1 Úvod

Poslední dobou se čím dál častěji setkávám s lidmi, kteří by chtěli začít psát texty v opravdu dobrém programu. A již v první větě se dopouštím chyby, protože následující text a další články nebudou o editoru, ale sázecím systému LaTeX. Těmto začínajícím uživatelům nabídnu sérii článků ve stylu HOWTO. Dnes nás čeká úvodní kapitola obsahující základní informace, odkazy na zdroje a také něco málo z instalace.

1.1 Na začátek pár poznámek

Je docela možné, že se v těchto článcích objeví několik drobných chyb či nepřesností. Avšak články nemají za úkol vysvětlit teorii a terminologii, jejích úkolem je naučit začátečníky běžným konstrukcím v LaTeXu, jako je sazba tabulek, obrázků, seznamů, ale také je seznámit se základní strukturou dokumentu a rozšiřujícími moduly.

V článcích se budeme věnovat i některým problémům, které jsou specifické pro českou sazbu, a problémům, s kterými jsem se trápil já nebo o kterých mi napíšete.

Snad poslední poznámka bude věnována tomu, že nejsem odborník na LaTeXani TeX (tedy tzv. TeXpert) a ani odborník na sazbu, jsem prostě obyčejný uživatel LaTeXu a chtěl bych vás seznámit s tímto skvělým a jedinečným sázecím programem. Z tohoto pohledu pochopte, že se v některých věcech mohu mýlit (autor se nechává slyšet, že bude dělat chyby :), případně zapisovat některé konstrukce neefektivně, a to z důvodu toho, že to lépe neumím nebo že by to kolidovalo se snahou pedagogicky působit na stránku názornosti (a tímto způsobem se autor bude snažit chybu zamaskovat :).

1.2 Informační zdroje

Ve světě se vyznavači TeXu sdružují do skupin - TeX Users Group (TUG). Pro naši a také Slovenskou republiku existuje společné sdružení uživatelů TeXu - Československé sdružení uživatelů TeXu (CSTUG). Na jejich stránkách `www.cstug.cz` najdete odkazy na mnoho velmi zajímavých stránek a dalších informačních zdrojů. Velmi užitečné jsou také diskusní a newsové skupiny.

Software týkající se nějakým způsobem TeXu je uchováván pohromadě v CTANu (Comprehensive TeX Archive Network) - je to obdobné jako s distribucemi Linuxu (někde někdo udržuje soubory pohromadě a ostatní z něj dělají zrcadla). České zrcadlo CTANu naleznete na `http://www.cstug.cz/`.

A ještě se podíváme po velmi pěkných dokumentech, které mi v mých počátcích, ale i nyní velmi pomáhají. Mezi nejstěžejnější pro začátečníky považuji publikaci *Ne příliš stručný úvod do systému LaTeX2e* s podtitulem *Neboli LaTeX2e v 73 minutách.* Dokument lze nalézt na stránkách Michala Kočera, který spolu s Pavlem Sýkorou tento dokument přeložili. Za tento počin bych jim chtěl významně poděkovat. Další základní dokumenty (ale již v angličtině) se instalují se systémem TeX (např. pro Debian i Red Hat je to balík tetex-doc).

Ještě bych chtěl upozornit na dva články, které vyšly již před nějakou dobou. V prvním článku je uvedena trocha teorie a ukázka programu LyX, ve druhém článku se pak můžete pokochat krásou dokumentu vytvořeného v LaTeXu. Doporučuji všem, kteří neví, o čem byla doposud řeč.

1.3 Instalace

Pro instalaci budeme potřebovat několik balíků. Ty, které se týkají TeXovských věcí, začínají na tetex. Pokračování jména balíčku pak určuje jeho obsah; počet a pojmenování balíčků záleží na distribuci Linuxu. Systému vůbec neublížíte (a budete bez starostí), pokud nainstalujete všechny balíčky, které začínají právě slovem tetex.

Dalšími užitečnými programy jsou *ghostscript* a *ghostview*, jsou většinou uloženy v balíčcích pojmenovaných `gs` a `gv`, balík gs bude pravděpodobně vyžadovat ještě další balíčky, ve kterých jsou uloženy postscriptové fonty (některé distribuce fonty obsahují již ve jmenovaných balíčcích, jiné je mají odděleny).

O instalaci a nastavení dalších věcí, které lze při práci s TEXem využít, si řekneme, až je budeme potřebovat (ono stejně skoro nic nezbylo).

1.4 Test

Vyzkoušíme nainstalovaný software, zda nám přeloží následující minimální dokument:

```
\documentclass{article}
\begin{document}
  Dokument.
\end{document}
```

Při překladu je TEX neuvěřitelně ukecaný (nepochází totiž z Unixu, kde se programy řídí heslem *pokud je vše v pořádku, není důvod otravovat uživatele* - a to ani hlášením o provedení a správnosti úkolu). Při použití příkazu `latex test.tex` se nejen vypíše hodně hlášení, ale ještě víc se jich uloží do souboru s příponou `.log`, také se vytvoří soubor s příponou `.aux`, který nás zatím nebude zajímat. Posledním a pro nás zajímavým vygenerovaným souborem je `test.dvi`. Tento formát je vytvořen tak, aby byl nezávislý na výstupním zařízení (ještě aby nebyl, když to má v názvu - DeVice Independent). Výstup už si můžeme prohlédnout programem `xdvi` (dodává se jako součást balíků tetex).

Většinou chceme daný text vytisknout. Na systémech unixového typu se pro tisk užívá často formát PS (PostScript). Konverzi z DVI na PS nám zařídí příkaz `dvips test.dvi -o test.ps` (zadání výstupního souboru je nutností, jinak program posílá data přímo na tiskárnu). Program opět generuje řadu informací o probíhajícím převodu.

Vygenerovaný PS soubor můžeme zkusit prohlédnout třeba programem gv a pokud uvidíte vpravo nahoře text `Dokument.`, je vše v pořádku. Abyste mohli srovnat, jsou zde mé výsledky našeho testovacího souboru: `test.dvi` a `test.ps`.

A co dělat, pokud se to nepovedlo? Tak to je problém, který vám nepomohu vyřešit. Takto na dálku snad jen radu: pokusit se programy přeinstalovat nebo se zeptat v diskusi, já nebo někdo jiný vám poradí.

1.5 Závěr

V příštím dílu se koukneme na balíčky, základní strukturu LATEXovského dokumentu, a také si to celé vyzkoušíme trochu víc naostro. Prostě budeme pomaličku přitvrzovat.

2 Pronikáme hlouběji

V dnešním dílu se podíváme na opravdový základ dokumentu napsaný v LATEXu. Vysvětlíme si, co je to balíček, také si ukážeme jednoduchý příklad. A nebude chybět ani nakouknutí do teorie :)

Ano, správně. Trocha teorie nikomu neuškodí, a tak se objeví v každém článku. Povětšinou nebude moc souviset s probíraným tématem, ale myslím si, že se bude občas někomu hodit.

2.1 Teorie: TEXvs. LATEX

TEXje vlastně sada příkazů, kterými se formátuje dokument. Tyto příkazy už se nemění - TEXje zakonzervovaný. A právě to je největší výhodou psaní dokument v TEXu. Protože když dnes v TEXu napíšete nějaký dokument, přeložíte jej se stejným výsledkem i za 20 let na úplně jiných strojích s jiným operačním systémem. Přeložíte jej stejně jak na počítači doma, tak v práci, ale třeba také v Tramtárii.

LATEXje sada maker (tedy nové příkazy vytvořené z TEXovských příkazů). Tato makra byla napsána tak, aby ulehčovala práci a zároveň dodržovala zásady správného vzhledu dokumentu (typografické zásady). V LATEXu nemusíte přemýšlet nad některými věcmi, o nichž byste museli přemýšlet při psaní v TEXu (je to tedy zjednodušení práce). Na druhou stranu se LATEXstále vyvíjí a jeho sada příkazů se může změnit. Zde tedy již neplatí to, že dokument přeložíte znovu do stejné podoby i za 20 let. Avšak s drobnými úpravami by se to jistě povedlo. Stále na tom budete nesrovnatelně lépe (jak ve výsledné kvalitě, tak s počtem ztracených nervových buněk) než při psaní dokumentů v jiných programech, kde je už problém při přenášení dokumentu z jednoho počítače na druhý.

2.2 Struktura dokumentu

Dokument v LaTeXu se skládá ze dvou částí. První část (nazvěme si ji hlavičkou) obsahuje základní nastavení celého dokumentu. V druhé části pak už budeme psát vlastní text. Než to ale uděláme, povíme si něco o příkazech.

Všechny příkazy začínají znakem \ (zpětné lomítko), za nímž následuje jméno příkazu. Pokud příkaz umožňuje zadání parametrů, píší se do bloků uvozených hranatými závorkami parametry nepovinné (oddělují se čárkou) a do složených závorek se vpisují parametry povinné. Posloupnost bloků nepovinných a povinných parametrů musí mít právě toto pořadí (v dřívějších verzích byl možný i opačný zápis).

Prvním příkazem v dokumentu je informace o stylu (třídě) psaného dokumentu (do verze 2.09 včetně se používal příkaz `\documentstyle` (nyní to je `\documentclass`), odtud tedy ten *styl* dokumentu, který se mi zdá pro naše území přijatelnější než *třída* dokumentu):

```
\documentclass{article}
```

Tímto úvodním příkazem jsme definovali, že budeme psát článek (*article*). LaTeX si tím nastaví hodnoty pro správný tisk dokumentu. Tento styl se používá nejčastěji. Lze také nastavit nepovinné parametry, např.:

```
\documentclass[11pt,titlepage]{article}
```

Tím byly uvedeny dva nepovinné parametry (velikost základního písma (*11pt*) a požadavek, že budeme chtít vygenerovat úvodní stránku samostatně (*titlepage*)).

Za tímto úvodním příkazem většinou následují příkazy pro zavedení balíčků. Jednotlivé balíčky rozšiřují možnosti LaTeXu o různá prostředí (jako je například zápis algoritmů) nebo o zvláštní fonty, případně dělají úplně něco jiného.
Pro zavedení balíčku se používá příkaz `\usepackage` s tím, že jako povinný parametr se uvádí jméno balíčku, který chceme zavést, jako nepovinný parametr pak nějaká bližší specifikace. Tedy například:

```
\usepackage[latin2]{inputenc}
\usepackage{czech}
```

Oba příkazy dělají to, co si pravděpodobně myslíte. První říká LaTeXu, že tento dokument bude mít vstupní kódování (*inputenc*) dle latin2. Díky tomuto příkazu bude LaTeXvědět, co dělat s diakritickými písmenky. Druhý příkaz je bez nepovinného parametru a oznamuje, že dokument je psaný v češtině (k tomu se také potom přihlédne při formátování dokumentu).

Pak již následují další příkazy, o nich však již nelze říci nic konkrétnějšího. Vlastní text je uvozen příkazem `\begin{document}` a ukončen `\end{document}`. Mezi tyto dva příkazy se vepíše náš text, který může být formátován příkazy. Vše za příkazem `\end{document}` se ignoruje.

Držme se pravidla, že názornost je nejlepší, proto následuje celkový příklad:

```
\documentclass[11pt]{article}
\usepackage[latin2]{inputenc}
\usepackage{czech}
\usepackage{a4wide}

\begin{document}
  Můj už druhý dokument v \LaTeX u,
 tentokrát s češtinou.
\end{document}
```

V textu jsme použili již první formátovací příkaz. Byl jím `\LaTeX`, který místo sebe dosadí správně zapsané logo LaTeXu. Všimněte si, že za příkazem `\LaTeX` byla mezera automaticky ignorována (příkaz by bylo možné zapsat také bez mezery, ale aby překladač poznal, kdy končí příkaz a začíná text, je nutné uvést prázdný blok povinných parametrů, tedy takto:
`...v \LaTeX{}u, tentokrát...`).

2.3 Styly

LaTeXmá předdefinováno několik základních stylů (tříd):

article je používán nejčastěji a určen především pro odborné články, krátké zprávy, seminární práce, dokumentace k programům, ...

report jeho použití není tak běžné, jelikož se používá až pro rozsáhlejší díla, která obsahují několik kapitol, mohou to být menší knihy nebo diplomové práce.

book z názvu je zřejmé, že právě toto je určeno pro psaní skutečných (plnokrevných) knih.

slide určeno pro blány používané při prezentacích na projektoru, pro tisk se použije velké bezpatkové písmo.

letter a poslední styl je určen pro psaní dopisů

V příkazu `\documentclass` lze použít tyto nepovinné parametry:

10pt, 11pt, 12pt určí výšku základního písma pro dokument. Výška je uvedena v tiskařských bodech. Pokud není uveden parametr, je implicitně použita velikost 10 bodů.

a4paper, ... udává velikost stránky, na kterou se bude provádět sazba dokumentu. Implicitní nastavení je letterpaper, lze ale použít některý z následujících: a5paper, b5paper, executivepaper, legalpaper. Narozdíl od američanů jsou evropané šetrní k lesům, a tak používají širší A4, proto se při sazbě u nás nepoužívá nastavení a4paper, ale použije se balíček a4wide (tedy příkaz \usepackage{a4wide}). Ten nastaví nové rozměry dokumentu, čímž se na stránku vleze více textu (toto nastavení se mi více líbí; nevypadá to, jako by autor neměl co říci a tak zúžil okraje, to je ovšem věc vkusu a američani jej asi mají jiný).

fleqn bude zarovnávat matematické rovnice na levou stranu, standardně jsou zarovnány na střed.

leqno a ještě jednou matematické rovnice, tentokrát se číslování rovnic bude sázet na levou stranu stránky (standardně to je vpravo).

titlepage, notitlepage Určí, zda titulní stranu sázet samostatně, či nikoliv. Pro styl article se normálně samostatná titulní stránka nepoužívá. Opakem jsou styly report a book, kde je primárně nastavena samostatná stránka.

twocolumn máte chuť na sazbu do dvou sloupců?

twoside, oneside jedná se o nastavení, které rozhoduje, je-li výstup určen pro dvoustranný, nebo jednostranný tisk. Sazba jednostranná se od dvoustranné liší v různých velikostech okraje pravé a levé strany pro vazbu (ovlivňuje také sazbu čísla stránky, případně záhlaví apod.). Pro styl article a report se používá jednostranná sazba, pro book pak oboustranná.

openright, openany Pokud styl umožňuje vytvářet kapitoly (article to neumí), volba openright zajistí, že nová kapitola bude začínat na první pravé stránce. Volba openany nechá začátek kapitoly na nejbližší volné stránce. Styl report má implicitně nastaveno openany, styl book pak používá openright.

2.4 Balíčky

V přehledu uvedu některé balíčky, které sám používám. V některých dalších dílech se na ně koukneme detailněji, zde uvedený výčet je uveden pro úplnost tohoto článku (a také aby byla vidět všestrannost):

inputenc specifikuje se kódování vstupního souboru

czech nastavuje zásady typografie pro českou sazbu

a4wide nastaví velikost A4 dle evropských (tedy i našich) rozměrů

graphicx pro vkládání obrázků

algorithmic lze jednoduše zapisovat algoritmy

bbm speciální dvojité fonty pro matematiku

psfrag prohledává vkládané PostScripty a překládá v nich texovské výrazy

makeidx jednoduchým způsobem umožňuje vytvářet rejstřík

... a mnoho dalších :)

2.5 Závěr

Omlouvám se, že je to dneska ještě takové faktografické a nedostali jsme se prakticky k ničemu, na druhou stranu nemáme kam spěchat :). V dalším dílu se podíváme, kde všude je potřeba nastavit TEX, aby správně podporoval češtinu, a vyzkoušíme si něco dalšího. Ti, kteří nemají správně nakonfigurovánu českou klávesnici a alespoň jeden editor, si nemusí zoufat, protože se této problematice budu věnovat ve zvláštním článku.

Pro slovenské uživatele: omlouvám se, že články jsou psány čistě pro češtinu, avšak v konfiguračních souborech je většinou sloveské nastavení hned vedle českého, a tak myslím nebude problém zakomentovat/odkomentovat řádek těsně vedle. A změna příkazu `\usepackage{czech}` na `\usepackage{slovak}` vás určitě napadla :-)

3 Struktura dokumentu

Každý dokument má nějakou strukturu. A dnes se protáhneme právě pojmy jako je kapitola, podkapitola či odstavec.

3.1 Teorie

Ačkoliv se překladač při formátování textu všemožně snaží, aby byl text co nejpěknější, stane se někdy, že se vše nepodaří tak, jak má. Dnes si řekneme o chybách nazývaných *vdova* a *sirotek*. I když zde budu mluvit pouze o případu při přechodu na další stránku, jedná se též o problém vícesloupcové sazby, kde přechodem na další stránku rozumíme také ukončení jednoho sloupce a přechod na nový sloupec na téže stránce.

Vdovou nazýváme první řádek nového odstavce, který zůstal na předchozí stránce. Tato chyba je zvláště neomluvitelná, pokud se to stane u titulku (tedy nadpis na předchozí stránce a text na další). Sirotek je chybou obdobnou, ale tentokrát poslední řádek odstavce přeteče na další stránku, kde se stane řádkem prvním. Tato chyba má také neodpustitelnou verzi, a to v případě, že stránka je pak dále prázdná (tzn. že na stránce je samostatný řádek (nedej bože, pokud se řádek skládá pouze z jednoho slova)).

Obě chyby se v kvalitních pracích (zvláště jako je diplomová práce) nesmí vyskytnout. Nejen že jsou ošklivé a hrubě porušují pravidla správné sazby, ale hlavně roztrhnou text a při čtení pak člověk ztrácí kontinuitu a hlavní téma.

Pro odstranění takové chyby máme několik možností:

1. jsme-li autorem textu, máme možnost přeformulovat odstavec tak, aby vdova/sirotek nevznikl.
2. vhodně spojíme/rozdělíme odstavce.
3. jiným způsobem naformátujeme předchozí odstavce.
4. řekneme překladači, kde má/nemá přejít na další stránku (nařídíme nebo zakážeme stránkový zlom).
5. na stránce (v sloupci) změníme počet řádků (porušíme tím tzv. řádkový rejstřík), je to krajně nevhodné ve sloupcové sazbě (řádky jednotlivých sloupců "nesedí" v jedné linii) nebo v oboustranné sazbě (kniha), kde pak "ruší" čtenáře prosvítání nesouhlasných řádků z druhé stránky listu (prostě řádky si na sebe nesedly). I přesto je to jeden z nejpoužívanějších způsobů, nevhodný však pro knižní sazbu, kde způsobuje výše uvedený problém.
6. automaticky odstraníme - kvalitní programy dovolují definovat velikost sirotka/vdovy, a ty se pak program snaží při sazbě odstranit.

Na příkladu simulované vícesloupcové sazby si ukážeme, jak tyto chyby vypadají:

Ve vícesloupcové sazbě tyto chyby nevypadají až tak závažně, přesto je to hrubé porušení správné sazby dokumentu.

Například tato **vdova**
by se rozhodně v kvalitní práci neměla vyskytovat.

A zcela neomluvitelný případ je, pokud by byla vdova zároveň nadpisem. Dalším prohřeškem vůči zásadám správné sazby je
tento **sirotek**.

Jeho závažnost se zvyšuje s volným místem okolo něj. Proto by se nikdy neměl vyskytnout před nadpisem nebo na samostatné stránce.

3.2 Členění textu

Pro členění se používá několik úrovní, každou úroveň nelze použít v každém stylu. Proto nejlépe formou přehledu:

Příkaz (řídící slovo)	**Log. úroveň**	**Poznámka**
`\part`	část	nepovinné
`\chapter`	kapitola	pouze styl book a report
`\section`	oddíl	již pro všechny styly
`\subsection`	pododdíl	
`\subsubsection`	podpododdíl	
`\paragraph`	odstavec	standardně není zahrnováno do obsahu
`\subparagraph`	pododstavec	nemá příliš velké praktické použití, není číslován a tudíž ani zahrnován do obsahu

Příkazy mají dva parametry, do povinného se zapisuje řetězec, který se má vytisknout a případně očíslovat. Do nepovinného lze zapsat řetězec, který se má vytisknout v obsahu. To se může hodit, pokud máme v nadpisu například nějaký matematický výraz a byli bychom radši, kdyby byl v obsahu pojmenován jménem, nikoliv symbolem.

Příkazy mají ještě svoji "hvězdičkovou" obdobu. Pokud za jméno příkazu (řídícího slova) uvedeme * (hvězdičku), nebude daný nadpis číslován a nebude zahrnut ani do obsahu. Příkazy, které standardně nejsou číslovány (`\paragraph` a `\subparagraph`) se v hvězdičkované verzi chovají úplně stejně jako bez hvězdičky.

Lze použít ještě jedno členění textu, které zde nebylo zmíněno. Jedná se o jeden nebo více prázdných řádků, které text dělí na "klasické" odstavce, jež nemají nadpisek, ale začátek je o něco málo odražen. Tento typ odstavce se nezahrnuje do obsahu ani nemá svůj vlastní čítač, pouze začátek dalšího textu poodstrčí od okraje.

Specialitkou je příkaz `\appendix`, kterým označujeme začátek příloh. Příkaz nastaví jiný způsob číslování a provede znovu nastavení čítačů na počáteční hodnoty.

3.3 Obsah

Důležitou součástí dokumentu je i obsah. Ten se generuje příkazem `\tableofcontents`. Při používání tohoto příkazu je nutné provést překlad dokumentu programem latex minimálně dvakrát.

To je způsobeno tím, že pro vytvoření obsahu je využíván soubor s příponou `.toc` (svou podstatou je to vygenerovaný zdrojový soubor pro LaTeX, který se vloží na místo příkazu). Tento soubor se ale vytváří teprve při překladu, a kompletní je až po jeho ukončení. Z tohoto důvodu je potřeba překlad spustit poprvé pouze za účelem vytvoření souboru s uloženým obsahem. A při druhém překladu se obsah vysází podle kompletního souboru. Nyní to vypadá, že takový způsob je velmi neohrabaný, ale má to jednu nespornou výhodu. Mezi oběma překlady lze soubor s obsahem editovat, a tak tam provést některé zajímavé (a třeba potřebné) úpravy. A pokud se překlad provádí na pomalejším stroji, není problém řádek s `\tableofcontents` zakomentovat (komentář se provádí znakem % (procento), vše za ním na řádku se ignoruje) a před finálním "dvojitým" překladem jej stačí odkomentovat.

Jedno drobné doporučení: protože se obdobným způsobem řeší více problémů při sazbě (křížové odkazy, rejstřík, ...), doporučuje se provést závěrečný překlad třikrát.

3.4 Čítače

Této problematice bude věnován zvláštní článek, ale nakousnout to můžeme již zde. A tak pro úplnost a dokreslení této kapitoly se dozvíme něco málo o čítačích (čítač je proměnná uchovávající číslo). Při členění textu do logických částí se používají tyto čítače:

Název čítače	**Popis**
`secnumdepth`	standardně se čísluje do třetí úrovně (`\subsubsection`) čítač lze nastavit v rozmezí $1-4$, kde 1 je pro `\section` a 4 pro `\paragraph`.
`tocdepth`	tento čítač patří až k vytváření obsahu, ale určuje téměř to samé, co předchozí čítač – určuje, do jaké hloubky se má vypisovat obsah příkazem `\tableofcontents`.
`part, chapter, ...`	až k `subparagraph`. V těchto čítačích se ukládají hodnoty pro očíslování jednotlivých úrovní. S každým příkazem dané logické úrovně se k patřičnému čítači přičte jednička. Jediný užitek změny těchto čítačů vidím pouze v případě, že dokument někam vkládáme (nezačínáme číslovat od jedničky) nebo do výsledného dokumentu něco vkládat budeme.

Hodnotu čítače lze nastavit příkazem `\setcounter` (zde se setkáváme poprvé s příkazem, který má více než jeden povinný parametr), získat hodnotu čítače lze příkazem `the{jméno_čítače}` (lze použít pouze pro čítače logických úrovní, nelze proto užít pro čítače secnumdepth nebo tocdepth). Tento příkaz pro zobrazení čítače formátuje uložené číslo. Z následujícího příkladu bude použití příkazů patrné:

```
\setcounter{section}{3}  % nastaví čítač na 3
\thesection              % vytiskne 3
```

3.5 Závěr

V příštím dílu si už na méně rozsáhlém příkladě vyzkoušíme formátování (zvýrazňování) textu a možná ještě něco navíc.

Rejstřík

\, 3

balík, 1, 3, *5*
  czech, 3
  inputenc, 3, *3*

Comprehensive TEXArchive Network, 1
CSTUG, 1
CTAN, 1
czech, *viz* balík

Debian, 1
DeVice Independent, *viz* DVI
distribuce
  linuxu, 1
dokument, 2, 3
  hlavička, 3
  minimální, 2
  struktura, 1, *3–4*, 5
  styl, 3, *4*
    article, 4
    book, 4
    letter, 4
    report, 4
    slide, 4
  třída, 3
DVI, *2*
dvips, 2
dvoustranná sazba, 4

ghostscript, 1
ghostview, 1, 2
gs, *viz* ghostscript
gv, *viz* ghostview

hlavička, *viz* dokument

inputenc, *viz* balík
instalace, 1

jednostranná sazba, 4

kapitola, 4–6
komentář, 7

LATEX, 1, *2–3*
latex, 2
latin2, 3
Linux, 1
LyX, 1

makro, 2

nadpis, 6
nepovinný parametr, *viz* parametr

obrázek, 1
obsah, 6, 7
oddíl, 6
odstavec, 5, 6
operační systém, 2

parametr, 3
  nepovinný, *3*
  povinný, *3*
podkapitola, 5
pododdíl, 6
pododstavec, 6
podpododdíl, 6
postscript, 2
postscriptové fonty, 1
povinný parametr, *viz* parametr
program, 1
prostředí, 3
prázdný blok, 4
PS, *viz* postscript
překlad, 7
příkaz, 2, 3
  jméno, 3
  parametr, *viz* parametr
příloha, 7

Red Hat, 1
rejstřík
  řádkový, 6
rovnice, 4

sazba, 1
seznam, 1
sirotek, *5–6*
skupina
  diskusní, 1
  news, 1
  uživatelů TEXu, 1
sloupcová sazba, 6
soubor
  .aux, 2
  .dvi, 2
  .log, 2
  .ps, *viz* postscript
  .toc, 7
struktura
  dokumentu, *viz* dokument
styl, *viz* dokument

symbol, 6

tabulka, 1
tetex, 1
tetex-doc, 1
TeX, 1, 2, *2–3*
TeX Users Group, 1
tisk, 2
titulní strana, 4
TUG, 1
třída, *viz* dokument

Unix, 2

vdova, *5–6*
vícesloupcová sazba, 5, 6

xdvi, 2

základní písmo, 4

Československé sdružení uživatelů TeXu, 1

česká sazba, 1
článek, 1
část, 6
čítač, *7*
    `secnumdepth`, 7
    `tocdepth`, 7

řádkový rejstřík, *viz* rejstřík

Rejstřík použitých příkazů

%, 7

appendix, 7

begin
- document, 3

chapter, 6

documentclass, *3*, 4
- 10pt, 4
- 11pt, 3, 4
- 12pt, 4
- a4paper, 4
- a5paper, 4
- article, 3, 4
- b5paper, 4
- book, 4
- executivepaper, 4
- fleqn, 4
- leqno, 4
- leter, 4
- letterpaper, 4
- notitlepage, 4
- oneside, 4
- openany, 4
- openright, 4
- report, 4
- slide, 4
- titlepage, 3, 4
- twocolumn, 4
- twoside, 4

documentstyle, *3*

end
- document, 3

LaTeX, 4
legalpaper, 4

paragraph, 6
part, 6

rovnice, 4

section, 6
setcounter, 7
subparagraph, 6
subsection, 6
subsubsection, 6

tableofcontents, 7

the, 7

usepackage, 3
- a4wide, 4, 5
- algorithmic, 5
- bbm, 5
- czech, 3, 5
- graphicx, 5
- inputenc, 3, 5
- latin2, 3
- makeidx, 5
- psfrag, 5
- slovak, 5