 2
```

Poznámka

Pokud používáte jiné rozdělení disku než jaké jsme používali v této kapitole, musíte odpovídajícím způsobem upravit použité příkazy. Přečtěte si proto příslušné manuálové stránky (**man 8 swriteboot** a **man 8 abootconf**).

Navíc můžete zjednodušit start Gentoo nastavením konfiguračního souboru pro aboot a proměnné SRM `boot_osflags`. Měli byste se také ujistit že proměnná `bootdef_dev` je správně nastavena (to lze jednodušeji provést z konzole než z Linuxu).

Obrázek 9.16. Vylepšení startu Gentoo

```
# echo '0:2/boot/vmlinux.gz root=/dev/sda2' > /etc/aboot.conf
# echo -n 0 > /proc/srm_environment/named_variables/boot_osflags
# echo -n '' > /proc/srm_environment/named_variables/boot_file
```

Pokud k instalaci používáte sériovou konzoli, nezapomeňte zahrnout "serial console boot flag" do souboru `aboot.conf`. Podívejte se do souboru `/etc/aboots.conf.example` ve kterém najdete další informace. Dále budete potřebovat umožnit přihlašování na sériovém portu:

Obrázek 9.17. Povolení přihlášení na sériovém portu

```
# echo tts/0 >> /etc/securetty
# echo 's0:12345:respawn:/sbin/agetty 9600 tts/0 vt100' >> /etc/inittab
```

Aboot je nyní nakonfigurován a připraven k použití. Nyní můžete pokračovat v kapitole Instalace potřebných systémových nástrojů.

9.5. Volitelně: MILO

Poznámka

MILO lze používat pouze na systémech s architekturou Alpha!

Než budeme pokračovat, měli byste se rozhodnout jak používat MILO. V tomto odstavci budeme předpokládat že chcete použít MILO startovací disketu. Pokud chcete startovat z MS-DOS diskového oddílu na svém pevném disku, měli byste odpovídajícím způsobem pozměnit použité příkazy.

Pro nainstalování zavaděče MILO použijte příkaz **emerge**.

Obrázek 9.18. Instalace zavaděče MILO

```
# emerge --usepkg milo
```

Po nainstalování zavaděče MILO by se měly MILO obrazy (images) objevit v `/opt/milo`. Příkazy uvedené níže vytvoří startovací disketu obsahující MILO. Nezapomeňte použít správné obrazy pro svůj Alpha systém.

Obrázek 9.19. Instalace zavaděče MILO na startovací disketu

```
(First insert a blank floppy)
# fdformat /dev/fd0
# mformat a:
# mcopy /opt/milo/milo-2.2-18-gentoo-ruffian a:\milo
# mcopy /opt/milo/linload.exe a:\linload.exe
(Only if you have a Ruffian:
  # mcopy /opt/milo/ldmilo.exe a:\ldmilo.exe
)
# echo -ne '\125\252' | dd of=/dev/fd0 bs=1 seek=510 count=2
```

Vaše MILO startovací disketě je nyní připravena k nastartování Gentoo Linuxu. Možná budete potřebovat nastavit proměnné prostředí ve svém ARCS Firmware tak aby nastartovaly MILO; jak to provést je vysvětleno v MILO-HOWTO [<http://tldp.org/HOWTO/MILO-HOWTO/>] s několika příklady na běžných systémech, a příklady příkazů používaných v interaktivním módu.

Nečtení MILO-HOWTO [???] je špatný nápad.

Nyní můžete pokračovat v kapitole Instalace potřebných systémových nástrojů.

9.6. Volitelně: SILO

Je čas nainstalovat a nakonfigurovat SILO, tj. Sparc Improved boot Loader.

Obrázek 9.20. Instalace zavaděče SILO

```
# emerge --usepkg silo
```

Otevřete svůj oblíbený editor (v této příručce je používán editor **nano**) a vytvořte soubor /

`etc/silo.conf`.

Obrázek 9.21. Vytvoření souboru `/etc/silo.conf`

```
# nano -w /etc/silo.conf
```

V následujícím vzorovém konfiguračním souboru `silo.conf` je použito naše vzorové schéma rozdělení disku a jako obraz jádra (`kernelimage`) je použit soubor `kernel-2.4.22`.

Obrázek 9.22. Příklad souboru `/etc/silo.conf`

```
partition = 1          # Boot partition
root = /dev/hda4       # Root partition
timeout = 15          # Wait 15 seconds before booting the default section

image = /boot/kernel-2.4.22
label = linux
```

Pokud použijete konfigurační soubor `silo.conf` získaný přes Portage, ujistěte se že jste zakomentovali všechny nepotřebné řádky.

Pokud máte oddělený diskový oddíl `/boot`, zkopírujte tento konfigurační soubor do adresáře `/boot` a spusťte příkaz `/sbin/silo`:

Obrázek 9.23. Pouze pokud máte `/boot` na samostatném diskovém oddíle

```
# cp /etc/silo.conf /boot
# /sbin/silo -C /boot/silo.conf
/boot/silo.conf appears to be valid
```

Pokud je váš adresář `/boot` umístěn na kořenovém diskovém oddílu, stačí pouze spustit `/sbin/silo`:

Obrázek 9.24. Spuštění `silo`

```
# /sbin/silo
/etc/silo.conf appears to be valid
```

Nyní můžete pokračovat v kapitole Instalace potřebných systémových nástrojů.

9.7. Volitelně: PALO

Poznámka

Zavaděč PALO je určen pouze pro systémy s architekturou HPPA!

Na platformě PA-RISC je používán zavaděč nazývaný palo. Jeho konfigurační soubor se nachází v `/etc/palo.conf`. Zde je ukázková konfigurace:

Obrázek 9.25. Vzorový soubor `/etc/palo.conf`

```
--commandline=2/vmlinux root=/dev/sdb2 HOME=/  
--recoverykernel=/vmlinux.old  
--init-partitioned=/dev/sdb
```

První řádek zavaděči palo oznamuje umístění jádra a jaké startovací parametry je potřeba použít. `2/vmlinux` znamená že jádro pojmenované `/vmlinux` se nachází na druhém diskovém oddíle. Pozor, cesta k jádru je uváděna vzhledem k diskovému oddílu nikoliv ke kořenovému adresáři.

Druhý řádek oznamuje jaké záložní jádro použít. Pokud je toto vaše první instalace a žádné záložní jádro nemáte, zakomentujte tento řádek. Třetí řádek definuje na kterém disku bude palo umístěno.

Až budete s konfigurací hotovi, stačí spustit **palo**.

Obrázek 9.26. Aplikace konfigurace PALO

```
# palo
```

Nyní můžete pokračovat v kapitole Instalace potřebných systémových nástrojů.

9.8. Volitelně: yaBoot

9.8.1. Úvod

Poznámka

Zavaděč yaBoot lze používat pouze na systémech s architekturou NewWorld PPC!

Existují dva způsoby jak nakonfigurovat váš yaBoot systém. Můžete použít nový a vylepšený **yaboot-config** dodávaný s `yaboot-1.3.8-r1` a novějšími pro automatické nastavení `yaboot`. Pokud z nějakého důvodu nechcete `yabootconfig` použít k automatickému vytvoření souboru `/etc/yaboot.conf`, můžete jedno-

duše upravit vzorový soubor nainstalovaný na vašem systému.

- Výchozí: Použití yabootconfig
- Alternativa: Ruční konfigurace yaBoot

9.8.2. Výchozí: Použití yabootconfig

Program **yabootconfig** automaticky nalezne diskové oddíly na vašem systému a nastaví dvojité a trojitě kombinace startu s Linuxem, Mac OS a Mac OS X.

Pro použití **yabootconfig** musí mít váš disk bootstrap diskový oddíl, a `/etc/fstab` musí být nakonfigurován s vašimi Linuxovými diskovými oddíly. Oba tyto kroky již byly v této příručce probírány. Před započítím se přesvědčete že máte nainstalovanu nejnovější verzi yaboot spuštěním příkazu **emerge --update yaboot**. To je nutné protože poslední verze je dostupná přes Portage, ale nemusí být zahrnuta do etapových balíčků.

Obrázek 9.27. Instalace yaboot

```
# emerge --usepkg --update yaboot
```

Nyní spusťte yabootconfig. Program poběží a bude žádat o potvrzení umístění bootstrap diskového oddílu. Zadejte **Y** pokud je hodnota správná. Pokud není, zkontrolujte obsah souboru `/etc/fstab`. Program yabootconfig poté prohledá nastavení vašeho systému, vytvoří soubor `/etc/yaboot.conf` a spustí za vás **mkofboot**. Program **mkofboot** je používán pro vytvoření bootstrap diskového oddílu a instalaci konfiguračního souboru yaboot do něj.

Možná budete chtít zkontrolovat obsah `/etc/yaboot.conf`. Pokud v souboru `/etc/yaboot.conf` uděláte nějakou změnu (jako je nastavení implicitního OS), spusťte **ybin -v** pro aplikaci změn na bootstrap diskový oddíl.

Nyní můžete pokračovat v kapitole Instalace potřebných systémových nástrojů.

9.8.3. Alternativa: Ruční konfigurace yaBoot

Nejdříve se ujistěte že máte na svém systému nainstalovanu poslední verzi balíčku yaboot:

Obrázek 9.28. Instalace yaboot

```
# emerge --usepkg --update yaboot
```

Níže najdete dokončený soubor `/etc/yaboot.conf`. Změňte ho dle vlastního uvážení:

Obrázek 9.29. /etc/yaboot.conf

```
## /etc/yaboot.conf
##
## run: "man yaboot.conf" for details. Do not make changes until you have!!
## see also: /usr/share/doc/yaboot/examples for example configurations.
##
## For a dual-boot menu, add one or more of:
## bsd=/dev/hdaX, macos=/dev/hdaY, macosx=/dev/hdaZ

## our bootstrap partition:

boot=/dev/hda9

##hd: is open firmware speak for hda
device=hd:
partition=11

root=/dev/hda11
delay=5
defaultos=macosx
timeout=30
install=/usr/lib/yaboot/yaboot
magicboot=/usr/lib/yaboot/ofboot

#####
## This section can be duplicated if you have more than one kernel or set of
## boot options
#####
image=/boot/vmlinuz
    label=Linux
    sysmap=/boot/System.map
    read-only
#####

macos=/dev/hda13
macosx=/dev/hda12
enablecdboot
enableofboot
```

Jakmile je yaboot nastaven jak potřebujete, spusťte **mkofboot -v** což aplikuje nastavení na bootstrap diskový oddíl. Nezapomeňte na tento krok! Pokud všechno proběhlo v pořádku, a pokud jste použili stejné nastavení jako v předchozí ukázce, po restartu se před vámi objeví jednoduché menu. Pokud později upravíte nastavení yaboot, bude pro aktualizaci bootstrap diskového oddílu stačit použít příkaz **ybin -v**, protože mkofboot se používá pouze při první instalaci.

Více informací o yabootu najdete na stránkách yaboot projektu [<http://penguinppc.org/projects/yaboot>]. Nyní můžete pokračovat v kapitole Instalace potřebných systémových nástrojů.

9.9. Volitelně: BootX

Poznámka

Zavaděč BootX lze používat pouze na systémech s architekturou OldWorld PPC!

Pokud chcete používat zavaděč BootX, potom nemusíte v této etapě dělat vůbec nic. Nicméně po restartu budete muset trochu nakonfigurovat BootX kontrolní panel (v MacOS). Budeme se tím zabývat později během instalace.

Nyní můžete pokračovat v kapitole Instalace potřebných systémových nástrojů.

9.10. Volitelně: MIPS ROM

MIPS nevyžaduje použití zavaděče. MIPS ROM se stará o celý proces startování, ale musíte nainstalovat jádro jako životaschopnou volbu.

Nejdříve nainstalujte nástroj dvhtool, potřebný ke zkopírování jádra do diskového oddílu SGI Volume Header.

Obrázek 9.30. Instalace nástroje dvhtool

```
# emerge dvhtool
```

Nyní zkopírujte jádro do diskového oddílu SGI Volume Header. Program **dvhtool** implicitně předpokládá že SGI Volume Header je na `/dev/sda`. Pokud to není váš případ (například pokud je IRIX nainstalován na `/dev/sda` a Gentoo/MIPS na `/dev/sdb`) potom o tom musíte dvhtool informovat pomocí přepínače `-d <zařízení>`.

Obrázek 9.31. Kopírování jádra do diskového oddílu SGI Volume Header

```
# dvhtool --unix-to-vh <kernel name in /boot> <name you want to give in volume header>
```

Pokud chcete vypsat všechny dostupné volby na SGI Volume Header, použijte přepínač `-print-volume-directory`:

Obrázek 9.32. Zobrazení všech položek v SGI Volume Header

```
# dvhtool --print-volume-directory
```

Pro nastavení vašeho systému tak aby naskočil Gentoo/MIPS si musíte pohrát s nastavením v MIPS ROM. Tato nastavení jsou popsána dále v této příručce v části věnující se krokům po dokončení instalace Gentoo.

Nyní můžete pokračovat v kapitole Instalace potřebných systémových nástrojů.

Kapitola 10. Instalace potřebných systémových nástrojů

Abstrakt

Jak již bylo uvedeno, Gentoo je o možnosti volby. V této kapitole vám pomůžeme zvolit a nainstalovat některé důležité nástroje.

10.1. Systémový logger

Když jsme se podrobněji zmínili o etapovém balíčku stage3, uvedli jsme že obsahuje všechny potřebné systémové nástroje pro které neexistuje alternativa. Řekli jsme také že později nainstalujeme všechny potřebné systémové nástroje. A ta chvíle právě nadešla :)

První nástroj u kterého se musíte rozhodnout bude poskytovat logovací zázemí pro váš systém. Unix a Linux mají skvělou historii logovacích nástrojů - pokud chcete můžete všechno co se odehraje na vašem systému zaznamenat do logovacích souborů (logfiles). A to se děje přes systémový logger.

Gentoo vám nabízí možnost výběru z několika systémových loggerů. Existuje `sysklogd`, což je tradiční balíček systémových logovacích démonů, `msyslog`, flexibilní systémový logger s modularizovaným designem, `syslog-ng`, vylepšený systémový logger, a `metalog` což je skvěle konfigurovatelný systémový logger.

Pokud si jeden z nich nedokážete vybrat, použijte `syslog-ng` protože to je mocný nástroj a navíc má skvělou výchozí konfiguraci.

K nainstalování zvoleného systémového loggeru použijte **emerge** a přidejte ho do výchozího runlevelu příkazem **rc-update**. Následující příklad instaluje `syslog-ng`. Pokud chcete, můžete ho samozřejmě nahradit svým vybraným systémovým loggerem:

Obrázek 10.1. Instalace systémového loggeru

```
# emerge --usepkg syslog-ng
# rc-update add syslog-ng default
```

10.2. Volitelně: Cronový démon

Další je cronový démon. Ačkoliv se jedná o volitelný a nepovinný systémový nástroj, je moudré ho nainstalovat. Ale co to vlastně cronový démon je? Cronový démon spouští naplánované příkazy. Je velice praktické pokud můžete některé příkazy spouštět pravidelně (například denně, týdně nebo měsíčně).

Gentoo poskytuje tři verze cronových démonů: `dcron`, `fcron` a `vixie-cron`. Instalace kteréhokoliv

z nich je podobná instalaci systémového loggeru. Avšak `dcrn` a `fcron` vyžadují další konfigurační příkaz, konkrétně `crontab /etc/crontab`. Pokud nevíte který zvolit, použijte `vixie-cron`.

Poznámka

Pokud instalujete Gentoo pomocí předkompilovaných balíčků, zkuste použít `vcron` namísto `vixie-cron` pokud `vixie-cron` nefunguje.

Obrázek 10.2. Instalace cronového démona

```
# emerge --usepkg vixie-cron
# rc-update add vixie-cron default
(Only if you have chosen dcrn or fcron:) # crontab /etc/crontab
```

10.3. Nástroje souborového systému

Dle toho který souborový systém používáte musíte pro tento souborový systém nainstalovat potřebné nástroje (pro kontrolu jeho integrity, vytváření dalších souborových systémů, atd.).

Následující tabulka obsahuje nástroje které potřebujete nainstalovat pro jednotlivé souborové systémy:

Souborový systém	Nástroj	Instalační příkaz
XFS	xfspg	<code>emerge --usepkg xfspg</code>
ReiserFS	reiserfspg	<code>emerge --usepkg reiserfspg</code>
JFS	jfsutils	<code>emerge --usepkg jfsutils</code>

Pokud pro připojení k Internetu nepotřebujete `rp-pppoe`, pokračujte v části Dokončení vaší instalace Gentoo. V opačném případě pokračujte v části Nástroje pro síť.

10.4. Volitelně: Nástroje pro síť

Pokud k připojení k Internetu potřebujete `rp-pppoe`, musíte si ho nainstalovat. Uživatelé GRP (předkompilované balíčky) musí použít následující příkaz:

Obrázek 10.3. Instalace rp-pppoe s použitím GRP

```
(Only for GRP users:)
# emerge --usepkgonly rp-pppoe
```

Ostatní uživatelé musí `rp-pppoe` nainstalovat následujícím příkazem:

Obrázek 10.4. Instalace rp-pppoe pro běžné uživatele

```
# USE="-X" emerge rp-pppoe
```

Volba USE="-X" zakáže instalaci XFree jako závislosti (**rp-pppoe** má i grafické nástroje, pokud je chcete povolit musíte si později **rp-pppoe** později překompilovat nebo nyní nainstalovat XFree - kompilace ale zabere dlouhou dobu).

Nyní pokračujte v části Dokončení vaší instalace Gentoo.

Kapitola 11. Dokončení vaší instalace Gentoo

Abstrakt

Už jsme skoro hotovi. Nyní už stačí vytvořit jednoho (nebo více) uživatelů pro váš systém a restartovat instalaci do zbrusu nového Gentoo systému.

11.1. Správa uživatelů

11.1.1. Nastavení hesla uživatele root

Než na to zapomenete, nastavte heslo uživatele root:

Obrázek 11.1. Nastavení hesla uživatele root

```
# passwd
```

Pokud chcete aby se root mohl přihlásit i přes sériovou konzoli, přidejte `ttyS0` do souboru `/etc/securetty`.

Obrázek 11.2. Přidání `ttyS0` do `/etc/securetty`

```
# echo "ttyS0" >> /etc/securetty
```

11.1.2. Přidání uživatele pro běžné používání

Pracovat jako root na Unixovém/Linuxovém systému je *nebezpečné* a měli byste se tomu vyhýbat jak to jenom jde. Tudíž vám silně doporučujeme přidat dalšího uživatele pro každodenní používání.

Například pro vytvoření uživatele se jménem *john* který je členem skupiny *wheel* (aby se mohl přepnout do role roota pomocí příkazu **su**), skupiny *users* (implicitní skupina pro všechny uživatele) a skupiny *audio* (aby mohl používat zvuková zařízení):

Obrázek 11.3. Přidání uživatele pro každodenní používání

```
# useradd john -m -G users,wheel,audio -s /bin/bash
```

```
# passwd john
Password: (Enter the password for john)
Re-enter password: (Re-enter the password to verify)
```

Pokud tento uživatel bude někdy potřebovat provést něco jako root, může použít příkaz **su** - tím dočasně získá práva uživatele root. Dalším možným řešením je balíček **sudo**, který je při správné konfiguraci velmi bezpečný.

11.2. Restartujte a kochejte se

11.2.1. Restart

Gratuluje! Váš Gentoo systém je nyní hotov. Spusťte příkaz **etc-update** kterým aktualizujete konfigurační soubory (pokud je to potřeba), ukončete chrootované prostředí a odpojte veškeré diskové oddíly a, pokud jste museli provést bind-mount na `/mnt/gentoo/dev`, nezapomeňte ho také odpojit. A teď už zadejte ten kouzelný příkaz na který celou dobu čekáte: **reboot**

Obrázek 11.4. Restart systému

```
# etc-update
# exit
# cd /
# umount /mnt/gentoo/boot /mnt/gentoo/proc /mnt/gentoo
# reboot
```

Samozřejmě nezapomeňte vyjmout startovací CD, jinak systém opět nastartuje z něj namísto z vaší nové instalace Gentoo Linuxu.

Uživatelé architektury OldWorld PPC nastartují svůj MacOS protože jejich zavaděč ještě není nainstalován. Tito uživatelé by si měli přečíst následující odstavec Volitelně: Konfigurace BootX. Uživatelé systémů s architekturou MIPS si při zprovoznění nové instalace Gentoo budou muset trochu pohrát s nastavením MIPS PROM. Tito uživatelé by si měli přečíst odstavec Volitelně: Zprovoznění Gentoo/MIPS. Uživatelé GRP mohou pokračovat v odstavci Volitelně: Instalace dalších balíčků, a ostatní mohou pokračovat v kapitole Kam se vydat nyní?

11.2.2. Volitelně: Konfigurace BootX

Poznámka

Tato část je určena pouze uživatelům architektury PPC kteří chtějí používat zavaděč BootX. Všichni ostatní by měli tento odstavec přeskočit.

Váš systém právě nastartoval do Mac OS, takže otevřete ovládací panel programu BootX. Zvolte Options a odškrtněte volbu Used specified RAM disk. Až se vrátíte do hlavního okna programu BootX, najdete zde nyní volbu pomocí které můžete nakonfigurovat kořenový disk a diskový oddíl systému. Zadejte příslušné hodnoty.

BootX lze nakonfigurovat i tak že nastartuje Linux hned po bootu. Pokud si to zvolíte, systém nejdříve začne startovat do MacOS, a během tohoto startu se spustí BootX a nastartuje se Linux. Další informace hledejte na domácí stránce programu BootX. [<http://penguinppc.org/projects/bootx/>]

Pokud používáte GRP můžete pokračovat v odstavci Volitelně: Instalace dalších balíčků, a jinak přeskočte až na kapitolu Kam se vydat nyní?

11.2.3. Volitelně: Zprovoznění Gentoo/MIPS

Po restartu se přepněte do System Maintenance Menu a zvolte položku Enter Command Monitor (5). Pokud chcete otestovat svou Instalaci Gentoo, můžete jednoduše zadat příkaz **root -f <jméno jádra>**. Aby váš systém vždy startoval do instalace Gentoo, musíte v MIPS PROM nastavit některé proměnné:

Obrázek 11.5. Konfigurace MIPS PROM pro start do Gentoo

```
1) Start System
2) Install System Software
3) Run Diagnostics
4) Recover System
5) Enter Command Monitor

Option? 5
Command Monitor. Type "exit" to return to the menu.

(<root device> = Gentoo's root partition, e.g. /dev/sda3)
>> setenv OSLoadPartition <root device>

(To list the available kernels, type "ls")
>> setenv OSLoader <kernel name>
>> setenv OSLoadFilename <kernel name>

(Declare the kernel parameters you want to pass)
>> setenv OSLoadOptions <kernel parameters>

(Provide the location of the Volume Header)
>> setenv SystemPartition scsi(0)disk(1)rdisk(0)partition(8)

(Automatically boot Gentoo)
>> setenv AutoLoad Yes

(Set the timezone)
>> setenv TimeZone EST5EDT

(Use the serial console - graphic adapter users should have "g" instead of "d1" (one))
```

```
>> setenv console dl
```

Nyní si můžete užívat Gentoo!

11.2.4. Volitelně: Instalace dalších balíčků

Poznámka

Tato část je určena výhradně uživatelům GRP. Zbývající uživatelé by ji měli přeskočit a pokračovat v kapitole Kam se vydat nyní?

Nyní když váš systém nastartoval, přihlaste se jako uživatel kterého jste vytvořili (například *john*) a použijte příkaz **su** - čímž získáte práva uživatele root:

Obrázek 11.6. Získání rootovských práv

```
$ su -  
Password: (Enter your root password)
```

Nyní potřebujeme přkopírovat předkompilované balíčky z druhého CD (CD-2) pokud ho máte. Nejdříve CD připojte:

Obrázek 11.7. Připojení CD-2

```
# mkdir /mnt/cdrom  
(Put CD-2 in the CD tray)  
# mount /mnt/cdrom
```

Nyní přkopírujte předkompilované balíčky z CD do `/usr/portage/packages`. Ujistěte se že používáte přesně ten samý příkaz **copy**!

Obrázek 11.8. Překopírování předkompilovaných balíčků

```
# cp /mnt/cdrom/packages/All/* /usr/portage/packages/All/
```

A nyní nainstalujte ty balíčky které chcete. CD-2 obsahuje několik předkompilovaných binárních balíčků, například KDE:

Obrázek 11.9. Instalace KDE

```
# emerge --usepkg kde
```

Ujistěte se že jste veškeré binární soubory nainstalovali nyní. Až použijete příkaz **emerge sync** k aktualizaci Portage (což se naučíte později), pravděpodobně dojde k tomu že předkompilované balíčky již nebudou odpovídat ebuildům v aktualizované databázi Portage. Můžete to zkusit obejít použitím **emerge -usepkgonly** namísto **emerge --usepkg**.

Gratulujeme, váš systém je nyní kompletní! Pokračujte v kapitole Kam se vydat nyní? abyste se dozvěděli více o Gentoo.

Kapitola 12. Kam se vydat nyní?

Abstrakt

Tak teď máte Gentoo systém, ale co dál?

12.1. Dokumentace

Gratulujeme! Nyní máte funkční Gentoo systém. Ale kam se nyní vydat? Jaké jsou vaše možnosti? Co prozkoumávat nejdříve? Gentoo poskytuje svým uživatelům mnoho možností, takže je zde mnoho dokumentovaných (a méně dokumentovaných) vlastností.

Rozhodně byste se měli podívat na druhou část Gentoo Handbook která nese název Práce s Gentoo, ve které vysvětlujeme jak udržovat svůj software aktuální, jak nainstalovat další programy, co jsou to USE příznaky, jak funguje inicializační systém Gentoo, atd.

Pokud se zajímáte o optimalizaci svého systému pro desktopové použití, nebo pokud se chcete naučit konfigurovat systém jako úplný desktopový systém, podívejte se do Desktop Configuration Guide [<http://www.gentoo.org/doc/en/desktop.xml>].

Pro úplný seznam dostupné dokumentace se podívejte na naši Documentation Resources page [<http://www.gentoo.org/doc/en/index.xml>].

12.2. Gentoo Online

Samozřejmě jste vždy vítáni v našich Gentoo diskusních fórech [<http://forums.gentoo.org/>] nebo na jednom z našich mnoha Gentoo IRC kanálů [<http://www.gentoo.org/main/en/irc.xml>].

Pro naše uživatele také existuje několik mailinglistů [<http://www.gentoo.org/main/en/lists.xml>]. Informace o tom jak se do nich přihlásit najdete na uvedené stránce.

A teď už zmlkneme a necháme vás vychutnat si instalaci :)

Část II. Práce s Gentoo

Naučte se pracovat s Gentoo: instalovat software, upravovat proměnné, měnit chování portage atd.

Obsah

13. USE příznaky	129
13.1. Co jsou USE příznaky?	129
13.1.1. Myšlenka na které stojí USE příznaky	129
13.1.2. Definování USE příznaků	129
13.1.3. Jaké USE příznaky existují?	129
13.2. Použití USE příznaků	130
13.2.1. Deklarace trvalých USE příznaků	130
13.2.2. Deklarace dočasných USE příznaků	131
13.2.3. Dědění USE příznaků	131
13.2.4. Priorita příznaků	131
13.3. USE příznaky specifické pro balíčky	132
13.3.1. Zobrazení dostupných USE příznaků	132
14. Portage a software	134
14.1. Získávání informací o balíčcích	134
14.1.1. Pán všech nástrojů: emerge	134
14.1.2. Portage Tree	134
14.1.3. Hledání balíčku	136
14.1.4. Zobrazení ChangeLogu	138
14.2. Aktualizace Portage	138
14.2.1. Úvod	138
14.2.2. Volba mirroru pro rsync	138
14.2.3. Aktualizace Portage	139
14.3. Údržba software	139
14.3.1. Kompilované nebo předkompilované	139
14.3.2. Instalace programů ze zdrojového kódu	140
14.3.3. Instalace předkompilovaných balíčků	141
14.3.4. Práce se závislostmi	142
14.3.5. Aktualizace systému	143
14.3.6. Odstraňování software	145
14.4. Dostupnost software	145
14.4.1. ARCH nebo ne?	146
14.4.2. Skryté balíčky	146
14.4.3. Blokované balíčky	147
15. Vlastnosti Portage	148
15.1. DistCC	148
15.1.1. Co je distCC?	148
15.1.2. Instalace DistCC	148
15.1.3. Aktivace	148
15.2. ccache	150
15.2.1. Co je to ccache?	150
15.2.2. Instalace ccache	150
15.2.3. Aktivace podpory v Portage	150
15.3. Binární balíčky	151
15.3.1. Vytváření binárních balíčků	151

15.4. Vlastnosti související s bezpečností	152
15.4.1. Sandbox	152
15.4.2. Uživatelská práva	152
15.4.3. Aktivace sandboxu a/nebo uživatelských práv	152
15.4.4. Striktní kontroly	152
15.4.5. Smart File System práva	153
15.5. Další vlastnosti	153
15.5.1. Portage náповěda	153
16. Kontrola chování Portage	154
16.1. Nastavení ochrany souborů	154
16.1.1. Ochrana?	154
16.1.2. Deklarace CONFIG_PROTECT	154
16.1.3. etc-update	155
16.2. Síťová nastavení	156
16.2.1. Mirrors	156
16.2.2. Mirrorselect	157
16.2.3. Stahování	158
16.2.4. Nastavení programu rsync	158
16.3. Umístění adresářů	159
16.3.1. Úvod	159
16.3.2. Portage Tree	159
16.3.3. Distfiles	160
16.3.4. Balíčky a RPM	160
16.3.5. Dočasné soubory Portage	161
16.3.6. Logování	161
16.4. Další nastavení Portage	161
16.4.1. Nice Value	161
16.4.2. SLOTované balíčky a automatické čištění	162
16.4.3. Kompilace souvisejících balíčků	162
16.4.4. Formátování výstupu	163
16.5. Zdroje	164
16.5.1. Manuálové stránky	164
16.5.2. Komentované příklady	164
17. Inicializační skripty	165
17.1. Runlevely	165
17.1.1. Start systému	165
17.1.2. Inicializační skripty	165
17.1.3. Jak to funguje	165
17.1.4. Co je to runlevel?	167
17.1.5. Práce s inicializačními skripty	167
17.2. Práce s rc-update	169
17.2.1. Co je rc-update?	169
17.2.2. Přidávání a odebírání služeb	169
17.3. Konfigurace služeb	170
17.3.1. Proč je potřeba další konfigurace?	170
17.3.2. Adresář /etc/conf.d	170
17.4. Psaní init skriptů	170
17.4.1. Musím?	170
17.4.2. Schéma	171

17.4.3. Závislosti	171
17.4.4. Určování pořadí	172
17.4.5. Standardní funkce	173
17.4.6. Přidávání vlastních voleb	173
17.4.7. Proměnné pro konfiguraci služby	174
18. Proměnné prostředí	175
18.1. Proměnné prostředí?	175
18.1.1. Co jsou zač	175
18.1.2. Důležité příklady	175
18.2. Definice globálních proměnných	176
18.2.1. Adresář /etc/env.d	176
18.2.2. Skript env-update	177
18.3. Definice lokálních proměnných	178
18.3.1. Dle sezení (session)	178

Kapitola 13. USE příznaky

Abstrakt

USE-příznaky jsou velice důležitá vlastnost Gentoo. V této kapitole se naučíte pracovat s USE-příznaky a pochopíte jak USE-příznaky působí ve vašem systému.

13.1. Co jsou USE příznaky?

13.1.1. Myšlenka na které stojí USE příznaky

Při instalaci Gentoo (nebo jakékoliv jiné distribuce, nebo i jiného operačního systému) činíte rozhodnutí v závislosti na prostředí se kterým pracujete. Nastavení pro server se bude lišit od nastavení pro pracovní stanici. Nastavení herního systému se bude lišit od nastavení 3D pracovní stanice.

To platí nejen pro to které balíčky vyberete pro instalaci, ale také toho jaké vlastnosti by tyto instalované balíčky měly podporovat. Když nepotřebujete OpenGL, proč byste měli instalovat balíčky s podporou pro OpenGL a zakompilovávat podporu OpenGL do části vašich balíčků. Pokud nechcete používat KDE, proč byste měli zdržovat kompilováním KDE podpory do balíčků které bez problémů fungují i bez ní?

Abychom uživatelům pomohli v rozhodování co nainstalovat/zapnout a co ne, chtěli jsme jim umožnit jednoduchý způsob definice prostředí. To nutí uživatele rozhodnout se co skutečně chce a zjednodušuje práci systému Portage, našeho systému pro správu balíčků.

13.1.2. Definování USE příznaků

Podívejme se na USE příznaky. Příznak je klíčové slovo které vyjadřuje podporu a informaci o závislosti pro určitý pojem. Pokud použijete některý USE příznak, Portage bude vědět že chcete podporu pro dané klíčové slovo. Samozřejmě tím také měníte informaci o závislostech balíčku.

Podívejme se na konkrétní příklad: klíčové slovo `kde`. Pokud toto slovo nemáte ve své proměnné `USE`, všechny balíčky které obsahují nepovinnou podporu KDE budou zkompileovány bez ní. Všechny balíčky které mají nepovinnou závislost na KDE budou nainstalovány bez instalování KDE knihoven. Pokud jste klíčové slovo `kde` použili, potom budou tyto balíčky nainstalovány s podporou KDE a KDE knihovny budou v důsledku závislosti nainstalovány.

13.1.3. Jaké USE příznaky existují?

Existují dva typy USE příznaků: globální a lokální USE příznaky.

- Globální USE příznaky jsou používány několika balíčky, v celém systému. Toto je nejobvyklejší způsob použití USE příznaků.
- Lokální USE příznaky jsou používány jediným balíčkem pro úpravu chování právě toho jednoho ba-

líčku.

Seznam dostupných USE příznaků lze najít na webu [<http://www.gentoo.org/dyn/use-index.xml>] nebo lokálně v souboru `/usr/portage/profiles/use.desc`. Krátký (absolutně neúplný) úryvek:

Obrázek 13.1. Krátký úryvek použitelných USE příznaků

```
gtk      - Adds support for x11-libs/gtk+ (The GIMP Toolkit)
gtk2     - Use gtk+-2.0.0 over gtk+-1.2 in cases where a program supports both.
gtkhtml  - Adds support for gnome-extra/gtkhtml
guile    - Adds support for dev-util/guile (interpreter for Scheme)
icc      - Use the Intel C++ Compiler if the package supports it
icc-pgo  - Enable PGO data generation or use when use icc.
imap     - Adds support for IMAP
```

13.2. Použití USE příznaků

13.2.1. Deklarace trvalých USE příznaků

V naději že už jsme vás přesvědčili o důležitosti USE příznaků vám nyní vysvětlíme jak USE příznaky používat.

Jak už bylo uvedeno, všechny USE příznaky jsou deklarovány v USE proměnné. Abychom uživatelům vyhledávání a volbu USE příznaků ulehčili, vytvořili jsme výchozí USE nastavení. Toto nastavení je několik USE příznaků o kterých si myslíme že jsou uživateli Gentoo obvykle používány. Toto výchozí nastavení je deklarováno v souboru `/etc/make.profile/make.defaults`, a my se na něj teď podíváme:

Obrázek 13.2. `/etc/make.profile/make.defaults` USE proměnná

```
USE="x86 oss apm arts avi berkdb crypt cups encode foomaticdb gdbm gif gpm gtk
    imlib jpeg kde gnome libg++ libwww mad mikmod motif mpeg ncurses nls
    oggvorbis opengl pam pdflib png python qt quicktime readline sdl slang
    spell ssl svga tcpd truetype X xml2 xmms xv zlib"
```

Jak se můžete přesvědčit, tato proměnná obsahuje poměrně velký počet klíčových slov. Pokud si chcete nastavení přizpůsobit, neměňte soubor `/etc/make.profile/make.defaults` protože při aktualizaci Portage budou tyto změny ztraceny!

Změny tohoto nastavení docílíte přidáním nebo odstraněním klíčových slov do USE proměnné. To uděláte globálně definicí proměnné USE v souboru `/etc/make.conf`. Do této proměnné přidáte další požadované USE příznaky, nebo odstraníte USE příznaky které nechcete. Odstranění docílíte přidáním

znaku – (minus) před klíčové slovo.

Například, abychom odstranili podporu pro KDE a QT ale abychom současně přidali podporu pro ldap, nadefinujeme v souboru `/etc/make.conf` proměnnou USE následujícím způsobem:

Obrázek 13.3. Příklad nastavení proměnné USE v souboru `/etc/make.conf`

```
USE="-kde -qt ldap"
```

13.2.2. Deklarace dočasných USE příznaků

Někdy budete potřebovat jednorázově nastavit určité USE příznaky. Namísto dvojité editace `/etc/make.conf` (jednou abyste nastavení USE příznaků provedli a jednou abyste nastavení vrátili do původní podoby) stačí pouze definovat USE proměnnou jako proměnnou prostředí.

Jako příklad dočasně vypneme podporu javy z nastavení USE příznaků během instalace mozilly.

Poznámka

Příkaz `emerge` bude podrobněji probírán v odstavci `Portage a software`.

Obrázek 13.4. Použití USE jako proměnné prostředí

```
# USE="-java" emerge mozilla
```

13.2.3. Dědění USE příznaků

Některé balíky nejen že přijímají USE příznaky, ale USE příznaky taktéž poskytují. Až takový balík nainstalujete, budou USE příznaky které poskytuje přidány do nastavení USE. Pro zobrazení seznamu balíčků poskytujících USE příznak se podívejte do souboru `/etc/make.profile/use.defaults`:

Obrázek 13.5. Úryvek z `/etc/make.profile/use.defaults`:

gnome	gnome-base/gnome
gtk	x11-libs/gtk+
qt	x11-libs/qt
kde	kde-base/kdebase
motif	x11-libs/openmotif

13.2.4. Priorita příznaků

Samozřejmě existuje jasná priorita ohledně nastavení USE příznaků. Asi nechcete použít **USE="-java"** a dočkat se překvapení že `java` je stejně deklarována někde jinde. Priorita v nastavení USE je (od nejnižší priority po nejvyšší):

- Implicitní nastavení USE v souboru `/etc/make.profile/make.defaults`
- Zděděné nastavení USE pokud je instalován balíček z `/etc/make.profile/use.defaults`
- Uživatelské nastavení USE v souboru `/etc/make.conf`
- Uživatelské nastavení USE v proměnné prostředí

Pokud se chcete podívat na konečné nastavení, tak jak se dostane k Portage, použijte příkaz **emerge info**. Ten vypíše všechny relevantní proměnné (včetně proměnné USE) s obsahem který použije Portage.

Obrázek 13.6. Spuštění příkazu `emerge info`

```
# emerge info
```

13.3. USE příznaky specifické pro balíčky

13.3.1. Zobrazení dostupných USE příznaků

V následující kapitole, která se věnuje Portage a software, vám vysvětlíme jak spravovat nainstalovaný software a jak pracovat s příkazem **emerge**. Nicméně v tuto chvíli vám ukážeme alespoň jak pomocí tohoto příkazu zobrazit které USE příznaky balíček může použít.

Vraťme se k příkladu s Mozillou: které USE příznaky tento balíček používá? Abychom to zjistili stačí použít přepínače `--pretend` (což skutečně nic neudělá) a `--verbose` (takže příkaz bude generovat podrobnější výstup):

Obrázek 13.7. Zobrazení používaných USE příznaků

```
# emerge --pretend --verbose mozilla
These are the packages that I would merge, in order:

Calculating dependencies ...done!
[ebuild N      ] net-www/mozilla-1.5-r1 +java +crypt -ipv6 -gtk2 +ssl +ldap
+gnome -debug +mozcalendar -mozaccess -mozxterm -moznoirc -moznomail
-moznocompose -moznoxf
```

Příkaz **emerge** není jediný dostupný nástroj pro tento účel. Ve skutečnosti existuje nástroj nazývaný **et-**

cat, obsažený v balíčku **gentoolkit**, určený právě k zobrazování informací o balíčcích. Takže nejdříve nainstalujeme **gentoolkit**:

Obrázek 13.8. Instalace balíčku **gentoolkit**

```
# emerge --usepkg gentoolkit
```

Nyní spusťte **etcat** s argumentem **uses** čímž zobrazíte USE příznaky pro určitý balík. Například pro balík **gnumeric**:

Obrázek 13.9. Použití příkazu **etcat** pro zobrazení používaných USE příznaků

```
# etcat uses gnumeric
[ Colour Code : set unset ]
[ Legend      : (U) Col 1 - Current USE flags          ]
[              : (I) Col 2 - Installed With USE flags ]

U I [ Found these USE variables in : app-office/gnumeric-1.2.0 ]
- - libgda : Adds GNU Data Access (CORBA wrapper) support for gnumeric
- - gnomedb : unknown
+ + python : Adds support/bindings for the Python language
+ + bonobo  : Adds support for gnome-base/bonobo (Gnome CORBA interfaces)
```

Kapitola 14. Portage a software

Abstrakt

Hlavním účelem Portage je údržba programů na vašem systému. V této části se kromě jiného naučíte získávat informace z balíčků, aktualizovat svou databázi balíčků a instalovat/odstraňovat/aktualizovat programy.

14.1. Získávání informací o balíčcích

14.1.1. Pán všech nástrojů: emerge

Hlavní nástroj z balíku Portage, který bude většina uživatelů používat nejčastěji je **emerge**. Používali jsme ho již během instalace Gentoo a v předchozí kapitole, ale jenom krátce jsme vysvětlili jak ho používat. V této kapitole se budeme tímto nástrojem zabývat pečlivě a naučíme vás jak používat emerge k uspokojení všech potřeb souvisejících s programovým vybavením.

Příkaz **emerge** je používán k instalaci, deinstalaci, dotazování a údržbě softwarových balíčků. Je to vlastně nadstavba nad programem **ebuild**; ti z vás kteří se chtějí stát Gentoo profíky se později naučí používat i tento příkaz. Pro tuto chvíli se zaměříme na příkaz **emerge** protože má schopnosti které **ebuild** postrádá (jako například řešení závislostí, prohledávání Portage Tree, atd.).

Protože **emerge** je pro uživatele Gentoo nejdůležitější nástroj, má vyčerpávající manuálovou stránku kterou si můžete přečíst po zadání příkazu **man emerge**. Příkazem **emerge --help** si můžete nechat zobrazit zabudovanou nápovědu.

Obrázek 14.1. Zobrazení nápovědy pro emerge

```
# man emerge
# emerge --help
```

14.1.2. Portage Tree

Než budeme pokračovat v popisu příkazu **emerge**, podívejme se na Portage Tree. Přesuňte se do adresáře `/usr/portage` a zobrazte si seznam dostupných adresářů. Pro zobrazení seznamu adresářů použijeme příkaz `ls --classify` protože adresáře zobrazí s `"/` na konci.

Obrázek 14.2. Zobrazení Portage Tree

```
# cd /usr/portage; ls --classify
app-admin/      dev-ml/         gnome-libs/     net-print/
app-arch/       dev-perl/       gnome-office/   net-wireless/
```

app-benchmarks/	dev-php/	header.txt	net-www/
app-cdr/	dev-python/	incoming/	net-zope/
app-crypt/	dev-ruby/	jython/	packages/
app-dicts/	dev-tcltk/	kde-apps/	profiles/
app-doc/	dev-tex/	kde-base/	releases/
app-editors/	dev-util/	kde-i18n/	scripts/
app-emacs/	distfiles/	kde-libs/	sec-policy/
app-emulation/	eclass/	licenses/	skel.ChangeLog
app-games/	experimental/	media-fonts/	skel.ebuild
app-gnustep/	files/	media-gfx/	skel.metadata.xml
app-i18n/	fresco-base/	media-libs/	snapshots/
app-misc/	games-action/	media-plugins/	sys-apps/
app-office/	games-arcade/	media-radio/	sys-build/
app-pda/	games-board/	media-sound/	sys-cluster/
app-portage/	games-emulation/	media-tv/	sys-devel/
app-sci/	games-engines/	media-video/	sys-fs/
app-shells/	games-fps/	metadata/	sys-kernel/
app-text/	games-kids/	net-analyzer/	sys-kmods/
app-vim/	games-misc/	net-apache/	sys-libs/
app-xemacs/	games-mud/	net-dialup/	unix2tcp/
berlin-base/	games-puzzle/	net-dns/	x11-base/
dev-ada/	games-roguelike/	net-firewall/	x11-libs/
dev-cpp/	games-rpg/	net-fs/	x11-misc/
dev-db/	games-server/	net-ftp/	x11-plugins/
dev-dotnet/	games-simulation/	net-im/	x11-terms/
dev-embedded/	games-sports/	net-irc/	x11-themes/
dev-games/	games-strategy/	net-libs/	x11-wm/
dev-haskell/	games-util/	net-mail/	xfce-base/
dev-java/	glep/	net-misc/	xfce-extra/
dev-lang/	gnome-apps/	net-nds/	
dev-libs/	gnome-base/	net-news/	
dev-lisp/	gnome-extra/	net-p2p/	

Jak vidíte, Portage Tree má rozličné podadresáře. Většina z nich jsou vlastně *kategorie* ve kterých jsou uloženy Gentoo balíčky, nazývané *ebuildy* (ebuilds). Podívejme se například na app-office:

Obrázek 14.3. Zobrazení kategorie

```
# cd app-office; ls --classify
abiword/      gnotime/      kmymoney2/    oodi/          plan/          timestamp.x
dia/          gnucash/      koffice/      oooqs/         qhacc/
dia2code/     gnumeric/     lxbank/       openoffice/    sc/
factualux/    ical/         lyx/          openoffice-bin/ scribus/
gaby/         kbudget/      mdbtools/     openoffice-ximian/ siag/
gnofin/       khacc/        mrproject/    phprojekt/     texmacs/
```

V kategorii najdete balíčky které do ní náleží, přičemž pro každý balíček je vyhrazen samostatný adre-

sář. Podívejme se na balíček `openoffice`:

Obrázek 14.4. Zobrazení balíčku

```
# cd openoffice; ls --classify
ChangeLog  files/          openoffice-1.0.3-r1.ebuild  openoffice-1.1.0-r2.ebuild
Manifest   metadata.xml    openoffice-1.1.0-r1.ebuild  openoffice-1.1.0.ebuild
```

Vzpomeňte si že jsme vám říkali že Gentoo balíčky se nazývají `ebuild`? Nuže, v ukázkovém adresáři jsou uloženy čtyři takové `ebuildy`. Jejich jména jsou téměř stejná: liší se pouze ve jménu verze. Obsah takového balíčku si klidně můžete zobrazit: jsou to pouze obyčejné skripty. Tím se v tuto chvíli nebudeme zabývat protože to pro vás není důležité pokud plánujete Gentoo jen používat.

Další soubory jsou `ChangeLog` (který obsahuje seznam všech změn provedených v `ebuild`ech), `Manifest` (obsahující kontrolní součty a velikosti všech souborů v adresáři) a `metadata.xml` (obsahující podrobnější informace o balíčku, jako je například skupina zodpovědná za vývoj -- nazývaná `herd` -- a další podrobný popis).

V adresáři `files` najdete další soubory, potřebné programem Portage: `digests` (kontrolní součty a přístupová práva souborů potřebná jednotlivými verzemi balíčků), `patches`, vzorové konfigurační soubory, atd.

Obrázek 14.5. Zobrazení dalších souborů

```
# cd files; ls --classify
1.0.3/  digest-openoffice-1.0.3-r1  digest-openoffice-1.1.0-r1
1.1.0/  digest-openoffice-1.1.0      digest-openoffice-1.1.0-r2
# cd 1.1.0; ls --classify
fixed-gcc.patch      ooffice-wrapper-1.3
newstlportfix.patch  openoffice-1.1.0-linux-2.6-fix.patch
no-mozab.patch       openoffice-1.1.0-sparc64-fix.patch
nptl.patch
```

Pokud se přesunete zpět do kořenového adresáře Portage Tree (`/usr/portage`) zjistíte že existují také další adresáře, které nejsou kategoriemi. O těch se budeme bavit dále v této kapitole.

14.1.3. Hledání balíčku

Pokud jste v Gentoo Linuxu noví, pravděpodobně nevíte jaký nástroj potřebujete na jakou úlohu. Ke zjednodušení hledání vám **emerge** poskytuje způsob prohledávání všech balíčků dostupných na vašem systému. Existují dva způsoby jak hledat balíčky: *dle jména* a *dle jména a popisu*.

Pro prohledávání Portage Tree dle jména, použijte příkaz **emerge search**. Například, abyste našli informace o **mozilla**:

Obrázek 14.6. Zobrazení informací o mozille

```
# emerge search mozilla
Searching...
[ Results for search key : mozilla ]
[ Applications found : 5 ]
(Some output removed to improve readability)
* net-www/mozilla
    Latest version available: 1.5-r1
    Latest version installed: 1.4-r3
    Size of downloaded files: 29,153 kB
    Homepage:      http://www.mozilla.org
    Description: The Mozilla Web Browser

* net-www/mozilla-firebird
    Latest version available: 0.7
    Latest version installed: [ Not Installed ]
    Size of downloaded files: 37,850 kB
    Homepage:      http://www.mozilla.org/projects/firebird/
    Description: The Mozilla Firebird Web Browser

(...)
```

Pokud chcete aby se vyhledávalo i v popisech, použijte přepínač `--searchdesc`:

Obrázek 14.7. Vyhledávání i v popisu

```
# emerge --searchdesc mozilla
Searching...
[ Results for search key : mozilla ]
[ Applications found : 10 ]
(Some output removed to improve readability)
* dev-libs/nss-3.8
    Latest version available: 3.8
    Latest version installed: 3.8
    Size of downloaded files: 2,782 kB
    Homepage:      http://www.mozilla.org/projects/security/pki/nss/
    Description: Mozilla's Netscape Security Services Library that implements PKI support
```

Jak vidíte, výstup příkazu **emerge** vás informuje o kategorii a jméně balíčku, dostupné verzi, aktuálně nainstalované verzi, velikosti stažených balíčků, domovské stránce a krátkém popisu.

Všimli jste si něčeho nového? Ano, *stažené soubory* (downloaded files). Až požádáte Portage o nainstalování tohoto balíčku, budou samozřejmě potřeba příslušné zdrojové kódy (nebo předkompilované balíčky). Proto zkontroluje obsah adresáře `/usr/portage/distfiles` (pro zdrojový kód) nebo `/usr/portage/packages/All` (pro předkompilované balíčky) aby zjistil zda jsou potřebné soubor-

ry k dispozici. Pokud nejsou, stáhne potřebné soubory a umístí je do příslušných adresářů.

14.1.4. Zobrazení ChangeLogu

Při procházení Portage Tree jste viděli že u každého balíčku je veden ChangeLog. Položky ChangeLogu náležejícím verzím od nainstalované až po nejnovější si také můžete zobrazit pomocí **emerge**. Použijte přepínače `--pretend` a `--changelog` (zkráceně `-pl`). Jako příklad se podíváme na ChangeLog pro **gnumeric**:

Obrázek 14.8. Zobrazení položek ChangeLogu pro gnumeric

```
# emerge --pretend --changelog gnumeric
(Some output removed to improve readability)
*gnumeric-1.2.2

 27 Nov 2003; foser <foser@gentoo.org> gnumeric-1.2.2.ebuild :
New release, requested in #34492
updated deps

 12 Nov 2003; Jason Wever <weeve@gentoo.org> gnumeric-1.2.0.ebuild:
Marked stable on sparc, fixes bug #32405.

 14 Oct 2003; Jason Wever <weeve@gentoo.org> gnumeric-1.0.8.ebuild:
Added ~sparc keyword. Fixes bug #31150.
```

14.2. Aktualizace Portage

14.2.1. Úvod

Prohledávání Portage je pěkné, ale pokud pravidelně neaktualizujete Portage Tree, zaseknete se na balíčcích a verzích dostupných na vašem systému. To znamená že váš systém brzy zastará a že vám budou chybět bugfixy a opravy možných bezpečnostních problémů.

Existuje několik způsobů jak aktualizovat Portage Tree. Nejpopulárnější metodou je použití jednoho z našich rsync mirrorů [<http://www.gentoo.org/main/en/mirrors.xml>]. Další spočívá v použití obrazu (snapshot) Portage (v případě že vám například firewall nebo síťová omezení brání v použití rsync serveru).

14.2.2. Volba mirroru pro rsync

Je rozumné zvolit si rychlý mirror blízko vás. To můžete udělat ručně (nastavením proměnné `SYNC` v souboru `/etc/make.conf`) nebo použít nástroj **mirrorselect** který to za vás provede automaticky. Protože proměnné `SYNC` bude probírána později, podíváme se nyní na příkaz **mirrorselect**. Nejdříve pomocí **emerge** nainstalujeme **mirrorselect**:

Obrázek 14.9. Instalace nástroje mirrorselect

```
# emerge --usepkg mirrorselect
```

Nyní spusťte mirrorselect k automatické volbě mirrorů blízko vás (dojde také k nastavení Portage tak aby používal mirror pro zdrojové kódy):

Obrázek 14.10. Spuštění nástroje mirrorselect

```
# mirrorselect -a -s3
```

14.2.3. Aktualizace Portage

K aktualizace Portage za použití **rsync** jednoduše stačí spustit **emerge rsync**:

Obrázek 14.11. Aktualizace Portage Tree pomocí emerge rsync

```
# emerge sync
```

Pokud toto selže (ať již kvůli problémům se sítí nebo například firewallu), můžete zkusit použít příkaz **emerge-webrsync** který stahuje obrazy Portage Tree pomocí nástroje **wget**. To také znamená že můžete použít proxy servery pokud chcete. Jak nakonfigurovat systém aby používal proxy servery jsme rozebírali během instalace Gentoo.

Obrázek 14.12. Aktualizace Portage Tree pomocí emerge-webrsync

```
# emerge-webrsync
```

14.3. Údržba software

14.3.1. Kompilované nebo předkompilované

Gentoo poskytuje ebuildy, Gentoo balíčky chcete-li. Ale když chcete instalovat takový ebuild, můžete si zvolit mezi kompilací balíčku a využitím předkompilované verze. Ale jaké jsou výhody/nevýhody obou přístupů a mohou být používány současně?

Jak asi dokážete odhadnout, kompilace balíčků zabere hodně času (zejména pokud máte malé zdroje nebo pokud chcete kompilovat velké balíčky jako jsou KDE, OpenOffice.org, atd.). Pokud balíček kompilujete, můžete použít nastavení USE k přizpůsobení vašemu systému. Samozřejmě při kompilaci také můžete nadefinovat silnou optimalizaci (v proměnných CFLAGS a CXXFLAGS).

Použití předkompilovaných balíčků zmenšuje časové nároky (protože kompilace již není potřeba), ale ztratíte výhody nastavení USE příznaků a proměnných CFLAGS a CXXFLAGS.

Jak již bylo řečeno, předkompilované balíčky jsou ukládány v adresáři `/usr/portage/packages/All`, zatímco zdrojové kódy jsou ukládány v adresáři `/usr/portage/distfiles`. Pokud jste dokončili instalaci balíčku, můžete odstranit zdrojové kódy nebo balíček z příslušného adresáře. Avšak možná si budete chtít zdrojové kódy nejnovější verze ponechat, jenom pro případ že byste chtěli balíček přeinstalovat (takže ho nebudete muset znovu stahovat).

14.3.2. Instalace programů ze zdrojového kódu

OK, dost řečí, podívejme se na to. Pro instalaci programů budete používat příkaz **emerge**. Pokud nechcete používat předkompilované balíčky, můžete používat **emerge <jméno-balíčku>** nebo **emerge <kategorie>/<jméno-balíčku>**. Jako příklad nainstalujeme program **gnumeric**:

Obrázek 14.13. Kompilace gnumeric

```
# emerge gnumeric
```

Tento příkaz stáhne zdrojový kód, rozbalí ho, zkompiluje a nainstaluje balíček na váš systém. To samé provede také pro všechny závislosti. Pokud chcete zjistit jaké závislosti budou s balíčkem nainstalovány, použijte přepínač `--pretend` (zkráceně `-p`):

Obrázek 14.14. Předstírání kompilace gnumeric

```
# emerge --pretend gnumeric
```

Pokud chcete stáhnout zdrojové kódy balíčku a všech závislostí, ale nechcete balíček kompilovat, použijte přepínač `--fetchonly` (zkráceně `-f`):

Obrázek 14.15. Stažení zdrojových kódů gnumeric

```
# emerge --fetchonly gnumeric
```

Pokud chcete vědět odkud **emerge** stahuje zdrojové kódy, zkombinujte přepínače `--fetchonly` a `-`

-pretend:

Obrázek 14.16. Zobrazení URL pro zdrojové kódy gnumeric

```
# emerge --fetchonly --pretend gnumeric
```

Můžete také zvolit instalaci konkrétní verze balíčku. Například, pokud chcete instalovat program **gnumeric** verze nejvýše 1.2 -- ať už z jakéhokoli důvodu :) můžete zadat:

Obrázek 14.17. Instalace konkrétní verze gnumeric

```
# emerge "<gnumeric-1.2"
```

Další možnosti jsou samozřejmě ">" (novější verze) a "=" (přesná verze).

14.3.3. Instalace předkompilovaných balíčků

Když chcete nainstalovat předkompilovaný balíček, měli byste použít přepínač `--usepkg` (zkráceně `-k`). Tento příkaz použije binární balíček z adresáře `/usr/portage/packages/All` pokud se balíček a verze aplikace kterou chcete nainstalovat shodují.

Obrázek 14.18. Instalace předkompilovaného balíčku gnumeric

```
# emerge --usepkg gnumeric
```

Pokud chcete nainstalovat binární balíček i pokud si verze neodpovídají, použijte přepínač `--usepkgonly` (zkráceně `-K`).

Obrázek 14.19. Instalace předkompilovaného balíčku gnumeric

```
# emerge --usepkgonly gnumeric
```

Pokud na svém systému předkompilovaný balíček ještě nemáte, můžete nechat **emerge** ať ho stáhne z mirroru, definovaného v proměnné `PORTAGE_BINHOST` deklarované v `/etc/make.conf`.

Ke stažení binárního balíčku který na vašem systému ještě vůbec neexistuje použijte přepínač `--getbinpkg` (zkráceně `-g`):

Obrázek 14.20. Stažení a instalace předkompilovaného balíčku gnumeric

```
# emerge --getbinpkg gnumeric
```

Tento příkaz stáhne balíček a související informace na váš systém, společně se závislostmi. Pokud chcete zjistit které závislosti budou nainstalovány, použijte přepínač `--pretend` (zkráceně `-p`):

Obrázek 14.21. Předstírané stažení a instalace předkompilovaného balíčku gnumeric

```
# emerge --getbinpkg --pretend gnumeric
```

Můžete si také zvolit stáhnout předkompilovaný balíček (a souvisejících informací) bez kontroly informací oproti vašemu lokálnímu systému a pokud zatím tento předkompilovaný balíček na vašem systému nepoužíváte, použijte přepínač `--getbinpkgonly` (zkráceně `-G`):

Obrázek 14.22. Instalace předkompilovaného balíčku gnumeric bez použití lokální informace

```
# emerge --getbinpkgonly gnumeric
```

Můžete si také zvolit specifickou verzi balíčku. Například, pokud chcete nainstalovat program **gnumeric** starší než 1.2 -- ať už z jakéhokoliv důvodu :) byste zadali:

Obrázek 14.23. Instalace konkrétní verze programu gnumeric

```
# emerge --usepkg "<gnumeric-1.2"
```

Další možnosti jsou samozřejmě `>` (novější verze) a `=` (přesná verze).

14.3.4. Práce se závislostmi

Portage má rozsáhlou podporu pro práci se závislostmi. Ačkoliv se tím obvykle vůbec nemusíte zabývat (protože závislosti jsou automaticky zpracovávány systémem Portage) někteří uživatelé by mohli chtít vědět jak **emerge** pracuje se závislostmi.

Například pokud chcete aby Portage předstíral že žádná ze závislostí balíčku není nainstalována, můžete použít přepínač **--emptytree** (zkráceně **-e**). To je užitečné s přepínačem **--pretend** ke zobrazení kompletního stromu závislostí libovolného balíčku. Bez přepínače **--pretend** emerge překompiluje veškeré vypsané balíčky. Nicméně `glibc` nebude z bezpečnostních důvodů mezi závislostmi uvedena.

Obrázek 14.24. Zobrazení všech závislostí programu gnumeric

```
# emerge --emptytree --pretend gnumeric
```

Dalším přepínačem je **--nodeps**, který požádá Portage aby se pokusil nainstalovat uvedený balíček bez kontroly závislostí. Je evidentní že toto může vést k chybám.

Obrázek 14.25. Instalace balíčku gnumeric bez ohledu na závislosti

```
# emerge --nodeps gnumeric
```

Opakem přepínače **--nodeps** je přepínač **--onlydeps**, který Portage říká aby nainstaloval všechny závislosti balíčku, ale nikoliv balíček samotný:

Obrázek 14.26. Instalace pouze závislostí programu gnumeric

```
# emerge --onlydeps gnumeric
```

14.3.5. Aktualizace systému

Portage zná dvě speciální značky pro označení množiny balíčků: `system` a `world`. První z nich jste viděli během instalace, pokud jste nepoužili etapový balíček `stage3`. Pro připomenutí: `system` je kolekce hlavních balíčků, potřebných pro funkci systému Gentoo.

Značka `world` označuje veškerý software nainstalovaný na vašem systému a informace o systému. Jinými slovy, pokaždé když nainstalujete balíček pomocí příkazu **emerge <jméno-balíčku>**, je `<jméno-balíčku>` zaregistrováno ve "world" souboru (`/var/cache/edb/world`). Závislosti nejsou částí "world" souboru, ale k tomu se dostaneme později.

Obrázek 14.27. Aktualizace systémových balíčků

```
# emerge --update system
```

Identický přístup lze použít pro "world" balíčky:

Obrázek 14.28. Aktualizace celého systému

```
# emerge --update world
```

A znovu, pokud chcete vidět které balíčky bude emerge aktualizovat, použijte přepínač `--pretend` společně s přepínačem `--update`:

Obrázek 14.29. Předstírání aktualizace celého systému

```
# emerge --pretend --update world
(Some output removed to improve readability)
[ebuild    U ] net-misc/wget-1.9-r1 [1.9]
[ebuild    UD] media-video/dvdauthor-0.5.0 [0.5.3]
[ebuild    U ] net-analyzer/ethereal-0.9.16 [0.9.14]
```

Hned vedle slova "ebuild" si všimnete písmene (nebo kombinace písmen) které vám podávají další informace o balíčku:

- B (blokù) Balíček nalevo blokuje emerge balíčku napravo.
- N (nový) Balíček je pro váš systém nový a bude nainstalován poprvé.
- R (reemerge) Balíček není nový, ale musí být nainstalován znovu.
- F (stáhnout) Balíček vyžaduje ruční stažení zdrojového kódu (například vzhledem k licenci).
- U (aktualizovat) Balíček na vašem systému již existuje a bude aktualizován.
- UD (downgrade) Balíček na vašem systému již existuje a bude nainstalována starší verze.
- U (slot warning) balíček nainstalovaný na vašem systému je označen jako balíček který nemůže ko-existovat s jinou verzí, ale vaše aktualizace ano. Bude nainstalována aktualizace a stará verze bude odstraněna.

V určitých případech může aktualizace znamenat downgrade (tj. přechod na starší verzi namísto přechodu na novější). Pokud nechcete aby se to stalo, použijte přepínač `--upgradeonly` (zkráceně `-U`):

Obrázek 14.30. Upgrade celého systému

```
# emerge --update --upgradeonly world
```

Samozřejmě, mluvíme zde o "system" a "world", ale stejné akce můžete provést pro jednotlivé softwarevé balíčky.

14.3.6. Odstraňování software

Pokud chcete ze svého systému odstranit software, můžete použít přepínač `unemerge` (zkráceně `-C`, tj. velké C):

Obrázek 14.31. Odinstalování balíčků

```
# emerge unmerge gnumeric
```

Pokud chcete odstranění pouze simulovat (ale neprovést jej), můžete opět použít přepínač `--pretend`:

Obrázek 14.32. Předstírání odinstalování balíčků

```
# emerge --pretend unmerge gnumeric
```

Varování

Portage neověřuje zda na balíčku závisí některý jiný balíček. Nebude vás také varovat pokud je balíček součástí systému, tj. pokud se jedná o klíčovou aplikaci potřebnou pro korektní fungování vašeho systému!

Jakmile `unemerge` začne pracovat, uvidíte dlouhý seznam jmen souborů náležících k vašemu zvolenému balíčku. Nalevo od některých z těchto jmen bude zobrazen přepínač (flag). Přepínače `!mtime`, `!empty` a `cfgpro` určují důvody proč tyto soubory nebyly odstraněny společně se zbytkem balíčku. Soubory bez těchto přepínačů byly ze systému odstraněny úspěšně. Uvedené tři přepínače označují následující důvody:

- `!mtime` : Zobrazený seznam byl od instalace změněn, pravděpodobně vámi nebo některým nástrojem.
- `!empty` : Zobrazený adresář není prázdný.
- `cfgpro` : Tento soubor je umístěn v chráněném adresáři a nebylo s ním manipulováno z důvodu bezpečnosti.

14.4. Dostupnost software

14.4.1. ARCH nebo ne?

Gentoo umísťuje svoje balíčky do dvou možných stavů, nazývaných `ARCH` a `~ARCH`. Neberte to doslova: stavy závisí na architektuře kterou používáte. Jinými slovy pro systémy s architekturou `x86` existují stavy `x86` a `~x86`, pro systémy s architekturou `ppc` existují stavy `ppc` a `~ppc`, atd.

Stav `~ARCH` znamená že balíček sice pracuje na této platformě, ale že nebyl otestován dostatečně důkladně k tomu aby mohl být zařazen do `ARCH`. Balíčky z `~ARCH` většinou přechází do `ARCH` po dostatečně dlouhé době bez chyby.

Váš systém bude implicitně používat balíčky z `ARCH`. Pokud chcete být na špici vývoje, nevadí vám že čas od času některý balíček spadne, víte jak se vypořádat s poškozeným systémem a pokud chcete posílat bugreporty na bugs.gentoo.org [<http://bugs.gentoo.org/>], potom můžete zvolit balíčky z `~ARCH`. Pro přechod k systému používajícímu `~ARCH`, upravte proměnnou `ACCEPT_KEYWORDS` v souboru `/etc/make.conf` tak aby obsahovala `~ARCH` (opět: pro `x86` systémy: `~x86` atd.).

Všimněte si že zdaleka není triviální (pokud ne dokonce nemožné) vrátit se z `ARCH` k `~ARCH`.

Pokud nyní chcete aktualizovat svůj systém, všimněte si že bude aktualizována spousta balíčků!

14.4.2. Skryté balíčky

Pokud chcete nainstalovat balíček, možná narazíte na následující zprávu:

Obrázek 14.33. Zpráva o skrytých balíčcích

```
Calculating dependencies
!!! all ebuilds that could satisfy <your package> have been masked.
```

Balíček může být skrytý ze dvou důvodů:

1. Balíček je `~ARCH` zatímco vy používáte `ARCH`.
2. Balíček je výslovně napevno skrytý.

Pokud je balíček skrytý z prvního důvodu, a pokud ho skutečně chcete nainstalovat (s vědomím že nejspíš existuje nějaký důvod proč není dostupný v `ARCH`), můžete dočasně povolit `~ARCH` balíčky:

Obrázek 14.34. Dočasné povolení `~ARCH` balíčků

```
# ACCEPT_KEYWORDS="~x86" emerge gnumeric
```

Balíček je výslovně skrytý, pokud je jmenován v souboru /

`usr/portage/profiles/package.mask`. Pokud se do tohoto souboru podíváte, najdete v něm také důvody proč je jako skrytý označen (obvykle se jedná o komentář). Pokud chcete i přes to balíček nainstalovat (navzdory všem varováním o "poškození systému", "poškození dalších balíčků", nebo "velmi potřebném otestování"), vytvořte soubor `/etc/portage/package.unmask` a uveďte v něm onen balíček (použijte stejný formát jako je použit v souboru `/usr/portage/profiles/package.mask`).

Neměňte soubor `/usr/portage/profiles/package.mask` protože při další aktualizaci Portage Tree o všechny změny přijdete. Pokud chcete napevno skrýt balíček, vytvořte soubor `/usr/portage/package.mask` a uveďte v něm onen balíček (opět ve stejném formátu jako v souboru `/usr/portage/profiles/package.mask`).

14.4.3. Blokové balíčky

Někdy se dostanete do situace kdy se na obrazovce objeví následující chybová hláška:

Obrázek 14.35. Blokování balíčku

```
[blocks B      ] gnome-base/bonobo-activation (from pkg gnome-base/libbonobo-2.4.0)
```

V předchozím příkladě balíček **bonobo-activation** blokuje instalaci balíčku **libbonobo**. Pro vyřešení tohoto problému stačí odstranit balíček **bonobo-activation** a pokračovat dále:

Obrázek 14.36. Řešení situace s blokem balíčku

```
# emerge unmerge bonobo-activation
```

Kapitola 15. Vlastnosti Portage

Abstrakt

Gentoo Portage bere v úvahu několik vlastností zlepšujících související činnosti, mezi jinými například datum sestavení balíčku (build time). Tato kapitola vysvětluje existující vlastnosti.

15.1. DistCC

15.1.1. Co je distCC?

Program **distcc** slouží k distribuci kompilace mezi několik, ne nutně identických, počítačů na síti. Klient **distcc** zasílá všechny potřebné informace dostupným **DistCC** serverům (na kterých běží **distccd**) takže mohou pro klienta kompilovat kousky zdrojového kódu. Důsledkem síťové kompilace je její rychlejší dokončení.

Více informací souvisejících s **distcc** (a informací o tom jak tento nástroj zprovoznit s Gentoo Linuxem) můžete nalézt v Gentoo Distcc Dokumentaci [<http://www.gentoo.org/doc/en/distcc.xml>].

15.1.2. Instalace DistCC

Nástroj DistCC je dodáván s grafickým nástrojem pro monitoring úloh které váš počítač odesílá ke zpracování. Pokud používáte Gnome potom do USE příznaků doplňte klíčové slovo `gnome`. Nicméně pokud Gnome nepoužíváte a přesto byste rádi tento monitorovací program používali, potom byste měli mezi USE příznaky měli doplnit `gtk`.

Instalace **distcc** je, stejně jako v případě všech ostatních nástrojů dostupných přes Gentoo Portage, extrémně jednoduchá:

Poznámka

Od této chvíle budeme v příkazech uvedených v Gentoo Handbook vynechávat přepínač `-usepkg`, protože už víte jak instalovat binární balíčky.

Obrázek 15.1. Instalace DistCC

```
# emerge distcc
```

15.1.3. Aktivace

Výborně, takže pokud instalace byla jednoduchá, měl by být jednoduchý i zbytek :) Čili rychle aktivujme v Portage podporu pro **distcc**.

Nejdříve ze všeho otevřete soubor `/etc/make.conf` a upravte proměnnou `FEATURES` tak aby obsahovala klíčové slovo **distcc**. Dále upravte proměnnou `MAKEOPTS` tak aby obsahovala `-jX` kde `X` je počet CPU na kterých **distccd** (včetně aktuálního systému) plus jedna:

Obrázek 15.2. Možné nastavení proměnné `MAKEOPTS` v `/etc/make.conf`

```
# Suppose you have 2 single-CPU distccd hosts excluding this host:
MAKEOPTS="-j4"
```

Nyní, stále ještě v souboru `/etc/make.conf`, odkomentujte řádek s proměnnou `PORTAGE_TMPDIR` a doplňte následující řádek na konec souboru:

Obrázek 15.3. Na konec souboru `/etc/make.conf` doplňte následující proměnnou specifickou pro `distcc`

```
# Don't forget to uncomment the PORTAGE_TMPDIR variable
DISTCC_DIR=${PORTAGE_TMPDIR}/portage/.distcc
```

Nyní spusťte `distcc-config` a zadejte seznam dostupných DistCC serverů. Například předpokládejme že dostupné DistCC servery jsou 192.168.1.102 (aktuální systém), 192.168.1.103 a 192.168.1.104 (dva vzdálené systémy):

Obrázek 15.4. Konfigurace `distcc` tak aby používal tři dostupné DistCC servery

```
# distcc-config --set-hosts "192.168.1.102 192.168.1.103 192.168.1.104"
```

Samozřejmě, nezapomeňte spustit také démona `distccd`:

Obrázek 15.5. Spuštění `distccd`

```
# /etc/init.d/distccd start
```

Gratuluje, váš systém bude nyní používat distribuovanou kompilaci! Pro podrobnější informace o DistCC a Gentoo se podívejte do naší Gentoo Distcc Dokumentace [<http://www.gentoo.org/doc/en/distcc.xml>].

15.2. ccache

15.2.1. Co je to ccache?

Nástroj **ccache** je vlastně rychlá cache pro překladač. Při překladu programu budou mezivýsledky cachovány, takže při další rekompilaci stejného programu bude potřeba daleko kratší čas. V obvyklém případě může dojít k 5 až 10 násobnému zkrácení doby kompilace.

Pokud vás zajímají pro a proti nástroje **ccache**, navštivte domácí stránku ccache [<http://ccache.samba.org/>].

15.2.2. Instalace ccache

Instalace **ccache** s Gentoo je lehounké. Stačí použít **emerge** a je to :)

Obrázek 15.6. Instalace ccache

```
# emerge ccache
```

15.2.3. Aktivace podpory v Portage

Nejdříve ze všeho otevřete soubor `/etc/make.conf` a změňte proměnnou `FEATURES` tak aby obsahovala klíčové slovo `ccache`:

Obrázek 15.7. Úprava proměnné FEATURES v /etc/make.conf

```
FEATURES="ccache"
```

Dále upravte (nebo vytvořte) proměnnou `CCACHE_SIZE` (také v `/etc/make.conf`) tak aby obsahovala velikost prostoru na disku který chcete vyhradit pro ccache:

Obrázek 15.8. Úprava proměnné CCACHE_SIZE v /etc/make.conf

```
CCACHE_SIZE="2G"
```

Od této chvíle bude Portage používat cache pro zrychlení kompilace kde to je jenom možné. Pokud si nejste jisti zda **ccache** pracuje, můžete spustit příkaz **ccache -s** kterým zobrazíte statistiku **ccache**:

Obrázek 15.9. Zobrazení statistik ccache

```
# ccache -s
```

15.3. Binární balíčky

15.3.1. Vytváření binárních balíčků

Už jsme probírali jak pracovat s předkompilovanými balíčky, ale jak vytvářet svoje vlastní binární balíčky?

Pokud je balíček již nainstalován, můžete použít příkaz `quickpkg` který vytvoří tar archiv nainstalovaných souborů. To je velice zajímavé pro účely zálohování!

Obrázek 15.10. Použití quickpkg

```
# quickpkg gnumeric
```

Pokud balíček ještě nainstalován není, můžete ho nainstalovat pomocí `emerge` a požádat o současné vytvoření binárního balíčku. K tomu slouží přepínač `--buildpkg` (zkráceně `-b`):

Obrázek 15.11. Instalace programu gnumeric a současné vytvoření binárního balíčku

```
# emerge --buildpkg gnumeric
```

Pokud chcete aby Portage toto dělal implicitně, měli byste klíčové slovo `buildpkg` doplnit do proměnné `FEATURES` deklarované v souboru `/etc/make.conf`.

Obrázek 15.12. Automatické vytváření binárních balíčků

```
FEATURES="buildpkg"
```

Pokud nechcete program instalovat, ale pouze vytvořit binární balíček, měli byste použít přepínač `-buildpkgonly` (zkráceně `-B`):

Obrázek 15.13. Vytvoření binárního balíčku pro gnumeric

```
# emerge --buildpkgonly gnumeric
```

15.4. Vlastnosti související s bezpečností

15.4.1. Sandbox

Během kompilace a instalace balíčků používá Portage k ochraně vašeho systému sandbox. To znamená že dokud balíček není nainstalován na vašem systému, nemůže přistupovat k žádným souborům mimo sandbox. To zaručuje že Portage pozná které soubory jsou vytvářeny a modifikovány balíčkem.

Po dokončení kompilace balíčku Portage "předinstaluje" balíček do sandboxu, zjistí které soubory jsou vytvářeny a na jakém místě. Poté tyto soubory přemístí na váš systém.

15.4.2. Uživatelská práva

Portage podporuje kompilaci balíčků pod běžným uživatelským účtem (přesněji jako uživatel "portage" ze skupiny "portage"). To zvyšuje bezpečnost během procesu kompilace. Můžete si zvolit uživatelská práva s použitím sandboxu nebo bez něj. Samozřejmě, je bez řečí že uživatelská práva společně s použitím sandboxu jsou nejdoporučovanější metodou :)

15.4.3. Aktivace sandboxu a/nebo uživatelských práv

Portage implicitně používá sandbox. Pokud chcete i uživatelská práva, měli byste do proměnné FEATURES doplnit potřebné klíčové slovo `userpriv`. Všimněte si že aktivace uživatelských práv zruší podporu sandboxu pokud současně nezapnete i jeho podporu pomocí klíčového slova `usersandbox`:

Obrázek 15.14. Aktivace uživatelských práv a sandboxu

```
FEATURES="userpriv usersandbox"
```

Varování

Neodstraňujte klíčové slovo `sandbox` z proměnné FEATURES!

15.4.4. Striktní kontroly

Portage může být požádán aby silně reagoval na potenciálně nebezpečné podmínky (jako jsou například chybějící nebo nesprávný Manifest soubory). Pro aktivaci těchto striktních kontrol stačí doplnit klíčo-

vé slovo `strict` do proměnné `FEATURES`:

Obrázek 15.15. Aktivace striktních kontrol

```
FEATURES="strict"
```

15.4.5. Smart File System práva

Systému Portage může být oznámeno aby automaticky zpracovával nebezpečná nastavení práv souborů, která by mohla představovat bezpečnostní riziko. Toho dosahuje odstraněním "others" a "groups" práva ke čtení na souborech se `setuid` bitem a "other" práva ke čtení na `setgid` souborech během "předinstalace". K zapnutí Smart File System práv doplňte klíčové slovo `sfperms` do proměnné `FEATURES`:

Obrázek 15.16. Aktivace Smart File System práv

```
FEATURES="sfperms"
```

15.5. Další vlastnosti

15.5.1. Portage nápověda

Existuje ještě několik dalších klíčových slov která lze použít v proměnné `FEATURES`. Většina z nich je cílena na vývojáře a není tak zajímavá pro běžného uživatele. Pokud se chcete naučit více o těchto vlastnostech (nebo obecně o Portage), nezapomeňte si přečíst manuálovou stránku kterou poskytujeme k `make.conf`.

Obrázek 15.17. Více informací týkajících se Portage

```
# man make.conf
```

Kapitola 16. Kontrola chování Portage

Abstrakt

Portage můžete přizpůsobit vlastním potřebám/prostředí. Naučíte se jak chránit soubory, vytvářet zrcadlové obrazy (mirrors), měnit umístění adresářů a další.

16.1. Nastavení ochrany souborů

16.1.1. Ochrana?

Portage zná pojem "chráněné soubory". To znamená že při aktualizaci software nebudou některé soubory automaticky přepsány, ale budete informováni o tom že existuje novější verze. Tato vlastnost je samozřejmě velice užitečná pro konfigurační soubory (soubory umístěvané do adresáře `/etc`).

Namísto přepsání takových souborů bude vytvořen nový soubor s názvem `._cfg0000_<jméno>` kde `<jméno>` je jméno původního souboru. Potom je na uživateli aby do existujícího souboru zanesl potřebné změny, přičemž ke zjednodušení tohoto úkolu je možné použít nástroj **etc-update** o kterém budeme mluvit později.

16.1.2. Deklarace CONFIG_PROTECT

Portage neumí chránit jednotlivé soubory, namísto toho chrání celé adresáře. Proměnná `CONFIG_PROTECT` obsahuje seznam všech chráněných adresářů, přičemž chráněny jsou i všechny jejich podadresáře. Proměnná `CONFIG_PROTECT` je definována v souboru `/etc/make.globals`, ale pokud ji chcete změnit předefinujte ji v souboru `/etc/make.conf` (aby byla udržena konzistence je veškerá konfigurace Portage umístěna do souboru `/etc/make.conf`).

Obrázek 16.1. Příklad nastavení proměnné CONFIG_PROTECT

```
CONFIG_PROTECT="/etc /usr/share/config /usr/kde/3.1/share/config"
```

Pokud chcete konkrétní adresář označit jako chráněný, ale nechcete jako chráněné označit také všechny jeho podadresáře, můžete je "odchránit" tak že je uvedete v proměnné `CONFIG_PROTECT_MASK` která je také definována v souboru `/etc/make.globals` a měněna by měla být v `/etc/make.conf`.

Obrázek 16.2. Příklad nastavení CONFIG_PROTECT_MASK

```
CONFIG_PROTECT_MASK="/etc/init.d"
```

Další informace o nastavení ochrany souborů lze najít v online nápovědě k příkazu **emerge**:

Obrázek 16.3. Zobrazení informací o nastavení ochrany souborů

```
# emerge --help config
```

16.1.3. etc-update

Nástroj **etc-update** pomáhá při slučování souborů `._cfg0000_<jméno>`. Poskytuje interaktivní nastavení sloučení a triviální změny umí zpracovat automaticky.

Použití **etc-update** je přímočaré:

Obrázek 16.4. Spuštění etc-update

```
# etc-update
```

Po sloučení triviálních změn vám bude předložen seznam chráněných souborů které je potřeba upravit. V dolní části jsou vám nabídnuty možné volby:

Obrázek 16.5. Volby etc-update

```
Please select a file to edit by entering the corresponding number.
      (-1 to exit) (-3 to auto merge all remaining files)
      (-5 to auto-merge AND not use 'mv -i'):
```

Pokud zadáte `-1`, `etc-update` skončí bez provedení jakýchkoliv změn. Pokud zadáte `-3` nebo `-5`, všechny vypsané konfigurační soubory budou přepsány novější verzí. Proto je důležité nejdříve vybrat konfigurační soubory které by automaticky upravovány být neměly. To je jednoduché, protože stačí zadat číslo vypsané nalevo od tohoto konfiguračního souboru.

Jako příklad jsme si vybrali konfigurační soubor `/etc/pear.conf`:

Obrázek 16.6. Aktualizace konkrétního konfiguračního souboru

```
Beginning of differences between /etc/pear.conf and /etc/._cfg0000_pear.conf
[...]
End of differences between /etc/pear.conf and /etc/._cfg0000_pear.conf
1) Replace original with update
```

- 2) Delete update, keeping original as is
- 3) Interactively merge original with update
- 4) Show differences again

Můžete se podívat na rozdíly mezi těmi dvěma soubory. Pokud si myslíte že novější soubor může být bez problémů použit, zadejte 1. Pokud si myslíte že novější soubor není vůbec potřeba, nebo neobsahuje žádné nové nebo potřebné informace, zadejte 2. Pokud chcete interaktivně aktualizovat aktuální konfigurační soubor, zadejte 3.

Na tomto místě se nemusíme zabývat interaktivní úpravou. Abychom měli čisté svědomí tak uvedeme možné příkazy které můžete při interaktivním slučování dvou souborů používat. Objeví se dva řádky (původní a nově navrhovaný) a příkazový řádek na který můžete zadat jednu z následujících voleb:

Obrázek 16.7. Příkazy použitelné při interaktivním slučování

```
ed:      Edit then use both versions, each decorated with a header.
eb:      Edit then use both versions.
el:      Edit then use the left version.
er:      Edit then use the right version.
e:       Edit a new version.
l:       Use the left version.
r:       Use the right version.
s:       Silently include common lines.
v:       Verbosely include common lines.
q:       Quit.
```

Až skončíte s úpravou důležitých konfiguračních souborů, budete moci automaticky upravit všechny ostatní konfigurační soubory. Až nebude moci najít žádné aktualizovatelné konfigurační soubory tak se program **etc-update** sám ukončí.

16.2. Síťová nastavení

16.2.1. Mirrors

Protože se Gentoo stává stále populárnější distribucí, doporučujeme vám použití mirrorů. Portage používá tři proměnné pro jejich nastavení: jedna pro `rsync mirror` (oproti kterým se aktualizuje Portage Tree), jedna pro `distfiles` (oproti kterým se aktualizuje zdrojový kód a jedna pro předkompilované balíčky.

Všechny mirror pro `distfiles` jsou vypsané v našem seznamu Gentoo mirrorů [<http://www.gentoo.org/main/en/mirrors.xml>]. Můžete také použít nástroj `mirrorselect` který zjednoduší nastavení mirrorů pro váš systém. Ale nejdříve se podívejme na jednotlivé proměnné ...

Proměnná `SYNC` obsahuje seznam `rsync-mirror`ů které chcete používat. Například pro použití

`rsync://rsync.namerica.gentoo.org/gentoo-portage` jako primárního serveru a `rsync://rsync.samerica.gentoo.org/gentoo-portage` jako sekundárního serveru bychom do `/etc/make.conf` přidali:

Obrázek 16.8. Definice SYNC v `/etc/make.conf`

```
SYNC="rsync://rsync.namerica.gentoo.org/gentoo-portage
      rsync://rsync.samerica.gentoo.org/gentoo-portage"
```

Proměnná `GENTOO_MIRRORS` obsahuje seznam mirrorů distfiles které chcete používat. Například pro použití `ftp://ibiblio.org/pub/Linux/distributions/gentoo` jako primárního serveru a `http://www.gtlib.cc.gatech.edu/pub/gentoo` jako sekundárního serveru bychom do `/etc/make.conf` přidali:

Obrázek 16.9. Definice GENTOO_MIRRORS v `/etc/make.conf`

```
GENTOO_MIRRORS="ftp://ibiblio.org/pub/Linux/distributions/gentoo
                http://www.gtlib.cc.gatech.edu/pub/gentoo"
```

Proměnná `PORTAGE_BINHOST` obsahuje seznam mirrorů s předkompilovanými balíčky, které chcete používat. Například pokud chcete použít server `ftp://login:pass@grp.mirror.site/pub/grp/i686/athlon-xp`, přidali byste do `/etc/make.conf`:

Obrázek 16.10. Definice PORTAGE_BINHOST v `/etc/make.conf`

```
PORTAGE_BINHOST="ftp://login:pass@grp.mirror.site/pub/grp/i686/athlon-xp"
```

16.2.2. Mirrorselect

Pokud chcete použít nástroj `mirrorselect`, nejdříve ho nainstalujte (pokud jste tak již neučinili).

Obrázek 16.11. Instalace nástroje mirrorselect

```
# emerge mirrorselect
```

Nyní můžete `mirrorselectu` říci aby automaticky zvolil nejlepší mirrorserver, tj. mirrorserver umístěný co

nejblíže, nebo zvolit mirrorserver ručně ze seznamu. Další informace o tom jak používat **mirrorselect** stačí pouze spustit **mirrorselect** z příkazové řádky - nabídne vám krátký přehled jak ho používat.

Obrázek 16.12. Spuštění nástroje mirrorselect

```
# mirrorselect
```

16.2.3. Stahování

Program využívaný Portage ke stahování archivů lze specifikovat nastavením proměnných `FETCHCOMMAND` a `RESUMECOMMAND`. Několik příkladů je uvedeno v `/etc/make.conf` a `/etc/make.globals`. Portage implicitně používá **wget**:

Obrázek 16.13. Výchozí nastavení proměnných FETCHCOMMAND a RESUMECOMMAND

```
FETCHCOMMAND="/usr/bin/wget -t 5 --passive-ftp -P \${DISTDIR} \${URI}"  
RESUMECOMMAND="/usr/bin/wget -c -t 5 --passive-ftp -P \${DISTDIR} \${URI}"
```

Proměnná `\${DISTDIR}` je nahrazena za adresář do kterého jsou ukládány stažené soubory (`/usr/portage/distfiles`), zatímco proměnná `\${URI}` je nahrazena za soubor který Portage potřebuje stáhnout.

Protože Portage implicitně používá **wget**, můžete proxy servery nakonfigurovat jednoduchým nadefinováním proměnných `http_proxy` a `ftp_proxy` (všimněte si malých písmen). Ačkoliv tyto proměnné můžete nastavit i v souboru `/etc/make.conf`, doporučujeme vám použít klasický postup protože `http_proxy` a `ftp_proxy` jsou používány i dalšími nástroji (soubor `/etc/make.conf` je používán pouze pro Portage). Přečtěte si prosím kapitolu o proměnných prostředí ve které se dočtete jak nastavovat systémové proměnné prostředí.

16.2.4. Nastavení programu rsync

Program **rsync** je používán nástrojem **emerge** k aktualizaci Portage Tree. Chování programu **rsync** ve vztahu k Portage lze ovlivňovat třemi proměnnými, a to `RSYNC_EXCLUDEFROM`, `RSYNC_RETRIES` a `RSYNC_TIMEOUT`.

Jedním ze způsobů jak chránit ebuildy před aktualizací nebo odstraněním po spuštění **emerge sync** je použití proměnné `RSYNC_EXCLUDEFROM`. Měla by být nastavena na soubor který je programem **rsync** používán k vyjmutí některých souborů a implicitně je umístěn `/etc/portage/rsync_excludes`. Použití této metody však není doporučováno, protože v případě neopatrnosti může dojít k porušení závislostí. Později si povíme o proměnné `PORTDIR_OVERLAY`, což je doporučovaný způsob jak dosáhnout stejného efektu. Další informace najdete v manuálové stránce programu **rsync**.

Obrázek 16.14. Proměnná RSYNC_EXCLUDEFROM

```
RSYNC_EXCLUDEFROM="/etc/portage/rsync_excludes"
```

Pokud **rsync** selže, pokusí se několikrát o stejný úkon než se přepne na další dostupný rsync server. Počet těchto opětovných pokusů je definován v proměnné **RSYNC_RETRIES** a implicitní hodnota je 3.

Obrázek 16.15. Proměnná RSYNC_RETRIES

```
RSYNC_RETRIES="3"
```

Pokud používáte (velice) pomalý rsync server, může dojít k vypršení časového limitu. Počet vteřin po které má program rsync čekat než přejde k dalšímu serveru je definován v proměnné **RSYNC_TIMEOUT** a přičemž výchozí hodnota je 180:

Obrázek 16.16. Proměnná RSYNC_TIMEOUT

```
RSYNC_TIMEOUT="180"
```

16.3. Umístění adresářů

16.3.1. Úvod

Každý aspekt chování Portage je konfigurovatelný, včetně toho které adresáře a soubory jsou pro jednotlivé úkoly používány. Změny výchozích hodnot (definovaných v `/etc/make.globals`) dosáhnete nadefinováním příslušných proměnných - kde jinde než :) - v souboru `/etc/make.conf`.

Varování

Pokud změníte proměnnou tak aby ukazovala na nové místo, nikdy neukončujte cestu znakem `"/`!"

16.3.2. Portage Tree

Umístění Portage Tree je definováno proměnnou `PORTDIR`, jejíž výchozí hodnota je `/usr/portage`:

Obrázek 16.17. Proměnná PORTDIR

```
PORTDIR="/usr/portage"
```

Pokud si chcete vedle "oficiálního" udržovat ještě lokální Portage Tree, potřebujete nastavit proměnnou `PORTDIR_OVERLAY`. Adresáře vypsané v této hodnotě nebudou ovlivňovány příkazem **emerge sync**: ebuildy v těchto adresářích nebudou aktualizovány nebo odstraňovány, ale přitom jsou součástí Portage Tree.

Obrázek 16.18. Proměnná `PORTDIR_OVERLAY`

```
PORTDIR_OVERLAY="/usr/local/portage"
```

16.3.3. Distfiles

Umístění staženého zdrojového kódu (takzvaných distfiles) je definováno v proměnné `DISTDIR`. Výchozí hodnota je `${PORTDIR}/distfiles`:

Obrázek 16.19. Proměnná `DISTDIR`

```
DISTDIR="${PORTDIR}/distfiles"
```

16.3.4. Balíčky a RPM

Umístění předkompilovaných balíčků určuje proměnná `PKGDIR`. Implicitní hodnota je `${PORTDIR}/packages`:

Obrázek 16.20. Proměnná `PKGDIR`

```
PKGDIR="${PORTDIR}/packages"
```

Umístění RPM balíčků (ano, některé balíčky jsou dostupné jako RPM) je definováno proměnnou `RPMDIR`. Výchozí umístění je `${PORTDIR}/rpm`:

Obrázek 16.21. Proměnná

```
RPMDIR="${PORTDIR}/rpm"
```

16.3.5. Dočasné soubory Portage

Ke kompilaci (sestavování) ebuildů používá Portage dočasný adresář. Jeho umístění je definováno v proměnné `PORTAGE_TMPDIR`, jejíž výchozí hodnota je `/var/tmp`:

Obrázek 16.22. Proměnná `PORTAGE_TMPDIR`

```
PORTAGE_TMPDIR="/var/tmp"
```

Výchozí nastavení je takové že v adresáři `PORTAGE_TMPDIR` dojde k vytvoření adresáře `portage`. To je definováno proměnnou `BUILD_PREFIX`:

Obrázek 16.23. Proměnná `BUILD_PREFIX`

```
BUILD_PREFIX="${PORTAGE_TMPDIR}/portage"
```

Pokud máte v úmyslu změnit umístění, ujistěte se že tento dočasný adresář je na diskovém oddílu s dostatkem volného prostoru, protože při kompilaci velkých softwarových balíků může být tento adresář i větší než 2 GB!

16.3.6. Logování

Proměnná `PORT_LOGDIR` je speciální a implicitně vůbec není nastavena. Pokud ji definujete, Portage bude v daném adresáři pro každý ebuild vytvářet log:

Obrázek 16.24. Proměnná `PORT_LOGDIR`

```
PORT_LOGDIR="/var/log/portage"
```

16.4. Další nastavení Portage

16.4.1. Nice Value

Portage umožňuje kompilaci s jinou nice-value (jakoby prioritou). Pokud chcete aby Portage kompiloval balíčky s vyšší nice-value (což má za následek hbitější odezvu systému během procesu kompilace, ale současně zvyšuje dobu kompilace) můžete nadefinovat proměnnou `PORTAGE_NICENESS` s kladným celým číslem:

Obrázek 16.25. Proměnná PORTAGE_NICENESS

```
PORTAGE_NICENESS="3"
```

16.4.2. SLOTované balíčky a automatické čištění

V některých situacích potřebujete mít v systému k dispozici různé verze jednoho balíčku (včetně knihoven). Portage toto podporuje definováním proměnné `SLOT` v ebuildch. Jako uživatel nepotřebujete vědět jak SLOTování funguje, ale je důležité vědět že tato vlastnost funguje.

Pokud instalujete novější verzi balíčku, Portage zkontroluje zda je v něm nadefinována proměnná `SLOT`. Pokud ano, a pokud je její hodnota pro oba balíčky (nový a starý) různá, Portage se starým balíčkem manipulovat nebude.

Avšak pokud jsou hodnoty proměnné `SLOT` různé (a to je obvyklý případ), starší balíček bude odstraněn. Aby mohl uživatel zabránit tomuto odstranění, bude Portage odpočítávat určitý počet vteřin. Tento počet je definován v proměnné `CLEAN_DELAY` a výchozí hodnota je 5 vteřin:

Obrázek 16.26. Proměnná CLEAN_DELAY

```
CLEAN_DELAY="5"
```

Pokud nechcete aby Portage automaticky odstraňoval starší balíčky (což je nazýváno "cleaning" tj. "čištění"), můžete nastavit proměnnou `AUTOCLEAN` na `no`:

Obrázek 16.27. Proměnná AUTOCLEAN

```
AUTOCLEAN="no"
```

16.4.3. Kompilace souvisejících balíčků

Už jsme probrali poměrně dost proměnných, ale ještě nejsme na konci. Ti z vás kteří instalovali Gentoo si pamatují proměnné `CHOST`, `CFLAGS` a `CXXFLAGS`, používané překladačem ke kompilaci a optimalizaci balíčků.

Další informace o těchto proměnných naleznete v manuálových stránkách k programu `gcc`, nebo na webu v GCC Online Manuals [<http://www.gnu.org/software/gcc/onlinedocs/>].

Obrázek 16.28. Zobrazení informací o CHOST, CFLAGS and CXXFLAGS

```
# info gcc
(Select "Invoking gcc")
(Select "Optimize options")
```

Pokud je proměnná `DEBUGBUILD` definována, Portage nebude binární soubory a knihovny čistit (strip) aby zjednodušil jejich ladění. To však zpomaluje váš systém a zvětšuje soubory.

Obrázek 16.29. Výchozí nastavení proměnné `DEBUGBUILD`

```
# Do not set this to "false"; instead remove the line. Portage does not
# check the value, it just checks if the variable is defined.
DEBUGBUILD="true"
```

Proměnná `MAKEOPTS` je používána příkazem **make**, což je nástroj zjednodušující kompilaci balíčku. Obvykle je definován tak aby provedl několik kompilací současně (zejména pokud máte víceprocesorový systém nebo pokud používáte nástroj **distcc** popsáný výše).

Aby **make** prováděl vždy tři kompilace současně, nastavte proměnnou `MAKEOPTS` na hodnotu `-j 3`:

Obrázek 16.30. Proměnná `MAKEOPTS`

```
MAKEOPTS="-j 3"
```

Proměnná `ROOT` by neměla být nastavena v souboru `/etc/make.conf`, ale raději jako proměnná prostředí. Portage kontroluje tuto proměnnou aby zjistil kam má nainstalovat balíček. Samozřejmě, implicitní hodnota je `/`. V příkladu je ukázáno jak nainstalovat program **gnumeric** do `/mnt/gentoo` namísto do vašeho běžícího systému:

Obrázek 16.31. Proměnná `ROOT`

```
# ROOT="/mnt/gentoo" emerge gnumeric
```

16.4.4. Formátování výstupu

Portage je implicitně nakonfigurován tak aby barevný výstup byl co nejpřehlednější. Pokud toto nechcete, nastavte proměnnou `NOCOLOR` na `true`:

Obrázek 16.32. Proměnná `NOCOLOR`

```
NOCOLOR="true"
```

16.5. Zdroje

16.5.1. Manuálové stránky

Pokud potřebujete seznam s krátkým popisem všech uvedených proměnných, podívejte se do manuálové stránky k souboru `make.conf`:

Obrázek 16.33. Zobrazení informací v `make.conf`

```
# man make.conf
```

16.5.2. Komentované příklady

Soubor `/etc/make.conf` obsahuje spoustu komentářů, včetně příkladů které by pro vás mohly být zajímavé. Avšak většina uživatelů soubor `/etc/make.conf` neupravuje a proto jim tyto informace unikají. Nejnovější `/etc/make.conf` můžete najít v našem Online ViewCVS Repository [http://www.gentoo.org/cgi-bin/viewcvs.cgi/*checkout*/portage/cnf/make.conf?rev=HEAD&cvsroot=gentoo-src&content-type=text/plain].

Kapitola 17. Inicializační skripty

Abstrakt

Gentoo používá zvláštní formát inicializačních souborů který, kromě jiných vlastností, umožňuje závislostmi řízená rozhodnutí (dependency-driven) a virtuální inicializační skripty. V této části jsou všechny tyto vlastnosti vysvětleny a to včetně toho jak tyto skripty používat.

17.1. Runlevely

17.1.1. Start systému

Když startujete systém, běhá po obrazovce spousta textu. Pokud budete dávat pozor, všimnete si že tento text je při každém startu systému stejný. Posloupnost všech těchto kroků se nazývá startovací posloupnost (boot sequence) a je víceméně staticky definována.

Nejdříve ze všeho zavaděč (boot loader) načte obraz jádra (definovaný v konfiguraci zavaděče) do paměti a poté přikáže procesoru aby toto jádro spustil. Když je jádro načteno a spuštěno, jsou inicializovány struktury potřebné pro jeho běh a pro běh úloh, a je spuštěn proces **init** (inicializační proces).

Tento proces zkontroluje že všechny souborové systémy (definované v `/etc/fstab`) jsou připojeny a připraveny k použití. Poté jsou spuštěny skripty umístěné v adresáři `/etc/init.d`, což způsobí spuštění všech služeb potřebných pro plně funkční systém.

Nakonec, když jsou všechny skripty spuštěny, aktivuje **init** terminály (ve většině případů virtuální konzole skryté za **Alt-F1**, **Alt-F2**, atd.) ke kterým je připojen speciální proces nazývaný **agetty**. Tento proces spuštěním programu **login** zajišťuje že se přes tyto terminály můžete přihlásit.

17.1.2. Inicializační skripty

Inicializační skripty umístěné v `/etc/init.d` však nejsou spouštěny v náhodném pořadí. Ba co víc, **init** dokonce nespouští všechny skripty umístěné v tomto adresáři ale jenom ty u kterých je mu to nařízeno. To které skripty budou spuštěny je rozhodnuto na základě obsahu adresáře `/etc/runlevels`.

Nejdříve ze všeho **init** spustí skripty z `/etc/init.d`, které mají symbolické odkazy do `/etc/runlevels/boot`. Skripty jsou obvykle spouštěny v abecedním pořadí, ale některé skripty v sobě mají definovanu závislost, čímž sdělují systému že před jejich spuštěním musí být spuštěn některý jiný skript.

Když jsou spuštěny všechny skripty odkazované z `/etc/runlevels/boot`, pokračuje **init** ve spuštění skriptů odkazovaných z `/etc/runlevels/default`. A opět, při spuštění skriptů bude použito abecední řazení, nicméně některé skripty mohou obsahovat informaci o závislosti a v tom případě je pořadí pozměněno tak aby byla posloupnost skriptů platná.

17.1.3. Jak to funguje

Samozřejmě proces **init** to vše nerozhoduje sám. Potřebuje konfigurační soubor který určuje které akce je nutno provést. Tento konfigurační soubor se jmenuje `/etc/inittab`.

Pokud si pamatujete na startovací posloupnost kterou jsme vám před chvílí vysvětlili, vzpomenete si že první akce prováděná procesem **init** je připojení všech souborových systémů. To je definováno následující řádkou v `/etc/inittab`:

Obrázek 17.1. Inicializace systému v `/etc/inittab`

```
si::sysinit:/sbin/rc sysinit
```

Tento řádek říká procesu **init** že musí spustit `/sbin/rc sysinit` který inicializuje systém. Skript `/sbin/rc` se stará o inicializaci, takže by se dalo říct že **init** vlastně sám nic moc nedělá - pouze předává úlohu inicializovat systém jinému procesu.

Zadruhé, proces **init** spouští všechny skripty na které existuje symbolický odkaz v adresáři `/etc/runlevels/boot`. To je definováno následující řádkou:

Obrázek 17.2. Pokračování inicializace systému

```
rc::bootwait:/sbin/rc boot
```

A opět, příslušné úlohy provádí skript `rc`. Všimněte si že volba předávaná skriptu `rc` (tj. `boot`) je stejná jako jméno podadresáře `/etc/runlevels`.

Proces **init** poté zkontroluje konfigurační soubor aby zjistil který runlevel má spustit. To udělá tak že ze souboru `/etc/inittab` přečte následující řádek:

Obrázek 17.3. Řádek `initdefault`

```
id:3:initdefault:
```

V tomto případě (který bude používat většina uživatelů Gentoo), je id runlevelu 3. Díky této informaci **init** ví že má nastartovat do runlevelu 3:

Obrázek 17.4. Definice runlevelů

```
10:0:wait:/sbin/rc shutdown
11:S1:wait:/sbin/rc single
```

```
l2:2:wait:/sbin/rc nonetwork
l3:3:wait:/sbin/rc default
l4:4:wait:/sbin/rc default
l5:5:wait:/sbin/rc default
l6:6:wait:/sbin/rc reboot
```

Řádek definující runlevel 3, opět, používá skript `rc` pro nastartování služeb (nyní s volbou `default`). Opět si všimněte že argument `rc` se shoduje se jménem podadresáře `/etc/runlevels`.

Když skript `rc` skončí, proces **init** se rozhodne které virtuální konzole by měly být aktivovány a které příkazy je potřeba na konzolích spustit:

Obrázek 17.5. Definice virtuálních konzolí

```
c1:12345:respawn:/sbin/agetty 38400 tty1 linux
c2:12345:respawn:/sbin/agetty 38400 tty2 linux
c3:12345:respawn:/sbin/agetty 38400 tty3 linux
c4:12345:respawn:/sbin/agetty 38400 tty4 linux
c5:12345:respawn:/sbin/agetty 38400 tty5 linux
c6:12345:respawn:/sbin/agetty 38400 tty6 linux
```

17.1.4. Co je to runlevel?

Viděli jste že proces **init** používá systém číslování k rozhodnutí které runlevely by měly být aktivovány. Runlevel je stav ve kterém váš systém běží a obsahuje skupinu skriptů (runlevel skripty nebo initskripty) které musí být spuštěny pokud chcete do tohoto runlevelu přejít.

V Gentoo existuje sedm definovaných runlevelů: tři z nich jsou interní a čtyři uživatelské. Interní runlevely se nazývají `sysinit`, `shutdown` a `reboot` a dělají přesně to co znamenají jejich jména: inicializaci systému, vypnutí systému a restart systému.

Uživatelské runlevely jsou ty které používají podadresáře `/etc/runlevels`: `boot`, `default` a `nonetwork` a `single`. Runlevel `boot` nastartuje nutné systémové služby používané všemi ostatními runlevely. Zbývající tři runlevely se liší pouze v tom které služby spouští: `default` je používán pro každodenní používání, `nonetwork` je používán v případě že nepotřebujete připojení k síti, a `single` je používán pokud potřebujete opravit systém.

17.1.5. Práce s inicializačními skripty

Skripty spouštěné procesem **rc** se nazývají inicializační skripty. Každý skript v `/etc/init.d` může být spouštěn s argumenty `start`, `stop`, `restart`, `pause`, `zap`, `status`, `inneed`, `iuse`, `needs-me`, `usesme` nebo `broken`.

Pro spuštění, ukončení nebo restart služby (a všech závislých služeb) lze použít volby `start`, `stop` a `restart`:

Obrázek 17.6. Spuštění postfixu

```
# /etc/init.d/postfix start
```

Poznámka

Zastaveny nebo restartovány jsou pouze ty služby které příslušnou službu potřebují. Další související služby (ty které ji používají ale nepotřebují) jsou ponechány.

Pokud chcete zastavit službu, ale nikoliv služby které na ní závisí, můžete použít volbu `pause`:

Obrázek 17.7. Zastavení postfixu, ale závislé služby stále běží

```
# /etc/init.d/postfix pause
```

Pokud chcete zjistit stav služby (spuštěná, zastavená, pozastavená, ...) můžete použít argument `status`:

Obrázek 17.8. Informace o stavu pro postfix

```
# /etc/init.d/postfix status
```

Pokud informace o stavu říká že služba běží, ale vy víte že nikoliv, můžete informaci o stavu resetovat na "zastavena" pomocí volby `zap`:

Obrázek 17.9. Resetování stavové informace pro postfix

```
# /etc/init.d/postfix zap
```

Pokud vás zajímá jaké služby jsou spojené s určitou službou, můžete použít volby `iuse` nebo `ineed`. Volba `ineed` vypíše služby nutné pro korektní funkci služby, a volba `iuse` naopak zobrazí seznam služeb které mohou být službou používány, ale nejsou nutné pro korektní funkci.

Obrázek 17.10. Zobrazení seznamu služeb vyžadovaných pro běh postfixu

```
# /etc/init.d/postfix ineed
```

Podobně se můžete zeptat které služby danou službu vyžadují (volba `needsme`) nebo používají (volba `usesme`):

Obrázek 17.11. Zobrazení seznamu služeb které pro svůj běh vyžadují postfix

```
# /etc/init.d/postfix needsme
```

Konečně, můžete se zeptat které závislosti služba vyžaduje a které přitom chybí:

Obrázek 17.12. Zobrazení seznamu závislostí které postfixu chybí

```
# /etc/init.d/postfix broken
```

17.2. Práce s rc-update

17.2.1. Co je rc-update?

Inicializační systém Gentoo používá k rozhodnutí v jakém pořadí je potřeba služby spouštět závislostní strom. Protože toto je nutná úloha kterou by uživatelé neměli dělat ručně, vytvořili jsme nástroje které ulehčí administraci runlevelů a initskriptů.

Pomocí programu **rc-update** můžete přidávat a odebírat z runlevelu. Nástroj **rc-update** poté automaticky spustí skript **depscan.sh** který znovu vytvoří závislostní strom.

17.2.2. Přidávání a odebírání služeb

Během instalace Gentoo už jste přidávali skripty do `default` runlevelu. V té době jste ještě nemuseli vědět co volba `default` znamená, ale nyní už to víte. Skript **rc-update** navíc vyžaduje ještě jeden argument definující akci: `add`, `del` nebo `show`.

Pro přidání nebo odebrání inicializačního skriptu stačí skriptu **rc-update** předat argument `add` nebo `del`, následovaný jménem init skriptu a runlevelu. Například:

Obrázek 17.13. Odstranění postfixu z runlevelu "default"

```
# rc-update del postfix default
```

Příkaz **rc-update show** zobrazí všechny dostupné init skripty a seznam runlevelů ve kterých budou

spouštěny:

Obrázek 17.14. Zobrazení seznamu dostupných init skriptů

```
# rc-update show
```

17.3. Konfigurace služeb

17.3.1. Proč je potřeba další konfigurace?

Init skripty mohou být dosti složité. Z toho důvodu není vhodné aby uživatelé tyto inicializační skripty přímo upravovali, protože by to bylo náchylné k chybám. Možnost konfigurace takových služeb je však důležitá. Například byste mohli chtít službě předávat více voleb.

Dalším důvodem pro umístění konfigurace mimo inicializační skript je možnost aktualizace inicializačních skriptů bez obav že přijdete o své nastavení.

17.3.2. Adresář `/etc/conf.d`

Gentoo poskytuje skutečně jednoduchý způsob jak takové služby konfigurovat: každý skript který může být konfigurován má konfigurační soubor v `/etc/conf.d`. Například inicializační skript `apache2` (nazývaný `/etc/init.d/apache2`) má konfigurační soubor s názvem `/etc/conf.d/apache2`, který může obsahovat volby které chcete při startu předat serveru Apache 2:

Obrázek 17.15. Proměnné definované v `/etc/conf.d/apache2`

```
APACHE2_OPTS="-D PHP4"
```

Takový konfigurační soubor obsahuje jen a pouze proměnné (stejně jako `/etc/make.conf`), díky čemuž je konfigurace služeb velice jednoduchá. Tento přístup také umožňuje připojit podrobnější informace o proměnných (formou komentářů).

17.4. Psaní init skriptů

17.4.1. Musím?

Ne. Psaní init skriptů obvykle není potřeba neboť Gentoo poskytuje hotové init skripty pro všechny poskytované služby. Avšak možná jste nainstalovali některou službu bez použití Portage, v kterémžto případě nejspíše budete muset vytvořit vlastní init skript.

Nepoužívejte init skript dodávaný se službou pokud není výslovně určen pro Gentoo: init skripty Gentoo

nejsou kompatibilní s init skripty pro ostatní distribuce!

17.4.2. Schéma

Základní schéma init skriptu je uvedeno níže.

Obrázek 17.16. Základní schéma init skriptu

```
#!/sbin/runscript

depend() {
    (Dependency information)
}

start() {
    (Commands necessary to start the service)
}

stop() {
    (Commands necessary to stop the service)
}

restart() {
    (Commands necessary to restart the service)
}
```

Každý init skript vyžaduje definici funkce `start()`. Všechny ostatní funkce jsou nepovinné.

17.4.3. Závislosti

Existují dva druhy závislostí které můžete definovat: `use` a `need`. Jak již bylo uvedeno, závislost `need` je striktnější než `use` závislost. Za uvedením typu závislosti uveďte službu na které vaše služba závisí, nebo virtuální závislost.

Virtuální závislost je závislost kterou služba poskytuje, ale která není poskytována výhradně touto službou. Váš init skript může záviset na systémovém loggery, ale existuje mnoho systémových logů (`metalogd`, `syslog-ng`, `sysklogd`, ...). Protože nepotřebujete všechny (žádný rozumný systém nepoužívá všechny tyto loggery najednou) stačí zajistit že všechny tyto služby poskytují (anglicky "provide") virtuální závislost.

Podívejme se na informace o závislostech pro službu **postfix**.

Obrázek 17.17. Informace o závislostech postfixu

```
depend() {
    need net
}
```

```
use logger dns
provide mta
}
```

Jak vidíte, služba **postfix**:

- vyžaduje (virtuální) závislost `net` (ta je poskytována, například, init skriptem `/etc/init.d/net.eth0`)
- používá (virtuální) závislost `logger` (ta je poskytována, například, init skriptem `/etc/init.d/syslog-ng`)
- používá (virtuální) závislost `dns` (ta je poskytována, například, init skriptem `/etc/init.d/named`)
- poskytuje (virtuální) závislost `mta` (což je obvyklá virtuální závislost poskytovaná mail servery)

17.4.4. Určování pořadí

V některých případech sice službu nevyžadujete, ale pokud je v systému dostupná a na stejném runlevelu tak ji můžete chtít spustit dříve (či později) než jinou službu (všimněte si podmínek - nejedná se o žádnou závislost mezi službami a jedná se pouze o služby ve stejném runlevelu). Toho můžete docílit pomocí voleb `before` a `after`.

Podívejme se například na službu `Portmap`:

Obrázek 17.18. Funkce `depend()` služby `portmap`

```
depend() {
    need net
    before inetd
    before xinetd
}
```

Přitom můžete použít standardní znak `"*"` kterým označíte všechny služby v daném runlevelu, ačkoliv tento postup nedoporučujeme.

Obrázek 17.19. Spouštění služby jako první v celém runlevelu

```
depend() {
    before *
}
```

17.4.5. Standardní funkce

Kromě `depend()` potřebujete nadefinovat také funkci `start()`. Ta obsahuje všechny příkazy potřebné k inicializaci vaší služby. doporučujeme vám používat funkce `ebegin` a `epend` k informování uživatele o tom co se děje:

Obrázek 17.20. Příklad funkce `start()`

```
start() {  
    ebegin "Starting my_service"  
    start-stop-daemon --start --quiet --exec /path/to/my_service  
    eend $?  
}
```

Pokud se chcete podívat na další příklady funkce `start()`, podívejte se do zdrojového kódu dostupných init skriptů v adresáři `/etc/init.d`. Navíc existuje skvělá manuálová stránka pro **start-stop-daemon**:

Obrázek 17.21. Zobrazení manuálové stránky pro `start-stop-daemon`

```
# man start-stop-daemon
```

Další funkce které můžete definovat jsou `stop()` a `restart()`. Tyto funkce nemusíte definovat! Náš init systém je dostatečně inteligentní na to aby si tyto služby doplnil sám, pokud používáte **start-stop-daemon**.

17.4.6. Přidávání vlastních voleb

Pokud chcete aby init skript podporoval více voleb než jen ty které jsme právě uvedli, musíte přidat tuto volbu do proměnné `opts`, a vytvořit funkci se stejným jménem jako je volba. Například pokud chcete přidat volbu `restartdelay`:

Obrázek 17.22. Doplnění podpory pro proměnnou `restartdelay`

```
opts="${opts} restartdelay"  
  
restartdelay() {  
    stop()  
    sleep 3      # Wait 3 seconds before starting again  
    start()  
}
```

17.4.7. Proměnné pro konfiguraci služby

Pro to aby váš skript podporoval konfigurační soubor v `/etc/conf.d` nemusíte udělat vůbec nic: pokud je váš skript spuštěn jsou automaticky načteny následující soubory (tj. je možno používat v nich definované proměnné):

- `/etc/conf.d/<your init script>`
- `/etc/conf.d/basic`
- `/etc/rc.conf`

Také, pokud váš init skript poskytuje virtuální závislost (jako je například `net`), bude načten také soubor asociovaný s touto závislostí (čili například `/etc/conf.d/net`).

Kapitola 18. Proměnné prostředí

Abstrakt

S Gentoo můžete jednoduše spravovat proměnné prostředí na vašem systému. Tato kapitola vysvětluje jak na to a popisuje často používané proměnné.

18.1. Proměnné prostředí?

18.1.1. Co jsou zač

Proměnná prostředí je pojmenovaný objekt obsahující informace využívané jednou nebo více aplikacemi. Mnoho uživatelů (a zejména těch kteří jsou v Linuxu noví) to považuje za podivné nebo neovladatelné. To je však velký omyl: používáním proměnných prostředí lze jednoduše měnit nastavení jedné nebo více aplikací.

18.1.2. Důležité příklady

V následující tabulce je uveden seznam proměnných používaných Linuxovým systémem společně s popisem jejich použití. Pod tabulkou jsou uvedeny příklady použití.

PATH	Tato proměnná obsahuje dvojtečkou oddělovaný seznam adresářů ve kterých systém hledá spustitelné soubory. Pokud zadáte jméno spustitelného souboru (jako jsou například ls , rc-update nebo emerge) a tento soubor není umístěn v některém z uvedených adresářů, váš systém program nespustí (pokud k němu nezadáte celou cestu, jako je například /bin/ls).
ROOTPATH	Má stejný význam jako PATH, ale obsahuje pouze adresáře které mají být kontrolovány pokud příkaz zadal root.
LDPATH	Obsahuje seznam adresářů oddělovaných dvojtečkou ve kterých dynamický linker hledá knihovny.
MANPATH	Obsahuje seznam adresářů oddělovaných dvojtečkou ve kterých příkaz man hledá manuálové stránky.
INFODIR	Obsahuje seznam adresářů oddělovaných dvojtečkou ve kterých příkaz info hledá info stránky.
PAGER	Proměnná obsahující cestu k programu používanému pro listování obsahem souborů (například less nebo more).
EDITOR	Tato proměnná obsahuje cestu k programu použí-

	vanému k úpravě obsahu souborů (například nano nebo vi).
KDEDIRS	Tato proměnná obsahuje seznam adresářů oddělených dvojtečkou, které obsahují materiál specifický pro KDE.
CLASSPATH	Proměnná obsahuje seznam adresářů oddělených dvojtečkou, které obsahují třídy jazyka Java (Java classes).
CONFIG_PROTECT	Proměnná obsahuje seznam adresářů oddělovaných <i>mezerou</i> , které mají být chráněny systémem Portage během aktualizace.
CONFIG_PROTECT_MASK	Proměnná obsahuje seznam adresářů oddělovaných <i>mezerou</i> , které nemají být chráněny systémem Portage během aktualizace.

Níže najdete příklad definice všech těchto proměnných:

Obrázek 18.1. Vzorové definice

```
PATH="/bin:/usr/bin:/usr/local/bin:/opt/bin:/usr/games/bin"
ROOTPATH="/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin"
LDPATH="/lib:/usr/lib:/usr/local/lib:/usr/lib/gcc-lib/i686-pc-linux-gnu/3.2.3"
MANPATH="/usr/share/man:/usr/local/share/man"
INFODIR="/usr/share/info:/usr/local/share/info"
PAGER="/usr/bin/less"
EDITOR="/usr/bin/vim"
KDEDIRS="/usr"
CLASSPATH="/opt/blackdown-jre-1.4.1/lib/rt.jar:."
CONFIG_PROTECT="/usr/X11R6/lib/X11/xkb /opt/tomcat/conf \
                /usr/kde/3.1/share/config /usr/share/texmf/tex/generic/config/ \
                /usr/share/texmf/tex/platex/config/ /usr/share/config"
CONFIG_PROTECT_MASK="/etc/gconf"
```

18.2. Definice globálních proměnných

18.2.1. Adresář /etc/env.d

Aby byly definice těchto proměnných centralizovány, Gentoo zavádí adresář `/etc/env.d`, ve kterém najdete mnoho souborů pojmenovaných například `00basic`, `05gcc`, atd. které obsahují proměnné potřebné aplikací uvedenou v jejich jméně.

Například, když jste nainstalovali **gcc**, byl balíčkem automaticky vytvořen soubor `05gcc` obsahující definice následujících proměnných:

Obrázek 18.2. /etc/env.d/05gcc

```
PATH="/usr/i686-pc-linux-gnu/gcc-bin/3.2"
ROOTPATH="/usr/i686-pc-linux-gnu/gcc-bin/3.2"
MANPATH="/usr/share/gcc-data/i686-pc-linux-gnu/3.2/man"
INFOPATH="/usr/share/gcc-data/i686-pc-linux-gnu/3.2/info"
CC="gcc"
CXX="g++"
LDPATH="/usr/lib/gcc-lib/i686-pc-linux-gnu/3.2.3"
```

Jiné distribuce vám říkají abyste definice proměnných prostředí měnili nebo přidávali v souboru /etc/profile nebo na jiných místech. Gentoo vám (a Portage) naopak zjednodušuje údržbu a správu proměnných prostředí aniž byste se museli starat o spoustu souborů obsahujících proměnné prostředí.

Například během aktualizace **gcc** je aktualizován i soubor /etc/env.d/05gcc bez nutnosti zásahu uživatele.

To není výhoda pouze pro Portage, ale i pro vás jako uživatele. V některých případech můžete být požádáni o globální nastavení některé proměnné. Jako příklad uvedeme proměnnou `http_proxy`. Namísto abychom se mazali se souborem /etc/profile, můžete jednoduše vytvořit soubor (/etc/env.d/99local) a vložit definici/e do něj:

Obrázek 18.3. /etc/env.d/99local

```
http_proxy="proxy.server.com:8080"
```

Pokud budete pro všechny svoje proměnné používat stejný soubor, budete mít stále přehled o tom jaké proměnné jste nadefinovali sami.

18.2.2. Skript env-update

Několik souborů v /etc/env.d definuje proměnnou `PATH`. To není chyba - když spustíte **env-update**, spojí těchto několik definic do jedné než aktualizuje proměnné prostředí, takže je jednoduché pro balíčky (nebo uživatele) přidávat svoje vlastní proměnné prostředí aniž by se dostali do sporu s již existujícími hodnotami.

Skript **env-update** spojí hodnoty v abecedním pořadí souborů /etc/env.d. To je důvod proč spousta souborů v /etc/env.d začíná číslem.

Obrázek 18.4. Pořadí aktualizace používané skriptem env-update

00basic

99kde-env

99local

```
+-----+-----+-----+
PATH="/bin:/usr/bin:/usr/kde/3.2/bin:/usr/local/bin"
```

Až spustíte **env-update**, vytvoří tento skript všechny proměnné prostředí a umístí je do souboru `/etc/profile.env` (který je používán souborem `/etc/profile`). Vytáhne také informace pro proměnnou `LDPATH` a umístí je do souboru `/etc/ld.so.conf`. Poté spustí **ldconfig** a znovu vytvoří soubor `/etc/ld.so.cache` používaný dynamickým linkerem.

Pokud chcete se efekt úpravy proměnných projevit okamžitě po spuštění **env-update**, spusťte následující příkaz který aktualizuje vaše prostředí. Uživatelé kteří si Gentoo instalovali sami si tento příkaz patrně budou pamatovat z instalačních instrukcí:

Obrázek 18.5. Aktualizace prostředí

```
# env-update && source /etc/profilePATH="${PATH}:/home/my_user/bin"
```

18.3. Definice lokálních proměnných

Obrázek 18.6. Aktualizace proměnné PATH pro lokální použití v `~/.bashrc`

```
PATH="${PATH}:/home/my_user/bin"
```

18.3.1. Dle sezení (session)

Někdy je potřeba ještě omezenější definice proměnné prostředí. Můžete například potřebovat spouštět binární soubory z vámi vytvořeného dočasného adresáře a nechcete kvůli té chvilce upravovat soubory `~/.bashrc` nebo `~/.bash_profile`.

V tom případě můžete definovat proměnnou `PATH` v aktuálním sezení pomocí příkazu **export**. Dokud se neodhlásíte, proměnná prostředí bude obsahovat tuto dočasnou hodnotu.

Obrázek 18.7. Změna proměnné PATH pouze pro aktuální session

```
# export PATH="${PATH}:/home/my_user/tmp/usr/bin"
```