

Zpravodaj Československého sdružení uživatelů TeXu

Ladislav Lhotka

Popis formátu dismac

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 1 (1991), No. 2, 24–33

Persistent URL: <http://dml.cz/dmlcz/148779>

Terms of use:

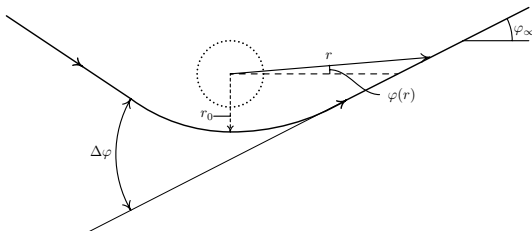
© Československé sdružení uživatelů TeXu, 1991

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>

způsobenému relativistickými účinky), dobře je však vidět řada zajímavých efektů, které METAFONT umožňuje.



Příklad ukazuje, že METAFONT umí nakreslit mnoho důležitých pomocných objektů, jako jsou šípky s libovolnou orientací, tečkované a čárkované čáry, atd.

Do zbraně!

Jedním z cílů tohoto článku je zažehnout oheň v srdcích horlivých mistrů METAFONTu. Komunita uživatelů $\text{T}_{\text{E}}\text{X}$ a METAFONTu by mohla mít užitek z vytvoření balíku maker, jenž by se měl k METAFONTu stejně, jako se má $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ nebo $\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}_{\text{E}}\text{X}$ k $\text{T}_{\text{E}}\text{X}$. Kdo se hlásí?

Literatura

- [1] Hobby, John D. A METAFONT-like system with PostScript output. *TUGboat* 10(4), str. 505–512, 1989.
- [2] Simpson, Richard O. Nontraditional uses of METAFONT. Sborník *T_EX: Applications, Uses, Methods* (editor Malcolm Clark), Ellis Horwood Ltd., London, 1990, str. 259–272.
- [3] Wilcox, Patricia. Metaplot: Machine-independent line graphics in $\text{T}_{\text{E}}\text{X}$, *TUGboat* 10(2), str. 179–187, 1989.

(Překlad: Danuše Lhotková)

Popis formátu `dismac`

Tento příspěvek popisuje formát `dismac`, který se hodí pro přípravu rozsáhlejších matematických a technických textů, jako jsou diplomové a disertační práce, různé zprávy a s určitými drobnými úpravami také knihy. Tak, jak je zde popsán, je určen pro psaní českých textů, ale velmi snadno jej lze modifikovat pro psaní ve slovenštině, v angličtině nebo jiném jazyku.

Před tím, než jsem na tomto formátu začal pracovat, jsem pochopitelně zvažoval, zda se pro daný účel nedá použít některý z obecně dostupných formátů.

Formát `plain` je na příliš nízké úrovni a nenabízí prakticky žádnou podporu pro logické strukturování textu, křížové reference apod. Jeho smyslem totiž je poskytnutí základních nástrojů, s jejichž pomocí je možno efektivně budovat specializované

soubory maker, jakým je třeba `manmac`, vytvořený D. Knuthem pro psaní série *Computers & Typesetting*.

Na základně plain $\text{T}_{\text{E}}\text{X}$ u je postaven také populární $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, který se snaží řešit většinu problémů spojených s kvalitní přípravou a sazbou technických textů. Jeho velmi obecné zaměření však vede k tomu, že téměř každý jeho uživatel aktivně používá jenom část z nabízených možností a zbytek pro něho představuje balast, který zabírá paměť a prodlužuje výpočet. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ důsledně využívá princip prostředí (*environment*), který je elegantní a pro začátečníky užitečný, avšak v řadě případů je spíše na obtíž—zejména při vytváření matematických vzorců, kde je podle mne *plain* $\text{T}_{\text{E}}\text{X}$ podstatně lepší a častá nekompatibilita $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u brání ve využívání *know-how* pro sazbu matematiky, jež je obsaženo v $\text{T}_{\text{E}}\text{X}$ booku, kap. 16–19.

Někde uprostřed mezi plain $\text{T}_{\text{E}}\text{X}$ em a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ em se nachází $\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}_{\text{E}}\text{X}$, který zachovává (až na malé výjimky) zpětnou kompatibilitu s *plain* $\text{T}_{\text{E}}\text{X}$ em. Je vhodný pro psaní matematických článků pro rozsáhlejší publikace s členěním do kapitol mu však chybí větší podpora logických struktur textu.

Návrh formátu `dismac` byl založen na následujících požadavcích:

- Logické členění textu do kapitol, oddílů a pododdílů;
- Další logické struktury (různé druhy seznamů, poznámky pod čarou aj.);
- Křížové reference, zejména na matematické vzorce, definice, věty apod.;
- Přiměřený výběr fontů a prostředky pro jejich přepínání;
- Automatická kompilace obsahu;
- Podpora pro zpracování rejstříku;
- Prostředky pro vkládání ilustrací do textu
- Dodatečná inteligence pro korekturní fázi
- Co nejvyšší míra zpětné kompatibility s formátem *plain*

Při psaní maker jsem se nechal inspirovat řadou zdrojů, ke hlavním přitom patřil formát `manmac` popsáný v $\text{T}_{\text{E}}\text{X}$ booku (Appendix E) a vynikající kniha [1].

Členění textu

Formát `dismac` umožňuje členit text do kapitol, oddílů a pododdílů, které jsou označovány standardním způsobem, například kapitola 3, oddíl 3.2 a pododdíl 3.2.5. Kapitola je ve zdrojovém textu vymezena řídicími slovy `\chapter{<název>}` a `\endChapter`, kde se jako parametr makra, `\chapter` uvádí název kapitoly. Makro `\chapter` má alternativní tvar s volitelným parametrem

$$\backslash\text{chapter}[\langle\text{číslo}\rangle]\{\langle\text{název}\rangle\}.$$

Tento tvar se použije v případě, že chceme dané kapitole přiřadit `<číslo>` mimo pořadí. Na začátku každé kapitoly se vypisuje při běhu $\text{T}_{\text{E}}\text{X}$ u na monitoru `Chapter <číslo>:`.

Jméno kapitoly se vypisuje také v záhlaví stránek. Pokud by bylo příliš dlouhé a nevešlo se dobře na jednu řádku, je třeba uvést náhradní (zkrácené) jméno, a to ihned za makro `\chapter` a jeho parametry pomocí deklarace `\headName{<zkrácenéjméno>}`.

Podobným způsobem jako kapitoly se vymezují oddíly (`\section{<název>}`) `\endSection`), které musí být vždy uvnitř kapitoly, a pododdíly (`\subsection{`

`\endSubsection`), které musí být vždy uvnitř oddílu. Také v těchto případech je možné změnit posloupnost číslování pomocí volitelného parametru v hranatých závorkách ihned za řídicím slovem.

Je sledováno řádné spárování a vnoření příslušných dvojic pro začátek a konec struktury a v případě nesrovnalosti se vydá chybové hlášení.

Kromě toho je možno používat zvláštní útvary na úrovni kapitoly, které se vymezují řídicími slovy `\looseChapter{<název>}` a `\endLooseChapter`. Od kapitol se liší tím, že nejsou číslované (ani nemění čítač kapitol) a nemohou obsahovat žádné oddíly a pododdíly. Hodí se proto například pro předmluvu, seznam literatury, rejstřík a podobně.

Další logické struktury

Formát *plain* umožňuje vytvářet seznamy položek maximálně o dvou úrovních pomocí dvojice maker `\item` a `\itemitem`. Značka pro vyznačení položky (odrážka) se přitom explicitně uvádí jako parametr. Položka přitom nemůže být členěna do odstavců. Formát `dismac` definuje několik druhů seznamů, přičemž používá strategii uvedenou v knize [1]. Každý seznam začíná řídicím slovem, v němž je vyznačen typ seznamu, a končí odpovídajícím řídicím slovem, jež začíná na `\end...` Položky jsou vyznačovány řídicím slovem `\item`¹. K dispozici jsou tyto základní druhy seznamů:

- `\ALPHAList... \endALPHAList`—položky jsou značeny velkými písmeny vzestupně podle abecedy, tj. A., B., C., atd.
- `\bulletList... \endBulletList`—položky odraženy značkou `•`.
- `\dashList... \endDashList`—položky odraženy pomlčkou (`—`).
- `\numberList... \endNumberList`—položky jsou číslovány vzestupně arabskými čísly.
- `\romanList... \endRomanList`—položky jsou číslovány vzestupně římskými čísly (minuskulemi), tj. (i), (ii), atd.

Tedy například zdrojový text

```
\bulletList
\item První položka
\item Druhá položka
...
\endBulletList
```

vytvoří seznam, v němž jsou položky odraženy značkou `•`.

Zvláštním případem je seznam vyznačený dvojicí

```
\labelList{<vzor>} ... \endLabelList{<vzor>},
```

který umožňuje používat libovolné značky explicitním zadáním. Parametr `<vzor>` (který musí být totožný na začátku a na konci konkrétního seznamu), představuje šablonu pro značky položek. Počítá se z něho šířka odsazení textu v seznamech, a proto

¹ Toto je jedna z mála konstrukcí nekompatibilních s *plain* T_EXem.

je obvykle třeba jej volit tak, aby reprezentoval nejdelsí značku, která se v seznamu vyskytuje. Značky položek se tentokrát zapisují jako `\item{značka}`. Zde je příklad:

```
\labelList{Gandalf}
\item{Bilbo} První položka
\item{Gandalf} Druhá položka
\item{Thror} Třetí položka
...
\endLabelList{Gandalf}
```

Seznamy je možné do sebe libovolně vnořovat, za předpokladu správného párování začátků a konců seznamů. Možné je tedy třeba toto:

```
\numberList
\item ...
\dashList
\item ...
\item ...
\ALPHAList
\item ...
\item ...
\endALPHAList
\item ...
\endDashList
\endNumberList
```

Pokud jsou položky v seznamu krátké a vejdou se na jednu řádku, je možné zmenšit vertikální rozestupy mezi položkami uvedením příkazu `\tightList` ihned za začátek seznamu před první položkou. Rozestupy se pak zmenší.

Seznam je rovněž možno přerušit příkazem `\suspend..List`, kde místo teček je třeba uvést druh seznamu (tedy např. `\suspendBulletList`). Další text se pak sází v celé šířce sloupce. Obnovení výčtu položek seznamu se zařídí příkazem `\continue..List`, kde je opět třeba uvést typ seznamu (např. `\continueDashList`). Speciálním případem je opět seznam s explicitními značkami, kde se u příkazů `\suspendLabelList` a `\continueLabelList` uvádí jako parametr vzor, který seznam identifikuje. Například

```
\labelList{Gandalf}
\item{Bilbo} První položka
\suapendLabelList{Gandalf}
... Text mimo seznam ...
\continueLabelList{Gandalf}
\item{Gandalf} Druhá položka
\item{Thror} Třetí položka
...
\endLabelList{Gandalf}
```

Poznámky pod čarou se vytvářejí příkazem `\note{text}`. Jako značky se u těchto poznámek v rámci jedné kapitoly střídají symboly †, ‡, § a ¶. Tento systém tedy funguje v případě, že na jednu stránku nevyjdou více než čtyři poznámky pod čarou. Pokud se jedna poznámka vztahuje k několika místům v textu, je možno použít makro `\lastNote`, které vysadí v daném místě textu značku předchozí poznámky pod čarou. Uživatel sám musí ošetřit případné problémy vzniklé například zalomením strany mezi dvěma místy se stejnou značkou.

Matematické struktury

Formát `dismac` v současné podobě nabízí pět typů číslovaných a zvláště vysazených matematických struktur. Jsou to:

- Definice, které jsou vymezeny klíčovými slovy `\definition` a `\endDefinition`.
- Věty, vymezené klíčovými slovy `\theorem` a `\endTheorem`.
- Tvzení, vymezená klíčovými slovy `\proposition` a `\endProposition`.
- Lemmata, vymezená klíčovými slovy `\lemma` a `\endLemma`.
- Důsledky, vymezené klíčovými slovy `\corollary` a `\endCorollary`.

Všechny tyto struktury mají společné číslování (tedy nečísľují se nezávisle definice, věty atd., což považují za méně přehledné). Ve stejné posloupnosti jsou navíc i čísla matematických rovnic (vysazených na zvláštní řádce), u nichž je deklarována identifikační značka (viz dále). Rovnice i matematické struktury jsou tedy průběžně číslovány při levém okraji, a to dvojicí čísel oddělených tečkou: prvním je číslo kapitoly a druhým číslo objektu v rámci kapitoly.² Rozsah čísel vyskytujících se na dané straně je vyznačen v záhlaví této strany, což usnadňuje hledání.

Makra pro začátek matematických struktur mohou mít jeden nepovinný parametr uzavřený v hranatých závorkách. Tento parametr je identifikační značkou pro další případné odkazy na tento útvar. Například

```
\theorem[main] Necht  $x$  je ...
...
\endTheorem
```

Důkazy začínají řídicím slovem `\proof`, které zahájí nový odstavec a vypíše v kurzívě slovo Důkaz. Na konci důkazu se pak uvede řídicí slovo `\QED`, které vysadí černý čtvereček a vloží vertikální odskok.

Křížové reference

Formát `dismac` umožňuje odkazovat v textu jak na obecné textové struktury (kapitoly, oddíly, pododdíly, strany) a na matematické objekty (rovnice, definice, věty atd.).

Odkazy mohou být jak zpětné (na objekt, který je v textu před místem odkazu), tak i dopředné. Vyskytují-li se v textu jenom zpětné reference, potom obyčejně stačí jeden průchod `TEX`. Výjimkou jsou stránkové odkazy, které ze známých důvodů

² V každé kapitole se tedy čísľuje od jedničky

musí být řešeny přes diskový soubor příkazem `\write`. Dopředné odkazy pak obecně vyžadují dva průchody³.

Chceme-li odkazovat na nějakou textovou strukturu, musíme v požadovaném místě takto deklarovat identifikační značku odkazu:

$$\backslash\text{make} \dots \text{Ref}\{\langle\text{značka}\rangle\},$$

přičemž tečky v řídicím slově nahradíme typem odkazu podle dále uvedeného výčtu. Vlastní odkaz (může být udělán i vícekrát z různých míst) se pak provede pomocí makra `\dots\text{Ref}\{\langle\text{značka}\rangle\}`, kde namísto teček opět doplníme typ odkazu. Značkování různých typů odkazů je navzájem nezávislé, takže při použití stejné značky pro různé typy odkazů nedojde k jejich kolizi. Mohou se použít tyto varianty:

- Odkazy na kapitolu—místo se vyznačí pomocí řídicího slova `\makeChapRef`, odkazuje se pomocí `\chapRef`.
- Odkazy na oddíl—místo se vyznačí pomocí řídicího slova `\makeSecRef`, odkazuje se pomocí `\secRef`.
- Odkazy na pododdíl—místo se vyznačí pomocí řídicího slova `\makeSubsecRef`, odkazuje se pomocí `\subsecRef`.
- Odkazy na stranu—místo se vyznačí pomocí řídicího slova `\makePageRef`, odkazuje se pomocí `\pageRef`.

Odkazy na matematické objekty se řeší poněkud odlišně kvůli jejich větší frekvenci. Rovnice, na něž chceme odkazovat, musí před uzavírající dvojicí `$$` deklarovat značku pomocí `\eqTag` (*značka*). Například

$$$$ 1 + 1 = 1.9999998. \text{\eqTag approx} $$$$

Zde je identifikační značkou pro odkaz `approx`. Všimněme si, že parametr makra `\eqTag` neuzavíráme do složených závorek (je delimitován pomocí `$$`). Případné mezery mezi značkou a `$$` se ignorují. Identifikační značky pro definice, věty aj. se deklarují, jak bylo již uvedeno, pomocí volitelného parametru v makrech `\definition`, `\theorem`, atd.

Odkazy na matematické objekty se provádějí příkazem `\stuffRef{\langle\text{značka}\rangle}`. Navíc je možné jednoduše odkazovat na předchozí objekt makrem `\lastStuff` a na nejbližší následující objekt makrem `\nextStuff`.

Fonty

Výběr fontů je ve formátu `dismac` poněkud bohatší než ve formátu `plain`, zdaleka však v tomto ohledu nekonkuruje `LATEX`u. Předpokládá se, že hlavní část textu je psána deseti- anebo dvanáctibodovým písmem. Z důvodu obecné dostupnosti se používají EC-fonty vyvinuté N. Schwarzem na základě dohody o rozložení akcentovaných znaků, učiněné na setkání `TEX90` v Corku.

Nezávisle je možno přepínat velikost i typ písma. K dispozici jsou tyto velikosti:

- desetibodové: zapíná se příkazem `\tenPoint`
- dvanáctibodové: zapíná se příkazem `\twelvePoint`
- sedmibodové: zapíná se příkazem `\sevenPoint`

³ Takových odkazů bývá v textu obvykle málo a v průběhu korektury není nutno se jimi zabývat

- osmibodové: zapíná se příkazem `\eightPoint`
- čtrnáctibodové: zapíná se příkazem `\fourteenPoint`
- pětadvacetibodové: zapíná se příkazem `\twentyfivePoint`

Ve všech případech se k příslušné velikosti automaticky doplňují vhodné velikosti indexů 1. a 2. řádu (`subscript` a `subsubscript`).

Z typů písma jsou k dispozici:

- standardní románské (přepínač `\rm`)
- kurzíva (přepínač `\it`)
- položené písmo (přepínač `\sl`)
- tučné písmo (přepínač `\bf`)
- kapitálky (přepínač `\sc`)

Přepínání velikosti a typu je ortogonální, to znamená, že při změně velikosti zůstává předchozí typ a naopak.

Kompilace obsahu

Chceme-li vytvořit automaticky obsah publikace, musíme na začátku dokumentu uvést příkaz `\makeToc`. V takovém případě se vytvoří soubor `toc.tex`, do něhož makra pro logické členění textu automaticky zapisují informace pro tvorbu obsahu. Každá položka má tvar

$$\backslash\text{tocEntry}\{\langle typ \rangle\}\{\langle číslo \rangle\}\{\langle název \rangle\}\{\langle strana \rangle\}.$$

Parametr $\langle typ \rangle$ vyjadřuje úroveň dotyčné strukturní jednotky (0-„looseChapter“, 1-kapitola, 2-oddíl, 3-pododdíl). Po případné editaci je možno soubor s obsahem natáhnout na zvoleném místě pomocí `\input`. Přitom je třeba uvážit možné posunutí stránkování (proto je obvykle zapotřebí jeden průchod navíc).

Tvorba rejstříku

Pomocí formátu `dismac` lze připravit podklady pro zpracování rejstříku hesel. Pro tento účel je potřebné vyznačit v textu slova, která mají do rejstříku přijít. Existují dvě varianty:

- Uzavřeme-li nějakou část textu do uvozovek `"..."`, zapíše se tato jako heslo do rejstříku, ale zároveň zůstane v textu.
- Pokud za první uvozovky napíšeme hvězdičku, tj. `"*..."`, potom se uzavřený text zapíše jako heslo do rejstříku, ale ve vlastním textu se neobjeví.

Formát `dismac` tedy nedovoluje používat uvozovky `"` jako náhražku dvojice apostrofů `'` což by však nemělo být na závadu. Je rozumné uvádět jako hesla pouze holý text, tj. bez vyznačení typu písma apod. V opačném případě se totiž komplikuje třídění hesel podle abecedy.

Pokud na začátku dokumentu napíšeme `\makeIndex`, zapisují se rejstříková hesla do souboru `index.tex` ve tvaru

$$\backslash\text{indexEntry}\{\langle hesla \rangle\}\{\langle strana \rangle\}.$$

V průběhu korektur není obvykle nutno tento soubor vyrábět, neboť hesla pro rejstřík se zapisují také na okraj té strany, na níž se vyskytují (viz dále).

Soubor `index.tex` je potřeba před zařazením do publikace ještě upravit. Nejprve se utřídí hesla podle abecedy, spojí se výskyty stejného hesla na více stranách, a nakonec se případně některá hesla typograficky upraví (odlišný typ písma atd.).

Sazba rejstříku by měla začínat řídicím slovem `\index`, které zahájí dvousloupcovou sazbu a nastaví některé parametry pro tento speciální účel. Konec rejstříku je pak vyznačen řídicím slovem `\endIndex`.

Vkládání obrázků

Obrázek, který chceme do textu vložit, je potřeba napřed uvést do formy přijatelné pro \TeX , tj. buďto jako zvláštní font (vytvořený například pomocí `META-FONTu`), anebo jako grafický rastr podporovaný použitým DVI ovladačem. Formát `dismac` nabízí podpůrné makro pro druhý z uvedených způsobů, a to pro ovladače `DVIDRV`, které jsou součástí volně dostupného `em \TeX u`, který je zřejmě relativně nejlepší implementací \TeX u na PC. Ovladače umožňují vkládat soubory typu `.PCX` a `.MSP`. Úprava pro jiný ovladač s podobnými možnostmi je víceméně triviální.

Pomocí příkazu

$$\backslash\text{specFig}\{\langle\text{soubor}\rangle\}\{\langle\text{šířka}\rangle\}\{\langle\text{výška}\rangle\}\{\langle\text{značky}\rangle\} \backslash\text{par}$$

se vytvoří box nazvaný `\figure`, který obsahuje

- rastr obrázku obsaženého v souboru, jehož jméno je `\langle\text{soubor}\rangle`. Další dva parametry zadávají šířku a výšku boxu `\figure` (skutečná šířka s výška obrázku je přitom irelevantní). Další požadavky (umístění souboru s rastrem v určitém adresáři aj.) jsou dány ovladači `DVIDRV`.
- značky „vlepené“ do obrázku přímo \TeX em. Tyto značky se zadávají pomocí maker `\placeMark`, která se zapisují jako argumenty makra `\specFig` (viz výše). Tyto argumenty jsou delimitovány řídicím slovem `\par`, resp. vynecháním prázdné řádky (což bude asi častější případ).

Syntaxe makra `\placeMark` je následující:

$$\backslash\text{placeMark}\{\langle\text{značka}\rangle\}\{\langle x\rangle\}\{\langle y\rangle\}\{\langle\text{lorg}\rangle\}.$$

Parametr `\langle\text{značka}\rangle` udává symbol či text, který se má do obrázku vepsat, `\langle x\rangle` a `\langle y\rangle` pak souřadnice referenčního bodu, k němuž se má značka, napsat (počítáno od levého dolního rohu boxu `\figure`—jednotky jsou uváděny podle zvyklostí \TeX u). Parametr `\langle\text{lorg}\rangle` konečně určuje polohu značky vzhledem k referenčnímu bodu. Může nabývat hodnot 1–9 s tímto významem:

- 1 – text vpravo nahoře
- 2 – text vpravo a vertikálně centrován
- 3 – text vpravo dole
- 4 – text nahoře a horizontálně centrován
- 5 – text centrován v obou směrech
- 6 – text dole a horizontálně centrován
- 7 – text vlevo nahoře
- 8 – text vlevo a vertikálně centrován
- 9 – text vlevo dole

Pro ilustraci těchto poněkud komplikovaných maker poslouží následující příklad:

```

\specFig{obrazek.pcx}{8cm}{15cm}
\placeMark{$x_1$}{35mm}{72mm}{3}
\placeMark{$\rho(x_1)$}{45mm}{5mm}{6}

```

Podstatné je vynechání řádku za posledním makrem `\placeMark`.

Pro vkládání obrázků (připravených buď výše popsaným způsobem, anebo jakkoliv jinak) do textu je možno použít jednak makra plain T_EXu pro plovoucí vsuvky, tj. `\topinsert`, `\midinsert` a `\pageinsert`, jednak je k dispozici makro `\rightInsert`, které vkládá obrázek k pravému okraji stránky se zúžením sázeného sloupce v potřebné výšce. Přitom se předpokládá, že hotový obrázek je připraven v boxu `\figure`.

Makro `\rightInsert` nevytváří plovoucí vsuvku (to ani dobře nejde), takže obrázek se umístí tam, kde je makro uvedeno. Tímto místem musí být začátek odstavce. Hlavně je však třeba zajistit, aby na straně pro obrázek zbývalo ještě dost místa. Syntaxe makra `\rightInsert` je tato:

```
\rightInsert{<popisek>}
```

Text uvedený v parametru `<popisek>` se vysadí pod obrázkem do sloupce o šířce rovné šířce boxu `\figure`.

Módy zpracování, korekturní informace a jiné

Formát `dismac` rozeznává tři hlavní módy zpracování dokumentu:

- korekturní mód, který se volí řídicím slovem `\proofMode` na začátku dokumentu.
- finální mód (první běh), volený řídicím slovem `\firstFinal`.
- finální mód (druhý běh), volený řídicím slovem `\secondFinal`.

Korekturní mód se nestará o reference, které nelze vyřídit při jednom běhu T_EXu. Na levý okraj textu zapisuje některé dodatečné informace, které jsou pro korektury užitečné. Jsou to jednak identifikační značky matematických objektů (rovnic, vět atd.), vypisující se vždy v místě, kde se v textu dotyčný objekt nachází. Dále se v levém horním rohu vypisují rejstříková hesla, která se na dané straně nacházejí. Při dolním okraji se dále vypisuje datum a počet minut uplynulých od začátku dne.

První běh finálního módu zapisuje do souboru `xrefs.tex` ty křížové odkazy, které je třeba řešit dvojným průchodem a vynechává samozřejmě všechny výše popsané korekturní informace.

Druhý běh finálního módu načítá soubor křížových referencí vzniklý v prvním běhu, avšak zároveň vytváří nový soubor těchto referencí, který se jmenuje `comp.tex`. Pokud po skončení druhého běhu zjistíme, že se soubory `xrefs.tex` a `comp.tex` od sebe liší, je nutno druhý běh opakovat. Taková situace však nastává jen v singulárních případech.

Pomocí dodatečných příkazů `\makeToc` a `\makeIndex` lze požádat o výpis souborů s obsahem a rejstříkovými hesly (viz výše).

Je možné část textu sazet ve dvou sloupcích. Začátek dvousloupcové sazby se vyznačí řídicím slovem `\twoColumns` a konec pak řídicím slovem `\endTwoColumns`.

Závěr

Formát `dismac` je nadstavbou formátu `csplain`, který je českou mutací standardního `plain` s EC-fonty namísto odpovídajících řezů Computer Modern a českými vzory pro dělení. Formát je tedy možno předkompilovat INITEXem:

```
tex /i /8 /o /c:kam2cork &csplain dismac.
```

Lomítkové parametry jsou specifické pro \TeX . Parametr `tt /c` zadává jméno souboru `kam2cork.tcp` pro konverzi vstupního kódu Kamenických na rozložení znaků EC-fontů.

Formát `dismac` dávám tímto k dispozici individuálním i kolektivním členům \CTUGu . Makra obsažená v tomto formátu jsem odzkoušel při přípravě své disertační práce, což ovšem neznamená, že jsou bez chyb, anebo dokonce dokonalá. Případní zájemci o vyzkoušení tohoto formátu jsou vítáni, na požádání jim zašlu zdrojový soubor. Naleznou-li nějaké chyby, prosím, aby mi je sdělili, avšak nešířili modifikované verze pod stejným názvem.

Literatura

[1] Appelt W. *\TeX für Fortgeschrittene*. Addison-Wesley, Bonn, 1988.

(Ladislav Lhotka)

e-mail: lhotka@csearn

\TeX a grafika

V poslední době se stále častěji objevují dotazy, jak je \TeX schopen začlenit grafické obrázky do textu. Stejně často — snad dokonce i častěji — je tato otázka diskutována na různých \TeX ových setkáních a v časopisech o \TeX u.

Protože se i na mne stále obrazejí uživatelé \TeX u s tímto dotazem, pokusím se v tomto příspěvku nastínit základní postupy začleňování grafiky do textu. Přitom se budu snažit zdůraznit výhody a nevýhody jednotlivých způsobů tak, jak jsem se s nimi setkával při praktických aplikacích. Přirozeně toto nebude vyčerpávající přehled, ale měl by čtenáři (a především mně) dát alespoň základní orientaci v této tematice. Ještě je potřeba zdůraznit, že slabý zrak a nedostatečné technické vybavení mi dovoluje zabývat se pouze černobílými obrázky.

Podrobnější popis jednotlivých způsobů tvorby a zpracování obrázků se pravděpodobně objeví v jednom z příštích čísel \TeX bulletinu (což je zároveň výzva k tomu, aby ti z vás, kteří mají nějaké zkušenosti v této oblasti, se o ně podělili s ostatními čtenáři).

Obecně lze celý postup rozdělit do tří fází.

- i) Příprava obrázku, jeho převedení do elektronické formy.
- ii) Případné opracování obrázku do vhodného tvaru (konverze do formátu vhodného pro další zpracování, ...).
- iii) Začlenění obrázku do textu.