

Zpravodaj

Československého
sdružení uživatelů T_EXu

TEX 2
92
bulletin

OBSAH

Jiří Veselý: Před Velkým třeskem?	49
Jana Chlebíková: Nebojme sa nových fontov	53
Oldřich Ulrych: Zkušenosti s METAFONTEM	60
Jiří Zlatuška: Záříjový EuroT _E X '92 v Praze — předběžný program	81
Další vývoj T _E Xu	84
Michal Voců: Integrované prostředí pro T _E X (zvané T _E XShell)	85
Poznámky ke koupi editoru	88
Pavel Sekanina a Jiří Rybička: Konverzní program z formátu Text602 do T _E Xu	93
Obsah TUGboatu 12 (3 & 4) 1991	95

- **PROSBA:** Veškeré platby (příspěvky, vložné na kongres, peníze za objednané publikace apod.) nám posílejte **odděleně**, protože informace o platbách ze spořitelny jsou velmi kusé a nepřesné. Od loňského roku totiž spořitelna přestala dodávat ústřížek se „zprávou od příjemce“ a jména na dodávaných výpisech jsou navíc zkomolená.
- Uzávěrky příštích čísel T_EXbulletinu: 3/92 k 7.9. 1992 a 4/92 do 30.11. 1992. Prosím: pište, překládejte, vymýšlejte, pomáhejte těm druhým. Nevíte jak? Máte zajímavé makro třeba na telefonní seznam instituce? Sem s ním! Umíte tisknout postavení pro GO? Pořídili jste si pěkný ruční scanner? Napsali jste v T_EXu svoji diplomku? O tom všem a o mnoha jiných věcech nám můžete napsat. Příspěvky pro T_EXbulletin posílejte na adresu: A. Najmanová, MÚ UK, Sokolovská 83, 186 00 Praha 8 nebo Karel Horák, MÚ ČSAV, Žitná 25, 115 67 Praha 1 (<horakk@csearn>).

Před Velkým třeskem?

Psát něco pravidelně svádí snadno ke stereotypu. Někdy to může být stereotyp působící veskrze kladně, např. držet se vždy věci, tj. v našem případě $\text{T}_{\text{E}}\text{X}$ u, nebo nevyhýbat se žádným problémům (např. nákup softwaru), případně snažit se brát vše s humorem. Jsou však i stereotypy značně škodlivé. Patří k nim nicnedělání, kterým se, pravda, nikdy nic nepokazí, předcházení nepříjemným věcem zasouváním hlavy do písku i aplikování politiky se vším vsudy tam, kde to je možné, ale ne žádoucí, atp. Okolnosti mne nutí psát i o věcech velmi nepříjemných a — jak stále věřím — $\text{T}_{\text{E}}\text{X}$ u bytostně cizích.

Jsme jednou z organizací, které mají „federální“ charakter; přidružujeme se stále zdánlivě již nemoderního jména. Staráme se o $\text{T}_{\text{E}}\text{X}$, programový produkt americké provenience, který se díky své filosofii i svému poslání stal mezinárodním standardem. Do jisté míry nás to vede k internacionalismu v tom dobrém slova smyslu. Na druhé straně to nevylučuje možnost vzniku jazykově či národně orientovaných uživatelských skupin. Příklady nalezneme snadno: existují jak jazykově orientované skupiny (DANTE působí v Německu, v Rakousku i ve Švýcarsku a podstatou členství je užívání $\text{T}_{\text{E}}\text{X}$ u a němčiny), tak územně laděné skupiny (Nordic Group je určen především pro uživatele ze Skandinávie). Dobře se též snáší existence TUGboatu s existencí $\text{T}_{\text{E}}\text{X}$ line v Anglii, byť jde o časopisy, které jsou oba určeny pro anglicky mluvící uživatele — koexistují, prezentují někdy tytéž a někdy naprosto odlišné názory, někdy zas dokonce i tytéž články. Existují však i speciální časopisy — pro německé a pro francouzské uživatele. Běžnou praxí je pak překládání článků; práva jsou většinou v rukách autorů, ne časopisů. V $\text{T}_{\text{E}}\text{X}$ ové oblasti vedle sebe koexistují komerční produkty i software, který je veřejně přístupný (public domain) nebo je typu „shareware“. Již i u nás existují někteří uživatelé, kteří s $\text{T}_{\text{E}}\text{X}$ em pracují nejen na IBM-kompatibilních osobních počítačích, ale i na Atari, na Amigách či na pracovních stanicích (a pochopitelně nejen pod operačním systémem DOS).

U nás vydáváme jeden dvojjazyčný časopis pro $\text{T}_{\text{E}}\text{X}$ (dosud jsme neotiskli žádné příspěvky v angličtině, francouzštině nebo němčině, nepočítáme-li obsahy TUGboatů), ve kterém je poměr míšení obou jazyků dán docházejícími příspěvky. Snažíme se vzájemně si pomáhat bez ohledu na národnost, místo působení, ba i politické přesvědčení. Pomocí $\text{T}_{\text{E}}\text{X}$ u se sází zleva doprava, zprava doleva i shora dolů, v latině, s akcenty i bez, se speciálními národními znaky, v azbuce, v kurentu, v řecké abecedě či japonských znacích. $\text{T}_{\text{E}}\text{X}$ nás totiž spojuje a ne rozděluje. Říkám tyto samozřejmosti záměrně jako příklady: lze jich nalézt podstatně více, ale my je snad zatím nemusíme pracně hledat. Tím podstatným je tolerance, vůle společně cosi tvořit, ne jen bořit.

Ať už dopadnou budoucí dohody mezi politickými stranami v Čechách a na Slovensku jakkoli a my všichni budeme žít v jednom či dvou státech, svázaných do „ranečku“ s názvem federace, konfederace nebo unie, nebo vedle sebe jen tak, měli bychom si pomáhat dál. Naše organizace by měla stejnou měrou pečovat o českou i o slovenskou národní verzi \TeX u a měla by členům poskytovat za těchže podmínek tytéž výhody. Snad by zatím ani nemusela měnit svůj název. Zase asi pro mnohé říkám samozřejmost, cítím ale, že je potřeba ji říci.

Neobávám se o členy ζ TUGu. Konec konců o všem patrně rozhodne referendum. Sám chci mít nadále své slovenské přátele a nacházet s nimi radost ze společných zájmů; věřím, že i oni chtějí totéž. V generalizování se skrývá velké nebezpečí. Nemožnost shromáždit kompletní informace o všem, co se nás týká, je zřejmá. Proto se náš obraz o situaci vždy podobá dětské skládačce, kterou se snažíme pochopit, ač se většina jejích kousků kdesi poztrácela. Navíc mohou přijít těžší chvíle, kdy se informací bude jaksi ještě více nedostávat a prostředí začne vykonávat své. Pak bude těžší nezapomenout a nezanevít na přátele. Tam také bude začínat zkouška, která ukáže, zda nás pojí pouze \TeX , nebo i o malinko více toho, k čemu nás jeho koncepce využívání přirozeně vedla a vede.

Pokud by přece jen došlo na dělení naší organizace, jistě jde o věc technicky, finančně i jinak zvládnutelnou. Jde o věc domluvitelnou a možnou. Jak se ale snažím naznačit, jde o věc zbytečnou, neúčelnou a do jisté míry i nežádoucí a škodlivou. Opět opakuji, co jsem patrně na tomto místě již jednou uvedl: dělat něco jen proto, že je to možné, bývá jen zřídka dobré.

Dost však politiky, přejdeme k jinému soudku: nadpis může předznamenávat i záležitosti s Euro \TeX em '92. Zatím běží přípravy vcelku dobře (nic jsme snad „neprošvihli“), připravený program je podle mého názoru hodnotný a zajímavý. Nemůžeme v bohatosti programu soupeřit s výroční konferencí TUGu v Portlandu, i když někteří přednášející se vyskytují na čestných místech obou seznamů. Na druhé straně není i pro dobře situovaného Evropana cesta do USA a náklady s účastí na konferenci zanedbatelným ohnosem a v tomto světle spolu s atraktivitou Prahy by měla být účast na Euro \TeX u velmi přitažlivá. Ostatně posuďte program sami, stačí si přečíst článek z pera Jiřího Zlatušky. (Zde jen malá odbočka náhradou za další úvahy, které jsem se na radu přátel rozhodl přenechat politikům. Protože zákon přirozené škodolibosti věcí je jedním z těch, které fungují dosti spolehlivě, bylo ve dnech dohodovování tohoto \TeX bulletinu spojení s Brnem částečně paralyzováno (také „velkým třeskem“: v tomto případě za to mohl blesk, bylo to tedy opět z vyšší moci); navíc byl náš programový šéf Jiří Zlatuška povolán k vojenským radovánkám, a tak situace s jeho příspěvkem nebyla růžová. Všechno však stihl, i když to neměl jednoduché.)

Z technických důvodů jsme se při poslední výborové schůzi sešli pouze v malém kroužku, nechtěli jsme proto přijímat žádná závazná rozhodnutí. Také probíhající anketa vyhlášená v prvním letošním \TeX bulletinu dávala jen nereprezentativní vzorek názorů. Dohodli jsme se proto jen předběžně na tom, že přikoupíme anglický korektor — tento v současné době testujeme. Kromě členů výboru úzce svázaných s redakcí \TeX bulletinu a kolegy J. Rybičky z Brna nám nikdo nedodal své uživatelské slovníky, které by posloužily ke zlepšení práce se zakoupeným korektorem. Od Barbary Beeton jsme získali soubory s řídicími slovy, které by měly vést k vytvoření uživatelských slovníků vhodných i pro korekci „ \TeX ové části“ korigovaného souboru. Originální zaslané soubory poskytneme rádi komukoli k experimentům, dosud však jsme nedošli k dohodě o filozofii tvorby pomocného slovníku pro tento účel.

Omylem nám z minulého čísla vypadl odpovědní lístek týkající se zájmu o překlad Spivakova $\mathcal{L}_{\mathcal{M}}\mathcal{S}\text{-}\text{\TeX}$ U. Patrně to bylo výjimečně šťastné opomenutí: Mirek Dont projevil přání poněkud modifikovat svoji nabídku překladu rozsáhlejšího textu. Zdá se, že bude velmi potřebné mít k dispozici co nejdříve manuál k nové verzi \LaTeX U (viz přednáška na konferenci). I když jde o materiál, který dosud **není** k dispozici, o jeho relativně velkém rozsahu a obrovské užitečnosti i pro členy ζTUG u nepochybuji. A právě tento materiál by pro nás byl Mirek Dont ochoten přeložit. Doufáme, že se podaří celý projekt dohodnout při Euro \TeX U v září v Praze.

K problematice zakoupení editoru se vracíme ve dvou příspěvcích; byly psány formou dopisu, nepovažujeme za nutné je však podstatně upravovat. Nákupem se budeme zabývat na nejbližší schůzi výboru. Ta je plánována na 7. července (jiný „velký třesk“, který způsobil její odložení o týden, mají na svědomí Spojené státy), to však toto číslo \TeX bulletinu bude již v tisku. Celkem došly do uzávěrky tohoto čísla 33 odpovědi a jeden dopis, který má charakter účasti v hlasování. Výsledek je následující: pro koupi ζSED u se vyslovilo 7 individuálních členů a 2 kolektivní (za 7 a 25 uživatelů). Další 23 individuálních členů doporučuje posečkat (koupě není tak naléhavá), totéž vyjadřuje 1 kolektivní člen (15 uživatelů). Konečně 1 člen v této věci neuveld žádnou preferenci. Jeden zástupce kolektivního člena (za 70 uživatelů) upozornil na nutnost řešení, přičemž navrhl porovnat ζSED s výhodami řešení pomocí \TeX shellu (viz příspěvek kolegy Voců). Konečně dva další obsáhlé příspěvky (od kolegů Demela a Marvana) jsme se rozhodli otisknout. Konstatuji nedostatek zájmu členů na věci, která by měla být pro všechny důležitá (viz následující informace o počtu členů). Výbor se pokusí sjednotit na nějakém rozumném stanovisku, budu navrhovat problém definitivně vyřešit při valném shromáždění v září: každý bude vědět, že se bude tato věc projednávat a při neúčasti bude moci příspěvek do diskuse poslat písemně (jeho stanovisko bude zveřejněno při jednání, ale nebude hlasovat). Tam již musíme dospět k řešení v každém případě, i kdyby

účast nebyla veliká. (Celý problém se táhne jako předvolební opakovaná jednání o státoprávním uspořádání.)

Pro zajímavost: k 20.6. má ζ STUG celkem 29 kolektivních členů (pouze jeden ubyl zánikem organizace a byl „řádně“ odhlášen), avšak 4 z těchto členů dluží stále příspěvek. Máme též 201 individuálních členů, ale 18 „loňských“ nám stále dluží příspěvek za tento rok. Toto je **poslední** číslo, které jim zasíláme, nezaplatí-li do září, vyjmeme je z evidence a $\text{T}_{\text{E}}\text{X}$ bulletin jim přestaneme posílat.

Zmiňuji to proto, že získání 35 odpovědí na anketu o editorech nesvědčí o velké přítulnosti našich členů a (eventuálních?) čtenářů $\text{T}_{\text{E}}\text{X}$ bulletinu k organizaci.

I když jsme plánovali celé toto číslo tematicky zaměřit k užívání METAFONTu a k problémům s tím úzce souvisejícím, je tento záměr realizován jen zčásti. Přesto věřím, že materiál z pera Oldřicha Ulrycha poskytne mnohým z vás další kratochvíle u počítače. Mimochodem, „Olin“ je zpět ze Španělska, a tak se snad brzo můžeme těšit na jeho knížku o $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}_{\text{E}}\text{X}$ u; chtěli bychom ji vytisknout během prázdnin.

Co říci nakonec ? Rád bych Vám všem popřál hezké prázdniny a vyslovil ještě jedno přání parafrázující písničku, která se nyní často hraje: Na shledanou (v lepších časech) na Euro $\text{T}_{\text{E}}\text{X}$ u '92!

Jiří Veselý

<jvesely@cspguk11>

- Na jiném místě uveřejňujeme výzvu k vytvoření koncepce pro další vývoj $\text{T}_{\text{E}}\text{X}$ u od J.L. Modernizace však probíhá tak říkajíc za pochodu: mnozí z vás již používají nový $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}_{\text{E}}\text{X}$, jiní se těší na nový $\text{L}\text{A}\text{T}_{\text{E}}\text{X}$. K vytvoření $\text{L}\text{A}\text{T}_{\text{E}}\text{X}$ u se vytváří $\text{L}\text{A}\text{T}_{\text{E}}\text{X}3$ Fund, do kterého se zapojují i naše partnerské organizace. Cílem projektu $\text{L}\text{A}\text{T}_{\text{E}}\text{X}3$, jehož kořeny sahají ke stanfordské konferenci TUG v r. 1989, je nejen vytvořit dobrý interface pro používání $\text{T}_{\text{E}}\text{X}$ u, ale i nebyvalé rozšíření možností: nové styly (style files) by měly být velmi flexibilní, měly by vzniknout i další pro zpracování dokumentů nových typů. Jejich další udržování a vývoj by měly být snadné . . .
- V posledním čísle anglického partnerského časopisu $\text{T}_{\text{E}}\text{X}$ line (14) se objevil velice zajímavý příspěvek Joachima Schroda (jde o článek, který vznikl dalším přepracováním autorových článků, které byly otištěny v „Baskervillu“ a v „Die $\text{T}_{\text{E}}\text{X}$ nische Komödie“). Jmenuje se **Součásti $\text{T}_{\text{E}}\text{X}$ u** (The Components of $\text{T}_{\text{E}}\text{X}$). Najde se dobrovolník (dobrovolnice), který by tento velmi pěkný materiál přeložil do češtiny nebo slovenštiny k otištění v $\text{T}_{\text{E}}\text{X}$ bulletinu? Prosím, kontaktujte J. Veselého (tel. (02) 231 60 00, l. 211, večer (02) 73 03 66, email [jvesely at cspguk11.bitnet](mailto:jvesely@cspguk11.bitnet)). Děkuji.

Nebojme sa nových fontov

Tento článok je mienený ako pomôcka užívateľom T_EXu pri používaní znakov z nových a neštandardných fontov. Za jazyk výkladu bol vybraný plainT_EX, pretože ho považujem za „pružnejší“. Všetkým stupenom L^AT_EXu sa preto ospravedlňujem.

Predpokladajme, že do svojej zbierky fontov sme získali nový prírastok a chceme zistiť, aké znaky obsahuje a ako ich môžeme používať. Pre „prezretie“ fontu nám stačí napr. nasledujúci program skompilovať pod plainT_EXom (počas behu programu odpovedať na otázky počítača) a dostávame tabuľku žiadaného fontu (viď príklad pre font *csr10*). Uvedený program je určený pre 8-bitové fonty,¹⁾ preto sa pri niektorých fontoch (napr. 7-bitových) môže objaviť veľa neobsadených pozícií.

```
\font\zoa=cmb10 scaled \magstep3
\nopagenumbers
\newcount\p\p=0\newcount\ps\ps=0
\newdimen\w\newcount\pr\pr=0\newcount\kon\kon=9
\def\zn{&&\za\char\p\global\advance\p by1
  \global\advance\ps by1}
\def\przn{&&\global\advance\ps by 1}
\def\riadok{\ifnum\ps<\kon \zn
  \riadok\else \ifnum\ps<9 \przn\riadok
  \else&\cr\fi\fi}
\def\riad{&\rm\the\pr\riadok}
\def\ri{\riad\noalign{\hrule}
  \global\ps=0\global\advance\pr by 1}
\def\tabulka{\ifnum\pr<24 \ri\tabulka\fi}
\def\rtt#1#2#3{\w=#3mm\multiply\w by11
  \font\za=#1 \centerline{
  \vbox{\hbox to \w{\hfil\zoa Font #1.\hfil}}
```

¹⁾ Dále uvedená tabulka se v rozmístění znaků ve své dolní polovině podstatně liší od té, kterou dostala při kompilaci autorka tohoto příspěvku, protože používá kódování KOI8, zatímco my se pořád ještě držíme kódování bratří Kamenických. Také se zatím vyhýbáme přímému použití osmibitových (Novákových) fontů. (Pozn. red.)

```

\vskip0.4cm
\offinterlineskip
\halign{\vrule###\vbox to #2mm{\vfil
\hbox to #3mm{\hfil##\hfil}\vfil}
&\vrule width0.05cm###&\vbox to #2mm{\vfil
\hbox to #3mm{\strut\hfil##\hfil}
\vfil}&\vrule##\cr
\noalign{\hrule}
&&&0&&1&&2&&3&&4&&5&&6&&7&&8&&9&\cr
\noalign{\hrule height 0.05cm}
\tabulka\cr
\global\kon=5\riad
\noalign{\hrule}}}}}}
\message{Zadaj nazov fonu, ktory chces zobrazit:
(napr. csr10)}
\readl6 to\naz
\message{Zadaj vysku stvorceka (v mm), napr. 7}
\readl6 to \vys
\message{Zadaj sirku stvorceka (v mm), napr. 9}
\readl6 to \sir
\rtt{\naz}{\vys}{\sir}
\bye

```

Ak chceme použiť nejaký znak z nového fonu, musíme najskôr tento font načítať. To sa deje príkazom

```

\font<pomenovanie fonu>=<názov fonu>,

```

prípadne

```

\font<pomenovanie fonu>=<názov fonu>
scaled \magstep<0...5, half>.

```

Pre načítanie napr. azbuky stačí teda napísať `\font\tenazb=wncyr10`.

Znaky v textovom reime.

Pokiaľ chceme znaky z príslušného fonu používať len v textovom režime, vystačíme s horeuvedeným. Po prepnutí do nového fonu (čo sa deje príkazom `\<pomenovanie fonu>`, v predchádzajúcom príklade teda `\tenazb`) sú nám jednotlivé znaky fonu prístupné niekoľkými spôsobmi:

- Znaky (takmer všetky) poradového čísla 33 až 127 (podľa tabuľky fonu) sú prístupné cez klávesy odpovedajúceho čísla podľa štandardnej ASCII tabuľky. Teda napr. `{\tenazb W}` dáva III, pretože pod

Font **csr10**.

	0	1	2	3	4	5	6	7	8	9
0	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	Φ	Ψ
1	Ω	ff	fi	fl	ffi	ffl	ı	ı	`	´
2	˘	˘	˘	˘	˘	ß	æ	œ	ø	Æ
3	Œ	Ø	-	!	”	#	\$	%	&	’
4	()	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	i	=	ı	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[“]	^	·	‘	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	-	—	”	~	¨		
13										
14		‰								
15			À				-	ı	«	»
16						Ĺ				Š
17		Ť			Ž					
18		Ŧ			à	š		ť		
19	ž		Ř	Á			Ä	Ĺ		
20	Č	É			Ě	Í		Ď		
21	Ň	Ó	Ô		Ö		Ř	Û	Ú	
22	Û	Ý			í	á			ä	Í
23			č	é			ě	í		ď
24			ň	ó	ô		ö		ř	ů
25	ú		ü	ý	„	“				

číslo písmena W podľa ASCII je vo fonte `\tenazb` znak III. V niektorých fontoch existujú výnimky pre špeciálne využitie niektorých

znakov. Napr. vo fonte *cmr10* znak číslo 37 (podľa ASCII) odpovedá %. Tento znak však nie je prístupný len jednoduchým napísaním %.

- Všetky znaky fontu sú prístupné skrz

```
\char<čís.riadku><čís.stĺpca>
```

podľa získanej tabuľky. Použitím `{\tenazb\char87}` dostávame III. Existuje tiež príkaz

```
\chardef\<názov>=<číslo>,
```

ktorý je skratkou pre

```
\def\<názov>={\char<číslo>}.
```

Použitie `{\tenazb\chardef\nom=87\nom}` dáva teda III.

Namiesto desiatkovej sústavy udávania čísel je možné použiť tiež osmičkovú alebo šestnástkovú. Teda `{\tenazb\char87}`, `\char'127`, `\char"57}` dáva vždy rovnaký znak III. V takomto prípade je vhodné pretransformovať tabuľku, aby zadávanie odpovedalo číslu riadku i stĺpca; viď nasledujúci príklad pre font *wncyr10*.

- Niektoré znaky sú prístupné ako ligatúry, čo spočíva v automatickom spájaní dvoch i viac vstupných znakov do jedného výstupného. Napr. `{\tenazb Sh}` dáva len jeden znak III, ale už `{\tenazb S{h}}` dáva Cx. Nemusíme snáď zdôrazniť, ako nám znalosť ligatúr môže uľahčiť písanie, a naopak ich neznalosť „spotvoríť“ dokument. Obvykle pre obyčajného užívateľa býva problém zistiť, ktoré kombinácie znakov vytvárajú v danom fonte ligatúry. (V \TeX u sú v latinke prístupné štandardné ligatúry, k ligatúram v azbukových fontoch *wncyr*, ... sa dostaneme neskôr.)

Znaky v matematickom režime.

Pokiaľ chceme nejaký znak používať i v matematickom režime a zabezpečiť, aby sa veľkosť znaku správne menila v indexoch (exponentoch), s vyššie uvedeným nevystačíme. Okrem iného treba zadať predpisy, ako sa má meniť veľkosť znaku v exponentoch a indexoch. Najjednoduchšie bude vysvetlenie na príklade:

```
\font\tenazb=wncyr10           % pomenovanie fontov
\font\sevenazb=wncyr7
\font\fiveazb=wncyr5
\newfam\azbfam                 % nzov skupiny fontov
\textfont\azbfam=\tenazb      % predpisujeme,
                               % ak font sa bude pouva
                               % pre zkladn vekos v matem. mde
\scriptfont=\azbfam=\sevenazb % predpisujeme font na 1.rove
                               % indexov a exponentov
```

Font wncyr10.

	'0	'1	'2	'3	'4	'5	'6	'7	
'00	Ѓ	Ѕ	Ц	Э	І	Є	Ђ	Ѓ	"0
'01	ђ	ѕ	ц	э	і	є	ђ	ѓ	
'02	Ю	Ж	Й	Ё	V	Θ	S	Я	"1
'03	ю	ж	й	ё	v	θ	s	я	
'04	"	!	"	Ђ	˘	%	'	,	"2
'05	()	*	Ѓ	,	-	.	/	
'06	0	1	2	3	4	5	6	7	"3
'07	8	9	:	;	«	ı	»	?	
'10	˘	A	B	Ц	Д	E	Ф	Г	"4
'11	X	И	J	K	Л	M	H	O	
'12	П	Ч	P	C	T	У	B	Щ	"5
'13	Ш	Ы	З	["]	Ь	Ъ	
'14	‘	a	b	ц	д	e	ф	г	"6
'15	x	i	j	к	л	m	h	o	
'16	п	ч	p	c	t	y	b	щ	"7
'17	ш	ы	з	–	—	№	ь	ъ	
	"8	"9	"A	"B	"C	"D	"E	"F	

```

\scriptscriptfont=\azbfam=\fiveazb % predpisujeme font na 2.rove
% indexov a exponentov
\def\azb{\fam\azbfam\tenazb} % nzov volania fontu

```

Takýchto „skupín fontov“ môže byť použitých (v matematickom režime) najviac 16, plain \TeX nahráva na začiatku štandardne 7.

Znaky v matematickom režime sú prístupné:

- Analogicky ako v textovom režime cez svoje klávesnicové ASCII ekvivalenty, napr. $\{\text{azb net}\}\$a+b$, $\{\text{azb V}^{\sim}\{V_V\}\$$ dáva: нет $a+b$, B^{B_B}. Takto samozrejme nie sú prístupné všetky znaky.

- Cez príkaz `\mathchar`, ktorý je analógiou `\char`, ale má viac parametrov. Napr. príkaz `\mathchar"2805` špecifikuje znak z ôsmej „skupiny fontov“ (v našom prípade teda z azbuky), pozícia znaku v danom fonte je "05 (šesťnástkovo) a prvá číslica 2 udáva, že sa jedná o binárny operátor. Takáto špecifikácia je dôležitá kvôli medzerovaniu okolo symbolu a riadkovým zlomom. (Viac o tomto je možno nájsť v `TEXbooku`.) V tomto prípade sme špecifikovali znak €, ktorý sa bude správať ako binárny operátor.

Existuje tiež príkaz

```
\mathchardef\<skratka>="<čísel. charakt.>
```

ktorý je skratkou pre

```
\def\<skratka>{\mathchar"<čísel. charakt.>}
```

Napr. ak `\def\je{\mathchar"2805}` potom `\je`, `2^{\je}` dáva: €².

Použitie takýchto definícií je viazané na matematický režim.

Pokiaľ chceme v matematickom režime používať napr. len 1 znak ako premennú (ktorú nebudeme používať v indexoch a exponentoch), môžeme voliť jednoduchší postup: načítať font rovnako ako pri použití v textovom režime a znak prístupniť cez vhodné makro. Napr. ak `\def\znak{\hbox{\tenazb\char"05}}` potom použitím `\a+\znak-b` dostaneme: $a + \text{€} - b$.

Povitie azbuky zo sady fontov $\mathcal{A}\mathcal{M}\mathcal{S}$ -font 2.0.

Kolekcia týchto fontov obsahuje niekoľko druhov azbukových fontov: `wncyr*`, `wncyb*`, `wncyi*`, `wncysc*`, `wncyss*` v rôznych veľkostiach. Na základe predchádzajúcich informácií vieme už s týmito fontami pracovať v oboch režimoch, ale pozor na ligatúry!!!

Tu je ich kompletný zoznam (ak vstupný kód ligatúry začína veľkým písmenom, ďalšie písmená môžu byť malé i veľké):

Ђ	→	Dj	ђ	→	dj
Ж	→	Zh	ж	→	zh
Љ	→	Lj	љ	→	lj
Њ	→	Nj	њ	→	nj
X	→	Kh	x	→	kh
Ц	→	Ts	ц	→	ts
Ч	→	Ch	ч	→	ch
Ш	→	Sh	ш	→	sh
Ш	→	Shch	щ	→	shch

Ю	→	Yu	ю	→	yu
Я	→	Ya	я	→	ya
№	→	NO			

Ešte pár poznámok k používaniu azbuky:

- Pozor na т a с nasledujúce v slovách za sebou, napr. советский. Ak т a с od seba neoddelíme (napr. т{ }с),²⁾ dostaneme ligatúru ц a „spotvorený“ výstup советский.
- Niektoré ligatúrové znaky môžeme písať jednoduchšie jedným klávesovým znakom, podľa uloženia tohto znaku v tabulke fonu:

с ↦ ц; h ↦ x; w ↦ ш; x ↦ ш; q ↦ ч.

- Niekedy môžeme dostať na výstupe nezmyselné slová, pokiaľ sa $\text{T}_{\text{E}}\text{X}$ bude snažiť rozdeliť azbukové slovo. Napr. shch odpovedajúce ligatúre ш sa môže pri troche smoly rozdeliť na ш-ч, с-хч, alebo na 7-х. Takéto delenia treba vždy prekontrolovať a opraviť napr. pomocou \-.

So sadou \mathcal{AMS} -fontov 2.0 bol originálne distribovaný i súbor CY-RACC.DEF, ktorý obsahuje makrá uľahčujúce prístup k „neklávesovému“ znakom. Obsahuje tiež makro umožňujúce ľahko písať písmená s naznačeným „ударением“.

Toľko k základným pravidlám použitia azbuky, ako i ostatných nových fontov. I keď článok danú problematiku nevyčerpáva, je snáď dostatočným pre zvládnutie práce i s tými najšpeciálnejšími fontami.

Jana Chlebíková
chlebikj@mff.uniba.cs

- Zájem o UNIX u nás silně roste. O další vývoj UNIXU pečuje v jistém smyslu mj. **Open Software Foundation**, Cambridge, MA. Stará se ve spolupráci s firmami DEC, IBM, HP, MIT o technologickou základnu pro DEC Unix(tm). Vyvíjený unixovský OSF/1(tm) nahradí ULTRIX patrně v průběhu 1993. V té době by měl být distribuován firmou DEC systém Alfa, patrně již vybavovaný OSF/1.

²⁾ Tento postup nefunguje stoprocentně, zvláště použijeme-li české či slovenské vzory dělení. Ligatúre spoľehlivo zamezí jedine italská korekce (tedy т\{ }с). (Pozn. redakce.)

Zkušenosti s METAFONTem

Důvody pro tento příspěvek

Tento článek v jistém smyslu navazuje na článek zkušeného METAFONTisty Karla Horáka (T_{EX}bulletin 3 (1991)), a chceme se v něm podělit s laskavými čtenáři o naše zkušenosti s používáním METAFONTu pro kreslení obrázků, se začleňováním obrázků do textu v T_{EX}u, obtékání obrázků... V tomto úvodu bychom mohli pokračovat dál dost dlouho. Pokud si však chceme nalít čistého vína (a nasypat popel na hlavu), musíme říct toto:

- a) Při kreslení obrázků s použitím METAFONTu jen velice těžko hledáme syntaxi jednotlivých příkazů METAFONTu v knize METAFONTbook.
- b) Nepamatujeme si přesně syntaxi svých vlastních definic, které při kreslení obrázků používáme, a také znovu a znovu zapomínáme pracně nabyté zkušenosti při kreslení obrázků.
- c) Nepamatujeme si přesně názvy řídicích posloupností (a jejich parametry), pomocí nichž začleňujeme obrázky vytvořené v METAFONTu (a nejen v něm) do textu v T_{EX}u.
- d) Vše, co nám z výše uvedeného činí problémy, chceme mít někde (podle našeho názoru) přehledně sepsané. Důvod, proč toto uveřejňujeme v T_{EX}bulletinu, je jednoduchý. Kde najít T_{EX}bulletin víme. Kam jsme dali naše poznámky, to si nepamatujeme. Pokud naše tabulky nejběžnějších příkazů a zkušenosti shledá užitečným i někdo jiný, budeme rádi. Ještě je potřeba se zmínit o tom, že nejběžnějšími příkazy jsou pro nás ty příkazy, které jsme dosud použili. Seznam příkazů uvedeme v tabulkách na konci článku.

Celkový postup návrhu obrázku, vytváření obrázku a začlenění obrázku do textu budeme popisovat v tomto pořadí. Movitějším doporučujeme koupit si čtverečkový sešit, ve kterém si mohou své ručně kreslené návrhy obrázků schovávat, neboť po čase člověk zjistí, že ve věčném koloběhu života kreslí znovu a znovu pár obrázků. Nebo že potřebuje opakovat trik, který nedávno použil, a už si ho nepamatuje.

V každém případě doporučujeme nejdříve alespoň nahlédnout do výše citovaného článku Karla Horáka, neboť v něm je stručně popsán postup, jak s METAFONTEM začít (jinak by tento článek mohl být spíše návodem, jak s METAFONTEM skončit).

Podmínky, které nás obklopují

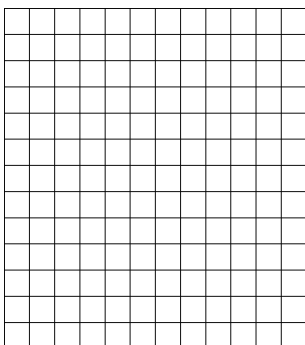
V naší praxi se vyskytují v převážné většině náčrtky, které si autor připraví jen tak od ruky na kusu papíru, a naší úlohou je tyto obrázky začlenit do matematického článku. Mnohdy se stane, že při korekturách autor řekne, že obrázek se mu zdá příliš malý, nebo příliš velký, nebo že by mohl mít jiný poměr stran. Jindy se stane, že by bylo potřeba obrázek mírně zmenšit, eventuálně zvětšit tak, aby hezky zapadal do místa, kam ho chceme umístit. Zvlášť nepříjemné jsou v tomto směru stránkové zlomy, když obrázek vychází přes něj a chybí pár milimetrů, aby se vešel tam, kde ho chceme mít.

Z tohoto vyplývá, že nejde ani o grafické zpracování výsledků, ani o kreslení složitých technických výkresů nebo o práci s přesnými rozměry.

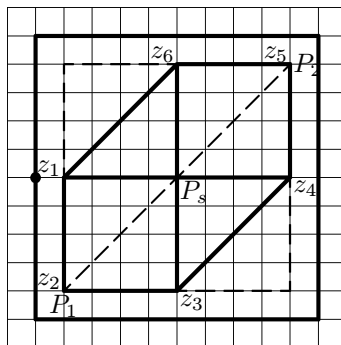
Je přirozené, že si autoři vymýšlejí různé popisky, které chtějí umístit do obrázku, a popřípadě pod obrázek napsat, co daný obrázek znázorňuje (pokud to z něj není patrné — což někdy nebývá). My bychom toto všechno chtěli dělat s co nejmenší námahou a co nejmenším úsilím. Proto jsme navrhli dva soubory maker, a to `incpic.mf` pro METAFONT a `incpic.tex` pro $\text{T}_{\text{E}}\text{X}$. Tato makra jsme navrhovali s ohledem na jejich případnou vzájemnou spolupráci. Snad jen na okraj poznamenejme, že všechny obrázky v tomto článku jsou vytvořeny a začleněny do textu s pomocí výše zmíněných dvou maker. Je velice jednoduché předat spolu s tímto článkem jedno makro pro $\text{T}_{\text{E}}\text{X}$, jedno makro a zdrojový soubor s obrázky pro METAFONT a nestarat se více o to, na jaké tiskárně bude $\text{T}_{\text{E}}\text{X}$ bulletin vytištěn. Tímto postupem nemá ani redakce téměř žádnou práci s tímto příspěvkem. Jak jsme tedy „vyráběli“ obrázky?

Jako relativně pohodlný se nám zdál tento postup, který budeme průběžně znázorňovat:

- Vezměme kus čtverečkováného papíru (takový, jaký je zobrazen na obrázku 1, a nakresleme na něj od ruky obrázek, o který nám jde. Do tohoto obrázku zanesme také všechny popis, který tam má být (např. popis souřadných os, atd.). Jednoduchý náčrtek s popisem jsme se snažili



Obr. 1 Čistý papír na začátku.



Obr. 2 Připravený náčrtek.

znázornit na obrázku 2. V následujícím textu budeme uvádět v závorkách hodnoty platné pro tento obrázek.

- Zarámujeme obrázek i s popisem do rámečku a určíme, kde bude mít tento rámeček referenční bod (tj. najdeme nějaký bod na **levém** okraji rámečku, od kterého budeme pohodlně počítat čtverečky). V obrázku 2 je tento bod vyznačen malým puntíkem. Čtverečkovaná síť totiž bude představovat jednoduchý pomocný souřadný systém. Spočítáme vodorovně počet čtverečků v rámečku (v případě obrázku 2 je to 10) a svisle počet čtverečků od referenčního bodu k hornímu okraji rámečku (5).
- Uděláme si první odhad, jak veliký by měl obrázek ve skutečnosti být. (Řekněme 45×45 mm, přičemž část nad referenčním bodem bude vysoká 22,5 mm a pod referenčním bodem hluboká také 22,5 mm). Tyto rozměry nemusí mít nic společného s velikostí náčrtku. Tyto hodnoty budeme spolu s hodnotami z předchozího odstavce zapisovat do zdrojového textu pro METAFONT.
- Vyznačíme si v obrázku pomocné body, které budeme používat pro nakreslení křivek a čar obrázku. Tyto body je možné číslovat např. pořadovými čísly (na obrázku 2 to jsou body z_i). Uděláme (kafe a) rozvalu o tom, jakým postupem obrázek nejlépe nakreslit (všímáme si různých symetrií, stejných částí lišících se pouze otočením či posunutím, atd.) Dobře provedená úvaha na začátku ušetří později trochu práce. Pokud např. kreslíme více podobných obrázků, kdy jeden lze odvodit z druhého a lze postupovat od jednoduššího ke složitějšímu,

lze k zrychlení práce zadávání obrázků použít funkci kopírování bloků v editoru nebo aparátu definic METAFONTu.

- Napíšeme zdrojový soubor pro náš obrázek (obrázky) pro METAFONT (např. `clanek.mf` — příklad takového souboru bude uveden dále) a ladíme a ladíme a ladíme... Většinou je možné mít v jednom souboru více obrázků. Postupně provádíme dva kroky. Provedeme změnu vstupního souboru a přeložíme ho METAFONTem v režimu zobrazování na obrazovce, tj. příkazem

```
mf clanek
```

Když jsme s obrázkem spokojeni (doporučujeme ladit obrázky postupně jeden za druhým), vytvoříme font např. příkazem

```
mf &plain \mode=hplaser; mag=1.; input clanek
```

(konkrétní tvar záleží na výstupním zařízení, které používáme pro zobrazování a pro tisk) a do zvláštního textového souboru (například `clanek.obr`) napíšeme zdrojový text pro \TeX , pro vysazení všech obrázků i s jejich případným popisem v obrázku a pod obrázkem (s využitím definic z makra `incpic.tex`). Tak, jak se ve fontu hromadí obrázky, hromadí se v tomto textovém souboru zápisy pro jejich vysazení. Tento soubor v \TeX u odladíme a zkontrolujeme. Jak bude později patrné z popisu makra `incpic.tex`, měly by být tyto obrázky v pomocném souboru v pořadí, v jakém se budou začleňovat do textu. Příklad tohoto souboru bude také uveden dále.

- Když jsme hotovi s obrázky, napíšeme text dokumentu (například `clanek.tex`) a začleníme do něj obrázky jednoduchými příkazy definovanými v souboru `incpic.tex`. (Pokud není obrázků „příliš mnoho“ (jsme omezeni pamětí počítače), je možné načíst soubor s obrázky jednoduše pomocí příkazu `\input clanek.obr`.)
- Text odladíme. Podle potřeby tu a tam můžeme upravit v METAFONTovém zdrojovém textu rozměry obrázků tak, aby se hodily do textu. Nic víc není potřeba měnit — pouze po změnách METAFONTového zdrojového textu generovat znovu font s obrázky.

Co obsahuje METAFONTové makro `incpic.mf`

Snaha po maximálním zjednodušení nás vedla k zavedení nových definic a k vlastnímu nastavení některých parametrů. Protože se pokusíme odhalit čtenáři i zákulisí tohoto makra, může se stát, že pro někoho může být některá věta nesrozumitelná. Pokud se vám to stane, nemusíte hned sahat po METAFONTbooku nebo odkládat T_EXbulletin, ale jednoduše takovou větu můžete přeskočit. Pravděpodobně pro vás není důležitá.

Činnost makra je následující:

- Definuje pracovní okno tak, aby obrazovka VGA (640 × 480 bodů) zobrazovala plochu asi 15 × 12 cm.
- Definuje režim práce `ouproof`, což je analogie režimu `proofmode` (tj. vytvářené znaky se zobrazují na monitoru, font se negeneruje). V režimu `ouproof` je nastaveno `showbox:=1`; `showmesh:=1`; což má za následek (pokud později není nastaveno jinak), že vytvářené znaky budou zarámovány a budou obsahovat čtverečkovanou síť.
- Pokud není explicitně zadán režim práce (např. na příkazové řádce při spouštění METAFONTu), je zvolen režim práce `ouproof` (viz předchozí bod).
- Dále makro definuje nové příkazy, které jsou uvedeny níž.
- V závěru pak toto makro nastaví `font_size 100mm#`; což znamená, že základní velikost fontu je 100 mm (tento údaj je důležitý, pokud chceme v T_EXu zavádět tento font v nějaké explicitně uvedené velikosti).
- Je iniciováno makro `mode_setup`; nastavuje další parametry.
- Je nastavena hodnota `thindestline:=0.4pt`; což je průměr pera pro kreslení nejtenčích čar.

Nyní následuje přehled dalších uživatelských definic a proměnných souboru `incpic.mf`, jejichž použití bude patrné také z příkladů uvedených dále:

- `beginch(⟨kód⟩,⟨šířka⟩,⟨výška⟩,⟨hloubka⟩,⟨x_čtv⟩,⟨y_čtv⟩)`; uvozuje písmeno, které budeme vytvářet, přičemž:

⟨kód⟩ je kód znaku, který budeme vytvářet (např. "A"), je povoleno používat pouze prvních 20 znaků abecedy, tedy písmena A–T, tj. jeden font může obsahovat nejvýše 20 obrázků;

\langle š \dot{r} řka) je rozměr, jak má být znak ve skutečnosti široký (např. 45mm# — tento údaj musí být v „ostrých“ jednotkách, tj. jednotka míry musí být následována znakem #);

\langle výška) je rozměr, jak má být znak ve skutečnosti vysoký (např. 22.5mm# — opět v „ostrých“ jednotkách);

\langle hloubka) je rozměr, jak má být znak ve skutečnosti hluboký (např. 22.5mm# — opět v „ostrých“ jednotkách);

\langle x_čtv) je vodorovný počet čtverečků v pomocném rámečku (např. 10);

\langle y_čtv) je svislý počet čtverečků v pomocném rámečku mezi referenčním bodem a horním okrajem rámečku (např. 5);

Toto makro také zajistí to, že se nastaví hodnota proměnné `current transform` tak, že souřadnice všech zadávaných bodů mohou být zadávány v bezrozměrných souřadnicích (tj. čtverečkováný papír je pro nás souřadný systém). Z toho plyne, že pokud nevíme, co děláme, raději s proměnnou `currenttransform` nepracujeme. Dále toto makro uloží do „nějakých“ proměnných `fontdimen` absolutní rozměr čtverečku sítě pro pozdější využití `TeX`em. Toto opět nemusí čtenáře příliš zajímat, protože s těmito hodnotami nebude nikdy přímo pracovat (využívá je pouze makro v `TeX`u).

- `endch`; ukončuje písmeno. Pokud je hodnota proměnné `showmesh` pozitivní, zakreslí se do znaku celá síť odpovídající čtverečkovánému papíru (hustota této sítě je dána parametry \langle x_čtv) a \langle y_čtv) v makru `beginch`). Pokud je hodnota proměnné `showbox` pozitivní, nakreslí kolem znaku rámeček.

`beginch` a `endch` znamenají také začátek a konec skupiny, tj. nastavení proměnných mezi nimi neovlivní nastavení proměnných v dalších znacích.

- `axes` \langle bod) nakreslí souřadné osy, které se budou protínat v bodě \langle bod). Toto makro bylo v podstatě převzato z článku K. Horáka, ale po mírné úpravě vyplývající z koncepce makra `incpic.mf`.

- `arrow` \langle délka), \langle úhel) je cesta pro šipku s délkou \langle délka) a směrem daným číslem \langle úhel). Co je to cesta, bude vysvětleno později. Toto je po úpravě také převzato z výše zmiňovaného článku.

- `sharparrow` je cesta ve tvaru šipky. Tato šipka je tvarově jiná, než `arrow`. Tato šipka je standardně dlouhá 3mm a ve své základní poloze směřuje vpravo (hrot je její počátek).

- `vector(<bod_1>, <bod_2>)`; nakreslí vektor (tj. úsečku s šipkou na konci) začínající v bodě `<bod_1>` a končící v bodě `<bod_2>`.
- `cerchovane(<bod_1>, <bod_2>)`; spojí body `<bod_1>`, `<bod_2>` čerchovanou **úsečkou**.
- `dashed(<bod_1>, <bod_2>)`; spojí body `<bod_1>`, `<bod_2>` čárkovanou **úsečkou**.
- `penfordots`; zvolí kruhové pero o průměru `10thinestline`. Jak již název napovídá, používáme toto pero na kreslení teček.
- `thickpen`; zvolí kruhové pero o průměru `4thinestline`. Toto pero používáme na kreslení tlustých čar.
- `middlepen`; zvolí kruhové pero o průměru `2thinestline`. Toto pero používáme na kreslení středně tlustých čar.
- `thinpen`; zvolí kruhové pero o průměru `thinestline`. Toto pero používáme na kreslení tenkých čar (popřípadě na kreslení čtverečkované sítě a rámečku kolem znaku).
- `drawdots(<bod_1>, <bod_2>, ...)`; kreslí postupně v bodech `<bod_1>`, `<bod_2>`, ... otisk zvoleného pera.
- `cdrawdot(<bod_1>)`; nakreslí v bodě `<bod_1>` otisk obrysu zvoleného pera (tj. pro kruhové pero udělá v daném bodě kroužek, nikoliv tečku).
- `cdrawdots(<bod_1>, <bod_2>, ...)`; bude kreslit postupně v bodech `<bod_1>`, `<bod_2>`, ... otisky obrysu zvoleného pera.

Nejběžnější příkazy pro kreslení obrázků

V METAFONTU lze používat proměnné pro různé typy objektů. Seznam možných deklarácí je uveden v tabulce 5.

Abychom mohli vůbec něco kreslit, je nutno zadávat body v obrázku. Body zadáváme buď explicitně (např. `(4.5, -12)`) nebo je označujeme písmenem `z` a číslem (něco jako index, ale indexem mohou být i desetinná čísla), např. `z5`, `z1.4`, `z[k]` (v tomto posledním případě si zjednodušíme psaní a v textu budeme takový bod označovat jako `zk`). Každý bod `zk` má dvě souřadnice `(xk, yk)`, a tedy výše označené body mají ekvivalentní zápis `(x5, y5)`, `(x1.4, y1.4)`, `(x[k], y[k])`. Body je možné označovat i jinými písmeny než `z`, ale v takovém případě je možné získat jejich jednotlivé souřadnice pomocí funkcí `xpart` a `ypart` uvedených v tabulce 7.

V METAFONTu existuje jednak znak pro přiřazení hodnoty proměnné ($:=$) a jednak znak pro rovnost dvou veličin ($=$). Jednotlivé souřadnice se mohou vyskytovat na obou stranách přiřazení i rovností, body (tj. dvojice souřadnic) pouze v rovnostech. Na základě přiřazení a rovností si METAFONT dopočítává souřadnice bodů. Obecně platí pravidlo, že rovnic a přiřazení musí být tolik, aby jednoznačně určovaly všechny body, které používáme. METAFONT umí řešit soustavy lineárních rovnic, k jejichž zápisu je možné využít právě znaku $=$. Na bod lze pohlížet také jako na vektor, neboť spojnice bodu s počátkem určuje směr i velikost. Názvy pro některé body (vektory), které se často používají, jsou uvedeny v tabulce 2.

Prostředky pro zadávání bodů mohou být velmi rozmanité. U bodů, u kterých je jejich poloha zcela zřejmá a lze je zapsat snadno pomocí souřadnic, můžeme tak učinit. Pro body, u kterých je určení jejich souřadnic obtížné, ale které splňují určité vlastnosti (např. leží v průsečíku přímek či křivek, atd.) stačí popsat tyto vlastnosti a nechat určení souřadnic bodů METAFONTu. S body je možné zacházet jako s vektory. S body lze provádět základní operace (sčítat, odčítat, násobit číslem, ...).

S proměnnými je možné provádět různé operace a úkony, seznam nejběžnějších operací je v tabulce 7.

Pod pojmem **cesta** budeme rozumět spojnici dvou nebo více bodů. Spojnice mezi dvěma body může být úsečka (což naznačujeme znaky $--$) nebo křivka, která je založena na Bezièrových polynomech třetího stupně (toto naznačujeme znaky $..$). Pokud spojujeme více bodů křivkou, bývá tato křivka ve vnitřních bodech hladká (tj. nemá v nich zlomy — viz dále). Pokud spojujeme dva body křivkou, můžeme průběh křivky mezi těmito dvěma body ovlivnit. Můžeme předepsat směr, v jakém má křivka do daného bodu vstupovat, resp. z něj vycházet, nebo můžeme předepsat „napětí“ mezi dvěma body v křivce, které říká, jak je křivka deformovaná. Protože pro nakreslení křivky spojující dva body si METAFONT vypočítá dva pomocné body (které nemusí ležet na křivce), je možné ovlivnit průběh křivky i předepsáním těchto pomocných bodů. Jednotlivé elementy, z nichž se může skládat popis cesty je v tabulce 4. Příklady cest budou uvedeny níže. Popsat cestu znamená říct:

Z bodu vychází (popř. v předepsaném směru) křivka, která může mít předepsané napětí mezi body (nebo se řídit podle předepsaných pomocných bodů) a přichází (popřípadě s předepsaným směrem) do dalšího bodu, a z něj vychází (popř. v předepsaném směru) křivka

... a přichází (popřípadě s předepsaným směrem) do posledního bodu. Pokud je místo posledního bodu napsáno `cycle`, znamená to, že je posledním bodem výchozí bod, a v takovém případě (pokud není předepsán směr) je uzavření cesty hladké.

Pokud potřebujeme opakovat některou část vícekrát (např. zadání stejných bodů v podobných obrázcích), je možné využít aparát definic `METAfontu`. Zde uvedeme pouze velice jednoduchý příklad definice bez parametrů. Definice začíná slovem `def`, za kterým následuje název definice, za kterým je rovnítko. Tělem definice je vše za tímto rovnítkem až po uzavírající slovo `enddef`; . Jestliže napíšeme například

```
def body =      % definice bodů společných pro více obrázků
    z1=(2,0); z2=(2,-4); z3=(6,-4);  z4=(10,0); z5=(10,4);
    z6=(6,4); z7=(6,0);  z8=(10,-4); z9=(2,4);
enddef;
```

pak slovo `body`; uvedené kdekoliv je totéž, jako bychom v takovém místě napsali tělo definice, tj. přiřazení souřadnic bodům `z1`, ..., `z9`.

Nyní jsme již řekli dost informací na to, abychom mohli dát snadnou odpověď na čtenářovu zvědavou otázku, jak jsme nakreslili obrázky 1 a 2. Příkaz `draw` kreslí zadanou křivku zvoleným perem, příkaz `drawdot` udělá „otisk“ zvoleného pera v daném bodě — příkazy pro kreslení a mazání jsou uvedeny v tabulce 3. Zde je zdrojový text souboru se dvěma obrázky (jde o obrázky 1 a 2):

```
\input incpic
```

```
def body =      % definice bodů společných pro více obrázků
    z1=(2,0); z2=(2,-4); z3=(6,-4);  z4=(10,0); z5=(10,4);
    z6=(6,4); z7=(6,0);  z8=(10,-4); z9=(2,4);
enddef;
```

```
beginch("A",40mm#,45mm#,0mm#,12,13);
showmesh:=1;      % zobrazit čtverečkovanou síť
endch;
```

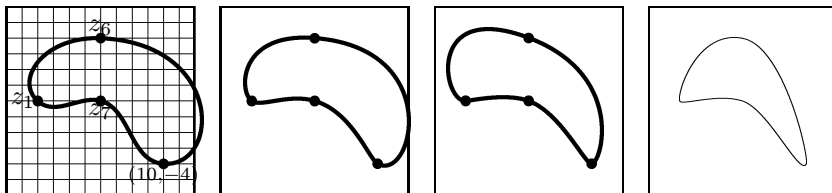
```
beginch("B",45mm#,22.5mm#,22.5mm#,12,6);
showmesh:=1;      % zobrazit čtverečkovanou síť
```

```

body;                % definice bodů
thickpen;           % volba tlustého pera
draw (1,-5)--(11,-5) % rámeček okolo obrázku
    --(11,5)--(1,5)--cycle;
draw z1--z2--z3--z4--z5--z6--cycle; % šestiúhelník
draw z1--z4; draw z3--z6;           % kříž v šestiúhelníku
middlepen;          % volba středně tlustého pera
dashed(z3,(10,-4)); % čárkované čáry
dashed((10,-4),z4); dashed(z6,(2,4));
dashed((2,4),z1);   dashed(z2,z5);
penfordots; drawdot (1,0); % znázornění referenčního bodu
endch;
end

```

Následující série obrázků 3–6 ukazuje, jak jsme vytvářeli obrázek profilu lopatky turbíny (naše zkušenost je, že v tomto případě metoda po-



Obr. 3 Poprvé. Obr. 4 Podruhé. Obr. 5 Potřetí Obr. 6 A je to.

```

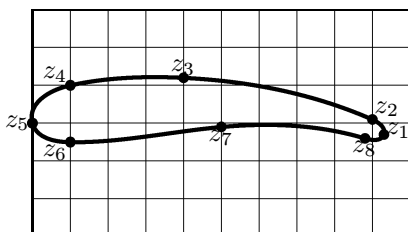
Obr. 3 draw z1..z7..(10,-4)..z6..cycle;
Obr. 4 draw z1..tension 2 and .8..z7..
    tension 8 and 3 ..(10,-4)..tension 2 and 1 ..z6..
    tension 8 and 2..cycle;
Obr. 5 draw z1..tension 5 and .9..z7..
    tension.9 and 5 ..(10,-4)..tension 5 and 1 ..z6..
    tension .8 and 2..cycle;
Obr. 6 draw z1..tension 5 and .9..z7
    ..tension .9 and 5 ..(10,-4)..tension 9 and 1.5
    ..z6..tension 1.2 and 5 ..cycle;

```

kusu a omylu vede nejrychleji k cíli a k nasbírání potřebných zkušeností — pak je těch omylů méně). Pod obrázky jsou uvedeny příkazy, kterými

byly dané křivky nakresleny. Body z_1 , z_6 , z_7 jsou stejné jako ve zdrojovém textu pro písmeno "B" výš. Na posledním obrázku je to, k čemu jsme chtěli dospět (tenkou čáru jsme zvolili proto, aby bylo vidět, že nám šlo o hladkou křivku). Zároveň poznamenejme to, že ze všech pokusů, které jsme dělali dříve s různými typy popisu průběhu křivek (zadávaní směrů, více bodů, atp.) nás nejrychleji dovedlo k cíli používání příkazu `tension` (vytvoření všech čtyř obrázků trvalo u všech všude 5 minut).

Protože už víme, co je to cesta, je vhodné se ještě zmínit o možnosti popisu jednotlivých bodů na cestě. Jestliže nějaká cesta je určena k body,



Obr. 7 Profil křídla.

pak existuje spojitě a vzájemně jednoznačné přiřazení čísel z intervalu $\langle 0, k - 1 \rangle$ bodům této cesty, přičemž platí, že celá čísla intervalu se zobrazují na zadané body cesty. Budeme říkat, že každému času $t \in \langle 0, k - 1 \rangle$ odpovídá bod na cestě. Na obrázku 7 je profil křídla, který lze parametrizovat na intervalu $\langle 0, 8 \rangle$ (neboť je zadáno 8 bodů a jde o uzavřenou cestu, tj. je spojeno 9 bodů — viz níž).

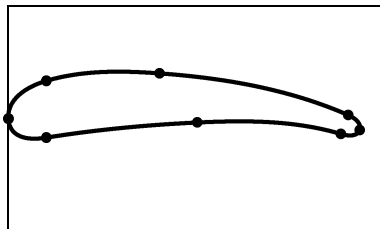
Použití některých operací z tabulek 8 a 7 uvedeme na příkladech. To, k čemu směřujeme, je na obrázku 9. Povšimli jsme si, že všechny tři profily si jsou velmi podobné a liší se natočením, umístěním a měřítkem. Proto jsme se začali zabývat nejdříve náčrtekem jednoho profilu, který je vidět na obrázku 7. Pro lepší názornost ukazujeme i čtverečkovanou síť. Zdrojový text tohoto obrázku pro METAFONT má tvar:

```
def proprofil = path profila;
  z1=(9.3,-0.3); z2=(9.0,0.1); z3=(4.0,1.2); z4=(1,1);
  z8=(8.8,-0.4); z6=(1,-0.5); z7=(5,-0.1); z5=(0,0);
enddef;

beginch("G",50mm#,15mm#,15mm#,10,3);
showmesh:=1;
proprofil;
profila=z1..z2..z3..z4..z5..z6..z7..z8..cycle;
thickpen; draw z1..z2..z3..z4..z5..z6..z7..z8..cycle;
penfordots; drawdots (z1,z2,z3,z4,z5,z6,z7,z8);
```


endch;

Poznamenejme jen, že druhý řádek způsobil, že je v obrázku viditelná i síť. Proměnnou `profil`, definovanou jako cesta, jsme zavedli právě kvůli tomu, že různými deformacemi a posunutími této cesty dosáhneme všech tří profilů (proto nám ani nevádí, že je tento základní profil („nos“ profilu) umístěn v počátku).



Obr. 8 Upravený profil.

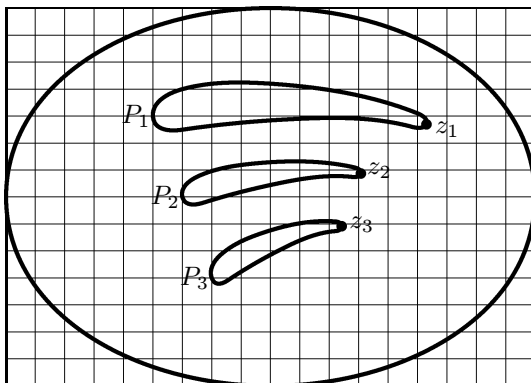
Protože se nám tento profil ještě nelíbil, mírně jsme upravili definici cesty takto:

```
profil:=z1..z2..z3..z4..z5..  
        tension 0.88..z6..z7..z8..cycle;
```

Na obrázku 8 je vidět malá změna, kterou pozměněním zdrojového textu dosáhli. Protože se nám tento výsledek líbil, přistoupili jsme k nakreslení celého obrázku 9. Zdrojový text je uveden pod ním.

Jak je vidět srovnáním se zdrojovým textem předcházejícího znaku, změny jsou nepatrné (tj. všechny tři profily byly nakresleny po natočení, zmenšení a posunutí základního profilu odladěného předem). Zde nastává malá obtíž, protože ve výsledném obrázku chceme mít popsány body, které jsme explicitně nezadali. Jejich umístění na čtverečkováném papíře v náčrtku a po vygenerování se neshodují, protože vygenerovaný obrázek není přesnou kopií náčrtku od ruky. Vyřešení tohoto problému je snadné, neboť si můžeme nechat při generování znaku zobrazit také čtverečkovanou síť (toto stačí udělat na obrazovce) a poznamenat si přibližné souřadnice bodů, do kterých umístíme popis. Toto byl jeden z hlavních důvodů, proč jsme zaváděli kreslení čtverečkové sítě. Výhodou tohoto postupu je to, že neustále pracujeme s relativními jednotkami (čtverečky), které jsou nezávislé na skutečných rozměrech obrázku. Proto změna rozměrů obrázku neovlivní umístění popisu a žádné další změny nebude potřeba v dokumentu ani v budoucnu provádět (tj. ani po změně skutečných rozměrů obrázku). Toto ovšem není jediný způsob, ale nám se zdál nejjednodušší (vzhledem k četnosti výskytu tohoto případu popisu).

```
beginch("I",70mm#,25mm#,25mm#,18,7);
```



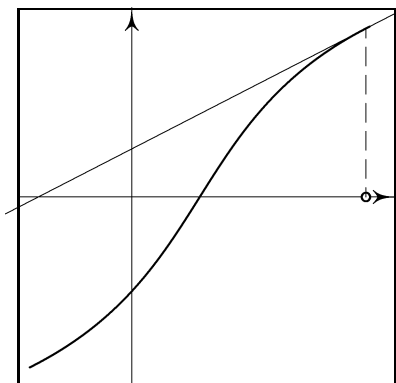
Obr. 9 Posunuté profily.

```

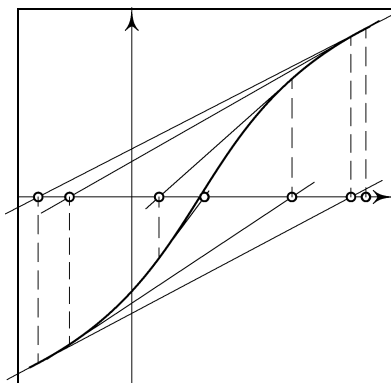
showmesh:=1;    proprofil;
profil:=z1..z2..z3..z4..z5..tension
              0.88..z6..z7..z8..cycle;
z9=(5,3);      z10=(6,0);    z11=(7,-3);
thickpen;     draw profila shifted z9;
draw profila rotated 10 scaled 0.66 shifted z10;
draw profila rotated 20 xscaled 0.5 yscaled 0.66
              shifted z11;
penfordots;   drawdot z1 shifted z9;
drawdot z1 rotated 10 scaled 0.66 shifted z10;
drawdot z1 rotated 20 xscaled 0.5 yscaled 0.66
              shifted z11;
thickpen;
draw (0,0)..(0.5w,-d)..(w,0)..(0.5w,h)..cycle;
endch;

```

V předposlední ukázce je použito několik užitečných funkcí z tabulky 6 a potvrzují se naše slova ze začátku. Přesto, že následující dva obrázky vypadají dost odlišně, zdrojový text pro METAFONT se podstatně liší pouze ve dvou řádcích. První obrázek znázorňuje jeden krok Newtonovy iterační metody na hledání nulových bodů funkce jedné proměnné. Po letmém nahlédnutí do tabulek z konce článku bude zřejmě příslušný zdrojový text zcela čitelný. Pozoruhodné je např. to, že křivka je zadána jen dvěma body (čím méně, tím lépe). Zadáním směrů v krajních bodech



Obr. 10 Jeden krok Newtonovy iterační metody.



Obr. 11 Newtonova iterační metoda (druhého řádu).

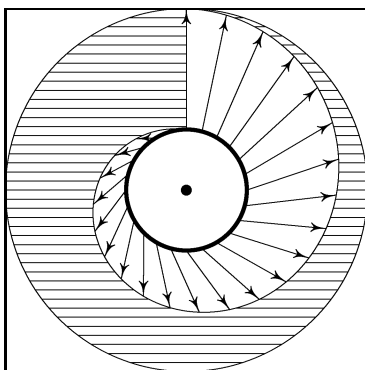
křivky jsme zaručili, že proces zobrazený na druhém obrázku bude konvergentní (čtenáře, jimž se zdá, že používáme mnoho matematických pojmů, prosím za prominutí). Dále je ve zdrojovém textu vidět, jak konstruujeme tečnu ke křivce.

```
beginch("J",60mm#,30mm#,30mm#,10,10);
path křivka;
numeric a,b;
thinpen; axes (3,0); % kreslení os
z1=(0.3,-9); z2=(9.3,9); % krajní body křivky
křivka:=z1{(x2,0)-z1}..{z2-(x1,0)}z2; % křivka
middlepen; draw křivka; % vykreslení křivky
z3=(9.2,0); % první bod na ose
penfordots; cdrawdot z3; % a jeho vykreslení
save a,b; (a,b) = křivka % určení časů průsečíku křivky
intersectiontimes ((x3,-d)--(x3,h)); % a svislice
z13 = point a of křivka; % určení průsečíku na křivce
thinpen; dashed(z3,z13); % čárkovaná svislá čára
z23 = direction a of křivka; % směr tečny ke křivce
z4 = z13+whatever*z23;
z4 = whatever*z3; % průsečík tečny s osou
draw 1.1[z13,z4]--1.1[z4,z13]; % kreslení tečny
endch;
```

Nyní je celkem zřejmé, že obrázek 11 dostaneme, pokud nahradíme číslo 3 indexem [i], číslo 4 indexem [i+1], číslo 13 indexem [i+10], číslo 23 indexem [i+20], a necháme i probíhat celá čísla od 3 do 9. Tedy ve zdrojovém textu pro obrázek 11 budou řádky, které jsou mírně odsazené od levého okraje, nahrazeny těmito řádky:

```
for i=3 upto 9 :           % počet tečen
  penfordots; cdrawdote z[i];
  save a,b; (a,b) = krivka intersectiontimes
              ((x[i],-d)--(x[i],h));
  z[i+10]= point a~of krivka;
  thinpen; dashed(z[i],z[i+10]);
  z[i+20] = direction a~of krivka;
  z[i+1]= z[i+10]+whatever*z[i+20]; z[i+1]=whatever*z[i];
  draw 1.1[z[i+10],z[i+1]]--1.1[z[i+1],z[i+10]];
endfor;
```

Na začátku jsme si řekli, že rámeček kolem náčrtku uděláme tak, aby



Obr. 12 Spirála.

z důvodů přehlednosti do obrázku nemalovali) je velikosti 6×6 čtverečků.

obsahoval i případný popis. Jedna z praktických námitek byla, že je o něco pohodlnější udělat rámeček pouze kolem samotného obrázku. Protože existují dobré důvody pro výše popsany postup, ukážeme na posledním příkladě způsob, jak vyhovět tomuto požadavku a mít počátek čtverečkových souřadnic kdekoli v obrázku.

Uvažujme velmi jednoduchý náčrtek, který je na obrázku 12. Referenční bod je znázorněn velkou tečkou a předpokládáme, že síť (kterou jsme

```
beginch("L",48mm#,48mm#,0mm#,6,6);
currenttransform:= % posunutí souřadného systému
  currenttransform shifted (3aux_sx,3aux_sy);
thinpen;           % pero pro vodorovné čáry
```

```

n:=20; % počet vodorovných čar nad osou
for i=-n upto n: % cyklus pro kreslení vodorovných čar
  draw (-3,i*3/n)--(3,i*3/n); % od okraje k okraji
endfor;
erase filldraw % vymazání "rohů čtverce"
  (0,-3)--(3,-3)--(3,3)--(-3,3)--(-3,-3)--(0,-3)
  ..(-3,0)..(0,3)..(3,0)..cycle;
path spirala,uplnaspirala;
numeric a,b;
r:=1; b:=2; % poloměr vnitřní a vnější kružnice
uplnaspirala:=(0,r+b) % spirála a uzavírající svislice
  ..(r+.75b,0)..(0,-r-.5b)..(-r-.25b,0)..(0,r)--cycle;
spirala:=subpath (0,4) of uplnaspirala; % spirála
erase fill uplnaspirala; % vymazání vnitřku spirály
thinpen draw fullcircle scaled(2r+2b);% vnější kružnice
thickpen draw fullcircle scaled 2r; % vnitřní kružnice
thinpen draw spirala; % spirála
n:=24; % počet vektorů
ar:=360/n; % přírůstek ve stupních
for i=0 upto n-1: % pro vykreslení vektorů
  a:=90-i*ar; % úhel a příslušný počáteční
  z[3i]=r*(cosd a,sind a); % bod na vnitřní kružnici
  z[3i+1]= % "koncový" bod na vnější kružnici
    (r+2)*(cosd(a+50i/n),sind(a+50i/n));
  z[3i+2]= % skutečný koncový bod na spirále
    (z[3i]--z[3i+1]) intersectionpoint spirala;
  vector(z[3i],z[3i+2]); % vykreslení vektoru
endfor;
penfordots drawdot (0,0); % počátek souřadného systému
endch;

```

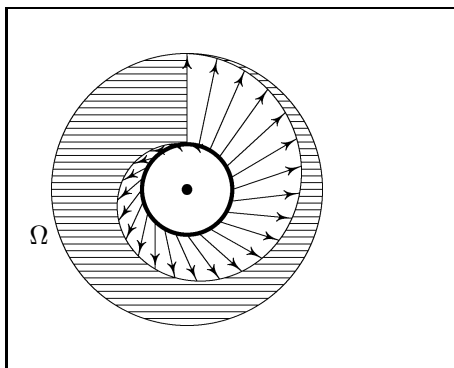
V druhém a třetím řádku je definováno posunutí všeho, co se bude dále kreslit (a tedy i počátku). Aniž bychom se pouštěli do hlubšího vysvětlování, zdůrazněme, že v tomto příkazu je nutno uvádět ve vektoru posunutí tajuplné jednotky `aux_sx`, `aux_sy` přesto, že čísla udávají počet čtverečků. Jestliže změněme rozměr obrázku a počty čtverečků v svislém i vodorovném směru a změněme posunutí počátku, tj. změněme první tři řádky na

```
beginch("M",60mm#,48mm#,0mm#,10,8);
currenttransform:=
    currenttransform shifted (4aux_sx,4aux_sy);
```

Výsledek pak bude vypadat tak, jak je nakresleno na obrázku 13.

Jak začlenit obrázky do textu

Protože začleňování obrázků do textu pomocí makra `incpic.tex` tvoří samostatnou kapitolu, uvedeme jej v příštím čísle \TeX bulletinu. Zde pouze uvedeme nej-



Obr. 13 Upravená spirála.

jednodušší způsob, jak vysázet vytvořený obrázek. Pokud jsme vytvořili například v znaku `A` obrázek v souboru `clanek.mf`, pak po zpracování `METAFONTem` a programem `gftopk` získáme font `clanek.pk` a příslušnou metriku `clanek.tfm`. Potom na začátek souboru pro \TeX stačí napsat `\font\obr=clanek` a v místě, kam chceme obrázek umístit, napsat `{\obr A}`.

Jak si makra opatřit

Makra `incpic.mf` a `incpic.tex` je možné získat v Matematickém ústavu UK.

Tabulky nejběžnějších příkazů

<code>+</code>	sčítání
<code>-</code>	odčítání
<code>*</code>	násobení
<code>/</code>	dělení
<code>**</code>	umocňování (např. <code>a**3</code> je totéž, co <code>a*a*a</code>)
<code>++</code>	pythagorejské sčítání (<code>a++b</code> znamená <code>a*a+b*b</code>)
<code>++-</code>	pythagorejské odčítání (<code>a++b</code> znamená <code>a*a-b*b</code>)
<code>λ[a,b]</code>	lineární kombinace (<code>3[a,b]</code> znamená <code>a+3(b-a)</code>)
<code>abs a</code>	absolutní hodnota čísla <code>a</code> (tj. $ a $)
<code>sqrt a</code>	je odmocnina z čísla <code>a</code> (tj. \sqrt{a})
<code>sind 40</code>	je číslo $\sin 40^\circ$
<code>cosd 40</code>	je číslo $\cos 40^\circ$
<code>dir 40</code>	je vektor $(\cos 40^\circ, \sin 40^\circ)$

Tab. 1 Operace s čísly

<code>a[z1,z2]</code>	je bod $z1+a(z2-z1)$ (<code>a</code> je číslo).
<code>whatever</code>	je proměnná, která při každém použití nabývá nějaké hodnoty (její hodnota je určena zpravidla ostatními vztahy). Např. rovnost <code>z1=z2+whatever*(z3-z4)</code> říká, že body <code>z1</code> a <code>z2</code> leží na přímce se směrnici <code>z3-z4</code> (body <code>z3</code> , <code>z4</code> musí být už určené).
<code>origin</code>	je totéž, co bod (0,0)
<code>right</code>	je totéž, co bod (1,0)
<code>up</code>	je totéž, co bod (0,1)
<code>left</code>	je totéž, co bod (-1,0)
<code>down</code>	je totéž, co bod (0,-1)

Tab. 2 Předdefinované hodnoty některých směrů a hodnot.

<code>pickup pencircle scaled č</code>	volba kruhového pera o průměru <code>č</code>
<code>drawdot z;</code>	nakreslí tečku v bodě <code>z</code> zvoleným perem (obecně nemusí být kruh)
<code>draw p;</code>	nakreslí cestu <code>p</code> zvoleným perem (tj. střed pera se pohybuje po zadané cestě)
<code>fill p;</code>	vyplní uzavřenou cestu <code>p</code> (tj. uzavřená cesta vymezuje plochu, která se vyplní)
<code>filldraw p;</code>	je totéž jako <code>fill p;</code> <code>draw p;</code>
<code>erase draw p;</code>	vymaže plochu ohraničenou obrysem zvoleného pera pohybujícího se po cestě <code>p</code>
<code>erase fill p;</code>	vymaže plochu ohraničenou uzavřenou cestou <code>p</code>

Tab. 3 Příkazy pro kreslení

<code>z[k]</code>	bod o souřadnicích $(x[k], y[k])$
<code>cycle</code>	znamená uzavřít křivku (vrátit se do prvního bodu)
<code>--</code>	úsečka mezi body
<code>..</code>	křivka mezi body
<code>..{dir 30}</code>	křivka přichází do bodu se směrnicí 30°
<code>..{z6}</code>	křivka přichází do bodu se směrnicí danou vektorem <code>z6</code>
<code>{dir 30}..</code>	křivka vychází z bodu se směrnicí 30°
<code>{z6}..</code>	křivka vychází z bodu se směrnicí danou vektorem <code>z6</code>
<code>tension 1.6 and .8</code>	křivka má napětí 1.6 u levého bodu a napětí 0.8 u druhého bodu
<code>tension 2</code>	je totéž co <code>tension 2 and 2</code>
<code>control z3 and z4</code>	křivka se má řídit pomocnými body <code>z3</code> a <code>z4</code>
<code>fullcircle</code>	jednotková kružnice se středem v počátku

Tab. 4 Elementy cest.

Deklarace	Do proměnných a a b lze ukládat	Příklad
numeric a,b; pair a,b; path a,b; boolean a,b; pen a,b; picture a,b; string a,b; transform a,b;	čísla body cesty logické hodnoty tvar pera celé obrázky řetězce transformace	a=4.8; b:=-12; a=(3.4,-6); a:=z1..z2--z3; b:=true; a:=pencircle scaled 1pt;

Tab. 5 Deklarace proměnných.

reverse p	znamená cestu p probíhanou od posledního bodu k prvnímu
p1&p2	výsledkem je cesta vzniklá spojením cest p1 a p2 (koncový bod první cesty musí být počátečním bodem druhé cesty).
length p1	je počet bodů, které cesta spojuje, zmenšený o 1
subpath (a,b) of p	je ta část cesty p, která odpovídá časům z intervalu (a, b).
point t of p	je bod na cestě, který odpovídá času t.
direction t of p	je vektor udávající směrnici tečny k cestě v bodě odpovídajícímu času t
directiontime z of p	je číslo (čas) t, pro který má tečna v odpovídajícím bodě cesty směr vektoru z
p1 intersectiontimes p2	je dvojice časů (s, t), kde čas s přísluší cestě p1 a odpovídá průsečiku cest p1 a p2 a čas t přísluší cestě p2 a odpovídá průsečiku cest p1 a p2
p1 intersectionpoint p2	je bod (s, t), ve kterém se cesty p1 a p2 protínají

Tab. 6 Operace s cestami (zde písmeno p, p1, p2 jsou nějaké cesty).

<code>+</code>	sčítání vektorů $\mathbf{z1+z2}$ je totéž, co $(x1+x2, y1+y2)$
<code>-</code>	odčítání vektorů $\mathbf{z1-z2}$ je totéž, co $(x1-x2, y1-y2)$
<code>*</code>	násobení vektoru skalárem $3*\mathbf{z1}+2(1,4)$ je totéž, co $(3x1+2, 3y1+8)$
<code>abs z</code>	absolutní hodnota vektoru \mathbf{z} (tj. $ z = \sqrt{x^2 + y^2}$)
<code>length z</code>	je totéž, co <code>abs z</code>
<code>angle z</code>	úhel, který svírá vektor \mathbf{z} s kladnou poloosou x
<code>xpart z</code>	je x -ová souřadnice vektoru \mathbf{z}
<code>ypart z</code>	je y -ová souřadnice vektoru \mathbf{z}
<code>z1 dotprod z2</code>	je skalární součin vektorů $\mathbf{z1}$ a $\mathbf{z2}$ (tj. $x1*y1+x2*y2$)

Tab. 7 Operace s body (vektory). Transformace (operace v dolní části tabulky jsou elementárními operacemi, z nichž je možné skládat transformace).

<code>identity</code>	identická transformace
<code>scaled č</code>	násobení číslem \check{c} v obou složkách (roztážení či smrštění)
<code>xscaled č</code>	násobení číslem \check{c} v x -ové souřadnici
<code>yscaled č</code>	násobení číslem \check{c} v y -ové souřadnici
<code>shifted z</code>	posunutí o vektor \mathbf{z}
<code>rotated č</code>	otočení o úhel \check{c}° .
<code>reflectedabout(z1,z2)</code>	znamená zrcadlový obraz vzhledem k přímce určené body $\mathbf{z1}$ a $\mathbf{z2}$
<code>rotatedaround(z1,č)</code>	znamená otočení o úhel \check{c}° okolo bodu $\mathbf{z1}$

Tab. 8 Elementární transformace

Oldřich Ulrych

Záříjový Euro \TeX '92 v Praze — předběžný program

V době uzávěrky tohoto čísla \TeX bulletinu končí přípravy programu záříjové evropské konference uživatelů \TeX u, Euro \TeX '92. Týdenní program konference (úterý 15.9. až sobota 19.9.) se bude skládat ze tří hlavních typů prezentací. Významní představitelé světa \TeX u (a tematicky úzce příbuzných oblastí), a to jak z Evropy, tak i ze zámoří, přednesou šest nebo sedm zvaných přednášek. Tyto přednášky budou tvořit páteř dopoledního programu konference. V odpoledních částech během úterka až čtvrtka bude dále předneseno sedmnáct přihlášených přednášek převážně evropských autorů. Dále budou organizovány tutoriály,³⁾ kterých se bude možno zúčastnit během čtvrtka až soboty.

Při přípravě programu se nepodařilo naplnit jedno z očekávání, které jsme při rozhodování o organizaci Euro \TeX u '92 u nás měli na mysli. Pořádali jsme za jasnou účast většího počtu příspěvků z Československa a okolních států bývalého „východního“ bloku (ve skutečnosti máme problém právě se zajištěním účasti reprezentantů některých těchto národních skupin). Je to škoda, protože jsme přesvědčeni, že v této oblasti mají naši kolegové z Československé komunity uživatelů \TeX u co nabídnout, a je možné jen litovat ne zcela využití příležitosti prezentovat se a navázat kontakty pro budoucnost. Vzhledem k nekomerčnímu charakteru vývoje \TeX u a dalších na něm postavených aplikací máme za podstatné vytváření podmínek k recipročnímu charakteru šíření \TeX pertízy vytvářené kolektivním přispěním jednotlivých uživatelů. Je to jeden ze způsobů, jak v jistém smyslu splácet takové kvalitní volně šířené produkty, jako je třeba $\text{em}\TeX$. (Mimoходом: Eberhard Mattes je jedním z těch, kteří již předběžně přislíbili účast.) Jako organizátoři doufáme, že se nám alespoň skladbou programu podařilo vytvořit dostatečně mnoho lákadel pro všechny zájemce o \TeX a pro jeho uživatele s nejrůznějším zaměřením: získali jsme vysoké procento příspěvků věnovaných perspektivním aplikacím, rozvoji \TeX u a $\text{L}\TeX$ u v budoucnu i používání \TeX u v neanglickém prostředí. Úspěch konference a perspektiva začlenění ζTUGu do

³⁾ Od anglického slova *tutor* = učitel, zvláště takový, který se věnuje jen jednomu nebo malé skupině žáků.

evropského kontextu jako **rovnocenného** partnera příbuzných západoevropských organizací (dnes však již ale nejen jich) závisí i na tom, jak dokážeme předvést skutečný zájem o T_EX u nás, a to jak počty zájemců, tak jejich kvalitou. Tím, že se konference zúčastníte, nezískáte jen nové znalosti o světě T_EXu, ale perspektivně pomůžete i ke kontaktu s aktuálním děním všech uživatelů T_EXu u nás.

Konkrétní náplň programu vypadá následovně:

Zvané přednášky:

- Yannis Haralambous (France): Scholar T_EX: Khmer T_EX and Enhanced Arabic T_EX;
- John Hobby (USA): MetaPost;
- Anita Hoover (USA): The Key to Successful Support: Knowing Your T_EX and L^AT_EX Users;
- Alan Hoenig (USA): Special Effects with METAFONT and T_EX;
- Daniel Taupin (France): MusicT_EX;
- Frank Mittelbach (Germany): L^AT_EX 3 project;

Přihlášené přednášky:

- Petr Božovský (ČSFR): ChiWriter or T_EX?
- Bart Childs (USA): Literate Programming;
- Daniel Flipo and Laurent Siebenmann (France): Hyphenation in the Presence of Accents;
- Harry Gaylord (The Netherlands): Integration of SGML and T_EX
- Bogusław Jackowski, and Marek Ryćko (Poland): POLISH T_EX: from Ready to Use to Handy in Use;
- Theo Jurriens (The Netherlands): T_EX for Everybody?
- Jörg Knappen (Germany): Changing the Appearance of Maths;
- Kees van der Laan (The Netherlands): Typesetting Crosswords via T_EX;
- Klaus Lagally (Germany): ArabT_EX;
- Michel Lavaud (France): A Solution to Help Ensuring the Future of T_EX: Make its Use Easier on Cheap Machines;
- Ирина А. Маховая (Russia): On T_EX Experience in “Mir” Publishers;
- Rama Porrat (Israel): Developments in Hebrew T_EX;
- Laurent Siebenmann (France): The Lion and the Mouse;
- Laurent Siebenmann (France): A Gentle Road via CM to Perfect Hyphenation;

- Petr Sojka, Rudolf Červenka, and Martin Svoboda (ČSFR): $\text{T}_{\text{E}}\text{X}$ for Database
- Philip Taylor (Great Britain): The Future of $\text{T}_{\text{E}}\text{X}$;
- Ondřej Vácha (ČSFR): $\text{T}_{\text{E}}\text{X}$ for Typesetting in a Publishing House;

Tutoriály:

- Yannis Haralambous (France): Internationalization of $\text{T}_{\text{E}}\text{X}$;
- Yannis Haralambous (France): METAFONT and PostScript;
- Philip Taylor (Great Britain): Advanced $\text{T}_{\text{E}}\text{X}$;
- Klaus Thull (Germany): WEB, SpiderWEB, etc.

Kromě toho počítáme s diskusí věnovanou nejruznějším problémům a hledání jejich řešení, budoucímu vývoji $\text{T}_{\text{E}}\text{X}$ u a s ním souvisejících programových i jiných prostředků, případně i diskusím v menších zájmových skupinách ustavených ad hoc podle skutečného zájmu. (Ti, kteří propásli termíny k přihlášení referátu, mohou mít ještě příležitost touto formou – je však nutné mě co nejdříve v této věci kontaktovat.) Během konference počítáme také s valnou hromadou členů $\mathcal{L}\mathcal{T}\text{UGu}$.

Programový i přípravný výbor se těší na setkání s co největším počtem členů $\mathcal{L}\mathcal{T}\text{UGu}$ v září v Praze – pokud jste ještě neposlali přihlášku, je na to nejvyšší čas! Na závěr připomínám, že se lze ubytovat na koleji Kajetánka již od 13.9. (řada zahraničních účastníků toho hodlá využít) a že uvítací party se koná v pondělí 14.9. v prostorách této koleje. Všichni členové $\mathcal{L}\mathcal{T}\text{UGu}$ dostali již poštou oznámení o konferenci s cenami, které jsou pro účastníky z Československa (a zejména pro členy $\mathcal{L}\mathcal{T}\text{UGu}$) mimořádně výhodné. Pokud někdo oznámení nedostal, rádi mu je pošleme v elektronické formě, resp. poštou – stačí se třeba telefonicky přihlásit u Hanky Najmanové.

Jiří Zlatuška

`zlatuska@cspuni12.bitnet`

Další vývoj T_EXu

Na každé T_EXové konferenci se od roku 1989 ozývají otázky spojené s dalším vývojem T_EXu. V Texasu to byla přednáška Franka Mittelbacha „Budoucnost T_EXu“. Další podobné příspěvky následovaly a diskuse o tomto palčivém tématu nekončí. Budoucnost T_EXu je již standardním námětem každé z posledních konferencí o T_EXu.

K nastoupení cesty k dalšímu vývoji T_EXu se rozhodla skupina německy píšících uživatelů DANTE vytvořit pracovní skupinu. Nemá jít o nějakou mezinárodní iniciativu DANTE, ale o spolupráci všech, které takový projekt zajímá. Každý je zván, aby přispěl svými nápady a požadavky, resp. i prací v tomto směru. I když dosud není nic definitivně rozhodnuto, jsou plánovány tři fáze.

1. etapa: Co je potřeba změnit?
2. etapa: Jak se dá těchto změn dosáhnout?
3. etapa: Realizace.

Projekt by měl začít na březnovém T_EXovém setkání DANTE v Hamburku. Tam by měla být vytvořena pracovní skupina, která by měla jednotlivé fáze projektu řídit.

Myslím, že další vývoj T_EXu je velmi důležitý, protože konkurenční firmy nezhálejí a jejich produkty jsou stále vylepšovány. Proto je důležité, aby T_EX, případně jeho pokračovatel zůstal tam, kde po celou řadu let byl, tj. aby zůstal stále tím opravdu nejlepším.

Joachim Lammarsh,
President DANTE e.V.
přeložil -ju-

Integrované prostředí pro T_EX (zvané T_EXShell)

Program T_EXShell představuje pohodlné, uživatelsky příjemné integrované prostředí pro T_EX. Umožní vám (po nastavení konfigurace T_EXu tak, jak ji používáte na svém počítači) snadnou přípravu dokumentů, jejich překlad T_EXem, zobrazení na obrazovce a vytištění na tiskárně. Je v podstatě náhradou za program MENU.EXE a systém *.BAT souborů dodávaných v balíku C_ST_EX. T_EXShell obsahuje vlastní integrovaný editor a všechny další potřebné programy jsou volány přímo z T_EXShellu (T_EX, ovladače pro obrazovku a tiskárnu, BIBT_EX, T_EXCad, případně externí editor a další uživatelem definované pomocné programy).

T_EXShell byl napsán v jazyce Turbo Pascal verze 6.0 firmy Borland a využívá systém unit zvaný Turbo Vision, který dává programu profesionální vzhled a snadné ovládání pomocí menu s plnou podporou myši. Dialog s uživatelem probíhá prostřednictvím oken a dialogových boxů, k dispozici je kontextově závislá nápověda. Škoda jen, že autor nevyužil schopnost stavového (spodního) řádku zobrazovat v každé situaci stručnou nápovědu.

Před prvním použitím T_EXShellu je nutné nastavit konfiguraci, tj. vytvořit správný soubor *.CFG. Ten v sobě zahrnuje údaje o adresářích pro zdrojové texty, grafické soubory, fonty a metriky, indexové styly a programy T_EXCad apod., dále jména překladače T_EXu, externího editoru, ovladačů tiskáren a obrazovky, prostředí (environments) pro jednotlivé volané programy a parametry příkazových řádků pro jednotlivá volání. Základní konfigurační soubor má jméno TEXSHELL.CFG, kromě něj je možné vytvořit konfiguraci ke každému zdrojovému textu. Příslušný soubor pak má jméno <text>.CFG. Tento soubor se automaticky otevírá při otevření příslušného textu a ukládá při jeho uzavření. Pokud T_EXShell nezná jméno .CFG souboru, použije standardní TEXSHELL.CFG.

Vestavěný editor je editor převzatý z balíku Turbo Vision. Není příliš pohodlný, neumožňuje například automatický přechod na novou řádku a maximální velikost zpracovávaného textu je 64K, takže se spíše hodí na opravy než k psaní textu. Pokud jej budete používat k tomuto účelu, jistě využijete jeho schopností. Kromě všech obvyklých operací vyhledávání a nahrazování řetězců to je například možnost otevřít najednou až

devět souborů (každý ve zvláštním okně), libovolně mezi nimi přenášet či kopírovat bloky textu, prohlížet si současně protokoly o překladu (soubory *.LOG), podle vlastní vůle a vkusu měnit polohu, velikost i barvu oken a podobně.

Pokud chcete již napsaný a přeložený text prohlédnout nebo vytisknout, $\text{T}_{\text{E}}\text{X}$ Shell zavolá příslušné obslužné programy. Pro ně je možné nastavit parametry – ovšem pomocí přepínačů na příkazovém řádku. Bylo by velmi pohodlné, kdyby $\text{T}_{\text{E}}\text{X}$ Shell nabízel při nastavování těchto parametrů jakési menu, kde by byly jednotlivé možné volby uvedeny v jazyce člověku přístupném, a do příkazového řádku se ukládaly automaticky.

Instalace programu je jednoduchá, autor ji totiž nechal zcela na vás. Stačí tedy vytvořit na disku vhodný adresář a do něj rozbalit patřičný archivní soubor. Po úspěšném rozbalení zbývá nastavit konfiguraci, což je patrně nejsložitější operace. Když totiž nastavujete v položce **OPTIONS** jednotlivé parametry, například adresáře, $\text{T}_{\text{E}}\text{X}$ Shell vždy kontroluje, zda uvedené adresáře či soubory existují. Pokud zadáte jméno neexistujícího souboru nebo adresáře, $\text{T}_{\text{E}}\text{X}$ Shell odmítne opustit dialogový box a nastaví kurzor na chybné jméno. Bohužel ale neposkytne žádné další informace ani možné varianty (například strom adresářů apod.). Jestliže tedy neznáte všechny cesty k souborům zpaměti, nezbývá vám než zrušit dialogový box a přijít tak o již nastavené údaje. Pomocí **OPTIONS** nelze nastavit $\text{T}_{\text{E}}\text{X}$ ové formáty, zde musíte přepsat konfigurační soubor přímo vhodným editorem (lze i $\text{T}_{\text{E}}\text{X}$ Shellovým).

Pokud $\text{T}_{\text{E}}\text{X}$ Shell z nějakých důvodů nemůže uložit na disk svůj *.TMP soubor při volání $\text{T}_{\text{E}}\text{X}$ u, DosShellu či dalších programů, oznámí chybu, ale pokračuje dál, jako by vše bylo v pořádku – že se vloudila chybička, poznáte v okamžiku, kdy je třeba vrátit se zpět do $\text{T}_{\text{E}}\text{X}$ Shellu. Nejde to, a v horším případě musíte znovu zavést operační systém. $\text{T}_{\text{E}}\text{X}$ Shell vytváří *.TMP soubor buď ve svém adresáři, nebo v adresáři daném systémovou proměnnou TMP. Příčinou chyby může být nedostatek místa, chybějící přístupová práva na síti, ale i neúmyslné smazání *.TMP souboru.

$\text{T}_{\text{E}}\text{X}$ Shell sice umožňuje nastavit všechny možné i nemožné parametry, ale nepodařilo se mi nastavit adresář pro *.DVI soubory. $\text{T}_{\text{E}}\text{X}$ Shell je ukládá vždy do adresáře se zdrojovým textem. Nepomohlo ani nastavení parametrů příkazové řádky pro překlad $\text{T}_{\text{E}}\text{X}$ em, protože v $\text{T}_{\text{E}}\text{X}$ Shellu můžete definovat jen část příkazové řádky mezi jménem *.EXE souboru a jménem zdrojového textu. $\text{T}_{\text{E}}\text{X}$ Shell totiž za vámi definovanou část

připojí jméno zdrojového souboru automaticky. Pokud tam toto jméno umístíte sami, příkazová řádka bude mít příliš mnoho parametrů.

Při psaní textu a jeho překladu se mi dvakrát stalo, že $\text{T}_{\text{E}}\text{X}$ Shell přestal reagovat na klávesy a „zatuhnul“. Bylo to v případě, kdy jsem se snažil prohlížet příliš dlouhý *.LOG soubor. Nejsem si jist, jestli je to skutečně chyba $\text{T}_{\text{E}}\text{X}$ Shellu, nebo pouze náhodná shoda okolností.

$\text{T}_{\text{E}}\text{X}$ Shell není napsán pro používání v zemích, které trápí problém vlastní podivné abecedy. Pro psaní češtiny je nutné před spuštěním $\text{T}_{\text{E}}\text{X}$ Shellu do paměti nainstalovat vhodný ovladač a před překladem musíte zkonvertovat soubor do kódu vhodného pro $\text{T}_{\text{E}}\text{X}$. Bohužel tím ztrácíte možnost použít k ovládání programu některé kombinace kláves (pokud je zapnuta česká klávesnice).⁴⁾

$\text{T}_{\text{E}}\text{X}$ Shell je uživatelsky příjemné integrované prostředí, které vám zjednoduší přípravu dokumentů v $\text{T}_{\text{E}}\text{X}$ u od jejich napsání až po vytisknutí. Trochu práce dá jeho přizpůsobení českým a vašim poměrům, ale tato námaha se vyplatí. Ovládání je velmi snadné – systém menu je přehledný a téměř všeho lze dosáhnout pomocí myši. Ve srovnání s menu, které je obsaženo v balíku $\zeta\text{T}_{\text{E}}\text{X}$, je tento systém velmi flexibilní, změna konfigurace nevyžaduje přepsání dávkových souborů. Dokumentace k programu je velmi podrobná. Některé nedostatky, kterými program trpí, se snad podaří v dalších verzích odstranit, a potom bude $\text{T}_{\text{E}}\text{X}$ Shell jistě velmi užitečným nástrojem.

Michal Voců,
MFF UK

- Jedním ze zajímavých editorů je i **Multi-Edit**, s jehož relativně plně funkční demo-verzí se patrně řada z vás měla možnost seznámit. Testovali jsme ji před několika měsíci, když nám ji přinesl kolega Jaroslav Antoch. Nyní nám Multi-Edit 500P nezávisle doporučil kolega Ivan Lukáč z Bratislavy. Plnou verzi distribuuje firma American Cybernetics, 455 S. 48th St. Suite 107, TEMPE, AZ 85281, USA. Pokud však budete o koupi uvažovat, rozhodně vám doporučujeme se nejdříve seznámit s demo-verzí.

⁴⁾ Toto neplatí pro každý ovladač. Rovněž není nutno konvertovat soubor psaný česky, používáme-li $\text{eT}_{\text{E}}\text{X}$ a vhodnou **tcp** tabulku při vytváření formátu. (Pozn. redakce.)

Poznámky ke koupi editoru

Domnívám se, že vytvoření dobrého $\text{T}_{\text{E}}\text{X}$ ového editoru představuje sice významnou, ale přesto jen počáteční etapu, a tu je potřeba v pravou chvíli uzavřít. Navrhuji proto koupit zatím jediný funkční editor (tedy asi $\zeta\text{E}D$) a smluvně zavázat autora, že do určitého termínu doplní podporu práce s myší, která zde zřejmě nejdříve schází.

Velmi citelně však postrádám jiný rys: při vyhledávání podřetězců by skutečně $\text{T}_{\text{E}}\text{X}$ ový editor neměl rozlišovat mezi mezerou a kombinací mezer a znaků pro konec řádku. Významné procento vyhledávání (např. při korekturách) skončí neúspěšně právě díky tomu, že hledaný podřetězec je ve zdrojovém textu zalomen přes řádek. Ale to lze odložit do další etapy.

V čem by podle mne další etapa měla spočívat? Nedávno jsme zde v Opavě získali od sponzora TEXTURES pro Macintosh (chtěl bych o nich brzy napsat do Bulletinu podrobněji). Možnost souběžného spuštění editoru, $\text{T}_{\text{E}}\text{X}$ u a prohlížení je natolik zajímavá, že jsem se pokusil o totéž s $\text{emT}_{\text{E}}\text{X}$ em a $\zeta\text{E}D$ em pod Windows. To se až na drobné chybičky povedlo (od P. Ševečka jsem sice získal podporu $\zeta\text{E}D$ u pro Windows, ale na černobílé VGA se mi v prvním sloupci ztrácí kurzor). Nebylo však možné dostat „view“ do okna ve funkční podobě.

Nyní je sice možné získat prohlížeče pod Windows, ale problém je i ve Windows samotných. Podařilo se mi nakonec zorganizovat $\text{T}_{\text{E}}\text{X}$ ovou smyčku tak, aby stačilo (1) označit jméno souboru, (2) ve správném pořadí spouštět a uzavírat editor, $\text{T}_{\text{E}}\text{X}$ a prohlížeč, ovšem výsledek má daleko ke komfortu poskytovanému TEXTURES .

Nehledě na to, že každý si nemůže spustit Windows v „enhanced“ módu.

Věřím proto, že nejspíše za rok se bude jednat o integrovaném prostředí pro české a slovenské uživatele a zasílám první sadu požadavků:

- Možnost souběžného spuštění editoru a $\text{T}_{\text{E}}\text{X}$ u a souběžného spuštění editoru a „view“ v oknech současně viditelných na obrazovce a snadného přepínání mezi nimi (běh programů v pozadí zřejmě není nutný).
- Plně $\text{T}_{\text{E}}\text{X}$ ový editor s podporou myši a podporou $\text{T}_{\text{E}}\text{X}$ ových konvencí při prohledávání (nerozlišovat mezi mezerou a znaky, které $\text{T}_{\text{E}}\text{X}$ interpretuje jako mezeru).
- Integrovaný spell-checker a další možnosti předzpracování dokumentu pro rutinní redakční úpravy (např. svázání všech neslabičných předložek s následujícím slovem atd.) v běžném textu, tj. mimo matematický mód.

- Možnost nastavení režimu T_EXové smyčky (zdrojový soubor, pracovní adresáře, národní podpora, tj. kódování diakritiky a další předzpracování souboru, T_EXový formát, tj. L^AT_EX resp. $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX a počáteční parametry prohlížeče).
- Nastavitelnost pracovních adresářů pro zdrojové soubory, jako *.tex, *.sty a pomocné soubory jako *.dvi, *.log a *.aux. Posledně jmenované lze výhodně ukládat na RAM-disk.
- Spouštění T_EXové smyčky s „view“ nebo bez něho, podle charakteru chyb nebo přání uživatele.
- Aktualizace prostředí „jedním tlačítkem“ (editovaný dokument není z pochopitelných důvodů vhodné přepisovat v době, kdy je rozpracován T_EXem). V podstatě jde jen o nové spuštění smyčky, ale nesmíme zapomenout ukončit starou, což může být zdlouhavé.

Doufám, že třetí etapa (která už možná nebude probíhat pod MS DOSem) nezačne dříve, než skončí dosud neuzavřená první.

M. Marvan, Opava

Vážení kolegové.

Když jsem v loňském prvním čísle T_EXbulletinu četl o zamýšlené koupi editoru, měl jsem *pro sebe* problém editoru pro T_EX (či spíše L^AT_EX) jakž takž vyřešen a měl jsem dojem, že ζ TUG k tomu má jen krůček. Navíc jsem v té době měl katastrofálně málo času a tak jsem byl zticha. Při čtení podmínek konkursu na editor mne sice napadaly všelijaké doplňky, ale konkurs již byl vypsán, takže jsem opět neměl důvod, proč se do toho míchat. Teprve nyní, když vidím, že v záležitosti editoru stále panuje jakási nerozhodnost, rozhodl jsem se poslat Vám několik připomínek.

Po přečtení recensí jsem rád, že korektor už byl zakoupen a že funguje nezávisle na editoru. Alespoň něco tedy funguje a bude to k dispozici i těm, kteří se rozhodnou nový editor nepoužívat. Korektor totiž používám vždy jen chvíli, když už je celý dokument napsán, takže jeho uživatelský interface není tak kritický. S editorem to je podstatně horší, neboť aspoň 90 % času stráveného u počítače něco edituji. A edituji nejen texty v T_EXu, ale i jiné texty, programy, různá data a podobně.

Zjistil jsem, že informace o editoru ζ ED, které mám k dispozici, jsou velice kusé, čímž nechci říci, že recenze jsou špatné. Jako recenze se to čte dobře, ale neposkytuje mi to informace, podle kterých bych se mohl rozhodnout. Nicméně editor K bych na základě recensí s klidným svědomím zamítl.

Na editoru ζ ED (a vlastně i na konkursních požadavcích) mi chybí několik „drobností“:

- Pokud se počítá s poměrně velkým množstvím maker, je naprosto nezbytná operace UNDO, která „zruší“ chybně zapsané makro. Velký počet maker si totiž málokdo spolehlivě zapamatuje, ale s operací UNDO se případný omyl dá snadno napravit.

Daleko raději bych však viděl plnohodnotnou operaci UNDO, která by fungovala na rozumně dlouhou posloupnost *libovolných* operací editoru.

- Kromě skoků mezi párovými závorkami by jistě všichni \LaTeX isté uvítali i skoky mezi párovými `\begin` a `\end`. Jsou to vlastně také závorky.
- Nikde jsem nenašel zmínku o operaci přeformátování odstavce. Víím, že \TeX formátuje sám, ale úhledně napsaný zdrojový soubor má taky své přednosti.
- Editace libovolně dlouhých souborů se jistě nevyužije denně, a znám i příkazy `\input` a `\include`, ale když mi někdo přinese soubor, který je delší, nechci být nucen nejprve shánět jiný editor, kterým bych to rozparceloval na kusy.

Užitečnost těchto funkcí mám ověřenu zhruba dvouletým používáním MultiEditu, k němuž jsem si naprogramoval (MultiEdit má svůj speciální programovací jazyk) několik podpůrných funkcí: automatické otevírání stejnojmenného log-souboru, skoky mezi párovými závorkami včetně `\begin` a `\end`, zápis řady \LaTeX ovských konstrukcí a několik klávesových maker. Většinou jsem se inspiroval tím, co MultiEdit nabízel pro podporu jiných programovacích jazyků. Požadavky ζTUGu jsem v té době neznal, ale většinu z nich by MultiEdit (po jistém programátorském úsilí) mohl splnit. Má to jediný háček: MultiEdit není public domain,⁵⁾ a proto ho nelze přihlásit do konkursu. Teprve při čtení recensí jsem si uvědomil, že by takto rozšířený MultiEdit mohl posloužit také jako experimentální nástroj k ověření důležitosti požadavků na kupovaný editor.

Zjistil jsem totiž, že ne každá operace je stejně užitečná, a neužitečnost některých se asi zjistí teprve pokusem. Měl jsem například naprogramovány skoky mezi sudými a lichými znaky $\$$, ale zjistil jsem, že je to velice pomalé, a tudíž skoro k ničemu, neboť diagnostika \TeXu je v tomhle docela slušná. Naproti tomu skoky mezi `\begin` a `\end` a mezi složenými závorkami jsou k nezaplacení, neboť zde je diagnostika \LaTeXu mizerná.

Ale abych se vrátil k recensím editoru ζED .

J. Chlebíková se zmínila, že by chtěla označit celou řádku jako blok, aniž by skákala na začátek řádky. K tomu bych dodal, že já bych chtěl manipulovat s bloky, tvořenými několika celými řádkami. V MultiEditu to jde a používám to možná častěji než ten „normální“ blok.

Mohl bych jmenovat další užitečnosti (třeba regulární výrazy při vyhledávání a náhradách), ale víím, že všechno chtít asi nemohu.

⁵⁾ Teprve před pár týdny, když se mi dostala do rukou demo-verze MultiEditu, jsem zjistil, že to funguje i s ní.

Požadavek skoku na další chybu se mi zdá trochu problematický a já bych na něm netrval, lze-li se snadno dívat do chybového souboru. Podle mých zkušeností oprava chyby poměrně často vyžaduje vložit nebo vymazat nějakou tu řádku, což ovšem částečně znehodnotí údaje z chybového souboru. Mohu sice opravovat chyby směrem od konce souboru, ale pak mám zase problémy se zavlečenými chybami: jediný překlep může způsobit mnoho chybových hlášení a já se takto setkám nejprve s tím posledním.

Trochu mne zarazilo, že proti práci s okny žádný z recesentů nic nenamítal, pouze J. Chlebíková si posteskla, že by mohlo být užitečné mít okna vedle sebe a ne nad sebou. Z této nenápadné věty usuzuji, že ζ ED umí zobrazit okna pouze pod sebou a že je tento stav pokládán za normální a uspokojivý. Je to samozřejmě lepší než nic, ale nadšen nejsem. Připomíná mi to editor ED z Editor ToolBoxu od Borlanda, který se stal vzorem (ne-li přímo východiskem) několika tuzemských editorů. Nebýt onoho postesku, mohl jsem si myslet, kdoví jaká práce s okny je v editoru ζ ED implementována.

Ono to platí všeobecně: pod tímž názvem a popisem funkce se může skrývat podstatně odlišná činnost editoru. Jako příklad uvedu zálohování práce po určitém čase. Pokud se zálohování provede i během mého desetiminutového přemýšlení, je to v pořádku. Ale třeba ChiWriter se zálohováním vždy počká, až stisknu klávesu. To je jednak k vzteku (když už konečně vím, co chci napsat, stisknu prvou klávesu a musím čekat, než se soubor uloží), jednak to je zcela neúčinné, když na počítač zapomenu a dojde k výpadku napájení. Z textu recense vůbec není patrné, jak to doopravdy funguje. Mimochodem MultiEdit zálohuje, když po nastavenou dobu (třeba jednu minutu) není stisknuta žádná klávesa. Je to celkem příjemné (zálohuje se, když přemýšlím), ale také to není bez chyb: čím jsem pilnější, tím méně se zálohuje a už jsem takto přišel o dvě hodiny práce.

Požadavek, aby bylo možno určit pořadí a názvy extensí při natahování souborů, pokládám za zbytečný, neboť to lze zajistit jednoduchou dávkou, která editor startuje. Použily by se v ní příkazy typu `if exist`.

Mám určité pochybnosti ohledně otázky, jaké procento členů ζ TUGu bude nový editor *skutečně používat*, bude-li koupěn. Trochu se bojím, že nový editor použijí zejména kolektivní členové, a to tak, že jej správci počítačů vnutí na disky ostatním uživatelům.

Zkusil jsem si chvíli hrát na pesimistu a napadly mne tyto důvody, proč by uživatel zůstal u starého editoru:

Každý uživatel \TeX u již *nějaký* editor používá, a kdo chce používat editor nový, musí se jej naučit. Ze zkušenosti vím, že zvládnout elementární úkony bývá snadné, ale pro rutinní práci je třeba si na nový editor zvyknout, naučit se využívat jeho přednosti, obházet jeho nedostatky a zejména dělat řadu věcí automaticky. Přejechání na nový editor se vším všudy zpravidla trvá dost

dlouho a nikdo by jej (z duševně-hygienických důvodů) neměl podstupovat zbytečně často. Zpravidla je k tomu člověk donucen (např. přechodem na jiný typ počítače nebo operačního systému), někdy se *odvážnější* uživatelé nechají zlákat možnostmi nového editoru. Radu uživatelů však *žádné* nové možnosti nezlákaají.

Jsou lidé, kteří používají sofistikovaný editor a jeho možnosti skutečně využívají. Tito lidé by přechodem něco získali a něco ztratili a těžko říci, čeho bude víc. Je také pravdou, že někteří z nich by možná byli ochotni za skutečně dobrý editor i zaplatit.

Je tu ovšem nezanedbatelný zástup těch uživatelů, kteří se naučí vždy pouze nejelementárnější povely, a další zástup těch, pro které je zvyk železnou košilí. Znam například lidi, kteří všechny texty (včetně \TeX) píšou ve starší verzi WordPerfectu, česká písmena dělají dvojhmatem s klávesou Alt (např. $\boxed{\text{Alt}}$ + $\boxed{\text{c}}$ = č), a to prostě proto, že před čtyřmi lety to byla první implementace češtiny na PC, která se jim dostala do rukou. Dokonale si na to zvykli a dosud nenašli pádný důvod k přechodu na něco „normálnějšího“.

Zakoupený editor asi nebude public domain — bude zdarma pouze pro členy \CSUG . Pak ovšem přechod na nový editor způsobí potíže těm individuálním členům, jejichž zaměstnavatelé nejsou členy kolektivními (např. pro Stavební fakultu ČVUT je 5 000 Kčs/rok příliš mnoho). Nový editor by tito členové směli používat pouze doma.

Řada uživatelů je nucena používat nějaký editor i mimo \TeX . Mnozí z nich mají tendenci používat svůj oblíbený editor tam i onde. Prozatím to vypadá tak, že pro \TeX používáme ten editor, který používáme jinde. A nový editor pro tak mnohostranné použití asi nebude moc šikovný. Ale konec konců — skoro každý editor se dá *nějak* použít.

Dosti sýčkování. Myslím, že *slušný* editor by se koupit měl, i kdyby jej používali pouze kolektivní členové. Oni konec konců nejvíce platí.

Výše zmíněný doplněk k MultiEditu jsem nabídl Martinovi Bílému do \TeX ovského archívu na `cs.felk.cvut.cs`. Bude-li zájem, mohu poslat i zdrojový text.

Jiří Demel
Čechova 6
170 00 Praha 7

e-mail: demel@csearn

Konverzní program z formátu Text602 do T_EXu

Ve čtvrtém čísle T_EX Bulletinu jsem apeloval na všechny naše členy, aby si své dobré nápady a dobré i špatné zkušenosti nenechávali jen pro sebe, ale aby se s nimi podělili s ostatními. A protože, jak praví klasičtí, „*Exempla trahunt*“ (a taky poté, co jsem na valné hromadě viděl, jak jsou odměňováni nejpilnější přispěvatelé), obtěžuji Vás nyní se svou informací o prográmku, který jsme u nás na Vysoké škole zemědělské v Brně vyvinuli pro svoji potřebu.

Text602 rozhodně není ideální prostředek pro tvorbu vstupních souborů pro T_EX, ale je široce rozšířen díky své dobré obchodní politice. Většina lidí na VŠZ (ale i mimo ni) má zatím velice malé vědomosti o možnostech počítačem podporované sazby. To vede k následujícímu postupu při jejich práci s počítačem.

Sledovaná osoba se rozhodne napsat své další dílo pomocí počítače. Rozhlédne se kolem a od sekretářky se dozví, že nejlepší je Text602 (jak by mohla ostatně poradit něco jiného, když nic jiného sama nezná). I začne v něm (v ní? ... ta 602 nebo ten 602) psát nejdříve stručný dopis, pak osnovu přednášky a nakonec skripta. Po napsání skript je odnese vytisknout na laserovou tiskárnu, aby „To vypadalo pěkně“. To je ovšem nesplnitelný požadavek, protože Text602 nemůže překročit svůj stín. A což teprve, když autor zvolil v některé části ‘elite’ nebo ‘condens’, například v tabulkách. To pak HP LaserJet vyrábí vyložené zmetky.

Nastane chvíle, kdy místo ke kováříčkovi přijdou ke kováři a dozví se, že „to mají psát v T_EXu, když už je Zemědělkou kolektivním členem \mathcal{G} TUGu.“ Ale text už je napsán, a zbavit se většiny udělané práce tím, že text exportujete ven z Text602 jako ASCII file — to není to pravé.

Proto jsme spolu s Jirkou Rybičkou vyrobili překladač 602toTEX. Samozřejmě, že neumí všechno. Ono to ani dost dobře nejde, pokud bude Text602 místo tabelátoru ukládat pět mezer nebo pokud bude vzorec napsán na tři různé řádky, kdy indexy umísťujete také pomocí ručně doplňovaných mezer. Ale jistou část práce zachová a následné úpravy nemusí být tak rozsáhlé a týkají se hlavně formátování výstupu, kontroly vysázení vzorců a sázení tabulek.

Tabulky jsou dalším cílem našeho úsilí, myslím si, že během března až dubna (zde původně bylo ledna až února) budeme mít hotov program konvertující tabulku Text602 do \LaTeX u.

A teď již k popisu programu: Program byl napsán v Turbo Pascalu verze 5.5. Rychlost nebyla nejvyšším kritériem — nepočítali jsme, že by někdo používal Text602 jako standardní editor k \TeX u. Jde spíše o jednorázový překlad do formátu \TeX u.

Při volání programu zadáváte tři parametry: vstupní soubor, výstupní soubor a tabulku substitucí. Tabulku nahrazení jsme umístili mimo tělo hlavního programu, protože má každý trochu jiné požadavky. Tabulka nahrazování je načítána při spouštění překladu a není určena pevně. Tabulky nahrazení si může každý vytvořit sám, několik standardních dodáváme přímo. Uživatel si může naeditovat zvlášť nahrazení při zapnutí fontu a zvlášť při vypnutí fontu. To umožňuje například nahrazení fontu „velké“ řetězcem `\velke{ ... }` a po vložení příkazu `\let\velke=\section` máte část práce ulehčenu.

Ve stejném článku jsem také nabádal k tomu, abychom si v rámci CSTUGu zachovali nekomerční charakter. Oznamujeme tedy tímto článkem, že program 602to \TeX je dán všem řádným členům \CSTUG u k dispozici. Samozřejmě, má své dětské chyby, ale pokusíme se je na základě Vašich upozornění a doporučení postupně odstraňovat. Ostatně „Darjonomu kaňů v zuby něsmótrjat“.

Na závěr snad jedinou poznámku — tento program nikdy nebude dokonalý a je lépe začít dokument psát okamžitě přizpůsobený \TeX u než jej později předělávat.

Mnoho štěstí v tomto roce Vám přejí⁶⁾

Pavel Sekanina a Jiří Rybička

⁶⁾ Autoři poslali svůj příspěvek na začátku tohoto roku, ale zřejmě poprvé v krátké historii našeho sdružení bylo nutno pro nedostatek místa přesunout několik článků do dalšího čísla.

Obsah TUGboat 12 (3 & 4) 1991

Pokračujeme v otiskování obsahu 12. ročníku TUGboat (viz $\text{T}_{\text{E}}\text{X}$ bulletin 2 a 3, 1991), který je dostupný v Praze, Brně a Bratislavě.

TUGBOAT 12 (3 & 4) 1991 (1991 TUG Conference Proceedings)

Part 1

Introduction	351	Nelson Beebe / <i>President's introduction</i>
Keynote Address	353	Nico A.F.M. Poppelier / <i>Two sides of the fence</i>
Publishing	359	Laurie J. Petrycki / <i>Comparing $\text{T}_{\text{E}}\text{X}$ and traditional typesetting for the composition of a textbook</i>
	367	Frederick H. Bartlett / <i>Contra-$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, or what really works in the publishing world</i>
	372	Howard Ratner and Kenneth Dreyhaupt / <i>$\text{T}_{\text{E}}\text{X}$ in a book production department</i>
	377	Berthold K.P. Horn / <i>dvi and eps: The ideal author-to-publisher interface?</i>
	382	Neil A. Weiss / <i>Producing a book using $\text{T}_{\text{E}}\text{X}$: How the process works</i>
	387	Samuel E. Rhoads / <i>Authors new to $\text{T}_{\text{E}}\text{X}$ publish a textbook with a publisher new to $\text{T}_{\text{E}}\text{X}$</i>
	393	Colleen Brosnan / <i>The "Five Cs": A guide to successful publication using $\text{T}_{\text{E}}\text{X}$</i>
	397	Anita Z. Hoover / <i>$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}/\text{T}_{\text{E}}\text{X}$ user: A typist, or typesetter?</i>
	401	John Lavagnino / <i>Simultaneous electronic and paper publication</i>
	406	Robert W. McGaffey / <i>SGML versus/and $\text{T}_{\text{E}}\text{X}$</i>
	409	Andrew E. Dobrowolski / <i>Typesetting SGML documents using $\text{T}_{\text{E}}\text{X}$</i>
	415	C. M. Sperberg-McQueen / <i>Specifying document structure: Differences in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ and TEI markup</i>
	422	Takashi Kakiuchi, Yuki Kusumi, Yoshiyuki Miyabe, and Kazu Tsuga / <i>A</i>

		<i>structured document preparation system—Autolayout version 2.0—an enhancement for handling multiple document types</i>
	430	Linda Williams / <i>Refining a process</i>
Future Issues	434	Luigi Semenzato and Edward Wang / <i>A text processing language should be first a programming language</i>
	442	Michael Vulis / <i>Should T_EX be extended?</i>
TUG Business	448	TUG 1992 annual meeting, Portland, Oregon
	449	Institutional members
Forms	451	TUG membership applications
Advertisements	453	T _E X consultants
	468	Index of advertisers
 Part 2		
Applications	471	Cay S. Horstmann / <i>Automatic conversion from a scientific word processor to T_EX</i>
	479	Dennis S. Arnon and Sandra A. Mamrak / <i>On the logical structure of mathematical notation</i>
	485	Kees van der Laan / <i>Math into BLUes</i>
	502	Michael J. Downes / <i>Dialog with T_EX</i>
	510	Jackie Damrau and Michael Wester / <i>Form letters with 3-across labels capability</i>
	517	Mark A. Roth / <i>Typesetting forms with L^AT_EX</i>
	523	Peter Flynn / <i>Developing a pop-up help facility for T_EX on PCs</i>
	528	Malcolm Clark, Brian Hamilton Kelly, and Niel Kempson / <i>7 bits good, 8 bits bad or “The eight-bit blight”</i>
Graphics and Fonts	534	Friedhelm Sowa / <i>Bitmaps and halftones with BM2FONT</i>
	539	Yannis Haralambous / <i>T_EX and those other languages</i>
	549	Don Hosek / <i>Siamese T_EX: Joining dvi files at the hip and other novel applications of vf files</i>

	554	Alan Hoenig / <i>When T_EX and talk: Typesetting on curved paths and other special effects</i>
Announcements	558	Participants at TUG'91
	562	Calendar
	563	The Donald E. Knuth Scholarship for 1992
	564	GUTenberg'92, Les Diablarets, Switzerland
	570	TUG 1992 annual meeting, Portland, Oregon
Advertisements	588	Index of advertisers

Vydalo: Československé sdružení uživatelů T_EXu
vlastním nákladem jako interní publikaci
Počet výtisků: 400
Tisk: HBT-Jet PRESS Neratovice
Adresa: ČS_TUG MÚ UK, Sokolovská 83, 186 00 Praha 8