

Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach

Žilina 6.-9. júla 2011

L^AT_EX NIE JE FARBA NA MAĽOVANIE, ALE NA PÍSANIE

BLAŠKO, Rudolf, (SK)

Abstrakt. Príspevok nadväzuje na článok „L^AT_EX nie je farba na maľovanie“, ktorý bol súčasťou tejto konferencie v minulom roku. Jeho cieľom je, aby každý čitateľ dokázal napísať svoj „vlastný dokument“ v L^AT_EX-u.

Kľúčové slová. T_EX, L^AT_EX, programovanie.

L^AT_EX IS NOT THE COLOR FOR PAINTING BUT FOR WRITING

Abstract. This contribution is the continuation of the paper „L^AT_EX is not the color for painting“, which was published in the previous year conference proceedings. The aim for the readers is to make them able to create the own documents in L^AT_EX.

Key words and phrases. T_EX, L^AT_EX, programming.

1 Úvod

Na úvod stručne zopakujeme hlavné zásady práce v programe L^AT_EX (L^AT_EX je typografický systém, ale kvôli prehľadnosti ho budeme označovať program). L^AT_EX má dve zdanlivo veľké nevýhody, ktoré odrádzajú väčšinu potenciálnych užívateľov, ale pre normálneho užívateľa nie sú problém. Prvú z nich predstavujú minimálne, ale predsa nejaké znalosti tohto programu. Druhou nevýhodou je, že sa dokument tvorí analogickým spôsobom ako program v klasickom programovacom jazyku. 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 na to postačia dve systémové okná (so zdrojovým textom a s výsledným grafickým súborom). Jedným príkazom alebo kliknutím myši sa zdrojový súbor automaticky preloží, aktualizuje sa výsledný súbor a hneď vidíme výsledok.

Táto zdanlivá nevýhoda je v skutočnosti výhoda. Nezáleží na tom, v akom operačnom prostredí pôsobíme, aké kódovanie používame, koľko a akých jazykov naraz používame, akú tlačiareň používame a „výsledok je vždy rovnaký“ (samozrejme až na možnosti zobrazovačov a tlačiarň). To znamená, že zdrojový súbor môžeme bez problémov modifikovať a beztriestne prekladať v ľubovoľnom inom operačnom systéme a „výsledok je vždy rovnaký“. Takýto stav absolútnej väčšine programov určených pre bežného užívateľa nehrozí. Jednoduchý text bez zložitých vzorcov si dokáže predstaviť každý. Čo si nedokáže predstaviť je jeho usporiadanie na stránke, ale to uvidí ihneď po preklade v druhom okne. S trochou praxe dokáže bez problémov každý priemerný počítačový užívateľ tvoriť aj zložitejšie vzorce.

L^AT_EX je určený predovšetkým na písanie vedeckých a technických dokumentov (z matematiky, fyziky, informatiky, chémie, biológie, . . .). Do textu sa dajú jednoducho vkladať súbory v rôznych formátoch (* . eps, * . ps, * . jpg, * . png, . . .). Dajú sa v ňom písať zložité notové zápisy (Mus_iX_TeX), ale samozrejme aj jednoduché texty.

Nesmieme zabúdať na najväčšiu výhodu L^AT_EX-u, ktorou sú nulové náklady na jeho zaobstaranie. To znamená, že to nie je komerčný produkt, ale produkt voľne šíriteľný, ktorý veľké množstvo nadšencov rozvíja a neustále zdokonaľuje.

Základnou činnosťou pri vytváraní publikácie v programe L^AT_EX je zápis zdrojových textov, t. j. súborov * . tex (majú pevnú štruktúru, zloženú z **preambuly** a **vlastného zdrojového textu** – vid' [1]). Pre naše účely budeme používať pracovný názov zdroj . tex . Súbor zdroj . tex môžeme písať v ľubovoľnom editore. Uložiť ho musíme ako obyčajný textový ASCII súbor bez formátovacích znakov editora, resp. operačného systému. Kvôli prehľadnosti a hlavne kvôli jednoduchým dodatočným opravám je výhodne vlastný zdrojový text štrukturovať do súborov podľa ucelených celkov (napr. kapitoly, podkapitoly). Tiež je vhodné každý dokument písať do nového zdrojového súboru. Myslím, že v L^AT_EX-u užívateľa ani nenapadne „zlepšovák“, ktorý sa často praktizuje: Písať všetky svoje dokumenty do jedného súboru a potom tlačiť iba potrebné strany.

Najmenšiu skupinu textu, ktorú L^AT_EX spracováva je odstavec. Pomocou dobre premyslených algoritmov upraví delenie slov, nastaví medzislovné medzery. Text ako celok spracováva na úrovni kapitoly, t. j. sa snaží sformátovať celý text tak, aby bola celá kapitola vyvážená (aby neboli medzi odstavcami veľké medzery, aby neboli strany prázdne, aby boli obrázky a tabuľky vhodne usporiadané, . . .). Inými slovami to znamená, že sa staráme iba o obsah textu a nie o jeho vizuálnu stránku.

Odskok textu na nový riadok zabezpečí príkaz `\newline` alebo príkaz `\\` (používa sa hlavne v tabuľkách). Odskok textu na novú stranu zabezpečí príkaz `\newpage` . Zvyšok strany ostane prázdny a text pokračuje na novej strane ako nový odstavec.

2 Začíname písať dokument

Na napísanie svojho prvého L^AT_EX–dokumentu už máme dosť informácií. Keďže sa všetko unifikuje a do popredia sa dostáva kódovanie Unicode¹ UTF-8, predpokladám, že v budúcnosti bude väčšina populácie používať toto kódovanie (nastavenia iných kódovaní vid' napr. [1], resp. [3, 5]). Súbor zdroj . tex pre jednostranný slovenský text, písaný v kódovaní Unicode UTF-8 na stránku formátu A4 s veľkosťou písma 12pt má tvar (ostatné nastavenia vrátane iných kódovaní vid' [1, 3, 5]):

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[slovak]{babel}
\include{mojeMakra.tex}
```

¹Vo všetkých distribúciách Linux-u a aj vo Windows 7 je implicitne nastavené toto kódovanie.

```

\begin{document}
  \include{listy/list01}
\end{document}

```

Náš text budeme písať do súboru `list01.tex`, ktorý je uložený v podadresári `listy` (aktuálneho adresára, t. j. adresára so súborom `zdroj.tex`). Vlastné makrá, môžeme ukladať do súboru `mojeMakra.tex` v aktuálnom adresári. Tento súbor (aj prázdny) musí existovať, v opačnom prípade hlási \LaTeX chybu.

3 Kapitoly a obsah

V \LaTeX -u sa obsah vytvára automaticky. Stačí v mieste textu, kde má obsah byť uložený, použiť príkaz `\tableofcontents`. V tomto prípade sa na nový riadok vysádza nadpis „Obsah“ (v anglickej mutácii „Contents“) a pod nim názvy jednotlivých častí s číslami strán.

Na dosiahnutie kompletného obsahu je nevyhnutné preložiť dokument `zdroj.tex` aspoň dvakrát. Pri prvom preklade sa vytvorí súbor `zdroj.toc`, do ktorého sa uložia potrebné informácie. Pri druhom preklade sa tieto informácie vložia do súboru `zdroj.dvi` (preklad pomocou `cslatex zdroj.tex` alebo `cslatex zdroj – koncovka.tex` nemusíme písať), resp. `zdroj.pdf` (preklad `pdfcslatex zdroj.tex`, resp. `pdfcslatex zdroj`).

Obsah zvykne byť v texte umiestnený na samostatnej strane, preto je rozumnejšie ho vložiť do súboru `obsah.tex` a na miesto výskytu napísať `\include{obsah}`. Súbor `obsah.tex` by mal okrem `\tableofcontents` obsahovať aj príkaz `\clearpage`, resp. `\cleardoublepage`. Príkaz² `\clearpage` zabezpečí prechod na novú stranu (zvyšok strany za posledným riadkom obsahu ostane prázdny). Pri obojstrannom dokumente (voľba `twoside` v príkaze `\documentclass`) sa použije analogický príkaz `\cleardoublepage`, ktorý zabezpečí prechod na nepárnu stranu.

Do obsahu sa automaticky pridávajú názvy jednotlivých častí (vrátane číslovania) definovaných príkazmi `\chapter` (trieda `book`), `\section`, `\subsection` `\subsubsection`. Tieto príkazy môžu mať nepovinný parameter [názov_do_obsahu], ktorý sa zobrazí v obsahu. Ak obsahujú hviezdičku `*`, potom sa nečísľujú a nezobrazujú v obsahu (viď [1]). Na ešte jemnejšie členenie textu existujú príkazy `\paragraph{názov}`, `\subparagraph{názov}`, ale tie sa nezobrazujú v obsahu.

Príkazom `\addcontentsline{toc}{úroveň}{text_do_obsahu}` môžeme do obsahu pridať ľubovoľný text (`toc` je koncovka súboru `zdroj.toc`, `úroveň` je jeden z parametrov `chapter`, `section`, `subsection`, resp. `subsubsection`). Na druhej strane príkazom `\addtocontents{toc}{text_do_obsahu}` tiež vložíme text do obsahu (môže byť formátovaný), ale bez čísla stránky. Uvedieme príklad použitia týchto príkazov a ich interpretáciu v obsahu:

```

\section{Číslovaná sekcia}\label{sek01}
  Text číslovanej sekcie \dots

```

²Ak použijeme tento príkaz v texte, potom sa okrem prechodu na novú stranu tiež vysádzajú všetky tzv. plávajúce objekty, t. j. obrázky a tabuľky.

```

\section[D-sekcia]{Druhá číslovaná sekcia}\label{sek02}
    Text druhej číslovanej sekcie \dots

\section*{Nečíslovaná sekcia}\label{sek03}
    Text nečíslovanej sekcie \dots

\subsection{Číslovaná subsekcia}\label{ssek01}
    Text číslovanej subsekcie \dots

\subsection[C-subsekcia]{Číslovaná subsubsekcia}\label{sssek01}
    Text číslovanej subsubsekcie \dots

\subsection*{Nečíslovaná subsubsekcia}\label{sssek02}
\addcontentsline{toc}{subsection}{N-subsubsekcia}
    Text nečíslovanej subsubsekcie \dots
\addtocontents{toc}{\dotfill toto je náš text \dotfill}

\paragraph{Náš paragraf}\label{p01}
\addcontentsline{toc}{subsection}{Náš paragraf}
    Text paragrafu \dots

\subparagraph{Subparagraf}\label{sp01}
    Text subparagrafu \dots

```

Preložený výsledný text je zobrazený v tabuľke 1 a obsah celého článku v tabuľke 2. V texte sú použité návestia `\label{návestie}`, na ktoré sa môžeme odkazovať (viď [1]).

Každá z úrovní textu má svoje číslovanie, reprezentované vlastným počítadlom, ktoré môžeme podľa potreby rôzne nastavovať a meniť. Počítať môžeme ľubovoľný objekt v texte, nemusí to byť iba kapitola alebo sekcia (viď [5]). \LaTeX má preddefinovaných niekoľko počítadiel, ktoré väčšinou korešpondujú s názvom počítaného príkazu alebo prostredia: `part` (príkaz `\part` má zvláštny význam a neovplyvňuje číslovanie ostatných úrovní), `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, `subparagraph` (evidujú čísla kapitol, sekcií, ...), `page` (číslo strany), `equation`, `figure`, `table` (čísla rovníc, obrázkov a tabuliek), `footnote` (číslo poznámok pod čiarou), `mpfootnote` (číslo poznámok pod čiarou v prostredí minipage), `enumi`, `enumii`, `enumiii`, `enumiv` (čísla rôznych úrovní v prostredí enumerate). Po každom použití sa hodnota daného počítadla zväčší o hodnotu 1, pri príkazoch s hviezdičkou `*` sa hodnota počítadla nemení.

Pre výpis aktuálnej hodnoty počítadla s názvom `meno_počítadla` sa používa príkaz `\themenom_počítadla`. Príkazom `\addtocounter{meno_počítadla}{n}` sa zväčší jeho hodnota o celé číslo n (resp. zmenší pre záporné n). Na hodnotu n nastavíme počítadlo `meno_počítadla` príkazom `\setcounter{meno_počítadla}{n}`.

Počítať objekty môžeme nielen arabskými číslami. Používajú sa na to nasledujúce príkazy `\arabic{meno_počítadla}` (arabské čísla – hodnoty môžu byť celé čísla), `\roman{meno_počítadla}`, `\Roman{meno_počítadla}` (malé a veľké rímske čísla – iba kladné celé čísla), `\alph{meno_počítadla}`, `\Alph{meno_počítadla}` (malé a veľké latinské písmená – iba prirodzené čísla, maximálna hodnota počítadla je 26).

Hodnotu počítadla môžeme nastaviť pre celý dokument v preambule alebo podľa potreby meniť na ľubovoľnom mieste dokumentu. Postupnosť príkazov

4 Číslovaná sekcia

Text číslovanej sekcie ...

5 Druhá číslovaná sekcia

Text druhej číslovanej sekcie ...

Nečíslovaná sekcia

Text nečíslovanej sekcie ...

5.1 Číslovaná subsekcia

Text číslovanej subsekcie ...

5.1.1 Číslovaná subsubsekcia

Text číslovanej subsubsekcie ...

Nečíslovaná subsubsekcia

Text nečíslovanej subsubsekcie ...

Paragraf Text paragrafu ...

Subparagraf Text subparagrafu ...

Tabuľka 1: Ukážka štrukturovaného textu

```
\setcounter{section}{-3}\thesection
\addtocounter{section}{36}\thesection
\renewcommand{\thesection}{\roman{section}}\thesection
\addtocounter{section}{-14}
\renewcommand{\thesection}{\Roman{section}}\thesection
\renewcommand{\thesection}{\alph{section}}\thesection
\renewcommand{\thesection}{\Alph{section}}\thesection
```

postupne nastaví hodnotu počítadla `section` na hodnoty: -3, 33, xxxiii, XIX, s, S.

Aby sme uzavreli túto časť nastavíme správne číslovanie sekcie na hodnotu 3 príkazmi `\setcounter{section}{3}` a `\renewcommand{\thesection}{\arabic{section}}`.

4 Úprava odstavca a výpočtové prostredia

\LaTeX implicitne zarovnáva text do odstavcov a delí slová (vid' [1, 3, 5]). Text môžeme na ľubovoľnom mieste rozdeliť na viacej stĺpcov a potom sa vrátiť k jednostĺpcovej sadzbe. Služi na to balíček `multicol`,

t. j. do preambuly musíme napísať príkaz `\usepackage{multicol}`.

Zápis textu do n stĺpcov dosiahneme pomocou prostredia `multicols` príkazmi (v našom prípade sú to 2 stĺpce, t. j. $n=2$):

Obsah	
1	Úvod 169
2	Začíname písať dokument 170
3	Kapitoly a obsah 171
4	Číslovaná sekcia 173
5	D-sekcia 173
5.1	Číslovaná subsekcia 173
5.1.1	C-subsubsekcia 173
	N-subsubsekcia 173
 toto je náš text
	Náš paragraf 173
4	Úprava odstavca a výpočtové prostredia 173
5	Tabuľky 177
6	Záver 180

Tabuľka 2: Obsah dokumentu

```
\begin{multicols}{n}
  Text ľubovoľného množstva textu sádzaný do n odstavcov \dots
\end{multicols}
```

Text (s viacerými odstavcami) môžeme bez problémov zarovnať na ľavý, resp. pravý okraj alebo centrovať na vertikálnu os (v tomto prípade sa nedelia slová). Takéto formátovanie odstavcov sa nazýva sadzba na zástavku (po česky sazba na praporek). Sádzbu na zástavku môžeme bez problémov kombinovať s viacstĺpcovou sadzbou.

Sádzbu na ľavý okraj zabezpečíme pomocou príkazu `\raggedright` alebo prostredia `flushleft`. Príkaz:

```
\raggedright{ľubovoľný text \dots}
je ekvivalentný príkazom:
\begin{flushleft}
  ľubovoľný text \dots
\end{flushleft}
```

Poznámky pod čiarou môžeme chápať ako špeciálny prípad odstavcov. Sú sádzané menším stupňom písma v dolnej časti danej strany „pod čiarou“ a zadávajú sa príka-

Sádzbu na pravý okraj zabezpečíme pomocou príkazu `\raggedleft` alebo prostredia `flushright`. Príkaz:

```
\raggedleft{ľubovoľný text \dots}
je ekvivalentný príkazom:
\begin{flushright}
  ľubovoľný text \dots
\end{flushright}
```

Centrovanú sadzbu zabezpečíme pomocou príkazu `\centering` alebo prostredia `center`. Príkaz:

```
\centering{ľubovoľný text \dots}
je ekvivalentný príkazom:
\begin{center}
  ľubovoľný text \dots
\end{center}
```

zom `\footnote[vlastné_číslo]{text_pod_čiarou}`, ktorý sa napíše za slovo, ktoré chceme označiť. Ak nie je zvolený nepovinný parameter `[vlastné_číslo]`, potom sa použije aktuálna hodnota počítadla `footnote`. V niektorých prostrediach (napr. `tabular`) príkaz `\footnote` nefunguje. Príkaz `\footnotemark[vlastné_číslo]` vytvorí v mieste uvedenia iba odkaz (nie poznámku pod čiarou) podľa rovnakých pravidiel ako `\footnote`. Príkaz `\footnotetext[vlastné_číslo]{text_pod_čiarou}` vytvorí poznámku pod čiarou s číslom identickým s posledným použitím príkazov `\footnote`, resp. `\footnotemark`, prípadne s hodnotou `[vlastné_číslo]`. Ak chceme dané miesto odkázať na už použitú poznámku pod čiarou (ľubovoľne v texte), môžeme použiť dvojicu príkazov `\label`, `\ref` (viď [1, 3, 5]). Uvedieme príklad použitia poznámok pod čiarou:

```
Náš text\footnote{Prvá poznámka
  pod čiarou.\label{p01}}
pokračuje A\footnotemark
\footnotetext{%
  Druhá poznámka pod čiarou.}
\footnotetext{%
  Tretia poznámka pod čiarou
  s rovnakým číslom ako druhá.}
a pokračuje B\footnote{77}{%
  Štvrtá poznámka pod čiarou
  s číslom 77.}
a pokračuje C\footnote{\ref{p01}}{%
  Piata poznámka pod čiarou
  s rovnakým číslom ako prvá.}
a pokračuje D\footnotetext{%
  Šiesta poznámka pod čiarou
  s rovnakým číslom ako druhá,
  ale bez značky.}
a pokračuje E\footnotemark{}
bez poznámky pod čiarou \dots
```

Vysádaná strana s poznámkami pod čiarou bude mať tvar (kvôli prehľadnosti je počítadlo `footnote` nastavené na začiatok):

Náš text¹ pokračuje A² a pokračuje B⁷⁷ a pokračuje C¹ a pokračuje D a pokračuje E³ bez poznámky pod čiarou . . .

-
- ¹Prvá poznámka pod čiarou.
²Druhá poznámka pod čiarou.
²Tretia poznámka pod čiarou s rovnakým číslom ako druhá.
⁷⁷Štvrtá poznámka pod čiarou s číslom 77.
¹Piata poznámka pod čiarou s rovnakým číslom ako prvá.
²Šiesta poznámka pod čiarou s rovnakým číslom ako druhá, ale bez značky.

Na zložitejšie členenie odstavcov formou číslovaných alebo nečíslovaných zarážiek môžeme použiť niekoľko prostredí. Môžeme ich do seba ľubovoľne vnárať do štvrtej úrovne. Pre označenie zarážiek sa vo všetkých z nich používa príkaz `\item[vlastná_značka]`.

Odsadenie zarážiek od ľavého okraja definujú registre `\leftmargini`, `\leftmarginii`, `\leftmarginiii`, `\leftmarginiv`. Príkazy `\renewcommand{\leftmargini}{.2cm}`, resp. `\leftmargini=.2cm`, resp. `\leftmargini 0.2cm` nastaví odsadenie prvej úrovne na 0,2 cm. Podobne `\leftmarginii 2.5\parindent` nastaví odsadenie druhej úrovne na 2,5-násobok `\parindent`. Zvislú medzeru medzi zarážkami meníme príkazom `\itemsep`, napr. `\itemsep 0pt` nastaví túto medzeru iba na medziriadkovú medzeru.

Prostredie `itemize` označuje zarážky postupne na jednotlivých úrovňach symbolmi

- (bullet), - (stredná pomlčka), * (hviezdička) a · (centrovaná bodka). Prostredie `enumerate` čísloje zarážky postupne na jednotlivých úrovňach arabskými číslami s bodkou (1. počítadlo `enumi`), malými latinskými písmenami v zátvorkách (a) počítadlo `enumii`), malými rímskymi číslami s bodkou (i. počítadlo `enumiii`) a veľkými latin-

skými písmenami s bodkou (A. počítadlo `enumiv`). Prostredie `description` predstavuje popisný výpis hesiel, ktoré sú definujú ako nepovinný parameter [vlastná_značka] príkazu `\item`. Tieto heslá sú automaticky na všetkých úrovniach sádzané tučným písmom.

Na označenie zarážiek môžeme použiť ľubovoľné symboly. Väčšina z nich je považovaná za matematické symboly a preto sa musí písať v matematickom móde, t. j. medzi dolármi `$` a `$`. Je ich veľmi veľké množstvo a čitateľ ich nájde napríklad v [2, 3, 4, 5].

```
Náš text rozdelíme na štyri úrovne,
aby sme videli odsadenia:
\begin{itemize}\itemsep 0.5em
\item Prvá úroveň \dots
\item[{$\heartsuit$}] Prvá úroveň \dots

\begin{itemize}\itemsep 0.25em
\item Druhá úroveň \dots
\item[{$\clubsuit$}] Druhá úroveň \dots

\begin{itemize}\itemsep 0em
\item Tretia úroveň a na
ukážku je tu dlhší text \dots
\item[{$\star$}] Tretia úroveň \dots

\begin{itemize}\itemsep 0em
\item Štvrtá úroveň \dots
\item[{$\pi$}] Štvrtá úroveň \dots
\end{itemize}
\end{itemize}
\item[{$\clubsuit$}] Druhá úroveň \dots
\item[{$\clubsuit$}] Druhá úroveň \dots
\end{itemize}
\item[{$\heartsuit$}] Prvá úroveň \dots
\end{itemize}
A text pokračuje \dots
```

```
\begin{description}
\item[Heslo 1] Text pre heslo~1 \dots
\begin{enumerate}
\item Prvý podtext \dots
\item Druhý podtext \dots
\item[7a] Tretí podtext \dots
\end{enumerate}

\item[Heslo 2] Text pre heslo~2 \dots
\renewcommand{\theenumi}{\alph{enumi}}
\begin{enumerate}
\item Prvý podtext \dots
\item[{$\ast$}] Druhý podtext \dots
\end{enumerate}

\item Text bez hesla \dots
\end{description}
```

Predchádzajúci text ilustruje použitie prostredia `itemize` do štvrtej úrovne. Vyšádzaný text vyzerá nasledovne:

```
Náš text rozdelíme na štyri úrovne, aby
sme videli odsadenia:

• Prvá úroveň ...
♡ Prvá úroveň ...

– Druhá úroveň ...
♣ Druhá úroveň ...

* Tretia úroveň a na ukážku
je tu dlhší text ...
* Tretia úroveň ...
  · Štvrtá úroveň ...
  π Štvrtá úroveň ...

♣ Druhá úroveň ...
♣ Druhá úroveň ...

♡ Prvá úroveň ...

A text pokračuje ...
```

Po preložení dostaneme výsledný text v nasledujúcom tvare:

```
Heslo 1 Text pre heslo 1 ...

1. Prvý podtext ...
2. Druhý podtext ...
7a Tretí podtext ...

Heslo 2 Text pre heslo 2 ...

a. Prvý podtext ...
* Druhý podtext ...

Text bez hesla ...
```


5 Tabuľky

Tabuľky sa na prvý pohľad robia zložito, ale nie je to pravda. Ak si užívateľ píše zdrojový text prehľadne (t. j. aby sa v ňom aj neskôr vyznal) a predvída potenciálne zmeny, potom sa tabuľky tvoria elegantne a pár príkazmi dokážeme meniť ich obsah a vzhľad.³ Pre sadzbu tabuliek sa používa prostredie `tabbing` (text môže presahovať na novú stranu), resp. `tabular` (text sa musí zmestiť na jednu stranu) a jeho matematická analógia `array`.

Prostredie `tabbing` umožňuje rozmiestniť text v riadku systémom podobným tabulátorom na písacích strojoch (viď múzeum), t. j. písať text do rovnakých pozícií v riadkoch. Tabulátory vo vnútri prostredia `tabbing` sa definujú príkazom `\=` a posun jednotlivých častí textu na pozíciu tabulátora zabezpečuje príkaz `\>`. Prechod na nový riadok sa zabezpečí príkazom `\\[výška]` (nepovinný parameter [výška] udáva veľkosť medzery medzi riadkami). Ak riadok končí príkazom `\kill`, potom sa riadok nezobrazí, nastavenie tabulátorov sa zachová a sadzba sa nastaví od začiatku riadku. Pri nastavení tabulátorov si musíme dať pozor, aby sa text zmestil do vymedzeného priestoru na šírku. Nasledujúci príklad ilustruje správne aj nesprávne použitie tabulátorov. Výsledok je zobrazený v tabuľke 3.

```
\begin{tabbing}
\= {\bf Tovar} \= {\bf Cena} \= {\bf Množstvo} \= {\bf Celkom} \\
\> Kolobežka \> 22,00 \euro \> 11 ks \> 242,00 \euro \\[-.2em]
\> Lopta \> 12,21 \euro \> 2 ks \> 24,42 \euro \\[.5em]

\= {\bf XXXXXXXXX} \= {\bf XXXXXXXX} \= {\bf XXXXXXXX} \= \kill
\> {\bf Tovar} \> {\bf Cena} \> {\bf Množstvo} \> {\bf Celkom} \\
\> Kolobežka \> 22,00 \euro \> 11 ks \> 242,00 \euro \\[-.2em]
\> Lopta \> 12,21 \euro \> 2 ks \> 24,42 \euro
\end{tabbing}
```

Jednotlivé prostredia `tabbing` sa nemôžu do seba vnárať. Toto prostredie poskytuje oveľa viac možností, ale ich vysvetlenie presahuje rozsah tohto článku (viď [3, 4, 5]).

Tovar	Cena	Množstvo	Celkom
Kolobežka	22,00 €	11 ks	242,00 €
Lopta	12,21 €	2 ks	24,42 €

Tabuľka 3: Prostredie `tabbing`

Na tvorbu tabuliek je vhodnejšie prostredie `tabular`. Toto prostredie má dve varianty:

```
\begin{tabular}[p]{formát_stĺpcov} \begin{tabular*}[p]{šírka}{formát_stĺpcov}
Vlastný text tabuľky \dots Vlastný text tabuľky \dots
\end{tabular} \end{tabular*}
```

³Na rozdiel od nemenovaných klikacích programov, v ktorých zmeniť veľkosť alebo štýl písma síce nie je veľký, ale veľmi pracný problém.

V modifikácii `tabular` sa šírka tabuľky určí podľa jednotlivých stĺpcov a v modifikácii `tabular*` sa šírka tabuľky definuje povinným parametrom⁴ `{šírka}` (napr. `{12em}`, `{5cm}`, resp. `{.8\linewidth}`), t. j. 80 % aktuálnej šírky riadku na strane). Nepovinný parameter `[p]` určuje spôsob pripojenia tabuľky k okolitému textu, môže nadobúdať hodnoty `t` (top – tabuľka sa k textu pripojí horným okrajom), `b` (bottom – tabuľka sa k textu pripojí dolným okrajom) a `c` (center – tabuľka sa k textu pripojí stredom – implicitné nastavenie).

V povinnom parametri `{formát_stĺpcov}` sa definuje spôsob zarovnávanía pre každý stĺpec tabuľky. Používajú sa na to symboly `l` (stĺpce zarovná k ľavému kraju), `r` (stĺpce zarovná k pravému kraju), `c` (stĺpce vycentruje), `p{šírka_stĺpca}` (text vypíše do bloku širokého `{šírka_stĺpca}`, širší text rozdelí na viac riadkov, pričom nedelí slová a zarovnáva na ľavú zástavku). Symbol `|` definuje zvislú čiaru medzi stĺpcami a `@{tvar_medzery}` definuje tvar medzery susediacich stĺpcov s týmto parametrom.

Ak použijeme `@{}`, potom medzi týmito stĺpcami nebude žiadna medzera, ale napríklad po použití `@{\quad=X=\quad}` sa v tabuľke v každom riadku medzi príslušnými stĺpcami objaví text `=X=` ohraničený širokými medzerami. Ak v tabuľke nasleduje za sebou niekoľko rovnako definovaných stĺpcov, potom môžeme použiť `*{počet}{formát_stĺpcov}`, kde číslo `{počet}` udáva koľkokrát sa má definícia `{formát_stĺpcov}` zopakovať (napr. `\begin{tabular}{|r||l|c|l|c|}` a `\begin{tabular}{|r||*{2}{l|c|}}` sú ekvivalentné príkazy). Príkaz `\setlength{\doublerulesep}{medzera}` zmení šírku medzery medzi dvojitémi vertikálnymi čiarami (`||`) na hodnotu `{medzera}`. Po nastavení `\setlength{\doublerulesep}{0pt}` sa táto medzera zruší a z dvojitej čiary sa stane tučnejšia čiara. Preddefinovaná vzdialenosť medzi vnútrom stĺpca a jeho okrajom je 6 pt a zmeniť ju môžeme predefinovaním registra `\tabcolsep` (napr. `\tabcolsep=3pt`).

Údaje sa zapisujú po riadkoch, pričom jednotlivé stĺpce sa oddeľujú symbolom `&` a prechod na nový riadok sa zabezpečí príkazom `\\[výška]` (viď str. 177). Horizontálna čiara cez celú šírku tabuľky medzi riadkami tabuľky sa definuje príkazom `\hline`. Príkaz `\cline{x-y}` definuje vodorovnú čiaru, ktorá začína pod stĺpcom s poradovým číslom `x` (počítané zľava) a končí pod stĺpcom s poradovým číslom `y`. Medzery medzi riadkami v tabuľke meníme predefinovaním príkazu `\arraystretch` (napr. `\arraystretch=1.5` ich zväčší na 1,5-násobok pôvodnej hodnoty).

Príkazom `\multicolumn{počet_stĺpcov}{formát_stĺpca}{text}` umiestnime text do viacerých stĺpcov (t. j. ich previažeme na šírku). Parameter `{počet_stĺpcov}` udáva koľko stĺpcov máme previazať, v parametri `{formát_stĺpca}` sa definuje formát tohto stĺpca a v parametri `{text}` sa nachádza vlastný text.

Balíček `array` ďalej rozširuje možnosti tvorby tabuliek na vertikálne prepojenia textu, nové stĺpce. . . Definuje sa v preambule príkazom `\usepackage{array}` (viď [2, 3, 4, 5]). Existujú aj balíčky (napr. `supertab`), ktoré riešia sadzbu „veľkých“ tabuliek presahujúcich niekoľko strán (prostredia `supertabular`, `longtable` – viď [2, 3, 5]).

⁴Celková šírka stĺpcov by mala korešpondovať so šírkou tabuľky.

Kedže sa nám mnohokrát nepodarí naformátovať text tak, aby sa tabuľky nelámali na konci strany, ukladáme ich (analogicky aj obrázky) ako tzv. **plávajúce objekty**. To znamená, že ich \LaTeX uloží na prvé vhodné miesto. Toto zabezpečuje plávajúce prostredie `table` (pre obrázky prostredie `figure`). \LaTeX zabezpečí ich automatické číslovanie a vhodné umiestnenie. Syntax pre vytvorenie plávajúcej tabuľky má tvar:

```
\begin{table}[p]\label{návestie}      resp. \begin{table}[p]\label{návestie}
      vlastná tabuľka                    \caption{popis_tabuľky}
\caption{popis_tabuľky}                vlastná tabuľka
\end{table}                             \end{table}
```

Voliteľný parameter `[p]` určuje miesto, kde sa má prednostne plávajúca tabuľka uložiť (pokiaľ je to možné). Tento parameter môže nadobúdať hodnoty (aj všetky naraz) `h` (here – prioritne na toto miesto), `t` (top – prioritne na hornú časť strany), `b` (bottom – prioritne na dolnú časť strany) a `p` (page – na samostatnú stranu, kde sú iba plávajúce objekty).

Príkaz `\caption[názov_do_obsahu]{popis_tabuľky}` definuje popis tabuľky pod alebo nad tabuľku (podľa jeho umiestnenia) a zabezpečuje číslovanie (počítadlo `table`). Popis má tvar `Tabuľka číslo: popis_tabuľky`. Ak tento príkaz chýba, potom sa tabuľka nečísluje. Ak je prázdny, t. j. `\caption{}`, potom sa vypíše iba `Tabuľka číslo: .` Obsah tabuliek, resp. obrázkov sa inicializuje (identicky ako normálny obsah) príkazmi `\listoftables`, resp. `\listoffigures`.

Do prostredia `table` môžeme vložiť aj externý obrázok (napr. `*.eps`, vid' [3,4,5]). Takto vložená tabuľka sa sádza od ľavého okraja strany, preto je vhodné ju centrovat'. Na záver ukážeme (v tomto prípade obsahovo nelogické) niektoré možnosti tabuliek. Dve plávajúce tabuľky vedľa seba umiestnime pomocou prostredia `minipage` (vid' tabuľky 4 a 5, 6).

```
\begin{tabular}{|||*{3}{l}|} \hline\hline
{\bf Tovar} & {\bf Cena} & {\bf Množstvo} & {\bf Celkom} \\\hline\hline
Kolobežka veľká & 22,00 \euro & 11 ks & 242,00 \euro \\\hline
Lopta & 12,21 \euro & 2 ks & 24,42 \euro \\\hline\hline
\end{tabular}
```

```
\begin{table}[ht]\begin{center} \begin{minipage}[t]{.48\linewidth}
  \setlength{\doublerulesep}{0pt}
\begin{tabular}[t]{||@{}p{5em}||*{3}{l}|} \hline\hline
{\bf Tovar} & {\bf Cena} & {\bf Množstvo} & {\bf Celkom} \\\hline\hline
Kolobežka veľká & 22,00 \euro & 11 ks & 242,00 \euro \\\hline
Lopta & 12,21 \euro & 2 ks & 24,42 \euro \\\hline\hline
\end{tabular}\caption{Prvá tabuľka}\end{minipage}
  \hfill
\begin{minipage}[t]{.48\linewidth} \caption{Druhá tabuľka}
  \setlength{\doublerulesep}{0.5pt}
\begin{tabular}[t]{||p{4em}||l|c|@{}r||} \hline\hline
{\bf Tovar} & {\bf Cena} & & {\bf Množstvo} & {\bf Celkom} \\\hline
\cline{1-2}\cline{4-4}
\multicolumn{2}{||c|}{Kolobežka veľká} & 11 ks & & 242,00 \euro \\\hline
Lopta & 12,21 \euro & & \multicolumn{1}{l}{2 ks} & \\
& 24,42 \euro & & & \\\hline\hline
\end{tabular}\end{minipage} \end{center}\end{table}
```

6 Záver

V zdrojových textoch sa používa symbol tilda `~`, ktorý zakazuje v danom mieste rozdelenie slov na nový riadok. V texte sa vyskytuje aj symbol pre euro (€). Keďže Eurozóna je podstatne mladší produkt ako \LaTeX , euro nie je jeho súčasťou. Tento problém rieši balíček `eurosym` a príkaz `\euro` (v preambule musíme deklarovat' `\usepackage{eurosym}`).

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

Tovar	Cena	Množstvo	Celkom
Kolobežka veľká	22,00 €	11 ks	242,00 €
Lopta	12,21 €	2 ks	24,42 €

Tabuľka 4: Prostredie tabular

Tovar	Cena	Množstvo	Celkom
Kolobežka veľká	22,00 €	11 ks	242,00 €
Lopta	12,21 €	2 ks	24,42 €

Tabuľka 5: Prvá tabuľka

Tabuľka 6: Druhá tabuľka

Tovar	Cena	Množstvo	Celkom
Kolobežka veľká		11 ks	242,00 €
Lopta	12,21 €	2 ks	24,42 €

Literatúra

- [1] BLAŠKO, R.: *\LaTeX nie je farba na maľovanie*, Otvorený softvér vo vzdelávaní, výskume a IT riešeniach, zborník medzinárodnej konferencie OSSConf 2010, Žilina, 1.–4. júla 2010, str. 43–52, ISBN 978-80-970457-0-8, <http://sospreskoly.org/files/OSSConf2010/ossconf2010-Blasko.pdf>; resp. Elektronický časopis openMagazin 09/2010, Liberix, str. 23–29, ISSN 1804-1426, <http://www.openmagazin.cz/pdf/2010/openMagazin-2010-09.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 príručka*, Brno, Computer Press, 2004, ISBN 80-722-6973-9.
- [4] OETIKER, T. – PARTL, H. – SCHLEGL, E. – HYNA, I. – KOČER, M. – SÝKORA, P.: *Ne príliš stručný úvod do systému $\LaTeX 2_{\epsilon}$* , „public domain“ dokument, 1998, <http://www.penguin.cz/~kocser/texty/lshort2e/lshort2e-cz.pdf>, resp. <http://ftp.cstug.cz/pub/tex/CTAN/info/lshort/slovak/Slshorte.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/>.