

# Zpravodaj Československého sdružení uživatelů TeXu

---

Jan Kybic; Karel Horák

Program IMARK: Nástroj usnadňující vytváření rejstříků v TeXu a LaTeXu

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 4 (1994), No. 4, 174–179

Persistent URL: <http://dml.cz/dmlcz/149726>

## Terms of use:

© Československé sdružení uživatelů TeXu, 1994

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>

---

---

# Program IMARK: Nástroj usnadňující vytváření rejstříků v T<sub>E</sub>Xu a L<sub>A</sub>T<sub>E</sub>Xu

JAN KYBIC

## 1. Úvod

Rejstřík je velmi důležitá a užitečná část každé vědecké či technické knihy. Umožňuje čtenáři velice rychle nalézt žádanou informaci. Bohužel ale jen málo dnes vydávaných knih rejstřík má, nemívají ho ani knihy vytvořené na počítači.

Zaměřím se nyní na dokumenty vysázené v T<sub>E</sub>Xu (čímž myslím i L<sub>A</sub>T<sub>E</sub>X a další makra). Uživatelé L<sub>A</sub>T<sub>E</sub>Xu mohou vytvářet rejstřík relativně jednoduše. Stačí označit ve zdrojovém textu všechna slova, která se v něm mají objevit. Po prvním průchodu L<sub>A</sub>T<sub>E</sub>Xem vznikne soubor `idx`, který je nutné programem `makeindex` (nebo `csindex`) převést na soubor typu `ind`. Po druhém průchodu L<sub>A</sub>T<sub>E</sub>Xem se už rejstřík objeví na svém místě. Uživatelé PlainT<sub>E</sub>Xu použijí v zásadě stejný přístup, pouze budou muset napsat několik jednoduchých maker.

Bez ohledu na další zpracování je však vždy nutné nějakým způsobem označit všechna slova, která přijdou do rejstříku. Někteří autoři doporučují po napsání dokumentu ho celý znovu projít a ručně označit každý významný výskyt klíčových slov a termínů. Je to ovšem zdlouhavá práce a je zde riziko chyb a přehlédnutí. Druhý přístup je proto použit k označování slov nějakým způsobem počítač. A to byl důvod k napsání programu *IMark*.

## 2. Popis činnosti

*IMark* umí vytvořit abecedně seřazený seznam slov ze zdrojových textů T<sub>E</sub>Xu. (Slovo se skládá z jednoho či více písmen). Je rovněž možné vytvořit jeden seznam z mnoha zdrojových souborů, máme-li například každou kapitolu knihy v jednom souboru, ale rejstřík chceme mít společný. Jelikož vzniklý seznam bývá opravdu rozsáhlý, je nutné jej ručně editovat. (Seznam je čistý ASCII textový soubor. Každá řádka obsahuje jedno slovo.) V seznamu ponecháme jen ta slova, která chceme zařadit do

rejstříku. Pokud do řádky v seznamu napíšete dvě slova, pak bude *IMark* ve zdrojovém textu vyhledávat první z nich a druhé slovo (zde je definice slova zobecněna na libovolný řetězec omezený mezerami) se stane argumentem indexovacího příkazu. Tímto způsobem lze ovlivnit formu konkrétní položky v rejstříku (např. změnit font) nebo posílat informace programu `makeindex`.

Máme-li připraven seznam slov patřících do rejstříku, zavoláme *IMark* ještě jednou, ovšem s jiným příkazem. *IMark* projde zdrojové texty, přičemž hledá slova ze seznamu. Pokud nějaké najde, připiše ze něj indexovací příkaz, což je implicitně `\index`. Argumentem tohoto makra je buď nalezené slovo samo, nebo druhé slovo na příslušném řádku v seznamu slov, jak je zmíněno výše.

Někdy bude nutné vzniklý zdrojový text pro  $\TeX$  lehce upravit, pokud například *IMark* umístí příkaz `\index` do argumentu nějakého makra, nebo pokud chceme zahrnout do rejstříku něco, co *IMark* nepovažuje za slovo. (*IMark* například odmítne zahrnout do rejstříku slovo  $\TeX$ , jelikož  $\TeX$  obsahuje `\`, což není písmeno.) Takové věci je nutné udělat ručně. Pokud by to *IMark* měl umět, bylo by nutné použít mnohem složitější, a tedy i pomalejší algoritmus.

A z tohoto bodu pokračujeme jako obvykle. Spustíme např.  $\LaTeX$ , potom `makeindex` a jsme hotovi (skoro).

### 3. Syntaxe

`IMARK` [přepínače] příkaz seznam [zdrojové soubory]

(*seznam* je soubor obsahující seznam slov, které mají být označeny ve zdrojových souborech pro  $\TeX$ )

**Přepínače** (předchází jim `,-` nebo `,/`):

- e** použij anglické řazení (implicitní je české)
- c** *cmd* nastav indexovací příkaz na *cmd* (implicitně `index`)
- o** *ext* nastav příponu výstupních souborů na *ext* (implicitně `new`)

**Příkazy:**

- h** zobraz nápovědu (anglicky)
- b** vytvoř nový seznam ze slov obsažených ve zdrojových souborech
- a** přidej do seznamu slova obsažená ve zdrojových souborech
- m** označ slova ze seznamu ve zdrojových souborech (*pro nové verze souborů použijte příponu a indexovací příkaz nastavený pomocí přepínačů*)

#### 4. Příklad

Celý proces ilustrujeme následujícím příkladem. Mějme zdrojový text pro  $\LaTeX$  nazvaný `test.tex`:

```
\documentstyle{article}
\begin{document}
Pro animaci je použita spline
interpolace.
\end{document}
```

Zavoláme-li *IMark* na tento soubor (`IMARK b test test`), dostaneme soubor `test.lw` se seznamem slov:

```
% List of words for IMARK, (C) Jan Kybic 1993
animaci
article
begin
document
interpolace
je
použita
spline
Pro
% End of the list
```

Tento soubor ručně upravíme na tvar:

```
% List of words for IMARK, (C) Jan Kybic 1993
animaci animace
interpolace
použita
spline interpolace!spline
% End of the list
```

Ještě jednou zavoláme *IMark* (`IMARK m test test`) a ten vytvoří ze souboru `test.tex` soubor `test.new`:

```
\documentstyle{article}
\begin{document}
Pro animaci\index{animace} je použita\index{použita}
spline\index{interpolation!spline}
interpolace\index{interpolace}.
```

```
\end{document}
```

A je to!

## 5. Podpora češtiny

*IMark* používá implicitně české třídění (anglické zapneme pomocí `-e`). Třídění je dvouprůchodové, v současné verzi řadím velká písmena až za malá, pokud by to někomu vadilo, stačí ve zdrojovém textu přepsat jedinou tabulku (`cstab` v modulu `csort.cpp`). Stejnou tabulku je nutné přepsat, chceme-li použít jiné kódování než Kamenických. V této souvislosti je nutné upozornit ještě na jednu vlastnost programu *IMark* (dodávám, že stejně úmyslnou jako ta předchozí) — začíná-li slovo velkým písmenem, např. na začátku věty, nevidí *IMark* žádnou souvislost s tímž slovem začínajícím malým písmenem.

## 6. Programátorské poznámky

*IMark* byl napsán v Borland C++ verze 2.0. Tvoří ho asi 600 řádek zdrojového textu (26 kB). Objektový přístup byl použit v maximální míře. Program používá objektovou knihovnu CLASSLIB, která poskytuje mnoho základních algoritmů pro práce s daty. Není ovšem naprogramována příliš efektivně, a proto bylo nutné v konečné verzi některé algoritmy přepsat, což však díky objektovosti nebylo obtížné. Připouštím, že výsledný program není tak malý a rychlý, jako by mohl být vzhledem k režii při práci s objekty, ale — něco za něco.

Rutiny českého třídění jsou obsaženy v jediném modulu a dají se proto použít i v jiných programech.

## 7. Závěr

Program lze získat pomocí anonymous ftp z `ftp.muni.cz` — je tam uložen v souboru `imark.zip`, což je zdrojový text, dokumentace a spustitelný MSDOS `imark.exe` soubor, vše zabalené PKZIPem. Přenos pod UNIX asi nebude jednoduchý, bylo by nutné buď přepsat, nebo přenést knihovnu CLASSLIB. Pod UNIXem však existují mocné nástroje, ze kterých se dá něco podobného jako *IMark* zbastlit. Pokud by byl přesto zájem, jsem ochoten se o konverzi pod UNIX časem pokusit.

Program dávám k dispozici jako *public domain* a věřím, že bude někomu užitečný. Uvítám samozřejmě jakékoli připomínky, návrhy na zlepšení a opravení chyb. Pokud použijete můj nápad nebo zdrojový text

jako základ vašeho programu, který bude jakýmkoli způsobem lepší než *IMark*, dejte mi, prosím, vědět.

## 8. Literatura

- [1] *Lamport Leslie*, MakeIndex: An Index Processor for L<sup>A</sup>T<sub>E</sub>X.
- [2] Dokumentace k programu MakeIndex.
- [3] *Wágner Zdeňek*, CsIndex v. 2.11 – Česká/Slovenská implementace MakeIndexu.
- [4] *Wágner Zdeňek*, Tvorba rejstříku, Zpravodaj C<sub>S</sub>TUGu 4/92.

*Jan Kybic*

Prosecká 681

190 00 Praha 9

e-mail: [kybic@earn.cvut.cz](mailto:kybic@earn.cvut.cz)

[kybic@csearn.bitnet](mailto:kybic@csearn.bitnet)

## Pár poznámek vděčného uživatele

Dovolím si drobný komentář k právě popisovanému programu, protože informace o něm mě zastihla právě v okamžiku, kdy se mi *velmi* hodil. Občas se totiž najde autor, který ví, co chce v rejstříku mít, ale nestíhá ani další dokončovací práce na knize, natož aby ještě jednou celé dílo pečlivě přečetl (i když už je půlka v tiskárně) a všechna slova odkazovaná v rejstříku v textu vyznačil. V případě jmenného rejstříku to dokonce snad ani zvláštní problémy nenesou, zvlášť vyskytuje-li se většina jmen jen jednou či dvakrát. V takovém případě velice pomůže, můžeme-li rychle zjistit, v jakém tvaru se příslušná slova v textu vyskytují, a pak si je nechat automaticky označit rejstříkovou značkou.

Příznám se, že stejně jako *IMark* píšu tuto značku obvykle rovněž na konec slova (dokonce jsem si na to napsal kopírovací makro v qeditu), ale z hlediska správného odkazu by asi bylo vhodnější psát odkazy před slovo, protože jen tak můžeme zabránit tomu, aby někde nezůstalo dole na stránce rozdělené slovo, na které bude rejstřík odkazovat následující stranou! Rejstříková makra v T<sub>E</sub>Xbooku (*manmac*) v sobě samozřejmě zahrnují zákaz dělení odkazovaných slov.

Co se týče třídící funkce zmíněného programu, nebylo by marné, kdyby bylo možno požadovat vypsání a utřídění jen těch slov, která začínají zvoleným písmenem (popříp. více vybranými písmeny). To by navíc notně zjednodušilo práci s rozsáhlejšími texty, protože počet slov, jež program

dokáže zpracovat v rámci jediného souboru, není z pochopitelných důvodů nijak závratný.

A nakonec pár „drobných nedostatků“ té verze, kterou jsem si stáhl z brněnského archivu v době, kdy jsem článek redigoval: program nevypisuje slova obsahující písmena s přehláskou (v kódu Kamenických, jiné kódování jsem nezkoušel) a plete si velká písmena O a P (to že třídí zvláště velká a zvláště malá počáteční písmena je někdy výhoda, někdy nevýhoda).

*Karel Horák*

---

---

## BMV — Jak na PostScript pod Linuxem

JAN KYBIC

Mnoho z nás již pro  $\text{T}_{\text{E}}\text{X}$ ování (a nejen pro  $\text{T}_{\text{E}}\text{X}$ ování) používá UNIX. Na počítačích třídy PC se často jedná o variantu zvanou **Linux**. K tomu nás vede kromě jeho nepopíratelné kvality i to, že je ho možné získat zadarmo, například pomocí **ftp**. Chcete-li se o **Linuxu** dozvědět více, zkuste si přečíst například *Bajt* číslo 46 nebo 41 z roku 1994.

Mnoho z nás také používá výstup z  $\text{T}_{\text{E}}\text{X}$ u do PostScriptu. Ptáte-li se proč, zalistujte několika posledními bulletinů, články o PostScriptu se to tam jen hemží. Převod souboru **dvi** do PostScriptu za nás udělá program **dvips**. Otázkou ovšem zůstává, jak si výsledný PostScriptový soubor prohlédnout. Ano, i pod UNIXem existuje volně šiřitelný PostScriptový interpret zvaný Ghostscript. Sám o sobě ale příliš komfortní není a o interaktivitě si můžeme nechat jen zdát. Zvětšení je možné pouze jediné — většinou zvolíme takové, aby se na obrazovku vešla celá stránka, pak je ale text příliš drobný, než aby se dal číst. Svou základní úlohu Ghostscript splní dobře — PostScriptový soubor převede na bitovou mapu, kterou pak podle naší volby buď zobrazí, pošle na tiskárnu nebo uloží do souboru. Co dělat, abychom se alespoň přiblížili ke komfortu Mattesova **dvicsr**?