

L^AT_EX workshop s Vítkem

„Studenti studentům“ na FEL ČVUT

Vít Valečka

18. prosince 2018

Slovo úvodem

Tento dokument, byť je součástí krátkého workshopu s tématem publikačního systému L^AT_EX, v žádném případě není kompletním popisem dovedností tohoto značkovacího jazyka, ani by neměl sloužit jako nějaká učebnice.

Cílem této práce je ukázat naprosté základy práce, upozornit na možná úskalí a navnadit čtenáře/posluchače k dalšímu studiu této problematiky, ať už samostatnému, nebo například prostřednictvím mezifakultního zápisu předmětu *01PSL Publikační systém L^AT_EX* vyučovaném na Fakultě jaderné a fyzikálně inženýrské (FJFI) ČVUT v Praze, který vyučuje doktor Petr Ambrož.

1 Trocha teorie

\TeX je program pro počítačovou sazbu. Vytvořil jej profesor Donald Ervin Knuth, který v 70. letech 20. století nebyl spokojen s tím, jak školní nakladatelství sázelo jeho skripta určená studentům. Ve skriptech se objevovalo mnoho chyb, především v matematických vzorcích, a ani typografie za mnoho nestála.

Je velmi populární zejména v akademických kruzích, zvláště v oborech jako je matematika, fyzika a informatika. Uplatnění ale může najít i v jiných oborech, např. humanitních – sazba je kvalitní nejen v případě matematiky, ale také v případě znaků různých abeced. Tímto programem byl z větší části vytlačen troff, formátovací systém preferovaný na mnoha Unixech. K dispozici je mnoho rozšiřujících balíků od velmi početných komunit uživatelů. Například existují balíky pro sazbu textu v podobě čárového kódu nebo vysázení obrázku situace šachové partie pomocí jediného příkazu. Taktéž existují komerční programové balíky.

\TeX je obecně považován za nejlepší nástroj pro sazbu složitých matematických vzorců. Je však hojně používán i v běžné sazbě, stejně tak jako jeho odvozeniny a nadstavby, zvláště pak balík maker \LaTeX . I Wikipedie používá na svých stránkách tento systém pro matematické výrazy.

Zajímavostí je způsob označování verzí programu: místo tradičního zvyšování čísla verze se označení verze \TeX u prodlužuje o další číslici desetinného rozvoje Ludolfova čísla. Za nalezení chyby v programu nabízí Donald Knuth odměnu, která v současné době činí \$327,68. Další zajímavostí je skutečnost, že Donald Knuth vyjádřil přání, ať bezprostředně po jeho smrti přestane být \TeX dále vyvíjen a verze aktuální v toto datum se stane verzí finální, označenou právě číslem pí.

\LaTeX (vyslovuje se [latech]), je balík maker programu \TeX , který umožňuje autorům textů sázet a tisknout svá díla ve velmi vysoké typografické kvalitě, přičemž autor používá profesionály předdefinovaných vzhledů dokumentu. \LaTeX byl původně napsán Leslie Lamportem. \LaTeX užívá programu \TeX jako sázecího stroje.

Od 90. let je \LaTeX rozšiřován týmem $\text{\LaTeX}3$, který vede Frank Mittelbach. Tento tým se snaží sjednotit všechny rozšiřující verze $\text{\LaTeX}u$, které postupně vznikaly od verze 2.09. Aby byla nějakým způsobem odlišena stará verze od nové, byla zatím aktuální označena jako $\text{\LaTeX}2\epsilon$.

2 Základní struktura dokumentu

2.1 Úvod do struktury

Dokument psaný v L^AT_EXu se dělí na dvě základní části – *preambuli* a *tělo dokumentu*.

Preamble obsahuje importy používaných balíčků, nastavení dokumentu (např. velikosti okrajů stránky, styly odstavců a textu atd.), informace o dokumentu samotném, definice vlastních příkazů a další.

Tělo dokumentu pak zahrnuje samotný obsah, který chceme sdělit. Patří sem tedy samotný text, kapitoly, obrázky...

Při psaní dokumentu je také dobré myslet na 5 základních vlastností:

- libovolný počet mezer v textu se bere jako jedna mezera
- mezery na začátku řádku jsou ignorovány
- prázdný řádek (popř. libovolný počet prázdných řádků) se bere jako nový odstavec

V L^AT_EXu také platí, že:

- všechny příkazy začínají \ (zpětným lomítkem)
- příkaz může mít hlavní a vedlejší parametry – `prikaz[vedlejsi]{hlavni}`
- speciální znaky se „escapují“ \ (zpětným lomítkem)

2.2 Části dokumentu

Každý L^AT_EXový dokument obsahuje tři základní značky:

`documentclass{class}` – určuje typ dokumentu (jedná se v podstatě o „šablonu“ pro vzhled). Za parametr „class“ se doplňuje název šablony. Mezi nejpoužívanější třídy dokumentu patří **article**, **book** a **report**.

`begin{document}` – značí začátek samotného dokumentu – zde se vše zapisuje.

`end{document}` – značí konec samotného dokumentu – vše za touto značkou se ignoruje.

Příklad 1 – Hello World!

Kód níže vytvoří dokument, jehož jediným obsahem bude text „Hello World!“.

```
\documentclass{article}
\begin{document}
  Hello World!
\end{document}
```

3 Načítání rozšiřujících balíčků

Načítání balíčků, které rozšiřují základní funkcionalitu se provádí v preambuli pomocí příkazu `\usepackage[]{}{}`.

Do prvních (hranatých) závorek se uvádí volitelné parametry, do druhých (složených) povinné parametry - typicky název balíčku, který načítáme.

4 Čeština v L^AT_EXu

Základní font, který je využíván, není přizpůsoben pro korektní zobrazování českého jazyka – konkrétně pak diakritiky. Z tohoto důvodu pro správnou sazbu využíváme balíček *Babel* a některou ze sad českých písem, typicky *fontenc*.

Pro korektní zobrazení speciálních znaků a češtiny je také dobré přepnout kódování dokumentu do UTF-8 popřípadě CHCP 1250.

`\usepackage[czech]{babel}` – načte balíček obsahující pravidla pro češtinu (formáty data, odstavců atd.).

`\usepackage[utf8]{inputenc}` – nastaví dokument, aby využíval pro kódování textu UTF-8.

`\usepackage[IL2]{fontenc}` – načte balíček s českým fontem.

Příklad 2 – Žluťoučký kůň

Kód níže vytvoří dokument, jehož jediným obsahem bude text „Hello World!“.

```
\documentclass{article}
\usepackage[czech]{babel}
\usepackage[utf8]{inputenc}
\usepackage[IL2]{fontenc}
\begin{document}
  Příliš žluťoučký kůň úpěl ďábelské ódy.
\end{document}
```

5 Formátování textu

Dále následuje seznam základních značek pro formátování textu.

5.1 Znaky a formát textu

`%` – jednořádková poznámka v kódu (L^AT_EX nemá víceřádkové poznámky)

`-` `(-)` – spojovník

`--` `(-)` – pomlčka

`---` `(—)` – dlouhá pomlčka

`emph{text}` (*text*) – zvýraznění textu (s ohledem na aktuálně použitou šablonu)

`textbf{text}` (**text**) – tučný text

`textit{text}` (*text*) – text psaný kurzívou

`textrm{text}` – normální text (pokud například styl dokumentu zavádí, že standardně je text kurzívou)

`underline{text}` (text) – podtržený text

`texttt{aimnl}` (`aimnl`) – neproporcionální text

`textsc{text}` (`TEXT`) – text v kapitálkách

`textsf{text}` (`text`) – bezpatkový text

`indent` – odsazení prvního řádku odstavce dle předpisů v šabloně, nebo uživatelem definovaných

`noindent` `(-)` – zruší odsazení prvního řádku odstavce

`uv{text}` („text“) – umístí text do uvozovek dle aktuálně použité jazykové sady

5.2 Tok textu

`\` – přechod na nový řádek

`newpage` – vloží konec stránky

`clearpage` – vloží konec stránky

`hspace{2em}` – vloží horizontální mezeru do textu (možno využít všechny obvyklé jednotky – px, em, cm, mm...)

`vspace{2em}` – vloží vertikální mezeru

`vspace*{2em}` – vloží vertikální mezeru mezi „něco“ a „nic“ (např. mezi text a okraj stránky)

`mbox{text}` – obaluje text, který nesmí být rozdělený na konci řádku

`begin{flushleft} ... end{flushleft}` – zarovnání textu dle zadaného parametru – existují tři možnosti `flushleft` – vlevo, `flushright` – vpravo a `center` – na střed.

5.3 Rozložení stránky

Studium tohoto tématu ponecháme na čtenáři. Omezíme se pouze na informaci, že aktuální rozložení stránky lze vypsát příkazem `layout`, zde jsou vypsány i příkazy, kterými lze jednotlivé parametry (okraje, šířku prostoru pro text...). Jako konkrétní příklad můžeme uvést např.:

```
setlength{\textheight}{24.7cm}
```

První parametr určuje, kterou vlastnost chceme měnit, druhý pak nastavenou velikost.

Lze také volit nastavení jen pro sudé nebo liché stránky:

```
setlength{\oddsidemargin}{-0.04cm}
```

A je také možné pracovat s jednotlivými již nastavenými parametry a provádět s nimi jednoduché matematické operace (sčítání, odčítání):

```
setlength{\topmargin}{-\headheight-\headsep}
```

6 Kapitoly

`section{nadpis}` – vloží číslovaný nadpis kapitoly

`subsection{nadpis}` – vloží číslovaný nadpis podkapitoly, přidáním dalších předpon „sub“ lze více vnořovat

`section*{nadpis}` – vloží nečíslovaný nadpis kapitoly, analogicky funguje také pro podkapitoly

7 Seznamy

7.1 Bodové odrážky

Příklad 3

<code>\begin{itemize}</code>	• semper vel quam at
<code>\item semper vel quam at</code>	
<code>\item rutrum tristique metus</code>	• rutrum tristique metus
<code>\item[:)] vivamus elementum lectus</code>	:) vivamus elementum lectus
<code>\end{itemize}</code>	

7.2 Číslované odrážky

Příklad 4

<code>\begin{enumerate}</code>	1. donec non hendrerit erat
<code>\item donec non hendrerit erat</code>	
<code>\item et suscipit turpis</code>	2. et suscipit turpis
<code>\item donec interdum turpis</code>	
<code>\end{enumerate}</code>	3. donec interdum turpis

7.3 Pojmenované odrážky

Příklad 5

```
\begin{description}
\item[Suspendisse] non risus ante. Donec sed pharetra massa...
\item[Quisque] non augue ut libero vehicula ullamcorper at quis...
\end{description}
```

Suspendisse non risus ante. Donec sed pharetra massa. Nam leo libero, condimentum in sollicitudin eget, porta sed quam.

Quisque non augue ut libero vehicula ullamcorper at quis mauris. Phasellus tincidunt velit eget dui malesuada ultrices.

8 Tabulky

Tvorba tabulek je v L^AT_EXu podobná tvorbě v HTML. Bez složitého úvodu si vše ukážeme na příkladu:

Příklad 6

```
\begin{tabular}{|l|l|c|p{4cm}|}  
\hline Kdo & Kolik & & Poznámka \\  
\hline Alice & 151 & & To je docela dost, otázka jak k tomu přišla. \\  
Bob & 21 & & Nic moc Bobe, zaber. \\  
\hline  
\end{tabular}
```

Kdo	Kolik	Poznámka
Alice	151	To je docela dost, otázka jak k tomu přišla.
Bob	21	Nic moc Bobe, zaber.

8.1 Formátování tabulky

`\begin{tabular}{params}` – začátek definice tabulky, slovo `tabular` specifikuje, že se jedná o tabulku, ve druhých složených závorkách je pak specifikováno rozložení tabulky (sloupce)

`\end{tabular}` – konec tabulky

`\hline` – vytvoří vodorovné ohraničení mezi řádky tabulky, uvádí se na první řádek pod ohraničením

`&` – přechod na novou buňku

`\ \` – přechod na nový řádek

8.2 Formátování sloupců

`r` – sloupec zarovnaný vpravo

`l` – sloupec zarovnaný vlevo

`c` – sloupec zarovnaný na střed

`p{4cm}` – sloupec s pevnou šířkou (zde 4 cm)

`|` – oddělí vedle sebe ležící sloupce čarou (přidá mezi ně ohraničení)

@{+++} – definice vlastního oddělovače sloupců (zde tři znaky +), lze ale zvolit libovolný jiný znak, nebo řetězec znaků

Příklad 7 - Zarovnání na desetinnou čárku (trik)

Alice	1,51
Bob	12,3

```
\begin{tabular}{lr@{,}l}
\hline Alice & 1 & 51 \\
Bob & 12 & 3 \\
\hline
\end{tabular}
```

Příklad 8 - Alternativní ohrazení (Booktabs)

D	P_u	u_u	β	G_f
[in]	[lbs]	[in]		[psi·in]
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

```
\begin{tabular}{ccccc}
\toprule[1pt] D & $P_u$ & $u_u$ & $\beta$ & $G_f$ \\
{[in]} & [lbs] & [in] & & [psi·in] \\
\midrule[1pt] 5 & 269.8 & 0.000674 & 1.79 & 0.04089 \\
\midrule 10 & 421.0 & 0.001035 & 3.59 & 0.04089 \\
\midrule 20 & 640.2 & 0.001565 & 7.18 & 0.04089 \\
\bottomrule[1pt]
\end{tabular}
```

Příklad 9 - Multirow, multicolumn

A	B	C	D	F					
1	0	6	230	35	40	55	25	40	35
	1	5	195	25	50	35	40	45	

```
\begin{tabular}{|c|c|c|c|p{1cm}p{1cm}p{1cm}p{1cm}p{1cm}p{1cm}|}
\hline
A & B & C & D & \multicolumn{7}{|c|}{F} \\
\hline
\multirow{2}{*}{1} & 0 & 6 & 230 & 35 & 40 & 55 & 25 & 40 & 35 \\
& 1 & 5 & 195 & 25 & 50 & 35 & 40 & 45 & \\
\hline
\end{tabular}
```

V případě příkazu `multicolumn{7}{c}{text}` první parametr určuje počet sloupců, které „multisloupec“ zabírá, druhý určuje zarovnání (včetně případného ohraničení) a třetí pak obsah.

U příkazu `\multirow{2}{*}{text}` první parametr určuje počet zabíraných řádek (řádek uvedení příkazu je první), druhý šířku (lze použít znak `*` - pak je šířka automatická) a poslední argument opět obsahuje text.

Za zmínku ještě stojí příkaz tzv. *dcolumn* – je to speciální typ sloupce, který umožňuje snadné zarovnávání podle určitého parametru – například konkrétního znaku. Při použití formulace `D{:}{:}{3.3}` dojde k zarovnání podle dvojtečky tak, že na každé straně bude místo na tři platné číselné pozice. První argument specifikuje, jaký je referenční znak pro zarovnání ve zdrojovém kódu, druhý určuje znak pro zarovnání ve výstupním PDF (je tedy možné použít jiný znak v kódu \LaTeX a jiný si nechat vypisovat do PDF), třetí pak počet platných číslic, které se mají vypsát. Je také možné specifikovat jiný počet číslic pro každou stranu zarovnání.

9 Matematika

Matematický zápis nám v \LaTeX umožňuje balíček AMSMath projektu $\mathcal{AMS}\text{-}\text{\LaTeX}$.

9.1 Základní pravidla

Matematický mód \LaTeX se řídí několika jednoduchými pravidly, která je při sázení dobré mít na paměti:

- mezery jsou ignorovány (libovolný počet mezer = 0 mezer)
- v sekci pro matematický zápis se nesmí objevit prázdný řádek
- výchozí styl písma je kurzívou
- každé písmeno \LaTeX považuje za samostatnou proměnnou
- v matematickém zápisu není možné užívat stejné příkazy, jako v běžném textu

9.2 Oddíly pro matematický zápis

Matematický zápis můžeme využívat ve dvou různých stylech. Jako součást odstavce souvislého textu – tzv. *in-line*, nebo jako samostatný blok zabírající celou šířku stránky – chová se jako *vlastní odstavec*.

Pro *in-line* zápis můžeme použít následující uvozující znaky:

- `\begin{math} ... \end{math}`
- `\(... \)`
- `$... $`

Pro zápis do *vlastního odstavce* můžeme zase použít následující uvozující znaky:

- `\begin{displaymath} ... \end{displaymath}`
- `\[... \]`
- `$$... $$`
- `\begin{equation} ... \end{equation}`

9.3 Znaky

mocnina/horní index $- x^2 - x^2$, resp. x^2

dolní index $- x_2 - x_2$, resp. x_2

odmocnina $- \sqrt{x}$, popř. $\sqrt[3]{x} - \sqrt[3]{x}$, popř. $\sqrt[3]{x}$

zlomek $- \frac{a}{b} - \frac{a}{b}$

suma $- \sum_{i=0}^n x^a - \sum_{i=0}^n x^a$ – zápis je v podstatě znak sumy s dolním a horním indexem

integrál $- \int_{i=0}^n x^a - \int_{i=0}^n x^a$ – obdobně jako v případě sumy se jedná o znak integrálu s dolním a horním indexem

produkt $- \prod_{i=0}^n x^a - \prod_{i=0}^n x^a$

funkce $- \sin(x) \cos(x) \tan(x) \max_{x \in M} \{x\} - \sin(x) \cos(x) \tan(x) \max_{x \in M} \{x\}$
– všechny základní funkce mají svůj vlastní příkaz

9.4 Mezery

Jak již bylo uvedeno, matematický zápis ignoruje všechny mezery. Pokud je tedy chceme do zápisu přidat, $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ nám poskytuje několik možností:

- \quad – mezera šířky písmene M
- \quad – 1/2 quadu
- \quad – 2 quady
- \quad – 3/18 quadu
- \quad – 4/18 quadu
- \quad – 5/18 quadu

9.5 Zápisy

Matematický zápis je v $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u na pokročilé úrovni. Často však uvnitř vlastního odstavce s rovnicemi chceme zapsat i prostý text. To nám umožňuje značka $\text{\texttt{\texttt{abc}}}$:

$$x^2 \geq 0 \text{ pro všechna } x$$

Problém může nastat při zápisu závorek. Ty by velikostně měly odpovídat velikosti obsahu. Z tohoto důvodu nemůžeme používat standardní znaky, ale místo nich speciální značky `\left(text a \right)` (analogicky pro hranaté závorky):

$$\left(\frac{a}{b}\right)^2$$

Často v matematice také používáme tři tečky jako výpustku. I ta má v `LaTeXu` svou vlastní značku. Je však třeba rozlišovat, zda potřebujeme tečky na úrovni základní linky textu (použijeme `\ldots`) nebo na středu řádku (použijeme `\cdots`). `LaTeX` také poskytuje do určité míry „chatré“ řešení v podobě značky `\dots`, ta by měla z kontextu (okolního textu) využít správný druh teček (ukázka v druhém řádku příkladu níže).

$$\begin{array}{l} i \in \{1, 2, \dots, n\} \quad x_1 + x_2 + \cdots + x_n \\ i \in \{1, 2, \dots, n\} \quad x_1 + x_2 + \cdots + x_n \end{array}$$

V neposlední řadě je dobré myslet také na zápis matic – ten je velmi podobný tabulkám a často se zde využívá v předešlé kapitole zmíněný `dcolumn` (kvůli zarovnání na desetinnou čárku). Příklad si můžete prohlédnout níže:

$$A = \begin{pmatrix} 1 & -24.5 \\ 3 & 5.22 \end{pmatrix}$$

```

$$
A=\left(
\begin{array}{@{}cD{.}{.}{.}{3.2}@{}}
1 & -24.5 \\
3 & 5.22
\end{array}\right)
$$
```

Další zajímavost se týká značky `\begin{equation}`, výsledná rovnice je totiž číslována a jde se na ni odkazovat pomocí značky `\label{xx}` a `\ref{xx}`, jak můžete vidět v rovnici 1:

$$\mathcal{X} = \frac{1}{n} \sum_{i=1}^r p_i x_i \quad \mathcal{S} = \sqrt{\frac{1}{n} \sum_{i=1}^r p_i (x_i - \mathcal{X})^2}, \quad (1)$$

```

\begin{equation}
\label{xx}
\mathcal{X}=\frac{1}{n}\sum_{i=1}^rp_{ix_i} \quad
\mathcal{S}=\sqrt{\frac{1}{n}\sum_{i=1}^rp_i(x_i-\mathcal{X})^2},
\end{equation}

```

Rovnice lze také spojovat do skupin (zde například skupina rovnic 2 obsahující rovnice 2a a 2b):

$$x = a + b \tag{2a}$$

$$y = c - d + 15 \tag{2b}$$

```

\begin{subequations}\label{eq:x}
\begin{align}
x &= a + b \label{eq:x1} \\
y &= c - d + 15 \label{eq:x2}
\end{align}
\end{subequations}

```

A také lze místo běžného číslování rovnice rovnou pojmenovávat názvy. Mějte ale na paměti, že názvy neslouží jako reference:

$$\lim_{n \rightarrow \infty} \sqrt[n]{\left(\frac{n}{n+1}\right)^{n^2}} = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^{-n} = \frac{1}{e} < 0 \tag{Cauchy}$$

```

\begin{gather}
\lim_{n \rightarrow \infty} \sqrt[n]{\left(\frac{n}{n+1}\right)^{n^2}} = \\
\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^{-n} = \\
\frac{1}{\mathrm{e}} < 0 \tag{Cauchy}
\end{gather}

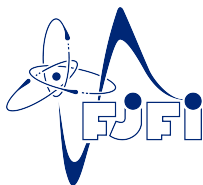
```

Často chceme také zapsat znak číselné množiny, jako jsou reálná čísla, přirozená čísla atd. K tomuto není nutný speciální příkaz, ale využití správného fontu z balíčku *AMSTeX*, který je k dispozici pouze v matematickém zápisu. Máme tedy reálná čísla – \mathbb{R} (`\mathbb{R}`), racionální čísla – \mathbb{Q} (`\mathbb{Q}`), celá čísla – \mathbb{Z} (`\mathbb{Z}`), přirozená čísla – \mathbb{N} (`\mathbb{N}`) a také komplexní čísla – \mathbb{C} (`\mathbb{C}`).

10 Obrázky

Vkládání obrázků je na první pohled složitější, než v případě např. Wordu. Jeho výsledek je také nejistý, protože o výsledném umístění obrázku nakonec rozhodne sám \LaTeX , a to bez vašeho vědomí.

Máme dvě možnosti, jak zaobalit obrázek pro vložení do dokumentu. Buď do značky `figure`, nebo `table`. Níže si ukážeme například první možnost:



Obrázek 1: Logo FJFI

```
\begin{figure}[!htp]
\begin{center}
\includegraphics[scale=0.1]{logo-fjfi.png}
\end{center}
\caption{Logo FJFI}
\label{fig:logo}
\end{figure}
```

U značky `figure`, první parametr určuje, kam se obrázek umístí. Možné hodnoty parametru jsou následující:

- `h` – here – umístění zde
- `t` – top – k hornímu okraji stránky
- `b` – bottom – k dolnímu okraji stránky
- `p` – page – samostatná stránka s obrázkem
- `!` – slouží k potlačení méně důležitých omezení pro umístění obrázku

Parametry lze samozřejmě libovolně kombinovat – \LaTeX se je bude snažit v uvedeném pořadí plnit. Často se jako parametr využívá řetězec `!htp`.

Další značky a parametry, které stojí za zmínku, jsou `\includegraphics[scale=0.1]{logo-fjfi.png}`, která se stará o samotné načtení obrázku. První (volitelný) parametr

udává násobící člen pro zvětšení/zmenšení obrázku, druhý (povinný) pak cestu k souboru s obrázkem.

Značka `caption` slouží jako viditelný popisec obrázku, přes `label` se lze na obrázek standardně odkazovat – viz. Obrázek 1.

Chcete-li si na začátku/konci dokumentu nechat vytvořit seznam všech obrázků, je k tomu možné využít příkazy `\listoffigures` nebo `\listoftables`, které jej pro vás automaticky vygenerují.

11 Reference, poznámky, odkazy

11.1 Reference

\LaTeX umožňuje snadné reference na stránky, nebo kapitoly. V případě *reference na kapitolu* stačí za příslušnou značku `section` doplnit `label{nazev}`, kde „nazev“ je zvolené jméno, přes které se pak budeme na referenci odvolávat. Kdekoliv v textu se pak můžeme odvolat na to, že v sekci 11.1 (`ref{reference}`) na stránce 17 (`pageref{reference}`) se něco nachází.

11.2 Poznámka pod čarou

Narozdíl například od Wordu se v \LaTeX u uvádí poznámka pod čarou přímo na místo, kde se nachází. Vždy se tak po vytvoření PDF bude nacházet na správné stránce¹.

Pro její vytvoření se využívá příkaz `footnote{text}`.

11.3 Použité zdroje

Použití zápisu využitých zdrojů ponecháme na čtenáři – omezíme se pouze na konstatování, že tuto funkcionalitu v \LaTeX u poskytuje balíček `BibTeX`.

Jednoduchou správu může poskytovat i \LaTeX sám, není ale příliš efektivní a výstupy nemusí splňovat podmínky pro formát uvádění zdrojů. Viz. příklad níže:

¹ Lorem ipsum dolor sit amet.

Reference

- [1] Leslie Lamport, *LaTeX: a document preparation system*, Addison Wesley, Massachusetts, 2nd edition, 1994.

```
\begin{thebibliography}{9}
\bibitem{lamport94}
Leslie Lamport,
\textit{LaTeX: a document preparation system},
Addison Wesley, Massachusetts,
2nd edition,
1994.
\end{thebibliography}
```

Je také možné se na jednotlivé zdroje odvolávat – [1, chapter, p. 215].

12 Vlastní příkazy

LaTeX umožňuje definování vlastních příkazů. Je tak možné doplnit určitou funkcionalitu, která není snadno dostupná, popřípadě zjednodušit složité zápisy, které se často opakují. Lze tak dosáhnout snazšího zápisu, který je mnohem lépe čitelný.

Pro definování vlastního příkazu se využívá příkaz `newcommand`, který píšeme do preambule dokumentu. Je specifikován takto:

```
newcommand{\R}[1]{\mathbb{R}_{#1}}
```

V prvních složených závorkách je uvedena značka vlastního příkazu, kterou se na něj budeme odvolávat v dokumentu. V hranatých závorkách je uveden počet argumentů, které lze k příkazu přidružit (maximálně 9). V posledních složených závorkách je pak řetězec, který říká, co značka dělá (v tomto případě vypíše v matematickém zápisu \mathbb{R} , jako značku množiny reálných čísel). V definici příkazu se lze na jednotlivé parametry odvolávat přes #, za kterou je uvedeno číslo označující pořadí parametru (indexováno je od 1).