

Sazba odstavců do textových oblastí různé šířky

Tento text popisuje problém sazby textu do textových oblastí. Textovou oblastí je zde myšlena obdélníková oblast na dané pozici na stránce, do níž se má vysázet text. Tento druh sazby se často využívá v novinových člancích. Sazba je charakteristická tím, že se v ní vyskytují řádky o různé šířce. Z pohledu \TeX u problém nastává v situaci, kdy je text jednoho odstavce rozdělen do více oblastí, které mají různou šířku. V důsledku pružnosti vertikálních mezer je v době sazby odstavce nemožné zjistit, kolik řádků odstavce je třeba vysázet při jedné šířce a kolik při druhé šířce.

Na prosincové přednášce v Brně padla otázka, zda je možné tento problém řešit na úrovni \TeX u. Zda to je, nebo není možné, je patrné z následujícího textu.

V popisovaném řešení se textové oblasti definují makrem `\threadshape`. Toto makro má syntaxi podobnou syntaxi primitivu `\parshape`. Makro má $4n + 1$ argumentů (oddělených mezerou), kde n je první argument. Použití

`\threadshape n x1 y1 w1 h1 ... xn yn wn hn` říká \TeX u, že stránka se skládá z n textových oblastí, přičemž levý horní roh i -té oblasti má souřadnice (x_i, y_i) , šířka i -té oblasti je w_i a výška i -té oblasti je h_i . Levý horní roh stránky má souřadnice $(0\text{pt}, 0\text{pt})$. Pokud by bylo třeba text vysázet na více než n oblastí, bude sazba pokračovat v oblas \TeX 1, 2, ... na další straně.

Tento dokument je vysázen s nastavením `\threadshape 10 1cm 1cm 4cm 19cm 55mm 1cm 232mm 3cm 55mm 45mm 4cm 155mm 10cm 45mm 187mm 3cm 10cm 8cm 4cm 12cm 145mm 8cm 142mm 3cm 145mm 115mm 4cm 85mm 19cm 115mm 97mm 3cm 19cm 15cm 4cm 5cm 235mm 15cm 52mm 5cm`. To znamená, že se stránka skládá z 10 oblastí. První oblast začíná 1cm od levého okraje stránky a 1cm od horního okraje stránky a je 4cm široká a 19cm vysoká. Podobně pro zbývající oblasti.

Pro zjednodušení zápisu budu dále používat makro `\name#1`, které se expanduje na

```
\csname#1\endcsname
```

Makro `\threadshape` své argumenty uloží do makra `\name{tsxi}`, `\name{tsyj}`, `\name{tswi}` a `\name{tshi}`, aby je bylo možné použít později.

Sazba odstavce do i -té oblasti je vyřešena makrem `\addonepar`. Jeho hlavní myšlenka je sazba odstavce při

```
\hsize = \name{tswi}
```

a při vhodném nastavení `\parshape` do vboxu a jeho rozsekání makrem `\splitparagraph` do oblastí $i, i + 1, \dots$ pomocí `\vsplit`. Toto sekání se provádí tak dlouho, dokud se počet řádků v jednotlivých oblas \TeX neustálí.

Při nastavování argumentů `\parshape` se používají data uložená do makra `\linesinregions`. V tomto makru jsou data uložena jako čísla oddělená středníky, přičemž

poslední číslo je -1 (tato hodnota se používá v $\text{tes}\TeX$ na ukončení cyklu). Pokud makro `\linesinregions` je definováno jako

```
r0; r1; ...; rk; -1;
```

znamená to, že se příslušný odstavec má být rozdělen do oblastí $i, i + 1, \dots, i + k$, přičemž v oblasti $i + j$ bude jeho r_j řádků. Samotné nastavení `\parshape` se provede makrem `\genparshape`. Toto makro nastaví první argument `\parshape` na hodnotu $\sum_{j=0}^k r_j$ a další argumenty nastaví tak, aby prvních r_0 řádků mělo šířku `\name{tswi}`, dalších r_1 řádků šířku

`\name{tsw}(i + 1)`, atd. (Stačilo by nastavit pouze prvních $\sum_{j=0}^{k-1} r_j + 1$ řádků, ale to by znamenalo složitější testování konce cyklu, než pouhý test na hodnotu -1 .) Celé nastavení `\parshape` je poté uloženo do makra `\the-parshape`.

Při hledání hodnot makra `\linesinregions` se využívá makro `\linesinbox`. Toto makro spočítá počet řádků textu (přesněji počet hboxů) v daném vboxu. Toho dosáhne rozebráním tohoto vboxu primitivem `\unvbox` a postupným odebráním hboxů. Při použití argumentů `\parshape` se `display` rovnice počítá jako tři řádky. Proto také makro `\linesinbox` musí `display` rovnici počítat jako tři řádky. Že

příslušný hbox obsahuje `display` rovnici, makro zjistí podle přítomné vertikální mezery velikosti `\abovedisplay(short)skip`. Aby nemohlo dojít k záměně s jinou vloženou mezerou, je hodnota `\abovedisplay(short)skip` nastavena na hodnotu, která se od původní hodnoty liší o 0.001pt .

Jádrum algoritmu na zlom odstavce do oblastí je makro `\splitparagraph`. Při prvním průchodu tohoto makra je nastaveno `\parshape` na nulovou hodnotu. S touto hodnotou je daný odstavec pokusně vysázen do vboxu `\region`, ve kterém je předtím uložený

text i -té oblasti. Tento vbox usekne primitivem `\vsplit` na výšku `\name{tshi}`. Odseknutou část usekne na výšku `\name{tsh}(i + 1)`, atd. Přitom zjistí, kolik řádků je v jednotlivých vboxech, a výsledek uloží do makra `\linesinregions`. Protože první hodnota `\linesinregions` je nesprávná (obsahuje také řádky předchozích odstavců), je tato hodnota makrem `\countlinesinparagraph` opravena. Poté je nastaveno nové `\parshape`. S tímto nastavením se provede druhý průchod

makra `\splitparagraph`, který stejným způsobem zjistí počet řádků v jednotlivých vboxech. Podobně se provede třetí a další průchod a tyto průchody se opakují tak dlouho, až dva sobě jdoucí průchody dají stejnou hodnotu `\linesinregions`. S touto hodnotou je odstavec vysázen „naostro“.

Může se stát, že ani po dostatečném počtu průchodů makra `\splitparagraph` se hodnota `\linesinregions` neustálí. To

může nastat, pokud odstavec končí blízko hranice oblasti a hodnoty `\clubpenalty` a `(display)widowpenalty` jsou vysoké. Na to reaguje makro `\splitparagraph` následovně. Pokud počet průchodů dosáhne hodnoty `\softSPpasses` (v tomto případě hodnoty 8), sníží se makrem `\set-`

`softpenalties` některé penalty (tím se sníží typografické požadavky na zlom do oblastí). Pokud se ani potom hodnota `\linesinregions` neustálí a počet průchodů dosáhne hodnoty `\maxSPpasses` (zde hodnoty 16), provede se (pravděpodobně nesprávný) zlom podle posledního průchodu a vypíše se varovné hlášení. Možné řešení pomocí primitivu `\looseness` je popsáno níže.

Text aktuálně sázené oblasti je vložen do vboxu `\region`. V případě, že je box `\region` zaplněný, je makrem `\addtopagebox` tento box vložen do vboxu `\pagebox`. V boxu `\pagebox` je uložen text aktuální strany (přesněji vboxy všech dosud zkompletovaných oblastí). Pokud jsou zaplněny všechny oblasti na straně, zavolá se makro `\forceoutput`, které primitivem `\shipout` uloží box `\pagebox` do DVI/PDF souboru. Poté se začne plnit následující strana. Na ní jsou textové oblasti rozmístěny stejně jako na předcházející straně. Pokud se \TeX dostane na konec dokumentu, je (dosud neukončený) box `\region` poslán do boxu `\pagebox` a poté je proveden `\shipout`.

Uvedené řešení problému s sebou nese několik důsledků. Odstavec je při své sazbě vložen do `\vbox`, a proto jsou všechna přiřazení uvnitř odstavce lokální a mají-li být globální, musí se použít prefix `\global`. Není možné, aby nějaká skupina začala uvnitř jednoho odstavce a končila uvnitř jiného.

Při lámání odstavce se neprovádí expanze a odstavec je jako celek poslán algoritmu řádkového zlomu. Proto může

obsahovat například příkazy pro změnu fontu nebo matematické vzorce. Ty navíc mohou být zlomeny, případně může být použito makro `\opakuj`. Podobně může odstavec obsahovat i `display` matematiku. Dále je možno používat některé registry ovlivňující řádkový zlom, jako je `\leftskip`, `\rightskip`, `\parfillskip` nebo `\looseness`. Naopak nelze (zatím) použít registry `\hangindent`, `\hangafter` a `\parshape`, protože parametry `\parshape` jsou algoritmem předefinovány.

V dalších verzích je třeba zautomatizovat makro `\addonepar` pomocí `\everypar`. V důsledku několikanásobné sazby odstavce je ale nutné s primitivem `\everypar` pracovat opatrně. Je také nutné lépe vyřešit vkládání jiného materiálu než textu, které je v tuto chvíli řešeno makrem `\addtoeoregion` a nemusí dávat správné výsledky.

V tomto odstavci jsou vidět možnosti uvedeného řešení. Je nastaveno

```
\parfillskip = 0pt
```

a je použito makro `\opakuj` Petra Olšáka. Za normálních okolností by tento odstavec končil na začátku osmé oblasti, ale nastavením

```
\looseness = 15
```

se docílilo toho, že odstavec končí až na konci deváté oblasti. Zopakujme si jednoduchou matematiku a počítejme od jedné do dvou set dvaceti dvou, $0 + 1 = 1, 1 + 1 = 2, 2 + 1 = 3, 3 + 1 = 4, 4 + 1 = 5, 5 + 1 = 6, 6 + 1 = 7, 7 + 1 = 8, 8 + 1 = 9, 9 + 1 = 10, 10 + 1 = 11, 11 + 1 = 12, 12 + 1 = 13, 13 + 1 = 14, 14 + 1 = 15, 15 + 1 = 16, 16 + 1 = 17, 17 + 1 = 18, 18 + 1 = 19, 19 + 1 = 20, 20 + 1 = 21, 21 + 1 = 22, 22 + 1 = 23, 23 + 1 = 24,$

$24 + 1 = 25, 25 + 1 = 26, 26 + 1 = 27, 27 + 1 = 28, 28 + 1 = 29, 29 + 1 = 30, 30 + 1 = 31, 31 + 1 = 32, 32 + 1 = 33, 33 + 1 = 34, 34 + 1 = 35, 35 + 1 = 36, 36 + 1 = 37, 37 + 1 = 38, 38 + 1 = 39, 39 + 1 = 40, 40 + 1 = 41, 41 + 1 = 42, 42 + 1 = 43, 43 + 1 = 44, 44 + 1 = 45, 45 + 1 = 46, 46 + 1 = 47, 47 + 1 = 48, 48 + 1 = 49, 49 + 1 = 50, 50 + 1 = 51, 51 + 1 = 52, 52 + 1 = 53, 53 + 1 = 54, 54 + 1 = 55, 55 + 1 = 56, 56 + 1 = 57, 57 + 1 = 58, 58 + 1 = 59, 59 + 1 = 60, 60 + 1 = 61, 61 + 1 = 62, 62 + 1 = 63, 63 + 1 = 64, 64 + 1 = 65, 65 + 1 = 66, 66 + 1 = 67, 67 + 1 = 68, 68 + 1 = 69, 69 + 1 = 70, 70 + 1 = 71, 71 + 1 = 72, 72 + 1 = 73, 73 + 1 = 74, 74 + 1 = 75, 75 + 1 = 76, 76 + 1 = 77, 77 + 1 = 78,$

$78 + 1 = 79, 79 + 1 = 80, 80 + 1 = 81, 81 + 1 = 82, 82 + 1 = 83, 83 + 1 = 84, 84 + 1 = 85, 85 + 1 = 86, 86 + 1 = 87, 87 + 1 = 88, 88 + 1 = 89, 89 + 1 = 90, 90 + 1 = 91, 91 + 1 = 92, 92 + 1 = 93, 93 + 1 = 94, 94 + 1 = 95, 95 + 1 = 96, 96 + 1 = 97, 97 + 1 = 98, 98 + 1 = 99, 99 + 1 = 100, 100 + 1 = 101, 101 + 1 = 102, 102 + 1 = 103, 103 + 1 = 104, 104 + 1 = 105, 105 + 1 = 106, 106 + 1 = 107, 107 + 1 = 108, 108 + 1 = 109, 109 + 1 = 110, 110 + 1 = 111, 111 + 1 = 112, 112 + 1 = 113, 113 + 1 = 114, 114 + 1 = 115, 115 + 1 = 116, 116 + 1 = 117,$

$117 + 1 = 118, 118 + 1 = 119, 119 + 1 = 120, 120 + 1 = 121, 121 + 1 = 122, 122 + 1 = 123, 123 + 1 = 124, 124 + 1 = 125, 125 + 1 = 126, 126 + 1 = 127, 127 + 1 = 128, 128 + 1 = 129, 129 + 1 = 130, 130 + 1 = 131, 131 + 1 = 132, 132 + 1 = 133, 133 + 1 = 134, 134 + 1 = 135, 135 + 1 = 136, 136 + 1 = 137, 137 + 1 = 138, 138 + 1 = 139, 139 + 1 = 140, 140 + 1 = 141, 141 + 1 = 142, 142 + 1 = 143, 143 + 1 = 144, 144 + 1 = 145, 145 + 1 = 146, 146 + 1 = 147, 147 + 1 = 148, 148 + 1 = 149, 149 + 1 = 150, 150 + 1 = 151, 151 + 1 = 152, 152 + 1 = 153, 153 + 1 = 154, 154 +$

$+ 1 = 155, 155 + 1 = 156, 156 + 1 = 157, 157 + 1 = 158, 158 + 1 = 159, 159 + 1 = 160, 160 + 1 = 161, 161 + 1 = 162, 162 + 1 = 163, 163 + 1 = 164, 164 + 1 = 165, 165 + 1 = 166, 166 + 1 = 167, 167 + 1 = 168, 168 + 1 = 169, 169 + 1 = 170, 170 + 1 = 171, 171 + 1 = 172, 172 + 1 = 173, 173 + 1 = 174, 174 + 1 = 175, 175 + 1 = 176, 176 + 1 = 177, 177 + 1 = 178, 178 + 1 = 179, 179 + 1 = 180,$

$180 + 1 = 181, 181 + 1 = 182, 182 + 1 = 183, 183 + 1 = 184, 184 + 1 = 185, 185 + 1 = 186, 186 + 1 = 187, 187 + 1 = 188, 188 + 1 = 189, 189 + 1 = 190, 190 + 1 = 191, 191 + 1 = 192, 192 + 1 = 193, 193 + 1 = 194, 194 + 1 = 195, 195 + 1 = 196, 196 + 1 = 197, 197 + 1 = 198, 198 + 1 = 199, 199 + 1 = 200, 200 + 1 = 201, 201 + 1 = 202, 202 + 1 = 203, 203 + 1 = 204, 204 + 1 =$

$= 205, 205 + 1 = 206, 206 + 1 = 207, 207 + 1 = 208, 208 + 1 = 209, 209 + 1 = 210, 210 + 1 = 211, 211 + 1 = 212, 212 + 1 = 213, 213 + 1 = 214, 214 + 1 = 215, 215 + 1 = 216, 216 + 1 = 217, 217 + 1 = 218, 218 + 1 = 219, 219 + 1 = 220, 220 + 1 = 221, 221 + 1 = 222.$

Mgr. Jan Šustek
Katedra matematiky
Přírodovědecká fakulta
Ostravská univerzita v Ostravě
<http://sustek.wz.cz>
jan.sustek@osu.cz