

Zpravodaj Československého sdružení uživatelů TeXu

Pavel Stríž

Proměnné záhlaví a zápatí

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 17 (2007), No. 1, 31–59

Persistent URL: <http://dml.cz/dmlcz/150026>

Terms of use:

© Československé sdružení uživatelů TeXu, 2007

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>

Tento článek se zabývá sazbou záhlaví a zápatí. Popisuje základní možnosti a způsoby, jak je vysázet. K tomu využívá ve většině ukázek standardního balíčku **fancyhdr**. Ukazuje možnosti přípravy proměnných prvků, které jsou obvykle závislé na aktuální stránce. V případě, že jsou proměnné prvky navíc navzájem provázané, článek zmiňuje metodu, která část nebo celý $\text{\TeX}$ ový dokument vygeneruje za pomoci PHP a MySQL.

Závěrečná ukázka, její zdrojový kód a všechny očíslované zdrojové partie jsou nezávisle publikovány na serveru <http://bulletin.cstug.cz/>.

Klíčová slova: Sazba a proměnlivost v záhlaví a zápatí, balíček **fancyhdr**, programování v PHP a MySQL.

This article deals with the typesetting of headings and footings. It describes basic opportunities and ways how to typeset them. It uses the standard package **fancyhdr** nearly on all examples. It shows the ways how to prepare variable objects which are usually page dependent. In case variable objects are dependent itself in addition, the article introduces a method which generates part or whole $\text{\TeX}$ document using PHP and MySQL tools.

Final example, its source code and all numbered source codes are independently published on <http://bulletin.cstug.cz/>.

Key words: Headings and footings typesetting and its variability, **fancyhdr** package, PHP and MySQL programming.

Предметом настоящей статьи является (типографский) набор колонтитулов. Описаны основные варианты набора (верхних и нижних) колонтитулов. Почти все примеры в тексте пользуются стандартным пакетом **fancyhdr**. Показаны способы определения изменяемых объектов обычно зависящие от содержания страницы. В случае зависимости изменяемых объектов от других изменяемых объектов, настоящая статья предоставляет методы генерирования частей или полного $\text{\TeX}$ документа с помощью средств PHP и MySQL.

Заключительный пример, его исходный код и все остальные нумерованные исходные коды опубликованы на www.cstug.cz.

Ключевые слова: Набор колонтитулов и его изменяемость, **fancyhdr** пакет, программирование на PHP и MySQL.

1. Seznámení s problémem

Se zajištěním proměnného záhlaví a zápatí, tedy s tímto problémem, jsem se setkal při výpomoci nad skripty psanými v programu Microsoft Word[®], kde byl požadavek (spíše přání) mít odděleně vysázeno na každé straně v zápatí jiný citát a jeho autora. To šlo pomocí nastavení pevných oddílů v závěru publikace zrealizovat, ale při požadavku na změnu pořadí, přidání, vypuštění citátů, nebo při zásazích do textu by nastal ruční blázelec.

Není proto úplně na škodu nahlédnout na nástroje T_EXu a mít vhodný nástroj po ruce na podobné situace do budoucna.¹

2. Sazba záhlaví a zápatí

Nejčastěji pracujeme s `\markboth`, který je vhodný i pro zrcadlovou sazbu, a za pomoci `\def\ps@` a příkazů `\def\@oddhead`, `\def\@evenhead`, `\def\@oddfoot` a `\def\@evenfoot`. Lze použít i balíček `fancyhdr` (nebo modifikovanou alternativu `nccfancyhdr`), se kterým můžeme bez problémů rozmístit objekty do dvanácti směrů při zrcadlové sazbě. Pokud bychom potřebovali rozmístit ještě více objektů, lze to zařídit dle našich vlastních představ přes příkaz `\makebox`.

U knih využíváme různě naformátované sekce a podsekce s číslováním nebo bez něj [4, str. 113 – 115], a to při zrcadlové sazbě. U zrcadlové sazby nastavujeme tzv. zrcadlo sazby. To má plně využít potenciálu oboustranného tisku a zajistit maximální čitelnost dokumentu. Pojem zrcadlová sazba může vyznít i trochu jinak (viz grafická vsuvka, str. 45).

U slovníků se lze setkat v záhlaví vlevo a vpravo s prvním a posledním pojmem vyskytujícím se na dané straně [4, str. 115]. Typograficky obdobnou situaci řešil i pan Olšák na přednášce v Brně v listopadu roku 2003 při sazbě inzertní přílohy časopisu DoteK [2].

S proměnným charakterem v záhlaví a zápatí se lze setkat v podsekcí *A movie* [4, str. 121]. Zde se předpokládá již předpřipravená sada obrázků a při odstranění jednoho z nich se sada obrázků za tímto vyřazeným obrázkem již nepřepočítává.

Pokusil jsem se s kolegou nastínit proměnné záhlaví a zápatí již dříve [5].

2.1. Style headings a myheadings

Začneme běžně zažitou ukázkou:

```
1 \documentclass[twoside]{article}
2 \pagestyle{headings}
```

¹ Motto *To i my jsme zamlada psávali jen tím uživatelsky přívětivým stylem empty ...*

Mé poděkování patří pánům Jozefu Říhovi, Radku Vicherkovi, Miroslavu Červenkovi a Haně a Martinu Strážovským za pečlivé pročtení rukopisu. Poděkování patří i panu Igoru Novozámskému za vytvoření zakázkové grafické vsuvky.

```

3 \begin{document}
4 %...                               Zde patří text těla dokumentu.
5 \end{document} \bye

```

Tímto si lze zajistit vhodné záhlaví a zápatí, kterým téměř nikdo nepohrdne. Pokud si čtenář není u editace jistý zdrojovým kódem, např. u `article.cls`, může využít o něco méně elegantní metodu za pomoci `myheadings`:

```

6 \documentclass[twoside]{article}
7 \pagestyle{myheadinds}
8 \begin{document}
9 \section[text_1]{text_2}\markboth{text_2}{text_5}
10 %...                               Běžný text.
11 \section[text_3]{text_4}\markboth{text_4}{text_6}
12 %...                               Běžný text.
13 \end{document} \bye

```

Bez hlubších znalostí pak lze při každé změně např. `\section` a `\subsection` měnit texty i v parametrech `\markboth`. Tím se můžeme dostat náročnější cestou k první ukázce. Vhodné je to však ve chvíli, kdy nechceme použít automatizovaných prvků. Anebo přesně obráceně, kdy chceme vysázet, např. do záhlaví, dohromady část, sekci i podsekci. Je to typografická zajímavost a čtenář se s tím setká jen málokdy. Minimálně to upoutá pozornost (viz podsekke 2.6).

2.2. Styl myfootings

Neodvracejme však naši tvář od automatizovaných prvků. Nejednomu začínajícímu $\text{\TeX}$ istovi se může zdát, že zápatí je příliš nevytížené a chce si záhlaví se zápatím u `myheadings` prohodit. Pak stačí jen vyhledat `article.cls`, konkrétně definování `myheadings` a `myfootings` si připravit takto:

```

14 \def\ps@myfootings{%
15     \let\@oddhead\@empty\let\@evenhead\@empty
16     \def\@evenfoot{\thepage\hfil\slshape\leftmark}%
17     \def\@oddfoot{\slshape\rightmark}\hfil\thepage}%
18     \let\@mkboth\@gobbletwo
19     \let\sectionmark\@gobble
20     \let\subsectionmark\@gobble }

```

Je možné si styl přidat do `article.cls`, do vlastního souboru, nebo si tyto řádky nechat v preambuli dokumentu.

Není úplně na škodu se také podívat v `article.cls` na přípravu `headings`, protože někdy stačí upravit jen málo, abychom byli se svým záhlavím a zápatím spokojeni. Nechť se začínající $\text{\TeX}$ ista nenechá odradit příkazem `\if@twoside`, protože to je klíč k různému záhlaví a zápatí u běžné a zrcadlové sazby.

Pokud chceme mít náš dokument z pohledu záhlaví a zápatí prázdný, použijeme `\pagestyle{empty}`. Pokud bychom chtěli zvolit konkrétní styl, byť právě na jedné straně (nyní stylem `empty`), použijeme `\thispagestyle{empty}`.

2.3. Balíček fancyhdr

U tříprvkových záhlaví a zápatí, kde jsou levé a pravé položky proměnných prvků nesymetrické, je střed (při použití `\hfill`) vychýlen a s tím nejsme spokojeni jako doposud. Buď začneme s přeměřováním objektů nebo využijeme balíčku `fancyhdr`, který si zajistíme v preambuli pomocí:

```
21 \usepackage{fancyhdr}
22 \pagestyle{fancy}
```

Je dobré si pročíst dokumentaci a zdrojový soubor k tomuto balíčku v podobě `fancyhdr.sty` (nebo `nccfancyhdr.sty`). Lze totiž načerpat řadu podnětů ke zpracování záhlaví a zápatí. Balík `fancyhdr` nám umožňuje při zrcadlové sazbě využít 12 míst a pomocí několika příkazů rozmístit prvky do záhlaví a zápatí našeho dokumentu (viz nákres níže).

```
23 \fancyhf{}
24 \renewcommand{\headrulewidth}{1.2pt}
25 \renewcommand{\footrulewidth}{0.8pt}
26 \fancyhead[LE,RO]{\slshape \rightmark}
27 \fancyhead[LO,RE]{\slshape \leftmark}
28 \fancyfoot[C]{\thepage}
```

Pomocí řádku 23 si vyčistíme naše záhlaví a zápatí, řádky 24 a 25 si nastavíme linky a v dalších třech řádcích říkáme co chceme mít zobrazeno. Jsme tam, kde jsme začali – u volby `headings`. Ovšem s tou výhodou, že můžeme velmi nenásilně operovat na celé dvoustraně, a to jak v záhlaví, tak v zápatí. Velmi rychle se můžeme přesvědčit i o nenásilné práci na více řádcích pomocí zalamování `\\`. Nastavení pracovních rozměrů záhlaví a zápatí můžeme nastudovat např. z [1, str. 110–112].

	přes <code>\hfill</code>	vlevo (left)	uprostřed (center)	vpravo (right)
	přes <code>\makebox</code>	vlevo (left)	uprostřed (center)	vpravo (right)
Lichá strana (odd)	<code>\fancyhead</code>	LO	CO	RO
		TĚLO DOKUMENTU		
	<code>\fancyfoot</code>	LO	CO	RO
		⋮		
Sudá strana (even)	<code>\fancyhead</code>	LE	CE	RE
		TĚLO DOKUMENTU		
	<code>\fancyfoot</code>	LE	CE	RE

Pokud bychom chtěli nastavit okraje záhlaví a zápatí delší než okraje těla dokumentu, nahlédneme na [4, str. 12 a 13].

2.4. Speciální rozmístění

Pokud chceme rozmístit objekty velmi přesně, a to nezávisle na předchozím sázeném textu (viz pokus o vycentrování v záhlaví nákresu výše), můžeme v záhlaví a zápatí při využití balíčku `fancyhdr` zkusit tuto alternativu:

```
29 \fancyhf{}
30 \newcommand{\vboxu}[1]{\makebox[0pt][l]{#1}}
31 \fancyhead[l]{\vboxu{\thepage\ text poměrně dlouhý,
32     mnohem delší než ten v zápatí}\hspace{9cm}\text}
33 \fancyfoot[l]{\vboxu{\thepage\ krátký text}\hspace{9cm}\text}
```

Tímto obratem na řádcích 29–33 jsme získali možnost přesně umístit text nebo grafiku tak, aby posuny nebyly ovlivněny délkou sazby textu nebo grafiky na tom samém řádku. Takto můžeme dle svého uvážení rozmístit typografické prvky. Pokud zvolíme vhodnou proměnnou místo devíti centimetrů, lze rozmístění řídit poměrně rozumně.

Na tomto příkladě je již vidět, že ačkoliv dokážeme automaticky rozmístit prvky, zatím nemáme nástroj na editaci jednotlivých položek dostatečně flexibilní vůči každé straně dokumentu. Doposud jsme jen zvažovali jisté změny v záhlaví a zápatí na úrovni `\section`, `\subsection` a dalších dokument zpřehledňujících prvcích.

2.5. Změna stylu

Podobně jako `myfootings` si můžeme připravit celou řadu zajímavých stylů a v dokumentu se mezi nimi přepínat. Nejčastěji potřebujeme v publikaci tři až čtyři styly. Na úvodní strany knihy se dává jeden styl (čísla stránek římskými číslicemi). Dokument je sazen v dalším stylu (nejčastěji `headings` nebo `myheadings`). U příloh a dodatků se setkáváme s jednodušším stylem bez čísel stran. U první strany a občas i u rozsáhlejších ukázek dáváme styl `empty`.

Přepínání realizujeme v textu následujícím způsobem:

```
34 \pagestyle{empty}           %Zavedení prvního stylu.
35 %...                         Text dokumentu.
36 \pagestyle{headings}       %Zavedení druhého stylu.
37 %...                         Text dokumentu.
38 \thispagestyle{myfootings} %Třetí styl. Určen pro jednu stranu.
39 %...                         Text dokumentu. Pokračuje se stylem headings.
40 \pagestyle{empty}           %Zpátky na první styl. Atd.
```

Pokud si nechceme definovat styly, můžeme si nadefinovat přímo příkazy:

```

41 \def\stylA{\fancyhf{}}
42 \def\stylB{\fancyhf{} \fancyhead[L]{Konference}
43   \fancyhead[R]{\thepage} \fancyfoot[L]{Místo a čas}}
44 \def\stylC{\thispagestyle{fancy} \fancyhf{}
45   \fancyhead[C]{\partname ~\thepart ~--- \leftmark} }

V textu se mezi styly balíčku fancyhdr pohybuje podobně:
46 \stylA                                %Zavedení stylu A.
47 %...                                  Text dokumentu.
48 \stylB                                %Zavedení stylu B.
49 %...                                  Text dokumentu.
50 \stylC                                %Zavedení stylu C. Určen pro jednu stranu.
51 %...                                  Text na jedné straně a pak se pokračuje stylem B.
52 \stylA                                %Zpátky na styl A. Atd.

```

2.6. Řízené změny stylu

Přepínat se jako v předchozí podsekcí není po určité době a častých změnách vždy vhodná možnost. Ukažme si způsob jak využít automatických čítačů na třech úrovních nadpisů. S každou ze zmíněných čtyř variant (vysázeno jako bloky) lze pracovat relativně samostatně.

Nejjednodušší variantou je zobrazit aktuální nadpis. Na aktuální straně se vysází vždy ten poslední nadpis. Rozšířili jsme tedy `\part` na `\urovenA`, `\section` na `\urovenB` a `\subsection` na `\urovenC`.

```

53 \def\urovenA#1{\part{#1}\fancyhf{}}
54   \fancyhead[C]{\partname\ \thepart\ \emph{#1}} }
55 \def\urovenB#1{\section{#1}\fancyhf{}}
56   \fancyhead[C]{Sekce \thesection\ \emph{#1}} }
57 \def\urovenC#1{\subsection{#1}\fancyhf{}}
58   \fancyhead[C]{Podsekce \thesubsection\ \emph{#1}} }

```

Strana 1	záhlaví	Podsekce 1.1 <i>Historické poznámky</i>
		Část I – Umělé neuronové sítě
		1 Kohonenovy sítě
		1.1 Historické poznámky
	zápatí	PRÁZDNÉ
		:
Strana n	záhlaví	Podsekce 3.7 <i>Citované zdroje</i>
		3.5 Průmyslové využití
		3.6 Příklady a cvičení
		3.7 Citované zdroje

zápatí	PRÁZDNÉ
--------	---------

Druhou variantou je sazba posledního nadpisu na stránce, ale v úplné formě. Vysází se název úrovně nadpisu, její hodnota a její název. U úrovně *B* se zobrazí i úroveň *A* a u úrovně *C* se zobrazí *A* i *B*. V zápatí se zobrazí zkrácená varianta jen za použití arabských číslic. Zvolíme tuto cestu:

```

59 \def\urovenA#1{\part{#1}\def\textA{\emph{#1}}\fancyhf{}
60   \fancyhead[C]{\partname\ \thepart\ \textA}
61   \fancyfoot[C]{\thepart} }
62 \def\urovenB#1{\section{#1}\def\textB{\emph{#1}}\fancyhf{}
63   \fancyhead[C]{\partname\ \thepart\ \textA\ ---\ Sekce\ %
64     \thesection\ \textB}
65   \fancyfoot[C]{\thepart\,--\,\thesection} }
66 \def\urovenC#1{\subsection{#1}\def\textC{\emph{#1}}\fancyhf{}
67   \fancyhead[C]{\partname\ \thepart\ \textA\ ---\ Sekce\ %
68     \thesection\ \textB\ ---\ Podsekce\ \thesubsection\ \textC}
69   \fancyfoot[C]{\thepart\,--\,\thesection\,--\,\thesubsection}}
```

Strana 1	záhlaví	Část I <i>JAVA</i> – Sekce 1 <i>Programátorské milníky</i> – – Podsekce 1.1 <i>Konec 20. století</i>
		Část I – JAVA 1 Programátorské milníky 1.1 Konec 20. století
	zápatí	I – 1 – 1.1
⋮		
Strana <i>n</i>	záhlaví	Část IX <i>Síťové protokoly</i> – Sekce 11 <i>Rodina TCP</i> – – Podsekce 11.1 <i>TCP/IP</i>
		Část IX – Síťové protokoly 11 Rodina TCP 11.1 TCP/IP
	zápatí	IX – 11 – 11.1

Alternativně vůči první variantě zmíníme sazbu nadpisů s tím rozdílem, že je vysázen nadpis na zkoumané straně první:

```

70 \newcount\stopka\newcount\posledni\posledni=0
71 \def\textA{}\def\textB{}\def\textC{}
72 \newcounter{hodnotaA}\newcounter{hodnotaB}\newcounter{hodnotaC}
73 \def\urovenA#1{\part{#1}
```



```

74 \ifnum\posledni=\thepage\else
75 \def\textA{\emph{#1}}\setcounter{hodnotaA}{\value{part}}
76 \fancyhf{}\fancyhead[C]{\partname\ \Roman{hodnotaA}\ \textA}
77 \posledni=\value{page}\fi }
78 \def\urovenB#1{\section{#1}
79 \ifnum\posledni=\thepage\else
80 \def\textB{\emph{#1}}\setcounter{hodnotaB}{\value{section}}
81 \fancyhf{}\fancyhead[C]{Sekce\ \thehodnotaB\ \textB}%
82 \posledni=\value{page}\fi }
83 \def\urovenC#1{\subsection{#1}
84 \ifnum\posledni=\thepage\else\def\textC{\emph{#1}}
85 \setcounter{hodnotaC}{\value{subsection}}
86 \fancyhf{}\fancyhead[C]{Podsekce\ \thehodnotaC\ \textC}%
87 \posledni=\value{page}\fi }

```

Strana 1	záhlaví	Část I <i>Umělé neuronové sítě</i>
		Část I – Umělé neuronové sítě
		1 Kohonenovy sítě
		1.1 Historické poznámky
	zápatí	PRÁZDNÉ
		:
Strana <i>n</i>	záhlaví	Podsekce 3.5 <i>Průmyslové využití</i>
		3.5 Průmyslové využití
		3.6 Příklady a cvičení
		3.7 Citované zdroje
	zápatí	PRÁZDNÉ

Čtvrtou a poslední variantou řízené změny zmíníme rozšířenou modifikaci druhé varianty. V záhlaví je úplný výpis nadpisu na prvním místě aktuální strany a v zápatí je zkrácený zápis. Opět pracujeme se třemi úrovněmi a zavádíme několik pracovních proměnných.

```

88 \newcount\stopka\newcount\posledni\posledni=0
89 \def\textA{}\def\textB{}\def\textC{}
90 \newcounter{hodnotaA}\newcounter{hodnotaB}\newcounter{hodnotaC}
91 \def\urovenA#1{\part{#1}\def\docasneA{\emph{#1}}
92 \ifnum\posledni=\thepage\else
93 \def\textA{\emph{#1}}\setcounter{hodnotaA}{\value{part}}
94 \fancyhf{}\fancyhead[C]{\partname\ \Roman{hodnotaA}\ \textA}
95 \fancyfoot[C]{\Roman{hodnotaA}}

```

```

96     \posledni=\value{page}\fi }
97 \def\urovenB#1{\section{#1}\def\docasneB{\emph{#1}}
98     \def\textA{\docasneA}\setcounter{hodnotaA}{\value{part}}
99     \ifnum\posledni=\thepage\else
100     \def\textB{\emph{#1}}\setcounter{hodnotaB}{\value{section}}
101     \fancyhf{}\fancyhead[C]{\partname\ \Roman{hodnotaA}\ %
102     \textA\ ---\ Sekce\ \thehodnotaB\ \textB}%
103     \fancyfoot[C]{\Roman{hodnotaA}\,--\,\thehodnotaB}
104     \posledni=\value{page}\fi }
105 \def\urovenC#1{\subsection{#1}
106     \def\textA{\docasneA}\setcounter{hodnotaA}{\value{part}}
107     \def\textB{\docasneB}\setcounter{hodnotaB}{\value{section}}
108     \ifnum\posledni=\thepage\else\def\textC{\emph{#1}}
109     \setcounter{hodnotaC}{\value{subsection}}
110     \fancyhf{}\fancyhead[C]{\partname\ \Roman{hodnotaA}\ %
111     \textA\ ---\ Sekce\ \thehodnotaB\ \textB\ ---\ Podsekce\ %
112     \thehodnotaC\ \textC\ \fancyfoot[C]{%
113     \Roman{hodnotaA}\,--\,\thehodnotaB\,--\,\thehodnotaC}
114     \posledni=\value{page}\fi }

```

Strana 1	záhlaví	Část I <i>JAVA</i>
		Část I – JAVA 1 Programátorské milníky 1.1 Konec 20. století
	zápatí	I
Strana <i>n</i>		:
	záhlaví	Část IX <i>Síťové protokoly</i>
		Část IX – Síťové protokoly 11 Rodina TCP 11.1 TCP/IP
	zápatí	IX

U třetí a čtvrté varianty jsme u pomocných čítačů nepoužili závislou proměnnou `section`, a tak je podsekce sázena jen jako prostý čítač, např. z „Podsekce 11.1 *TCP/IP*“ se vysází „Podsekce 1 *TCP/IP*“. Pokud chceme původní formát, jedna z možností je smazat řádky 112 a 113 a rozšířit je na tyto tři:

```

115     \thehodnotaB .\thehodnotaC\ \textC\ \fancyfoot[C]{%
116     \Roman{hodnotaA}\,--\,\thehodnotaB\,--\,
117     \thehodnotaB .\thehodnotaC}

```

Používaná varianta je sazba nadpisů v záhlaví a zápatí, jejichž text se první (poslední) vyskytuje na aktuální straně. Tohle úspěšně vyřešil pan Olšák [2].

Lze zvolit alternativu pomocí ukládání začátků a konců nadpisů buď přes citování (s uloženým odkazem na stranu lze pracovat jako s číslem) nebo přes ukládání do dvou pracovních proměnných. Tyto metody považuji za nehotové a dovolím si je neuvádět. Ať laskavý čtenář promine, pokud právě tyto alternativně vyřešené partie v článku očekával.

3. Práce s `\thepage`

Proměnná `\thepage` vzniká automaticky po nastavení čítače `page`. Velmi často je třeba změnit jeho formát, a to zařídíme takto:

```
118 \renewcommand\thepage{\roman{page}}
```

Pokud se chceme dostat zpět k zápisu arabskými číslicemi, tak přes:

```
119 \renewcommand\thepage{\arabic{page}}
```

Některým $\text{\TeX}$ istům vyhovuje více alternativa `\pagenumbering{arabic}`. Vedle `arabic` a `roman` lze použít `Roman`, `alph` nebo `Alph`. Vlastní formát čísla strany lze připravit příkazy `\ifnum` nebo `\ifcase` (viz podsektce 3.1 a 3.2).

V další části textu budeme již pracovat s `\ifodd` a `\ifcase`. Toto je jedna z cest, jak si zajistit vhodný formát čítače `page`. Druhou cestou nám může být práce přímo s `\value{page}`, což je na formátu nezávislé.

3.1. Práce s `\ifodd` a `\ifnum`

Chceme-li vysázet text nebo vykreslit grafiku s ohledem na to, jsme-li na sudé nebo liché stránce, učiníme to takto:

```
120 Jsem na \ifodd\value{page}
```

```
121 liché \else
```

```
122 sudé \fi straně.
```

Výpis sudé/liché stránky lze přes novou proměnnou zařídit i pomocí příkazu `\ifnum` touto cestou:

```
123 \newcount\citatel \citatel=\thepage
```

```
124 \divide\citatel by 2 \multiply\citatel by 2
```

```
125 \ifnum\citatel=\value{page} Sudé, dvěma dělitelné \else
```

```
126 Liché, dvěma nedělitelné \fi číslo strany.
```

3.2. Práce s `\ifcase`

Pokud chceme do textu vypsát na které jsme straně, použijeme `\ifcase` takto:

```

127 \ifcase\value{page}
128 Jsem na nulté\or
129 Jsem na první\or
130 Jsem na druhé\else
131 Nejsem ani na nulté, první nebo druhé
132 \fi straně.

```

Pokud používáme formát `arabic`, můžeme si dovolit nahradit řádek 127 za `\ifcase\thepage`. Je dobré si uvědomit, že první záznam patří nule. Takový záznam můžeme přeskočit při změně řádku 128 na `\relax\or` nebo jen `\or`.

3.3. Posun v `\ifcase`

Uvedeme si jednoduchý příklad, v němž chceme slovně vypsát na které jsme straně, ale vůči kapitole, která je hluboko za nultou stranou. Pak je méně efektivní vypisovat sérii příkazů `\or` vedle sebe.

Můžeme použít učebnicového příkladu pana Olšáka [3, str. 375, `\ifcase`] za pomoci nové proměnné:

```

133 \newcount\tatostrana \newcount\odskok
134 \tatostrana=\value{page}

```

Tím si do proměnné `\tatostrana` uložíme začátek (např. kapitoly), a kdekoliv v textu pak můžeme operovat následujícím způsobem:

```

135 \odskok=\value{page} %Následují a/nebo předcházejí jiné strany.
136 \advance\odskok by -\tatostrana \advance\odskok by 1
137 \ifcase\odskok\or
138 První \or Druhá \or Třetí \else
139 Čtvrtá nebo vyšší \fi strana této kapitoly.

```

Můžeme však zasáhnout přímo do čítače `page`. Ten upravíme před a po `\ifcase`. Hodnota čítače mimo `\ifcase` zůstane stejná; dle svých zásahů však pracujeme s upravenou hodnotou tohoto čítače. Realizujeme to např. takto:

```

140 \addtocounter{page}{-6}
141 \ifcase\value{page}
142 Nultý \or První \or Druhý \or Třetí \else Jiný \fi záznam.
143 \addtocounter{page}{6}

```

3.4. Uvnitř záhlaví a zápatí

Nyní jsme se dostali do velice příjemné situace, kdy můžeme uvnitř záhlaví a zápatí použít `\ifcase`, a to přímo ve vztahu k čítači `page`. Obzvláště přes balíček `fancyhdr` můžeme operovat i s jinými čítači velmi nenásilně.

Příklad. Chceme ve středu zápatí mít informaci, je-li strana menší než 100, přesně rovna 100 nebo je-li strana větší než 100.

Pomocí `\ifcase` bychom ke kýženému výsledku také dospěli, ale použijeme přímo `\ifnum`.

```
144 \fancyfoot[C]{\ifnum\value{page}<100 Jsem pod stranou sto.\else
145 \ifnum\value{page}=100 Jsem na straně sto. \else
146 Jsem na straně vyšší než sto. \fi \fi }
```

Pokud bychom všechny „100“ a „sto“ nahradili za proměnné, tak máme opět velmi flexibilní řešení. Úkol splněn.

Příklad. Chceme mít v záhlaví u vnitřního okraje dle čísla stránky v zrcadlové sazbě název měsíce.

Realizace může vypadat například takto:

```
147 \fancyhf{}
148 \fancyhead[LO,RE]{\ifcase\value{page}\or
149 Leden\or Únor\or Březen\or Duben\or Květen\or Červen\or
150 Červenec\or Srpen\or Září\or Říjen\or Listopad\or Prosinec
151 \else \relax \fi }
```

Velmi rychle lze změnit popisky na libovolný jiný jazyk. Úkol splněn.

Najdeme-li vhodnou metodu generování zdrojového kódu pro záhlaví a zápatí, budeme mít zajištěno jak automatické přechíslování, tak čistý $\text{\TeX}$ ový dokument. K tomuto bodu se ještě vrátíme.

4. Hrátky s grafikou

U sazby grafiky nesmíme zapomenout uvést `\usepackage{graphicx}` v preambuli dokumentu. Osvědčila se mi práce v prostředí `picture`.

4.1. Sazba v oblasti záhlaví a zápatí

Uvedme si ukázkou sazby textu a grafiky (vkládaný soubor `obr1` reprezentuje jednoduchou dekoraci). Horizontální linka nám v naší ukázce odděluje záhlaví od těla dokumentu.

```
152 \fancyhead[L]{\begin{picture}(0,0)
153 \put(0,0){\rotatebox[origin=r]{180}{
154 \includegraphics[height=0.75em,width=5em]{obr1}
155 text TEXT text \rotatebox[origin=r]{180}{
156 {\includegraphics[height=0.75em,width=5em]{obr1}}
157 } } \end{picture} }
```

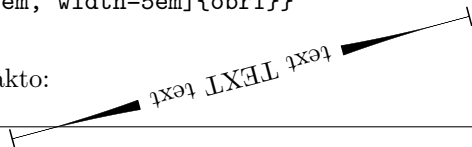
Příklad může vypadat například takto:

4.2. Sazba mimo oblast záhlaví a zápatí

I zde nám v příkladu horizontální linka odděluje záhlaví od těla dokumentu. Při změně parametrů na umístění u `\put` a změně natočení u `\rotatebox` jsme se octli mimo oblast záhlaví, i když to typograficky stále pod záhlaví patří.

```
158 \fancyhead[L]{\begin{picture}(0,0)
159 \put(170,-15){\rotatebox[origin=r]{195}{
160 \includegraphics[height=0.75em, width=5em]{obr1}
161 text TEXT text \rotatebox[origin=r]{180}
162 {\includegraphics[height=0.75em, width=5em]{obr1}}}
163 } } \end{picture} }
```

Příklad může vypadat například takto:



Nyní tedy můžeme využít celé plochy papíru, především všech okrajů, a nenarušíme tím sazbu v těle dokumentu.

5. Jádرو proměnného záhlaví a zápatí

Jeden z nápadů lze objevit v podsektci *A movie* [4, str. 121]. Někde si však obrázky musíme připravit a k tomu nemusíme použít přímo $\text{\TeX}$. Podobné je to i s textem. Například u citátů, které si průběžně ukládáme do textového souboru, internetové databáze nebo např. do tabulkového procesoru v Open Source balíku OpenOffice.org. Převody mezi těmito třemi softwarovými součástmi nečiní většinou velké potíže.

5.1. Stránkově nezávislé prvky

Pokud se pohybuje na úrovni stránkově nezávislých prvků, můžeme editovat přímo hlavní dokument nebo vhodněji z pohledu přehlednosti editujeme zvláštní soubor, který si do hlavního dokumentu načteme. Načítáme ovšem i partie vztahující se k záhlaví a zápatí, abychom nevkládali soubor do prostředí.

V hlavním dokumentu načteme soubor `zahnavi.tex`, nejlépe v preambuli, pomocí `\input{zahnavi.tex}`. Soubor `zahnavi.tex` může vypadat takto:

```
164 \fancyhead[L0,RE]{\ifcase\value{page}\or
165 3,1\or
166 3,14\or
167 3,141\or
168 3,1415\or
169 3,14159\else
170 $\pi$ \fi }
```

```

171 \fancyhead[LE,R0]{\ifcase\value{page}\or
172 Kolega Hlaváč\or
173 Kolegyně Novotná\or
174 Kolegyně Tichá\or
175 Kolega Kafka\or
176 Kolega Havelka\else
177 --- \fi }

```

Je výhodné mít každou položku na jednom řádku – kvůli poměrně přehledné možnosti editace.

Ačkoliv máme všechny údaje v záhlaví, není mezi těmito údaji souvislost a můžeme každý proměnný prvek editovat zvlášť. Na tuto formu editace stačí editovat přímo tento soubor. Kdyby totiž tyto proměnné prvky byly závislé a chtěli bychom odebrat nebo přehodit údaje v prvním proměnném prvku, museli bychom ty samé kroky udělat i u druhého proměnného prvku. To při rozsáhlejších úpravách příjemné již není.

Přímá editace souboru `zahnavi.tex` je výhodná tam, kde jsou proměnné prvky stránkově nezávislé nebo je proměnný prvek jen jeden.

5.2. Stránkově závislé prvky

Za stránkově závislé prvky považujeme takové prvky, které při editaci prvního prvku dávají do souvislosti položky z druhého prvku, a toto přiřazení je z jakéhokoliv hlediska nepřijatelným. Za typické odděleně sázené závislé prvky uveďme: obrázek auta a jeho SPZ, autora příspěvku a jeho univerzitu, citát a jeho autora, citát a původ citátu, úryvek knihy a název knihy nebo logo univerzity a kontaktní osobu, jméno a příjmení, rodiče a iniciály dítěte, knihu a její ISBN, zemi a její měnu, oficiální jazyk a vlajku země, atd.

Pokud sázíme na jednom místě záhlaví nebo zápatí jméno a příjmení dohromady, jsou sice logicky spojené, ale nejsou sázeny odděleně a tudíž je nepovažujeme za závislé prvky.

Uveďme si konkrétní příklad sazby logicky provázaných prvků sázených odděleně. Mějme dva autory, a to pana Jiřího Hlavsu s článkem *E-learningová omezení* a paní Helenu Mokrou s článkem *Tvorba univerzitní šablony*.

```

178 \fancyhead[C]{\ifcase\value{page}\or
179 Jiří Hlavsa\or
180 Helena Mokrá\else \relax \fi }
181 \fancyfoot[C]{\ifcase\value{page}\or
182 E-learningová omezení\or
183 Tvorba univerzitní šablony\else \relax \fi }

```

Pokud u prvního proměnného prvku (jméno autora) chci tyto dva autory prohodit, dostáváme:

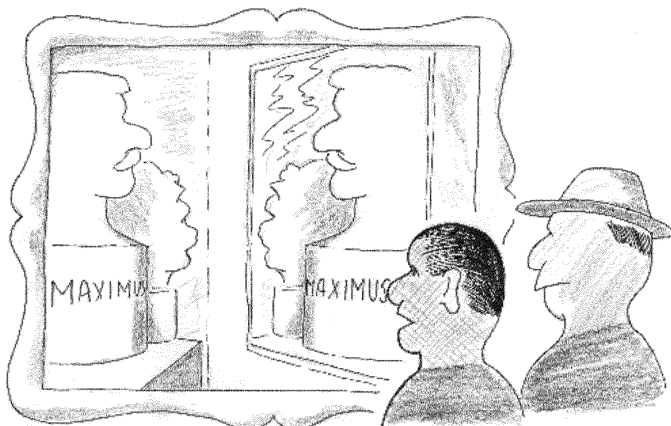
```

184 \fancyhead[C]{\ifcase\value{page}\or
185 Helena Mokrá\or                                %Zde nastala změna.
186 Jiří Hlavsa\else \relax \fi }                    %Zde nastala změna.
187 \fancyfoot[C]{\ifcase\value{page}\or
188 E-learningová omezení\or
189 Tvorba univerzitní šablony\else \relax \fi }

```

Teď je však naše záhlaví a zápatí chybné, protože paní Mokrá nikdy článek *E-learningová omezení* nenapsala. Podobně pan Hlavsa neposlal příspěvek *Tvorba univerzitní šablony*. Proto také musíme prohodit dvě položky ve druhém proměnném prvku.

Je to řešení spíše nepřijatelné, zvláště když si uvědomíme, že těchto stránkově závislých proměnných prvků můžeme mít na straně poměrně hodně, nepočítaje množství nastalých změn.



Malíř uměl rozhodně zrcadlově kreslit,
ale neuměl zrcadlově psát!

Grafická vsuvka: *Jak na zrcadlovou sazbu?*

6. Generování pomocí PHP a MySQL

Nechme raději soubor `zahnavi.tex` vhodně vygenerovat. Uvedme si ukázkou pomocí programu v PHP zpracovávající data z MySQL, ve které máme uloženy potřebné informace (včetně uložených odkazů na galerie obrázků, rodinných fotek, apod.).

Při přidání, mazání a jiném řazení položek necháme soubor znovu vygenerovat a $\text{\TeX}$ má neustále aktuální vstupní soubor `zahlavi.tex`.

6.1. Program `meetings.php`

Připravíme si soubor `meetings.php` a ten přes webový prohlížeč spustíme. Soubor vypadá např. takto:

```
190 <?php
191 @$spojeni=mysql_connect("localhost","spravce","heslo_spravce");
192 @mysql_select_db("conference");
193 @$result=mysql_query("SELECT * FROM papers");
194 $mp=fopen("zahlavi.tex","w");
195 fwrite($mp,"%Dne: ".date("F d Y H:i:s.",getlastmod())."\r\n");
196 fwrite($mp,"\\fancyhead[R]{\\ifcase\\value{page}\\or\r\n");
197 while($line = mysql_fetch_array($result)) {
198     fwrite($mp,$line["article"]);
199     fwrite($mp,"\\or \r\n"); }
200 fwrite($mp,"\\relax\\else\\relax\\fi }\\r\n");
201 mysql_data_seek($result, 0);
202 fwrite($mp,"\\fancyfoot[R]{\\ifcase\\value{page}\\or\r\n");
203 while($line = mysql_fetch_array($result)) {
204     fwrite($mp,$line["authors"]);
205     fwrite($mp,"\\or \r\n"); }
206 fwrite($mp,"\\relax\\else\\relax\\fi }\\r\n");
207 fclose($mp);
208 @mysql_close($spojeni);
209 echo "Soubor zahlavi.tex byl vygenerován: ";
210 echo date ("F d Y H:i:s.", getlastmod()); ?>
```

Program `meetings.php` se řádkem 191 připojí k nyní fiktivnímu serveru, řádkem 192 k databázi `conference`, a žádá řádkem 193 o informace z tabulky `papers`. Řádkem 195 do souboru `zahlavi.tex` vypíše informace o vzniku tohoto souboru, na řádcích 196–200 generuje záhlaví a na řádcích 202–206 pak generuje zápatí. Další řádky programu jsou uzavírající sekvence a kontrolní výpis pro uživatele o dokončení běhu programu.

Jediným problémem snad může být kódování, ale `ledacos` se dá napravit při využití příkazu `mb_convert_encoding`. Není však problém využít externího převaděče a uzpůsobit kódování výstupu z `MySQL` pro potřeby $\text{\TeX}$ u před vlastním načtením tohoto souboru.

6.2. Výstup z programu meetings.php

Výstupem je soubor `zahlavı.tex`, který může být poměrně velký. To platí i pro náš příklad, a proto uvádíme vždy první a poslední záznam a řádky 214 a 219, a nahrazujeme vnitřní údaje třemi tečkami. Ukázka z databáze o konferenci může vypadat následujícím způsobem:

```
211 %Dne: February 02 2005 05:41:52.  
212 \fancyhead[R]{\ifcase\value{page}\or  
213 Internet Interfaces\or  
214 %... Zde následuje druhý až předposlední záznam.  
215 Exact Procedures : Two Independent Samples\or  
216 \relax\else\relax\fi }  
217 \fancyfoot[R]{\ifcase\value{page}\or  
218 Gustav Horník\or  
219 %... Zde následuje druhý až předposlední záznam.  
220 Saskie Burešová\or  
221 \relax\else\relax\fi }
```

Na stránkách, počínaje první, se v záhlaví vpravo objeví název příspěvku a v zápatí vpravo jméno autora. Na první straně bude uveden příspěvek *Internet Interfaces* od pana Horníka a jako poslední záznam, pokud strana existuje, bude uveden příspěvek *Exact Procedures : Two Independent Samples* od paní Burešové. Pokud je stran dokumentu více než je položek v databázi, tak se provádí `\relax`.

Pokud chceme udělat změny, uděláme je přímo v databázi a necháme znovu vygenerovat `zahlavı.tex` spuštěním `meetings.php`. Tím se nejen berou aktuální údaje, ale uloží se do poznámky i doba, kdy se soubor `zahlavı.tex` vygeneroval.

Přímo v MySQL lze připravit generující skript za pomoci příkazu `concat`. V tabulkových procesorech je na přípravu souboru `zalozka.tex` jednou z alternativ příkaz `concatenate`. Slouží ke spojování různých řetězců v jeden textový, což se může hodit zvláště k napojení příkazu `\or`.

Pokud máme v databázi i texty určené do těla dokumentu, můžeme je pomocí PHP také vygenerovat. Velkým pomocníkem je pak příkaz `TEXu \newpage`.

6.3. Program rozmisti.php

Pomocí nástrojů PHP a MySQL můžeme udělat řadu jiných vzorových řešení. Ukážeme si řešení následující situace. Máme přepsány texty v databázi a chceme je rozmístit tak, aby první položka byla na první straně v záhlaví, druhá v jejím zápatí, třetí na druhé straně v záhlaví, čtvrtá pak v zápatí druhé strany, atd. Podobně uvažujeme o rozmístění do rohů stránek, na okrajích dokumentu ve směru hodinových ručiček, atd.

Rozšíříme a připravíme si program `rozmisti.php` následující cestou:

```

222 <?php @$spojeni=mysql_connect("PC","uzivatel","heslo");
223 @mysql_select_db("umeni");
224 @$result=mysql_query("SELECT * FROM basen001");
225 $mp=fopen("zahnavi.tex","w");
226 fwrite ($mp,"%".date ("F d Y H:i:s.", getlastmod())."\r\n");
227 fwrite ($mp,"\\setlength{\\headheight}{24pt}\\r\n");
228 fwrite($mp,"\\renewcommand{\\headrulewidth}{2.4pt}\\r\n");
229 fwrite($mp,"\\renewcommand{\\footrulewidth}{1.6pt}\\r\n");
230 fwrite($mp,"\\fancyhf{}\\r\n");
231 fwrite($mp,"\\lhead{\\ifcase\\thepage\\or\\r\n");
232 $citac=0;while ($line = mysql_fetch_array($result)) {
233 if($citac==0){
234     fwrite($mp,"\\emph{". $line["radkove"]."}");
235     fwrite($mp,"\\or \\r\n"); $citac++;}
236     else {$citac=0;} }
237 fwrite($mp,"\\relax\\else\\relax\\fi }\\r\n");
238 mysql_data_seek($result, 0);
239 fwrite($mp,"\\lfoot{\\ifcase\\thepage\\or\\r\n");
240 $citac=1;
241 while ($line = mysql_fetch_array($result)) {
242 if($citac==0){
243     fwrite($mp,"\\emph{". $line["radkove"]."}");
244     fwrite($mp,"\\or \\r\n"); $citac++; }
245     else {$citac=0;} }
246 fwrite($mp,"\\relax\\else\\relax\\fi }\\r\n");
247 pclose($mp); @mysql_close($spojeni);
248 echo "Verze ze dne: ".date ("F d Y H:i:s.", getlastmod()); ?>

```

Jistě bychom našli řadu programátorských postupů a fint, jak se s takovým generováním T_EXových zdrojů zabývat i mnohem hlouběji. V tomto programu jsme na řádce 227 navíc nastavili i pracovní velikost záhlaví, protože úryvky básní jsou dvouřádkové.

6.4. Výstup z programu rozmisti.php

Výstupem jsou zdravotní napomenutí z kapitoly *Lokšan kontra Griespek*² a soubor *zahnavi.tex* vypadá takto:

```

249 %February 06 2005 02:34:10.
250 \setlength{\\headheight}{24pt}
251 \renewcommand{\\headrulewidth}{2.4pt}
252 \renewcommand{\\footrulewidth}{1.6pt}

```

² Rudolf Richard Hofmeister. *Břežnické obrázky*. Šolc a Šimáček, Praha 1932.

```

253 \fancyhf{}
254 \lhead{\ifcase\thepage\or
255 \emph{Ledna pouštění nech, ráno\\\ \ \ %
256   traňk pí, potom dobré víno.\ ---}\or
257 \emph{Března střídmy jdi do lázně,\\\ \ \ %
258   pulej pí, a drž se vážně.\ ---}\or %
259 %... Měsíce by zde pokračovaly.
260 \relax\else\relax\fi }
261 \lfoot{\ifcase\thepage\or
262 \emph{Února pust z palce, mej se,\\\ \ \ %
263   hodu, zvěř jez, hry varuj se.\ ---}\or
264 \emph{Dubna přijmi počištění,\\\ \ \ %
265   traňk pí, pouštěj, jez koření.\ ---}\or
266 %... Měsíce by zde pokračovaly.
267 \relax\else\relax\fi }

```

Vysázené záhlaví a zápatí by vypadalo následujícím způsobem:

Strana 1	záhlaví	<i>Ledna pouštění nech, ráno traňk pí, potom dobré víno. —</i>
		TĚLO DOKUMENTU
	zápatí	<i>Února pust z palce, mej se, hodu, zvěř jez, hry varuj se. —</i>
Strana 2	záhlaví	<i>Března střídmy jdi do lázně, pulej pí, a drž se vážně. —</i>
		TĚLO DOKUMENTU
	zápatí	<i>Dubna přijmi počištění, traňk pí, pouštěj, jez koření. —</i>
		⋮

7. Generování záložky na okraji

Tím, že jsme použili PHP, nemůžeme říci, že $\text{\TeX}$ em nelze nic generovat. Jak se budeme moci přesvědčit, tak to lze – samotný $\text{\TeX}$ však pro mě spíše není databázovým systémem na webové záležitosti. Můžete však se mnou nesouhlasit.

Budeme generovat okrajovou záložku tímto způsobem. Nejprve si potřebné údaje při prvních bězích $\text{\TeX}$ u vygenerujeme do souboru. Při závěrečných bězích $\text{\TeX}$ u si tento soubor načítáme a údaje z něj necháme vysázet. Okrajovou záložku budou tvořit části a sekce, a aktuální jedna nebo dvě sekce budou na dvoustraně

orámovány. Sazba se provede vpravo shora dolů. Velmi podobně lze připravit sazbu vlevo na dvoustraně, na každé straně, atd.

7.1. Definování proměnných

Začněme jedinou neautomatizovanou proměnnou `posun`. Pomocí ní `TEX`u říkáme, zda-li jsme v úvodních (někdy jen v prvním) nebo již v závěrečných bězích `TEX`u (obecně můžeme říci: v posledním běhu `TEX`u).

```
268 \newcount\posun
```

```
269 \posun=1 %Implicitně je jedna. Pro vysázení měníme na \posun=0.
```

Pokud nejsme v závěrečném běhu `TEX`u, nechávejme `\posun=1`. Ve chvíli kdy končíme práci na dokumentu, dejme místo jedničky jinou hodnotu, např. `\posun=0`.

Na této proměnné závisí řada dějů. Jedním z nich je otevření a možný zápis do souboru `zalozka.tex`:

```
270 \ifnum\posun=1
```

```
271 \immediate\newwrite\tak%
```

```
272 \immediate\openout\tak zalozka.tex \fi
```

Definujeme proměnnou `\podel` na zjištění sudé/liché strany. Dále si založíme proměnné `stranapr`, `zaznampr` a `aktualni`, pomocí nichž řídíme vlastní proces generování a zápisu do souboru.

```
273 \newcount\podel \newcount\stranapr
```

```
274 \newcount\zaznampr \zaznampr=0
```

```
275 \newcounter{aktualni} \setcounter{aktualni}{1}
```

Jak budeme moci vypořádat dále, první a poslední bloky zápisů jsou rozdílné vůči ostatním. To realizujeme nastavením proměnné `\zaznampr=0`.

7.2. Definování příkazů

Příkaz `\zapsano` slouží k formulaci výstupní procedury:

```
276 \newcommand\zapsano{\Roman{aktualni}--\thesection}
```

Dále potřebujeme rozšířit příkaz `\section`. Tento způsob není nejjednodušší a vyžaduje více zkušeností s `TEX`em.

Lze to obejít příkazem, který bude stát mimo příkaz `\section` a nebude na něm přímo závislý. Nazvěme ho `\gnadpis` a definujme jej tímto způsobem:

```
277 \newcommand\gnadpis{\newpage \ifnum\posun=1%
```

```
278 \stranapr=\value{page}%
```

```
279 \ifodd\stranapr\advance\stranapr by -2
```

```
280 \else\advance\stranapr by -1\fi%
```

```
281 \ifnum\zaznampr=0%
```

```

282 \immediate\write\tak{\string\i fnum\string\t hepage>%
283 \the\stranapr} \zaznampr=1\podel=\value{page}%
284 \ifodd\podel\advance\podel by 1\else\advance\podel by 2\fi%
285 \else%
286 \podel=\value{page}\advance\podel by -1
287 \ifodd\podel\advance\podel by 1\else\advance\podel by 2\fi%
288 \immediate\write\tak{\string\i fnum%
289 \string\t hepage<\the\podel
290 \string\framebox{\zapsano} \string\else \zapsano \ %
291 \string\fi\string\else \zapsano}%
292 \immediate\write\tak{\string\fi}%
293 \immediate\write\tak{\string\i fnum%
294 \string\t hepage>\the\stranapr}
295 \setcounter{aktualni}{\value{part}}
296 \fi \fi }

```

Na řádce 281 rozlišujeme, zda-li se jedná o náš první zápis nebo o další. První zápis formulují řádky 282 a 283, další se řídí řádky 288 až 294. Zápis zpětného lomítka není ideálním řešením, ale splňuje to svůj účel.

Příkaz `\gnadpis` tedy zajišťuje výstup do souboru `zalozka.tex`, a to vyjma závěrečných sekvencí.

Poslední zápis pro testování přítomnosti nadpisu na dvoustraně definujeme zvlášť jako příkaz `\poslednpr`:

```

297 \newcommand\poslednpr{\ifnum\posun=1%
298 \podel=\the\page \advance\podel by 1%
299 \immediate\write\tak{\string\i fnum\string\t hepage<\the\podel
300 \string\framebox{\zapsano}\string\else \zapsano
301 \ \string\fi\string\else \zapsano}%
302 \immediate\write\tak{\string\fi}%
303 \immediate\closeout\tak \fi}

```

Tento příkaz použijeme jen jednou, a to na poslední straně, u které ještě zkoumáme přítomnost nadpisu, a kde povolujeme možnost vysazení záložky. Pokud tento příkaz nepoužijeme, bere se za poslední testovanou stranu poslední strana dokumentu. Pomocí řádky 303 uzavíráme soubor `zalozka.tex`.

7.3. Vygenerovaný soubor `zalozka.tex`

Uvedme si jednu ukázkou po vlastním generování. Na straně 9 máme sekci první, druhou máme na straně 11 a nadpis třetí sekce je na straně 19. Balíkem `fancyhdr` načítaný soubor může již po prvním běhu vypadat např. takto:

```

304 \ifnum\the\page>7
305 \ifnum\the\page<12\framebox{I--1} \else I--1\ \fi\else I--1

```

```

306 \fi
307 \ifnum\thepage>9
308 \ifnum\thepage<20\framebox{I--2} \else I--2 \fi\else I--2
309 \fi
310 \ifnum\thepage>17
311 \ifnum\thepage<22\framebox{II--3}\else II--3 \fi\else II--3
312 \fi

```

U každého `\zapsano` byla zjišťována přítomnost na dvoustraně a zapsána přes příkazy `\ifnum`. Pokud je sázena záložka a zjištěna přítomnost na dvoustraně, `\zapsano` je zarámováno vůči těm, které na dvoustraně nejsou. Poněvadž každá sekce je vysázena na nové straně, musí na straně předchozí být také předchozí sekce. U částí se používá vlastní styl a nemusí nás tedy trápit.

7.4. Styl zalozka

Pro snazší přepínání mezi styly si definujeme styl `zalozka` takto:

```

313 \def\zalozka{\fancyhead[RE]{\thepage}%
314 \fancyhead[RO]{\thepage\ifnum\posun=0%
315 \begin{picture}(0,0)\put(3,-688){%
316 \rotatebox[origin=r]{270}{\input{zalozka.tex}}
317 } \end{picture} \fi } }

```

7.5. Použití v hlavním dokumentu

Přípravu pomocného souboru řídíme příkazem `\posun=1` nebo `\posun=0`. Pokud má být záložka správně vysázena, musí dokument proběhnout $\text{\TeX}$ em minimálně dvakrát. Poprvé a minimálně jednou pro `\posun=1` a podruhé (v závěrečném běhu $\text{\TeX}$ u) a minimálně také jednou pro `\posun=0`.

Dokument může vypadat následujícím způsobem:

```

318 \documentclass[twoside]{article}
319 \usepackage{graphicx} \usepackage{fancyhdr} \pagestyle{fancy}
320 %... Definování všech příkazů z této sekce.
321 %... Nastavení příkazu \posun=1 a na posledním běhu \posun=0.
322 \zalozka %Spuštění námi připraveného stylu.
323 \begin{document}
324 \part{První...}\thispagestyle{empty}
325 \gnadpis\section[Ada]{První...} %V ukázce na str. 9.
326 \gnadpis\section[Eda]{Druhá...} %V ukázce na str. 11.
327 \part{Druhá...}\thispagestyle{empty}
328 \gnadpis\section[Oto]{Třetí...} %V ukázce na str. 19.
329 \poslednipy

```

```

330 \section{Čtvrtá...}           %Tuto sekci nechci mít v záložce.
331 %...                          %Následují další pasáže dokumentu.
332 \end{document} \bye

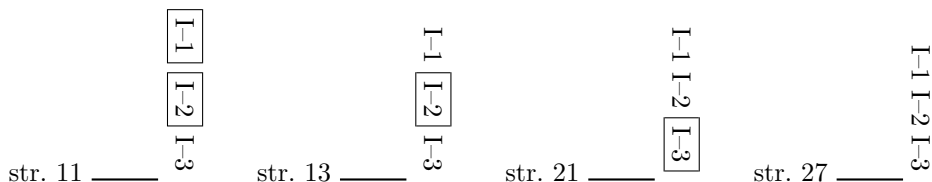
```

V českém dokumentu lze do preambule přidat `\usepackage{czech}` a nastavit si vstupní kódování, atd. To není náplní tohoto příspěvku. Nastavené x a y u příkazu `\put` jsou pevné a spíše méně vhodné, ale nechávám na čtenáři jejich případnou modifikaci, příp. využití zápatí k vysázení záložky. Příkaz `\gnadpis` je nastaven tak, aby byl vždy před příkazem `\section`. Jsme tedy v oblasti proměnných prvků, ale s určitým krokem zobrazování.

7.6. Záložka na dvoustraně

Tato záložka je připravena tak, že každá změna je řízena příkazem `\gnadpis` a všechny odhalené `\zapsano` se vysází. Jen aktuální sekce na dvoustraně budou zarámované pomocí `\framebox`.

Horizontální linky nám v tomto případě oddělují tělo dokumentu od jeho zápatí. Následují ukázky z několika stran při načtení zmíněného `zalozka.tex`:



Na dvoustraně se stranou 11 se vyskytují texty dvou sekcí, na straně 13 a 21 pak texty jen jedné ze sekcí.

Strana 27 je zvláště zajímavá, protože to je strana, která následovala až po zadání `\posledni`. Mohou to být přílohy, doplňky, atd., které k textům s nadpisy nepatří. Vytvořená záložka je vysázena, ale žádná sekce není zarámována.

Pozorný čtenář si již sám jistě upraví záložku položek vysázených na jedné straně.

Pro hloubavého čtenáře: Čtenář může dle chuti vyzkoušet svůj um, ať už se jedná o pevné umístění položek záložky, anebo umístění proměnně závislé od počtu částí, sekcí nebo počtu stran dokumentu. První nechť se vysází nahoře u záhlaví a pak rovnoměrně dle délky těla dokumentu až po poslední, která může být dole těsně u zápatí.

Např. sekce třetí (ze čtyř) na straně 25 (z celkových 125) se vysází přesně ve třech čtvrtinách (sazba závislá na počtu kapitol; $3/4$) nebo v jedné pětině (sazba závislá na stránkovém umístění; $25/125$) délky mezi záhlavím a zápatím.

8. Souhrnná ukázka

Ukažme si ještě jeden příklad. Ten také využije balíčku `fancyhdr` a bude postupně obsahovat informace v záhlaví (text) a zápatí (grafiku) z MySQL databáze s externími obrázky za pomoci vygenerovaného souboru skriptovacím jazykem PHP, jednoduchou ukázkou dekorace (chceme-li ornamentů) mimo oblast záhlaví a zápatí a se záložkou na stranách. V této souvislosti se dále používá sazba pasovacích, ořezových a zaměřovacích prvků.

Záhlaví a zápatí se generuje z databáze, kde máme uloženy citáty a odkazy na `.eps` soubory například s japonskými ekvivalenty. Soubory by měly být uloženy v adresáři s hlavním dokumentem, anebo při generování nastavíme absolutní cestu.

Záložka je téměř samogenerující a zobrazuje se na lichých stránkách. Dekorace už jen vhodně přidáme pomocí prostředí `picture` dle našeho grafického cítění.

V prvním kroku si vhodně upravíme generující `.php` soubor, a to touto cestou:

```
333 <?php
334 @$spojeni=mysql_connect("localhost","root","triadpass");
335 @mysql_select_db("skripta");
336 @$result=mysql_query("SELECT * FROM citaty");
337 $mp=fopen("zahlavi.tex","w");
338 fwrite($mp,"%".date("F d Y H:i:s.", getlastmod())."\r\n");
339 fwrite($mp,"\\renewcommand{\\headrulewidth}{1.2pt}\\r\n");
340 fwrite($mp,"\\renewcommand{\\footrulewidth}{0.8pt}\\r\n");
341 fwrite($mp,"\\fancyhf{}\\r\n");
342 fwrite($mp,"\\lhead{\\ifcase\\thepage\\or\\r\\n");
343 while ($line = mysql_fetch_array($result)) {
344     fwrite($mp,"\\emph{".$line["english"]."}");
345     fwrite($mp,"\\or \\r\\n"); }
346     fwrite($mp,"\\relax\\else\\relax\\fi }\\r\\n");
347 mysql_data_seek($result, 0);
348 fwrite($mp,"\\lfoot{\\ifcase\\thepage\\or\\r\\n");
349 while ($line = mysql_fetch_array($result)) {
350     fwrite($mp,"\\includegraphics[height=10pt]{");
351     fwrite($mp,$line["japanese"].".eps}");
352     fwrite($mp,"\\or \\r\\n"); }
353     fwrite($mp,"\\relax\\else\\relax\\fi }\\r\\n");
354 pclose($mp);
355 @mysql_close($spojeni);
356 echo "Last modified: ".date("F d Y H:i:s.", getlastmod()); ?>
```

Takto připravený .php soubor spustíme a výstupem je soubor zahlavi.tex, který vypadá následujícím způsobem (opět vnitřní položky vynecháváme, tedy řádky 363 a 368):

```

357 %February 05 2005 01:14:40.
358 \renewcommand{\headrulewidth}{1.2pt}
359 \renewcommand{\footrulewidth}{0.8pt}
360 \fancyhf{}
361 \lhead{\ifcase\thepage\or
362 \emph{Empty barrels make the most noise.}\or
363 %... Zde následují další položky načtené z MySQL databáze.
364 \emph{Security is the greatest enemy.}\or
365 \relax\else\relax\fi }
366 \lfoot{\ifcase\thepage\or
367 \includegraphics[height=10pt]{empty.eps}\or
368 %... Zde následují další položky načtené z MySQL databáze.
369 \includegraphics[height=10pt]{security.eps}\or
370 \relax\else\relax\fi }

```

Nyní je téměř vše připraveno pro výsledný T_EXový dokument. Ten strukturuje takto:

```

371 \documentclass[twoside]{article}
372 %... Načteme minimálně tyto tři balíčky.
373 \usepackage{graphicx} \usepackage{fancyhdr} \pagestyle{fancy}
374 \input{zahlavi.tex} %Soubor z předchozí strany tohoto článku.
375 %... Načteme čtvrtý zmiňovaný styl ze sekce 2.6.
376 %... Načteme zdrojové řádky ze sekcí 7.1., 7.2. a 7.4.
377 % V načtených řádcích přepínáme proměnnou \posun=1 a \posun=0.
378 \zalozka %Rozšíříme styl zavedený v zahlavi.tex o záložku.
379 \fancyhead[C]{\begin{picture}(0,0)\put(0,15){
380 \rotatebox[origin=r]{180}{\includegraphics%
381 [width=6.9cm, height=0.5cm]{obr1}} }
382 \put(-200,15){\includegraphics[width=7cm, height=0.5cm]{obr1}}
383 \end{picture} } %Prostředí picture rozšiřuje styl o grafiku.
384 \begin{document}
385 \urovenA{Vítejte!} %Úroveň part. Bez zápisu do záložky.
386 \gnadpis\urovenB{Sazba záhlaví} %Zápis záložky. Úroveň section.
387 \gnadpis\urovenB{Sazba záložky} %Zápis záložky. Úroveň section.
388 \gnadpis\urovenB{Volení stylů} %Zápis záložky. Úroveň section.
389 \urovenC{0 stylu 1} %Bez zápisu do záložky. Úroveň subsection.
390 \urovenC{0 stylu 2} %Bez zápisu do záložky. Úroveň subsection.
391 \poslednopr %Uzavření záložky. Nenásledují příkazy \gnadpis.
392 \end{document}

```

Výstup našich snah může vypadat například takto:

Strana 3	záhloví	Empty barrels make the most noise.	3	I-1
		TĚLO DOKUMENTU		I-2
	zápatí	あさせにあだなみ	I-1-1.5	
		⋮		
Strana 7	záhloví	Security is the greatest enemy.	8	I-1
		TĚLO DOKUMENTU		I-2
	zápatí	ゆだんたいてき	I-2-2.8	

Co se stane, pokud máme takto načtených přísloví méně než je stran dokumentu? V dalších řádcích se provádí `\relax`, tedy v záhlaví nebo zápatí na dalších stranách se nezobrazuje nic nového. Co se stane, pokud máme načtených řádků více než je stran dokumentu? Překladač o nich ví, ale nemá příležitost je v dokumentu zobrazit. Zobrazí se nanejvýš tolik přísloví, kolik je stran dokumentu.

Zdrojový kód `ukazka.tex` a ukázky v souborech `ukazka.pdf` a `ukazka.ps` hledejte na serveru `CTUGu`. Dále zde najdete vygenerovaný a uměle vytvořený soubor `zahnavi.tex` a předpřipravený soubor `zalozka.tex`. Grafický doplněk je vytvořen pomocí souboru `obr1` a poslední přiložené soubory jsou `empty.eps` a `security.eps`.

9. Slovo závěrem

Zajímavých záhlaví, zápatí a záložek se dá připravit určitě celá řada a budu rád, pokud mi na ukázkou svá nebo i cizí díla, případně odkazy na ně, pošlete. To samozřejmě platí pro libovolná „živá záhlaví a zápatí“, jak tento termín někteří typografové používají, včetně vždy vítaných neřešených kousků.

Stejné formátovací prvky, řadu dalších textů a především styl mluvy se čtenářem lze očekávat v budoucnosti u připravovaných skript *DTP a elektronické publikování* a *DTP and Electronic Publishing* pro stejnojmenné předměty na Fakultě managementu a ekonomiky Univerzity Tomáše Bati ve Zlíně.

Post mortem (Ponaučení)

Doufám, že čtenář tiše prominul, ale osobně velice rád čísluji nejlépe vše a všude. Zvláště to platí u řádků zdrojových kódů, na které se odkazují pomocí balíčku `lineno`. Pokud bych tyto řádky nečísloval, nebyl by tento *Post mortem* napsán.

Číslování přináší problém v podobě rychlého použití z `.ps` nebo `.pdf` souboru a přetažení do svého zdrojového dokumentu, protože číslo se umístí v `.txt` souboru pod příslušný řádek (extrahováno z `.ps`) nebo na konci řádku (extrahováno z `.pdf`). To vyžaduje další nutné úpravy (např. opět pomocí skriptovacího jazyka PHP), než si čtenář může zdrojový kód ve svém dokumentu vyzkoušet. Nemluvě o úporném přepisu z tištěného podkladu.

Poněvadž zdrojové kódy píší strukturovaně, vždy na začátku řádku zadaným příkazem `\verb` a na konci ukončené znakem `|` nebo příkazem `\linelabel` (příkazem `\verb` textový odstavec nikdy nezačínám), lze tyto údaje vyextrahovat z `.tex` dokumentu za pomoci PHP. Pomocí čítače můžeme za zdrojový řádek přidat informaci o čísle řádku v příspěvku, ale již jen jako poznámku.

Bloky zdrojových kódů začínají `\linenumbers` a jsou ukončeny příkazem `\citk`, který definujeme takto:

```
393 \def\citk{\nolinenumbers\medskip}
```

Toho můžeme využít a ve výstupním souboru přidat pro přehlednost volný řádek. Program jsem nazval `extrakce.php` a vypadá takto:

```
394 <?php
395 $vstup=fopen("zdroj.tex","r"); $zapis=fopen("kod.tex","w");
396 $citac=0; $delka=0;
397 while (!feof($vstup)) {$radek=fgets($vstup); $label="";
398 if (substr($radek,0,6)=="\verb".chr(124)){
399     $label=strrchr($radek,chr(124)); }
400 if (substr($radek,0,5)=="\citk"){
401     echo "<br>"; fwrite($zapis,"\r\n"); }
402 if ($label!=""){ $citac++;
403     $delka=strlen($radek)-strlen($label)-6;
404     echo substr($radek,6,$delka)."%". $citac."<br>";
405     fwrite($zapis,substr($radek,6,$delka)."%". $citac."\r\n"); }
406 } fclose($vstup); fclose($zapis); ?>
```

Soubor `zdroj.tex` je zdrojový kód tohoto článku nebo jakýkoli dokument se stejnou strukturou u číslovaných řádků zdrojových kódů. Soubor `kod.tex` je vygenerovaným souborem a konkrétně pro tento článek je zpřístupněn na serveru `ČTUGu`. Pokud změníme na řádku 395 výstupní soubor na `kod_bez.tex` a z řádků 404 a 405 odstraníme `"%". $citac.`, dostaneme soubor bez komentářů. To je výhodné u PHP, protože zde se nekomentuje pomocí znaku procenta.

Nechť samotný článek a další zveřejněné soubory dobře slouží.

Slovníček pojmů

Uvedme si vysvětlení některých pojmů použitých v textu.

Dekorace. Výzdoba zkrášlující sazbu textu.

Ornament. Grafický prvek zkrášlující sazbu textu.

Ořezové značky. Pomůcky pro správné nařezání papíru. Typickým příkladem je příprava vizitek a A5 bulletinů (z vytištěných papírů A4).

Pasovací značky. Zobrazují postupné zvětšování mezery mezi stránkami pro vnější archy svazečku tak, aby po ořezání stránky lícovaly. Citováno z [2].

Položka proměnného prvku. Nejčastěji jedna položka z databáze, která se sází v textu právě jednou, a to nejčastěji za pomoci příkazu `\ifcase`.

Proměnný prvek. Je to takový typografický objekt, který se se změnou čítače `page`, obecně však vlivem jakéhokoliv čítače, mění v sázeném textu.

Prvek. Objekt. Typografický prvek. Proměnný prvek. V podstatě jakákoli diskutovaná pasáž sazby.

Stránkově nezávislé. Jakákoli položka z proměnného prvku může být umístěna v záhlaví nebo zápatí na libovolné straně a neztratí se přitom významové souvislosti mezi položkami proměnných prvků.

Stránkově závislé. Nelze vysázet položku jednoho proměnného prvku na jinou stranu, aniž by se položky jiných proměnných prvků v záhlaví nebo zápatí bez ztráty významu musely také přesouvat.

Zaměřovací značky. Slouží ke správné identifikaci (umístění a natočení) prvků na straně. Typickým příkladem je skenování testů pro jejich automatické vyhodnocení.

Živé záhlaví a zápatí. Jakákoli sazba v záhlaví nebo zápatí, která má čtenáři usnadnit orientaci v publikaci.

Slovníček Top 40

Uvedme si některé pojmy ve stylu EN→CZ.

advance [əd'vɑ:ns]	rozšířit	dependent [dɪ'pendənt]	závislý
alpha [ælfə]	abecedně	divide [dɪ'vaɪd]	podělit
arabic ['ærəbɪk]	arabsky	empty ['emptɪ]	prázdný
article ['ɑ:tɪkəl]	článek	even ['i:vən]	sudý
box [bɒks]	schránka	fancy ['fænsɪ]	pestrý
conference ['kɒnfərəns]	konference	footing ['fʊtɪŋ]	zápatí
connect [kə'nekt]	připojit se	heading ['hedɪŋ]	záhlaví
count [kaʊnt]	spočítat	height [haɪt]	výška
decoration [dekə'reɪʃən]	výzdoba	immediate [ɪ'mɪ:diət]	okamžitě

independent [ɪndrɪ'pendənt] .. nezávislý	result [rɪ'zʌlt] výsledek
leftmark [leftmɑ:k] levá značka	rightmark [raɪtmɑ:k] pravá značka
movie ['mʊvi] sekvence obrázků	rotate [rəʊ'tert] pootočit
multiply ['mʌltɪplai] vynásobit	section ['seksjən] sekce
odd [ɒd] lichý	select [srɪ'lekt] vybrat
origin ['ɒrɪdʒɪn] původní	string [strɪŋ] textový řetězec
page [peɪdʒ] strana	subsection [sʌb'sekʃən] podsekce
paper ['peɪpə] příspěvek	typesetting ['taɪpsetɪŋ] sazba
picture ['pɪktʃə] obrázek	value ['vælju:] hodnota
query ['kwɪəri] dotaz	variable ['veəriəbəl] proměnná
relax [rɪ'læks] odpočívát	width [wɪdθ] šířka

Reference

- [1] Tobias Oetiker, Hubert Partl, Irene Hyna and Elisabeth Schlegl. *The Not So Short Introduction to L^AT_EX 2_ε*.
<http://www.ctan.org/tex-archive/info/lshort/>.
- [2] Petr Olšák. *Jak používám T_EX?* 4. ukázka: Inzertní příloha časopisu DoteK.
<http://www.olsak.net/texpraxe.html>.
- [3] Petr Olšák. *T_EXbook naruby*. Public edition 1996, Konvoj 1997. 468 pp, ISBN 80-85615-64-9.
<http://math.feld.cvut.cz/olsak/tbn.html>.
- [4] Piet van Oostrum. *Page layout in L^AT_EX*. MAPS, 16:108–125, 1996.
http://www.ntg.nl/maps/pdf/16_29.pdf.
- [5] Jozef Říha, Pavel Stríž. *T_EX-Collection 2003 a 2004*. Informační Bulletin České statistické společnosti, 4(15), pp 4–7, Praha 2004. ISSN 1210–8022.
<http://www.statapol.cz/bulletiny/ib-04-4.pdf>.

Pavel Stríž, ÚIS FaME UTB ve Zlíně
striz@fame.utb.cz, pavel.striz@email.cz