

DTP1

(příprava textu pomocí počítače)

Kapitola 7 / T_EX

Petr Lobaz, 8. 4. 2010

Struktura T_EXu

- input procesor
 - konverze vstupu do platformově nezávislé podoby
- token procesor
 - tvorba posloupností tokenů
- expand procesor
 - expanze maker na primitivní příkazy
- hlavní procesor
 - tvorba sazby, výstup

- činnost řízena hlavním procesorem

Input procesor

- překódování vstupu do ASCII
- T_EX obsahuje i inverzní algoritmus – pro výstup na terminál příkazem `\write`

Token procesor

- řídicí sekvence interpretuje jako token (např. `\par`)
- skupinu mezer interpretuje jako jednu mezeru
- prázdný řádek interpretuje jako token `\par`
- ostatní znaky se stávají tokenem znak-kategorie (např. A_{11})
- některé kategorie zpracuje token procesor sám (např. $\%_{14}$)

Token procesor

KATEGORIE

0	uvození řídicí sekvence	\
1	otevření skupiny	{
2	uzavření skupiny	}
3	matematický režim	\$
4	oddělovač buněk	&
5	konec řádku	^^M (ASCII 13)
6	označení parametrů maker	#
7	horní index	^
8	dolní index	_
9	znak, který se ignoruje	^^@ (ASCII 0)
10	mezera	␣

Token procesor

11	písmeno	a-z, A-Z
12	ostatní znaky	
13	aktivní znaky	~
14	komentář	%
15	nedovolený znak	^^? (ASCII 127)

- nastavení kategorie
`\catcode`\znak = <číslo>`

Token procesor

PŘÍKLADY

```
\catcode \@=11
\def\m@jemakro{...}
\def\uzivatelske{...\m@jemakro...}
\catcode \@=12
```

```
{\catcode \*=13
\def*{\bullet}
Ted' snadno sázíme puntíky: * * *}
```

Expand procesor

DEFINICE MAKER

<code>\def</code>	definice makra
<code>\edef</code>	definice makra, ve fázi učení probíhá expanze
<code>\global</code>	definice je platná i mimo skupinu
<code>\long</code>	umožní token <code>\par</code> v parametru
<code>\outer</code>	makro <code>\outer\def</code> se nesmí objevit v těle jiných definic
<code>\gdef</code>	<code>\global\def</code>
<code>\xdef</code>	<code>\global\edef</code>

Expand procesor

DEFINICE MAKER

```
\def<řídící sekvence><maska parametrů>  
  {<tělo definice>}
```

- řídicí sekvence
 - znak s kategorií 0
 - znaky s kategorií 11 (písmena) nebo jeden znak s jinou kategorií
- tělo definice
 - „balancovaný“ text – stejný počet { a }
 - je-li třeba „nebalancovaně“ otvírat/zavírat skupinu, můžeme použít `\begingroup`, `\endgroup`

Expand procesor

- maska parametrů
 - nepovinná, začíná za řídicí sekvencí, končí {
 - `<separátor>#1<separátor>#2<separátor>#3...#9<separátor>`
 - `<separátor>` nepovinný
 - obsahem parametru „balancovaný“ text

```
\def\sazbamailu#1@#2\konec{{\bf #1}@#2}  
\def\mail#1{\sazbamailu #1\konec}
```

Expand procesor

špatně:

```
\def\beginpункiky{\begingroup  
  \catcode`*=13  
  \def*{$\bullet$}}  
\def\endpункiky{\endgroup}
```

správně:

```
{\catcode`*=13  
  \gdef\beginpункiky{\begingroup  
    \catcode`*=13  
    \def*{$\bullet$}}}  
\def\endpункiky{\endgroup}
```

Expand procesor

ZMĚNA POŘADÍ EXPANZE

- primitivní příkaz `\expandafter`

`\expandafter <token1> <token2> ⇒`
`<token1> <výsledek expanze token2>`

`\expandafter <token1>\expandafter <token2><token3>⇒`
`<token1> <token2> <výsledek expanze token3>`

Expand procesor

PŘÍKLADY

špatně:

```
\uppercase{\romannumeral\num}
```

dobře:

```
\expandafter\uppercase  
  \expandafter{\romannumeral\num}
```

Hlavní procesor

REGISTRY

- 256 uživatelských registrů každého typu
- obsah platný v rámci skupiny

- číslo: `\count0` – `\count255`
- rozměr: `\dimen0` – `\dimen255`, `\hsize`, ...
- mezera: `\skip0` – `\skip255`, `\baselineskip`, ...
- matematická mezera
- řetězec tokenů: `\toks0` – `\toks255`, `\everypar`, ...
- box: `\box0` – `\box254`

Hlavní procesor

PRÁCE S REGISTRY

`\count0 = 1`

`\countdef\pageno=0 \pageno=1`

`\newcount\mujcitac \mujcitac=5`

- alokační příkazy
`\newcount`, `\newdimen`, `\newskip`, `\newmuskip`,
`\newtoks`, `\newbox`

Hlavní procesor

PRÁCE S REGISTRY

`\count5 = 3`

`\count\count5 = 18`

`\advance\count5 by 1`

`\dimen0 = 5mm`

`\dimen1 = 1.3\dimen0`

`\dimen2 6mm`

Hlavní procesor

PRÁCE S REGISTRY

`\skip0 = 1pt plus 3 pt minus 2pt`

`\skip1 = 1pt`

`\skip2 = 0pt plus 1fil`

POUŽÍVANÉ ZKRATKY

`\hfil` `\hskip 0pt plus 1fil`

`\hfill` `\hskip 0pt plus 1fill`

`\hss` `\hskip 0pt plus 1fil minus 1fil`

`\vfil` `\vskip 0pt plus 1fil`

`\vfill` `\vskip 0pt plus 1fill`

`\vss` `\vskip 0pt plus 1fil minus 1fil`

Hlavní procesor

PODMÍNĚNÝ PŘÍKAZ

`\if... <podmínka><akce> \else <akce> \fi`

- primitivní příkazy
`\if, \ifx, \ifcat, \ifnum, \ifodd, \ifdim, \ifeof, \iffalse, \iftrue, \ifhbox, \ifvbox, \ifvoid, \ifhmode, \ifvmode, \ifmmode, \ifinner, \ifcase`

Hlavní procesor

- implementace cyklu

```
\def\loop#1\repeat{\def\body{#1}\iterate}  
\def\iterate{\body \let\next\iterate  
                \else\let\next\relax\fi \next}  
\let\repeat=\fi
```

příklad:

```
\newcount\citac \citac=1  
\loop  
  \the\citac\l\advance\citac by 1  
  \ifnum\citac<10\repeat
```

Režimy hlavního procesoru

- hlavní vertikální mód
 - výchozí režim práce T_EXu
- odstavcový horizontální mód
 - sazba odstavce
 - začíná, jakmile se objeví první prvek odstavce (např. písmeno)
 - končí příkazem `\par` nebo prázdnou řádkou; T_EX vyhodnotí řádkový zlom a vrátí se do vertikálního módu

Režimy hlavního procesoru

- vnitřní vertikální mód
 - neumožňuje stránkový zlom, uvnitř `\vbox`
- vnitřní horizontální mód
 - neumožňuje řádkový zlom, uvnitř `\hbox`
- display matematický mód
 - sazba matematiky na zvláštní řádek
 - začíná i končí sekvencí `$$`
 - neumožňuje řádkový zlom
- vnitřní matematický mód
 - sazba matematiky v řádce
 - začíná i končí znakem `$`
 - za jistých podmínek umožňuje zalomení

Režimy hlavního procesoru

VERTIKÁLNÍ MÓD

- tvoří vertikální seznam materiálu
 - box
 - linka (`\hrule`)
 - výplněk (`\kern`)
 - pružný výplněk (`\vskip, ...`)
 - penalta (`\penalty`)
 - další prvky

Režimy hlavního procesoru

HORIZONTALNÍ SEZNAM

- tvoří horizontální seznam materiálu
 - znak
 - box
 - linka (`\vrule`)
 - výplněk (`\kern`)
 - pružný výplněk (`\hskip, \ ,...`)
 - penalta (`\penalty`)
 - dělení slova (`\discretionary`)
 - další prvky

Režimy hlavního procesoru

BOXY

- prvky sazby, složeny z prvků
 - `\hbox` – horizontálního seznamu
 - `\vbox`, `\vtop`, `\vcenter` – vertikálního seznamu
- parametry šířka, výška, hloubka

```
\setbox0=\hbox{pokusný box}  
\the\wd0 \the\ht0 \the\dp0  
\copy0\box0
```

```
\def\obvod#1{\vbox{%  
  \hrule\hbox{\vrule #1 \vrule}\hrule}}  
\obvod{text}
```

Režimy hlavního procesoru

- box o daném rozměru
 - `\hbox to 1cm{pokus\hfil}`
 - `\hbox spread \hsize{toto je text}`
 - `\setbox0\hbox{pokus}\wd0=1cm`
- změna umístění boxu
 - pro `\hbox`: `\raise`, `\lower`
 - pro `\vbox`: `\moveleft`, `\moveright`

Režimy hlavního procesoru

MEZERY VE VERTIKÁLNÍM SEZNAMU

- `\baselineskip` (12 pt)
- `\lineskip` (1 pt)
- `\lineskiplimit` (0 pt)
- `\parskip` (0 pt)

Zalamování

ALGORITMY T_EXU

- řádkový zlom
- stránkový zlom

MÍSTA ZLOMU

- pružný výplněk
- penalta
- dělení slov (`\discretionary`)

```
\def\podpis#1\par{\unskip  
\nobreak\hfill\penalty100\hskip2em\hbox{ }\nobreak\hfill  
\hbox{\it #1}\par}
```

Zalamování

\DISCRETIONARY

- `\discretionary{<pre>}{<post>}{<nobreak>}`

- příklady

```
10\discretionary{}{tisíc}{\kern.2em000}
```

```
5 \discretionary{\ až}{}{--} 10
```

```
\- je \discretionary{-}{}{}
```

```
po\-kus\-ný text
```

Zalamování

ŘÁDKOVÝ ZLOM

- začíná povelom `\par`
- T_EX vloží
`\unskip\penalty10000\hskip\parfillskip\penalty-10000`
- nalezení míst pro řádkový zlom – rozložení seznamu do boxů šířky `\hsize`
 - ovlivněno `\leftskip`, `\rightskip`, `\hangindent`,
`\hangafter`, `\parshape`
 - algoritmus minimalizuje „badness“

Zalamování

- sestavení odstavce
 - vložení řádek do vertikálního seznamu
 - za každý řádek materiál z `\vadjust`
 - vložení penalt mezi řádky – `\interlinepenalty`,
`\clubpenalty`, `\widowpenalty`, `\brokenpenalty`

Zalamování

TVAR ODSTAVCE

- `\parindent`, `\parfillskip`
- `\leftskip`, `\rightskip`, `\hangindent`, `\hangafter`, `\parshape`

- vynucení kratšího východového řádku
`\parfillskip=2em plus 1fil`
- sazba na levý prapor
`\parindent=0pt`
`\leftskip=0pt`
`\rightskip=0pt plus 1fil`

Zalamování

- sazba do bloku s centrovaným východovým řádkem
 - `\leftskip=0pt plus 1fil`
 - `\rightskip=0pt plus -1fil`
 - `\parfillskip=0pt plus 2fil`
 - `\parindent=0pt`

Zalamování

- **iniciála**

```
\font\inicialfont=csbx10 scaled 2700
```

```
\newbox\inicialbox
```

```
\def\iniciala#1{\setbox\inicialbox
```

```
  \hbox{\lower12pt\hbox{\inicialfont #1}}%
```

```
  \hskip.2em}%
```

```
\hangindent=\wd\inicialbox%
```

```
\dp\inicialbox=0pt%
```

```
\hangafter=-2%
```

```
\noindent%
```

```
\hskip-\wd\inicialbox\box\inicialbox}
```

```
\iniciala První odstavec textu, ...
```

Zalamování

STRÁNKOVÝ ZLOM

- plnění strany
 - převod materiálu z přípravné oblasti na akt. stranu
 - hledání nejlepšího zlomu
 - spuštění uzavření strany
 - registry `\pagetotal`, `\pagegoal (= \vsize)`
- uzavření strany
 - vrácení materiálu za místem zlomu do přípravné oblasti
 - kompletace strany do `\box255`
 - vyvolání výstupní rutiny `\output`
 - spuštění algoritmu plnění strany

Zalamování

VÝSTUPNÍ RUTINA

```
\output={\shipout\vbox{%  
  \line{Záhlaví\hfil\the\pageno}  
  \bigskip\box255}  
\global\advance\pageno by 1}
```

Zalamování

VÝSTUPNÍ RUTINA PLAINU

```
\newtoks\headline \headline={\hfil}
\newtoks\footline \footline={\hss\tenrm\folio\hss}
\def\folio{\ifnum\pageno<0 \romannumeral-\pageno \else\number\pageno \fi}
\def\advancepageno{\ifnum\pageno<0 \global\advance\pageno by -1
\else\global\advance\pageno by 1 \fi}

\output{\plainoutput}
\def\plainoutput{\shipout\vbox{\makeheadline\pagebody\makefootline}%
\advancepageno
\ifnum\outputpenalty>-20000 \else\dosupereject\fi}
\def\pagebody{\vbox to\vsizel{\boxmaxdepth\maxdepth \pagecontents}}
\def\makeheadline{\vbox to 0 pt{\vskip -22.5 pt
\line{\vbox to 8.5 pt{\}\the\headline}\vss}\nointerlineskip}
\def\makefootline{\baselineskip 24 pt\lineskiplimit 0 pt\line{\the\footline}}
\def\dosupereject{\ifnum\insertpenalties>0
\line{\}\kern-\topskip\nobreak\vfill\supereject\fi}
```

Zalamování

CHYBOVÁ HLÁŠENÍ

- `underfull \hbox` – podtečení
 - řádka nebo `hbox` obsahuje málo materiálu ⇒ bylo nutno příliš roztáhnout mezery
 - `badness` – jak „ošklivé“ je roztažení, max. 10 000
- `overfull \hbox` – přetečení
 - řádka nebo `hbox` obsahuje mnoho materiálu ⇒ část materiálu přesahuje `box`
 - hlášení o rozměru přesahujícího materiálu
- `underfull \vbox`, `overfull \vbox`
 - pro stránku nebo `vbox`

Činnost T_EXu

VSTUP T_EXU

- dokument (.tex), formát (binární forma)
- metrika fontu (.tfm)
- v L^AT_EXu navíc .cls, .sty, .toc, .lof, .lot, .aux, ...

VÝSTUP T_EXU

- tiskové strany (.dvi), zpráva o překladu (.log)
- v L^AT_EXu navíc .toc, .lof, .lot, .aux, .fd, ...

ZPRACOVÁNÍ DVI

- bitmapové fonty (.pk), volání Metafontu (.mf, ...)
- virtuální fonty (.vf), mapovací tabulky (.map)