



\LaTeX NIE JE FARBA NA MAĽOVANIE

BLAŠKO, Rudolf, (SK)

Abstrakt. Príspevkom chceme popularizovať používanie systému \LaTeX širokou verejnosťou, nielen vedeckou a odbornou obcou. Na začiatku porovnáme jeho výhody a nevýhody, a následne ponúkneme návod, ako môže aj úplný laik vytvárať v tomto systéme svoje jednoduché dokumenty.

1 Úvod

Poznáme mnoho programov, pomocou ktorých môžeme prezentovať v písanej alebo lepšie povedané v grafickej forme. Niektoré sú komerčné, niektoré nie. Niektoré sú viac pohodlné, niektoré menej. Niektoré sa dajú používať priamo bez akýchkoľvek znalostí, k niektorým sú potrebné aspoň elementárne poznatky a k niektorým je potrebné zložiť „maturitu“. Niektoré majú minimálne nároky na softvér a hardvér a niektoré naopak. A z niektorých sa normálny užívateľ s priemernými alebo podpriemernými počítačovými zručnosťami môže zblázniť. Pre jednoduchosť budeme všetky tieto produkty nazývať programy.

Medzi najrozšírenejšie programy medzi verejnosťou patrí známy nemenovaný komerčný produkt a jeho voľne šíriteľný ekvivalent OpenOffice.org. Ich veľkou výhodou je jednoduchosť ovládania – samozrejme pokiaľ máte verziu v jazyku, ktorému rozumiete. Bez akýchkoľvek znalostí môžete písať. Problém je iba v tom, že musíte používať veľmi často myš. Pre bežného užívateľa je to vcelku priateľné riešenie. A to aj napriek tomu, že sú tieto programy „svojhlavé“ a mnohokrát vnucujú užívateľovi svoju vôľu a prispôsobenie tohto počínania na svoj obraz stojí veľkú námahu. Problém je aj v tom, že prvý program funguje iba na platforme operačného systému Microsoft Windows, ktorá je spoplatnená. Kompatibilita medzi uvedenými programami je vcelku dobrá, aj keď nie bezproblémová. Ale ako už bolo spomenuté, pre bežného užívateľa pre normálne kancelárske používanie, pokiaľ sa nekladú vysoké nároky na výstupnú úpravu, sú tieto programy postačujúce a vcelku vhodné.

Alternatívou k uvedeným programom je používať úplne bezplatný systém $\text{T}_{\text{E}}\text{X}$, resp. jeho najpoužívanejší formát $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. Nevýhodou $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u je, že úplný laik nedokáže bez predchádzajúcich (síce minimálnych, ale predsa nejakých) znalostí tento program používať. Ďalšia vec, ktorá väčšinou odradí potenciálnych užívateľov, je to, že pisateľ hneď nevidí, čo píše. $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ funguje podobne ako klasické programovacie jazyky. Napíše sa zdrojový súbor *.tex (je to obyčajný textový súbor), ktorý sa preloží prekladačom do *.pdf alebo *.dvi súboru a ten sa následne vhodným prehliadačom zobrazí. V praxi to nie je také zložité – postačia na to dve systémové okná – v jednom okne je zdrojový textový súbor a v druhom výsledný grafický súbor. Jedným príkazom alebo jedným kliknutím myši sa automaticky *.dvi súbor preloží a hneď sa aktualizuje výsledný súbor a vidíme výsledok svojej práce. Existujú aj programy (napr. $\text{T}_{\text{E}}\text{X}$ Shell, WinShell, $\text{T}_{\text{E}}\text{X}$ Maker), ktoré tieto problémy riešia za užívateľa.

Existujú aj programy, ktoré fungujú tak, že človek hneď vidí svoje písanie, ale sú komerčné. Koniec koncov je to iba otázka zvyku a priemerne inteligentný človek s tým nemá žiadny problém. Výsledok stojí za to. Nezáleží na tom, v akom operačnom prostredí pôsobíme, aké kódovanie používame, koľko a akých jazykov naraz používame, aký prehliadač používame, akou tlačiarňou tlačíme výsledok a „výsledok je vždy rovnaký“ (samozrejme až na kvalitu a možnosti zobrazovačov, resp. tlačiarňí). To znamená, že už napísaný zdrojový súbor môžeme bez problémov modifikovať a beztrešne prekladať v ľubovoľnom inom operačnom systéme a „výsledok je vždy rovnaký“.

Dôležitá je tiež skutočnosť bezpečnosti a jednoduchej možnosti archivácie našej práce. V predtým spomínaných programoch nám jeden zmenený bit znehodnotí celý súbor. Pri $\text{T}_{\text{E}}\text{X}$ u, resp. $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u stačí archivovať zdrojové súbory, ktoré sú (vrátane všetkých podporovaných a konfiguračných) textovými súbormi, to znamená ľahko čitateľnými súbormi a menej náchylnými na poškodenie. Nie nepodstatná je tiež veľkosť archivovaných zdrojových dát. Zdrojové súbory mojej poslednej knihy, ktorá vyšla ako vysokoškolská učebnica v tlačenej forme a mala asi 200 strán popísaných matematickým textom, sa zmestili na klasickú disketu (dnes už nepoužívanú), a spolu mali 1.159.741 bytov. Výsledný súbor *.dvi mal 2.884 kB, súbor *.ps mal 3.874 kB a súbor *.pdf mal 1.721 kB. Súbor *.doc (ak zamlčíme skutočnosť, že uvedená kniha sa nedá napísať v tomto formáte v uspokojivom tvare) by obsadil podľa môjho odhadu asi 12.000 kB, t. j. desiatšesťkrát viac!

V Linuxe a jeho klonoch je $\text{T}_{\text{E}}\text{X}$ a $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ väčšinou súčasťou inštalácie, do ostatných operačných systémov ho treba doinštalovať. V súčasnej dobe to už nie je problém a inštalácia je pohodlná. Keďže prakticky každý národ (okrem Angličanov) má svoju špecifickú abecedu, sú aj rôzne národnostné modifikácie $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u. Pre češtinu a slovenčinu je to $\text{C}_{\text{S}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.

Nadšenci a užívatelia $\text{T}_{\text{E}}\text{X}$ u sú združení v medzinárodnej organizácii TUG „ $\text{T}_{\text{E}}\text{X}$ Users Group“ (Združenie užívateľov $\text{T}_{\text{E}}\text{X}$ u, <http://www.tug.org/>). V Čechách a na Slovensku je jeho súčasťou $\text{C}_{\text{S}}\text{TUG}$ „Československé združenie užívateľů $\text{T}_{\text{E}}\text{X}$ u“ (Czechoslovak $\text{T}_{\text{E}}\text{X}$ Users Group, <http://www.cstug.cz/>). Bezplatné inštalácie ($\text{T}_{\text{E}}\text{X}$ Live 2009, resp. $\text{T}_{\text{E}}\text{X}$ Live 2010 pretest) môže každý záujemca nájsť na <http://www.tug.org/texlive/>, resp. www.tug.org/texlive/pretest.html.

2 Začíname s L^AT_EXom

T_EX (vyslovujeme „tech“) je počítačový program vytvorený profesorom Donaldom Ervinom Knuthom. Je určený pre sadzbu textu a matematických rovníc pri zachovaní vysokej typografickej úrovne výsledného dokumentu. L^AT_EX (vyslovujeme „la-tech“, resp. „lej-tech“) je otvorený systém rôznych preddefinovaných maker, ktoré zjednodušujú a zlepšujú prácu pri vytváraní výsledného dokumentu a na sadzbu výsledného textu využíva T_EX.

Každý užívateľ (a nielen začiatočník) L^AT_EXu by mal mať vo svojej knižnici minimálne publikáciu *Ne príliš stručný úvod do systému L^AT_EX 2_ε* [4], ktorá je voľne prístupná (vrátane zdrojových súborov) na internete a Rybičkovu knihu *L^AT_EX pro začátečníky* [5].

2.1 Zdrojový text

Hlavnou a najdôležitejšou činnosťou pri vytváraní publikácie v programe L^AT_EX je zápis zdrojových textov, t. j. súborov *.tex. Tieto súbory môžeme písať v ľubovoľnom editore, uložiť ich ale musíme ako obyčajné textové ASCII súbory bez formátovacích znakov. To znamená, že sa do nich nesmú dostať riadiace príkazy daného editora. V tomto súbore sú okrem vlastného textu aj „príkazy“, pomocou ktorých L^AT_EX vysádza výsledný dokument.

Neviditeľné znaky (medzera `\` , tabulátor, koniec riadku) sú spracované L^AT_EXom ako medzera. Viac po sebe nasledujúcich neviditeľných znakov má význam ako jedna medzera.

Prázdny riadok medzi textom označuje koniec odstavca (za prázdny riadok začína nový odstavec). Viac prázdnych riadkov za sebou je spracovaných ako jeden prázdny riadok.

Špeciálne znaky `# $ % ^ _ { } ~ \` majú pre L^AT_EX zvláštny význam. Pre ich aplikáciu v texte použijeme príkazy `\# \$ \% \^ _ \{ \} \~ \`.

Príkazy v L^AT_EXe začínajú symbolom `\`. Za symbolom `\` nasleduje špeciálny, resp. nealfanumerický znak (napr. `_ \, \& \l \r \l`) alebo nasleduje meno príkazu zložené výlučne z písmen, pričom sa veľké a malé písmena rozlišujú (napr. `\bf \TeX \end{document}`).

Povinné parametre príkazov sa ohraničujú zátvorkami `{ }`. Ak má príkaz nepovinný parameter, ohraničuje sa zátvorkami `[]`. V texte majú zátvorky `[]` svoj normálny význam.

Zátvorky `{ }` majú špeciálny význam. Vo všeobecnosti ohraničujú parameter príkazu alebo ohraničujú skupinu. **Skupina** je úsek textu (vrátane príkazov), pre ktorý predpokladáme nejaké spoločné vlastnosti, resp. pre ktorý chceme nastaviť pôsobenie nejakého príkazu. Druhý spôsob pre ohraničenie skupiny je pomocou dvojice príkazov `\begin{meno}` a `\end{meno}`, kde meno je špeciálne slovo (napr. center, document, displaymath). Takto ohraničený úsek textu sa tiež nazýva **prostredie**.

Prostredia a skupiny sa môžu do seba ľubovoľne vnárať, ale nemôžu sa vzájomne krížiť.

Symbol percenta `%` má špeciálny význam pre poznámkovanie textu. Všetky znaky, ktoré stoja za symbolom `%`, sú až do konca riadku T_EXom pri preklade ignorované.

2.2 Štruktúra dokumentu

Vstupný súbor je zdrojový súbor, na ktorý sa \LaTeX nastaví a ktorý prekladá. Vo vstupnom súbore môžu byť odkazy na ďalšie zdrojové súbory, ktoré sa následne bez prerušenia preložia.

Keď \LaTeX spracováva vstupný súbor, očakáva jeho pevnú štruktúru v tvare:

```
\documentclass[vol'by]{trieda}
...preambula...
\begin{document}
...vlastný text...
\end{document}
```

`\documentclass[vol'by]{trieda}` je príkaz, ktorý \LaTeX očakáva na začiatku vstupného súboru. **Trieda** je povinný parameter a môže nadobúdať niektorú z hodnôt:

- `article` – pre články, krátke správy, listy, prezentácie, ... ,
- `report` – pre dlhšie správy, menšie knihy, seminárne a diplomové práce, ... ,
- `book` – pre skutočné knihy, veľké dokumenty, ...

Vol'by predstavujú nepovinný parameter, ktorý bližšie špecifikuje vlastnosti triedy dokumentu. Najpoužívanejšie z nich sú (ak ich je viac, sú oddelené čiarkou):

- `letterpaper`, `a4paper`, `a5paper`, `b5paper`, `legalpaper` – nastaví veľkosť stránky, ak nie je parameter spomenutý, t. j. implicitne je nastavené `letterpaper`,
- `10pt`, `11pt`, `12pt` – veľkosť písma dokumentu (implicitne je nastavené `10pt`),
- `oneside`, `twoside` – jednostranný výstup (implicitne pre `article` a `report`), resp. dvojstranný výstup (implicitne pre `book`),
- `twocolumn` – text sa bude sádzať do dvoch stĺpcov na stránku,
- `landscape` – výstup bude formátovaný na šírku, t. j. rozmery šírky a výšky sa vymenia.

Preambula je časť vstupného súboru, v ktorej sa nachádzajú tzv. globálne príkazy vzťahujúce sa k celému dokumentu (napr. šírka textu, hlavička, číslovanie stránok, ...). Môžu sa tu nachádzať naše vlastné príkazy, tzv. **makrá**. Tu sa k dokumentu pripájajú balíčky príkazov tzv. **packages** (súbory `*.sty`), v ktorých sa nachádzajú mnohé rozšírenia \LaTeX u (jazykové doplnky, farby, grafické a postskriptové doplnky, matematické a rôzne symboly, rozhodovacie makrá, indexovanie, obtekanie obrázkov, ...). Vloženie týchto balíčkov sa vykonáva opakovaným použitím príkazu `\usepackage[vol'by]{balíček}`, resp. `\usepackage{balíček1,balíček2,balíček3...}`. Parameter **vol'by** je nepovinný a bližšie špecifikuje daný balíček.

Pre ilustráciu uvedieme niektoré z týchto balíčkov:

- `czech`, `slovak` – umožní spracovať český, resp. slovenský text vrátane správneho delenia a názvov textových objektov v danom jazyku (Kapitola, Obrázok, ...),

- `amsmath`, `amssymb`, `amsfonts`, `amscd` – ďalšie príkazy pre matematickú sadzbu,
- `color` – podpora pre prácu s farbami,
- `ifthen` – podporuje rozhodovacie príkazy,
- `makeidx` – podpora pre tvorbu registra,
- `curves`, `bezier` – podpora pre kreslenie kriviek v prostredí `picture`,
- `array`, `hline` – podpora pre rozšírenie tabuliek,
- `inputenc` – špecifikácia kódovania pre zdrojové súbory,
- `fontenc` – špecifikácia kódovania fontov pre výstupné cieľové dokumenty.

Časť vstupného dokumentu medzi príkazmi `\begin{document}` a `\end{document}` predstavuje hlavnú časť zdrojového textu. Tu sa píše všetok text, ktorý sa má zobraziť vo výstupnom súbore. Tento text môže byť uložený aj v iných zdrojových súboroch a vtedy sa do vstupného dokumentu vkladá príkazmi `\include{meno}` (vloží text na novú stranu), resp. `\input{meno}` (vloží text na miesto, kde sa nachádza príkaz `input`).

Parameter **meno** môže obsahovať cestu k danému súboru, napr. `\input{makra/subor}`. Ak je prípona daného súboru `.tex`, potom príponu nemusíme písať. L^AT_EX prehľadáva najprv aktuálny adresár, kde sa nachádza vstupný zdrojový súbor a potom adresáre, ktoré má nastavené systém T_EX v parametri `path`.

2.3 Vlastný dokument

Teraz už môžeme písať vlastný dokument. Ak píšeme po anglicky a nemáme žiadne špeciálne požiadavky, potom nám postačí nasledujúca štruktúra vstupného súboru:

```
\documentclass{article}
\begin{document}
  English text ...
\end{document}
```

Ak sa vstupný súbor nazýva `subor.tex`, potom preklad do súborov `subor.dvi`, resp. `subor.pdf` dosiahneme príkazmi `latex subor.tex`, resp. `pdflatex subor.tex` (koncovka `.tex` je nepovinná, takže ju nemusíme písať).

Pre konverziu do postskriptového súboru `subor.ps` musíme použiť nejaký externý program (napr. `dvips`).

Ak vložíme daný text „English text ...“ do súboru `subor1.tex`, potom identický výsledok dostaneme pre:

```
\documentclass{article}      resp.      \documentclass{article}
\begin{document}            \begin{document}
  \input{subor1}             \include{subor1}
\end{document}              \end{document}
```

Súbor `subor1.tex` môžeme do dokumentu vložiť aj niekoľkokrát (nemusíme ho prepisovať). Nasledujúce dva príklady nie sú ekvivalentné:

```
\documentclass{article}      resp.      \documentclass{article}
\begin{document}             \begin{document}
\input{subor1}               \input{subor1}
\input{subor1}
\end{document}               \input{subor1}
                              \end{document}
```

V prvom prípade sa text z druhého vloženia pripojí priamo za prvý vstup, t. j. bude tam dvakrát. V druhom prípade je medzi vstupmi medzera, t. j. druhý vstup začne ako nový odstavec. Zato nasledujúce dva príklady sú ekvivalentné. V oboch prípadoch sa uvedený text zobrazí dvakrát a vždy na novej strane.

```
\documentclass{article}      resp.      \documentclass{article}
\begin{document}             \begin{document}
\include{subor1}            \include{subor1}
\include{subor1}
\end{document}              \include{subor1}
                              \end{document}
```

Ak píšeme po slovensky, resp. po česky musíme v preambule túto skutočnosť \LaTeX oznámiť (v opačnom prípade by preklad ignoroval znaky s diakritikou). Máme dve možnosti.

```
\documentclass[a4paper,12pt]{article}
\usepackage{slovak}
\usepackage[latin2]{inputenc}
\usepackage{amsmath,amssymb,amsfonts,amscd}
\begin{document}
Slovenský text v kódovaní ISO Latin-2 ...
\end{document}
```

V tomto prípade sú zdrojové súbory napísané v kódovaní ISO Latin-2 (kódová stránka ISO-8859-2). Ak vynecháme príkaz `\usepackage[latin2]{inputenc}`, potom kódovanie zdrojových súborov musí byť rovnaké ako implicitné kódovanie v systéme \TeX .

Preklad do výstupných súborov `subor.dvi`, resp. `subor.pdf` dosiahneme príkazmi `cslatex subor.tex`, resp. `pdfcslatex subor.tex`.

V súčasnosti sa do popredia dostáva tzv. „babelizovaný \LaTeX “, ktorý umožňuje v jednom zdrojovom súbore používať viacero jazykov.

```
\documentclass[a4paper,12pt]{article}
\usepackage{slovak}{babel}
\usepackage[IL2]{fontenc}
\usepackage[latin2]{inputenc}
\usepackage{amsmath,amssymb,amsfonts,amscd}
\begin{document}
Slovenský text v kódovaní ISO Latin-2 ...
\end{document}
```

Pre kódovania PC-latin2 (cp852), resp. win1250 (cp1250) má 4. riadok tvar:

```
\usepackage[cp852]{inputenc}    resp.    \usepackage[cp1250]{inputenc}
```

Pre kódovanie unicode UTF8 majú 3. a 4. riadok tvar:

```
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}    resp.    \usepackage[utf8x]{inputenc}
```

Pre preklad súborov v tomto prípade nemusíme použiť C_SL^AT_EX, postačí L^AT_EX. To znamená, že môžeme použiť príkazy `latex subor.tex`, resp. `pdflatex subor.tex`.

Na záver tejto časti uvedieme príklady súčasného použitia viacerých jazykov. Na prvý pohľad prepínanie medzi angličtinou a slovenčinou nemá význam, však pri zapnutej slovenčine dokážeme písať aj anglický text. Problém je v rozdeľovaní slov na konci riadku (každý jazyk má svoje špecifiká v rozdeľovaní slov) a v slovách, ktoré dopĺňajú text (napr. „Kapitola“, „Chapter“ alebo „Obrázok“, „Obrázek“, „Figure“, ...). Medzi jednotlivými jazykmi sa prepína príkazom `\selectlanguage{jazyk}`.

```
\documentclass[11pt]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[russian,slovak,english]{babel}

\begin{document}
\selectlanguage{english}
  English text ...

\selectlanguage{russian}
  Ruskij tekst ...

\selectlanguage{slovak}
  Slovenský text ...

\selectlanguage{english}
  English text ...

\selectlanguage{russian}
  Ruskij tekst ...

  ...

\end{document}
```

Príkaz `\selectlanguage{english}` za `\begin{document}` nemusíme písať, pretože angličtina sa nastaví ako implicitný jazyk (do T_EXu sa načíta `\language=0`).

Trochu iný je nasledujúci príklad. Ak by sme ponechali štruktúru dokumentu v nasledujúcom tvare (bez načítania ruštiny), T_EX by vyhlásil chybu, pretože ruštinu zatiaľ nepozná

(opäť sa načíta `\language=0`). Musíme mu to explicitne oznámiť príkazom `\selectlanguage{russian}`.

```
\documentclass[11pt]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[russian,slovak]{babel}

\begin{document}
  Ruskij tekst ...

\selectlanguage{slovak}
  Slovenský text ...

\end{document}
```

2.4 Vzhľad stránky

L^AT_EX má veľkú výhodu, že o vzhľad stránky a dokumentu sa užívateľ nemusí starať. Štýl stránkovania definuje vzhľad hlavičky a päty stránky a zadáva sa väčšinou v preambule príkazom `\pagestyle{štýl}` (ovplyvní celý dokument od miesta výskytu po koniec dokumentu, prípadne po jeho nový výskyt). Jeho analógia, ktorá ovplyvní iba aktuálnu stránku má tvar `\thispagestyle{štýl}`. Povinný parameter **štýl** môže mať tvar:

- `plain` – hlavička je prázdna, v päte je vycentrované číslo strany (implicitné nastavenie pre triedy `article` a `plain`),
- `empty` – hlavička a päta sú úplne prázdne, nie sú ani čísla strán,
- `headings` – hlavička obsahuje čísla strán a titulkové informácie (nadpisy kapitol a sekcií), päta je prázdna (implicitné nastavenie pre triedu `book`),
- `myheadings` – podobný tvar ako `headings`, ale titulkové informácie musíme nastaviť pomocou príkazov `\markright{pravé_záhlavie}` pre jednostrannú tlač a `\markboth{ľavé_záhlavie}{pravé_záhlavie}` pre obojstrannú tlač `twoside`.

Viac možností ponúka balíček `fancyhdr`.

Príkaz `\pagenumbering{štýl_číslovanie}` špecifikuje číslovanie strán, povinný parameter **štýl_číslovanie** môže mať tvar:

- `arabic` – čísloje sa arabskými číslami (implicitné nastavenie),
- `roman`, `Roman` – čísloje sa malými, resp. veľkými rímskymi číslami,
- `alph`, `Alph` – čísloje sa malými, resp. veľkými latinskými písmenami.

Štandardne sa čísloje od prvej hodnoty, pre zmenu číslovanie môžeme použiť príkaz `\setcounter{page}{nové_číslo_strany}`, resp. `\addtocounter{page}{zväčšenie_číslo_strany_o_túto_hodnotu}`.

2.5 Vzhľad odstavca

Odstavec je ohraničený na začiatku a na konci prázdny riadkom a o jeho tvar sa stará L^AT_EX. To znamená, že nezáleží na formáte ako napíšeme zdrojový kód odstavca¹ (nezáleží na počte medzislovných medzier, nezáleží na koľkých riadkoch je text napísaný, ...).

Na začiatku je odstavec odsadený o hodnotu `\parindent`, ktorú môžeme zmeniť príkazmi: `\setlength{\parindent}{nová_veľkosť}`, resp.

`\addtolength{\parindent}{zváčšenie_veľkosti_o_túto_hodnotu}`.

Príkaz `\setlength{\parindent}{10cm}` zmení hodnotu `\parindent` na 10 cm. Toto je nevhodné použitie, pretože keď zmeníme veľkosť písma dokumentu, odsadenie ostane 10 cm. Výhodnejšie je použiť relatívnu veľkosť vzťahujúcu sa k veľkosti písma dokumentu definovaného v príkaze `\documentclass`. Napr. `\setlength{\parindent}{5em}` zmení `\parindent` na 5-násobok šírky písmena „M“ a `\setlength{\parindent}{5.5ex}` zmení `\parindent` na 5,5-násobok výšky písmena „x“. V parametroch príkazov sa desatinné čísla používajú v anglickom tvare s desatinnou bodkou.

Riadkovanie je určené parametrom `\baselinestretch`, ktorý môžeme zmeniť príkazom `\renewcommand{\baselinestretch}{faktor}`, pričom **faktor** je ľubovoľné desatinné číslo. Napríklad hodnota 1.5 zväčší medziriadkové medzery o 50 %.

Pomocou príkazov `\hangafter=číslo`, `\hangindent=veľkosť_zmeny` môžeme meniť tvar odstavca. Tieto príkazy patria medzi základné príkazy systému T_EX, z ktorých sú zložené všetky ostatné a nazývajú sa **primitívy**. Vo formáte L^AT_EX (ako rozšírení T_EXu) tieto primitívy bez problémov fungujú a ich zoznam nájdeme napríklad na adrese <http://petr.olsak.net/ftp/olsak/bulletin/primitiv.ps>.

Ak je parameter **číslo** kladný (celé číslo), potom určuje počet riadkov plnej šírky, t. j. takýto počet prvých riadkov bude mať nezmenenú šírku. Ak je tento parameter záporný, potom určuje počet riadkov zmenenej šírky.

Ak je parameter **veľkosť zmeny** kladný (napr. 1.5cm, 2em), potom budú modifikované riadky krátené zľava o túto hodnotu. Ak je tento parameter záporný, potom krátenie nastane sprava. Ak je nulový, potom krátenie nenastane. Implicitné nastavenie je² `\hangafter=1 \hangindent=0pt`.

Po použití príkazov `\hangafter=5 \hangindent=5em` sa v odstavci zobrazia prvých 5 riadkov v normálnej šírke a ostatné sa skrátia zľava o hodnotu 5 em.

Po použití `\hangafter=-3 \hangindent=\parindent` sa v odstavci zobrazia prvé 3 riadky skrátené o hodnotu `\parindent` a ostatné sa zobrazia v normálnej šírke.

Ak použijeme na začiatku odstavca (musí byť prvým príkazom v odstavci) príkaz `\noindent`, potom sa odstavec neodsadí a začne sa sádzať od začiatku riadku. Naopak po použití príkazu `\indent` sa odstavec odsadí o hodnotu `\parindent`. Tento príkaz môžeme použiť v odstavcoch, ktoré sa automaticky neodsadia.

L^AT_EX implicitne neodsadí prvý odstavec na začiatku sekcie (nepomôže ani príkaz

¹ Pokiaľ ho neprerušíme prázdny riadkom alebo nepoužijeme formátovacie príkazy na jeho zmenu.

² 1 in [palec] = 2,54 cm = 25,4 mm = 72,27 pt [tlačiarenský bod]. Všetky tieto jednotky môžeme použiť.

`\indent`). Na odstránenie tohto nedostatku môžeme definovať vlastný príkaz, napríklad `\def\Prvyindent{\hangafter=-1 \hangindent=\parindent}`.

Ak použijeme predchádzajúce príkazy v preambule dokumentu, majú globálny význam.

2.6 Členenie textu

Pre lepšiu orientáciu v texte je vhodné ho rozčleniť na jednotlivé časti – na kapitoly, sekcie, paragrafy, ... \LaTeX obsahuje pre tento účel špeciálne príkazy, ktoré automaticky určia veľkosť a tvar fontu pre daný nadpis, automaticky ho očísľujú, zistia jeho polohu a následne ho zaradia do obsahu aj s jeho číslom. To textu sa zadávajú podľa nasledujúceho vzoru `\chapter[názov_do_obsahu]{názov}`. Ak chýba nepovinný parameter **názov_do_obsahu**, potom sa do obsahu dosadí celý **názov**.

Výsledkom použitia príkazu `\chapter` je na novej strane nadpis³ „Kapitola 1“ a pod tým **názov**. Jediný problém tohto príkazu je, že sa nedá použiť v triede `article`.

Príkaz `\chapter*{názov}` vypíše iba **názov** a to bez čísla a bez slova „Kapitola“.

Kapitoly môžeme potom deliť na ďalšie podčasti a to v nasledujúcom poradí:

<code>\section[názov_do_obsahu]{názov}</code>	resp.	<code>\section*{názov}</code>
<code>\subsection[názov_do_obsahu]{názov}</code>		<code>\subsection*{názov}</code>
<code>\subsubsection[názov_do_obsahu]{názov}</code>		<code>\subsubsection*{názov}</code>
<code>\paragraph[názov_do_obsahu]{názov}</code>		
<code>\subparagraph[názov_do_obsahu]{názov}</code>		

Príkazy `\section`, `\subsection`, `\subsubsection` sú číslované, kým `\paragraph` a `\subparagraph` nie sú. Po ich zadaní sa vypíše iba zodpovedajúce číslo časti a **názov**.

V texte si môžeme nejaké miesto označiť neviditeľnou značkou `\label{návestie}` a potom sa naň odvolávať. Príkazy `\ref{návestie}`, resp. `\pageref{návestie}` vypíšu číslo časti, resp. číslo stránky, kde sa `\label{návestie}` nachádza.

2.7 Vlastný text

V \LaTeX e treba rozlišovať medzi textom v obyčajnom a tzv. matematickom móde. Matematické časti textu sa vo vnútri odstavca uzatvárajú medzi znaky `$ a $`, resp. `\(a \)`.

Rozsiahlejšie matematické vzorce je výhodnejšie sádzať na špeciálne riadky. Pre tento prípad existujú matematické prostredia uzavreté medzi znaky `$$ a $$`, resp. `\[a \]`, resp. `\begin{displaymath}` a `\end{displaymath}`, resp. `\begin{equation}` a `\end{equation}`, resp. `\begin{eqnarray}` a `\end{eqnarray}`.

Teraz sa budeme zaoberať normálnym textovým módom. Aj keď mnohé poznatky, ktoré spomenieme sú aplikovateľné aj pre matematický mód.

Prácu nám môžu uľahčiť vlastné makrá, ktoré môžeme na ľubovoľnom mieste dokumentu definovať príkazmi `\newcommand{\meno}[číslo]{definícia_makra}`, resp.

`\def{\meno}parametre{definícia_makra}`.

³V anglickej mutácii sa napíše „Chapter 1“, t. j. číslo kapitoly.

Takto definované makrá môžu obsahovať niekoľko znakov, ale aj celé strany vrátane skupín a prostredí. Môžu obsahovať až 9 parametrov, ktoré do nich vstupujú.

Príkazom `\newcommand` musíme definovať makro, ktorého **meno** je nové⁴ (doteraz v dokumente alebo v systéme T_EX nepoužívané). Na premenovanie⁵ existujúceho makra použijeme `\renewcommand{\meno}[číslo]{definícia_makra}`. Číslo vyjadruje počet parametrov, ktoré môžu vstupovať do makra (maximálne 9). V časti **definícia_makra** sa tieto parametre vyjadrujú ako #1 (parameter1), ... #N (parameterN). Názov **meno** môže obsahovať iba písmena, pričom sa rozlišuje ich veľkosť. Takto vytvorené makro použijeme príkazom `\meno`, resp. `\meno{parameter1}...{parameterN}`.

Uvedieme niekoľko príkladov na použitie príkazu `\newcommand`:

```
\newcommand\MPMakro{Vidím ťa.}           \MPMakro
\newcommand>List[2]{#1 píše list #2.}   \List{Jano}{Ferovi}
\renewcommand\MPMakro[1]{Vidím #1.}    \MPMakro{Fera}
```

Príkaz `\def` je T_EXový primitív a funguje podobne ako `\newcommand`, ale ak makro s názvom **meno** existuje, potom ho bez opýtania prepíše. Navyše musíme všetky **parametre** vypísať aj s ohraničením do definície, okrem štandardného ohraničenia zátvorkami `{}`. Predchádzajúce príklady potom budú vyzeráť nasledovne:

```
\def\MPMakro{Vidím ťa.}                 \MPMakro
\def>List[#1]#2{#1 píše list #2.}       \List{Jano}{Ferovi}
\def\MPMakro#1{Vidím #1.}              \MPMakro{Fera}
```

Na poznámky písané pod čiarou slúži príkaz `\footnote{text_pod_čiarou}`.

Text zvýrazníme pomocou príkazu `\emph{zvýraznený_text}`. Ak je pôvodný text písaný kurzívou, potom sa **zvýraznený_text** prepne do kolmého módu a naopak z kolmého módu sa **zvýraznený_text** prepne do kurzívy.

Ak použijeme prepínač `\it`, potom do konca dokumentu, resp. do ďalšieho výskytu prepínača písma bude text vysádzaný kurzívou. Ak príkaz použijeme v tvare `{\it náš_text}`, resp. `\textit{náš_text}`, potom kurzívou bude vypísaný iba **náš_text**.

Analogicky fungujú príkazy `\bf`, `\textbf` (**polotučné písmo**), `\sl`, `\textsl` (*sklonené písmo*), `\sc`, `\textsc` (KAPITÁLKY), `\tt`, `\texttt` (strojopisné písmo), `\em`, `\textem` (vznačovacie písmo – analogické s `\emph`), `\rm`, `\textrm` (antikva – kolmé písmo) a `\sf`, `\textsf` (bezpätkové písmo).

Na zmenšenie alebo zväčšenie písma môžeme použiť nasledujúce príkazy. Pomer zmeny sa nemení a veľkosť všetkých písiem je odvodená od implicitného nastavenia veľkosti normálneho písma. Tiež sú prepínače (medzi rôznymi veľkosťami písma) a ich použitie je úplne rovnaké ako v predchádzajúcom odstavci a môžeme ich navzájom medzi sebou kombinovať: `\tiny` (najmenšie písmo), `\tiny\bf` alebo `\bf\tiny` (najmenšie polotučné písmo), `\scriptsize` (veľmi malé), `\footnotesize` (malé), `\small` (menšie), `\normalsize`

⁴ Ak makro s týmto názvom existuje, systém T_EX nám vynadá.

⁵ Ak makro s týmto názvom neexistuje, systém T_EX nám vynadá.

(normálne písmo), `\large` (väčšie), `\Large` (veľké), `\LARGE` (veľké), `\huge` (veľké) a `\Huge` (najväčšie písmo).

\TeX a \LaTeX majú veľmi veľký potenciál a dokážu toho oveľa oveľa viac. Ich možnosti ďaleko prekračujú kapacitu tohto článku. Sú to napríklad farebné možnosti (balíček `color`), obrázky (prostredie `picture`, balíček `graphicx`), tabuľky (prostredie `tabular`, balíček `longtable`), veľké množstvo rôznych matematických a aj nematematických symbolov, register pojmov, rôzne fonty písam, ľubovoľné zmenšenie či zväčšenie písma, prostredia na tvorenie zoznamov, možnosť vkladania rôznych externých súborov, atď. Na záver spomeniem ešte jeden odkaz <http://www.cheat-sheets.org/saved-copy/latexsheet.pdf>, kde sa nachádza na dvoch stranách stručný prehľad základných príkazov \LaTeX u.

Ďakujem všetkým nadšencom, ktorí sa venujú rozvíjaniu \TeX u. Táto práca vznikla na propagáciu \LaTeX u.

Literatúra

- [1] BALDA, M.: *Výpočty a diagramy v \LaTeX u*. Praha, Zpravodaj Československého sdružení uživatelů TeXu, č. 2, ročník 14, C₅TUG, 2004, str. 54–110, ISSN 1211-6661, http://bulletin.cstug.cz/pdf/bul_042.pdf.
- [2] KNUTH, D. E.: *The \TeX book*, Volume A of *Computers and Typesetting*, Addison-Wesley Publishing Company (1984), ISBN 0-201-13448-9.
- [3] KOPKA, H. – DALY, P. W.: *\LaTeX – Podrobný průvodce*, Brno, Computer Press, 2004, ISBN 80-722-6973-9.
- [4] OETIKER, T. – PARTL, H. – SCHLEGL, E. – HYNA, I. – KOČER, M. – SÝKORA, P.: *Ne příliš stručný úvod do systému $\LaTeX 2_{\epsilon}$* , „public domain“ dokument, 1998, <http://www.penguin.cz/~kocer/texty/lshort2e/lshort2e-cz.pdf>, resp. <http://ftp.cstug.cz/pub/tex/CTAN/info/lshort/slovak/lshorte.pdf> (Buša J. ml. a st., slovenský preklad).
- [5] RYBIČKA, J.: *\LaTeX pro začátečníky*, Brno, KONVOJ 2003, ISBN 80-7302-049-1.

Kontaktná adresa

RNDr. Rudolf BLAŠKO, PhD., Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská Republika, beerb@frcatel.fri.uniza.sk, <http://frcatel.fri.uniza.sk/~beerb/>.