

Zpravodaj

Československého
sdružení uživatelů T_EXu

TEX 3
92
bulletin

OBSAH

Jiří Veselý: Téměř poslední	97
Oldřich Ulrych: Zkušenosti s METAFONTem (pokračování)	99
Janka Chlebíková: Čo s matematickými diagramami a grafmi? . . .	111
Martin Bílý: Jak implementovat T _E X	122
Zbyněk Linhart: T _E X a UNIX	128
Petr Škoda: . . . editory	137
Petr Olšák: Program MNU: Konfigurovatelné menu pro spouštění aplikací pod DOSem	141
Nová konfigurace \LaTeX u	148
Oceano T _E X	150
Obsah prvních ročníků TUGboatu	153
!!	

Vlastníkům publikace

Lichá M., Ulrych O.: \LaTeX verze 2.1

Vážení majitelé.

Kromě drobných kosmetických nedostatků, které naši publikaci „zdobí“, došlo při jejím závěrečném zpracování k dvěma podstatným chybám. V obsahu jsou čísla stránek u některých kapitol a podkapitol posunuta o 1 směrem dolů (tj. některé kapitoly a odstavce je potřeba hledat na následující stránce oproti číslu uvedenému v obsahu). V rejstříku jsou všechny odkazy na stránky posunuty o 4 směrem nahoru (tj. od všech odkazů na čísla stránek v rejstříku je nutno odečíst číslo 4).

Všem čtenářům se za zmíněné chyby omlouváme. **Naši** chybou bylo, že jsme neověřili, zda při závěrečném překladu (na jiném počítači) používáme poslední verzi makra pro tuto publikaci. Pokud tato trpká zkušenost poslouží jako varování i jiným uživatelům T_EXu, bude to pro nás alespoň částečnou útechou.

Michaela Lichá, Oldřich Ulrych

Úvodní text píšu s pocitem dluhu: cítím, že nelze vyjmenovat všechny, kdo se podíleli na organizaci Euro \TeX u a vyjádřit tak zasloužené díky za obětavou, mnohdy poněkud nudnou, ale nutnou práci, kterou si příprava i zajištění zdárného průběhu vyžádaly. Alespoň tedy anonymně děkuji kolegům i přátelům z $\mathcal{C}\text{STUGu}$ i mimo něj, studentům i mnoha dalším, kteří nám obětavě pomohli. Děkuji i jménem mnoha těch, od nichž jsem dostal v posledních dnech e-maily vyjadřující díky i chválu: Euro \TeX '92 se líbil a jistě posloužil k propagaci dobrého jména $\mathcal{C}\text{STUGu}$ ve světě.

A nyní — poněkud telegraficky — pár informací. Někteří „domácí“ přihlášení na Euro \TeX nedorazili (někdy z vážných důvodů: jistě mluvím za všechny, přeji-li Štefanu Porubskému brzké uzdravení), jiní přijeli „v zastoupení“ nebo v poněkud jinou dobu. Se všemi se pokusím vyřídit finanční záležitosti do konce října. V současné době je dost neméně důležité práce: proplácíme docházející faktury, kupujeme materiál, který jsme si pro hladký průběh Euro \TeX u musili vypůjčit apod. Za upozornění v této souvislosti stojí i to, že se mění číslo našeho běžného účtu. A ještě jedna poznámka: dobrou představu o Euro \TeX u vám dá sborník příspěvků. Je to hezká knížka o 330 stránkách, kterou budeme opět prodávat členům za režijní cenu, ne však dříve než na Valném shromáždění $\mathcal{C}\text{STUGu}$ v prosinci. Pouze výtisky pro kolektivní členy budou k dispozici u Hanky Najmanové od 1.11. t.r.

Během Euro \TeX u proběhlo u příležitosti jakéhosi „Dne otevřených dveří“ ve středu 16. září i krátké Valné shromáždění členů $\mathcal{C}\text{STUGu}$. Na něm jsme rozhodli o zakoupení editoru $\mathcal{C}\mathcal{S}\mathcal{E}\mathcal{D}$ a pohovořili o nejbližší perspektivě sdružení. Pro pohodlnější práci s $\mathcal{T}\mathcal{E}\mathcal{X}$ em budou kromě editoru $\mathcal{C}\mathcal{S}\mathcal{E}\mathcal{D}$ k dispozici také upravený $\mathcal{T}\mathcal{E}\mathcal{X}$ shell, obsahující skromnější editor, a menu Petra Olšáka. To bylo k dispozici již v průběhu Euro \TeX u a bude součástí inovované „standardní české/slovenské instalace“, na které se již pracuje. S autorem $\mathcal{C}\mathcal{S}\mathcal{E}\mathcal{D}$ u Pavlem Ševečkem budeme jednat o začlenění korektoru takovým způsobem, aby ho bylo možno používat i „uvnitř“, ale v případě potřeby i „vně“ tohoto editoru. Poznávám, že korektor je nyní upraven ke korekci českého, slovenského i anglického textu. Znovu opakuji výzvu: dejte nám k dispozici vaše pomocné slovníky se

slovy, která korektor neznal, stále ještě máme dle smlouvy rok na provedení korekcí.

Uvítáme také nové příspěvky pro $\text{T}_{\text{E}}\text{X}$ bulletin. Na nedostatek si nestěžujeme, ale chceme si v budoucnu i vybírat a vám chceme přinášet stále více užitečných a zajímavých informací. Jak vidíte z ukázky v tomto čísle, nevyhýbáme se ani poezii související s $\text{T}_{\text{E}}\text{X}$ em. (Zde se sluší poděkovat dvěma lidem, kteří se kromě autora o příspěvek zasloužili: paní Anně Fučíkové, kterou znají téměř všichni, jimž vyšel někdy článek ve CMUCu, a kterou privilegovaní mohli oslovovat „paní Fu“ — ta připravila překlad; na něm pak ještě s ohledem na terminologické úpravy nemálo zapracoval kol. Arnold Santus z ČVUT.) Rozsah dnešního čísla by se měl dle možnosti stát pravidlem, ne výjimkou.

V souvislosti s dalším vývojem $\mathcal{C}\text{STUGu}$ je dobré přemýšlet o některých dalších aktivitách. Máme sice např. pokryt vhodnými texty jakýsi základ $\text{T}_{\text{E}}\text{X}$ u, rádi bychom však přešli ve vydavatelské politice na vyšší úroveň při zachování maxima ze současných výhod, mezi něž počítáme relativní rychlost a finanční výhodnost. Rádi bychom i dosáhli finančně výhodného dovozu publikací, které si nebudeme moci dovolit z různých důvodů přeložit. Ukázky produkce nakladatelství Addison-Wesley, které byly vystaveny na Euro $\text{T}_{\text{E}}\text{X}$ u, si budete moci prohlédnout nebo i zapůjčit v Praze, v Brně a v Bratislavě. Seznam knížek a jejich uložení uveřejníme v příštím čísle. Chceme přinášet i recenze komerčních $\text{T}_{\text{E}}\text{X}$ ových produktů (první krok v tomto směru se snad již podařilo uskutečnit — celnice nám avizovala před několika dny slibně popsanou zásilku). Je ovšem lehké mít takové nápady a přání, podstatné je však umět ty dobré realizovat. Proto vás prosím, napište nám nejen o tom, co bychom se měli pokusit v příštích letech realizovat, ale i **jak si představujete realizaci** námětů a jakým způsobem k ní chcete **osobně** přispět. S ohledem na připravované volby nás upozorněte na kolegy, které rádi uvidíte v budoucím výboru: musíme na příští valné shromáždění připravit kandidátku.

Náměty na činnost i na kandidáty nového výboru pošlete laskavě na obvyklou adresu: A. Najmanová, Sokolovská 83, 186 00 Praha 8 s poznámkou „Valné shromáždění“ na obálce.

-jv-

Zkušenosti s METAFONTem (pokračování)

Oldřich Ulrych

Úvod

V minulém čísle (viz [1]) byla uveřejněna první část tohoto dvoudílného cyklu, ve které jsme se snažili objasnit postup a prostředky METAFONTu, s jejichž pomocí vytváříme v MÚ jednoduché obrázky pro časopis CMUC. Protože samotné vytvoření obrázku ještě neznamená jeho vytištění, je potřeba říci, jak jsme si zjednodušili život při vlastní sazbě obrázků v plainTEXu a $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u. To je smyslem tohoto příspěvku. Protože však život je velmi dynamický, podlehl i makro `incpic.mf` vývojovým změnám, nicméně v platnosti zůstává téměř vše, co již bylo řečeno v [1]. Z tohoto důvodu bude tento příspěvek začínat nejdříve tím, co je v makru `incpic.mf` nového, pak popíšeme řídicí slova z makra `incpic.tex`, která na závěr uvedeme v tabulce. V závěru článku také budou uvedeny příklady použití prakticky všech zde uváděných řídicích slov.

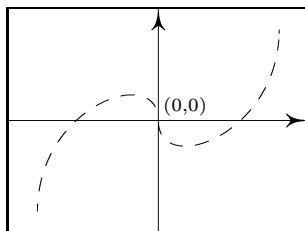
Rozšíření makra `incpic.mf`

Makro `incpic.mf` bylo na přání některých mých kolegů doplněno o dvě nové definice a o několik ekvivalentních názvů již existujících definic pro definice již zavedené, neboť prapůvodní verze tohoto makra obsahovala české názvy, které byly nahrazeny anglickými. Vážným čtenářům odstavců o příkazech METAFONTu doporučujeme přečíst si nejdříve článek [1]. Nejprve tedy nové definice:

- `dashedpath(<cesta>)`; kreslí cestu `<cesta>` přerušovanou čarou (viz obr. 1). Protože toto makro provádí dost náročné numerické výpočty (ve snaze zachovat čárky ekvidistantní), jeho používání značně prodlužuje dobu zpracování.
- `shiftorigin(<bod>)`; znamená, že referenční bod má souřadnice `<bod>` (měřeno v uživatelově souřadném systému čtverečkové sítě). Toto makro umožňuje libovolné umístění počátku uživatelova pomocného souřadného systému a zjednodušuje zápis, který byl použit v po-

slední ukázce v [1]. Jestliže počátek souřadnic splývá s referenčním bodem, není potřeba toto řídicí slovo uvádět. Postup naznačený v [1] s použitím proměnné `currenttransform` nezaručuje, že případné popisky sázené \TeX em a vztahované k pomocnému souřadnému systému budou správně umístěny. Pokud je potřeba tuto definici použít, pak je nejlépe ji umístit hned na začátek znaku uvozeného pomocí `beginch`. Definice `shiftright` se zadanou nenulovou x -ovou souřadnicí bodu $\langle bod \rangle$ je možné použít pouze u znaků A–I. V opačném případě může METAFONT ohlásit syntaktickou chybu a \TeX informaci o posunutí ignoruje. Jinak lze generovat obrázky pouze pod znaky A–O.

Obě tyto definice byly použity při vytváření obrázku, který je vidět na obr. 1. Referenční bod tohoto obrázku byl zvolen v levém dolním rohu rámečku a uživatelský souřadný systém má počátek v bodě $(0,0)$. Je zřejmé, že tato volba není jediná možná, ale byla zvolena z důvodů snadného (symetrického) zadávání bodů pro kreslení. Zdrojový text tohoto obrázku pro METAFONT je:



Obr. 1 Ukázka čárkované čáry.

```
beginch("A",40mm#,30mm#,0,10,10);
shiftright(-5,-5);
thinpen; axes(0,0);
middlepen;
dashedpath(-4,-4){up}..{down}(0,0)..{up}(4,4);
endch;
```

V následující tabulce uvádíme seznam všech definic a nastavení některých parametrů z makra `incpic.mf`. Protože až do provedení těchto úprav nebyla žádná verze tohoto makra veřejně šířena, nevznikají žádné problémy s otázkou, jestli verze, kterou si opatříte, v sobě obsahuje výše popsané rozšíření. Obsahuje. V první části tabulky je seznam definic, na kterých se nic neměnilo nebo které jsou nové; v druhé části jsou definice, ke kterým byly zavedeny ještě další, ekvivalentní názvy; poslední část ukazuje přehled implicitně nastavených hodnot v makru `incpic.mf`. Protože tato tabulka by měla být logicky začleněna do článku [1], doporučuji celou stránku s touto tabulkou z tohoto čísla vytrhnout a vlepít si ji do čísla předcházejícího, kam logicky patří:

<pre> beginch(<kód>,<šířka>,<výška>,<hloubka>,<šířka čtv.>,<výška čtv.>); endch; sharparrow axes(<bod>); arrow(<délka>,<úhel>); vector(<počátek>,<konec>); cerchovane(<počátek úsečky>,<konec úsečky>); dashedpath(<cesta>); shiftright(<bod>); </pre>	
<pre> penfordots; penfordots; thinpen; middlepen; thickpen; drawdots(<b₁₂₁₂₁₂</pre>	<pre> peronatecky; peronapuntiky; tenkepero; strednipero; tlustepero; puntiky(<b₁₂₁₂₁₂</pre>
<pre> showmesh:=0; eps_:=0.001; s_r:=10; </pre>	<pre> showbox:=0; len_given:=2; thindestline:=0.4pt; </pre>

Tabulka všech definic a nastavení parametrů z makra `incpic.mf`

Charakteristika „objektů“ začleňovaných do textu

Objekty, které nejsou textového charakteru a které zařazujeme do textu, je z našeho pohledu možné rozdělit do čtyř skupin:

1. Vynechání prázdného místa předepsaných rozměrů, např. pro pozdější nalepení fotografie, kterou nechceme elektronicky zpracovávat.

2. Prázdné místo předepsaných rozměrů s uvedením povelů pro začlenění obrázků při zpracování dvi souborů (např. pomocí řídicího slova `\special{...}`).
3. Objekt vytvořený v \TeX u, např. tabulka.
4. Obrázek vytvořený METAFONTem pomocí makra `incpic.mf`.

Protože skupina 2 zahrnuje skupinu 1, nebudeme tyto dvě skupiny nadále rozlišovat. Pro zjednodušení vyjadřování budeme mít pod slovem „obrázek“ na mysli kterýkoliv z výše uvedených objektů.

Nároky na začlenění obrázků do textu

Protože je dost často nutno dělat rozhodnutí o změně umístění obrázku či vynechaného místa, kladli jsme na makro pro začlenění obrázku do dokumentu požadavky, které vyplynuly z našich potřeb:

1. Snadná manipulovatelnost s obrázky.
2. Možnost umístit různé značky do obrázku (popis do obrázku) — např. `text (0,0)` v obr. 1.
3. Možnost umístit popis pod obrázek.
4. Jestliže je příkaz pro začlenění v odstavci, obrázek má být umístěn za řádek, ve kterém se tento příkaz vyskytl (pomocí řídicího slova `\adjust{...}` — implicitní nastavení je obrázek umístěný centrováně).
5. Jestliže je příkaz pro začlenění obrázku mezi odstavci s vyznačením, že má být začleněn do následujícího odstavce nalevo či napravo (případně s vyznačením počtu řádků sázených nad obrázkem na plnou šířku zrcadla), bude obtékán textem. Jinak bude vysázen v místě, kde je uvedeno, centrováně. V těchto případech by mělo být umožněno používat v popise pod obrázkem poznámky pod čarou. Pokud v takovém místě (v případě obtékání textem) obrázek umístit nelze, měl by být přesunut na následující stránku a zároveň by se mělo objevit varování v `.log` souboru.
6. V ostatních případech umístit obrázek v místě, kde je uvedeno.
7. Možnost záramovat libovolný obrázek.
8. Referenční bod obrázku by měl být (implicitně) na levém spodním okraji obrázku (případný popis pod obrázkem by tedy byl pod tímto referenčním bodem), nebo nastavitelný na levý horní roh obrázku nebo

levý dolní roh (případného) popisu obrázku. Všechny tři případy jsou na obrázku 5 vyznačeny puntíky.

9. Možnost jedním řídicím slovem zarámovat všechny obrázky.
10. Možnost jedním řídicím slovem znázornit zarámováním jen prázdné místo, které budou obrázky zaujímat, ale vlastní obrázky tam neumísťovat. Tato možnost zrychluje prohlížení dokumentu, pokud potřebujeme upravovat pouze text a kontrolovat provedené změny.

Řídicí slova uvádějící obrázky

Celé makro `incpic.tex` bylo navrženo ze zorného úhlu výše uvedených dvou výtčů. Začlenění obrázku do textu probíhá ve dvou etapách. V první etapě se obrázek spolu se značkami uloží do zásobníku obrázků a jeho případný popis, který má být pod ním, do zásobníku popisů. Ve druhém kroku se vyjme obrázek a odpovídající popis ze zásobníků a umístí se spolu do požadovaného místa. Oba zásobníky jsou typu FIFO (First In First Out), umožňují nejdříve si všechny obrázky do zásobníku „naskládat“ a pak je postupně ze zásobníků vyjmát a vkládat do textu. Všechny obrázky mohou být umístěny např. ve zvláštním souboru a v pořadí, v jakém se mají vkládat do textu. Tento soubor stačí načíst pomocí řídicího slova `\input` a pak v příslušných místech pouze uvádět příkazy pro vložení obrázku. Druhým extrémem je, že obrázek do zásobníku vložíme a hned jej vyjmeme a vložíme do textu.

Pokud tento postup není jasný, lze jej přiblížit na příkladě ze života, kdy obrázek je totéž, co kornout na zmrzlinu, \TeX je zmrzlinářem a zásobník obrázků je zásobník na kornouty. Zmrzlinář by v prvním případě nasypal kornouty na zmrzlinu do válcového zásobníku shora, a to nejvýše tolik, jakou má zásobník kapacitu (u \TeX u je toto omezení dáno pamětí), a pokud jich tam chce nasypat více, musí nejdříve odebráním zespod zásobník alespoň částečně vyprázdnit. Druhým případem je, že vhodí shora jeden kornout do zásobníku kornoutů a hned jej zdola vytáhne a natočí do něj zmrzlinu. Nikdy nenarazí na problém, že se mu kornout do zásobníku nevejde, ale o trošku víc se nadře.

Řídicí slova pro ukládání obrázků do zásobníku jsou tři a odrážejí tři skupiny objektů začleňovaných do textu. Protože poslední dva argumenty těchto řídicích slov jsou stejné, uvádíme níže jejich popis pro všechna tři řídicí slova společně.

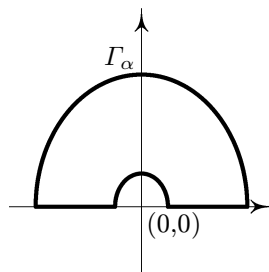
- `\emptyplace{⟨šířka⟩⟨cokoliv⟩}{⟨výška⟩}{⟨značky⟩}⟨popis⟩\par` vytvoří místo o velikosti $\langle\text{šířka}\rangle \times \langle\text{výška}\rangle$, přičemž do tohoto místa je vloženo $\langle\text{cokoliv}\rangle$ (zde lze např. uvést příkaz `\special{...}` nebo také nic). O argumentech $\langle\text{značky}\rangle$ a $\langle\text{popis}\rangle$ je pojednáno níže.
- `\boxplace{⟨vbox⟩}{⟨značky⟩}⟨popis⟩\par` znamená, že obrázek je tvořen prvním parametrem $\langle\text{vbox}\rangle$. Tento argument bude uzavřen do vertikálního boxu. Tím chceme naznačit, že tento argument se začne zpracovávat ve vnitřním vertikálním režimu. O argumentech $\langle\text{značky}\rangle$ a $\langle\text{popis}\rangle$ je pojednáno níže.
- `\fontcharplace{⟨font⟩}{⟨kód⟩}{⟨značky⟩}⟨popis⟩\par` znamená, že obrázek bude tvořen znakem, jehož kód je $\langle\text{kód}\rangle$ z fontu $\langle\text{font}\rangle$. Poznamenejme, že $\langle\text{font}\rangle$ musí být řídicí slovo, kterým je označen font s obrázky, a $\langle\text{kód}\rangle$ je kód znaku, pod kterým je obrázek uložen. Tedy např. pro znak A je možné zadat jeho kód ve formě `\A` nebo 65. O argumentech $\langle\text{značky}\rangle$ a $\langle\text{popis}\rangle$ je pojednáno níže.

Význam argumentů $\langle\text{značky}\rangle$ a $\langle\text{popis}\rangle$ je tento:

- $\langle\text{značky}\rangle$ jsou značky, které mají být umístěny do obrázku. V těchto argumentech by měla být použita řídicí slova `\placechar` nebo `\fontchar` (toto však pouze pro umístování značek u obrázku z fontu, který byl vytvářen s pomocí METAFONTového makra `incpic.mf`). Tato řídicí slova jsou popsána v následujícím odstavci.
- $\langle\text{popis}\rangle$ je text, který má být napsán pod obrázkem na šířku, která je určena šířkou obrázku. Při formátování popisu pod obrázek jsou nastavována řídicí slova `\leftskip` a `\rightskip` na určité hodnoty (tato informace je pro ty uživatele, kteří v popise pod obrázkem tato řídicí slova chtějí používat).

Značky v obrázku

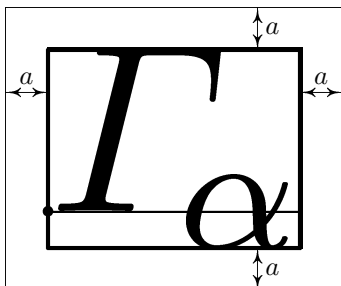
Uvažujme následující velice jednoduchý obrázek s popisem (viz obr. 2). Předpokládáme, že popis chceme do obrázku doplnit \TeX em. Tedy do obrázku potřebujeme doplnit do přešpaných míst značky $(0, 0)$ a Γ_α . K tomu je potřeba přesně určit, že jeden bod značky bude splývat s jedním bodem obrázku. Zde v tomto obrázku jsme se rozhodli, že umístíme značku $(0,0)$ vpravo dole od průsečíku os a značku Γ_α vlevo nahoru od průsečíku křivky s osou. Abychom



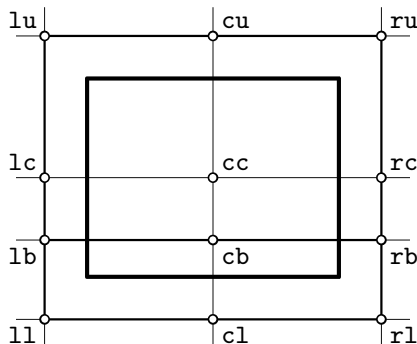
Obr. 2 Cesta integrace.

mohli tyto dva požadavky přesně sdělit T_EX u, podívejme se blíže na značku Γ_α .

Tato značka, kterou jsme zvětšenou zobrazili na obrázku 3, tvoří sama o sobě krabičku. Tato krabička je na obrázku 3 vyznačena tlustou čarou s referenčním bodem v bodě naznačeném puntíkem a má určitou šířku, výšku a hloubku (měřeno od referenčního bodu). Řekněme, že takový box



Obr. 3 Značka s okrajem a .



Obr. 4 Body na značce

chceme zvětšit ve všech směrech o nějakou malou hodnotu a (implicitně nastaveno na 2pt), neboť se chceme vyhnout případu, že se značky dotýkají os apod.

Po takovém zvětšení má značka velikost, kterou jsme znázornili na obrázku 3 tenkou čarou. Tato nově vzniklá krabička je charakterizovatelná (např.) jedním z dvanácti bodů, které jsou popsány dvojicemi písmen tak, jak je znázorněno na obrázku 4. První písmeno může být **l**, **c**, **r** a znamená, že jde o bod na levém okraji (**left**), na svislé ose (**centered**) nebo na pravém okraji (**right**) značky. Druhé písmeno může být **u**, **c**, **b**, **l** a znamená, že jde o bod na horním okraji (**upper**) uprostřed celkové výšky (**centered**), na základně (**baseline**), dolním okraji značky (**lower**). Tyto body a příslušná písmena jsou vyznačeny na obrázku 4.

Pro umístění značek do obrázků jsou navržena tato řídicí slova:

- `\placemark <ph><pv>_<xdim>_<ydim>_<značka>_` je řídicí slovo, které umístí značku $\langle značka \rangle$ (což může být libovolný text — například $\$\varGamma_\alpha\$$) do předepsaného místa v obrázku. Bod, do kterého patří umístit značku, je zadán hodnotami $\langle xdim \rangle$ a $\langle ydim \rangle$ (např. `22mm_15pt_`). Tyto hodnoty jsou vztaheny k referenčnímu bodu obrázku (což je v případě řídicího slova `\emptyplace` levý dolní roh vynechaného

místa, v případě řídicího slova `\boxplace` referenční bod výsledného boxu a v případě řídicího slova `\fontcharplace` referenční bod znaku, ve kterém je vytvořen obrázek METAFONTem. Bod, kterým patří značku (po rozšíření o okraj — viz popis výš) umístit do předepsaného bodu obrázku, je určen dvěma písmeny $\langle ph \rangle \langle pv \rangle$. Prvním písmenem $\langle ph \rangle$ může být `l`, `c` nebo `r` a druhým písmenem $\langle pv \rangle$ může být `u`, `c`, `b` nebo `l`. Význam těchto písmen je popsán výš. Povšimněte si, že kromě prvního argumentu, který je jednopísmenný, musí být všechny ostatní argumenty ukončeny (tj. vyznačeny) mezerou. Tento způsob se nám osvědčil v praxi jako nejpohodlnější.

- `\fontmark` $\langle ph \rangle \langle pv \rangle \square \langle xrel \rangle \square \langle yrel \rangle \square \langle značka \rangle \square$ je řídicí slovo, pro které platí totéž, co bylo řečeno o řídicím slově `\placemark`, ale je možné je použít pouze pro obrázky vytvořené METAFONTem pomocí makra `incpic.mf` a vkládané do textu pomocí řídicího slova `\fontcharplace`. Bezrozměrné hodnoty $\langle xrel \rangle$, $\langle yrel \rangle$ popisují v uživatelské pomocném souřadném systému (čtverečkový papír) bod, do kterého má být značka umístěna. O vyznačování argumentů mezerami platí totéž, co u řídicího slova `\placemark`.

Další užitečné definice

Výše uvedené definice ještě nepokrývají všechny požadavky, které jsme si kladli na snadnou manipulaci s obrázky. Proto makro `incpic.tex` obsahuje ještě tyto definice:

- `\allframed` je řídicí slovo, které způsobí, že všechny následující obrázky vkládané do textu budou zarámovány tenkou čarou.
- `\picturemargins{\langle dimen1 \rangle}{\langle dimen2 \rangle}` je řídicí slovo, které nastavuje pro následující obrázky začleňované do odstavce na levou či pravou stranu (které jsou tedy obtékané textem) velikost mezery mezi obrázkem a okrajem textu na hodnotu $\langle dimen1 \rangle$ a velikost mezery mezi obrázkem a okrajem zrcadla na hodnotu $\langle dimen2 \rangle$. Hodnotám $\langle dimen1 \rangle$ a $\langle dimen2 \rangle$ odpovídají hodnoty \check{s}_1 a \check{s}_2 . Implicitně jsou tyto hodnoty nastaveny na 15pt a 0pt.
- `\descriptionmargins{\langle dimen \rangle}` nastavuje velikost okraje, který je přidáván k následujícím značkám. Hodnota $\langle dimen \rangle$ udává velikost vyznačenou na obrázku 3 písmenem a . Implicitně nastavená hodnota je 2pt.

- `\picname{<návěští>}` je řídicí slovo, které vysází od levého okraje tučně text `<návěští>` a následující text bude zalomen do odstavce, který bude mít všechny řádky odražené zleva o šířku textu `<návěští>`. Toto řídicí slovo bylo navrženo ke snadnějšímu přidávání popisu pod obrázky.

Řídicí slovo začleňující obrázky do textu

• `\inspicture` je řídicí slovo, které vybere další obrázek ze zásobníku obrázků a umístí ho podle kontextu, kde se vyskytuje a podle dalších volitelných parametrů do textu. Za tímto řídicím slovem mohou být tyto parametry:

- `l` nebo `c` nebo `r` znamená, že obrázek má být umístěn nalevo na stránce, vprostřed na stránce nebo napravo na stránce (implicitně je nastaveno doprostřed). V případě, že je obrázek začleňován do odstavce vlevo či vpravo, je měněn obsah `\everypar{}`.

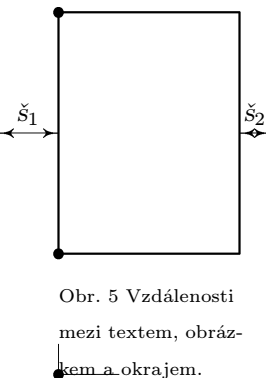
- `(<n>)` znamená, že obrázek má být zařazen do následujícího odstavce, přičemž nad obrázkem má být vysázeno `<n>` řádků na plnou šířku zrcadla. Toto je bráno v úvahu, pouze pokud jsou zařazovány obrázky do odstavce pomocí parametrů `l` nebo `r` (a pokud se řídicí slovo `\inspicture` vyskytuje mezi odstavci. Implicitně nastavená hodnota je 0 řádků.

- `!` znamená, že obrázek se má umístit bezprostředně v místě, kde je řídicí slovo `\inspicture` uvedeno (v tomto případě jsou všechny předcházející volby ignorovány).

- `^` nebo `_` znamená, že pro srovnávání s okolním textem je uvažována horní hrana obrázku nebo dolní řádek popisu (pokud popis chybí, pak dolní hrana obrázku). Viz body vyznačené puntíky v obrázku 5.

- `|` znamená, že obrázek má být zarámován tenkou čarou.

- `{<text>}` znamená, že `<text>` bude přidán k textu pod obrázkem (za text, který je zadán jako poslední argument řídicího slova `\emptyplace` nebo `\boxplace` nebo `\fontcharplace`).



Obr. 5 Vzdálenosti mezi textem, obrázkem a okrajem.

Příklady

Protože jsme již vyčerpali veškerou suchou teorii, téměř na samý závěr uvádíme několik příkladů opatření obrázků popisem a jejich začlenění do textu. Aby bylo vidět, jak jsme postupovali od návrhu obrázků v METAFONTu, uvedeme zde začátky znaků s obrázky tak, jak jsme je měli ve zdrojovém souboru pro METAFONT (toto je důležité, aby bylo vidět, jakou síť jsme měli v jednotlivých případech zvolenou).

```
beginch("C",52.125mm#,45mm#,0mm#,139,120); % ram pro značku
beginch("B",35mm#,35mm#,0mm#,10,8); % integracni cesta
beginch("A",40mm#,30mm#,0,10,10); % carkovana cesta
```

Příkazy, kterými byly obrázky 1, 2, 3 a 6 popsány a uloženy do zásobníku obrázků, byly (`bull` je název fontu, ve kterém byly obrázky vygenerovány a `bull.pcx` je soubor s obrázkem ve formátu PCX):

```
\font\bullpic=bull
\fontcharplace{\bullpic}{'\A}{
  \fontmark ll 0 0 \sevenrm(0,0)
  }\picname{Obr. 1} Ukázka čárkované čáry.

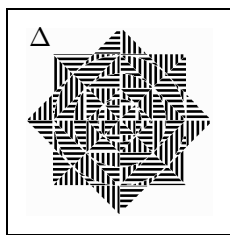
\fontcharplace{\bullpic}{'\B}{
  \fontmark lu 5 2 (0,0)
  \fontmark rl 5 6 $\varGamma_\alpha$
  }\Obr. 2 Cesta integrace.

\fontcharplace{\bullpic}{'\C}{\descriptionmargins{0pt}
  \fontmark ll 25 25 \box\znak{
  \descriptionmargins{3pt}
  \fontmark lc 99 17.5 $a$ \fontmark lc 99 102.5 $a$
  \fontmark cl 17.5 80 $a$ \fontmark cl 121.5 80 $a$
  }\hskip 15pt Obr. 3 Značka s~okrajem~$a$.

...
\emptyplace{30mm\special{em:graph bull.pcx}}{30mm}{
  \placemark ll 3mm 25mm $\Delta$
  }\hfil Obr. 6
```

K výše uvedenému zdrojovému textu uvedme ještě několik poznámek. Souřadnice pro umístování značek do obrázků byly odečteny ze zdrojového textu obrázků pro METAFONT. U obrázku 3 jsme měli v boxu `\znak`

připravena zvětšená písmena Γ_α , která jsou do obrázku sestávajícího jen z čar, šipek a teček vložena jako značka. Protože je potřeba tuto značku Γ_α přiložit levým dolním rohem přesně do předepsaného bodu, je předtím nastavena velikost okrajů značek na 0pt. Protože značka musí být ukončena mezerou, jsou za řídicím slovem `\znak` složené závorky. Okraj dalších značek je zvětšen na 3pt proto, že značky jsou umísťovány nad šipky, a aby tudíž nedocházelo k jejich překrytí. Popis pod obrázkem začíná skokem `\kern15pt`, neboť ve zdrojovém textu pro METAFONT jsou uvedeny stejné rozměry obrázků 3 a 4. V obrázku 6 je vidět, jakým způsobem jsme vložili do vynechaného místa informace o začlenění PCX souboru do daného místa. Zároveň jsme také umístili číslo obrázku cent-



Obr. 6

rovaně pod obrázek. Toto je díky tomu, že za textem „Obr. 6“ je rovněž `\hfil` implicitně vložen z `\parfillskip`.

Nyní již zbývá jen popsat, jakým způsobem jsme všechny obrázky do textu vložili tak, aby se objevily v místech, kde jsou vidět. Abychom vložili do textu obrázek 1, umístili jsme příkaz

```
\inspicture| (2)r
```

před začátek odstavce *Obě tyto definice byly ...* Svislá čára jako argument způsobila, že je obrázek zarámován.

Obrázek 2 jsme vložili do textu příkazem

```
\inspicture(2)r
```

uvedeným před odstavcem „• *⟨popis⟩* je popis ...“

Obrázky 3 a 4 jsme vložili do textu tím způsobem, že v odstavci ve zdrojovém textu bylo napsáno:

```
... Řekněme, že takový
\adjust{ \line{\inspicture \hfil \inspicture } }%
box chceme zvětšit ve všech směrech o~nějakou ...
```

Obrázek 5 jsme vložili do textu příkazy

```
\picturemargins{0pt}{0pt}
\inspicture(3)r
```

před začátek odstavce popisujícího řídicí slovo `\picturemargins`.

Obrázek 6 jsme vložili do textu příslušného odstavce tím, že jsme ve zdrojovém textu psali:

```
... PCX souboru do daného místa.\inspicture|
Zároveň jsme také umístili číslo obrázku ...
```

Tabulka příkazů makra `incpic.tex`

V následující tabulce uvádíme seznam řídicích slov z makra `incpic.tex`. V hranatých závorkách u řídicích slov `\placemark` a `\fontmark` jsou uvedena všechna možná písmena, z nichž je nutno vybrat právě jedno. U řídicího slova `\inspicture` je uveden seznam všech parametrů, které za tímto řídicím slovem **mohou** být uvedeny.

<pre>\emptyplace{⟨šířka⟩⟨cokoliv⟩}{⟨výška⟩}{⟨značky⟩}⟨popis⟩\par \boxplace{⟨vbox⟩}{⟨značky⟩}⟨popis⟩\par \fontcharplace{⟨font⟩}{⟨kód⟩}{⟨značky⟩}⟨popis⟩\par \placemark [lcr][ucbl]⟨xdim⟩⟨ydim⟩⟨značka⟩ \fontmark [lcr][ucbl]⟨xrel⟩⟨yrel⟩⟨značky⟩ \allframed \picturemargins{⟨dimen_{text-obrázek}⟩}{⟨dimen_{obrázek-okraj}⟩} \descriptionmargins{⟨dimen⟩} \picname{⟨návěští⟩} \inspicture lcr(⟨n⟩)!!^_{⟨text⟩}</pre>

Literatura:

- [1] Ulrych O., *Zkušenosti s METAFONTem*, Zpravodaj Československého sdružení uživatelů T_EXu, 2 (1992), str. 60–80.

Oldřich Ulrych

Čo s matematickými diagramami a grafmi?

Janka Chlebíková

Každý užívateľ $\text{T}_\text{E}\text{X}$ u, ktorý potrebuje do svojho článku zahrnúť i pomerne jednoduchý matematický diagram či graf, si asi povzdychne nad nie príliš kamarátskym vzťahom $\text{T}_\text{E}\text{X}$ u a grafiky. Na kreslenie jednoduchých matematických obrázkov má užívateľ v podstate nasledujúce možnosti:

Jedna možnosť je použiť METAFONT (mimochodom tí, ktorých okúzliť $\text{T}_\text{E}\text{X}$, pravdepodobne podľahnú i čaru METAFONT u), pomocou ktorého sa dá zvládnuť každý diagram či graf až s puntičkárskou presnosťou. Toto riešenie má nesmiernu výhodu v nezávislosti na výstupnom driveri (t.j. ľubovoľný $\text{T}_\text{E}\text{X}$ ový driver, obrazovkový či tlačiarenský, ktorý vie pracovať s $\text{T}_\text{E}\text{X}$ ovým dokumentom, vie pracovať i s metafontovými obrázkami). Nič sa nestratí ani na bezproblémovej prenositeľnosti dokumentu.

Druhá možnosť je použiť na kreslenie nejaký grafický editor (napr. Paintbrush, ...). Nevýhodou tohto prístupu je závislosť na výstupnom driveri. Obrázky musia byť často vytlačené zvlášť, pokiaľ nie je k dispozícii $\text{T}_\text{E}\text{X}$ ový driver spolupracujúci s daným konkrétnym grafickým formátom obrázku. V tomto prípade je tiež veľmi obtiažna „spolupráca“ $\text{T}_\text{E}\text{X}$ ového textu s obrázkami (napr. na popis obrázku).

Užívateľ, ktorý potrebuje nakresliť len jeden-dva relatívne jednoduché obrázky by asi dal (pre celý rad výhod) prednosť nejakej $\text{T}_\text{E}\text{X}$ ovej možnosti. Takáto možnosť sa mu ponúka vo využití vhodného balíka makier na kreslenie obrázkov v samotnom $\text{T}_\text{E}\text{X}$ u. Problém často býva „nájsť“ existujúce vhodné makrá, pokiaľ ich nechceme vytvárať sami.

$\text{L}\text{A}\text{T}_\text{E}\text{X}$ isti často vystačia s „picture“ okolím, v ktorom s pomocou interaktívneho $\text{T}_\text{E}\text{X}$ cadu (z balíku $\text{em}\text{T}_\text{E}\text{X}$) je kreslenie obrázkov veľmi jednoduché a pohodlné. Iná možnosť je použiť pomerne rozsiahly $\text{P}\text{T}\text{C}\text{T}_\text{E}\text{X}$ [3], pomocou ktorého sa dajú kresliť i zložitejšie obrázky. Tento balík však často nie je moc vhodný v prípade diagramov a grafov.

V súčasnosti existuje rad rôznych iných makier, vždy zameraný na kreslenie nejakej špeciálnej matematickej štruktúry (napr. hierarchickej stromovej štruktúry, pre schémy dôkazov v matematickej logike, ...). Tieto makrá sú však obvykle veľmi úzko zamerané a neposkytujú užívateľovi zväčša veľkú voľnosť použitia. Výnimkou v tomto smere je ba-

lík makier \Xy-PIC^1) (podporovaný vlastnou sadou metafontových fontov), ktorý je špeciálne vhodný na kreslenie matematických diagramov a grafov a poskytuje užívateľovi dostatočnú variabilitu. Navyše umožňuje spoluprácu so všetkými základnými nadstavbami \TeX u (\plainTeX , \AMS-TeX , \LATEX , \AMSLATEX , ...), pretože používa jazyk základného \TeX u. Tento fakt je zvlášť potešiteľný pre početnú skupinu stúpcov \AMS-TeX u, ktorá obvykle dosť ťažko zápasí i s veľmi jednoduchými obrázkami.

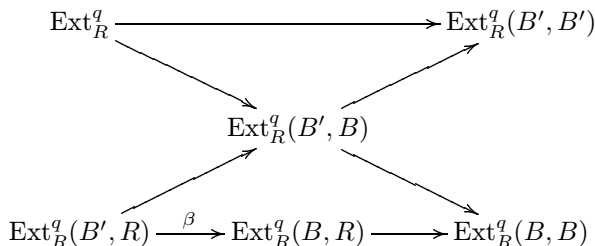
Reklama \Xy-PIC u na úvod

Pozorný čitateľ Spivakovej príručky *The Joy of \TeX* [2] si iste spomína na poznámku, že iné ako obdĺžnikové diagramy sa v \AMS-TeX u jednoducho kresliť nedajú (na strane 142 (viď obr. 1) je uvedený diagram, ktorý sa už v \AMS-TeX u nedá jednoducho nakresliť). Použitím \Xy-PIC makier uvedený diagram môžeme však v svojom \AMS-TeX ovskom článku nakresliť veľmi jednoducho.

```

 $\def\ExtqR{\text{Ext}}^q_R$ 
\diagram
\ExtqR\rrto\drto & & \ExtqR(B',B') \\
& \ExtqR(B',B) \urto\drto \\
\ExtqR(B',R) \urto\rto^{\beta} & & \\
& \ExtqR(B,R) \rto & \ExtqR(B,B) \\
\enddiagram

```



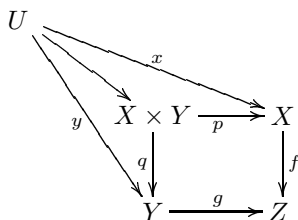
Obr. 1.

¹⁾ Autorom týchto makier je Kristoffer H. Rose z Kodane a makrá s fontami, ako i krátkym manuálom sú prístupné cez anonymous ftp na internetovskej adrese ftp.diku.dk [129.142.96.1], prípadne u autorky tohto príspevku.

Nasledujúca časť príspevku sa snaží vystihnúť všetky základné možnosti tohto balíka makier a ukázať jeho základnú koncepciu. Niektoré príklady pre ilustráciu boli vybrané zo spomínaného autorovho manuálu [1].

Základná koncepcia Xy-PICových diagramov

- Diagram je popisovaný maticovou štruktúrou, prvky matice zodpovedajú objektom v diagrame. Objekty sú automaticky zarovnávané v riadkoch a stĺpcoch. Možno tiež meniť štandardne nastavenú vzdialenosť medzi riadkami a stĺpcami. Popis diagramu



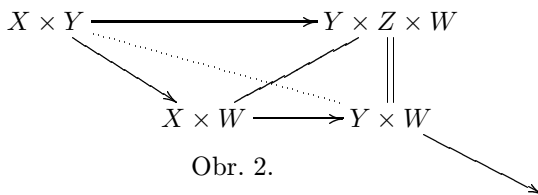
vyzerá veľmi jednoducho:

```

 $\begin{array}{ccccc}
U & & & & \\
& \searrow & & \searrow & \\
& X \times Y & \xrightarrow{p} & X & \\
& \searrow q & & \searrow f & \\
& Y & \xrightarrow{g} & Z &
\end{array}$ 

```

- Ľubovoľný objekt môže byť spojený s ľubovoľným iným objektom v diagrame použitím spojovacej čiary (ktorých je k dispozícii niekoľko druhov). Navyše je možné viesť z objektu spojovaciu čiaru kamkoľvek, poloha je udávaná relatívne alebo absolútne vzhľadom k danému ľubovoľnému objektu. Spojovacia čiara spája stredy objektov a je kreslená od „boxov“ ohraničujúcich objekty. Dĺžka kreslenej časti čiary sa mení automaticky so zmenou polohou objektov. Príkladom môže byť nasledujúci diagram



Obr. 2.

s popisom

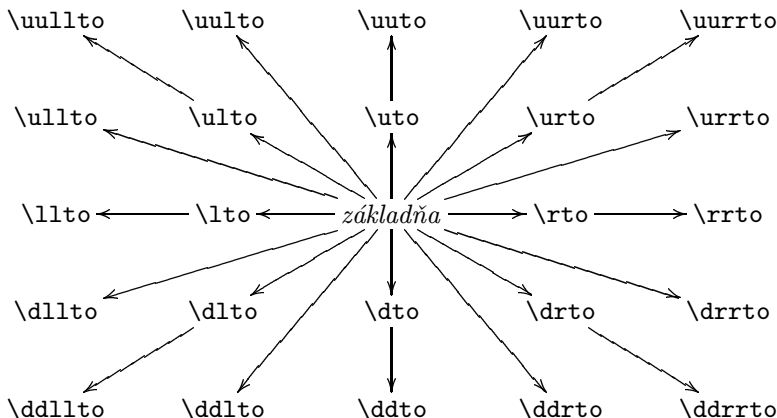
```

$$
\diagram
  X \times Y \xrightarrow{[0,2]} Y \times Z \times W
  \xrightarrow{[1,1]} X \times W \xrightarrow{[1,3]} Y \times W
  \xrightarrow{[0,0]+<2cm,-1cm>}
\enddiagram
$$

```

Spojovacie čiary

• Z počiatočného do cieľového objektu môžeme smer spojovacej čiary popísať pomocou preddefinovaných smerov a vzdialeností



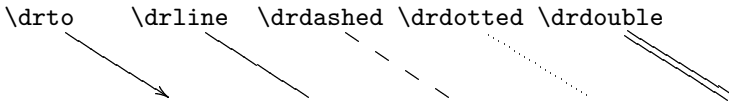
alebo (pokiaľ nevystačíme s predchádzajúcim) relatívne alebo absolútne k počiatočnému objektu. Napríklad diagram na obr. 2 môže byť tiež popísaný:

```

$$
\diagram
  X \times Y \rrto \drto \xdotted"2,3"
                                     && Y \times Z \times W \ll
  & X \times W \underline{\rto}
  & Y \times W \udouble \xto[0,0]{<2cm,-1cm}
\enddiagram
$$

```

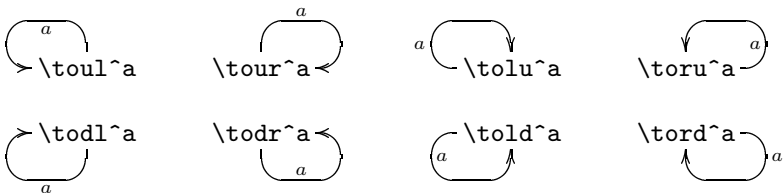
- Existuje niekoľko základných typov spojovacích čiar (je možné definovať si i ďalšie):



Okrem toho je možnosť výberu špeciálnych spojovacích čiar



a množiny slučiek



- Je možné vybrať si ľubovoľné umiestnenie popisu nad alebo pod spojovaciu čiaru, napríklad nasledujúci obrázok spolu s popisom

$$\frac{0}{c} \cdot \xrightarrow{+} \frac{1}{c}$$

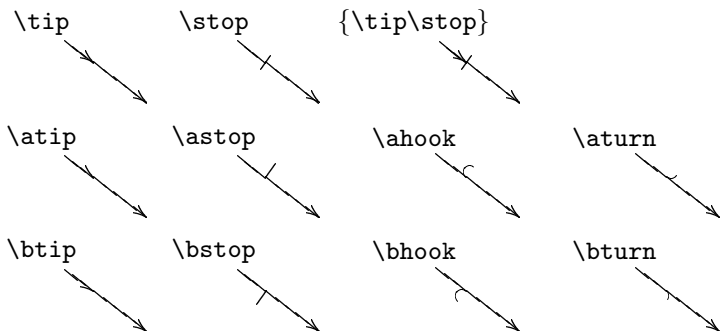
možno dosiahnuť makrami

```

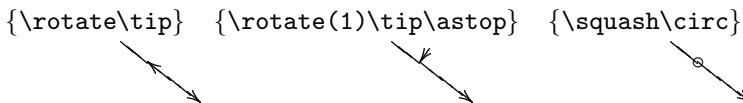
$$
\diagram
== \rrto ^{+} ^<{.} ^<<{,} _<>(.5)c ^>>{,}
^>{.} ^{(0)0 ^{(1)1 && =====
\enddiagram$$

```

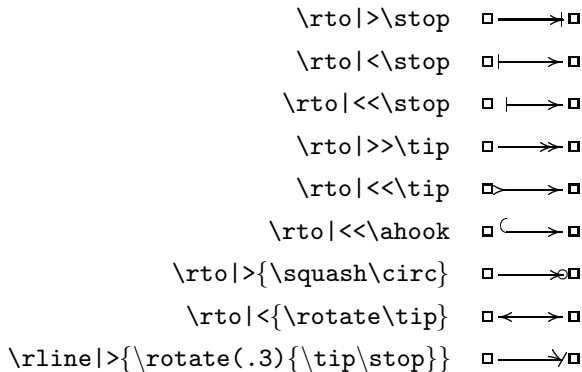
- Spojovacie čiary môžu obsahovať štandardné i špeciálne značky, ktoré môžu byť tiež rôzne kombinované



a otočené i o ľubovoľný uhol:



Špeciálne efekty možno tiež dosiahnuť umiestňovaním týchto značiek na okraje spojovacích čiar:



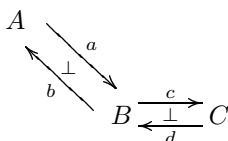
- Je tiež možné použiť niekoľko spojovacích čiar medzi dvoma objektami, pričom máme možnosť ľubovoľne meniť umiestnenie týchto čiar. Napríklad makro

```

 $\$ \backslash \text{diagram}$ 
  A \drto<1ex>^a\_bot \ \
    & B \ulto<1ex>^b \rto<1ex>^c & C \lto<1ex>^d\_bot
\enddiagram
 $\$ \$$ 

```

bude kresliť



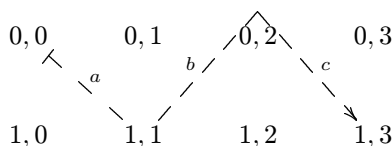
- Spojovacie čiar z počiatočného do cieľového objektu, ktoré vedú cez niekoľko iných objektov, v ktorých môžu meniť smer je možné popisovať ako „cestičku“ z počiatočného do cieľového objektu. Úseky cesty na seba navzájom naväzujú (vždy čiastkový cieľový objekt sa stáva v nasledujúcom kroku počiatočným objektom), ako je vidieť podľa nasledujúceho príkladu. Výsledkom makra

```

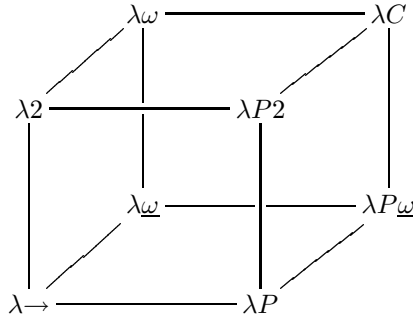
 $\$ \backslash \text{diagram } 0,0$ 
  \xdashed '[1,1]^a|\<\stop '[0,2]+<0pt,1em>^b
    [1,3]^c|\>\tip
    & 0,1 & 0,2 & 0,3 \ \
  1,0 & 1,1 & 1,2 & 1,3
\enddiagram\mathbb{D}

```

bude diagram



Uvedeným spôsobom môžeme tiež veľmi jednoducho nakresliť kocku



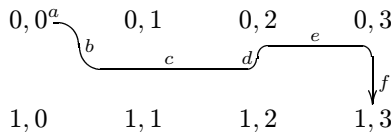
použitím makra

```

 $\def\omega{\underline{\omega}}\diagram
&\lambda\omega \rrline\xline' [1,0] [2,0]
&&\lambda C \ ddline \ \
\lambda 2\urline \rrline\ddline
&&\lambda P 2 \urline\ddline \ \
&\lambda\omega \xline' [0,1] [0,2]
&&\lambda P\omega \ \
\lambda\{to\} \rrline\urline
&&\lambda P \urline \enddiagram$ 

```

- Špeciálny druh predstavujú „cestičky“, ktoré vedú z počiatočného do cieľového objektu, pričom zachovávajú vždy smer k čiastkovému cieľovému objektu a zmena smeru sa uskutočňuje „hladko“, t.j. bez zalomenia čiary. Je možné meniť polomer krivosti v „zátačkách“. Napríklad obrázok



bol nakreslený pomocou makra

```

 $\diagram 0,0 \xto 'r[1,0]^a ' [0,2]^b '/4pt[0,2]^c
' [1,3]^d ' [1,3]^e [1,3]^f
& 0,1 & 0,2 & 0,3 \ \$ 

```



```
1,0 & 1,1 & 1,2 & 1,3 \enddiagram
```

\$\$

Objekty

- Je možné vytvárať objekty, ktorých umiestnenie nezodpovedá presne „maticovému vstupu“ a vzťahovať spojovacie čiary práve k takýmto objektom. Napríklad

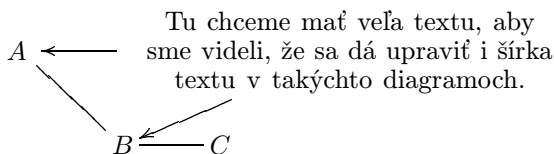
```

$$\diagram
  A \drline & \save \go+<3cm,0cm>\Drop{\Text<6cm>
    {Tu chceme mať veľa textu, aby sme
      videli, že sa dá upraviť i šírka textu
      v~takýchto diagramoch.}}
    \lto \dto \restore \\
    & B \rline & C \enddiagram

```

\$\$

kreslí diagram



zatiaľ čo výsledkom nasledujúcich makier

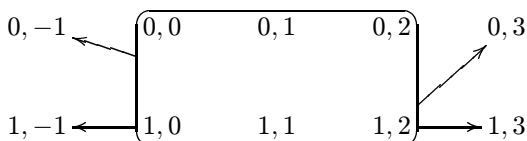
```

$$\diagram
  A \drline & \Text<6cm>{Tu chceme mať veľa textu,
    aby sme videli, že sa dá upraviť
    i šírka textu v~takýchto
      diagramoch.}
    \lto\dto \\
    & B \rline & C \enddiagram

```

\$\$

objektu, čo bude mať vplyv na zmenu dĺžky spojovacej čiary. Napríklad diagram



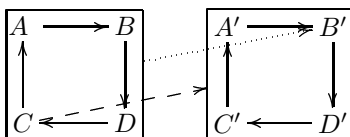
môže byť nakreslený pomocou makra

```

 $\$$ \diagram 0,{-1} & 0,0
    \save\go[1,2]\merge\framed<5pt>
    \xto[0,-1]\xto[1,-1]\xto[0,3]\xto[1,3]
    \restore
    & 0,1 & 0,2 & 0,3 \\
    1,{-1} & 1,0 & 1,1 & 1,2 & 1,3 \enddiagram
 $\$$ 

```

Rôznymi kombináciami uvedených možností možno vytvoriť s veľkou presnosťou i pomerne zložité obrázky, napríklad nasledujúci diagram



možno dosiahnuť makrami

```

 $\$$ 
\def\group#1{\save\go[1,-1]\merge
  \go[1,-1];[0,0):(0.5,0)\merge \framed\go="#g1"\restore}
\diagram
  A \rto & B \dto \group1 & A' \rto & B' \dto \group2 \\
  C \uto & D \lto & C' \uto & D' \lto
  \save \go#g1" \xdotted"1,4"|>\tip \restore
  \save \go#g2" \xdashed"2,1"|>\tip \restore
\enddiagram $\$$ 

```

Slovo na závěr

Predchádzajúce príklady snád' dostatočne 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_EXu venovať hlbšie. . .

Literatúra

1. Kristoffer H. Rose, *Typesetting diagrams with X_Y-P_IC*, User manual, 1992.
2. M. D. Spivak, *The Joy of T_EX*, American Mathematical Society, 1990.
3. M. J. Wichura, *The P_IC_TE_X manual*, T_EX Users Group, Providence, R.I., 1987.

Janka Chlebíková
chlebi_kj@mff.uniba.cs

Jak implementovat T_EX

Martin Bílý

Představte si situaci, kdy Vás navštíví přítel a poté, kdy mu na svém PC předvedete možnosti T_EXu, Vás požádá, zda byste mu „neudělal“ T_EX i pro jeho ZX Spectrum. Tenhle příklad je sice trochu nadsazený, ale má naznačit, že se v článku budu zabývat postupem při implementaci T_EXu. 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_EXu pod operačními systémy UNIXového typu.

Jazyk WEB

Velký kouzelník DEK při tvorbě T_EXu, 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

stanic firma DEC. Protože společně se samotným „výkonným“ softwarem se dodávají manuálové stránky (Manual Pages) popisující použití jednotlivých programů, měl by tento článek být zejména přehledem, které manuálové stránky souvisí s $\text{T}_{\text{E}}\text{X}$ em.

Ještě bych chtěl poznamenat, že mi je jasné, že problematiku používání a instalování $\text{T}_{\text{E}}\text{X}$ u pod UNIXem nelze obsáhnout v jednom krátkém článku a že takovýto článek bude nutně pouze přehledný. Jeho přínos by měl být v tom, že zahájí (papírově) pronikání $\text{T}_{\text{E}}\text{X}$ istů pod UNIX (resp. do (heterogenních) sítí vůbec).

Užívání $\text{T}_{\text{E}}\text{X}$ u

V této části popíšeme proces zpracovávání zdrojového textu za předpokladu, že máme „standardní“ instalaci. Pod slovem „standardní“ budeme rozumět instalaci, která je provozována v MÚ UK Praha (a byla už poskytnuta MFF UKo Bratislava a VUT Brno) — tato instalace má podobnou strukturu adresářů (popis následuje později) jako instalace $\text{T}_{\text{E}}\text{X}$ u pod DOSem, která je šířena $\zeta\text{TUG-em}$.

Editor

Jako editor pro pořizování zdrojových textů používám a doporučuji používat GNU Emacs. Tento editor je public domain a například firma Digital ho šíří jako „unsupported subset“ ve své základní instalaci operačního systému. Emacs již má standardně v sobě zabudovanou určitou podporu $\text{T}_{\text{E}}\text{X}$ u — při editaci souboru s názvem končícím na `.tex` se automaticky přepíná do tzv. $\text{T}_{\text{E}}\text{X}$ -módu — v tomto režimu je přístupná základní sada maker pro klávesnici (popis jak přepínání různých módů, tak volání maker je v základním manuálu k Emacsu, který je součástí jeho instalace). Popis $\text{T}_{\text{E}}\text{X}$ -módu je také v časopise holandského TUGu, č. 1 z roku 1992, v článku „ $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ Editing support“ (od pana Nelsona Beebeho). Tento časopis je k dispozici v MÚ UK. Celý Emacs je napsán v Lispu a v tomto jazyce lze také dopisovat další funkce podporující zpracovávání textu $\text{T}_{\text{E}}\text{X}$ em, jako např. volání překladu, preview a tisku přímo z editoru.

Úpravy tohoto typu napsal (pravděpodobně mezi jinými) pan Kristen Krab Thorup (e-mail adresa `krab@iesd.auc.dk`) z University of Aalborg (Dánsko). Tato makra jsou šířena pod názvem AUC- $\text{T}_{\text{E}}\text{X}$; pro diskusi o použití a dalším vývoji těchto maker byl založen mailing list. Informace o nových verzích se také objevují ve skupinách

`comp.text.tex` a `gnu.emacs.sources` zpráv (NetNews) šířených po USENETu. Příspěvky (články) pro `auc-tex` mailing list lze zasílat na adresu `auc-tex@iesd.auc.dk`, podobně žádosti lze posílat na adresu `auc-tex-request@iesd.auc.dk`. Adresa současného správce instalace `auc-tex` je `auc-tex-mgr@iesd.auc.dk`. Instalace maker `auc-tex` sestává ze zdrojových textů napsaných v Lispu a z několika souborů popisujících použití této nadstavby (soubor `INTRO`; v souboru `README` je popsán obsah jednotlivých souborů a také popis instalace). Poznamenejme ještě, že Emacs má i další vhodné vlastnosti — např. možnost rozdělení okna na dvě a že existují i public domain verze Emacsu pro PC (MS-DOS).

Překlad

Již výše jsme se zmínili o možnosti volat překlad přímo z Emacsu. Standardní cestou, jak volat překlad je zavolat příslušný (pro `plainTeX`, `AMS-TeX`, `LATeX`) vykonatelný binární soubor na zdrojový text z libovolného okna, ve kterém běží emulátor řádkového terminálu (`xterm`, `DECTerm`). Stejně tak si může uživatel vytvořit ve svém domovském adresáři libovolně pojmenované symbolické linky na zmiňované binární soubory uložené v adresářích, do kterých nemá oprávnění zapisovat, popř. si ve svém adresáři vytvořit své vlastní binární soubory pro překlad. Výstupní soubory (tj. `.log`, `.dvi`) jsou zapisovány do adresáře, ze kterého byl překlad volán, takže uživatel musí volat překlad z adresáře, do kterého má povolenou zapisovat.

Stejně jako pod DOSem i pod UNIXem jsou částí instalace utility pro další opracovávání `.dvi` souborů; v krátkosti je připomeneme. Program `dviconcat` spojuje dané soubory do daného výstupního souboru, vytvářejíce tak nový `.dvi` soubor použitelný k dalšímu zpracovávání. Program `dviselect` vybírá z daného `.dvi` souboru stránky a zapisuje je do nového `.dvi` souboru. Program `dvicopy` kopíruje `.dvi` soubor do jiného `.dvi` souboru, přičemž lze opět vybírat stránky a volit jiné zvětšení. `Dvitype` překládá `.dvi` soubor do čitelného tvaru. Slouží také jako program kontrolující správnost `.dvi` formátu (tj., pokud utilita `dvitype` je schopna daný soubor přečíst, potom je v pořádku) a jako příklad programu schopného číst `.dvi` soubor (pro budoucí ovladače periférií). Poznamenejme, že tento program také používá proměnnou `TEXFONTS` pro hledání `.tfm` souborů. Programy konvertující `.dvi` soubory do jiných formátů budou zmíněny dále.

Fonty

Důležitou otázkou je vztah mezi jednotlivými typy (formáty) fontů — mezi \TeX ovými `.pk` fonty a PostScript-ovými fonty při používání PostScriptu jako jazyka pro popis stránky ať už pro preview, nebo pro tisk a vztah mezi `.pk` fonty a fonty používanými aplikacemi pod X windows.

Poznamenejme nejprve, že stejně jako pod DOS-em používá i \TeX pod UNIX-em nastavení svých okolí pro hledání fontů, vstupních souborů, ... atp. Při výstavbě \TeX u pod UNIX-em se cesty k příslušným adresářům nastavují v souboru `site.h` (který je součástí balíku `web2c`), popř. si je může uživatel individuálně upravit nastavením z promptu (popř. nastavením ze svého startovacího souboru po přihlášení se počítači). Takže např. nastavením proměnné `TEXFONTS` (nebo `TEXFONTS.SUBDIR`, chceme-li po \TeX u, aby hledal ve všech adresářích od dané úrovně níže) udáváme cestu k `.tfm` souborům. Popis tohoto nastavování okolí je popsán také v manuálových stránkách k \TeX u, str. 3.

Součástí instalace DECwindows jsou i PostScriptové fonty — jejich začlenění do sazby \TeX em je řešitelné následovně: `.tfm` soubory pro PostScriptové fonty lze vytvořit z příslušných `.afm` (Adobe font metrics) souborů pomocí programu `afm2tfm`. Tento program vytváří také „virtual property list“, který lze dále překonvertovat do virtuálního fontu pomocí `vptovf`. (Tento postup je opět popsán v manuálových stránkách pro `afm2tfm` a `vptovf` — problém s `.tfm` soubory vytvořenými pouze programem `afm2tfm` je ten, že v těchto souborech chybí informace o ligaturách a dalších věcech standardních u Computer Modern fontů, zatímco hlavní výhodou virtuálních fontů je to, že používají stejné kódování jako CM fonty.)

Standardní umístění `.afm` souborů je v adresáři `/usr/lib/font/metrics`, zatímco PostScriptový popis tvarů znaků je v adresáři `/usr/lib/DPS/outline/decwin`.

Stejně jako pod DOS-em se utilita (jenda z možných) pro převod `.dvi` souboru do popisu stránky v PostScriptu jmenuje `dvips` (jde o konvertor, který napsal Thomas Rokicki). Výstup tohoto programu lze poslat na PostScriptovou tiskárnu použitím volby `-Pjménotiskárny` (podrobný popis jednotlivých voleb je opět v manuálových stránkách); pokud volba `-P` není použita, program se dívá do okolí na nastavení proměnné `PRINTER`. K nastavení voleb programu `dvips` slouží dále několik konfiguračních souborů. Každý uživatel může mít ve svém adresáři soubor `.dvipsrc` — tento soubor slouží k nastavení uživatelských voleb a program `dvips` se

do něj dívá hned po prohlédnutí souboru `config.ps`. Parametry, které se liší tiskárnu od tiskárny, mohou být nastaveny v jednotlivých konfiguračních souborech `config.jménotiskárny`. Poznamenejme ještě, že zmiňované soubory hledá `.dvips` v adresáři nastaveném proměnnou `TEXCONFIG` a že každá řádka konfiguračního souboru zachycuje nastavení jedné volby.

Další otázkou je — kde `dvips` hledá fonty, protože v jednom dokumentu mohou být použity jak Adobe fonty, tak CM (`.pk`) fonty a informace o použitých znacích musí být zapsána ve výsledném PostScriptovém popisu stránky. Situace je přehlednější v případě Computer Modern fontů, protože volbou na příkazové řádce je možné zadat cestu k `.pk` souborům, popř. je to možné zařídit nastavením proměnné `TEXPKS` (příčímž nastavení této proměnné má prioritu před výše zmíněnou volbou na příkazové řádce).

Co se týká PostScriptových fontů (ať už rezidentních v tiskárně nebo uložených na disku ve formě `.pfb` (Adobe Type 1 font program) souborů), existuje soubor `psfonts.map` v adresáři `/usr/local/lib/tex/ps/dvips` (toto je adresář daný nastavením proměnné `TEXCONFIG`), který dává programu `dvips` o existenci těchto fontů.

```
rphvb Helvetica-Bold
rphvbrn Helvetica-Narrow-Bold
        "/Helvetica-Bold .82 ExtendFont"
rpopb Optima-Bold <Optima-Bold.pfb
rpopbo Optima-BoldOblique <Optima-Bold.pfb
        "/Optima-Bold .1944 SlantFont"
```

Syntaxe řádek tohoto souboru je opět popsána v manuálových stránkách pro `afm2tfm`.

Ovladač `dvips` navíc poskytuje možnost zařadit do \TeX em sázeného textu obrázky ve formě zabalených (encapsulated) PostScriptových souborů. Cesta k adresáři pro tyto vkládané soubory lze opět nastavit volbou `S` na příkazové řádce, přičemž přednost před touto volbou má opět nastavení proměnné `TEXINPUT`. Další možností jak kombinovat \TeX pro sazbu textu a PostScript pro grafické efekty je opracovávat přímo výstupní `.ps` soubor z `dvips`.

Poznamenejme ještě, že dalším konvertorem do PostScriptu, který se šíří ve zdrojových kódech, je `dvitps` pana von Bechtolsheim.

Preview

Existují alespoň tři aplikace, které slouží k prohlížení \TeX em sázených dokumentů na obrazovce. Tyto aplikace mají společné to, že (stejně jako ostatní aplikace pod X windows resp. DEC windows) si na obrazovce otevírají své vlastní okno, se kterým lze standardním způsobem manipulovat — určovat obrazovku, na které chceme dané okno zobrazit, pohybovat s oknem, měnit jeho velikost, ... atd. Počáteční hodnoty nastavené pro start aplikací jsou v příslušných souborech v adresářích `/usr/lib/X11/app-defaults` a `/usr/local/lib/X11/app-defaults`.

PostScript Previewer

PostScriptový prohlížeč je součástí standardní instalace DEC windows — je zahrnut v části označené UDWDECW420 a jeho standardní umístění na disku po instalaci je `/usr/bin/dxpsview`. Tento prohlížeč je nutné používat tehdy, když kombinujeme \TeX s PostScriptem (i když tato kombinace znamená třeba jenom používání Adobe fontů).

xdvi

`Xdvi` je public domain program pod X windows, který slouží k prohlížení `.dvi` souborů.

Mimo jiné lze nastavit počáteční (relativní) zmenšení (v rozsahu 1 až 4), hustotu použitou při zmenšování bitmap fontů (větší hustota má za důsledek lehčí font — standardní nastavení hustoty je 40, zdá se mi, že při lehce menší hodnotě jsou znaky čitelnější. Před (znovu)zobrazováním stránky se `xdvi` dívá, jestli se `.dvi` soubor nezměnil od doby, kdy byl naposledy použit pro preview. Pokud se změnil, pak `xdvi` se automaticky znovu inicializuje s novým `.dvi` souborem. Toto umožňuje prohlížet více verzí toho samého dokumentu při jednom spuštění `xdvi`. Aplikace `.xdvi` má v sobě zabudovanou lupu (která funguje stejně při jakémkoliv nastavení zmenšení), která ukazuje nezmenšený obraz okolí kurzoru (velikost tohoto okolí závisí na tom, které z tlačítek myši použijeme k zavolání lupy). Zvětšovaná oblast se pohybuje současně s pohybem myši.

`Xdvi` používá bitmap-ové fonty ve formátech `.pk`, `.gf`, `.pxl`. Tyto fonty hledá podle nastavení proměnné `XDVIFONTS`. Další proměnná, `XDVISIZES`, ukazuje, které velikosti jsou dostupné

xtex

`Xtex` je další public domain aplikací pod X windows sloužící k prohlížení `.dvi` souborů. Běh této aplikace velmi závisí na nastavení hodnot v souboru `/usr/local/lib/app-defaults/Xtex` a bez instalace (a vyladění) tohoto souboru nebude (řádně) pracovat. Tento prohlížeč povoluje

také použití PostScriptových fontů (cesta k nim je dána proměnnou `TEX-FONTS`). `Xtex` používá fonty pro X windows ve formátu `.pcf` (portable compiled font) uložené v adresáři `/usr/local/lib/tex/xtex/pcf`. Tyto soubory jsou vytvořeny z `.pk` souborů následujícím způsobem: nejprve se vytvoří `.bdf` (bitmap distribution format) soubory utilitou `mftobdf` — popis, jak spočítat správnou velikost fontu, je popsána v manuálových stránkách pro `xtex` (tato hodnota závisí na rozlišení obrazovky, požadovaném zvětšení a na původní hustotě `.pk` fontů); ve druhém kroku se potom již vytvoří `.pcf` fonty použitím překladače `dxfc`, který je částí instalace DECwindows. Poznamenejme ještě, že další programy související s tímto problémem, jsou `TEXtoXfont`, `bdftopcf`, `mkfontdir`.

Tisk

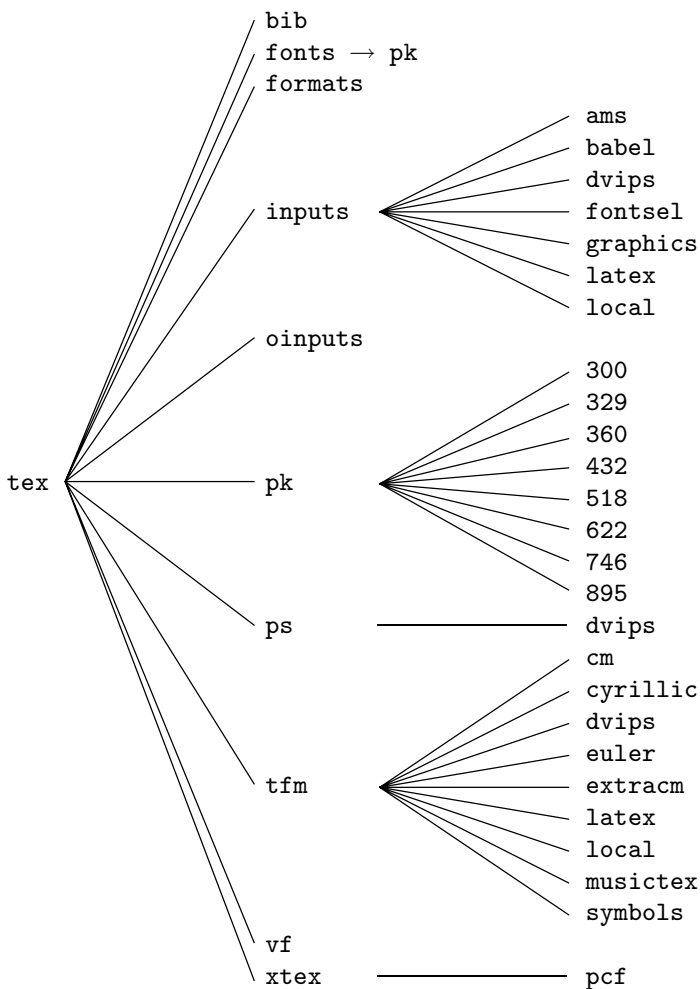
Rozdíl v tištění `TEX`ových dokumentů pod UNIXem je dán pouze jinou správou tiskáren (a pochopitelně jinými příkazy). Zpracování `.dvi` souboru příslušnými ovladači (`dvips`, `dvi2lj`) je stejné jako v DOSu (popř. jako u `preview`).

Instalace T_EXu

Struktura adresářů instalace, kterou popisujeme, je uvedena na následujícím schématu:

Hlavní rozdíl mezi instalováním `TEXu` na počítač, který provozuje MS-DOS a na počítač používající některou z variant UNIXu, je ten, že zatímco pod DOSem kopírujeme již připravené binární soubory, tak pod UNIXem je třeba příslušné binární soubory vytvořit ze zdrojových (napsaných v jazyce C) kódů (pokud nemáme k dispozici tyto binární soubory již připravené a fungující na binárně kompatibilním počítači). Úspěšný překlad předpokládá editování příslušných `makefile` souborů (popř. dalších souborů — `site.h`, `local-defs`, ...) a v nich nastavení proměnných používaných při překladu tak, aby odpovídaly dané architektuře počítače, danému operačnímu systému a zvolené struktuře adresářů. Soubory, které je potřeba editovat, jsou vyjmenovány v README souborech u každé jednotlivé aplikace. Např. správné nastavení proměnné `TEXPKS` tak, aby `dvips` našel `.pk` soubory (v naší instalaci) je „`TEXPKS = /usr/local/lib/tex/pk/%d/%f.%p`“.

/usr/local/lib/tex



Doplňující informace

Některé z informací o dostupnosti T_EXu a souvisejícího softwaru a materiálů po počítačových sítích jsou již „obecně“ známé díky (i v ČSFR)

relativně rozšířené síti EARN (a byly publikovány i v tomto bulletinu), některé souvisí se sítí INTERNET (resp. USENET), tedy přirozeně s UNIXem, a proto je zde uvedeme.

Přímo T_EXu je věnovaná „NetNews group“ `comp.text.tex`, informace o T_EXu (fontech) lze najít i ve skupině `comp.fonts`.

Informace o tom, na kterých uzlech je dostupný jaký T_EXový software, byla (leden 1992 — snad stále je) v souboru `pub/raymond/tex.supplement.Z` na adrese `math.berkeley.edu`. Pokud byl tento soubor již někam přenesen, potom informaci o nové adrese by měl vědět majitel adresy `spunky.cs.nyu.edu`, který je (?) správcem zmiňovaného souboru. Tento soubor obsahuje seznam hlavních FTP serverů, rady, jak získat po síti instalace T_EXu jak pro UNIX tak pro MS-DOS a hlavně abecední seznam public domain T_EXového sw spolu s adresami příslušných FTP serverů, odkud lze daný sw získat.

Články věnované T_EXu pod UNIXem se objevují i v TUGBOAT-u. Podle jednoho z těchto článků lze získat „skoro všechny“ T_EXový (UNIXový) sw od jedné z pracovních skupin sdružení DECUS; adresa je DECUS Library, 219 Boston Post Road, Marlboro, MA 01752-1850.

Zbyněk Linhart

Jako volnou ilustraci k článku Zbyňka Linharta T_EX a UNIX uvnitř tohoto čísla přinášíme ukázkou toho, co dovede METAFONT v rukou šikovného programátora *Ládi Lhotky*, kterého tímto zároveň vítáme zpátky na československé půdě. Na jeho další (tentokrát už vědomé) příspěvky do Zpravodaje se jistě všichni těšíme.



Vážení $\text{T}_{\text{E}}\text{X}$ isté,

jelikož mi již černé svědomí nedalo spát, musím se alespoň touto formou ozvat k otázce nákupu editoru ζTUGu (prý se rozhodne už v září).

Dosavadní diskusi na stránkách $\text{T}_{\text{E}}\text{X}$ bulletinu jsem nevěnoval příliš pozornosti s tím, že recenzenti editoru s ním budou jistě dokonale seznámeni a budou mít široké zkušenosti z vlastních pracovišť.

S hrůzou jsem ale přečetl číslo 2, podle kterého panuje velká nerozhodnost a nezájem o tuto otázku. Proto cítím jako svou morální povinnost vyjádřit se k ζEDu .

Trochu historie: Na kterési $\text{T}_{\text{E}}\text{X}$ ové akci v Karlíně se Petr Sojka zmínil, že jeho známý Petr Ševeček nabízí konzistentní instalaci $\text{emT}_{\text{E}}\text{X}$ u (za cenu disket) spolu s vlastním editorem a přejatým českým a anglickým spelling-checkerem za asi 5 000 Kčs. Protože na našem pracovišti (astronomický ústav ČSAV) začala popularita $\text{T}_{\text{E}}\text{X}$ u vzrůstat a ukradený $\text{PCT}_{\text{E}}\text{X}$ měl spoustu nedostatků (a navíc byl nelegální), rozhodli jsme se systém a editor ζED tehdy verze 1.2 koupit.

Překvapil na první pohled. Měl perfektně řešenou češtinu s možností vlastní definice klávesnic i screen fontů, zabudovaný spellchecker fy Delta, pull-down menu a spoustu komfortních funkcí, obdélníkové bloky, budíček, hodiny a obrovské možnosti maker. Sám se adaptoval na grafickou kartu a na rozdíl od T602 tehdejší verze 1.x nevypadal tak „pouťově“ — myslím tvary znaků a jejich čitelnost. A měl až 9 oken s možností přesunu bloku mezi nimi bez clipboardu. Navíc byl zaměřen velmi efektivně programátorsky, a tak nezobrazoval WYSIWYG fonty, ale tiskový program se vyvolával zvláště s možností definice tiskáren atd. Umožňoval změnu kódování, a to dokonce podle Novákových fontů $\text{T}_{\text{E}}\text{X}$ u (ty jsme ale bohužel neměli). Spolu s ním přišel i konvertor, který konvertoval mezi editorem a T602, WP a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ovými makry $\backslash\text{it}$ $\backslash\text{bf}$ atd. Samozřejmostí bylo ovládání podle Wordstaru resp. Borland Turbo editorů. A hlavně s $\text{emT}_{\text{E}}\text{X}$ em sám skákal na řádek s chybou při překladu (při použití jednoduchého menu) a skákal i na další chybu podle logu.

Začátkem letošního roku jsme upgradeovali na novou verzi 1.4., které jsme navíc dostali překladové slovníky č-a a a-č podle Oplatka (zabudované do menu). To bude patrně recenzovaná verze. Proto nyní můj pohled na tuto verzi, popř. doplnění recenzí:

klady:

- možnost uložení konfigurací i maker do konfiguračního souboru, který lze udat na příkazové řádce, popř. použít soubor v uživatelském adresáři, a pokud není, použije se standardní nastavení z adresáře editoru. Tak lze mít v každém adresáři soubor s jinými makry a parametry
- možnost definice vlastní grafické karty (i větší rozlišení podle kódu pro int10 — jako `dviscr emTeX`)
- budíček (mně funguje dobře a používám jej, nefunguje pouze, jsem-li v DOS shellu, ale po návratu zpět mě hned upozorní)
- práce s bloky, i obdélníkové, snadný přesun mezi okny
- vyhledávání párových závorek
- možnost abecedního řazení i českého v daném sloupcovém bloku
- dokonce i sčítání čísel pod sebou v bloku (v `TeX` se asi neužije moc)
- vyhledávání a substituce jdou i v rámci bloku
- možnost vlastní definice klávesnice, a to i polohy horké klávesy pro háček, lze např. udělat i háček a š, což dá # (nemusím přepínat sclock do US klávesnice)
- fajnovosti záměny textu — např. v daném bloku nahradit velká písmena malými nebo opačně, zrušit diakritiku, nahradit měkké mezery tvrdými atd.
- korekce pro `TeX` (ignorování command sekvencí a závorek)
- oddělení tiskového programu zvlášť, většinou se moc nepoužije, protože tisknout lze přímo blok do PRN, lze definovat i vlastní fonty pro DOWNLOAD
- automatické zálohování, které proběhne samo po dané době (a nemusím něco zmáčknout jako u ChiWriteru), naopak pokud jsem soubor nezměnil, nedělá se. To je podle mě ideální a odpovídá to požadavku kolegy Demela. (Provede se vždy po uplynutí nastavené doby v celou minutu, tedy pokud jsem nastavil dobu na 1 minutu a napíšu něco v 56. sekundě, uloží se to se zdvořilým upozorněním „moment prosím“ za 4 sekundy.)

- možnost zadat jak 1 DOS příkaz (např. používám nc, popř. logování příslušné síťové tiskárny) tak po ENTERu skočit do DOS Shellu (pro J. Chlebíkovou)
- spousta dalších funkcí, která mně jako uživateli ζ EDU připadá samozřejmá, pro uživatele jiného editoru by to bylo překvapení

zápory:

- nepracuje s myší, a co je horší, ani s různými simulátory klávesových sekvencí, dodávaných s myší (Genius gmmenu, Logitech logimenu atd.)
- neumí pracovat s libovolně dlouhými soubory jako Norton Editor. To by opravdu chtělo dodělat;
- nemá funkci UNDO, která by si pamatovala veškeré sekvence (nejen deletované řádky) — to je potřebné při nahrazování pomocí složitějších maker;
- má poměrně stručný help, kde není vše, co v manuálu — zde nechápu, co je v recenzi zmiňovaný kontextově závislý help k $\text{T}_{\text{E}}\text{X}$ u, patrně novinka, kterou nám autor zapomněl asi oznámit v rámci upgrade programu. (To bych mu asi vytkl obecně, protože o nové verzi jsem se musel dozvědět sám z reklamy na INVEXU — výhody legální instalace i vyplněné registrační karty se jaksi vytratily.)
- Při zapnutí režimu word wrap sice hezky formátuje zdrojový text, ale dá tam měkký konec řádky 0E 0A, což $\text{T}_{\text{E}}\text{X}$ nebere a hlásí chybný znak $\text{^}K$, přesněji řečeno to asi ignoruje jako mezeru, a teprve je-li řádek dlouhý (odhadem 350 znaků?), přeteče. Zajímavé je, že si toho nikdo z recenzentů nevšiml. Já potom musím označit celý text jako blok a dát funkci $\text{^}KG$ — rozdělení odstavce na samostatné řádky. Proto radši WW vypínám a dělím si řádky sám enterem, což ale je nepohodlné, a při psaní bez sledování obrazovky mám dlouhé řádky, které musím dělit dodatečně. Navíc soubory s 0E 0A nelze používat v jiných programech, které vyžadují 0D 0A.

Podle mého mínění, jde o editor, který má asi nejbliž k požadavkům uživatele $\text{T}_{\text{E}}\text{X}$ u a má i slibnou budoucnost pro použití jako univerzální editor pro programátory. Nyní připomínky, co by se mělo zlepšit a co by mělo naše sdružení požadovat v případě zakoupení multilicence:

- 1) práce s myší, a to tak, aby kurzor myši byl nezávislý na kurzoru klávesnice (to např. T602 nemá). Tak si mohu označit začátek bloku kurzorem a konec myší a opačně. Volitelná tlačítka myši by měla mít možnost konfigurace např. pro volání blokových operací, popř.

uživatelských maker — např. abych měl pravé tlačítko na přesun bloku a prostřední na volání svého makra např. na $\sim W$

- 2) práci s libovolně velkými soubory (jako např. Norton Editor)
- 3) plnohodnotné UNDO
- 4) možnost konfigurace tak, aby se měkké konce řádky nedělaly a bylo možné použít wordwrap mód, nebo lépe aby se při uložení souboru nahradily samy měkké konce tvrdými 0D 0A.

Podle možností by bylo dobré respektovat i připomínky recenzentů, které zde shrnu:

- práce s okny vedle sebe, popř. se překrývajícími. Možnost posuvu i změny velikosti ze všech rohů myši.
- oprava zobrazení dlouhé cesty ve stavové řádce
- možnost vypnutí pamatovací řádky dole, popř. doplnění všech příkazů od dané sekvence i s momentálním zvětšením — tím chci říci, že zmáčknou-li ctrl-K, objeví se mi všechny možnosti pokračování, nejen vybrané, pokud je jich moc, udělat 2–3 řádky dole.
- možnost označení celé řádky jako bloku bez skoku na začátek
- u hledání párových závorek nepřesouvat textem, pokud mám závorky v okně
- vyhledávání párů Begin, End popř. dolaru a dvojdolaru (tak aby hledal právě 1 nebo právě 2 dolary.
- Možnost při vyhledávání řetězců hledat kombinace více mezer a tvrdých konců řádků — prostě \TeX ovsky — toto ale udělat vypínatelné, někdy je třeba hledat slovo s právě 1 mezerou — u delimitovaných parametrů, popř. možnost nějak vložit jako součást vyhledávacího řetězce Newline (např. Norton Editor to umí).
- možnost zrušení barevného označování jiných druhů písma, protože mám-li v textu tabulátor 09, tak je zvýrazněn celý řádek červeně jako kurzíva, což ruší
- bylo by krásné, kdyby editor umožňoval i měnit texty v binárních souborech, tzn., že by nezobrazitelné znaky zobrazil jako např. tečku a nekládal by nic, jen by soubor rozřezal podle nastavených okrajů (musel jsem jednou např. nahrazovat v dvi souboru odkazy na amr fonty odkazy na cmr, a jelikož jsem nenašel na PC žádný editor, musel jsem psát program v Pascalu), kdyby ještě umožňoval vkládat znaky, které mohou dávat pomocí ALT xxx z klávesnice, hlavně form feed, newline a 000, a měl možnost zobrazovat a editovat v hexa módu, stal by se jistě i oblíbeným editorem programátorů — ne \TeX istů. Ale to

jsem se dostal od požadavku na \TeX editor daleko, nebo myslíte, že ne?

Vzhledem ke kódování povelů jako pro Turbo editory se domnívám, že by přechod na \zS ED nebyl pro nikoho tak těžký, a jistě by výhody nového \zS EDu rychle docenili i ne \TeX isté (zvláště při splnění posledního bodu). Uživatelé na našem ústavu si ani jiný editor pro \TeX ování a psaní fortranských programů než \zS ED představit neumí. Po Norton editoru sahají jen kvůli dlouhým souborům!!

Nechť je tento příspěvek považován zároveň za kladné hlasování nejméně 10 lidí (z kolektivního člena) a mě individuálního za zakoupení a hlavně vylepšení \zS EDu pro potřeby \zS TUGu.

Petr Škoda,
astronomický ústav ČSAV Ondřejov
aststel@csearn

Program MNU: Konfigurovatelné menu pro spouštění aplikací pod DOSem

Petr Olšák

Československé sdružení uživatelů \TeX u zpřístupnilo svým členům i veřejnosti program \TeX pro počítače IBM PC ve formě balíku programů em \TeX , který pochází z dílny Eberharda Mattese. Do tohoto balíku byl zařazen už dříve používaný systém MENU, jehož autorem je Jiří Hnat. MENU umožňuje spouštět jednotlivé programy, související s přípravou dokumentu v \TeX u.

Tento systém MENU má mnoho předností a několik nevýhod. Inspiroval jsem se jeho přednostmi a poučil z jeho nevýhod a vytvořil jsem jiný program, který by mohl stávající systém MENU nahradit.

Aby se to nepletlo, nazval jsem tento nový program podobně: MNU. Výsledek své práce jsem předal k volnému šíření \zS TUGu jako dárek ke konferenci konané v září.

Následuje stručný „výtažek“ z asi 25 stránkové dokumentace, která je napsaná v Plain \TeX u. Tento text má umožnit udělat si obrázek o tom, co program dovede a co ne.

25. 8. 1992

Petr Olšák

1. Stručná charakteristika programu MNU

Programem MNU lze pomocí řídicí dávky (`.bat`) opakovaně spouštět různé aplikační programy. Byl vyvinut pro použití v systémech \TeX , ale lze jej užít i jinde.

Spuštěním dávky `demotex.bat` spustíte demonstraci chodu jednak vlastního programu MNU, jednak jeho navrhované konfigurace `texcfg.mnu`, která je vhodná pro systémy \TeX .

Dokumentace je určena pro uživatele, kteří hodlají měnit stávající konfigurace systémů, v nichž je program použit, nebo navrhovat nové. Proto předpokládá určitou hlubší znalost DOSu, zvláště pak zkušenosti s navrhováním dávek (`.bat`) v DOSu.

Program vytváří zdání integrovaného prostředí pro spouštění jednotlivých procesů. Toto prostředí lze zcela nakonfigurovat, včetně barev a tvarů oken, jmen a způsobu použití jednotlivých položek menu, kontextového helpu apod. Přitom řízení procesu je zajištěno řídicí dávkou. Tato dávka v jistém okamžiku zavolá program MNU s parametrem — názvem konfiguračního souboru (např. `texcfg.mnu`). Program MNU umožní uživateli komfortně vybrat požadovanou položku. Po výběru položky vrátí program příslušný chybový kód odpovídající číslu vybrané položky a ukončí svou činnost. V dávce se nyní „zjistí“ číslo vybrané položky pomocí skupiny příkazů `if errorlevel` a podle toho se dávka větví na jednotlivé aplikační procesy. Po provedení aplikace dávka většinou znovu zavolá program MNU a uživatel znovu vybere nějakou položku a tak pořád dokola.

Dávka může s programem MNU komunikovat i pomocí „environment proměnných“ DOSu a vstupních a výstupních souborů. Program může (je-li to řečeno v konfiguračním souboru) zobrazovat a editovat obsahy těchto proměnných. Navíc existuje jedna speciální environment proměnná s názvem MNU, která ovlivní start programu — po spuštění programu se nabídne jako aktivní ta položka, která je specifikována právě v této proměnné MNU. Uživatel proto nemusí dlouho vybírat následující položku, nevymyká-li se postup jeho činnosti normálu. Napří-

klad po ukončení překladu (programu $\text{T}_{\text{E}}\text{X}$) lze v dávce zjistit příkazem `if errorlevel`, zda tento překlad proběhl bez chyby. Pokud ano, pak se provede nastavení proměnné `MNU` tak, aby po vyvolání menu byla nabídnuta jako aktivní položka `View`. Uživatel pak nemusí v prostředí menu dělat nic jiného, než mačkat `Enter`.

V následující tabulce je seznam souborů, které jsou součástí dodávky programu `MNU`.

<code>!readme.txt</code>	Základní informace o programu
<code>mnu.doc</code>	Podrobná dokumentace k programu
<code>mnutrick.doc</code>	Popis některých triků použitých v dávkách
<code>mnu.cs</code>	Obsah předchozích tří souborů v Plain $\text{T}_{\text{E}}\text{X}$ u
<code>demotex.bat</code>	Demonstrační dávka.
<code>mnu.exe</code>	Vlastní program <code>MNU</code>
<code>texcfg.mnu</code>	Navrhovaná konfigurace programu <code>MNU</code> pro $\text{T}_{\text{E}}\text{X}$
<code>totext.exe</code>	Vrací obrazovku do textového módu 80×25
<code>*.bat</code>	Navrhované dávky pro konfiguraci em $\text{T}_{\text{E}}\text{X}$ u
<code>kalk.exe, kalk.hlp</code>	Kalkulačka — ukázka aplikačního programu

Software, obsažený ve výše zmíněných souborech, je volně šiřitelný. Autor jej předává sdružení ζTUG , od kterého očekává, že zajistí popularizaci tohoto produktu a další jeho šíření. Není dovoleno uvedené soubory ani jejich modifikace používat k výdělečným účelům. To znamená, že není dovoleno tento software prodávat (s výjimkou manipulačního poplatku za kopírování disket) ani zařazovat do systémů, jejichž provoz je určen výhradně k výdělečné činnosti.

Autor nebude v zásadních koncepčních otázkách tento software dále upravovat a vylepšovat. Bude ale vděčný za všechny připomínky, které povedou k odstranění chyb, které bohužel u žádného software nejsou vyloučeny.

2. Popis programu

Program MNU pracuje v textovém režimu obrazovky 80×25 . Je-li výchozí stav obrazovky jiný, program nezobrazuje správně. V takovém případě je vhodné nejprve volat program `totext.exe`, který vrací obrazovku do uvedeného textového režimu.

V textovém režimu není program MNU samotný závislý na způsobu reprezentace národních znaků. Veškeré texty, které zobrazuje, čte z konfiguračního souboru, který musí být připraven v závislosti na použitém kódování. Správné zobrazování národních znaků na obrazovce nespadá do kompetence programu a je třeba použít jiné technické prostředky.

Program se spouští s aspoň jedním parametrem. Tímto parametrem je název konfiguračního souboru (popřípadě i s cestou). Další parametry jsou nepovinné. Jedná se o názvy vstupních a výstupních souborů a program s nimi může nadále pracovat podle příkazů z konfiguračního souboru.

Pokud je program spuštěn bez parametru, ohlásí chybu a ukončí činnost s návratovým chybovým kódem 255. Stejně tak v případě, že konfigurační soubor nelze otevřít ke čtení.

Pravidlo: Kdykoli program ukončí činnost kvůli nějaké chybě, vypíše se znění chyby a návratový chybový kód je 255.

3. Údaje v konfiguračním souboru

Z konfiguračního souboru program načte následující údaje:

- Znění (texty) jednotlivých položek
- Čísla položek — každá položka má své číslo (nezáporné číslo typu integer)
- Způsob reakce programu při výběru položky
- Seskupení položek do jednotlivých oken
- Struktura vzájemného volání jednotlivých položek
- Typ položek (editovatelná a needitovatelná)
- Doprovodné texty v jednotlivých oknech
- Zobrazení obsahu souborů, resp. environment proměnných v oknech
- Pro každé okno jeho velikost a umístění na obrazovce
- Všechny potřebné atributy barev
- Zvýrazněná písmena v textech jednotlivých položek

- Texty helpů, jejich napojení na jednotlivé položky
- Velikosti, umístění a barvy oken helpů
- Způsob komunikace s vnějším prostředím (vstupní a výstupní soubory, environment proměnné)
- Zvuková odezva programu při nesprávně zmáčknuté klávese

V konfiguračním souboru je plno dalších informací. Stručně se dá říci, že nakonfigurovat lze prakticky všechno.

4. Volba aktivní položky při startu programu

Po spuštění programu se nabídne jako aktivní ta položka, jejíž číslo je shodné s číslem environment proměnné MNU.

Příklad volání programu :

```
set MNU=17
mnu config.mnu
```

V tomto případě se nastaví při startu jako aktivní položka s číslem 17. (Předpokládáme, že v souboru `config.mnu` je taková položka definována).

Pokud taková položka je v nějakém dílčím „pod-menu“, které vyplývá ze struktury vzájemné vazby oken a položek, otevřou se automaticky všechna „rodičovská“ okna v této struktuře a potom teprve okno s položkou.

Jestliže environment proměnná MNU neexistuje nebo její obsah není číslo typu integer, bere se, jako by měla nulovou hodnotu.

Není-li v konfiguračním souboru položka s číslem MNU definována, program ohlásí chybu a ukončí činnost.

5. Výběr položky z klávesnice

Uživatel mění aktivní položku pomocí šipek v rámci jednoho okna. Výjimku tvoří tzv. „roletové“ menu, kde šipky doprava a doleva vyvolají přeskok do sousedních oken (to je nutné ovšem přesně definovat v konfiguračním souboru).

Klávesa `Esc` vrací výběr položek k „rodičovskému“ oknu. Neexistuje-li takové okno (aktivní okno je „kořenem“ struktury), pak program vyhledá položku s číslem 0. Pokud taková položka neexistuje, program ukončí činnost a vrátí řízení DOSu s kódem 0. Pokud položka existuje, program

otevře příslušné okno a nabídne tuto položku jako aktivní. Tento postup umožňuje nakonfigurovat závěrečnou otázku typu „Opravdu chcete ukončit činnost?“, přičemž položka s číslem 0 může mít znění **Ano**.

Reakce klávesy **Enter** může být trojí:

- Otevře se nové okno s položkami. Přesněji: nabídne se specifikovaná položka jako aktivní.
- Program ukončí svou činnost, pozavírá všechna otevřená okna a vrátí chybový kód rovný číslu vybrané položky. Je-li číslo položky větší než 254, program vrátí kód 255.
- Program spustí proces definovaný jedním řádkem DOSu (tzv. **řádkové spouštění procesu**). V tom případě program neopouští paměť počítače, spustí požadovaný proces a po ukončení procesu nabídne další specifikovanou položku jako aktivní.

Způsob reakce klávesy **Enter** se definuje v konfiguračním souboru.

Zvláštním typem je tzv. „editovatelná“ položka. Umožňuje editovat v místě položky. Výsledek editace se ukládá do speciálního bufferu pro pozdější užití v „řádkovém“ spouštění procesu nebo při zařazení do výstupních souborů. Tímto způsobem lze změnit obsah environment proměnných DOSu, nebo připravit parametry pro nějaký řádkový proces apod.

Zabudovaný editor položek pracuje v „insert“ módu. Klávesy **Esc**, **Enter**, šipky, **Del** a **Backspace** reagují běžným způsobem. Je-li jako první klávesa stisknuta šipka, nabízený text lze editovat. Jinak původní text mizí a je možno editovat nový.

Klávesa **F1** vyvolá help závislý na právě aktivní položce. Okna helpu se konfiguruji v konfiguračním souboru a mohou obsahovat kdekoliv další položky (zvýrazněné pojmy), z nichž se skáče na další okna helpu. Tím lze dosáhnout strukturovaný help. Help se opustí klávesou **Esc**. V případě, že okno helpu neobsahuje žádnou položku, pak klávesa **Enter** má stejný význam, jako klávesa **Esc**.

Pokud je v položce definováno tzv. „zvýrazněné písmeno“, lze vybrat danou položku pouhým zmáčknutím klávesy s tímto písmenem. Reakce programu je stejná, jako při použití šipek a zmáčknutí **Enter**.

V případě jakékoli nesrovnalosti v konfiguračním souboru (např. nelze nalézt číslo navazující položky) program ohlásí chybu a ukončí činnost.

6. Výběr položky myši

Je-li instalována myš, na obrazovce se objeví myší kurzor. Zmáčknutí levého tlačítka myši s kurzorem nad položkou kdekoli na obrazovce způsobí výběr této položky (jako **Enter**). Zmáčknutí téhož tlačítka mimo okno reaguje podobně jako **Esc**, tj. zavře se naposledy otevřené okno. Pravé tlačítko volá a odvolává helpy. Prostřední tlačítko je bez funkce.

Měl jsem možnost odladit program pouze s myší „genius mouse“. Je možné, že jiná myš nebude poslouchat (např. „Mickey mouse“).

7. Program MNU jako strukturovaný help

Protože program MNU dokáže zobrazovat nejen položky, ale i okolní text v oknech, která jsou plně konfigurovatelná, je zřejmé, že programem MNU lze vytvořit strukturovaný help. Položky jsou v tomto případě zvýrazněné pojmy helpu, po jejichž zvolení klávesou **Enter** program skáče na další příslušné okno a tak pořád do omrzení.

Každý slušný editor umožňuje spustit řádek DOSu a má navíc možnost definovat makra. Když se tyto dvě věci spojí dohromady, může se například klávesou F1 v editoru vyvolat program MNU, který začne zobrazovat help třeba k \LaTeX u. Bylo by pěkné mít stručný přehled příkazů \LaTeX u v helpech.

Zde by se asi projevila nevýhoda programu MNU — že informace o každém okně je vyhledávána sekvenčně řádek po řádku od počátku souboru, a tedy na některých počítačích dlouho. Chtělo by proto učinit několik experimentů, zda by to stálo za námahu. Můžete uspořádat závody: Někdo vyhledá příslušný pojem programem MNU v rozsáhlém helpu a jiný učiní totéž ručně v knize [Leslie Lamport: *A document preparation system*].

8. Výhody a nevýhody programu

Mezi výhodu programu asi patří, že v okamžiku běhu aplikace nepřekáží vůbec v paměti. Dále v řídicí dávce lze prostředky DOSu vyjádřit vše, co je v konfiguračních možnostech jednotlivých aplikačních programů. Navíc program MNU je plně konfigurovatelný. Vůbec není omezen jen nějakými druhy aplikace (systém \TeX je jen příklad jeho užití).

Nevýhoda programu je v tom, že je plně konfigurovatelný. Divíte se, že výhoda je zde shodná s jeho nevýhodou? Problém je v tom, že popis for-

mátu konfiguračního souboru programu je z uvedených důvodů relativně složitý a tvorba nové konfigurace u tak komplikovaných systémů, jako je TEX , vůbec není „user friendly“. Je potřeba vytvořit nejen konfigurační soubor programu, ale také řídicí dávku, která bude řídit spouštění jednotlivých procesů. To vyžaduje jisté zkušenosti s vytvářením dávkových souborů DOSu.

Na druhé straně se ale program odvděčí tvůrci takové konfigurace tím, že umožní řídit nakonfigurovaný balík programů „user friendly“ způsobem.

Aby se kompenzovala nevýhoda programu, spočívající v obtížnosti tvorby nových konfigurací, je s programem dodávána řídicí dávka `texbat.bat`, další podružné dávky a konfigurační soubor programu s názvem `texcfg.mnu`. Tyto soubory slouží jako ilustrace pro vytváření Vašich konfigurací. Jsou zkopírovány z fungující konfigurace em TEX u a v dokumentaci k programu MNU je rozsáhlý komentář k obsahům uvedených souborů. Forma tohoto komentáře je podobná jako popis PLAINu v Appendixu B knihy [Donald E. Knuth: *The TEX book*].

Navíc program MNU umožňuje vytvářet instalační dávky, které se chovají „user friendly“. Věřím, že pokud ζTUG bude distribuovat systém TEX s tímto programem, tak někdo vytvoří takovou instalační dávku, že novopečený uživatel TEX u po napsání příkazu `install` bude slušně dotázán na to, co si přeje mít a kde si to přeje mít, a vše se samo nainstaluje. Pokud nebude chtít nějaké „speciality“, nebude vůbec muset rozumět vnitřní konfiguraci systému.

Nová konfigurace $\zeta\text{T}\text{E}\text{X}$ u

Na valné hromadě ζTUG u během konference Euro TEX '92 bylo řečeno, že kolega Petr Olšák, který věnoval ζTUG u nové prostředí pro práci s TEX em a podpůrnými programy, je ochoten se podílet na přípravě nové verze $\zeta\text{T}\text{E}\text{X}$ u, v němž by byl jeho program použit.

Tato nová instalační verze by pokud možno měla vyhovovat co nejširšímu okruhu zájemců. Měla by být také snadněji přizpůsobitelná provozu na počítačových sítích a měla by obsahovat osmibitové fonty.

Žádáme proto všechny uživatele \TeX u, kterým z jakýchkoliv důvodů dřívější instalace nevyhovuje, necht' laskavě zašlou veškeré své připomínky či návrhy do tří týdnů po obdržení tohoto \TeX bulletinu na některou z adres:

Oldřich Ulrych, MÚ UK, Sokolovská 83, Praha 8 – Karlín
e-mail: oulrych@cspguk11.bitnet

Petr Olšák, Katedra matematiky, FEL ČVUT, Žitkova 4,
Praha 6 – Dejvice.

e-mail: olsak@csearn.bitnet

Oldřich Ulrych



OBNOVIT ČLENSTVÍ V NAŠEM SDRUŽENÍ.

Karel Horák

Oceano T_EX

I

Elle contemple l'écran, tout vert dans les ténèbres,
Où les chiffres s'affichant jettent des leurs d'algèbre.
Viendra-t-il, le chapitre inscrit dans les mémoires?
O touches, que vous savez de lugubres histoires!
Combien de mots, de lignes, se sont évanouis,
Dans l'aveugle machine à jamais engloutis!

II

Mais le menu s'affiche, rassurant, familier,
Tiens, dit-elle en souriant, le voici tout entier.

III

Sur la table, entassé, l'article et ses progrès,
Certain, démontré presque, et probable à peu près,
Attend, bloc annoté, qu'on veuille bien l'imprimer
Symbole après symbole, comme Sisyphe con rocher.
Ratures, sens perdu, doute, feuillet manquant,
Partout la question triple: –Comment? Où? Quand?

IV

Mais le mot, qu'on le sache, est un être vivant,
La main de l'auteur vibre et tremble en le frappant.
La pagination à l'infini s'échappe,
A chaque instant lacune, embûche, chausse-trappe.
Faut-il mettre un backslash ou bien une cédille,
Au bout du théorème et de ses codicilles?
Regardons le manuel. Ces choses-là sont rudes,
Il faut pour les comprendre avoir fait ses études!

V

La tâche se termine, le dernier mot s'inscrit.

Elle appelle T_EX. Il vient, suppute, renifle, écrit,
Réclame deux dollars avec une parenthèse,
Accentue, calcule en toute hypothèse
Le lieu mobile, obscur, capricieux, changeant.
Où se plaît la virgule aux negeoires d'argent.
Oh cliquetis des phrases, tohu-bohu, rumeur,
De l'univers des mots le terrible écumeur!

VI

T_EX termine son travail; après tant de souffrance,
Tant de labeur obtient enfin sa récompense.
L'imprimante achevant lentement ses devoirs
Eclaire tout à coup dans ses jambages noirs
Le théorème orné de tous ses corollaires,
Eblouissant Shakespeare et ravissant Euler!

Victor Hugo
p.c.c. *Bernard Beauzamy*

I

Ono to děvče luští displej, temnej a celej zelenej,
z něj srší drzé číslice šajn nepochopitelnej.
Zjev se mi, kódičku, zapsaný v paměti,
když moje klávesky nemohou věděti
nic o slovech, řádcích a chybové latině,
co navždy zmizely v tý hloupý mašině.

II

A vida, menu je tady, důvěrné a jasné,
konečně ozdobi úsměv líc dívky krásné.

III

Mrak předloh a šmíráků na stole leží,
něco už načisto, na něčem nezáleží.
Těhotný poznáemek blok na zázrak tisku čeká,

za znakem znak jak Sisyfos pod tíhou touhy heká.
Jak trápí ji ty škrtance, změněný smysl, vytržený list.
A stále ptá se jen: Jak a kdy a kde já odpověď mám číst?

IV

Člověče, věz, že slovo živá bytost je,
i ruka autora se nad tím zachvěje.
Než v nekonečnu se ztratí stránek řada,
na každém kroku léčka, temno, zrada!
Co dělat, když větu ukončiti chceš,
prásknout tam zavináč, anebo radši backslash?
Snad manuál ti poví, jak tyhle trable zdolat,
než získáš rutinu, musíš jej prostudovat.

V

Když končí práci, poslední slovo píše,
zaklíná dívka \TeX , volá jej, sotva dýše.
On zjeví se, šrotí a kalkuluje,
ještě dvě závorky a dolar vyžaduje.
Všechno se mění — obskurní, vrtkavá hra,
rej čar a křivek s ploutvemi ze stříbra.
Ohňostroj frází, miš-maš vět nesmírný,
všude se valí slova jako prach vesmírný.

VI

Čaroděj \TeX už dokouzlil a ustal
a v ruce dívky zářivý dýmant zůstal.
Pokorná tiskárna jí zvolna poklad vysází,
jak slunce jasné když černý tisk vychází.
Toť čistá pravda, co nás teď oslňuje:
Shakespeare snad dojímá, Euler však triumfuje!

Victor Hugo
p.c.c. Bernard Beauzamy
český překlad *Anna Fučíková a Arnold Santus*

Obsah prvních ročníků TUGboatu

Díky péči Olina Ulrycha se podařilo získat v elektronické podobě obsahy dnes už historických čísel TUGboatu. Budeme se snažit o jejich postupné otiskování, protože všechna čísla by měla být dostupná v Praze, Brně i Bratislavě.

TUGBOAT 1 (1) October 1980

	2	Editor's Comments / <i>Robert Welland</i>
General Delivery	3	Message from the Chairman / <i>Richard Palais</i>
	7	Publishing & T _E X / <i>Ellen Swanson</i>
	10	AMS-T _E X—"A very friendly product" / <i>Michael Spivak</i>
	12	Minutes of the first TUG meeting, February 1980 / <i>Robert Morris</i>
Interface Software	16	The status of the Pascal implementation of T _E X / <i>Ignacio Zabala and Luis Trabb-Pardo</i>
	17	The format of T _E X's DVI files / <i>David Fuchs</i>
Site Reports	19	University of Minnesota CDC Cyber site report / <i>Thea Hodge</i>
Warnings & Limitations	20	Troubles with trace and Other oddities / <i>Barbara Beeton</i>
Macros	20	
Appendix A		An indexing facility for T _E X / <i>Terry Winograd</i> and <i>Bill Paxton</i>
Questions & Answers	21	
Letters	22	<i>Gérard Emch</i> and <i>Arnold Pizer</i>
Miscellaneous		Order form for <i>The Joy of T_EX</i>
Supplements		T _E X Errata TUG mailing list

	2	Addresses of authors	
General Delivery	3	Site Coordinators / <i>Robert Welland</i>	
	3	Chairman's report / <i>Richard Palais</i>	
	5	Report on the January 1981 TUG Steering Committee meeting / <i>Robert Morris</i>	
	6	1980 TUG Treasurer's report / <i>Samuel B. Whidden</i>	
	6	Informal TUG session / <i>Robert Morris</i>	
	7	A position on T _E X maintenance / <i>Robert Morris</i>	
	9	T _E X support / <i>Samuel B. Whidden</i>	
	10	Update on Pascal / <i>Scott Kim</i>	
	Interface Software	11	Erratum: The format of T _E X's DVI files / <i>David Fuchs</i>
		11	T _E X-Pascal and Pascal compilers (a status report) / <i>Ignacio Zabala</i>
12		T _E X Font Metric files / <i>David Fuchs</i>	
17		T _E X support programs / <i>Phil Sherrod</i> and <i>Alan Wright</i>	
20		Brief functional characterization of the procedures in the T _E X/Pascal compilation unit, SYSDEP / <i>C. L. Lawson, I. Zabala</i> and <i>M. Díaz</i>	
32		Detailed specifications of procedures in the T _E X/Pascal compilation unit, SYSDEP / <i>C. L. Lawson, I. Zabala</i> and <i>M. Díaz</i>	
48		Output devices: A new column / <i>Barry Doherty</i>	
Site Reports			
	· IBM Group	48	Status of T _E X on the Stanford 370/3033 systems / <i>Eagle Berns</i>
	· DEC 20	48	AMS site report / <i>Barry Doherty</i> and <i>Barbara Beeton</i>
	· VAX/UNIX	49	The status of VAX/T _E X at Brown / <i>Janet Incerpi</i>
	· CDC Cyber	51	Report from the North Star, or, T _E X at the University of Minnesota / <i>T. D. Hodge</i>
	· Univac 1100	51	T _E X is available for Univac 1100 systems / <i>Ralph Stromquist</i>

	· Nord 100	51	Report on the use of T _E X at Computas A/S, Norway; Nord 100 computer: 16 bit “big mini” / <i>Helge Totland</i>
Warnings & Limitations		53	Disappearing digits; Undisciplined uppercase / <i>Barbara Beeton</i>
Macros		53	How to prepare a file for publication in <i>TUGboat</i> / <i>Barbara Beeton</i>
		55	T _E X macro package / <i>Max Díaz</i>
		56	Anatomy of a T _E X macro package / <i>Arthur M. Keller</i>
		87	NOFILL program with Pascal source / <i>Lynne A. Price and Patrick Milligan</i>
		98	List macros / <i>Lynne A. Price</i>
		111	Table of contents macros / <i>Lynne A. Price</i>
		119	Utility macros / <i>Patrick Milligan and Lynne A. Price</i>
		122	Hebrew letter (with source) / <i>Lynne A. Price</i>
		125	Two slides / <i>Lynne A. Price</i>
Letters		127	Report from an early $\mathcal{A}\mathcal{M}\mathcal{S}$ -T _E X user / <i>David Eck</i>
		127	<i>Graeme Hirst</i>
Miscellaneous		128	Appendix A. Output samples from a paper by David Eck
Supplements			T _E X Errata
			TUG membership list

TUGBOAT 2 (2) July 1981

	2	Addresses of officers, authors and others
	3	Official announcements
General Delivery	3	Editor’s remarks / <i>Robert Welland</i>
	3	Chairman’s report / <i>Michael Spivak</i>
	4	Report on the May 1981 TUG Steering Committee meeting / <i>Robert Morris</i>
	5	TUG Treasurer’s report / <i>Samuel B. Whidden</i>
	5	Proposal for institutional support of TUG / <i>Robert Morris</i>
	6	Report: T _E X Implementors’ Workshop, Stanford, 14–15 May 1981 / <i>Barry Doherty</i>

	8	Workshop attendees
	8	Preliminary announcement: TUG meeting, Cincinnati, January 1982 / <i>Tom Pierce</i>
	9	Ask not what TUG can do for you, ask what you can do for TUG! / <i>Patrick Milligan</i>
	10	A proposal for a machine independent tape interchange standard / <i>Patrick Milligan</i>
Software	12	The format of T _E X's DVI files, Version I / <i>David Fuchs</i>
	16	Some feedback from PTEX installations / <i>Ignacio Zabala</i>
	19	T _E X and hyphenation / <i>Frank M. Liang</i>
	21	T _E X on small machines / <i>Kent S. Harris</i> and <i>Robert M. McClure</i>
Output Devices	25	Output device news flash (APS-5 and Linotron 202) / <i>David Fuchs</i>
	25	Summary of computing equipment and output devices
	26	Imagen (Canon LBP-10) / <i>Luis Trabb-Pardo</i>
Site Reports		
· CDC Cyber	28	T _E X under the North Star / <i>Thea Hodge</i> and <i>Michael Frisch</i>
· DEC 10 & 20	28	DECSys _{tem} -10/20 Implementation Workshop announcement / <i>Phil Sherrod</i>
	29	AMS site report / <i>Barbara Beeton</i>
	29	T _E X at the 1981 Spring DECUS U.S. Symposium / <i>Patrick Milligan</i>
	29	T _E X at NIH / <i>Rachel Schwab</i>
· Univac 1100	30	An implementation report for the Univac 1100 / <i>Bill Kelly</i>
· VAX/VMS	33	Oregon Software implementation for VAX/VMS / <i>Monte Nichols</i> and <i>Barry Smith</i>
Fonts	34	TUG Font Committee / <i>Barry C. W. Doherty</i>
	35	The status of at OCLC / <i>Thomas B. Hickey</i>
	39	Font development at the A _M S / <i>William J. LeVeque</i>
	40	TPHON / <i>Ronald Whitney</i>
	40	Proofmode and magnification / <i>Barry Doherty</i> and <i>Ronald Whitney</i>
Warnings & Limitations	42	Uppercase update; Fickle fonts / <i>Barbara Beeton</i>
Macros	43	Greetings from the Editor / <i>Lynne Price</i>
	43	Macros on microfiche

	43	Erratum: NOFILL program / <i>Patrick Milligan</i>
	44	\DefineFont / <i>Patrick Milligan</i>
	46	A macro menagerie / <i>Brendan McKay</i>
	50	Macro madness / <i>Michael Spivak</i>
Appendix A		Fácil T _E X / <i>Max Díaz</i>
Problems	54	Greetings from the Editor / <i>Lynne Price</i>
	54	Problems from the T _E Xarcana course
	57	Balancing columns of text and translation / <i>Johnny Stovall</i>
	57	Input-dependent macro redefinition / <i>Michael Spivak</i>
Letters	57	<i>Lynne Price</i>
Dreamboat	58	<i>Lynne Price</i>
Miscellaneous		Instructions for submitting TUGboat articles on magnetic tape
		Membership application and order form
Supplements		T _E X and Errata; TUG membership list

TUGBOAT 2 (3) November 1981

	2	Addresses of officers, authors and others
	3	Official announcements
General Delivery	3	Message from the Chairman / <i>Michael Spivak</i>
	4	TUG Treasurer's report / <i>Samuel B. Whidden</i>
	5	The current state of things / <i>Donald Knuth</i>
	7	T _E X Users Group Winter 1982 meeting, January 11–12, 1982, Cincinnati, Ohio / <i>Tom Pierce</i>
Software	7	Pascal-coded T _E X errata / <i>Arthur Samuel</i>
	8	The format of PXL files / <i>David Fuchs</i>
Output Devices	13	Sample of output from an IDS-640 / <i>John Sauter</i>
	14	Output device index / <i>Rilla Thedford</i>
	14	A Varian output driver in VAX/VMS Fortran / <i>Jim Mooney</i>
	15	Diabolic T _E X / <i>Timothy Murphy</i>
Site Reports	21	News from the home front (Stanford) / <i>David Fuchs</i>
· CDC Cyber	22	T _E X under the North Star / <i>Michael Frisch</i>

· DEC 20	23	A <i>TUGboat</i> tour: Excerpts from the $\text{T}_{\text{E}}\text{X}$ nician's log / <i>Barbara Beeton</i>
· HP 3000	25	$\text{T}_{\text{E}}\text{X}$ for the HP3000 / <i>Lance Carnes</i>
· IBM Group	26	$\text{T}_{\text{E}}\text{X}$ for the IBM 370 / <i>Susan Plass</i>
	27	$\text{T}_{\text{E}}\text{X}$ in Israel / <i>Jeffrey S. Rosenschein</i>
	28	$\text{T}_{\text{E}}\text{X}$ at the University of Michigan, summary of progress / <i>David Rodgers</i>
· VAX/UNIX	29	VAX on UNIX / <i>Robert Morris</i>
	29	$\text{T}_{\text{E}}\text{X}$ at CalTech / <i>Calvin Jackson</i>
	32	The status of VAX/ $\text{T}_{\text{E}}\text{X}$ at Brown / <i>Janet Incerpi</i>
· VAX/VMS	32	VAX/VMS site report / <i>Monte C. Nichols</i>
	32	Enhancements of VAX/VMS $\text{T}_{\text{E}}\text{X}$ at Calma / <i>John Blair</i>
	34	“Poor man’s” $\text{T}_{\text{E}}\text{X}$ / <i>John Sauter</i>
Fonts	35	Font update / <i>Ronald Whitney</i>
Warnings & Limitations	35	Don't just <code>\let</code> $\text{T}_{\text{E}}\text{X}$ hang, <code>\raise</code> or <code>\lower</code> it / <i>Barbara Beeton</i>
Macros	36	Editor's introduction / <i>Lynne Price</i>
	36	<i>TUGboat</i> macro index
	37	Bubbles: A $\text{T}_{\text{E}}\text{X}$ tension in search of a $\text{T}_{\text{E}}\text{X}$ pert / <i>Timothy Murphy</i>
	39	Charting your grammar with $\text{T}_{\text{E}}\text{X}$ / <i>Michael F. Plass</i>
	57	Chemical notation using $\text{T}_{\text{E}}\text{X}$ / <i>Monte Nichols, Barbara Beeton</i>
Problems	58	<i>Lynne Price</i> Balancing columns of text and translation (<i>Johnny Stovall</i>) Input-dependent macro redefinition (<i>Mike Spivak</i>) A macro that prints its name (<i>Anthony Siegman</i>) Testing the equivalence of strings
	61	Problems from the $\text{T}_{\text{E}}\text{X}$ arcana class: Answers, and another problem
Letters et alia	66	<i>J. M. Cole</i>
	66	Formatting a book with $\text{T}_{\text{E}}\text{X}$: Experiences and observations / <i>Michael Sannella</i>

Miscellaneous	Instructions for submitting <i>TUGboat</i> articles on magnetic tape
	Membership application and order form
Supplements	TeX and Errata
	TUG membership list

TUGBOAT 3 (1) March 1982

	2	Addresses of officers, authors and others
	3	Official announcements
General Delivery	3	Statement of editorial policy / <i>Barbara Beeton</i>
	4	Report on the January 1981 TUG Steering Committee meeting / <i>Robert Morris</i>
		TUG Winter meeting, January 11–12, 1982, Cincinnati, Ohio
	5	Program
	6	Attendees
	7	Open letter to TUG / <i>Robert McGaffey</i> and <i>Keith Penny</i>
	10	Report on the ANSI X3J6 meeting / <i>Lynne A. Price</i>
Software	10	Fixed-point glue setting—an example of WEB / <i>Donald Knuth</i>
Site Reports		
· CDC Cyber	28	CDC TeX / <i>Michael J. Frisch</i>
· HP 3000	28	HP3000 site report / <i>Lance Carnes</i>
· IBM Group	29	TeX at the University of Milan / <i>G. Canzii</i> , <i>D. Lucarella</i> and <i>A. Pilenga</i>
	30	TeX news from Pisa / <i>L. Aiello</i> and <i>S. Pavan</i>
· VAX/VMS	32	VAX/VMS site report / <i>Monte C. Nichols</i>
“small” TeX	32	Editor’s introduction / <i>Lance Carnes</i>
Warnings & Limitations	33	Another hangup / <i>Barbara Beeton</i>
Macros	33	Editor’s introduction / <i>Lynne Price</i>
	33	<i>TUGboat</i> macro index
	35	Display of a font in table form / <i>Roger L. Beeman</i>

	39	Seating charts / <i>Roger L. Beeman</i>
Problems	43	Problems from the T _E Xarcana class: Answer to the challenge problem
Late-Breaking News		TUG financial reports / <i>Samuel B. Whidden</i>
	45	TUG Treasurer's report
	46	1982 TUG Budget
Miscellaneous		Instructions for submitting <i>TUGboat</i> articles on magnetic tape
		Membership application and order form
Supplements		T _E X and Errata
		TUG membership list

Z obsahu příštího čísla

Oldřich Ulrych: Jak jsem se sázel svisle

Vydalo: Československé sdružení uživatelů T_EXu
vlastním nákladem jako interní publikaci
Počet výtisků: 600
Tisk: HBT-Jet PRESS Neratovice
Adresa: ČG TUG MÚ UK, Sokolovská 83, 186 00 Praha 8