

# Zpravodaj Československého sdružení uživatelů TeXu

---

Martin Bílý

Jak implementovat TeX

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 2 (1992), No. 3, 122–128

Persistent URL: <http://dml.cz/dmlcz/149634>

## Terms of use:

© Československé sdružení uživatelů TeXu, 1992

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>

## Slovo na závěr

Předcházející příklady snád' dostatočně ilustrovali možnosti spomínaného balíka makier. Uvedené makrá môžu byť užitočné pre každého, kto potrebuje nakresliť diagram, či graf do svojho článku a pritom sa nechce (bezpochyby nádhernému!) METAFONTu a T<sub>E</sub>Xu venovať hlbšie. . .

## Literatúra

1. Kristoffer H. Rose, *Typesetting diagrams with X<sub>Y</sub>-P<sub>I</sub>C*, User manual, 1992.
2. M. D. Spivak, *The Joy of T<sub>E</sub>X*, American Mathematical Society, 1990.
3. M. J. Wichura, *The P<sub>I</sub>C<sub>T</sub>E<sub>X</sub> manual*, T<sub>E</sub>X Users Group, Providence, R.I., 1987.

Janka Chlebíková  
chlebi<sub>k</sub>j@mff.uniba.cs

---

## Jak implementovat T<sub>E</sub>X

---

Martin Bílý

Představte si situaci, kdy Vás navštíví přítel a poté, kdy mu na svém PC předvedete možnosti T<sub>E</sub>Xu, Vás požádá, zda byste mu „neudělal“ T<sub>E</sub>X i pro jeho ZX Spectrum. Tenhle příklad je sice trochu nadsazený, ale má naznačit, že se v článku budu zabývat postupem při implementaci T<sub>E</sub>Xu. Pokusím se objasnit, jaké prostředky jsou použity při tvorbě nové implementace a jak je zaručena jejich vzájemná kompatibilita. V závěru uvedu i rozšířený postup pro implementaci T<sub>E</sub>Xu pod operačními systémy UNIXového typu.

## Jazyk WEB

Velký kouzelník DEK při tvorbě T<sub>E</sub>Xu, METAFONTu a jejich podpůrných programů měl na mysli i jejich snadnou přenositelnost na jakýkoliv

typ počítače. V počtu existujících implementací ostatně tkví jeden z půvabů  $\text{T}_{\text{E}}\text{X}$ u. Z toho důvodu jsou psány v jazyce **WEB** vytvořeném právě pro projekt  $\text{T}_{\text{E}}\text{X}$ . Jazyk **WEB** je určen pro tvorbu snadno přenositelných rozsáhlých programových celků metodou shora dolů. Mezi jeho charakteristické vlastnosti patří modularita, používání maker a možnost vytvořit současně program i jeho dokumentaci včetně vzájemných referencí. Pokud jde o vlastní příkazy jazyka **WEB**, jsou velice podobné Pascalu.

Program v jazyce **WEB** se skládá ze sekcí. Sekce lze pro snadnou čitelnost programu sestavit z několika sekcí nižší úrovně. Jinak řečeno, sekce lze vzájemně vnořovat do sebe. U každé sekce může být připojena její dokumentační část. Nenechme se zmást podobností s procedurami z jiných programovacích jazyků. Ty jsou zde rovněž k dispozici. Dělení programu na sekce není úzce svázáno se strukturou programu tak jako je tomu u procedur. Sekce vytvářím pro své pohodlí programátora. Mohu například napsat sekci, ve které budu jen nastavovat počáteční hodnoty proměnných. Sekci začlením do úvodu hlavního programu nebo ji vsunu do některé inicializační procedury. Pokud později začnu psát jinou část programu, budu potřebovat další proměnné a budu je chtít inicializovat. Nic mi nebrání v tom abych prohlásil: zde pokračuje inicializační sekce, zařaď do ní následující příkazy. Z toho vyplývá, že posbírání částí sekcí roztroušených po celém programu a rozvinutí jejich vzájemného vnořování je vlastně jen textové zpracování, jehož analogii bychom našli v preprocesoru jazyka C. Vlastní členění **WEB**ovského programu na procedury je shodné s jinými jazyky tohoto typu. Pokud jde o používání maker, opět bych poukázal na analogii s jazykem C. Několika větami a bez ukázek není možno přesně zachytit všechny vlastnosti jazyka **WEB**. Případný zájemce nechť se laskavě obrátí na manuál `webman.tex`, kde je uvedena přesná definice jazyka.

## **Překladač tangle**

Pro práci s jazykem **WEB** jsou k dispozici dva překladače, `tangle` a `weave`. Překladač `tangle` program z jazyka **WEB** převádí na jeho ekvivalentní zdrojový text v jazyce Pascal. Protože na každém slušném počítači se v době vzniku celého projektu  $\text{T}_{\text{E}}\text{X}$  předpokládala existence pascalského překladače, byla takto elegantně otázka přenositelnosti vyřešena. Můžete ovšem namítnout, že se jedná jen o přesunutí problému někam jinam. Přes všechny snahy o dosažení jednotnosti Pascalu v jeho implementacích existují rozdíly. Jednak ve vazbě na vstupně/výstupní operace, ve

vazbě na operační systém i ve způsobu uložení datových struktur v operační paměti. Tyto potenciální problémy jsou řešeny dvěma cestami.

Jednak text generovaný tanglem využívá jen podmnožinu jazyka Pascal. Nejsou využity dynamické proměnné a konformní pole, zavádí se vlastní správa paměti a pro dosažení potřebné přesnosti na všech typech procesorů je použita i vlastní aritmetika. Druhá cesta se pak vztahuje k implementaci WEBovského programu pro konkrétní procesor, operační systém a konečně i kompilátor Pascalu. Tangle zpracovává dva vstupní soubory. Jedním je zdrojový soubor typu `.web`, druhým je změnový soubor typu `.ch` (*change file*). Ve změnovém souboru jsou zapsány dvojice *vyhledej* a *nahrad*. V případě části *vyhledej* se jedná o pasáže rozsahu obvykle od jedné do několika řádek, které se musejí přesně shodovat se zdrojovým souborem `.web`. Mohou být nahrazeny libovolným textem, nejčastěji konkrétními pascalskými příkazy.

Jako příklad si uveďme práci s binárními soubory, kde Pascal přiliš mnoho nástrojů neposkytuje, ale v jeho konkrétních implementacích bývají k dispozici. Případně mohou být poměrně snadno dostupné voláním služeb operačního systému (to ovšem záleží na té které implementaci Pascalu). V některých implementacích se ve změnovém souboru řeší převod bytově orientovaných operací se soubory na blokové operace, které jsou obvykle efektivnější.

Výstupem programu tangle je soubor typu `.pas` nebo `.p` s pascalským programem. V případě  $\text{T}_{\text{E}}\text{X}$ u a  $\text{M}_{\text{E}}\text{T}_{\text{A}}\text{F}_{\text{O}}\text{N}_{\text{T}}$ u je druhým výstupem soubor typu `.poo` (*string pool*).  $\text{T}_{\text{E}}\text{X}$  používá vlastní operace pro práci s řetězcy znaků proměnné délky. V tomto souboru jsou uloženy všechny řetězce použité v programu (diagnostické zprávy a základní řídicí příkazy).

Vytvořením konkrétního změnového souboru vznikají jednotlivé implementace všech programů z rodiny  $\text{T}_{\text{E}}\text{X}$ u. Sám program  $\text{T}_{\text{E}}\text{X}$  navíc existuje ve verzích  $\text{I}_{\text{N}}\text{T}_{\text{E}}\text{X}$  a  $\text{V}_{\text{I}}\text{R}_{\text{T}}\text{E}}\text{X}$ . Běžný uživatel pracuje s  $\text{V}_{\text{I}}\text{R}_{\text{T}}\text{E}}\text{X}$ em, pomocí  $\text{I}_{\text{N}}\text{T}_{\text{E}}\text{X}$ u se zpracovávají definiční soubory balíků maker `plain`, `lplain`, `slitex` atd., které se na závěr činnosti  $\text{I}_{\text{N}}\text{T}_{\text{E}}\text{X}$ u uloží na disk jako soubor typu `.fmt`. Následné opakované čtení tohoto souboru  $\text{V}_{\text{I}}\text{R}_{\text{T}}\text{E}}\text{X}$ em je mnohokrát rychlejší, než kdyby se při běžné práci měla všechna makra načítat ze zdrojových souborů. Drobnou úpravou změnového souboru `.ch` lze zvolit, zda tangle má generovat pascalský text pro  $\text{I}_{\text{N}}\text{T}_{\text{E}}\text{X}$  nebo pro  $\text{V}_{\text{I}}\text{R}_{\text{T}}\text{E}}\text{X}$ . V prvním případě se řetězce `init` a `tini` nahradí řetězci `begin` a `end`, ve druhém případě se nahradí komentářovými závorkami. Uvedené platí i pro program  $\text{M}_{\text{E}}\text{T}_{\text{A}}\text{F}_{\text{O}}\text{N}_{\text{T}}$ . V některých

implementacích jsou INIT $\TeX$  a VIRT $\TeX$  sloučeny do jediného programu, režim činnosti se volí až při jejich aktivaci nějakým specifickým přepínačem z příkazové řádky.

Uvedený standardní postup implementace se běžně používá u velkých počítačů. Bohužel u počítačů PC je k dispozici jen relativně malá operační paměť pro překlad rozsáhlého pascalského programu nebo výsledný produkt by byl natolik rozsáhlý, že by při svém běhu obsadil většinu operační paměti (pokud by se do ní ovšem vešel). Proto se implementace pro PC a jiné „malé“ počítače provádí různými náhradními způsoby a zejména u PC se lze setkat s několika alternativními řešeními (em $\TeX$ , PC $\TeX$ , SB $\TeX$ , DOST $\TeX$ ).

### Testy trip a trap

Každá implementace programu  $\TeX$  musí úspěšně projít testem zvaným `trip`. Teprve pak smí program nést jméno  $\TeX$ . Jedině tak lze zaručit vzájemnou kompatibilitu všech existujících implementací. Otestovat (verifikovat) program je obecně i teoreticky náročná úloha. Tím spíše pro rozsáhlý program, jakým  $\TeX$  bezesporu je. V tomto případě byl zvolen nekonvenční postup, který v roce 1960 použil DEK pro testy překladače jazyka Algol. Test vychází ze zásady, že nemá valný smysl soustředit se na ty pasáže programu, které jsou mnohokrát aktivovány při běžném chodu programu a případné chyby se tak vlastně najdou „samy“. O to větší úsilí by mělo být věnováno ostatním místům programu, ale sestavit pro ně kvalitní test je i tak dost náročné.

Ukázalo se, že lepší je nechat pracovat program v režimu zcela odlišném od běžného používání, ba až téměř nesmyslném. V takovém případě je větší pravděpodobnost, že se objeví i chyby, na které by se jinak nepřišlo. Po proběhnutí testu stačí porovnat vygenerovaný soubor se vzorovým souborem dodávaným spolu s testem. Pokud se číselné výsledky drobně liší v nejméně významném řádu, lze to ještě připustit. Pokud jsou rozdíly větší, měl by se autor implementace poněkud zamyslet. Podobný typ testu je určen i pro METAFONT, jmenuje se `trap`.

### Překladač weave

Druhým překladačem jazyka WEB je `weave`. I on vedle hlavního souboru `.web` zpracovává změnový soubor `.ch`. Výsledkem jeho činnosti je soubor `.tex` s dokumentací programu. V dokumentaci je pro snadnou čitelnost dodrženo dělení na sekce. Uvedena je jak dokumentační část sekce, tak i program, který ji realizuje. Průběžně jsou uváděny křížové reference

proměnných, maker, procedur i sekcí. Na závěr je připojen shrnující index a seznam sekcí, obojí s odkazy na místo definice i použití. Z vlastní zkušenosti mohu říci, že orientace v takto dokumentovaném i značně rozsáhlém programu pak čtenáři nečiní potíže. Kniha [1] je vlastně produktem tohoto postupu, přičemž vstupním souborem byl `tex.web`.

## Další programy

Mezi programy, které mají něco společného s  $\text{T}_\text{E}\text{X}$ em, patří a jsou stejným způsobem přenositelné i další programy. Stručnou charakteristiku většiny z nich uvádím: programy:

- set        Program  $\text{X}_\text{E}\text{T}$  je obdoba programu  $\text{T}_\text{E}\text{X}$  s tím jediným rozdílem, že text je sázen zprava doleva.
- patgen    Na základě předloženého velkého množství správně rozdělených slov program vygeneruje tabulku pravidel pro dělení slov.
- bibtex    Tento program je součástí  $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ u, slouží k práci s rozsáhlou bibliografickou databází.
- mft       Program načítá soubor `.mf` se zdrojovým textem pro METAFONT a vytváří z něj text v  $\text{T}_\text{E}\text{X}$ u použitelný jako dokumentace dotyčného fontu. Svým určením je tedy podobný programu `weave`.
- gftodvi   Používá se ve spojitosti s METAFONTEM pro zkušební tisky nově připravovaných fontů.
- gftopx    METAFONT generuje fonty ve tvaru generic font `.gf`. Ty se pro běžnou práci prakticky nepoužívají. Pixel fonty `.pxl` se častěji používaly dříve, dnes se používají `.pk` fonty.
- pktofg    Tyto programy doplňují sadu konverzních programů pro práci s fonty.
- pktofx    Packed fonts `.pk` jsou dnes převažujícím formátem fontů.
- gftype    Zobrazení obsahu `.gf` souboru v lidsky čitelné podobě.
- pktype    Program pro výpis obsahu `.pk` souboru s fonty.
- pooltype  Zobrazení všech řetězců z poolu (soubor `.poo`) a kontrola celistvosti.
- dvitype   Program kontroluje celistvost a formální správnost `.dvi` souboru. Rovněž se používá k získání lidsky čitelného výpisu obsahu souboru. Původně se zřejmě předpokládalo, že bude použit k tvorbě DVI-driverů, ale to se neujalo. Dnes se drivery píše v jazyce C a na tomto základě se podporuje jejich přenositelnost. Asi nejznámější je Beebeho stavebnice driverů.

tftopl	Převod metrik fontů (soubory <code>.tfm</code> ) do textového souboru.
pltotf	Opačný program k tftopl. Oba se používají při přenosu metrik do jiného prostředí, případně jako pokus odstranit formální chyby v metrice. $\text{T}_{\text{E}}\text{X}$ perti mohou vyrábět různé efekty.
vftovp	Konverze virtuálního fontu <code>.vf</code> a metriky <code>.tfm</code> na soubor <code>.vpl</code> — virtual property list.
vptovf	Protějšek k vftovp.

Řada dalších programů je součástí konkrétní implementace a její použití bývá omezeno jen pro ni. Dokladem jsou programy `pkedit`, `fontlib` a `maketcp` z `em $\text{T}_{\text{E}}\text{X}$` .

### Jak s UNIXem?

Na počítačích s operačním systémem UNIXového typu je postup při implementaci  $\text{T}_{\text{E}}\text{X}$ u poněkud komplikován tím, že pascalské kompilátory nejsou běžně dostupné, jsou drahé a ne dostatečně kvalitní. Proto se zde šlo cestou ve světě UNIXu obvyklou, totiž přes jazyk C a standardní podporu pro syntaktickou analýzu `yacc` a `lex`. Program převedený z `WEBu` do obecného Pascalu se dále programem `convert` převede do C, pomocí `fixwrite` se samostatně přeloží pascalské příkazy vstupu a výstupu. Může následovat program `regfix` pro vyhledání nejčastěji používaných proměnných a jejich deklarování jako registrových proměnných. Převod z Pascalu do C končí programem `splitup`, který má za úkol rozdělit rozsáhlý soubor s programem v jazyce C do několika samostatně kompilovatelných souborů. To z toho důvodu, že ne každý kompilátor jazyka C je schopen přeložit opravdu rozsáhlý soubor.

Pro celý proces počínaje kompilací programu `tangle` přes vytvoření  $\text{T}_{\text{E}}\text{X}$ u, `METAFONTu` a služebních programů až po instalaci do požadovaných adresářů a inicializaci je popsán několika vnořenými soubory `Makefile`. Proto má instalátor  $\text{T}_{\text{E}}\text{X}$ u v UNIXovém prostředí usnadněnu práci. Uvážlivou modifikací souboru `Makefile` a `site.h` lze definovat druh a verzi operačního systému, typ kompilátoru C jazyka, potřebné adresáře a řadu dalších detailů. Uvažují se počítače od PC přes pracovní stanice nejdůležitějších výrobců až po sálové počítače. Stejně tak se předpokládají jak běžné kompilátory `cc`, tak i `gnu c i s` s jejich známými chybami. Po několika iteracích by neměl být problém  $\text{T}_{\text{E}}\text{X}$  instalovat. Podobný postup se užívá při generování driverů. V systému X-windows se obvykle používá `preview xdvi`, k tisku nebo případným dalším úpra-

vám lze s výhodou použít postscriptový driver `dvips`. Ale to je námět pro samostatný článek.

Lze se setkat s několika překvapeními, která vyplývají z možností a zvyklostí UNIXu, ale pro člověka obeznámeného s UNIXem by zas tak velkým překvapením být neměla.  $\TeX$ em v prostředí UNIX se zabývá Pierre MacKay z washingtonské univerzity.  $\TeX$  a DVI-drivery jsou k dispozici na počítači `byron.u.washington.edu`, jejich kopie jsou i na `cs.felk.cvut.cs`, v komprimovaném tvaru mají zhruba 15 MB.

### Závěr

Na svém pracovišti spravuji  $\TeX$  na počítačích PC (em $\TeX$ ). Pro pracovní stanice SUN 3 i SUN Sparc v prostředí X-windows a pro počítače VAX pod operačním systémem VMS jsem byl nucen  $\TeX$  vytvořit postupem zmíněným v článku, přičemž změnové soubory byly k dispozici. Pokud má člověk trochu zkušenosti s dotyčným typem počítače, není nutno se kompletní instalace  $\TeX$ u ze zdrojových souborů nijak obávat.

- [1] Donald E. Knuth.  $\TeX$ : The program. Addison Wesley 1986.
- [2] Helmut Kopka.  $\LaTeX$ , Erweiterungsmöglichkeiten mit einer Einführung in METAFONT, Addison Wesley (Deutschland) 1991

*Martin Bílý*  
`bily@cs.felk.cvut.cs`

---

## $\TeX$ a UNIX

Zbyněk Linhart

### Úvod

Cílem tohoto článku je podat úvodní přehledné vysvětlení instalace a užívání  $\TeX$ u pod operačním systémem UNIX s grafickým rozhraním X windows. Přesněji řečeno, následující popis se bude týkat instalace pod operačním systémem Ultrix v. 4.2, ale nejsou využity žádné specifické vlastnosti Ultrixu. Tento operační systém používá u svých pracovních