

# Zpravodaj Československého sdružení uživatelů TeXu

---

Petr Olšák

OPmac – makra rozšiřující možnosti plainTeXu

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 22 (2012), No. 1, 20–41

Persistent URL: <http://dml.cz/dmlcz/150200>

## Terms of use:

© Československé sdružení uživatelů TeXu, 2012

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>

OPmac je balík jednoduchých doplňujících maker k plain $\TeX$ U umožňující uživatelům základní  $\LaTeX$ ovou funkcionalitu: změny velikosti písma, automatickou tvorbu obsahu a rejstříku, práci s **bib** databázemi, referencemi, možnost proložení referencí hyperlinkovými odkazy atd. V tomto článku jsou ukázány významné vlastnosti makra OPmac. Soubor maker a úplná uživatelská i technická dokumentace jsou ke stažení na <http://petr.olsak.net/opmac.html>.

## Úvod

V říjnu roku 2012 jsem se rozhodl, že společně s novým  $\mathcal{C}\mathcal{S}$ plainem zveřejním svá makra, která už léta používám pro nejrůznější účely. Stačilo je trochu pročistit, přidat k nim uživatelskou a technickou dokumentaci a zveřejnit je. Základním záměrem bylo nabídnout uživatelům plain $\TeX$ U sadu maker, která řeší nejčastěji se vyskytující úlohy. Aby nemuseli pořád nanovo vymýšlet kolo. Vedlejším efektem je demonstrace faktu, že  $\TeX$ ová makra lze dělat jednoduše a účelně, což je fenomén, který nebývá v  $\LaTeX$ U obvyklý.

Při tvorbě balíku OPmac jsem se řídil následujícími zásadami:

- V jednoduchosti je síla.
- Makra nejsou univerzální, ale jsou čitelná a srozumitelná.
- Uživatel si makra může snadno předefinovat k obrazu svému.

Každé makro je napsáno s cílem co největší srozumitelnosti pro lidi, kteří to budou chtít číst, porozumět tomu a měnit to. Balík nabízí čtenáři inspiraci, jak se programují  $\TeX$ ová makra. Z kódu maker je cítit jistá elegance. Technická část dokumentace k OPmac by mohla sloužit jako učebnice programování  $\TeX$ ových maker. To je zásadní rozdíl od koncepce  $\LaTeX$ U. Když se člověk podívá do  $\LaTeX$ ového souboru `latex.1tx`, vystřeví se na něj množství zavináčů a makra, ze kterých je často cítit topornost a mnohdy nepochopení vnitřní koncepce  $\TeX$ U. Skoro nikdo se v tom nevyzná. Soubor `latex.1tx` obsahuje 8000 řádků a schopnosti  $\LaTeX$ U jsou navíc ukryty v desítkách či stovkách různých dalších makro souborech. Naproti tomu v OPmac vidíte vše pohromadě a názorně v jednom malém souboru. V některých věcech OPmac výrazně překračuje možnosti běžného [1] i rozšířeného [2]  $\LaTeX$ U: generování rejstříků bez externího programu, práce s předgenerovanou databází bibliografických záznamů, listingy externích souborů, pohodlnost a účelnost, s jakou se makra ovládají na uživatelské úrovni.

Balík OPmac nabízí podobně jako L<sup>A</sup>T<sub>E</sub>X autorům textů *rozhraní*, tj. smlouvenou sadu značek na vymezení struktury dokumentu. Je jiná, než v L<sup>A</sup>T<sub>E</sub>Xu, možná umožní napsat zdrojový text článku poněkud přehledněji a oku více lahodícím způsobem. Balík OPmac ovšem neřeší typografický vzhled dokumentu. Bez doplňujících maker vyleze jednoduchý střízlivý dokument. Předpokládá se, že autor dodatečných plainT<sub>E</sub>Xových maker ušije vzhled dokumentu na míru konkrétnímu požadavku. Například pro nastavení vzhledu tohoto článku podle typografie časopisu Zpravodaj stačilo dopsat 30 doplňujících řádků maker.

Uživatelům OPmac nabízím konzultace po emailu a uvítám hlášení o chybách. Pojdme společnými silami odstranit případné mouchy a mušky.

V následujícím textu je přepsána větší část uživatelské dokumentace balíku OPmac. Vážní zájemci o OPmac si mohou přečíst i technickou dokumentaci dostupnou na síti, která podrobně popisuje interní řešení jednotlivých maker.

## Záhlaví dokumentu

Na začátek dokumentu můžete napsat třeba toto:

```
\input opmac      % zavedení makra OPmac
\chyph           % zapnutí češtiny
\input lmfonds   % Latin Modern fonty (od verze CSplainu Nov.2012)
\typosize[12/14] % nastavení základní velikosti sazby
```

## Velikosti fontů a řádkování

Všechna makra popsaná v této sekci nastavují změny ve fontech a dalších parametrech jen lokálně, takže po ukončení skupiny se nastavení vrací k původním hodnotám.

Makro `\typosize[⟨velikost fontu⟩/⟨řádkování⟩]` nastaví velikost textových i matematických fontů a řádkování. Je-li některý z parametrů prázdný, makro nastaví jen údaje plynoucí z neprázdného parametru. Parametry neobsahují jednotku, jednotka pt se doplní v makru. Příklady

```
\typosize[10/12] % to je implicitní nastavení
\typosize[11/12.5] % font velikosti 11pt, řádkování 12.5pt
\typosize[8/] % font velikosti 8pt, řádkování nezměněno.
```

Začátek dokumentu tedy může vypadat takto:

```
\input opmac \typosize[11.5/13] % stupeň písma 11.5pt, řádkování 13pt
```

Makro `\typoscale[⟨faktor-font⟩/⟨faktor-řádkování⟩]` zvětší nebo zmenší velikost textových i matematických fontů resp. řádkování *⟨faktor⟩*krát aktuální velikost fontů resp. řádkování. Faktor je celé číslo, přitom 1000 znamená jedničku

(jako za slovem `scaled` v příkazu `\font`). Je-li parametr prázdný, je to stejné, jako by byl roven 1000.

```
\typoscale[800/800] % fonty i řádkování se zmenší na 80 %  
\typoscale[\magstep2/] % \magstep2 je 1440, písmo se zvětší 1,44krát
```

Tato makra můžeme použít pro nadpisy nebo poznámky s tím, že velikost bude relativní vzhledem k základnímu písmu celého dokumentu.

Pokud zavedete font příkazem `\font\prepinač=<metrika>` (následovaný případným `at` nebo `scaled`) pak `\prepinač` přepíná do pevně stanoveného fontu, který není ovlivněn makry na nastavování velikosti. Ale co není, může být. Stačí font registrovat pomocí `\regfont\prepinač` a nyní i `\prepinač` přepíná do fontu podle velikosti nastavené pomocí `\typosize` nebo `\typoscale`. Příklad:

```
\font\zapfchan=pzcmi8z \regfont\zapfchan  
\typosize[20/] Taký \zapfchan přepne do Zapf-Chancery velikosti 20pt.
```

Při inicializaci `CSplainu` je registrováno pět fontových přepínačů ukrytých v makrech `\rm`, `\it`, `\bf`, `\bi`, `\tt`. Takže tato makra implicitně nastavují font do stanovené velikosti.

Na místo `\typosize` a `\typoscale` je možno použít makra na změnu velikosti jen textových fontů `\textfontsize[<velikost>]` a `\textfontscale[<faktor>]`. Tato makra nemění matematické fonty ani řádkování.

Makra `\thefontsize[<velikost-fontu>]` nebo `\thefontscale[<faktor>]` změní velikost jen aktuálního fontu, nemění žádné jiné fonty a nemění řádkování. Tato makra používají interní fontový přepínač `\thefont`. Je možné si jej „vypůjčit“ příkazem `\let` pro pozdější použití:

```
\tenrm \thefontscale[700]\let\sevenrm=\thefont \tenrm
```

Díky přepínači `\tenrm` na konci ukázky sazba dále pokračuje neškálovaným fontem, ovšem škálovaný font máme uložen v přepínači `\sevenrm`.

Všechna zde uvedená makra na změnu velikosti fontů jsou vybavena inteligencí: hledají metriku, která má svou designovanou velikost nejbližše požadované velikosti. Takže při požadavku na velikost 13pt se použije metrika `csr12 at13pt`, zatímco při velikosti 7.5pt se použije metrika `csr8 at7.5pt`. Data pro tuto inteligenci jsou přečtena ze souboru `ams-math.tex`, kde je najdete u příkazů `\regtfm`.

Poslední poznámka se týká makra `\em`, které sice nepřepíná velikost fontů, ale přepíná jejich variantu. Je to kontextové makro, které pracuje v závislosti na aktuálně zvoleném fontu. Implicitně přepíná na `\it`. Pokud ale je aktuálním fontem `\it`, přepne na `\rm`. Je-li aktuálním fontem `\bf` přepne na `\bi` a obráceně. Makro navíc správně doplní italské korekce ke slovu před jeho použitím a za jeho použitím. Takže se o italské korekce není nutno starat. Příklad:

```
To je {\em zdůrazněný} text. %jako: To je {\it zdůrazněný\} text.  
\it To je {\em zdůrazněný} text. %jako: To je\ { \rm zdůrazněný} text.  
\bf To je {\em zdůrazněný} text. %jako: To je {\bi zdůrazněný\} text.  
\bi To je {\em zdůrazněný} text. %jako: To je\ { \bf zdůrazněný} text.
```

## Členění dokumentu

Dokument se může skládat z kapitol, kapitola ze sekcí a sekce z podsekcí. Titul dokumentu vyznačte pomocí `\tit <titul><prázdný řádek>`, kapitolu zahajte `\chap<titul><prázdný-řádek>` a podobně novou sekci je možné zahájit makrem `\sec<titul><prázdný-řádek>` a podsekcí `\secc<titul><prázdný-řádek>`. Takže třeba:

```
\chap Brouci
```

```
\sec Chrousti
```

```
\secc 0 nesmrtnosti chroustů
```

Bla bla bla bla ...

Bla bla bla a ještě bla.

Kapitoly se automaticky číslují jedním číslem, sekce dvěma čísly (číslo kapitoly.sekce) a podsekcí třemi čísly. Pokud dokument neobsahuje kapitoly, číslo kapitoly chybí, tj. sekce má jedno číslo a podsekcí dvě.

Implicitní vzhled nadpisů kapitol, sekcí a podsekcí je definován v makrech `\printchap`, `\printsec` a `\printsecc`. Můžete se na obsah těchto maker podívat do technické dokumentace nebo do `opmac.tex`. Můžete se těmi makry inspirovat a třeba je předefinovat podle vlastního typografického návrhu.

První odstavec za titulem kapitoly, sekce a podsekcí není odsazen. Pokud jej chcete mít odsazen jako ostatní odstavce, napište `\let\firstnoindent=\relax`.

Jestliže je název kapitoly, sekce nebo podsekcí příliš dlouhý, rozlomí se do řádků. V takovém případě je někdy lepší rozdělit název do řádků manuálně. K tomu slouží makro `\n1`, které odřádkuje v místě použití (newline). Toto makro se navíc v obsahu chová jako mezera.

## Další číslované objekty a odkazy na ně

Kromě kapitol, sekcí a podsekcí se automaticky číslují ještě rovnice a popisky pod obrázky a pod tabulkami.

Pokud na konci `display` módu je uvedeno `\eqmark`, tato rovnice bude číslovaná. Formát číslování je implicitně jediné číslo v kulaté závorce resetované při každém zahájení nové sekce. Příklad: `$$ a^2 + b^2 = c^2 \eqmark $$` vytiskne

$$a^2 + b^2 = c^2 \tag{1}$$

Je-li potřeba očíslovat jedním číslem více rovnic sestavených například pomocí `\eqalignno`, pak použijte `\eqmark` v posledním sloupci, například takto:

\$\$

```
\eqalignno{a^2+b^2 &= c^2 \cr
            c &= \sqrt{a^2+b^2} & \eqmark \cr}
```

\$\$

Ukázka dává tento výsledek:

$$\begin{aligned} a^2 + b^2 &= c^2 \\ c &= \sqrt{a^2 + b^2} \end{aligned} \tag{2}$$

Dalšími číslovanými objekty jsou popisky. Popisek pod obrázky je potřeba uvést slovem `\caption/f` a popisek pod tabulkami slovem `\caption/t`. Pak následuje text popisku ukončený prázdným řádkem. Příklad:

```
\hfil\table{rl}{věk & hodnota \cr\l\lnoalign{\smallskip}
                0-1 & neměřitelná \cr
                1-6 & projevující se \cr
                6-12 & výrazná \cr
                12-20 & extrémní \cr
                20-40 & mírnější \cr
                40-60 & různá \cr
                60-$\infty$ & umírněná} % vytvoření tabulky
\par\nobreak\medskip
\caption/t Závislost závislosti na počítačích na věku.
```

Tato ukázka vytvoří:

věk	hodnota
0–1	neměřitelná
1–6	projevující se
6–12	výrazná
12–20	extrémní
20–40	mírnější
40–60	různá
60–∞	umírněná

**Tabulka 1** Závislost závislosti na počítačích na věku.

Vidíme, že makro `\caption/t` doplnilo slovo „Tabulka“ následované číslem. Podobně se chová `\caption/f`, jen místo slova „Tabulka“ se v textu zjeví slovo „Obrázek“. Obrázky a tabulky jsou číslovány nezávisle. Popisek je centrován. Je-li popisek delší na více řádcích, je centrován poslední řádek.

Způsob číslování lze změnit jinou definicí makra `\thednum` (pro rovnici), `\thetnum` (pro tabulky) a `\thefnum` (pro obrázky).

Makro `OPmac` vloží slovo „Tabulka“ v závislosti na nastaveném jazyce příkazem `\chyp`, `\shyph`, `\ehyph`. Při `\shyph` dostaneme „Tabulka“ a při `\ehyph` máme

„Table“. Podobně se chovají slova „Obrázek/Obrázok/Figure“ a „Kapitola/Kapitola/Chapter“. Jiná automaticky generovaná slova OPmac nepoužívá.

Předefinovat tato slova lze pomocí `\sdef`, jak ukazuje následující příklad, který zamění celá slova za zkratky.

```
\sdef{mt:t:cz}{Tab.} \sdef{mt:t:sk}{Tab.} \sdef{mt:t:en}{Tab.}
\sdef{mt:f:cz}{Obr.} \sdef{mt:f:sk}{Obr.} \sdef{mt:f:en}{Fig.}
```

L<sup>A</sup>T<sub>E</sub>Xoví uživatelé jsou zvyklí, že jim tabulky a obrázky plavou v dokumentu, přičemž inklinují k horní části stránky. To se při použití OPmac implicitně neděje, ale je možno plavání zařídit pomocí plainT<sub>E</sub>Xového makra `\topinsert` resp. `\midinsert`. Například:

```
\topinsert
  \hfil\table{r1}{...} % vytvoření tabulky
  \medskip
  \caption/t Závislost závislosti na počítačích na věku.
  \par
\endinsert
```

Na automaticky číslované objekty je nutno se občas v textu odkazovat. Protože dopředu nevíme, pod jakým číslem se rovnice, sekce, tabulka atd. vytiskne, je potřeba použít interní lejbílky k označování odkazovaných objektů. K tomu slouží makro `\label[⟨lejbílka⟩]`, které musí předcházet makru, jež generuje číslo. Není nutné, aby `\label` předcházel těsně danému makru. Tedy například:

```
\label[chroust] \sec 0 nesmrtelnosti chroustů
```

```
\label[zavislaci]
\hfil\table{r1}{...} % vytvoření tabulky
\caption/t Závislost závislosti na počítačích na věku.
```

```
\label[pythagoras]
$$ a^2 + b^2 = c^2 \eqmark $$
```

Nyní můžeme hovořit o~sekci~\ref[chroust] na straně~\pgref[chroust] nebo taky o~rovnici~\ref[pythagoras] na straně~\pgref[pythagoras]. Dále bude potřeba upozornit na tabulku~\ref[zavislaci] na straně~\pgref[zavislaci], která shrnuje jistý druh závislosti.

Jestliže se v textu vyskytují dopředné reference (tj. odkazujeme na objekt, který ještě není vytištěn) nebo text odkazuje na stránky (`\pgref`), je nutné T<sub>E</sub>Xovat dokument dvakrát.

## Odrážky

Jednotlivé myšlenky je občas potřeba vypíchnout odrážkami. Prostředí s odrážkami se vymezuje sekvencemi `\begitem`s a `\enditem`s. Uvnitř tohoto prostředí je hvězdička aktivním znakem, který zahajuje odrážky. Prostředí s odrážkami je možné vnořit do sebe. Pomocí `\style <znak>` hned za slovem `\begitem`s je možné vymezit některé z předdefinovaných vzhledů odrážek:

```
\style o % malý puntík
\style 0 % velký puntík $\bullet$ (implicitní volba)
\style - % spojovník
\style n % odrážky číslované 1., 2., 3., ...
\style N % odrážky číslované 1), 2), 3), ...
\style i % odrážky číslované (i), (ii), (iii), (iv), ...
\style I % odrážky číslované I, II, III, IV, ...
\style a % odrážky s písmeny a), b), c), ...
\style A % odrážky s písmeny A), B), C), ...
\style x % malý čtvereček
\style X % velký čtvereček
```

Příklad:

```
\begitem \style n
* Tady je první myšlenka.
* A tady druhá, která je rozdělena na
  \begitem \style a
  * podmyšlenku
  * a hned následuje další podmyšlenka,
  * poslední podmyšlenka.
  \enditem
* Tady je třetí myšlenka.
\enditem
```

vytvoří následující výstup:

1. Tady je první myšlenka.
  2. A tady druhá, která je rozdělena na
    - a) podmyšlenku
    - b) a hned následuje další podmyšlenka,
    - c) poslední podmyšlenka.
3. Tady je třetí myšlenka.

Chcete-li uvnitř prostředí s odrážkami vytisknout hvězdičku, pište `\char‘\*`.

Pomocí `\sdef{item:⟨písmeno⟩}{⟨text⟩}` si můžete dodefinovat vzhled odrážek podle svých představ. Implicitní odrážku můžete předefinovat pomocí `\def \normalitem{⟨text⟩}`.



Jednotlivá prostředí s odřádkami se odsazují podle registru `\iindent`, který je nastaven na hodnotu `\parindent` v době čtení souboru `opmac.tex`. Pokud později změníte `\parindent`, doporučuji na stejnou hodnotu nastavit `\iindent`. Vertikální mezera nad a pod prostředím s odřádkami je řízena makrem `\iiskip`.

## Tvorba obsahu

Makro `\maketoc` vytiskne v místě svého použití obsah dokumentu bez nadpisu, jen jednotlivé řádky obsahu. Odsazení jednotlivých řádků je nastaveno na násobky registru `\iindent`. Často je potřeba dokument  $\TeX$ ovat vícekrát, než se obsah zjeví a než se stránkové číslice srovnají správně, protože po prvním zjevení obsahu se mohou stránky posunout jinam.

Nadpis pro obsah většinou nebudeme psát pomocí `\chap` nebo `\sec`, protože by se údaj o obsahu dostal do obsahu, což obvykle není žádoucí. Je proto potřeba použít vnitřního makra `\printsec` a před jeho použitím vyprázdnit obsah `\thesectnum`, abychom neměli tuto sekci číslovanou. Takže obsah vyrobíme třeba takto:

```
\def\thesectnum{\printsec{\unskip Obsah}\maketoc}
```

Tituly kapitol, sekcí a podsekcí zapisuje OPmac pro účely sestavení obsahu do externího souboru `*.ref`. Může se stát, že uživatel v těchto textech použije nějaké komplikované makro, které se pak v souboru „rozsype“ do takového stavu, že nejde vzápětí přelit. V takovém případě je potřeba makro zabezpečit proti expanzi při zápisu do souboru pomocí deklarace `\addprotect\makro`. Takto deklarované makro je pak zabezpečené proti expanzi do `*.ref` souboru. Například OPmac deklaruje:

```
\addprotect~ \addprotect\TeX \addprotect\thefontsize \addprotect\em
```

a mnoho dalších. Není možné ale předvídat všechno, co může uživatel nacpat do titulu sekce nebo kapitoly. Pokud se tedy „rozsype“ REF soubor, je potřeba si tímto způsobem zabezpečit používané makro.

## Sestavení rejstříku

Makro pro zanášení slov do rejstříku je navrženo s ohledem na optimalizaci počtu úhozů na klávesnici. Autor už napsal své dílo, má daný termín odevzdání a nyní ho čeká úmorná práce vyhledávání slov v textu, která by měla přijít do rejstříku, a jejich vyznačování. Je třeba mu tuto práci co nejvíce usnadnit.

Pro zanesení slova do rejstříku slouží makro `\ii`. Je to zkratka za „insert to index“. Jeho parametr je  $\langle slovo \rangle$  bez mezery ukončené mezerou (obecnější tvar parametru uvedeme později). Toto slovo se přepíše do rejstříku, ve kterém jsou

všechna takto deklarovaná slova seřazena podle abecedy a jsou k nim připojena čísla stránek, na kterých bylo použito odpovídající makro `\ii` (*slovo*). Příklad:

Tady mluvím o jistém

`\ii rezistor`

rezistoru, který provokoval moji zvědavost.

Makro `\ii` viditelně neudělá v sazbě nic. Přilepí se na následující slovo (v našem příkladě slovo „rezistoru“) jako skrytá značka. Číslo strany, kde se ta značka objeví, bude v rejstříku vedle slova „rezistor“.

Je-li `\ii` zapsáno ve vertikálním módu, zahájí se v daném místě odstavec, aby se mohla neviditelná značka z `\ii` nalepit na následující slovo. Pokud si to z nějakých důvodů nepřejete, použijte interní variantu makra `\iiindex{<slovo>}`, která nezahajuje odstavec.

Pokud se v rejstříku má objevit stejné slovo jako v textu, není nutno je psát dvakrát. Stačí použít makro `\iid` (zkratka za `\ii double`):

Zásady jsou `\iid nestrannost` , `\iid pravdomluvnost` a `\iid odvaha` .

To povede ke stejnému výsledku jako

Zásady jsou `\ii nestrannost nestrannost`,

`\ii pravdomluvnost pravdomluvnost` a `\ii odvaha odvaha`.

Povšimněte si, že čárky a tečky jsou odstrčeny od dublovaného slova, protože mezera je ukončovací znak parametru `\iid`. Do textu se mezera vrátí právě tehdy, když nenásleduje tečka nebo čárka. V našem příkladě před spojkou „a“ mezera ve výsledku je, ale před tečkou nebo čárkou mezera není.

Vlastnosti makra `\iid` jsou tímto popsány zcela. Vraťme se k makru `\ii`, které poskytuje další možnosti.

Parametr `\ii` je vždy ukončen mezerou. Může obsahovat čárky (bez mezer), které naznačují, že se do rejstříku dává více slov:

`{\bf Definice.}`

`\ii lineární~prostor,vektorový~prostor`

`{\em Lineárním prostorem}` (nebo též vektorovým prostorem) rozumíme ...

Dostaneme totéž jako při `\ii lineární~prostor` `\ii vektorový~prostor` a tato ukázka demonstruje ještě jednu věc: je-li potřeba do parametru `\ii` dostat mezeru, pište vlnku.

Pokud se v rejstříku objeví hesla skládající se z více slov, obvykle chceme, aby u hesla, které opakuje první slovo, se toto slovo v rejstříku nevypisovalo opakovaně, ale aby bylo nahrazeno pomlčkou. Například:

lineární podprostor 12, 16, 18, 29

— prostor 12, 16–32, 51

— závislost 18–20, 34

Při takovém požadavku pište místo vlnky mezi slovy lomítko. Příklad:

`\ii lineární/prostor,vektorový/prostor`

Někdy je vhodné kromě hesla `lineární/prostor` zařadit i `prostor/lineární`. Aby se to nemuselo psát dvakrát, je k dispozici zkratka `@` napsaná za čárku na konci parametru:

```
\ii lineární/prostor,vektorový/prostor,@
% je totéž jako \ii lineární/prostor,vektorový/prostor
%                \ii prostor/lineární,prostor/vektorový
```

Počet lomítek v hesle pro rejstřík není omezen. Můžete tedy vytvořit víceúrovňový rejstřík. Nicméně je třeba vědět, že zkratka `@` nevytváří všechny permutace, ale jen prohazuje první údaj před lomítkem se všemi ostatními. Takže `\ii a/b/c,@` je totéž jako `\ii a/b/c \ii b/c/a`.

Samotný rejstřík vznikne v místě příkazu `\makeindex`. Rejstřík obsahuje data z předchozího zpracování dokumentu `TEX`em, takže je potřeba `TEX`ovat aspoň dvakrát. Makro `\makeindex` abecedně seřadí data v rejstříku podle českých a slovenských pravidel řazení a upraví odkazy na stránky (aby se stránky neopakovaly a inklinovaly k zápisu ve tvaru 26–28). Makro `\makeindex` se nestará o prostředí, do kterého sazbu vyvrhne, ani o nadpis. To musíme udělat sami. `OPmac` nabízí pro sazbu do více sloupců makra `\begmulti <počet-sloupců> ... \endmulti`.  
Příklad:

```
\sec Rejstřík\par
\begmulti 3 \makeindex \endmulti
```

Do rejstříku musejí být zařazena jen „čistá“ slova, která neobsahují makra expandující na primitivní příkazy `TEX`u. Pokud chcete vytisknout v rejstříku něco komplikovanějšího, můžete sestavit slovník výjimek pomocí sady maker `\iis <heslo><mezera>{\tisk}` (název makra můžeme číst jako `\ii speciální`). Funkci si vysvětlíme na příkladu:

```
\iis chikvadrat {$\chi$-kvadrát}
\iis relax {{\tt \char`\relax}}
\iis Goedelova/věta/o-neúplnosti {G"odelova/věta/o-neúplnosti}
\iis věta/o-neúplnosti/Goedelova {věta/o-neúplnosti/G"odelova}
```

Po takové deklaraci je možné psát `\ii relax` nebo `\ii chikvadrat` nebo třeba `\ii Goedelova/věta/o-neúplnosti,@`. `OPmac` abecedně řadí podle těchto hesel, ale když dojde na potřebu heslo vytisknout do rejstříku, vytiskne místo těchto hesel materiál, který je uveden na pravé straně slovníku. Příklad ukazuje, že tím lze řešit nejen tisk hesel, která je potřeba ošetřit speciálními makry (v příkladu slovo `relax`), ale také výjimky abecedního řazení (písmeno `\o` nemáme v české abecedě, kdoví, kam by se to zařadilo). Slovník výjimek je možný zapsat kamkoli před `\makeindex`, typicky se píše na začátek dokumentu.

Výjimku z řazení dvojhlásky `ch` (například ve slově `mochнатý`, tj. `mnohonohý`) je možné zařadit pomocí tečky, která má stejně jako ostatní interpunctní znaky,

nulovou řadičí platnost (OPmac hesla řadí, jakoby tam interpunkce nebyla). Takže třeba takto:

```
... \ii moc.hnátý ...
\iis moc.hnátý {mochnátý}
```

Je-li při zpracování `\makeindex` zapnutý anglický jazyk (implicitní nastavení nebo po přepínači `\ehyph`), pak se `ch` neinterpretuje jako dvojhláska. Ostatní pravidla řazení zůstávají nezměněna.

Pro různé speciální znaky můžete využít znak `@`, který se řadí před celou abecedou. Speciální znak pak nahradíte až ve slovníku výjimek. Takže třeba `\ii Ernst-@~Young` pro řazení a `\iis Ernst-@~Young {Ernst \& Young}` pro tisk.

## Barvy, vodoznaky

Makra uvedená v této sekci nastavují barvy jen při použití pdf $\TeX$ u a při výstupu do PDF. Jinak tato makra neudělají nic.

Barvu textu můžete nastavit pomocí přepínačů `\Blue`, `\Red`, `\Brown`, `\Green`, `\Yellow`, `\White`, `\Grey` a `\Black`. Je potřeba si uvědomit, že tyto přepínače pracují globálně nezávisle na skupině uvnitř boxu i mimo, prostě kdekoli. Barvu jinou než černou je potřeba nakonec vypnout přepínačem `\Black`. Pokud barvu jinou než černou zapnete na některé stránce a text v této barvě přeteče do stránek dalších, při opakovaném  $\TeX$ ování se objeví správná barva i na následujících stránkách. Tuto skutečnost si totiž  $\TeX$  musí ujasnit prostřednictvím pomocného souboru `s odkazy`.

Společně s barvou textu tyto přepínače zapínají i barvu větších ploch vytvořených pomocí `\hrule` a `\vrule`. Ve specifikaci formátu PDF je nicméně ještě jeden přepínač barev nezávislý na barvě textu a větších ploch. Jsou to barvy linek, které mají tloušťku menší než 1pt. Tento „druh barvy“ se přepíná stejnými přepínači `\Blue`, ..., `\Black` ovšem před tímto přepínačem musí bezprostředně předcházet slovo `\linecolor`. Takže `\linecolor\Red` zapne barvu linek na červenou, ale barva textu zůstane původní. Nebo `\Black` vrátí do normálu barvu textu, zatímco `\linecolor\Black` způsobí, že také linky budou dále černé.

Kromě uvedených barevných přepínačů si můžete „namíchat“ v barevném prostoru CMYK i barvy vlastní. Stačí se inspirovat, jak jsou uvedené přepínače definovány:

```
\def\Red{\setcmykcolor{0.0 1.0 1.0 0}}
\def\Brown{\setcmykcolor{0 0.67 0.67 0.5}}
...
```

Následuje příklad, kterým vytvoříme podbarvený text:

```
\def\podbarvi#1#2#3{\setbox0=\hbox{#3}\leavevmode
  \rlap{#1\strut\vrule width\wd0}#2\box0\Black}
\podbarvi\Yellow\Brown{Tady je hnědý text na žlutém pozadí.}
```

**Vodoznakem** je míněn šedý text opakující se na každé stránce, který je vytištěn pod obvyklým textem. Například OPmac nabízí makro `\draft`, které způsobí, že každá stránka obsahuje šikmo napsaný veliký šedý nápis DRAFT. Můžete se inspirovat v technické dokumentaci, jak je to uděláno.

## Klikací odkazy

Pokud napíšete na začátek dokumentu `\hyperlinks{⟨color-in⟩}{⟨color-out⟩}`, pak se v dokumentu při výstupu do PDF stanou klikacími:

- čísla generovaná pomocí `\ref` a `\pgref`,
- čísla kapitol, sekcí, podsekcí a stránek v obsahu,
- čísla nebo značky generované pomocí `\cite` (odkazy na literaturu),
- texty tištěné pomocí makra `\url`.

Poslední z uvedených odkazů je externí a bude mít barvu `⟨color-out⟩`, zatímco ostatní čísla jsou interními odkazy a budou mít barvu `⟨color-in⟩`. Příklad:

```
\hyperlinks{\Blue}{\Green}    % vnitřní odkazy modré, URL zelené
```

Je možné zobrazit rámečky ohraničující aktivní plochu pro klikání. Tyto rámečky jsou viditelné jen v PDF prohlížeči, při tisku na tiskárně se nezobrazují. Stačí těmto rámečkům „namíchat“ barvu (tentokrát RGB) a definovat některé ze sekvencí `\pgborder`, `\tocborder`, `\citeborder`, `\refborder` a `\urlborder`. První část jména kontrolní sekvence určuje, jakých odkazů se to týká. Příklad:

```
\def\tocborder{1 0 0} % odkazy v obsahu vlevo: červený rámeček  
\def\pgborder{0 1 0} % odkazy na stránky: zelený rámeček  
\def\citeborder{0 0 1} % odkazy na publikace: modrý rámeček
```

Implicitně tato makra nejsou definována, což znamená, že se rámečky netvoří.

Cíl odkazu lze specifikovat makrem `\dest[⟨typ⟩:⟨lejblák⟩]` a klikací text makrem `\link[⟨typ⟩:⟨lejblák⟩]{⟨color⟩}{⟨text⟩}`. Parametr `⟨typ⟩` je typ odkazu (toc, pg, cite, ref nebo další). Parametr `⟨výška⟩` určuje vzdálenost cíle nad účařím.

Makro `\url{⟨text⟩}` vytiskne odkaz do internetu. Text je psán strojopisem a může se lámat do řádků za lomítky. Například `\url{http://petr.olsak.net}` vytvoří `http://petr.olsak.net`. Je-li nastaveno `\hyperlinks`, stává se tento text aktivním vnějším odkazem.

Obsah dokumentu se dá přesunout do levé záložky PDF prohlížeče tak, že klikáním na něj se přechází v dokumentu na požadované místo. Ve specifikaci PDF se tomu říká „outlines“. Makro, které uvedenou věc zařídí, se jmenuje `\outlines{⟨úroveň⟩}`. Záložky budou implicitně rozevřeny do `⟨úrovně⟩` včetně. Takže při `⟨úroveň⟩=0` jsou vidět jen úrovně kapitol, při `⟨úroveň⟩=1` vidíme i sekce a při `⟨úroveň⟩≥2` vidíme rozevřeno všechno. Bohužel písmo v záložkách typicky

nezvládá správně česká a slovenská písmena. Proto OPmac konvertuje texty do záložek tak, že tam jsou bez hacku a carek. Chcete-li vypnout tuto konverzi, napište `\def\cnvhook#1{}`.

Samotný řádek do záložek vložíte makrem `\insertoutline{<text>}`. Text v tomto případě nepodléhá konverzi. V sazbě se neobjeví nic, jen se stane cílem, když uživatel na záložku s `<textem>` klikne. Obsah se do záložek vloží celý během činnosti makra `\outlines`, takže další řádky vložené pomocí `\insertoutline` tomuto obsahu předcházejí nebo následují podle toho, zda předcházejí nebo následují místu, kde je použito `\outlines`.

## Verbatim texty

Vytisknout část textu verbatim „tak jak je“ bez interpretace speciálních znaků lze v prostředí vymezeném makry `\begtt` a `\endtt`.

Je-li před zahájením `\begtt` nastaven registr `\ttline` na nezápornou hodnotu, bude makro číslovat řádky. První řádek má číslo `\ttline+1` a po práci makra se registr `\ttline` posune na číslo posledního vtištěného řádku. Takže v dalším prostředí `\begtt ... \endtt` číslování pokračuje tam, kde přestalo. Implicitně je `\ttline=-1`, takže číslování neprobíhá.

Odsazení každého řádku v `\begtt... \endtt` je nastaveno na `\ttindent`. Tento registr má výchozí hodnotu rovnou `\parindent` (v době čtení souboru `opmac.tex`). Vertikální mezera nad a pod verbatim výpisem je vložena makrem `\ttskip`.

Makro `\begtt` zahájí skupinu a v ní nastaví všem speciálním znakům plain- $\TeX$ u kategorii 12. Pak spustí makro `\tthook`, které je implicitně prázdné. V něm je možno nastavit další kategorie znaků podle potřeby. Definicí aktivních znaků je potřeba udělat pomocí `\adef<znak>{<text>}`. Normální `\def` nefunguje, důvod je vysvětlen v TBN, str. 26. Příklad:

```
\def\tthook{\adef!{?}}
```

```
\begtt
```

Nyní se každý vykřičník promění v otazník. Že nevěříte? Vyzkoušejte!

```
\endtt
```

Jednou definovaný `\tthook` funguje ve všech verbatim výpisech, dokud jej nepředefinujete jinak. Tipy:

```
\def\tthook{\typosize[9/11]} % jiná velikost verbatim výpisů  
\def\tthook{\ttline=0} % všechny výpisy číslovány od jedničky  
\def\tthook{\adef{ }{\char` \ }} % místo mezer budou vaničky
```

Verbatim může být i v řádku uvnitř odstavce. Pomocí `\activettchar<znak>` si uživatel zvolí znak, který bude aktivní a bude zahajovat i končit verbatim výpisy uvnitř odstavce. Verbatim výpis se v odstavci nikdy nerozlomí (je v boxu). Autor makra OPmac obvykle nastavuje `\activettchar"`, takže pak může psát třeba toto:

Je-li před zahájením "\begtt" nastaven registr "\ttline" na ...

Znak nastavený pomocí \activettchar má lokální platnost a ruší se také pozdějším nastavením \activettchar na jinou hodnotu. Při zahájení každého řádkového verbatim výpisu se spustí makro \intthook, které je implicitně prázdné. **Upozornění:** deklaraci \activettchar⟨znak⟩ proveďte až po přečtení všech makrosouborů. Důvod: \activettchar nastavuje ⟨znak⟩ jako aktivní, což může při čtení souborů maker vadit.

Verbatim listingy je možné tisknout z externího souboru. Například

```
\verbatim (12-42) program.c
```

vytiskne ve stejné úpravě, jako při použití \begtt, ... \endtt, řádky 12 až 42 ze souboru program.c. Parametry v kulaté závorce mohou vypadat také takto:

```
\verbatim (-60) program.c % výpis od začátku souboru do řádku 60
\verbatim (61-) program.c % výpis od řádku 61 do konce souboru
\verbatim (-) program.c % výpis celého souboru
\verbatim (70+10) program.c % čtení od řádku 70, přečteno 10 řádků
```

V dalších ukázkách OPmac čte od řádku, který následuje za naposledy přečteným řádkem souboru z předchozího volání \verbatim. Je-li soubor čten poprvé, začíná číst prvním řádkem. Tento prvně čtený řádek je označen v komentářích jako n.

```
\verbatim (+10) program.c % čtení deseti řádků od řádku n
\verbatim (+) program.c % čtení od řádku n do konce souboru
\verbatim (-5+7) program.c % vynechá 5 řádků, od n+5 tiskne dalších 7
\verbatim (-3+) program.c % vynechá 3 řádky, tiskne do konce souboru
```

Narazí-li čtení na konec souboru dřív, než je vytištěno vše, co si žádá uživatel, přepis souboru je ukončen a žádná chyba se nezjeví.

Výpisy provedené makrem \verbatim jsou ovlivněny registrem \ttindent a makrem \tthook stejně, jako i prostředí \begtt... \endtt. Při \ttline<-1 se netisknou čísla řádků. Je-li \ttline=-1, čísluje se podle řádků souboru. Je-li \ttline nezáporné, čísluje se od \ttline+1.

## Jednoduché tabulky

L<sup>A</sup>T<sub>E</sub>Xoví uživatelé jsou zvyklí při vymezení pravidel zarovnávání v tabulce používat deklarace typu {cclr}. Každé písmeno vymezí jeden sloupec v tabulce, přitom písmeno c znamená centrováný sloupec, l je sloupec zarovnaný doleva a r sloupec zarovnaný doprava. Podobnou možnost deklarace jednoduchých tabulek nabízí OPmac v makru \table{⟨deklarace⟩}{⟨data⟩}. Příklad:

```
\table{||lcr||}{\cr1
Měsíc & zboží & cena\hfil \crli \tskip.2em
```

```

leden      & nořas      & 14 kKř      \cr
ůnor       & skejt       & 2 kKř       \cr
řervenec  & jachtiřka    & 3,4 MKř     \crl}

```

Uvedený pŕíklad povede na následující vŕsledek:

Měsíc	zboží	cena
leden	nořas	14 kKř
ůnor	skejt	2 kKř
řervenec	jachtiřka	3,4 MKř

Ve skutečnosti vŕsledek nebude uprostřed řádku, ale tam, kam `\table` napíšete. Kromě písmen `c`, `l`, `r` se v *(deklaraci)* mohou objevit znaky „svislítko“, které vymezují svislou čáru mezi sloupci.

Místo symbolu pro konec řádku `\cr` je možno použít `\crl` (přidá jednoduchou vodorovnou čáru) nebo `\crl1` (přidá dvojitou čáru), `\crl1i` (přidá čáru přerušenou svislými dvojitými linkami) a `\crl11i` (přidá dvojitou čáru přerušenou svislými dvojitými linkami). Těsně za `\cr`, `\crl` atd. může následovat `\tskip<dimen>`, což vytvoří vertikální mezeru velikosti *<dimen>*, přitom se nepřeruší svislé čáry v tabulce.

Za povšimnutí stojí, že v ukázce u slova „cena“ je připojeno `\hf1l`, což vloží pružnou mezeru vpravo od položky. Protože sloupec `r` obsahuje implicitní stejnou pružnou mezeru vlevo, je slovo „cena“ centrováno, zatímco ostatní údaje ve sloupci jsou zarovnány napravo.

Makro `\table` pracuje s předdefinovanými hodnotami, které můžete změnit, pokud chcete dosáhnout jiný vzhled tabulky:

```

\def\tabiteml{\enspace} % co vkládá vlevo každé datové položky
\def\tabitemr{\enspace} % co vkládá vpravo každé datové položky
\def\tabstrut{\strut}   % podpěra vymezující výšku řádků
\def\vvkern{1pt}       % velikost mezery mezi dvojitou svislou linkou
\def\hhkern{1pt}       % velikost mezery mezi dvojitou vodorovnou linkou

```

Vyzkoušejte si tabulku po `\def\tabiteml{\}\def\tabitemr{\}`. Sloupce budete mít na sebe nalepeny bez mezer. Příklad definice `\tabstrut`:

```

\def\tabstrut{\vrule height11pt depth3pt width0pt}

```

Tento příklad vymezuje v tabulce vzdálenost mezi účařím 14pt, z toho 11pt je rezervováno pro přetahy nad účařím a 3pt pro přetahy pod účařím. Vyskytne-li se větší písmeno, zvětší to v daném místě řádkování.

OPmac definuje `\strut` závislý na zvoleném řádkování (při použití příkazu `\typosize`) zhruba takto:

```

\def\strut{\vrule height.709<baselineskip>depth.291<baselineskip>width0pt}

```

Tip: zkuste `\def\tabiteml{\$}\enspace} \def\tabitemr{\enspace$}`. Ty dolary způsobí, že každá datová položka bude zpracována v matematickém módu. Makro `\table` se nyní podobá L<sup>A</sup>T<sub>E</sub>Xovému prostředí `array`.



Makro `\frame{<text>}` vytvoří rámeček kolem textu s vnitřními okraji o velikosti `\vbkern` a `\hhkern`. Například `\frame{ahoj}` vytvoří ahoj. Povšimněte si, že účarí rámovaného textu zůstalo nezměněno. Pokud chcete mít tabulku s dvojitými čarami, je výhodné ji vytvořit po stranách a nahoře a dole s jednoduchými čarami a celou ji zabalit do `\frame`:

```
\begtt
\frame{\table{|c|||r|}{\crl
 \multispan3\vrule\hss\bf Nadpis\hss \vrule\tabstrut \crl
 \noalign{\kern\hhkern}\crl
 první & druhý & třetí \crl
 sedmý & osmý & devátý \crl}}
\endtt
```

<b>Nadpis</b>		
první	druhý	třetí
sedmý	osmý	devátý

Tloušťka všech čar je v  $\TeX$ u implicitně 0,4pt. Makro `OPmac` umožňuje tuto implicitní tloušťku nastavit jinak pomocí `\rulewidth=<šířka>`, například `\rulewidth=1.5pt`.

Další příklad použití makra `\table` najdete na straně 24. Pokud potřebujete vytvořit komplikovanější tabulky, nezbude než prostudovat TBN, kapitolu čtvrtou.

## Vkládání obrázků

Nejprve je potřeba nastavit šířku obrázku do registru `\picw` a pak je možné použít makro `\inspic <jméno>.<přípona>`. Obrázek se vloží jako `\hbox` dané šířky `\picw`. Registr `\picw` si ponechá svou hodnotu, takže další obrázek bude mít stejnou šířku, pokud ji nezměníte. Přípony souboru s obrázkem mohou být: `png`, `jpg`, `jbig2`, `pdf`.

Obrázek je vyhledán v adresáři `\picdir`. Toto makro je implicitně prázdné, tj. obrázek je vyhledán v aktuálním adresáři.

O umístění obrázku v sazbě se musíte postarat vlastními prostředky. Například:

```
\picw=.5\hsize \centerline{\inspic hodiny.jpg }\nobreak\medskip
\caption/f Hodiny na brněnském náměstí Svobody.
```

Makro není vhodné použít při opakovaném použití stejného obrázku v dokumentu (opakující se grafika na každé straně nebo obrázek jako odrážka ve výčtu položek). V takovém případě je vhodnější natáhnout obrázek do PDF dokumentu jen jednou pdf $\TeX$ ovým příkazem `\pdfximage` a dále opakovat jeho zobrazení na různých místech dokumentu pomocí `\pdfrefximage`. Dokumentace k pdf $\TeX$ u řekne víc.

Makro `\inspic` pracuje jen v pdf $\TeX$ u při výstupu do PDF. Pokud máte nastaven výstup do DVI, můžete použít makro `epsf.tex`. Vzhledem k omezeným možnostem (obrázek jen ve formátu EPS) není tento způsob práce s obrázky v makru `OPmac` podporován.

## PDF transformace

Veškerá sazba v pdfTeXu může podléhat lineární transformaci, která je daná transformační maticí `\pdfsetmatrix{⟨a⟩ ⟨b⟩ ⟨c⟩ ⟨d⟩}`. Tato matice se v lineární algebře zapisuje do dvou řádků:

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix}, \text{ např. zvětšení: } \begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix}, \text{ nebo rotace: } \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix}.$$

PdfTeXové primitivy `\pdfsetmatrix`, `\pdfsave` a `\pdfrestore` bohužel nejsou v dokumentaci pdfTeXu uvedeny, tak je musím dokumentovat aspoň zde. Příkaz `\pdfsave` uloží stávající transformační matici a aktuální bod sazby. V době konání příkazu `\pdfrestore` se matice vrátí do původní podoby a aktuální bod sazby v té době musí být na stejném místě, jako byl v době `\pdfsave`, jinak se nám sazba rozjede a pdfTeX nadává. Toho se dá docílit např. pomocí `\pdfsave... \rlap{⟨text⟩} \pdfrestore`. Transformační matice se nastavují pomocí `\pdfsetmatrix`. Opakované použití `\pdfsetmatrix` způsobí pronásobení transformační matice novou maticí, takže to funguje jako skládání zobrazení. OPMac nabízí 2 užitečná makra `\pdfscale{⟨vodorovně⟩}{⟨svisle⟩}` a `\pdfrotate{⟨úhel⟩}`. Parametr `⟨úhel⟩` je interpretován ve stupních. Tato makra provedou odpovídající `\pdfsetmatrix`.

## Poznámky pod čarou a na okraji

**Poznámku pod čarou** vytvoříte pomocí `\fnote{⟨text⟩}`. V místě tohoto zápisu v textu se objeví automaticky generovaná značka a pod čarou dole na stránce je tato značka zopakovaná a vedle ní je `⟨text⟩`.

Značka je implicitně definovaná jako číslo v exponentu následované závorkou. Číslování poznámek je na každé stránce započato jedničkou. Čísla jsou vygenerována správně až po opakovaném TeXování. Při prvním zpracování jsou místo čísel otazníky.

Implicitní značkování je možné změnit předefinováním makra `\thefnote`. Nápříklad po

```
\def\thefnote{\ifcase\locfnum\or  
*or**or***or$^{\dag}$\or$^{\ddag}$\or$^{\dag\dag}$\fi}
```

bude první poznámka mít hvězdičku, druhá dvě hvězdičky atd. Uvedená definice předpokládá, že na jedné stránce nebudete mít více než 6 poznámek.

Makro `\fnote` je možné zapsat jen v běžném textu odstavce, nikoli v boxu (například v tabulce). Chcete-li odkazovat třeba z tabulky, je nutné v tabulce vytvořit jen značky a mimo tabulku (ovšem tak, aby to neuteklo na jinou stránku) zapíšete texty poznámek. K vytvoření značky použijte `\fnotemark{⟨číslo⟩}` a text (bez značky) vytvoří `\fnotetext{⟨text⟩}`. Příklad:

```

{\typoscale[/1200]\table{||lc|r||}{\crl1T
Měsíc      & zboží                               & cena\hfil \crlM \tskip.2em
leden      & noťas\fnote\mark1      & 14 kKč    \cr
únor       & skejt\fnote\mark2      & 2 kKč     \cr
červenec   & jachtička\fnote\mark3  & 3,4 MKč   \crl1B}}
\par\nobreak
\fnote\text{notebook}\fnote\text{scateboard}\fnote\text{jachta}

```

Čísla za slovy `\fnote\mark` jsou vztažena vzhledem k jedné tabulce a nemusejí souviset se skutečným číslem poznámky. Například, je-li na stejné stránce nad tabulkou z ukázky normální `\fnote`, bude mít vytištěno číslo 1, odkazy v tabulce budou mít čísla 2, 3, 4 a případná další poznámka pod tabulkou na stejné stránce obdrží číslo 5.

**Poznámku na okraji stránky** vytvoříte pomocí sekvence `\mnote{<text>}`. Poznámka je vlevo (na pravou zarážku) na sudé stránce a je vpravo (na levou zarážku) na liché stránce. Tuto vlastnost mají poznámky až po opakovaném  $\TeX$ ování. Při prvním  $\TeX$ ování jsou všechny poznámky vpravo. Chcete-li mít poznámky i při opakovaném  $\TeX$ ování jen vpravo nebo jen vlevo, pište do úvodu dokumentu `\fixmnotes\right` nebo `\fixmnotes\left`.

Řídící sekvenci `\mnote{<text>}` můžete napsat do odstavce nebo před odstavec. S odstavcem samotným to nic neudělá. Řádek odstavce, kde je `\mnote` vložena jako neviditelná značka, je na stejné úrovni, jako první řádek textu poznámky.

Text poznámky je od sazby odsazen o `\mnoteindent` a maximální šířka poznámky je `\mnotesize`. Text poznámky se rozlomí do více řádků, aby nepřesáhl `\mnotesize`.

Není ošetřen případ, kdy je `\mnote` víceřádková a je umístěna na úroveň například posledního řádku strany. Pak text poznámky přechuhuje poněkud dolů ze strany. Je tedy nutné `\mnote` použít jen na velmi krátké poznámky a případně si tento jev pohlídat a ošetřit před definitivní sazbou manuálně.

## Bibliografické údaje

Pomocí `\cite{<lejblik>}` nebo `\cite{<lejblik1>,<lejblik2>,<lejblik3>}` (atd.) vytvoříme v textu odkazy na položky v seznamu literatury. V seznamu literatury je třeba uvést záznamy, které mají odkazované lejbliky. Tyto záznamy dostanou v seznamu automaticky vygenerovaná čísla a sekvence `\cite` se pak promění na číselné odkazy, například [27] nebo [18, 24, 42] (atd.). Souvislé řady čísel [1, 2, 3, 5, 6] se promění v intervaly [1–3, 5–6] jen tehdy, když je v úvodní deklaraci dokumentu napsáno `\shortcitations`.

Seznam literatury je možné vložit do dokumentu třemi různými způsoby:

- Manuálně: pomocí jednotlivých položek `\bib{<lejblik>}` přímo v dokumentu.
- S využitím Bib $\TeX$ u makrem `\usebibtex{<bib-báze>}{<bst-styl>}`.

- Využitím jednou vygenerované databáze makrem `\usebbl/⟨typ⟩ ⟨bbl-báze⟩`.

Jednotlivé způsoby jsou níže probrány podrobněji.

**Manuálně vložený seznam literatury** v dokumentu vypadá například takto:

```
\bib[tnb] P. Olšák. {\it\TeX{}}book naruby.} 468~s. Brno: Konvoj, 1997.
```

```
\bib[tst] P. Olšák. {\it Typografický systém \TeX.}
269~s. Praha: CSTUG, 1995.
```

Výše uvedená ukázka dá následující výstup:

[1] P. Olšák. *TeXbook naruby*. 468 s. Brno: Konvoj, 1997.

[2] P. Olšák. *Typografický systém TeX*. 269 s. Praha: CSTUG, 1995.

**Využití BibTeXu.** Předpokládá se, že uživatel má soubor `⟨bib-báze⟩.bib`, ve kterém jsou nashromážděny bibliografické údaje ve formátu, v jakém je čte program BibTeX. V TeXové distribuci jistě najdete nějaký `*.bib` soubor, tak se do něj podívejte. Lejblíkem je první údaj u každého bibliografického záznamu. Soubor `⟨bib-báze⟩.bib` by měl obsahovat bibliografické údaje, které jsou nadmnožinou toho, co potřebuje uživatel vypsat ve svém dokumentu. Na místo, kde budete chtít vypsat seznam literatury, vložte následující pokyn:

```
\usebibtex{⟨bib-báze⟩}{⟨bst-styl⟩}
```

Parametr `⟨bib-báze⟩` je jméno souboru bez přípony `.bib`, ve kterém jsou připraveny bibliografické záznamy. Parametr `⟨bst-styl⟩` je jméno stylového souboru bez přípony `.bst`, který použije BibTeX pro konverzi ze zdroje `⟨bib-báze⟩.bib` do výstupu `⟨dokument⟩.bbl`. Tento výstup pak bude makrem `\usebibtex` přečten a vložen do dokumentu. Typicky používané `⟨bst-styly⟩` jsou `plain`, `alpha`, `apalike`, `ieeetr`, `unsrt`. Styl `alpha` způsobí, že se místo čísel začnou zjevovat v dokumentu zkratky, a to jednak v seznamu literatury a jednak v místě výskytu příkazů `\cite`. V takovém případě pochopitelně `\shortcitations` nefunguje a pokud byste se o něj pokusili, tak makro havaruje. Existují desítky, možná stovky dalších `.bst` stylů, viz internet.

Při prvním zpracování dokumentu makro `\usebibtex` připraví vstupní pokyny pro BibTeX do souboru `<dokument>.aux` a zjistí, že `<dokument>.bbl` zatím neexistuje. To dá najevo na terminálu:

```
WARNING: .bbl file doesn't exist.
```

```
Use the "bibtex <dokument>" command.
```

Přejděte tedy na příkazový řádek a napíšeme `bibtex <dokument>`. Tím se spustí program BibTeX, který přečte ze souboru `⟨dokument⟩.aux` vstupní pokyny (kterou otevřít `.bib` databázi, který `.bst` styl a jaké lejblíky jsou požadovány) a na základě toho vygeneruje soubor `⟨dokument⟩.bbl`, který obsahuje výběr jen těch záznamů, které byly uživatelem citovány pomocí `\cite`. Soubor `⟨dokument⟩.bbl` je

navíc zkonvertovaný z .bib formátu do formátu čitelného T<sub>E</sub>Xem. Tato konverze je řízena stylem .bst.

Když znovu T<sub>E</sub>Xujete dokument, makro `\usebibtex` v tomto případě shledá, že soubor `<dokument>.bbl` existuje, načte jej a vytvoří seznam literatury. Seznam obsahuje jen citované položky. Druhé spuštění T<sub>E</sub>Xu obvykle nestačí, protože příkazy `\cite` jsou typicky dopřednými referencemi, takže zatím nemají ponětí o přiřazení čísel k `<lejblíkům>` v seznamu literatury. To se dozvedí až v místě použití `\usebibtex`, což je typicky na konci dokumentu. Takže teprve třetí T<sub>E</sub>Xování dá vše do pořádku.

Seznam literatury obsahuje po použití BibT<sub>E</sub>Xu jen citovaná dílka. Pokud chcete do seznamu zařadit další položky, které nejsou v textu explicitně odkazovány příkazem `\cite`, použijte `\nocite[<lejblík>]`. Toto makro dá BibT<sub>E</sub>Xu pokyn, aby do seznamu zahrnul i položku s `<lejblíkem>`, ale v místě použití tohoto makra se nevytiskne nic. Konečně pomocí `\nocite[*]` dáváme BibT<sub>E</sub>Xu vzkaz, že chceme mít v seznamu literatury celou .bib databázi.

Zdroj bibliografických záznamů může být ve více .bib souborech. Pak stačí jejich názvy oddělit čárkou: `\usebibtex{<bib-báze1>,<bib-báze2>}{<bst-styl>}`.

Někdy se stane, že autoři .bib databází nebo .bst stylů neopustili při tvorbě těchto souborů L<sup>A</sup>T<sub>E</sub>Xový způsob myšlení a občas jim uklouzne nějaká L<sup>A</sup>T<sub>E</sub>Xová konstrukce z prstů až do počítače. Odtud se dostane do čteného .bbl souboru a náš plainT<sub>E</sub>X si s tím nebude vědět rady. K tomu slouží seznam `\bibtexhook`, kde můžete uvést definice těchto L<sup>A</sup>T<sub>E</sub>Xových konstrukcí. Tyto definice budou mít lokální platnost jen při čtení .bbl souboru. Například

```
\def\bibtexhook{\def\emph##1{\em##1}}\def\frac##1##2{{##1\over##2}}
```

**Využití jednou vygenerované databáze.** Tvorba seznamů literatury pomocí BibT<sub>E</sub>Xu má jistou nevýhodu. Pokud později do dokumentu vložíte další `\cite[<lejblík>]`, musíte veškerou anabázi s BibT<sub>E</sub>Xem provést znovu. A protože v současné době probíhá inflace odborných publikací způsobená tím, že se podle kvanta publikací a citačí daňový poplatník rozhodl odměňovat vědce, je každé zjednodušení práce s bibliografickými záznamy přínosné. Makro OPmac navrhuje řešení, při kterém stačí použít BibT<sub>E</sub>X pro mnoho nových článků jen jednou.

1. Vytvořte si zvláštní dokument `<mojebáze>.tex`, do kterého napíšete:

```
\input opmac \genbbl{<bib-báze>}{<bst-styl>} \end
```

2. Po T<sub>E</sub>Xování dokumentu `<mojebáze>.tex` spusťte `bibtex <mojebáze>`. Tím se vytvoří soubor `<mojebáze>.bbl`.
3. Zpracujte T<sub>E</sub>Xem soubor `<mojebáze>.tex` ještě jednou. Vytvoří se seznam veškeré literatury, který byl v souboru `<bib-báze>.bib`, přitom každá položka je označena svým `<lejblíkem>`. Vytiskněte si tento výstup a dejte si jej na nástěnku.
4. Uložte soubor `<mojebáze>.bbl` někam, kde jej umí přečíst T<sub>E</sub>X bez ohledu na to, v kterém pracujete adresáři.

5. Přejděte k editaci svého dokumentu, pište `\cite` nebo `\nocite` podle potřeby a v místě seznamu literatury dejte sekvenci `\usebbl/⟨typ⟩ ⟨moжебáze⟩`. Údaj `⟨typ⟩` má tyto možnosti:

```
\usebbl/a ⟨moжебáze⟩ % vypsát kompletně celou ⟨moжебaze⟩ (a=all),
\usebbl/b ⟨moжебáze⟩ % jen cite údaje řadit dle ⟨moжебaze⟩ (b=base),
\usebbl/c ⟨moжебáze⟩ % jen cite řadit podle pořadí citace (c=cite).
```

Kroky 2 až 4 budete muset opakovat pouze tehdy, když budete chtít přidat do `⟨moжебáze⟩.bbl` další údaj, tj. po aktualizaci souboru `⟨bib-báze⟩.bib`. Prudí-li různí odběratelé vaší vědecké činnosti požadavky na různé `⟨bst-styly⟩`, stačí si vygenerovat podle různých stylů různé soubory `mybbl-plain.bbl`, `mybbl-ieee.tr.bbl` a další.

## Okraje

Plain $\TeX$  nastavuje levý okraj 1 in a šířku sazby (`\hspace`) nastavuje tak, aby i pravý okraj při formátu papíru „letter“ byl 1 in. Také horní a dolní okraj (do kterého přesahuje záhlaví a stránková číslice) jsou nastaveny na 1 in při formátu papíru „letter“ a tím je určena výška sazby (`\vsize`).  $\mathcal{C}\text{Splain}$  dělá totéž, tj. okraje jsou 1 in, ale formát papíru je A4.<sup>1</sup> OPmac umožňuje toto nastavení změnit příkazem:

```
\margins/⟨pg⟩ ⟨formát⟩ (⟨levý⟩,⟨pravý⟩,⟨horní⟩,⟨dolní⟩)⟨jednotka⟩
```

například:

```
\margins/1 b5 (2,2,2,2)cm % všechny okraje na 2 cm pro papír b5.
```

```
⟨pg⟩... 1 = shodné okraje pro všechny stránky,
```

```
⟨pg⟩... 2 = okraje pro liché stránky, sudé prohozují ⟨levý⟩/⟨pravý⟩,
```

```
⟨formát⟩... a3, a4, a5, b5, letter nebo uživatelem definovaný,
```

```
⟨levý⟩,⟨pravý⟩,⟨horní⟩,⟨dolní⟩... velikosti okrajů,
```

```
⟨jednotka⟩... mm, cm, in, pt, pc, bp, dd, cc.
```

Každý z parametrů `⟨levý⟩`, `⟨pravý⟩`, `⟨horní⟩`, `⟨dolní⟩` může být prázdný. Jsou-li prázdné oba `⟨levý⟩` i `⟨pravý⟩`, je zachováno nastavení `\hspace` a levý i pravý okraj je stejný. Je-li jen jeden z parametrů `⟨levý⟩`, `⟨pravý⟩` prázdný, zůstává zachováno `\hspace` a neurčený okraj se dopočítá. Jsou-li `⟨levý⟩` i `⟨pravý⟩` neprázdné, jsou oba okraje určeny a je podle nich upraveno `\hspace`. Analogické pravidlo platí pro `⟨horní⟩`, `⟨dolní⟩` v souvislosti s výškou sazby `\vsize`. Například

```
\margins/2 a4 (,18,,)mm % vnější okraj na dvojstraně 2*a4 je 18mm
% \hspace, \vsize beze změny.
```

---

<sup>1</sup>Přesněji:  $\mathcal{C}\text{Splain}$  nastavuje výšku sazby `\vsize=239.2mm`, což vede k dolnímu okraji o 7 mm většímu než 1 in.

Uživatel může před použitím `\margins` definovat vlastní *⟨formát⟩* papíru pomocí příkazu `\sdef{pgs:⟨formát⟩}{⟨šířka⟩ ⟨výška⟩}`. OPmac například implicitně definuje:

```
\sdef{pgs:a4}{(210,297)mm}   \sdef{pgs:letter}{(8.5,11)in}
\sdef{pgs:b5}{(176,250)mm}
```

Celou sazbu na úkor okrajů je možno zvětšit nebo zmenšit pomocí makra `\magscale[⟨factor⟩]`. Například `\magscale[500]` zmenší sazbu na polovinu. Při této změně zůstává na místě „Knuthův bod“, tj. bod o souřadnicích 1 in, 1 in od horního a levého okraje. Sazba samotná je zalomena zcela stejně. Jednotky použité v dokumentu jsou od této chvíle relativní. Například po `\magscale[2000]` je použitá jednotka v dokumentu 1mm ve skutečnosti 2mm. Makro `\magscale` ponechává nezměněny jen rozměry stránek dané formátem stránek (A4, A3, atd.). Příklad použití makra: `\magscale[1414] \margins/1 a4 (,,)mm` umístí sazbu, která je určena pro tisk na A5, doprostřed stránky A4 a odpovídajícím způsobem ji zvětší, aby se to korektorům lépe četlo.

## Reference

- [1] L. Lamport. *LaTeX: A Document Preparation System: User's Guide and Reference Manual*. Addison-Wesley, Reading, Massachusetts, second edition, 1994.
- [2] M. Goossens, F. Mittelbach, a A. Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.

## Summary: OPmac — The Macro Package to Enrich the Possibilities of PlainTeX

The OPmac package is set of simple additional macros to plainTeX. It enables to users to take advantage of basic LaTeX functionality: the font size changing, automatic creation of table of contents and index, working with bibliography databases, tables, references including hyperlinks option, etc. In this paper, the significant properties of the OPmac are shown. The complete source of the macros and user and technical documentation are available on the <http://petr.olsak.net/opmac.html>.

**Key words:** OPmac, plainTeX extension macros, Csplain.

*petr@olsak.net*