

Zpravodaj Československého sdružení uživatelů TeXu

Zdeněk Wagner
LaTeXová kuchařka/3

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 7 (1997), No. 3, 140–167

Persistent URL: <http://dml.cz/dmlcz/149795>

Terms of use:

© Československé sdružení uživatelů TeXu, 1997

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ*:
The Czech Digital Mathematics Library <http://dml.cz>

- [DVIPS] Tom Rokicki. *dvips — dvi to PostScript*. Manuál k programu `dvips` ve formátu `*.tex` přiložen v distribuci programu.
- [EC] Milan Zamazal. Balík pro počestění Emacsu `emacs-czech`, <http://www.fi.muni.cz/~pdm/emacs-czech.html>.
- [ENCTEX] Petr Olšák. Balík `encTeX` rozšiřující standardní `TeX` o možnost manipulace s `xord/xchr` vektory. <ftp://math.feld.cvut.cz/pub/olsak/enc tex/>. Petr Olšák: *EncTeX — změny konverzních tabulek TeXu*. Zpravodaj Československého sdružení uživatelů `TeXu`, **3** (7), 109–118 (1997).
- [TBN] Petr Olšák. *TeXbook naruby*. Vyjde v nakladatelství Konvoj. V elektronické podobě plný text k dispozici na <http://math.feld.cvut.cz/olsak/tbn/>.
- [TST] Petr Olšák. *Typografický systém TeX*. CSTUG 1995.

LaTeXová kuchařka/3

ZDENĚK WAGNER

Třetí díl LaTeXové kuchařky naváže na předchozí část. Naposledy se zastavíme u `maker` pro změnu velikosti písma. Poté nás čeká drobné intermezzo týkající se robustních a křehkých příkazů a pak už se můžeme věnovat vzhledu obsahu.

13. Změna řádkového prokladu

Na začátku této kapitoly mi připadá nemilá povinnost, a tou je oprava omylu z předchozí části [1]. Týká se definice makra `\@setfontsize`. První parametr je použit jinak, než bylo uvedeno. Slouží totiž k tomu, aby se aktuální velikost písma uložila do `\@currsize`. Definice vypadá přibližně takto:

```
\def\@setfontsize#1#2#3{%  
  \let\@currsize #1\fontsize{#2}{#3}\selectfont}
```

Prvním parametrem je tedy makro, v jehož definici se `\@setfontsize` vyskytuje. Další dva parametry specifikují velikost písma a vzdálenost účarí. Poslední dvě hodnoty budou uloženy do vnitřních maker `\f@size` a `\f@baselineskip`. Tyto uschované hodnoty lze s výhodou použít v případech, kdy chceme makrem `\fontsize` změnit pouze jednu z nich.

Zde vidíme další důvod, proč bychom neměli používat `\fontsize` přímo. Pak totiž `\@currsize` obsahuje nesprávný příkaz a balíky, které na toto makro spoléhají, mohou způsobit katastrofu.

Zatím jsme si ale neřekli, jak skutečně funguje nastavení vzdálenosti účaří, potažmo řádkového prokladu. Musíme se podívat, jak \TeX láme odstavce na řádky. Každý rádek je uložen do boxu a příslušné boxy jsou skládány pod sebe tak, aby vzdálenost účaří byla rovna hodnotě registru `\baselineskip`. Přitom se kontroluje vzdálenost horního okraje právě vkládaného boxu od spodního okraje předchozího boxu. Ta nesmí být menší než `\lineskiplimit`. Pokud by tato podmínka nebyla splněna, vložil by \TeX mezi boxy vertikální mezeru velikosti `\lineskip`. Hodnoty `\lineskip` i `\lineskiplimit` můžeme předefinovat. Navíc můžeme změnit hloubku předchozího boxu, kterou \TeX uložil do registru `\prevdepth`.

Příkaz pro změnu řádkového prokladu se nabízí sám. Stačí změnit hodnotu `\baselineskip`. Před takovým přístupem je ale nutno důrazně varovat! Registr `\baselineskip` je pro uživatele \LaTeX u tabu. Je pravda, že v jednoduchých dokumentech to funguje podle očekávání, ale často se dočkáte nemilých překvapení. Může za to právě makro `\fontsize`, které mění kromě velikosti písma i vzdálenost účaří. Stačí tedy jakákoliv změna velikosti písma a vaše nastavení `\baselineskip` je ztraceno. Lze namítat, že změny velikosti provádíme obvykle lokálně uvnitř skupiny, takže po jejím uzavření se `\baselineskip` obnoví. To je sice pravda, ale nesmíme zapomínat, že řada standardních maker v sobě obsahuje například `\normalsize`, a tím o své nastavení registru `\baselineskip` přijdeme.

V \LaTeX u máme tři metody, jak změnit řádkový proklad. První z nich jsme si ukázali již v předchozí části kuchařky. Stačí předefinovat `\normalsize`, `\small`, `\large` a ostatní makra pro změnu velikosti písma. V odpovídajících příkazech `\fontsize` uvedeme požadovanou vzdálenost účaří. Tuto metodu použijeme, pokud píšeme vlastní styl pro dokument určitého typu. Tak je to nastaveno i ve stylu pro sazbu tohoto Zpravodaje. Nevýhodou takového postupu je skutečnost, že řádkový proklad je definován kdesi uvnitř chráněných maker a nelze jej tudíž snadno ovlivňovat na uživatelské úrovni.

Autor se často dostává do situace, že by potřeboval v první fázi tvorby rukopisu zvětšit řádkový proklad, aby si mezi řádky mohl vpisovat poznámky. I na to \LaTeX pamatuje. Musíme si ovšem uvědomit, jak funguje makro `\fontsize`. Když opomeneme příkazy pro uschování aktuálních hodnot, můžeme jeho definici zjednodušeně zapsat:

```
\def\fontsize#1#2{%
  \set@fontsize\baselinestretch{#1}{#2}}
```

Změna velikosti je makrem `\set@fontsize` pouze zaznamenána, ale provede ji až `\size@update` při provádění příkazu `\selectfont`. Za normálních okolností nedělá `\size@update` nic. Makro `\set@fontsize` uschová své tři parametry do `\f@linespread`, `\f@size` a `\f@baselineskip`. Hodnota `\f@linespread` je pak vrácena zpět do `\baselinestretch`. Potom je přede-

finováno makro `\size@update` tak, aby se vzdálenost účaří nastavila na součin `\f@linespread` a `\f@baselineskip`. Vlastní činnost je pak provedena během expanze makra `\selectfont`. Změny řádkování tedy docílíme předefinováním `\baselinestretch`, například:

```
\renewcommand\baselinestretch{1.2}
```

\LaTeX 2_ε nabízí navíc makro `\linespread`, které je definováno (přibližně):

```
\def\linespread#1{\set@fontsize{#1}\f@size\f@baselineskip}
```

Stejného výsledku jako výše dosáhneme nyní velmi jednoduchým повеlem `\linespread{1.2}`.

Zde je nutno připojit komentář pro ty, kdo ještě používají zastaralý \LaTeX 2.09. Jak patrně, nová hodnota `\baselinestretch` se projeví až po změně velikosti písma. Bohužel pouhý `\normalsize` k tomu nestačí. Toto makro totiž bylo optimalizováno a ignorovalo se, pokud již byla zvolena základní velikost. Uživatel tedy musel napřed zmenšit velikost například na `\small` a potom vrátit na `\normalsize`. Pro \LaTeX 2_ε stačí použít `\selectfont`. Toto makro automaticky provede změnu velikosti písma i v tom případě, že současná hodnota `\baselinestretch` neodpovídá uschované `\f@linespread`.

Třetí metodou změny řádkového prokladu je přímé použití příkazu `\fontsize`. To má opodstatnění v krátkých tiskovinách, jako jsou třeba vizitky, na titulních stránkách, nebo jako součást maker, která mění řádkový proklad pouze lokálně v malé části dokumentu. Nejbližší `\normalsize` totiž vrátí `\baselineskip` zpět na standardní hodnotu.

14. Křehká a robustní makra

Tato odbočka je nesmírně důležitá pro pochopení následujícího textu. Často se osobně i v diskusním listu setkávám s dotazy, proč \LaTeX hlásí chybu v `\caption`, ale přitom je vše vytištěno správně. Zřejmě již tušíte, že je to způsobeno použitím křehkého makra. Jak ale předem poznáme křehké makro od robustního? Odpověď není těžká, ale zatím ji odložíme.

Začneme trochu zešíroka. Víme, že \TeX čte vstupní proud a předává jej k dalšímu zpracování. Pokud při čtení objeví nějaký token, který se dá expandovat, provede expanzi a výsledek vloží zpět do vstupního proudu. Ten může být předmětem další expanze. Jsou však situace, kdy k expanzi nedojde. To je případ již zmíněného makra `\caption`. \LaTeX jej totiž definoval jako makro s parametry. Při jeho expanzi tedy \TeX nejprve parametry hledá. Předpokládejme, že nepovinný parametr nebyl uveden, takže \TeX vezme text mezi složenými závorkami a uschová jej jako parametr makra. Tento text je pak vysázen a přitom dojde k plné expanzi všech použitých maker. To nečiní žádné potíže. Problém však nastává, když se parametr zapisuje do pomocného souboru, z něhož se při dalším průchodu vytvoří seznam obrázků či seznam tabulek. Zápis provádí primitiv

`\write` a ten také expanduje veškerá makra v zapisovaném textu. Některá makra však v této situaci expandovat nelze, a právě to způsobí chybu. Pokud netisknete seznam obrázků ani seznam tabulek, chyba se v dokumentu neprojeví. Robustní makro je ochráněno před takovou předčasnou expanzí. Do pomocného souboru se zapíše bez expanze a ke skutečné expanzi dojde až v okamžiku tisku.

Křehké makro musíme ochránit použitím `\protect`. Toto ochranné makro má proměnný význam podle situace, v níž se vyskytuje. Dokumentovaný zdrojový kód \LaTeX u uvádí pět možností, které mohou nastat. Makro `\protect` může být definováno jako:

- `\relax` – Toto se používá při normální sazbě. Potom `\protect\foo` provede `\foo`. Dochází k plné expanzi všech maker. Uživatel nesmí `\protect` pro tuto základní situaci změnit. Řada vnitřních maker na tuto definici spoléhá. Pokud potřebujete jinou definici makra `\protect` pro běžnou sazbu, musíte stejným způsobem předdefinovat i `\@typeset@protect`. Nejvhodnější je předdefinovat toto pomocné makro a pro definování makra `\protect` použít `\set@typeset@protect`.
- `\string` – Používá se pro výpis na obrazovku. Makra `\protect\foo` pak vypíší `\foo`. Tuto hodnotu lze do makra `\protect` vložit příkazem `\set@display@protect`.
- `\noexpand` – To je vhodné pro zápis do souboru, který se bude dále zpracovávat (\LaTeX) em. Zde potřebujeme, aby za makrem byla vynechána mezer. Sekvence `\protect\foo` nyní skutečně zapíše `\foo` s mezerou.
- `\@unexpandable@protect` – Občas potřebujeme do souboru zapsat makro tak, aby bylo robustní a sneslo další zacházení. Výsledkem zápisu `\protect\foo` musí být opět `\protect\foo` následované mezerou. Již jsme uvedli, že část úkolu splní `\noexpand\foo`. Stejným způsobem potřebujeme předem zapsat `\protect`, takže kompletní sekvence příkazů bude `\noexpand\protect\noexpand\foo`. Z úsporných důvodů definuje \LaTeX zkratku:

```
\def\@unexpandable@protect{\noexpand\protect\noexpand}
```

Makro se používá i při definicích prostřednictvím `\edef` a `\xdef`. Pro tyto účely máme další zkratky, `\protected@edef` a `\protected@xdef`. Pokud po `\xdef` nepotřebujeme obnovit původní `\protect` (protože definici provádíme uvnitř skupiny a obnovu tudíž zařídí \TeX), můžeme použít `\unrestored@protected@xdef`. Definice najdete v souboru `ltxdefns.dtx`. Zde je drobná ukáзка:

```
def\protected@edef{%
  \let\@@protect\protect
  \let\protect\@unexpandable@protect
  \afterassignment\restore@protect
  \edef
}
\def\restore@protect{\let\protect\@@protect}
```

- `\@unexpandable@noexpand` – Původně bylo určeno pro odložený zápis uvnitř `\edef`. Mělo způsobit zapsání `\foo` a mezery do souboru. Definice tohoto makra byla:

```
\def\@unexpandable@noexpand{\noexpand\noexpand\noexpand}
```

Všimněte si minulého času. Toto makro totiž nebylo nikdy využito, takže již v \LaTeX u definováno není.

Řekli jsme si, že `\protect` chrání křehká makra před expanzí. Víme také, že \TeX v jistých situacích expanduje vše až na primitivy. Z tohoto hlediska jsou tedy všechna definovaná makra křehká. Některá makra jsou definována tak, že je lze bezpečně expandovat ve všech režimech, ale jsou případy, kdy si to dovolit nemůžeme. Předpokládejme, že `\Macro` patří do této kategorie. Zkusme nyní přidat následující definici:

```
\def\MyMacro{\protect\Macro}
```

Snadno se přesvědčíme, že `\MyMacro` bude v běžném textu dělat totéž co `\Macro` – až na drobné výjimky¹. Navíc bude nové makro fungovat i v pohyblivých argumentech. Jeho expanzí totiž vznikne `\protect\Macro`. Naše `\MyMacro` je tedy robustní.

Při definici robustních maker musíme být pečliví. V jednom svém dokumentu jsem často potřeboval psát název programu – „Gepard“. Abych nemusel psát backslash nebo prázdné složené závorky za makrem, vytvořil jsem si `\g+`. Makro se mělo vyskytovat i v názvech kapitol a z nich se mělo přenést do obsahu, muselo tedy být robustní. Zcela mechanicky jsem pak zapsal:

```
\def\g+{\protect\pg}
```

```
\def\pg{Gepard}
```

V běžném textu vše fungovalo správně. Mezera po `\g+` nebyla spolknuta. Do pomocného souboru se však zapsalo `\protect\pg` následované mezerou. Při následujícím zpracování ale \TeX tuto mezeru spolkne, takže za slovem „Gepard“ mezera chyběla. Definice měla správně vypadat:

```
\def\g{\protect\pg}
```

```
\def\pg+{Gepard}
```

Tak se definují robustní makra. Nevýhodou tohoto postupu je nutnost definice dvou maker. Řekneme uživateli, že má pro určitou činnost použít `\MyMacro` a že je toto makro robustní. Nyní si uživatel nadefinuje `\Macro` a dojde ke katastrofě. Musíme tedy ještě informovat, jaké makro uživatel předefinovat nesmí. Tuto nevýhodu obejdeme použitím definičního příkazu `\DeclareRobustCommand`. Používá se naprosto stejně jako `\newcommand`. Při definici robustního makra se vytváří vnitřní pomocné makro s tak obskurním jménem, že se jej nikomu nepodaří předefinovat – kromě největších čarodějů, ale ti vědí, co dělají.

¹Pokud expandované `\Macro` začíná číslicí, pak `1\Macro` bude mít jiný účinek než `1\MyMacro`.

Jak tedy poznáme robustní makro? Požádáme T_EX, aby vypsal jeho definici. Zkusme třeba:

```
\show\MyMacro
```

Na obrazovku i do log souboru se vypíše:

```
> \MyMacro=macro:  
->\protect \Macro .
```

Podobný výpis dostanete pro každé robustní makro.

15. Vytváření obsahu

Řekli bychom, že vytvoření obsahu v L^AT_EXovém dokumentu je triviální záležitostí. Stačí přece napsat `\tableofcontents`. Toto tvrzení je sice pravdivé, ale pouze částečně. Uvedeným příkazem totiž vysázíme obsah v předpřipraveném formátu. Často však potřebujeme vzhled obsahu změnit. V následujícím textu si povíme, jak lze nejrůznějších změn snadno dosáhnout.

15.1. Jak L^AT_EX vytváří obsah

Chceme-li zasahovat do tvorby obsahu, musíme nejprve něco vědět o mechanismu, který standardní L^AT_EXové styly pro sazbu obsahu používají. Makra `\part`, `\chapter`, `\section` a další ukládají název části či kapitoly včetně čísla stránky do pomocného souboru s příponou `.toc`. Nebudeme popisovat podrobně mechanismus, uvedeme pouze, že tento pomocný soubor obsahuje pro každou kapitolu makro `\contentsline` s příslušnými parametry.

Makro `\contentsline` má tři parametry. Prvním parametrem je typ záznamu, tedy například `chapter` nebo `section`, druhým parametrem je vlastní text příslušného názvu a třetím parametrem je číslo stránky. Číslo kapitoly je uváděno v argumentu makra `\numberline`. Ve Zpravodaji č. 3/1996 obsahuje pomocný soubor mimo jiné:

```
\contentsline {section}{\numberline {A}Úvod}{134}  
\contentsline {section}{\numberline {B}Původ}{135}  
\contentsline {subsection}{\numberline {1}Co je to \TeX {}}{135}  
\contentsline {subsection}%%  
  {\numberline {2}Jak mám vyslovovat \uv {\TeX }}{136}  
\contentsline {subsection}{\numberline {3}Co to je \MF {}}{136}  
\contentsline {section}%%  
  {\numberline {C}Dokumentace a\nobreakspace {}nápověda}{145}  
\contentsline {subsection}%%
```

```

{\numberline {18}Knihy o \TeX {}u a příbuzná literatura}{145}
\contentsline {subsection}%%
{\numberline {19}Kde lze najít tento dokument}{146}

```

Některé řádky jsou příliš dlouhé, proto musely být ručně zlomeny, aby se vešly do tiskového zrcadla. Místo ručního zlomu je označeno třemi procenty a odsazením druhé části zlomeného řádku.

Makro `\tableofcontents` vysází nadpis obsahu, obvykle příkazem:

```
\chapter*{\contentsname}
```

Ve stylu `ARTICLE` se pochopitelně použije `\section*`. Makro `\contentsname` je definováno ve všech jazykových stylech tak, aby fungovalo v angličtině, češtině, francouzštině, jakož i v jiných jazycích. Po vysazení nadpisu se zavolá makro `\@starttoc`, které načte pomocný soubor a otevře jej znovu pro zápis. Soubor `article.cls` obsahuje následující definici:

```

\newcommand\tableofcontents{%
  \section*{\contentsname
    \@mkboth{%
      \MakeUppercase\contentsname}{\MakeUppercase\contentsname}}%
  \@starttoc{toc}%
}

```

Makro `\@mkboth` navíc vytvoří živé záhlaví. Pokud nevíte, proč je použito makro `\MakeUppercase` místo primitivu `\uppercase`, přečtete si článek Petra Olšáka: Putování písmene ř z klávesnice na papír [3].

Makro `\@starttoc` má jeden parametr, a tím je přípona pomocného souboru. Pak tedy `\@starttoc{toc}` vysází obsah, `\@starttoc{lof}` je voláno makrem `\listoffigures` a vytiskne seznam obrázků s použitím údajů z pomocného souboru s příponou `.lof`. Z pomocného souboru se načítají příkazy `\contentsline{typ}`, které se expandují na `\l@typ`. Pokud se nám tedy nelíbí, jak se v obsahu tiskne název kapitoly, musíme předefinovat `\l@chapter`.

\LaTeX zapisuje do obsahu pouze kapitoly a podkapitoly do určité úrovně. Představte si, že chcete počet úrovní v obsahu rozšířit. Pohledem do pomocného souboru zjistíte, že příslušné údaje v něm jsou, ale v obsahu se neobjeví. \LaTeX má totiž určitý mechanismus, jímž se určuje, jaká nejhlubší úroveň nadpisů se v obsahu vytiskne. Přehled úrovní najdete v tabulce 1. Všimněte si, že se mění úroveň nadpisu `\part` podle třídy dokumentu. Je to tak proto, aby nadpisy `\section` a méně významné měly stejnou úroveň, čímž se další programování maker významně zjednoduší. K této tabulce se vrátíme ještě v dalších dílech.

Úroveň nadpisů, které se objeví v obsahu, ovlivníme čítačem `tocdepth`. Pokud jej nastavíme na hodnotu 2 příkazem:

```
\setcounter{tocdepth}{2}
```


Tabulka 1: Přehled úrovní standardních nadpisů

<code>\part</code> (book, report)	-1	<code>\part</code> (article)	0
<code>\chapter</code>	0	<code>\section</code>	1
<code>\subsection</code>	2	<code>\subsubsection</code>	3
<code>\paragraph</code>	4	<code>\subparagraph</code>	5

budeme v obsahu mít všechny nadpisy do úrovně 2 včetně, tedy `\part`, `\chapter`, `\section` a `\subsection`.

Ukážeme si to na příkladech. Pro jednoduchost vynecháme vlastní nadpis „Obsah“ a všechny ukázky budou odděleny od textu vodorovnými linkami. Nejprve uvedeme část obsahu Zpravodaje 3/96 (viz obsah pomocného souboru na str. 145) s čítačem `tocdepth` nastaveným na hodnotu 1.

A Úvod	134
B Původ	135
C Dokumentace a nápověda	145

Obsah je velmi stručný. Podrobnější verzi obsahu získáme nastavením čítače `tocdepth` na hodnotu 2.

A Úvod	134
B Původ	135
1 Co je to $\text{T}_{\text{E}}\text{X}$	135
2 Jak mám vyslovovat „ $\text{T}_{\text{E}}\text{X}$ “	136
3 Co to je METAFONT	136
C Dokumentace a nápověda	145
18 Knihy o $\text{T}_{\text{E}}\text{X}$ u a příbuzná literatura	145
19 Kde lze najít tento dokument	146

15.2. Kde a jak je definován vzhled obsahu?

Předvedli jsme si způsob, jak rozhodneme, které nadpisy budou do obsahu zahrnuty. Nyní se již budeme věnovat změnám jeho vzhledu. Nejprve nás bude ovšem zajímat, odkud $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ příslušné definice bere.

Pozorný čtenář si jistě pamatuje, že se do pomocného souboru zapisují příkazy `\contentsline{typ}`, které se během tisku obsahu expandují na `\l@typ`. Makro `\section` tedy zapíše do pomocného souboru `\contentsline{section}`, a to se bude později expandovat na `\l@section`. Posledně jmenované makro společně s dalšími makry pro tisk obsahu je uvedeno v souboru definujícím třídu dokumentu. Tak například soubor `article.cls` obsahuje (`\l@part` úmyslně vynecháme):

```
\newcommand*\l@section[2]{%
  \ifnum \c@tocdepth >\z@
    \addpenalty\@secpenalty
    \addvspace{1.0em \@plus\p@}%
    \setlength\@tempdima{1.5em}%
    \begingroup
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      \leavevmode \bfseries
      \advance\leftskip\@tempdima
      \hskip -\leftskip
      #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2}\par
    \endgroup
  \fi}
\newcommand*\l@subsection{\@dottedtocline{2}{1.5em}{2.3em}}
\newcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
\newcommand*\l@paragraph{\@dottedtocline{4}{7.0em}{4.1em}}
\newcommand*\l@subparagraph{\@dottedtocline{5}{10em}{5em}}
\newcommand*\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}
\let\l@table\l@figure
```

Obvykle požadujeme tečkovanou linku mezi názvem a číslem strany. Pro usnadnění používáme `\@dottedtocline`, o němž se ještě za chvíli zmíníme. Jak bylo ukázáno v předchozích příkladech, `\section` se v obsahu vysází jiným písmem, bez tečkované linky a s vertikální mezerou. Proto je také definice `\l@section` odlišná. Na první pohled vypadá složitě, ale věřte, že to ve skutečnosti vůbec složitě není. Význam budeme odhalovat postupně s výkladem, jak lze vzhled změnit k obrazu svému.

Začneme sdělením, že všechna makra `\l@typ` požadují dva parametry. Prvním parametrem je název (pod)kapitoly, druhým je číslo strany. Tyto informace se tisknou ke konci definice `\l@section`. Připomeneme, že `\hb@xt@` je zkratka za `\hbox to` a šetří několik tokenů (to je významné zejména při generování formátu). Celá konstrukce

```
\hb@xt@\@pnumwidth{\hss #2}
```

je funkčně ekvivalentní známému (alespoň pro \LaTeX isty) zápisu

```
\makebox{\@pnumwidth}[r]{#2}
```

Makro `\makebox` ale prochází dosti složitou expanzí, a proto je výrazně méně efektivní než použité `\hb@xt@`.

Pozorný čtenář má již delší dobu na jazyku jedovatou poznámku. Makro `\l@section` má sice dva parametry, o čemž svědčí dvojka v hranatých závorkách, ale `\l@subsection` ani další makra žádné parametry nemají! Pozorný čtenář má pravdu, ale pouze částečnou. Trik spočívá v tom, že makro `\@dottedtocline` požaduje pět parametrů, ale v definicích jsou použity pouze tři. `\l@subsection` své dva parametry nespotřebuje a nechá je ve vstupním proudu netknuté. Po expanzi si je vezme `\@dottedtocline` jako čtvrtý a pátý parametr.

Poslední dva parametry makra `\@dottedtocline` již tedy známe. Nyní vysvětlíme první tři. Z pohledu na definice a kontrolou tabulky 1 asi tušíte, že prvním parametrem je úroveň nadpisu. Ta se porovnává s hodnotou čítače `tocdepth`, čímž se zjistí, zda se má příslušná položka v obsahu vytisknout. Makro `\l@section` provádí tento test na začátku primitivem `\ifnum`. Druhý parametr určuje odsazení od levého okraje tiskového zrcadla a třetí parametr vyhrazuje prostor pro číslo (pod)kapitoly. Stejná hodnota se použije též pro odsazení, pokud je název víceřádkový. Tehdy musíme také definovat vzdálenost, kde se mají řádky zlomit. Ta je uložena v makru `\@tocmarg`. Podobně do makra `\@pnumwidth` vložíme požadovanou šířku boxu pro stránkovou číslici. Příkladem takových definic může být:

```
\renewcommand*\@tocmarg{23mm}%
\renewcommand*\@pnumwidth{10mm}%
\renewcommand*\l@section{\@dottedtocline{1}{17mm}{5mm}}
```

Všechny rozměry jsme uvedli v milimetrech, abyste si je mohli odměřit obyčejným pravítkem. Příklad výstupu je uveden na obrázku 1.

Obrázek 1: Demonstrace parametrů makra `\@dottedtocline`

odsazení		\@tocmarg
4	Toto je velmi dlouhý název kapitoly, který se nevejde na jeden řádek, což je sice blbost vyskytující se jen zřídka, ale pro demonstraci sazby obsahu se docela hodí	159
	Zde je něco velmi podobného, ale tato kapitola nemá číslo, takže si můžete všimnout, jak se bude text odsazovat v tomto případě	163
		\@pnumwidth

Číslice 4 je odsazena od textu, protože je uzavřena v `\numberline{4}`. Toto makro má následující definici:

```
\def\numberline#1{\hbxt@\@tempdima{#1\hfil}}
```

Hodnota rozměru \@tempdima se vezme z třetího parametru makra \@dottedtocline. V případě makra \l@section je to však trochu jinak. Podíváme se na jeho definici znovu.

Makro začíná svoji činnost porovnáním hodnoty čítače tocdepth s nulou. Je-li hodnota čítače větší, má být nadpis v obsahu vytištěn, v opačném případě se přeskočí vše až k primitivu \fi včetně. Primitiv \ifnum, stejně jako dále se vyskytující \begingroup a \endgroup, již nejsou čistě L^AT_EXové konstrukce a pro jejich pochopení si musíte pročíst alespoň jednu z knih [4] nebo [5]. Pokud je tedy podmínka splněna, přidá se záporná penalizace, která informuje o tom, že zde je vhodné místo pro stránkový zlom. Dále následuje vertikální mezer. Ta bude samozřejmě vypuštěna, pokud by se dostala na začátek stránky. Zařídí to automaticky algoritmy pro stránkový zlom. Potom vložíme vhodnou velikost prostoru pro číslo kapitoly do proměnné \@tempdima. Uvnitř skupiny nastavíme parametry odstavce a vše vytiskneme tučným písmem.

Teď již umíme dělat jednoduché zásahy do vzhledu obsahu. Podívejme se znovu na náš příklad. Názvy kapitol nám tam připadají příliš tučné, chybí u nich tečkovaná linka, zatímco u podkapitol je zbytečně velký prostor pro číslo podkapitoly. Změníme tedy definice následujícím způsobem:

```
\renewcommand*\l@section{\@dottedtocline{1}{0mm}{1.5em}}
\renewcommand*\l@subsection{\@dottedtocline{2}{1.5em}{1.5em}}
```

Obsah pak bude vypadat takto:

A	Úvod	134
B	Původ	135
	1 Co je to T _E X	135
	2 Jak mám vyslovovat „T _E X“	136
	3 Co to je METAFONT	136
C	Dokumentace a nápověda	145
	18 Knihy o T _E Xu a příbuzná literatura	145
	19 Kde lze najít tento dokument	146

Při sazbě knihy s velmi hlubokým členěním je někdy odsazování na závadu. Stačí ovšem jednoduchá předefinice, např.:

```
\renewcommand*\l@section[2]{\@dottedtocline{1}{0mm}{1.5em}%
\bfseries #1}{\bfseries #2}}
\renewcommand*\l@subsection{\@dottedtocline{2}{0mm}{1.5em}}
```

Výsledkem pak bude:

A Úvod	134
B Původ	135
1 Co je to T _E X	135
2 Jak mám vyslovovat „T _E X“	136
3 Co to je METAFONT	136
C Dokumentace a nápověda	145
18 Knihy o T _E Xu a příbuzná literatura	145
19 Kde lze najít tento dokument	146

Také můžeme zcela opustit makro `\dottedtocline` a vytvořit nové definice třeba takto:

```
\renewcommand*\l@section[2]{\@tempdima 1.5em
    \noindent\hb@xt@\@pnumwidth
    {\hss\bfseries #2:}\enspace \textbf{#1}\par}
\renewcommand*\l@subsection[2]{\@tempdima 1.5em
    \noindent\hb@xt@\@pnumwidth
    {\hss #2:}\enspace #1\par}
```

Zde vidíte výsledek:

134: A Úvod
135: B Původ
135: 1 Co je to T _E X
136: 2 Jak mám vyslovovat „T _E X“
136: 3 Co to je METAFONT
145: C Dokumentace a nápověda
145: 18 Knihy o T _E Xu a příbuzná literatura
146: 19 Kde lze najít tento dokument

15.3. Jak se dostanou informace do pomocného souboru?

Můj školitel kdysi říkával, že život není žádná humanita. Platí to beze zbytku i v L^AT_EXu. Ukázali jsme si, že záznamy z pomocného souboru lze využít k vysázení obsahu v mnoha různých podobách. V praxi se nám ale může stát, že potřebná informace v pomocném souboru není. O její uložení do pomocného souboru se musíme postarat sami. Proto se nejprve podíváme, jakým způsobem provádí zápisy do pomocného souboru sám L^AT_EX.

Nebudeme se zabývat podrobným rozбором všech maker, protože to uživatel k ničemu nepotřebuje. Nebudou nás zajímat všechny cestičky, jimiž se budou

písmenka ubírat, než dojdou na správné místo správného souboru. Chceme pouze najít vhodné makro, které splní námi požadovaný úkol, a bude stačit, když mu porozumíme na uživatelské úrovni.

Pro zápis informací, které se mají objevit v obsahu, máme dvě makra: `\addcontentsline` a `\addtocontents`. Každé z nich má své použití.

Makro `\addtocontents` tvoří základ pro zápis informací do obsahu. Obvykle je automaticky voláno makry jako `\section` a `\caption`. Má tři parametry: typ souboru (uvádí se přípona, nejčastěji `toc = table of contents`, `lof = list of figures`, `lot = list of tables`), typ záznamu (`chapter`, `section`, `figure`, apod.) a vlastní text nadpisu. Je zřejmé, že křehká makra použitá v nadpisu vyžadují `\protect`. Jako příklad si uvedeme makro `\chapter`, které během své expanze použije:

```
\addcontentsline{toc}{chapter}%  
    {\protect\numberline{\thechapter}nadpis}
```

Zatím se nebudeme starat, jak se dostane *nadpis* z parametru makra `\chapter` do třetího parametru `\addcontentsline`. To si vysvětlíme v jednom z dalších pokračování, až se budeme zamýšlet nad vzhledem nadpisů. Všimneme si hlavně použití makra `\thechapter`, které obsahuje tisknutelnou podobu čítače `chapter`. V zásadě lze říci, že každý typ nadpisu má přiřazen stejnojmenný čítač, takže pro `\subsection` bychom použili `\thesubsection`, pro nadpis obrázku (`figure`) použijeme `\thefigure` apod.

Existuje typická situace, kdy použijeme `\addcontentsline` sami. Chceme například v knize nadpis jedné kapitoly vytisknout bez čísla. K tomuto účelu slouží varianta nadpisového makra s hvězdičkou. Pak se nám ale nadpis nedostane do obsahu. Zde tedy musíme připsat `\addcontentsline`. Ukážeme si to na příkladu kapitoly „Úvod“, přičemž navíc požadujeme, aby text byl v obsahu odsazen o místo vyhrazené pro číslo kapitoly. V dokumentu proto uvedeme:

```
\chapter*{Úvod}  
\addcontentsline{toc}{chapter}{\protect\numberline{}Úvod}
```

Možná se ptáte, jak se do obsahu dostane číslo stránky. Při své expanzi jej makro `\addcontentsline` doplní samo. Proto musí být `\addcontentsline` uvedeno *bezprostředně po* vytištění odpovídajícího nadpisu, protože jinak by číslo stránky v obsahu mohlo být nesprávné.

Makro `\addtocontents` má jen dva parametry: typ souboru a text. Používá se ke vkládání rozličných informací nesouvisejících s nadpisy; proto se také nedoplňuje číslo stránky. Můžeme například vložit vertikální mezeru příkazem

```
\addtocontents{toc}{\protect\vspace{1ex}}
```

Tímto způsobem také můžeme měnit definice maker. Makro `\@starttoc` totiž načítá pomocný soubor uvnitř skupiny. Pokud vložíme do druhého

parametru makra `\addtocontents` definice maker, budou tyto definice platit pouze do okamžiku ukončení tisku obsahu.

Podle návodu v této podkapitole dokážeme sestavit další typy obsahu. Představme si, že v dokumentu máme velké množství definic. Budeme je uvozovat makrem `\define`, jehož parametrem bude název definice. Tento název chceme vytisknout v seznamu definic. Všechny definice budeme číslovat s použitím čítače `definice` a pro vytištění seznamu si vymyslíme makro `\seznamdefinic`. Příslušná makra mohou vypadat přibližně takto:

```
\newcommand*\define[1]{%
  \refstepcounter{definice}% inkrementace čítače
  % Zde budou příkazy pro tisk čísla a názvu definice
  \addcontentsline{def}{definice}%
    {\protect\numberline{\thedefinice}#1}}
\newcommand*\l@definice{\@dottedtocline{0}{0em}{1.3em}}
\newcommand*\seznamdefinic{\chapter*{Seznam definic}
  \@starttoc{def}}
```

15.4. Poněkud špinavé triky

Celý mechanismus vytváření obsahu jsme si již popsali z uživatelské stránky. Nic nám nebrání ve zkoumání, co vše L^AT_EX snese. Jako úvod pro další experimenty lze uvést několik příkladů. Nebudou to ukázky akademické. Nejrůznější formy vzhledu obsahu budou demonstrovány na tom, co přinesl život, tedy přesněji zákazníci, kteří měli zcela vyhraněné požadavky.

Začneme Zpravodajem, který právě čtete. Víte, že existuje styl `csbul.sty`², který se pro sazbu používá. Víte také, že `\section` se do obsahu nepřenese, a použitá třída `ARTICLE` nezná `\chapter`. Jak je to zde uděláno? Vše řeší následující makra:

```
\newif\ifautkn@wn
\renewcommand\maketitle[1][\z@]{%
  % příkazy pro tisk nadpisu
  \ifautkn@wn % true, pokud je za \begin{clanek} uveden autor
    \addcontentsline{toc}{clanek}{\protect\cl@nek{\auth@r}{\t@tle}}%
  \else
    \addcontentsline{toc}{clanek}{\t@tle}%
  \fi
  % další příkazy
}
\def\cl@nek#1#2{#1: #2}
\def\l@clanek#1{% příkazy pro výstup do HTML
  \@dottedtocline{-2}{\z@}{2em}{#1}}
```

²<ftp://ftp.icpf.cas.cz/wagner/cstex/bulletin/csbuldoc.zip>

Všimněte si, že jsme vůbec nemuseli měnit definici `\tableofcontents`. Ve skutečném stylu `csbul.sty` ji sice měníme, ale z jiného důvodu. Chceme totiž současně vytisknout obsah ve formátu HTML pro zveřejnění na WWW. Proto i `\l@clanek` má několik dalších příkazů navíc, které používají hodnotu parametru. Pro tvorbu tištěného obsahu jsou však výše uvedená makra zcela postačující.

Složitější příklad představuje německý překlad knihy Pověsti českých hradů [6]. K některému hradu je uvedena pouze jedna pověst, u jiných je pověstí více. Vždy však začínáme jménem příslušného hradu. Pokud má pověst vlastní název, uvádíme jméno hradu makrem `\hrad`. V obsahu se číslo stránky v tomto případě neuvádí, protože souhlasí se stránkou, kde začíná první pověst. Pokud se ale ke hradu váže jediná pověst, která nemá vlastní název, uvedeme jméno hradu makrem `\shrad` a do obsahu musíme číslo stránky vytisknout. Pro název pověsti vždy použijeme `\section`. Navíc nakladatel vyžadoval netradiční vzhled, kdy položky obsahu jsou centrovány a stránková číslice je oddělena lomítkem. Vše zařídíme následujícími definicemi. (Makra `\@afterindenttrue`, `\@afterindentfalse` a `\@afterheading` vysvětlíme v jednom z dalších pokračování. Spokojíme se s tím, že jejich úkolem je zajištění, aby se nezlomila stránka hned pod názvem kapitoly nebo za prvním řádkem prvního odstavce.)

```
\def\hrad#1{% příkazy pro tisk nadpisu
  \addcontentsline{toc}{hrad}{#1}
  \@afterindentfalse \@afterheading}
\def\shrad#1{% příkazy pro tisk nadpisu
  \addcontentsline{toc}{shrad}{#1}
  \@afterindentfalse \@afterheading}
\def\section#1{% příkazy pro tisk nadpisu
  \addcontentsline{toc}{section}{#1}
  \@afterindentfalse \@afterheading}
\def\l@hrad#1#2{\penalty-50\vskip 1ex{\centering\textbf{#1}\par}\nobreak}
\def\l@shrad#1#2{\penalty-50\vskip 1ex{\centering\textbf{#1}\nobreak
  \hskip.8em/\nobreak\hskip.8em#2}\par}\nobreak}
\def\l@section#1#2{\centering#1\nobreak
  \hskip.8em/\nobreak\hskip.8em#2\par}}
```

Abychom šetřili místem, vypíšeme si jen část pomocného souboru.

```
\contentsline {section}{Vorwort}{7}
\contentsline {hrad}{Libušín}{9}
\contentsline {section}{Gericht und Wahl}{9}
\contentsline {hrad}{Vyšehrad}{18}
\contentsline {section}{Die Burg über der Moldau}{18}
\contentsline {section}{Der gro\ss{}e Sprung}{26}
\contentsline {hrad}{Děvín}{33}
\contentsline {section}{Kampf und Verrat}{33}
```



```

\contentsline {hrad}{Karlštejn}{44}
\contentsline {section}{Die Karlštejner Teufel}{44}
\contentsline {section}{Von einer List}{48}
\contentsline {section}{Die treue Freundschaft}{52}
\contentsline {shrad}{Hluboká}{108}
\contentsline {hrad}{Orlík}{118}
\contentsline {section}{Das Adlerness}{118}
\contentsline {shrad}{Radyně, Kašperk}{120}
\contentsline {shrad}{Přimda}{124}
\contentsline {shrad}{Bezděz, Blatná}{132}
\contentsline {hrad}{Kost}{138}
\contentsline {section}{Die knochenharte Burg}{138}

```

V ukázce si můžete prohlédnout, jak to dopadne po vytištění.

Vorwort / 7

Libušín

Gericht und Wahl / 9

Vyšehrad

Die Burg über der Moldau / 18

Der große Sprung / 26

Děvín

Kampf und Verrat / 33

Karlštejn

Die Karlštejner Teufel / 44

Von einer List / 48

Die treue Freundschaft / 52

Hluboká / 108

Orlík

Das Adlerness / 118

Radyně, Kašperk / 120

Přimda / 124

Bezděz, Blatná / 132

Kost

Die knochenharte Burg / 138

Nyní si předvedeme poněkud složitější příklad. Výsledek je stručný, proto uvidíte celý obsah knihy Mahájánské texty Heleny Petrovny Blavacké [7]. Požadavky na vzhled obsahu byly upřesňovány během sazby, takže mezi makry

zůstaly jakési relikty. Úmyslně je ponecháváme i v tomto příkladu. Napřed se ale na makra podíváme.

```

\def\chapter#1{\clearpage\thispagestyle{plain}
  \refstepcounter{chapter}
  \addcontentsline{toc}{chapter}{\protect\numberline{\Roman{chapter}}#1}
  % Další příkazy...
}
\def\chap#1{\clearpage\thispagestyle{plain}
  \setcounter{section}{0}
  \addcontentsline{toc}{chapter}{#1}
  % Další příkazy...
}
\def\l@chapter{\@dottedtocline{1}{\z@}{\z@}}
\def\l@section{\@dottedtocline{2}{\t@dim}{\Zdim}}
\def\numberline#1{{\scshape Zlomek #1.} }
\def\ln{\protect\p@ln}
\def\p@ln{\par}
\let\c@ln\relax
\def\toc@cln{\let\footnote@gobble \let\p@ln\space \let\c@ln\}
\AtBeginDocument{\addtocontents{toc}{\protect\toc@cln}}
\setcounter{tocdepth}{1}

```

Všimněte si, že v knize jsou použita dvě různá makra pro nadpis kapitol. Standardní `\chapter` čísluje kapitoly a zapisuje informace pro obsah. Varianta `\chap` se liší pouze tím, že kapitoly nejsou číslovány. Mohli bychom sice použít hvězdičkovou verzi `\chapter*`, ale pak bychom se o zápis do obsahu museli starat sami v dokumentu. Takto je situace uživatelsky příjemnější.

Definice maker `\l@chapter` a `\l@section` není nijak zvláštní. Za zmínku snad stojí jen použití rozměrů, jejichž hodnoty se měly vypočítat podle nějakého vzorce. Minulý čas v této větě osvětlíme za okamžik. Následuje definice makra `\numberline`. Ukazuje nové možnosti, ale definice sama je triviální.

Některé názvy mohou být dlouhé, takže se nevejdou na řádek. Ponechat rozdělení automatu není šťastné řešení, neboť zejména název má vypadat graficky příjemně a má se snadno číst. Přitom název obvykle sázíme jiným písmem (přínejmenším jinou velikostí) než obsah. Zlom řádku je proto velmi často jiný ve vlastním nadpisu a v obsahu. Nadefinujeme si tedy robustní makro `\ln`, které zlomí řádek v nadpisu, ale v obsahu vloží pouze mezeru. Podobné makro `\c@ln` zlomí řádek v obsahu, ale při sazbě nadpisu se chová jako `\relax`, tj. nedělá nic. Protože `\relax` je primitiv, nedochází při zápisu do pomocného souboru k jeho expanzi, a tudíž `\c@ln` nepotřebuje `\protect`. Při tisku obsahu musíme definice vyměnit a navíc potřebujeme zrušit tisk poznámek pod čarou. To zařídí makro `\toc@cln`, které prostřednictvím `\AtBeginDocument` zapíšeme do pomocného souboru na začátek obsahu (v této knize je obsah vytištěn na konci).

Poslední příkaz demonstruje, že jsme se rozhodli netisknout v obsahu názvy `\section`. Nakonec bylo dokonce předefinováno i makro `\section` tak, že se v něm `\addcontentsline` vůbec nevyskytuje. Makro `\l@section` je tedy relikv, jež při zpracování knihy není použit. Současně chybí algoritmus na výpočet rozměrů `\tib@dim` a `\Zdim`. Tyto relikvy zůstaly ve stylovém souboru pro případ, že by nakladatel změnil rozhodnutí.

Podívejme se nyní na výpis pomocného souboru. Při zpracování knihy došlo k expanzi českých znaků na \TeX ové sekvence, takže záznamy nejsou zrovna čitelné.

```
\toc@cln
\contentsline {chapter}{\`UVOD KOMENT\`ATORA}{5}
\contentsline {chapter}{P\v REDMLUVA\p@ln HELENY PETROVNY BLAVACK\`E}{8}
\contentsline {chapter}{\numberline {\uppercase {i}}HLAS TICHA}{11}
\contentsline {chapter}{\numberline {\uppercase {ii}}DV\v E STEZKY}{51}
\contentsline {chapter}{\numberline {\uppercase {iii}}SEDM BRAN}{92}
\contentsline {chapter}{POZN\`AMKY Heleny Petrovny
  Blavack\`e\footnote {Upozorn\v en\`i vydavatele: Pozn\`amky H.\,P.\,B.
  odr\`a\v zej\`i jej\`i vlastn\`i n\`azory; p\v r\`ipadn\`e omyly nebo
  nep\v resnosti byly v\nobreakspace {}textu ponech\`any tak,
  jak je autorka formulovala.}}{150}
\contentsline {chapter}{VYSV\v ETLIVKY}{165}
```

Výsledný obsah ovšem čitelný je, neboť $\LaTeX 2_{\epsilon}$ umí správně složit \TeX ové sekvence na znaky ζ -fontů (včetně `\`i`).

ÚVOD KOMENTÁTORA	5
PŘEDMLUVA HELENY PETROVNY BLAVACKÉ	8
ZLOMEK I. HLAS TICHA	11
ZLOMEK II. DVĚ STEZKY	51
ZLOMEK III. SEDM BRAN	92
POZNÁMKY Heleny Petrovny Blavacké	150
VYSVĚTLIVKY	165

Ještě větší oříšek představovaly dva svazky Tajemství Tibetu [8], [9]. Bylo nutno se přizpůsobit anglickému vydání [10]. První svazek má sedm dílů, druhý svazek má dva díly. Anglická verze je typograficky nejednotná, neboť v každém dílu je členění kapitol odlišné a jiným způsobem se uvádějí nadpisy v obsahu. Přesto mají oba svazky mnoho společného, a bylo tudíž vhodné vyřešit vše společným stylovým souborem. Zde uvedeme pouze makra, která se týkají tvorby obsahu. Přesto jich bude poměrně mnoho.

% Counters

```
\def\T@fem#1{\ifcase#1 \or PRVNÍ\or DRUHÁ\or TŘETÍ\or ČTVRTÁ\or
PÁTÁ\or ŠESTÁ\or SEDMÁ\or OSMÁ\else \@latexerr{Too large number}\fi}
\def\T@masc#1{\ifcase#1 \or PRVNÍ\or DRUHÝ\or TŘETÍ\or ČTVRTÝ\or
PÁTÝ\or ŠESTÝ\or SEDMÝ\or OSMÝ\else \@latexerr{Too large number}\fi}
```

% Parts

```
\def\part{\clearpage\null\thispagestyle{plain}\secdef\@part\@spart}
\def\@part[#1]#2{\par
\refstepcounter{part}
\addcontentsline{toc}{part}{\protect\part@line{\thepart}{#1}}
\T@switch
% Další příkazy...
}
```

% Switches

```
\def\B@switch{\setcounter{secnumdepth}{0}}
\def\R@switch{%
\ifcase\c@part \or %1
\setcounter{secnumdepth}{2}%
\def\thesection{\Roman{section}.}%
\addtocontents{toc}{\protect\part@I}%
\let\sec@wpul\textsc
\or %2
\setcounter{secnumdepth}{4}%
\def\thesection{ČÁST~\Roman{section}.}%
\def\thesubsection{}%
\def\thesubsubsection{\arabic{subsubsection}.}%
\addtocontents{toc}{\protect\part@II}%
\let\sec@wpul\textsc
\or %3
\setcounter{secnumdepth}{3}%
\def\thesection{Část~\arabic{section}:}%
\def\thesubsection{\arabic{subsection}.}%
\addtocontents{toc}{\protect\part@III}%
\let\sec@wpul\wpul
\or %4
\setcounter{secnumdepth}{3}%
\def\thesection{}%
\def\thesubsection{\arabic{subsection}.}%
\addtocontents{toc}{\protect\part@IV}%
```

```

\let\sec@wpul\wpul
\or %5
\setcounter{secnumdepth}{3}%
\def\thesection{Část\arabic{section}:}%
\def\thesubsection{\arabic{subsection}.}%
\addtocontents{toc}{\protect\part@III}%
\let\sec@wpul\textsc
\or %6
\setcounter{secnumdepth}{3}%
\def\thesection{Část\arabic{section}:}%
\def\thesubsection{\arabic{subsection}.}%
\let\sec@wpul\textsc
\or %7
\setcounter{secnumdepth}{3}%
\def\thesection{Část\arabic{section}:}%
\def\thesubsection{\arabic{subsection}.}%
\let\sec@wpul\textsc
\fi
}

% Contents

\def\tableofcontents{\part*\contentsname}\starttoc{toc}\par
\addtocontents{toc}{\protect\toc@cfn}}
\def\l@part#1#2{\addpenalty{-\@highpenalty}%
\addvspace{2.25ex plus\p@}%
\begingroup
\parindent\z@ \rightskip\@pnumwidth \parfillskip\z@ plus 1fill
\interlinepenalty\@M
\leavevmode\bfseries#1\par\nobreak
\endgroup}
\def\part@line#1#2{\partname~#1\\#2}
\def\partline#1{\clearpage
\addtocontents{toc}{\protect\l@part{#1}{0}}}
\def\l@chapter{\@dottedtocline{1}{\z@}{\z@}}
\def\l@section{\@dottedtocline{2}{\tib@dim}{\Zdim}}
\def\l@subsection{\@dottedtocline{3}{3.7em}{1em}}
\def\l@subsubsection{\@dottedtocline{4}{4.7em}{1em}}
\def\part@I{\settowidth\Zdim{XXVIII. }\let\numberline\r@numberline}
\def\part@II{\global\Zdim=4.7em \let\numberline\l@numberline}
\def\part@III{\global\Zdim=3.7em\relax}
\def\part@IV{\global\Zdim=\z@}
\newdimen\Zdim
\def\l@numberline#1{\hbox to\@tempdima{#1\hfil}}

```

```

\def\r@numberline#1{\hbox to\@tempdima{\hfil#1\ }}
\let\cIn\relax
\def\toc@cIn{\let\cIn\}
\setcounter{tocdepth}{4}
\newdimen\tib@dim

% Options

\DeclareOption{Ruzenec}{%
  \def\partname{KNIHA}
  \def\thepart{\T@fem@c@part}
  \let\T@switch\R@switch
  \tib@dim=\z@
  \let\sec@wpul\wpul
}
\DeclareOption{Bardo}{%
  \def\partname{DÍL}
  \def\thepart{\T@masc@c@part}
  \let\T@switch\B@switch
  \tib@dim=2em
  \let\sec@wpul\textsc
}
\ProcessOptions

```

Nejprve se zastavíme u čítačů. Později budeme požadovat tisk hodnoty slovně, a to někdy v mužském, jindy v ženském rodě. Proto jsme si vytvořili makra `\T@masc` a `\T@fem`. Hodnota smí být pouze v rozmezí 1–8, pokud chceme tisknout větší hodnotu, udělali jsme určitě někde chybu (takové číslo totiž v podkladech od nakladatele není).

V makru `\part` si povšimněte hlavně makra `\T@switch`. To provádí specifické přepínání, k němuž se ještě dostaneme. `\B@switch` není zajímavý, ale `\R@switch` se větví podle čísla dílu a v každé větvi se zapíše do pomocného souboru přepínací makro.

Trochu jsme změnili i definici `\tableofcontents`. Makra `\l@typ` jsou již důvěrně známá. Za povšimnutí stojí pouze skutečnost, že rozměry požadované v `\@dottedtocline` budou vypočteny později, a to v přepínacích makrech.

Nejzajímavější je část, kde definujeme volitelné parametry stylového souboru. Zde se nastaví některé rozměry a zejména se definuje chování makra `\part`.

Opět si ukážeme část pomocného souboru prvního svazku.

```

\toc@cIn
\contentsline {chapter}{PŘEDMLUVA}{13}
\contentsline {part}{\part@line {PRVNÍ}{RŮŽENEC
  SKVOSTNÝCH DRAHOKAMŮ}}{29}

```

```

\part@I
\contentsline {chapter}{PŘEDMLUVA}{29}
\contentsline {chapter}{VZÝVÁNÍ A\nobreakspace {}ÚVOD}{32}
\contentsline {chapter}{DVACET OSM ZÁKLADNÍCH PŘÍKAZŮ JÓGY}{33}
\contentsline {section}{\numberline {\uppercase {i}.}Deset
příčin bolesti}{33}
\contentsline {section}{\numberline {\uppercase {ii}.}Deset podmínek}{37}
\contentsline {part}{\part@line {DRUHÁ}{STEZKA K\nobreakspace {}NIRVÁNĚ:
JÓGA VELKÉHO \cIn SYMBOLU}}{141}
\part@II
\contentsline {chapter}{PŘEDMLUVA}{141}
\contentsline {chapter}{VZÝVÁNÍ A\nobreakspace {}ÚVOD}{144}
\contentsline {section}{\numberline
{ČÁST\nobreakspace {} \uppercase {i}.}PŘÍPRAVNÉ NAUKY}{145}
\contentsline {section}{\numberline
{ČÁST\nobreakspace {} \uppercase {ii}.}PODSTATNÝ OBSAH}{153}
\contentsline {subsection}{\numberline {}Obvyklá cvičení}{153}
\contentsline {subsubsection}{\numberline {1.}Jóga jednodovosti}{153}
\contentsline {subsubsection}{\numberline {2.}Jóga nestvořeného}{179}
\contentsline {part}{\part@line {TŘETÍ}{STEZKA VĚDĚNÍ:
JÓGA ŠESTI NAUK}}{212}
\part@III
\contentsline {chapter}{PŘEDMLUVA}{212}
\contentsline {chapter}{VZÝVÁNÍ A\nobreakspace {}ÚVOD}{215}
\contentsline {chapter}{I. NAUKA O\nobreakspace {}PSYCHICKÉM TEPLE}{217}
\contentsline {section}{\numberline
{Část\nobreakspace {}1:}Pět přípravných cvičení}{217}
\contentsline {subsection}{\numberline {1.}Vidění
těla jako prázdnoty}{217}
\contentsline {subsection}{\numberline {2.}Vidění
psychického nervového systému jako prázdnoty}{218}
\contentsline {subsection}{\numberline {3.}Vidění ochranného kruhu}{221}

```

Všimněte si, jak přepínání významu makra \numberline změnilo vzhled jednotlivých částí obsahu.

PŘEDMLUVA 13

KNIHA PRVNÍ

RŮŽENEC SKVOSTNÝCH DRAHOKAMŮ

PŘEDMLUVA	29
VZÝVÁNÍ A ÚVOD	32
DVACET OSM ZÁKLADNÍCH PŘÍKAZŮ JÓGY	33
I. Deset příčin bolesti	33
II. Deset podmínek	37

KNIHA DRUHÁ
STEZKA K NIRVÁNĚ: JÓGA VELKÉHO
SYMBOLU

PŘEDMLUVA	141
VZÝVÁNÍ A ÚVOD	144
ČÁST I. PŘÍPRAVNÉ NAUKY	145
ČÁST II. PODSTATNÝ OBSAH	153
Obvyklá cvičení	153
1. Jóga jednobodovosti	153
2. Jóga nestvořeného	179

KNIHA TŘETÍ
STEZKA VĚDĚNÍ: JÓGA ŠESTI NAUK

PŘEDMLUVA	212
VZÝVÁNÍ A ÚVOD	215
I. NAUKA O PSYCHICKÉM TEPLE	217
Část 1: Pět přípravných cvičení	217
1. Vidění těla jako prázdnoty	217
2. Vidění psychického nervového systému jako prázdnoty	218
3. Vidění ochranného kruhu	221

Další příklad představuje obsah knihy Milaräpa, velký tibetský jógin [11]. V této knize každá kapitola začíná krátkým shrnutím, které chceme přenést do obsahu. Nejprve se znovu podíváme na makra, jimiž toho bylo dosaženo.

`% Counters`

```
\def\T@asc#1{\ifcase#1 \or PRVNÍ\or DRUHÝ\or TŘETÍ\or ČTVRTÝ\or  
PÁTÝ\or ŠESTÝ\or SEDMÝ\or OSMÝ\else \latexerr{too large number}\fi}
```

`% Chapters`

```
\newtoks\chap@toc  
\def\chapter{\ifmode \par \fi \@ifstar{\s@chapter}{\n@chapter}}  
\def\s@chapter#1{\def\chap@head{#1}\chap@toc{#1}\zw@chapter}  
\def\n@chapter#1{%  
  \refstepcounter{chapter}  
  \def\chap@head{KAPITOLA \thechapter\#\1}  
  \chap@toc{\protect\numberline{\thechapter}#1}\zw@chapter}  
\newcommand\zw@chapter[1][\relax]{%  
  \ifnewpage \clearpage \fi \newpagetrue  
  \def\zw@{#1}\def\zw@next{\relax}  
  \ifx\zw@\zw@next
```



```

\let\zw@\relax
\edef\zw@next{\noexpand\addcontentsline
               {toc}{chapter}{\the\chap@toc}}
\else
\edef\zw@next{\noexpand\addcontentsline{toc}{chapter}%
               {\noexpand\protect\noexpand\cpt@entry{\the\chap@toc}{#1}}}
\fi
\zw@next
% Další příkazy...
\@afterindenttrue\@afterheading}

% Contents

\def\tableofcontents{\cleardoublepage {\toc@cln
\centerline{\Large\textbf{OBSAH}}\vskip2\baselineskip
\@starttoc{toc}\par}}
\def\l@part#1#2{\penalty-100\addvspace{2\baselineskip}
\begin{center}\Large \interlinepenalty\@M
\def\numberline##1{DÍL \T@masc ##1\}\#1
\end{center}\nobreak}
\def\l@chapter#1#2{\addvspace{1ex plus 1ex minus 2pt}
\begingroup
\def\numberline##1{KAPITOLA~##1: }%
\leftskip\parindent \rightskip\@tocrmarg
\parfillskip -\rightskip
\interlinepenalty\@M
\noindent \null \hskip -\leftskip #1\nobreak
\leaders\hbox{$\m@th \mkern\dotsep mu
\hbox{.}\mkern\dotsep mu$}\hfill
\nobreak \hb@xt@\@pnumwidth{\hfil
\normalfont\normalcolor#2}\par
\endgroup}
\def\cpt@entry#1{#1\}
\def\l@section{\@dottedtocline{1}{\z@}{\Zdim}}
\def\numberline#1{\hb@xt@\@tempdima{\hss#1.\hskip.5em}}
\def\l@subsection{\@dottedtocline{2}{\parindent}{\Zdim}}
\newdimen\Zdim \Zdim=3em
\def\nl{\protect\p@ln}
\def\p@ln{\par}
\let\cIn\relax
\def\ccolon{\protect\c@colon}
\def\c@colon{\par}
\def\toc@cln{\let\footnote@gobble \let\p@ln\empty
\def\c@colon{: }\let\cIn\}

```

```

\setcounter{tocdepth}{2}
\setcounter{secnumdepth}{1}
\newdimen\tib@dim

```

Definici čítače již nebudeme komentovat. Všimneme si však definice `\chapter`. Makro `\@ifstar` přepíná mezi dvěma dalšími makry podle toho, zda za `\chapter` byla hvězdička. Ta připraví informaci do registru `\chap@toc` buď s číslem nebo bez čísla kapitoly a zavolají `\zw@chapter`. Posledně jmenované makro má nepovinný parametr, který obsahuje ono zmíněné shrnutí. Primitivem `\ifx` testujeme přítomnost nepovinného parametru a podle okolností nadefinujeme vhodně `\zw@next`. Při definici potřebujeme expandovat token registr `\chap@toc`, proto jsme makro definovali pomocí `\edef`. Nesmíme ještě expandovat `\addcontentsline`, proto před něj připišeme `\noexpand`. Nesmíme expandovat ani `\cpt@entry` a před ním stojící `\protect`, proto ta záhadná skupina primitivů `\noexpand`.

Makro `\tableofcontents` jsme mírně upravili a makra `\l@typ` již známe. Za povšimnutí stojí `\l@part` a `\l@chapter`, která si lokálně předefinovávají `\numberline`, a `\chap@toc` používané pro tisk souhrnu kapitoly. Makro `\toc@cln` i jeho „přátele“ již také známe z předchozích příkladů. Zde se navíc musíme postarat o dvojtečku.

Můžeme tedy přistoupit k zobrazení části pomocného souboru:

```

\contentsline {chapter}{PŘEDMLUVA KOMENTÁTORA}{13}
\contentsline {chapter}{0\nobreakspace }{ČESKÉM PŘEKLADU}{27}
\contentsline {chapter}{PŘEDMLUVA W.\,Y. EVANS-WENTZE}{31}
\contentsline {chapter}{ÚVOD}{33}
\contentsline {section}{\numberline {\uppercase {i}}Význam
  Džecün Kambumu}{33}
\contentsline {section}{\numberline {\uppercase {ii}}Historická
  hodnota vyprávění}{34}
\contentsline {part}{\numberline {i}STEZKA TEMNOTY}{67}
\contentsline {chapter}{\cpt@entry {\numberline {\uppercase {i}}PŮVOD
  A\nobreakspace }{NAROZENÍ}{Ráčhungüv sen vedoucí k\unhbox \voidb@x
  \penalty \@M \ }napsání tohoto životopisu; \cfn Milaräpüv původ
  a\unhbox \voidb@x \penalty \@M \ }narození.}}{67}
\contentsline {chapter}{\cpt@entry {\numberline {\uppercase {ii}}CHUŤ
  STAROSTÍ}{Smrt a\unhbox \voidb@x \penalty \@M \ }poslední vůle
  Milaräpova otce; protiprávní prisvojení si majetku strýcem
  a\unhbox \voidb@x \penalty \@M \ }tetou
  z\unhbox \voidb@x \penalty \@M \ }otcovy strany;
  z\unhbox \voidb@x \penalty \@M \ }toho vzniklé starosti, jež musel
  Milaräpa snášet se svou matkou
  a\unhbox \voidb@x \penalty \@M \ }sestrou.}}{78}
\contentsline {chapter}{\cpt@entry {\numberline {\uppercase {iii}}ŠKOLENÍ

```

```

V\nobreakspace {}ČERNÉM UMĚNÍ}{Džecünův guru
a\unhbox \voidb@x \penalty \@M \ {}jeho ovládání černého umění.
Džecün zničí magii pětatřicet svých nepřátel
a\unhbox \voidb@x \penalty \@M \ {}bohatou
žeň ječmene ostatních.}}{85}

```

Zde je pak vidět výsledek.

PŘEDMLUVA KOMENTÁTORA	13
O ČESKÉM PŘEKLADU	27
PŘEDMLUVA W. Y. EVANS-WENTZE	31
ÚVOD	33
I. Význam Džecün Kambumu	33
II. Historická hodnota vyprávění	34

DÍL PRVNÍ

STEZKA TEMNOTY

KAPITOLA I: PŮVOD A NAROZENÍ

Rächungův sen vedoucí k napsání tohoto životopisu; Milaräpův původ a narození. 67

KAPITOLA II: CHUŤ STAROSTÍ

Smrt a poslední vůle Milaräpova otce; protiprávní přisvojení si majetku strýcem a tetou z otcovy strany; z toho vzniklé starosti, jež musel Milaräpa snášet se svou matkou a sestrou. 78

KAPITOLA III: ŠKOLENÍ V ČERNÉM UMĚNÍ

Džecünův guru a jeho ovládání černého umění. Džecün zničí magii pětatřicet svých nepřátel a bohatou žeň ječmene ostatních. 85

15.5. Obsah před každou kapitolou

Článek o \TeX Live v minulém čísle Zpravodaje [12] postavil redakci před další problém. Článek má totiž vlastní obsah a jeho vypuštěním by došlo ke snížení informační hodnoty. Ladění nových maker je pracné, zvláště když požadovaný účel nabízí balík MINITOC. Ten ovšem zapisuje miniobsah za `\chapter`, ale pohledem do návodu ke psaní článků do Zpravodaje zjistíme, že takové makro se zde nevyskytuje. Souvisí to přirozeně s tím, že používáme třídu ARTICLE. Přesto existuje rychlé a jednoduché řešení, i když \TeX ování pak není zcela efektivní.

Nejprve si nadefinujeme čítač pro kapitoly a falešné makro `\chapter`. Čítač inkrementujeme pomocí `\stepcounter`, protože nechceme křížové odkazy na kapitoly. Makro také pouze učiní zápis do pomocného souboru, ale nebude tisknout žádný nadpis. Nakonec zařídíme, aby uživatel ve svém dokumentu nesměl `\chapter` napsat.

```
\newcounter{chapter}
\def\thechapter{\arabic{chapter}}
\def\chapter{\secdef@chapter@schapter}
\def\chapter[#1]#2{\stepcounter{chapter}% no references
  \addcontentsline{toc}{chapter}%
    {\protect\numberline{\thechapter}#1}}
\def@schapter#1{}
\AtBeginDocument{\let\chapter\undefined} % No real \chapter's
```

Balík MINITOC vyžaduje též přítomnost `\l@chapter` v obsahu. Do pomocného souboru jsme potřebné makro zapsali, ale nechceme nic tisknout. Proto v definici, kterou vidíte níže, použijeme úroveň nadpisu `\@M`, což znamená 10 000. Museli bychom tedy udělat moc velkou chybu, aby se do obsahu tyto záznamy dostaly.

```
\def\l@chapter{\@dottedtocline{\@M}{\z@}{\z@}}
```

Uvedený příklad demonstruje, že při troše znalostí lze i tak zdánlivě neproveditelnou věc lze v \LaTeX u naprogramovat za méně než pět minut.

16. Literatura

Prvních pět citací se týká knih a článků, které lze použít k dalšímu studiu a k osvětlení vybraných problémů, o nichž jsme se z různých důvodů nezmínili. Další odkazy se týkají knih, z nichž byly převzaty ukázky.

1. Z. Wagner: *L^AT_EXová kuchařka/2*. Zpravodaj Československého sdružení uživatelů \TeX u, **6** (4), 260–289 (1996).
2. M. Goossens, F. Mittelbach, A. Samarin: *The L^AT_EX Companion*. Addison Wesley, Reading 1994, ISBN 0-201-54199-8.
3. P. Olšák: *Putování písmene ř z klávesnice na papír*. Zpravodaj Československého sdružení uživatelů \TeX u, **7** (3), 129–140 (1977).
4. D. E. Knuth: *The T_EXbook*. Addison Wesley, Reading 1984. ISBN 0-201-13448-9.
5. P. Olšák: *T_EXbook naruby*. Vyjde v nakladatelství KONVOJ. Elektronická verze je k dispozici na <http://math.feld.cvut.cz/olsak/tbn/>.
6. E. Petiška: *Sagenschatz der böhmischen Burgen*. Martin, 1994.

7. K. Minařík – H. P. Blavacká: *Mahájánské texty*. Canopus, Praha 1995. ISBN 80-85202-26-3.
8. K. Minařík: *Tajemství Tibetu 1*. Sedm tibetských textů. Canopus, Praha 1994, 1996. ISBN 80-85202-23-9.
9. K. Minařík: *Tajemství Tibetu 2*. Tibetská kniha mrtvých (Bardo thödol). Canopus, Praha 1994, 1996. ISBN 80-85202-24-7.
10. W. Y. Evans-Wentz: *Tibetan Yoga And Secret Doctrines*. Oxford University Press, London 1935.
11. W. Y. Evans-Wentz, K. Minařík: *Milaräpa, velký tibetský jógin*. Canopus, Praha 1996. ISBN 80-85202-28-X.
12. S. Rahtz, M. Goossens: *The T_EX Live Guide, version 2*. Zpravodaj Československého sdružení uživatelů T_EXu, **7** (1–2), 34–88 (1997).

Zdeněk Wagner
wagner@mbox.cesnet.cz

Zápis z valného shromáždění Československého sdružení uživatelů T_EXu konaného dne 12. 10. 1997

Valné shromáždění bylo zahájeno ve 14.00 hod na půdě fakulty informatiky MU v Brně. Před tím, v dopoledních hodinách, zde proběhla přednáška Phila Taylora o projektu NTS a o ε -T_EXu. Přednášku ve spolupráci s fakultou informatiky uspořádal ζ TUG. Vstup byl otevřen pro veřejnost. V kuloárech se rozdávaly pozvánky na EuroT_EX98 ve St. Malo, Bretagne.

Valného shromáždění se zúčastnilo 30 řádných členů sdružení, převážně z Brna.

Petr Sojka informoval v rámci zprávy o činnosti o aktivitách jednotlivých členů výboru. Daří se vydávat Zpravodaj díky panu Wagnerovi a časopis dokonce dostal přiděleno ISSN. Rovněž administrativní zázemí ζ TUGu se zlepšilo zvláště díky panu ing. Váchovi a paní ing. Váchové. Neplatícím členům sdružení byla zaslána s časopisem 1–2/97 upomínka a připravuje se aktualizace databáze členů. Členství budou zbaveni dlouhodobě neplatící členové a z databáze se odstraní