

Zpravodaj Československého sdružení uživatelů TeXu

Zdeněk Wagner

LaTeXová kuchařka / 5

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 9 (1999), No. 3, 161–167

Persistent URL: <http://dml.cz/dmlcz/149850>

Terms of use:

© Československé sdružení uživatelů TeXu, 1999

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>

V této části seriálu bude více než jinde platit to, co bylo uvedeno v prvním dílu. Půjde zde zejména o demonstraci rozličných maker, s jejichž pomocí lze ovlivňovat vzhled dokumentu. Přitom některé ukázky mohou z hlediska typografie vypadat ošklivě. Než tedy taková makra ve svých dokumentech použijete, zvažte důkladně, zda tím svůj text neznehodnotíte.

26. Kde L^AT_EX vezme rozměry stránky

Rozměry stránky očekává T_EX v registrech `\hsize` a `\vsize`. L^AT_EXisté však nastavují `\textwidth` a `\textheight`. L^AT_EX je ale jen velkým balíkem maker a zpracovává se stejným programem. Mechanismus, jak funguje nastavování rozměrů stránky, je před uživateli ukryt. My se o něm stručně zmíníme, aby další text byl srozumitelný.

Plain T_EX neobsahuje žádné makro pro vícesloupcovou sazbu. Chceme-li sázet text do více sloupců, musíme se o to postarat sami. Text se musí vsázet do dlouhého úzkého sloupce. Dále musíme T_EXu vnutit jinou `\output` rutinu, která tento polotovar rozdělí na sloupce a usadí je vedle sebe. L^AT_EX obsahuje podporu pro dvousloupcovou sazbu. Lze buď použít volbu `twocolumn` nebo přepínat pomocí příkazů `\twocolumn` a `\onecolumn`. V obou případech se vyjde z celkové šířky stránky zadané v `\textwidth`. Při jednosloupcové sazbě je tím definováno vše, při dvousloupcové sazbě musíme vypočítat šířku sloupce. Odečteme tedy mezeru mezi sloupci zadanou v proměnné `\columnwidth` a výsledek vydělíme dvěma. Šířku sloupce pak vložíme do `\hsize`, `\linewidth` a `\colwidth`. Tuto činnost provede nejprve `\begin{document}`, potom přepínací makra `\twocolumn` a `\onecolumn`. Podobně se zachází s registrem `\vsize`. Vidíme tedy, že L^AT_EX nakonec vloží rozměry sloupce (před rozdělením) do registrů `\hsize` a `\vsize`. Právě hodnota registru `\hsize` bude pro nás v dalším textu důležitá.

27. Rozměry odstavce

Základní rozměr odstavce je uložen v registru `\hsize`. Z něho je odvozena šířka řádku. Na levé straně řádku je vynecháno místo velikosti `\leftskip`, na pravé straně je volné místo velikosti `\rightskip`. Tyto registry jsou inicializovány

na nulovou hodnotu, takže text zabírá přesně šířku `\hsize`. Drobný rozdíl je v prvním a posledním řádku odstavce. Na začátku prvního řádku je navíc meze-
zera velikosti `\parindent`, pokud ovšem není potlačena příkazem `\noindent`.
Na konci posledního řádku je meze-
zera velikosti `\parfillskip`. Ta má obvykle nekonečnou roztažitelnost.

V tomto odstavci si demonstrujeme použití `\leftskip` a `\rightskip`. Levý okraj byl nastaven na 2 cm, pravý na 1 cm. Můžete si to snadno ověřit pravítkem. Všimněte si, že první řádek má navíc odstavcovou zarážku stejné velikosti jako jiné odstavce. Je to tím, že hodnota registru `\parindent` nebyla změněna. Zůstala zachována i hodnota `\parfillskip`, takže nedojde k problémům při zlomu posledního řádku. Registrům lze přiřadit i záporné hodnoty. Text pak bude přesahovat mimo sazební obrazec. Dále si musíme uvědomit, že algoritmus řádkového zlomu použije ty hodnoty, které jsou platné na konci odstavce. Pokud měníme hodnoty `\leftskip`, `\rightskip` nebo jiných registrů jen lokálně, musíme ukončit odstavec ještě před koncem skupiny.

Vynulováním registrů `\parindent` a `\parfillskip` lze sázet do bloku, jehož východová řádka bude zcela zaplněna. Vypadá to efektně, ale většinou tím způsobíme přetečené a podtečené boxy. Jedinou nápravou je v takovém případě zásah do textu. I zde byla poslední věta upravována tak dlouho, až to vyšlo.

Někdy potřebujeme vykousnout do textu pravoúhlý otvor. Pro tyto účely můžeme využít registrů `\hangafter` a `\hangindent`. Hodnota vložená do registru `\hangindent` určuje velikost odsazení. Je-li hodnota kladná, odsadí se text zleva.

Záporná hodnota způsobí odsazení vpravo. Registr `\hangafter` určuje, po kolika řádcích odstavce se má začít odsazovat. Pokud je hodnota záporná, pak je tím naopak označeno, kolika počátečních řádků se odsazení týká. Uvažuje se samozřejmě absolutní hodnota. V tomto odstavci jsme do `\hangafter` vložili hodnotu 3 a odsazení bylo nastaveno na 2 cm.

Zde jsme nastavením `\hangafter -4 \hangindent -1cm` přestěhovali otvor do protilehlého rohu odstavce. Vhodnou změnou znamének lze otvor vynechat i ve dvou zbývajících rozích. Všimněte si, že v obou případech nám navíc zůstala odstavcová zarážka v prvním řádku. Její velikost je rovna `\parindent`. V souvislosti s tímto příkladem ale musíme zdůraznit velmi důležitou vlastnost stylů CZECH a SLOVAK. Použijí-li se s parametrem `split`, je sice automaticky při dělení slova opakován spojovník na začátku následujícího řádku, ale uživatel je za to penalizován změnou kategorie znaku „-“ na 13. Následkem toho nelze v dokumentu zadávat záporná čísla. Lze to řešit tím, že v preambuli dokumentu, dokud ještě kategorie změněny nejsou, použijeme definici `\def\minus{-}`. Výše uvedenou hodnotu registru `\hangafter` pak

budeme zadávat jako `\minus 4`. Podobně naložíme i s rozměrovým registrem `\hangindent`.

Musíme se ještě zmínit o příčině nejčastější chyby, která způsobí, že primitivy `\hangafter` a `\hangindent` nefungují. \TeX totiž na začátku odstavce tyto hodnoty vynuluje. Pokud je tedy zadáme ještě před prvním znakem odstavce, jsme stále ve vertikálním režimu a v okamžiku zpracování prvního znaku o ně přijdeme. Pokud je z estetických důvodů nechceme nastavovat uvnitř odstavce nebo na jeho konci, musíme před těmito příkazy použít `\leavevmode` a nesmíme za nimi nechat prázdný řádek.

Podívejme se nyní, jakého efektu dosáhneme prostředím `CENTERLAST`, jehož definice je uvedena níže:

```
\newenvironment{centerlast}{%
  \parindent 0mm
  \leftskip 0mm plus 1fil
  \rightskip -\leftskip
  \parfillskip 0mm plus 2fil
  \parskip\baselineskip
  \def\{\unskip\hskip\parfillskip\break\ignorespaces}}%
  {\par\vspace{\parskip}}
```

Toto nastavení způsobí, že poslední řádek odstavce bude vycentrován. Jak je toho dosaženo? Nejprve je odstraněna odstavcová zarážka. Dále vložíme nekonečnou roztažitelnost do registru `\leftskip`. Roztažitelnost v registru `\rightskip` je také nekonečná, ale se záporným znaménkem. Při sazbě řádků se tedy vyruší. Všechny řádky proto budou zabírat plnou šířku `\hsize`. Poslední řádek má kromě `\rightskip` ještě `\parfillskip`. Zde \TeX sečte hodnoty roztažitelnosti a výsledkem bude `1fil`. Ta je stejná jako v `\leftskip`. Tím je zařízeno centrování posledního řádku.

Odstavce by nepůsobily příliš estetickým dojmem, kdyby byly sesazeny těsně za sebe. Proto do registru `\parskip` vložíme hodnotu `\baselineskip`. Mezi odstavci tak bude vynechán právě jeden volný řádek.

Někdy nám poslední řádek vyjde extrémně krátký. Ani to nepůsobí vzhledně. Lze si pomoci tím, že zkrátíme předposlední řádek a necháme jej vycentrovat. Pro přechod na nový řádek používáme v \LaTeX u makro `\`. Aby fungovalo i zde, musí mít jinou definici. Pro jistotu nejprve odebereme předchozí mezeru. Pak musíme vložit stejnou mezeru, jakou používáme na konci odstavce, a přidat `\break`, což vynutí řádkový zlom. Pak budeme ignorovat všechny následující mezery.

Příklad použití je zde.

Aby nemusel uživatel pamatovat na to, že musí ponechat prázdný řádek před `\end{centerlast}`, vložíme do druhé části definice prostředí explicitní `\par`. Na konci prostředí bude také nastavena původní hodnota registru `\parskip`. Vertikální mezeru tedy také vložíme zde. V tomto příkladu mlčky předpokládáme, že původní hodnota `\parskip` byla nulová. Pokud mohla být nenulová a nechceme mít na konci tohoto prostředí vertikální mezeru jiné velikosti, budeme potřebovat poněkud chytřejší makro. Tuto úlohu však ponecháme na čtenářích.

Ještě složitější vzhled odstavce dosáhneme použitím primitivu `\parshape`. Tento primitiv má lichý počet parametrů. Prvním parametrem je počet řádků, které jsou dále specifikovány. Pro každý řádek je uvedena dvojice rozměrů: odsazení a šířka řádku. Má-li odstavec méně řádků než kolik dvojic rozměrů je uvedeno, jsou nadbytečné informace ignorovány. V opačném případě se poslední dvojice rozměrů odsazení/šířka použije pro všechny následující řádky. Podobně jako `\hangafter` je i `\parshape` vynulován na začátku odstavce. Musíme si tudíž dát pozor, abychom jej nenastavovali ve vertikálním režimu.

Příklad použití zde neuvedeme. Najdete jej například ve Zpravodaji č. 1 z roku 1996. V něm byl tímto způsobem vsazen do textu obrázek čarodějnice na straně 38. Ještě důmyslnější využití popsal Petr Olšák v knize *T_EXbook* naruby na straně 236.

Ruční odměřování parametrů pro `\parshape` není zrovna pohodlné. Proto lze na CTAN najít styly `SHAPEPAR`, `WINDOW` a řadu jiných, které umožní zadání tvaru odstavce a parametry se vypočtou vhodným makrem.

28. Usazujeme iniciály

Cílem této kapitoly je předvedení dalšího způsobu, jak lze využít parametrů `\hangafter` a `\hangindent`. Ukážeme si zde, jak lze usazovat iniciály na začátek odstavce. Z cvičných důvodů použijeme iniciálu na začátku každého odstavce. V běžném textu to není zvykem, iniciály se obvykle používají pouze na začátku kapitoly.

Na začátku odstavce musíme vynechat obdélníkový otvor. Jak jsme si již ukázali, můžeme pro tento účel využít `\hangafter` a `\hangindent`. Šířku vynechaného otvoru přizpůsobíme šířce písmene. Často se první řádek připojuje k iniciále těsněji než řádky následující. Toho snadno docílíme nastavením záporné hodnoty do registru `\parindent`. Nyní si předvedeme definici prostředí, které tuto činnost provede.

```
\newdimen\Iwidth
\newbox\InicBox
```

```

\newcommand\Iniciala
  {\usefont{IL2}{pp1}{m}{n}\fontsize{44}{44}\selectfont}
\newenvironment{inicialy}{\parindent -5dd
  \def\iniciala##1{\setbox \InicBox=\hbox
    {\kern-2dd\Iniciala ##1\kern -\parindent}}%
  \Iwidth=\wd\InicBox
  \leavevmode \hangafter -3 \hangindent\Iwidth
  \vadjust{\{\everypar{\}\noindent
    \smash{\lower 20dd \hbox to \z@ {\box\InicBox\hss}}}}}%
  \everypar{\iniciala}}{\par}

```

Nejprve jsme si definovali pomocný rozměr `\Iwidth`, box `\InicBox` a font pro tisk iniciály. Uvnitř prostředí pak nastavíme záporný `\parindent`. Dále si nadefinujeme makro `\iniciala`, které vyžaduje jeden parametr. Tímto parametrem je požadované písmeno, které máme vytisknout jako iniciálu. Definici zapisujeme uvnitř definice prostředí, proto musíme znak `#` zdvojit.

Makro `\iniciala` začíná svoji činnost tím, že vysadí iniciálu do pomocného boxu. Opticky vypadá iniciála lépe, když poněkud vyčnívá doleva. Proto je na začátku boxu záporný `\kern`. Za iniciálou je další `\kern`. Jeho velikost je záporně vzatá hodnota `\parindent`. Bez této mezery by druhý a třetí řádek byly těsně nalepeny na iniciálu a první řádek by do ní zasahoval. V následující instrukci změříme šířku takto vytvořeného boxu a vložíme ji do `\Iwidth`.

Touto definicí jsme ukončili přípravné akce a můžeme zahájit sestavení odstavce. Nejprve je nutno zajistit přechod do horizontálního režimu. Použijeme k tomu `\leavevmode`. Dále nastavíme příslušné hodnoty registrů `\hangindent` a `\hangafter`. O vložení záporné mezery specifikované v registru `\parindent` se postará algoritmus řádkového zlomu. Iniciálu pak vložíme na správné místo pomocí `\vadjust`. Musíme přejít do horizontálního režimu, abychom směli použít `\lower`. Tím totiž usazujeme box s iniciálou o dva řádky níže. Makro `\smash` vynuluje hloubku a výšku boxu. Bez této operace by \TeX vnechal patřičnou vertikální mezeru, což my ale nechceme. K registru `\everypar` se vrátíme za malý okamžik.

Závěrečným krokem je zajištění, aby se naše makro volalo automaticky na začátku každého odstavce. Využijeme k tomu další mechanismus, který nám \TeX poskytuje. Na začátek odstavce se totiž automaticky vloží obsah registru `\everypar`. Do tohoto registru tedy vložíme token `\iniciala`. Jakmile \TeX přejde do horizontálního režimu, vloží do vstupní fronty obsah registru `\everypar` následovaný textem odstavce. Na začátku fronty je tudíž token `\iniciala`. Ten odebere jako parametr první token odstavce. Bohužel to nemusí být zrovna znak – pokud odstavec nezačne písmenem nebo číslicí, dojde ke katastrofě.

Vraťme se nyní k usazování iniciály s použitím `\vadjust`. Řekli jsme si, že musíme přejít do horizontálního seznamu. V tom okamžiku ovšem \TeX vloží do fronty obsah registru `\everypar`, který skrývá makro pro sazbu iniciály. Pokud bychom na začátku `\vadjust` registr `\everypar` nevyprázdnili, snažil by se \TeX vyzáčet iniciálu v iniciále v iniciále v iniciále... Tento nekonečný cyklus by skončil zahlcením paměti. Navíc je prvním tokenem tohoto odstavce `\noindent`, což by způsobilo další chyby. Vyprázdnění `\everypar` provádíme lokálně uvnitř skupiny, aby \TeX opět použil makro pro sazbu iniciály v dalším odstavci.

Obvykle nebudeme sázet iniciálu na začátku každého odstavce. Problém s `\everypar` nám tím odpadne. Častěji budeme vyžadovat iniciálu na začátku kapitoly. Nebudeme tedy definovat prostředí `inicialy`, ale makro `\iniciala` bude posledním příkazem v makru `\chapter`. Zde nás čeká jiná obtíž: pokud pod příkazem `\chapter` vynecháme volný řádek, bude parametrem makra `\iniciala` token `\par`. Vyřešíme to tím, že posledním příkazem makra `\chapter` bude `\inic` s následující definicí:

```
\def\inic#1{\ifcat #1A\def\next{\iniciala #1}\else
\let\next\inic\fi
\next}
```

Tato definice zajistí, že všechny tokeny kromě písmen s kategorií 11 budou mlčky ignorovány. Nejsou tím ovšem odstraněny všechny potenciální problémy. Mohou nastat případy, kdy je nutné před začátkem prvního odstavce kapitoly provést nějakou akci. Pak budeme muset v makru `\inic` uvést vhodný příkaz `\if`, který potřebnou činnost provede.

Skutečná makra pro sazbu iniciál budou muset být poněkud složitější. Tvar písmen se liší a často bude nutné upravit odstup odstavce od iniciály. Také písmeno **V** požaduje jiné zacházení. Jak se můžete přesvědčit výše, nevypadá díra vedle této iniciály nijak hezky. Navíc iniciála nemusí být jen písmeno z nějakého fontu. Můžeme použít obrázek buď naskenovaný nebo vektorový. Již dříve jsme zmínili Zpravodaj č. 1 z roku 1996. Libor Sýkora použil iniciály v článku Vkládání obrázků do \LaTeX u. Každou iniciálu měl ovšem v jiné velikosti, což bylo redakčním zásahem sjednoceno. Tím se vysvětlí, proč nejsou všechny parametry makra použity (jinak bych musel zasahovat přímo do textu článku). Iniciala ovšem není na začátku každého odstavce, takže se nekládá pomocí `\everypar`. Definici upraveného makra, které bylo v článku použito, uvádíme níže bez vysvětlení.

```
\newlength{\inicl}
\newbox\inibox
\newcommand\inic[2]{%
\setbox\inibox=\hbox{\epsfysize=30dd \relax
\epsffile{#1#1.eps}}%
```

```

\inicl=\wd\inibox \advance\inicl .5em
\noindent \hangafter\minus3 \hangindent\inicl
\vadjustf{\smash{\vbox{\noindent
                \hbox to0pt{\lower20.5dd\box\inibox\hss}}}}%
\hspace*{\minus4pt}\ignorespaces}

```

Další příklady použití parametrů `\leftskip` a `\rightskip` (například pro sazbu veršů) budou uvedeny v příštím pokračování \LaTeX ové kuchařky.

29. Literatura

První citace odkazuje na příklad, který je v tomto článku popisován. V dalších knihách je uveden podrobný popis algoritmů řádkového zlomu.

1. Libor Sýkora: *Vkládání obrázků do \LaTeX u*. Zpravodaj Československého sdružení uživatelů \TeX u, **6** (1), 37–43 (1996).
2. D. E. Knuth: **The \TeX book**. Addison Wesley, Reading 1984. ISBN 0-201-13448-9.
3. P. Olšák: **\TeX book naruby**. Konvoj, Brno 1997. ISBN 80-85615-64-9. Elektronická verze je k dispozici na <http://math.feld.cvut.cz/olsak/tbn/>.

Zdeněk Wagner
wagner@mbox.cesnet.cz

TUG 2000: Call for papers

The 21st annual meeting of the \TeX users Group is to be held in Oxford, UK between Sunday 13th August and Wednesday 16th August 2000.

Details of the conference are available at the conference Web site:

<http://tug2000.tug.org>

Papers are invited on any subject relevant to \TeX .