Zpravodaj Československého sdružení uživatelů TeXu

Vít Zýka

Používáme pdfTeX V: aktuální pozice sazby

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 17 (2007), No. 2, 67–72

Persistent URL: http://dml.cz/dmlcz/150032

# Používáme pdfTEX V: aktuální pozice sazby

Vít Zýka

Jedním z posledních dosud v tomto seriálu nepublikovaných rozšíření, o které pdfTEX obohatil TEX, je zjištění aktuální pozice sazby. Ta může být později využita pro sazbu jiných objektů na takto zapamatovaném místě. Tento článek přináší popis souvisejících nových primitiv a ukázku jejich použití. V závěru uděláme tečku za tímto seriálem.

## Motivace

TEX při sazbě strany dokumentu postupuje sekvenčně. Klade jeden box (znak, čáru, mezeru) ke druhému a posouvá přitom aktuální bod sazby. Pokud potřebujeme klást více boxů přes sebe (např. barevné pozadí, vodoznak), musíme z aktuálního bodu box „vysunout“ do požadované polohy a zajistit, aby se nám přitom aktuální bod sazby neposunul. Tento způsob relativního umístění objektu vůči jednomu bodu sazby není omezující, ale je poněkud nešikovný.

Na makrojazykem TEXu nepřelezitelnou bariéru narazíme, pokud chceme umístit objekt v závislosti na dvou a více nezávislých bodech sazby. Příkladem je podbarvení či orámování odstavců v hlavním vertikálním seznamu tak, aby se zachovala vertikální pružnost a možnost stránkového zlomu, kreslení diagramů se šipkami mezi buňkami umístěnými v nepravidelné mřížce či šipky od jednoho slova v odstavci ke druhému. K tomu bychom potřebovali si místo sazby zapamatovat k pozdějšímu vykreslení závislého objektu. Dosud se tato potřeba řešila přes PostScriptoté operátory. pdfTEX nám však nabízí přímočařejší a universálnější řešení – pomocí nových primitiv.

## Nová primitiva

Pro práci s aktuální pozicí sazby slouží tři nová primitiva. Prvním je

```
\pdfsavepos
```

Slouží k uložení značky do zpracovávaného seznamu. Po zformátování strany, během akce `\shipout`, se tato značka zpracuje – uloží se do paměti absolutní pozice tohoto bodu sazby vzhledem k levému dolnímu rohu stránky (média). Tato $(x, y)$ pozice se pak dá přečíst primitivy

```
\pdflastxpos
\pdflastypos
```

Každý z nich vrací celočíselnou hodnotu reprezentující vzdálenost v jednotce `sp` (přesnost TEXu).

Protože `\pdfsavepos` muselo být zpracováno až při povelu `\shipout`, je třeba získané hodnoty uložit do souboru pomocí `\write` a načíst a použít je až při dalším průchodu TEXu.

Protože použití není zcela přímočaré, ukážeme si jej na příkladě. Spojíme v něm čarou dvě označená slova v odstavci.

## Příklad: čára mezi slovy v odstavci

Naším cílem bude *nakreslit* čáru z místa A do místa B. Podmínkou bude, aby obě místa ležela na téže straně, i když třeba uvnitř odstavce. Každé místo označíme pomocí značky `\posMark{A}`, kde A je zvolené jméno této značky. Dvě místa tímto způsobem definovaná a pojmenovaná pak využijeme k nakreslení *čáry* mezi nimi, jak ukazuje tento odstavec. Ukažme si část jeho zdrojového kódu:

```
Naším cílem bude {\it nakreslit\/}\posMark{A} čáru z~místa A ...
...
k~nakreslení \posMark{B}{\it čáry\/} mezi nimi, jak ukazuje ...
```

Nejprve si připravíme pomocná makra.[1] Pro vykreslení čáry z jednoho bodu do druhého využijeme PDF či PostScriptové operátory. Čára bude šedá a široká 2 pt. Pomocí makra `\ifpdf` ze stejnojmenného balíku zaručíme, aby makro fungovalo jak při výstupu do PDF, tak i do DVI:

```
\ifpdf
  \def\Line#1,#2--#3,#4{% 1,2=start x y; 3,4=stop x y <jednotka bp>
    \pdfliteral page {0.7 G 2 w #1 #2 m #3 #4 l S }}
\else
  \def\Line#1,#2--#3,#4{%
    \special{" 0.7 setgray 2 setlinewidth #1 #2 moveto
               #3 #4 lineto stroke }}
\fi
```

Dále budeme potřebovat makro, které odřízne jednotku *pt* z výpisu registru typu dimen:

```
{\catcode`\p=12 \catcode`\t=12 \gdef\removePT#1pt{#1}}
```

A poslední přípravnou akcí bude převod čísla v jednotkách *sp* do jednotek *bp*, ve kterých se vyjadřují vzdálenosti v PDF a PostScriptu:

[1] Pro povely, které přímo nesouvisí s tematikou tohoto článku, využijeme LATEXové balíky či makra `eso-pic`, `ifpdf`, `afterpage` a `\InputIfFileExist`, aby článek zůstal přehledný.

```
\def\defBPfromSP#1#2{% 1=identifikator 2=cislo <sp>
  \bgroup
  \dimen0=#2sp
  \dimen0=.013837\dimen0
  \dimen0=72\dimen0
  \expandafter\xdef\csname#1\endcsname{\expandafter\removePT\the\dimen0 }%
  \egroup}
```

Nyní přistupme k hlavnímu tématu – uložení pozice sazby. Napsali jsme již, že pozice je známa až v okamžiku provedení `\shipout`, a že si ji tedy musíme uložit do pomocného souboru. Nazveme jej podle hlavního souboru, ale s koncovkou `.pos`. Následující makra otevřou na začátku dokumentu tento soubor pro zápis a na jeho konci jej zavřou:

```
\newwrite\posHandle   \def\posFile{\jobname.pos }

\def\posOpen{\openout\posHandle=\posFile}
\def\posClose{\closeout\posHandle}

\AtBeginDocument{\posLoad\posOpen}
\AtEndDocument{\posClose}
```

Zápis pozice sazby do souboru obstará uživatelské makro `\posMark{jmeno}`. V něm použijeme všechna tři nová primitiva:

```
\def\posMark#1{% 1=jmeno
  \pdfsavepos
  \write\posHandle{%
    \string\posDef\string{#1\string}%
    \string{\the\pdflastxpos\string}\string{\the\pdflastypos\string}}}
```

Po prvním zpracování dokumentu se nám vytvoří soubor s obsahem:

```
\posDef{A}{9507532}{10913469}
\posDef{B}{23416299}{8599377}
```

Soubor načteme makrem `\posLoad` umístěným na řádku 25 před otevřením `.pos` souboru makrem `\posOpen`:

```
\def\posLoad{\InputIfFileExists{\posFile}{}{}}
```

Soubor `.pos` čteme jen tehdy, když soubor existuje, což bude až ve druhém a dalším průchodu TEXem. Aby načtení neskončilo chybou, musíme nadefinovat

následující makro `\posDef{jmeno}{x-pos}{y-pos}`. Jeho úkolem bude vytvořit dvě makra `\pos-x-sp-jmeno` a `\pos-y-sp-jmeno` obsahující náležité číselné hodnot v jednotce *sp* a dvě `\pos-x-bp-jmeno`, `\pos-y-bp-jmeno` v jednotce *bp*:

```
\def\posDef#1#2#3{% 1=jmeno 2=x-pos 3=y-pos
  \expandafter\def\csname pos-x-sp-#1\endcsname{#2}%
  \posDefXbp{#1}%
  \expandafter\def\csname pos-y-sp-#1\endcsname{#3}%
  \posDefYbp{#1}}
```

Převod mezi jednotkami obstaralo makro

```
\def\posDefXbp#1{\defBPfromSP{pos-x-bp-#1}{\posGetX{#1}}} % 1=jmeno
\def\posDefYbp#1{\defBPfromSP{pos-y-bp-#1}{\posGetY{#1}}} % 1=jmeno
```

Volání takto automaticky vytvořených maker si zjednodušíme definicí

```
\def\posGetX#1{% 1=jmeno
  \expandafter\ifx\csname pos-x-sp-#1\endcsname\relax0
  \else\csname pos-x-sp-#1\endcsname\fi\relax}
\def\posGetY#1{% 1=jmeno
  \expandafter\ifx\csname pos-y-sp-#1\endcsname\relax0
  \else\csname pos-y-sp-#1\endcsname\fi\relax}
\def\posGetXbp#1{% 1=jmeno
  \expandafter\ifx\csname pos-x-bp-#1\endcsname\relax0
  \else\csname pos-x-bp-#1\endcsname\fi\relax}
\def\posGetYbp#1{% 1=jmeno
  \expandafter\ifx\csname pos-y-bp-#1\endcsname\relax0
  \else\csname pos-y-bp-#1\endcsname\fi\relax}
```

Takto získané hodnoty použijeme pro vykreslení čáry v absolutních souřadnicích. Vhodné místo je při akci `\shipout`, když je nastaven základní souřadnicový systém. Využijeme k tomu balíku `eso-pic`:

```
\def\AbsLine#1,#2--#3,#4{% 1,2=start x y; 3,4=stop x y <jednotka bp>
  \AddToShipoutPicture{\AtPageLowerLeft{\Line#1,#2--#3,#4}}}
```

A konečně, zde je finální uživatelské makro používající pojmenované body:

```
\def\AbsLineFromTwoMarks#1#2{% 1=jmeno-A 2=jmeno-B
  \AbsLine\posGetXbp{#1},\posGetYbp{#1}--\posGetXbp{#2},\posGetYbp{#2}}
```

Takže po našem úvodním odstavci jsme museli uvést ještě tyto dvě řádky, aby se nám vykreslila požadovaná čára:

```
\AbsLineFromTwoMarks{A}{B}
\afterpage{\ClearShipoutPicture}
```

Druhý řádek zařídí, aby se nám tato čára netiskla na každé další straně.

## Závěr

Tento díl seriálu *Používáme pdfTEX* je pravděpodobně poslední. Hlavní témata, o která pdfTEX rozvíjí klasický TEX, byla popsána a samotný vývoj se přesunul k luaTEXu [1].

V této souvislosti je dobré připomenout, že pdfTEX vznikl jako diplomová a později disertační práce vietnamského studenta Hàn Thế Thành na Masarykově universitě v Brně pod vedením Jiřího Zlatušky a Petra Sojky. Později se vývoj přenesl na Martina Schrödera, Hartmuta Henkela, Taco Hoekwatera a Hanse Hagena. Hàn Thế Thành zůstal hlavním koordinátorem.

pdfTEX nepřinesl jen nový výstupní formát – PDF – k původnímu DVI, ale též mnohá rozšíření. Jejich použití usnadnují makrobalíky. Pro LATEX jmenujme několik nejdůležitějších:

- microtype (h-z algoritmus, prostrkání okrajů sazby, ligatury),
- hyperref (hypertext a formuláře),
- graphicx (PDF, JPG, PNG obrázky, ořez, zvětšení, rotace),
- color (barvy),
- pdfpages (slučování dokumentů, vyřazování),
- movie15 (vkládání videí),
- animate (vytváření PDF animací),
- acrotex (formuláře, interaktivní dotazníky, JavaScript).

V ConTEXtu je vše integrováno do jeho jádra, žádné přídavné balíky nejsou potřeba. Uživatelé plainTEXu musí studovat referenční manuálu PDF [2] a využívat přímo nová primitiva popsaná v manuálu pdfTEXu [3]. Sepsal jej Hàn Thế Thành spolu se Sebastianem Rahtzem. Grafickou úpravu zpracoval Hans Hagen. Dnes manuál spravuje Pawel Jackowski.

Úspěch pdfTEXu spočíval na 100% zpětné kompatibilitě s TEXem, používání mainstreamových formátů pro vstup i výstup, kvalitativních přínosech typografii a testování při praktickém používání. Jeho nástupce bude k úspěchu pravděpodobně potřebovat splnit stejná kritéria. Těžko říci, zda luaTEX, ke kterému přesunula své úsilí značná část vývojového týmu pdfTEXu, bude stejně úspěšný. Potenciál na to má. Držme vývojářům palce, podpořme je finančně či přiložme ruku k dílu. Je to šance, aby TEX obstál i v dalším desetiletí.

## Reference

[1] Taco Hoekwater and Hans Hagen. *LuaTEX reference manual*, 2007. http://context.aanhet.net/luatex/snapshot/manual/luatexref-t.pdf.
[2] Adobe systems Incorporated. *PDF reference manual, v. 1.7*, 6 edition, 2006. http://www.adobe.com/devnet/acrobat/.
[3] Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, and Hartmut Henkel. *The pdfTEX user manual*, 2007. http://sarovar.org/docman/view.php/106/66/pdftex-s.pdf.

## Summary: Using pdfTEX V: current typesetting position

One of the extension by which pdfTEX improved TEX is a possibility to obtain a current typesetting possition. It can be used for placing objects later. This article brings a description of related primitives and examples of their usage.

*Vít Zýka* vit.zyka@seznam.cz

# Vkládání JavaScriptů pdfTEXem prakticky

Robert Mařík

## Úvod

V poslední době se na CTAN objevuje stále více balíčků, využívajících vkládání JavaScriptů do PDF dokumentů (animate, animfig, cooltooltips, dps, eCards, exerquiz, fancytooltips, jeopardy, jj_game, pdfanim)

JavaScripty umožňují získat dokumenty, které mají větší možnosti než klasické statické dokumenty, jejich funkčnost se však projeví pouze v prohlížeči Adobe Reader nebo Adobe Acrobat (což je poněkud omezující pro uživatele alternativních programů na čtení PDF).

Cílem tohoto článku je ukázat několik jednoduchých příkladů na vkládání JavaScriptů, které by po příslušném rozšíření umožnily dosáhnout podobných možností, jaké mají výše uvedené balíčky. Budeme se přitom vždy snažit udělat příklady názorné a co nejjednodušší. Článek je doprovázen souborem `testjs.tex`, který obsahuje všechny popisované JavaScripty a některé doplňující komentáře a souborem `testjs.pdf`, který vznikl překladem pdfLATEXem. Tyto soubory jsou