

Sekce Othersubrs

Jak bylo uvedeno dříve, je jazyk písem typu 1 jen malou podmnožinou jazyka PostScript a zpracovává se také jiným interpretem (totiž rastrovačem písem typu 1, zabudovaným v interpretu); jinými slovy příkazy jazyka PostScript není možno v definici písma typu 1 použít. Přesto se ale časem ukázala tato možnost jako potřebná. Její implementace je však jistý problém: jak víme, písma typu 1 lze rastrovat nejenom interpretem jazyka PostScript (resp. v něm zabudovaným rastrovačem písem typu 1), ale také např. pomocí ATM (což je víceméně samostatný rastrovač písem typu 1). Pokud by ale písmo obsahovalo příkazy jazyka PostScript, bylo by jej možno rastrovat pouze úplným interpretem PostScriptu; nebylo by jej možno rastrovat pomocí ATM a tak by se z něj vlastně stalo písmo typu 3.

Tento problém vyřešila firma Adobe elegantním způsobem: definice písma typu 1 byla rozšířena o sekci OtherSubrs, což je pole, jehož jednotlivými prvky jsou *procedury* v jazyce PostScript (nepleťte prosím s *rutinami*, které jsou v sekci Subrs); tyto procedury lze z definice litery volat operátorem `callothersubr`. Funkce a použití jednotlivých procedur je ovšem přesně definováno (jinými slovy návrhář písma nemůže vytvořit své vlastní procedury; může použít jen předdefinované); postupem času přitom může firma Adobe definovat podle potřeby některé nové procedury. Z jiného úhlu pohledu můžeme volání konkrétní procedury ze sekce OtherSubrs chápat také jako nový operátor jazyka typu 1. Zpracování těchto procedur je následující:

- je-li písmo rastrováno v interpretu jazyka PostScript, provede *kód postscriptové procedury* samotný interpret PostScriptu. Jinými slovy, tyto procedury se provedou i tehdy, kdy v době vzniku tiskárny (tedy interpretu) ještě nebyla jejich funkce definována.
- je-li písmo rastrováno v ATM (nebo jiném samostatném rastrovači, jako je např. Font Foundry), potom se všechny procedury realizují tak, že se *místo nich* provede kód, který je přímo součástí ATM a jehož funkce je shodná s odpovídajícím postscriptovým kódem (ten se tedy ignoruje). Takto je pochopitelně možno realizovat jen ty procedury, které byly definovány v době vzniku ATM, nikoli ty, které vznikly později.
- je-li písmo rastrováno v ATM a příslušná procedura v něm není zabudována (neboť nebyla v době jeho vzniku ještě definována), tak ji ATM jednoduše vynechá. Důsledkem bude nejčastěji horší kvalita vyrastrovaného písma; toto jisté omezení je ale v praxi spíše teoretické, neboť ATM je (na rozdíl od tiskárny) snadné vyměnit za novější verzi a často se to také děje, už vzhledem k neustálým změnám a vylepšením operačních systémů, v nichž ATM pracuje (stačí vzpomenout na řadu změn jak na počítačích Macintosh počínaje systémem verze 6, přes verzi 7, 7.1 až k 7.5, nebo k řadě rozdílů mezi Windows 3.1, 3.11 a chystanou verzí 4).

Tato implementace procedur jazyka PostScript tedy umožňuje víceméně neustále doplňovat písma typu 1 o nové funkce a přitom zůstat kompatibilní se stávajícími tiskárnami či samostatnými rastrovači. To umožnilo firmě Adobe celou řadu vylepšení původní definice jazyka typu 1, ať už se jedná o změnu pokynů, funkci flex nebo dokonce o písma MultipleMaster. V tomto článku zmíníme pouze první dvě.

Spolu se sekci OtherSubrs a operátorem `callothersubr` je ještě třeba zmínit operátor `pop`, který převede výsledek postscriptové procedury ze zásobníku interpretu jazyka PostScript do zásobníku rastrovače, a dále operátor `setcurrentpoint`. Zájemce najde podrobnější informace v literatuře.

Změna pokynů

Jak bylo uvedeno dříve, nemohou se svíslé či vodorovné pásy, definované pokyny `vstem` a `vstem3` resp. `hstem` a `hstem3`, navzájem překrývat. To je jisté omezení, které může být v praxi na obtíž. Povšimněte si detailu litery E na obrázku: vodorovný pokyn (`hstem`) potřebný pro vodorovný tah litery se překrývá s užším vodorovným pokynem

potřebným pro vykreslení serifu. Toto je poměrně častá situace; vyskytuje se mimo jiné běžně v případě svislých pásů u liter s akcenty. Řešením je funkce označovaná jako změna pokynů (hint replacement). Při popisu tvaru litery nejprve definujeme nejdůležitější pokyny, které se vzájemně nepřekrývají (tedy například pokyn pro vodorovný tah litery na obrázku). Pokyny, které se s těmito hlavními pokyny překrývají (a které jsou méně významné) zapíšeme *mimo* tělo popisu litery jako samostatnou rutinu do sekce Subrs (např. č. 10). V okamžiku, kdy budeme vykreslovat část litery, pro níž tyto vedlejší pokyny potřebujeme (například serif litery z našeho obrázku), zavoláme postscriptovou proceduru číslo 3 (ze sekce OtherSubrs), které jako parametr předáme číslo rutiny (ze sekce Subrs) s vedlejšími pokyny - v našem případě tedy 10:

```
10 1 3 callothersubr pop callsubr
```

(10 je předávaný parametr, 1 je počet parametrů). Postscriptová procedura 3 potom:

- je-li písmo rastrováno v tiskárně, která umí měnit pokyny, tak zruší stávající pokyny a vrátí svůj parametr, tedy číslo rutiny ze sekce Subrs (v našem případě 10). Operátorem pop se tento parametr přenesou ze zásobníku interpretu jazyka PostScript na zásobník rastrovače a následující callsubr zavolá příslušnou rutinu ze sekce Subrs (v našem případě č. 10), která obsahuje nové pokyny.
- pokud tiskárna pokyny měnit neumí, neprovede postscriptová procedura nic a vrátí číslo 3, což je vždy prázdná rutina ze sekce Subrs. Ta se potom zavolá posloupností pop callsubr. Staré pokyny tedy zůstanou v platnosti.

Během kresby litery můžeme měnit pokyny mnohokrát. Je také třeba poznamenat, že všechny současné interprety PostScriptu firmy Adobe měnit pokyny umějí. Tuto funkci neměly jen úplně první modely postscriptových tiskáren.

Funkce flex

Funkce flex (Flex Feature) je určena pro vykreslení jemných oblouků, které jsou vodorovné nebo svislé, například spodních okrajů serifů. V malých velikostech vyrastrovaného písma (vzhledem k rozlišovací schopnosti zařízení) je totiž vhodné tyto oblouky nahradit rovnou čarou, protože jinak by oblouk (s hloubkou jednoho pixlu, což je minimum) vypadal opticky příliš velký. Oblouk musí splňovat některé předpoklady: musí být vodorovný nebo svislý (shodná y-ová nebo x-ová souřadnice koncových bodů), musí se skládat ze dvou křivek a jeho hloubka musí být menší než 20 (v soustavě 1 000 x 1 000) a menší než BlueShift. Hlavní myšlenka spočívá v tom, že oblouk (tedy dvě křivky popsané dvěma operátory rrcurveto), nahradíme *myšlenou* lomenou čarou, která prochází koncovými body tečen těchto křivek, tedy sedmi operátory rmoveto. Za každým operátorem rmoveto přitom zavoláme ještě rutinu 2 ze sekce Subrs. Oblouk bude tedy popsán sedmi příkazy typu 30 0 rmoveto 2 callsubr

Rutina 2 volá přímo postscriptovou proceduru 1 (má tedy tvar 0 2 callothersubr), která myšlenou čáru buď nahradí dvěma oblouky (ve větších velikostech litery), nebo rovnou čarou při menších velikostech.

Na závěr poznamenejme, že pokyny, jejich záměna a funkce flex se v kvalitních písmech firmy Adobe používají velmi často (pokyny a jejich záměna jsou naprosto v každém písmu) a jejich absence jednoznačně ukazuje na nevhodné nástroje, použité pro tvorbu písma. Pomocí nich lze také snadno rozlišit technicky kvalitní tiskové písmo od obyčejného fontu.