Zpravodaj Československého sdružení uživatelů TeXu

Lucie Schaynová; Jan Šustek

Aplikace parametrů řádkového zlomu a output rutiny k formátování sazby v TeXu

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 29 (2019), No. 1-4, 44–65

Persistent URL: http://dml.cz/dmlcz/150169

Aplikace parametrů řádkového zlomu a output rutiny k formátování sazby v TEXu

Lucie Schaynová, Jan Šustek

V článku projdeme vnitřnosti programu TEX a ukážeme si, jakou cestou se jednotlivé znaky vstupního souboru .tex postupně dostanou až do výstupního souboru .pdf. Zdržíme se u algoritmu řádkového zlomu, který bez debat výrazně předběhl svou dobu. Vhodnou kombinací jeho parametrů lze nastavit nejen zarovnání textu do bloku, na střed nebo na praporek, ale ukážeme si i mnoho dalších možných způsobů zarovnání textu. Na konci této cesty se nachází output rutina, která má za úkol umístit vysázený text na stránku. Ukážeme si, jak lze nastavit různá záhlaví a zápatí a jak lze jednoduše vytvořit hlavičkový papír. Také si ukážeme různé praktické aplikace output rutiny, například k vysázení slajdů pro přípravu prezentací. Přijde řeč i na problematiku zjišťování pozice konkrétního bodu sazby na stránce a využití této informace při kreslení obrázků v METAPOSTu.

Článek vychází z přednášky druhého autora na konferenci OSSconf 2018.

Klíčová slova: TEX, řádkový zlom, output rutina

1 Úvod

TEX je program, který vezme text ze vstupního souboru a vysází jej podle daných instrukcí. V tomto článku si stručně popíšeme, jakou cestou projdou jednotlivé znaky vstupního souboru .tex, než se vytvoří výstupní soubor .pdf. Tuto cestu si lze představit jako montážní linku s desítkou na sebe navazujících nástrojů, kde vstupem prvního nástroje je vstupní soubor, vstupem dalších nástrojů je výstup předchozího nástroje a výstupem posledního nástroje je výsledný soubor .pdf. Z těchto nástrojů se dále v článku budeme podrobněji zabývat řádkovým zlomem a output rutinou.

Každý nástroj začne pracovat okamžitě, když od předchozího nástroje dostane dostatek materiálu, aby mohl začít pracovat (princip nutného minima). Zájemcům o hlubší pochopení problematiky doporučuji [1] nebo [2], zájemcům o maximální pochopení doporučuji během dlouhých zimních večerů [3].

V textu budeme pracovat v PlainTEXu s načteným OPmacem, a to z důvodu, aby lépe vynikly jednotlivé ukázky. Jelikož je ale většina uvedených příkazů obsažena přímo v jádru (pdf)TEXu, budou ukázky po drobných modifikacích fungovat i v jiných nadstavbách (pdf)TEXu.

 doi: 10.5300/2019-1-4/44

1.1 Input procesor

Funkcí input procesoru je načíst jednotlivé řádky textu z konkrétního vstupního souboru, případně z jiného vstupního proudu.[1] Je třeba si uvědomit, že možnosti názvů souborů mohou být v různých operačních systémech různé. Dále mohou být různé i způsoby ukončování řádků v textových souborech. Chování input procesoru tak nutně je systémově závislé. Na druhou stranu je input procesor jediná systémově závislá část TEXu a výstup input procesoru již je systémově nezávislý.

1.2 Token procesor

Token procesor přečte znaky z výstupu input procesoru a vytvoří z nich tzv. tokeny a ty vloží do tzv. čtecí fronty. Token typu řídicí sekvence odpovídá názvu makra, názvu příkazu, názvu registru apod. Pokud se ze znaků nevytvoří token typu řídicí sekvence, vytvoří se token typu dvojice (*znak*, *kategorie*), kde *kategorie* určuje, jak se daný *znak* bude chovat. Zatímco u tokenů typu řídicí sekvence není dopředu známo, jak se budou chovat, a uživatel si je může předefinovat, u tokenů typu dvojice je jeho chování pevně dáno kategorií, která je dále neměnná.

Token procesor tedy identifikuje vše, co se v textu nachází, a tyto informace pošle dalšímu nástroji, expand procesoru.

1.3 Expand procesor

Expand procesor načte jeden token ze čtecí fronty. Pokud se jedná o expandovatelný token (například makro), načte i jeho případné argumenty a token expanduje podle příslušné definice. Vzniklé tokeny vrátí zpět do čtecí fronty. Tento proces se opakuje tak dlouho, dokud expand procesor nenarazí na neexpandovatelný token (například na primitivní příkaz, písmeno nebo konstruktor exponentu). Tento token pak předá hlavnímu procesoru. Na výstupu expand procesoru tak mohou být pouze primitivní příkazy nebo znaky s daným chováním.

1.4 Hlavní procesor

Hlavní procesor příslušné příkazy vykoná. V běžném dokumentu je nejčastějším příkazem vysázení konkrétního znaku. Dalšími častými příkazy jsou definice nových maker, změna fontu nebo práce s registry. V neposlední řadě je třeba zmínit příkaz na ukončení běhu TEXu, který bývá často zmiňován až na posledním řádku.

Znaky se ukládají do tzv. horizontálního seznamu, což si lze představit jako jeden dlouhý řádek textu, který později bude rozlámán na řádky. Do horizontálního seznamu se dále ukládají různé výplňky (horizontální mezery), penalty a další objekty.

[1] Vstup lze načítat z terminálu. Nicméně například v Linuxu lze načítat vstup i z pojmenované roury, protože ta se navenek tváří jako soubor.

1.5 Řádkový zlom

Cílem řádkového zlomu je načíst z horizontálního seznamu celý odstavec textu a ten rozlámat na jednotlivé řádky. Algoritmus najednou najde místa zlomu celého odstavce, aby byl optimální s ohledem na hodnoty mnoha parametrů.

Výsledné řádky textu se ukládají do tzv. vertikálního seznamu, což si lze představit jako jeden vysoký sloupec sazby, který bude později rozlámán na stránky. Do vertikálního seznamu se dále ukládají různé výplňky (vertikální mezery), penalty a další objekty.

1.6 Stránkový zlom

Jakmile je vertikální seznam již dostatečně plný, zavolá se algoritmus stránkového zlomu, který určí, jaká část textu se umístí na aktuální stránce. Protože nalezení optimálního stránkového zlomu vyžaduje obrovskou výpočetní složitost, není TEXový algoritmus natolik propracovaný a je jen pár parametrů, kterými lze stránkový zlom ovlivnit.

1.7 Output rutina

Output rutina se nachází na konci cesty k výstupnímu souboru .pdf. Jejím úkolem je umístění příslušného textu na stránku. Na stránku se umístí ale také například záhlaví, zápatí, poznámky pod čarou nebo plovoucí objekty.

Důležitým úkolem output rutiny je export takto vytvořené stránky do souboru .pdf. Protože až v tuto dobu je přesně známo, na které stránce bude konkrétní text, uloží se nyní také informace o křížových odkazech.

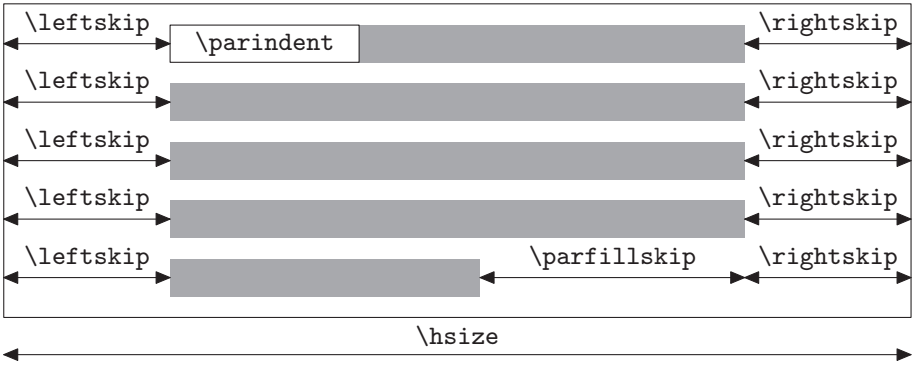
2 Řádkový zlom

Knuthův-Plassův algoritmus řádkového zlomu výrazně předběhl svou dobu. Jeho detaily, matematické odvození a množství příkladů lze najít v šedesátistránkovém článku [4].[2] Algoritmus je závislý na mnoha parametrech:

```
\adjdemerits \doublehyphendemerits \emergencystretch
  \exhyphenpenalty \finalhyphendemerits \fontdimen
\hangafter \hangindent \hsize \hyphenpenalty \leftskip
   \linepenalty \looseness \parfillskip \parindent
    \parshape \pretolerance \rightskip \tolerance
```

V dalším textu si ukážeme vliv některých parametrů na výsledný zlom. Přesný význam každého z nich se lze dočíst v [1] nebo [2].

[2] Nelze opomenout související Liangův algoritmus dělení slov [5], který se při lámání řádků v TEXu využívá.



Obrázek 1: Základní parametry řádkového zlomu

Zarovnání odstavce je určeno registry `\leftskip` a `\rightskip`, mezery těchto velikostí se vkládají na příslušný okraj každého řádku. Jako odstavcovou zarážku TEX vloží `\hbox` šířky `\parindent`. Na konec posledního řádku odstavce se vloží mezera velikosti `\parfillskip`. Situaci zachycuje obrázek 1. Jednotlivá zarovnání budeme v rámečcích na stranách 48–53 demonstrovat na textu z [6].

3 Příkaz `\shipout`

3.1 Minimální příklad

Primitivní příkaz `\shipout` do souboru .pdf okamžitě vloží stránku s vysázeným boxem, který za příkazem `\shipout` následuje. Příkaz `\shipout` neprovede nic dalšího – nevloží záhlaví, nezmění číslo strany a podobně.

Vyzkoušejme si vysázet jednoduchý dokument.

```
1 graf%
2 \shipout\hbox{ahoj}%
3 ika
4 \bye
```

Na řádku 1 hlavní procesor začne plnit horizontální seznam. Na řádku 2 se do souboru .pdf vloží stránka s textem „ahoj“. Na řádku 3 se pokračuje v plnění horizontálního seznamu. Příkaz `\bye` na řádku 4 ukončí horizontální seznam, vyvolá algoritmus řádkového zlomu, algoritmus stránkového zlomu a na závěr pomocí output rutiny do souboru .pdf vloží stránku s textem „grafika“. (Proč se nevysází text „grafika“, se čtenář dočte například v [2] na straně 103.) Výsledný soubor .pdf tak bude mít dvě strany, přičemž stránka ručně vložená pomocí příkazu `\shipout` předchází stránce, na které byl příkaz `\shipout` použit.

`\leftskip=0pt`
`\rightskip=0pt`
`\parindent=20pt`
`\parfillskip=0pt plus 1fil`
	V plainu jsou implicitně registry nastaveny na uvedené hodnoty. Při tomto nastavení je text zarovnaný do bloku, přičemž poslední řádek může být kratší.

Sto roků v šachtě žil, mlčel jsem, sto roků kopal jsem uhlí, za sto let v rameni bezmasém svaly mi v železo ztuhly. Uhelný prach sedl do očí, rubíny ze rtů mi uhly, ze vlasů, z vousů a z obočí visí mi rampouchy uhlí. Chléb s uhlím beru si do práce, z roboty jdu na robotu, při Dunaji strmí paláce z krve mé a z mého potu. Sto roků v kopalně mlčel jsem, kdo mi těch sto roků vrátí? Když jsem pohrozil kladivem, kdekdo se začal mi smáti. Abych měl rozum, šel v kopalnu zas, pro pány robil jak prve: máchl jsem kladivem – teklo vráz na Polské Ostravě krve! Všichni vy na Slezské, všichni vy, dím, nech je vám Petr neb Pavel, mějž prs kryt krunýřem ocelovým, tisícům k útoku zavel, všichni vy na Slezské, všichni vy, dím, hlubokých páni vy dolů: přijde den, z dolů jde plamen a dým, přijde den, zúčtujem spolu!

`\leftskip=0pt`
`\rightskip=0pt plus 1fil`
`\parindent=20pt`
`\parfillskip=0pt`
	Zarovnání na praporek docílíme přidáním nekonečné pružnosti k registru `\rightskip`. Nekonečná pružnost způsobí, že TEX preferuje vytvoření kratšího řádku před případným rozdělením slova. Registr `\parfillskip` můžeme vynulovat.

Sto roků v šachtě žil, mlčel jsem, sto roků kopal jsem uhlí, za sto let v rameni bezmasém svaly mi v železo ztuhly. Uhelný prach sedl do očí, rubíny ze rtů mi uhly, ze vlasů, z vousů a z obočí visí mi rampouchy uhlí. Chléb s uhlím beru si do práce, z roboty jdu na robotu, při Dunaji strmí paláce z krve mé a z mého potu. Sto roků v kopalně mlčel jsem, kdo mi těch sto roků vrátí? Když jsem pohrozil kladivem, kdekdo se začal mi smáti. Abych měl rozum, šel v kopalnu zas, pro pány robil jak prve: máchl jsem kladivem – teklo vráz na Polské Ostravě krve! Všichni vy na Slezské, všichni vy, dím, nech je vám Petr neb Pavel, mějž prs kryt krunýřem ocelovým, tisícům k útoku zavel, všichni vy na Slezské, všichni vy, dím, hlubokých páni vy dolů: přijde den, z dolů jde plamen a dým, přijde den, zúčtujem spolu!

3.2 Vložení strany s obrázkem

S využitím příkazu `\shipout` lze do dokumentu jednoduše vkládat strany jiného dokumentu. Níže použité makro `\margins` uvnitř skupiny lokálně nastaví nulové okraje, aby vložená strana byla přes celou stranu dokumentu.

```
\begingroup
\margins/1 a4 (0,0,0,0)mm
\pdfximage{zadaniVSKP.pdf}
\shipout\hbox{\pdfrefximage\pdflastximage}
\endgroup
```

`\leftskip=0pt plus 1fil`
`\rightskip=0pt plus 1fil`
`\parindent=0pt`
`\parfillskip=0pt`
	Pokud stejnou nekonečnou pružnost přidáme i k `\leftskip`, bude odstavec zarovnaný na střed. Nesmíme ale zapomenout vynulovat `\parindent` a `\parfillskip`.

Sto roků v šachtě žil, mlčel jsem, sto roků kopal jsem uhlí, za sto let v rameni bezmasém svaly mi v železo ztuhly. Uhelný prach sedl do očí, rubíny ze rtů mi uhly, ze vlasů, z vousů a z obočí visí mi rampouchy uhlí. Chléb s uhlím beru si do práce, z roboty jdu na robotu, při Dunaji strmí paláce z krve mé a z mého potu. Sto roků v kopalně mlčel jsem, kdo mi těch sto roků vrátí? Když jsem pohrozil kladivem, kdekdo se začal mi smáti. Abych měl rozum, šel v kopalnu zas, pro pány robil jak prve: máchl jsem kladivem – teklo vráz na Polské Ostravě krve! Všichni vy na Slezské, všichni vy, dím, nech je vám Petr neb Pavel, mějž prs kryt krunýřem ocelovým, tisícům k útoku zavel, všichni vy na Slezské, všichni vy, dím, hlubokých páni vy dolů: přijde den, z dolů jde plamen a dým, přijde den, zúčtujem spolu!

`\leftskip=0pt plus 1fil`
`\rightskip=0pt plus -1fil`
`\parindent=0pt`
`\parfillskip=0pt plus 2fil`
	Zarovnání do bloku s posledním řádkem na střed lze použít při sazbě popisků obrázků. Hodnoty registrů jsou voleny tak, aby jejich pružnost byla nekonečná a aby platilo `\leftskip` + `\rightskip` = `0pt` a zároveň `\leftskip` = `\rightskip`+`\parfillskip`.

Sto roků v šachtě žil, mlčel jsem, sto roků kopal jsem uhlí, za sto let v rameni bezmasém svaly mi v železo ztuhly. Uhelný prach sedl do očí, rubíny ze rtů mi uhly, ze vlasů, z vousů a z obočí visí mi rampouchy uhlí. Chléb s uhlím beru si do práce, z roboty jdu na robotu, při Dunaji strmí paláce z krve mé a z mého potu. Sto roků v kopalně mlčel jsem, kdo mi těch sto roků vrátí? Když jsem pohrozil kladivem, kdekdo se začal mi smáti. Abych měl rozum, šel v kopalnu zas, pro pány robil jak prve: máchl jsem kladivem – teklo vráz na Polské Ostravě krve! Všichni vy na Slezské, všichni vy, dím, nech je vám Petr neb Pavel, mějž prs kryt krunýřem ocelovým, tisícům k útoku zavel, všichni vy na Slezské, všichni vy, dím, hlubokých páni vy dolů: přijde den, z dolů jde plamen a dým, přijde den, zúčtujem spolu!

Strana vložená do souboru .pdf příkazem `\shipout` má rozměry podle nastavení platného v době zavolání příkazu `\shipout`. Příslušný box se umístí s ohledem na nastavený levý a horní okraj strany.

Primitivní příkazy `\pdfximage`, `\pdfrefximage` a `\pdflastximage` ke vkládání obrázků jsou popsány například v [7] v sekci 11.7.

3.3 Vložení speciální strany

Praktické využití může mít vložení velké tabulky samostatně na další stranu dokumentu. V ukázce se tabulka vysází na stránku na šířku. Nesmíme zapomenout na řádku 20 zvýšit číslo strany.

`\leftskip=0pt`
`\rightskip=0pt plus 60pt`
`\parindent=0pt`
`\parfillskip=0pt plus 1fil`
	Pokud má registr `\rightskip` konečnou pružnost, pak budou pružit i mezislovní mezery. TEX bude preferovat delší řádky i za cenu případného rozdělení slov.

Sto roků v šachtě žil, mlčel jsem, sto roků kopal jsem uhlí, za sto let v rameni bezmasém svaly mi v železo ztuhly. Uhelný prach sedl do očí, rubíny ze rtů mi uhly, ze vlasů, z vousů a z obočí visí mi rampouchy uhlí. Chléb s uhlím beru si do práce, z roboty jdu na robotu, při Dunaji strmí paláce z krve mé a z mého potu. Sto roků v kopalně mlčel jsem, kdo mi těch sto roků vrátí? Když jsem pohrozil kladivem, kdekdo se začal mi smáti. Abych měl rozum, šel v kopalnu zas, pro pány robil jak prve: máchl jsem kladivem – teklo vráz na Polské Ostravě krve! Všichni vy na Slezské, všichni vy, dím, nech je vám Petr neb Pavel, mějž prs kryt krunýřem ocelovým, tisícům k útoku zavel, všichni vy na Slezské, všichni vy, dím, hlubokých páni vy dolů: přijde den, z dolů jde plamen a dým, přijde den, zúčtujem spolu!

`\leftskip=0pt`
`\rightskip=0pt`
`\parindent=0pt`
`\parfillskip=0pt`
	Při nulovém nastavení všech registrů bude odstavec zarovnaný do obdélníku. V tomto případě je ale pravděpodobné, že vznikne řádek s extrémně širokými mezislovními mezerami.

Sto roků v šachtě žil, mlčel jsem, sto roků kopal jsem uhlí, za sto let v rameni bezmasém svaly mi v železo ztuhly. Uhelný prach sedl do očí, rubíny ze rtů mi uhly, ze vlasů, z vousů a z obočí visí mi rampouchy uhlí. Chléb s uhlím beru si do práce, z roboty jdu na robotu, při Dunaji strmí paláce z krve mé a z mého potu. Sto roků v kopalně mlčel jsem, kdo mi těch sto roků vrátí? Když jsem pohrozil kladivem, kdekdo se začal mi smáti. Abych měl rozum, šel v kopalnu zas, pro pány robil jak prve: máchl jsem kladivem – teklo vráz na Polské Ostravě krve! Všichni vy na Slezské, všichni vy, dím, nech je vám Petr neb Pavel, mějž prs kryt krunýřem ocelovým, tisícům k útoku zavel, všichni vy na Slezské, všichni vy, dím, hlubokých páni vy dolů: přijde den, z dolů jde plamen a dým, přijde den, zúčtujem spolu!

```
\begingroup
\margins/1 a4l (2,2,2,2)cm
\shipout\vbox{
  \centerline{%
    \table{rcl}
      {ob & rov & ská \cr
       ta & bul & ka \cr}
    }\medskip
  \caption/t Popisek
  \par}
\global\advance\pageno1
\endgroup
```

`\hsize=96mm`
`\leftskip=0pt`
`\rightskip=0pt`
`\parindent=0pt`
`\parfillskip=0pt`
	Pokud potřebujeme odstavec vysázet do obdélníku, ale nezáleží nám na jeho šířce, můžeme se pokusit vyhnout extrémně širokým mezerám změnou šířky sazby `\hsize`.

Sto roků v šachtě žil, mlčel jsem, sto roků kopal jsem uhlí, za sto let v rameni bezmasém svaly mi v železo ztuhly. Uhelný prach sedl do očí, rubíny ze rtů mi uhly, ze vlasů, z vousů a z obočí visí mi rampouchy uhlí. Chléb s uhlím beru si do práce, z roboty jdu na robotu, při Dunaji strmí paláce z krve mé a z mého potu. Sto roků v kopalně mlčel jsem, kdo mi těch sto roků vrátí? Když jsem pohrozil kladivem, kdekdo se začal mi smáti. Abych měl rozum, šel v kopalnu zas, pro pány robil jak prve: máchl jsem kladivem – teklo vráz na Polské Ostravě krve! Všichni vy na Slezské, všichni vy, dím, nech je vám Petr neb Pavel, mějž prs kryt krunýřem ocelovým, tisícům k útoku zavel, všichni vy na Slezské, všichni vy, dím, hlubokých páni vy dolů: přijde den, z dolů jde plamen a dým, přijde den, zúčtujem spolu!

`\leftskip=0pt`
`\rightskip=0pt`
`\parindent=0pt`
`\parfillskip=0pt`
`\emergencystretch100pt`
	Pokud potřebujeme odstavec vysázet do obdélníku a šířka je daná, pak si musíme pohrát s pružností mezislovních mezer. Nastavení registru `\emergencystretch` na kladnou hodnotu způsobí, že TEX nebude vnímat velké mezery jako příliš nevhodné. Bude se tak preferovat řádkový zlom s více řádky s velkými mezerami před zlomem s jedním řádkem s obrovskou mezerou.

Sto roků v šachtě žil, mlčel jsem, sto roků kopal jsem uhlí, za sto let v rameni bezmasém svaly mi v železo ztuhly. Uhelný prach sedl do očí, rubíny ze rtů mi uhly, ze vlasů, z vousů a z obočí visí mi rampouchy uhlí. Chléb s uhlím beru si do práce, z roboty jdu na robotu, při Dunaji strmí paláce z krve mé a z mého potu. Sto roků v kopalně mlčel jsem, kdo mi těch sto roků vrátí? Když jsem pohrozil kladivem, kdekdo se začal mi smáti. Abych měl rozum, šel v kopalnu zas, pro pány robil jak prve: máchl jsem kladivem – teklo vráz na Polské Ostravě krve! Všichni vy na Slezské, všichni vy, dím, nech je vám Petr neb Pavel, mějž prs kryt krunýřem ocelovým, tisícům k útoku zavel, všichni vy na Slezské, všichni vy, dím, hlubokých páni vy dolů: přijde den, z dolů jde plamen a dým, přijde den, zúčtujem spolu!

3.4 Řízený tisk obrázků

Předpokládejme, že máme stovku obrázků uloženou v souborech obrazek1.jpg až obrazek100.jpg. Úkolem bude vytisknout všechny tyto obrázky ve stejné velikosti, v našem případě na stránku A4 na šířku s 5mm okraji.

Zkušenosti autorů ukazují, že u ovladačů tiskáren je možné nastavovat mnoho

`\leftskip=0pt`
`\rightskip=0pt`
`\parindent=0pt`
`\parfillskip=0pt`
`\tolerance500`
	Nastavení velké hodnoty `\tolerance` způsobí, že TEX bude zkoumat i možné řádkové zlomy s většími mezerami a z těchto zlomů najde ten nejlepší.

Sto roků v šachtě žil, mlčel jsem, sto roků kopal jsem uhlí, za sto let v rameni bezmasém svaly mi v železo ztuhly. Uhelný prach sedl do očí, rubíny ze rtů mi uhly, ze vlasů, z vousů a z obočí visí mi rampouchy uhlí. Chléb s uhlím beru si do práce, z roboty jdu na robotu, při Dunaji strmí paláce z krve mé a z mého potu. Sto roků v kopalně mlčel jsem, kdo mi těch sto roků vrátí? Když jsem pohrozil kladivem, kdekdo se začal mi smáti. Abych měl rozum, šel v kopalnu zas, pro pány robil jak prve: máchl jsem kladivem – teklo vráz na Polské Ostravě krve! Všichni vy na Slezské, všichni vy, dím, nech je vám Petr neb Pavel, mějž prs kryt krunýřem ocelovým, tisícům k útoku zavel, všichni vy na Slezské, všichni vy, dím, hlubokých páni vy dolů: přijde den, z dolů jde plamen a dým, přijde den, zúčtujem spolu!

`\leftskip=0pt plus 20pt`
`\rightskip=0pt plus 20pt`
`\parindent=0pt`
`\parfillskip=0pt`
	Při použití ozdobného písma není nutné, aby okraje sazby byly zcela rovné. Pak můžeme zarovnání do obdélníku docílit nastavením malé pružnosti u `\leftskip` a `\rightskip`.

Sto roků trisekci zkoušel jsem, sto roků sekal jsem úhly. Za sto let Galois přišel sem, nálady dobré mi uhly. Rovnice padly mi do očí, kořeny z tělesa uhly. Cardano vzorcem svým útočí, najde ty rozseklé úhly. Kružítko beru si do práce, pravítko beru si taky. Eukleidés radí mi nevzdát se, konstrukcí dělal už mraky. Za sto let kružítka mám už dost, kdy se mi snahy ty vrátí? Od Knutha kamarád Metapost námahu výrazně zkrátí. Abych měl rozum, já sekal jsem zas, novinky zkusil však prve. Zadeha matika úhly vráz namísto sekání urve. Všechny vás nezdolám, všechny ne, vím, úhlové velcí neb malí, s kvantifikátorem existenčním řešení přichází z dáli. Všechny vás nezdolám, všechny ne, vím, všelikých úhly vy stupňů. Angulus rectus, ten já roztřetím a na tři třicítky utnu!

parametrů. Ne vždy je však jednoduché nastavit parametry ovladačů tak, aby byl text nebo obrázek vytištěný přesně v požadovaném rozměru a na požadované pozici. Ukázalo se jednodušším umístit text nebo obrázek přesně v požadovaném rozměru a na požadované pozici do pdf souboru, v němž rozměr stránky odpovídá rozměru stránky v tiskárně. Ovladači tiskárny se pak řekne, aby dokument se vytiskl přímo bez dalších úprav a transformací.

Proto náš úkol vyřešíme vytvořením jednoduchého pdf dokumentu, do nějž v cyklu na jednotlivé stránky vložíme příslušné obrázky.

```
\input opmac
\margins/1 a4l (5,5,5,5)mm
\tmpnum0
```

`\leftskip=1cm`
`\rightskip=1cm`
`\parindent=-1cm`
`\parfillskip=-1cm`
	Při sazbě obsahu nebo rejstříku bývají neprvní řádky odsazeny zleva a neposlední řádky odsazeny zprava.

Sto roků v šachtě žil, mlčel jsem, sto roků kopal jsem uhlí, za sto let v rameni bezmasém svaly mi v železo ztuhly. Uhelný prach sedl do očí, rubíny ze rtů mi uhly, ze vlasů, z vousů a z obočí visí mi rampouchy uhlí. Chléb s uhlím beru si do práce, z roboty jdu na robotu, při Dunaji strmí paláce z krve mé a z mého potu. Sto roků v kopalně mlčel jsem, kdo mi těch sto roků vrátí? Když jsem pohrozil kladivem, kdekdo se začal mi smáti. Abych měl rozum, šel v kopalnu zas, pro pány robil jak prve: máchl jsem kladivem – teklo vráz na Polské Ostravě krve! Všichni vy na Slezské, všichni vy, dím, nech je vám Petr neb Pavel, mějž prs kryt krunýřem ocelovým, tisícům k útoku zavel, všichni vy na Slezské, všichni vy, dím, hlubokých páni vy dolů: přijde den, z dolů jde plamen a dým, přijde den, zúčtujem spolu!

`\parshape12`
`  0mm 96mm 13mm 96mm 18mm 96mm`
`  21mm 96mm 23mm 96mm 24mm 96mm`
`  24mm 96mm 23mm 96mm 21mm 96mm`
`  18mm 96mm 13mm 96mm 0mm 96mm`
`\parindent=0pt`
`\parfillskip=0pt`
	Příkazem `\parshape` lze nastavit libovolný tvar odstavce. První parametr určuje počet nastavovaných řádků a za ním následuje odsazení a šířka každého řádku. I v tomto případě jsou na příslušná místa vloženy mezery z `\parindent`, `\leftskip`, `\rightskip` a `\parfillskip`.

Sto roků v šachtě žil, mlčel jsem, sto roků kopal jsem uhlí, za sto let v rameni bezmasém svaly mi v železo ztuhly. Uhelný prach sedl do očí, rubíny ze rtů mi uhly, ze vlasů, z vousů a z obočí visí mi rampouchy uhlí. Chléb s uhlím beru si do práce, z roboty jdu na robotu, při Dunaji strmí paláce z krve mé a z mého potu. Sto roků v kopalně mlčel jsem, kdo mi těch sto roků vrátí? Když jsem pohrozil kladivem, kdekdo se začal mi smáti. Abych měl rozum, šel v kopalnu zas, pro pány robil jak prve: máchl jsem kladivem – teklo vráz na Polské Ostravě krve! Všichni vy na Slezské, všichni vy, dím, nech je vám Petr neb Pavel, mějž prs kryt krunýřem ocelovým, tisícům k útoku zavel, všichni vy na Slezské, všichni vy, dím, hlubokých páni vy dolů: přijde den, z dolů jde plamen a dým, přijde den, zúčtujem spolu!

```
\loop
  \ifnum\tmpnum<100 \advance\tmpnum1
  \pdfximage width\hsize{obrazek\the\tmpnum.jpg}
  \shipout\hbox{\pdfrefximage\pdflastximage}
\repeat
\end
```

3.5 Stránková montáž

Pomocí příkazu `\shipout` je možné provádět stránkovou montáž, například vysázet dokument jako brožuru. O tomto bylo více pojednáno v článku [8].

4 Output rutina

V kapitole 23 TEXbooku Donald Knuth o output rutině píše:

> *Chapter 22 taught you how to be a TEX master, i.e., a person who can produce complicated tables using* `\halign` *and* `\valign`*; the following material will take you all the way to the rank of Grandmaster, i.e., a person who can design output routines. When you are ready for this rank, you will be pleased to discover that—like alignments—output routines are not really so mysterious as they may seem at first.*

V následujícím textu si na příkladech ukážeme, že i běžný smrtelník si může output rutinu TEXu upravit k obrazu svému.

Output rutina je posloupnost tokenů uložená v token registru `\output`, kterou TEX zpracuje v okamžiku, kdy algoritmus stránkového zlomu přesně určí, jaký vertikální materiál se vysází na aktuální stránce. Algoritmus stránkového zlomu tento materiál uloží do vboxu `\box255` a úkolem output rutiny zpravidla je umístit tento vbox na stránku a vložit stránku do souboru `.pdf`. Proto by output rutina měla obsahovat příkaz `\shipout`.

Nepříjemnou vlastností output rutiny je, že dopředu není známo, ve kterém okamžiku se zavolá. Na základě principu nutného minima se output rutina zavolá až po ukončení algoritmu řádkového zlomu, kdy se do vertikálního seznamu vloží celý odstavec. To bývá většinou v místě, které se poté vysází na následující straně. V případě odstavců delších než jedna strana se může stát, že se output rutina zavolá až v místě, které se poté vysází o několik stran dále.

Jedním důsledkem předchozího je, že TEX interně musí output rutinu zpracovat uvnitř skupiny. Proto je také nastavení platné napříč dokumentem nutné uvnitř output rutiny měnit globálně. To se týká například registru `\pageno`, ve kterém bývá zvykem mít uloženo číslo strany. (V LATEXu se používá registr `\c@page`, k němuž lze přistupovat například přes LATEXová makra `\arabic{page}` nebo `\setcounter{page}`.)

Druhým důsledkem je, že nemá smysl v textu dokumentu registr `\pageno` používat. Pokud bychom jej použili, neměli bychom vůbec jistotu, že by v něm bylo uloženo číslo aktuální strany. (Na řádku 20 jsme k registru `\pageno` mimo output rutinu přistupovali. Nezajímala nás však jeho hodnota, ale pouze jsme jej potřebovali zvýšit o jedničku.)

4.1 Minimální příklad

Vyzkoušejme si vysázet jednoduchý dokument.

```
31 \output{\shipout\box255}
32
33 Hello
34
35 \vfil\break
36
37 World
38
39 \bye
```

Nejprve si na řádku 31 nadefinujeme minimální output rutinu. Ta pouze vezme obsah boxu 255 a vloží jej do pdf souboru. Výsledný dokument tak bude mít dvě stránky, kde na první stránce bude pouze text „Hello“ a na druhé stránce text „World“. Žádné další objekty uvedené v podsekci 1.7 se na stránku nevloží.

4.2 Jednoduchá output rutina

Nyní si vytvoříme jednoduchou output rutinu, která na stránku vysází `\box255` a pod něj vysází číslo strany zarovnané na střed.

```
40 \output{
41   \shipout\vbox{
42     \vbox to\vsize{\unvbox255}
43     \bigskip
44     \centerline{\tenrm\the\pageno}}
45   \global\advance\pageno1}
```

Musíme si uvědomit, že `\box255` může mít pokaždé jinou výšku a output rutina musí zařídit, aby navenek působil jako box stejné výšky. V našem případě bude výška rovna `\vsize`. V opačném případě by na každé stránce bylo číslo strany v jiné výšce, což určitě není žádoucí.

Na řádku 44 je pomocí `\tenrm` nastaven font, v němž bude vysázeno číslo strany. Pokud bychom font nenastavili, mohlo by se stát, že by na každé stránce bylo číslo strany vysázeno jiným fontem, konkrétně fontem, kterým byl aktuální v okamžiku zavolání output rutiny. (Uživatelé LaTeXu použijí jiný přepínač pro změnu fontu včetně velikosti.)

A opět nesmíme zapomenout globálně zvýšit číslo strany.

Výsledek může čtenář vidět na aktuální stránce.

4.3 Záhlaví a zápatí

V plainu jsou pro sazbu záhlaví a zápatí vyhrazeny token registry `\headline` a `\footline`. Do těchto registrů uživatel vloží horizontální materiál, který se má vysázet v řádku se záhlavím a zápatím.

Následující řádky vysázejí číslo strany v záhlaví vždy na vnějším okraji stránky. Na sudých stránkách bude nekonečně pružná mezera `\hfil` napravo od čísla a číslo tak bude vlevo. Na lichých stránkách bude více nekonečně pružná mezera `\hfill` nalevo od čísla a číslo tak bude vpravo. Podobně je v zápatí použita nulová mezera s nekonečnou pružností, která se roztáhne na celý řádek.

```
46 \headline{\ifodd\pageno\hfill\fi \tenrm\the\pageno\hfil}
47 \footline{\hss}
```

Uživatel si může samozřejmě nastavit jiné záhlaví a zápatí. Obsah registrů `\headline` a `\footline` pak v output rutině vysázíme pomocí příkazu `\the`.

```
48 \output{
49   \shipout\vbox{
50     \vbox to0pt{\vss\line{\the\headline}\bigskip}
51     \vbox to\vsize{\unvbox255}
52     \bigskip
53     \line{\the\footline}}
54   \global\advance\pageno1}
```

Na řádku 50 se vysází vertikální box nulové výšky, který neovlivní pozici sazby, přičemž materiál použitý v tomto boxu bude z boxu vyčnívat nahoru.

Výsledek může čtenář vidět na aktuální stránce.

Pokud chce čtenář tuto output rutinu použít v LaTeXu, musí si předefinovat makro `\line`

```
55 \def\line{\hbox to\hsize}
```

protože LaTeX definuje makro `\line` jiným způsobem. Tuto definici lze použít také lokálně, například vložit řádek 55 mezi řádky 48 a 49. Dále si čtenář musí v LaTeXu registry `\headline` a `\footline` nejprve nadeklarovat pomocí `\newtoks\headline` a `\newtoks\footline`. V LaTeXu se implicitně text záhlaví namísto do token registrů `\headline` a `\footline` ukládá do maker `\@oddhead`, `\@evenhead`, `\@oddfoot` a `\@evenfoot`, jejichž existence je před uživatelem utajena a musí k nastavení těchto maker použít jiná makra.

4.4 Čára pod záhlavím

Záhlaví je možné oddělit od textu horizontální čarou. Musíme si však uvědomit, že horizontální čáru je možné použít pouze ve vertikálním módu, přičemž obsah

registru `\headline` se expanduje v horizontálním módu. K tomu lze využít příkaz `\vadjust`, který vloží příslušný vertikální materiál pod aktuálně sázený horizontální box. Při použití output rutiny ze strany 56 můžeme registry `\headline` a `\footline` nastavit následovně.

```
\headline{\ifodd\pageno\hfill\fi \tenrm\the\pageno\hfil
  \vadjust{\vskip3pt\hrule\vskip-3.4pt}}
\footline{\hss}
```

Vertikální mezera `3pt` slouží k oddělení záhlaví od čáry. Čára má implicitní tloušťku `0.4pt`, a proto vertikální mezera `-3.4pt` posune sazbu zpět do původního bodu.

Čtenář si může promyslet, kde by bylo třeba použít příkaz `\vskip` a s jakými parametry, aby se vysázela čára nad zápatím.

Alternativní možností je zakomponovat horizontální čáru přímo do output rutiny. Uživatel pak nemusí používat konstrukci z řádku 57.

```
\output{
  \shipout\vbox{
    \vbox to0pt{\vss\line{\the\headline}
      \vskip3pt\hrule\vskip-3.4pt
      \bigskip}
    \vbox to\vsize{\unvbox255}
    \bigskip
    \line{\the\footline}}
  \global\advance\pageno1}
```

Výsledek může čtenář vidět na aktuální stránce.

4.5 Plovoucí záhlaví

Do záhlaví je možné vložit také text, který je závislý na textu vyskytujícím se na aktuální stránce. Například v knihách se do záhlaví vkládá nadpis aktuální sekce, ve slovnících se do záhlaví vkládá rozsah slov vyskytujících se na aktuální stránce.

Vytvoříme si makro `\sekce`, které vysází nadpis sekce a tento nadpis uloží do makra `\nazevsekce`. Cílem bude, aby poté output rutina mohla použít název sekce v záhlaví. V našem případě bude v záhlaví název sekce umístěn naproti číslu strany.

```
\def\sekce#1{\par\bigskip
  \noindent
  \def\nazevsekce{#1}%
  {\bf#1}\par
  \nobreak\bigskip}
```

```
\headline{\tenrm\strut
  \ifodd\pageno
    \nazevsekce\hss\the\pageno
  \else
    \the\pageno\hss\nazevsekce
  \fi}
```

Takto definované makro však nebude správně fungovat. Očekávali a chtěli bychom, aby se v zápatí vyskytl název sekce, která je na aktuální stránce. Nicméně makro `\nazevsekce` se expanduje až v output rutině a v tom okamžiku v něm bude uložen poslední název sekce, který byl definován před zavoláním output rutiny. Z důvodů popsaných na straně 54 se může jednat o nadpis, který se vyskytne až na následující straně.

Pro tyto účely je v TEXu implementován příkaz `\mark`. Tento příkaz uloží svůj argument do paměti a v output rutině jej expanduje. Konkrétně se v output rutině expanduje

- `\topmark` na argument posledního `\mark` použitého na předchozí straně,
- `\firstmark` na argument prvního `\mark` použitého na aktuální straně,
- `\botmark` na argument posledního `\mark` použitého na aktuální straně.

Pokud tedy budeme chtít, ať se do záhlaví dostane název první sekce na aktuální straně, musíme v makru `\sekce` použít příkaz `\mark` a v záhlaví použít příkaz `\firstmark`.

```
\def\sekce#1{\par\bigskip
  \noindent
  \mark{#1}%
  {\bf#1}\par
  \nobreak\bigskip}
\headline{\tenrm\strut
  \ifodd\pageno
    \firstmark\hss\the\pageno
  \else
    \the\pageno\hss\firstmark
  \fi}
```

Čtenář nechť si sám vyzkouší, na co se budou expandovat příkazy `\topmark`, `\firstmark` a `\botmark` v případě, že na aktuální straně nezačíná žádná sekce, a v případě před nastavením prvního `\mark`.

Použité makro `\strut` vytvoří vertikální podpěru pro text, aby záhlaví bylo na každé straně ve stejné výšce, nezávisle na tom, jestli text záhlaví je celý nad účařím, nebo zda zasahuje pod účaří.

Výsledek může čtenář vidět na aktuální stránce.

4.6 Poznámky pod čarou

Nyní naši output rutinu naučíme pracovat s poznámkami pod čarou. V PlainTEXu se vysázené poznámky pod čarou ukládají do vboxu s číslem `\footins`.[3] Proto do naši output rutiny přidáme test, zdali je tento vbox prázdný. A pokud není, vložíme vertikální mezeru velikosti `\skip\footins`, vhodně umístíme horizontální čáru[4] a pak vysázíme tento vbox.

```
\output{\shipout\vbox{
  \vbox to0pt{\vss\line{\the\headline}
    \vskip3pt
    \hrule
    \vskip-3.4pt
    \bigskip}
  \vbox to \vsize{
    \unvbox255
    \ifvoid\footins\else
      \vskip\skip\footins
      \vskip-2.4pt
      \hrule width1cm
      \vskip2pt
      \unvbox\footins
    \fi
    \vfil}
  \bigskip
  \line{\the\footline}}
  \global\advance\pageno1}
```

Výsledek může čtenář vidět na aktuální stránce.

4.7 Jednoduché slajdy

V této podsekci vytvoříme output rutinu, kterou lze použít při sazbě slajdů. Jako rozměr stránky vezmeme formát A5 na šířku a nastavíme okraje pro text slajdů.

```
\margins/1 a5l (15,15,15,10)mm
```

V output rutině si lokálně nastavíme nulové okraje. Pomocí příkazů `\hrule` a `\vrule` vytvoříme obdélníky dané barvy. Pomocí `\hskip` a `\vskip` musíme zajistit, aby se po nakreslení obdélníku pozice sazby dostala zpět na patřičné místo. Výšku a hloubku obdélníků na řádcích 116 a 126 volíme tak, aby jejich součet odpovídal okrajům nastaveným na řádku 109, přičemž hloubka obdélníků

[3] Přesný mechanismus sazby poznámek pod čarou je uveden v [2] v sekci 6.7.

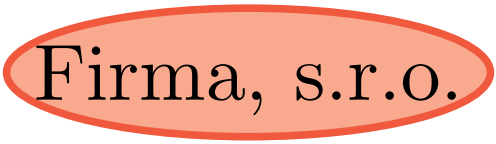
[4] Podle této čáry se poznámky nazývají pod čarou.

určuje vertikální umístění účaří příslušného textu v rámci obdélníku. Samotný `\box255` na slajdu vycentrujeme vertikálně i horizontálně na řádcích 120–123.

```
\output{\margins/1 a5l (0,0,0,0)mm
  \shipout\vbox to\vsize{
    \setcmykcolor{0 0 0.4 0.1}
    \hrule height\vsize width\hsize
    \vskip-\vsize
    \line{\setcmykcolor{0.5 0 0.5 0}%
      \vrule height10mm depth5mm width\hsize
      \hskip-\hsize \hskip15mm
      \typosize[20/]\bf\setcmykcolor{0 0 0 1}%
      \the\headline\hss}
    \vss
    \centerline{\setcmykcolor{0 0 0 1}%
      \vbox{\unvbox255\unskip}}
    \vss
    \line{\setcmykcolor{0.5 0 0.5 0}%
      \hskip\hsize \hskip-10mm
      \vrule height6mm depth4mm width10mm
      \hskip-10mm
      \typosize[12/]\bf\setcmykcolor{0 0 0 1}%
      \hss\the\pageno\hss
      }
    \hrule height0pt
  }
  \global\advance\pageno1
}
\headline{\firstmark}
```

Výsledek může čtenář vidět na aktuální stránce. (Ve skutečnosti byla output rutina drobně upravena, s ohledem na formát časopisu.)

Podobnou myšlenku můžeme použít při sazbě slovníku, kdy na hranu stránky umístíme počáteční písmeno slov na aktuální stránce.



4.8 Hlavičkový papír

Zcela analogicky můžeme vytvořit hlavičkové papíry nějaké firmy. Předpokládejme, že v souboru hlavicka.pdf máme uložen obrázek, který má stejné rozměry jako stránka a který se vloží na pozadí každé stránky dokumentu. Přes tento obrázek se pak bude sázet samotný text.

Přestože se obrázek bude vkládat na každou stránku dokumentu, my jej na řádcích 136–137 vložíme do dokumentu jen jednou a v output rutině budeme pouze vkládat odkaz na tento obrázek uložený v dokumentu. Tímto rozdílným přístupem oproti řádku 8 můžeme výrazně zmenšit velikost výsledného souboru, pokud bude dokument vícestránkový. Pro detaily opět odkazujeme čtenáře například na sekci 11.7 v [7].

Protože v output rutině lokálně na řádku 138 nastavujeme nulové okraje stránky, určuje vertikální mezera na řádku 145 vertikální odstup čísla strany od hrany stránky.

```
136 \pdfximage{hlavicka.pdf}
137 \edef\cislohlavicky{\the\pdflastximage}
138 \output{\margins/1 a5 (0,0,0,0)mm
139   \shipout\vbox to\vsize{
140     \pdfrefximage\cislohlavicky
141     \vskip-\vsize \vskip30mm
142     \centerline{\vbox{\unvbox255}}
143     \vss
144     \line{\hss\tenrm\the\pageno\hss}
145     \vskip20mm
146   }
147   \global\advance\pageno1
148 }
```

Výsledek může čtenář vidět na aktuální stránce.

Závěrem této sekce si uvědomme důležitý význam output rutiny. Autor dokumentu se stará pouze o obsah textu, tedy o `\box255`. Naproti tomu, grafik se *nezávisle* stará o umístění `\box255` a dalších náležitostí na stránku, tedy o output rutinu.

Ulice 256, Město, 655 36 +420 123 456 789 email@firma.cz

5 Pozice sazby

Může se stát, že autor bude potřebovat propojit grafiku s určitými částmi textu. Například když bude v prezentaci chtít zdůraznit, co s čím souvisí. Pro tyto účely může použít mechanismus pdfTEXu[5] na zjištění skutečné pozice konkrétního bodu sazby na stránce. Tento mechanismus úzce souvisí s output rutinou, protože skutečná pozice sazby je známa až v okamžiku vysázení sazby. Nicméně pro jeho použití do output rutiny zasahovat nemusíme.

5.1 Zjištění pozice

Pozici bodu TEX určuje v jednotkách `sp` měřených od levého dolního okraje stránky. Například pozice tohoto bodu je (12685637, 26756928). V bodě, jehož pozici chceme zjistit, použijeme příkaz `\pdfsavepos`. Souřadnice tohoto bodu pak získáme zápisem hodnot registrů `\pdflastxpos`, `\pdflastypos` do nějakého souboru.[6]

Konkrétně můžeme v dokumentu postupovat následovně. Pomocí příkazů

```
149 \newwrite\soubor
150 \openout\soubor pozice.txt
```

vytvoříme a otevřeme soubor `pozice.txt`. V dokumentu pak můžeme psát[7]

```
151 ... pozice tohoto
152 \pdfsavepos
153 \write\soubor{(\the\pdflastxpos, \the\pdflastypos)}%
154 bodu je ...
```

Poté, co zavřeme soubor `pozice.txt` příkazem

```
155 \closeout\soubor
```

bude v tomto souboru uložen řádek

```
156 (12685637, 26756928)
```

Jestliže nám jednotky `sp` nevyhovují, můžeme použít příkaz `\dimexpr`, který načte délkový výraz (v našem případě `12685637sp`), a příkaz `\the`, který hodnotu tohoto výrazu vypíše v jednotkách `pt`. Místo řádku 153 pišme

```
157 \write\soubor{\the\dimexpr\pdflastxpos sp,
158   \the\dimexpr\pdflastypos sp)}%
```

[5] Jedná se až o záležitost pdfTEXu, původní TEX tímto mechanismem vybaven není.

[6] Zápis do souborů TEX provádí až během exportu příslušného boxu do pdf souboru příkazem `\shipout`.

[7] Musíme si dát pozor, ať na řádku 153 nezavlečeme do textu nechtěnou mezeru.

Potom bude v souboru pozice.txt uložen řádek

```
159 (193.56746pt, 408.27832pt)
```

Samozřejmě si řádky 152 a 157–158 můžeme zautomatizovat vhodnými makry.

5.2 Propojení s METAPOSTem

Nyní si popíšeme, jak je možné pozice bodů, zjištěné v předchozí podsekci, předat programu METAPOST, který pak podle nich vytvoří příslušnou grafiku.

Vytvoříme si makro `\bod#1`, které do souboru pozice.txt zapíše pozici aktuálního bodu sazby tak, aby ji METAPOST uložil do proměnné `bod[#1]`.

```
160 \def\bod#1{\pdfsavepos
161   \write\soubor
162     {bod[#1]:=(\the\dimexpr\pdflastxpos sp,
163       \the\dimexpr\pdflastypos sp);}}
```

Po použití

```
164 ... pozice tohoto \bod{1}bodu je ...
```

bude v souboru pozice.txt uložen řádek

```
165 bod[1]:=(193.56746pt, 408.27832pt);
```

Dále si vytvoříme makro `\obr#1`, které v místě svého použití vloží do pdf souboru METAPOSTový obrázek s názvem obr.#1. Makro uloží aktuální pozici sazby do METAPOSTové proměnné[8] `bod[-#1]`, která bude sloužit jako referenční bod pro přesné umístění obrázku. Makro dále zjistí, zdali soubor s názvem obr.#1 existuje,[9] a pokud existuje, pak obrázek vloží na aktuální pozici makrem `\convertMPtoPDF`. Toto makro je definováno v souboru supp-pdf.tex.[10]

```
166 \input supp-pdf
167 \def\obr#1{\quitvmode\bod{-#1}%
168   \openin\testin obr.#1 \ifeof\testin\else
169     \closein\testin \convertMPtoPDF{obr.#1}{1}{1}%
170   \fi}
```

Makro `\obr` můžeme použít na libovolném místě na stránce, na kterou má být příslušný obrázek vložen.

[8] Indexy METAPOSTových proměnných mohou být libovolná čísla, konkrétně tato čísla mohou být i záporná nebo necelá.

[9] Uživatelé nepoužívající OPmac si nejdříve musejí příkazem `\newread\testin` alokovat řídicí sekvenci `\testin` pro načítaní souboru.

[10] Pokud LaTeXoví uživatelé načetli balíček color.sty, mají již soubor supp-pdf.tex interně načtený a řádek 166 mohou vynechat.

Zdrojový soubor METAPOSTových obrázků, v souladu s předchozím odstavcem, nazveme obr.mp. V tomto souboru musíme nejdříve deklarovat pole proměnných bod[] typu pair a načíst náš soubor pozice.txt. Pak vytvoříme jednotlivé obrázky obr.n, jak jsme zvyklí pomocí příkazů beginfig(n) a endfig, přičemž souřadnice bodu sazby, kde bylo použito makro \bod{k}, jsou uloženy v proměnné bod[k]. Na závěr každému obrázku jako jeho okraj nastavíme obdélník s nulovými rozměry umístěný do bodu bod[-n]. Tím zajistíme správné umístění obrázku v místě použití makra \obr{n} a zároveň zajistíme, že obrázek bude mít nulové rozměry a neposune se v tomto místě sazba.

```
pair bod[];
input pozice.txt
beginfig(n)
  ... obrázek obsahující bod[...]
  setbounds currentpicture to bod[-n]--cycle;
endfig;
end
```

Pro vygenerování dokumentu spustíme TEX tolikrát, než se ustálí pozice sazby, což je zpravidla dvakrát. Pak METAPOSTem vygenerujeme obrázky. A nakonec ještě jednou spustíme TEX, čímž obrázky vložíme.

5.3 Propojení s METAPOSTem a output rutinou

Uvedený postup můžeme ještě více zautomatizovat tím, že makro \obr umístíme přímo na nějaké[11] vhodné[12] místo output rutiny. Přitom můžeme využít registr \pageno a vložit obrázek příslušný pro danou stránku.

```
\output{...\obr{\the\pageno}...}
```

V METAPOSTu pak můžeme definovat makra

```
def beginobr(expr n)=
  CisloObr:=n;
  beginfig(n)
enddef;
def endobr=
  setbounds currentpicture to bod[-CisloObr]--cycle;
  endfig;
enddef;
```

[11] Makro \convertMPtoPDF můžeme použít pouze v horizontálním módu. V opačném případě se nejdříve horizontální mód zahájí, čímž se ale posune pozice sazby.

[12] Není vhodné obrázek umísťovat například do zápatí, neboť bychom uvnitř \shipout nejdříve vysázeli \box255 s textem stránky a přes něj bychom překreslili obrázek, čímž bychom zakryli text.

a řádky 173–176 zjednodušit následovně.

```
beginobr(n)
   ... obrázek obsahující bod[...]
endobr;
```

Odkazy

1. KNUTH, Donald Ervin. *Computers & Typesetting, Volume A: The TEXbook.* Addison-Wesley, 1986.
2. OLŠÁK, Petr. *TEXbook naruby.* Konvoj, 2001.
3. KNUTH, Donald Ervin. *Computers & Typesetting, Volume B: TEX: The Program.* Addison-Wesley, 1986.
4. KNUTH, Donald E.; PLASS, Michael F. *Breaking Paragraphs into Lines.* 1981.
5. LIANG, Frank. *Word Hy-phen-a-tion by Com-put-er.* 1983. Disertační práce. Stanford University, Department of Computer Science.
6. BEZRUČ, Petr. *Slezské písně.* Československý spisovatel, 1951.
7. OLŠÁK, Petr. *TEX pro pragmatiky.* CSTUG, 2016.
8. ŠUSTEK, Jan. Načítání souboru s argumenty v TEXu. *Zpravodaj CSTUG.* 2015, roč. 25, č. 1–2, s. 86–94. ISSN 1211-6661. Dostupné z DOI: 10.5300/2015-1-2/86.

Summary: Parameters of the Line Breaking Algorithm and the Output Routine and Their Applications for Typesetting in TEX

In the paper we go through the inner parts of TEX and we show how the particular characters of the input file .tex get to the output file .pdf. We focus on the line breaking algorithm explaining how its parameters affect the paragraph alignment. Then we focus on the output routine showing how to put the typeset text on the page. Finally we mention the way how to find the exact position of a particular point on the page with an application in METAPOST figures.

Keywords: TEX, line break, output routine

schaynova.lucie@seznam.cz, jan.sustek@osu.cz
Ostravská univerzita, Přírodovědecká fakulta, Katedra matematiky
30. dubna 22, CZ-701 03 Ostrava, Czech Republic