bulletin

Československé sdružení uživatelů TEXu
CS TUG

OBSAH

K druhému TEXbulletinu...

Sotva jsme trochu vydechli po semináři „**TEX, LinoTEX a Linotype**“, který jsme pořádali pro členy CSTUGu i pro širší veřejnost, je nutno dodělat druhý bulletin. Materiál jsme stále tak trochu sháněli, proto se na Vás obracím ještě jednou s žádostí o příspěvky.

Vraťme se nejprve k semináři. I když byl dlouho připravován, při samotném průběhu bylo nutno trochu improvizovat. Někteří přednášející se nemohli uvolnit na plánovanou dobu, přibyla na poslední chvíli velmi pěkná přednáška Ericha Neuwirtha, kterou jsme zorganizovali jen díky pomoci partnerské organizace DANTE a která jistě přispěla k zajímavosti celého semináře. Seminář ukázal, že kvalitní sazba pomocí TEXu např. na expoziční jednotce Linotronic je možná, pokud uživatel věnuje čas k seznámení s veřejně dostupným speciálním TEXwarem, PostScriptem (v malém rozsahu) a získá přístup k expoziční,jednotce s RIPem. Ekonomické stránky tohoto u nás dosud neběžného přístupu se zatím nepodařilo upřesnit, dle mého názoru je však zřejmé, že přechod časopisů na TEXnologii je výhodný a že má perspektivu přechodu k vysoké „knižní“ kvalitě (případné přechodné užití laserových tiskáren a klasického procesu neznamená ostatně podstatné zhoršení kvality).

K rodině časopisů tištěných TEXem přibývají další, mj. i časopisy popularizující nebo určené širší veřejnosti (Pokroky MFA, Rozhledy). Osobně věřím, že jich brzo hude podstatně více. Na Slovensku se jeví přechod „akademických“ časopisů na TEX jako oficiálně podporovaná cesta úspor a je možné, že vydání základních příruček a materiálů bude částečně subvencováno.

I když stále intenzívně jednáme, nemůžeme Vám dodat ještě speciální TEXový editor. Kromě úpravy korektorů a některých nedostatků, které — jak věřím — se podaří brzo zvládnout, se vyskytla právní obtíž u slovenského korektoru. Příslušná jednání hodláme uzavřít v září; jsme vedeni snahou o to, aby členové CSTUGu dostali dle možnosti pomocníka k přípravě vstupních textů v co možná nejlepší kvalitě.

Nová verze programového vybavení vytvořeného na bázi emTEXu Oldřichem Ulrychem již obsahuje programové prostředky, šířené pouze v rámci CSTUGu a jejich rozsah postupně poroste. V zásadě však veřejně přístupné programy, soubory a TEXware distribuujeme pro všechny, tedy i pro nečleny. Poněkud nás mrzí, využíváli na jeden individuální příspěvek pro CSTUG poskytnuté programové vybavení celé pracoviště.

S růstem CSTUGu se zamýšlíme nad zjednodušením způsobu šíření nových verzí. Uvažujeme o využití emailu a pro ty, kteří k němu nemají přístup, plánujeme zakoupení disket, které budou půjčovány (pro kolektivní členy vytvoříme kolovací systém).

K problematice editorů se dnes vracíme článkem o veřejně dostupných editorech od Miroslava Donta. Počítali jsme i se zařazením materiálu o vývoji a kvalitách populární „šestsetdvojky“, podrobná recenze by však měla vyjít v krátké době v BAJTu a navíc jsme slíbený materiál nedostali; proto jsme od tohoto plánu nuceni upustit.

První slovenské publikace o TEXu dostupné širokému okruhu čtenářů, na sebe jistě nedají dlouho čekat; zaregistrovali jsme preprint od Janky Chlebíkové. Po nezbytných úpravách vyjde i druhé vydání upraveného překladu knížky Michaela Dooba, neboť první vydání je již rozebráno. Bude mj. obsahovat i obsáhlou část o instalaci a užívání CSTEXu, tj. výše zmíněného programového balíku pro přípravu textů

v češtině a slovenštině. Použili jsme se souhlasem Oldřich Ulrycha upravené dokumentace (i když tato část bude rychle stárnout, bude užitečné ji mít v tištěné podobě — zdá, se, že je např na světě i první verze slovenského dělicího algoritmu; který by měl být zařazen do další připravované verze CSTEXu). Další publikace o TEXu jsou plánovány, připravují se. i další překlady. Poněvadž by již šlo o knížky, na něž je nutno získat. souhlas s překladem za (dle možnosti nevysoký) poplatek, je to poněkud náročnější.

Na téma grafika a TEX bylo napsáno již mnoho prací. V tomto TEXbulletinu přinášíme i české původní příspěvky k této problematice. Ta bude i předmětem semináře; který jsme oznamovali v předchozím čísle. Dnes přinášíme některá upřesnění; týkající se programu. Upozorňujeme, že jsme se rozhodli k dvoudennímu programu 9. letní školy o počítačové fyzice přidat (na základě průzkumu zájmu) jeden den. Seminář o TEXu. který tvoří samostatnou část školy, proběhne ve dnech 17.–20. září 1991. Příhlášky odešlete (adresa viz podrobnější sdělení) co nejdříve, počet míst je omezen.

Rád bych Vám na závěr, nežli se začtete do následujících stránek, popřál hezké prázdniny s TEXováním a bez něj — po prázdninách brzo na shledanou.

(*Jiří Veselý*)
e-mail: ummjv@csearn

Seminář o TEXu v hotelu Skalský Dvůr v Lísku u Bystřice pod Pernštejnem

Na závěr 9. letní školy o počítačové fyzice se bude konat v hotelu Skalský dvůr v Lísku u Bystřice pod Pernštejnem ve dnech 15.–20. 9. 1991 seminář o systému TEX. Zúčastní se ho přední pracovníci z oblasti TEXových aplikací, zvláště z německé uživatelské skupiny DANTE. Úplný seznam přednášek bude znám začátkem července; zatím lze uvést pouze následující:

Joseph Brandt, Technische Universität München: LATEX Introduction

Norbert Schwarz, Ruhr-Universität Bochum: Advanced features of TEX

Joachim Lammarsch, Universität Heidelberg: Exchange of experience in TEX environment

Friedhelm Sowa, Heinrich-Heine-Universität Düsseldorf: Integration of gralahics and pictures to TEX documents

Kromě toho se jedná s nakladatelstvím Elsevier o vyslání specialisty na využití TEXu v nakladatelské praxi a o další přednášce, která by se měla týkat vazby mezi TEXem a dalšími strukturovanými textovými systémy, zvl. SGML.

Na základě předběžných přihlášek bylo rozhodnuto prodloužit program semináře o jeden den, tj. 15.9. Vložné na seminář bude činit maximálně 150,-Kčs, stravné činí 60,-Kčs na den a ubytování v hotelu ve dvoulůžkových pokojích 80,-Kčs za den (v chatičkách 25,-Kčs za den). Přihlášky zašlete na adresu dr. Jaroslav Nadrchal, CSc. Fyzikální ústav ČSAV Cukrovarnická 10 162 00 Praha 6

Využijte výjimečné příležitosti k setkání se zahraničními kolegy z DANTE. Počet míst je omezen; proto se laskavě přihlašte co nejdříve.

(*Jaroslav Nadrchal*)

Volně dostupné editory

Volně dostupný software je v současné době v podstatě dvojího druhu. Především jsou to ty programy, které jsou opravdu volné, je možné je volně používat a volně dále šířit. Jedinou podmínkou bývá ta, že program je určen pouze k soukromému nekomerčnímu použití a neměl by být dále prodáván a to ani samostatně ani jako součást jiných softwarových balíků. Pokud se týče šíření tohoto programu, bývá vítáno, podmínkou však většinou je, že musí byt dále šířen úplný a v původním tvaru beze změny. Takovéto programy bývají v dokumentaci, která je provází, označovány jako „public domain“ nebo, free („free of charge“), často však tato specifikace bohužel chybí. Krásným příkladem public domain softwaru jsou všechny věci týkající se TEXu – zde jsou komerční programy spíše výjimkou (v našich podmínkách určitě).

Druhý způsob volně přístupných programů ‚jsou tzv. „shareware“ programy. Zde „shareware“ znamená, že program je volně přístupný, je možné jej dále volně šířit (opět úplný a v nezměněné podobě) a je možné ho volně zkoušet. Často bývá explicitně řečeno. jak dlouho je možné tento program volně zkoušet (nebo používat). Typicky bývá tato doba 30 dní; nejkratší takto určenou dobu jsem viděl 15 dní, nejdelší 90 dní (to bylo u ‚jednoho balíku různých objektů pro Turbo Pascal 5.5, kde jenom dokumentace obsahovala více než 700 Kb). Po uplynutí této doby je však uživatel žádán, aby se rozhodl a program buď vymazal nebo autorovi poslal registrační poplatek. Formulář pro registraci, kde ‚je stanovena cena a další podmínky, bývá součástí dokumentace. Program sám buď po spuštění nebo ukončení většinou upozorňuje uživatele, že je shareware a měl by být zaplacen. Po zaplacení registračního poplatku by uživatel měl většinou obdržet jednu další verzi programu, tištěný manuál apod. – tyto podmínky bývají stanoveny v dokumentaci.

V tomto článku stručně popíši některé volně dostupné editory, které lze získat pomocí e-mailu. Většinu z; nich lze nalézt v archivu SIMTEL20 (od nás přes TRICKLE@AWIWUW11), některé jsou z University of Vaasa (MAILSERV@GARBO.UWASA.FI), jeden z Heidelbergu (LISTSERV@DHDURZ1). Soustředím se na public domain editory, o shareware editorech se zmíním pouze okrajově. Nezmíním se zde ale ani, zdaleka o všech public domain editorech, které lze získat např. ze SIMTELu. Výběr budce přitom jednoznačně subjektivní a bude dán tím, že mám zkušenosti pouze s editory na PC, které mají příkazy podobné příkazům WordStaru jako např. vestavěný editor Turbo Pascalu. Navíc nebudu mluvit o editorech, které, se snaží být word processorem, tj, např. umožňují tisk přímo z editoru s různými efekty jako podtržené a tučné znaky apod. – toto vše není potřeba při psaní souborů v TEXu. Při psaní TEXových souborů je ale užitečné např. když editor má „word wrap“ (tj. automatický přechod na novou řádku po dosažení určité délky řádku), je dobré mít možnost: použití maker a vhodná je také funkce vyhledávání párových závorek. I mezi public domain editory existují některé, které všechny tyto vlastnosti mají.

Popis začnu některými malými editory, přičemž „malý“ zde rozumím ve smyslu velikosti příslušného EXE nebo COM souboru (měl bych zde spíše používat označení miniaturní, neboť naprostou většinu public domain editorů lze zařadit mezi tzv. malé editory). Mezi těmito editory je určitě na prvním místě TED – Tiny EDitor Toma Kihlkena z PC MAG z r. 1988 (VOL 7, No 19). Samotný editor je jediný soubor TED.COM, má pouze 2984 bytů a lze jej tedy považovat za ukázku miniaturizace softwaru. Spolu s TED.COM je v PC MAG k dispozici zdrojový kód. TED je „full-screen“ editor a umožňuje editovat soubory do velikosti 64 Kb. Základní pohyb kurzoru je pomocí kurzorových šipek; dále PgUp, PgDn umožňuje listování, CTRL-PgUp; CTRL-PgDn je přechod na začátek a konec souboru, Home, End přechod na začátek a konec řádku. Soubor může mít údajně libovolně dlouhé řádky; zobrazení části řádku, která je mimo obrazovku, je pomocí CTRL-šipka. Přechod na, novou řádku a lámání řádky je pomocí Enter (opětné spojení řádky pomocí Del na konci řádky). Další příkazy jsou realizovány pomocí kláves F1 – F10 (po spuštění editoru se ve spodní části obrazovky objeví pevný řádek s nápovědou o funkci těchto kláves). F2 je návrat znaku vymazaného pomocí Del. Klávesa F8 je výmaz od místa kurzoru do konce řádku, F9 výmaz celého řádku, F10 návrat vymazaných částí po F8, F9. TED umožňuje: také jednoduchou práci s bloky. Klávesa F4 je začátek práce s blokem pohybem kurzoru označíme blok (části bloku jsou celé řádky). Pomocí F5 uložíme obsah bloku do bufferu; označení bloku přitom zmizí. Později můžeme obsah bufferu zkopírovat na místa kurzoru pomocí F6 (i vícenásobné). Klávesa F3 způsobí odeslání textu označeného jako blok na tiskárnu. Výstup z editoru je buď pomocí F7 – uložení a výstup (předchozí verze souboru je uložena s koncovkou BAK) nebo F1 – výstup bez uložení poslední verze souboru. Po spuštění je editor v insert modu; přepínání do overwrite modu a, zpět je pomocí klávesy Insert. Psát je možné i znaky z rozšířené ASCII tabulky pomocí klávesy ALT a čísla (numerická klávesnice). Na řádku jsou rozmístěny tabelátory pravidelně po osmi znacích; přechod na další tabelátor je pomocí Tab. TED konfiguruje sám sebe podle použité grafické karty; přitom umí používat EGA a VGA textové mody (pokud jsou předtím nastaveny).

Existuje modifikace TEDu pod názvem TEDPLUS; rozsah kódu je tentokát 3412 bytů. V editoru TEDPLUS nemá klávesa F9 funkci vymazávání řádku; řádek je možno vymazat pomocí F8, je-li kurzor v prvním sloupci. Klávesa F9 má. nyní funkci hledání – po vyhledání je slovo odlišeno barevně a bliká. Opakované hledání je pomocí CTRL-F9. Oproti TED používá TEDPLUS na barevném monitoru barvy.

Popisoval jsem TED tak podrobně, protože se zdá. vskutku zajímavé, že editor může opravdu fungovat a mít tolik možností při tak malém rozsahu kódu (zdrojové texty jsou ovšem větší). Dále již budu stručnější.

Další z velice malých editorů je EZEDIT 1.0; COM soubor má velikost 5502 bytů. Podobně jako TED, konfiguruje EZEDIT sám sebe podle použité grafické karty a umožňuje psaní libovolného ASCII znaku pomocí ALT. Délka, řádku tentokrát nemůže být libovolná, ale je omezena 128 znaky (při načítání souboru jsou delší řádky roztrhány). Počet řádků může být přitom nejvýše 4096. Při spuštění je editor v overwrite modu a lze přepnout do insert modu. Jestliže při vsunování do řádku v insert modu dosáhne pravý okraj řádku sloupec 128, je to oznámeno zvukově. Klávesy Home. End způsobují přechod do sloupce 1 a 128. EZEDIT má funkci vyhledávání. Podobně jako TED, umožňuje EZEDIT uložení bloku do bufferu a vsunutí obsahu bufferu na místo kurzoru. Bloky mohou být přitom řádkové (blok obsahuje; celé řádky)

nebo obdélníkové (blok je obdélníková část textu – blok je určen pohybem kurzoru). EZEDIT dovoluje současnou editaci dvou souborů – přesouvání bloku do bufferu tedy umožňuje přesouvání částí textu mezi soubory. Zvláštností EZEDITu jsou tzv. draw mode a repeat mode. Draw mode umožňuje pohybem kurzoru kreslit dvojitou čáru (obsahující rohy a kříže). V repeat modu se při pohybu kurzoru (ve všech čtyřech směrech) opakuje znak, který je momentálně nad kurzorem (včetně mezery nebo znaků z rozšířené ASCII tabulky).

Rovněž editor E 1.0 (E10-EDIT) je malý editor (písmeno E je samozřejmě frekventované v názvech editorů – v tomto případě byla tato skutečnost dovedena do důsledků; v archivech se vyskytuje pod názvem E10-EDIT), jeho exekuční soubor má velikost 6461 bytů. Verze 1.0 editoru E pochází z března 1990 (několik předchozích verzí bylo v rychlém sledu vyrobeno v lednu 1990). Na rozdíl od dvou předchozích, velice jednoduchých editorů má E už trochu víc možností, přestože není o mnoho větší. Délka řádku u E je ovšem omezena na 80 znaků. Základní příkazy jsou opět stejné jako ve WordStaru. Navíc má např. označení místa a přechod na označené místo, přechod na určenou řádku, vymazání slova apod. Mimo kopírování bloku do a z bufferu je možné také kopírovat blok do a ze souboru na disku. E má word wrap; je možné nastavit pravý i levý okraj a tabelátory, je možné přepnout do autoindent modu (po přechodu na nový řádek se levý okraj nastaví podle předchozího řádku). Toto nastavení je možné provést během editace nebo „napevno“ pomocí konfiguračního programu. Mimo funkce hledání má funkci hledání s nahrazováním. Jako zvláštnost lze uvést možnost automatického zarovnávání textu na levý i pravý okraj (při nastavených okrajích). Dále E umožňuje dočasný výstup do DOSu nebo přímé spuštění některých BAT souborů. E má krátký help (klávesa F1).

Spolu s E.COM je dodáván program ECONFIG.EXE – konfigurační program pro E.COM. ECONFIG je už o trochu větší – má něco přes 18 Kb (ECONFIG je psaný v Pascalu; zdrojové texty jsou k dispozici). Umožňuje nastavit barvu informační řádky, barvu textu a dále postavení tabelátorů, insert nebo, overwrite mode, pravé a levé okraje.

CSE 3.10 je stále ještě velmi malý editor – jeho COM soubor má něco přes 13 Kb. Základní znaky editoru CSE jsou následující. Editovat lze současně až 16 souborů a využívat lze všechnu volnou paměť dostupnou DOSu. Při práci s bloky lze bloky posunovat, kopírovat, mazat. Délka řádky je omezena na 255 znaků. CSE nemá word wrap. Klávesy používané pro příkazy lze předefinovat – COM soubor je provázen souborem PROFILE.CSE, kde lze tyto i jiné změny provést. Oproti tomu, na co je většinou zvyklý ten, kdo používá pouze PC, má CSE příkazovou řádku, což je zvyk z editorů na velkých počítačích. Přesněji: obrazovka CSE je rozdělena na editační okno, příkazovou řádku, informační řádku (obsahující informace o názvu editovaného souboru, poloze kurzoru (řádek, sloupec), modu (insert overwrite) a velikostí volné paměti) a řádku pro hlášení editoru. Pomocí klávesy ESC lze přecházet z editačního okna do příkazové řádky a zpět. Příkazy pro základní pohyby kurzoru jsou stejné jako ve WordStaru, další příkazy se již liší. Příkazy lze zadávat z příkazové řádky; mnohé z nich jsou předdefinovány v připojeném makro souboru PROFILE.CSE pro použití s funkčními klávesami F1 – F10, resp. ALT-F# nebo CTRL-F# (jak jsem již poznamenal, tyto funkce lze předefinovat právě v souboru PROFILE.CSE). Např. F1 je vyvolání helpu, F2 uložení souboru atd. CSE je public domain editor, ale na rozdíl od některých jiných volných editorů je provázen pouze velice skoupou dokumentací.

V této dokumentaci se mluví o existenci souboru ADVANCED.DOC, který je však dostupný pouze registrovaným uživatelům (nejsou mi známy podmínky registrace CSE). Tento soubor má obsahovat podrobný popis všech příkazů a také pravidla pro výrobu makro souborů jako je PROFILE.CSE (při editaci je možné volat různé makro soubory a tím měnit různá nastavení a předefinovat klávesy).

Další editory z těch public domain editorů, které znám, jsou většinou již větší. Popíši přitom jenom některé z nich a začnu tím, který se mi nejvíce líbil (z public domain editorů) – jak jsem již poznamenal, výběr popisovaných editorů je silně subjektivní. Tento editor se jmenuje EDWIN.

EDWIN je vedlejší produkt firmy TurboPower Software a byl napsán pomocí Editor Toolboxu (v dokumentaci se říká, že vznikl jako koníček; že je public domain, a jsou tam stručně vyjmenovány komerční výrobky firmy); je označen jako programátorský editor. Základní charakteristiky jsou, že má systém.menu, připojený help, vícenásobná okna, přesouvání bloků mezi okny, systém maker, možnost dočasného výstupu do DOSu, velikost editovaných souborů je omezena pouze dostupnou pamětí; příkazy jsou z WordStaru, ale klávesnice je plně konfigurovatelná.

Hlavní soubor je EDWIN.COM (61 Kb). K provozu jsou však nutné ještě tyto soubory: EDWIN.000 (overlay soubor), EDWIN.KEY (definice klávesnice), EDWIN.MSG (hlášení při editaci). Další tři soubory mohou ale nemusí být přítomny: EDWIN.HLP (help), EDWIN.ERR, (chybová hlášení), EDWIN.MAC (základní makra). Dále je k dispozici soubor EDWINST.COM (46 Kb), což je instalační program. EDWINST má také menu, kde jsou nabízeny tyto základní možnosti: Obrazovka („kontrola sněžení“, volba barev a typu kurzoru), Příkazy (předefinování klávesnice), Volby (cesty k pomocným souborům, nastavení editoru při spuštění atd.), Makra (editor maker).

Pro kurzor lze volit dvě možnosti. Buď plný neblikající kurzor v libovolné barvě nebo obyčejný blikající kurzor, který se mění na tučný nebo tenký podle toho, zda editor je v insert nebo overwrite modu.

Pokud je předefinována klávesnice, EDWINST automaticky generuje soubor EDWIN.HLP, který odpovídá tomuto předefinování (v tomto souboru jsou také napsány názvy základních maker).

Při editaci maker je možné editovat buď již existující makra nebo přidávat makra do existujícího souboru maker nebo vytvářet nový soubor maker. Makra jsou pojmenovaná a jejich přiřazení ke klávesám je možné při instalaci v části Příkazy. Editor maker má omezený, ale plně postačující počet příkazů. Délka makra je omezena 255-ti znaky (znak z rozšířené ASCII tabulky se počítá jako dva znaky).

Jak jsem již poznamenal, EDWIN má systém menu. Volbu souboru k editaci je možné provést napsáním jména souboru nebo výběrem z adresáře nebo při spouštění je možné do příkazového řádku napsat názvy až tří souborů (včetně cest). EDWIN může pracovat zároveň až se šesti okny. Každé okno má informační řádku, která obsahuje následující informace: název souboru, číslo aktuální řádky a sloupce, počet znaků od začátku souboru až po kurzor, nastavení pravého okraje (pokud je editor ve word wrap modu), indikace, zda je editor v insert nebo overwrite modu, indikace autoindent modu, indikace změny souboru. Jak jsem již řekl, délka souboru je omezena dostupnou pamětí (v dokumentaci je popsána interní potřeba paměti pro každý řádek atd.). Délka řádku může být maximálně 999 znaků; při načtení souboru jsou delší řádky lámány. Maximální počet řádků je 32767.

Příkazy jsou převzaty z WordStaru (přesněji řečeno z editoru Turbo Pascalu) a nebudu je zde popisovat. Jenom jako zajímavost mohu např. uvést, že příkaz pro přechod na danou řádku umožní buď přejít na řádku s daným číslem nebo přejít o daný počet řádků dopředu či dozadu, pokud před číslo napíšeme + nebo −.

Mimo funkce vyhledávání a vyhledávání s nahrazováním má EDWIN funkci vyhledávání s následnou aplikací makra (zde se makrem myslí momentálně definované makro obsahující některé jednoduché příkazy). Ve volbách při vyhledávání existuje mimo jiné možnost vyhledávání pouze v označeném bloku. Pokud se týče práce se soubory a bloky, mimo zápisu bloku do souboru má EDWIN ještě možnost připojení bloku k existujícímu souboru (při zápisu bloku do souboru je možné volit název souboru PRN nebo LPT# k odeslání souboru na tiskárnu).

Jak jsem již poznamenal, EDWIN má word wrap; mimo to má možnost dodatečného přeformátování odstavce.

EDWIN byl psaný pomocí Editor Toolboxu. Existuje několik dalších public domain editorů psaných tímto způsobem. Jeden z nich je velice hezký editor TPE – Turbo Powered Editor v. 2.0. Předchozí verze TPE byly psány firmou TurboPower Software pro vlastní použití; tato verze je public domain (některé firmy nebo samostatní programátoři využívají svého public domain softwaru k propagaci svých komerčních produktů nebo třeba jenom jména firmy). TPE patří mezi tzv. programátorské editory, tj. editory určené k psaní zdrojových textů programů; dá se ovšem použít i k psaní jakéhokoliv jiného textu. Jako programátorský editor má TPE oproti obyčejnému textovému editoru přímý přístup ke kompilátoru, assembleru a debugeru (které ovšem nejsou součástí TPE). Mimo to, že TPE připomíná editor integrovaný v Turbo Pascalu (příkazy WordStaru jsou samozřejmostí), má. tyto další základní vlastnosti (neuvádím zde všechny):

- Na obrazovce může být až 8 oken různých velikostí pro editaci různých souborů (nebo různých částí téhož souboru).
- Možnost editace souboru až clo velikosti volné paměti.
- Přenášení dat mezi okny.
- Plně přestavitelná klávesnice.
- Word wrap, pevné a pohyblivé tabelátory, autoindent mode, příkazy pro zarovnávání a formátování textových bloků.
- Vyhledávání párových závorek.
- Čtyři různé možnosti kreslení čar.
- Systém maker.
- Připojený help.

V samotném manuálu je uvedeno, že TPE v porovnání s nejlepšími komerčními programátorskými editory má pouze tři nevýhody – neumožňuje editovat větší soubor než se vejde do paměti, navracení vymazaného textu je jednoduché a také systém maker je pouze jednoduchý.

V archivech dostupných e-mailem existuje více public domain nebo shareware editorů psaných tímto způsobem a připomínajících editor Turbo Pascalu. Jeden z nich je AE – Another Editor v. 1.4, který byl umístěn v SIMTELu teprve nedávno (realizace editoru je datována v březnu 1991). AE je public domain, je psaný v Turbo Pascalu 5.5 a je dostupný spolu se zdrojovým textem. Spolu se samotným editorem

jsou v příslušném balíku dva další programy (opět se zdrojovými texty); jeden na roztrhávání velkých souborů (pro možnost postupné editace souborů, které se nevejdou do paměti), druhý na opětné spojování. Jelikož AE má příkazy WordStaru a vestavěný help, je možné jej okamžitě používat bez čtení manuálu (který je ostatně poměrně stručný). Má word wrap a jednoduchý systém maker (10 maker). Jeho velikost je asi 42 Kb.

Za zmínku stojí ještě jeden editor psaný pomocí Turbo Editor Toolboxu. Je to AHED – Ad Hoc EDitor v. 0.6 z r. 1986. Má tu zvláštnost, že umožňuje editovat velmi dlouhé řádky – v dokumentaci se uvádí, že je možné editovat řádky až do délky 32767 znaků. Editor automaticky zvýrazňuje párové závorky. Dále má word wrap, autoindent mode a help, přičemž je tvořen jediným souborem o 64 Kb.

Velice solidní editor pro psaní textových souborů, zvláště souborů v TEXu je DOCEDIT. Vyznačuje se především velice dobrou dokumentací, avšak v němčině. Editor byl vypracován jako diplomová práce na univerzitě v Karlsruhe. Je poměrně komfortní. Má bohatý systém menu, podrobný help a dokonce připojený „tutor“ – všechno německy. Manuál je psaný v TEXu a je k dispozici již jako DVI soubor. Má poměrně dobrý systém maker; připojená předefinovaná makra zjednodušují psaní textu v LATEXu.

Některé z předchozích editorů byly méně, některé více komfortní, měly k dispozici méně nebo více funkcí, ale základ – totiž příkazy WordStaru – byl v podstatě u všech stejný. Druhou velkou skupinou editorů, která je do značné míry odlišná od skupiny předchozí, je skupina tvořená editory podobnými EMACSu. Původní tvar editoru EMACS napsal kolem roku 1970 R. Stallman v MIT pro Digital Equipment. Postupně vzniklo značné množství editorů podobných EMACSu a velké množství z nich je public domain. Tyto editory nebyly původně vůbec psány pro PC, ale většinou jsou schopny práce pod různými operačními systémy na různých typech počítačů a existují tedy i verze pro PC a MS-DOS. Možnost používání téhož editoru (téhož ve smyslu stejného vzhledu a chování) na různých systémech je jedna z výhod editorů typu EMACS.

Uživatel zvyklý pracovat s editory s příkazy WordStaru, který navíc nemusí pracovat na jiném počítači než na PC, nebude pravděpodobně přecházet na EMACSové editory, neboť na první pohled se systém příkazů dost liší (a volba a používání editoru je především otázka zvyku). Mimo existence verzí pro různé operační systémy mají ale editory typu EMACS některé další pozoruhodné vlastnosti. V archivu SIMTEL20 existuje několik různých EMACSových editorů schopných práce pod MS-DOS. Dva z nich tam mají dokonce svůj vlastní adresář – MicroEMACS a FREEMACS (vlastní adresář má v SIMTELu např. také velice hezký editor QEDIT který není typu EMACS; ten je však shareware – 55 dolarů). Zmíním se zde stručně alespoň o editoru MicroEMACS. Verze MicroEMACSu umístěná v SIMTELu je označena jako 3.10 – v současné době je např. v GARBO.UWASA.FI dostupná testovací beta verze 3.11 realizovaná v březnu 1991; v květnu by měla být dostupná konečná verze 3.11 (předpokládám, že dostupná už je, ale nevím, zda již je umístěna v některém archivu v síti).

MicroEMACS je schopný provozu např. na počítačích VAX 70 (operační systémy OS, UNIX, VMS), SUN, HP9000, IBM PC, HP150 atd. (je k dispozici zdrojový kód v C a pro některé systémy, jako např. PC, jsou k dispozici také exekuční soubory). Je public domain, ale je možné také získat komerční licenci připouštějící připojení

k jinému komerčnímu softwaru. Pokud uživatel sám překládá zdrojový kód (na PC to lze např. pomocí Turbo C v. 2.0), je možné připojit hlášení v jiných jazycích než angličtině. Jsou dostupná hlášení např. německá, francouzská, španělská, italská a holandská, (v dokumentaci pro beta verzi 3.11 jsem viděl zmínku o tom, že existuje verze MicroEMACSu pro IBM 5550, což je prý japonská verze IBM PC umožňující editování vedle latinky v různých typech japonského písma a autoři hledají dobrovolníka pro překlad hlášení do japonštiny).

Jak jsem již poznamenal, příkazy ze u EMACSových editorů na první pohled liší od příkazů jiných, běžně používaných editorů na PC. Mimo klávesy CTRL se užívá tzv. META klávesa, která na klávesnicích bez takto označené klávesy je nahrazena klávesou ESC (u některých EMACSových editorů klávesou ALT). Užívání jiných příkazů je ovšem jenom otázka zvyku – základní funkce jsou ale stejné jako u jiných editorů (ostatně např. na PC lze k základním pohybům kurzoru užít šipky, k listování klávesy PgUp, PgDn a navíc lze u EMACSových editorů funkce kláves snadno předefinovat). V některých funkcích se však EMACSové editory liší od jiných editorů na PC nejenom po vnější stránce. Např. funkce vyhledávání má některé zajímavé vlastností. Kromě základní funkce vyhledávání a vyhledávání s nahrazováním daného řetězce je možné např. hledat daný řetězec jenom na začátku nebo na konci řádku, je možné hledání řetězce, kde na určených místech mohou být libovolné znaky nebo znaky v určitém daném rozsahu (např. jenom samohláska nebo jenom souhláska), řetězce, kde některé znaky se mohou a nemusí vyskytovat; řetězce, které nejsou následovány písmenem a mnoho dalších možností a různé kombinace těchto možností. Toto je možné v tzv. „magic modu“. Přepínání do různých modů je přitom víc. Vedle insert a overwrite modu, wrap modu (word wrap), asave modu (automatické ukládání po napsání určitého počtu znaků), umožňuje MicroEMACS např. následující mody (některé mody mohou být nastaveny zároveň). View mode umožňuje pouze prohlížení souboru, ale zabraňuje jeho změnám. Cmode usnadňuje psaní programů v C (je automaticky nastaven u .c nebo .h souborů); mimo jiné v cmodu pracuje autoindent, avšak s výjimkou řádky, kdy předchozí řádka končí otevírající závorkou. Při napsání uzavírajícího výrazu), nebo kurzor automaticky ukáže na příslušnou párovou závorku (nebo) a zase se vrátí (pokud párová závorka je momentálně na monitoru). Pokud je editor v crypt modu, je při zápisu soubor zakódován a při čtení dekódován; uživatel zadává kódovací klíč. V zakódovaném souboru se nevyskytují znaky z rozšířené ASCII tabulky a je proto vhodný k posílání e-mailem. Kódování a dekódování v MicroEMACSu je přitom strojově nezávislé.

EMACSové editory mívají neobyčejně zajímavý systém maker. Mimo textová makra mají především možnost maker nazývaných někdy funkční makra, která mohou být tvořena různými příkazy. Těchto příkazů bývá velké množství a je možné je psát podle jistých, přesně stanovených pravidel, která vlastně tvoří určitý programovací jazyk. V základních příkazech editoru jsou různé EMACSové editory většinou téměř totožné, ale liší se jazykem maker. Jazyky maker mají u některých EMACSových editorů dokonce své názvy (např. u FREEMACSu je to MINT); v některých případech vedle editoru je samostatný překladač maker a editor potom používá už přeložená makra. Např. o FREEMACSu se mluví jako o programovatelném editoru – samotný editor je poměrně malý soubor (21 Kb), který obsahuje pouze tzv. primitivní příkazy a interpret maker. Další příkazy a funkce (a např, také definice klávesnice) jsou obsaženy v souborech maker.

Mimo MicroEMACSu a FREEMACSu jsou v SIMTELu dostupné například tyto EMACSové editory:

MG2A – malý EMACSový editor, který se od ostatních liší tím, že nemá jazyk pro funkční makra; má však bohatší možnosti při vyhledávání než třeba MicroEMACS. Je k dispozici manuál psaný v LaTeXu.

ME-CD je označen jako malý EMACSový editor (i když příslušný EXE soubor má přes 71 Kb). Má bohatý jazyk funkčních maker pod názvem programovací jazyk MUTT a samostatný překladač maker. K dispozici jsou zajímavé ukázky maker a to od jednoduchého makra pro výpočet faktoriálu přes makro pro editaci obrázků až po kalkulačku a hru Gomoku.

JOVE – Jonatan's Own Version of Emacs – další z relativně malých EMACSových editorů.

Všechny zde uvedené EMACSové editory jsou public domain.

Stejně jako existuje mnoho public domain editorů, velké množství editorů je shareware. Tyto editory už bývají větší, často mají mnoho funkcí jako word processory a většinou (ne vždy!) bývají o třídu lepší než public domain editory. Už jsem se zmínil o shareware editoru QEDIT. Dále mne zaujal editor TE25 – Technical Editor v. 2.5, který se vyznačuje především tím, že umožňuje editovat větší soubory než se vejdou do paměti. Údajně umožňuje editovat soubory až do velikosti 32 Mb. Začátek práce s velkým souborem je relativně dlouhý, avšak potom už je práce stejně rychlá jako by soubor byl celý v paměti (např. přechod od začátku na konec souboru pomocí CTRL-PgDn je okamžitý). Velikost EXE souboru je 80 Kb, cena 40 dolarů.

Další shareware editory zde nebudu popisovat – nejspíše bych je jenom vyjmenoval. Zájemcům doporučuji prohlédnout si adresář MSDOS.EDITOR v SIMTELu. Pouze bych poznamenal, že kdo má chuť vidět velkolepé počítačové barevné show, má možnost si ze SIMTELu nechat poslat např. demonstrační verzi (která je zároveň funkční!) ME400A (Multi-Edit). Je to skutečně krásná ukázka dobrého profesionálního editoru.

Nakonec se zmíním o existenci public domain nebo shareware spell checkerů, tj. programů na kontrolu pravopisu. Stejně tak jako neznám český public domain nebo shareware editor, neznám ani český public domain spell checker, budu tedy mluvit o anglických spell checkerech.

Pokud se týče užití spell checkeru na soubory psané v TeXu, je třeba poznamenat, že potíže způsobují různé příkazy TeXu, které budou hlášeny jako chyby (použití spell checkeru je možné, ale je zdlouhavější). Nevím, zda existuje nějaký spell checker, který by přímo tento problém řešil. Existuje však jednoduchý a krátký program s názvem UNRETEX (public domain), který ze souboru odstraní všechny příkazy (začínající \) a později (po opravě spell checkerem) je navrátí zpět.

Neznám public domain editor, který by v sobě obsahoval spell checker (některé shareware editory ho mají). Editoru se spell checkerem se však blíží FREEMACS. Spolu s FREEMACSem lze získat Clarkson Speller, což je rezidentní program, který (je-li instalován před spuštěním FREEMACSu) lze volat z FREEMACSu. Dále např. spolu s MicroEMACSem lze získat MicroSPELL, což je spell checker, který lze používat buď samostatně (výstupem bude soubor obsahující seznam chybných slov) nebo jej lze používat spolu s MicroEMACSem, který umožní opravování (dočetl jsem se, že

MicroSPELL lze mimo jiné také použít na IBM 370 pod operačním systémem CMS spolu s editorem XEDIT).

Zastavím se zde stručně u třech public domain spell checkerů, které pracují jako samostatné programy. Nejmenší z nich je EZSPELL (samotný program má název EZS). Soubor EZS.COM má velikost 18 Kb, slovník pod názvem EZSPELL.DCT (v původním stavu) má velikost 54 Kb. EZS pracuje tak, že nejprve přečte soubor, který má kontrolovat, oznámí počet slov a řádků, které mají být zkontrolovány, potom čte slovník. Oznámí kolik slovníkových slov bylo čteno a počet „neznámých“ slov. Nakonec podá přehled neznámých slov a nabízí tyto možné akce – přidání slova do slovníku, oprava slova v původním souboru, označení slova pro pozdější opravu, ignorování chyby, možnost přechodu na předchozí slovo (např. když jste si rozmysleli předchozí opravu).

EZSPELL nepřipouští znaky z rozšířené ASCII tabulky, délku slova připouští do 40-ti znaků, délku řádku do 255-znaků. Velikosti souborů připouští podle velikosti dostupné paměti, přičemž na jedno slovo potřebuje 130 bytů (počítá se počet různých slov). Velikost slovníku údajně není omezena.

Program LSPELLA je psán v Turbo Pascalu v. 3. LSPELLA.COM má velikost necelých 39 Kb. Slovník PABSP2.DCT má velikost 217 Kb a podle manuálu obsahuje 15000 slov a jelikož LSPELLA umí pracovat s příponami, skutečný počet slov je asi trojnásobný. Mimo to LSPELLA umožňuje pracovat s krátkým uživatelským slovníkem až do 1500 slov, kam uživatel ukládá další slova. LSPELLA pracuje se slovy nejvýše o 20-ti znacích, slova kratší než 3 znaky vynechává. Nepřipouští znaky z rozšířené ASCII tabulky.

TSPELL je spell checker Tima Salmiho (jméno, na které narazí jistě každý, kdo se zajímá o public domain programy pro PC). Jak autor uvádí, program je dále ve vývoji; v současné době je k dispozici verze 2.4 (datovaná 17. 3. 1991). V dodávaném balíku (TSPELL24.ARC) kromě slovníku a samotného spell checkeru je dále editor slovníku, program na připojování slov do slovníku a program na počítání frekvence slov. Nová slova se do slovníku přidávají pomocí editoru slovníku, nikoli při běhu spell checkeru; je možné navíc také slova, ze slovníku odstraňovat. Obsah slovníku je omezen počtem 22800 slov. Použití editoru slovníku je poměrně pomalé – navíc je tu tedy program na rychlé přidávání slov. Spell checker pracuje v obráceném pořadí než např. EZSPELL – nejprve čte slovník a potom kontroluje slova v daném souboru. Délka kontrolovaných slov je nejvýše 16 znaků, délka kontrolovaného textu není omezena (doporučuje se nicméně delší soubory kontrolovat po menších částech).

Nakonec se zmíním o existenci jednoho shareware spell checkeru. Je to SS – ShareSpell v. 2.3. Příslušný balík obsahuje mimo dokumentaci soubory SS.EXE – spell checker, ACROP.DIC – slovník a DICMAN.EXE – program na práci se slovníkem (doplňování apod.). V průběhu kontroly se program zastavuje u neznámých slov, nabízí možný správný tvar slova a nahrazení tímto tvarem, ruční opravu, přidání slova do slovníku, ignorování nebo označení slova pro další opravu. Dodávaný slovník má 250 Kb a údajně obsahuje 112000 slov. Cena za registraci je 20 dolarů.

(*Miroslav Dont*)

Dostupnost TEXu v počítačových sítích

Tento článek si klade za cíl seznámit uživatele TEXu s celosvětovými počítačovými sítěmi a přístupem k souborům pomocí těchto sítí. Podává přehled vybraných uzlů sítě, v nichž jsou umístěny soubory s materiály o TEXu. Uvádí i způsob práce s těmito uzly.

Od konce roku 1990 pracuje v Praze v Oblastním výpočetním centru vysokých škol československý uzel počítačové sítě EARN. EARN je zkratka plného názvu European Academic Research Network, tedy sítě propojující evropská vysokoškolská a výzkumná pracoviště. V síti EARN jsou zapojeny nejčastěji sálové počítače IBM, lze se však setkat i s VAXy a jinými počítači. Jako členská země EARNu platí ČSFR roční příspěvky na provoz sítě. Velikost příspěvku se odvozuje od dosaženého národního důchodu. Kromě toho je třeba uhradit pronájem komunikační linky mezi Prahou a Linzem, kde je umístěn rakouský národní uzel. K národnímu uzlu se mohou samozřejmě připojovat i další počítače. Náš národní uzel se jmenuje `csearn`; je k němu připojeno několik dalších počítačů umístěných v Praze na pracovištích vysokých škol a Akademie věd.

Nemohu opomenout ani síť EUNET, která je v ČSFR rovněž dostupná. Podle mých informací je provozována na počítačích PC pod operačním systémem UNIX. Národní uzel jev Bratislavě a je připojen na Vídeň. Další uzly jsou v Praze na Vysoké škole chemicko-technologické a na Vysoké škole ekonomické. Data jsou přenášena po komutovaných telefonních linkách a na rozdíl od sítě EARN není spojení trvalé, ale operátoři sítě jej navazují zhruba každé dvě hodiny.

Většina uživatelů TEXu pochází z akademického prostředí a k síti EARN má nebo postupně získává přístup. Správný fanoušek TEXu začne brzy v síti vyhledávat materiály o TEXu, nové verze programů, časopisy a diskusní fóra věnovaná TEXu. Tomu by měl napomoci i tento příspěvek. Neuvádím zde filozofii operačních systémů sálových počítačů ani konkrétní příkazy pro práci s nimi. Tomu jsou věnována školení uživatelů. Základní informace o práci v síti lze najít i v [2]. Věnuji se spíše možnostem přístupu do jiných sítí než je EARN. Můj příspěvek by měl posloužit jako vstupní informace pro ty, kteří se věnují i něčemu jinému než jen pátráním po zajímavých materiálech kdekoliv v síti.

Práce v síti

Jednotliví uživatelé sítě si mezi sebou mohou zasílat elektronické dopisy (e-mail). Výhody jsou nasnadě. Dopis si jako textový soubor v klidu připravím a odešlu adresátovi. Pokud adresát právě pracuje na počítači, operační systém mu oznámí, že přišla nová pošta. Ve vhodném okamžiku si ji adresát může vyzvednout a přečíst, případně vytisknout. Pokud adresát zrovna s počítačem nepracuje, došlá pošta se neztratí, ale uschová pro pozdější vyzvednutí. Dopisy si mezi sebou mohou vyměňovat uživatelé téhož počítače, ale samozřejmě i uživatelé počítačů zapojených v odlišných sítích. Doručení dopisu odkudkoliv kamkoliv trvá za běžných okolností několik minut, nejvýše desítky minut. Proto se někdy normálnímu poštovnímu styku říká hlemýždí pošta.

Některé počítače (uzly) počítačové sítě poskytují služby všem uživatelům sítě. Vžil se pro ně název servery (dle [2] obslužné stanice). Většinou se jedná o file servery

(obslužné stanice souborů). Pod pojem file server zde zahrnujeme jak dostatečný diskový prostor, ve kterém jsou uloženy soubory určené uživatelům sítě, tak samozřejmě programové vybavení takového počítače. To musí být schopno přijmout náš příkaz a provést jej. Typickými příkazy pro server jsou zjištění obsahu nějakého adresáře a požadavek na přenos některého souboru ze serveru na domovský počítač uživatele.

Jiným druhem služeb jsou elektronické časopisy a fóra. Elektronické časopisy vycházejí pravidelně nebo nepravidelně v okamžiku, kdy se shromáždí dostatek příspěvků. Lze se setkat s periodou dvě vydání za týden až jedno vydání za čtvrt roku. Zájemce si musí časopis objednat. Žádné dolarové vyúčtování samozřejmě neobdrží, pouze je zanesen do distribučního seznamu. Příspěvky do elektronického fóra (diskusního klubu) se okamžitě rozesílají všem zájemcům. Objevují se zde zprávy typu: Můžete mi někdo poradit s ...? V časopisech jsou obvykle šířeji zaměřené články. Zaregistrovat se jako čtenář časopisu nebo účastník fóra lze buď příkazem `subscribe` nebo je třeba vydavateli zaslat prosbu o zařazení do seznamu uživatelů, obvykle i s uvedením jména a adresy pracoviště.

Většina serverů sítě EARN umí přijímat jak interaktivní příkazy, tak i dopisy. Text dopisu je tvořen řádky obsahujícími jednotlivé příkazy pro server. Na síti BITNET/EARN mají servery jméno `listserv`. File server provozovaný na uzlu `csearn` má síťovou adresu `listserv@csearn`. Na tuto adresu lze zasílat dopisy obsahující příkazy pro server. Lze použít i interaktivní příkaz

`tell listserv at csearn`
příkaz

Síť EARN je propojena i s jinými světovými sítěmi. Sít BITNET (americký kontinent) splývá z uživatelského pohledu se sítí EARN. Uživatel mnohdy ani nerozpozná, do které ze sítí ten který uzel patří. Do sítí JANET (Velká Británie) a Internet (americký kontinent a postupně i Evropa) se lze dostat jen prostřednictvím pošty.

Na servery pracující v jiných sítí se můžeme dostat poštou adresovanou na `mailserver@...`, `texserver@...` a podobně. Každý ovšem může být ovládán jinými příkazy. Zde platí jediné doporučení: zkusit serveru poslat příkaz help. Jiné uzly povolují anonymní přístup pomocí FTP (File Transfer Protocol). To se týká sítě Internet. Protokol FTP umožňuje interaktivní práci s takovými uzly. Uživatel se musí přihlásit, pak se může volně pohybovat v jemu dostupných adresářích a přenášet soubory. Na závěr se odhlásí. Ze sítě BITNET/EARN nemůžeme FTP protokol používat přímo. Na adrese `bitftp@pucc` existuje ale služba, která nám přístup k FTP zprostředkuje. Stačí na uvedenou adresu zaslat dopis, který obsahuje jednotlivé příkazy FTP. Například:

```
ftp rusmv1.rus-uni-stuttgart.de
dir
cd soft/tex
dir
quit
```

Odpovědi na příkazy zasílané serverům mimo síť BITNET/EARN podle okolností přijdou zpět za 5 minut, ale také až příští den. To závisí na mnoha okolnostech,

mezi něž patří momentální zatížení serverů, sítě, den v týdnu, apod. Server někdy rozesílá soubory jen v mimopracovní době dotyčného pracoviště.

Jak zacházet s přenesenými soubory

Při přenosu textových souborů na náš domovský uzel a z něj potom na PC obvykle nenastávají problémy. Lze se setkat jen se záměnou znaků tilda (~) a caret (^) v souborech ze serveru v Astonu.

U binárních souborů (EXE, ZIP, ARC, ...) musíme být obezřetnější. Pokud server poskytuje volby typu souboru (textový, binární), musíme zvolit binární soubor (obvykle příkaz binary). Vyžádaný soubor může přijít jako pošta (MAIL) nebo jako soubor (RSCS). V případě souboru máme o starost méně, jen při přenosu na PC je třeba nastavit program Kermit do režimu přenosu binárních souborů.

Pokud nám server umí posílat soubory jen v MAIL formátu, je v případě binárních souborů situace složitější. Poštovní soubor ale může obsahovat jen tištitelné znaky. Pro takový případ servery používají algoritmy, které binární soubor transformují do obyčejného textového souboru. To se ovšem děje za cenu nárůstu jeho velikosti. Zpětnou transformaci lze provést na domovském uzlu nebo až na PC. Transformační programy jsou běžně dostupné buď na dotyčných serverech, v nouzi na populární adrese `trickle@awiwuw11`. Různé servery používají různé transformační algoritmy; někdy jich server umí i několik. O konkrétních možnostech serveru a způsobech, jak si transformaci vyžádat, se lze informovat v popisu příkazů jednotlivých serverů.

Nejčastěji používané mechanismy jsou BtoA (Binary to Ascii) a zpět AtoB. UUencoding s dekódovacím programem UUDECODE, XXencoding s dekódovacím programem XXDECODE jsou další rovněž používané metody. Z došlého dopisu je třeba editorem odstranit služební záhlaví a pak jej dekódovacím programem převést do původní podoby. Pokud je zakódovaný soubor příliš rozsáhlý, může jej server z důvodu efektivního využívání přenosové kapacity sítě rozdělit do několika samostatných dopisů. V takovém případě je nutné jednotlivé dopisy ve správném pořadí sestavit dohromady.

Některé zdroje TEXware

V sítích lze nalézt:

- Zdrojové soubory TEXu, Metafontu a řady pomocných programů v jazyce `WEB`.
- Změnové soubory (change files) pro přizpůsobení jednotlivým počítačům a jejich operačním systémům.
- Zdrojové soubory fontů (písmem Computer Modern počínaje a šachovými figurkami a notami konče).
- LATEX, nejrůznější styly,
- DVI-drivery i ve zdrojovém tvaru.
- Konverzní programy do/z TEXu.
- Dokumentaci ke všemu již zmíněnému.
- Elektronické časopisy a fóra věnovaná TEXu.

- Vše z public domain a shareware oblasti, co je věnováno TEXu a není uvedeno v předchozích bodech.

Z mnoha serverů poskytujících téměř cokoliv z oblasti TEXu jsem vybral jen několik pro nás nejzajímavějších; inspiroval jsem se v [1] a něco zestručnil, něco naopak přidal. Doufám, že žádný z kategorie podstatně zajímavých serverů při pátrání v sítích neunikl mé pozornosti.

`labrea.stanford.edu` (StanfordUniversity)
Přístup: FTP

Zde je uložen TEX tak říkajíc přímo od pramene. Na serveru je uloženo velké množství původních souborů. K dispozici jsou i materiály pro implementaci TEXu pro UNIX.

`hstserv@hearn` (Katholiecke Universiteit Nijmegen)
Přístup: interaktivně, mail

Uzel je součástí sítě `earn`. Zde si lze přeplatit časopis `texmag-l`. Vychází zhruba čtyřikrát do roka a přináší obecněji platné informace o TEXu a jeho budoucím vývoji.

`hstserv@dhdurzl` (Heidelberg University Computer Centre)
Přístup: interaktivně, mail

Uzel je součástí sítě earn. Na tomto uzlu jsou soustředěny materiály pracovní skupiny DANTE pro oblast německy hovořících zemí. Lze začít pracovat např. příkazem `get index` nebo `get index dante-l`.

`listserv@frulm11` (École Normale Supérieure, Paris)
Přístup: interaktivně, mail

Uzel je součástí sítě earn. Kolem tohoto uzlu se soustředila skupina GUTenberg orientovaná na frankofonní problematiku TEXu.

`texserver@tex.ac.uk` (Aston University, Birmingham)
Přístup: mail, hlemýždí pošta

V Astonu je nejrozsáhlejší evropský archiv TEXu. Kromě programového vybavení pro nejrůznější počítače a operační systémy jsou zde i časopisy TEXhax, UKTEX a TEXline. Server je provozován na počítači VAX s operačním systémem VMS. Oproti informacím uvedeným v helpu může již první řádek dopisu obsahovat příkaz pro server. Ve specifikacích souborů uložených v podadresářích je třeba dodržovat konvence VMS — jména adresářů se oddělují tečkami a celé uzavírají do hranatých závorek. S výhodou lze používat specifikace zahrnující více adresářů. Například `directory [texarchive.latex...]` dodá abecedně setříděný seznam všech souborů z podadresáře `latex` a jemu podřízených adresářů. V základním adresáři jsou udržovány soubory `00last7days.files` a `00last30days.files` se změnami za poslední období.

`fileserv@shsu` (Sam Houston State University)
Přístup: mail

Server s elektronickým fórem `info-tex`. Soubory serveru jsou soustředěny do problémově orientovaných balíků, které lze získat jediným příkazem sendme *jméno balíku*. Server je součástí BITNETu, ale umí zpracovávat pouze příkazy zasílané mu poštou.

rusmv1.rus.uni-stuttgart.de (Technical University Stuttgart)Přístup: FTP

Na tomto serveru jsou v adresáři `soft/tex/machines/pc/emtex` a jeho podadresářích uloženy jednotlivé části emTEXu, který je v současné době zřejmě nejpropracovanější implementací TEXu pro public domain oblast osobních počítačů. Za zmínku stojí i program `mfjob` pro podporu generování ucelených souborů fontů.

`mail-server@rusmv1.rus.uni-stuttgart.de` (Technical University Stuttgart)
Přístup: mail

Adresováním pošty na `mail-server` se vyhneme zprostředkovávanému protokolu FTP. Příkazy pro `mail-server` jsou ale jiné, než známe pro FTP.

`ymir.claremont.edu` (Claremont Colleges Networking, USA)
Přístup: FTP

Jeden z populárních serverů USA s materiály o TEXu. Dalším oblíbeným serverem je `sun.soe.clarkson.edu`, rovněž s přístupem pomocí FTP.

`bitftp@pucc` (Princeton University)
Přístup: mail

Za pomoci tohoto serveru mohou uživatelé sítí; které neumějí FTP protokol, získat zprostředkovaný přístup k FTP serverům.

`trickle@awiwuw11` (Wirtschaftsuniversität Wien)
Přístup: mail, interaktivně

Populární server s bohatou zásobou programového vybavení public domain a shareware. V adresáři `<msdos.starter>` jsou základní konverzní programy, v adresáři `<mmsdos.tex>` je i několik souborů týkajících se TEXu a emTEXu.

Literatura

[1] Peter Flynn: Network Sources of TEXware.

[2] Pavel Vachek: Bitnet userhelp. Překlad stejnojmenného manuálu od Christophera Condona.

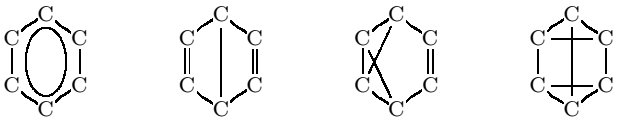
(*Martin Bílý*)
e-mail: `tepbm@csearn`

TEX a chemické vzorce

V tomto krátkém příspěvku bych chtěl v krátkosti popsat soubor `chemstructure.tex`, který dává (nepochybně ne jedinou) možnost sázet plain TEXem chemické strukturované vzorce. Předem se omlouvám za možné nepřesné použití chemického názvosloví.

Toto makro vytvořil Dr. Michael Ramek z Technische Univrsität Graz a jeho verze 1.0 z roku 1987 je, jakožto public-domain software, k dispozici v MU UK Praha. Celý soubor `chemstru.tex` má v zaarchivované podobě 25 kB a obsahuje i krátký popis použití samotné konstrukce .. Makro nevyžaduje dodatečné speciální fonty.

Makro vvytváří hbox, tedy jej lze okamžitě vysázet nebo uschovat pro další použití. Základna tohoto boxu je shodná se základnou prvního atomu zadaného makra, další rozměry se dopočítávají. Poznamenejme, že rozměry výsledného boxu se počítají dvakrát; poprvé pro spočítání potřebného prostoru (bez sazby) a podruhé se správným umístěním včetně sazby. První běh lze potlačit (v případě, že začínáme atomem, který je nejvíce vlevo) uvedením slova jakožto prvního argumentu makra .. Použitá metoda sazby umožňuje sázet necyklické, cyklické i neobvyklé vícecyklické řetězce (viz obr.l), ale tato možnost má nevýhodu ve značné potřebě paměti a času při tisku většího množství vícecyklických řetězců na jedné straně textu.



Obr.l: Čtyři různé benzenové struktury

Jako další ukázka následuje opět jeden z příkladů použitých ve výše zmíněném popisu, následovaný zdrojovým textem pro ilustraci (ne)náročnosti sazby při použití „sekundárních maker“ typu:

`\atom#1`	– vysázení atomu #1
`\phantatom#1`	– vynechání místa odpovídajícího atomu #1
`\sesingle`	– vazba mezi atomy (se=south–east)
`\edouble`	– dvojitá vazba mezi atomy
`\side`	– připojení bočního řetězce. . .

Z tohoto příkladu je také vidět konstrukce výsledného boxu a použití řídícího slova `\nopositioncheck`.

Obr. č.2.

```
\vbox{
\centerline{
\structure{\nopositioncheck\phantatom{H}\sephantom\atom{N}
\side{\wswabove\atom{H}}\side{\wnwbelow\atom{H}}\nnesingle\atom{C}
\side{\nnwabove\atom{H}}\side{\nwbelow\atom{H}}\esingle\atom{C}
\side{\nnedouble\atom{O}}\ssesingle\atom{O}\sswsingle\atom{H}}
```

```
\hskip 1cm{$\rightleftharpoons$}\hskip 1cm
\structure{\phantatom{H}\swphantom\atom{N}\side{\sseabove\atom{H}}
\side{\sebelow\atom{H}}\nnesingle\atom{C}\side{\nnwabove\atom{H}}
\side{\nwbelow\atom{H}}\esingle\atom{C}\side{\nnedouble\atom{O}}
\ssesingle\atom{O}\esingle\atom{H}}
}}
```

(*Zbyněk Linhart*)

Krátký příspěvek Zbyňka Linharta by měl povzbudit naše kolegy, kteří se zabývají sazbou chemických textů. Uvítali bychom z jejich praxe podrobnější informaci o tom, co existuje v oblasti veřejně dostupných programů, co se dá koupit, a pod.

(*-ov,-, -jv-*)

Značkování obrázků v dokumentech TEXu

Alan Hoenig, John Jay College/City University of New York
Email: `ajhjj@cunyvm`

Přeloženo z *TUGboatu* 12(1), 1991, str. 125–128.

Úvod

Problém včleňování obrázků do TEXovských dokumentů, který se dotýká řady lidí, ve skutečnosti už neexistuje. Díky příkazu `\special` je možné do tištěných dokumentů zařadit rozmanité druhy grafiky. Obecný postup je následující.

Požádejte TEX, aby pro obrázek uvolnil v dokumentu bílé místo vhodných rozměrů. Připravte nějakým jiným programem samostatný soubor obsahující instrukce pro vytvoření obrázku. „Vtáhněte“ tento samostatný soubor do dvi souboru pomocí příkazu `\special`. Nakonec použijte ovladač, který přijímá vnější grafické soubory. Takových ovladačů již existuje mnoho.

Tento přístup však má i své nevýhody. Především grafický soubor není doopravdy součástí zdrojového souboru TEXu. To znamená, že nebude možné prohlížet tento soubor na obrazovce, nepoužíváte-li ovšem zvlášť chytrý obrazovkový ovladač (takový, který reaguje na příkazy `\special` a je natolik inteligentní, že ví, jak tento materiál zobrazit).

Potřeba značek v obrázku

Jiná nevýhoda se ukáže, jakmile je potřeba vpisovat nějaké značky přímo do obrázku. Jak je tam vložit, aby byly správně umístěny? Někteří přidávají značky přímo jako součást grafiky, což ovšem vede k vizuální nesourodosti, poněvadž font použitý pro tyto značky nejspíš nepatří do rodiny *Computer Modern*, k níž inklinuje TEX. V ideálním případě bychom chtěli, aby značky sázel sám TEX tak pěkně, jako celý zbytek dokumentu.

V tomto článku bych rád popsal jeden možný přístup ke značkování, který se mi dobře osvědčil. S použitím METAFONTu vytvořím vlastní obrázek v podobě fontu. Přitom se průběžně rozhoduji, které body v obrázku budou značkovány (avšak nemusím určovat značky samotné), a METAFONT zaznamená souřadnice těchto bodů do fontu. Potom, když TEX včleňuje obrázek do textu, může tyto souřadnice použít ke stanovení, o kolik je potřeba box s obrázkem posunout doprava či doleva a nahoru či dolů.

Další práce v této oblasti

Na tomto poli se toho zřejmě zatím moc neudělalo. Rick Simpson byl nejspíš prvním, kdo (přinejmenším písemně) navrhl použít jako nástroj pro kreslení METAFONT. Jeho článek [2] stojí za přečtení.

Dalším příspěvkem je pak balík. maker Metaplot, vytvořený Patricií Wilcoxovou [3]. Patricia ukázala, jak je možné pojmout rozhraní mezi uživatelem a METAFONTem. Použila k tomu jazyk jednoho z běžně používaných souřadnicových zapisovačů. Jeho různé příkazy byly přeloženy do příkazů METAFONTu. Vznikl tak balík: s nímž je možné tvořit důmyslnou i pěknou grafiku. (Patricia je jedinec požehnaný značně nadprůměrnými uměleckými dovednostmi.) Tyto možnosti jsou bohatě ilustrovány ve výše zmíněném článku. (Některé z mých strategií byly inspirovány návrhy Patricie Wilcoxové a Toma Rokickiho, jimž tímto s potěšením děkuji za jejich bezděčnou pomoc.)

John Hobby [1] volí jiný přístup. Upravil METAFONT tak, že dělá výstup přímo v PostScriptu namísto *generic font* formátu, který METAFONT normálně produkuje. I když by bylo možné generovat *Computer Modern* fonty ve formátu PostScriptu, je tento program *MetaPost* především prostředkem pro tvorbu obrázků a emblémů. Pro včleňování značek do obrázku užívá jeho program odlišný přístup.

Úloha METAFONTu

Nejprve se podíváme na to, jak pracuje METAFONTovská část tohoto projektu. Dejme tomu že každý obrázek je znakem „A“ ve fontu vytvořeném METAFONTem pro tento zvláštní účel a že tento font neobsahuje žádné jiné znaky. TEX nikterak nevyžaduje, že „A“ musí vypadat jako opravdové „A“ a nemá žádné apriorní představy o správné velikosti takového znaku, takže si nikdo nebude stěžovat, pokud METAFONT požádáme o nakreslení „A“, které nebude vypadat vůbec jako „A“, ale jako obrázek, který potřebujeme v dokumentu. Předpokládejme, že zdrojový soubor METAFONTu, obsahující instrukce pro nakreslení obrázku, se jmenuje `figfont.mf`. Potom, za předpokladu, že jste všechny věci spojené s METAFONTem dobře zvládli, můžete vysadit dotyčný obrázek po deklaraci fontu

```
\font\figfont=figfont
```

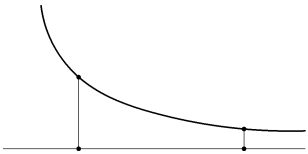
zadáním příkazu

```
{\figfont A}
```

na místě, kde se má obrázek objevit. (Nezapomeňte na uzavření do složených závorek!)

Následuje obrázek, který by se mohl dobře objevit v nějakém matematickém textu. (Ve skutečnosti se vyskytuje kdesi v prvním svazku Knuthova díla *Art of*

Computer Programming. Křivka reprezentuje část grafu funkce $f(x) = 1/a$. (Proto se zdrojový soubor pro METAFONT jmenuje `1onx.mf`.)



Bude užitečné probrat instrukce METAFONTu, jimiž se tento obrázek stvoří.

```
mode_setup;
u#=12pt; nib#=.5pt;
if (mode=smoke) or (mode=proof):
u#:=.5pt#; nib#:=.1pt; fi
define_pixels(u, nib);
beginchar("A", 12u#, 6u#, 0);
pickup pencircle scaled 2nib;
z1=(w/8,h); z2=(w/4, h/2);
z3=(w/2, h/4); z4=(w,h/8);
 % points on the curve
path p; p=z1..z2..z3..z4;
draw p; % draw the $1/x$ curve
z.x= point 2.6 of p; % another point
z5=(x2,0); z6=(x.x,0);
 % points on the $x$-axis
pickup pencircle scaled nib;
draw origin--(w,0); % bottom axis
draw z2--z5; draw z.x--z6;
 % vertical struts
pickup pencircle scaled 6nib;
drawdot z2; drawdot z.x;
drawdot z5; drawdot z6;
endchar; bye.
```

Tyto příkazy jsou ve své většině rozumně samovysvětlující, jste-li ovšem seznámeni se základy syntaxe METAFONTu. Proměnná `nib#` udává průměr kreslícího pera a `u#` je *ad hoc* zvolená jednotka délky, která je vhodná s ohledem na celkové měřítko obrázku. Z dalších proměnných tohoto programu jsou `w` a `h` šířka a výška obrázku, a veličiny `z1`, `z2`, atd. (včetně `z.x`) se vztahují k význačným bodům v obrázku.

Řádky

```
u#=12pt; nib#=.5pt;
if (mode=smoke) or (mode=proof):
 u#:=.5pt#; nib#:=.1pt; fi
```

asi popudí skalního METAFONTistu; plyne z nich, že `u#` a `nib#` nabývají hodnot závislých na zařízení, což je v rozporu s duchem METAFONTu. Tato část zdrojového textu je diktována omezeními mého monitoru. METAFONT je zvyklý pracovat se znaky o velikosti (zhruba) deset na deset tiskařských bodů. Během korektur na obrazovce (když mode je buď `smoke` nebo `proof`) používá METAFONT celou obrazovku pro zobrazení znaku. Je-li charakteristická délka znaku větší než několik jednotek pc, zobrazí METAFONT jen část písmena, pokud nezvolíme `u#` tak malé, aby bylo na monitoru místo pro zobrazení celého znaku. Výše uvedené řádky se pokoušejí implementovat tuto strategii. Charakteristické rozměry `u#` a `nib#` mají velikost, kterou bychom chtěli pro definitivní vytvoření fontu. Během korektur jsou však mnohem menší, aby umožnily prohlížet dílo na obrazovce počítače.

Komunikace METAFONTu s TEXem

TEX bychom mohli nyní použít pro popisný text k obrázku, víc bychom však obrázku pomohli připojením značek k některým jeho bodům. Knuth doplnil ke každé z okrouhlých teček značku—můžeme to udělat také?

Počítačoví experti obyčejně charakterizují TEX a METAFONT jako plnohodnotné programovací jazyky, METAFONT ovšem tuto roli tak docela neplní. S výjimkou souborů `log` a `gf` totiž nemá jiné výstupní možnosti. Proto nemůžeme zapsat souřadnice značených bodů do souborů a předat je v této formě TEXu.

Klíčovým bodem je rozpoznání užitečnosti parametru `fontdimen`. METAFONT normálně používá tyto parametry k zaznamenání univerzálních konstant pro určitý font, jakou je třeba šířka výplně *quad* nebo odstup mezi slovy. Nezdá se, že by byla nějaká horní mez počtu dovolených parametrů fontdimen, ani omezení způsobu jejich aplikace. Proto jsem si dovolil použít je k uschování souřadnic každého značeného bodu. Každý bod zabírá. dva parametry `fontdimen`—jeden pro souřadnici x a druhý pro souřadnici y.

Soubor `convert.mf` obsahuje definice maker jako `convertz_`, jež transformuje souřadnice do použitelného formátu. Hned za `mode_setup` je tedy třeba uvést příkaz

```
input convert;
```

Před instrukcí endchar pak přidáme další řádek

```
fontdimen10: convertz_(z2,z.x,z5,z6);
```

Makro `convertz_` převádí seznam párů na čísla, která odpovídají syntaxi pro `fontdimen`.

Úloha TEXu

Jak již bylo řečeno, „zMETAFONTované“ obrázky lze snadno včlenit do TEXovského dokumentu. Je však také snadné připojit značky. Pro toto použijeme sadu maker, jejichž obdoba je popsána v TEXbooku, Appendix D. Jsou-li `\x` a `\y` TEXovské *dimen* registry, můžeme je jednoduše naplnit hodnotami z vhodného parametru `\fontdimen`:

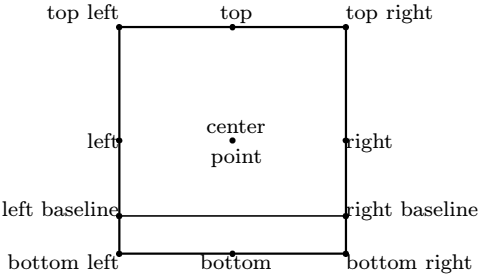
```
\advance\fontdimencount by 1
\x=\expandafter \the\fontdimen
\the\fontdimencount \figfont
```

(a podobně pro \y). Poté, co do boxu \labelbox vložíme text pro značku, konstrukce

```
\rlap{\kern\x \raise\y \box\labelbox}
```

zařídí usazení značky do žádané pozice.

To vlastně ale není ještě celá pravda. Existuje jedenáct referenčních bodů spojených s každým boxem, jak ukazuje následující diagram:



Není těžké napsat makra, která umístí zvolený referenční bod boxu do bodu, který je se značkou spojen. Tedy například nápis „top right“ byl vysazen pomocí příkazu

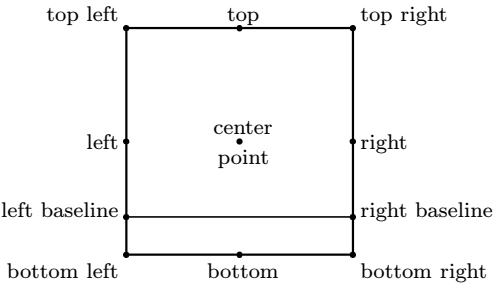
```
\blpoint{top right}
```

tj. levý dolní roh hboxu obsahujícího „top right“ má být umístěn ve značeném bodu.

A ještě to není celá historie. Čtenář, který prozkoumá obrázek podrobně, zaznamená, že značky se zdají být trochu natěsnané na značený bod. Je asi třeba ponechat navíc trochu místa. Příkazy typu \hskip2pt nebo \hskip-3pt posunou značku doprava nebo doleva, a je snadné definovat makra \up a \down taková, že (například)

```
\up3pt
```

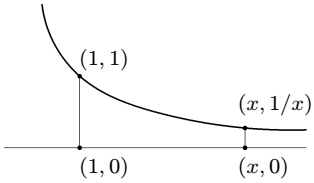
pomůže doladit vertikální umístění. Můžeme těchto úvah využít k vylepšení diagramu s referenčními body:



Následující tabulka vypisuje všechna zaměřovací makra a s nimi spojené orientace. Některé referenční body mají dvě makra. Uživatel, který potřebuje umístit levý horní roh značkového boxu, může použít \tlpoint i \ltpoint, takže nevznikají problémy se zapamatováváním správného pořadí písmen.

Orientace	Jméno makra
vlevo nahoře	`\tlpoint`
	`\ltpoint`
nahoře	`\tpoint`
vpravo nahoře	`\trpoint`
	`\rtpoint`
vpravo	`\rpoint`
vpravo dole	`\brpoint`
	`\rbpoint`
dole	`\bpoint`
vlevo dole	`\blpoint`
	`\lbpoint`
vlevo	`\lpoint`
základní čára vlevo	`\point`
	`\lBpoint`
	`\Blpoint`
základní čára vpravo	`\rBpoint`
	`\Brpoint`
středový bod	`\cpoint`

S těmito pomůckami můžeme předělat Knuthův obrázek z *ACP*, tentokráte i se značkami:



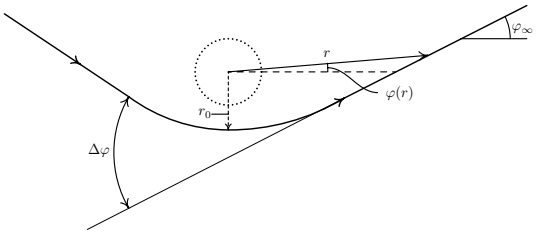
Potřeboval jsem k tomu tato makra:

```
\font\figfont=1onx
\beginfig
\lbpoint{$(1,1)$}%
\lbpoint{\up2pt $(x,1/x)$}%
\tpoint{\down 2pt $(1,0)$}%
\tpoint{\down 2pt$(x,0)$}%
\endfig
```

Příklad

Coby závěrečný příklad následuje ještě obrázek vybraný z textu o obecné relativitě. Skutečný kontext či smysl obrázku není důležitý (vztahuje se k ohybu světla

způsobenému relativistickými účinky), dobře je však vidět řada zajímavých efektů, které METAFONT umožňuje.



Příklad ukazuje, že METAFONT umí nakreslit mnoho důležitých pomocných objektů, jako jsou šipky s libovolnou orientací, tečkované a čárkované čáry, atd.

Do zbraně!

Jedním z cílů tohoto článku je zažehnout oheň v srdcích horlivých mistrů METAFONTu. Komunita uživatelů TEXu a METAFONTu by mohla mít užitek z vytvoření balíku maker, jenž by se měl k METAFONTu stejně, jako se má LATEX nebo $\mathcal{AMS}$-TEX k TEXu. Kdo se hlásí?

Literatura

[1] Hobby, John D. A METAFONT-like system with PostScript output. *TUGboat* **10**(4), str. 505–512, 1989.

[2] Simpson, Richard O. Nontraditional uses of METAFONT. Sborník *TEX: Applications, Uses, Methods* (editor Malcolm Clark), Ellis Horwood Ltd., London, 1990, str. 259–272.

[3] Wilcox, Patricia. Metaplot: Machine-independent line graphics in TEX, *TUGboat* **10**(2), str. 179–187, 1989.

(*Překlad: Danuše Lhotková*)

Popis formátu `dismac`

Tento příspěvek popisuje formát `dismac`, který se hodí pro přípravu rozsáhlejších matematických a technických textů, jako jsou diplomové a disertační práce, různé zprávy a s určitými drobnými úpravami také knihy. Tak, jak je zde popsán, je určen pro psaní českých textů, ale velmi snadno jej lze modifikovat pro psaní ve slovenštině, v angličtině nebo jiném jazyku.

Před tím, než jsem na tomto formátu začal pracovat, jsem pochopitelně zvažoval, zda se pro daný účel nedá použít některý z obecně dostupných formátů.

Formát *plain* je na příliš nízké úrovni a nenabízí prakticky žádnou podporu pro logické strukturování textu, křížové reference apod. Jeho smyslem totiž je poskytnutí základních nástrojů, s jejichž pomocí je možno efektivně budovat specializované

soubory maker, jakým je třeba `manmac`, vytvořený D. Knuthem pro psaní série *Computers & Typesetting*.

Na základně plain TEXu je postaven také populární LATEX, který se snaží řešit většinu problémů spojených s kvalitní přípravou a sazbou technických textů. Jeho velmi obecné zaměření však vede k tomu, že téměř každý jeho uživatel aktivně používá jenom část z nabízených možností a zbytek pro něho představuje balast, který zabírá paměť a prodlužuje výpočet. LATEX důsledně využívá princip prostředí (*environment*), který je elegantní a pro začátečníky užitečný, avšak v řadě případů je spíše na obtíž—zejména při vytváření matematických vzorců, kde je podle mne *plain* TEX podstatně lepší a častá nekompatibilita LATEXu brání ve využívání *know-how* pro sazbu matematiky, jež je obsaženo v TEXbooku, kap. 16–19.

Někde uprostřed mezi plain TEXem a LATEXem se nachází $\mathcal{AMS}$-TEX, který zachovává (až na malé výjimky) zpětnou kompatibilitu s *plain* TEXem. Je vhodný pro psaní matematických článků pro rozsáhlejší publikace s členěním do kapitol mu však chybí větší podpora logických struktur textu.

Návrh formátu dismac byl založen na následujících požadavcích:

- Logické členění textu do kapitol, oddílů a pododdílů;
- Další logické struktury (různé druhy seznamů, poznámky pod čarou aj.);
- Křížové reference, zejména na matematické vzorce, definice,věty apod.;
- Přiměřený výběr fontů a prostředky pro jejich přepínání;
- Automatická kompilace obsahu;
- Podpora pro zpracování rejstříku;
- Prostředky pro vkládání ilustrací do textu
- Dodatečná inteligence pro korekturní fázi
- Co nejvyšší míra zpětné kompatibility s formátem *plain*

Při psaní maker jsem se nechal inspirovat řadou zdrojů, ke hlavním přitom patřil formát `manmac` popsaný v TEXbooku (Appendix E) a vynikající kniha [1].

Členění textu

Formát `dismac` umožňuje členit text do kapitol, oddílu a pododdílů, které jsou označovány standardním způsobem, například kapitola 3, oddíl 3.2 a pododdíl 3.2.5. Kapitola je ve zdrojovém textu vymezena řídicími slovy `\chapter{(název)}` a `\endChapter`, kde se jako parametr makra, `\chapter` uvádí název kapitoly. Makro `\chapter` má alternativní tvar s volitelným parametrem

`\chapter`[⟨*číslo*⟩]{⟨*název*⟩}.

Tento tvar se použije v případě, že chceme dané kapitole přiřadit ⟨číslo⟩ mimo pořadí. Na začátku každé kapitoly se vypisuje při běhu TEXu na monitoru `Chapter` ⟨číslo⟩:.

Jméno kapitoly se vypisuje také v záhlaví stránek. Pokud by bylo příliš dlouhé a nevešlo se dobře na jednu řádku, je třeba uvést náhradní (zkrácené) jméno, a to ihned za makro `\chapter` a jeho parametry pomocí deklarace `\headName`{⟨*zkrácenéjméno*⟩}.

Podobným způsobem jako kapitoly se vymezují oddíly (`\section`{⟨*název*⟩} ... `\endSection`), které musí být vždy uvnitř kapitoly, a pododdíly (`\subsection`{

⟨*název*⟩} ... `\endSubsection`), které musí být vždy uvnitř oddílu. Také v těchto případech je možné změnit posloupnost číslování pomocí volitelného parametru v hranatých závorkách ihned za řídicím slovem.

Je sledováno řádné spárování a vnoření příslušných dvojic pro začátek a konec struktury a v případě nesrovnalosti se vydá chybové hlášení.

Kromě toho je možno používat zvláštní útvary na úrovni kapitoly, které se vymezují řídicími slovy `\looseChapter`{⟨*název*⟩} a `\endLooseChapter`. Od kapitol se liší tím, že nejsou číslované (ani nemění čítač kapitol) a nemohou obsahovat žádné oddíly a pododdíly. Hodí se proto například pro předmluvu, seznam literatury, rejstřík a podobně.

Další logické struktury

Formát plain umožňuje vytvářet seznamy položek maximálně o dvou úrovních pomocí dvojice maker `\item` a `\itemitem`. Značka pro vyznačení položky (odrážka) se přitom explicitně uvádí jako parametr. Položka přitom nemůže být členěna do odstavců. Formát dismac definuje několik druhů seznamů, přičemž používá strategii uvedenou v knize [1]. Každý seznam začíná řídicím slovem, v němž je vyznačen typ seznamu, a končí odpovídajícím řídicím slovem, jež začíná na `\end`... Položky jsou vyznačovány řídicím slovem `\item`[1]. K dispozici jsou tyto základní druhy seznamů:

- `\ALPHAList`...`\endALPHAList`—položky jsou značeny velkými písmeny vzestupně podle abecedy, tj. A., B., C., atd.
- `\bulletList`...`\endBulletList`—položky odraženy značkou •.
- `\dashList`...`\endDashList`—položky odraženy pomlčkou (—).
- `\numberList`...`\endNumberList`—položky jsou číslovány vzestupně arabskými čísly.
- `\romanList`...`\endRomanList`—položky jsou číslovány vzestupně římskými čísly (minuskulemi), tj. (i), (ii), atd.

Tedy například zdrojový text

```
\bulletList
\item První položka
\item Druhá položka
...
\eadBulletList
```

vytvoří seznam, v němž jsou položky odraženy značkou •.

Zvláštním případem je seznam vyznačený dvojicí

`\labelList`{⟨*vzor*⟩} ... `\endLabelList`{⟨*vzor*⟩},

který umožňuje používat libovolné značky explicitním zadáním. Parametr ⟨*vzor*⟩ (který musí být totožný na začátku a na konci konkrétního seznamu), představuje šablonu pro značky položek. Počítá se z něho šířka odsazení textu v seznamech, a proto

[1] Toto je jedna z mála konstrukcí nekompatibilních s *plain* TEXem.

je obvykle třeba jej volit tak, aby reprezentoval nejdelší značku, která se v seznamu vyskytuje. Značky položek se tentokrát zapisují jako `\item`{⟨*značka*⟩}. Zde je příklad:

```
\labelList{Gandalf}
 \item{Bilbo} První položka
 \item{Gandalf} Druhá položka
 \item{Thror} Třetí položka
 ...
\endLabelList{Gandalf}
```

Seznamy je možné do sebe libovolně vnořovat, za předpokladu správného párování začátků a konců seznamů. Možné je tedy třeba toto:

```
\numberList
 \item ...
 \dashList
  \item ...
  \item ...
  \ALPHAList
   \item ...
   \item ...
  \endALPHAList
  \item ...
 \endDashList
\endNumberList
```

Pokud jsou položky v seznamu krátké a vejdou se na jednu řádku, je možné zmenšit vertikální rozestupy mezi položkami uvedením příkazu `\tightList` ihned za začátek seznamu před první položkou. Rozestupy se pak zmenší.

Seznam je rovněž možno přerušit příkazem `\suspend...List`, kde místo teček je třeba uvést druh seznamu (tedy např. `\suspendBulletList`). Další text se pak sází v celé šířce sloupce. Obnovení výčtu položek seznamu se zařídí příkazem `\continue...List`, kde je opět třeba uvést typ seznamu (např. `\continueDashList`). Speciálním případem je opět seznam s explicitními značkami, kde se u příkazů `\suspendLabelList` a `\continueLabelList` uvádí jako parametr vzor, který seznam identifikuje. Například

```
\labelList{Gandalf}
 \item{Bilbo} První položka
\suapendLabelList{Gandalf}
... Text mimo seznam ...
\continueLabelList{Gandalf}
 \item{Gandalf} Druhá položka
 \item{Thror} Třetí položka
 ...
\endLabelList{Gandalf}
```

Poznámky pod čarou se vytvářejí příkazem `\note`{text}. Jako značky se u těchto poznámek v rámci jedné kapitoly střídají symboly †, ‡, § a ¶. Tento systém tedy funguje v případě, že na jednu stránku nevyjdou více než čtyři poznámky pod čarou. Pokud se jedna poznámka vztahuje k několika místům v textu, je možno použít makro `\lastNote`, které vysadí v daném místě textu značku předchozí poznámky pod čarou. Uživatel sám musí ošetřit případné problémy vzniklé například zalomením strany mezi dvěma místy se stejnou značkou.

Matematické struktury

Formát dismac v současné podobě nabízí pět typů číslovaných a zvlášť vysazených matematických struktur. Jsou to:

- Definice, které jsou vymezeny klíčovými slovy `\definition` a `\endDefinition`.
- Věty, vymezené klíčovými slovy `\theorem` a `\endTheorem`.
- Tvrzení, vymezená klíčovými slovy `\proposition` a `\endProposition`.
- Lemmata, vymezená klíčovými slovy `\lemma` a `\endLemma`.
- Důsledky, vymezené klíčovými slovy `\corollary` a `\endCorollary`.

Všechny tyto struktury mají společné číslování (tedy nečíslují se nezávisle definice, věty atd., což považuji za méně přehledné). Ve stejné posloupnosti jsou navíc i čísla matematických rovnic (vysazených na zvláštní řádce), u nichž je deklarována identifikační značka (viz dále). Rovnice i matematické struktury jsou tedy průběžně číslovány při levém okraji, a to dvojicí čísel oddělených tečkou: prvním je číslo kapitoly a druhým číslo objektu v rámci kapitoly.[2] Rozsah čísel vyskytujících se na dané straně je vyznačen v záhlaví této strany, což usnadňuje hledání.

Makra pro začátek matematických struktur mohou mít jeden nepovinný parametr uzavřený v hranatých závorkách. Tento parametr je identifikační značkou pro další případné odkazy na tento útvar. Například

```
\theorem[main] Nechť $x$ je ...
 ...
\endTheorem
```

Důkazy začínají řídicím slovem `\proof`, které zahájí nový odstavec a vypíše v kurzívě slovo Důkaz. Na konci důkazu se pak uvede řídicí slovo `\QED`, které vysadí černý čtvereček a vloží vertikální odskok.

Křížové reference

Formát `dismac` umožňuje odkazovat v textu jak na obecné textové struktury (kapitoly, oddíly, pododdíly, strany) a na matematické objekty (rovnice, definice, věty atd.).

Odkazy mohou být jak zpětné (na objekt, který je v textu před místem odkazu), tak i dopředné. Vyskytují-li se v textu jenom zpětné reference, potom obyčejně stačí jeden průchod TEXu. Výjimkou jsou stránkové odkazy, které ze známých důvodů

[2] V každé kapitole se tedy čísluje od jedničky

musí být řešeny přes diskový soubor příkazem `\write`. Dopředné odkazy pak obecně vyžadují dva průchody[3].

Chceme-li odkazovat na nějakou textovou strukturu, musíme v požadovaném místě takto deklarovat identifikační značku odkazu:

`\make ... Ref`{⟨*značka*⟩},

přičemž tečky v řídicím slov ř nahradíme typem odkazu podle dále uvedeného výčtu. Vlastní odkaz (může být udělán i vícekrát z různých míst) se pak provede pomocí makra `\...Ref`{⟨*značka*⟩}, kde namísto teček opět doplníme typ odkazu. Značkování různých typů odkazů je navzájem nezávislé, takže při použití stejné značky pro různé typy odkazů nedojde k jejich kolizi. Mohou se použít tyto varianty:

- Odkazy na kapitolu–místo se vyznačí pomocí řídicího slova `\makeChapRef`, odkazuje se pomocí `\chapRef`.
- Odkazy na oddíl-místo se vyznačí pomocí řídicího slova `\makeSecRef`, odkazuje se pomocí `\secRef`.
- Odkazy na pododdíl–místo se vyznačí pomocí řídicího slova `\makeSubsecRef`, odkazuje se pomocí `\subsecRef`.
- Odkazy na stranu-místo se vyznačí pomocí řídicího slova `\makePageRef`, odkazuje se pomocí `\pageRef`.

Odkazy na matematické objekty se řeší poněkud odlišně kvůli jejich větší frekvenci. Rovnice, na něž chceme odkazovat, musí před uzavírající dvojicí `$$` deklarovat značku pomocí `\eqTag` ⟨*značka*⟩. Například

```
$$ 1 + 1 = 1.9999998. \eqTag approx $$
```

Zde je identifikační značkou pro odkaz approx. Všimněme si, že parametr makra `\eqTag` neuzavíráme do složených závorek (je delimitován pomocí `$$`). Případné mezery mezi značkou a `$$` se ignorují. Identifikační značky pro definice, věty aj. se deklarují, jak bylo již uvedeno, pomocí volitelného parametru v makrech `\definition`, `\theorem`, atd.

Odkazy na matematické objekty se provádějí příkazem `\stuffRef`{⟨*značka*⟩}. Navíc je možné jednoduše odkazovat na předchozí objekt makrem `\lastStuff` a na nejbližší následující objekt makrem `\nextStuff`.

Fonty

Výběr fontů je ve formátu dismac poněkud bohatší než ve formátu *plain*, zdaleka však v tomto ohledu nekonkuruje LaTeXu. Předpokládá se, že hlavní část textu je psána deseti- anebo dvanáctibodovým písmem. Z důvodu obecné dostupnosti se používají EC-fonty vyvinuté N. Schwarzem na základě dohody o rozložení akcentovaných znaků, učiněné na setkání TeX90 v Corku.

Nezávisle je možno přepínat velikost i typ písma. K dispozici jsou tyto velikosti:

- desetibodové: zapíná se příkazem `\tenPoint`
- dvanáctibodové: zapíná se příkazem `\twelvePoint`
- sedmibodové: zapíná se příkazem `\sevenPoint`

[3] Takových odkazů bývá v textu obvykle málo a v průběhu korektury není nutno se jimi zabývat

- osmibodové: zapíná se příkazem `\eightPoint`
- čtrnáctibodové: zapíná se příkazem `\fourteenPoint`
- pětadvacetibodové: zapíná se příkazem `\twentyfivePoint`

Ve všech případech se k příslušné velikosti automaticky doplňují vhodné velikosti indexů 1. a 2. řádu (subscript a subsubscript).

Z typů písma jsou k dispozici:

- standardní románské (přepínač `\rm`)
- kurzíva (přepínač `\it`)
- položené písmo (přepínač `\sl`)
- tučné písmo (přepínač `\bf`)
- kapitálky (přepínač `\sc`)

Přepínání velikosti a typu je ortogonální, to znamená, že při změně velikosti zůstává předchozí typ a naopak.

Kompilace obsahu

Chceme-li vytvořit automaticky obsah publikace, musíme na začátku dokumentu uvést příkaz `\makeToc`. V takovém případě se vytvoří soubor toc.tex, do něhož makra pro logické členění textu automaticky zapisují informace pro tvorbu obsahu. Každá položka má tvar

`\tocEntry`{⟨*typ*⟩}{⟨*číslo*⟩}{⟨*název*⟩}{⟨*strana*⟩}.

Parametr ⟨*typ*⟩ vyjadřuje úroveň dotyčné strukturní jednotky (0-„looseChapter“, 1-kapitola, 2-oddíl, 3-pododdíl). Po případné editaci je možno soubor z obsahem natáhnout na zvoleném místě pomocí `\input`. Přitom je třeba uvážit možné posunutí stránkování (proto je obvykle zapotřebí jeden průchod navíc).

Tvorba rejstříku

Pomocí formátu dismac lze připravit podklady pro zpracování rejstříku hesel. Pro tento účel je potřebné vyznačit v textu slova, která mají do rejstříku přijít. Existují dvě varianty:

- Uzavřeme-li nějakou část textu do uvozovek `"..."`, zapíše se tato jako heslo do rejstříku, ale zároveň zůstane v textu.
- Pokud za první uvozovky napíšeme hvězdičku, tj. `"*..."`, potom se uzavřený text zapíše jako heslo do rejstříku, ale ve vlastním textu se neobjeví.

Formát dismac tedy nedovoluje používat uvozovky `"` jako náhražku dvojice apostrofů `''` což by však nemělo být na závadu. Je rozumné uvádět jako hesla pouze holý text, tj. bez vyznačení typu písma apod. V opačném případě se totiž komplikuje třídění hesel podle abecedy.

Pokud na začátku dokumentu napíšeme `\makeIndex`, zapisují se rejstříková hesla do souboru `index.tex` ve tvaru

`\indexEntry`{⟨*hesla*⟩}{⟨*strana*⟩}.

V průběhu korektur není obyčejně nutno tento soubor vyrábět, neboť hesla pro rejstřík se zapisují také na okraj té strany, na níž se vyskytují (viz dále).

Soubor index.tex je potřeba před zařazením do publikace ještě upravit. Nejprve se utřídí hesla podle abecedy, spojí se výskyty stejného hesla na více stranách, a nakonec se případně některá hesla typograficky upraví (odlišný typ písma atd.).

Sazba rejstříku by měla začínat řídicím slovem `\index`, které zahájí dvousloupcovou sazbu a nastaví některé parametry pro tento speciální účel. Konec rejstříku je pak vyznačen řídicím slovem `\endIndex`.

Vkládání obrázků

Obrázek, který chceme do textu vložit, je potřeba napřed uvést do formy přijatelné pro TEX, tj. buďto jako zvláštní font (vytvořený například pomocí METAFONTu), anebo jako grafický rastr podporovaný použitým DVI ovladačem. Formát dismac nabízí podpůrné makro pro druhý z uvedených způsobů, a to pro ovladače DVIDRV, které jsou součástí volně dostupného emTEXu, který je zřejmě relativně nejlepší implementací TEXu na PC. Ovladače umožňují vkládat soubory typu `.PCX` a `.MSP`. Úprava pro jiný ovladač s podobnými možnostmi je víceméně triviální.

Pomocí příkazu

$$\texttt{\textbackslash specFig}\{\langle soubor\rangle\}\{\langle šířka\rangle\}\{\langle výška\rangle\}\langle značky\rangle\ \texttt{\textbackslash par}$$

se vytvoří box nazvaný `\figure`, který obsahuje

- rastr obrázku obsaženého v souboru, jehož jméno je $\langle soubor\rangle$. Další dva parametry zadávají šířku a výšku boxu `\figure` (skutečná šířka s výška obrázku je přitom irelevantní). Další požadavky (umístění souboru s rastrem v určitém adresáři aj.) jsou dány ovladači DVIDRV.
- značky „vlepené“ do obrázku přímo TEXem. Tyto značky se zadávají pomocí maker `\placeMark`, která se zapisují jako argumenty makra `\specFig` (viz výše). Tyto argumenty jsou delimitovány řídicím slovem `\par`, resp. vynecháním prázdné řádky (což bude asi častější případ).

  Syntaxe makra `\placeMark` je následující:

$$\texttt{\textbackslash placeMark}\{\langle značka\rangle\}\{\langle x\rangle\}\{\langle y\rangle\}\{\langle lorg\rangle\}.$$

Parametr $\langle značka\rangle$ udává symbol či text, který se má do obrázku vepsat, $\langle x\rangle$ a $\langle y\rangle$ pak souřadnice referenčního bodu, k němuž se má značka, napsat (počítáno od levého dolního rohu boxu `\figure`—jednotky jsou uváděny podle zvyklostí TEXu). Parametr $\langle lorg\rangle$ konečně určuje polohu značky vzhledem k referenčnímu bodu. Může nabývat hodnot 1–9 s tímto významem:

- 1 – text vpravo nahoře
- 2 – text vpravo a vertikálně centrován
- 3 – text vpravo dole
- 4 – text nahoře a horizontálně centrován
- 5 – text centrován v obou směrech
- 6 – text dole a horizontálně centrován
- 7 – text vlevo nahoře
- 8 – text vlevo a vertikálně centrován
- 9 – text vlevo dole

Pro ilustraci těchto poněkud komplikovaných maker poslouží následující příklad:

```
\specFig{obrazek.pcx}{8cm}{15cm}
 \placeMark{$x_1$}{35mm}{72mm}{3}
 \placeMark{$\rho(x_1)$}{45mm}{5mm}{6}
```

Podstatné je vynechání řádku za posledním makrem `\placeMark`.

Pro vkládání obrázků (připravených bud výše popsaným způsobem, anebo jakkoliv jinak) do textu je možno použít jednak makra plain TEXu pro plovoucí vsuvky, tj. `\topinsert`, `\midinsert` a `\pageinsert`, jednak je k dispozici makro `\rightInsert`, které vkládá obrázek k pravému okraji stránky se zúžením sázeného sloupce v potřebné výšce. Přitom se předpokládá, že hotový obrázek je připraven v boxu `\figure`.

Makro `\rightInsert` nevytváří plovoucí vsuvku (to ani dobře nejde), takže obrázek se umístí tam, kde je makro uvedeno. Tímto místem musí být začátek odstavce. Hlavně je však třeba zajistit, aby na straně pro obrázek zbývalo ještě dost místa. Syntaxe makra `\rightInsert` je tato:

`\rightInsert`{⟨*popisek*⟩}.

Text uvedený v parametru ⟨*popisek*⟩ se vysadí pod obrázkem do sloupce o šířce rovné šířce boxu `\figure`.

Módy zpracování, korekturní informace a jiné

Formát `dismac` rozeznává tři hlavní módy zpracování dokumentu:

- korekturní mód, který se volí řídicím slovem `\proofMode` na začátku dokumentu.
- finální mód (první běh), volený řídicím slovem `\firstFinal`.
- finální mód (druhý běh), volený řídicím slovem `\secondFinal`.

Korekturní mód se nestará o reference, které nelze vyřídit při jednom běhu TEXu. Na levý okraj textu zapisuje některé dodatečné informace, které jsou pro korektury užitečné. Jsou to jednak identifikační značky matematických objektů (rovnic, vět atd.), vypisující se vždy v místě, kde se v textu dotyčný objekt nachází. Dále se v levém horním rohu vypisují rejstříková hesla, která se na dané straně nacházejí. Při dolním okraji se dále vypisuje datum a počet minut uplynulých od začátku dne.

První běh finálního módu zapisuje do souboru `xrefs.tex` ty křížové odkazy, které je třeba řešit dvojím průchodem a vynechává samozřejmě všechny výše popsané korekturní informace.

Druhý běh finálního módu načítá soubor křížových referencí vzniklý v prvním běhu, avšak zároveň vytváří nový soubor těchto referencí, který se jmenuje `comp.tex`. Pokud po skončení druhého běhu zjistíme, že se soubory `xrefs.tex` a `comp.tex` od sebe liší, je nutno druhý běh opakovat. Taková situace však nastává jen v singulárních případech.

Pomocí dodatečných příkazů `\makeToc` a `\makeIndex` lze požádat o výpis souborů s obsahem a rejstříkovými hesly (viz výše).

Je možné část textu sázet ve dvou sloupcích. Začátek dvousloupcové sazby se vyznačí řídicím slovem `\twoColumns` a konec pak řídicím slovem `\endTwoColumns`.

Závěr

Formát dismac je nadstavbou formátu csplain, který je českou mutací standardníhoplainu s EC-fonty namísto odpovídajících řezů Computer Modern a českými vzory pro dělení. Formát je tedy možno předkompilovat INITEXem:

```
tex /i /8 /o /c:kam2cork &csplain dismac.
```

Lomítkové parametry jsou specifické pro emTEX. Parametr tt /c zadává jméno souboru `kam2cork.tcp` pro konverzi vstupního kódu Kamenických na rozložení znaků EC-fontů.

Formát `dismac` dávám tímto k dispozici individuálním i kolektivním členům CSTUGu. Makra obsažená v tomto formátu jsem odzkoušel při přípravě své disertační práce, což ovšem neznamená, že jsou bez chyb, anebo dokonce dokonalá. Případní zájemci o vyzkoušení tohoto formátu jsou vítáni, na požádání jim zašlu zdrojový soubor. Naleznou-li nějaké chyby, prosím, aby mi je sdělili, avšak nešířili modifikované verze pod stejným názvem.

Literatura

[1] Appelt W. *TEX für Fortgeschrittene.* Addison-Wesley, Bonn, 1988.

(*Ladislav Lhotka*)
e-mail: `lhotka@csearn`

TEX a grafika

V poslední době se stále častěji objevují dotazy, jak je TEX schopen začlenit grafické obrazce do textu. Stejně často — snad dokonce i častěji — je tato otázka diskutována na různých TEXových setkáních a v časopisech o TEXu.

Protože se i na mne stále obracejí uživatelé TEXu s tímto dotazem, pokusím se v tomto příspěvku nastínit základní postupy začleňování grafiky do textu Tem. Přitom se budu snažit zdůraznit výhody a nevýhody jednotlivých způsobů tak, jak jsem se s nimi setkával při praktických aplikacích. Přirozeně toto nebude vyčerpávající přehled, ale měl by čtenáři (a především mně) dát alespoň základní orientaci v této tematice. Ještě je potřeba zdůraznit, že slabý zrak a nedostatečné technické vybavení mi dovoluje zabývat se pouze černobílými obrázky.

Podrobnější popis jednotlivých způsobů tvorby a zpracování obrázků se pravděpodobně objeví v jednom z příštích čísel TEXbulletinu (což je zároveň výzva k tomu, aby ti z vás, kteří mají nějaké zkušenosti v této oblasti, se o ně podělili s ostatními čtenáři).

Obecné lze celý postup rozdělit do tří fází.

i) Příprava obrázku, jeho převedení do elektronické formy.
ii) Případné opracování obrázku do vhodného tvaru (konverze do formátu vhodného pro další zpracování,...).
iii) Začlenění obrázku do textu.

Jak vytvořit obrázek?

Tato otázka vyvstává v okamžiku, kdy se rozhodneme začlenit obrázek nebo grafiku do textu. Máme tedy představu, jak by měl obrázek vypadat a je potřeba jej získat v elektronické podobě. Zde záleží především na programovém vybavení a zařízení, které máme k dispozici. Následující neúplný výčet dává alespoň orientační představu základních možností:

i) Jsme vlastníky scanneru (a příslušného softwaru) a obrázek již máme ve formě předlohy (např. černobílé stokoruny). V tomto případě je možné vytvořit elektronický obraz obrázku ve formě bitové mapy a to pravděpodobně i ve velikosti, v jaké jej potřebujeme do pořizovaného textu. Nevýhodou je, že jsme omezeni rozlišovací schopností scaneru, kterou by pravděpodobně bylo možné předchozí manipulací s xeroxem a originální předlohou poněkud zvýšit. Nicméně je zde problém snadné změny velikosti. Rovněž tak kvalita pořízeného elektronického obrazu předlohy je závislá na samotné předloze. Výsledkem tedy je bitová mapa (popřípadě zkomprimovaná). Tuto cestu jsem neměl možnost v praxi dostatečně ověřit.

ii) Věnovali jsme určité úsilí k získání vhodného programu (věříme, že legálně) a k získání zručnosti v ovládání vhodného programu, který budu nazývat „grafický editor“ — tedy něco, čím jsme schopni vytvářet obrázky podle svých vlastních potřeb (např. černobílé stokoruny s novým státním znakem). Tyto grafické editory mívají různou „inteligenci“ a je možné s nimi provádět různá kouzla. Podstatné je to, že umějí zpravidla nakreslený obrázek nebo jeho část vytisknout na různých výstupních zařízeních (např. v PostScriptu, na laserové tiskárně, atp.) a to buď přímo, nebo přes soubor. Dále umožňují vytvořený obrázek uschovat v souboru v některém z grafických formátů. Máme-li grafický editor s vyšší inteligencí, pak je možné při tisku měnit měřítko, tj. velikost obrázku. V této oblasti jsem měl možnost udělat několik pokusů, jejichž výsledky je možné shrnout do následujících bodů:

   α) Program MATLAB umožňuje vykreslovat různé křivky a plochy (buď analyticky nebo tabulkově zadané) a to několika způsoby a v různých pohledech. Tyto obrázky je možné uložit do souboru a zpracovat grafickým postprocesorem (dodávaným s MATLABem) do tvaru vhodného pro tisk na různých zařízeních (z mého pohledu byla nejzajímavější tiskárna HP LaserJet — později bude vidět proč); verze MATLABu, kterou jsem měl k dispozici neumožňovala tisk přes PostScript. Nevýhodou bylo to, že obrázky byly automaticky vykresleny na plnou šířku stránky (otázku zvětšování či zmenšování jsem neuměl uspokojivě vyřešit).

   β) Program AUDOCAD umožňuje kreslit různé technické obrázky, jejichž komplikovanost záleží na dovednosti uživatele (toto nemá být placenou reklamou, proto ta stručnost). Z AUDOCADu lze tisknout na různých výstupních zařízeních (i přes PostScript) a to i přes soubor. Další výhodou AUDOCADu je to, že při tisku lze obrázkům v dostatečně velikém rozsahu měnit měřítko, tj. lze apriori říci, jak má být obrázek velký. Nevýhodou tohoto programu je jeho vyšší cena.

   γ) Existuje nadstavba AUDOCADu, která umí jednoduché obrázky převést do příkazů TEXu (slovem jednoduchý mám na mysli obrázky složené z jed-

noduchých geometrických útvarů, které TEX zvládá). Tuto cestu jsem neměl možnost v praxi ověřit.

δ) Programy typu PAINTBRUSH, které jsou podle mého názoru vhodné víc pro rozvíjení fantazie, než pro kreslení různých náčrtků a technických obrázků. Nicméně je nutno je také zahrnout do prostředků, které poskytují možnost tvorby obrázků. Navíc převážná-většina těchto programů umožňuje pracovat s barvami a při tisku tyto barvy konvertuje do rastrů různé hustoty, čímž se vytváří dojem barevnosti. Obrázky je možné uschovávat v různých grafických formátech (PCX, BMP, ...). Nevýhodou těchto programů, jak jsem je měl možnost poznat já, bylo to, že při zvětšování při tisku docházelo k značnému zkreslení hladkostí šikmých čar a obrysů. Toto bylo způsobeno tím, že při zvětšování se zvětšovaly jednotlivé body, tj. při dvojnásobném zvětšení se z každého černého bodu 1×1 stane černý čtvereček 2×2. Měl jsem možnost pracovat se třemi programy tohoto typu a to MICROSOFT PAINTBRUSH (verze z roku 1987), PAINTBRUSH z WINDOWS 3.0 a DR.GENIUS.

ε) Existuje ještě celá řada dalších programů pro vytváření obrázků (HARWARD GRAPHICS, atd.), avšak ty nebylo možné z důvodů omezené délky lidského života prozkoumat.

φ) Za zmínku ještě stojí krátká informace o programovém balíku zvaném GRAPHIC WORKSHOP, který je velice pohodlným nástrojem pro konverzi mezi různými grafickými formáty. Přesto, že je v dokumentaci uvedena možnost zvětšování a zmenšování obrázků, mé pokusy se zvětšováním byly naprosto neuspokojivé (vyrobit z obrázku Mony Lisy vodorovnou čáru dlouhou 5 cm dokážu i jinak a rychleji, než pomocí operace zvětšování z GRAPHIC WORKSHOP).

κ) Jako poslední bod tohoto výčtu chci uvést naprosto neuspokojivé zkušenosti se všemi programy, které pouze ukládají bitovou mapu obrazovky do souboru. Tyto programy jsou nedostatečné pro zamýšlený účel, neboť pro tisk obrázku na laserové tiskárně v hustotě 300 dpi odpovídá např. obrazovka VGA (s rozlišením 640 bodů) ploše 2.13×1.6 palce, což je přibližně 54×40 mm (viz obr. 1).

iii) Umíme METAFONT.(?) Původně byl METAFONT vytvořen pro návrh písem. Je to však propracovaný nástroj pro vytváření nejen písem, ale i obrázků různého charakteru. Je možné vytvářet nejen různá loga, ale i znázorňovat funkce, kreslit šrafované plochy a další rozmanité obrazce podle toho, nakolik jsme s ovládáním METAFONTu obeznámeni. Na obrázek je možné totiž pohlížet jako na jeden (někdy dosti veliký) znak nějaké abecedy. Důležité je pouze to, abychom byli schopni požadovaný obrázek popsat analyticky s použitím prostředků, které METAFONT nabízí. A ovládnout v této úrovni METAFONT znamená naučit se další soubor příkazů a povelů metajazyka a malinko pochopit způsob, jakým METAFONT bude interpretovat zadávané příkazy. Je pochopitelné, že METAFONT pravděpodobně nebude nejvhodnější nástroj pro kreslení obrázku raketoplánu, ale na druhé straně nakreslit obrázek hyperbolometrických funkcí METAFONTem nečiní větších obtíží (pokud METAFONT ovládáte). Poznamenejme ještě, že METAFONT je stejně jako TEX veřejně dostupný prakticky bez finančních nákladů.

iv) Pro jednoduché obrázky mohou být dostatečné nástroje, kterými je vybaven TEX. Existují různá makra, která umožňují vytvářet jednoduché obrázky přímo v TEXu, nebo jednoduché obrázky vytvořené v jiném prostředí převést do posloupností přímo zpracovatelných TEXem (jde např. o PICTEX, okolí `picture` v LATEXu, programové vybavení TEXCAD, výše zmíněnou konverzi jednoduchých obrázků z AUTOCADu do TEXu atd.) Cáry se v tomto prostředí skládají z teček vhodně umisťovaných vedle sebe. Tím ovšem narůstají nároky na paměť a dost značně se prodlužuje doba zpracování dokumentu Čem. Na semináři o TEXu v Cikháji v roce 1989 bylo možné zahlédnout v LATEXu vytvořený obrázek obrysu republiky (tehdy ještě Československé — i když, na druhé straně, zase socialistické). Nicméně nebylo řečeno, jak dlouho vytvoření takového obrázku trvalo. Do této kategorie patří i způsoby vytváření obrázků tak, že se popíšou vhodnými makry v TEXu, která do `.dvi` souboru uloží vhodné příkazy, které se post-procesorem rozvinou do správného tvaru.

Závěrem této části je tedy to, že způsobů k vytvoření obrázků je mnoho a záleží na konkrétních podmínkách, pro který se rozhodneme. Ne všechny jsou stejně vhodné pro každého a pro každý účel. Některé vyžadují investovat něco úsilí do zvládnutí nové věci, některé naopak neposkytují požadovanou kvalitu (černobílé stokoruny), jiné jsou pracné, další pak vyžadují nemalé náklady na pořízení programového nebo technického vybavení. Z druhé strany lze získat obrázky, jejichž zmenšování či zvětšování je obtížné, nebo obrázky, které mohou být nezávislé na výstupním zařízení (přechod mezi oběma způsoby tvoří obrázky ve tvaru fontů).

Nicméně i když máme obrázek vytvořený, ještě není vyhráno. Je potřeba začlenit obrázek do textu, který má být zpracován TEXem.

Jak naložit s vytvořeným obrázkem.

Ještě jednou shrneme výsledek předchozího odstavce a to z pohledu, v jakém tvaru je obrázek připraven pro začlenění do textu.

i) Obrázek je ve tvaru bitové mapy, nebo v některém z grafických formátů, které mohou být zpracovány dále.

ii) Obrázek je ve tvaru, který je připraven pro tisk na výstupním zařízení.

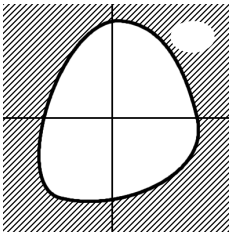
iii) Obrázek je ve fontu (obrázek se může skládat pouze s jednoho nebo i z více písmen. Zpravidla pak může být přístupný i v různých velikostech, tj. daný font lze vygenerovat (např. METAFONTem) ve zvětšeních, obvyklých pro běžné fonty.

iv) Obrázek je popsán v souboru příkazy TEXu.

Nyní je na místě se zmínit o tom, že případ zmíněný v bodě ii) je možné v některých případech převést na případ i) nebo iii). Jde o to, že na jedné straně je možné obrázky ve vhodném formátu (např. v PostScriptu) považovat za bitové mapy a na straně druhé lze některé grafické formáty (bitovou mapu v PCL, grafické formáty CUT, TIFF, BMP, atd.) převést do fontů. Např. obrázky ve tvaru PCL (což je popis stránky používaný např. laserovou tiskárnou HP LaserJet II), je možné konvertovat do fontu programem PCLPIC, jehož první verze je šířena v rámci CSTUGu. Pokud grafické editory umí při tisku do souboru ve tvaru PCL obrázek zvětšovat (např. jako AUTOCAD), je možné vytvářet tentýž obrázek jako fonty v různých zvětšeních.

Nadále nebudeme případ ii) tedy uvažovat.

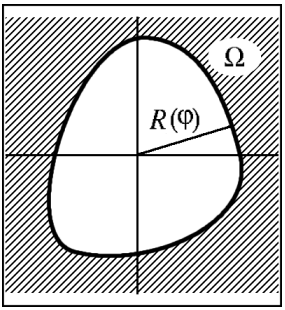
Je-li obrázek ve tvaru souboru, který je možné zpracovat ovladačem tiskárny (v některých případech i ovladačem obrazovky), je možné použít pro začlenění obrázku do textu instrukce TEXu `\special{...}`, kde do závorek se píše příkaz pro ovladač a název příslušného souboru s grafikou. Nicméně je nutno příkazy TEXu vynechat správně veliké místo v textu. Umístění popisu do takto začleňovaného obrázku je však obtížnější a neobejde se zpravidla bez tisku a měření pravítkem. Takto byl začleněn do textu obrázek č.1. Obrázek byl připraven pomocí programu PaintBrush (příprava trvala asi minuty). Nelze však snadno měnit jeho velikost tj. nemáme možnost tento obrázek zvětšit, neboť byl ve tvaru grafického souboru PCX tak, jak ho vytvořil PaintBrush. (Popis v obrázku — viz obrázek č.2 — byl vynechán záměrně.)



Obr.1 Obrázek ve formátu PCX začleněný instrukcí `\special`.

Nyní jde o to, jak má být obrázek začleněn do testu. TEX a jeho základní nadstavby (plain, $\mathcal{AMS}$-TEXa LATEX) mají v záhadní výbavě pouze jednoduché nástroje, tj. umožňují vynechat místo pro obrázek (vynechávají celou šířku stránky) a to podle potřeby uprostřed stránky, nebo nahoře či dole na. stránce. Zde je potřeba pouze říct, jak je obrázek vysoký, a specifikovat jeho umístění, eventuálně doplnit text k obrázku (nikoliv do obrázku).

Nicméně existuje řada maker jak pro plain, tak pro LATEX, která umožňují začlenit obrázek na levou či pravou stranu stránky a text odstavce či odstavců okolo obrázku oblomit. Tímto způsobem však nelze řešit všechny případy, a mnohdy je nutné ruční doladění umístění obrázku. Také začleňování obrázků do sazby ve více sloupcích je značně obtížné a dosud jsem se nesetkal s uspokojivým řešením.



Obr.2 Obrázek převedený do fontu.

Obrázek č.2 byl pořízen tak, že z PaintBrushe byl obrázek č.1 tištěn do souboru pro laserovou tiskárnu (soubor PCL) při určitém zvětšení a následně zpracován do fontu programem PCLPIC, šířeným v rámci CSTUG. Pak byl začleněn do textu pomocí makra pro začleňování obrázků do textu.

Popis byl doplněn TEXem rovněž pomocí tohoto makra tak, žena obrazovce byly odměřeny (pomocí nástrojů ovladače obrazovky) příslušné rozměry. Zároveň je vidět, že PainBrush není pro zvětšování obrázků vhodným prostředkem, protože při zvětšování se zvětšují jednotlivé body

a tudíž se ztrácí hladkost čar. Rámeček okolo obrázku byl doplněn rovněž makrem v TEXu a ukazuje výřez, který byl při tisku PaintBrushem zvolen (tj. již v PaintBrushi byl zvolen malinký okraj okolo obrázku).

Pokud je obrázek ve tvaru přikazů TEXu (bod ii), pak tyto příkazy mohou být součástí zdrojového textu a nejsou prakticky žádné další problémy s dalším zpracováním. Rovněž tak velikost obrázku může být zjištěna přímo TEXem.

Pokud je obrázek ve tvaru fontu (bod iii), je výstup závislý na určité konkrétní tiskárně, v závislosti na hustotě, v jaké byl font připraven. Výhodou je opět to, že rozměry obrázku si může TEX zjistit automaticky. Dále je možné (v případě tvorby obrázku METAFONTem) přenášet i jiné charakteristické rozměry obrázku do TEXu a tím učinit případný popis v obrázku nezávislý na změně velikosti obrázku (při jeho novém vytváření v pozměněné velikosti — pokud je to potřeba).

Závěrem.

Věřím, že v jednom z příštích čísel bude věnována značná pozornost zkušenostem získaným z vytváření a zpracování obrázku a to jednak kMETAFONTem a jednak programy konvertujícími obrázky do fontů.

Literatura

Wichura M. J., PICTEX: *Macros for Drawing PiCturcs*, TUGboat, Vol 9 (1988). No.2, pp.193–197.

Wilcox P.P., *METAPLOT: Maschine Independent line Graphics for TEX*, TUGboat, Vol 10 (1989), No.2, pp.179–187.

Nagy D., *A Bar Chart in LATEX*, TUGboat, Vol.10 (1989); No.2, pp.239–240.

Salomon D., *DDA Methods in TEX*, TUGboat, Vol.10 (1989), No.2, pp.207–216.

Olejniczak-Burkert R., *texpic – Design and Implementation of a Picture Graphics Language in TEX à la pic*, TUGboat, Vol.10 (1989), Vol.4, pp.627–637.

Kneser T., *LATEX Paragraphs Floating around Figures*, TUGboat, Vol.12 (1991); Vol.1, pp.28–30.

Sowa F.[4], *Integration of Graphics into TEX*, TUGboat, Vol.12 (1991), Vol.1, pp.58–63.

Maclenan M.P., Burns G.M., *An Approach to Drawing Circuit Diagrams for Text Books*, TUGboat, Vol.12 (1991), No.1, pp.66–73.

(*Oldřich Ulrych*)
e-mail: `ummou@csearn`)

[4] Friedhelm Sowa je jedním z přednášejících semináře, který proběhne od 18. do 30.9.1991 ve Skalském Dvoře.

Obsah TUGboatu 12(2) 1991

Počínaje tímto číslem TEXbulletinu začínáme uvádět obsahy TUGboatů. TUGboaty budou k dispozici (počínaje ročníkem 12) ke krátkodobému zapůjčení na adresách:

Anna Najmanová, Matematický ústav UK, Sokolovská 83, 186 00 Praha 8.
Karol Nemoga, Matematický ústav SAV, Štefánikova 49, 814 73 Bratislava.
Petr Sojka, UVM MU, Burešova 20, 601 77 Brno.

Vydalo: Československé sdružení uživatelů TEXu vlastním nákladem jako interní publikaci
Počet výtisků: 400
Písmo: Computer Modern
Vytištěno na: MFF UK v Praze