

latex.ltx リーディング

第5回資料

東大 T_EX 愛好会

2015 年 6 月 12 日

1 水平スペース (朝倉)

今回は I^AT_EX の水平スペースについて簡単に見ていくことにする。例によって、関連する定義を記載する。

```
1308 \DeclareRobustCommand\hspace{\@ifstar\@hspacer\@hspace}
1309 \def\@hspace#1{\hskip #1\relax}
1310 \def\@hspacer#1{\vrule \@width\z@\nobreak
1311         \hskip #1\hskip \z@skip}
1312 \newskip\fill
1313 \fill = 0pt plus 1fill
1314 \def\stretch#1{\z@ \@plus #1fill\relax}
1315 \def\thinspace{\kern .16667em }
1316 \def\negthinspace{\kern -.16667em }
1317 \def\enspace{\kern .5em }
1318 \def\enskip{\hskip .5em\relax}
1319 \def\quad{\hskip 1em\relax}
1320 \def\qqquad{\hskip 2em\relax}
```

ほとんど T_EX のプリミティブをわかりやすく使用しているだけであり、解説するようなことはあまりない。ひとまず、`\hspace` から順に見ていくことにする。

1308 行目の定義から明らかなように、`\hspace` は堅い制御綴として定義されており、また * の有無によって分岐が行われるようになっている。* がない場合には `\@hspace` が展開されて、プリミティブ `\hskip` を用いた任意幅のグルーが挿入されるようになっている。後ろに `\relax` が置かれている理由は後述する。

一方、`\hspace*` は一般に強制改行 `\\` の直後で利用されるものだが、この場合は `\@hspacer` が展開される。こちらの方が、先ほどの `\@hspace` よりはやや定義が長くなるが、いずれにせよそれほど複雑ではない。水平モードに切り替えるため幅 0pt の `\vrule` が挿入されている。なお、ここに登場する `\@width` は単なる文字列の置き換えマクロである。ついでなので、`latex.ltx` の関連する部分をまとめて引っ張っておく。

```
563 \def\@height{height} \def\@depth{depth} \def\@width{width}
564 \def\@minus{minus}
565 \def\@plus{plus}
566 \def\hb@xt@{\hbox to}
```

こうした置換えは、T_EX の処理時にメモリ使用量を軽減するための工夫らしい。

また、`\z@skip` は 355 行目で `\newskip\z@skip \z@skip=0pt plus0pt minus0pt` と定義されている。`\newskip` は簡単に言うと寸法レジスタを生成するマクロである。後半の `plus0pt minus0pt` はそれぞれ伸長度と収縮度を指定しており、要するに `\z@skip` では一切の伸縮を許容していないことになる。なお、`\hspace` などの定義末尾に `\relax` が置かれているのはこれらの命令の後ろにたまたま “plus” などの文字列が現れた際、それが伸長度や伸縮度を指定する意味に解釈されないようにするためである。

1313 行目に登場する `fill` は特殊な単位であり、簡単に言えば「無限」の長さを表現する。因みに `fil` や

filll という単位も存在し、1 の数が多いほどより強力な無限を表現する.

■疑問 1 1311 行目の `\hskip \z@skip` は `\relax` ではだめなのか.

■疑問 2 `\hskip` と `\kern` の使い分けはどのように行われているのか.