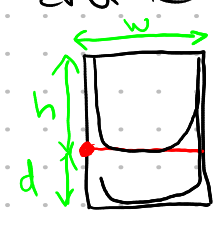


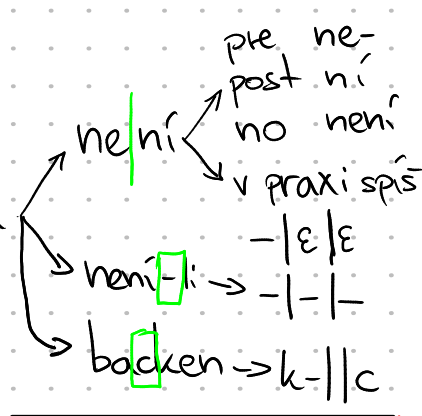
- TEX nepracuje s fonty, jen rámečky
 - dnes už PDFTEX -
 - děti už rovnou pdf

občas předefinová makra pod sebou - vommit
 L^AT_EX ConT_EXt OpMac
 p_ki_nT_EX ← většína maker, která známe
 primitiva



Zalamování textu

- jeden dlouhý horizontální seznam
- box, písmenko, ligatura
- role
- discretionary break - rozdělování jednoho slova
 - pre, post, no break
- whatsit - není vidět ve výsledném textu
- vertikální materiál - např. poznámky pod čarou



za zabmením vše zmiží

- lepidlo - mezera s ideálním rozměrem, stláčitelností a roztahitelností
 - může obsahovat leader - Bruh.....strana 9
- kern - pevná mezera
- penalta - hodnocená pokuta za zlom -10 000 až 10 000
- math on/off

musím zlomit
 nesmím zlomit

- místa zlomu (potenciální)



- lepidlo, před kterým je nondiscardable a jsme mimo matematiku
- před kernem, je-li za ním lepidlo a není v \$
- v math-off, je-li za ním lepidlo

- discretionary break
- penalta

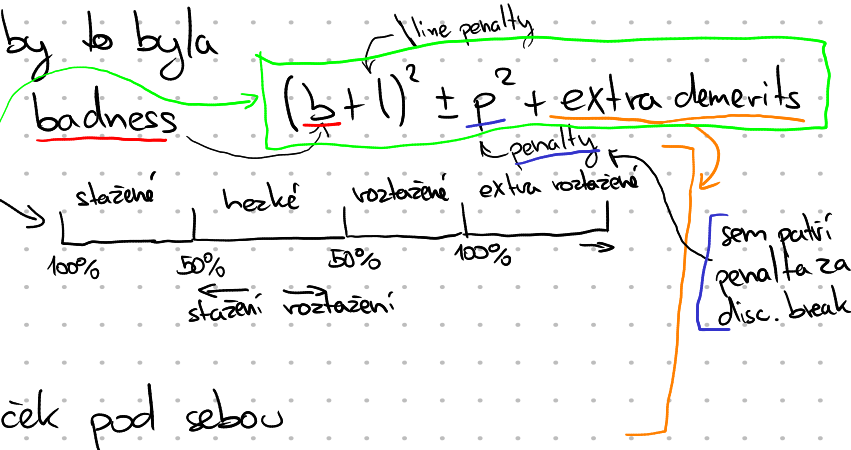
- pro každý blok h. materiálu se ho pokusíme vložit do řádku a spočítáme

"badness" - jak špatná situace by to byla

- tvoří se graf, hodnoty hran jsou badness

- čtyři typy řádku dle rozložení

- penalizace, pokud se řádky liší o více než jednu kategorii
- penalizace, pokud děláme víc pomlček pod sebou



- tvoří se DAG, v něm se hledá nejlepší cesta

• nejdrív se zahodí vše větší než |pre tolerance

• poté se pustí hyphenation alg., který přidá |

• druhý průchod, zahazuje se dle tolerance

• emergency stretch - k roztaženosti celého řádku se přičítá emergency stretch

- |looseness -> chceme o řádek více či méně - kvůli lámání stránek např.

pre	post	no
-		

rozdelí se mezi jednotlivá lepidla

Jak vyrobit neodstranitelnou mezeru.

- |penalty 0 |hbox{} |penalty 10000 <glue>

↑ |null

↳ horizontální Pakovací algoritmus

- počítá se pro každý box
- to (...) nebo spread(...)
- ① spočítá se:
 - základní rozměr
 - roztažitelost (f_0, f_1, f_2, f_3)
 - smrštitelost -||-

- ② když musím roztažovat
 - hledám první nenulový součet f_0, f_1, \dots
 - rozdělím rozměr, o který je třeba se roztažnout v poměru mezi prázdné prvky

- ③ určíme zbylé rozměry

- výška a hloubka - maximum ze všech unitů
- neurčitě rozměry - výška lvrule může odpovídat
- šířka
- badness : $\min(1000, \lceil 100 \cdot (\frac{\text{deformace}}{\text{průzornost}})^3 \rceil)$

- podpora
 - min. výška a hloubka řádku
 - ručně - lvrule s width 0
 - lstrut - odpovídá defaultním fontům

- luhbox
- lraise, llower

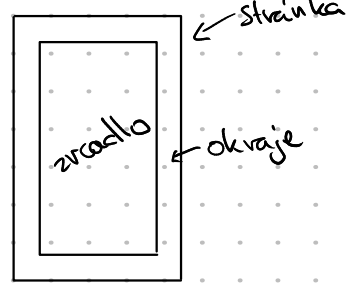
Vertikální pakování

- funguje stejně až na spočtení rozměrů
- šířka - maximum
- výška a hloubka
 - součet výšek a hloubek
 - jak rozdělit mezi výšku a hloubku?
 - pokud poslední prvek je box, pak hloubka je z posledního boxu
 - jinak hloubka 0

Jednotky

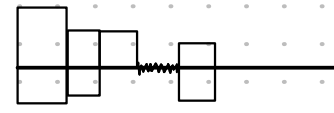
- mm, cm, in, pt, bp
- všechno se převádí na sp
- em, ex - závisí na šířce M
resp. výšce x | tex
- syntax glue: <...> plus <...> minus <...>
- doplňují se jednotky nebo nekonečna - fil, fill, filll

in=72,27pt
in=72bp
ε



umožňuje alg. přebíhovat
|hfuzz=0.1pt
|hbadness - max. akcept. badness
|showthe|badness

50% - 13
100% - 100
200% - 800



|vbox - alignuje poslední účarí
|vtop - || - ale první ← výška odpovídá prvnímu řádku, zbytek je hloubka

omezeno |maxdepth a |boxmaxdepth

Vznik horizontálního seznamu

- box-like
 - písmenko
 - ligatura (slitek) (\mathbb{f} , \mathbb{F} , \mathbb{fl}), pomlčky, vozovky
 - \hbox , \lvbox
 - akcenty - jenom jeden, ale z libovolného znaku
- linky
 - \lrvule
- discretionary - explicitně nebo alg. na dělení slov
- kern
 - \lkern
 - automaticky - dle metricky fontů - $A/v \rightarrow Av$
 - italská korekce - při přechodu z italicu na normální text - \l
- glue

• \hskip

plain TeX

• \hfil - \hskip \Oplus \fil

• \hss - nulová, smí se libovolně stáhnout i vztahnout

• \quad - 1em, \qqquad - 2em

• mezislovní mezery

• leaders

- penalta

• explicitně - \penalty , \nobreak , \break , \allowbreak

- vertikální materiál

• \lradjust - vypadne ven z h. seznamu do v. seznamu

$\text{\line}\{...\} = \text{\hbox}$ to \hsize

$\text{\leftline}\{x\} = \text{\line}\{x\} \text{\hss}$, center, right

$\text{\line}\{\text{\hfil}\text{\hfil}x\text{\hfil}\}$

$\text{\rlap}\{x\} = \text{\hbox}$ to $\text{\Oplus}\{x\} \text{\hss}$

$\text{\smash}\{...\} =$ výška a hloubka je nulová

$\text{\phantom}\{...\} =$ bílé místo velikosti obsahu

plain TeX

← 10000

← -10000

→ 0

fontdimen

- 1) sklon
- 2) std. mezera
- 3) roztažnost
- 4) sřtačitelnost
- 5) em
- 6) ex
- 7) extra mezera

$\backslash sfcode <kód\ znaku> = <číslo>$

$\backslash spacefactor$

znak: $sf \leftarrow sfcode$

ale: pokud $sf < 1000 < sfcode$
 $sf \leftarrow 1000$

box/linka: $sf \leftarrow 1000$

mezera: velikost = $fd2 + (fd7$ pokud $sf \geq 2000)$

roztažnost = $fd3 \cdot \frac{sf}{1000}$

sřstítelnost = $fd4 \cdot \frac{1000}{sf}$

$\backslash (x) spaceskip = glue$

Plainové sf codes

$\backslash nonfrenchspacing$ (jano v brit. imr.)

A-Z : 999

a-z : 1000

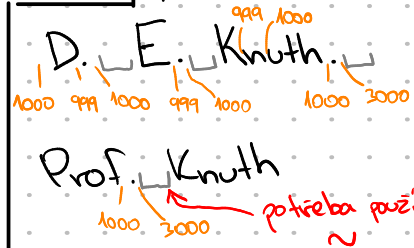
! ? : 3000

, : 1250

() : 0

{} : 2000

! : 1500



Dělení slov

① $\backslash discretionary$

② $\backslash hyphenation$ {nej-vět-ší} - výjimky, když to ③ nevládne, ! - - ekv. automatické pomůcky

③ patterns *rozhraní slova*

• ne 1

→ liché = smí se lámat, sudé nesmí se lámat

↳ hledá se maximum na každém místě z každého patternu

→ součásti initEXu, nedá se měnit za běhu

$\backslash left / right hyphenmin$ ← nesmí být menší než určitá hodnota v patternu, jinak nebudou patterny fungovat

$\backslash language = \backslash Czech$

„sřzení na praporek“ =

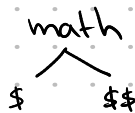
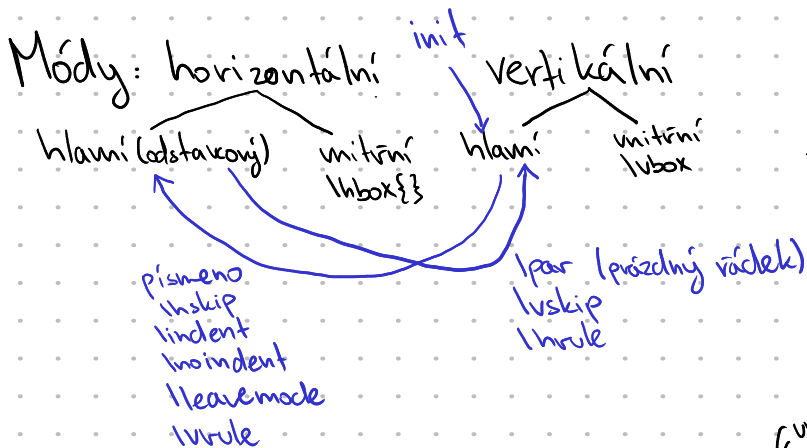
- $\backslash ligatur$ vznikají kouzelná discretionary: $\backslash f - \backslash fi - \backslash ffi$ efficient

TEX má tabulku překladač znaků zeřt na unicode - kopírování z PDF

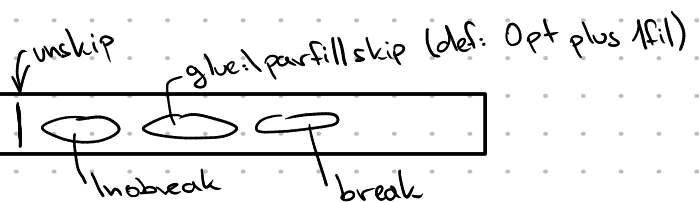
- když je ve slově hyphenchar není -li → | - | - | → vzniká explicitní discretionary => neláme se podle ② a ③

• csplain navrhuje spojovníky jiným stejným znakem

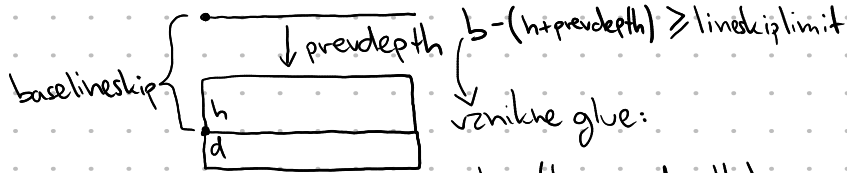
když chceme sřzet česky: csplain, $\backslash chyp$



$\backslash indent$
 $\backslash parindent = dimen$



predepth ↙ na začátku - 1000pt
a. po rule



! baselineskip < glue >

! lineskip < glue >

! lineskip limit < dimen >

vznikne glue:

$b - (h + \text{predepth})$

jinak se přidá lineskip

Zalomení stránek

- vertikální seznam

• box

• whatsit

• mark

• insertion

• glob of glue

• kern

• penalty

- zalomení

• nejlepší dostupná cena zlomu

• p if $(b < \infty)$ & $p \leq -10\,000$ & $q < 10\,000$

• $b + p + q$ if $(b < 10\,000)$ & $-10\,000 < p < 10\,000$ & $q < 10\,000$

• $10\,000$ if $(b = 10\,000)$ & $-10\,000 < p < 10\,000$ & $q < 10\,000$

• ∞ if $(b = \infty \vee q \leq 10\,000)$ & $p < 10\,000$

• když najdu ∞ , zalomím v nejlepším dostupném místě

- inserty

• !count < 0-255 >

• !dimen < 0-255 >

• !skip < 0-255 >

• !box < 0-255 >

• !advance < reg > by < value >

• !multiply < reg > by < value >

• !divide < reg > by < value >

Pr. $\text{lcount1} = 50$

$\text{ldimen2} = \text{lcount1pt}$

$\text{ldivide lcount1 by 8}$

$\text{lskip2} = -10\text{pt plus lcount1fil minus dimen2}$

$\text{lmultiply lskip2 by -lcount1}$

$\text{ldivide lskip2 by ldimen2}$

$\text{lcount6} = \text{lskip2}$

$\text{lskip1} = 0,5 \text{ldimen2 plus lskip2 minus lcount lcount1fil}$

$\text{lmultiply lskip2 by lskip1}$

$\text{advance lskip1 by -lskip2}$

$\text{rsplit } N \text{ to } \langle \text{dimen} \rangle$ - vysází vbox z reg. N , do dimen , zbytek zůstane v N

poze pro zabírací alg.

(output routine - čísla stránek, plavací objekty)

Inserty

- poznámky pod čarou (i s čarou), poznámky v okrajích, obrázky

- 255 tříd insertů

• $\text{lbox } N$ - materiál předán loutput

• $\text{ldimen } N$ - prostor, který smí zabrat na stránce

• $\text{lskip } N$ - vert. mezera, pokud se insert objeví

• $\text{lcount } N$ - % kolik zabere na stránce oproti vysázení - dvoosloopové 1... - 500

• $\text{linsert } N \{ \text{vert. materiál} \}$ - z h. materiálu vypadne pod řádek

• $\text{lnewinsert } \{xyz\}$ - naobkluje vše

! Insert nikdy neopustí svůj box

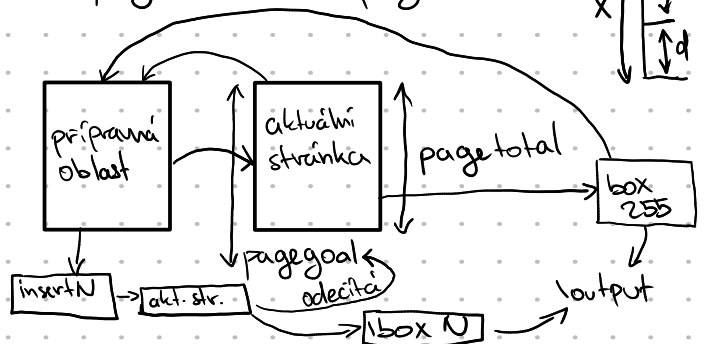
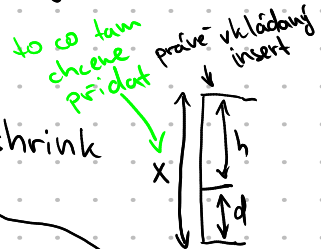
- algoritmus

• $g = \text{pagegoal}$, $t = \text{page total}$, $q = \text{insertpenalties}$, $d = \text{pagedepth}$, $z = \text{pageshrink}$

$f = \text{lcount } N / 1000$, $\text{lskip } N = w \oplus ? \ominus ?$

1) $g = h f + w$ *to, co už tam bylo*

2) pokud lámu, $q += \text{floating penalty} \rightarrow \text{konec}$



3) vejde se bez lámaní? $\lfloor \text{height } N + x \leq \lfloor \text{dimen } N$

$$x_f > 0 : t + d + x_f - z \leq g$$

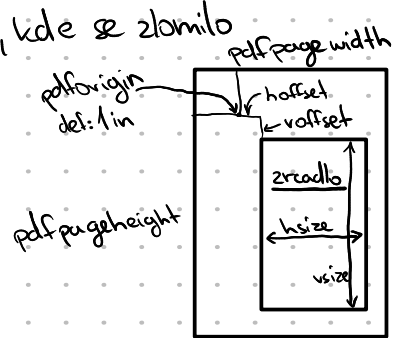
Pokud ano $\rightarrow g = x_f \rightarrow \text{konec}$

4) $\min(g - (t + d), \lfloor \text{dimen } N - h) = v$

$\lfloor \text{split to } v$

u - přirozená výška, r - penalta místa, kde se zlomilo

$$g = u \cdot f \quad q = +r$$



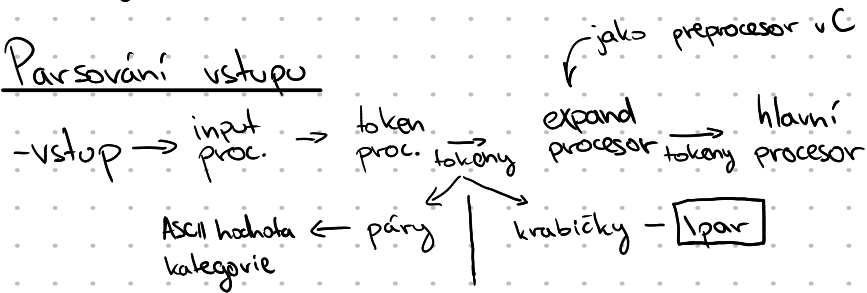
Output rutina

- vstupy: box 255, box N, marks, layout penalty
- výstupy: $\lfloor \text{shipout}$, vert. materiál zřít do připravené oblasti

Plain output

- $\lfloor \text{headline}$, $\lfloor \text{footline} = \{ \dots \}$
- $\lfloor \text{folio}$ - číslo stránky - $\langle 0 \Rightarrow$ říms. č.
- $\lfloor \text{footnote} \{ \text{značka} \} \{ \text{text} \}$
- $\lfloor \text{topinsert} \dots$, $\lfloor \text{endinsert}$, $\lfloor \text{midinsert}$, $\lfloor \text{pageinsert}$
- $\lfloor \text{eject}$, $\lfloor \text{supereject}$
- $\lfloor \text{raggedbottom}$

Parsování vstupu



- celý proces iniciuje hlavní procesor
- požádá expand o exp. nebo neexp. token
- expand žádá token...

input proc: odstraní mezery na konci řádku, přidá endlchar (CR-13)

token proc

- $\lfloor \text{catcode}$

- 3 stavy - N, M, S ← skipping spaces

- kat. 7 - ^ - po dvou takových znacích - buď dvě hexové číslice - kód znaku

- kat. 0-1 - nebo \oplus s 40 (jako Ctrl v term.)

• pokud není kat. 11 (písmeno) - krabička s tím znkem

• jinak krabička se všemi kat. 11 \rightarrow S

0		8 -
1	{	9 ignorováno
2	}	10 mezera
3	\$	11 písmeno
4	&	12 ostatní
5	newl.	13 aktivní (N, M)
6	#	14 %
7	^	15 nepovoleno

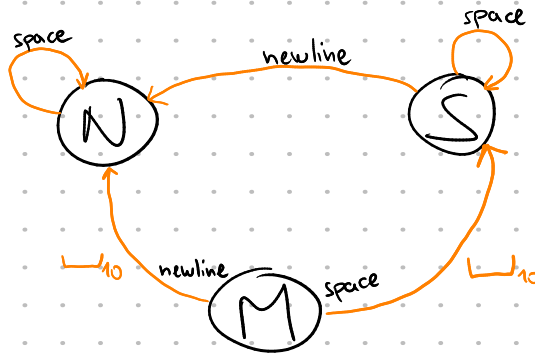
- kat. 1, 2, 3, 4, 6, 7, 8, 11, 12, 13 → C_t → (M)

catcode 127 = 9
Jak napsat číslo
* - číslo znaku
"7f - hex
'177 - oct.
'\w

- kat. 9 - nic, 15 - nic + err.

- kat. 6 (N) (S) → nic (M) → L₁₀ → (S)

- kat. 14 - % → řekneme inputu at' skipne zbytek řádku



expand procesor

- expandujeme makro (pořád dokola) - \def \makro{...} \makro

- argumenty #1

- dva seznamy
 - parametry - #1
 - náhrady - 1, 2
 ← omezený, neomezený?

\def \a #1{(#1)}

\def \b{1b}

\b 123 → \a 123 → (1) 23

Hlavní procesor:

\let \a = \b ← uložení významu tokenu

\gdef → \global \def

\edef - expanduje makro už při definici

\xdef → \global \edef

\long\def - makra nesmí obsahovat \par - s tímto ano

\outer\def - lze použít pouze mimo {}

Expand procesor:

\the - expanze registru
 \noexpand - neexpanduje následující token (jednou)

\string \cs - vzniknou znaky
 _12 C_12 S_12

\csname ... \endcsname

vytvoří krabičku
 - krabičky s názvem mimo {}
 - asociativní pole

\number, \romannumeral

\meaning

\expandafter AB → A exp(B)

\ea \let \csname ... \endcsname = \b

\ea \a \csname ... \endcsname ^{odkládají se}

\ea A \ea B C → A B exp(C)

\ea \ea \ea A \ea B C → \ea AB exp(C) → exp(B) exp(C)

`\uppercase{...}` `\uccode X=Y`

`\uppercase{\romannumeral42}` ← pokusi se vdelet upper na 42

`\lea\uppercase\lea{\roman...}` → `\up{xlii}` (první lea nemusí být protože wtf)

Podmínky

`\if`, `\else`, `\fi`, `\or`

- končí expanze když narazíme na true

- když narazíme na else a neexpandujeme, začne (a jsme v nejnižší úrovni if (zbytek se expandoval))

Vyhodnocení podmínky

`\if A B` ← stejný znak nebo dojí krabicka

`\ifcat AB` ← stejná kat.

`\ifx AB` ← stejný meaning

`\ifvoid` `\ifvbox` `\ifhbox`

`\ifnum <cis> > <cis>`

`\ifdim`

`\ifodd`

`\ifvhm()mode`

`\ifinner`

`\ifeof`

`\iftrue`, `\iffalse`

`\ifcase`

`\ifx|xyz|undefined`

`\ifcsname... \endcsname`

Cykly (ocasatá rekurze)

- např.: `\scan#1{%`

`\ifx#1:`

`\let\next=\relax`

`\else`

`<pozij #1>`

`\let\next=\scan`

`\fi`

`\next`

`\loop A... \if... B... \repeat` (↑ if)

\after... \every...

`\everypar={}`

`\aftergroup <neexp>` ← vloží se na konec grupy

`\afterassignment <neexp>` ← vloží se za assignment

```
\def\gcdnum{\afterassignment\gcdx\l a=}
\def\gcdx{\afterassignment\gcdy\l b=}
\def\gcdy{\gcd\l a}\l b}
```

`\futurelet <cs> <tok1> <tok2>`

```
\obr{soubor.png}
\obr[0.5\hsize]{soubor.png}
\def\obr{\futurelet\tmp\obrx}
\def\obrx{%
  \if\tmp{\expandafter\obrarg
  \else\expandafter\obrplain}
  \fi
}
\def\obrarg[#1]#2{}
\def\obrplain{#1}
```

`\newif|if X` (see plain.tex)

`|Xtrue`

`|Xfalse`

`\begin{group} ... \end{group}`

`\bgroup ... \egroup`

`\unskip`

`\toksN \newtoks\l \l = {<neexp>}`

`\the\l`

spouští se input a token

`\newread\l \openin\l = soubor`

`\read\l to lcs`

`\closein\l`

`\newwrite\l \openout\l`

`\write\l{}`

`\closeout`

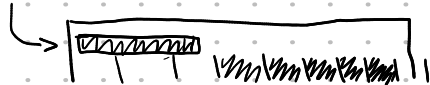
deje se až v shipoutu

`\immediate` - hned

(c/x)

`\leaders <box or rule> <skip>`

`\hrulefill, \vrulefill, \dotfill`



C -> vyplní se a vycentruje

X -> vyplní se a rozloží rovnoměrně

Tabulky

`\tabskip` se nastavuje mezi sloupky

`\halign[to size]{# & # & # \cr`
... & ... & ... \cr}

`(((\lvalign)))`

- doplní obsah sloupečků za # v hlavičce
- spočítá se maximum z šířek buněk v sloupcu
- na okraje buněk se hodí `\tabskip`
- `\lvalign` vypadne pod rádek
- v preambuli se expanduje pouze `\span <token>` a `\tabskip`
- `\omit` - nepoužij šablonu
- `\span` (uvnitř tabulky) - spoj s následující buňkou
- expanze v buňce
 - expanduje se dokud to jde a zjistí se přítomnost `\omit, \cr`

• dějou se trochu záry, část obsahu buňky se expanduje před prefixem

Příklad:

$\backslash rule \hskip 0.5em \strut \rightarrow \dots$

```
\halign{\hfil#. \hskip = 1em & \# \hfil & \hfil \# \hskip = 0 sp \cr
  \omit Pořadí \hfil & Jméno & Body \cr
  \dots
  17 & Pokusný králik & 137 \cr
}
```

Matematika

- spousta cool symbolů, které si Honza nezapsal, protože dělal něco jiného

Styly

Display, Text, Script, SS

$\backslash abovewithdelims : D \rightarrow \frac{T}{T'}, T \rightarrow \frac{S}{S'}, S \rightarrow \frac{SS}{SS'}$

$x^1 : D_{S'}^S \quad T_{S'}^S \quad S_{SS'}^{SS} \quad SS_{SS'}^{SS}$

$\Gamma : D \rightarrow D' \dots$

Font families: D, T: rm 10, S: 7, SS: 5

$\backslash displaystyle ? \rightarrow D$

$\backslash textstyle ? \rightarrow T$

$\backslash scriptstyle ? \rightarrow S$

$\backslash scriptscriptstyle ? \rightarrow SS$

$\backslash math choice \{D\}\{T\}\{S\}\{SS\}$

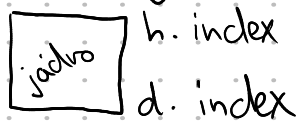
$\backslash math palette \{t \# 1 \rightarrow \backslash math choice \{t \backslash displaystyle \# 1\}$

Matematický seznam

- atomy
- dočasné značky (`\left`)
- glue, mathglue - math unit - mu - relativní jednotka (nonscript)
- kern, mathkern
- přepínače stylu
- mathchoice
- horiz. materiál - linky, penalty, discr.
- vert. materiál - mark, vadjust, inserty
- whatsit
- above withdelims

vypadne
ven

Atomy

 h. index
d. index

Typy:

0	Ord	x	\aleph	\hbox	\}
1	Op		\sum		
2	Bin		+ -		
3	Rel		< =		
4	Open		([
5	Close)]		
6	Punct		,		
7	Inner		\left \right		
8	Over		\overline{xxx}		
9	Under		\underline{xxx}		
10	Acc		\hat{x}		
11	Rad		$\sqrt{\quad}$		
12	Vcen		\vcenter		

`\mathcode` kód znaku = "TFXY"
 type ← T, font family ← F, pozice ve fontu ← XY

T=7: rodina (fam) pokud {0...15}
 jinak F

118000 - active

Velikosti závorek

`(((... (` $\left. \begin{array}{l} \text{stacksize} \\ \leftarrow \rightarrow \end{array} \right\}$

V použití s `\left`, jinak nejmenší

`\delcode` kód znaku = "F₁X₁Y₁F₂X₂Y₂"

`\delimiter` "TF₁X₁Y₁F₂X₂Y₂"

`\radical` $\sqrt{\quad}$

Mezery

`\L` \emptyset

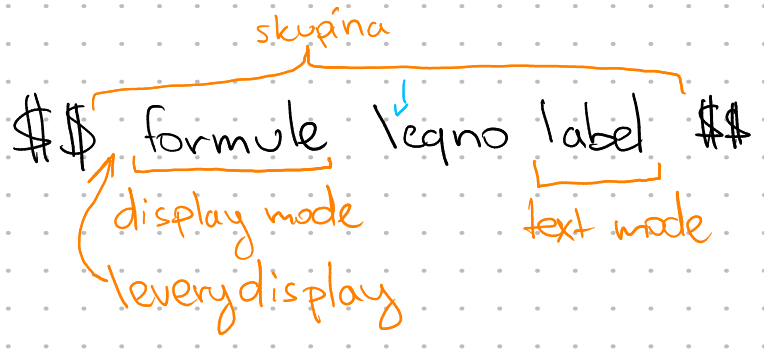
`\L` ~ textově

`\mu` = $\frac{1}{18}$ em pro fam=0

`|_` $\left| \right.$ `|>` $\left| \right.$ `|j` $\left| \right.$
 thin, med, thick, thin

Lámání

- pouze na explicitní penaltě nebo `\binoppenalty` `\lpenalty`



Usazeni matematiky

`formule`

- bez značky - centruje se a smrštuje

`formule` $\geq |2n|$

`\left/right skip` X

`formule` $\geq \frac{1}{2}em$

`\hang...`
`\parshape` } ~~\$\$~~ ~~\$\$~~

`glue formule` $\geq 1em$

`\displaywidth`
`\displayindent`

`\prelisplaypenalty`
`\abovedisplayskip` (short)
`\belowdisplayskip` (short)
`\postdisplaypenalty`
značky nové velikosti
! nezačíná nový odstavec

`$$ \halign{...} $$`

`\eqalign{... &= ... \cr ...}`

`\eqalignno{}`

Fonty

- r - roman
- ti - text italic
- mi - math italic
- b - bold
- bx - bold extended ← `\bf`
- tt - monospace
- sl - slanted
- ss - sans serif
- u - unslanted italic

`\font\cs = cmr10` at 14pt
rodina design size scaled 1400
`\cs`

`\textfont <fam> = \cs`

`\fontdimen <̄> \cs (\font)`

`\fontname \font`

`\magstep 1 → 1200`

Rodokmen rodiny TEXových

