Zpravodaj Československého sdružení uživatelů TeXu

Vít Zýka

TeX a PDF

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 12 (2002), No. 3-4, 140–151

Persistent URL: http://dml.cz/dmlcz/149899

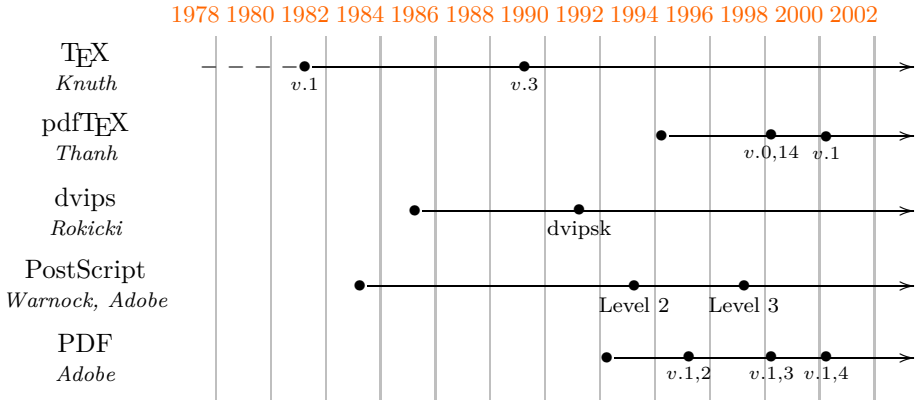
TEX a PDF[1]

Vít Zýka

Program TEX je systém pro velmi kvalitní počítačovou sazbu vyznačující se přenositelností zdrojového kódu mezi platformami i v čase. Portable Document Format (PDF) je formát pro elektronické publikování. Tento příspěvek ukazuje možnosti elektronického publikování v PDF uživateli zvyklými na TEX.

1. Troška historie

Není mým cílem zde rozebírat historii programů TEX, `dvips` a pdfTEX stejně jako historii formátů PostScript (dále jen PS) a PDF. Kdo by chtěl nahlédnout do podmíněností vzniku těchto softwarových nástrojů počítačové sazby, tomu doporučuji skvělý článek Philipa Taylora [16].



2. Co přináší PDF oproti PostScriptu?

Koncem osmdesátých let byl PS prakticky standardem pro popis vzhledu stránek. Přesto firma Adobe začala od roku 1990 pracovat na specifikaci jiného standardu. V té době již bylo zřejmé, že komunikace a výměna dokumentů prostřednictvím sítí je mnohem flexibilnější než práce s dokumenty v tištěné podobě. Chyběl však formát, který by v sobě spojoval typografickou kvalitu PS a

[1] Článek byl poprvé publikován ve sborníku Jan Kasprzak, Petr Sojka: SLT 2002, KONVOJ 2002, ISBN 80-7302-043-2, na stranách 69–77. Publikováno se souhlasem vydavatele a autorů.

hypertextové a multimediální možnosti elektronického publikování. Tyto okolnosti vedly ke vzniku PDF. Pokusme se charakterizovat jeho vlastnosti [15, 18].

1. Začneme tím, v čem je rozdílnost minimální. Je to *kreslicí model*, který se v PS osvědčil, umožňující komplexní grafický popis stránky. Oba standardy používají stejnou afinní transformaci pro převod mezi uživatelským (na výstupním zařízení nezávislým) souřadným systémem a systémem výstupního zařízení. Používat lze bitmapovou i vektorovou grafiku. Základním kamenem pro vektorový popis textu i grafiky jsou kubické Bézierovy křivky. Volit lze z různých druhů tahů, způsobů napojení a rastrování rohů. Uzavřené křivky lze vyplňovat vzory. Používat lze různé barevné modely (RGB, CMYK, CIE). Verze PDF 1.4 umí pracovat s průhledností.
2. Velké rozdíly najdeme v *jazyku* dokumentu. Zatímco PS je plnohodnotný programovací jazyk zásobníkového typu, PDF neobsahuje procedury, řídicí struktury a proměnné. Jde o seznam grafických operací. V PS se operátory vyšší úrovně vyjadřují přímo jazykem PS z elementárních grafických příkazů. V PDF musí být tyto vyšší operace přímo implementovány aplikací zpracovávající dokument. Důsledkem je efektivnější zpracování a zobrazování PDF dokumentu. Na druhou stranu je snížena flexibilita vyjádření. Například balík PSTricks [23] expanduje kreslící příkazy do PS kódu. Tento kód je relativně složitý a obsahuje programové konstrukce, např. cykly. Předpokládá se, že nakonec bude dokument zpracován nějakým PS RIPem. Protože např. pdfTEX takový RIP neobsahuje, nelze tento balík v pdfTEXu použít. Řešením je vložit takový obrázek do zvláštního dokumentu, ten zpracovat standardním TEXem, pomocí `dvips -E` vyrobit encapsulated PS, ten převést skriptem `epstopdf` využívajícím GhostScript do PDF a načíst do našeho dokumentu. O zautomatizování tohoto procesu se snaží balík PDFTRICKS [12]. Spouští při tom externí programy přímo z TEXu pomocí konstrukce `\write18{command}`. Tuto možnost nabízí např. web2c instalace TEXu. V našem případě spustíme následující tři programy:

   ```
   \write18{(la)tex tmp.tex; dvips -E tmp.dvi; ps2pdf tmp.ps}
   ```

3. PS byl původně textový kód bez jakékoliv pevné struktury. Dodatečně vydaná technická zpráva [7] doporučuje strukturní konvence PS (Document Structuring Conventions, DSC) dokumentů a značkování pomocí komentářů. Pak lze oddělit jednotlivé stránky dokumentu bez nutnosti interpretovat PS. Na přítomnosti těchto komentářů je založen balík PSUTILS [2]. Náhodný přístup k libovolné části dokumentu řeší až pevná struktura PDF. Dokument PDF se skládá z objektů, jejichž umístění v dokumentu popisuje tabulka křížových referencí. Každá stránka je také samostatným objektem a krom sazby textu obsahuje i seznam svých zdrojů (fonty, anotace), informace o velikosti stránky ap. Díky tomu stačí pro zobrazení stránky interpretovat jen ty objekty, které tvoří tuto stránku. Také je snadné s do-

kumentem manipulovat, měnit pořadí stránek a jejich velikost. Také můžeme snadno vložit stránku jako obrázek do jiného dokumentu (toho využívá balík pdfpages [9]). Další užitečnou vlastností PDF je možnost jeho jednoprůchodového generování. Je to zajištěno systémem nepřímých odkazů na objekty. Například potřebujeme-li zadat uvnitř objektu A jeho velikost, kterou před jeho uzavřením jestě neznáme, uvedeme v něm pouze nepřímý odkaz na objekt B. Objekt B bude obsahovat velikost objektu A. Příklad nejjednoduššího PDF dokumentu ukazuje obrázek 1.

```
%PDF-1.3
3 0 obj <<
/Length 101
>>
stream
1 0 0 1 91.925 759.927 cm
BT
/F51 9.963 Tf 0 0 Td[(Hallo)-250(PDF!)]TJ
211.584 -654.747 Td[(1)]TJ
ET
endstream
endobj
2 0 obj <<
/Type /Page
/Contents 3 0 R
/Resources 1 0 R
/MediaBox [0 0 595.276 841.89]
/Parent 5 0 R
>> endobj
1 0 obj <<
/Font << /F51 4 0 R >>
/ProcSet [ /PDF /Text ]
>> endobj
5 0 obj <<
/Type /Pages
/Count 1
/Kids [2 0 R]
>> endobj
6 0 obj <<
/Type /Catalog
/Pages 5 0 R
>> endobj
7 0 obj <<
/Producer (pdfTeX-0.14h)
/Creator (TeX)
/CreationDate (D:20021104154200)
>> endobj
xref
0 8
0000000004 65535 f
0000000278 00000 n
0000000167 00000 n
0000000009 00000 n
0000000000 00000 f
0000000346 00000 n
0000000403 00000 n
0000000452 00000 n
trailer
<<
/Size 8
/Root 6 0 R
/Info 7 0 R
>>
startxref
546
%%EOF
```

Obrázek 1: Nejjednodušší PDF Hallo PDF připravené pdfTEXem pomocí kódu `\pdfcompresslevel=0 \font\tenrm=tir\tenrm Hallo PDF!\end`.

4. PDF podporuje standardní kompresní formáty (JPEG, CCITT Group 3 a 4, LZW a Run Length). Jejich dekompresi musí zajistit aplikace zpracovávající PDF. Kromě výrazného zmenšení dokumentu (důležité pro přenos po síti) a nezmenšeném konfortu pro uživatele (žádné odzipovávání) je díky této podpoře snadné i vkládání komprimovaných bitmapových obrázků (TIFF, PNG, JPEG). Nevýhodou ovšem je, že nelze snadno na úrovni TEXových maker vyhledávat a měnit řetězce textu tak, jak to například dělá balík PSFRAG [3] pro editaci popisek obrázků často generovaných

programy s méně variabilním grafickým výstupem (fonty, matematická sazba). I zde by se pro pdfTEX hodila makra podobná PDFTRICKS pro automatické externí zpracování obrázku přes PS.

5. Z běžných formátů fontů umožňuje PS pracovat hlavně s Type 1 fonty. Bitmapové fonty rastrované METAFONTem jsou vkládány ve formátu Type 3. PDF zvládá oba tyto formáty, a navíc umí pracovat s True Type formátem. Musíme jen vyrobit TEXovou metriku pomocí dvojice programů `ttf2afm` a `afm2tfm`.
   Každá aplikace PDF musí obsahovat 14 základních fontů. Tyto fonty se standardně nevkládají do dokumentu. Bohužel, nemůžeme se spoléhat na to, že metriky těchto fontů jsou ve všech aplikacích stejné. Liší se dokonce verze od verze Adobe Acrobatu. Chceme-li mít jistotu, že se náš dokument všude vysází stejně, musíme i tyto fonty vložit. Zařídíme to v konfiguračním souboru fontů `psfonts.map` odkázaném v konfiguračním souboru pdfTEXu `pdftex.cfg`. Na řádku příslušného fontu provedeme dvě změny: a) změníme název fontu a b) vložíme font pomocí dvouznaku `<<`. Pro příklad uvedeme řádek pro Times-Roman:
   ```
   ptmr8r Times-RomanEmbed <8r.enc << utmr8a.pfb
   ```
6. PDF podporuje předtiskovou úpravu dokumentu (hranice media, stránky a zobrazení, tisk ořezových značek, barevné separace, ...).
7. Hlavním přínosem PDF bylo jeho interaktivní rozšíření využitelné při elektronickém zobrazení dokumentu. Jde o hypertextové prvky usnadňující orientaci (odskoky, záložky, náhledy), dále zpřístupnění videa a zvuku a možnost vyplňování a zasílání formulářů.
8. V PDF existuje podpora pro indexaci a vyhledávání slov.
9. Dokument je možné také modifikovat a to pouze připojením změněných nebo nových objektů na konec dokumentu spolu s novou tabulkou křížových referencí. Původní část dokumentu tedy zůstává nedotčena. Snadno se pak můžeme vrátit k předchozí verzi.
10. Formát byl navržen tak, aby byl systémově nezávislý, šlo jej rozšířit o nové technologie se zachováním zpětné kompatibility.

Po přečtení tohoto výčtu můžeme nabýt dojmu, že PDF má vše, co potřebujeme pro potřeby tisku, elektronického publikování nebo prezentací. Specifikace formátu je však věc jedna a jeho implementace a dostupnost vhodných nástrojů věc druhá. Sama Adobe se snažila prosadit PDF tím, že zdarma poskytla nástroj Adobe Acrobat Reader pro prohlížení a tisk PDF dokumentů na různých platformách. Díky tomu se jí také *de facto* podařilo prosadit PDF jako standard elektronického dokumentu. Přesto existují problémy, které používání formátu ztěžují. Napadají mě tyto:

1. Volně šířený Acrobat Reader ani plný Acrobat umožňující modifikovat PDF dokument neimplementuje celou specifikaci formátu (uveďme např. nastavení hlasitosti v `/Sound` anotaci, grafické znázornění `/Sound` anotace

podle jejího stavu, specifikace `/Sound` anotace podle URL, vložení video souboru pro `/Movie` anotaci interně do dokumentu). Kromě toho existují části specifikace systémově závislé (spouštění externích programů, zvukové a video anotace).

2. Implementace Acrobatu obsahuje odlišnosti od specifikace (chyby). Například obsahuje-li encoding vektor fontu vynechané nedefinované znaky, tiskne Acrobat verze pět chybně. Dále aplikování transformační matice je podle specifikace možné několika způsoby, Acrobat však rozumí jen některým. Že software obsahuje chyby, je vcelku pochopitelné, co je však pobuřující, je postoj Adobe k jejich odstraňování (a bohužel typický pro velkou komerční firmu). Znalým uživatelem pohrdají, chybu většinou nepřiznají, a pokud ano, její odstranění trvá léta.
3. Podle specifikace musí mít každý interpret PDF k dispozici 14 základních fontů. Je to proto, aby malé dokumenty zbytečně nenarůstaly vkládáním fontů. Co si však myslet o tom, že Adobe přibaluje k různým verzím Acrobatu tyto fonty s různými metrikami?
4. Adobe preferuje v některých oblastech podivné strategie vedoucí k mizerné kvalitě dokumentu. Příkladem je záměrně špatná rasterizace Type 3 fontů na obrazovku Acrobatem.
5. Ačkoliv je Acrobat Reader distribuován pro různé platformy, na některé je k dispozici s výrazným časovým skluzem (Linux) a pro jiné se vývoj zastavil (OS/2).

Přenositelnost dokumentu není z těchto důvodů naplněna. Je pravda, že pokud se omezíme na jednoduchý hypertext bez videa a externích volání, vložíme všechny fonty do dokumentu a přizpůsobíme svůj program tak, aby negeneroval chybu v prohlížeči, i když je syntakticky správně, slušné záruky o přenositelnosti dokumentu dosáhneme. Problém nevidím ve specifikaci PDF, ale v implementaci této specifikace. Konkurence z oblasti open-source (GhostScript, xpdf) udělala velký kus práce, ale i tak je v rozsahu pokrytí specifikace za Adobe. Chybí hlavně podpora prezentačního módu, videa a java skriptu. Nechť je to výzva pro open-source programátory.

3. Tři cesty od TEXu k PDF

Existují v podstatě tři cesty, jak vytvořit PDF dokument z TEXového zdrojového kódu:

1. Generovat jej přímo ze zdrojového kódu. Umí to modifikace TEXu zvaná pdfTEX. Program vznikl na Masarykově univerzitě jako magisterská a doktorská práce Hàn Thế Thành [19, 20].
2. Standardním TEXem vytvořit DVI a zkonvertovat jej do PDF, například programem `dvipdf` [22].

3. Vytvořit z DVI PS a ten pak převést do PDF pomocí GhostScriptového skriptu `ps2pdf` nebo komerčním Adobe Distillerem.

Vkládání interaktivních prvků je v pdfTEXu umožněno zavedením nových primitiv. Zbylé dva způsoby využívají značkování TEXovým primitivem `\special{...}` a PS operátorem `/PDFMark`.

S programem `dvipdf` nemám osobní zkušenost, takže jej hodnotit nebudu. Ale je to nejméně používaná cesta. Kvalita převodu pdfTEXem a přes PS je výborná. Distiller měl oproti GhostScriptu navrch, ale v poslední době se tento rozdíl stírá. Vyplatí se mít GhostScript nejnovější verze. Distiller umožňuje optimalizovat dokument z hlediska jeho použití (osvit, tiskárna, obrazovka). Pokud by chtěl tyto zařízení rozlišit uživatel pdfTEXu, musel by ručně přizpůsobit každý obrázek (rozlišení, barevný model), probrat specifikaci PDF a nastavit patřičné parametry.

Při generování PDF je dobré se vyhnout Type 3 fontům, tj. fontům generovaných METAPOSTem. Dnes již většina METAPOST fontů existuje v dobré kvalitě i jako Type 1. Pro první a třetí způsob generování PDF se nastavení PS fontů zajistí na úrovni konfiguračních souborů (ve web2c instalaci je to implicitní nastavení `pdftex.map` uvedený v `pdftex.cfg`, respektive `config.pdf` pro DVI ovladač `dvips` spuštěný s parametrem `-Ppdf`).

Výhody pdfTEXu oproti výrobě PDF přes DVI jsou následující:

- Můžeme vkládat bitmapovou grafiku ve formátech JPEG, PNG a TIFF. Při klasickém překladu do PS jsme většinou omezeni převodem obrázků do EPS, čímž jejich velikost velmi naroste (běžně 10×) a naroste tak i velikost výsledného PS. V případě jednostránkových plakátů tak běžně jde o PS soubor veliký stovky MB. Také musíme započítat paměťovou a časovou režii při převodu obrázků.
- Můžeme používat TrueType fonty.
- Můžeme využívat mikrotypografická rozšíření, kterými se zabývá Thànhova disertační práce [20]. Jde o znaky visící z bloku textu a horizontální zvětšování/zmenšování znaků u řádků s vysokou hodnotou škaredosti (badness). Jejich cílem je kompaktnější a homogennější globální vzhled odstavce při okem nepostřehnutelných lokálních odchylkách. Obě techniky jsou implementovány do TEXového optimalizačního algoritmu zlomu odstavce. Více viz v sekci 4.1 v bodě 4.
- Lze uložit aktuální bod sazby a znovu jej nastavit. Tato funkce je užitečná pro sazbu blokových diagramů nebo podbarvení odstavců s možným stránkovým zlomem a plovoucími objekty.

V další sekci se zaměřím na přímé generování PDF pdfTEXem.

4. Program pdfTEX

Program pdfTEX má svoji vlastní elektronickou diskusní skupinu vedený v anglickém jazyce. Její adresa je `pdftex@tug.org` a její archiv lze nalézt na `http://tug.org/pipermail/pdftex/`. Přihlásit se lze na stránce `http://tug.org/mailman/listinfo/pdftex`světového Sdružení uživatelů TEXu TUG.

4.1. Nová primitiva

Jak jsme již uvedli, pdfTEX umožňuje využívat možností PDF pomocí nových primitiv. Jejich popis (bohužel neúplný) uvádí pdfTEX manual [21], úplný seznam čtenář najde v [17]. My si je zde funkčně roztřídíme bez požadavku na úplnost:

1. Hlavní přepínač `\pdfoutput`, jehož kladná hodnota přepíná výstup z DVI do PDF. Jeho nastavení má význam jen před výstupem první stránky pomocí `\shipout`. Tento registr se testuje, chceme-li automaticky rozlišit, zda je dokument zpracováván klasickým TEXem nebo pdfTEXem s výstupem do DVI či pdfTEXem s výstupem do PDF:
   ```
   \newif\ifPDF
   \ifx\pdfoutput\undefined
   \else\ifnum\pdfoutput>0 \PDFtrue\fi
   \fi
   ```
2. Rozměrové parametry (např. `\pdfpagewidth`, `\pdfhorigin`). Pozor! Pokud nastavíte `\pdfhorigin` nebo `\pdfvorigin` na nulu, pdfTEX ji změní na 1 in. Proto v takovém případě použijte např. `\pdfhorigin=1sp`).
3. Parametry dokumentu (komprese `\pdfcompresslevel`, numerická přesnost `\pdfdecimaldigits`, informační údaje `\pdfinfo`, způsob zobrazování dokumentu [15, str. 83] `\pdfcatalog`) a jednotlivých stránek [15, str. 88] (`\pdfpagesattr`, `\pdfpageattr`).
4. Mikrotypografická rozšíření.
   - Visící znaky (protrude characters). Každému znaku lze přiřadit hodnotu, o kolik má tento znak přesahovat levý (`\lpcode`) a pravý (`\rpcode`) okraj bloku textu. Syntax obou primitiv je stejná jako u primitiva `\catcode`. Hodnota 1000 znamená převis o 1 em. Visící znaky zapínáme kladnou hodnotou `\pdfprotrudechars`. Je-li jinak menší jak 2, pak se řádky nalámou standardním algoritmem TEXu a následně se doplní převis. Je-li 2 a více, optimalizuje se zlom i s aktuálními hodnotami visících zanků. Rozhoduje hodnota tohoto registru při uzavření odstavce. Příklad: chceme-li vpravo přesah tečky o 5 % em a rozdělovacího znaménka o 8 % em v aktuálním fontu, napíšeme:
     ```
     \pdfprotrudechars=2
     ```

```
\rpcode\font`\.=50
\rpcode\font\hyphenchar\font=80
```

Mezi visící znaky pečliví typografové zařazovali interpunkční znaménka (tečky, čárky, uvozovky, rozdělovací znaménka) pro jejich přílišnou světlost narušující vertikální hranu textu. Příklad použití je v tomto odstavci.

- Horizontální zvětšování/zmenšování znaků (font expansion). Někdy je obtížné zlomit odstavec tak, aby neobsahoval velké díry mezi slovy. Často se to stává při sazbě do úzkého sloupce. Horizontální zvětšování/zmenšování znaků nám přidává další stupeň volnosti pro rovnoměrné vyplnění řádky. Každému fontu přidělíme maximální hodnoty roztažení/stažení pomocí primitivu `\pdffontexpand`. Také se zde nastavuje krok, protože stažení i roztažení se děje diskrétně. Změna velikosti znaku musí však být prováděna s citem, aby čtenář nic nepoznal, i když se pod sebou sejde nejvíce protažená a smrsknutá řádka. Doporučené hodnoty jsou okolo 2 %. I zde máme možnost ovlivnit, o kolik se může jedno písmeno roztáhnout/stáhnout vůči globálním hodnotám pomocí hodnoty `\efcode`. Přepínač se jmenuje `\pdfadjustspacing` a i zde má tři polohy stejně jako u visících znaků. Ke každému takto použitému fontu musíme připravit metriky a jde-li o bitmapové (METAPOST) fonty, musíme zajistit i vygenerování těchto bitmap. Syntax názvu fontů je následující: `fontname-shrink.tfm` nebo `fontname+stretch.tfm`. Příklad:

```
\newcount\N
\loop\efcode\font\N=1000\advance\N by 1
  \ifnum\N<256 \repeat
\pdfadjustspacing=2
\pdffontexpand\font 20 20 5 1000
```

Číselné hodnoty u `\pdffontexpand` mají následující význam: roztažení, stažení, krok a měřítko.

5. Primitiva pracující s fonty (`\pdffontname`, `\pdffontobjnum`, `\pdfincludechars`).
6. Parametry bodu sazby (`\pdfsavepos`, `\pdflastxpos`, `\pdflastypos`). Není zatím dokumentováno.
7. Primitiva vkládající obrázky [24] či boxy sazby (`\pdfximage`, `\pdfrefximage`, `\pdflastximage`, `\pdfimageresolution`, `\pdfxform`, `\pdfrefxform`, `\pdflastxform`). Málo známý je registr `\pdflastximagepages`, který udává počet stran PDF dokumentu načteného pomocí `\pdfximage`.

8. Primitiva vytvářející hypertextové odkazy [25] (odkazy `\pdfstartlink`, `\pdfendlink`, `\pdflinkmargin`, doskoky `\pdfdest`, `\pdfdestmargin`, záložky `\pdfoutline` a zřetězení článku `\pdfthread`, `\pdfstartthread`, `\pdfendthread`).
9. Primitiva zařazující anotace, např. zvukové soubory a videa [26] (`\pdfannot` a `\pdflastannot`).
10. Primitiva vkládající obecné objekty včetně proudů [26] (`\pdfobj`, `\pdfrefobj`, `\pdflastobj`,) a obecné grafické instrukce (`\pdfliteral`). Příklady použití:
    - Transformace sazby.

$$[\ x' \quad y' \quad 1\ ] = [\ x \quad y \quad 1\ ] \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$$

R

```
\pdfliteral{q 0 1 -1 0 0 0 cm}%
  \smash{R}%
\pdfliteral{Q}%
```

    Aby se nám nerozjely body sazby TEXu a PDF, musíme dát obsah trasformace do boxu nulové velikosti nebo TEXovými primitivy zajistit posun sazby o tolik, o kolik se posune včetně PDF transformace. TEX sám totiž o PDF transformaci nemůže nic vědět.
    - Průhlednosti (od verze PDF 1.4.

```
\def\HR#1#2{\vrule width.5cm height#1ex depth#2ex}%
\let\ex=\expandafter \let\im=\immediate
\im\pdfobj{<< /Type /ExtGState /ca 1
              /BM /Normal /AIS false >>}%
\edef\NmTrA{\the\pdflastobj\space 0 R }%
\im\pdfobj{<< /Type /ExtGState /ca .8
              /BM /Screen >>}%
\edef\NmTrB{\the\pdflastobj\space 0 R }%
\im\pdfobj{<< /Tr0 \NmTrA /Tr1 \NmTrB >>}%
\edef\NmTr{/ExtGState \the\pdflastobj\space 0 R }%
\ex\global\ex\pdfpageresources\ex{\NmTr}%

\pdfliteral{1 .8 .8 rg /Tr1 gs}%
\HR{2}{1}\hskip-0.25cm\HR{2.5}{.5}%
\pdfliteral{/Tr0 gs 0 g}%
```

    - Změna barvy.

Gray Black
```
\pdfliteral{0.8 g}Gray
\pdfliteral{0 g}Black
```

RGB Black
```
\pdfliteral{0 0 1 rg}RGB
\pdfliteral{0 g}Black
```

CMYK Black
```
\pdfliteral{0 0 1 0 k}CMYK
\pdfliteral{0 g}Black
```

11. Podpora UNICODE. Zatím není dokumentováno.
12. Parametry programu pdfTEX (`\pdftexversion`, `\pdftexrevision`).

4.2. Makrobalíky

Velice šikovným nástrojem je balík `thumbpdf.sty` od Heiko Oberdieka, který perlovým skriptem ve spolupráci s GhostScriptem vytvoří a vloží do dokumentu *náhledy stránek*. Je napsaný v plain TEXu. Použití je nesmírně jednoduché:

1. Vlož makra do dokumentu: `\input thumbpdf.sty`
2. PřeTEXuj.
3. Spusť skript Perlu: `thumbpdf file_without_ext`
4. PřeTEXuj.

Uživatelé LATEXu mohou využít balíku Sebastiana Rahtze `hyperref` [13, 10]. Nejenomže zjednodušuje vkládání *hypertextových odkazů* a automaticky generuje odkazy ze standardních LATEXových referenčních mechanismů. Umožňuje dokonce učinit kód nezávislým na způsobech generování PDF popsaných v sekci 3. Použijeme-li parametr `pdftex`, najdeme podporu v balících `color` od Davida Carlisleho, `graphicx` od Davida Carlisleho a Sebastiana Rahtze a `geometry` pro nastavení rozměrů dokumentu od Hideo Umekiho. Dále jmenujme již zmíněné `pdftricks` a `pdfpages`.

Myslím, že oblast, elektronického publikování, jehož rozvoj teprve nastane, je vyplňování *formulářů*. I zde existuje velmi silný nástroj pro LATEX. Jmenuje se AcroTEX a jeho autorem je D. P. Story [14]. Tento balík obsahuje i podporu Javascriptu.

A snad nejsilnější podporu pro elektronické publikování a prezentace má balík ConTEXt Hanse Hagena. Je to nástroj využívající to nejlepší ze tří různých programů: sazbu pdfeTEXu, vektorovou grafiku METAPOSTu a výpočty, textové manipulace a řízení procesu Perlu. Od letošního roku je konečně k dispozici výborná dokumentace v angličtině [5, 6].

5. Prezentace pdfTEXem

V této části budou ukázány možnosti využití PDF při prezentacích včetně odkazů na nejpoužívanější prezentační makrobalíky. O ConTEXtu jsem se již zmínil. Originální varianty Hagenových prezentací přináší dokument [4]. Seznam prezentačních balíků pro LATEX uvádí prezentace [1]. Velmi oblíbený je pdfScreen od C. Radhakridhnana [11]. Umí paralelně vytvářet prezentace a textový dokument. Inkrementální přírůstky textu na stránce zvládá balík TEXPower od Stephana Lehmke [8].

6. Závěr

Formát PDF je standardem pro elektronické publikování a je vhodný i pro dataprezentace. Uživatelé připravující dokumenty TEXem si mohou vybrat několik cest, jak připravit velmi kvalitní PDF. Usnadní jim v tom dnes již silná podpora makrobalíků. Po prudkém rozvoji v této oblasti koncem devadesátých let se situace stabilizuje. Týká se to i programu pdfTEX, který přináší několik technologických vylepšení. Největší problémy dnes nenastávají na straně vytváření PDF, ale v jeho zobrazování a tisku. Neexistuje přenositelný nástroj na zpracování méně obvyklých prvků specifikace PDF.

Reference

[1] David M. Allen. LATEX presentation packages, únor 2002. `http://www.ms.uky.edu/~allen/presentations.pdf`.

[2] Angus Duggan. Psutils. Free program package.

[3] Michael C. Grant, David Carlisle. *The PSfrag System, version 3*, 1998.

[4] Hans Hagen. The ConTEXt presentation styles, 2001. `http://www.tug.org/tug2001/authors/presentations/hagen/hagenIIIslides.pdf`.

[5] Hans Hagen. *ConTEXt the manual*. `http://www.pragma-ade.com/`, listopad 2001.

[6] Hans Hagen. *METAFUN*. `http://www.pragma-ade.com/`, leden 2002.

[7] Adobe Systems Incorporated. PostScript language document structuring conventions specification. Technická zpráva 5001, září 1992.

[8] Stephan Lehmke. TEXPower package. Home page: `http://texpower.sourceforge.net`.

[9] Andreas Matthias. *The pdfpages Package*, 2002. Email: `amat@kabsi.at`.

[10] Heiko Oberdiek. *PDF information and navigation elements with hyperref, pdfTEX, and thumbpdf*, 1999. `TeXLive6:\$TEXMF/doc/latex/hyperref/slides.pdf`.

[11] C. V. Radhakrishnan. *Pdfscreen manual*, 2002. `http://ftp.cstug.cz/pub/tex/CTAN/macros/latex/contrib/supported/pdfscreen/manual-screen.pdf`.

[12] C. V. Radhakrishnan, C. V. Rajagopal, a Chambert-Loir Antoine. *Trivial experiments with PSTricks manipulation*, září 2001. `http://ftp.agh.edu.pl/pub/tex/macros/latex/contrib/supported/pdftricks/manual.pdf`.

[13] Sebastian Rahtz. *Hypertext marks in LATEX: the* `hyperref` *package*, 1998. `TeXLive6:\$TEXMF/doc/latex/hyperref/manual.pdf`.

[14] D. P. Story. The AcroTEX eDucation Bundle. Home page: `http://www.math.uakron.edu/~dpstory/webeq.html`.

[15] Adobe Systems Incorporated. *PDF reference manual, v. 1.4*, second edition, 2000. `http://partners.adobe.com/asn/developer/acrosdk/DOCS/PDFRef.pdf`.
[16] Philip Taylor. Computer typesetting or electronic publishing? New trends in scientific publication. V *Zpravodaj Československého sdružení uživatelů TEXu*, number 1–4, strany 61–89, 1995.
[17] Hàn Thế Thành. Seznam nových primitiv pdfTEXu. Soubor: `http://www.fi.muni.cz/~thanh/download/pdftex/syntax.txt`.
[18] Hàn Thế Thành. Alternativní výstup programu TEX – PDF. *Zpravodaj Československého sdružení uživatelů TEXu*, (2):69–85, 1996.
[19] Hàn Thế Thành. Přenositelný formát dokumentu a sázecí systém TEX. Diplomová práce, Masarykova univerzita v Brně, Fakulta informatiky, 1996.
[20] Hàn Thế Thành. *Micro-typographic extensions to the TEX typesetting system.* Ph.D. thesis, Masarykova univerzita v Brně, Fakulta informatiky, 2000.
[21] Hàn Thế Thành, Sebastian Rahtz, Hans Hagen. *The pdfTEX user manual*, 2001. `http://www.tug.org/applications/pdftex/pdftex-s.pdf`.
[22] Mark A. Wicks. Program dvipdf. Home page: `http://gaspra.kettering.edu/dvipdfm/`.
[23] Timothy van Zandt. *PSTricks: PostScript macros for Generic TEX*, 1993. User's guide.
[24] Vít Zýka. Používáme pdfTEX: vkládání obrázků. *Zpravodaj Československého sdružení uživatelů TEXu*, 11(4):181–186, prosinec 2001.
[25] Vít Zýka. Používáme pdfTEX II: prezentace fotografií aneb jak na hypertext. *Zpravodaj Československého sdružení uživatelů TEXu*, 12(1):13–21, březen 2002.
[26] Vít Zýka. Používáme pdfTEX III: video a zvuk v prezentaci. *Zpravodaj Československého sdružení uživatelů TEXu*, 12(2):47–55, březen 2002.

Summary: TEX and PDF

TEX is a system for quality typesetting with portability of source code within platforms and time. Portable Document Format (PDF) is format for electronic publishing. This paper shows possibilities of electronic publishing in PDF for pdfTEX users.

Vít Zýka
České vysoké učení technické, Fakulta elektrotechnická
Centrum aplikované kybernetiky
Email: `zyka@cmp.felk.cvut.cz`