

Zpravodaj Československého sdružení uživatelů TeXu

Lincoln Durst

Vytváření rejstříků: Část 1

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 1 (1991), No. 4, 30–37

Persistent URL: <http://dml.cz/dmlcz/148820>

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>

- Při rozdělení obrazovky na více oken možnost rolovat oběma směry v předcházejícím okně (předcházejícím oproti oknu, ve kterém je kurzor, tj. které je aktivní), aniž by kurzor přecházel do jiného okna.
- Při rozdělení obrazovky na více oken možnost dělat ZOOM aktivního okna (tj. dočasné roztažení na celou obrazovku a uvedení do původního stavu).
- Při nataženém chybovém a zdrojovém souboru a rozdělení obrazovce na dvě okna (v aktivním okně zdrojový a ve druhém okně chybový soubor) nalézt v chybovém souboru další chybu, zobrazit ji a ve vstupním souboru umístit kurzor do řádku, ve kterém je chyba (jedná se o nalezení vykřičníku na začátku řádku v chybovém souboru).
- Umožňovat automatické uschovávání editovaného souboru po určitém čase nebo po vstupu určitého počtu znaků.
- Uživatel by měl mít možnost konfigurovat významové kombinace kláves podle svých zvyklostí, aby nebyl nucen učit se nové ovládání.
- Možnost určit pořadí a názvy extenzí souborů pro natahování. V případě, že je uveden pouze název souboru, bude natažen ten soubor, jehož extenze je uvedena v seznamu „nejvýše“. Pokud neexistuje žádný soubor s extenzí uvedenou v seznamu, založí se nový soubor zadaného jména (bez extenze).
- Při všem, co je uvedeno výše, by editor měl být malý a rychlý (rychle se spouštět a rychle provádět často užívané časově náročné operace — např. natahování souborů, hledání a nahrazování řetězců znaků, přesun bloků, a pod.).

(Oldřich Ulrych)
e-mail: ummou@csearn

(Jiří Veselý)
e-mail: ummjv@csearn

Vytváření rejstříků: Část 1

Lincoln Durst

Tři předchozí příspěvky této řady jsou v dřívějších číslech *Tugboatu* [10#3 (November 1989, strany 390–394), 11#1 (April 1990, strany 62–68) a 11#4 (November 1990, strany 580–588)]. Tato část je víceméně na předchozích nezávislá; přístup k problému je nicméně v podobném duchu. Ve skutečnosti některé myšlenky inspirovaly vývoj popsany v předchozích částech.

Při vytváření rejstříků pomocí $\text{T}_{\text{E}}\text{X}$ u jsou nutné přibližně čtyři hlavní kroky stejně jako za starých časů, kdy se rejstříky vytvářely ručně; a to 1) výběr položek, které jsou rozumnými kandidáty na rejstříkové položky, 2) výběr vhodného druhu jejich uspořádání (abecedního pro pořádání slov, číselného pro stránkování), 3) spojení, anebo vyloučení duplikátů, odstranění nesrovnalostí mezi položkami z různých částí knihy, které se po setřídění třeba octnou blízko sebe, a 4) uspořádání rejstříku do podoby, ve které je ho možno vytisknout. Doba se ve skutečnosti příliš nezměnila;

tyto kroky uvedené v přirozeném pořadí jsou stále i přes technický pokrok uvedeny v pořadí rostoucí obtížnosti.

Označit vhodné termíny pro rejstřík a zapsat je do souboru v pořadí, v jakém se objevují v textu, je snadné. Uspořádání tohoto souboru však není tak jednoduché jako uspořádání souboru maker pro položky v bibliografii, protože jsou spojeny s čísly stránek, a pokud nejste dost opatrní a mazání, jednoduché setřídění podle ASCII kódu vám třeba může dát

$$1 < 11 < 111 < 2 < 21 \dots$$

(znalci to nazývají lexikografickým uspořádáním). Po automatickém setřídění nastanou další problémy, které vyplývají z rozdílů psaní velkých a malých začátečních písmen, jednotného a množného čísla apod. A nakonec je potřeba rozlišit hlavní a vedlejší položky.

Záleží na vkusu, zda poslední problém budete řešit jako první, nebo jako poslední. Problém je to filosofický, protože se redukuje na to, zda má být logické rozlišení určeno před začátkem psaní (tzn.: že problém co uspořádat či přerovnat je tu zcela abstraktní otázkou není co pořádat]), nebo zda se tím budeme trápit až po sestavení seznamu položek. Dlouhodobí čtenáři *TUGboat* si mohli všimnout, že zdejší varianty „slepice a vejce“ jsou jen opakováním dilematu předloženého Richardem Southallem před šesti lety [5#2, November 1984, strana 80], zda je vhodné (nebo možné) vytvořit zdrojový text ještě předtím, než je jasná jeho grafická úprava při vytištění.

Tato část je věnována prvním dvěma krokům a pojednává jednak o vytvoření souboru poznámek, jednak o některých problémech, se kterými je možno se setkat při jejich výběru.

Vytvoření souboru poznámek. Zde si vezmeme zjednodušenou verzi Knuthova rejstříkového makra z *TEXbook* popsanou v Appendixu E.⁵

Knuth stanovil čtyři druhy položek svého rejstříku, např.

Arabic, by, \char, <dimen>.

Obyčejný smrtelník obvykle vystačí jen s jedním druhem položek. Jisté bude jednodušší to pochopit, jestliže omezíme pozornost jen na první z Knuthových čtyř případů. Jakmile porozumíte základnímu principu, pusťte se klidně vlastní cestou, pokud se náhodou rozhodnete napsat knihu o \TeX u.

Knuth to dělá tak, že označuje v textu souboru termíny, které chce zahrnout do rejstříku (*The TEXbook*, strany 415–416, 423–424). Označování se může provádět během psaní, anebo až při kontrole textu. Knuth o nich mluví jako o „index reminders“, protože tyto značky nebudou ještě automaticky tvořit konečné položky rejstříku; mnohé budou teprve editovány v dalším průběhu konstrukce rejstříku. Předpokládejme, že „Gauss“ je zmíněn v textu, potom slovo „Gauss“ bude v textu vyznačeno, i když odpovídající položka rejstříku se nakonec změní na

Gauss, Karl Friedrich (1777–1855) . . .

Knuthův způsob označení termínů v textu je buď $\sim\{\text{Gauss}\}$, nebo $\sim\sim\{\text{Gauss}\}$; v prvním případě je slovo „Gauss“ vytištěno v textu, zatímco ve druhém ne, ačkoliv

⁵ Jiné způsoby přípravy rejstříku byly popsány v dřívějších *TUGboatech*, počínaje prvním svazkem u září 1980 (1#1, Winograd & Paxton, Appendix A, str. 1–12, případně později v listopadu 1989 (10#3, David Salomon, str. 394–400). Pro čtenáře, které zajímají možné alternativy, by mohlo být srovnání těchto dvou pramenů užitečné.

je bráno v úvahu při vytváření rejstříku. Např. zdrojový text pro obr.1 obsahuje následující pasáž:

```
By means of  $\text{\textasciitilde}\{stops\}$  the performer has
within his pover a number of combinations
for varying the  $\text{\textasciitilde}\{tone\}$  and
 $\text{\textasciitilde}\{power, dynamic\}$ 
dynamic power.
```

V *TEXbooku* se položky druhé kategorie, které mají dvě „stříšky“, $\text{\textasciitilde}\dots$, označují jako „tiché“. (To je důvod — viz níže — pro umístění tohoto tichého vstupu právě před slovo (slova), jehož se to týká; jejich oddělení na zvláštní řádek může být pro čtenáře textu na obrazovce poněkud snazší, protože všechny ostatní řádky pak může přeskočít).

Knuth určil kód, kterým se poznámky zapisují do souboru (v pořadí, v jakém se slova vyskytují v textu), a kód, který provádí jejich výpis na okraji stránky, jak ukazuje výsledek na obr. 1. Zde je zkrácená verze [*The TEXbook*, strany 415, 423]:

```
%%% index.rem, first fragment %%%
\newinsert\margin \dimen\margin=\maxdimen
\count\margin=0 \skip\margin=0pt
\nevf\ifsilent \newwrite\inx
\immediate\openout\inx=\jobname.inx
\def\specialhat{%
  \ifmode\def\next{\textasciitilde}%
  \else\let\next=\beginxref
  \fi
  \next}
\catcode'\textasciitilde=\active
\let \textasciitilde=\specialhat
```

Nejprve určíme vsuvku ($\text{\textasciitilde}\insert$), která bude obsahovat seznam na okraji. Rozměr této vsuvky může být výška celé stránky ($\text{\textasciitilde}\maxdimen$), ale ve vertikálním prostoru zabírá nulovou výšku ($\text{\textasciitilde}\count\margin=0$), tedy nezabírá žádnou textovou plochu běžné stránky: jak jsme už nejednou viděli, $\text{\textasciitilde}\text{TEX}$ může bez viditelného odporu přijmout některá tvrzení, která si mohou hrubě odporovat. Pokud se chcete o vsuvkách (insecích), jež jsou v *TEXbooku* pojednány s charakteristickou stručností, dozvědět něco více, nahlédněte do článku Davida Salomona v listopadovém vydání *TUGboatu* z roku 1990 (D. Salomon, 11#4, stránky 588–605). Následuje $\text{\textasciitilde}\if$ -příkaz, který je určen k rozlišení mezi jedno- a dvoustříškovými položkami, a poté se otevře soubor, který obsahuje jednotlivé položky. Nakonec je \textasciitilde definován jako aktivní symbol a $\text{\textasciitilde}\specialhat$ jako jeho náhrada. Je samozřejmě nezbytné zdůraznit skutečnost, že \textasciitilde může být v matematickém módu použit obvyklým způsobem, takže v případě splnění podmínky $\text{\textasciitilde}\ifmode$ je $\text{\textasciitilde}\specialhat$ obvyklá \textasciitilde , pokud ovšem nejsme v matematickém módu, stane se z něj $\text{\textasciitilde}\beginxref$, následující položka v naší agendě [*dtto*, str. 423].

```
%%% index.rem, second piece %%%
\def\beginxref{\futurelet\next
\beginxrefswitch}
\def\beginxrefswitch{%
  \ifx\next\specialhat
```

```

\let\next=\silentref
\else
\silentfalse\let\next=\xref
\fi
\next}
\def\silentref^{\silenttrue\xref}

```

Tady musíme být opatrní a lstiví. T_EX našel $\hat{}$ mimo matematický modus, takže $\backslash next$ je nyní $\backslash beginxref$. První otázka, kterou si musí položit, je, zda následuje další $\hat{}$. Uvažme uvedené dva případy, $\hat{}$ {Gauss} a $\hat{}$ {Gauss}: v prvním případě dalším znakem, jenž T_EX najde, bude $\{ \hat{}$, zatímco v druhém případě to je „ $\hat{}$ “, tj. $\backslash specialhat$. Příkaz $\backslash futurelet$ (viz *The T_EXbook*, strana 307) T_EXu umožňuje zaň dosadit $\backslash next$; pak dojde k expanzi $\backslash beginxrefsvitch$. Jestliže nastal druhý případ, příkaz $\backslash silentref$ odstraní $\hat{}$, nastaví přepínač $\backslash ifsilent$ na pravdivou hodnotu a zavolá příkaz $\backslash xref$; v druhém případě se rovněž volá $\backslash xref$, ale podmínka $\backslash ifsilent$ zůstává nepravdivá. Diskuse o $\backslash futurelet$ na str. 207 je jistě příliš stručná; čtenáři, kteří by rádi viděli i jiné příklady jeho použití, mohou konzultovat poučný článek Stephana v. Bechtolsheima v *TUGboat* **9#3** December 1988, na str. 276 a dalších.] V každém případě se nejprve expanduje příkaz $\backslash beginxrefswitch$, po němž je správně nastaven přepínač $\backslash ifsilent$ a zavolá se $\backslash xref$.

```

Nyní je již čas na  $\backslash xref$ :          %% index.rem, third part %%
\def\xref#1{\def\text{#1}%
\let\next=\text\makexref}
\def\makexref{%
\ifproofmode\insert\margin{%
\hbox{\marginfont\text}}%
\xdef\writeit{\write\inx{%
\text\space!0\space
\noexpand\number\pageno.}}%
\writeit
\else
\ifhmode\kern0pt\fi
\fi
\ifsilent\ignorespaces\else\next\fi}

```

První dva řádky zde nahrazují patnáct (mnohem delších) řádek kódu uprostřed str. 424, kde dochází ke čtyřnásobnému větvení podle čtyř typů rejstříkových položek, s kterými Knuth kouzlí. Všimněte si, že dvě kopie textu v závorkách jsou vytvořeny pomocí $\backslash xref$, jedna z nich ($\backslash text$) je určena pro rejstříkový soubor a poznámku na okraj textu a druhá ($\backslash next$) je tou položkou, která se objeví v textu v případě $\backslash silentfalse$. Právě popsané kroky se uskutečňují pomocí $\backslash makexref$. Všimněte si $\backslash space!0\space$ v definici; to se používá k oddělení jednotlivých terminů pro rejstřík od čísla stránky, na které se položka objevila. Jediným kritériem pro výběr vhodného znaku pro oddělovač je, že by to mělo být něco, co se nikdy neobjeví jako kandidát pro zápis do rejstříku. Tady vyhovuje libovolná dostatečně nepravděpodobná kombinace znaků. Knuth pro své čtyři případy používá !0, !1, !2 a !3 a jeho označení se s naším shoduje. Tečka za $\backslash pageno$ označuje konec položky. Jestliže vaše číslování stránek obsahuje tečky (viz níže), měli byste ukončit svoji položku některým jiným znakem (např. :). Dvě jemnosti: (1) $\backslash kern0pt$ potlačuje dělení slov, pokud nejsme

v korekturním módu (`\ifproofmode`), to pro shodu s výsledkem v případě vsuvky (viz Appendix H, *The T_EXbook*, strany 454, 455). (2) `\ignorespaces` potlačuje každou mezeru, explicitní, nebo implicitní na konci položky „`^^Gauss`“, což znamená, že v textu

```
$1+i$ is a ^^Gauss gaussian prime.
```

nebo

```
$1+i$ is a
^^Causs
gaussian prime.
```

nemůže dojít k zalomení stránek mezi `^Caus` a `gaussian`: Tyto dvě položky jsou spolu pevně spojeny (jako `superlepidlem`, které ovšem nemá žádnou vůli).

K dokončení konstrukce obr. 1 musíme poněkud zacvičit s výstupem stránky, abychom dostali vsuvku na okraj, když jsme v korekturním módu. Knuthova verze je na str. 416, naše je o něco jednodušší, protože nezkoušíme dělat vše, co musel dělat on:

```
%%% index.rem, output routine %%%
\newdimen\pageheight \pageheight=\vsize
\def\onepageout{\shipout
  \vbox{\offinterlineskip
    \makeheadline\pagebody}\advancepageno}
\def\makeheadline{\vbox to 2pc{%
  \ifodd\pageno\rightheadline
  \else\leftheadline\fi\vfill}}
\def \pagebody{\vbox to\pageheight{%
  \ifproofmode\InsertNotes\fi
  \unvbox255}}
\output={\onepageout}
\newdimen\remkern \remkern=-8pc
\def\InaertNotes{\ifvoid\margin\else
  \rlap{\kern\remkern\vbox to 0pt
    {\box\margin\vss}}\fi}
```

Tím přepíšeme `\output` definovaný v `plain.texu`, neboť ho nahrazujeme příkazem `\onepageout`; příkaz `\offinterlineskip` skládá oba boxy (`\makeheadline` a `\pagebody`) těsně k sobě, aniž by ponechal mezi nimi jakýkoli prostor, a `\vbox255` je box, který obsahuje hotovou stranu textu. Příkaz `\InsertBotes` přidává obsah vsuvky `\margin`, je-li jaký. Konečně potřebujeme (víceměň) vhodný řez písma pro marginálie:

```
%%% index.rem, reminders in margin %%%
\font\remfont=cmtt8 \font\strutfont=cmr9
\newbox\rembox
\setbox\rembox=\hbox{\strutfont })
\def\remstrut{\vrule height 1\ht\rembox
  depth 1\dp\rembox width 0pt}
\def\marginfont{\remstrut\remfont}
```

Knuth dává tyto poznámky na pravý okraj, v našem případě jsme zvolili levou, protože pravá strana je již přepracována. Nepsaná moudrost profesionálních technických redaktorů (upravovatelů knih a jiných tištěných materiálů) opírající se o studii

čitelnosti říká, že tištěný text má obsahovat průměrně deset nebo méně slov v řádku. To dává pro šířku textu (`\hsize`) v případě desetibodových typů asi 24 cicer a nejvýše 30 cicer pro dvanáctibodové typy. To znamená, že jeden sloupec desetibodového textu na stránce formátu A4 by měl vždy mít levý a pravý okraj, jejichž celková šířka je skoro 10 cm, tj. téměř polovina šířky papíru, což nechává spoustu volného místa pro poznámky na obou stranách textu. Viz např. Southall, *dto*, str. 86. Rovněž *Designing with type* od Jamese Craiga (New York, Watson-Guptill; London, Pitman; opravené vydání 1980), str. 128. *Typography: How to make it most legible* od Rolfa F. Rehe (Carmel, Indiana, Design Research International; 5. vydání, 1984) obsahuje rozsáhlou bibliografii výzkumných zpráv týkajících se čitelnosti.⁶

Jako obvykle máme tu něco pro `prepare.tex`:

```
\newif\ifIndRemMark \IndRemMarkfalse
\def\IndexRemindersMarked{\IndRemMarktrue}
\newif\ifproofmode \proofmodefalse
\def\NriteIndexReminders{\proofmodetrue}
```

a něco pro `compose.tex`: `\ifIndRemMark\input index.rem\fi`

A navíc další dvě možnosti pro řídicí soubor:

```
\IndexRemindersMarked
\irriteIndexReminders
```

První volba vkládá do zdrojového souboru `index.rem`, aby se se mohl vypořádat se všemi `\specialhat` a druhá zapisuje označené položky do souboru a umožňují jejich vytištění na okraji textu v době, kdy je rejstřík připravován, ještě před závěrečným \TeX ováním. (O souborech `\prepare.tex`, `\compose.tex` a řídicích souborech pojednává druhá část tohoto seriálu, viz výše.)

Varování. Kvůli jednoduchosti se v rejstříkových položkách kromě běžné interpunkce vyvarujte všech znaků, jež nejsou písmeny, nebo se dočkáte příšerných zážitků při třídění souboru. Nepoužívejte žádná diakritická znaménka, vlnky (~), ani řídicí slova libovolného druhu a za každou cenu se vyvarujte matematického módu. Tato opomenutí mohou být opravena, když se rejstříkový soubor upravuje do konečné podoby. V nezbytném případě je nutno psát

```
... described in detail in his
^^{Heiligenstaat Testament}
Hei\~li\~gen\~staat Test\~a\~ment. ...
```

(Viz též Dodatek dále.)

Uspořádání souboru položek. První věc, kterou musíme se souborem udělat, je jeho abecední uspořádání. Pokud jste nadáni stručností a čísla vašich stránek nemají víc než jednu číslici (nebo když se z nějakého důvodu stane, že všechny mají shodný počet číslic), je problém jednoduchý; použijte třídící utilitu, kterou poskytuje váš operační systém. Jinak bude patrně potřeba těžší kanón.

Kernighan & Ritchie ve své knize *The C programming language* (2. vydání), Prentice-Hall 1988 (odstavec 5.6, strany 108–110) ukazují, jak třídít soubory pomocí Hoarova rychlého třídění. Jestliže dáváte přednost Shellovu třídění, podívejte se do

⁶ Je možné, že v USA jsou vysoké školy, jejichž záměry vyžadují, aby diplomka měla obtížně čitelnou či dokonce vůbec nečitelnou formu? Dozajista boj proti volnému šíření myšlenek!

knihy Harbison & Steele, *C: A reference manual* (2. vydání), Prentice-Hall 1987, strany 210, 211, na smíšenou Knuthovu-Sedgewickovu-Harbisonovu-Steelovu verzi a vložte ji do K & R kódu. V každém případě není těžké upravit kód tak, aby části řádků předcházejících „!0“ byly uspořádány abecedně a části následující „!0“ numericky. Trik spočívá ve změně kritéria pro prohazování řádků, aby dva řádky byly zaměněny; jestliže řetězce písmen (a) jsou ve špatném pořadí, nebo (b) jsou stejné, ale čísla stránek nejsou správně seřazena. Když jste umínění a trváte na stránkování ve formě 1-12 [IBM, tady 1-2 < 1-12], 2.17 [Zenith, tady 2.2<2.17], K-9 [PCT_EX], atd., následky si poneseťe sami. Můžete je ovšem chápat jako výzvu.⁷

Jak už jsem naznačil, existuje jiný způsob, jak chytit kočku za ocas, pokud čísla vašich stránek jsou stará dobrá přirozená čísla. Naštěstí jsou ASCII kódy (na rozdíl od telefonních číselníků a klávesnic psacího stroje) uspořádány tak, že 0 předchází všechny ostatní číslice. Připsáním dostatečného počtu nul před `\pageno` tak, aby všechna čísla stránek měla shodný počet číslic, můžeme dosáhnout shody lexikografického uspořádání s číselným. Jestliže nemáte víc než tisíc stránek (máte-li, je třeba vám požeňnat), můžete použít následující kód

```
\def\Pageno{%
  \ifnum\pageno>99
    \noexpand\number\pageno
  \else\ifnum\pageno>9
    0\noexpand\number\pageno
  \else
    00\noexpand\number\pageno
  \fi\fi}
```

a dejte „`\Pageno`.“ do definice `\makehref` místo „`\noexpand\number\pageno`.“ S touto změnou můžete použít třídící utilitu, kterou nabízí váš operační systém, a po netřídění můžete odstranit teď už nadbytečné nuly.

Dodatek. Při jiné příležitosti navrhuje uvažovat kroky, s jejichž pomocí lze setříděný soubor vybraných položek přeměnit na docela dobrý rejstřík. Je třeba počítat s tím, že nějaká „ruční práce“ bude nutná, i kdyby šlo pouze o vložení druhého jména, dat či jiných podrobností, jež autor považuje za relevantní nebo důležité. Ale pořád zbývá dost rutinní práce, která může být zautomatizována. Navíc při troše chytrosti je vždy možno omezení týkající se zakázaných diakritických znamének a jiných řídicích příkazů uvnitř vybraných položek nějak překonat.

Poznámka. Současná verze zdroje, o němž je zmínka v předchozí části tohoto seriálu, spolu s již zmíněnými soubory obsahuje i soubor `index.rem` a zdrojový text pro obr. 1.

Je příjemné, jako ostatně vždy, poděkovat za pomoc, vedení a podporu všem, kteří nějak přispěli ke vzniku právě uvedené série článků. Ke všem obvyklým pochybnostem (o nichž jsem se tu již několikrát zmiňoval) musím vyjádřit svoji vděčnost

⁷ Menší výzvu představují tiskové chyby v pasážích citovaných na začátku tohoto odstavce. V K & R kódu nahraďte `maxlines` výrazem `MAXLINES`; ve výkladu H & S nahraďte „nejmenší číslo“ spojením „největší číslo“. (Chyby se objevily přinejmenším v prvních výtiscích obou těchto druhých vydání.)

Costovi Mylonasovi z Brown University a z Atén (Řecko), který mne vyzval, abych vytvořil soubor pokud možno jednoduchých a pružných prostředků, jež by mohly být užitečné pro ty autory, kteří mají chuť sedět u svých počítačů a psát knihy. Série mých učebních textů je výsledkem této výzvy. Doufám, že budou pro někoho užitečné, nebo alespoň poučné.

Lincoln Durst
46 Walnut Road
Barrington, RI 02806
lkd@math.ams.com

(Překlad: *Blanka Koubová*)

Kolik časopisů je vydáváno T_EXem v Československu?

Vydáváte časopis sázený T_EXem a kladete si podobnou otázku? Při rozhovorech s lidmi zabývajícími se sazbou časopisů je tuto otázku často slyšet. Útržkovité informace a neúplný přehled je málo přesvědčivým důkazem o rozvoji T_EXu v Československu. Rád bych tedy opatřil potřebné informace nejen pro sebe, ale především pro vás a všechny zájemce, kteří váhají s přechodem časopisů na elektronickou sazbu pomocí T_EXu. Proto vás prosím o zaslání odpovědí na následující otázky:

1. Úplný název časopisu (ev. doplněný jeho zkratkou)
2. V jakém jazyku či jazycích jsou příspěvky přijímány.
3. Kdo časopis vydává (včetně adresy redakce).
4. Rozsah časopisu (kolikrát ročně vychází a rozsah stran jednotlivého čísla).
5. Náklad časopisu.
6. Od kterého roku(ročníku) je sázen Tem.
7. Jaké makro je při sazbě používáno.
8. Jaká je technologie výroby.
9. Stručná charakteristika časopisu a poznámky, které jsou podle vás užitečné.

Pro vaši lepší představu uvedu, jak jsem tyto položky vyplnil pro časopis CMUC:

```
\nazev      CMUC - Comentationes Mathematicae Universitatis Carolinae
\jazyk     výhradně anglicky
\vydavatel  Matematický ústav UK, Sokolovska 83,
            186 00 Praha 8 -- Karlín
\rozsah    4xročně, přibližně 200 stran/číslo
\naklad     600 výtisků
\texem     od roku 1989 (ročník 30)
\makro     AmSTeX a upravený styl amsppt.sty
\technologie Tiskové předlohy z-laserové tiskárny v hustotě 300 dpi jsou
            zpracovány dále fotocestou.
\poznamky  Specializovaný matematický časopis.
```