

Příručka k sázecímu systému \LaTeX

Tomáš Zdráhal

22. června 2004

1 Úvod

Cílem této práce bylo napsat příručku k sázecímu systému \LaTeX . Snahou není vytvořit úplnou příručku, která by obsahovala vyčerpávající popis tohoto jinak propracovaného systému, ale spíše úvod do pravidel psaní v sázecím systému \LaTeX . Předpokládá se, že příručka bude sloužit jako pomůcka při psaní textů v systému \LaTeX , jako např.: diplomové práce nebo studijních textů. Motivací byla snaha o usnadnění tvorby různých forem vědeckých dokumentů.

Jednou z oblastí lidské činnosti, kterou výrazně poznamenalo rozšíření osobních počítačů, je zpracování textů. Postupně mizí ruční psaní delších textů a v poslední době se upouští i od psaní strojem a pole stále více ovládá výpočetní technika. Mezi první publikované programové prostředky pro sazbu textů a grafických prvků byl program \TeX a později jeho uživatelsky přístupnější nadstavba \LaTeX . Důvodem pro široké rozšíření a využívání obou programů je nejen dokonalost s jakou je výsledný text vysázen, ale i skutečnost, že jsou tyto systémy volně šiřitelné. V obou je také zabudován princip přizpůsobení národnímu prostředí. Zatímco vývoj \TeX u je ukončen, vývoj \LaTeX u nadále pokračuje. Kromě u nás nejrozšířenější verze \LaTeX 2.09 je k dispozici dočasná verze označovaná jako $\text{\LaTeX} 2_{\epsilon}$, která představuje okamžitý stav vývoje \LaTeX u. Novější verze by měla odstranit řadu slabých míst, především v málo rozvinuté grafice. Přičemž platí, že dokumenty pořízené ve starší verzi jsou zpracovatelné i ve verzi novější.

Obsah

1	Úvod	2
	Obsah	4
2	Základní principy	5
2.1	Pravidla sazby	7
2.2	Sazba českých textů	9
2.2.1	Balíčky	9
2.2.2	Dělení slov a diakritická znaménka	9
3	Řídící sekvence	10
3.1	Zpětné lomítko	10
3.2	Řídící slova	10
3.3	Řídící symboly	10
3.4	Změna typu a stupně písma	11
3.4.1	Typografické míry	11
3.4.2	Typ písma	11
3.4.3	Stupeň písma	12
4	Textová a jiná prostředí	13
4.1	Prostředí quote, quation a verse	13
4.1.1	Quote	13
4.1.2	Quotation	14
4.1.3	Verse	14
4.2	Prostředí verbatim	14
4.3	Prostředí center	15
4.4	Prostředí flushleft a flushright	15
4.5	Výčtová prostředí	15
4.5.1	Prostředí itemize	15
4.5.2	Prostředí enumerate	16
4.5.3	Prostředí description	16
5	Členění dokumentu	16
5.1	Kapitoly a jiné oddíly	16
5.2	Vytváření obsahu	17
5.3	Poznámka pod čarou	17
5.4	Okrajové poznámky	17
5.5	Číslování stránek	18
6	Tabulky	18
6.1	Prostředí tabbing	18
6.2	Vlastní tabulky	19
6.3	Prostředí table	20

7	Matematická sazba	21
7.1	Prostředí math	21
7.1.1	Dolní a horní indexy	22
7.2	Další matematická prostředí	22
7.3	Zápis matematických objektů	23
7.3.1	Zlomky	23
7.3.2	Odmocniny	23
7.3.3	Matematické značky	23
7.4	Balík amsmath	26
7.4.1	Prostředí amsmath	27
7.4.2	Matice	28
7.4.3	Integrál	28
8	Odkazy	28
8.1	Tvorba obsahů a odkazů	29
9	Obrázky	29
9.1	Balík color	30
9.2	Balík graphicx	30
	Literatura	32

2 Základní principy

Chceme-li pracovat s L^AT_EXem¹ musíme dodržet následující zásady:

1. Do vstupního textového souboru doplníme příkazy L^AT_EXu, které říkají, jak má text vypadat. Tyto příkazy určují „typ textu“, změny v typu písma aj.
2. Svůj vlastní soubor zpracujeme L^AT_EXem. Tím dostaneme výstup, který má podobu tzv. *.dvi souboru² (deviindependent). Tedy soubor nezávislý na zařízení.
3. Výstupní soubor *.dvi vytiskneme na tiskárně nebo na obrazovce.

Jestliže L^AT_EX ohlásil, že nerozumí našemu vstupu, nebo nevypadá-li výsledný text tak, jak jsme si představovali, musíme se vrátit a vstupní soubor opakováním uvedených kroků upravovat tak dlouho, dokud nebudeme s výsledkem spokojeni.

Vstupní soubor pro L^AT_EX je normální textový soubor, který si můžeme vytvořit pomocí „libovolného“ textového editoru. Jedinou podmínkou je, aby vzniklý textový soubor neobsahoval žádné řídicí znaky ani vnitřní příkazy (například Text602 nebo Word). Jak jsme již uvedli vstupní soubor obsahuje kromě textu i příkazy L^AT_EXu a jeho název může být jakýkoli s rozšířením *.tex .

Vstupní soubor by měl vypadat asi takto:

```
\documentclass[volby]{třída}[datum vytvoření]
      :
      : preamble
\begin{document}
      :
      : textová část
\end{document}
```

Povinný úvodní příkaz `documentclass` má parametr *třída*, který definuje styl sazby, jakým má být dokument vypracován. Máme tyto standardní třídy: `article` (pro článek), `report` (pro zprávu), `book` (pro knihu), `letter` (pro dopis). Potom je tu nepovinný parametr *datum*. Dále můžeme ještě zadat volitelný parametr, jehož pomocí lze modifikovat činnost příkazů ve zvolené třídě. Příklady některých tříd:

- `11pt`– sazba dokumentu bude provedena písmem o velikosti 11pt. Podobně lze zvolit volbu `10pt`, `12pt` atd.

¹čti „latechem“

²Výstupní soubor `dvi` čtou ovladače jednotlivých výstupních zařízení: ovladače obrazovky (prohlížeč obrazovky `dvi`) a ovladač tiskárny (tiskový program `dvi`). Soubor s tímto rozšířením vlastně „zviditelní“ práci L^AT_EXu v podobě hotové sazby.

- `twoside`– bude prováděno rozlišování levých a pravých stránek (liché stránky jsou tištěny na pravé straně).
- `a4paper`– nastavení formátu stránky na A4. Podobně existují volby `a5paper`, `b5paper`, `letterpaper`, `legalpaper`, `executivepaper`.

Jednoduchý text není nic jiného než posloupnost slov, vět a odstavců. Slova jsou od sebe oddělena jednou nebo více mezerami, tabulátory nebo koncem řádku (konec řádku je také znak, přestože není viditelný). Na počtu mezer oddělujících slova nezáleží. Jakýkoli počet mezer mezi slovy bude brán jako jedna mezera. L^AT_EX člení text na odstavce tam, kde ve vstupním textu vložíme prázdný řádek nebo vložíme příkaz `\par`. Ve všech ostatních případech má konec řádku přesně stejný význam jako mezera.

Ještě tu je deset *speciálních znaků*, které mají v L^AT_EXu svůj vlastní význam:

\$ % & { } ~ _ ^ \

Hlavní práci při realizaci publikace v systému L^AT_EX je zápis zdrojového textu společně s příkazy. Zápis těchto příkazů se skládá:

- buď z jednoho z prvních čtyř uvedených *speciálních znaků*,
- nebo z posloupnosti *zpětné lomítka* a příslušný znak (např. `\%`, `\#`),
- nebo z posloupnosti `\název`, kde název je posloupnost písmen, tato posloupnost představuje příkaz, např. `\tiny`, `\it`, `\beta`.

Systém rozlišuje mala a velká písmena. Proto `\Large` a `\large` mají odlišný význam.

Nyní si předvedeme jak by náš vstupní soubor `text.tex` mohl vypadat:

```
\documentclass[12pt]{article}
\begin{document}
  Toto je můj první text, skládající se ze dvou odstavců.
  Na dělení řádků si nedávám pozor.   Nový
  odstavec vyznačím prázdným řádkem.

  Druhý odstavec mi \LaTeX{ }odsadí sám.
  Velké mezery mu           mezi slovy nevadí.
\end{document}
```

Tento soubor nyní zpracujeme L^AT_EXem. Uvedeme si jeden z příkladů jak by to mohlo vypadat³. V MS-DOS spustíme příkaz⁴: `cslatex text.tex`

³Přesný postup vypadá různě na různých operačních systémech.

⁴ Aby tento příkaz pracoval, je třeba nastavit správné cesty k souborům

Poté L^AT_EX zpracovává náš vstupní soubor a na obrazovce vypisuje zprávy o tom, co dělá. Celé toto hlášení také současně ukládá do souboru `text.log`, který si můžeme v případě potřeby znovu přečíst. Pokud si zobrazíme další výsledný soubor `*.dvi` pomocí libovolného prohlížeče `dvi`, pak dostáváme výstup:

Toto je můj první text, skládající se ze dvou odstavců. Na dělení řádků si nedávám pozor. Nový odstavec vyznačím prázdným řádkem.

Druhý odstavec mi L^AT_EX odsadí sám. Velké mezery mu mezi slovy nevadí.

Všimněmi si, že o případné dělení slov na konci řádku se také nestaráme. V uvedené ukázce je použit příkaz `\LaTeX` jehož vysazením vznikne slovo L^AT_EX. Za tímto příkazem jsou v ukázce použité tyto závorky `{ }`, aby nebyla ignorována mezera uvnitř těchto závorek. Pokud bychom nepoužili těchto závorek dopadlo by to takto: L^AT_EXodsadím.

2.1 Pravidla sazby

V této kapitole si ukážeme v praxi velmi potřebná pravidla pro sazbu různých znaků, která dále uvedeme včetně jejich realizace.

Mezera - Několik mezer za sebou systém chápe jako jednu mezeru.

Dále se v sazbě používá zúžená mezera, která se vysází příkazem `\`, bez okolních mezer.

Nezlomitelná mezera je taková mezera, v níž nikdy nenastane řádkový zlom(konec řádku). Zapisuje se pomocí symbolu `~`. Používá se v případě, kdy chceme svázat dvě slova pevně k sobě. Příklad je na str. 13 v části 4.1.1

Pomlčky - Klávesnice osobního počítače má jen jeden druh pomlčky.

V textu se používá několik druhů pomlček, které jsou různě dlouhé. Nejkratší pomlčkou je *spojovník*, který se vyskytuje ve složených slovech. Spojovník sázíme tak, že použijeme právě jediné pomlčky. Delší je tzv. „půlčtverčiková pomlčka“, kterou užíváme při udávání číselného rozsahu, např. „str. 27–39“. Vysází se dvěma po sobě jdoucími spojovníky (`--`). Nejdelší pomlčka je tzv. „čtverčiková pomlčka“, kterou vysázíme tak, že napíšeme tři jednoduché spojovníky za sebou. Dále bychom měli rozlišovat znak minus, který má odlišný rozměr (viz tab. 1).

Uvozovky - Při sazbě uvozovek jsou rozlišovány počáteční uvozovky a koncové uvozovky. Protože však angličtina nepoužívá uvozovky

Význam	Zápis	Sazba
spojovník	-	-
půlčtverčík	--	—
čtverčík	---	---
záp. znaménko	\$-3\$	−3

Tabulka 1: Příklad použití

dole, toto je vyřešeno v balíku⁵ czech pomocí příkazu `\uv{slovo}`⁶. V anglicky psané literatuře se používají uvozovky, které lze vytvořit pomocí apostrofů (“ahoj”)

Procento - Všechn text, který se nachází ve vstupním souboru za znakem % bude ignorován. Tedy za tento znak lze psát poznámky ve vstupním souboru, které se nezobrazí ve výsledku. Chceme-li zapsat znak procento, musíme uvést zápis `\%`

Stupeň - Sazí se pomocí matematického symbolu umístěného v exponentu. V případě, že se jedná o jednotku je oddělen zúženou mezerou od hodnoty: `15\,$^\circ$C` (15 °C)

Dolar - Znak \$ je chápán jako uvození matematického prostředí. Je tedy nutné po výskytu tohoto znaku, aby někde následoval další, který toto prostředí uzavře⁷.

Hvězdička a mečík - Používají-li se jako znaménko pro narození a úmrtí, sázejí se před letopočet (datum) se zúženou mezerou. Např.: `*\,1942 \dag\,2001` \Rightarrow *1942 †2001

Tři tečky - Nahrazují nevýslovný nebo vynechaný text. Tečky mají mezi sebou zvláštní mezerování, které zařizuje příkaz `\dots`. Od slova se odděluje zúženou mezerou. Příklad: `Já vím, ale \, \dots` \Rightarrow Já vím, ale Dále se používají tečky, vedené středem řádku, diagonálně nebo svisle. Pro všechny uvedené případy jsou k dispozici příkazy: `\cdots`, `\vdots` a `\ddots`.

Další znaky - Na klávesnici jsou ještě některé další neabecední znaky, jimž musíme věnovat pozornost. Některé vysázíme tak, že je prostě stiskneme na klávesnici (@, *, +, =). Pak tu jsou znaky, které se

⁵Vysvětlení jak se používá balík bude na str. 9

⁶Kde *slovo* představuje řetězec, který chceme do uvozovek vložit („*slovo*“)

⁷viz. tabulka 1

v normálním textu běžně nevyskytují ($>$, $<$, $|$). Pokud přesto použijeme těchto znaků (s výjimkou `\tt` a matematického prostředí), objeví se místo nich znaky $\grave{}$, $\grave{}$, — .

2.2 Sazba českých textů

2.2.1 Balíčky

V této kapitole je popsáno zpracování českých textů. Nejdříve si objasníme jak používat *balík* (packages). Připojení balíku se provádí příkazem `\usepackage`, jehož sestava parametrů je:

`\usepackage[volby]{balík}[datum vytvoření]`

Tento příkaz se musí psát do *preamble*, kde se také nachází příkazy platné pro celý text. Balík je vlastně textový soubor s rozšířením **.sty*, který se nalézá někde na disku⁸. Příkazy uvedené v tomto balíku pak mohou být použity v celém dokumentu. Pro ilustraci uvedeme názvy některých standardních balíků, které lze nalézt v distribuci:

amstex - definuje mnoho dalších příkazů pro sazbu matematiky,

color - umožňuje práci v barvách. Psání textu barevně aj,

czech - umožňuje zpracovávat dokumenty s českými prvky (české kódování vstupního textu, české dělení slov na konci řádků, české názvy - Kapitola, Obrázek, Tabulka, názvy měsíců atd.),

graphics - umožňuje geometrické transformace textu jako zvětšování, rotace, zrcadlový obraz,

latexsym - definuje některé speciální symboly,

makeidx - definuje příkazy pro tvorbu rejstříku,

pict2e - definuje příkazy pro kreslení objektu bez omezení sklonů čar, tloušťek čar a velikost kružnic.

2.2.2 Dělení slov a diakritická znaménka

Při spuštění L^AT_EXu je určeno pomocí balíku **czech**, zda se bude dělit podle pravidel českého jazyka, nebo nikoli. Pokud neuvedeme balík **czech** bude dělení slov prováděno implicitním nastavením. Tedy bude docházet k dělení slov podle pravidel anglického jazyka.

V dokumentech zpracovávaných L^AT_EXem se zavedeným balíčkem **czech** lze používat rozšířenou sadu znaků a využívat tak psaní slov s diakritickými znaménky. Původní verze L^AT_EXu obsahuje jak příkazy pro umístění akcentů nad písmeny, tak i umožňuje použití speciálních písmen.

⁸K balíku musí být známá cesta, aby ho L^AT_EX při překladu našel

Jak už bylo uvedeno výše, balík czech taky umožňuje sazbu českých uvozovek a to příkazem `\uv{}`

3 Řídící sekvence

3.1 Zpětné lomítko

Znak (zpětnélomítko) je v L^AT_EXu velice důležitý. Řetězec uvedený za tímto zpětným lomítkem „ztrácí“ svůj obvyklý textový význam a nabývá speciálního významu pro L^AT_EX. Tedy bezprostředně za zpětným lomítkem následuje speciální posloupnost znaků, které L^AT_EXu říkají, co chceme, aby provedl. Takové posloupnosti jsou pro L^AT_EX příkazy a nazývají se *řídící sekvence* (control sequences). Se dvěma základními typy řídicích sekvencí jsme se už seznámili:

řídící slovo - je tvořeno zpětným lomítkem, za nímž následuje jeden nebo více abecedních znaků (písmen). První *neabecední znak* (číslo, interpunkční znaménko nebo mezera) ukončuje řídící slovo. Příklad řídicího slova: `\today`⁹

řídící symbol - je složen ze zpětného lomítka následovaným jedním *neabecedním znakem*. Příkladem řídicího symbolu je: `\%`

3.2 Řídící slova

Mezera, skupina mezer a konec řádku po libovolném řídícím slově je ignorována (resp. potlačena jejich sazba). Tato malá nepříjemnost se dá vyřešit tak, že se pro ukončení řídicího slova použije jiný neabecední symbol (jako např. složené závorky). Mezery nebo skupina mezer za tímto symbolem již budou vysázeny normálně. Např.: `\LaTeX{}` `p...` nebo `\LaTeX{ }` `p...`, jejichž výsledek bude stejný.

3.3 Řídící symboly

Některé řídící symboly jsme už uvedli (např.: `\%`), ale existují i další jako `\&`, `\$`, `\^`, `\#`, atd. Všechny uvedené řídící symboly slouží k vysázení znaků `%`, `&`, `$`, `^` a `#` (jinak totiž mají tyto symboly svůj vlastní význam). Pomocí další skupiny řídicích symbolů máme prostředky pro sazbu i v jiných jazycích. Za pozornost stojí možnost vkládat různé akcenty nad latinková písmena a vytvářet tak symboly z některých evropských jazyků.

⁹tato řídící sekvence vysází aktuální datum viz. dále

$\grave{a} \Rightarrow \backslash k\{a\}$	$\tilde{a} \Rightarrow \backslash \sim\{a\}$	$\check{a} \Rightarrow \backslash v\{a\}$	$\grave{a} \Rightarrow \backslash c\{a\}$
$\underline{a} \Rightarrow \backslash b\{a\}$	$\acute{a} \Rightarrow \backslash .\{a\}$	$\check{a} \Rightarrow \backslash H\{a\}$	$\grave{a} \Rightarrow \backslash d\{a\}$
$\hat{a} \Rightarrow \backslash ^\{a\}$	$\ddot{a} \Rightarrow \backslash "\{a\}$	$\mathring{a} \Rightarrow \backslash t\{aa\}$	$\grave{a} \Rightarrow \backslash u\{a\}$

Příkazy pro vytvoření akcentu jsou uvedeny v předchozí tabulce. Jako příklad je v této tabulce použito písmeno „a“, ale je možné použít jakékoliv jiné. Musíme si dát pozor na písmena „i“ a „j“. Při vložení akcentu nad ně je třeba použít jejich beztečkovou variantu. To se provede zápisem `\i` a `\j` (jinak by došlo k vysázení tohoto znaku *i*)

3.4 Změna typu a stupně písma

3.4.1 Typografické míry

Měrné jednotky	
pt – typografický bod, 1pt = 0,351mm	mm – milimetr
pc – pica = 12pt	dd – Didôtův bod, 1dd = 0,3759mm
cm – centimetr	cc – cicero = 12dd
in – palec (inch) = 25,4mm	sp – „scaled point“, 1pt = 65 536sp

Tabulka 2: Měrné jednotky

Základní jednotkou je *typografický bod*. Další vztahy mezi ostatními měrnými jednotkami jsou naznačeny v tabulce 2.

Kromě uvedených jednotek jsou v systému ještě k dispozici dvojice relativních jednotek, jejichž skutečná velikost je ovlivněna právě nastaveným písmem. Jedná se o jednotku `em` – je rovna stupni nastaveného písma a `ex` – rovna polovině nastaveného stupně písma. Existence těchto relativních jednotek umožňuje nastavovat některé rozměry, jež se budou vhodně měnit s použitým písmem.

3.4.2 Typ písma

Některé řídicí slova mění typ písma. Jsou to příkazy, které vytvářejí **polotučné** nebo *skloněné písmo* a další.

`\rm`– text bude vysázen normálním písmem (*roman*)

`\it`– výstup bude psán *kurzívou* (*italic*)

`\bf`– výstupní text bude psán **polotučně** (**bold face**)

`\sl`– přepíná výstup do *skloněného písma* (*slanted*)

`\sf`– přepíná výstup do bezpatkového písma (*sans serif*)

`\sc`– přepíná výstup do KAPITÁLEK (SMALL CAPS)

`\tt`– přepíná do strojopisného typu (typewriter type)

`\em`– umožňuje *zdůraznit písmo*¹⁰ (*emphasized*)

Uvedená řídící slova ovlivňují text libovolného rozsahu od místa svého uvedení. Pokud jsou ovšem použita uvnitř skupiny, bude jejich rozsah působnosti pouze v rámci skupiny. Druhým ekvivalentem těchto příkazů pro změnu typu písma jsou příkazy začínající `\text{??}{...}`, kde se místo dvou otazníků uvádí zkratky `rm`, `it`, `bf`, atd. Výjimku tvoří řídící slovo `\em`, jehož ekvivalentem je příkaz `\emph{...}`. Druhá varianta příkazů ovlivňující text má příslušný text ve svém parametru, čímž je přesně určeno pole jejich působnosti.

Pokud máme v úmyslu vysázet nějakou část *textu v kurzívě* a jinou část neskloněným písmem, je vhodné přidat mezi tento přechod řídící symbol `\/`.

3.4.3 Stupeň písma

Je třeba upozornit, že ne každý počítač bude mít k dispozici všechny typy písma a ve všech velikostech.¹¹ Stupeň písma lze nastavit pomocí řídících slov, uvedených v tab. 3. Kdykoli změníme stupeň písma,

Příkaz	Ukázka	Velikost
<code>\normalsize</code>	Vzorek	10 pt
<code>\large</code>	Vzorek	12 pt
<code>\Large</code>	Vzorek	14 pt
<code>\LARGE</code>	Vzorek	17 pt
<code>\huge</code>	Vzorek	20 pt
<code>\Huge</code>	Vzorek	25 pt
<code>\small</code>	Vzorek	9 pt
<code>\footnotesize</code>	Vzorek	8 pt
<code>\scriptsize</code>	Vzorek	7 pt
<code>\tiny</code>	Vzorek	5 pt

Tabulka 3: Stupně písma

systém automaticky přechází do písma normálního. Jestliže chceme napsat část textu v nějaké kombinaci typu a stupně písma, musíme příkaz

¹⁰ implicitně je na zdůraznění písma nastavena kurzíva

¹¹ Takové situace by se neměly stát pravidlem, ale spíše výjimkou. Snažíte-li se použít písma, které v dané instalaci není, L^AT_EX vypíše varovné hlášení.

pro změnu stupně před příkazem pro změnu typu písma. (`\Large\bf text`)

Ještě jedna poznámka ke změně typu či stupně písma. Chceme-li zdůraznit slovo „květiny“ ve větě, můžeme to udělat více způsoby.

Zítřa přines `\bf květiny\rm` a `\dots`

Zítřa přines `{\bf květiny}` a `\dots`

V druhém způsobu jsme použili tzv. *skupinu*, která začíná `{` a končí `}` těmito znaky. Libovolné změny uvnitř této skupiny, přestanou platit po jejím opuštění.

4 Textová a jiná prostředí

Pro označení zvláštních částí textu, které mají být vysázeny jinak než odstavec, slouží tzv. „prostředí“ (např. tabulka). Toto prostředí je uzavřeno příkazem

`\begin{název}`

tělo prostředí

`\end{název}`

Každé prostředí se chová jako skupina. Stejně jako skupina i prostředí se nesmí překrývat. Musíme tedy nejprve uzavřít „vnitřní“ prostředí a teprve pak „vnější“¹². Je možné vložit jedno prostředí do jiného.

4.1 Prostředí `quote`, `quation` a `verse`

Jednotlivá prostředí lze vkládat do jiných, avšak tato prostředí se nesmí překrývat (viz. výše)

4.1.1 Quote

Text vysázený v tomto prostředí bude užší zleva i zprava o jednu zarážku. Odstavce jsou nezarovnané a jsou odděleny užším prázdným řádkem.

Příklad:

`\begin{quote}`

Juroj Jánošík je historickou postavou, ale pověst o jeho životě vytvořili lidoví vypravěči.

Podle pověsti dali Janošíka jeho chudí rodiče studovat na kněze, aby se stal pánem. Jednou nešel s^otcem na robotu, protože`\,``\dots`

`\end{quote}`

¹²Není na škodu již při psaní vstupního souboru napsat `\begin` a `\end`. Potom teprve vkládat do tohoto prostředí vše potřebné.

Juroj Jánošík je historickou postavou, ale pověst o jeho životě vytvořili lidoví vypravěči.

Podle pověsti dali Janošika jeho chudí rodiče studovat na kněze, aby se stal pánem. Jednou nešel s otcem na robotu, protože...

4.1.2 Quotation

Prostředí `quotation` je téměř identické s prostředím `quote`. Rozdíl je jen v tom, že jednotlivé odstavce v prostředí `quotation` mají zarážky.

Příklad:

Juroj Jánošík je historickou postavou, ale pověst o jeho životě vytvořili lidoví vypravěči.

Podle pověsti dali Janošika jeho chudí rodiče studovat na kněze, aby se stal pánem. Jednou nešel s otcem na robotu, protože...

4.1.3 Verse

Prostředí se užívá pro sazbu veršů. Jednotlivé verše jsou zakončeny příkazem `\\`.

4.2 Prostředí verbatim

Pomocí prostředí `verbatim` je možné vysázet řídicí sekvence tak, jak jsou napsané ve vstupním souboru, aniž by byl brán zřetel na jejich význam. Příkazy jsou tedy zpracovány jako normální text. Veškerý výstup je sázen strojopisným písmem (typewriter). Také jsou zobrazeny všechny nové řádky a mezery, jak jsou napsané ve vstupním souboru. Jediný příkaz, který se nesmí objevit v prostředí `verbatim` je příkaz `\end{verbatim}`. Tento příkaz by totiž znamenal konec prostředí.

Je také možné použít řídicí slovo `\verb` nebo `\verb*`¹³. Podobně jako prostředí `verbatim` slouží tyto příkazy k potlačení formátování. U obou případů následuje bezprostředně za příkazem hraniční znak (jakýkoli *neabecední znak*). Potom následuje text, jehož formátování chceme potlačit a na konci je opět námi zvolený hraniční znak. Vše může vypadat následovně:

`\verb+Text` a za ním příkaz `\par +` \Rightarrow `Text` a za ním příkaz `\par` .

Prostředí `verbatim` ani příkaz `\verb` se nesmí objevit v parametru makra (např.: v poznámce pod čarou). Je důležité se také zmínit, že v těle příkazu nesmí být konec řádku.

¹³řídicí slovo `\verb*` nám dá na výstup místo mezer „vaničku“ (`_`).

4.3 Prostředí center

Jak už název tohoto prostředí napovídá, jeho funkcí je vycentrovat vložený text. Text nebude zarovnán a také se v něm nedělí slova.

`\begin{center}` Nadpis `\end{center}` \Rightarrow

Nadpis

4.4 Prostředí flushleft a flushright

Sazba na prapor používá stejné mezislovní mezery základní velikosti, ale je zarovnáno pouze na jedné straně. Praporek vlaje buď vpravo¹⁴ nebo praporek vlaje vlevo. Každý konec řádku musí být explicitně vyznačen, jinak bychom totiž dostali odstavec s nezarovnaným okrajem.

Příklad:

`\begin{flushright}`

Sazba na praporek, $\backslash\backslash$

který vlaje vlevo.

`\end{flushright}`

\Rightarrow

Sazba na praporek,

který vlaje vlevo.

4.5 Výčtová prostředí

L^AT_EX obsahuje prostředí, která umožňují snadno formátovat výčty a seznamy. Mezi nejdůležitější patří prostředí `itemize`, `enumerate` a `description`.

4.5.1 Prostředí itemize

Jednotlivé položky výčtu jsou standardně označeny „kuličkou“ (*bullet*). Před každou položkou výčtu je řídicí slovo `\item`. Všechny takto označené položky budou řazeny do odstavců, které budou stejně zarovnané.

Příklad:

`\begin{itemize}`

Prvky výčtu budou vyznačeny

symbolem \bullet `\bullet`

`\uv{kuličkou}` `\item` Zde \Rightarrow

bude další složka našeho

výčtu `\item` Třetí a poslední

položka

- Prvky výčtu budou vyznačeny symbolem \bullet „kuličkou“

- Zde bude další složka našeho výčtu

- Třetí a poslední položka

¹⁴zarovnaný okraj je vlevo (`flushleft`).

4.5.2 Prostředí `enumerate`

Toto prostředí vytváří číslovaný seznam položek. Čísla jsou přidělována automaticky. Prostředí `enumerate` je velmi podobné prostředí `itemize`. Rozdíl je v tom, že každá položka `\item` se očísluje. V předchozí ukázce by tedy na místo „kuliček“ přišly čísla (konkrétně: 1. , 2. , 3.). Budeme-li vkládat několik numerických výčtů do sebe, pak vnitřní výčet bude označen jinak než vnější¹⁵.

4.5.3 Prostředí `description`

Popisné prostředí, které popisuje význam určitých hesel. Tato zvolená hesla jsou automaticky sázena tučně. Do výčtu se hesla vkládají pomocí parametru¹⁶ řídicího slova uvedeným mezi závorkami `[_]`. Chceme-li vysázet hranaté závorky uvnitř hesla, musíme celé heslo uzavřít do složených závorek.

Příklad:

<code>\begin{description}</code>		
<code>\item[{Pole}]</code>	Chtěl jsem	Pole Chtěl jsem jenom ukázat
jenom ukázat příklad s		příkladem s hranatými závorkami v hesle.
hranatými závorkami v hesle.	⇒	
<code>\item[Myš]</code>	Malé zvířátko	Myš Malé zvířátko.
<code>\end{description}</code>		

5 Členění dokumentu

Členění dokumentu je v L^AT_EXu umožněno pomocí příkazů, které usnadňují vytváření kapitol, podkapitol, poznámek atd. Je možné vytvořit obsah a to automaticky.

5.1 Kapitoly a jiné oddíly

Členění textu do oddílu závisí na typu textu. Pro typ textu `report` je možné členění na kapitoly (chapters), oddíly (section), pododdíly (subsection), podpododdíly (subsubsection), odstavec (paragraphs) a pododstavec (subparagraphs). Podobné členění má i typ `book`. Typ textu `article` nemá členění na kapitoly, jeho jednotkou členění je oddíl.

Uvození kapitoly nebo podkapitoly se zadává příkazy `\chapter{název}`, `\section{název}`, `\subsection{název}`, L^AT_EX všechny tyto oddíly očíslovuje. Pododdíly bude číslovat více číslicemi, kde první číslice znamená číslo kapitoly, další by znamenalo číslo podkapitoly atd. (např.

¹⁵Odlišné označení dvou vnořených výčtů může být vytvářeno pomocí číslic a písmen

¹⁶parametr je nepovinný

3.1). Číslování je prováděno automaticky, při vkládání nové kapitoly se tedy nemusíme o přečíslování starat. Dále je postaráno o odsazení a velikost písma. Existuje i „hvězdičková“ modifikace příkazu `\section*`, který způsobí vytisknutí neočíslovaného nadpisu.

Je možné uvést v *preambuli* řídicí slova `\title{...}`, `\author{...}` a `\date{datum}`. Po zadání titulu, autora a datumu, je možné tyto údaje zpracovat příkazem `\maketitle`, který všechny tyto údaje vysází a odsadí. Jestliže je autorů dokumentu více, pak se autoři (v `\author{}`) oddělují příkazem `\and`.

5.2 Vytváření obsahu

Vytváření obsahu je poměrně jednoduché. Vše se provede jedním příkazem `\tableofcontents`, umístěným tam, kde se má vytvořit obsah. Budou postupně brány názvy kapitol, nadpisů atd., které budou umísťovány do obsahu spolu s číslem stránky. V případě nutnosti lze vložit jiný text do obsahu než je název nadpisu.

`\subsection[Začínáme]{Naše podkapitola}`. V obsahu se objeví řádek *Začínáme.....* „číslo strany“. Pokud se má obsah umístit na novou stránku, použije se příkaz `\newpage`. Jestliže nastane situace, že se neobjevil náš požadovaný obsah, je to proto, že při prvním překladu se potřebné informace (názvy a čísla stránek) zapíšou do souboru `text.aux`. Po druhém překladu se teprve vysází obsah, který vychází z informací čerpaných ze souboru `text.aux`.

5.3 Poznámka pod čarou

V některých částech dokumentu jsou poznámky pod čarou nezbytnou součástí. Poznámky jsou sázeny menším stupněm písma (footnote-size). Uvedme si rovnou příklad: Poznámka se vytváří jednoduše `\footnote{text poznámky}`. Poznámky se automaticky číslovají, takže o značení poznámek se nemusíme starat.

Vlastnoručně číslovaná poznámka `\footnote[5]{text ...}` s číslem pět.

5.4 Okrajové poznámky

Okrajové poznámky jsou sázeny vždy na vnějším okraji dokumentu ve formě odstavce. Tedy pokud bude dokument psán ve dvoustranném režimu, bude poznámka na levém okraji na každé sudé straně. Poznámka se vytváří příkazem `\marginpar{text}`

Naše první okrajová poznámka

5.5 Číslování stránek

Změna způsobu číslování stránek se provádí příkazem `\pagenumbering{}`, jehož parametry mohou být `roman` (čísluje se malými římskými číslicemi) a `arabic` (normální číslování). Jiná možná nastavení, poskytuje příkaz `\pagestyle` s parametry:

plain - číslo stránky je uvedeno uprostřed paty stránky.

empty - číslování není vypisováno

headings - v hlavičce je umístěno číslo stránky a název aktuální kapitoly

myheadings - lze nastavit vlastní úpravu umístění čísla pomocí příkazů `\markright` nebo `\markboth`

6 Tabulky

V této části si ukážeme, práci s tabulkami a jejich umisťování v textu na vhodné místo.

6.1 Prostředí `tabbing`

Prostředí `tabbing` umožňuje rozmístit části textu ve vodorovném směru. Produktem tohoto prostředí vlastně není tabulka, ale uspořádaný text do sloupců a řádků. Řídící sekvence mají speciální funkce, které umožňují formátovat text v jednoduché tabulce, viz tab. 4

Sekvence	Popis
<code>\=</code>	slouží pro nastavení zarážky v tabulce
<code>\></code>	přechod k následující zarážce
<code>\<</code>	vrátí se k předcházející zarážce
<code>\\</code>	přesune se na nový řádek sazby
<code>\kill</code>	řádek, na kterém je příkaz uveden, nebude vysázen
<code>\+</code>	posun levého okraje o jednu zarážku vpravo
<code>\-</code>	nastaví levý okraj o jednu zarážku vlevo
<code>\'</code>	umístí text jeho pravým okrajem od zarážky
<code>\‘</code>	posune text k pravému okraji zarážky
<code>\pushtabs</code>	uložení pozic všech zarážek
<code>\poptabs</code>	obnoví uchované nastavení zarážek, které byly uloženy <code>\pushtabs</code>
<code>\a=</code>	tento příkaz (spolu s <code>\a'</code> , <code>\a‘</code>) slouží pro vytváření akcentu

Tabulka 4: příkazy pro prostředí `tabbing`

Příklad:

```

\begin{tabbing}
Hmotnost \= Jejich cena \=      Hmotnost Jejich cena Výsledek
Výsledek \\\                     3 kg      2031      levné
3 kg \> 2031 \> levné \\\          2 kg      5000      drahé
2 kg \> 5000 \> drahé \\\
\end{tabbing}

```

Řídící sekvence `\=` v příkladu nastavuje zarážku v dané pozici na řádku. Jak vidíme ve vysázeném výsledku příkladu i text oddělený těmito sekvencemi se zobrazí na výstupu. Často budeme potřebovat nastavit zarážky podle nejširších položek v tabulce. Potom na první řádek dáme nejširší části (položek) jednotlivých sloupců, odděleny sekvencemi `\=`, ale nebudeme končit řádek obvyklým způsobem `\\`, ale užijeme `\kill`. Příkaz `\kill` zachová nastavení zarážek, ale tento řádek se ne výstupu neobjeví.

Příklad:

```

\begin{tabbing}
procesor \= cena \= počet \kill
Zboží \> Cena \> \‘Počet \\\
procesor \> 2100 \> \‘1 \\\
monitor \> \+ \> \‘1
\end{tabbing}

```

6.2 Vlastní tabulky

Prostředí `tabular` slouží k vytváření tabulek. Šířky sloupců v těchto tabulkách jsou určovány automaticky. Lze také jednoduše vytvářet čáry oddělující sloupce nebo řádky. Prostředí má dvě varianty:

1. základní variantu – `\begin{tabular}[par]{sloupce} ... \end{tabular}`
2. variantu s hvězdičkou – `\begin{tabular*}[par]{šířka}{sloupce} ... \end{tabular*}`

V základní variantě šířka tabulky odpovídá obsahu jednotlivých sloupců, zatímco varianta s hvězdičkou umožňuje vytvořit tabulku, jejíž celková šířka je zadána dalším povinným parametrem (*šířka*). Nepovinný parametr *par* určuje způsob připojení tabulky k okolnímu textu. Není-li parametr uveden, tabulka je připojena středem (*t* – připojení horním okrajem, *b* – připojení dolním okrajem).

Příklad:

```

\begin{tabular}{|c|c|}
\hline
Zboží & Cena \\
\hline
procesor & 2100 \\
tiskárna & 5000 \\
\hline
\end{tabular}

```

⇒

Zboží	Cena
procesor	2100
tiskárna	5000

V parametru *sloupce* se uvádí počet sloupců a svislých čar. Znak `|` určuje, kde se v tabulce objeví svislé čáry. Písmena určují zarovnání obsahu sloupců, které jsou:

l – obsah je zarovnán k levému okraji

r – obsah je zarovnán k pravému okraji

c – sazba textu na střed

V uvedeném příkladu tabulky, budou obsahy obou sloupců sázeny na střed. Dále jsme v tomto příkladu použili příkaz `\hline`, který vysází vodorovnou čáru. Podobně lze použít příkaz `\cline{x-y}`, kde *x* je číslo prvního sloupce a *y* je číslo posledního sloupce. Tento příkaz vysází vodorovnou čáru přes uvedené sloupce v parametru.

Je možné použít k oddělení dvou sloupců jiného znaku. Chceme-li například, aby mezi dvěma sloupci byl místo svislé čáry znak \Rightarrow , zapíšeme do hlavičky tabulky výraz `@{ \Rightarrow }`.

Příklad:

```

\begin{tabular}{l@{ $\Rightarrow$ }r}
\hline
\multicolumn{2}{|c|}{Slovíčka} \\
\hline
boy & chlapec \\
girl & dívka \\
\hline
\end{tabular}

```

⇒

Slovíčka	
boy \Rightarrow	chlapec
girl \Rightarrow	dívka

V příkladu je použit příkaz `\multicolumn`, který umístí text přes více sloupců. Způsob zápisu je: `\multicolumn{počet}{šířka}{text}`. Kde *počet* je číslo, udávající přes kolik sloupců chceme umístit *text* s uvedeným způsobem v parametru *šířka*.

6.3 Prostředí table

Toto prostředí způsobí, že jeho obsah bude umístěn odděleně od okolního textu. Přesné umístění tabulky závisí na velikosti volného prostoru a na volitelném parametru.

`\begin{table}[par]`. Význam tohoto parametru je následující:

h – pokusí se umístit tabulku na aktuálním řádku (here),

b – pokud je to možné, umístit tabulku v dolní části stránky (bottom),

t – bude preferovat umístění tabulky v horní polovině stránky (top),
p – umístí tabulku na novou stránku (page).

Tyto parametry lze kombinovat, a to zápisem znaků za sebou ([hp]). Snaha umístit tabulku, bude v pořadí parametrů. Tedy nejdříve bude snaha ji umístit na aktuální pozici nebo pro nedostatek místa se vyčlení nová stránka pro umístění.

Popis tabulky s automatickým číslováním lze dostat užitím příkazu `\caption{popisek}`. Chceme-li se v textu odkázat na tuto tabulku, umístíme za příkaz `\caption` sekvenci `\label{označení tab.}`. V textu, kde má být umístěn odkaz na tabulku, jenom použijeme příkaz `\ref{označení tab.}`.

7 Matematická sazba

V mnoha vědeckých a technických dokumentech se vyskytuje matematická sazba. L^AT_EX má velice dobré prostředky pro takovou sazbu, kterými jsou matematická prostředí. V těchto prostředích je automaticky pro sazbu zvolena italika a navíc jsou ignorovány mezery. Pro sazbu mezer se používají příkazy viz tab. 5. Je-li za potřebí v matematickém prostředí zapsat nějaký text, uzavíráme ho do parametru příkazu `.` Text bude vysázen typem písma, který byl zvolen před začátkem matematického prostředí.

Příkaz ¹⁷	Ukázka \Rightarrow Výsledek	Příkaz	Ukázka \Rightarrow Výsledek
	<code>\$s_s\$</code> \Rightarrow <i>ss</i>	<code>\,</code>	<code>\$s\,s\$</code> \Rightarrow <i>s s</i>
<code>\!</code>	<code>\$s\!s\$</code> \Rightarrow <i>ss</i>	<code>\:</code>	<code>\$s\:s\$</code> \Rightarrow <i>s s</i>
<code>\;</code>	<code>\$s\;s\$</code> \Rightarrow <i>s s</i>	<code>\quad</code>	<code>\$s\quad s\$</code> \Rightarrow <i>s s</i>
<code>_</code>	<code>\$s_s\$</code> \Rightarrow <i>s s</i>	<code>\qquad</code>	<code>\$s\qquad s\$</code> \Rightarrow <i>s s</i>

Tabulka 5: Mezery v mat. prostředí

V matematickém režimu jsou řídicí sekvence pozměněny tak, aby usnadnila sázení rovnic a vzorců.

7.1 Prostředí math

Toto prostředí slouží pro sazbu vzorců v textu jako „ $x^2 + 4x + 5 = 0$ “. Místo hraničních příkazů `\begin{math}` a `\end{math}`, umožňuje L^AT_EX psát `\(` a `\)` nebo znak `$` pro začátek i konec prostředí.

Příklad:

Okolní text před `$x^2+4x+5= 0$` \Rightarrow Okolní text před $x^2 + 4x + 5 = 0$ a
a po matematickém režimu. po matematickém režimu.

¹⁷Uvedené příkazy platí pro všechna matematická prostředí.

7.1.1 Dolní a horní indexy

V příkladu z předcházející kapitoly bylo použito horního indexu¹⁸ pro sazbu x^2 . Jak vidíme horní index se sází pomocí znaku `^`. Naopak pro sazbu dolního indexu se používá znak `_`. Vše budeme demonstrovat v následujícím příkladu.

Příklad:

<code>\(\mbox{Horní Index:}x^{5y}\quad x^{y^2}\)</code>	\Rightarrow	Horní Index: x^{5y} x^{y^2}
Dolní index: <code>\$X_1\,,\dots X_n\$</code>	\Rightarrow	$X_1 \dots X_n$
Obojí současně: <code>\$X^{2}_{n}\$</code>		X_n^2

7.2 Další matematická prostředí

Prostředí `displaymath`

Podobně jako u prostředí `math` lze místo příkazů `\begin{displaymath}` a `\end{displaymath}` použít ekvivalentní dvojice `\[` pro začátek a `\]` pro konec nebo také `$$` pro začátek i konec. Jsou zde také zakázány prázdné řádky.

Příklad:

		Okolní text před
Okolní text před <code>\$\$x^2+4x+5= 0</code>	\Rightarrow	$x^2 + 4x + 5 = 0$
<code>\$\$</code> a po matematickém režimu.		a po matematickém režimu.

Prostředí `equation`

Je stejné jako předchozí prostředí, jen navíc čísluje rovnice na pravém okraji.

Příklad:

Okolní text před		Okolní text před
<code>\begin{equation}</code>		
<code>equation x^2+4x+5= 0</code>	\Rightarrow	$x^2 + 4x + 5 = 0$ (1)
<code>\end{equation}</code>		
a po matematickém režimu.		a po matematickém režimu.

Prostředí `eqnarray`

Používají se pro sazbu soustavy rovnic, které mají být zarovnány. Každou rovnici je třeba rozdělit na tři části. Rozdělení na části bude prováděno stejně jako v prostředí `tabular` (viz 6.2).

Příklad:

<code>\begin{eqnarray}</code>		
<code>4x + 2y & = & 20 + z \\\</code>	\Rightarrow	$4x + 2y = 20 + z$ (2)
<code>6z - 3y & = & 0 \\\</code>		$6z - 3y = 0$ (3)
<code>x & = & 1/4</code>		$x = 1/4$ (4)
<code>\end{eqnarray}</code>		

Je-li potřeba na některém řádku vynechat číslování, uvedeme na

¹⁸Zapisování horních a dolních indexů je stejné pro všechna matematická prostředí.

něm příkaz `\nonumber`. Pro vynechání číslování u všech rovnic uvedených v tomto prostředí, stačí použít jeho „hvězdičkovou“ variantu.

7.3 Zápis matematických objektů

7.3.1 Zlomky

Často používáme zápis zlomku se šikmou zlomkovou čarou ($\frac{1}{4}$), ale někdy potřebujeme výraz zapsat pomocí vodorovné zlomkové čáry. K tomu slouží řídicí sekvence `\frac` se dvěma parametry: čitatel a jmenovatel.

Příklad:

$$\text{\$ \frac{x^2}{x+2} \cdot \frac{x+1}{3} \$} \quad \Rightarrow \quad \frac{x^2}{x+2} \cdot \frac{x+1}{3}$$

7.3.2 Odmocniny

Pro vysázení odmocniny se používá řídicí sekvence `\sqrt[řád]{argument}`, kde *řád* označuje volitelný parametr udávající řád odmocniny a argument představuje povinný parametr.

Příklad:

$$\text{\$f(x)=\sqrt[3]{x^2+5} \$} \quad \Rightarrow \quad f(x) = \sqrt[3]{x^2+5}$$

7.3.3 Matematické značky

L^AT_EX nabízí řídicí slova jako `\log`, která v matematickém režimu automaticky produkují odpovídající matematickou značku. Taková řídicí slova mohou zaznamenat názvy funkcí apod. Některá z nich mají zajímavé chování ve vztahu k dolním indexům. Příklady řídicích slov ukazuje tab. 6.

<code>\cos</code>	<code>\sin</code>	<code>\arcsin</code>	<code>\arccos</code>	<code>\deg</code>	<code>\sup</code>	<code>\inf</code>	<code>\limsup</code>
<code>\sinh</code>	<code>\cosh</code>	<code>\cot</code>	<code>\tan</code>	<code>\arctan</code>	<code>\dim</code>	<code>\ker</code>	<code>\liminf</code>
<code>\exp</code>	<code>\min</code>	<code>\max</code>	<code>\tanh</code>	<code>\lim</code>	<code>\ln</code>	<code>\log</code>	<code>\sec</code>
<code>\infty</code>	<code>\bmod</code>	<code>\pmod</code>					

Tabulka 6: Mat. značky

V následujících tabulkách (tab. 7, 8, 9 a 10) si ukážeme další používané matematické značky.

α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	δ	<code>\delta</code>	ϵ	<code>\epsilon</code>
ε	<code>\varepsilon</code>	ζ	<code>\zeta</code>	η	<code>\eta</code>	θ	<code>\theta</code>	ϑ	<code>\vartheta</code>
ι	<code>\iota</code>	κ	<code>\kappa</code>	λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>
ξ	<code>\xi</code>	o	<code>o</code>	π	<code>\pi</code>	ϖ	<code>\varpi</code>	ρ	<code>\rho</code>
ϱ	<code>\varrho</code>	σ	<code>\sigma</code>	ς	<code>\varsigma</code>	τ	<code>\tau</code>	v	<code>\upsilon</code>
ϕ	<code>\phi</code>	φ	<code>\varphi</code>	χ	<code>\chi</code>	ψ	<code>\psi</code>	ω	<code>\omega</code>
Γ	<code>\Gamma</code>	Δ	<code>\Delta</code>	Θ	<code>\Theta</code>	Λ	<code>\Lambda</code>	Ξ	<code>\Xi</code>
Π	<code>\Pi</code>	Σ	<code>\Sigma</code>	Υ	<code>\Upsilon</code>	Φ	<code>\Phi</code>	Ψ	<code>\Psi</code>
Ω	<code>\Omega</code>								

Tabulka 7: Řecká abeceda

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\Lleftarrow	<code>\Lleftarrow</code>
\Longleftarrow	<code>\Longleftarrow</code>	\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\longleftrightarrow	<code>\longleftrightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\searrow	<code>\searrow</code>	\nleftarrow	<code>\nleftarrow</code>	\nrightarrow	<code>\nrightarrow</code>

Tabulka 8: Šipky

Některé matematické symboly lze pro vyjádření negace přeškrtnout, pokud je před ním použit příkaz `\not` ($x \not< y \Rightarrow x \not\leq y$).

Příklad:

$$\begin{array}{ccc}
 \lim_{n \rightarrow \infty} & & \sum_{i=1}^n x_i f_i^2 \\
 \text{\texttt{\$ \$ \lim_{n \rightarrow \infty} \texttt{\$ \$}} & \Rightarrow & \text{\texttt{\$ \sum_{i=1}^n x_i f_i^2 \$}} \\
 \text{\texttt{\$ \$ \sum_{i=1}^n x_i f_i^2 \$ \$}} & & \sum_{i=1}^n x_i f_i^2
 \end{array}$$

V příkladu jsou výrazy umístěny nad a pod symbol, v textovém vzorci jsou umístěny vedle symbolu. Příkazem `\limits` lze nařídit, aby se výraz umístil nad a pod symbol bez ohledu na prostředí. Opačně pracuje příkaz `\nolimits`. Oba se píší bezprostředně za symbol, kterého se týkají.

\geq	<code>\geq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>	\leq	<code>\leq</code>
\perp	<code>\perp</code>	\simeq	<code>\simeq</code>	\parallel	<code>\parallel</code>	\approx	<code>\approx</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>

Tabulka 9: Relace

\pm	<code>\pm</code>	\mp	<code>\mp</code>	\cap	<code>\cap</code>	\cup	<code>\cup</code>
\oplus	<code>\oplus</code>	\otimes	<code>\otimes</code>	\div	<code>\div</code>	\odot	<code>\odot</code>
\vee	<code>\vee</code>	\wedge	<code>\wedge</code>	\times	<code>\times</code>	$*$	<code>\ast</code>
\star	<code>\star</code>	\setminus	<code>\setminus</code>	\bullet	<code>\bullet</code>	\diamond	<code>\diamond</code>
\emptyset	<code>\emptyset</code>	\forall	<code>\forall</code>	\exists	<code>\exists</code>	\neg	<code>\neg</code>
\sum	<code>\sum</code>	\prod	<code>\prod</code>	\oint	<code>\oint</code>	\int	<code>\int</code>

Tabulka 10: Binární operátory a jiné symboly

$($	<code>(</code>	$)$	<code>)</code>	$[$	<code>[</code>	$]$	<code>]</code>	$\{$	<code>\{</code>
$\}$	<code>\}</code>	\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>	\lceil	<code>\lceil</code>	$ $	<code> </code>
\rceil	<code>\rceil</code>	\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	\parallel	<code>\parallel</code>		

Tabulka 11: Závorky

Příklad:

```
\oint\limits_\gamma x^2\,dx + \implies \oint_\gamma x^2 dx + x^5 y dy
x^5 y\,dy $
```

Oddělovače uvedené v tabulce 11, mohou měnit velikost podle výrazu, který ohraničují. Pro dosažení správné velikosti je třeba před příslušný oddělovač zapsat sekvenci `\left` a `\right`. Tyto dvě sekvence tvoří vždy pár. Chceme-li vysázet oddělovač jen na jedné straně, použijeme u párové sekvence místo zápisu oddělovače pouze tečku.

V prostředí `array` slouží pro vytváření matic. Počáteční příkaz má povinný parametr, který má stejné složení jako u prostředí `tabular` viz 5.2. Dále má ještě jeden volitelný parametr, který určuje způsob připojení k okolnímu textu viz 5.2.

Příklad:

```
\left( \frac{x+1}{|x+3|} \right) $
\ln \left| \frac{z+2}{z} \right| $
$ \left| \begin{array}{ccc}
4 & 5 & 1 \\
0 & 2 & 1 \\
1 & 3 & -2 \end{array} \right|
\end{array} \right| $
$ f(x)= \left\{ \begin{array}{c}
0, \text{ \& } x>0 \\
1, \text{ \& } x=0 \end{array} \right.
\end{array} \right. $
```

$$\left(\frac{x+1}{|x+3|} \right)$$

$$\ln \left| \frac{z+2}{z} \right|$$

$$\implies \left| \begin{array}{ccc} 4 & 5 & 1 \\ 0 & 2 & 1 \\ 1 & 3 & -2 \end{array} \right|$$

$$f(x) = \begin{cases} 0, & x > 0 \\ 1, & x = 0 \end{cases}$$

Z mnoha dalších příkazů v matematickém režimu ještě uvedme příkazy `\overbrace` a `\underbrace`. Těmito příkazy lze vytvořit vodorov-

nou svorku nad nebo pod daným objektem. `\vec` je další důležitý příkaz pro sazbu vektorů¹⁹.

7.4 Balík `amsmath`

`Amsmath` je balík L^AT_EXu, který rozšiřuje a ulehčuje možnosti sazby matematických částí textu. Vše co nám poskytuje balík `amsmath`, dokážeme vysázet i bez tohoto balíku. Takové psaní dokumentu je zdlouhavé a tudíž nepraktické.

Balík `amsmath` je distribuován společně s dalšími balíky²⁰, které nějakým způsobem rozšiřují jeho působnost. Bohužel si řekneme jenom o balíku `amsmath`, a to okrajově.

Jak již jsme řekli v části 2.2.1, balík se připojuje příkazem `\usepackage`. Jako volbu lze u `amsmath` uvést:

`centertags` – tato volba je nastavena automaticky. Slouží pro umístění čísla rovnice, které bude zarovnané podle čísla nejdelší rovnice.

`sumlimits` – umístí index nad a pod symbol suma. Opět je tato volba nastavena automaticky a není ji třeba uvádět.

`nosumlimits` – vždy umístí index vedle symbolu suma.

`intlimits` – podobná volba jako u `sumlimits`, ale nyní se jedná o symbol integrálu.

`nointlimits` – opak `intlimits`, ale oproti `intlimits` je tato volba nastavena implicitně.

`namelimits` – index bude sázen pod operator (např.: `det`, `min`, `max`, `lim`, ...), pokud se vyskytnou v prostředí `displaymath`.

`nonamelimits` – opak `namelimits`

`leqno` – čísla rovnic se umístí na levou stranu od rovnice

`reqno` – čísla rovnic se vysází vpravo

Tyto volby lze samozřejmě mezi sebou kombinovat. V takovém případě více voleb oddělujeme čárkou.

V tomto balíku existuje několik prostředí, např.:

`equation`, `gather`, `multline`, `align`, `flalign`, `alignat` a `split`.

Všechna prostředí až na `split`, mají i hvězdičkovou variantu, kde bez hvězdičková forma automaticky čísluje jednotlivé rovnice. Je možné

¹⁹Je-li potřeba vysázet něco takového \vec{i} , potom je nutné odstranit příkazem `\imath` tečku nad písmenem `i` (podobně pro písmeno `j` slouží příkaz `\jmath`).

²⁰Mezi takové další balíky patří: `amstext`, `amsbsy`, `amscd`, `amsxtra`.

potlačit číslování u rovnic, kde vložíme příkaz `\notag` před `\\`. Symbol `\\` značí konce řádku a části rovnice se oddělují znakem `&` (stejně jako v prostředí `tabular`).

Prostředí `split` je podřízené prostředí, které může být jenom v jiném z uvedených prostředí, kromě `multline`.

7.4.1 Prostředí `amsmath`

Multline

Toto prostředí je obdobou `equation` (viz 7.2) s tím rozdílem, že se používá tam, kde nestačí pro napsání rovnice jeden řádek. Tato rovnice bude číslována jen jednou. Samozřejmě je možné potlačit číslování příkazem `\notag`. Příkaz `\shoveleft`, resp. `\shoveright` pro umístění neokrajových řádků vlevo, resp. vpravo.

Příklad:

$$\begin{array}{l} \backslash\mathrm{begin}\{\mathrm{multline}\} \\ a+b+c+d+e \\ +2f+3j+6a+k+3c \\ +3e+f+b+a \\ \backslash\mathrm{end}\{\mathrm{multline}\} \end{array} \quad \Rightarrow \quad \begin{array}{l} a+b+c+d+e \\ +2f+3j+6a+k+3c \\ +3e+f+b+a \end{array} \quad (5)$$

Split

Podobně jako `multline`, slouží prostředí `split` pro sázení rovnic, které jsou příliš dlouhé pro sazbu na jeden řádek. `Split` provádí připojení mezi řádky pomocí značky přípojného bodu `&`. Rovnice tady nejsou číslovány, protože o to se postará prostředí, ve kterém je `split` umístěn.

Příklad:

$$\begin{array}{l} \backslash\mathrm{begin}\{\mathrm{equation}\} \\ \backslash\mathrm{begin}\{\mathrm{split}\} \\ f(x,y)= & x^2+3x-2y^3 \\ & -3y+10 \\ \backslash\mathrm{end}\{\mathrm{split}\} \\ \backslash\mathrm{end}\{\mathrm{equation}\} \end{array} \quad \Rightarrow \quad \begin{array}{l} f(x,y)=x^2+3x+y^3 \\ -3y+10 \end{array} \quad (6)$$

Align

Prostředí `align` se používá pro zápis dvou nebo více rovnic na jeden řádek. Je také možné přidat poznámku aplikací příkazu `\text`

Příklad:

$$\begin{array}{l} \backslash\mathrm{begin}\{\mathrm{align}\} \\ x&=y \quad \& \quad x^2+3&=y \\ x'&=y' \quad \& \quad x'+3&=y'+1 \\ f(x)&=x+2 \quad \&\& \quad \backslash\mathrm{text}\{\mathrm{Lineární} \\ \mathrm{rovnice}\} \quad \backslash\mathrm{notag} \\ \backslash\mathrm{end}\{\mathrm{align}\} \end{array} \quad \Rightarrow \quad \begin{array}{l} x=y \quad x^2+3=y \quad (7) \\ x'=y' \quad x'+3=y'+1 \quad (8) \\ f(x)=x+2 \quad \text{Lineární rovnice} \end{array}$$

Podobně jako příkazem `\text` lze dosáhnout vysázení poznámky,

je možné příkazem `\intertext` vložit do matematické sazby nějaký krátký text.

Příklad:

$$\begin{aligned}
 &\begin{array}{l} \backslash\begin{array}{l} \backslash\cos^2(x) + \backslash\sin^2(x) \\ \backslash\intertext{nebo} \backslash\sin^2(x) \&=1 - \backslash\cos^2(x) \end{array} \end{array} \quad \Rightarrow \quad \text{nebo} \quad \sin^2(x) = 1 - \cos^2(x) \quad (9)
 \end{aligned}$$

7.4.2 Matice

Balík `amsmath` také poskytuje prostředí pro tvorbu matic. Jsou to:

`pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` a `Vmatrix`

Tato prostředí vychází ze standardního prostředí L^AT_EXu `array`, které vytvoří oddělovače popořadě `()`, `[]`, `{}`, `||` a `|||`. Příkaz `\hdotsfor{číslo}` vysází na řádek tečky přes uvedený počet sloupců.

Příklad:

$$\begin{aligned}
 &\begin{array}{l} \backslash\begin{array}{l} \backslash\begin{array}{l} \backslashhdotsfor{3} \\ \backslashend{array} \end{array} \end{array} \quad \Rightarrow \quad \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & \dots \end{pmatrix}
 \end{aligned}$$

Je možné vysázet „malou“ matici v textu pomocí `smallmatrix` (např.: $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$).

7.4.3 Integrál

Dvojměrné a trojměrné integrály se v balíku `amsmath` vysází pomocí příkazů `\iint`, `\iiint`, `\iiiint`. Porovnejme integrál vysázený pomocí příkazu `\iint` resp. `\int\int`.

Příklad:

$$\begin{aligned}
 &\begin{array}{l} \backslash\begin{array}{l} \backslash\int\int \frac{x^2}{x+2} dx \\ \backslash\iint \frac{x^2}{x+2} dx \end{array} \end{array} \quad \Rightarrow \quad \begin{array}{l} \int\int \frac{x^2}{x+2} dx \\ \iint \frac{x^2}{x+2} dx \end{array}
 \end{aligned}$$

8 Odkazy

V dokumentech je nutné často použít odkaz na jinou část textu (Např.: „... další příklad je na str. 20“). Bylo by velice nepraktické, nutně znát předem číslo stránky (číslo kapitoly, obrázku nebo tabulky) u každého uvedeného odkazu. Všechny takové informace jsou určeny až při překladu, proto je v L^AT_EXu k dispozici příkaz `\label`. Tento příkaz propojí

informace získané při překladu s jediným povinným parametrem (*symbolické jméno*). V místě textu, kde hodláme vysázet příslušný odkaz, jenom uvedeme příkaz `\ref{symbolické jméno}`²¹ (pro číslo strany použijeme `\pageref{symbolické jméno}`). Při prvním překladu se na místě odkazu nejprve objeví dva symboly ??, teprve při druhém překladu se ?? nahradí číslem stránky (kapitoly, obrázku nebo tabulky).

Příklad:

V textu na straně 14 napíšeme:

```
\label{Juraj}   Juraj Janošik
je historickou postavou, ale
...
```

Při prvním překladu:

Umístění odkazu ze str. ?? v
oddíle ??, jsem ...

⇒

Odkaz:

Umístění odkazu ze str.

```
\pageref{Juraj}   v oddíle
\ref{Juraj}, jsem ...
```

Po druhém překladu:

Umístění odkazu ze str. 14 v
oddíle 4.1.2, jsem ...

8.1 Tvorba obsahů a odkazů

Příkaz `\tableofcontents` produkuje obsah. Na začátku čte L^AT_EX dosavadní verzi souboru s rozšířením `.toc` a vytváří obsah i s nadpisem. Soubor s rozšířením `.toc` vznikl při předchozím překladu, kde jsou zapisovány titulky s čísly stránek. Podobně pracují i příkazy `\listoffigures` a `\listoftables`, které vytvářejí seznam obrázků, resp. tabulek. U nečíslovaných titulků se obsah nevytváří. K tomu slouží příkaz `\addcontentsline{obsahový soubor}{úroveň textu}{text}`, kde obsahový soubor může být `toc` (pro obsah), `lot` (pro obrázky) a `lof` (pro tabulky). Dále úroveň textu představuje titulek příslušné úrovně.

Odkaz na citaci v textu lze vyjádřit buď číslem, nebo zkratkou jména autora a rokem vydání. Odkaz na položku seznamu citací vytváří příkaz `\cite{název}`. Pokud seznam citací píšeme do dokumentu, použijeme k tomu účelu prostředí `thebibliography`. Položky seznamu začínají příkazem `\bibitem`, jehož povinným parametrem je `název odkazu`, použitý v příkazu `\cite`. Příkaz `\bibitem` automaticky produkuje pořadové číslo dané literatury. Dále prostředí obsahuje `text`, představující jméno autora, název knihy (literatury), atd.

9 Obrázky

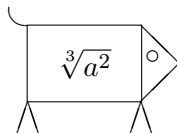
V L^AT_EXu je možné vkládat už vytvořené obrázky do textu. Je také možné obrázky přímo „kreslit“ (spíše jde o zaznamenání souřadnic kon-

²¹Pokud je uveden příkaz `\label` v prostředí `table`, potom příkaz `\ref` vyprodukuje číslo tabulky (podobně pro prostředí `figure`)

cových bodů přímek atd.), ale tato činnost přesahuje rámec této příručky.

V této příručce se snažím zaměřit na základy psaní v L^AT_EXu, tedy není žádoucí ukázat všechny zákoutí tohoto systému. Určitě je někdy lepší nakreslit si svůj obrázek (graf, nákres, diagram atd.) v jiném kreslicím programu a hotový produkt jenom přidat do našeho textu na příslušné místo. Takové vytváření potřebného grafického útvaru, může být jednodušší, protože se k jeho tvorbě nevyužívá jenom klávesnice. Samozřejmě není nutné vždy vytvářet nový obrázek, stačí použít již vytvořený.

Jako demonstraci příkladu, který je možné vytvořit pomocí L^AT_EXu, je obrázek 1. Tento obrázek je popsáný zhruba deseti řádky příkazů.



Obrázek 1: Ukázka obrázku

9.1 Balík color

Tento balík předpokládá existenci ovladače, který je schopen pracovat s vkládanými obrázky. Je možné měnit barvu textu nebo si definovat vlastní barvy.

Definice barev

Barvy jako `black`, `red`, `green`, `blue`, `cyan`, `magenta`, `yellow` jsou už předdefinované. Definování vlastních barev je umožněno prostřednictvím příkazu `\definecolor{name}{model}{specifikace}`, kde `name` představuje jméno definované barvy a `specifikace` číselne vyjádření zvoleného modelu (příklad modelu je `rgb` nebo `gray`).

Příklad

```
\definecolor{modra}{rgb}{0.8,0.85,1}
```

```
\textcolor{red}{Obsah} \Rightarrow \textcolor{red}{Obsah}
```

Používání definovaných barev se děje pomocí příkazu `\color{name}`, kde `name` souhlasí s jménem definované barvy. Podobný příkaz pro změnu barvy je `\textcolor{name}{text}`, který vysází pouze text uvedený v závorkách.

9.2 Balík graphicx

V balíku `graphicx` se předpokládá existence ovladače, který je schopen pracovat s obrázky. Balík vyžaduje výstup v jazyce PostScript. Vý-

stupní soubor .dvi, který je získán obvyklým překladem, musí být tedy konvertován. Je možné použít různé programy na konvertování, jako například dvips. Výsledek takového programu je postscriptový soubor, který je možné vytisknout nebo zobrazit pomocí nástrojů, schopny přijímat postscriptový vstup. Součástí tohoto balíku jsou příkazy, které manipulují s obrazem vysázeného textu a umožňují zvláštní efekty.

Rotace a jiné úpravy textu

Rotaci textu umožňuje příkaz `\rotatebox{úhel}{text}`, kde úhel je vyjádřený ve stupních. Dále je možné zrcadlové převrácení textu (`\reflectbox`), změna velikosti textu v ose x a y (`\scalebox`) a vyplnění textu do připraveného obdélníka (`\resizebox`).

Příklad

<code>\rotatebox{90}{Text}</code>	\Rightarrow	
<code>\reflectbox{Flop}</code>		

Vkládání obrázku ze souboru

Pomocí L^AT_EXu nebo L^AT_EXových doplňujících balíků je možné vytvořit i obrázek. Mnohem jednodušším způsobem, jak dostat grafiku do dokumentu, je vytvořit daný obrázek speciálním programem. Potom vložit hotový obrázek do dokumentu. Řekneme si o jednoduchém a často používaném způsobu prostřednictvím EPS (ENCAPSULATED POSTSCRIPT).

Nejprve musíme mít tento obrázek uložený jako EPS soubor²². Na vložení souboru do dokumentu slouží příkaz

`\includegraphics[klic=hodnota]{soubor}` .

Za volitelný parametr se bere seznam čárkami oddělených klíčů a jejich hodnot. Klíče se mohou použít na změnu šířky (width), výšky (height) a otáčení vkládané grafiky (angle). Hvězdičková varianta tohoto příkazu je schopna vyříznout obdélník, jehož levý dolní roh a pravý horní roh je definován volitelnými parametry.

Příklad

`\includegraphics*[0mm,5mm][12mm,10mm]{pokus.eps}`

Pomocí prostředí `figure` poskytuje L^AT_EX základní možnosti na práci s plovoucími objekty (obrázky a jiné). V tomto prostředí nemůže nastat stránkový zlom. Tedy nestane se, aby obrázek byl „rozpůlen“ do dvou stránek. Prostředí `figure` se chová stejně jako prostředí `table`, rozdíl spočívá jen v jejich číslování a označování. Další popis prostředí `figure` viz kapitola 6.3.

²²Takový grafický prvek můžeme vytvořit a poté uložit v grafickém programu, který podporuje formát EPS

Literatura

- [1] DOOB, M.: *Jemný úvod do $T_{\text{E}}X$* . Praha, 1990
- [2] OLŠÁK, P.: *$T_{\text{E}}X$ book naruby*. Brno, Konvoj 2001
- [3] OLŠÁK, P.: *Typografický systém $T_{\text{E}}X$* . Brno, Konvoj 2000
- [4] RYBIČKA, J.: *\LaTeX pro začátečníky*. Brno, Konvoj 2003
- [5] RECKDAHL, K.: *Using Imported Graphics in $\text{\LaTeX} 2_{\epsilon}$* . 1997
- [6] CARLISLE, D.: *Packages in the ‘graphics’ bundle*. 1999