Nie príliš stručný úvod do systému $\LaTeX\,2_\varepsilon$

Alebo $\LaTeX\,2_\varepsilon$ za 93 minút

Tobias Oetiker

Hubert Partl, Irene Hyna a Elisabeth Schlegl

verzia 3.13, 23. február 2000

Preklad **Ján Buša ml. a st.**

posledná úprava 17. marca 2002

Vďaka!

Väčšina materiálu použitého v tejto príručke pochádza z rakúskeho *Úvodu do LATEXu 2.09*, napísaného po nemecky týmito autormi:

Hubert Partl `<partl@mail.boku.ac.at>`
Zentraler Informatikdienst der Universität für Bodenkultur Wien

Irene Hyna `<Irene.Hyna@bmwf.ac.at>`
Bundesministerium für Wissenschaft und Forschung Wien

Elisabeth Schlegl `<no email>`
in Graz

Ak máte záujem o nemecký variant tohoto manuálu, jeho najnovšiu LATEX 2_ε verziu napísanú Jörgom Knappenom môžete nájsť na adrese
`CTAN:/tex-archive/info/lshort/german`

Počas prípravy tohoto dokumentu som požiadal o zaslanie recenzie na `comp.text.tex`. Dostal som množstvo ohlasov. Nasledujúci jednotlivci mi pomohli opravami, návrhmi a poskytli mi materiál na zlepšenie tejto práce. Venovali veľa úsilia, aby mi pomohli dostať tento dokument do jeho terajšej podoby. Rád by som im úprimne poďakoval. Prirodzene, všetky chyby, ktoré nájdete v tejto knižke, sú moje. Ak nájdete slovo, ktoré je vyslabikované správne, musel ma na to upozorniť písomne niekto z nižšie uvedených ľudí.

Rosemary Bailey, Friedemann Brauer, Jan Busa, Markus Brühwiler,
David Carlisle, José Carlos Santos, Mike Chapman,
Christopher Chin, Carl Cerecke, Chris McCormack, Wim van Dam,
Jan Dittberner, Michael John Downes, David Dureisseix, Elliot,
David Frey, Robin Fairbairns, Jörg— Fischer, Erik Frisk, Frank,
Kasper B. Graversen, Alexandre Guimond, Cyril Goutte,
Greg Gamble, Neil Hammond, Rasmus Borup Hansen,
Joseph Hilferty, Björn Hvittfeldt, Martien Hulsen, Werner Icking,
Jakob, Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones,
Johannes-Maria Kaltenbach, Michael Koundouros, Andrzej Kawalec,
Alain Kessi, Christian Kern, Jörg Knappen, Kjetil Kjernsmo,
Maik Lehradt, Alexander Mai, Martin Maechler,
Aleksandar S Milosevic, Claus Malten, Kevin Van Maren,
Lenimar Nunes de Andrade, Hubert Partl, John Refling,

Mike Ressler, Brian Ripley, Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Hanspeter Schmid, Craig Schlenter, Christopher Sawtell, Geoffrey Swindale, Josef Tkadlec, Didier Verna, Fabian Wernli, Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zaccone, and Mikhail Zotov.[1]

[1]Prekladatelia ďakujú najmä L. Ševčovičovi a M. Takáčovi, ktorí upozornili na niekoľko (stoviek) chýb a nepresností prekladu.

Úvod

LaTeX [1] je typografický systém, ktorý je veľmi vhodný na výrobu vedeckých a matematických dokumentov vysokej typografickej kvality. Je tiež vhodný na prípravu všetkých druhov iných dokumentov, od jednoduchých listov až po celé knihy. LaTeX používa TeX [2] ako formátovací jazyk.

Tento krátky úvod opisuje LaTeX 2_ε a mal by byť dostatočným pre väčšinu aplikácií LaTeXu. Podrobnejšie sa o LaTeXu môžete dočítať v [1, 3].

LaTeX je dostupný pre väčšinu počítačov od PC a Mac-ov až po veľké systémy UNIX a VMS. Na mnohých univerzitných sieťach je už inštalovaný a pripravený na prácu. Informácie o prístupe k lokálnej inštalácii LaTeXu by mali byť uvedené v *Local Guide* [4]. Ak máte problémy so spustením, obráťte sa na osobu, ktorá vám poskytla túto brožúru. Cieľom tohoto dokumentu *nie je* povedať vám ako nainštalovať a nastaviť systém LaTeX, ale naučiť vás, ako písať dokumenty tak, aby mohli byť spracované LaTeXom.

Tento úvod je rozdelený do piatich kapitol:

Kapitola 1 sa venuje základnej štruktúre dokumentov, písaných v LaTeXu. Dozviete sa tiež niečo z histórie LaTeXu. Po prečítaní tejto kapitoly by ste mali mať približný obraz o LaTeXu. Bude to zatiaľ len rám obrazu, ale umožní vám to poskladať informácie z ďalších kapitol do veľkého obrazu.

Kapitola 2 rozvádza do podrobností písanie vášho dokumentu. Vysvetľuje najdôležitejšie LaTeXové príkazy a prostredia. Po prečítaní tejto kapitoly budete schopní napísať váš prvý dokument.

Kapitola 3 vysvetľuje ako písať v LaTeXu vzorce. Množstvo príkladov vám znova pomôže pochopiť jednu zo silných stránok LaTeXu. Na konci tejto kapitoly nájdete tabuľky, obsahujúce zoznamy všetkých matematických symbolov prístupných v LaTeXu.

Kapitola 4 vysvetľuje vytváranie indexu a zoznamu literatúry, vkladanie `eps` grafiky[2] do vášho dokumentu a niektoré ďalšie užitočné rozšírenia.

[2] Obrázkov vo formáte ENCAPSULATED POSTSCRIPT (pozn. prekl.).

Kapitola 5 obsahuje určitú potenciálne nebezpečnú informáciu o tom ako dosiahnuť alternatívnu úpravu voči štandardnej úprave, ktorú vytvára LaTeX. Dozviete sa z nej ako zmeniť veci tak, že krásny výstup z LaTeXu začne vyzerať dosť zle.

Je dôležité, aby ste čítali kapitoly po poradí. Okrem toho táto kniha nie je až taká rozsiahla. Pozorne si prečítajte príklady, pretože veľká časť informácií je obsiahnutá práve v týchto príkladoch, ktoré nájdete v celej knihe.

Ak potrebujete akýkoľvek materiál súvisiaci s LaTeXom, pozrite sa do niektorého z `CTAN` ftp archívov. Pre USA je to `ftp.shsu.edu`, pre Nemecko `ftp.dante.de`, pre Veľkú Britániu `ftp.tex.ac.uk` alebo `ftp.muni.cz` u nás. Ak nie ste v niektorej z týchto krajín, zvoľte si archív, ktorý je ku vám najbližšie.

Ak chcete spustiť LaTeX na vašom vlastnom počítači, pozrite sa, čo sa dá získať z `CTAN:/tex-archive/systems`.

Ak vás napadne niečo, čo by mohlo byť pridané, vypustené, alebo zmenené v tomto dokumente, prosím, oznámte mi to. Názory LaTeXových začiatočníkov ma zaujímajú najviac – čo z tohoto úvodu (do LaTeXu) ľahko pochopili a čo by podľa nich mohlo byť vysvetlené lepšie.

Tobias Oetiker `<oetiker@ee.ethz.ch>`
Department of Electrical Engineering,
Swiss Federal Institute of Technology

Aktuálna verzia originálu tohoto dokumentu je prístupná na
`CTAN:/tex-archive/info/lshort`

Ján Buša st. `<busaj@tuke.sk>`
Katedra matematiky FEI,
Technická univerzita v Košiciach

Aktuálna verzia tohoto dokumentu je prístupná na
`www.cstug.sk`

Obsah

Zoznam obrázkov

Zoznam tabuliek

Kapitola 1

Veci, ktoré potrebujete vedieť

V prvej časti tejto kapitoly sa dozviete trochu o filozofii a histórii LaTeXu 2_ε. Druhá časť kapitoly sa venuje základnej štruktúre LaTeXového dokumentu. Po prečítaní tejto kapitoly by ste mali mať hrubú predstavu o tom, ako pracuje LaTeX. Prečítanie tejto kapitoly vám pomôže poskladať všetky ďalšie informácie do veľkého obrazu.

1.1 The Name of the Game

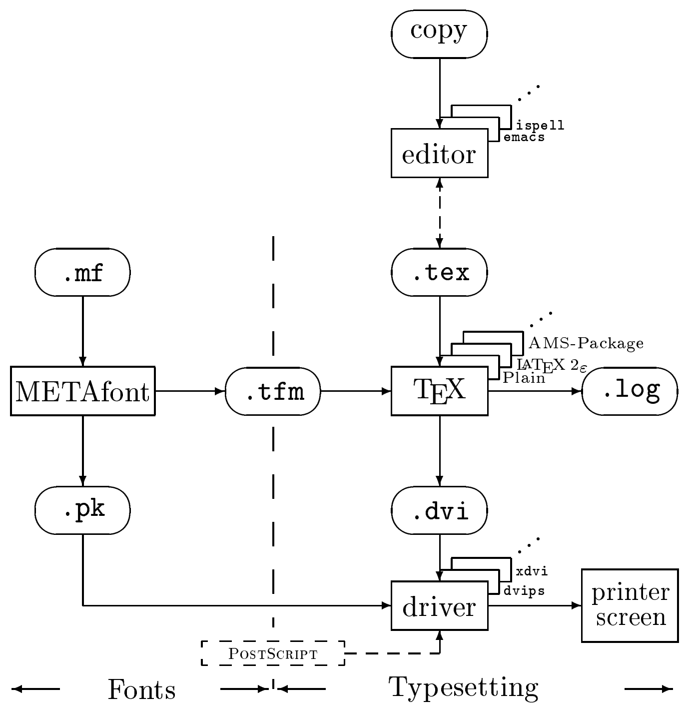
1.1.1 TeX

TeX je počítačový program, ktorého autorom je Donald E. Knuth [2]. Je určený na kvalitnú tlač textu a matematických vzorcov. Knuth začal písať typografický nástroj TeX v roku 1977 aby preskúmal potenciál digitálneho tlačového vybavenia, ktoré v tom čase začalo pomaly prenikať do typografického priemyslu, najmä v nádeji, že dokáže zvrátiť trend zhoršujúcej sa typografickej kvality postihujúci jeho vlastné knihy a články. TeX, ktorý používame dnes, bol uvoľnený v roku 1982 s niekoľkými drobnými rozšíreniami pridanými v roku 1989 na lepšiu podporu 8-bitových písmen a využitie mnohonásobných jazykov. TeX je slávny svojou extrémnou stabilitou, schopnosťou použitia na mnohých rôznych typoch počítačov a skutočnou bezchybnosťou. Číslo verzie konverguje k π a v súčasnosti je to 3.14159.

TeX sa číta ako „Tech“, s „ch“ ako v nemeckom slove „Ach“ alebo škótskom „Loch“. V prostredí `ASCII` sa TeX píše `TeX`.

1.1.2 LaTeX

LaTeX je balík makier, ktorý umožňuje autorovi písať a následne tlačiť svoje práce v najvyššej typografickej kvalite, používajúc preddefinované profe-



Obr. 1.1: Zložky systému TeX

sionálne rozmiestnenie obsahu dokumentu. Pôvodným autorom LaTeXu je Leslie Lamport [1]. Používa TeX ako typografický nástroj.

V roku 1994 bol balík LaTeX aktualizovaný LaTeX3 tímom vedeným Frankom Mittelbachom tak, aby obsahoval niektoré dlho požadované vylepšenia a aby sa zjednotili všetky záplatové verzie, ktoré vznikli od vytvorenia LaTeXu 2.09 pred niekoľkými rokmi. Nato, aby sa nová verzia dala odlíšiť od starej, bola pomenovaná LaTeX 2_ε. Táto publikácia sa zaoberá práve už spomínanou verziou LaTeX 2_ε.

LaTeX sa vyslovuje „Lay-tech“ alebo „Lah-tech“. LaTeX sa v prostredí `ASCII` píše `LaTeX`. LaTeX 2_ε sa vyslovuje „Lej-tech dva e“ a píše sa `LaTeX2e`.

Hore uvedený obrázok 1.1 ukazuje spoluprácu TeXu a LaTeXu2_ε. Tento obrázok je prevzatý z `wots.tex` od Kees van der Laana.

1.2 Základy

1.2.1 Autor, knižný dizajnér a sadzač

Keď chcú autori niečo publikovať, odnesú svoj rukopis do vydavateľstva. Knižný dizajnér vydavateľstva potom rozhodne o úprave dokumentu (o šírke stĺpcov, druhoch písma, medzerách pred a po nadpisoch, ...). Dizajnér napíše inštrukcie do rukopisu a dá ho sadzačovi, ktorý pripraví knihu podľa týchto inštrukcií.

Ľudský dizajnér sa snaží zistiť, čo si autor myslel počas písania rukopisu. Na základe svojich profesionálnych vedomostí a obsahu rukopisu rozhoduje o nadpisoch kapitol, citáciách, príkladoch, vzorcoch, atď.

V prostredí LaTeXu berie na seba úlohu dizajnéra LaTeX a používa TeX ako svojho sadzača. Ale LaTeX je tiež „len“ program a preto potrebuje ďalšie vedenie. Autor musí poskytnúť dodatočné informácie, ktoré opisujú štruktúru jeho práce. Tieto informácie sa vpisujú do textu ako „LaTeXové príkazy“.

Toto je trochu odlišné od prístupu WYSIWYG[1], ktorý využíva väčšina moderných procesorov ako *MS Word* alebo *Corel WordPerfect*. Pri týchto aplikáciách, autori určujú výstup interaktívne počas písania textu do počítača. Počas celej práce takto môžu vidieť na monitore, ako bude práca nakoniec vyzerať, keď bude vytlačená.

Pri LaTeXu zvyčajne nie je možné počas písania vidieť konečný výstup. Výsledný tvar textu sa však môže prezerať na monitore po spracovaní LaTeXom. Ďalšie úpravy sa môžu urobiť pred vlastnou tlačou dokumentu.

1.2.2 Výzor výstupu

Typografický dizajn je remeslo. Neskúsení autori sa často dopúšťajú vážnych chýb, mysliac si, že výzor knihy je hlavne vec estetiky-„Ak dokument vyzerá dobre po umeleckej stránke, potom je navrhnuý.“ Keďže však má dokument slúžiť na čítanie a nie visieť niekde v galérii, čitateľnosť a zrozumiteľnosť sú najdôležitejšie a až potom nasleduje pekný výzor. Príklady:

- veľkosť textu a číslovanie kapitol sa má vyberať tak, aby štruktúra kapitol a odsekov bola jasná pre čitateľa,
- riadky majú byť dostatočne krátke, aby nenamáhali zrak čitateľa a zároveň dosť dlhé, aby pekne zaplnili stranu.

Použitím systémov WYSIWYG autori často dosahujú esteticky zrak potešujúce dokumenty s nedostatočnou alebo nedôslednou štruktúrou. LaTeX predchádza takýmto chybám pri formátovaní, nútiac autora vysvetliť *logickú* štruktúru svojho dokumentu. LaTeX následne zvolí výstup, najviac zodpovedajúci požiadavkám autora.

[1] What you see is what you get. – Čo vidíš, to dostaneš.

1.2.3 Výhody a nevýhody

Keď ľudia z WYSIWYGového sveta stretnú ľudí, ktorí používajú LaTeX, najčastejšie sa diskutuje „o výhodách LaTeXu v porovnaní s normálnym textovým procesorom“ alebo naopak. Najlepšie urobíte, ak takýto rozhovor začne, keď budete v úzadí, keďže sa takáto diskusia často vymyká z rúk. Avšak niekedy niet úniku...

Teda tu je troška streliva. Najväčšie výhody LaTeXu oproti normálnym textovým procesorom sú nasledujúce:

- Sú dostupné profesionálne vytvorené úpravy stránky, takže dokument skutočne vyzerá, ako keby bol „vytlačený“.
- Písanie matematických vzorcov je zabezpečené pohodlným spôsobom.
- Užívateľovi sa stačí naučiť niekoľko ľahko pochopiteľných príkazov, definujúcich logickú štruktúru dokumentu. Takmer nikdy nemusí „fušovať“ do nastavenej úpravy dokumentu.
- Ľahko môžu byť vygenerované aj zložité štruktúry, ako poznámky pod čiarou, odkazy, obsah, použitá literatúra a register.
- Na riešenie mnohých typografických úloh priamo nepodporovaných základným LaTeXom existujú voľne pripojiteľné balíky makier. Napríklad môžete pomocou nich vložiť do textu PostScriptovú grafiku alebo vytvoriť bibliografie dodržujúc presné štandardy. Mnoho z týchto voľne pripojiteľných makier je opísaných v *The LaTeX Companion* [3].
- LaTeX povzbudzuje autorov písať texty s kvalitnou štruktúrou, pretože toto je spôsob, ako LaTeX pracuje — špecifikovaním štruktúry.
- TeX, formátovací motor LaTeXu2$_\varepsilon$, je bezplatný a ľahko prenositeľný. Preto beží takmer na každej dostupnej hardvérovej platforme.

LaTeX má aj určité nevýhody, ale myslím si, že je pre mňa trochu obtiažne nájsť nejakú rozumnú, hoci som si istý, že iní ľudia by vám mohli povedať o stovkách `;-)`

- LaTeX nefunguje dobre ľuďom, ktorí predali svoju dušu...
- Hoci pri preddefinovaných formátoch strán sa dajú upravovať niektoré parametre, vytvorenie nového formátu je náročné a zaberie veľa času[2].
- Je veľmi ťažké napísať neštruktúrovaný a chaotický dokument.

[2]Hovorí sa, že to bude jeden z kľúčových prvkov na ktoré sa upriami pozornosť pri pripravovanom systéme LaTeX3.

- Váš škrečok by nemusel byť nikdy schopný, navzdory niektorým povzbudzujúcim prvým krokom, úplne pochopiť pojem logického značkovania (Logical Markup).

1.3 Vstupné súbory LaTeXu

Zdrojový súbor pre LaTeX je jednoduchý súbor v `ASCII` formáte. Môžete ho vytvoriť ľubovoľným textovým editorom. Obsahuje text dokumentu rovnako ako aj príkazy, ktoré hovoria LaTeXu ako vysádzať text.

1.3.1 Medzery

Znaky „bielych miest“, ako medzera alebo tabulátor, sa v LaTeXu interpretujú zhodne ako medzera („space“). *Viacero za sebou nasledujúcich* medzier sa považuje za *jednu* medzeru. Voľné miesto na začiatku riadku sa vo všeobecnosti ignoruje a samotný koniec riadku sa považuje za jednu medzeru.

Prázdny riadok medzi dvoma riadkami textu definuje koniec odstavca. *Viacero* voľných riadkov sa berie rovnako ako *jeden* voľný riadok. Príkladom je nižšie uvedený text. Na ľavej strane vidíte zdrojový text a na pravej strane je preformátovaný výstup.

```
Nezáleží na tom, či
zadáte za slovom jednu,
alebo   viac     medzier.

Voľný riadok začína nový
odstavec.
```

Nezáleží na tom, či zadáte za slovom jednu, alebo viac medzier.

Voľný riadok začína nový odstavec.

1.3.2 Špeciálne znaky

Nasledujúce znaky sú rezervovanými symbolmi, ktoré buď majú pre LaTeX zvláštny význam, alebo nie sú dostupné vo všetkých fontoch. Ak ich napíšete priamo do textu, zvyčajne sa pri tlači nevytlačia, ale väčšinou donútia LaTeX robiť také veci, ktoré ste nemali v úmysle.

```
$ & % # _ { } ~ ^ \
```

Ako uvidíte, tieto symboly môžete vložiť do vášho dokumentu pridaním obráteného lomítka pred ne:

```
\$ \& \% \# \_ \{ \}
```

$ & % # _ { }

Zvyšné symboly (a mnoho ďalších) môžu byť vytlačené pomocou špeciálnych príkazov v matematickom prostredí alebo ako diakritické znamienka. Znak obráteného lomítka \ *nemôžeme* získať pridaním ďalšieho spätného lomítka pred neho (`\\`), táto postupnosť sa používa na zalamovanie riadkov.[3]

1.3.3 Príkazy LaTeXu

Príkazy LaTeXu rozlišujú veľké a malé písmená a majú jeden z nasledujúcich dvoch tvarov:

- Začínajú sa obráteným lomítkom `\` a potom pokračujú názvom zloženým len z písmen. Názvy príkazov sú zakončené medzerou, číslom alebo nejakým iným „nepísmenom“.
- Skladajú sa z obráteného lomítka a práve jedného znaku (číselného alebo špeciálneho).

LaTeX ignoruje medzery za príkazmi. Ak chcete dostať za príkazom medzeru, musíte za príkaz napísať `{}` a medzeru alebo použiť špeciálny príkaz po zadaní vášho príkazu. Zátvorky (`{}`) zabránia LaTeXu požierať všetky medzery za názvom príkazu.

```
Čítal som, že Knuth rozdeľuje
ľudí používajúcich \TeX{} na
\TeX{}nikov a \TeX pertov. \\
Dnes je \today.
```

Čítal som, že Knuth rozdeľuje ľudí používajúcich TeX na TeXnikov a TeXpertov.
Dnes je 17. marca 2002.

Niektoré príkazy vyžadujú parameter, ktorý sa vkladá do zložených zátvoriek `{ }` za názvom príkazu. Niektoré príkazy podporujú nepovinné parametre, ktoré sa pridávajú za názov príkazu do hranatých zátvoriek `[ ]`. Nasledujúci príklad používa niektoré LaTeXové príkazy. Netrápte sa pre ne, neskôr budú vysvetlené.

```
Môžete ma \textsl{nakloniť}!
```

Môžete ma *nakloniť*!

[3] Vyskúšajte namiesto toho príkaz `$\backslash$`. Vytvorí znak ‚\‘. Príkaz `\texttt{\char'134}` vytvorí znak \.

```
Začnite, prosím, nový riadok
práve tu!\newline
Ďakujem!

Začnite nový riadok
práve tu!\linebreak[3]
Ďakujem!

Začnite nový riadok
práve tu!\linebreak[4]
Ďakujem!
```

Začnite, prosím, nový riadok práve tu!
Ďakujem!

Začnite nový riadok práve tu! Ďakujem!

Začnite nový riadok práve tu!
Ďakujem!

1.3.4 Poznámky

Ak LaTeX narazí počas spracovávania vstupného súboru na znak percenta `%`, bude ignorovať zvyšok daného riadku, znak ukončenia riadku a všetky medzery na začiatku nasledujúceho riadku.

Toto sa dá použiť na písanie poznámok do vstupného súboru, ktoré sa neukážu vo vytlačenej verzii.

```
Toto je  % hlúpy
% Lepšie: poučný <----
príklad: super%
          predĺ%
      žené slovo
```

Toto je príklad: superpredĺžené slovo

Znak % môže byť tiež použitý na rozdelenie dlhých vstupných riadkov, v ktorých sú zakázané medzery, respektíve konce riadkov.

Na dlhšie poznámky musíte použiť prostredie `lscomment` poskytované balíkom verbatim.

```
Toto je ďalší
\begin{comment}
dosť hlúpy,
ale užitočný
\end{comment}
príklad začlenenia komentárov do vášho dokumentu.
```

Toto je ďalší príklad začlenenia komentárov do vášho dokumentu.

1.4 Štruktúra zdrojového súboru

Keď LaTeX 2_ε spracováva zdrojový súbor, očakáva, že bude mať určitú štuktúru. Teda každý zdrojový súbor musí začínať príkazom

```
\documentclass{...}
```

Toto špecifikuje, aký štýl dokumentu chcete napísať. Potom môžete zadať príkazy, ktoré majú ovplyvniť štýl celého dokumentu, alebo môžete natiahnuť balíky (makrá), ktoré pridajú do systému LaTeX nové vybavenie. Na natiahnutie takýchto balíkov použite príkaz

```
\usepackage{...}
```

Keď je všetko nastavené[4], začnete samotný text príkazom

```
\begin{document}
```

Ďalej napíšete text, pomiešaný s niekoľkými užitočnými LaTeXovými príkazmi. Na konci dokumentu pridáte príkaz

```
\end{document}
```

ktorý povie LaTeXu, že tu končí váš text. Všetko, čo nasleduje za týmto príkazom, bude LaTeX ignorovať.

Obrázok 1.2 ukazuje obsah minimálneho LaTeXového súboru, trochu zložitejší zdrojový súbor ukazuje obrázok 1.3.

```
\documentclass{article}
\usepackage{slovak}
\begin{document}
Malé je pekné.
\end{document}
```

Obr. 1.2: Minimálny LaTeXový súbor

1.5 Úprava dokumentu

1.5.1 Štýly dokumentov

Prvá informácia, ktorú potrebuje LaTeX počas formátovania súboru, je typ dokumentu, aký chcel autor vytvoriť. To definuje príkaz `\documentclass`:

`\documentclass[`*voľby*`]{`*trieda*`}`

Parameter *trieda* špecifikuje štýl (triedu) dokumentu, ktorý chceme vytvoriť. Tabuľka 1.1 uvádza štýly dokumentov, ktoré sú v tejto príručke obsiahnuté. V distribúcii LaTeX 2_ε sa stretávame aj s ďalšími triedami pre iné dokumenty,

[4]Priestor medzi `\documentclass` a `\begin{document}` sa nazýva *preambula*. Ak používate v CSTeXu diakritiku, musíte zadať príkaz `\usepackage{slovak}`, respektíve `\usepackage{czech}`.

vrátane listov a fólií. Parameter *voľby* upravuje základné nastavenia dokumentu. Voľby musia byť oddelené čiarkami. Najbežnejšie voľby (nastavenia) pre štandardné štýly dokumentu nájdete v tabuľke 1.2.

Príklad: Vstupný súbor pre LaTeX môže začínať riadkom

```
\documentclass[11pt,twoside,a4paper]{article}
```

Tento riadok prikazuje LaTeXu vysádzať dokument ako *článok* so základnou veľkosťou fontov *jedenásť bodov*, pričom výstup musí byť vhodný na *obojstrannú* tlač na papieri formátu *A4*.

```
\documentclass[a4paper,11pt]{article}
\usepackage{slovak,latexsym}
\author{H.~Partl}
\title{Minimalizmus}
\frenchspacing
\begin{document}
\maketitle
\tableofcontents
\section{Začiatok}
Nuž, a tu začína môj prekrásny článok.
\section{Koniec}
\ldots{} a tu končí.
\end{document}
```

Obr. 1.3: Príklad realistického časopiseckého článku

Tabuľka 1.1: Štýly (triedy) dokumentov

`article` na články vo vedeckých časopisoch, prezentácie, krátke správy, dokumentáciu programu, pozvánky, …

`report` na dlhšie správy obsahujúce niekoľko kapitol, krátke knihy, dizertačné práce, …

`book` na skutočné knihy

`letter` na písanie listov (korešpondenciu)

`slides` na fólie. Tento typ používa veľké bezpätkové písmená (sans serif). Namiesto toho sa môžete rozhodnúť pre používanie FoilTeXu[a].

[a] CTAN:/tex-archive/macros/latex/packages/supported/foiltex

Tabuľka 1.2: Nastavenia dokumentu (voľby príkazu `\documentclass`)

`10pt`, `11pt`, `12pt` — nastavuje veľkosť hlavného fontu, použitého v dokumente. Ak sa voľba nepoužije, predpokladá sa `10pt`.

`a4paper`, `letterpaper`, ... — definuje formát strany. Východzia veľkosť je `letterpaper`. Okrem toho sa môžu použiť aj formáty `a5paper`, `b5paper`, `executivepaper` a `legalpaper`.

`fleqn` — namiesto centrovania zobrazí vzorce (rovnice) zarovnané doľava.

`leqno` — umiestňuje číslovanie vzorcov (rovníc) na ľavú stranu (vzorca) namiesto pravej.

`titlepage`, `notitlepage` — špecifikuje, či po názve dokumentu má byť začatá nová strana, alebo nie. Pri dokumente typu `article` nie je implicitne nastavené, aby po názve bola začatá nová strana, kým pri triedach `report` a `book` to tak nastavené je.

`twocolumn` — hovorí LaTeXu, aby vysádzal dokument v dvoch stĺpčekoch.

`twoside, oneside` — upresňuje, či výstup má byť jedno- alebo obojstranný. Pri triedach `article` a `report` je základom jednostranná a pri štýle `book` je to obojstranná tlač. Voľba `twoside` *neoznámi* vašej tlačiarni, že má skutočne vyrobiť dvojstranný výstup.

`openright, openany` — zabezpečí, aby kapitoly začínali buď iba na pravej strane alebo na ktorejkoľvek nasledujúcej voľnej strane. Toto nastavenie nepracuje pri štýle `article`, pretože tento typ dokumentu nepozná kapitoly. Pri dokumentoch typu `report` štandardne začína nová kapitola na najbližšej voľnej stánke a pri triede `book` na najsledujúcej pravej strane.

1.5.2 Balíky – packages

Počas písania dokumentu pravdepodobne nájdete miesta, kde samotný základný LATEX nemôže vyriešit váš problém. Keď chcete do vášho dokumentu vložiť grafiku, farebný text alebo zdrojový text zo súboru, musíte zväčšiť schopnosti LATEXu. Takéto rozšírenia sa nazývajú balíky makier. Tieto balíky sa aktivujú pomocou príkazu

```
\usepackage[voľby]{balík}
```

kde *balík* je názov balíka a *voľby* je zoznam kľúčových slov, ktoré spúšťajú špeciálne vlastnosti daného balíka. Niektoré balíky sú obsiahnuté v základnej distribúcii LATEXu2$_\varepsilon$ (pozri tabuľku 1.3). Ďalšie sú šírené samostatne. Viac informácií o balíkoch inštalovaných na vašej užívateľskej stanici sa môžete dočítať vo vašom *Local Guide* [4]. Základným zdrojom informácií o LATEXu je *The LATEX Companion* [3]. Obsahuje popisy stoviek balíkov, spolu s informáciami, ako napísať vaše vlastné rozšírenia pre LATEX 2$_\varepsilon$.

1.5.3 Štýly strán

LATEX podporuje tri preddefinované kombinácie záhlavia/päty strán nazývané štýly strán (page styles). Príkaz

```
\pagestyle{štýl}
```

definuje, ktorú z možných kombinácií treba použiť. Tabuľka 1.4 uvádza preddefinované štýly.

Štýl aktuálnej stránky sa môže meniť príkazom

```
\thispagestyle{štýl}
```

V *The LATEX Companion* [3] a v oddieli 4.5 na strane 65 nájdete popis, ako si môžete vytvoriť vaše vlastné záhlavie a pätu strany.

1.6 Súbory, na ktoré môžete natrafiť

Pri práci s LATEXom sa čoskoro ocitnete v bludisku súborov s rôznymi koncovkami (extensions) a pravdepodobne bez pochopenia ich pôvodu. Doleuvedný zoznam vám napovie niečo o rôznych typoch súborov, na ktoré môžete natrafiť pri práci s TEXom. Prosím, vezmite však na vedomie, že táto tabuľka si nenárokuje stať sa úplným zoznamom koncoviek a ak nájdete takú, o ktorej ste presvedčení, že je dôležitá, napíšte nám pár riadkov.

.tex Vstupný (zdrojový) súbor LATEXu alebo TEXu. Môže ho kompilovať `latex`, respektíve `tex`.

Tabuľka 1.3: Niektoré balíky, distribuované s LaTeXom

---

doc umožňuje dokumentáciu LaTeXových programov.
Popis je v `doc.dtx`[a] a v *The LaTeX Companion* [3].

exscale poskytuje stupňované verzie rozšíreného matematického fontu.
Popis v `ltexscale.dtx`.

fontenc stanovuje, aké kódovanie fontov má LaTeX použiť.
Popis v `ltoutenc.dtx`.

ifthen poskytuje príkazy typu 'ak ... potom vykonaj ... ináč vykonaj ....
Popis v `ifthen.dtx` a v *The LaTeX Companion* [3].

latexsym musíte použiť na sprístupnenie fontu symbolov LaTeXu.
Popis v `latexsym.dtx` a v *The LaTeX Companion* [3]

makeidx poskytuje príkazy na tvorbu indexov.
Popis v kapitole 4.4 a v *The LaTeX Companion* [3].

syntonly spracuje dokument bez toho, aby ho vysádzal.
Popis v `syntonly.dtx` a v *The LaTeX Companion* [3]. Toto je užitočné na rýchle preverenie chýb.

inputenc umožňuje špecifikovať vstupné kódovanie ako ASCII, ISO Latin-1, ISO Latin-2, kódovanie strán 437/850 IBM, Apple Macintosh, Next, ANSI-Windows alebo definované užívateľom.
Popis v `inputenc.dtx`.

---

[a] Tento súbor by mal byť inštalovaný na vašom systéme a mali by ste mať možnosť získať `dvi` súbor pomocou príkazu `latex doc.dtx`. Toto isté platí aj o ostatných balíkoch, uvádzaných v tejto tabuľke.

.sty LaTeXovký balík makier. Je to súbor, ktorý môžete zaradiť do vášho LaTeXovského dokumentu príkazom `\usepackage`.

.dtx Dokumentovaný TeX. Je to základný formát rozširovaných LaTeXových štýlových súborov. Ak spracujete .dtx súbor, získate dokumentovaný kód makier LaTeXovského balíka obsiahnutého v .dtx súbore.

.ins Je to inštalátor pre súbory obsiahnuté v označenom .dtx súbore. Ak si stiahnete LaTeXovský balík zo siete, väčšinou získate súbory .dtx a .ins. Spustením LaTeXu so súborom .ins rozbalíte súbor .dtx.

.cls Súbory tried určujú vzhľad vašich dokumentov. Zavádzajú sa príkazom `\documenentclass`.

Nasledujúce súbory sa vygenerujú ak spustíte LaTeX so svojím zdrojovým súborom:

.dvi Device Independent file (súbor nezávislý na zariadení). Toto je hlavný výsledok kompilovania LaTeXom. Jeho obsah si môžete pozrieť pomocou DVI prehliadača (DVI previewer program) alebo ho môžete poslať do tlačiarne po použití programu `dvips` alebo podobnej aplikácie.

.log Podáva podrobnú správu o priebehu poslednej kompilácie.

.toc Sem sa ukladajú hlavičky oddielov. Načíta sa pri nasledujúcej kompilácii a použije sa na vytvorenie obsahu (table of content).

.lof Podobne ako .toc súbor, avšak obsahuje zoznam obrázkov (list of figures).

.lot Znova to isté pre zoznam tabuliek (list of tables).

.aux Ďalší súbor prenášajúci informáciu z jednej kompilácie do ďalšej. Súbor s príponou .aux sa okrem iného používa na uloženie informácií súvisiacich s krížnymi odkazmi.

Tabuľka 1.4: Preddefinované štýly strán LaTeXu

`plain` vytlačí čísla strán na spodok strany do stredu päty strany. Toto je štandardný štýl strany.

`headings` vytlačí do záhlavia každej strany názov aktuálnej kapitoly (oddielu) a číslo strany, pričom päta dokumentu ostane prázdna. (Takýmto štýlom je napísaný tento dokument.)

`empty` nastavuje, aby záhlavie ako aj päta stránky ostali prázdne.

.idx Ak váš dokument obsahuje register. LaTeX sem ukladá všetky slová, ktoré sa majú objaviť v registri. Tento súbor musí byť spracovaný programom `makeindex`. Ďalšiu informáciu o tvorbe indexu nájdete v oddiele 4.4 na strane 63.

.ind Je to výsledok spracovania súboru .idx pripravený na zaradenie do vášho dokumentu počas nasledujúceho kompilačného cyklu.

.ilg Súbor hlásení o priebehu programu `makeindex`.

1.7 Veľké projekty

Počas práce s veľkými dokumentmi môžete chcieť rozdeliť vstupný súbor na niekoľko častí. LaTeX má dva príkazy, ktoré vám s tým pomôžu.

Príkaz

`\include{`*menosúboru*`}`

môžete použiť v texte na vloženie obsahu iného súboru *menosúboru.tex*. Zapamätajte si, že LaTeX pred spracovaním vloženého materiálu z *menosúboru.tex* začne novú stranu.

Druhý príkaz môže byť použitý v preambule. Umožňuje vám to prikázať LaTeXu, aby vložil len niektoré „`include`“ súbory.

`\includeonly{`*menosúboru*,*menosúboru*,...`}`

Po umiestnení tohoto príkazu v preambule dokumentu, budú vykonané iba tie príkazy `\include`, ktoré sú uvádzané v `\includeonly`. Zapamätajte si, že v názve súboru (menosúboru) sa nesmie vyskytovať čiarka ani medzera.

Príkaz `\include` začína tlač vkladaného textu na novej strane. Toto je nápomocné, keď používate `\includeonly`, pretože zalamovania strán (pagebreaks) sa neposúvajú ani v prípade, ak niektoré z vkladaných súborov vynecháme. Niekedy to však nemusí byť žiadúce. V takom prípade môžete použiť príkaz

`\input{`*menosúboru*`}`

Jednoducho vloží do dokumentu špecifikovaný súbor. Nepridávajú sa žiadne reťazce.

Ak chcete, aby LaTeX rýchlo skontroloval váš dokument, môžete použiť balík syntonly. Tento donúti LaTeX pri prechádzaní vaším dokumentom kontrolovať len správnosť syntaxe a použitia príkazov, nevytvára sa však žiaden (DVI) výstup. Keďže pri tomto režime beží LaTeX rýchlejšie, môžete ušetriť váš drahocenný čas. Použitie je veľmi jednoduché:

```
\usepackage{syntonly}
\syntaxonly
```

Ak chcete vytvoriť strany, stačí zakomentovať druhý riadok (pridaním znaku percenta).

Kapitola 2

Sádzanie textu

Po prečítaní predchádzajúcej kapitoly by ste mali vedieť o základnom materiále, z ktorého je dokument v LaTeXu vyrobený. V tejto kapitole doplním zvyšnú štruktúru, ktorú potrebujete poznať na tvorbu materiálov zo skutočného sveta.

2.1 Štruktúra textu a jazyka

Hlavným cieľom napísania textu (ak vylúčime určitú súčasnú literatúru typu DACC[1]) je vyjadrenie myšlienok, sprostredkovanie informácie alebo vedomostí čitateľovi. Čitateľ text lepšie pochopí, ak budú myšlienky dobre štruktúrované a uvidí a precíti túto štruktúru omnoho lepšie, ak bude typografická forma odrážať logickú a sémantickú štruktúru obsahu.

LaTeX sa od iných typografických systémov líši práve tým, že mu musíte oznámiť logickú a sémantickú štruktúru textu. Potom odvodí typografickú formu na základe „pravidiel" zadaných v súbore definujúcom triedu a v rozličných štýlových súboroch.

Najdôležitejšia jednotka textu v LaTeXu (a v typografii) je odsek (paragraph). Nazývame ho „textovou jednotkou", pretože odsek je typografická forma, ktorá by mala odrážať jednu súvislú myšlienku alebo jeden pojem. V nasledujúcich oddieloch sa naučíte, ako si môžete vynútiť zalomenie riadku napríklad pomocou \\ a ukončenie odseku, napríklad vynechaním prázdneho riadku v zdrojovom kóde. Preto, ak sa začína nová myšlienka, mal by sa začať aj nový odsek a ak nie, mali by byť použité len zalomenia riadkov. Ak ste na pochybách ohľadom ukončení odsekov, predstavte si svoj text, ako sprostredkovateľa pojmov a myšlienok. Ak máte ukončený odsek, ale stará myšlienka pokračuje, mal by byť nový odsek zrušený. Ak sa v tom istom odseku vyskytne nejaký úplne nový tok myšlienok, tak by mal byť odsek rozdelený.

[1]Different At All Cost, preklad švajčiarsko-nemeckého UVA (Um's Verrecken Anders).

Väčšina ľudí úplne podceňuje dôležitosť správneho členenia odsekov. Mnohí dokonca ani nepoznajú zmysel ukončenia odseku alebo, najmä pri použití LaTeXu, ukončujú odseky bez toho, aby si to uvedomovali. Táto druhá chyba sa dá obzvlášť jednoducho urobiť pri použití rovníc v texte. Pozrite si nasledujúce príklady a premyslite si, prečo sú prázdne riadky (ukončenia odsekov) niekedy použité pred a za rovnicami a niekedy nie sú použité. (Ak ešte dosť dobre nechápete všetky príkazy, aby ste porozumeli týmto príkladom, prečítajte si, prosím, túto a nasledujúcu kapitolu a potom si prečítajte tento oddiel znova.)

```
% Príklad 1
\ldots keď Einstein zaviedol svoj vzťah
\begin{equation}
  e = m \cdot c^2 \; ,
\end{equation}
ktorý je zároveň najznámejším a najmenej
pochopeným fyzikálnym vzorcom.

% Príklad 2
\ldots z~ktorého vyplýva Kirchoffov zákon pre prúdy:
\begin{equation}
  \sum_{k=1}^{n} I_k = 0 \; .
\end{equation}

Kirchhoffov zákon pre napätia môže byť odvodený \ldots

% Príklad 3
\ldots ktorý má mnohé výhody.

\begin{equation}
  I_D = I_F - I_R
\end{equation}
je jadro veľmi odlišného tranzistorového modelu. \ldots
```

Ďalšia menšia textová jednotka je veta. V anglických textoch sa dáva väčšia medzera za bodku, ktorá ukončuje vetu, ako za bodku, ktorá sa používa na konci skratky. LaTeX sa pokúša určiť, ktorú z nich chcete mať. Ak to LaTeX urobí nesprávne, musíte mu povedať, čo chcete. Toto bude vysvetlené neskôr v tejto kapitole.

Štruktúrovanie textu pokračuje dokonca do častí viet. Väčšina jazykov má veľmi zložité interpunkčné pravidlá, ale v mnohých jazykoch (vrátane nemčiny a angličtiny) dostanete takmer všetky čiarky správne, ak si zapamätáte, čo reprezentujú: krátku zastávku v toku reči. Ak si nie ste istí v tom, kam umiestniť čiarku, prečítajte vetu nahlas a zľahka sa nadýchnite pri každej čiarke. Ak to na niektorom mieste pôsobí nemotorne, zrušte túto čiarku, ak cítite potrebu nadýchnuť sa (alebo urobiť krátku prestávku) na nejakom inom mieste, vložte tam čiarku.

Nakoniec odseky textu by mali byť logicky štruktúrované na vyššej úrovni tým, že sú zoskupované do kapitol, oddielov, pododdielov, atď. Nakoniec typografický účinok napísania napríklad `\section{Štruktúra textu a jazyka}` je tak jasný, že je takmer úplne evidentné, ako majú byť tieto štruktúry vyššej úrovne používané.

2.2 Lámanie riadkov a strán

2.2.1 Zarovnávanie odsekov

Väčšinou sa knihy tlačia tak, že každý riadok má rovnakú dĺžku. LaTeX láme riadky a vkladá medzery medzi slová tak, aby dosiahol optimálny vzhľad celého odseku. V prípade, že je to nutné, slová, ktoré sa nezmestia pohodlne do riadku, rozdelí. Od druhu dokumentu závisí, ako sú odseky sádzané. Normálne je prvý riadok odseku pevne stanovený a medzi dvoma odsekmi nie je žiadne voľné miesto. Viac informácií nájdete v kapitole 5.3.2.

V špeciálnych prípadoch môžeme prikázať LaTeXu, aby zlomil riadok (ukončil ho) v danom mieste:

`\\` alebo `\newline`

začína nový riadok bez toho, aby začal nový odsek,

`\\*`

dodatočne zakazuje zalomenie strany po vyplnení príkazu na zalomenie riadku.

`\newpage`

začína novú stranu.

`\linebreak[`*n*`]`, `\nolinebreak[`*n*`]`, `\pagebreak[`*n*`]` a `\nopagebreak[`*n*`]`

robia presne to, čo hovoria ich názvy. Umožňujú autorovi ovplyvniť ich činnosť pomocou parametra n. Tento parameter môže nadobúdať hodnoty od 0 do 4. Nastavením parametra n na hodnotu nižšiu ako 4, necháváte LaTeXu možnosť ignorovať váš príkaz v prípade, ak by výstup vyzeral veľmi zle. Nepleťte si tieto „break“ príkazy s príkazmi „new“. Hoci zadávate príkaz „break“ (ukonči), LaTeX sa stále pokúša vyrovnať pravý okraj strany a vytvoriť stranu s požadovanou celkovou dĺžkou, ako to bude popísané v nasledujúcom oddiele. Ak chcete skutočne začať „nový riadok“, použite odpovedajúci príkaz. Uhádnite jeho názov!

LaTeX sa vždy pokúša dosiahnuť čo najlepšie zarovnávanie riadkov. Ak nevie nájsť spôsob, ako zarovnať riadky v súlade s jeho vysokými požiadavkami, nechá jeden riadok prečnievať za pravý okraj odseku. Počas spracovania vstupného súboru sa potom LaTeX sťažuje na „Overfull “hbox“. Toto sa najčastejšie stáva, keď LaTeX nevie nájsť vhodné miesto na rozdelenie slova.[2] Pomocou príkazu `\sloppy` môžete LaTeXu prikázať, aby trochu povolil v svojich požiadavkách. V takom prípade predchádza pretečeným riadkom zväčšením medzier medzi písmenami — aj keď konečný výstup nie je optimálny. Aj v takomto prípade vás LaTeX varuje „Underfull “hbox“. Vo väčšine týchto prípadov nevyzerá výstup príliš dobre. Príkaz `\fussy` vráti LaTeX na jeho štandardné správanie!

2.2.2 Delenie slov

LaTeX rozdeľuje slová vždy, keď je to potrebné. V prípade, že TeX nevie na základe svojich algoritmov nájsť vhodné miesto na rozdelenie slova, môžete túto situáciu napraviť a nasledujúcim príkazom mu povedať o výnimke.

Príkaz

`\hyphenation{`*zoznam slov*`}`

spôsobí, že slová uvedené v parametroch budú delené len na miestach označených „`-`“. Argument by mal obsahovať len slová zložené z normálnych písmen alebo radšej znakov, ktoré sú považované za normálne písmená v aktívnom kontexte. Rozdeľovacie pravidlá sa ukladajú pre jazyk, ktorý je aktívny v momente, keď sa príkaz objaví. To znamená, že ak umiestnite rozdeľovací príkaz do preambuly vášho dokumentu, ovplyvní anglické delenie slov. Ak príkaz umiestnite za `\begin{document}` a ak použijete nejaký balík pre národnú podporu ako babel, tak príkazy delenia budú aktívne v jazyku aktivovanom pomocou babel.

Nižšie uvedený príklad umožní rozdeliť slovo „rozdeľovanie“ rovnako, ako „Rozdeľovanie“, ochráni slová „FORTRAN“, „Fortran“ a „fortran“ pred rozdelením v celom texte. Stav písmen (veľké—malé) je ignorovaný. V zadaní slov nie sú povolené žiadne špeciálne znaky alebo symboly.

Príklad:

```
\hyphenation{FORTRAN roz-de-ľo-va-nie}
```

Príkaz `\-` umožňuje slovo v danom mieste rozdeliť. Len takéto miesto sa zároveň stáva miestom, kde môže byť dané slovo rozdelené. Tento príkaz

[2] Hoci vás LaTeX varuje, keď sa to stane („Overfull “hbox“), nie je vždy jednoduché nájsť takéto riadky. Ak v príkaze `\documentclass` použijete voľbu `draft`, budú tieto riadky označené hrubou čiernou čiarou na pravom okraji.

je zvlášť užitočný pre slová obsahujúce špeciálne znaky (napríklad znaky s diakritikou), pretože LaTeX takéto slová automaticky nedelí.[3]

```
Myslím, že je to: su\-per\-cal\-%
i\-frag\-i\-lis\-tic\-ex\-pi\-%
al\-i\-do\-cké
```

Myslím, že je to: supercalifragilisticexpialido-cké

Viacero slov môžeme udržať na jednom riadku pomocou príkazu

`\mbox{`*text*`}`

Zabezpečí, že jeho obsah ostane v každom prípade spolu.

```
Budem mať nové telefónne číslo ---
\mbox{0116 291 2319}.

Parameter
\mbox{\emph{filename}} musí
obsahovať názov súboru.
```

Budem mať nové telefónne číslo — 0116 291 2319.

Parameter *filename* musí obsahovať názov súboru.

2.3 Reťazce pripravené na použitie

V niektorých príkladoch na predchádzajúcich stranách ste videli veľmi jednoduché LaTeXovské príkazy na vytlačenie špeciálnych textových reťazcov:

Command	Example	Description
`\today`	17. marca 2002	Priebežný dátum v zapnutom jazyku
`\TeX`	TeX	Meno vášho obľúbeného sadzača
`\LaTeX`	LaTeX	The name of the Game
`\LaTeXe`	LaTeX 2_ε	Súčasná inkarnácia LaTeXu

2.4 Špeciálne znaky a symboly

2.4.1 Úvodzovky

Na vytvorenie úvodzoviek sa *nepoužíva* príkaz `"`, ako na písacom stroji. Pri tlači existujú špeciálne otváracie a zatváracie úvodzovky. V LaTeXu sa používajú dva znaky ` na otvorenie a dva ' na zatvorenie úvodzoviek.[4]

[3]Toto platí vo všeobecnosti. Napríklad CSLaTeX ich delí (pozn. prekl.).
[4]Pri použití štýlu `slovak.sty`, respektíve `czech.sty`, môžete písať „naše“ úvodzovky pomocou príkazu `\uv{naše}` (pozn. prekl.).

```
``Prosím, stlačte kláves `x'.''
```

“Prosím, stlačte kláves ‘x’.”

2.4.2 Pomlčky a rozdeľovníky

LaTeX pozná štyri druhy pomlčiek. Tri z nich dostanete pomocou rôzneho počtu po sebe idúcich pomlčiek. Štvrtý znak nie je vlastne ani pomlčka, je to matematický znak mínus:

```
X-násobný\\
strany 13--67\\
áno---alebo nie? \\
$0$, $1$ a $-1$
```

X-násobný
strany 13–67
áno—alebo nie?
0, 1 a -1

Názvy týchto pomlčiek sú: ‚-‘ rozdeľovník (spojovník, hyphen), ‚–‘ krátka pomlčka (en-dash), ‚—‘ pomlčka (em-dash) a ‚$-$‘ znak mínus.

2.4.3 Vlnka ($\sim$)

Znak, ktorý môžeme často vidieť, ako súčasť URL adresy je vlnka. V LaTeXu ju môžete vygenerovať použitím `\~`, ale výsledok: ˜ nie je v skutočnosti to, čo chcete. Namiesto toho skúste:

```
http://www.rich.edu/\~{}bush \\
http://www.clever.edu/$\sim$demo
```

http://www.rich.edu/˜bush
http://www.clever.edu/$\sim$demo

2.4.4 Bodky (...)

Na písacom stroji zaberá čiarka alebo bodka rovnaké miesto, ako každý iný znak. Pri tlači kníh zaberajú tieto znaky len veľmi malú plochu a nachádzajú sa veľmi blízko predchádzajúceho písmena. Preto nemôžete napísať „bodky“ len napísaním troch bodiek, pretože medzery medzi nimi by boli zlé. Namiesto toho sa používa špeciálny príkaz, nazývaný

`\ldots`

```
Nie ako ..., ale takto:\\
New York, Tokyo, Budapest,\ldots
```

Nie ako ..., ale takto:
New York, Tokyo, Budapest,…

2.4.5 Ligatúry

Niektoré písmena sa nesádzajú len tak, že sa vytlačia, ako rôzne písmená vedľa seba, ale niekedy sa vytlačia s použitím špeciálnych znakov.

ﬀ ﬁ ﬂ ﬃ... namiesto ff fi fl ffi ...

Týmto takzvaným ligatúram môže byť zabránené vložením `\mbox{}` medzi písmená, ktorých sa to týka. Toto môže byť nevyhnutné pri zložených slovách.

```
Nie shelfful\\
ale shelf\mbox{}ful
```

Nie shelﬀul
ale shelfful

2.4.6 Diakritika a špeciálne znaky

LaTeX podporuje používanie diakritiky a špeciálnych znakov z mnohých jazykov. Tabuľka 2.1 ukazuje všetky druhy diakritiky na písmene o. Prirodzene na ostatných písmenách to funguje tiež.

Aby sa dal použiť niektorý znak na i alebo j, musia sa najprv odstrániť ich bodky. Toto je zabezpečené napísaním `\i` a `\j`.

```
H\^otel, na\"\i ve, \'el\`eve,\\
sm\o rrebr\o d, !`Se\~norita!,\\
Sch\"onbrunner Schlo\ss{}
Stra\ss e
```

Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße

Tabuľka 2.1: Diakritika a špeciálne znaky

ò	`` \`o ``	ó	`\'o`	ô	`\^o`	õ	`\~o`
ō	`\=o`	ȯ	`\.o`	ö	`\"o`	ç	`\c c`
ŏ	`\u o`	ǒ	`\v o`	ő	`\H o`	ǫ	`\c o`
ọ	`\d o`	o̲	`\b o`	o͡o	`\t oo`		
œ	`\oe`	Œ	`\OE`	æ	`\ae`	Æ	`\AE`
å	`\aa`	Å	`\AA`				
ø	`\o`	Ø	`\O`	ł	`\l`	Ł	`\L`
ı	`\i`	ȷ	`\j`	¡	`` !` ``	¿	`` ?` ``

2.5 Medzinárodná jazyková podpora

Ak chcete písať dokument v inom jazyku ako anglickom, existujú dve oblasti, v ktorých musí byť LaTeX správne nakonfigurovaný:

1. Všetky automaticky vytvárané textové reťazce[5] musia byť prispôsobené na nový jazyk. Pre mnohé jazyky môžu byť tieto zmeny vykonané použitím balíka babel od Johannesa Braamsa.

2. LaTeX potrebuje poznať pravidlá delenia slov nového jazyka. Dostať pravidlá delenia slov do LaTeXu je trošku zložitejšie. Znamená to prebudovanie formátu so sprístupnením odlišných vzorov delenia slov. Váš *Local Guide* [4] by mal o tomto poskytnúť viac informácií.

Ak je už váš systém náležite nastavený, môžete aktivovať balík babel pridaním príkazu

```
\usepackage[jazyk]{babel}
```

nasledujúcim po príkaze `\documentclass`. V *Local Guide* [4] by malo byť uvedené, ktoré *jazyky* podporuje váš systém. Babel bude automaticky aktivovať vhodné pravidlá delenia slov jazyka, ktorý ste zvolili. Ak váš formát LaTeXu (napríklad `cslatex`) nepodporuje delenie slov vo vami zvolenom jazyku, babel bude aj tak funkčný, ale neumožní delenie, ktoré by malo úplne negatívny vplyv na vizuálny zjav vášho dokumentu.

Pre niektoré jazyky špecifikuje babel aj nové príkazy zjednodušujúce vkladanie špeciálnych znakov. Napríklad v nemeckom jazyku existuje mnoho špeciálnych znakov (äöü). Pri použití babelu môžete napísať ö pomocou príkazu `"o` namiesto `\"o`.

Niektoré počítačové systémy umožňujú vkladanie špeciálnych znakov priamo z klávesnice. LaTeX vie spracovať aj takéto znaky. Od decembrového vydania LaTeXu2$_\varepsilon$ v roku 1994 obsahuje rozširovaná verzia LaTeXu2$_\varepsilon$ podporu rôznych vstupných kódovaní. Pozrite si balík inputenc. Počas používania tohoto balíka si musíte uvedomiť, že iní ľudia nemusia mať možnosť zobraziť si váš zdrojový súbor na svojom počítači, pretože môžu používať iné kódovanie. Napríklad „nemecký" znak ä je na PC označený ako 132 a na niektorých UNIXovských systémoch, používajúcich ISO-LATIN 1, je označený ako 228. Preto používajte tieto nastavenia opatrne.

Iná vec je kódovanie fontu. Definuje, na ktorej pozícii TeXovského fontu je uložené každé písmeno. Originálne TeXovské fonty Computer Modern obsahujú len 128 znakov starej 7-bitovej ASCII tabuľky. Ak sú požadované akcentované znaky, vytvára ich TeX kombinovaním normálnych písmen a príslušného akcentu. Hoci výsledný výstup môže vyzerať bezchybne, tento

[5] Obsah, Zoznam obrázkov, …

prístup neumožňuje automatické delenie slov obsahujúcich písmená s diakritikou.

Našťastie súčasné distribúcie TEXu obsahujú EC fonty. Tieto fonty vyzerajú ako fonty Computer Modern, ale obsahujú aj zvláštne znaky pre akcentované písmená, používané v európskych jazykoch. Použitím týchto fontov môžete skvalitniť delenie slov v neanglických dokumentoch. EC fonty sa sprístupňujú použitím balíka fontenc v preambule vášho dokumentu

```
\usepackage[T1]{fontenc}
```

2.6 Medzery medzi slovami

Aby LATEX dosiahol vo výstupe zarovnaný pravý okraj, vkladá medzi slová medzery rôznych veľkostí. Za vetou vkladá napatrne väčšiu medzeru, čo robí text čitateľnejším. LATEX sa domnieva, že vety končia bodkou, otáznikom alebo výkričníkom. Ak nasleduje bodka za veľkým písmenom, nie je to považované za koniec vety, nakoľko je to jav typický pre skratky.

Každá výnimka z tohoto pravidla musí byť upresnená autorom. Opačné lomítko pred medzerou vytvára medzeru, ktorá nebude zväčšená. Znak vlnovky ‚~' vytvára medzeru, ktorá nesmie byť zväčšená a naviac zabraňuje ukončeniu riadku v danom mieste. Príkaz `\@` pred bodkou určuje, že táto bodka ukončuje vetu, aj keď je pred ňou veľké písmeno.

```
Dr.~Smith bol rád, že ju videl\\
napríklad~obr.~5\\
Mám rád BASIC\@. A~čo vy?
```

Dr. Smith bol rád, že ju videl
napríklad obr. 5
Mám rád BASIC. A čo vy?

Zväčšeniu medzery za bodkou môžete zabrániť použitím príkazu

```
\frenchspacing
```

ktorý prikáže LATEXu, aby *nevkladal* o nič väčšie medzery za bodkou ako za normálnym znakom. Toto je bežné pre neanglické jazyky okrem bibliografií. Ak použijete `\frenchspacing`, príkaz `\@` nie je potrebný.[6]

2.7 Tituly, kapitoly a oddiely

Aby ste pomohli čitateľovi alebo čitateľke nájsť cestu vaším dokumentom, mali by ste ho rozdeliť na kapitoly, odddiely a pododdiely. Toto LATEX za-

[6] Ak používate voľbu `slovak`, respektíve `czech`, je príkaz `\frenchspacing` automaticky aktívny (pozn. prekl.).

bezpečuje pomocou špeciálnych príkazov, ktoré používajú názov oddielu ako svoj argument. Je na vás, aby ste ich použili v správnom poradí.

Pre štýl `article` sú dostupné nasledujúce druhy delenia:

```
\section{...}            \paragraph{...}
\subsection{...}         \subparagraph{...}
\subsubsection{...}      \appendix
```

Pre štýly `report` a `book` môžete naviac použiť nasledujúce dva príkazy:

```
\part{...}               \chapter{...}
```

Keďže `article` nepozná kapitoly, je úplne jednoduché pridávať články, ako kapitoly do knihy. Medzery medzi oddielmi, číslovanie a veľkosť fontov, ich názvov sú nastavené automaticky LaTeXom.

Dva z uvedených príkazov na vytvorenie oddielov sú trochu špeciálne:

- Príkaz `\part` nemá vplyv na číselné poradie kapitol.
- Príkaz `\appendix` nemá parameter. Mení iba číselné označenie kapitol na písmená[7].

LaTeX vytvára obsah použitím názvov oddielov a im príslušných čísel strán, ktoré získal pri predchádzajúcom spracovaní textu. Príkaz

```
\tableofcontents
```

vytvorí obsah a umiestni ho na miesto, kde sa daný príkaz nachádza. Nový dokument musí byť spracovaný („preLaTeXovaný“) dvakrát, aby sme dostali správny obsah. V niektorých prípadoch môže byť potrebné spracovať dokument tretí krát. LaTeX vám povie, kedy je to potrebné.

Všetky oddielové príkazy uvedené vyššie existujú aj v „hviezdičkovej“ forme. „Hviezdičková“ verzia príkazu sa vytvára pridaním znaku hviezdičky `*` za názvom príkazu. Tieto príkazy vytvoria hlavičku oddielu, ktorý nebude číslovaný a ani uvedený v obsahu. Napríklad namiesto príkazu `\section{Help}` napíšeme príkaz `\section*{Help}`.

Štandardne je názov oddielu vypísaný v obsahu presne tak, ako bol napísaný v texte. Niekedy to však nie je možné, pretože názov oddielu je príliš dlhý na to, aby sa do obsahu zmestil. Údaj pre obsah môže byť preto špecifikovaný ako nepovinný parameter pred aktuálnym nadpisom.

```
\chapter[Čítajte to! Je to napínavé]{Toto je veľmi dlhý
                        a zvlášť nudný nadpis}
```

[7] Pri štýle `article` mení číslovanie oddielov.

Titul celého dokumentu je vytváraný príkazom

`\maketitle`

Obsah titulu sa musí definovať príkazmi

`\title{...}`, `\author{...}` a prípadne `\date{...}`

predtým, ako je zavolaný príkaz `\maketitle`. V argumente `\authors` môžete vložiť viacero mien oddelených príkazmi `\and`.

Príklad niektorých vyššie spomenutých príkazov môžete nájsť na obrázku 1.3 na strane 9.

Okrem oddielových príkazov vysvetlených vyššie, LaTeX 2_ε pozná naviac 3 príkazy, ktoré sa používajú v triede `book`:

`\frontmatter`, `\mainmatter` a `\backmatter`

Sú užitočné na rozdelenie vašej publikácie. Prerobia nadpisy kapitol a číslovanie strán tak, ako by ste to očakávali v knihách.

2.8 Krížové odkazy

V knihách, správach a článkoch sú často krížové odkazy na obrázky, tabuľky a špeciálne časti textu. LaTeX poskytuje na vytváranie krížových odkazov nasledujúce príkazy

`\label{`*značka*`}`, `\ref{`*značka*`}` a `\pageref{`*značka*`}`

kde *značka* je identifikátor, zvolený autorom. LaTeX nahrádza `\ref` číslom oddielu, pododdielu, obrázku, tabuľky, rovnice alebo vety zodpovedajúcemu miestu, kde bol príkaz `\label` zadaný. `\pageref` vytlačí číslo strany zodpovedajúcej miestu, kde bol príkaz `\label` zadaný.[8] Rovnako ako názvy oddielov sú aj čísla použité z predchádzajúceho behu LaTeXu.

```
Odkaz k~tomuto pododdielu
\label{sec:tento} vyzerá:
\uv{pozri oddiel~\ref{sec:tento} na
strane~\pageref{sec:tento}.}
```

Odkaz k tomuto pododdielu vyzerá: „pozri oddiel 2.8 na strane 27.“

[8] Všimnite si, že tieto príkazy si neuvedomujú, na čo odkazujú. `\label` len uloží posledné automaticky vygenerované číslo.

2.9 Poznámky pod čiarou

Pomocou príkazu

`\footnote{`*text poznámky pod čiarou*`}`

sa vytvorí na konci danej strany poznámka pod čiarou. Poznámky by mali byť vždy použité za slovom[9] alebo za vetou, na ktorú sa vzťahujú.[10]

```
Poznámky pod čiarou\footnote{Toto
je poznámka pod čiarou} sú
často používané ľuďmi,
používajúcimi \LaTeX.
```

Poznámky pod čiarou[a] sú často používané ľuďmi, používajúcimi LaTeX.

[a]Toto je poznámka pod čiarou

2.10 Zvýraznené slová

V rukopise vytvorenom na písacom stroji sa dôležité slová zvýrazňujú obyčajne podčiarknutím. V tlačených knihách sa však takéto slová *zvýrazňujú* vytlačením *kurzívou*. LaTeX na *zvýraznenie textu* poskytuje príkaz

`\emph{`*text*`}`

Jeho argumentom je text, ktorý chceme zvýrazniť. Čo príkaz v danom prípade urobí so svojím argumentom, závisí od kontextu:

```
\emph{Ak použijete
\emph{zvýrazňovanie} v~už
zvýraznenom texte,
\LaTeX{} použije
na zvýraznenie \emph{vzpriamený}
typ písma (\emph{antikvu}).}
```

Ak použijete zvýrazňovanie *v už zvýraznenom texte, LaTeX použije na zvýraznenie* vzpriamený *typ písma (*antikvu*).*

Všimnite si, prosím, rozdiel medzi prikázaním LaTeXu niečo *zvýraniť* a požiadaním ho použiť iný *font*:

```
\textit{Môžete tiež
  \emph{zvýrazniť} text ak
  je sádzaný kurzívou,}
\textsf{vo fonte
  \emph{sans-serif},}
\texttt{alebo štýlom
  \emph{typewriter}.}
```

Môžete tiež zvýrazniť *text ak je sádzaný kurzívou,* vo fonte *sans-serif*, alebo štýlom *typewriter*.

[9]Napríklad týmto.

[10]Poznámky, vzťahujúce sa ku vete alebo jej časti, by preto mali byť uvedené za čiarkou alebo bodkou.

2.11 Prostredia

Aby sa dal napísať text rôzneho typu, LaTeX má definovaných množstvo prostredí na viac rôznych druhov formátovania:

`\begin{`*názov*`}` *text* `\end{`*názov*`}`

Názov prostredia je *názov*. Prostredia môžu byť volané viackrát (aj jedno v druhom), pokiaľ je dodržané ich poradie

```
\begin{aaa}...\begin{bbb}...\end{bbb}...\end{aaa}
```

V nasledujúcich podkapitolách sú opísané všetky dôležité prostredia.

2.11.1 Odrážky `itemize`, číslované zoznamy `enumerate` a popisy `description`

Prostredie `itemize` je vhodné na vytváranie jednoduchých odrážok, prostredie `enumerate` na vytváranie číslovaných zoznamov a `description` je prostredie na vytváranie popisov.

```
\begin{enumerate}
\item  Prostredia môžete
 kombinovať podľa chuti:
\begin{itemize}
\item Ale môže to začať
vyzerať zle.
\item[-] S~pomlčkou.
\end{itemize}
\item Preto si pamätajte:
\begin{description}
\item[Hlúposti] sa nestanú
rozum\-ný\-mi preto,
že budú v~zozname.
\item[Rozumné] veci však môžu
byť podané v~zozname.
\end{description}
\end{enumerate}
```

1. Prostredia môžete kombinovať podľa chuti:
   - Ale môže to začať vyzerať zle.
   - S pomlčkou.
2. Preto si pamätajte:

   Hlúposti sa nestanú rozumnými preto, že budú v zozname.

   Rozumné veci však môžu byť podané v zozname.

2.11.2 Sadzba na vlajku a centrovanie

Prostedia `flushleft` a `flushright` vytvárajú odseky, ktoré sú zarovnané buď doľava alebo doprava. Prostredie `center` vytvára centrovaný text. Ak nezadáte `\\`, aby ste upresnili lámanie riadkov, LaTeX ho určí sám.

```
\begin{flushleft}
Tento text je\\ zarovnaný doľava.
\LaTeX{} sa nepokúša vytvoriť
každý riadok rovnakej dĺžky.
\end{flushleft}
```

Tento text je
zarovnaný doľava. LaTeX sa nepokúša vytvoriť každý riadok rovnakej dĺžky.

```
\begin{flushright}
Tento text je zarovnaný\\ doprava.
\LaTeX{} sa nepokúša vytvoriť
každý riadok rovnakej dĺžky.
\end{flushright}
```

Tento text je zarovnaný
doprava. LaTeX sa nepokúša vytvoriť každý riadok rovnakej dĺžky.

```
\begin{center}
V~strede \\ Zeme
\end{center}
```

V strede
Zeme

2.11.3 Dlhšie citácie a verše

Prostredie `quote` je vhodné na citácie, dôležité frázy a príklady.

```
Typografické pravidlo palca
pre dĺžku riadku:
\begin{quote}
Žiaden riadok nemá obsahovať
viac ako 66~znakov.

Preto majú strany \LaTeX u~nastavené
ako východzie také veľké okraje.
\end{quote}
Preto sa v~tlači novín často
používa viacero stĺpcov.
```

Typografické pravidlo palca pre dĺžku riadku:

> Žiaden riadok nemá obsahovať viac ako 66 znakov.
>
> Preto majú strany LaTeXu nastavené ako východzie také veľké okraje.

Preto sa v tlači novín často používa viacero stĺpcov.

Prostredia `quotation` a `verse` sú podobné. Prostredie `quotation` sa používa na dlhšie citácie skladajúce sa z viacerých odsekov, pretože odseky vie odsadzovať. Prostredie `verse` je užitočné na písanie básní, kde je potrebné lámať riadky. Riadky sú delené pomocou príkazu `\\`, použitého na konci riadku alebo pomocou voľného riadku za každým veršom.

```
Naspamäť poznám len jednu anglickú
báseň. Je o Humpty Dumpty.
\begin{flushleft}
\begin{verse}
Humpty Dumpty sat on a wall:\\
Humpty Dumpty had a great fall.\\
All the King's horses and all
the King's men\\
Couldn't put Humpty together
again.
\end{verse}
\end{flushleft}
```

Naspamäť poznám len jednu anglickú báseň.
Je o Humpty Dumpty.

Humpty Dumpty sat on a wall:
Humpty Dumpty had a great
fall.
All the King's horses and all
the King's men
Couldn't put Humpty together
again.

2.11.4 Doslovná tlač

Text, uzavretý medzi `\begin{verbatim}` a `\end{verbatim}`, bude priamo vytlačený tak, ako by bol napísaný na písacom stroji so všetkými zakončeniami riadkov a medzerami bez uskutočnenia akéhokoľvek LaTeXového príkazu.

Vo vnútri odseku, môžete dosiahnuť rovnaký výsledok aj pomocou príkazu

\verb+*text*+

`+` je len oddeľovací znak. Možete použiť ľubovoľný znak okrem písmen, znaku `*` a medzier. Množstvo LaTeXových príkladov v tejto brožúre je vysádzaných pomocou tohto príkazu.

```
Príkaz \verb|\ldots| \ldots

\begin{verbatim}
10 PRINT "HELLO WORLD ";
20 GOTO 10
\end{verbatim}
```

Príkaz `\ldots` …

```
10 PRINT "HELLO WORLD ";
20 GOTO 10
```

```
\begin{verbatim*}
verzia prostredia
      verbatim
s~hviezdičkou zvýrazňuje
medzery    v~texte
\end{verbatim*}
```

```
verzia␣prostredia
␣␣␣␣␣␣verbatim
s~hviezdičkou␣zvýrazňuje
medzery␣␣␣␣v~texte
```

Príkaz `\verb` sa môže použiť v podobnom tvare s hviezdičkou:

```
\verb*|napríklad   takto :-) |
```

`napríklad␣␣␣takto␣:-)␣`

Prostredie `verbatim` a príkaz `\verb` sa nesmú používať v parametroch iných príkazov.

2.11.5 Vytváranie tabuliek

Prostredie `tabular`[11] sa používa na sadzbu prekrásnych tabuliek s voliteľnými vertikálnymi a horizontálnymi čiarami. LaTeX určuje šírku stĺpcov automaticky.

Argument *typ stĺpca* príkazu

`\begin{tabular}{`*typ stĺpca*`}`

definuje formát tabuľky. Parameter `l` vytvorí stĺpec zarovnaný doľava, `r` doprava, `c` centrovaný stĺpec, parameter `p{`*šírka*`}` vytvorí stĺpec danej šírky (zarovnaný doľava) a nakoniec príkaz `|` vytvorí zvislú čiaru.[12]

V prostredí `tabular` sa pomocou znaku `&` prepínate do nasledujúceho stĺpca, `\\` začína nový riadok, `\hline` vkladá horizontálnu čiaru a `\cline` podčiarkne zvolené stĺpce.

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimal \\
3700 & octal \\ \cline{2-2}
11111000000 & binary \\
\hline \hline
1984 & decimal \\
\hline
\end{tabular}
```

7C0	hexadecimal
3700	octal
11111000000	binary
1984	decimal

```
\begin{tabular}{|p{4.7cm}|}
\hline
Vitajte v~Boxyho odseku.
Úprimne dúfame, že sa
Vám bude naša show páčiť.
\\ \hline
\end{tabular}
```

Vitajte v Boxyho odseku. Úprimne dúfame, že sa Vám bude naša show páčiť.

[11] Tabuľky vytvorené v prostredí `tabular` nemôžu byť rozdelené na dve strany. Na dlhé tabuľky sa používajú štýly `supertab.sty` a `longtabl.sty` (pozn. prekl.).

[12] Balík `hhline.sty` umožňuje rôzne spôsoby, akými sa majú čiary v tabuľkách pretínať (pozn. prekl.).

Pomocou príkazu `@{...}` je možné špecifikovať oddelovač stĺpcov. Tento príkaz ruší medzistĺpcové medzery a nahrádza ich tým, čo sa nachádza v zložených zátvorkách. Jeden zo spôsobov použitia tohoto príkazu bude uvedený nižšie pri probléme zarovnania podľa desatinnej bodky. Iný možný spôsob použitia je potlačenie okrajových medzier v tabuľke pomocou príkazu `@{}`.

```
\begin{tabular}{@{} l @{}}
\hline
žiadne okrajové medzery\\\hline
\end{tabular}
```

žiadne okrajové medzery

```
\begin{tabular}{l}
\hline
medzery napravo a naľavo\\
\hline
\end{tabular}
```

medzery napravo a naľavo

Keďže nie je zabudovaný spôsob na zarovnávanie číselných stĺpcov podľa desatinnej bodky[13], môžeme trochu zašvindľovať a použiť dva stĺpce: doprava zarovnaný pre celé čísla a doľava zarovnaný pre desatinnú časť. Príkaz `@{.}` použitý v riadku `\begin{tabular}` nahrádza bežnú medzistĺpcovú medzeru znakom „.“, pričom vytvára dojem, že daný stĺpček je zarovnaný podľa desatinnej bodky. Nezabudnite nahradiť desatinnú bodku vo vašich číslach oddeľovačom stĺpcov (`&`)! Označenie stĺpca môžeme nad náš „číselný stĺpec“ umiestniť pomocou príkazu `\multicolumn`, ktorým sa spája viac stĺpcov.

```
\begin{tabular}{c r @{.} l}
Výraz Pí      &
\multicolumn{2}{c}{Hodnota} \\
\hline
$\pi$               & 3&1416  \\
$\pi^{\pi}$         & 36&46   \\
$(\pi^{\pi})^{\pi}$ & 80662&7 \\
\end{tabular}
```

Výraz Pí	Hodnota
π	3.1416
π^{π}	36.46
$(\pi^{\pi})^{\pi}$	80662.7

[13] Ak máte nainštalovaný balík tools, pozrite sa na balík dcolumn.

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}%
     {\textbf{V strede}} \\
\hline
Vľavo & Vpravo \\
\hline
\end{tabular}
```

V strede	
Vľavo	Vpravo

2.11.6 Nastavenie tabulátorov

Prostredie tabbing[14] umožňuje autorovi pracovať s tabulátormi. Presnejší popis môžete nájsť napríklad v [12].

Na rozdiel od prostredia tabular sa v prostredí tabbing sa môže objaviť koniec stránky, teda jeho dĺžka je neobmedzená. Tabulátory je možné používať napríklad na formátovanie zápisov štruktúrovaných programov. Nasleduje stručný popis jednotlivých príkazov.

\= – nastavenie tabulačnej zarážky

\> – prechod na ďalšiu zarážku

\< – prechod na predchádzajúcu zarážku

\\ – nový riadok

\kill – ukončenie „vzorového“ riadku

\+ – posunutie ľavého okraja textu na ďalšiu zarážku

\- – posunutie ľavého okraja textu o zarážku doľava

\' – umiestnenie pravého okraja textu na pravý okraj predchádzajúcej zarážky (alebo 1\tabbingsep pred aktuálnu zarážku)

\` – posunie nasledujúci text na pravý okraj riadku

\pushtabs – uloženie pozícií všetkých zarážok

\poptabs – vyvolanie pozícií zarážok

\a. . . – v prostredí tabbing nefungujú príkazy \=, \', \` vytvárajúce akcenty. Na tieto akcenty v prostredí tabbing používame príkazy \a=, \a', \a`.

Nasledujúci príklad ilustruje použitie „vzorového“ riadku a všetkých vyššie uvedených príkazov s výnimkou \pushtabs a \poptabs. TEX sa nastaví na zarážku nezávisle od toho, či sa text bude prekrývať alebo nie.

[14] Pridané pri slovenskom preklade (pozn. prekl.).

```
raz dva  tri štyri
    dva
         tri
             štyri
         tri
raz                        dva
     tri
             štyrikrát
    dva
štyrtri  dvaraz
```

```
\begin{tabbing}
raz \=dva \quad \=tri\kill
raz \>dva \>tri \=štyri \+\\
dva\+\\
tri\+\\
štyri\\
\<tri\\
\<\<\<raz \`dva \\
\<tri\'\\
štyrikr\a'at\-\-\\
dva\-\\
štyri\>tri\>dva\>raz\\
\end{tabbing}
```

2.12 Plávajúce objekty

V súčasnosti obsahuje väčšina publikácií množstvo obrázkov a tabuliek. Tieto elementy si vyžadujú zvláštny prístup, pretože sa nemôžu rozdeľovať na viacero strán. Jedným zo spôsobov riešenia tohoto problému je začať novú stranu zakaždým, keď je obrázok alebo tabuľka príliš veľká, aby sa zmestila na aktuálnu stranu. Tento prístup by zanechal strany čiastočne prázdne, čo v konečnom dôsledku vyzerá veľmi zle.

Riešením tohoto problému je „plávanie“ obrázku alebo tabuľky, ktorá nebude umiestnená na danej strane, ale na niektorej z nasledujúcich strán, kým daná strana bude zaplnená textom. LaTeX poskytuje dve prostredia na výrobu plávajúcich objektov. Jedno pre tabuľky a jedno pre obrázky. Aby ste mohli plne využiť výhody týchto dvoch prostredí, je dôležité, aby ste približne chápali, ako s nimi LaTeX interne narába. V opačnom prípade môžu byť tieto plávajúce objekty hlavným zdrojom frustrácie, pretože LaTeX ich nikdy nedá tam, kde by ste ich chceli mať.

Pozrime sa na príkazy, ktoré nám LaTeX ponúka pre plávajúce objekty:

Všetok materiál umiestnený v prostrediach `figure` a `table` je považovaný za plávajúci. Obe tieto plávajúce prostredia podporujú nepovinný parameter

`\begin{figure}[`*špecifikátor umiestnenia*`]` alebo `\begin{table}[`*špecifikátor umiestnenia*`]`

nazývaný *špecifikátor umiestnenia*. Tento parameter hovorí LaTeXu, kam môže umiestniť daný plávajúci útvar. *Špecifikátor umiestnenia* je tvorený reťazcom *podmienok na umiestnenie plávajúceho objektu*. Pozri tabuľku 2.2.

Tabuľka môže začínať napríklad nasledujúcim riadkom

```
\begin{table}[!hbp]
```

Tabuľka 2.2: Podmienky na umiestnenie plávajúceho objektu

Špec.	Povolenie na umiestnenie plávajúceho objektu
`h`	*here (tu)* práve na tom mieste textu, kde bol daný príkaz vydaný. Toto je užitočné najmä pre malé plávajúce objekty.
`t`	na vrchu strany (*top*)
`b`	na spodu strany (*bottom*)
`p`	na zvláštnej strane, obsahujúcej len plávajúce objekty (*page*)
`!`	bez ohľadu na vnútorné parametre[a], ktoré môžu zabrániť umiestneniu plávajúceho objektu

[a]Ako napríklad maximálny počet plávajúcich objektov na jednej strane.

Špecifikátor umiestnenia `[!hbp]` povoľuje LaTeXu umiestniť tabuľku priamo tu (`h`) alebo na spodok nejakej strany (`b`) alebo na špeciálnu stranu pre plávajúce objekty (`p`) a to aj v prípade, že to nebude vyzerať veľmi dobre (`!`). Pokiaľ nie je zadaný žiaden špecifikátor, LaTeX pracuje s `[tbp]`.

LaTeX umiestni každý plávajúci objekt podľa špecifikátora umiestnenia, zadaného autorom. Ak plávajúci objekt nemôže byť umiestnený na aktuálnej strane, je zaradený do poradovníka *obrázkov* alebo *tabuliek*[15]. Keď sa začne nová strana, LaTeX najprv skontroluje, či nie je možné vytvoriť špeciálnu stranu zaplnenú len plávajúcimi objektami z poradovníkov. Ak to nie je možné, LaTeX vezme prvý objekt z poradovníka, ako keby bol len teraz umiestnený v texte a pokúša sa ho znovu umiestniť na základe špecifikátorov umiestnenia, ktoré boli pri ňom zadané (okrem „`h`", ktoré už nie je prijateľné). Všetky ďalšie plávajúce objekty, ktoré sa v texte objavia, sú umiestnené do príslušného poradovníka. LaTeX prísne dodržuje pôvodné poradie vzhľadom na každý druh plávajúcich objektov. To je príčinou toho, že obrázok, ktorý nemôže byť umiestnený, vytlačí všetky ďalšie obrázky na koniec dokumentu. Preto:

> Ak LaTeX neumiestňuje plávajúce objekty podľa vašich očakávaní, je to často kvôli nejakému plávajúcemu objektu, ktorý upcháva jeden z dvoch poradovníkov plávajúcich objektov.

Keď sme si už vysvetlili náročnejšiu časť, existuje ešte pár vecí, ktoré by sme mali o prostrediach `table` a `figure` spomenúť. Príkazom

`\caption{`*názov plávajúceho objektu*`}`

môžete definovať názov plávajúceho objektu. Priebežné poradové číslo a označenie „Obrázok" alebo „Tabuľka" doplní LaTeX.

[15]tieto sú FIFO – ‚first in, first out' – zásobníky

Dva príkazy

`\listoffigures` a `\listoftables`

pracujú podobne ako príkaz `\tableofcontents`. Vytlačia zoznam obrázkov, respektíve tabuliek. V týchto zoznamoch je spomenutý celý názov daných objektov. Ak ste náchylní k používaniu dlhých názvov obrázkov (tabuliek), musíte mať kratšiu verziu názvu, ktorá pôjde do zoznamu. Toto sa zabezpečuje napísaním kratšej verzie názvu do hranatých zátvoriek za príkazom `\caption`.

```
\caption[Krátky]{DDDllllhhhhýýýý}
```

Príkazmi `\label` a `\ref` môžete vytvoriť odkazy na plávajúce objekty vo vnútri textu.

Nasledujúci príklad kreslí štvorec a vkladá ho do dokumentu. Toto môžete použiť, ak chcete nechať miesto na obrázky, ktoré chcete vložiť (vlepiť) do dokončeného dokumentu.

```
Obrázok~\ref{biela} je príkladom Pop-Art\,u.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Päťkrát päť centimetrov} \label{biela}
\end{figure}
```

Vo vyššie uvedenom príklade[16] sa LaTeX *skutočne ťažko* (`!`) pokúša umiestniť obrázok priamo *tu* (`h`). Ak to nie je možné, pokúša sa umiestniť obrázok na *spodu* (`b`) strany. Neuspejúc pri pokuse umiestniť obrázok na danej strane zisťuje, či by nebolo možné vytvoriť stranu plávajúcich objektov obsahujúcu tento obrázok a možno niekoľko tabuliek z poradovníka tabuliek. Ak nemá dostatok materiálu na stranu plávajúcich objektov, LaTeX začne novú stranu a ešte raz bude narábať s obrázkom, ako keby sa práve teraz objavil v texte.

Za určitých podmienok môže byť nevyhnutné použitie príkazu

`\clearpage` alebo dokonca `\cleardoublepage`

Tento prikazuje LaTeXu okamžite umiestniť zvyšné plávajúce objekty nachádzajúce sa v poradovníkoch a až potom začať novú stranu. Na začiatok novej pravej strany nás nastaví príkaz `\cleardoublepage`.

Neskôr sa v tomto úvode do LaTeXu dozviete, ako vkladať PostScriptové obrázky do vašich LaTeXových dokumentov.

[16] za pedpokladu, že poradovník obrázkov je prázdny

Kapitola 3

Sadzba matematických vzorcov

Teraz ste pripravení! V tejto kapitole zaútočíme na najsilnejšie miesto TEXu: sadzbu matematiky. Ale varujem vás, táto kapitola iba pohladká povrch. Keďže tu opísané veci postačujú väčšine ľudí, nezúfajte, keď nebudete vedieť nájsť riešenie vašich potrieb matematickej sadzby. Je vysoko pravdepodobné, že váš problém je vyriešený v $\mathcal{A}\mathcal{M}\mathcal{S}$-LATEXu[1] alebo v niektorom inom balíku.

3.1 Všeobecné

LATEX má zvláštny režim na sadzbu matematiky. Matematický text je v odseku písaný medzi `\begin{math}` a `\end{math}`, medzi `$` a `$` alebo medzi `\(` a `\)`.

```
Sčítajte $a$ na druhú a $b$
na druhú tak, aby ste dostali $c$
na druhú. Alebo použijúc
matematickejší prístup:\\
$c^{2}=a^{2}+b^{2}$
```

Sčítajte a na druhú a b na druhú tak, aby ste dostali c na druhú. Alebo použijúc matematickejší prístup:
$c^{2}=a^{2}+b^{2}$

```
\TeX{} sa vyslovuje ako
$\tau\epsilon\chi$.\\[6pt]
100~m$^{3}$ vody\\[6pt]
Toto vychádza z~môjho $\heartsuit$.
```

TEX sa vyslovuje ako $\tau\epsilon\chi$.

100 m^{3} vody

Toto vychádza z môjho $\heartsuit$.

Väčšie matematické vzorce alebo rovnice je výhodnejšie sádzať ako samostatné výrazy a nie len ich písať na samostatných riadkoch. Preto ich mu-

[1] `CTAN:/tex-archive/macros/latex/packages/amslatex`

síte uzatvárať medzi `\[` a `\]`, alebo medzi príkazy `\begin{displaymath}` a `\end{displaymath}`. Tieto príkazy vytvoria vzorce, ktoré nie sú číslované. Ak chcete, aby ich LaTeX čísloval, môžete použiť prostredie `equation`.

```
Sčítajte $a$ na druhú a $b$
na druhú tak, aby ste dostali $c$
na druhú. Alebo použijúc
matematickejší prístup:
\begin{displaymath}
c^{2}=a^{2}+b^{2}
\end{displaymath}
A~práve jeden ďalší riadok.
```

Sčítajte a na druhú a b na druhú tak, aby ste dostali c na druhú. Alebo použijúc matematickejší prístup:

$$c^{2}=a^{2}+b^{2}$$

A práve jeden ďalší riadok.

Pomocou `\label` a `\ref` sa v texte môžete odvolávať na rovnice.

```
\begin{equation} \label{eq:eps}
\epsilon > 0
\end{equation}
Z~(\ref{eq:eps}) sme dostali
\ldots
```

$$\epsilon > 0 \tag{3.1}$$

Z (3.1) sme dostali ...

Zapamätajte si, že výrazy budú vytlačené rôznym spôsobom, ak budú rôzne zadané:

```
$\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$
```

$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

Medzi *matematickým režimom* a *textovým režimom* sú rozdiely. Napríklad v *matematickom režime*:

1. Väčšina medzier a ukončení riadkov nemá význam, všetky medzery sú buď odvodené z logického obsahu matematických výrazov alebo musia byť zadané pomocou príkazov ako `\,`, `\quad` alebo `\qquad`.

2. Prázdne riadky nie sú dovolené. Iba jeden odsek na jeden výraz.

3. Každé písmeno sa považuje za názov premennej a je vysádzané ako premenná. Ak chcete vo výraze použiť normálny text (normálny vzpriamený font a normálne vzdialenosti písmen), potom musíte použiť na napísanie textu príkaz `\textrm{...}`.

```
\begin{equation}
\forall x \in \mathbf{R}:
\qquad x^{2} \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R}: \qquad x^{2} \geq 0 \tag{3.2}$$

```
\begin{equation}
x^{2} \geq 0\qquad
\textrm{pre všetky }x\in\mathbf{R}
\end{equation}
```

$$x^{2} \geq 0 \qquad \textrm{pre všetky } x\in\mathbf{R} \tag{3.3}$$

Matematici môžu byť veľmi úzkostliví na symboly, ktoré sa používajú: konvenčne by sa tu mal používať font „blackboard bold“, ktorý je dosiahnuteľný príkazom `\mathbb` z balíka `amsfonts` alebo `amssymb`. Posledný príklad by vyzeral

```
\begin{displaymath}
x^{2} \geq 0\qquad
\textrm{pre všetky }x\in\mathbb{R}
\end{displaymath}
```

$$x^{2} \geq 0 \qquad \textrm{pre všetky } x\in\mathbb{R}$$

3.2 Zoskupovanie v matematickom režime

Väčšina príkazov matematického režimu funguje len na nasledujúcom znaku. Teda ak chcete, aby sa daný príkaz týkal viacerých po sebe idúcich znakov, musíte ich uzavrieť do zložených zátvoriek: `{...}`.

```
\begin{equation}
a^x+y \neq a^{x+y}
\end{equation}
```

$$a^x+y \neq a^{x+y} \tag{3.4}$$

3.3 Sádzanie matematických výrazov

V tejto časti budú opísané najdôležitejšie príkazy. Kompletný zoznam všetkých matematických znakov je v podkapitole 3.10 na strane 52.

Malé grécke písmená sa zadávajú ako `\alpha`, `\beta`, `\gamma`, ..., veľké písmená[2] sa zadávajú ako `\Gamma`, `\Delta`, ...

[2]V LaTeXu nie je definovaná veľká Alfa, pretože vyzerá rovnako ako normálne A. Ak dôjde niekedy k zmenám v matematickom kódovaní, zmení sa to.

```
$\lambda,\xi,\pi,\mu,\Phi,\Omega$
```

$\lambda,\xi,\pi,\mu,\Phi,\Omega$

Exponenty a dolné indexy môžu byť zadané použitím znakov ^ a _.

```
$a_{1}$ \qquad $x^{2}$ \qquad
$e^{-\alpha t}$ \qquad
$a^{3}_{ij}$\\
$e^{x^2} \neq {e^x}^2$
```

a_{1} $\qquad$ x^{2} $\qquad$ $e^{-\alpha t}$ $\qquad$ a^{3}_{ij}
$e^{x^2} \neq {e^x}^2$

Druhá odmocnina sa zadáva príkazom `\sqrt`, a n–tá odmocnina pomocou `\sqrt[`n`]`. Veľkosť znaku odmocniny vypočíta LaTeX automaticky. Ak potrebujete len znak odmocniny, je potrebné použiť príkaz `\surd`.

```
$\sqrt{x}$ \qquad
$\sqrt{ x^{2}+\sqrt{y} }$
\qquad $\sqrt[3]{2}$\\[3pt]
$\surd[x^2 + y^2]$
```

$\sqrt{x}$ $\qquad$ $\sqrt{ x^{2}+\sqrt{y} }$ $\qquad$ $\sqrt[3]{2}$
$\surd[x^2 + y^2]$

Príkazy `\overline` a `\underline` vytvoria **horizontálne čiary** priamo nad alebo pod výrazom.

```
$\overline{m+n}$
```

$\overline{m+n}$

Príkazy `\overbrace` a `\underbrace` vytvoria dlhé **horizontálne svorky** nad alebo pod výrazom.

```
$\underbrace{ a+b+\cdots+z }_{26}$
```

$\underbrace{ a+b+\cdots+z }_{26}$

Aby ste mohli pridať k premenným akcenty, respektíve značky, ako, napríklad malé šípky, vlnovky atď., môžete použiť príkazy uvedené v tabuľke 3.1. Dlhé striešky a vlnovky pokrývajúce viacero znakov sa generujú pomocou príkazov `\widetilde` a `\widehat`. Symbol `'` dáva čiarku (prime).

```
\begin{displaymath}
y=x^{2}\qquad y^\prime=2x\qquad y''=2
\end{displaymath}
```

$$y=x^{2}\qquad y^\prime=2x\qquad y''=2$$

Vektory sú často označované pridaním malého znaku šípky nad premennú. Toto sa robí pomocou príkazu `\vec`. Na vyznačenie vektora z bodu A do bodu B sú užitočné dva príkazy `\overrightarrow` a `\overleftarrow`.

```
\begin{displaymath}
\vec a\quad\overrightarrow{AB}
\end{displaymath}
```

$$\vec a \quad \overrightarrow{AB}$$

Názvy funkcií (napríklad logaritmických) sa často sádzajú zvislým fontom (antikvou) a nie šikmým (kurzívou) ako premenné. Preto LaTeX obsahuje nasledujúce príkazy na sádzanie názvov najdôležitelších funkcií:

`\arccos`	`\cos`	`\csc`	`\exp`	`\ker`	`\limsup`	`\min`	`\sinh`
`\arcsin`	`\cosh`	`\deg`	`\gcd`	`\lg`	`\ln`	`\Pr`	`\sup`
`\arctan`	`\cot`	`\det`	`\hom`	`\lim`	`\log`	`\sec`	`\tan`
`\arg`	`\coth`	`\dim`	`\inf`	`\liminf`	`\max`	`\sin`	`\tanh`

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x}=1$$

Pre funkciu modulo existujú dva príkazy: `\bmod` pre binárny operátor „$a \bmod b$“ a `\pmod` pre výrazy ako napríklad „$x \equiv a \pmod b$“.

Zlomky sa sádzajú pomocou príkazu `\frac{...}{...}`. Často sa uprednostňuje tvar zlomkov ako 1/2 (najmä v textovom režime), pretože to vyzerá lepšie, ak zlomky neobsahujú veľké výrazy.

```
$1\frac{1}{2}$~hodiny
\begin{displaymath}
\frac{ x^{2} }{ k+1 }\qquad
x^{ \frac{2}{k+1} }\qquad
x^{ 1/2 }
\end{displaymath}
```

$1\frac{1}{2}$ hodiny

$$\frac{ x^{2} }{ k+1 }\qquad x^{ \frac{2}{k+1} }\qquad x^{ 1/2 }$$

Na napísanie binomických koeficientov alebo podobných štruktúr, môžete použiť príkaz `{... \choose ...}` alebo `{... \atop ...}`. Príkaz spomínaný ako druhý vytvorí rovnaký výstup ako prvý, avšak bez zátvoriek. (Upozornime, že použitie týchto príkazov starého štýlu je výslovne zakázané v balíku amsmath. Sú nahradené príkazmi `\binom` a `\genfrac`. Ten druhý zahŕňa všetky odpovedajúce konštrukcie, napríklad môžete dostať pomocou `\newcommand{\newatop}[2]{\genfrac{}{}{0pt}{1}{#1}{#2}}` niečo podobné ako `\atop`.)

```
\begin{displaymath}
{n \choose k}\qquad {x \atop y+2}
\end{displaymath}
```

$$\binom{n}{k}\qquad \genfrac{}{}{0pt}{}{x}{y+2}$$

Pre binárne relácie môže byť užitočné umiestnenie symbolov jeden nad druhým. `\stackrel` umiestni symbol daný ako prvý argument vo veľkosti indexu nad druhý symbol, ktorý bude stáť na svojej obvyklej pozícii.

```
\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}
```

$$\int f_N(x) \stackrel{!}{=} 1$$

Znak integrálu sa vytvára pomocou `\int` a **znak sumy** pomocou príkazu `\sum`. Horné a dolné hranice[3] sa zadávajú pomocou `^` a `_`, ako dolný a horný index.[4]

```
\begin{displaymath}
{\textstyle\sum_{i=1}^{n}} \qquad
\sum_{i=1}^{n} \qquad
\int_{0}^{\frac{\pi}{2}} \qquad
\int\limits_{0}^{\frac{\pi}{2}}
\end{displaymath}
```

$${\textstyle\sum_{i=1}^{n}} \qquad \sum_{i=1}^{n} \qquad \int_{0}^{\frac{\pi}{2}} \qquad \int\limits_{0}^{\frac{\pi}{2}}$$

Pre **zátvorky** a ostatné oddeľovače existujú v TEXu všetky typy znakov (napríklad $[\ \langle\ \|\ \updownarrow$). Okrúhle a hranaté zátvorky môžu byť priamo zadané z klávesnice pomocou príslušných klávesov, zložené zátvorky pomocou príkazu `\{`, všetky ostatné oddeľovače sú vytvárané pomocou špeciálnych príkazov (napríklad príkazu `\updownarrow`). Zoznam všetkých oddeľovačov je v tabuľke 3.8 na strane 54.

```
\begin{displaymath}
{a,b,c}\neq\{a,b,c\}
\end{displaymath}
```

$${a,b,c}\neq\{a,b,c\}$$

Pokiaľ zadáte pred otvárací oddeľovač príkaz `\left` a pred uzatvárací oddeľovač príkaz `\right`, TEX automaticky vypočíta správnu veľkosť oddeľovača. Zapamätajte si, že každé `\left` musíte uzatvoriť príslušným `\right`. Ak nechcete napravo nič, použite neviditeľné „`\right.`“!

```
\begin{displaymath}
1 + \left( \frac{1}{ 1-x^{2} }
    \right) ^3
\end{displaymath}
```

$$1 + \left(\frac{1}{ 1-x^{2} } \right) ^3$$

V niektorých prípadoch je nevyhnutné zadať správnu veľkosť matematického oddeľovača ručne. Na to môžete používať príkazy `\big`, `\Big`, `\bigg`

[3] $\mathcal{AMS}$-LATEX má navyše viacriadkové horné a dolné indexy.

[4] Príkazy `\textstyle`, `\displaystyle` a `\limits` menia spôsob ich umiestnenia (pozn. prekl.).

a `\Bigg` ako predpony ku väčšine oddeľovačov.[5]

```
$\Big( (x+1) (x-1) \Big) ^{2}$\\
$\big(\Big(\bigg(\Bigg($\quad
$\big\}\Big\}\bigg\}\Bigg\}$\quad
$\big\|\Big\|\bigg\|\Bigg\|$
```

$\Big((x+1) (x-1) \Big) ^{2}$

$\big(\Big(\bigg(\Bigg($ $\quad$ $\big\}\Big\}\bigg\}\Bigg\}$ $\quad$ $\big\|\Big\|\bigg\|\Bigg\|$

Na vloženie **troch bodiek** do vzorca môžete použiť niekoľko príkazov. `\ldots` vysádza bodky v riadku, `\cdots` vycentrované. Okrem toho ešte existujú príkazy `\vdots` pre vertikálne (zvislé) a `\ddots` pre diagonálne bodky. V časti 3.5 môžete nájsť ďalšie príklady.

```
\begin{displaymath}
x_{1},\ldots,x_{n} \qquad
x_{1}+\cdots+x_{n}
\end{displaymath}
```

$$x_{1},\ldots,x_{n} \qquad x_{1}+\cdots+x_{n}$$

3.4 Medzery v matematike

Pokiaľ medzery medzi vzorcami (vo vzorcoch), zvolené TEXom, nie sú uspokojujúce, môžu byť upravené vložením špeciálnych príkazov pre medzery. Najdôležitejšie sú: `\,` na maličké medzery $\frac{3}{18}$ quad (␣), `\:` na $\frac{4}{18}$ quad (␣) a `\;` na $\frac{5}{18}$ quad (␣). `\␣` na stredne veľké medzery (␣ je znakom „medzery“). `\quad` (␣) a `\qquad` (␣) vytvárajú veľké medzery. Veľkosť `\quad` zodpovedá šírke písmena ‚M‘ aktuálneho fontu. Príkaz `\!` vytvára zápornú medzeru (posun, priestor) $-\frac{3}{18}$ quad (␣).

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int_{D} g(x,y)
  \, \ud x\, \ud y
\end{displaymath}
namiesto
\begin{displaymath}
\int\int_{D} g(x,y)\ud x \ud y
\end{displaymath}
```

$$\int\!\!\!\int_{D} g(x,y) \, \mathrm{d} x\, \mathrm{d} y$$

namiesto

$$\int\int_{D} g(x,y)\mathrm{d} x \mathrm{d} y$$

Všimnime si, že ‚d‘ v diferenciále sa zvykne písať antikvou – typom roman.

[5] Tieto príkazy nefungujú ako sa od nich očakáva, ak bol použitý príkaz na zmenu veľkosti písma alebo bola zvolená možnosť `11pt` alebo `12pt`. Na opravenie tohoto správania použite balík exscale alebo amsmath.

$\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX poskytuje ďalšie cesty na jemnú úpravu medzier medzi znakmi integrovania, konkrétne príkazy `\iint`, `\iiint`, `\iiiint` a `\idotsint`. Pri natiahnutom balíku `amsmath` môže byť vyššie uvedený príklad vysádzaný nasledujúcim spôsobom:

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint_{D} \, \ud x \, \ud y
\end{displaymath}
```

$$\iint_{D} \, \mathrm{d} x \, \mathrm{d} y$$

Ďalšie podrobnosti nájdete v elektronickom dokumente testmath.tex (rozširovaný spolu s $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeXom) alebo v Kapitole 8 *The LaTeX Companion* [3].

3.5 Vertikálne zarovnaný materiál

Na vysádzanie **matíc** sa používa prostredie `array`. Pracuje podobne ako prostredie `tabular`. Príkaz `\\` sa používa na ukončenie riadkov.

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

$$\mathbf{X} = \left(\begin{array}{ccc} x_{11} & x_{12} & \ldots \\ x_{21} & x_{22} & \ldots \\ \vdots & \vdots & \ddots \end{array} \right)$$

Prostredie `array` sa môže používať aj na sádzanie výrazov, ktoré majú len jeden veľký oddeľovač. Namiesto druhého sa použije znak bodky („`.`“), ako neviditeľný pravý (`\right`) oddeľovač:

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
 a & \textrm{ak $d>c$}\\
 b+x & \textrm{ráno}\\
 l & \textrm{počas celého dňa}
  \end{array} \right.
\end{displaymath}
```

$$y = \left\{ \begin{array}{ll} a & \textrm{ak } d>c \\ b+x & \textrm{ráno} \\ l & \textrm{počas celého dňa} \end{array} \right.$$

Rovnako, ako vnútri prostredia `tabular`, môžete kresliť čiary aj v prostredí `array`, napríklad na oddelenie prvkov matice:

```
\begin{displaymath}
\left(\begin{array}{c|c}
 1 & 2 \\
\hline
3 & 4
\end{array}\right)
\end{displaymath}
```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}\right)$$

Na sádzanie vzorcov, ktoré dĺžkou prekračujú hranicu jedného riadku alebo na sádzanie sústav rovníc môžete namiesto prostredia `equation` použiť prostredia `eqnarray` a `eqnarray*`. V prostredí `eqnarray` je každý riadok automaticky číslovaný. V prostredí `eqnarray*` riadky nie sú číslované.

Prostredia `eqnarray` a `eqnarray*` pracujú ako 3-stĺpcová tabuľka tvaru `{rcl}`, kde sa stredný stĺpec používa pre znak rovnosti alebo nerovnosti alebo iný znak, ktorý budete považovať za vhodný. Príkaz `\\` ukončuje („láme“) riadky.

```
\begin{eqnarray}
f(x) & = & \cos x     \\
f'(x) & = & -\sin x   \\
\int_{0}^{x} f(y)\mathrm{d}y &
= & \sin x
\end{eqnarray}
```

$$\begin{array}{rcll} f(x) & = & \cos x & (3.5) \\ f'(x) & = & -\sin x & (3.6) \\ \int_{0}^{x} f(y)\mathrm{d}y & = & \sin x & (3.7) \end{array}$$

Všimnite si, že na oboch stranách stredného stĺpca, znaku rovnosti, je veľa voľného miesta. Ako uvidíte v nasledujúcom príklade, toto voľné miesto môže byť zmenšené príkazom `\setlength\arraycolsep{2pt}`.

Dlhé rovnice nie sú automaticky rozdeľované na pekné časti. Autor musí sám určiť, kde sa majú rozdeliť a nakoľko ich treba odsadiť. Na dosiahnutie tohoto sa najčastejšie používajú nasledujúce dve metódy:

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x & = & x -\frac{x^{3}}{3!}
     +\frac{x^{5}}{5!}-{}
                    \nonumber\\
 & & {}-\frac{x^{7}}{7!}+{}\cdots
\end{eqnarray}}
```

$$\begin{array}{rcll} \sin x & = & x - \frac{x^{3}}{3!} + \frac{x^{5}}{5!} - & \\ & & {}- \frac{x^{7}}{7!} + \cdots & (3.8) \end{array}$$

```
\begin{eqnarray}
\lefteqn{ \cos x = 1
     -\frac{x^{2}}{2!} +{} }
                  \nonumber\\
 & & {}+\frac{x^{4}}{4!}
     -\frac{x^{6}}{6!}+{}\cdots
\end{eqnarray}
```

$$\begin{aligned} &\cos x = 1 - \frac{x^2}{2!} + \\ &\qquad + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \end{aligned} \tag{3.9}$$

Príkaz `\nonumber` spôsobí, že LaTeX danú rovnicu neočísluje.

Môže byť náročné vytvoriť pomocou týchto metód vertikálne zarovnané rovnice, ktoré vyzerajú dobre. Balík amsmath poskytuje viac možností na riešenie tohoto problému (pozri prostredia `split` a `align`).

3.6 Duch

Nemôžeme vidieť duchov (fantómov), ale oni ešte stále obývajú určité miesto v mysliach mnohých ľudí. LaTeX nie je iný. Túto okolnosť môžeme použiť na niektoré zaujímavé triky s medzerami.

Pri vertikálnom zarovnávaní textu pomocou `^` a `_` môže byť niekedy LaTeX tiež trochu osožný. Použitím príkazu `\phantom` môžete vymedziť priestor na písmená, ktoré nebudú v konečnom výstupe ukázané. Najlepšie to je vidieť na nasledujúcich príkladoch.

```
\begin{displaymath}
{}^{12}_{\phantom{1}6}\textrm{C}
\qquad \textrm{namiesto} \qquad
{}^{12}_{6}\textrm{C}
\end{displaymath}
```

$${}^{12}_{6}\textrm{C} \qquad \textrm{namiesto} \qquad {}^{12}_{6}\textrm{C}$$

```
\begin{displaymath}
\Gamma_{ij}^{\phantom{ij}k}
\qquad \textrm{namiesto} \qquad
\Gamma_{ij}^{k}
\end{displaymath}
```

$$\Gamma_{ij}^{k} \qquad \textrm{namiesto} \qquad \Gamma_{ij}^{k}$$

3.7 Veľkosť matematického fontu

V matematickom režime volí TeX veľkosť fontu v závislosti od kontextu. Napríklad horné indexy sú sádzané menším fontom. Ak by ste chceli napísať v rovnici text fontom roman a použili by ste príkaz `\textrm`, mechanizmus na zmenu veľkosti fontu by nefungoval pretože príkaz `\textrm` dočasne prešiel do textového režimu. Namiesto príkazu `\textrm` musíte použiť príkaz `\mathrm`, aby mechanizmus na menenie veľkostí fontov fungoval aj naďalej.

Ale dávajte si pozor, `\mathrm` funguje dobre len na malých úsekoch. Medzery stále nie sú považované za znaky, negunguje ani diakritika.[6]

```
\begin{equation}
2^{\textrm{nd}} \quad
2^\mathrm{nd}
\end{equation}
```

$$2^{\textrm{nd}} \quad 2^\mathrm{nd} \tag{3.10}$$

Niekedy napriek tomu, že LaTeX vie vypočítať správnu veľkosť fontov, potrebujete mu zadať správnu veľkosť. V matematickom režime sa veľkosť písma zadáva pomocou štyroch príkazov:

`\displaystyle` $(\displaystyle 123)$, `\textstyle` $(\textstyle 123)$, `\scriptstyle` $(\scriptstyle 123)$ and `\scriptscriptstyle` $(\scriptscriptstyle 123)$.

Menenie štýlov ovplyvňuje tiež spôsob, akým sa zobrazujú hranice.

```
\begin{displaymath}
\mathop{\mathrm{corr}}(X,Y)=
 \frac{\displaystyle
   \sum_{i=1}^n(x_i-\overline x)
   (y_i-\overline y)}
  {\displaystyle\biggl[
 \sum_{i=1}^n(x_i-\overline x)^2
\sum_{i=1}^n(y_i-\overline y)^2
\biggr]^{1/2}}
\end{displaymath}
```

$$\mathop{\mathrm{corr}}(X,Y)= \frac{\displaystyle \sum_{i=1}^n(x_i-\overline x) (y_i-\overline y)} {\displaystyle\biggl[\sum_{i=1}^n(x_i-\overline x)^2 \sum_{i=1}^n(y_i-\overline y)^2 \biggr]^{1/2}}$$

Toto je jeden z tých príkladov, keď potrebujeme väčšie hranaté zátvorky, ako nám poskytujú príkazy `\left[ \right]`.

3.8 Vety, zákony, ...

Keď budete písať matematické dokumenty, pravdepodobne budete potrebovať spôsob, ako napísať „lemy“, „definície“, „axiómy“ a podobné štruktúry. LaTeX toto zabezpečuje príkazom

`\newtheorem{`*meno*`}[`*čítač*`]{`*text*`}[`*oddiel*`]`

Argument *meno* je krátke kľúčové slovo, ktoré sa používa na označenie „teorémy“ v texte. Pomocou argumentu *text* definujete názov „teorémy“, ktorý sa objaví vo výslednom dokumente.

[6] Balík $\mathcal{A}\mkern-5mu\mathcal{M}\mkern-3mu\mathcal{S}$-LaTeX pri práci s príkazom `\textrm` rieši tieto problémy fontov.

Argumenty v hranatých zátvorkách sú nepovinné. Oba sa používajú na bližšie určenie číslovania, použitého pri danej „teoréme“. Argumentom *čítač* môžete bližšie určiť *meno* skôr uvedenej „teorémy“. Nová „teoréma“ tak bude číslovaná v nadväznosti na skôr uvedenú „teorému“. Argument *oddiel* vám dovoľuje určiť oddiel, v ktorom chcete, aby bola vaša „teoréma“ číslovaná.

Po zadaní príkazu `\newtheorem` do hlavičky vášho dokumentu môžete vo vašom dokumente použiť nasledujúce príkazy.

```
\begin{názov}[text]
Toto je moja zaujímavá veta
\end{názov}
```

Dosť bolo teórie. Nasledujúce príklady by, dúfajme, mali odstrániť aj posledné zvyšky pochybností a jasne ukázať, že prostredie `\newtheorem` je príliš zložitá cesta na pochopenie.

```
% deifnície pre dokument
% hlavička
\newtheorem{zakon}{Zákon}
\newtheorem{porota}[zakon]{Porota}
%v dokumente
\begin{zakon} \label{zakon:box}
Neskrývajte sa v~svedkovom boxe.
\end{zakon}
\begin{porota}[Dvanásti]
Môžete to byť vy! Takže si dávajte
pozor a pozrite zákon
\ref{zakon:box}.\end{porota}
\begin{zakon}Nie, Nie, Nie\end{zakon}
```

Zákon 1 *Neskrývajte sa v svedkovom boxe.*

Porota 2 (Dvanásti) *Môžete to byť vy! Takže si dávajte pozor a pozrite zákon 1.*

Zákon 3 *Nie, Nie, Nie*

Teoréma „porota“ používa rovnaké číslovanie, ako teoréma „zákon“. Preto dostane poradové číslo, ktoré nasleduje za uvedenými teorémami „zákon“. Argument v hranatých zátvorkách sa používa na bližšie určenie názvu alebo niečoho podobného danej teoréme.

```
\newtheorem{mur}{Murphy}[section]
\begin{mur}
Ak existujú dva alebo
viac spôsobov, ako niečo
urobiť a jeden z~týchto
spôsobov môže spôsobiť
katastrofu, potom si niekto
tento spôsob vyberie.\end{mur}
```

Murphy 3.8.1 *Ak existujú dva alebo viac spôsobov, ako niečo urobiť a jeden z týchto spôsobov môže spôsobiť katastrofu, potom si niekto tento spôsob vyberie.*

„Murphyho“ veta dostane číslo, ktoré sa viaže na číslo aktuálneho oddielu. Môžete použiť aj iné celky, ako napríklad kapitola (chapter) alebo pododdiel (subsection).

3.9 Tučné znaky

V LaTeXu je pomerne zložité dosiahnuť tučné znaky: je to tak asi úmyselne, pretože amatérski sadzači ich zvyknú nadmerne používať. Príkaz na zmenu fontov `\mathbf` síce dáva tučné písmo, ale to je roman (vzpriamené), kým matematické symboly sú obyčajne italic (naklonené). Existuje príkaz `\boldmath`, ale *funguje iba mimo matematického režimu* [7].

```
\begin{displaymath}
\mu, M \qquad \mathbf{M} \qquad
\mbox{\boldmath $\mu, M$}
\end{displaymath}
```

$$\mu, M \qquad \mathbf{M} \qquad \boldsymbol{\mu, M}$$

Všimnite si, že čiarka je tiež tučná, čo nemusí byť žiaduce.

Balík amsbsy (obsiahnutý v amsmath) toto veľmi zjednodušuje. Zahŕňa príkazy `\boldsymbol` a „poor man's bold“ `\pmb`, ktorého účinok je zvlášť významný u systémov, ktoré nemajú fonty potrebné pre tučné symboly.

```
\begin{displaymath}
\mu, M \qquad
\boldsymbol{\mu}, \boldsymbol{M}
\qquad \pmb{\mu}, \pmb{M}
\end{displaymath}
```

$$\mu, M \qquad \boldsymbol{\mu}, \boldsymbol{M} \qquad \pmb{\mu}, \pmb{M}$$

[7] Príkaz `\mathbf` funguje aj pre znaky.

3.10 Zoznam matematických symbolov

V nasledujúcich tabuľkách nájdete zoznam všetkých symbolov bežne dostupných v *matematickom režime.*

Aby ste mohli použiť symboly uvedené v tabuľkách 3.12–3.16[8], musíte v hlavičke zdrojového súboru natiahnuť balík amssymb a musíte mať vo svojom systéme nainštalované matematické $\mathcal{AMS}$ fonty. V prípade, že uvedený balík ani fonty vo vašom systéme nainštalované nemáte, sú dostupné na adrese `CTAN:/tex-archive/macros/latex/packages/amslatex`

Tabuľka 3.1: Diakritika v matematickom režime

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`	$\acute{a}$	`\acute{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`	$\widehat{A}$	`\widehat{A}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\breve{a}$	`\breve{a}`	$\widetilde{A}$	`\widetilde{A}`

Tabuľka 3.2: Malé grécke písmená

α	`\alpha`	θ	`\theta`	o	`o`	υ	`\upsilon`
β	`\beta`	ϑ	`\vartheta`	π	`\pi`	ϕ	`\phi`
γ	`\gamma`	ι	`\iota`	ϖ	`\varpi`	φ	`\varphi`
δ	`\delta`	κ	`\kappa`	ρ	`\rho`	χ	`\chi`
ϵ	`\epsilon`	λ	`\lambda`	ϱ	`\varrho`	ψ	`\psi`
ε	`\varepsilon`	μ	`\mu`	σ	`\sigma`	ω	`\omega`
ζ	`\zeta`	ν	`\nu`	ς	`\varsigma`		
η	`\eta`	ξ	`\xi`	τ	`\tau`		

Tabuľka 3.3: Veľké grécke písmená

Γ	`\Gamma`	Λ	`\Lambda`	Σ	`\Sigma`	Ψ	`\Psi`
Δ	`\Delta`	Ξ	`\Xi`	Υ	`\Upsilon`	Ω	`\Omega`
Θ	`\Theta`	Π	`\Pi`	Φ	`\Phi`		

[8]Tieto tabuľky boli odvodené zo súboru `symbols.tex`, vytvoreného Davidom Carlisleom a neskôr zmeneného a doplneného na návrh Jozefa Tkadleca.

Tabuľka 3.4: Binárne relácie

Príslušné negácie môžete vytvoriť pridaním príkazu `\not` ako predpony pred nasledujúce symboly.

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq` aj `\le`	$\geq$	`\geq` aj `\ge`	$\equiv$	`\equiv`
$\ll$	`\ll`	$\gg$	`\gg`	$\doteq$	`\doteq`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubset$	`\sqsubset` [a]	$\in$	`\in`	$\Join$	`\Join` [a]
$\sqsubseteq$	`\sqsubseteq`	$\vdash$	`\vdash`	$\bowtie$	`\bowtie`
$\sqsupset$	`\sqsupset` [a]	$\ni$	`\ni` , `\owns`	$\propto$	`\propto`
$\sqsupseteq$	`\sqsupseteq`	$\dashv$	`\dashv`	$\models$	`\models`
$\mid$	`\mid`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\smile$	`\smile`	$\frown$	`\frown`	$\asymp$	`\asymp`
$:$	`:`	$\notin$	`\notin`	$\neq$	`\neq` or `\ne`

[a] Na dosiahnutie tohto symbolu použite balík latexsym

Tabuľka 3.5: Binárne operátory

$+$	`+`	$-$	`-`		
$\pm$	`\pm`	$\mp$	`\mp`	$\triangleleft$	`\triangleleft`
$\cdot$	`\cdot`	$\div$	`\div`	$\triangleright$	`\triangleright`
$\times$	`\times`	$\setminus$	`\setminus`	$\star$	`\star`
$\cup$	`\cup`	$\cap$	`\cap`	$\ast$	`\ast`
$\sqcup$	`\sqcup`	$\sqcap$	`\sqcap`	$\circ$	`\circ`
$\vee$	`\vee` , `\lor`	$\wedge$	`\wedge` , `\land`	$\bullet$	`\bullet`
$\oplus$	`\oplus`	$\ominus$	`\ominus`	$\diamond$	`\diamond`
$\odot$	`\odot`	$\oslash$	`\oslash`	$\uplus$	`\uplus`
$\otimes$	`\otimes`	$\bigcirc$	`\bigcirc`	$\amalg$	`\amalg`
$\bigtriangleup$	`\bigtriangleup`	$\bigtriangledown$	`\bigtriangledown`	$\dagger$	`\dagger`
$\lhd$	`\lhd` [a]	$\rhd$	`\rhd` [a]	$\ddagger$	`\ddagger`
$\unlhd$	`\unlhd` [a]	$\unrhd$	`\unrhd` [a]	$\wr$	`\wr`

Tabuľka 3.6: Veľké operátory

$\sum$	`\sum`	$\bigcup$	`\bigcup`	$\bigvee$	`\bigvee`	$\bigoplus$	`\bigoplus`
$\prod$	`\prod`	$\bigcap$	`\bigcap`	$\bigwedge$	`\bigwedge`	$\bigotimes$	`\bigotimes`
$\coprod$	`\coprod`	$\bigsqcup$	`\bigsqcup`			$\bigodot$	`\bigodot`
$\int$	`\int`	$\oint$	`\oint`			$\biguplus$	`\biguplus`

Tabuľka 3.7: Šípky

$\leftarrow$	`\leftarrow` aj `\gets`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\rightarrow$	`\rightarrow` aj `\to`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\longleftrightarrow$	`\longleftrightarrow`	$\leftrightarrow$	`\leftrightarrow`	$\updownarrow$	`\updownarrow`
$\Longleftarrow$	`\Longleftarrow`	$\Leftarrow$	`\Leftarrow`	$\Uparrow$	`\Uparrow`
$\Longrightarrow$	`\Longrightarrow`	$\Rightarrow$	`\Rightarrow`	$\Downarrow$	`\Downarrow`
$\Longleftrightarrow$	`\Longleftrightarrow`	$\Leftrightarrow$	`\Leftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\iff$	`\iff` (väčšie medzery)	$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\rightharpoondown$	`\rightharpoondown`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\leftharpoonup$	`\leftharpoonup`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`	$\leadsto$	`\leadsto` [a]	$\nearrow$	`\nearrow`

[a] Na dosiahnutie tohto symbolu použite balík latexsym

Tabuľka 3.8: Oddeľovače

$)$	`)`	$\uparrow$	`\uparrow`	$\Uparrow$	`\Uparrow`
$]$	`]` aj `\rbrack`	$\downarrow$	`\downarrow`	$\Downarrow$	`\Downarrow`
$\}$	`\}` aj `\rbrace`	$\updownarrow$	`\updownarrow`	$\Updownarrow$	`\Updownarrow`
$\rangle$	`\rangle`	$\vert$	`|` aj `\vert`	$\Vert$	`|` aj `\Vert`
$\rfloor$	`\rfloor`	$\lceil$	`\lceil`	$\rceil$	`\rceil`
$($	`(`	$[$	`[` aj `\lbrack`	$\{$	`\{` aj `\lbrace`
$\langle$	`\langle`	$\lfloor$	`\lfloor`	$/$	`/`
$\backslash$	`\backslash`		. (prázdny doplnok)		

Tabuľka 3.9: Veľké oddeľovače

$\lgroup$	`\lgroup`	$\rmoustache$	`\rmoustache`	$\lmoustache$	`\lmoustache`	$\bracevert$	`\bracevert`
$\rgroup$	`\rgroup`	$\arrowvert$	`\arrowvert`	$\Arrowvert$	`\Arrowvert`		

Tabuľka 3.10: Rôzne znaky

$\dots$	\dots	$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots
$\hbar$	\hbar	$\imath$	\imath	$\jmath$	\jmath	ℓ	\ell
$\Re$	\Re	$\Im$	\Im	$\aleph$	\aleph	$\wp$	\wp
$\forall$	\forall	$\exists$	\exists	$\mho$	\mho [a]	$\angle$	\angle
$\spadesuit$	\spadesuit	$\emptyset$	\emptyset	$\Diamond$	\Diamond [a]	$\prime$	\prime
$\clubsuit$	\clubsuit	$\triangle$	\triangle	$\Box$	\Box [a]	$'$	'
$\heartsuit$	\heartsuit	$\top$	\top	∂	\partial	$\surd$	\surd
$\diamondsuit$	\diamondsuit	$\bot$	\bot	∇	\nabla	∞	\infty
$\neg$	\neg aj \lnot	$\flat$	\flat	$\natural$	\natural	$\sharp$	\sharp

[a] Na dosiahnutie tohto symbolu použite balík latexsym

Tabuľka 3.11: Nematematické znaky

Tieto znaky sa môžu používať aj v textovom režime.

†	\dag	§	\S	©	\copyright
‡	\ddag	¶	\P	£	\pounds

Tabuľka 3.12: $\mathcal{AMS}$ oddeľovače

$\ulcorner$	\ulcorner	$\urcorner$	\urcorner	$\llcorner$	\llcorner	$\lrcorner$	\lrcorner

Tabuľka 3.13: $\mathcal{AMS}$ grécke a hebrejské znaky

$\digamma$	\digamma	$\varkappa$	\varkappa	$\beth$	\beth	$\daleth$	\daleth	$\gimel$	\gimel

Tabuľka 3.14: $\mathcal{AMS}$ binárne relácie

$\doteqdot$	`\doteqdot` aj `\Doteq`	$\lessdot$	`\lessdot`	$\gtrdot$	`\gtrdot`
$\leqslant$	`\leqslant`	$\geqslant$	`\geqslant`	$\risingdotseq$	`\risingdotseq`
$\eqslantless$	`\eqslantless`	$\eqslantgtr$	`\eqslantgtr`	$\fallingdotseq$	`\fallingdotseq`
$\leqq$	`\leqq`	$\geqq$	`\geqq`	$\eqcirc$	`\eqcirc`
$\lll$	`\lll` aj `\llless`	$\circeq$	`\circeq`	$\lesssim$	`\lesssim`
$\ggg$	`\ggg` aj `\gggtr`	$\gtrsim$	`\gtrsim`	$\triangleq$	`\triangleq`
$\lessapprox$	`\lessapprox`	$\gtrapprox$	`\gtrapprox`	$\bumpeq$	`\bumpeq`
$\lessgtr$	`\lessgtr`	$\gtrless$	`\gtrless`	$\Bumpeq$	`\Bumpeq`
$\lesseqgtr$	`\lesseqgtr`	$\gtreqless$	`\gtreqless`	$\thicksim$	`\thicksim`
$\lesseqqgtr$	`\lesseqqgtr`	$\gtreqqless$	`\gtreqqless`	$\thickapprox$	`\thickapprox`
$\preccurlyeq$	`\preccurlyeq`	$\succcurlyeq$	`\succcurlyeq`	$\approxeq$	`\approxeq`
$\curlyeqprec$	`\curlyeqprec`	$\curlyeqsucc$	`\curlyeqsucc`	$\backsim$	`\backsim`
$\precsim$	`\precsim`	$\succsim$	`\succsim`	$\backsimeq$	`\backsimeq`
$\precapprox$	`\precapprox`	$\succapprox$	`\succapprox`	$\vDash$	`\vDash`
$\subseteqq$	`\subseteqq`	$\supseteqq$	`\supseteqq`	$\Vdash$	`\Vdash`
$\Subset$	`\Subset`	$\Supset$	`\Supset`	$\Vvdash$	`\Vvdash`
$\blacktriangleleft$	`\blacktriangleleft`	$\sqsubset$	`\sqsubset`	$\sqsupset$	`\sqsupset`
$\blacktriangleright$	`\blacktriangleright`	$\because$	`\because`	$\varpropto$	`\varpropto`
$\vartriangleright$	`\vartriangleright`	$\shortparallel$	`\shortparallel`	$\between$	`\between`
$\trianglerighteq$	`\trianglerighteq`	$\smallfrown$	`\smallfrown`	$\pitchfork$	`\pitchfork`
$\vartriangleleft$	`\vartriangleleft`	$\shortmid$	`\shortmid`	$\backepsilon$	`\backepsilon`
$\trianglelefteq$	`\trianglelefteq`	$\smallsmile$	`\smallsmile`	$\therefore$	`\therefore`

Tabuľka 3.15: $\mathcal{AMS}$ šípky

$\dashleftarrow$	`\dashleftarrow`	$\dashrightarrow$	`\dashrightarrow`	$\multimap$	`\multimap`
$\leftleftarrows$	`\leftleftarrows`	$\rightrightarrows$	`\rightrightarrows`	$\upuparrows$	`\upuparrows`
$\leftrightarrows$	`\leftrightarrows`	$\rightleftarrows$	`\rightleftarrows`	$\downdownarrows$	`\downdownarrows`
$\Lleftarrow$	`\Lleftarrow`	$\Rrightarrow$	`\Rrightarrow`	$\upharpoonleft$	`\upharpoonleft`
$\twoheadleftarrow$	`\twoheadleftarrow`	$\twoheadrightarrow$	`\twoheadrightarrow`	$\upharpoonright$	`\upharpoonright`
$\leftarrowtail$	`\leftarrowtail`	$\rightarrowtail$	`\rightarrowtail`	$\downharpoonleft$	`\downharpoonleft`
$\leftrightharpoons$	`\leftrightharpoons`	$\rightleftharpoons$	`\rightleftharpoons`	$\downharpoonright$	`\downharpoonright`
$\leftrightsquigarrow$	`\leftrightsquigarrow`	$\rightsquigarrow$	`\rightsquigarrow`	$\Lsh$	`\Lsh`
$\looparrowleft$	`\looparrowleft`	$\looparrowright$	`\looparrowright`	$\Rsh$	`\Rsh`
$\curvearrowleft$	`\curvearrowleft`	$\curvearrowright$	`\curvearrowright`		
$\circlearrowleft$	`\circlearrowleft`	$\circlearrowright$	`\circlearrowright`		

Tabuľka 3.16: $\mathcal{AMS}$ negované binárne relácie a šípky

$\nless$	\nless	$\ngtr$	\ngtr	$\varsubsetneqq$	\varsubsetneqq
$\lneq$	\lneq	$\gneq$	\gneq	$\varsupsetneqq$	\varsupsetneqq
$\nleq$	\nleq	$\ngeq$	\ngeq	$\nsubseteqq$	\nsubseteqq
$\nleqslant$	\nleqslant	$\ngeqslant$	\ngeqslant	$\nsupseteqq$	\nsupseteqq
$\lneqq$	\lneqq	$\gneqq$	\gneqq	$\nmid$	\nmid
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\nparallel$	\nparallel
$\nleqq$	\nleqq	$\ngeqq$	\ngeqq	$\nshortmid$	\nshortmid
$\lnsim$	\lnsim	$\gnsim$	\gnsim	$\nshortparallel$	\nshortparallel
$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox	$\nsim$	\nsim
$\nprec$	\nprec	$\nsucc$	\nsucc	$\ncong$	\ncong
$\npreceq$	\npreceq	$\nsucceq$	\nsucceq	$\nvdash$	\nvdash
$\precneqq$	\precneqq	$\succneqq$	\succneqq	$\nvDash$	\nvDash
$\precnsim$	\precnsim	$\succnsim$	\succnsim	$\nVdash$	\nVdash
$\precnapprox$	\precnapprox	$\succnapprox$	\succnapprox	$\nVDash$	\nVDash
$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq	$\ntriangleleft$	\ntriangleleft
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq	$\ntriangleright$	\ntriangleright
$\nsubseteq$	\nsubseteq	$\nsupseteq$	\nsupseteq	$\ntrianglelefteq$	\ntrianglelefteq
$\subsetneqq$	\subsetneqq	$\supsetneqq$	\supsetneqq	$\ntrianglerighteq$	\ntrianglerighteq
$\nleftarrow$	\nleftarrow	$\nrightarrow$	\nrightarrow	$\nleftrightarrow$	\nleftrightarrow
$\nLeftarrow$	\nLeftarrow	$\nRightarrow$	\nRightarrow	$\nLeftrightarrow$	\nLeftrightarrow

Tabuľka 3.17: $\mathcal{AMS}$ binárne operátory

$\dotplus$	\dotplus	$\centerdot$	\centerdot	$\intercal$	\intercal
$\ltimes$	\ltimes	$\rtimes$	\rtimes	$\divideontimes$	\divideontimes
$\barwedge$	\barwedge	$\Cap$	\Cap aj \doublecap	$\boxminus$	\boxminus
$\veebar$	\veebar	$\Cup$	\Cup aj \doublecup	$\boxdot$	\boxdot
$\boxplus$	\boxplus	$\smallsetminus$	\smallsetminus	$\circleddash$	\circleddash
$\boxtimes$	\boxtimes	$\doublebarwedge$	\doublebarwedge	$\circledcirc$	\circledcirc
$\leftthreetimes$	\leftthreetimes	$\rightthreetimes$	\rightthreetimes	$\circledast$	\circledast
$\curlyvee$	\curlyvee	$\curlywedge$	\curlywedge		

Tabuľka 3.18: $\mathcal{AMS}$ rôzne

$\hbar$	\hbar	$\complement$	\complement	$\Bbbk$	\Bbbk
$\hslash$	\hslash	$\blacksquare$	\blacksquare	$\circledS$	\circledS
$\vartriangle$	\vartriangle	$\blacktriangle$	\blacktriangle	$\square$	\square
$\triangledown$	\triangledown	$\blacktriangledown$	\blacktriangledown	$\Game$	\Game
$\lozenge$	\lozenge	$\blacklozenge$	\blacklozenge	$\bigstar$	\bigstar
$\angle$	\angle	$\measuredangle$	\measuredangle	$\diagdown$	\diagdown
$\diagup$	\diagup	$\sphericalangle$	\sphericalangle	$\Finv$	\Finv
$\nexists$	\nexists	$\backprime$	\backprime	$\mho$	\mho
$\eth$	\eth	$\varnothing$	\varnothing		

Tabuľka 3.19: Matematické abecedy

Príklad	Príkaz	Potrebný balík
ABCdef	\mathrm{ABCdef}	
ABCdef	\mathit{ABCdef}	
$\mathnormal{ABCdef}$	\mathnormal{ABCdef}	
$\mathcal{ABC}$	\mathcal{ABC}	
$\mathscr{ABC}$	\mathcal{ABC}	mathrsfs
$\mathcal{ABC}$	\mathcal{ABC}	eucal s: mathcal alebo
	\mathscr{ABC}	eucal s: mathscr
$\mathfrak{ABCdef}$	\mathfrak{ABCdef}	eufrak
$\mathbb{ABC}$	\mathbb{ABC}	amsfonts alebo amssymb

Kapitola 4

Špeciality

Pri spájaní veľkého dokumentu vám LaTeX pomôže riešiť niektoré špeciálne problémy, akými sú vytváranie registra, manažment bibliografických údajov a iné veci. Podstatne úplnejší popis špecialít a rozšírení prístupných v LaTeXu môžete nájsť v *LaTeX Manual* [1] a *The LaTeX Companion* [3].

4.1 Vkladanie EPS grafiky

Pomocou prostredí `figure` a `table` poskytuje LaTeX základné možnosti na prácu s plávajúcimi objektmi, napríklad s obrázkami alebo grafikou.

Pomocou LaTeXu alebo LaTeXových doplňujúcich balíkov je možné vytvárať aj obrázky. Nanešťastie väčšina užívateľov považuje tento spôsob za náročný na pochopenie. Z tohoto dôvodu sa ďalej nebudeme vytváraním obrázkov v LaTeXu v tomto manuále zaoberať.[1] Bližšie informácie o tvorbe obrázkov nájdete v *The LaTeX Companion* [3] a *LaTeX Manual* [1].

Oveľa jednoduchším spôsobom, ako dostať grafiku do dokumentu, je vytvoriť si daný obrázok špeciálnym grafickým programom[2] a potom vložiť hotový obrázok do dokumentu. LaTeX tu znovu ponúka širokú škálu balíkov, ktoré ponúkajú množstvo spôsobov, ako to urobiť. V tomto úvode sa budeme zaoberať len typom Encapsulated PostScript (EPS), nakoľko je pomerne jednoduchý a často používaný. Aby ste mohli používať obrázky vo formáte EPS, musíte mať k dispozícii PostScriptovú tlačiareň.[3]

Prospešnú sadu príkazov na vkladanie grafiky poskytuje balík graphicx, ktorého autorom je D. P. Carlisle. Je to časť celej rodiny balíkov nazývanej „grafický“ uzol.[4]

[1] V slovenskom preklade sa budeme, pozri oddiel 4.2 na strane 61. (pozn. prekl.).

[2] Ako napríklad XFig, CorelDraw!, Freehand, Gnuplot, ...

[3] Druhý spôsob, ako použiť PostScript, je pomocou programu GhostScript dostupného na adrese `CTAN:/tex-archive/support/ghostscript`.

[4] `CTAN:/tex-archive/macros/latex/packages/graphics`

Ak pracujete na systéme, ktorý má k dispozícii PostScriptovú tlačiareň a má nainštalovaný balík graphicx, tak môžete používať nasledujúci návod, kde sa krok za krokom dozviete, ako vložiť grafiku do dokumentu:

1. Uložte svoj obrázok v grafickom programe vo formáte EPS.[5]

2. Aktivujte balík graphicx v hlavičke zdrojového súboru príkazom

   ```
   \usepackage[ovládač]{graphicx}
   ```

   ovládač je názov vášho „dvi to postscript“ prekladača.[6] Tento názov je požadovaný balíkom, pretože neexistuje štandard na včlenenie grafiky do TEXu. Poznajúc *ovládač*, balík graphicx zvolí správnu metódu na vloženie informácie o obrázku do súboru `.dvi` tak, aby ju tlačiareň pochopila a mohla správne začleniť EPS súbor.

3. Na vloženie *súboru* do dokumentu použite príkaz

   ```
   \includegraphics[kľúč=hodnota, ...]{file}
   ```

   Za voliteľné parametre sa berie zoznam čiarkami oddelených *kľúčov* a ich hodnôt. *Kľúče* sa môžu použiť na zmenu šírky, výšky a otáčanie vkladanej grafiky. Tabuľka 4.1 uvádza zoznam najdôležitejších kľúčov.

Tabuľka 4.1: Názvy kľučov pre balík graphicx

`width`	zmení obrázok na danú šírku
`height`	zmení obrázok na danú výšku
`angle`	otočí obrázok v smere hodinových ručičiek
`scale`	zmení mierku obrázka

Nasledujúci príklad, dúfajme, urobí vo veciach jasno:

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=0.5\textwidth]{test.eps}
\end{center}
\end{figure}
```

[5]Ak váš softvér nedokáže exportovať do formátu EPS, môžete si skúsiť nainštalovať ovládač PostScriptovej tlačiarne (napríklad nejaký Apple LaserWriter) a potom tlačiť s týmto ovládačom do súboru. Pri troche šťastia bude tento súbor vo formáte EPS. Poznamenajme, že EPS obrázok nesmie obsahovať viac ako jednu stranu. Niektoré ovládače tlačiarní môžu byť explicitne nakonfigurované na vytváranie formátu EPS.

[6]Najrozšírenejší program sa volá `dvips`.

Príkaz vloží obrázok uložený v súbore `test.eps`. Obrázok je *najprv* otočený o 90 stupňov a *potom* rozšírený na konečnú šírku rovnajúcu sa polovici šírky štandardného odseku. Pomer výšky a šírky sa nezmenil, nakoľko sme nedefinovali žiadnu špeciálnu výšku. Parametre šírky a výšky môžu byť tiež zadané v absolútnych jednotkách. Pozrite sa na tabuľku 5.4 na strane 76. Ak sa o tejto téme chcete dozvedieť viac, určite si prečítajte [8] a [11].

4.2 Prostredie `picture`

Na tomto mieste doplníme stručný popis prostredia `picture`.[7] Podrobnejší popis nájdete v knižke [12]. LaTeXovské prostredie `picture` umožňuje priamo v LaTeXu vytvárať jednoduché obrázky, pozostávajúce z rovných čiar, šípiek, kružníc a oválov (respektíve polkružníc, štvrťkružníc). Okrem toho vyrába Bezierove krivky. Toto prostredie je vhodné aj na umiestňovanie popisov a to aj popisov ku grafike, vytváranej mimo TeXu, napríklad k bitmapovým obrázkom. Takýmto spôsobom sa dosiahne jednota textu a popisov obrázkov. Nasledujúci obrázok 4.1 ilustruje uvedené možnosti. Bol vytvorený programom TeXcad 3.2 (autori G. Horn a J. Winkelmann), ktorý je aj súčasťou distribúcie CSTeXu. Tento program je vhodný na prvé zoznámenie sa s prostredím `picture`, používateľ môže neskôr písať príkazy samostatne. Pri použití príkazov `\emline` a `\bezier` sú potrebné štýly `emlines2.sty` a `bezier.sty`. Navyše nie všetky ovládače zobrazia výsledok príkazu `\emline`.

Syntax prostredia je približne nasledujúca:

```
\begin{picture}(šírka,výška)(x_r,y_r)
\put(x,y){objekt}
\end{picture}
```

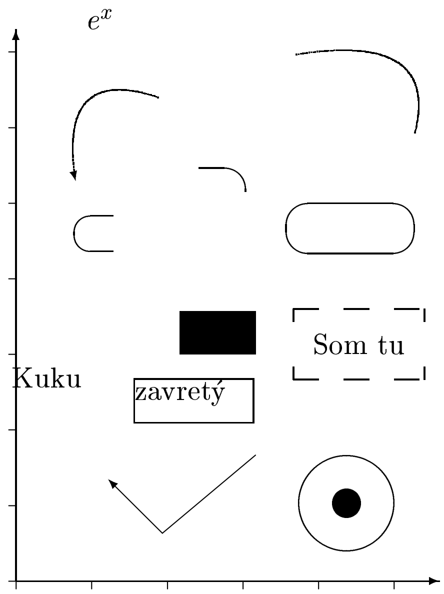
Parametre *šírka* a *výška* udávajú rozmery vytvoreného „boxu“. Zadanie $(0,0)$ spôsobí, že vytvorený objekt má nulový rozmer, teda TeX ostáva na mieste, kde bol pred zadaním prostredia `picture`. Nepovinné (x_r, y_r) sú súradnice „referenčného bodu“, teda bodu, v ktorom sa momentálne TeX nachádza, v súradnicovom systéme prostredia `picture`.

```
Uprostred slo\begin{picture}(0,0)(0,0)
\unitlength=1mm
\put(2,2.5)
    {\framebox(4,2.5)[tr]{$\bullet$}}
\end{picture}va môžem na chvíľu odísť.
```

Uprostred slova môžem na chvíľu odísť.

[7] Doplnené pri slovenskom preklade (pozn. prekl.).

```
\special{em:linewidth 0.4pt} \unitlength 1mm \linethickness{0.4pt}
\begin{picture}(56.00,73.00)
\put(0,0){\vector(1,0){56}} \put(0,0){\vector(0,1){73}} % osi x,y
\multiput(0,0)(10,0){6}{\emline{0}{0}{1}{0}{-1}{1}} % znacky osi x
\multiput(0,0)(0,10){8}{\emline{0}{0}{1}{-1}{0}{1}} % znacky osi y
\put(8.67,28.33){\makebox(0,0)[rt]{Kuku}}
\put(15.67,21.00){\framebox(15.67,5.67)[lt]{Zavretý}}
\put(36.67,26.67){\dashbox{3.33}(17.33,9.33)[cc]{Som tu}}
\put(21.67,30.00){\rule{10.00\unitlength}{5.67\unitlength}}
\emline{31.67}{16.67}{1}{19.33}{6.33}{2}
\put(12.33,13.33){\vector(-1,1){0.2}}
\emline{19.33}{6.33}{3}{12.33}{13.33}{4} %\end
\put(43.67,10.33){\circle{13.33}} \put(43.67,10.33){\circle*{4.00}}
\put(12.83,46.00){\oval(10.33,4.67)[l]}
\put(24.17,51.50){\oval(12.33,6.33)[rt]}
\put(44.17,46.67){\oval(17.00,6.67)[]}
\bezier{132}(52.67,59.33)(56.00,72.67)(37.00,69.67)
\bezier{116}(18.67,64.00)(5.67,68.33)(7.67,53.67)
\put(7.67,53.67){\vector(1,-4){0.2}} \put(9.33,73.00){$e^x$}
\end{picture}
```



Obr. 4.1: Príklad použitia prostredia `picture`

4.3 Zoznam použitej literatúry

Prostredím `thebibliography` môžete vytvárať zoznam použitej literatúry. Každý záznam sa začína príkazom

`\bibitem{`*značka*`}`

značka sa potom používa na citovanie knihy v dokumente.

`\cite{`*značka*`}`

Číslovanie záznamov je automatické. Maximálny počet záznamov určuje parameter za príkazom `\begin{thebibliography}`. V nižšie uvedenom príklade vytvárame zoznam literatúry, ktorý nebude mať viac ako 99 položiek.

```
Partl~\cite{pa}
navrhol, že \ldots

\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Vol.~9, No.~1 ('88)
\end{thebibliography}
```

Partl [1] navrhol, že ...

Literatúra

[1] H. Partl: *German TEX*, TUGboat Vol. 9, No. 1 ('88)

Pri väčších projektoch sa vám môže hodiť program BibTEX. Ten je súčasťou väčšiny TEXovských distribúcií. Umožní vám udržiavať bibliografickú databázu a potom z nej vyberať odkazy týkajúce sa prác, ktoré citujete vo vašom článku. Vizuálna prezentácia literatúry vytvorenej BibTEXom je založená na koncepcii štýlov, čo vám umožní vytvárať literatúru podľa požiadaviek širokej množiny rôznych zriadených vzorov.

4.4 Tvorba registrov

Veľmi prospešnou stránkou mnohých kníh je ich register. LATEXom a podporným programom `makeindex`[8] sa registre dajú vytvárať pomerne jednoducho. V tomto návode budú vysvetlené len základné príkazy na tvorbu registrov. Hlbšie znalosti získate z knižky *The LATEX Companion* [3].

Aby sa samotný register mohol vytvárať, v hlavičke musí byť natiahnutý balík makeidx:

```
\usepackage{makeidx}
```

a špeciálne príkazy na vytváranie registra musia byť aktivované umiestnením

[8] V systémoch, ktoré nepodporujú názvy dlhšie ako 8 znakov sa môže nazývať `makeidx`. Program `csindex`, vytvorený pánom Z. Wagnerom, je česká/slovenská implementácia programu `makeindex` (pozn. prekl.).

Tabuľka 4.2: Príklady syntaxe indexového kľúča

Príklad vstupu	**Výstup**	**Poznámka**
`\index{hello}`	hello, 1	základná položka
`\index{hello!Peter}`	Peter, 3	podpoložka položky ‚hello'
`\index{Sam@\textsl{Sam}}`	*Sam*, 2	formátovaný vstup
`\index{Lin@\textbf{Lin}}`	**Lin**, 7	podobne ako vyššie
`\index{Jenny|textbf}`	Jenny, **3**	formát čísla strany
`\index{Joe|textit}`	Joe, *5*	podobne ako vyššie

príkazu

```
\makeindex
```

do hlavičky zdrojového súboru.

Obsah registra sa určuje príkazmi

```
\index{kľúč}
```

kde *kľúč* je záznam v registri. Príkazy na tvorbu registra umiestňujete na miesto v texte, na ktoré chcete, aby index odkazoval. Tabuľka 4.2 popisuje syntax parametra *kľúč* s niekoľkými príkladmi.

Keď LaTeX prekladá zdrojový súbor, každý príkaz `\index` zapisuje príslušný indexový záznam spolu s číslom strany do špeciáleho súbora. Tento súbor má rovnaký názov ako LaTeXový zdrojový dokument, avšak líši sa koncovkou (`.idx`). Tento `.idx` súbor potom môže byť spracovaný programom `makeindex`:

```
makeindex menosúboru
```

Tento vytvorí usporiadaný register s rovnakým názvom, avšak v tomto prípade s príponou `.ind`. V prípade, ak bude LaTeX tento zdrojový súbor znovu prekladať, bude tento zoradený register umiestnený v dokumente na mieste, kde sa nachádza príkaz

```
\printindex
```

Balík showidx, ktorý je distribuovaný spolu s LaTeX 2_ε, vytlačí všetky indexované údaje na ľavom okraji textu. Toto je celkom prospešné pri kontrolnom čítaní dokumentu a overovaní registra.

4.5 Exotické hlavičky

Balík fancyhdr (od spojenia Fancy Headers)[9] napísaný Pietom van Oostrum poskytuje niekoľko jednoduchých príkazov, ktoré vám umožnia prispôsobiť si riadky v hlavičke alebo v päte vášho dokumentu. Ak sa pozriete na vrch strany, ktorú práve čítate, môžete vidieť možné použitie tohto balíka.

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% týmto sa uistíme, že hlavičky pre chapter a section
% budú písané malými písmenami.
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{}  % ruší aktuálne nastavenia pre hlavičku a pätu
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % vytvára medzeru pre čiaru
\fancypagestyle{plain}{%
   \fancyhead{} % zbaví sa hlavičky na stránkach plain
   \renewcommand{\headrulewidth}{0pt} % aj čiary
}
```

Obr. 4.2: Príklad nastavenia fancyhdr

Skryté a neočakávané ťažkosti sa môžu vyskytnúť, keď sa snažíme do hlavičky alebo päty dostať niečo ako priebežné oddiely a kapitoly. LaTeX to vykoná dvojstupňovým prístupom. V definícii hlavičky a päty použijete príkazy `\rightmark` a `\leftmark` na reprezentáciu názvu aktuálnej kapitoly, respektíve oddielu. Hodnoty týchto dvoch príkazov budú prepísané pri každom vykonaní príkazu `\chapter` alebo `\section`.

Pre konečnú pružnosť príkaz `\chapter` s kamarátmi nepredefinujú samotné `\rightmark` a `\leftmark`. Volajú ešte ďalšie príkazy `\chaptermark`, `\sectionmark` alebo `\subsectionmark`, ktoré sú zodpovedné za predefinovanie obsahu `\rightmark` a `\leftmark`.

Takže ak chcete zmeniť vzhľad názvu kapitoly v hlavičkovom riadku, musíte jednoducho „obnoviť“ príkaz `\chaptermark`.

Obrázok 4.2 ukazuje možné nastavenia pre balík fancyhdr, ktoré vytvoria hlavičky vyzerajúce približne tak ako hlavičky v tejto brožúrke. V každom

[9] Dostupný na `CTAN:/tex-archive/macros/latex/contrib/supported/fancyhdr`.

prípade vám odporúčam stiahnuť si dokumentáciu k balíku na adrese uvedenej v poznámke.

4.6 Balík verbatim

V našej knižke ste sa už skôr zoznámili s *prostredím* `verbatim`. V tomto oddiele sa naučíte niečo o *balíku* verbatim. Balík verbatim je v princípe reimplementácia prostredia `verbatim`, ktorá obchádza niektoré ohraničenia pôvodného prostredia `verbatim`. Toto samo osebe nie je nič mimoriadne, ale s touto implementáciou balíka verbatim boli pridané aj nové funkcie, čo bolo dôvodom na to, že tu tento balík spomínam. Balík verbatim sprístupňuje príkaz

`\verbatiminput{`*menosúboru*`}`

ktorý vám umožní začleniť surový ASCII text do vášho dokumentu ako keby bol vo vnútri prostredia `verbatim`.

Keďže balík verbatim je časťou uzla ‚tools‘, mali by ste ho nájsť nainštalovaný vo väčšine systémov. Ak sa o tomto balíku chcete dozvedieť viac, určite si prečítajte [9].

4.7 Ochrana krehkých príkazov

Text zadaný ako argument príkazov typu `\caption` alebo `\section` sa môže ukazovať v dokumente viackrát (napríklad v obsahu rovnako ako v samotnom texte). Niektoré príkazy skrachujú, ak sú použité ako argument príkazov typu `\section`. Tieto príkazy sa nazývajú krehké príkazy. Krehký príkaz je napríklad `\footnote` alebo `\phantom`. Na to, aby tieto príkazy fungovali, potrebujú ochranu (a my všetci nie?). Môžete ich ochrániť použitím príkazu `\protect` umiestneným pred nimi.

`\protect` sa vzťahuje len na príkaz nasledujúci bezprostredne za ním, nevzťahuje sa dokonca ani na jeho argumenty. Nadbytočný príkaz `\protect` vo väčšine prípadov neuškodí.

```
\section{Som ohľaduplný
   \protect\footnote{a chránim svoje poznámky pod čiarou}}
```

Kapitola 5

Prispôsobovanie LATEXu

Vzhľad dokumentov vytvorených na základe príkazov, ktoré ste sa doteraz naučili bude prijateľný pre široký kruh užívateľov. Hoci nemusia vyzerať fantasticky, budú vyhovovať ustanoveným pravidlám dobrej typografie, čo ich urobí ľahko čitateľnými a na pohľad príjemnými.

Napriek tomu sú situácie, keď LATEX neposkytuje príkazy alebo prostredia odpovedajúce vašim potrebám, prípadne výstup získaný na základe existujúcich príkazov neuspokojuje vaše požiadavky.

V tejto kapitole sa pokúsim trochu naznačiť ako naučiť LATEX nové triky a ako ho donútiť vytvoriť výstup, ktorý vyzerá ináč, ako to poskytuje štandard.

5.1 Nové príkazy, prostredia a balíky

Mohli ste si všimnúť, že všetky príkazy, ktoré som uviedol v tejto knihe, sú sádzané v rámčekoch a že sú vytlačené v registri na konci knižky. Namiesto priameho použitia príkazov LATEXu potrebných na dosiahnutie týchto úloh som vytvoril balík, v ktorom som za týmto účelom definoval nové príkazy a prostredia. Teraz jednoducho napíšem:

```
\begin{lscommand}
\ci{dum}
\end{lscommand}
```

`\dum`

V tomto príklade používam oboje: nové prostredie s názvom `command`, ktoré je zodpovedné za nakreslenie rámčeka okolo príkazu aj nový príkaz `\ci`, ktorý vytlačí názov príkazu a zároveň vytvorí odpovedajúcu položku registra. Môžete sa o tom presvedčiť, ak sa pozriete do registra na konci knihy a nájdete položku príkazu `\dum`, ukazujúcu na každú stranu, kde som sa o príkaze `\dum` zmienil.

Ak by som sa hocikedy rozhodol, že už viac nemám rád príkazy sádzané v rámčekoch, môžem jednoducho zmeniť definíciu prostredia `command` a vytvoriť jeho nový vzhľad. Je to oveľa ľahšie ako prechádzať celým dokumentom a hľadať všetky miesta, kde som použil niektoré základné príkazy LaTeXu na vykreslenie rámčeku okolo nejakého slova.

5.1.1 Nové príkazy

Na pridanie vašich nových príkazov použite príkaz

`\newcommand{`*meno*`}[`*počet*`]{`*definícia*`}`

Príkaz potrebuje v podstate dva argumenty. *meno* – názov príkazu, ktorý chcete vytvoriť a *definíciu* príkazu. Argument *počet* v hranatých zátvorkách je nepovinný a určuje počet parametrov, ktoré používa nový príkaz (maximálne 9). Ak chýba, štandardná hodnota je 0, t. j. nie je dovolený žiaden argument.

Nasledujúce dva príklady by vám mali pomôcť pochopiť hlavnú myšlienku. Prvý príklad definuje príkaz nazývaný `\npsu` (toto je skratka pre „Nie príliš stručný úvod do systému LaTeX 2ε“). Takýto príkaz môže byť veľmi praktický, keď musíte písať názov tejto knihy znovu a znovu.

```
\newcommand{\npsu}
    {Nie príliš stručný úvod
        do systému \LaTeXe}
% v dokumente:
Toto je \uv{\npsu} \ldots{} \uv{\npsu}
```

Toto je „Nie príliš stručný úvod do systému LaTeX 2ε“ … „Nie príliš stručný úvod do systému LaTeX 2ε“

Nasledujúci príklad ukazuje, ako sa definuje nový príkaz s parametrom. Značka `#1` bude nahradená parametrom, ktorý zadáte. Ak chcete použiť viac ako jeden parameter, použijete `#2` a tak ďalej.

```
\newcommand{\txsit}[1]
    {Toto je \emph{#1} stručný
      úvod do systému \LaTeXe}
% v dokumente:
\begin{itemize}
\item \txsit{nie príliš}
\item \txsit{veľmi}
\end{itemize}
```

- Toto je *nie príliš* stručný úvod do systému LaTeX 2ε
- Toto je *veľmi* stručný úvod do systému LaTeX 2ε

LaTeX vám nedovolí vytvoriť nový príkaz, ktorý predefinuje už existujúci. Ale v prípade, ak chcete urobiť práve toto, máte k dispozícii zvláštny príkaz `\renewcommand`. Používa rovnakú syntax ako príkaz `\newcommand`.

Za určitých podmienok môžete tiež chcieť použiť príkaz `\providecommand`. Funguje podobne ako `\newcommand`, ale ak príkaz s týmto názvom je už definovaný, LaTeX ho bude potichu ignorovať.

Pár slov by bolo dobré povedať o medzerách nasledujúcich za príkazmi LaTeXu. Pre podrobnejšiu informáciu pozrite stranu 6.

5.1.2 Nové prostredia

Podobný príkaz ako `\newcommand` existuje aj na vytváranie vašich vlastných prostredí. Počas písania tohoto úvodu do LaTeXu som si vytvoril špeciálne prostredia pre štruktúry, ktoré som opakovane používal: „príklady“, „úryvky zo zápisov“ a „rámčeky definícií príkazov“. Príkaz `\newenvironment` má nasledujúcu štruktúru:

`\newenvironment{`*meno*`}[`*počet*`]{`*pred*`}{`*potom*`}`

Ako pri príkaze `\newcommand`, aj príkaz `\newenvironment` môžete použiť s parametrom alebo bez neho. Materiál bližšie určený v parametri *pred* je spracovaný predtým ako text v danom prostredí. Materiál bližšie určený v parametri *potom* je spracovaný až po vykonaní príkazu `\end{`*meno*`}`.

Nižšie uvedený príkaz ukazuje používanie príkazu `\newenvironment`.

```
\newenvironment{kral}
 {\rule{1ex}{1ex}%
      \hspace{\stretch{1}}}
 {\hspace{\stretch{1}}%
      \rule{1ex}{1ex}}

% použite toto v časti textu
\begin{kral}
Moje skromné prípady \ldots
\end{kral}
```

■ Moje skromné prípady … ■

Argument *počet* sa používa rovnakým spôsobom ako to bolo v prípade príkazu `\newcommand`. LaTeX spoľahlivo zabezpečuje, aby ste nenadefinovali prostredie, ktoré už existuje. Ak budete chcieť niekedy zmeniť existujúci príkaz, môžete použiť príkaz `\renewenvironment`. Používa sa rovnakým spôsobom ako príkaz `\newenvironment`.

Príkazy použité v tomto príklade budú vysvetlené neskôr. Na príkaz `\rule` sa pozrite na stranu 81, príkaz `\stretch` nájdete na strane 75. Podrobnejšia informácia o `\hspace` sa dá nájsť na strane 75.

5.1.3 Váš vlastný balík

Ak definujete veľké množstvo prostredí a príkazov, bude preambula vášho dokumentu príliš dlhá. V tomto prípade je dobrým nápadom vytvorenie LaTeXovského balíka obsahujúceho všetky definície vašich príkazov a prostredí. Na sprístupnenie balíka vo vašom dokumente môžete použiť príkaz `\usepackage`.

```
% Demo Package by Tobias Oetiker
\ProvidesPackage{demopack}
\newcommand{\npsu}{Nie príliš stručný úvod do systému \LaTeXe}
\newcommand{\txsit}[1]{Toto je \emph{#1} stručný úvod
                            do systému \LaTeXe}
\newenvironment{king}{\rule{1ex}{1ex}%
     hspace{\stretch{1}}}{\hspace{\stretch{1}}\rule{1ex}{1ex}}
```

Obr. 5.1: Príklad balíka

Napísanie balíka spočíva v podstate v skopírovaní obsahu vašej preambuly do samostatného súboru s názvom s koncovkou `.sty`. Existuje jeden zvláštny príkaz,

`\ProvidesPackage{`*názov balíka*`}`

používaný na úplnom začiatku súboru vášho balíka. `\ProvidesPackage` oznámi LaTeXu názov balíka a umožní mu vydať výrazné chybové hlásenie, ak sa pokúsite balík inicializovať druhý raz. Obrázok 5.1 ukazuje malý príklad balíka obsahujúceho príkazy definované vo vyššie uvedených príkladoch.

5.2 Fonty a veľkosti

5.2.1 Príkazy na zmenu písma

LaTeX volí vhodný font a jeho veľkosť na základe logickej štruktúry dokumentu (oddiely, poznámky pod čiarou, ...). V niektorých prípadoch môže niekto chcieť zmeniť fonty a ich veľkosti ručne. Toto sa dá urobiť pomocou príkazov, ktoré sú uvedené v tabuľkách 5.1 a 5.2. Skutočná veľkosť každého fontu je výsledok dizajnu a závisí od triedy dokumentu a jej volieb. Tabuľka 5.2 ukazuje absolútne bodové veľkosti pre tie príkazy, ktoré sú súčasťou štandardných tried dokumentu.

Tabuľka 5.1: Fonty

`\textrm{...}`	roman (antikva)	`\texttt{...}`	`typewriter`
`\textsf{...}`	sans serif (bezpätkové)		
`\textbf{...}`	**bold face** (tučné)	`\textmd{...}`	medium
`\textit{...}`	*italic* (kurzíva)	`\textup{...}`	upright
`\textsl{...}`	*slanted* (naklonené)	`\textsc{...}`	SMALL CAPS
`\emph{...}`	*emphasised*	`\textnormal{...}`	bežné

```
{\small The small and
\textbf{bold} Romans ruled}
{\Large all of great big
\textit{Italy}.}

{\small Malí,
\textbf{tuční} Rimania vládli}
{\Large celej veľkej
\textit{Itálii}.}%
\footnote{roman\,--\,antikva,
italic\,--\,kurzíva, bold\,--\,tučný
--- v~angličtine je to pekná
typografická slovná hračka (pozn. prekl.)}
```

The small and **bold** Romans ruled all of great big *Italy.*

Malí, **tuční** Rimania vládli celej veľkej *Itálii.*[a]

[a]roman – antikva, italic – kurzíva, bold – tučný — v angličtine je to pekná typografická slovná hračka (pozn. prekl.)

Jedna dôležitá vlastnosť systému LaTeX 2ε je, že vlastnosti fontov sú nezávislé. To znamená, že môžete zmeniť počiatočnú veľkosť alebo dokonca typ fontu a pritom zachovať skôr zadaný príkaz na tučné alebo naklonené písmo. [1]

V *matematickom režime* môžete používať na dočasné opustenie *matematického režimu* a napísanie normálneho textu príkazy na zmenu aktuálneho typu písma. Ak chcete v matematickom režime použiť nejaký iný typ písma, existuje ďalšia sada príkazov. Tieto fonty sú podrobne uvedené v tabuľke 5.3.

V súvislosti s príkazmi na veľkosť a tvar písma majú významné postavenie zložené zátvorky. Používajú sa na vytváranie *skupín.* Skupiny obmedzujú pôsobenie väčšiny LaTeXových príkazov.

```
Má rád {\LARGE veľké a
{\small malé} písmená}.
```

Má rád veľké a malé písmená.

[1]Aj keď je toto zrejmé pre niekoho, kto sa učí LaTeX 2ε, nie je to také zrejmé pre niekoho, kto používal LaTeX 2.09.

Tabuľka 5.2: Veľkosti fontov

`\tiny`	drobný font	`\Large`	väčší font
`\scriptsize`	veľmi malý font	`\LARGE`	veľmi veľký font
`\footnotesize`	dosť malý font		
`\small`	malý font	`\huge`	obrovský font
`\normalsize`	normálny font		
`\large`	veľký font	`\Huge`	najväčší font

Tabuľka 5.3: Matematické fonty

Príkaz	*Príklad*	*Výstup*
`\mathcal{...}`	`$\mathcal{B}=c$`	$\mathcal{B}=c$
`\mathrm{...}`	`$\mathrm{K}_2$`	K_2
`\mathbf{...}`	`$\sum x=\mathbf{v}$`	$\sum x=\mathbf{v}$
`\mathsf{...}`	`$\mathsf{G\times R}$`	$\mathsf{G\times R}$
`\mathtt{...}`	`$\mathtt{L}(b,c)$`	$\mathtt{L}(b,c)$
`\mathnormal{...}`	`$\mathnormal{R_{19}}\ne R_{19}$`	$\mathnormal{R_{19}}\ne R_{19}$
`\mathit{...}`	`$\mathit{ffi}\neq ffi$`	$\mathit{ffi}\neq ffi$

Príkaz na zmenu veľkosti fontov môže zmeniť aj riadkovanie, avšak iba ak odstavec končí za účinku príkazu na veľkosť fontu. Zatváracia zložená zátvorka `}` preto nesmie prísť príliš skoro. Všimnite si polohu príkazu `\par` v nasledujúcich dvoch príkladoch.

```
{\Large Nečítajte to! To nie
je pravda. Môžete mi veriť!\par}
```

Nečítajte to! To nie je pravda. Môžete mi veriť!

```
{\Large Toto nie je pravda,
ale nezabúdajte: som klamár.}\par
```

Toto nie je pravda, ale nezabú-
dajte: som klamár.

Ak chcete uplatniť príkaz na zmenu veľkosti pre celý odstavec textu alebo dokonca na viac textu, môže byť užitočné použiť na zmenu veľkosti syntax typu prostredí.

```
\begin{Large}
To nie je pravda. Ale potom
znova, čo sú dnešné dni\ldots
\end{Large}
```

To nie je pravda. Ale potom znova, čo sú dnešné dni…

Toto vás ochráni pred počítaním množstva zložených zátvoriek.

5.2.2 Pozor, nebezpečenstvo!

Ako bolo poznamenané na začiatku tejto kapitoly, je nebezpečné preplniť váš dokument explicitnými príkazmi, pretože tieto potláčajú základnú myšlienku LaTeXu, ktorou je oddelenie logického a vizuálneho značkovania vášho dokumentu. To znamená, že ak chcete použiť rovnaký príkaz na zmenu písma na viacerých miestach pre to, aby ste zvýraznili zvláštny druh informácií, mali by ste použiť príkaz `\newcommand` na definovanie „príkazu logického vyčlenenia“, ktorý zmení font.

```
\newcommand{\oops}[1]{\textbf{#1}}
\oops{Nevstupujte} do tejto
miestnosti, je obsadená\oops{strojom}
neznámeho pôvodu a účelu.
```

Nevstupujte do tejto miestnosti, je obsadená**strojom** neznámeho pôvodu a účelu.

Tento prístup má tú výhodu, že vám neskôr umožní rozhodnúť sa, či nechcete použiť nejakú inú vizuálnu reprezentáciu nebezpečenstva, ako je použitie `\textbf` bez toho, aby ste sa brodili svojím dokumentom, vyhľadávali všetky výskyty príkazu `\textbf` a potom pre každý z nich určovali, či bol použitý na vyznačenie nebezpečenstva alebo na nejaký iný účel.

5.2.3 Rada

Na záver tejto cesty do krajiny fontov a ich veľkostí vám ponúkam malú radu:

Pamätajte! *Čím* VIAC fontov použijete vo **vašom** dokumente, tým *krajším* a ČITATEĽNEJŠÍM *sa* stane.[2]

[2]Pozor! Toto je trochu satira. Dúfam, že ste si to všimli!

5.3 Medzery

5.3.1 Riadkovanie

Ak chcete väčšie medzery medzi riadkami svojho dokumentu, môžete to urobiť pomocou príkazu

`\linespread{`*faktor*`}`

umiestneného v hlavičke zdrojového súboru. Príkaz `\linespread{1.3}` zabezpečí riadkovanie $1\frac{1}{2}$, pomocou príkazu `\linespread{1.6}` dosiahneme riadkovanie 2. Obvykle sa medzery medzi riadkami nezväčšujú, preto je základný zväčšovací koeficient rovný 1.

5.3.2 Formátovanie odsekov

V LaTeXu existujú dva príkazy ovplyvňujúce úpravu odsekov. Pomocou príkazov

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

umiestnených v hlavičke[3] zdrojového súboru môžete meniť nastavenie odsekov. Tieto dva príkazy zväčšujú miesto medzi odsekmi, pričom rušia odsadzovanie prvých riadkov odsekov. V kontinentálnej Európe sa odseky často oddeľujú voľným miestom a neodsadujú sa ich prvé riadky. Ale pozor, toto má vplyv aj na obsah. Aj riadky obsahu budú oddeľované väčším priestorom. Aby ste sa tomu vyhli, môžete umiestniť tieto dva príkazy na nejaké miesto za príkazom `\tableofcontents` alebo sa môžete rozhodnúť, že ich nepoužijete vôbec, pretože ste zistili, že väčšina profesionálnych kníh používa odsadzovanie odsekov bez medzery medzi nimi.

Ak chcete odsadiť odsek, ktorý nie je odsadený, môžete použiť na začiatku odseku príkaz[4]

`\indent`

Samozrejme, že to môže fungovať len v prípade, ak nie je `\parindent` nastavený na nulu.

[3] Medzi príkazy `\documentclass` a `\begin{document}`.

[4] Ak chcete odsadiť prvý odsek v každej kapitole, použite balík indentfirst z uzla ‚tools'.

Ak chcete, aby daný odsek nebol odsadený, môžete na začiatku odseku použiť príkaz

```
\noindent
```

Môže sa vám to zísť, ak začínate dokument samotným textom a nie príkazom pre kapitolu, respektíve oddiel.

5.3.3 Vodorovné medzery

LaTeX určuje medzery medzi slovami a vetami automaticky. Ak chcete zväčšiť vodorovnú (horizontálnu) medzeru, musíte použiť príkaz:

\hspace{*dĺžka*}

Ak sa má takáto medzera zachovať aj v prípade, že pripadne na koniec alebo začiatok riadku, použite príkaz `\hspace*` namiesto `\hspace`. V najjednoduchších prípadoch je *dĺžka* len číslo a jednotka. Najdôležitejšie jednotky sú uvedené v tabuľke 5.4.

```
Táto\hspace{1.5cm}medzera
má 1.5 cm.
```

Táto   medzera má 1.5 cm.

Príkaz

\stretch{*n*}

vytvára špeciálnu mäkkú (gumenú) medzeru. Rozpína sa, až kým nie je vyplnený celý zvyšný priestor riadku. Ak sa v jednom riadku nachádzajú dva príkazy `\hspace{\stretch{`*n*`}}`, narastajú primerane svojmu rozširovaciemu koeficientu.

```
x\hspace{\stretch{1}}
x\hspace{\stretch{3}}x
```

x   x   x

5.3.4 Zvislé medzery

Medzera medzi odsekmi, oddielmi, pooddielmi, ... je určovaná automaticky LaTeXom. V prípade, že je to potrebné, môžete zvislú (vertikálnu) medzeru zväčšiť pomocou príkazu:

\vspace{*dĺžka*}

Tento príkaz sa normálne oddeľuje voľným riadkom pred a za príkazom.

Tabuľka 5.4: Jednotky TeXu

`mm`	milimeter $\approx 1/25$ inch
`cm`	centimeter $= 10$ mm
`in`	inch $= 25.4$ mm
`pt`	point $\approx 1/72$ inch $\approx \frac{1}{3}$ mm
`em`	približne výška daného fontu
`ex`	približne výška `x` v danom fonte

V prípade, že sa má medzera udržovať aj na začiatku (konci) strany a medzi dvoma stránkami, použite namiesto `\vspace` hviezdičkovú verziu príkazu `\vspace*`.

Príkaz `\stretch` spolu s príkazom `\pagebreak` sa môže použiť na napísanie textu na posledný riadok strany, prípadne na zvislé vycentrovanie strany.

```
Nejaký text \ldots

\vspace{\stretch{1}}
Toto bude napísané na poslednom riadku strany.\pagebreak
```

V prípade, že chceme zväčšiť medzery medzi riadkami *toho istého* odseku alebo vo vnútri tabuľky, môžeme namiesto príkazu `\\` použiť príkaz:

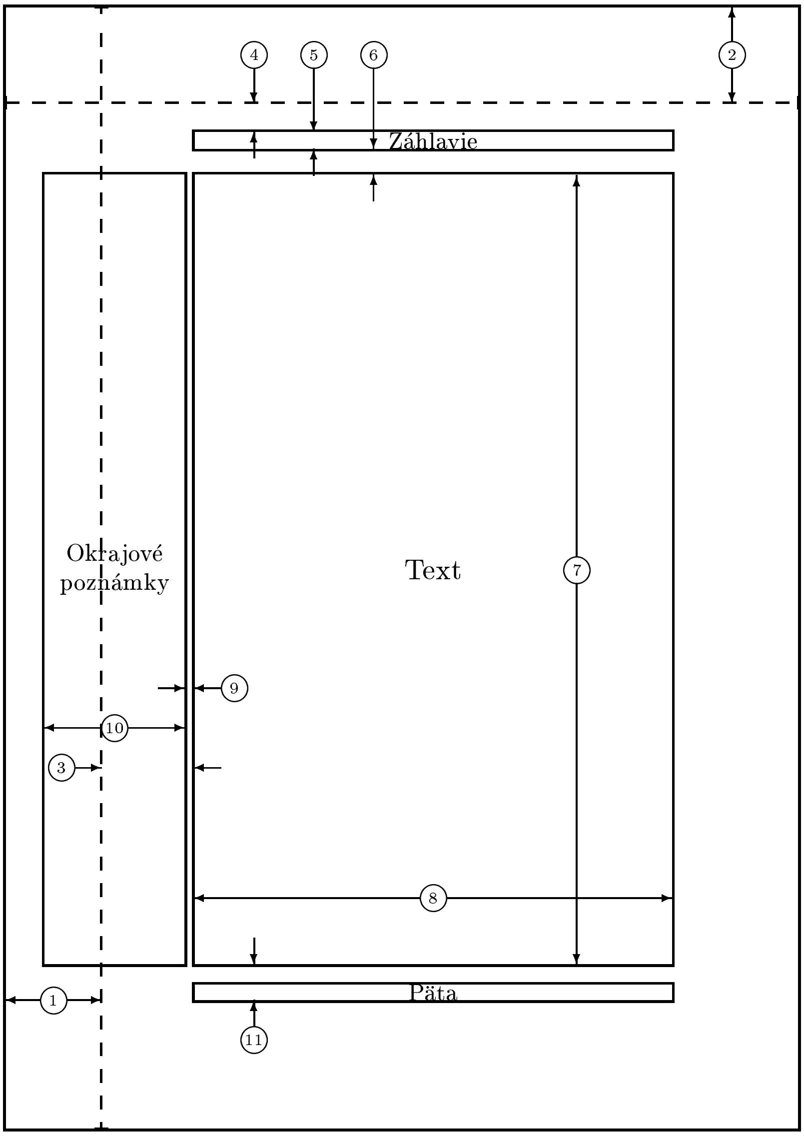
`\\[`*dĺžka*`]`

5.4 Rozvrhnutie stránky

LaTeX 2_ε vám umožňuje zadať veľkosť papiera v príkaze `\documentclass`. LaTeX potom automaticky zvolí správne rozmery textu. V niektorých prípadoch vám však preddefinované hodnoty nemusia vyhovovať. Prirodzene, môžete ich zmeniť. Obrázok 5.2 ukazuje všetky parametre, ktoré sa dajú meniť. Obrázok bol vytvorený pomocou balíka layout z uzla tools.[5]

POČKAJTE! …skôr než sa pustíte do šialených aktivít typu: „Rozšírme trošku túto úzku stranu!“, porozmýšľajte pár sekúnd. Rovnako ako v prípade väčšiny vecí v LaTeXu, nájdu sa dobré dôvody aj nato, aby bol vzhľad stránky taký, aký je.

[5] `CTAN:/tex-archive/macros/latex/packages/tools`



1	`jeden palec + \hoffset`	2	`jeden palec + \voffset`
3	`\evensidemargin = 70pt`	4	`\topmargin = 22pt`
5	`\headheight = 13pt`	6	`\headsep = 19pt`
7	`\textheight = 595pt`	8	`\textwidth = 360pt`
9	`\marginparsep = 7pt`	10	`\marginparwidth = 106pt`
11	`\footskip = 27pt`		`\marginparpush = 5pt (nezobrazené)`
	`\hoffset = 0pt`		`\voffset = 0pt`
	`\paperwidth = 597pt`		`\paperheight = 845pt`

Obr. 5.2: Parametre ovplyvňujúce rozvrhnutie stránky

Samozrejme, v porovnaní so stránkou MS Wordu vyzerá hrozne úzka. Ale pozrite sa na vašu obľúbenú knihu[6] a spočítajte počet písmen na štandardnom textovom riadku. Zistíte, že tam nie je viac písmen ako okolo 66. Teraz urobte to isté so svojou LaTeXovskou stranou. Zistíte, že aj tam sa nachádza približne 66 znakov na riadok. Skúsenosti ukazujú, že ak sa na riadku vyskytuje viac písmen, čítanie sa stáva obtiažnejším. Je to preto lebo očiam je náročné presúvať sa z konca riadku na začiatok ďalšieho. Toto je tiež dôvod nato, aby sa noviny tlačili do viacerých stĺpcov.

Teda ak zväčšíte šírku vášho textu, uvedomte si, že komplikujete život čitateľom vašej práce. Ale dosť bolo výstrah, sľúbil som, že vám poviem, ako sa to robí...

LaTeX poskytuje dva príkazy na zmenu týchto parametrov. Väčšinou sa používajú v hlavičke dokumentu.

Prvým príkazom sa nastavuje určitá hodnota ktoréhokoľvek z daných parametrov:

`\setlength{`*parameter*`}{`*dĺžka*`}`

Druhý príkaz pridáva určitú hodnotu ku ktorémukoľvek z daných parametrov:

`\addtolength{`*parameter*`}{`*dĺžka*`}`

Tento druhý príkaz je v skutočnosti prospešnejší než príkaz `\sethlength`, pretože môžete pracovať podobne, ako pri existujúcich nastaveniach. Aby ste k šírke stránky pridali jeden centimeter, do hlavičky treba pridať nasledujúce príkazy:

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

V tejto súvislosti vás možno zaujme balík calc umožňujúci použitie aritmetických operácií v argumente príkazu `\setlength` a na iných miestach, kde môžete zadávať numerické hodnoty ako argumenty funkcií.

5.5 Viac zábavy s dĺžkami

Vždy, keď je to možné, vyhýbam sa používaniu absolútnych dĺžok v LaTeXovských dokumentoch. Radšej sa snažím založiť veci na dĺžke, výške alebo inom prvku strany. Pre šírku obrázku to môže byť `\textwidth` – takto zaplní stranu.

[6] Mám na mysli skutočnú tlačenú knihu vyrobenú vydavateľom s dobrou povesťou.

Nasledujúce 3 príkazy umožňujú určiť šírku, výšku a hĺbku textového reťazca.

```
\settoheight{príkaz}{text}
\settodepth{príkaz}{text}
\settowidth{príkaz}{text}
```

Nasledujúci príklad ilustruje možné použitie týchto príkazov.

```
\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1:\ }
  \makebox[0pt][r]{#1:\ }}{}

\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}

\begin{vardesc}{Kde}$a$,
$b$ -- sú odvesny pravouhlého
       trojuholníka.

$c$ -- je prepona trojuholníka
       a cíti sa osamelo.

$d$ -- sa tu nakoniec vôbec nevyskytuje. Nie je to popletené?
\end{vardesc}
```

$$a^2+b^2=c^2$$

Kde: a, b – sú odvesny pravouhlého trojuholníka.

c – je prepona trojuholníka a cíti sa osamelo.

d – sa tu nakoniec vôbec nevyskytuje. Nie je to popletené?

5.6 Škatuľky, krabičky, schránky, skrinky, boxy

LaTeX vytvára svoje strany rozmiestňovaním boxov (škatuliek, krabičiek, schránok, skriniek). Po prvé, každé písmeno je malá krabička, ktorá sa ďalej nalepuje na iné písmená, vytvárajúc slová. Tieto sa zasa prilepujú k ďalším slovám, ale špeciálnym glejom[7], ktorý je pružný. Preto môže byť rad slov stlačený alebo roztiahnutý tak, aby mohol presne vyplniť riadok na strane.

Pripúšťam, že toto je veľmi zjednodušená verzia toho, čo sa skutočne stane, ale myšlienka je taká, že TeX narába s glejom (lepidlom) a so škatuľkami (boxmi). Ale nielen písmená môžu tvoriť box. Do krabičky môžete položiť skutočne všetko, vrátane iných boxov. LaTeX zaobchádza s každým boxom, akoby to bolo jedno písmeno.

V predchádzajúcich kapitolách ste sa už niektorými boxmi stretli, hoci som vám to nepovedal. Prostredie `tabular` a príkaz `\includegraphics`, napríklad, oboje vytvárajú box. To znamená, že ľahko môžete umiestniť dve

[7] Anglicky glue (pozn. prekl.)

tabuľky alebo dva obrázky vedľa seba. Len si musíte byť istí, že ich spoločná šírka nie je väčšia ako nastavená šírka strany.

Rovnako môžete do boxu zabaliť vami zvolený odsek a to buď príkazom

`\parbox[`*pozícia*`]{`*šírka*`}{`*text*`}`

alebo pomocou prostredia

`\begin{minipage}[`*pozícia*`]{`*šírka*`}` text `\end{minipage}`

Parameter *pozícia* môže byť jedno z písmen `c`, `t` alebo `b`. Týmto sa riadi vertikálne uloženie boxu vzhľadom na základnú čiaru okolitého textu. Parameter *šírka* je dĺžkový argument určujúci šírku boxu. Základný rozdiel medzi príkazmi `minipage` a `parbox` je, že vo vnútri `parbox`u nemôžete používať všetky príkazy a prostredia, zatiaľ čo vo vnútri `minipage` je možné takmer všetko.

Zatiaľ čo `\parbox` zabalí celý odsek so zalomením riadkov, existuje aj trieda príkazov vytvárajúcich boxy, ktoré pracujú len s horizontálne uloženým materiálom. Už poznáme jeden z nich. Nazýva sa `\mbox` a jednoducho spája do jedného boxu rad boxov. Môže byť použitý na zabránenie LaTeXu rozdeliť dve slová. Keďže dovnútra boxov môžete uložiť ďalšie boxy, tieto príkazy na horizontálne balenie boxov vám poskytujú obrovskú flexibilitu.

`\makebox[`*šírka*`][`*pozícia*`]{`*text*`}`

šírka určuje šírku výsledného boxu videného zvonku.[8] Oddelene od dĺžkových výrazov môžete použiť aj `\width`, `\height`, `\depth` a `\totalheight` v parametri šírky. Tieto sa nastavia podľa hodnôt získaných meraním vytlačeného *text*u. Parameter *pozícia* nadobúda jednu z hodnôt: **c**entrovanie, **l**eft flush (zarovnanie naľavo), **r**ight flush (zarovnanie vpravo) alebo **s**, ktorý roztiahne text, obsiahnutý v boxe, na celú šírku boxu.

Príkaz `\framebox` funguje presne tak ako `\makebox`, ale navyše nakreslí okolo textu rámček.

Nasledujúci príklad vám ukáže niektoré veci, ktoré môžete robiť príkazmi `\makebox` a `\framebox`.

[8]To znamená, že môže byť menšia ako materiál obsiahnutý vo vnútri boxu. Dokonca môžete nastaviť šírku na 0pt, takže text vo vnútri boxu sa vytlačí bez vplyvu na okolité boxy.

```
\makebox[\textwidth]{%
    c e n t r o v a n i e}\par
\makebox[\textwidth][s]{%
    r o z t i a h n u t i e}\par
\framebox[1.1\width]{Tuším
            som zarámovaný!} \par
\framebox[0.8\width][r]{Ty nešika,
   som príliš široký} \par
\framebox[1cm][l]{nevadí,
     ja som tiež}
Dokážeš to prečítať?
```

c e n t r o v a n i e

r o z t i a h n u t i e

Tuším som zarámovaný!

Ty nešika, som príliš široký

nevadí, ja som tiež Dokážeš to prečítať?

Teraz, keď sme zvládli vodorovné riadenie, očividne nasledujúci krok je prechod na vertikálne.[9] Žiaden problém pre LaTeX. Príkazom

`\raisebox{`*nadvihnutie*`}[`*hĺbka*`][`*výška*`]{`*text*`}`

môžete určiť vertikálne vlastnosti boxu. V prvých troch parametroch môžete použiť **`\width`**, **`\height`**, **`\depth`** a **`\totalheight`**, aby ste ovplyvnili veľkosť boxu pre argument *text*.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{a}%
\raisebox{-2.2ex}{c}%
\raisebox{-4.5ex}{h}}}
 -- zakričal, ale dokonca ani
najbližší v~rade si nevšimol, že
sa mu prihodilo niečo strašné.
```

Aaaaaaaach – zakričal, ale dokonca ani najbližší v rade si nevšimol, že sa mu prihodilo niečo strašné.

5.7 Čiary a rozpery

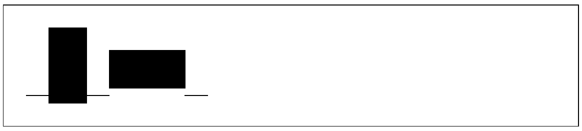
Pred niekoľkými stranami ste si mohli všimnúť príkaz

`\rule[`*nadvihnutie*`]{`*šírka*`}{`*výška*`}`

Pri obyčajnom použití vytvorí jednoduchý čierny obdĺžnik.

[9] Úplnú kontrolu nad všetkým môžeme získať len kombináciou horizontálneho a vertikálneho....

```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```



Toto je užitočné na kreslenie vertikálnych a horizontálnych čiar. Napríklad čiara na titulnej strane bola vytvorená príkazom `\rule`.

Zvláštnym prípadom je čiara s nulovou šírkou, ale s určitou výškou. V profesionálnej typografii sa to anglicky nazýva strut.[10] Používa sa na zabezpečenie toho, aby prvok stránky mal určitú minimálnu výšku. Môžete ju využiť v prostredí `tabular` na zaistenie určitej minimálnej výšky riadku.

```
\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}Viditeľná podpera
\ldots\\ \hline
\rule{0pt}{4ex}Rozpera\\
\hline
\end{tabular}
```

Viditeľná podpera ...
Rozpera

[10] Žiaľ, neviem, aký je slovenský typografický názov pre niečo také. Snáď to vystihuje slovo rozpera. (pozn. prekl.)

Literatúra

[1] Leslie Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.

[2] Donald E. Knuth. *The TeXbook,* Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9.

[3] Michel Goossens, Frank Mittelbach a Alexander Samarin. *The LaTeX Companion.* Addison-Wesley, Reading, Massachusetts, 1994, ISBN 0-201-54199-8.

[4] Každá LaTeXová inštalácia by mala obsahovať tzv. *LaTeX Local Guide* (miestneho sprievodcu programom LaTeX), ktorý popisuje špeciality miestneho systému. Mal by sa nachádzať v súbore `local.tex`. Nanešťastie niektoré lenivé systémy tento dokument neposkytujú. V takom prípade musíte o radu poprosiť miestneho LaTeXového guru.

[5] LaTeX3 Project Team. *LaTeX 2_ε pre autorov.* V inštalácii LaTeX 2_ε ako `usrguide.tex`.

[6] LaTeX3 Project Team. *LaTeX 2_ε pre autorov Tried a Balíkov.* V inštalácii LaTeX 2_ε ako `clsguide.tex`.

[7] LaTeX3 Project Team. *LaTeX 2_ε voľba fontov.* V inštalácii LaTeX 2_ε ako `fntguide.tex`.

[8] D. P. Carlisle. *Balíky zo zväzku ‚graphics'.* So zväzkom ‚graphics' je distribuovaný aj súbor `grfguide.tex`, dostupný z rovnakého zdroja ako inštalácia LaTeX.

[9] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of LaTeX's verbatim Environments.* Je distribuovaný so zväzkom ‚tools' ako `verbatim.dtx`, dostupný z rovnakého zdroja ako inštalácia LaTeX.

[10] Graham Williams. *The TeX Catalogue* je jeden z najúplnejších zoznamov mnohých balíkov súvisiacich s TeXom a LaTeXom. Dostupný online z `CTAN:/tex-archive/help/Catalogue/catalogue.html`

[11] Keith Reckdahl. *Using EPS Graphics in LaTeX 2ε Documents* vysvetľuje všetko a oveľa viac ako by ste si kedykoľvek priali vedieť o EPS-s[boroch a ich použití v LaTeXovských dokumentoch. Dostupný online z `CTAN:/tex-archive/info/epslatex.ps`

[12] J. Rybička. *LaTeX pro začátečníky.* Konvoj, Brno, 1995 (144).

Register