

Zpravodaj Československého sdružení uživatelů TeXu

David Antoš

TeX Versioning System aneb jak všechny zdrojové soubory uložit

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 10 (2000), No. 1-3, 47–49

Persistent URL: <http://dml.cz/dmlcz/149868>

Terms of use:

© Československé sdružení uživatelů TeXu, 2000

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>

1. Convert a DVI file to a plain text.
2. Lines can be maximaly 32 characters long, so we have to split them and words, which are too long, have to be split too.
3. Put Brailles' marks into the text. (Mark the text.)
4. Convert the text to a character set which is suitable for printing on a braille printer.

The first step can be done with help of the perl module *DVIParse* (available from any CPAN site in the `authors/id/JANPAZ/` directory). The callbacks are defined for each of the DVI instructions in the module. Thus the only thing which you have to do is to program them. For the next step the perl module from the same author, called *hyphenation* (available on the same site as the *DVIParse*), can be used. There is defined the function *hyphenate* in the module. It takes a word as its argument and returns an array of numbers saying where the word can be hyphenated. The third step seems to be the hardest. Firstly we have to know the semantics of the text to put brailles' marks correctly, for example a fraction $\frac{3}{4}$. In the plain text the fraction takes 3 lines. The program doesn't know if it is a fraction or if there is underlined number 3 in the text and by chance there is number 4 under it. Each of these possibilities is marked differently. The last step is easy. You can do it with help of the script *ctocs* (in the case you are converting a czech text).

In my opinion this way is suitable only for converting a plain text not for structured text or text with higher mathematics.

T_EX Versioning System aneb jak všechny zdrojové soubory uložit

DAVID ANTOŠ

Tohle se vám už někdy muselo stát: Používáte T_EX v práci. Doma máte jinou distribuci. Doma jste pracně napsali a krásně zalomili dokument používající několik souborů s makry. Přišli jste do práce, zdroják na disketě, spustili T_EX a po chvíli I can't find file 'blabla.tex'. Naštvaně přemýšlíte, proč v instalaci v práci zrovna tenhle soubor chybí.

Nebo něco jiného. Máte vysázenou knihu, jasný bestseller, který určitě budete za rok či dva potřebovat vysadit znovu. Rádi byste si uložili všechny zdrojové

texty, na kterých vzhled dokumentu závisí. Takže se ponoříte do logu a hledáte je ručně. Tak to vás upřímně lituji, protože nevíte, že nemusíte.

Pro oba případy může posloužit **TVS**, **T_EX** Versioning System. Tento program umí projít logy **T_EX**u, najít v nich jména souborů (což zas není tak jednoduché) a tyto soubory zkopírovat do zadaného adresáře. Může najít nejen soubory potřebné přímo k překladu hlavního dokumentu, ale i soubory, které jsou potřeba k vygenerování formátu.

Přítom umí vynechávat soubory odpovídající regulárnímu výrazu – například se zadanými příponami. Navíc dokáže zacházet se jmény souborů docela inteligentně, dokáže zachovat strukturu vašich vlastních zdrojových souborů za předpokladu, že jsou umístěny v adresářové struktuře v podstromu adresáře, ve kterém je hlavní dokument.

Zatím experimentálně zařazujeme i možnost projít log **dvipsu**, který tento program vytváří většinou s volbou **-d**. Takto je možno získat fonty odkazované z **.dvi** souboru. Bohužel se takto dostanou pouze bitmapy, přesněji řečeno soubory **.pk**, což není příliš užitečné. Nicméně zkušený uživatel může získaný výpis použít jako vodítko pro ruční posbírání zdrojových souborů pro generování fontů.

Činnost **TVS** je samozřejmě limitována tím, co lze najít v logu. Bohužel trochu ruční práce zbývá, pokud dokument závisí na něčem, co lze sice považovat za zdrojový text, ale nepoužívá se to přímo v posledním běhu **T_EX**u. Příkladem může být soubor s bibliografickými údaji. Z něj se vyberou ty, na které je odkazováno z dokumentu a ty se třídí a doplní informacemi o stylu. Teprve tento derivát se začleňuje do dokumentu. Pokud chceme archivovat i původní soubor, nic nám nebrání ho k posbíraným zdrojům přidat.

Získané kompletní zdroje dokumentu je možno přeložit na jakékoli distribuci **T_EX**u. Stačí, aby cílová distribuce umožnila nastavit, aby **T_EX** hledal soubory nejdříve v aktuálním adresáři a jeho podadresářích a pak teprve v systémových adresářích **T_EX**u. Není třeba, aby systém **TVS** byl na cílovém stroji instalován.

TVS je napsán v Perlu ve snaze o maximální platformovou nezávislost. Je nezávislý na tom, jestli používáte **L^AT_EX**, plain nebo cokoli jiného. Zvládá konvence jmen souborů obvyklá v **M\$** světě i v **Un*xu**¹, počítá i s tak patologickými jevy, jako jsou mezery v názvech souborů, které se kupodivu vyskytují v **T_EX**u na Macu.

TVS jsem napsal jako bakalářský projekt na Fakultě informatiky Masarykovy university v Brně pod vedením a za vydatné podpory RNDr. Petra Sojky.

Domovská stránka je <http://www.fi.muni.cz/~xantos/TVS> a tamtéž si můžete **TVS** stáhnout včetně jeho podrobné dokumentace ve formátu Texinfo, na stránce je dokumentace dostupná také ve formátech HTML a PostScript. **TVS** je šířen zdarma za podmínek GPL licence.

¹měl jsem nutkání napsat „i v operačních systémech“ ...

Summary: T_EX Versioning System or How to Save All Source Files

TVS (T_EX Versioning System) collects all files needed to re-typeset T_EX documents exactly the same way on another T_EX installation. It is a platform-independent Perl script. TVS parses T_EX log files. This article discusses basic features of TVS, complete documentation is written in English.

Spolupráce databáze s L^AT_EXem

ZDENĚK WAGNER

Tento článek demonstruje výhody spojení databázového programu s L^AT_EXem. Předvádí, jak lze pomocí makrojazyka z téhož souboru generovat zcela odlišné výstupy s jiným využitím. Celý systém je předveden na programovém projektu, který je již řadu let využíván při organizaci konferencí CHISA.

Historie

Začátek použití výpočetní techniky při organizaci konferencí CHISA spadá již do šedesátých let. Tehdy to však byly počítače sálové, jejichž jediným vstupním médiem byly děrné štítky. Zpracování databáze, uložené na děrných štítcích, bylo tedy zcela neinteraktivní. Později byly údaje ukládány na magnetická média, ale vstupem nových dat i příkazů byly stále děrné štítky. Až koncem osmdesátých let se začaly používat osobní počítače, nejprve osmibitové s operačním systémem CP/M a později počítače šestnáctibitové se systémem MS-DOS. Konference měla tradičně okolo 1 500 příspěvků v několika odborných sekcích a v adresáři potenciálních zájemců bylo přibližně 16 000 záznamů. Počítač byl ovšem vybaven pouze jednotkami pro osmipalcové diskety, jejichž kapacita byla 256 KB. Taková databáze se na jednu disketu nevešla a komerční programy, které se navíc v tehdejších dobách sháněly velmi nesnadno, nedokázaly pracovat s databází rozložené na větším počtu disket. Byl proto vyvinut speciální databázový systém inspirovaný příkladem, který byl součástí kompilátoru Turbo-Pascal. Později byl tento systém přepsán do C++ a přeložen pro MS-DOS. Přes koncepční zastaralost se po mnoha postupných úpravách používá dodnes.

Správa takto rozsáhlé databáze vyžaduje značné úsilí a není pohodlné provádět všechny kontroly na obrazovce. Kontrolní výtisky však spotřebují velké