LaTeX
Stručný popis

Hubert Partl*
Elisabeth Schlegl†
Irene Hyna*
Pavel Sýkora‡

Verze 1.1 (č)
25. 1. 1993

LaTeX [1, 2, 4, 5, 6, 7, 8] je systém pro sázení textů, který se hodí zejména pro zhotovování vědecké dokumentace obsahující matematické vzorce. LaTeX můžeme ale použít také pro mnoho dalších druhů písemností — od jednoduchých dopisů až ke kompletní knize. LaTeX je postaven na programu TeX [10, 11, 12, 13, 14, 15].

Tato stručná dokumentace popisuje verzi 2.09 LaTeXu a měla by stačit pro většinu použití LaTeXu. Kompletní popis je uveden v *manuálu LaTeXu* [1] nebo ve výborných publikacích [4, 5].

LaTeX je implementován na nejrůznějších typech počítačů. Jaký je v dané instalaci přístup na LaTeX a které složky jsou k dispozici je popsáno v *Local Guide* [3].

*EDV-Zentrum der Technischen Universität Wien, Abt. Digitalrechenanlage
†EDV-Zentrum der Karl-Franzens-Universität Graz
‡České vysoké učení technické v Praze, fakulta strojní, katedra strojů a zařízení pro strojírenskou výrobu

Tento popis byl zhotoven v LaTeXu pomocí dokumentního stylu (document style) pro článek v češtině a příkazů LaTeXu pro nadpis, obsah, obrázky, tabulky, přehled literatury atd. Popis slouží zároveň jako ukázka možností LaTeXu. Německý originál zdrojových souborů jsou k dispozici na serveru `listserv@dhdurz1.bitnet` *(Heidelberg). Tuto českou verzi lze získat pomocí anonymního FTP z uzlu* `cs.felk.cvut.cs` *nebo elektronickou poštou ze serveru* `mailserv@vax.felk.cvut.cs`*. Oproti německému originálu je kapitola o sazbě německého jazyka nahrazena kapitolou o sazbě češtiny, některé části mírně rozšířeny nebo upraveny a doplněn dodatek o chybách a možnostech rozšíření. Dále byl seznam literatury upraven a rozšířen o další literaturu, zvláště o tu, která vyšla u nás a v Německu.*

Tento překlad vznikl pro potřebu Katedry strojů a zařízení pro strojírenskou výrobu strojní fakulty ČVUT v Praze. Nejsem LaTeXpert, typograf ani germanista. Do překladu jsem se pustil, abych se zdokonalil ve všech těchto oblastech. V dokumentu se však z tohoto důvodu mohou vyskytovat chyby nebo nepřesné formulace. Naleznete-li nějaké, můžete je zaslat na mou adresu.

Pavel Sýkora
ČVUT FSI K 235
Horská 3
128 00 Praha 2
e-mail: `sykora@csearn.bitnet`
`sykora@earn.cvut.cs`

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

1.1 The Name of the Game

1.1.1 TEX

TEX (vyslov „tech“, může se také psát „TeX“) je počítačový program, který sestavil Donald E. Knuth [10, 11, 12, 13, 14, 15]. Tento program slouží k sázení a tisku textů a matematických vzorců.

1.1.2 LATEX

LATEX (vyslov „latech“ nebo „lejtech“, může se také psát „LaTeX“) je tzv. balík maker Leslie Lamporta [1, 2, 4, 5, 6, 7, 8], který využívá TEX. Umožňuje autorovi textu jednoduchým způsobem sázet a vytisknout jeho dokument v kvalitě knihtisku v některém z předem připravených formátů[1].

1.2 Základní koncepce

1.2.1 Autor, typograf a sazeč

Pro publikaci předává autor nakladateli obvykle rukopis psaný na stroji. Typograf nakladatelství pak rozhodne o úpravě písemnosti (délka řádku, druh písma, odstupy před a za kapitolou atd.) a napíše sazeči k tomu nezbytné příkazy.

LATEX je, řekněme, typograf, TEX je jeho sazeč. Příkazy zadané LATEXu se přeloží na úroveň nižších příkazů TEXu.

Typograf-člověk pozná úmysly autora (např. nadpisy kapitol, citace, příklady, vzorce, ...) z obsahu rukopisu hlavně na základě svých odborných znalostí. LATEX naproti tomu je „jen“ program, proto zde potřebuje doplňkové informace autora, které udávají logickou strukturu textu. Tyto informace se zadávají ve formě tzv. „příkazů“ v textu.

V protikladu k tomu stojí příprava písemností pomocí programu pro zpracování textu jako je např. Word, WordPerfect nebo Text602. V tomto případě autor určuje úpravu textu interaktivně. Přitom vidí na obrazovce (údajně) přesně to, co bude vytištěno na stránce. Takové systémy, které podporují optický návrh, se nazývají WYSIWIG („what you see is what you get“).

Při práci s LATEXem autor při psaní vstupního souboru ještě nevidí, jak text bude po formátování vypadat. Může ale po spuštění odpovídajícího počítačového programu pokaždé udělat zkušební výtisk a podle toho svůj vstupní soubor patřičně upravit a pokračovat v práci.

> Pozn.: Na některých typech počítačů (např. osobní počítače kompatibilní s IBM PC, MacIntosh aj.) je k dispozici tzv. funkce

[1] tzv. dokumentní styly (angl. document styles) — jsou to soubory s příponou `.STY`

„preview“, která umožňuje i dosti podrobné prohlížení výsledného dokumentu na grafické obrazovce.

1.2.2 Úprava textu

Typografický návrh je ruční práce, kterou se musí člověk naučit. Necvičení autoři dělají často vážné chyby formátování. Mnoho laiků se nesprávně domnívá, že tento návrh je především otázka estetiky — když dokument vypadá dobře z umělecké stránky, je dobře navržen. Poněvadž ale dokumenty nebudou pověšeny v muzeu, nýbrž je lidé budou číst, je snadná čitelnost a dobrá srozumitelnost důležitější než umělecký vzhled.

Například bychom měli zvolit takovou výška písma a číslování nadpisů, aby byla zřetelná struktura kapitol a podkapitol. Délky řádků bychom měli zvolit takové, aby nedocházelo k namáhavým pohybům očí čtenáře, nikoliv takové, aby byl papír co nejkrásněji zaplněn.

Pomocí interaktivních systémů autoři obvykle vytvářejí dokumenty hezčí z estetického hlediska, ale špatně strukturované. LaTeX takovým chybám formátování zabraňuje, neboť je v něm autor nucen zadat logickou strukturu textu a pak pro něj použít nejvhodnější formát.

1.2.3 Výhody a nevýhody

Oproti ostatním programům pro zpracování textu se LaTeX vyznačuje následujícími výhodami:

- Je k dispozici několik profesionálně vytvořených formátů, se kterými dokumenty vypadají „jako profesionálně vytištěné“.
- Zvlášť dobře je podporována sazba matematických vzorců.
- Uživatel musí zadávat jen několik lehce srozumitelných příkazů, které se týkají logické struktury dokumentu, a (téměř) nepotřebuje zabývat se technickými detaily tisku.
- Bez velké námahy mohou být vytvořeny také složité struktury jako poznámky pod čarou, seznamy literatury, obsahy, tabulky atd. stejně jako odkazy na stránku, čísla kapitol tabulek, obrázků, rovnic atd.

LaTeXmá ovšem také nevýhody:

- Dokonalejší algoritmy pro formátování kladou větší nároky na čas zpracování a paměť než u jednodušších programů pro zpracování textu.
- Výstup textu je možný pouze na grafických zařízeních (laserové, inkoustové nebo jehličkové tiskárny, grafické obrazovky), nikoliv na levných znakově orientovaných rychlotiskárnách.

- Ačkoli lze v předem připravených dokumentních stylech snadno přizpůsobit jednotlivé parametry, zásadní změny předem stanovených formátů vyžadují větší námahu (návrh nového dokumentního stylu).

1.3 Vstupní soubor

Vstupní soubor LaTeXu je editorem připravený textový soubor obsahující jak text, který má být vytištěn, tak i „příkazy“, pomocí kterých se LaTeX dozví, jak má být text vysázen.

1.3.1 Nastavování mezer

„Neviditelné“ znaky jako mezera (blank) a konec řádku (carriage return) zpracovává LaTeX jako mezeru. *Více* mezer je zpracováno jako *jedna* mezera. Prázdný řádek mezi textovými řádky označuje konec odstavce. *Více* prázdných řádků je zpracováno jako *jeden* prázdný řádek.

Pokud autor požaduje jiné rozestupy slov nebo řádků než normální, nedosáhne jich zadáváním dodatečných mezer nebo prázdných řádků, nýbrž pouze odpovídajícími příkazy LaTeXu (viz kap 4.2).

1.3.2 Speciální znaky

Následující symboly jsou rezervované znaky, které mají v LaTeXu zvláštní význam nebo je nelze použít ve všech druzích písma:

```
$ & % # _ { }  ~  ^ "  \  | < >
```

Následující znaky mohou být vytištěny zadáním dvojice znaků „obrácené lomítko“ (\) a požadovaného znaku:

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

Zbývající symboly a ještě mnoho dalších znaků lze vytisknout speciálními příkazy jako akcenty (diakritická znaménka) nebo v matematických vzorcích[2].

1.3.3 Příkazy LaTeXu

Většina příkazů LaTeXu má následující formát: Buď začínají znakem „obrácené lomítko“ (\) a mají název vytvořený pouze z písmen, nebo se skládají z obraceného lomítka a právě jednoho zvláštního znaku nebo číslice. Velká a malá písmena mají v názvu příkazu *odlišný* význam. Název příkazu složený z písmen musí být oddělen od následujícího textu (tj. ukončen) jednou nebo

[2] V LaTeXu existuje také prostředí `verbatim` (viz kap. 2.9.4), ve kterém se všechny znaky včetně vícenásobných mezer opisují do výstupního textu.

více mezerami, zvláštním znakem nebo číslicí. Pokud chceme za názvem příkazu dosáhnout mezeru, musíme název ukončit pomocí `{}` nebo zvláštního příkazu pro prázdnou pozici.

Dnes je 1. září 1987. Nebo: Dnes je 1. září 1987. Špatně: Dne 1. září 1987prší. Správně: Dne 1. září 1987 sněží. Nebo: Dne 1. září 1987 neprší.

```
Dnes je \today.
Nebo: Dnes je \today .
Špatně: Dne \today prší.
Správně: Dne \today{} sněží.
Nebo: Dne \today\ neprší.
```

Mnoho příkazů má parametry, které se uzavírají mezi složené závorky. Mnoho příkazů má parametry, které lze vynechat nebo uzavřít mezi hranaté závorky. Mnoho příkazů má variace, které se rozliší přidáním hvězdičky za název příkazu.

Složené závorky mohou být také použity k vytvoření „skupin“. Účinnost příkazů, které byly zadány uvnitř skupin nebo „prostředí“, končí vždy na konci skupiny nebo prostředí. Ve výše uvedeném příkladu je `{}` prázdná skupina, která nemá kromě ukončení názvu příkazu `\today` žádný účinek.

1.3.4 Komentáře

Vše, co stojí za znakem procento (`%`) až na konec řádku vstupního souboru, je LaTeXem ignorováno. Toho lze využít pro poznámky a komentáře, které nemají být vytištěny.

To je příklad.

```
To je % hloupý
% Lépe: poučný <----
příklad.
```

1.3.5 Struktura vstupního souboru

První příkaz[3] vstupního souboru pro LaTeX musí být příkaz

```
\documentstyle
```

(viz kap. 1.4.1). Pak mohou následovat další definice, které mají platit pro celý dokument. Příkazem

```
\begin{document}
```

začíná sázení dokumentu. Zde následuje text a všechny příkazy LaTeXu, které mají vliv na tisk dokumentu. Vstupní soubor musí být zakončen příkazem

[3] V tomto popisu se autoři občas dopouštějí drobných nepřesností pro zjednodušení. Toto je jeden z příkladů. Nebuďte proto překvapeni, naleznete-li ve vstupním souboru pro LaTeX před příkazem `\documentstyle` některé speciální příkazy, a pamatujte na přísloví „Na slova učitelova nepřísahej!“

\end{document}.

Cokoliv následuje za tímto příkazem, je LaTeXem ignorováno.

Obr. 1 ukazuje minimální vstupní soubor pro LaTeX. O něco složitější soubor je na obr. 2.

```
\documentstyle{article}
\begin{document}
Small is beautiful.
\end{document}
```

Obrázek 1: Minimální vstupní soubor LaTeXu

1.4 Vzhled dokumentu

1.4.1 Dokumentní styly

Na začátku každého souboru musí být definován typ dokumentu příkazem

\documentstyle[*volby*]{*styl*}.

Mezi složenými závorkami *musí* být zadán název jednoho dokumentního stylu (document style) z tabulky 1.

Tabulka 1: Dokumentní styly

`article` pro články v odborných časopisech, přednášky, seminární práce, kratší zprávy, návrhy, posudky, manuály k programům atd.

`report` pro delší zprávy, které obsahují více kapitol, diplomové a disertační práce, skripta aj.

`book` pro knihy

`letter` pro dopisy

Mezi hranaté závorky se *může* uvádět jedna nebo více „voleb“ vytvářejících variace a doplňky standardního dokumentního stylu. Nejdůležitější volby jsou uvedeny v tabulce 2.

Příklad: Vstupní soubor pro tento dokument začíná příkazem

```
\documentstyle[11pt,czech,twoside]{article}
```

```
\documentstyle[11pt,czech]{article}
\author{N.~I.~Kdo}
\title{Krátký nebo dlouhý}
\setlength{\parindent}{0pt}
\setlength{\parskip}{5pt plus 2pt minus 1pt}
\sloppy

\begin{document}
\maketitle
\begin{abstract}
Příklad odborného článku v~češtině.
\end{abstract}
\tableofcontents

\section{Začátek}

Zde začíná moje krásná práce\dots

\section{Konec}

\dots\ a zde končí.

\end{document}
```

Obrázek 2: Struktura článku

Tabulka 2: Volby k dokumentním stylům

`11pt` pro písmo vysoké 11 bodů — jako v tomto dokumentu.

`12pt` pro písmo vysoké 12 bodů — asi jako na psacím stroji

Pokud není zadána žádná z výše uvedených voleb, použije LaTeX písmo vysoké 10 bodů, což je obvyklá velikost při knihtisku.

`fleqn` matematické rovnice zarovná nalevo místo centrování

`leqno` číslování rovnic vlevo místo vpravo od rovnice

`titlepage` pro samostatnou titulní stranu (pouze pro typ `article`)

`twocolumn` pro tisk do dvou sloupců

`twoside` pro oboustranný tisk (rozdílné okraje pravé a levé strany pro vazbu)

`czech` pro dokumenty v češtině (viz kap. 2.5)

`german` pro dokumenty v němčině

1.4.2 Formát stránky

Formát stránky *můžeme* nastavit příkazem

`\pagestyle{`*styl*`}`:

Parametr `plain` (nebo pokud příkaz `\pagestyle` není vůbec zadán) určuje, že čísla stran budou tištěna na spodním okraji. Zadání parametru `headings` udává tištění názvu kapitoly a čísla strany na horním okraji. Parametr `empty` potlačuje číslování stran.

V *manuálu LaTeXu* [1] je uvedeno, jak se řídí formát stránky a jejich číslování příkazy `\thispagestyle`, `\pagenumbering`, `\twocolumn` a `\onecolumn`.

2 Sázení textu

2.1 Lámání řádků a stránek

2.1.1 Sazba do bloku

Normální text se sází do bloku, tj. se zarovnaným levým i pravým okrajem. LaTeX automaticky láme řádky a stránky. Přitom se pro každý odstavec určí nejlepší rozvržení slov na řádky a — pokud je to nezbytné — se slova automaticky rozdělí.

Konce slov a vět jsou rozpoznány podle mezer. Nezáleží na tom, zda je zadána jedna nebo 100 mezer.

Jeden nebo více prázdných řádků označuje konec odstavce.

```
Konce slov a vět jsou
rozpoznány podle mezer.
Nezáleží na tom, zda je
zadána jedna nebo       100
mezer.

Jeden nebo více prázdných
řádků označuje konec
odstavce.
```

Jak budou odstavce vysázeny, závisí na dokumentním stylu. V článcích, zprávách a knihách je vždy první řádek odstavce odsazen — jako ve všech ostatních kapitolách tohoto popisu. V dopisech nebo při definici

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{5pt plus 2pt minus 1pt}
```

se odstavce oddělují vertikálně volným prostorem bez odsazení prvního řádku — jako v této kapitole. Jako vedlejší efekt se mění formátování odstavců také v prostředích a při sázení matematických rovnic.

Pomocí prostředí popsaných v kap. 2.9 můžeme sázet zvláštní části textu jinak.

Kromě toho lze ovlivnit lámaní řádků a stránek těmito příkazy: Příkazem `\\` nebo `\newline` je možno přejít na nový řádek aniž se ukončí odstavec, příkaz `\\*` zajistí totéž, avšak v tomto místě nesmí dojít k zlomení stránky. Příkaz `\newpage` zlomí stránku. Pomocí příkazů `\linebreak[`n`]`, `\nolinebreak[`n`]`, `\pagebreak[`n`]` a `\nopagebreak[`n`]` lze zadat, zda na určitém místě je zlomení řádku nebo stránky spíše příznivé nebo spíše nepříznivé. Míru vhodnosti nebo nevhodnosti udává parametr n (1, 2 nebo 3).

LaTeX se snaží lámat řádky tak, aby byly co nejhezčí. Pokud nenajde žádnou přísným pravidlům odpovídající možnost, ponechá řádek delší a vydá o tom chybové hlášení („overfull hbox"). Tato situace nastává zejména tehdy, když nenalezne žádné vhodné dělení slov. Příkazem `\sloppy` lze zadat, aby byl LaTeX ve svých nárocích méně přísný a vyhnul se takovým přeplněným řádkům tím, že se zvětší rozestupy slov — bude-li třeba až nehezky. V tomto případě ovšem jsou hlášena varování („underfull hbox"), výsledek je ale většinou docela upotřebitelný.

2.2 Dělení slov

LaTeX dělí slova automaticky. Pokud jednotlivé výsledky automatického dělení slov nejsou správné, můžete tyto „výjimky" napravit níže uvedenými příkazy. Obzvláště to může být nutné při použití cizojazyčných slov nebo např. v němčině při skládání slov.

Příkaz `\hyphenation` oznamuje, že v něm uvedená slova se mohou *vždy* rozdělit jen a pouze v místech označených `-`. Příkaz by se měl zadávat na začátku vstupního souboru. Příkaz se nehodí pro slova s akcenty zadanými pomocí příkazů LaTeXu a pro slova se speciálními znaky nebo s číslicemi. Příklad:

```
\hyphenation{ kniho-vna,
  kniho-vny, FORTRAN }
```

Příkazem `\-` zadáváme, že dané slovo se *tentokrát* může rozdělit jen a pouze v místech označených `\-`. Tento příkaz se hodí i pro slova obsahující číslice, zvláštní znaky nebo příkazy pro akcenty.

pododdělení, podražit, podrážet

```
pod\-od\-dě\-le\-ní,
po\-dra\-žit,
pod\-rá\-žet
```

Pro speciální případy slouží příkaz `\discretionary`. Je jím možno popsat možnost rozdělování slov se změnou pravopisu. Mohl by se uplatnit například v německém textu (rozdělování slov ve skupině „ck" apod.), avšak v LaTeXu při přepnutí do německé varianty existují pohodlnější možnosti.

Příkaz `\mbox{...}` označuje, že argument se nemůže dělit.

Naše telefonní číslo už není (0222) 5601-3694.
Parametr *filename* určuje jméno souboru.

```
Naše telefonní číslo už není
\mbox{(0222) 5601-3694}.
Parametr \mbox{\it filename\/}
určuje jméno souboru.
```

2.3 Speciální znaky

2.3.1 Uvozovky

Píšete-li text na psacím stroji, můžete jako uvozovky použít pouze znak „uvozovky nahoře" (`"`). V knihtisku se však otevírací a uzavírací uvozovky sázejí různými znaky. Navíc jsou otevírací i uzavírací uvozovky v anglickém textu odlišné od příslušných uvozovek v textu českém, slovenském, německém apod. Ve vstupním souboru s anglickým textem se otevírací uvozovky zadávají jako dva znaky „obrácený apostrof“, uzavírací jako dva znaky „apostrof“. Zapamatujte si, že uvozovky v textu se *nikdy* nezadávají pomocí klávesy `"`. Při úpravách textů původně určených pro jiný sázecí systém nebo textový procesor nezapomeňte takto zadané uvozovky opravit. Špatně zadané uvozovky nevypadají v hotovém dokumentu vůbec dobře.

“No,” he said, “I don’t know!”	`` ``No,'' he said, ``   `` ``I don't know!'' ``

Uvozovky v češtině, němčině apod. vypadají jinak a mohou se zadávat několika způsoby (viz kap. 2.5.3).

2.3.2 Pomlčky a podobné znaky

V knižním tisku se používají pomlčky různých délek. Nejkratší je *spojovník*, pak následuje krátká (půlčtverčíková) pomlčka, pak dlouhá (čtverčíková) pomlčka. Poněkud odlišné je i matematické znaménko „mínus“. Chceme-li, aby hotový dokument vypadal esteticky, je nezbytné pomlčky používat správně. Krátkou pomlčku používáme pro znázornění intervalu. Neodděluje se mezerou nebo jen zúženou. Dlouhá pomlčka naznačuje přestávku v řeči nebo od sebe výrazně odděluje části textu; významově je rovna čárce. V českých a německých textech se sází s mezerami z obou stran, v anglických bez mezer. Při řádkovém zlomu zůstává na konci řádku, nový řádek jí nesmí začínat.

půjde-li to	`půjde-li to \\`
10–18 hodin	`10--18~hodin \\`
ano — nebo ne?	`ano --- nebo ne? \\`
yes—or no?	`yes---or no? \\`
0, 1 a -1	`0, 1 a $-$1`

2.3.3 Tečky (dots)

Na rozdíl od strojopisného textu, kde každá tečka nebo čárka zabírá stejnou šířku jako ostatní znaky, při knihtisku se tečky a čárky sázejí užší. Pro tři tečky ve smyslu „a tak dále“ se zde používá příkazu `\ldots` nebo `\dots`.

Takhle ne … nýbrž takto:	`Takhle ne ... nýbrž takto: \\`
Praha, Plzeň, …	`Praha, Plzeň, \dots`

2.3.4 Slitky a podřezávání

V knižním tisku je zvykem sázet některé kombinace písmen jako jeden znak — tzv. *slitky* neboli *ligatury*.

ﬀ ﬁ ﬂ ﬃ... místo ff fi fl ffi...

Podřezávání (kerning) je přiblížení některých dvojic znaků k sobě (přesněji řečeno přes sebe), aby text opticky vypadal lépe.

AV Te... místo AV Te...

Program LaTeX (resp. TeX) vytváří slitky i podřezávání automaticky. Slitky se však nepoužívají ve složených slovech (nebo ve slovech s předponou či příponou), jejichž části se stýkají kombinací písmen ff, fl apod. Z mně známých jazyků se jedná pouze o velmi málo anglických slov a nemnoho slov německých (v češtině jsem na žádné takové slovo nepřišel). V těchto případech je třeba použít příkaz `\/` nebo `{\kern0pt}`.

Nikoli shelﬀul nebo Auﬂage (Au-ﬂ-age), nýbrž shelfful a Auflage (Auf-lage)	`Nikoli shelfful nebo Auflage (Au-fl-age),\\` `nýbrž shelf\/ful a Auf\/lage (Auf-lage)`

2.3.5 Akcenty a speciální znaky

LaTeX umožňuje použít akcenty (diakritická znaménka) a zvláštní znaky mnohých jazyků (viz tab. 3). Akcenty jsou zde ukázány na příkladě písmena o, ovšem principiálně není vyloučeno jejich použití s kterýmkoli jiným písmenem. Pokud má být akcent vysázen nad písmenem i nebo j, musí být použito písmeno „i bez tečky" nebo „j bez tečky". Tyto znaky se vysázejí pomocí příkazu `\i` nebo `\j`.

Hôtel, naïve, smørebrød.
¡Señorita!
Brüel & Kjær
	`H\^otel, na\"\i ve,`
	`sm\o rebr\o d. \\`
	`!'Se\~norita!\\`
	`Br\"uel \&\ Kj\ae r`

2.4 Mezery

Aby bylo dosaženo zarovnaného pravého okraje, LaTeX zvětšuje nebo zmenšuje mezery mezi slovy. Při sazbě anglických textů se navíc sází po tečkách, otaznících apod. na konci věty větší mezera, což zvyšuje čitelnost textu. V českých a německých textech se sázejí mezery stejné. LaTeX předpokládá, že tečka, která následuje za velkým písmenem, označuje zkrácení (např. křestního jména), a že všechny ostatní tečky ukončují větu.

Výjimky z těchto pravidel musíme LaTeXu sdělit následujícími příkazy: Obrácené lomítko následované mezerou označuje, že mezera ve výsledném textu v tomto místě nesmí být širší. Znak `~` (vlnka, tilda) označuje mezeru, která nesmí být širší a v tomto místě nesmí dojít k řádkovému zlomu. V českých textech bychom neměli nechávat jednohláskové předložky a spojky i, Tabulka 3: Akcenty a zvláštní znaky

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

I, A, na konci řádku. Za těmito spojkami a předložkami tedy místo mezery píšeme vlnku. Existuje i program, který tuto činnost provádí automaticky. Další případ nepovoleného řádkového zlomu, kdy tedy musíme použít vlnku, je například mezera mezi číselnou hodnotou a jednotkou, mezera mezi titulem nebo zkratkou jména a příjmením apod. Příkaz `\@` před tečkou označuje, že touto tečkou končí věta, ačkoli je před ní velké písmeno.

Firma Čtverec a spol. Vám dodá kval. kolečka.
... 5 m široký.
Dr. Šedý bydlí v 1. patře.
Potřebuji vitamín C. Ty ne?

```
Firma Čtverec a spol.\ Vám
dodá kval.\ kolečka.  \\
\dots\ 5~m široký. \\
Dr.~Šedý bydlí v~1.~patře. \\
Potřebuji vitamín~C\@.
Ty ne?
```

Jinou možností je použít příkazu `\frenchspacing`, který zajistí stejnou mezeru na konci věty jako mezi slovy. Tato konvence se používá při sázení jiných textů než anglických. V tomto případě není nutno používat příkazy `\ ` a `\@`. Při vytváření českých textů se používá předdefinovaný formát `czech.sty`, který příkaz `\frenchspacing` aktivuje automaticky. K obnovení konvence sázení mezer anglických textů slouží příkaz `\nonfrenchspacing`.

2.5 Sazba českých textů

V této kapitole je popsáno zpracování českých textů systémem $\mathcal{CS}$TEX (systém emTEX upravený pro češtinu), který šíří Československé sdružení uživatelů TEXu (CSTUG).

Originální americká verze TEXu podporuje pouze dokumenty v angličtině. Původní systém emTEX vznikl v Německu, podporuje tedy dokumenty anglické i německé. $\mathcal{CS}$TEX podporuje standardně češtinu a angličtinu, navíc jej snadno lze upravit pro podporu dokumentů v češtině, angličtině i němčině.

Při zpracování dokumentů různými instalacemi LATEXu bude nutné připojit použité nestandardní styly. Vstupní soubory s českými znaky by navíc měly být převedeny na kód sedmibitový (ASCII znaky s příkazy pro akcenty).

2.5.1 Dělení slov

Při spuštění LATEXu je určeno, zda se bude dělit podle pravidel českého, anglického, německého případně jiného jazyka. Výjimky lze v daném dokumentu zadat pomocí příkazů uvedených v kap. 2.2. Při správném vytváření vícejazyčných dokumentů se pravidla pro dělení přepínají podle příslušného jazyka.

2.5.2 Písmena s akcenty

Původní verze LATEXu obsahuje jak příkazy pro umístění akcentů nad (v některých jazycích i pod) písmena, tak i umožňuje použití písmen speciálních. Nicméně používat tyto příkazy je dosti nepohodlné. Dokumenty zpracovávané LATEXem obsaženým v $\mathcal{CS}$TEXu lze proto při použití volby `czech` v příkazu `\documentstyle` na počítačích kompatibilních s IBM PC používat rozšířenou sadu ASCII znaků — například u nás nejpoužívanější kód Kamenických. Toto řešení umožňuje použít dělicí algoritmus a příkaz `\hyphenation` i na slova s diakritickými znaménky.

2.5.3 Uvozovky

Uvozovky v českém textu vypadají jinak než v anglickém. Pokud použijeme volbu `czech`, můžeme použít pro české uvozovky příkaz `\uv{`*text, který má být v uvozovkách*`}`. Pro anglické uvozovky platí původní příkazy — ``` `` ``` pro otevírací a `''` pro zavírací. Jinou možností je na začátku textu použít příkaz `\csprimeson`, po kterém příkazy ``` `` ``` (dva obrácené apostrofy) a `''` (dva apostrofy) vytvářejí uvozovky české. V textu se ovšem mohou vyskytnout slova s apostrofem (např. l'Hospitalovo pravidlo), který by byl v tomto případě vysázen nesprávně. Pak je možné použít příkaz `\csprimesoff`, který obnoví původní chování příkazů ``` `` ``` a `''` a tedy i apostrofu. Jiné možnosti vytváření a kombinací anglických a českých uvozovek jsou uvedeny v dokumentaci ke stylu `czech` (soubor `czech.doc` nebo `czech.sty`).

„Ne,“ řekl, „nevím nic!“	`\uv{Ne,} řekl,` `\uv{nevím nic!}`

2.5.4 Nadpisy a data

V originální verzi LaTeXu jsou použity pro nadpisy kapitol, obsahu, označení tabulek a obrázků a názvy měsíců v datech anglické termíny. Jejich použití by bylo v českém textu krajně nevhodné. Použitím stylu `czech` se tyto názvy automaticky změní na jejich české ekvivalenty. Existují sice i jiné cesty, jejich popis však překračuje rámec tohoto popisu.

2.6 Kapitoly a nadpisy (section)

Začátek kapitoly nebo podkapitoly a jejich nadpis se zadává příkazy `\section{...}`. Přitom se musí zachovat logická struktura.

U článků:

```
\section  \subsection  \subsubsection
```

U zpráv a knih:

```
\chapter  \section  \subsection  \subsubsection
```

Články mohou být tedy relativně snadno vloženy do knihy jako kapitoly.

Odsazení mezi kapitolami, číslování a velikost písma určuje LaTeX automaticky.

Nadpis celého článku nebo titulní stranu dokumentu se vysází příkazem `\maketitel`. Obsah titulní strany nebo nadpisu musí být předem zadán příkazy `\titel`, `\author` a `\date`.

Příkaz `\tableofcontents` způsobí vytištění obsahu. Nadpisy a čísla stran se dozví LaTeX z předcházejícího zpracování vstupního souboru. Proto při úpravách nebo doplnění kapitoly je nutno zpracovat dokument programem LaTeX dvakrát (někdy i třikrát), aby bylo dosaženo správného výstupu.

Příkaz `\section` existuje také ve formě `\section*{...}`, kdy se nadpis nečísluje a není uveden v obsahu.

Příklady lze nalézt na obr. 2 na str. 11.

Pomocí příkazů `\label` a `\ref` se lze odkázat na čísla kapitol automaticky vytvořená LaTeXem. Místo `\ref{...}` vysází LaTeX číslo kapitoly, obrázku nebo tabulky označené pomocí `\label{...}`. Také v tomto případě se berou informace o číslech kapitol apod. z předešlého zpracování. Příklad:

```
\section{Algoritmy}
...
Toto je dokázáno v~kapitole~\ref{dukaz}.
...
\section{Důkazy} \label{dukaz}
...
```

2.7 Poznámky (footnote)

Poznámky[4] pod čarou jsou automaticky číslovány a jsou vytištěny vždy v dolní části stránky.

```
Poznámky\footnote
{Toto je příklad}
pod čarou jsou
automaticky ...
```

2.8 Zdůrazněná slova (emphasize)

Ve strojopisných textech bývá zdůrazněný text podtržen, v knižním tisku se k tomu používají různé druhy písma. Příkaz `\em` (emphasize) přepíná na „zdůrazněný“ typ písma. Tento typ písma zůstává až do konce aktuální skupiny (nebo do jiného příkazu změny písma), proto příkaz `\em` musí být *uvnitř* složených závorek.

Tyto závorky jsou *před* příkazem, *ne za* ním.

```
Tyto závorky jsou {\em před\/}
příkazem, {\em ne za\/} ním.
```

LaTeXpoužívá pro zdůrazněný text *kurzívu.* V kurzívě jsou všechny znaky skloněné doprava, proto na konec se dává příkaz `\/`, který zajišťuje, aby poslední skloněné písmeno nezasahovalo do následujícího kolmého písmena nebo aby nezmenšilo následující mezeru (tzv. kurzívní korekce). Kurzívní korekce se nepoužívá, následuje-li tečka nebo čárka.

Šel *podél* lesa.
Šel *podél* lesa.
*pod*le
*pod*le

```
Šel {\em podél} lesa.
Šel {\em podél\/} lesa.
{\em pod}le
{\em pod\/}le
```

Pokud použijeme uvnitř zdůrazněného textu znovu *zdůraznění, LaTeX použije* kolmé *písmo.*

[4] Toto je příklad

2.9 Prostředí (environment)

Pro označení zvláštních částí textu, které mají být vysázeny jinak než odstavec do bloku, slouží tzv. „prostředí“ (environment) ve formě:

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

Každé prostředí se chová jako *skupina*. Prostředí mohou být do sebe vnořena, přičemž však je třeba dbát na správné řazení:

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

2.9.1 Citace (quote, quotation, verse)

Prostředí `quote` se hodí pro kratší citace, zdůrazněné věty a příklady. Text se sází užší zleva i zprava o jednu zarážku:

Jedno typografické pravidlo pro délku řádku říká:

> Žádný řádek by neměl obsahovat více než 66 písmen.

Kvůli tomu se v novinách používá více sloupců vedle sebe.

```
Jedno typografické pravidlo
pro délku řádku říká:
\begin{quote}
Žádný řádek by neměl
obsahovat více než 66 písmen.
\end{quote}
Kvůli tomu se v~novinách
používá více sloupců vedle
sebe.
```

Prostředí `quotation` se hodí pro delší citace, které se skládají z více odstavců.

Prostředí `verse` se hodí pro básně a pro příklady, ve kterých je předem známé dělení na řádky. Verše (řádky) se dělí pomocí `\\`, sloky pomocí prázdného řádku.

2.9.2 Výčty (itemize, enumerate, description)

Prostředí `itemize` se hodí pro jednoduché výčty (viz obr. 3). Prostředí `enumerate` se hodí pro číslované výčty (viz obr. 4). Prostředí `description` se hodí pro popisy (viz obr. 5). Výčty se mohou do sebe vnořovat až do čtvrté úrovně.

2.9.3 Sazba na prapor a centrování (flushleft, flushright, center)

V prostředí `flushleft` resp. `flushright` se sázený text doráží na levý resp. pravý okraj (tzv. „sazba na prapor“ — tj. bez zarovnání okraje), prostředí `center` text umisťuje na osu. Jednotlivé řádky se dělí příkazem `\\`. Pokud nepoužijeme příkaz `\\`, určí LaTeX řádkové zlomy sám. V těchto prostředích se slova nerozdělují.

Výčty:

- U `itemize` jsou položky vyznačeny body nebo jinými symboly.
- Výčty mohou být i vnořeny:
  - Maximální úroveň vnoření je 4.
  - Odsazení a označení položek se mění automaticky.
- atd.

```
Výčty:
\begin{itemize}
\item U~{\tt itemize} jsou
položky ...
\item Výčty mohou být
i vnořeny:
  \begin{itemize}
  \item Maximální ...
  \item Odsazení ...
  \end{itemize}
\item atd.
\end{itemize}
```

Obrázek 3: Příklad na `itemize`

Výčty:

1. U `enumerate` jsou položky označeny čísly nebo písmeny.
2. Číslování je automatické.
3. Výčty mohou být také vnořeny:
   (a) Maximální úroveň vnoření je 4.
   (b) Označování a odsazení položek se mění automaticky.
4. atd.

```
Výčty:
\begin{enumerate}
\item U~{\tt enumerate} jsou
položky ...
\item Číslování ...
\item Výčty ...
  \begin{enumerate}
  \item Maximální ...
  \item Označování ...
  \end{enumerate}
\item atd.
\end{enumerate}
```

Obrázek 4: Příklad na `enumerate`

Malá zoologie:
 	
Had: u nás malé zvíře, které se plazí po zemi.
 	
Lev: velké zvíře, žijící v poušti a v zoologických zahradách.
 	
Ptakopysk: středně velké zvíře, které je zde jen kvůli délce jména.
	`Malá zoologie:`
	`\begin{description}`
	`\item[Had:]`
	`   u~nás malé ...`
	`\item[Lev:]`
	`   velké zvíře ...`
	`\item[Ptakopysk:]`
	`   středně ...`
	`\end{description}`

Obrázek 5: Příklad na `description`

vlevo
nalevo
	`\begin{flushleft}`
	`vlevo \\`
	`nalevo`
	`\end{flushleft}`

vpravo
napravo
	`\begin{flushright}`
	`vpravo  \\`
	`napravo`
	`\end{flushright}`

V
říši
středu
	`\begin{center}`
	`V~\\ říši \\ středu`
	`\end{center}`

2.9.4 Přímý výstup (verbatim, verb)

Řádky nacházející se mezi `\begin{verbatim}` a `\end{verbatim}` jsou vytištěny tak, jak byly zadány, tj. se všemi mezerami a ukončeními řádků a bez interpretace zvláštních znaků i příkazů LaTeXu. To se hodí např. pro vytištění krátkých počítačových programů.

Jednotlivé kombinace znaků nebo krátké úseky textu v odstavci mohou být také vytištěny „doslova“, pokud se uvedou mezi `\verb|` a `|`. Těmito příkazy jsou například vysázeny všechny příkazy LaTeXu v tomto manuálu.

Příkaz `\dots` …	`Příkaz \verb|\dots| \dots`

Prostředí `verbatim` a příkaz `\verb` se nesmějí používat v parametrech jiných příkazů a také v prostředí `tabular`.

Obrázek 6: R. Černý: Celý v bílém

2.9.5 Obrázky (figure)

Text mezi `\begin{figure}` a `\end{figure}` (nebo místo pro nalepení obrázku vytvořené příkazem `\vspace`) se automaticky vysází na místo, kam se vejde celý, aniž by byl přerušen stránkovým zlomem. Příkazem `\caption{...}` se sází popis obrázku. Zadává se pouze text popisu, zkratku „Obr.“ a číslo obrázku doplní LaTeX. U obrázku je zvykem označení *pod* obrázek. Pomocí příkazů `\label` a `\ref` se můžeme odkazovat v textu na číslo obrázku.

Obr. 6 ukazuje příklad tzv. Pop-Art.

```
Obr.~\ref{bily} ukazuje
příklad tzv. Pop-Art.
\begin{figure}
\vspace{6cm}
\caption{R.~Černý: Celý
  v~bílém} \label{bily}
\end{figure}
```

2.9.6 Tabulky (table)

Tabulky se sázejí podobně jako obrázky, avšak mezi `\begin{table}` a `\end{table}`, také příkazy `\caption`, `\label` a `\ref` se chovají podobně. Označování tabulek je možné oběma konvencemi — označení se sází vždy *nad* nebo vždy *pod* tabulku.

Pro vysázení vlastní tabulky se používá hlavně prostředí `tabbing` nebo `tabular`, které jsou popsány níže. Mohou se však použít i jiné struktury (např. `enumerate` nebo `description`). Obr. 7 obsahuje příklad takového vnoření prostředí.

```
\begin{table}
\caption{...} \label{...}
  \begin{center}
    \begin{tabular}{...}
    ...
    \end{tabular}
  \end{center}
\end{table}
```

Obrázek 7: Použití prostředí `table`

2.9.7 Tabulátory (tabbing)

V prostředí `tabbing` se mohou nastavovat a používat tabulátory podobně jako na psacím stroji. Příkaz `\=` nastavuje tabulační pozici, `\kill` znamená, že „vzorový řádek“ nemá být vytištěn, `\>` skočí na další pozici tabulátoru a `\\` dělí řádky.

IČO	Název	Kurs akc.
451926	Silnice Ostrava	51/100
451954	Ingstav Opava	45/100

```
\begin{tabbing}
444444\quad \= Silnice
Ostrava\quad \= \kill
IČO \> Název \> Kurs akc.\\
451926 \> Silnice Ostrava
\> 51/100\\ 451954 \>
Ingstav Opava \> 45/100
\end{tabbing}
```

2.9.8 Vlastní tabulky (tabular)

Prostředí `tabular` slouží k sázení tabulek, kde LaTeX automaticky určí potřebné šířky sloupců. Lze u nich zajistit i speciální vlastnosti jako zarovnání vpravo nebo pomocné čáry.

V parametru příkazu `\begin{tabular}{...}` se zadává formát. Zde znamená `l` sloupec s textem zarovnaným vlevo, `r` sloupec s textem zarovnaným vpravo, `c` s centrovaným textem, `p{`*šířka*`}` sloupec zadané šířky s víceřádkovým textem (text se zformátuje do odstavce), `|` svislou čáru.

V tabulce znamená `&` skok na další sloupec tabulky, `\\` dělí řádky, `\hline` (jako samostatný řádek) sází vodorovnou čáru v šířce tabulky.

7C0	hexadecimálně
3700	oktalově
11111000000	binárně
1984	decimálně

```
\begin{tabular}{|rl|}
\hline
7C0 & hexadecimálně \\
3700 & oktalově \\
11111000000 & binárně \\
\hline
1984 & decimálně \\
\hline
\end{tabular}
```

2.10 Linky, rámečky, diagramy (rule, picture)

LaTeX obsahuje příkazy i pro tvorbu linek, rámečků a jednoduchých diagramů složených také z šipek a kroužků malého průměru.

Pro vytváření linek (mimo prostředí `tabular`) slouží příkaz `\rule` se dvěma parametry. Tento příkaz vytváří černý pravoúhelník, jehož šířku udává první a výšku druhý parametr. Pokud jeden z parametrů má hodnotu „0cm“, na výstupu se objeví „neviditelná“ linka, což můžeme využít pro vyhrazení místa. Příkaz může mít i jeden nepovinný parametr (pak se uvádí v hranatých závorkách jako první), který určuje zdvih linky nad základní čáru.

```
\rule{3cm}{1mm}\\
\rule{0.5mm}{5mm}
```

Další příkazy používáme v prostředí `picture`.

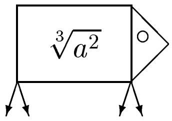
```
\setlength{\unitlength}{1mm}
\begin{picture}(x,y)
\end{picture}
```

Před použitím tohoto prostředí je nutno definovat „jednotkovou délku“. Souřadnice a délky se pak uvádějí jako parametry v kulatých závorkách *bez* udání jednotek. Ve výše uvedené definici příkazu znamená x horizontální a y vertikální rozměr plochy „obrázku“.

V takto definované ploše „obrázku“ umisťujeme jednotlivé objekty na pozici x, y příkazem

`\put(`x,y`){`*objekt*`}`.

Další příkazy si ukážeme pouze na příkladu. Podrobnější popis je uveden v *manuálu LaTeXu* [1], publikaci [4] aj.




```
\setlength{\unitlength}{1mm}
\begin{picture}(30,20)
\put(10,10){\framebox(15,10)
  [cc]{$\sqrt[3]{a^{2}}$}}
\put(30,15){\line(-1,1){5}}
\put(30,15){\line(-1,-1){5}}
\put(26.5,16){\circle{1.25}}
\put(10,10)
  {\vector(-1,-3){1.5}}
\put(10,10)
  {\vector(1,-3){1.5}}
\put(25,10)
  {\vector(-1,-3){1.5}}
\put(25,10)
  {\vector(1,-3){1.5}}
\end{picture}
```

3 Sázení matematických vzorců

3.1 Úvod

Matematické části textu v odstavci se uzavírají mezi `\(` a `\)` nebo mezi `$` a `$` nebo mezi `\begin{math}` a `\end{math}`. Za „matematický" text se považují jak kompletní matematické vzorce, tak i jednotlivé proměnné vztahující se ke vzorcům, řecká písmena, indexy nahoře i dole a různé zvláštní znaky.

Nechť a a b jsou odvěsny a c přepona, pak platí $c^2 = a^2 + b^2$ (Pythagorova věta).

```
Nechť $a$ a $b$ jsou
odvěsny a $c$ přepona,
pak platí $c^{2}=a^{2}+b^{2}$
(Pythagorova věta).
```

TEX se vyslovuje jako $\tau\epsilon\chi$.

100 m^2 obytné plochy

Se $\heartsuit$-ným pozdravem

```
\TeX\ se vyslovuje jako
 $\tau\epsilon\chi$.\\
100~m$^{2}$ obytné plochy\\
Se $\heartsuit$-ným
 pozdravem
```

Rozsáhlejší matematické vzorce nebo rovnice se sázejí lépe na zvláštní řádky. V tomto případě se píší mezi `\[` a `\]` nebo mezi `\begin{displaymath}` a `\end{displaymath}` — rovnice zde ovšem nejsou číslovány. Pokud uvedeme rovnice mezi `\begin{equation}` a `\end{equation}`, bude vysázený text obsahovat i čísla rovnic.

Nechť a a b jsou odvěsny a c přepona, pak platí

$$c^2 = a^2 + b^2 \tag{1}$$

(Pythagorova věta).

```
Nechť $a$ a $b$ jsou
odvěsny a $c$ přepona,
pak platí
\begin{equation}
c^{2}=a^{2}+b^{2}
\end{equation}
(Pythagorova věta).
```

Pomocí `\label` a `\ref` se lze odkazovat v textu na čísla rovnic.

$$\epsilon > 0 \tag{2}$$

Z rovnice (2) plyne …

```
\begin{equation} \label{eps}
\epsilon > 0
\end{equation}

Z~rovnice (\ref{eps})
plyne \dots
```

Sazba v matematickém režimu se liší od sazby v textovém režimu zvláště v těchto bodech:

1. Mezery a ukončení řádku ve vstupním textu nemají žádný význam, neboť vše se sází podle pravidel matematické sazby, případně se mezery zadávají speciálními příkazy jako `\,` nebo `\quad`.

$$\forall x \in {\rm R}: \qquad x^{2} \geq 0 \tag{3}$$

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

2. Prázdné řádky jsou zakázány (tj. matematický vzorec musí být v jednom odstavci).

3. Každé písmeno je považováno za jméno proměnné a je vysázeno odpovídajícím způsobem (tzv. matematická kurzíva). Chceme-li uvnitř matematického textu sázet normální text, musíme jej uzavřít do `\mbox{...}`.

$$x^{2} \geq 0\qquad \mbox{pro všechna } x \in {\rm R} \tag{4}$$

```
\begin{equation}
x^{2} \geq 0\qquad
\mbox{pro všechna }
x \in {\rm R}
\end{equation}
```

3.2 Objekty matematických vzorců

V této kapitole jsou stručně popsány nejdůležitější objekty, které se používají v matematických vzorcích. Seznam všech použitelných symbolů obsahuje kap. 3.5.

Malá **řecká písmena** se zadávají jako `\alpha`, `beta`, `\gamma` atd., velká řecká písmena jako `{\rm A}`, `{\rm B}`, `\Gamma`, `\Delta` atd.

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

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

Dále existuje mnoho **matematických symbolů**: od $\in$ přes $\Rightarrow$ až k ∞ (viz kap. 3.5).

Exponenty a indexy se zadávají pomocí znaků `^` a `_`.

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

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

Značka pro **odmocninu** se zadává pomocí `\sqrt`, n-tá odmocnina pomocí `\sqrt[`n`]`. Velikost odmocnítka volí LaTeX automaticky.

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

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

Příkaz `\overline` resp. `\underline` zajistí **nadtržení** resp. **podtržení** požadované části matematického vzorce.

$\overline{m+n}$

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

Příkaz `\overbrace` resp. `\underbrace` vysází vodorovnou **složenou závorku** nad resp. pod požadovanou částí matematického vzorce.

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

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

Příkazy pro sázení matematických **„akcentů“** jako šipka (tj. označení vektoru) nebo vlnovka nad proměnnou obsahuje tabulka 4. Delší vlnovky a stříšky, které mohou být nad více (max. 3) znaky dostaneme pomocí `\widetilde` a `\widehat`. Označení derivace se zadává pomocí `'` (apostrof).

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

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

Matematické **funkce** se v literatuře sází obvykle nikoliv kurzívou (jako jména proměnných), nýbrž „normálním“ typem písma. K tomu slouží následující příkazy:

`\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`

Pro funkci „modulo“ existují dva různé příkazy: `\bmod` pro binární operátor $a \bmod b$ a `\pmod{...}` pro zadání ve formě $x \equiv a \pmod{b}$. V naší technické literatuře se však jinak označují funkce „tangens“ apod. Tyto lze v LaTeXu v případě potřeby snadno dodefinovat příkazy typu

```
\newcommand{\tg}{\mathop{\rm tg}\nolimits}
```

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

```
\begin{displaymath}
\lim_{x \to 0}
\frac{\sin x}{x}=1
\end{displaymath}
```

Zlomek (angl. fraction) se sází příkazem `\frac{...}{...}`. Pro jednoduché zlomky lze však také použít operátor `/`.

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

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

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

Binomické koeficienty lze zadávat ve formě `{... \choose ...}`. Příkazem `\atop` dostaneme totéž bez závorek.

$${ n \choose k } \qquad { x\atop y+2 }$$

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

Značka **integrálu** se sází příkazem `\int`, značka **sumy** příkazem `\sum`. Horní a dolní mez se zadává pomocí `^` a `_` (jakoby horní a dolní index).

Meze integrálu se normálně sázejí *vedle* značky integrálu (aby se šetřilo místo). Pokud požadujeme meze nad a pod značkou, použijeme příkaz `\limits`.

Naproti tomu meze u sumy jsou sázeny nad a pod značku, pouze u vzorců v textu nebo použitím příkazu `\nolimits` se meze vysázejí vedle značky.

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

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

Pro **závorky** a jiné ohraničující elementy obsahuje TEX mnoho různých symbolů (např. $[\ \langle\ \|\ \updownarrow$). Kulaté a hranaté závorky se zadávají odpovídajícími znaky, složené závorky pomocí `\{` a `\}`, ostatní speciálními příkazy (např. `\updownarrow`).

Uvedeme-li před otevírací závorkou příkaz `\left` a před odpovídající zavírací závorkou příkaz `\right`, LATEX automaticky zvolí velikost závorek. Zde je nutno upozornit, že každému příkazu `\left` musí odpovídat `\right`, i když bychom nechtěli uzavírací závorku sázet. Pak se používá příkaz `\right` s tečkou (tj. „`\right.`“).

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

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

V některých případech je vhodné velikost závorek nastavit „ručně". K tomu slouží příkazy `\bigl`, `\Bigl`, `\biggl` a `\Biggl` místo `\left` a podobně `\bigr` atd. místo `\right`.

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

```
\begin{displaymath}
\Bigl( (x+1) (x-1)
\Bigr) ^{2}
\end{displaymath}
```

Tři tečky se ve vzorcích zadávají příkazy `\ldots` a `\cdots`. Příkaz `\ldots` sází tečky na základní čáru (low), `\cdots` sází tečky doprostřed výšky znaků (centered). Kromě toho existuje příkaz `\vdots` pro vertikální a `\ddots` pro diagonální tečky.

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

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

Příkaz `\stackrel` se dvěma argumenty vysází jeden symbol nad druhý. První argument je sázen menší velikostí.

$$\vec{x} \stackrel{\rm def}{=} (x_{1},\ldots,x_{n})$$

```
\begin{displaymath}
\vec{x} \stackrel{\rm def}{=}
(x_{1},\ldots,x_{n})
\end{displaymath}
```

V matematickém režimu lze také příkazem `\cal` přepnout implicitní matematickou kurzívu na psané písmo (skript). Můžeme pak ovšem použít pouze velká písmena.

$$z \in Z \quad z \in {\cal Z}$$

```
\begin{displaymath}
z \in Z \quad
z \in {\cal Z}
\end{displaymath}
```

3.3 Sazba vzorců vedle sebe

Nejsme-li spokojeni ve vzorcích s **mezerami**, které zvolil TEX, můžeme je změnit explicitními příkazy. Mezi nejdůležitější patří `\,` pro malou mezeru, `\ ` pro střední, `\quad` a `\qquad` pro velké mezery. Příkaz `\!` mezeru zmenšuje (záporná malá mezera).

$$F_n = F_{n-1} + F_{n-2} \qquad n \ge 2$$

```
\begin{displaymath}
F_{n} = F_{n-1} + F_{n-2}
 \qquad n \ge 2
\end{displaymath}
```

$$\int\!\!\!\int_{D} dx\,dy \quad \mbox{místo} \quad \int\int_{D} dx dy$$

```
\begin{displaymath}
\int\!\!\!\int_{D} dx\,dy
\quad \mbox{místo} \quad
\int\int_{D} dx dy
\end{displaymath}
```

3.4 Sazba na více řádků

Pro **matice** aj. se používá prostředí `array`, které se chová podobně jako prostředí `tabular`. Příkazem `\\` se dělí řádky.

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

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

Víceřádkové vzorce nebo systémy rovnic se sázejí pomocí prostředí `eqnarray` a `eqnarray*` místo `equation`. Při použití `eqnarray` obsahuje každý řádek číslo rovnice (pokud to u některého řádku nezakážeme), při `eqnarray*` nikoli (jako v `displaymath`). Systémy rovnic, které jsou číslovány jako jeden objekt, se mohou zadávat v prostředí `array` uvnitř prostředí `equation`.

$$\delta_{ij} = \left\{ \begin{array}{ll} y & \mbox{je-li}\ y>0, \\ z+y & \mbox{jinak.} \end{array} \right. \qquad (5)$$

```
\begin{equation}
\delta_{ij} = \left\{
  \begin{array}{ll}
    y & \mbox{je-li}\ y>0,\\
    z+y  & \mbox{jinak.}
  \end{array}
\right.
\end{equation}
```

Prostředí `eqnarray` a `eqnarray*` se chovají jako třísloupcové tabulky ve formě `{rcl}`, přičemž prostřední sloupec se používá pro znaménka rovnosti nebo nerovnosti, podle kterých mají být řádky vyrovnány. Příkaz `\\` dělí řádky.

$$\begin{aligned} f(x) &= \cos x & (6)\\ f'(x) &= -\sin x & (7)\\ \int_0^x f(y)dy &= \sin x & (8)\end{aligned}$$

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

LaTeX *nedělí* automaticky příliš dlouhé rovnice do několika řádků . Autor musí určit, na které pozici se má dělit řádek a jak široká má být zarážka. Nejčastěji se používá některá z následujících variant.

$$\begin{aligned} \sin x &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\ &\quad -\frac{x^7}{7!} + \cdots & (9)\end{aligned}$$

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

$$\begin{aligned} &\cos x = 1 - \frac{x^2}{2!} + \\ &\qquad + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots & (10)\end{aligned}$$

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

Příkaz `\nonumber` zajistí, že pro daný řádek se nesází číslo rovnice. Příkaz `\lefteqn` umožňuje výjimku z vertikálního členění uvnitř `eqnarray`. Přesnější informace obsahuje *manuál LaTeXu* [1].

V našich publikacích nebývají samostatné vzorce centrovány, nýbrž zarovnány nalevo a odsazeny. Tento způsob sazby matematických textů můžeme nastavit uvedením `fleqn` mezi volby příkazu `\documentstyle`. Odsazení např. 1 cm můžeme nastavit příkazem

```
\setlength{\mathindent}{1cm}
```

Číslování rovnic vlevo místo vpravo můžeme zajistit uvedením `leqno` mezi volby příkazu `\documentstyle`.

3.5 Seznam matematických symbolů

V následujících tabulkách jsou uvedeny všechny symboly, které lze standardně použít v matematickém režimu.

Tabulka 4: Matematické akcenty

`\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`	$\ddot{a}$	`\breve a`	$\breve{a}$
`\bar a`	$\bar{a}$	`\vec a`	$\vec{a}$

Tabulka 5: Malá řecká písmena

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

Tabulka 6: Velká řecká písmena

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

Tabulka 7: Různé jiné symboly

$\aleph$	`\aleph`	$\prime$	`\prime`	$\forall$	`\forall`
$\hbar$	`\hbar`	$\emptyset$	`\emptyset`	$\exists$	`\exists`
$\imath$	`\imath`	∇	`\nabla`	$\neg$	`\neg`
$\jmath$	`\jmath`	$\surd$	`\surd`	$\flat$	`\flat`
ℓ	`\ell`	$\top$	`\top`	$\natural$	`\natural`
$\wp$	`\wp`	$\bot$	`\bot`	$\sharp$	`\sharp`
$\Re$	`\Re`	$\|$	`|`	$\clubsuit$	`\clubsuit`
$\Im$	`\Im`	$\angle$	`\angle`	$\diamondsuit$	`\diamondsuit`
∂	`\partial`	$\triangle$	`\triangle`	$\heartsuit$	`\heartsuit`
∞	`\infty`	$\backslash$	`\backslash`	$\spadesuit$	`\spadesuit`
$\mho$	`\mho`	$\Box$	`\Box`	$\Diamond$	`\Diamond`

Tabulka 8: Velké operátory

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

Tabulka 9: Binární operátory

$+$	`+`	$-$	`-`		
$\pm$	`\pm`	$\cap$	`\cap`	$\vee$	`\vee`
$\mp$	`\mp`	$\cup$	`\cup`	$\wedge$	`\wedge`
$\setminus$	`\setminus`	$\uplus$	`\uplus`	$\oplus$	`\oplus`
$\cdot$	`\cdot`	$\sqcap$	`\sqcap`	$\ominus$	`\ominus`
$\times$	`\times`	$\sqcup$	`\sqcup`	$\otimes$	`\otimes`
$\ast$	`\ast`	$\triangleleft$	`\triangleleft`	$\oslash$	`\oslash`
$\star$	`\star`	$\triangleright$	`\triangleright`	$\odot$	`\odot`
$\diamond$	`\diamond`	$\wr$	`\wr`	$\dagger$	`\dagger`
$\circ$	`\circ`	$\bigcirc$	`\bigcirc`	$\ddagger$	`\ddagger`
$\bullet$	`\bullet`	$\bigtriangleup$	`\bigtriangleup`	$\amalg$	`\amalg`
$\div$	`\div`	$\bigtriangledown$	`\bigtriangledown`		

Tabulka 10: Relace

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq`	$\geq$	`\geq`	$\equiv$	`\equiv`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\ll$	`\ll`	$\gg$	`\gg`	$\asymp$	`\asymp`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni`	$\Join$	`\Join`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\smile$	`\smile`	$\mid$	`\mid`	$\doteq$	`\doteq`
$\frown$	`\frown`	$\parallel$	`\parallel`	$\perp$	`\perp`
				$\propto$	`\propto`

Tabulka 11: Negace

$\not<$	`\not<`	$\not>$	`\not>`	$\not=$	`\not=`
$\not\leq$	`\not\leq`	$\not\geq$	`\not\geq`	$\not\equiv$	`\not\equiv`
$\not\prec$	`\not\prec`	$\not\succ$	`\not\succ`	$\not\sim$	`\not\sim`
$\not\preceq$	`\not\preceq`	$\not\succeq$	`\not\succeq`	$\not\simeq$	`\not\simeq`
$\not\subset$	`\not\subset`	$\not\supset$	`\not\supset`	$\not\approx$	`\not\approx`
$\not\subseteq$	`\not\subseteq`	$\not\supseteq$	`\not\supseteq`	$\not\cong$	`\not\cong`
$\not\sqsubseteq$	`\not\sqsubseteq`	$\not\sqsupseteq$	`\not\sqsupseteq`	$\not\asymp$	`\not\asymp`

Tabulka 12: Šipky

$\leftarrow$	`\leftarrow`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\rightarrow$	`\rightarrow`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleft...`	$\updownarrow$	`\updownarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleft...`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`			$\leadsto$	`\leadsto`

Tabulka 13: Levé závorky

$($	`(`	$[$	`[`	$\{$	`\{`
$[$	`\lbrack`	$\lfloor$	`\lfloor`	$\lceil$	`\lceil`
$\{$	`\lbrace`	$\langle$	`\langle`		

Tabulka 14: Pravé závorky

$)$	`)`	$]$	`]`	$\}$	`\}`
$]$	`\rbrack`	$\rfloor$	`\rfloor`	$\rceil$	`\rceil`
$\}$	`\rbrace`	$\rangle$	`\rangle`		

Tabulka 15: Synonyma

Pro některé symboly lze použít více různých příkazů:

$\neq$	`\ne` or `\neq`	`\not=`
$\leq$	`\le`	`\leq`
$\geq$	`\ge`	`\geq`
$\{$	`\{`	`\lbrace`
$\}$	`\}`	`\rbrace`
$\to$	`\to`	`\rightarrow`
$\leftarrow$	`\gets`	`\leftarrow`
$\ni$	`\owns`	`\ni`
$\wedge$	`\land`	`\wedge`
$\vee$	`\lor`	`\vee`
$\neg$	`\lnot`	`\neg`
$\vert$	`\vert`	`|`
$\Vert$	`\Vert`	`\\|`

Tabulka 16: Nematematické symboly

Následující symboly se používají v textovém režimu:

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

4 Speciality

Všechny speciality, které LaTeX nabízí, jsou popsány v *manuálu LaTeXu* [1]. Zde jsou uvedeny jen některé „bonbónky“.

4.1 Druhy a velikosti písma

Normálně LaTeX volí vhodný druh a velikost písma na základě příkazů, které udávají logickou strukturu textu (nadpisy, zdůraznění atd.). Ve zvláštních případech lze změnit druh nebo velikost písma pomocí příkazů uvedených v tab. 17 a 18. Tyto příkazy se používají uvnitř skupin podobně jako příkaz `\em`.

Malí **tuční** Římané ovládli celou velkou *Itálii*.

le 2$^{\text{ème}}$ régime

```
{\small Malí {\bf tuční}
Římané ovládli }{\large
celou velkou {\it Itálii\/}.}
\\
le $2^{\mbox{\scriptsize
\`eme}}$r\'egime
```

Příkazy pro změnu velikosti písma z tabulky 18 přepnou typ písma vždy na `\rm`[5]. Například velké tučné písmo zvolíme pomocí příkazů `\Large\bf`, *nikoli* pomocí `\bf\Large`.

Příkazy pro změnu velikosti písma také změní vzdálenosti mezi řádky na vhodnou hodnotu — ale jen tehdy, leží-li prázdný řádek, který ukončuje odstavec, uvnitř oboru platnosti příkazu pro změnu velikosti.

Základní pravidlo pro volbu typů písma je *střídmost a jednotnost*. Čím *méně* různých typů písma použijeme, tím čitelnější a hezčí bude výsledný dokument.

[5]Neplatí pro NFSS — viz str. 48

Tabulka 17: Druhy písma

`\rm`	normální písmo — antikva (roman)
`\bf`	**polotučné písmo (boldface)**
`\it`	*kurzíva (italic)*
`\sl`	*skloněné písmo (slanted)*
`\sf`	bezserifové písmo (sans serif)
`\sc`	PÍSMO „CAPS AND SMALL CAPS“
`\tt`	`strojopisné písmo (typewriter)`
`\boldmath`	polotučný tisk v matematických vzorcích

Tabulka 18: Velikosti písma

`\tiny`	nejmenší písmo
`\scriptsize`	velmi malé písmo (jako indexy)
`\footnotesize`	malé písmo (jako poznámky)
`\small`	menší písmo
`\normalsize`	normální písmo
`\large`	větší písmo
`\Large`	velké písmo
`\LARGE`	velmi velké písmo
`\huge`	obrovské písmo
`\Huge`	větší než obrovské

4.2 Mezery

4.2.1 Proklad mezi řádky

Pokud je v dokumentu nutné použít větší mezery mezi řádky (proklad), než jsou určeny v dokumentním stylu, musíme změnit příkaz `\baselinestretch` na hodnotu větší než 1.

pro řádkování „jeden a půl“:
`\renewcommand{\baselinestretch}{1.3}`

pro řádkování „dva“:
`\renewcommand{\baselinestretch}{1.6}`

4.2.2 Speciální horizontální mezery

Mezery mezi slovy a větami sází LaTeX automaticky. Ostatní horizontální mezery lze zadat příkazem

`\hspace{`*délka*`}`.

Pokud má mezera zůstat i na začátku nebo konci řádku, musíme použít `\hspace*` místo `\hspace`. Délka v nejjednodušším případě zadává jako číslo a jednotka. Nejdůležitější jednotky jsou uvedeny v tab. 19.

Zde   je mezera 1.5 cm.

```
Zde\hspace{1.5cm}je mezera
1.5~cm.
```

Příkazy v tab. 20 vytvářejí speciální horizontální mezery. Příkaz `\hfill` vytváří mezeru, která se „roztáhne“ co možná nejvíce. Příklady použití příkazů `\,` a `\hfill`:

Tabulka 19: Jednotky pro zadávání délek

`mm`	milimetr
`cm`	centimetr = 10 mm
`in`	palec (inch) $\approx$ 25 mm
`pt`	bod (point) $\approx \frac{1}{72}$ in $\approx \frac{1}{3}$ mm
`em`	šířka písmena M aktuálního typu písma
`ex`	výška písmena x aktuálního typu písma

Tabulka 20: Horizontální mezery

`\,`	malá (zúžená) mezera
`\enspace`	mezera široká jako číslice
`\quad`	mezera široká jako je vysoké písmeno (čtverčík)
`\qquad`	mezera dvakrát tak široká jako `\quad`
`\hfill`	mezera, která se může natáhnout od 0 do ∞.

„ ‚Parsifal‘ zní déle než ‚Cats‘.“

```
\uv{\,‚Parsifal‘ zní déle
než ‚Cats‘.}
```

vlevo vpravo

```
vlevo\hfill vpravo
```

4.2.3 Speciální vertikální mezery

Mezery mezi odstavci, kapitolami atd. určuje LaTeX sám. Ve zvláštních případech lze zvětšit mezeru *mezi dvěma odstavci* příkazem

`\vspace{`*délka*`}`.

Tento příkaz bychom měli zadávat vždy mezi dvěma prázdnými řádky. Pokud má mezera zůstat také na začátku nebo na konci stránky, musíme použít příkaz `\vspace*` místo `\vspace`. Příkazy v tab. 21 vytvářejí speciální vertikální mezery. Příkaz `\vfill` ve spojení s `\pagebreak` slouží k vysázení textu na dolní okraj stránky nebo k vertikálnímu centrování textu.

Mezeru mezi řádky *uvnitř* odstavce můžeme zvětšit příkazem

`\\[`*délka*`]`.

Tabulka 21: Vertikální mezery

`\smallskip`	přibližně 1/4 řádku
`\medskip`	přibližně 1/2 řádku
`\bigskip`	přibližně 1 řádek
`\vfill`	mezera, která se může natáhnout od 0 do ∞.

4.3 Přizpůsobení formátu papíru

Papír formátu A4, který se u nás obvykle používá, je vyšší a o něco užší než papír obvykle užívaný v Americe, který podporují standardní dokumentní styly LaTeXu. Pomocí příkazů

```
\addtolength{\topmargin}{-...}
\addtolength{\textheight}{...}
```

lze zmenšit horní a dolní okraj, a tím zvětšit počet řádků na stránce. Délku `\textheight` bychom vždy měli měnit o celočíselné násobky délky `\baselineskip`, která závisí na velikosti písma. Doporučené hodnoty jsou:

Velikost písma	10pt	11pt	12pt
`baselineskip`	12pt	13.6pt	15pt
Δ`topmargin`	-30pt	-34pt	-30pt
Δ`textheight`	60pt	68pt	60pt

Chybí-li záhlaví nahoře či dole, můžeme změnit také `\topmargin` o větší či menší hodnotu. Například tento dokument obsahuje příkazy

```
\addtolength{\topmargin}{-27pt}
\addtolength{\textheight}{68pt}
```

Obecně však *nemusíme* takové změny zadávat přímo, nýbrž můžeme použít zvláštní volby příkazu`\documentstyle`, pro sázení na různé formáty papíru (viz [3]).

Délku řádku a pravý a levý okraj bychom neměli měnit přímo, neboť bychom mimoto rovněž museli změnit mnoho příkazů LaTeXu, které na nich závisejí. Informace o takových změnách formátu mohou pokročilí uživatelé nalézt v [9].

4.4 Dopisy

Pokud zadáme styl dokumentu `letter` (nebo jeho variantu — viz [3]), můžeme mezi `\begin{document}` a `\end{document}` napsat jeden nebo více dopisů.

Pomocí `\signature` a `\address` zadáváme jméno a adresu odesílatele. Dopis začínáme příkazem `\begin{letter}{...}`. Jméno a adresu příjemce uvádíme jako parametr. Oslovení a závěrečný pozdrav zadáme pomocí `\opening{...}` a `\closing{...}`. Pod závěrečný pozdrav se automaticky doplní podpis definovaný v `\signature`. Dopis ukončíme `\end{letter}`.

Pokud uvedeme před `\begin{document}` příkaz `\makelabels`, vytisknou se kromě toho také štítky s adresami.

Příklad dopisu je na obr. 8.

```
\documentstyle[12pt,czech]{letter}
\address{ČVUT v~Praze, fak. strojní, K\,235 \\
         Horská 3 \\
         128\,00 Praha 2}
\signature{Pavel Sýkora}
\begin{document}
\begin{letter}{Pan Jiří Zlatuška \\
      Masarykova universita \\
      Burešova 20\\ 601\,77 Brno}
\opening{Vážený pane kolego!}
Zasílám Vám .....

.....
\closing{S pozdravem}
\end{letter}
\end{document}
```

Obrázek 8: Příklad dopisu

4.5 Seznam literatury

Pomocí prostředí `thebibliography` můžeme vysázet seznam literatury. Každá jeho položka začíná `\bibitem`. Jako parametr se uvádí jméno, které používáme jako symbolický odkaz v příkazu `\cite`. Za parametrem následují vlastní údaje o použitém pramenu. Číslování probíhá automaticky. Parametr u `\begin{thebibliography}` udává maximální šířku tohoto čísla, tedy např. `{99}` pro maximálně dvouciferná čísla.

Pro citaci literatury v textu se používá obvykle příkaz `\cite`. Příklad:

Partl [1] navrhoval, že ...

Reference

[1] H. Partl: *German TEX,* TUGboat Vol. 9, No. 1 (1988)

```
Partl~\cite{pa1}
navrhoval, že ...

\begin{thebibliography}{99}
\bibitem{pa1}
H.~Partl: {\it German \TeX,}
TUGboat Vol.~9, No.~1 (1988)
\end{thebibliography}
```

4.6 Robustní a křehké příkazy

Většina příkazů LATEXu je „robustních“, t.j. jejich použitím dostaneme vždy žádaný výsledek.

Kromě nich jsou i tzv. „křehké“ příkazy, které v jistých situacích (v tzv. „plovoucích“ parametrech) pracují správně pouze tehdy, uvedeme-li před takový příkaz `\protect`. Ke křehkým příkazům patří kromě jiných příkazy pro změnu velikosti písma (viz tab. 18) a příkaz `\footnote`. Je také několik málo zřídka se vyskytujících případů, kdy musíme psát např. `\protect\small` místo `\small`. V *manuálu LATEXu* [1] je uvedeno, kdy se tyto případy vyskytují.

DODATKY

A Chyby

To, že naše požadavky na vzhled dokumentu zadáváme pomocí příkazů ve vstupním souboru, přináší jednu nevýhodu. Můžeme příkaz zadat nesprávně — vznikne syntaktická chyba. Je to cena za otevřenost a rozsáhlé možnosti systému LaTeX. Při zpracování vstupního souboru pak LaTeX tuto chybu nahlásí. Chybová hlášení LaTeXu jsou dosti rozsáhlá a nezkušenému uživateli se mohou zdát poněkud záhadná. Důležité je si uvědomit, že mnoho vypsaných informací je možné ignorovat. Ze začátku bychom se měli naučit rozpoznat na kterém řádku vstupního souboru se chyba vyskytla a co asi chybu způsobilo.

Nejčastější chybou je nesprávné zadání názvu prostředí nebo příkazu. Řekněme, že na sedmém řádku vstupního souboru napíšeme `\begin{quotr}` místo správného `\begin{quote}`. LaTeXnám vypíše následující hlášení:

```
LaTeX error.  See LaTeX manual for explanation.
              Type  H <return>  for immediate help.
! Environment quotr undefined.
\@latexerr ...diate help.}\errmessage {#1}

l.7 \begin{quotr}

?
```

Popis chyby následuje za vykřičníkem (environment quotr undefined = není definováno prostředí quotr). Důležitý je i řádek začínající `l.7 ...`, který říká, že k chybě došlo na sedmém řádku a vypisuje vstupní řádek až k místu, kde se chyba vyskytla. Pokud se pod tímto hlášením objeví otazník, je LaTeX v interaktivním módu a čeká na náš vstup. Nejjednodušší možností je zadání písmene `x` následovaného klávesou Enter. LaTeX tím okamžitě skončí zpracování vstupního souboru.

LaTeX ovšem může být i v režimu, kdy na vstup uživatele nečeká a pokračuje ve zpracování vstupního souboru. Do interaktivního režimu jej pak můžeme dostat stisknutím kombinace kláves Ctrl-Break, nebo jej můžeme nechat zpracovávat soubor a průběžně sledovat další chybová hlášení. První chyba může ovšem způsobit další chyby často s komplikovanými hlášeními. Celý protokol zpracování se také nahrává do souboru s příponou `.log`.

Nejčastější chyby:

- Nesprávně uvedený název příkazu nebo prostředí.
- Nespárované závorky `{` a `}` nebo příkazy `\begin` a `\end`.
- Pokus o použití některého speciálního znaku z kap. 1.3.2 jako tištitelného znaku. (Pozn.: Některé z těchto znaků zapínají s varovným hlášením `! Missing $ inserted` matematický režim. Na výstupu pak je část textu matematickou kurzívou bez mezislovních mezer.)
- Chybí parametr příkazu.
- Chybí prostředí `document`.
- Chybí příkaz `\documentstyle`. (Pozn.: Tato chyba se hlásí jako `! Undefined control sequence`, *nehlásí* se však zde `LaTeX error`!)

Někdy LaTeX vypíše hvězdičku (`*`) a zastaví se bez chybového hlášení. To se stává, zapomeneme-li příkaz `\end{document}` na konci souboru, ale mohou jej vyvolat i jiné chyby. V tomto případě napíšeme `\stop` a stiskneme klávesu Enter.

B A co dál?

Nyní umíte LaTeX natolik, abyste mohli vytvářet širokou škálu dokumentů. Další informace naleznete v *manuálu LaTeXu* [1] nebo ve výborných publikacích [4, 5]. Náročným uživatelům se po čase stane, že narazí na jistá omezení LaTeXu. TeX však používají tisíce uživatelů na celém světě (zvláště v akademickém prostředí) a mnoho věcí je již hotovo. V mezinárodních počítačových sítích jsou uloženy stovky megabytů programů, maker a informací o TeXu, LaTeXu a dalších podobných systémech. Chcete-li sledovat novinky z této oblasti a téměř zdarma získat kvalitní programy usnadňující práci s LaTeXem, můžete se stát členy Československého sdružení uživatelů TeXu. Stačí napsat na adresu

CSTUG MÚ UK
Sokolovská 83
186 00 Praha 8

Na závěr si uvedeme stručný přehled několika z mnoha hotových rozšíření LaTeXu:

NFSS — New Font Selection Scheme — nový způsob změny typů a velikostí písma v LaTeXu. Každý příkaz změny mění pouze jeden atribut písma, tedy například příkazy `\tt\sl\bf\large` přepnou na větší skloněné tučné strojopisné písmo. NFSS umožňuje použití i jiných než standardních písem (např. Postscriptové fonty).

$\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX — LaTeX se značně rozšířenými možnostmi sazby matematických textů. Mnoho nových symbolů a typů písma. Některé příkazy, které jsou ve standardním LaTeXu těžkopádné (sazba matic) jsou rozšířeny.

array — sada souborů z dílny F. Mittelbacha a D. Carlisleho rozšiřuje možnosti prostředí `tabular`. Umožňuje například definovat sloupce zarovnané na desetinnou tečku, používat dvojité linky, vytvářet tabulky přes více stránek aj.

lslide — volba příkazu `\dokumentstyle` pro pohodlné vytváření fólií pro zpětné projektory. Jednodušší práce než s balíkem SliTeX, ale jen jednobarevné fólie.

cprog — pro přehledné formátování v dokumentu zařazených výpisů programů v jazyku C/C++, Pascal a Modula-2.

answers — pro pohodlné sázení řešených příkladů.

fancyheadings — vytváření několikařádkových záhlaví s linkami na stránce nahoře i dole.

Reference

[1] L. Lamport: *LaTeX, A Document Preparation System, User's Guide and Reference Manual*, Addison-Wesley Publishing Company (1985), ISBN 0-201-15790-X.

[2] K. Pala: *Úvod do systému LaTeX,* České vysoké učení technické v Praze (1990), ISBN 80-01-00395-7.

[3] Ke každé instalaci by měl být (např. ve výpočetním středisku) k dostání manuál *LaTeX Local Guide*, ve kterém jsou uvedeny údaje specifické pro danou instalaci — např. příkazy potřebné pro spuštění programů a soubory, předdefinované formáty a fonty, které jsou k dispozici.

[4] H. Kopka: *LaTeX — Eine Einfürung,* Addison-Wesley Deutschland (1991), ISBN 3-89319-338-3.

[5] H. Kopka: *LaTeX — Erweiterungsmöglichkeiten,* Addison-Wesley Deutschland (1991), ISBN 3-89319-356-1.

[6] R. Wonneberger: *Kompaktführer LaTeX,* Addison-Wesley Deutschland (1987), ISBN 3-925118-46-2.

[7] L. Schuman: *Professioneller Buchsatz mit TeX,* R. Oldenboroug, Mnichov (1991).

[8] J. Warbrick: *Essential LaTeX,* (1992).

[9] H. Partl: *Layout-Änderungen mit LaTeX,* EDV-Zentrum der Technischen Universit"at Wien (1988). Zdrojové soubory pro LaTeX jsou k dispozici na serveru `listserv@dhdurz1.bitnet` (Heidelberg).

[10] D. E. Knuth: *The TeXbook,* Svazek A řady *Computers and Typesetting*, Addison-Wesley Publishing Company (1984), ISBN 0-201-13448-9.

[11] M. Doob: *Jemný úvod do TeXu,* Univerzita Karlova Praha (1990), CSTUG (1992).

[12] P. W. Abrahams: *TeXfor the Impatient,* Addison-Wesley (1990).

[13] N. Schwarz: *Einführung in TeX,* Addison-Wesley Deutschland (1987, 1991), ISBN 3-925118-25-X, ISBN 3-89319-345-6.

[14] W. Appelt: *TeX für Forgeschrittene,* Addison-Wesley Deutschland (1988), ISBN 3-89319-115-1.

[15] B. Roßboth: *TeXanisches ABC,* c't computer roč. 1990, č. 6, 7, 8.

[16] M. Lichá, O. Ulrych: *AMS-TeX verze 2.1,* CSTUG (1992).