

## TeX a PostScript v grafike programovacích jazykov

Na svete je viacero programovacích jazykov a väčšina z nich má aj grafickú knižnicu. Málokto z nich však umožňuje zápis L<sup>A</sup>T<sub>E</sub>Xovských a PostScriptových príkazov priamo do zdrojového kódu programu napísaného v tomto jazyku. Podľa ankety *Ako TeXujeme* je jasné, že okolo 80% TeXistov svoje obrázky aj programuje – čo je potešiteľné – tak ma napadlo opísať Vám spôsob, ako sa mi to podarilo pre programovací jazyk FORTRAN. Vytvoril som balík programov, ktoré z FORTRANovského kódu obsahujúceho L<sup>A</sup>T<sub>E</sub>Xové a PostScriptové príkazy vytvoria PostScriptové obrázky a L<sup>A</sup>T<sub>E</sub>Xový zdrojový kód popisu týchto obrázkov spolu s príkazmi na zaradenie obrázkov na príslušné miesta v L<sup>A</sup>T<sub>E</sub>Xovom kóde. Tesne pred dokončením tohto článku som sa dozvedel, že existuje makro (plaincad.tex), ktoré by umožnilo prekladať tieto obrázky aj plainTeXom.

S týmto článkom Vám chcem priblížiť moje starosti a radosti pri hľadaní spôsobu, ako využiť TeX aj na úplne iný účel než bol určený – na kreslenie obrázkov. K napísaniu nasledujúcich pár strán ma viac-menej vyprovokovala *Anketa užívateľů TeXu* od pána Polácha. Osobne používam TeX na všetko možné, ale tento grafický balík mi narobil najviac problémov. Vďaka TeXu a PostScriptu ho používam do dnešných dní a fakticky s ním vytváram všetky svoje obrázky.

### GPVTV

Ešte dakedy na začiatku 80tych rokov minulého storočia vznikol na pôde Výpočtového strediska Slovenskej akadémie vied (VS SAV) súbor podprogramov napísaných v jazyku FORTRAN-77, vhodný pre grafické vyhodnotenie vedecko-technických výpočtov. Dostal názov GPVTV (zrejme skratka **G**rafické **P**odprogramy pre **V**edecko-**T**echnické **V**ýpočty). Celkový počet podprogramov je okolo 150 – od jednoduchých čiar až po konštruovanie a zobrazenie trojrozmerných scén. Posledné prídavky/úpravy sa datujú na začiatok 90tych rokov. Pamätám sa, že väčšina programátorov, ktorí chodili počítať do VS SAV tento grafický balík používala. Po objavení sa rôznych grafických programov balík rýchlo strácal na popularite. Bolo to predovšetkým preto, lebo nepoužíval žiadne fonty – obrázky i text boli vektorovo zapísané dvoma príkazmi (MOVE a LINE). Dalo sa síce všetko otáčať, zmenšovať, zväčšovať, skláňať – ale všetko pozostávalo len z jednoduchých čiar. Ja som sa ho tak ľahko nechcel vzdať, lebo som veľmi oceňoval (a dodnes oceňujem) jeho programovateľnosť.

### Prekódovanie do L<sup>A</sup>T<sub>E</sub>Xu

Na prekódovanie výstupu z GPVTV do L<sup>A</sup>T<sub>E</sub>Xového kódu – s pomocou EMLINE – stačil celkom jednoduchý program. Po preštudovaní štruktúry grafického súboru som bol schopný napísať program, ktorý správne prečítal jeho obsah. Potom stačilo príkazy MOVE a LINE správne rozložiť do príkazov EMLINE v prostredí PICTURE. Aby

som vyrobil krajšie obrázky, začal som ich programovať vo FORTRANe bez popisiek, ktoré som potom ručne dopisoval v L<sup>A</sup>T<sub>E</sub>Xu.

Bolo to trochu nepraktické a hneď som myslel nato, že by bolo výborné, keby som mohol L<sup>A</sup>T<sub>E</sub>Xové príkazy písať priamo vo FORTRANovskom zdrojáku. Lenže zdrojové kódy balíka GPVTV neboli k dispozícii a ja som chcel, aby grafika L<sup>A</sup>T<sub>E</sub>Xové príkazy zaregistrovala ako súčasť balíka. Chcel som, aby tie príkazy fungovali ako FORTRANovské premenné, na ktoré môžem aplikovať matematické operátory a mohol ich aj ukladať do indexovaných dátových polí (napr. aby som mohol bez problémov vypísať L<sup>A</sup>T<sub>E</sub>Xom riadiacu premennú FORTRANovského cyklu).

### **Pridávanie L<sup>A</sup>T<sub>E</sub>Xových príkazov**

Skúšal som ťažkopádne a krkolomné veci, až nakoniec som prišiel nato, že to pôjde celkom jednoducho. Využijem jeden veľmi krátky a už nepotrebný podprogram z balíku GPVTV, ktorý mal jediný parameter na výber pera pre kresliace zariadenie (DIGI-GRAF). Každý nový podprogram označím na začiatku s týmto nepotrebným podprogramom (pre každý nový podprogram iné číslo) a dám ho zapísať do grafického súboru. V prekódovacom programe ich nájdem podľa rôznych čísiel tej istej inštrukcie (výber pera). Potom to už išlo pomerne rýchlo. Vytvoril som si okolo desať podprogramov na zápis textu, čísla, na popis osí (aj logaritmických, dátumových, popisy s opačným smerom, atď.). Obmedzenia DOSu a EMLINE som dokázal obísť a mohol som vytvárať aj obrovské grafické súbory.

Bol som celkom spokojný až nato, že som nemohol pohodlne otáčať popisky. Písal sa rok 1994 a o PostScripte som vedel iba to, že to je v tlačiarňach. Musel som vtedy pristúpiť na vetvenie súboru. Vo FORTRANe som si vytvoril podprogram na vertikálny zápis (stačili mi 90 a 270 stupňové otočenia). Bol taký istý ako ten na horizontálny zápis (veľmi krátky – iba štyri riadky), len mal iné číslo. Vetvenie bolo zariadené až v prekódovacom programe. Keď sa v grafickom súbore našla inštrukcia na výpis vertikálneho textu, tak do L<sup>A</sup>T<sub>E</sub>Xového súboru sa zapísal príkaz na vloženie obrázka (text1 – 01.pcx, text2 – 02.pcx, text3 – 03.pcx...) a do pomocného súboru sa zapisovali jednotlivé zvislé texty (každý text na samostatnú stranu). Po prekódovaní GPVTV súboru do L<sup>A</sup>T<sub>E</sub>Xového sa muselo pristúpiť na preklad pomocného súboru a hneď po ňom na výrobu PCX obrázkov s pomocou E. Mattesových (EM) ovládačov. Až po týchto úkonoch prišiel na rad preklad L<sup>A</sup>T<sub>E</sub>Xového súboru, ktorý sa medzitým vygeneroval. Išlo to celkom svižne aj s viacerými obrázkami v jednom súbore (samostatné strany), hoci cesta z editora FORTRANovského zdrojáku po DVISCR bola celkom kľukatá. Vyzerala takto: editor, preklad FORTRANovského kódu, zlinkovanie a vytvorenie spustiteľného programu, spustenie programu, prekódovanie GPVTV výstupu do L<sup>A</sup>T<sub>E</sub>Xového kódu, prípadná výroba vertikálnych popisiek, preklad T<sub>E</sub>Xom, DVISCR. Napriek tomu som videl drvivú väčšinu obrázkov v priebehu 2–4 sekúnd.

Takto som teda vytváral svoje obrázky od začiatku roku 1995. Niektoré som vkladal

priamo v  $\text{\LaTeX}$ ovom kóde (cez príkaz `\input`), ale väčšinou som ich prekódoval do grafického formátu PCX cez EM ovládače. Bol som vcelku spokojný – išlo to celkom svižne, diakritikami fungovala, mal som v  $\text{\LaTeX}$ ovom súbore vložené obrázky s  $\text{\LaTeX}$ ovými popiskami, mohol som už zmeniť hrúbku čiary. Aj v GPVTV boli nato príkazy, lenže nikdy sa mi to nepodarilo – iba som to imitoval s viacerými čiarami. Hoci GPVTV umožňoval aj prácu s farbami, do  $\text{\LaTeX}$ u som to vtedy nemohol vložiť (a vtedy neboli ani také požiadavky).

### Prekódovanie do PostScriptu cez DVIPS

Prišiel rok 1997 a na  $\text{\TeX}$  konferencii sa množili otázky a odpovede o PostScripte. Dozvedel som sa veľa zaujímavých vecí – ako dostať DVI do PostScriptu, ako vkladať PostScriptové obrázky do  $\text{\LaTeX}$ u, keď chcem dostať do  $\text{\LaTeX}$ u farebné obrázky, tak cez PostScript, atď. Nainštaloval som si DVIPS a mohol som mať aj obrázky v PostScripte. Niekedy v septembri 1998 som si stiahol zo siete krátku príručku – Rudolf Musil: *Základy programování v PostScriptu*. Aby som trochu pochopil PostScript, začal som skúšať jednotlivé základné príkazy, ktoré som chcel zamontovať do obrázkov tak, aby to DVIPS vedel správne vyrobiť. Keď mi to už ako tak išlo, skúsil som vyrábať FORTRANovské podprogramy na PostScriptové príkazy. Tu ma čakalo sklamanie. Okrem nastavenia a zmeny šedosti a farby sa mi nepodarilo donútiť DVIPS, aby správne interpretoval zapísané príkazy cez EMLINE. Snažil som sa, ale po čase som pochopil, že to nepôjde. Videl som nemalé možnosti PostScriptu a tak som si povedal, že to skúsím priamo prekódovať do PostScriptu. Lámal som si dlho hlavu, ako potom tam dostanem  $\text{\LaTeX}$ , ako zosúladiť grafiku FORTRANu a PostScriptu s  $\text{\LaTeX}$ om. Lákalo ma to aj preto, lebo by potom DVIPS prekódoval iba  $\text{\LaTeX}$ ovú časť obrázkov – bolo by to teda rýchlejšie.

### Prekódovanie priamo do PostScriptu

Grafika FORTRANu a  $\text{\LaTeX}$ u sa tvorí z ľavého horného rohu, PostScriptu z ľavého dolného – čo mi veľmi sťažilo situáciu. Grafický kód GPVTV nebol až taký problém – tam sú len dva vykonávajúce príkazy, MOVE a LINE. Horšie bolo to všetko zosúladiť s  $\text{\LaTeX}$ om. Najprv som začal skúšať prekódovanie do príkazov `\special{ps: ...}`. Obrázky sa mi rozladili a ani na viaceré pokusy sa mi nepodarilo týmto spôsobom ich vytvárať. Zase som bol sklamaný, že som to nedokázal. Po dlhšom čase ma napadlo (niekedy pred Vianocami 1999), že čo sa stane keď to skúsím oddeliť – PostScript do jedného súboru a  $\text{\LaTeX}$  do druhého a potom ich dajako spojím. Takže všetky MOVE a LINE do jedného PostScriptového súboru a výstup z  $\text{\LaTeX}$ ových podprogramov do druhého, v ktorom bude aj príkaz na vloženie PostScriptového obrázka. Trošku mi to trvalo kým som to zosúladiť. Potom som už mohol pridávať PostScriptové príkazy do FORTRANovského zdrojáku. To už išlo rýchlo a konečne som mohol obdivovať krásu a presnosť PostScriptových obrázkov, ktoré boli zrazu omnoho menšie – zaberali

menej miesta na disku.

Hneď som si uvedomil, že ušetrím ešte ďaleko viac, keď sa mi podarí implementovať priame PostScriptové príkazy, ktoré vykonáva až samotný PostScript. Najlepšie to vysvetlím na kruhu. Vo FORTRANe si vyrobím kruh, ktorý vytvorí jeden príkaz MOVE a okolo 360 až 720 (niekedy aj viac) príkazov LINE. Keď sa mi podarí kruh zapísať PostScriptovým príkazom, tak bude stačiť jeden jediný príkaz a to je obrovská úspora. Takisto rôzne trhané čiary (čiarkovaná, bodkovaná., atď.). Keď som vo FORTRANe ťahal čiarkovanú čiaru z jedného bodu do druhého, tak to bolo zapísané viacerými príkazmi (podľa dĺžky ich mohlo byť aj viac než 100) MOVE LINE, MOVE LINE. Keď nastavím typ čiary priamo PostScriptovým príkazom, tak to bude len jediné MOVE LINE. Tak som postupne pridával podprogramy na PostScriptové príkazy. S potešením som potom zisťoval, že niektoré súbory sa mi zmenšili až neskutočne, teda aj výroba obrázkov bola rýchlejšia. Nakoniec som pochopil PostScript natolik (preštudoval som aj článok vo Zpravodaji 1–3 2001 – Arnošt Štědrý: *PostScript pro programátory, vědce i inženýry*), že som mohol naprogramovať aj rotujúci  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ový text. V súčasnosti mám vo FORTRANe okolo 20 PostScriptových podprogramov. Môžem zapísať všetky PostScriptové príkazy (niečo ako `\special{...}` v  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u), ale to je pre užívateľa zložitejšie, lebo tento spôsob vyžaduje určité znalosti PostScriptu.

### Rýchlosť vytvárania obrázkov

Ako rýchlo vytvoríme obrázok(y) záleží na viacerých faktoroch. Aký máme editor, ako vieme programovať, aký rýchly máme počítač, aká rýchla je tá aplikácia, či je vhodná na náš obrázok, atď. FORTRANovský kód môžeme zapísať ľubovoľným editorom, ale pokladal som za veľmi dôležité, aby bol trošku prispôbostený k FORTRANu. Hneď ako som dostal prvé PC (1988), naprogramoval som si taký editor a zabezpečil, aby som mohol priamo z neho pustiť preklad, link a samotný beh programu. Svoje programy si píšem priamo z hlavy, takže môžem s ním za 1–3 minúty vyprodukovať aj 100 riadkový kód, ktorý na drvivú väčšinu obrázkov stačí.

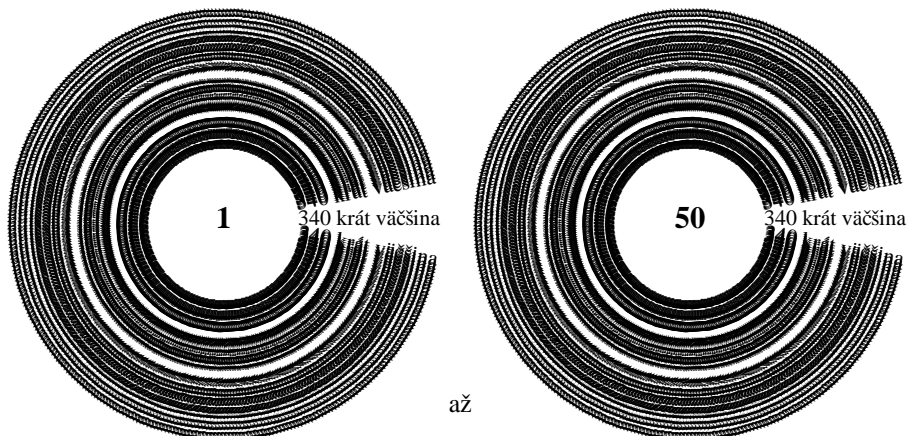
Druhá vec je vymyslieť ten kód. V tomto nám nepomôže sebestlepší editor, najrýchlejší počítač, sebestlepšia aplikácia – musíme to jednoducho naučiť a skúšať, aby to druhý krát išlo rýchlejšie. Nadarmo budem schopný za minútu napísať 50 riadkov, keď tých 50 riadkov zfunkčím a odladím za tri dni. Niekedy sa nám podarí za minútu vyprodukovať 100 obrázkov a niekedy sa potrápime s jedným obrázkom aj 100 minút a viac.

Hoci výroba obrázkov FORTRANom od editora až po GhostView (GV) je dlhá – 1. editor, 2. preklad, 3. link a vytvorenie spustiteľného programu, 4. spustenie a výroba GPVTV súboru, 5. prekódovanie do  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u a PostScriptu, 6. preklad  $\text{T}_{\text{E}}\text{X}$ om, 7. DVIPS, 8. GV – zdá sa mi však celkom svižná. Napadlo ma, že skúsím porovnať tento môj postup s rýchlosťou METAPOSTu. Vedel som, že aj v METAPOSTe sa dá niečo zapísať, čo prejde  $\text{T}_{\text{E}}\text{X}$ om. Tu je cesta z editora po GV kratšia – 1. editor, 2. preklad

METAPOSTom do PostScriptu a do  $\TeX$ u, 3. preklad  $\TeX$ om, 4. DVIPS, 5. GV. Krok číslo 2 je podrobne vysvetlený v nižšie spomenutom článku o METAPOSTe.

### Malé porovnanie s METAPOSTom

Najviac spomaľujú výrobu obrázkov  $\TeX$ ovské popisky a z nich hlavne rotujúce. Vymyslel som si jednoduchý obrázok, kde bude 340 krát vypísaný rotujúci text. S týmto postupom sa vyrobí 50 obrázkov tak (každý na novej strane), že v strede bude vy-



### Zdrojový kód

```
cgtpc
  character a*20,t*200
  call openg
  a='\bf'
  t='\hspace*{17mm}340 krát väčšina'
  do 2 j=1,50
    call begpic(11,10,3)
    do 1 k=12,350
      r=k
1    call larget(0.,0.,'1',t,r)
    call larget(0.,0.,'1',t,0.1)
    call latex(0.,0.,'c',a,j)
    call move(-7.,4.)
2    call eopic
  end
```

### Komentár

cesta od 1(editor) po 8(GV) zabezpečená  
definovanie premenných pre text a číslo  
otvorenie grafického súboru  
font pre číslo  
 $\LaTeX$ ový text do premennej t  
otvorenie cyklu pre obrázky  
otvorenie obrázka v grafickom súbore  
otvorenie cyklu pre rotáciu  
do premennej r (real) sa vloží k (integer)  
výpis textu s rotáciou r a koniec cyklu 1  
výpis textu s rotáciou 0.1  
výpis čísla j (integer)  
posunutie obrázka  
koniec obrázka a koniec cyklu 2  
koniec programu

Obr. 1. porovnávacie obrázky

písané číslo obrázka (tiež  $\text{\LaTeX}$ om). Rotujúci text (*340 krát väčšina*) sa zapísal teda  $50 \times 340 = 17000$  krát. Obr. 1 znázorňuje prvý a posledný obrázok, pod ktorým je FORTRANovský zdrojový kód s krátkym komentárom. V editore som vygeneroval dva cykly a do vnútorného som zapísal text s otočením o riadiacu premennú cyklu. Do vonkajšieho cyklu som zapísal text s nulovou rotáciou a do stredu som zapísal riadiacu premennú vonkajšieho cyklu. Behom pár minút som to mal napísané, odladené, vyskúšané.

Nemám nainštalovaný METAPOST a mám s ním nulové skúsenosti. Z  $\text{\TeX}$  konferencie a aj zo ZPRAVODAJA 1–3/2001 (Miroslava Krátká: METAPOST a mfpic) viem, že je to veľmi mocný nástroj. Kým by som sa dopracoval k METAPOSTovým kódom, tak by možno prešiel aj mesiac. Napísal som preto pánovi Kubenovi a potom aj pánovi Klocovi, či by neboli ochotní vyrobiť podobný súbor s pomocou METAPOSTu. Obidvaja sa zachovali ako praví  $\text{\TeX}$ isti a prisľúbili, že mi pošlú riešenia. Nevedel som, že METAPOST nemá nástroje na stránkovanie (nedá sa jednoducho vyrobiť viac obrázkov do jedného súboru – METAPOST vytvára bez problémov aj viac obrázkov, ale vždy ich dáva do samostatných súborov) a tak nechtiac som ich obral o viac času, než som si myslel (za ich námahu a pochopenie im aj touto cestou ďakujem). Pán Kuben mi poslal tri riešenia na tom istom stroji a pán Kloc jedno riešenie na troch rôznych strojoch – čiastočne preto, lebo som mu napísal, aby zbytočne neštrácal čas a v tom nepokračoval. Výsledky som zhrnul do Tab. 1.

Ťažko sa tieto postupy porovnávajú, bol som len zvedavý, či môj postup nezaostáva rádovo od postupu cez METAPOST. Z tej tabuľky dobre vidieť, že počítačový čas za chvíľu nebude hrať žiadnu rolu – procesory nad 2GHz to už zvládnu bleskovo. Pri tých pomalších je situácia iná, tam záleží viac na veľkosti spracovaného súboru, operačnej pamäti, rýchlosti disku, zbernice, systému, od použitých verzií programov atď. Pán Šurda vytvoril čisto plain $\text{\TeX}$ ový kód, ktorý som bez problémov preložil na svojom počítači (predposledný riadok tabuľky). Preklad  $\text{\TeX}$ om v tomto prípade je veľmi rýchly, lebo sa prekladá malý súbor a kvôli zaujímavosti som ho zaradil do tabuľky. Na poslednom riadku uvádzam výsledok môjho snaženia optimalizovať jednotlivé kroky výroby obrázka. V tomto prípade však to nemá príliš veľký význam, lebo obyčajne potrebujeme malé množstvo rotujúceho textu. Vyskúšal som otvorenie všetkých piatich PostScriptových súborov v GV s nasledujúcim výsledkom: 4s, 5s, 6s, 7s, 13s.

### Možnosti a použitie

Skúsim v krátkosti naznačiť možnosti a použitie grafického balíka GPVTV obohateného o  $\text{\LaTeX}$ ové a PostScriptové príkazy.

1. balík umožňuje zapísať vo FORTRANovskom zdrojáku  $\text{\LaTeX}$ ové a PostScriptové príkazy.  $\text{\LaTeX}$ ové sú implementované ako FORTRANovské textové a číselné premenné, teda sú aj FORTRANovsky programovateľné, čo môžeme využiť nie-

Tab. 1. Porovnávací tabuľka  $\text{\TeX}$ ovej rýchlosti METAPOSTu a FORTRANovského grafického balíka GPVTV.

N	PC MHz/RAM	sys	2-5/2 [s]	$\text{\TeX}$ [s]	DVIPS [s]	GV [s]	Total [s]	zdroj [bajt]	PS [MB]	ZIP [kB]
LB	590/192	W98/D	4	8	2	6	20	265	4.2	99
LB	1000/256	W00/D	3	5	2	4	14	265	4.2	99
LB	2200/256	WXP/D	2	3	1	2	8	265	4.2	99
JK	2400/512	eCS1.1	2	2	2	2	8	2615	8.2	1117
JK	2400/512	eCS1.1	2	2	2	2	8	545	8.2	1117
JK	2400/512	eCS1.1	2	2	2	2	8	950	8.2	1117
LK	2400/512	eCS1.1/D	4					*500	21.9	2836
LK	900/256	W00	7					*500	21.9	2836
LK	90/64	Linux	55					*500	21.9	2836
LB	133/16	W98/D	17					265	4.2	99
AŠ	590/192	W98/D		2	2	5	9	300	3.2	130
LB	590/192	W98/D	2	4	2	4	12	265	2.5	94

Poznámky k jednotlivým stĺpcom:

N (name) – LB-Ladislav Bittó, JK-Jaromír Kuben, LK-Luboš Kloc, AŠ-Anton Šurda

PC – rýchlosť procesora/operačná pamäť

sys – operačný systém, /D znamená DOS okno (okrem GV všetko DOSové programy)

2–5/2 – kroky 2–5 cez FORTRAN/krok 2 cez METAPOST

$\text{\TeX}$ , DVIPS, GV, Total nepotrebuje komentár

zdroj – veľkosť zdrojového kódu, \* znamená neúplné riešenie

PS – veľkosť výsledného súboru 50tych obrázkov

ZIP – veľkosť archivovaného súboru 50tych obrázkov.

len na popisky. Napríklad: pracujem na obrázku, kde by som chcel vykresliť 10 rôznych  $\text{\TeX}$ ovských tabuliek do 10tych rôznych bodov a to tak, že ani neopustím FORTRANovský zdroják. Údaje ešte ani nepoznám, budú vyrátané. Dám ich vygenerovať do 10tych rôznych súborov a v obrázku ich vložím do príslušných bodov cez `\input`, alebo vygenerujem do jedného súboru, ako rôzne makrá. Môžem ich dokonca skôr vykresliť, než vygenerovať – na poradí nezáleží, lebo sa  $\text{\TeX}$ uje až po prekódovaní grafického výstupu z GPVTV. PostScriptové príkazy sú generované vo FORTRANovských podprogramoch, ktorých parametre sú FORTRANovsky programovateľné.

2. pohodlná tvorba viacerých obrázkov do jedného súboru (obrázky sú na samostatných stranách)
3. pohodlná tvorba viacerých obrázkov na jednu stranu – umožňuje to grafický

systém GPVTV s posuvom súradnej sústavy

4. pohodlná práca s množstvom dátových súborov
5. Vyhodnotenie vedecko-technických výpočtov. Tento pojem je veľmi široký. GPVTV má okolo 150 podprogramov, z toho je väčšina už nepotrebná (rôzne ovládače a iné), ale okolo 50–60 môžem aj teraz využívať. Sú tam podprogramy na 2-D a aj na 3-D grafiku. Ja osobne využívam z GPVTV iba 4–5 výkonných podprogramov a okolo 10, ktoré sú potrebné na definovanie grafického systému a súradnej sústavy.
6. obrázky môžeme vytvárať aj priamo vo výpočtových programoch
7. grafické vyhodnotenie rôznych registračných (dátových) záznamov
8. kreslenie diagramov (aj živých – kde čiary idú s objektom), elektrických schém, pohyby planét, planétok, komét po oblohe, hviezdne mapy, ročné priebehy časovej rovnice, atď.
9. balík je súčasťou programovacieho jazyka, teda možno vytvoriť aplikáciu (vykonávateľný súbor s príponou EXE). Takto s ním môže vyrábať obrázok aj ten, kto nepozná ani FORTRAN, ani  $\text{\LaTeX}$  a ani PostScript (s pomocou dátových súborov).
10. presné ručné kreslenie. Trošku som si prispôbil editor a vytvoril pohyblivý milimetrák, aby som mohol pohodlne kresliť aj také obrázky, ktoré sa nedajú programovať. Teraz už ani nemusím väčšinou nakresliť obrázok na milimetrový papier, ale rovno na obrazovku. Začínam tak, že si vykreslím milimetrák na obrazovku (v jej hornej časti) a postupne kreslím objekty (v dolnej časti obrazovky) vo FORTRANovskom zdrojáku. K úplnej spokojnosti mi chýba už len to, aby GV pracoval aj vtedy keď nie je aktívny – naskytla by sa možnosť vidieť pridávanie príkazov okamžite (čo by som si doprogramoval do editora).
11. umožňuje proste všetko, čo sa dá vo FORTRANe s pomocou GPVTV a s mojimi  $\text{\LaTeX}$ ovými a PostScriptovými podprogramami naprogramovať. Môžem napísať a ľahko pridať ďalšie podprogramy, keď to bude potrebné.

Osobne týmto balíkom vytváram 99.5% svojich obrázkov a za 20 rokov som s ním nakreslil (a pomohol nakresliť) viac než dvestotisíc. Niekomu sa môže zdať tento spôsob zložitý, ale snád' z kódu tých porovnávacích obrázkov je zrejmé, že pre užívateľa to nie je až také zložené. Pracujú s ním aj kolegovia, ktorí ináč nepoužívajú  $\text{\TeX}$  (paradoxne je ich viac) a PostScript je pre nich neznáma veličina. FORTRAN bol vymyslený pre vedcov a patrí medzi najjednoduchšie programovacie jazyky (výsledný kód medzi najrýchlejšie). Moje vedomosti o FORTRANe sú slabé, z GPVTV využívam len malú časť, z  $\text{\LaTeX}$ u ovládam minimum a v PostScripte som na samom začiatku – čo mi však nebránilo vo vytváraní podporných programov pre svoj postup. Aj z mála sa niekedy dajú robiť zložité veci – treba len chcieť a to málo ovládať, ostatné príde.



## **Záverom**

Teraz by sa patrilo napísať, kde je ten balík k dispozícii (aj keď sme presvedčení, že on by nebol veľký záujem). Skúsil som zohnať zdrojové kódy podprogramov spolu s textom manuálu k GPVTV, ale nepodarilo sa mi to (nenašli sa). Ja mám len zlinkovanú DOSovú verziu – čo môžem s povolením VS SAV vystaviť – a jednu príručku o GPVTV (má 170 strán). Berte preto tento článok ako propagáciu T<sub>E</sub>Xu. Keď sa niečo podarí s T<sub>E</sub>Xom vytvoriť, treba sa o tom podeliť. Viem ako ťažké je v súčasnosti k nemu niekoho pritiahnuť. Okolo 1990 to bolo ešte niečo iné. Vytvoril som vtedy špeciálny editor pre T<sub>E</sub>X a oboznámil som s tým svoje okolie. Bol som rád, že som s tým pritiahol niekoľkých ľudí k práci v tomto sádzacom systéme. Zastavovali ma úplne neznámi ľudia, či by som im nenainštaloval T<sub>E</sub>X s tým svojím editorom. Sem tam som musel robiť aj školenia. Pár krát sa mi stalo, že som nevedel kam skôr ísť. Ľudia však idú s menším odporom, prišiel WORD a ostatné poznáme.

Som od prírody veľmi lenivý, takže stále vymýšľam, že ako by som si mohol ušetriť prácu. Niekedy sa natrápim a je to k ničomu, ale nestážujem sa. Naučil som sa, že chyby mám hľadať predovšetkým u seba a keď ich vyriešim vlastnými silami, tak som spokojný. Keď sa chceme pustiť do rôznych projektov, tak najdôležitejší je vždy (a obvyčajne aj najťažší) prvý krok s ktorým pustíme na seba tú lavínu problémov. Potom už je len na nás, či to ľahko vzdáme alebo budeme riešiť jeden problém za druhým.

Niekoho môže napadnúť, že načo sa s tým trápim, keď na svete je mnoho programov rôzneho druhu – treba si len vybrať, nainštalovať (niekedy poriadne zaplatiť) a už to aj funguje. Lenže všetko čo je krásne pre človeka, rastie pomaly. Je to trošku aj moje dieťa, takže ho mám rád. Do jeho výchovy mi nikto nezasahuje a teším sa zo všetkého, čo som schopný ho naučiť. Pár rokov ešte potrvá, kým dorastie (ak vôbec). Dôležité je, aby človek mal s čím bojovať, mal čo objavovať pre seba, potom sa môže postarať aj o to najcennejšie čo má – o svoje zdravie.

## **PodĎakovanie**

Autor článku ďakuje pánovi Kubenovi a pánovi Klocovi za METAPOSTové riešenie porovnávacích obrázkov, za poskytnutie ich zdrojových kódov a výsledného PostScriptového súboru a za ich pripomienky k článku. Moja vďaka patrí aj pánovi Šurdovi, ktorý prejavil značnú dávku trpezlivosti a ochoty pri opravovaní mojej nie celkom spisovnej slovenčiny (taká ľudová made in Hungary) a hlavne poradil čo pridať/vypustiť, aby článok bol zrozumiteľný pre každého.

Bittó Ladislav  
December 2003