

Zpravodaj Československého sdružení uživatelů TeXu

Miroslav Dont

LAmSTeXové tabulky

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 2 (1992), No. 1, 6–21

Persistent URL: <http://dml.cz/dmlcz/149609>

Terms of use:

© Československé sdružení uživatelů TeXu, 1992

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

za jakkoli krátký příspěvek do tohoto zpravodajství vděční. Může jím být vaše zkušenost s odesíláním recenzí pro **MR** emailem, s prací v jiných sítích, s \TeX em na pracovních stanicích, atp. Mimochodem, časopisů připravovaných pomocí \TeX u u nás přibývá, ale informace o nich nám nedocházejí.

Těším se, že od vás dostaneme vyplněný list co nejdříve, ne však později než do 5. května 1992. Chcete vědět proč? Inu tak: bylo by dobré se sejít např. ten den k oslavě malého jubilea.

Jiří Veselý

\LaTeX - \TeX ové tabulky

V minulém čísle \TeX bulletinu jsem se zmínil o \LaTeX - \LaTeX u — rozšíření \LaTeX u o některé možnosti \LaTeX - \TeX u (a o NFSS — mimo jiné). V tomto článku jsem se také v krátkosti zmínil o existenci \LaTeX - \TeX u, což je, zhruba řečeno, rozšíření \LaTeX - \TeX u o některé vlastnosti \LaTeX u (vše provedené úplně jiným způsobem než v \LaTeX u). Tato poznámka není věnována \LaTeX - \TeX u, ale pouze jedné její části — tvorbě tabulek. K samotnému \LaTeX - \TeX u bych poznamenal, že původně šlo o komerční výrobek (autor M. Spivak), který byl později uvolněn pro public domain a zdrojové texty \LaTeX - \TeX u jsou nyní volně dostupné v různých archivech pomocí e-mailu (volně dostupný není ovšem manuál).

Výroba tabulek v textu obecně nepatří mezi jednoduché záležitosti, zvláště je-li tabulka větší a má-li složitější strukturu. Zatímco v \LaTeX u existuje okolí `tabular`, které umožňuje poměrně snadnou tvorbu tabulek (nerámovaných i rámovaných), uživatel `PLAIN`u nebo \LaTeX - \TeX u je odkázán sám na sebe. Jak jsem již poznamenal, jedna z vlastností, kterou \LaTeX - \TeX u přebírá od \LaTeX u, je možnost snadné tvorby tabulek. Za pozoruhodnou vlastnost této tvorby tabulek v \LaTeX - \TeX u je ale možné označit skutečnost, že tabulky vytvořené v \LaTeX - \TeX u lze použít i v rámci souborů psaných v `PLAIN`u nebo \LaTeX - \TeX u nebo dokonce v \LaTeX u. Např. tento článek byl původně psán v \LaTeX u, ale všechny

ukázky tabulek byly vyráběny v $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u (říkám *původně*, neboť nevím, na jaký tvar bude soubor upraven těsně před tiskem; článek o $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ u v minulém čísle byl také původně psán v $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ u, ale před tiskem byl převeden do $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u, takže např. diagram na str. 17 vyšel poněkud jinak, než vyjít měl). O této vlastnosti tvorby tabulek $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ em se zmíním nejdříve. Je to ostatně vlastnost, pro kterou může být $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ (daná část $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u) zajímavý i pro ty uživatele $\mathcal{T}\mathcal{E}\mathcal{X}$ u, kteří jinak $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ používat v úmyslu nemají.

Zatímco v $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ u jsou makra pro tvorbu tabulek nedílnou součástí $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ u (soubor `latex.tex`), v $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u naopak tato makra tvoří zcela samostatnou součást. M. Spivak říká, že těchto instrukcí je tak mnoho a zabírají příliš místa v paměti počítače, že se je rozhodl osamostatnit (velikostí srovnává soubor obsahující makra pro tvorbu tabulek s částí $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u obsahující makra pro tvorbu komutativních diagramů). Hlavní soubor obsahující $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ je `lamstex.tex`; podle toho, co jsem již naznačil, tento soubor neobsahuje definice maker pro tvorbu tabulek. Tato makra jsou definována v souboru `lamstbl.tex`. Zatímco běžný soubor psaný v $\mathcal{L}\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u bude začínat např. řádky

```
\input lamstex
\docstyle{book}
\document
...
```

(pokud si nevyrobíme příslušný formátový soubor `.fmt`), bude soubor obsahující tabulky začínat podobně řádky

```
\input lamstbl
\tblstyle{book}
\document
...
```

Použití programu dvipaste

Především bych zdůraznil, že soubory obsahující text a obsahující tabulky jsou dva samostatné soubory, text a tabulky se „ $\mathcal{T}\mathcal{E}\mathcal{X}$ ují“ zvlášť a při volání souboru `lamstbl.tex` se *nevolá* soubor `lamstex.tex`.

Předpokládejme na chvíli, že jsme vytvořili dva soubory `text.tex` a `tabulky.tex`. První z nich obsahuje řádek

```
\input lamstex
```

a dále nějaký text. Druhý soubor obsahuje řádek

```
\input lamstbl
```

a dále nějaké řádky

```
\table
...
\endtable
\table
...
\endtable
...
```

kde jsou „napsány tabulky“. Jestliže tyto soubory „projedeme \TeX em“ (dále budu říkat „přeložíme“), dostaneme dva `.dvi` soubory, z nichž první obsahuje pouze text a druhý pouze tabulky. Nyní je potřeba ovšem tyto dva `.dvi` soubory spojit a tabulky „vsunout“ do textu. To umožní program `dvipaste` (autorem je opět M. Spivak a jeho zdrojový text v jazyku C je k dispozici spolu s \LaTeX -em) a malý soubor maker `dvipaste.tex` — tento soubor je automaticky volán soubory `lamstex.tex` a `lamstbl.tex`. Popíšme trochu blíže, o co se zde jedná.

Při překladu souboru `tabulky.tex` bude každá tabulka (ohrazená příkazy `\table` a `\endtable`) umístěna na samostatnou stránku a pod tabulkou bude vypsán název souboru a pořadí tabulky v tomto souboru. Mimo soubor `tabulky.dvi` bude vytvořen krátký soubor `tabulky.dat`, který obsahuje informace o rozměrech tabulek generovaných souborem `tabulky.tex`. Chceme-li nyní některou z těchto tabulek umístit do „hlavního“ textu, učiníme tak pomocí příkazu `\paste`, který má syntaxi `\paste{<soubor>}{<číslo>}`. Jestliže tedy např. chceme umístit tabulku, která byla vytvořena v souboru `tabulky.tex` v pořadí jako pátá tabulka, použijeme příkaz `\paste{tabulky}{5}` (přičemž použijeme další příkazy, které určí, jakým způsobem bude tabulka v textu umístěna). Po překladu se však samotná tabulka v textu neobjeví, ale bude pouze

vynecháno příslušné místo podle velikosti tabulky (v době překladu již musí existovat soubor `tabulky.dat`, tj. soubor `tabulky.tex` musí být přeložen dříve než soubor `text.tex`). Poslední fáze je samotné vsunutí tabulky ze souboru `tabulky.dvi` do souboru `text.dvi` programem `dvipaste`, což se provede jednoduše příkazem (z DOSu) `dvipaste text`. Příkaz `\paste`, který jsme použili v souboru `text.tex`, je definován v souboru `dvipaste.tex` (který, jak jsem již poznamenal, je součástí hlavního \LaTeX -ového souboru `lamstex.tex`). Přitom však použití tohoto příkazu není vázáno na použití pouze v rámci \LaTeX - \TeX u. Chceme-li tedy použít příkaz `\paste` na vsunutí tabulky třeba do \LaTeX -ového souboru, stačí, když tento soubor začíná např. řádky

```
\input dvipaste
\documentstyle[ ... ]{ ... }
```

Samotné umístění tabulky může potom vypadat třeba takto:

```
\begin{center}
\mbox{\paste{tabulky}{5}}
\end{center}
```

Použití programu `dvipaste` je ovšem stejné.

Nakonec bych poznamenal, že použití Spivakova programu `dvipaste` a souboru `dvipaste.tex` není ani zdaleka omezeno pouze na vsunování tabulek vyrobených v \LaTeX - \TeX u, ale můžeme takto vsunovat i jiné objekty, např. obrázky vyrobené v \PCTeX u. V souboru `dvipaste.tex` je definován příkaz `\sendout`, který to umožňuje. Použití je následující. Předpokládejme tedy, že chceme mít samostatný soubor `obraz.tex`, který bude obsahovat pouze obrázky psané v \PCTeX u. Soubor `obraz.tex` bude začínat řádky

```
\input dvipaste
\input pictex
...
```

a dále bude obsahovat „popis“ obrázků. Přitom každý obrázek bude opět umístěn na samostatné stránce příkazem `\sendout` a to tak, že obrázek nejprve uložíme do samostatného `\boxu` (nebo definujeme jako příkaz)

a potom tento `\box` „odešleme“ příkazem `\sendout`. Celý postup může být následující:

```
\setbox0=\hbox{\picture
...      % Samotný text vytvářející obrázek
\endpicture }
\sendout{\box0}
```

(při výrobě tabulek v \LaTeX je ve skutečnosti příkaz `\sendout` volán příkazem `\endtable`). Umístění obrázku v hlavním textu je potom stejné jako umístění tabulky. Kdo někdy použil \PCTeX pro výrobu obrázků, jistě vidí, že takto lze ušetřit spoustu času při editaci textu, neboť není třeba při každé opravě v textu vždy znova překládat obrázky.

Tabulky

Vraťme se ale konečně k samotným tabulkám v \LaTeX . Zmíním se pouze o některých základních možnostech. Výběr bude víceméně náhodný.

Jak jsem již řekl, příkaz, který vždy začíná tabulku, je `\table`. I další příkaz je určen jednoznačně a je to příkaz `\tss`. Zde bych poznamenal, že Špivák často používá pro názvy příkazů různé zkratky, které nemusí být vždy zcela průhledné. Tak např. `\tss` je zkratka pro „Top Specification“ a jedná se o příkaz, kterým je definován počet sloupců, mezery mezi sloupci, zarovnávání v jednotlivých sloupcích, typ sloupce atd. (v \LaTeX jsou tyto údaje obsaženy v parametru okolí `\tabular`). Ukončení příkazu `\tss` je potom pomocí `\`. Formální rámec tabulky tedy bude mít tvar

```
\table
\tss ... \
...
\endtable
```

Základní typy sloupců jsou určeny příkazy `\l` (zarovnávání na levý okraj), `\c` (středovaný), `\r` (zarovnání na pravý okraj). Na rozdíl od tabulek v \LaTeX mezery mezi sloupci tabulek v \LaTeX si vždy uživatel určuje sám příkazem `\g<rozměr>` (zde `\g` znamená „gap“). Symbol `&` značí přechod do nového sloupce (používá se v rámci `\tss` i v samotném textu tabulky). Tak např. napíšeme-li

```
\table
\tss \l \g(10pt) & \g(10pt) \c \g(10pt) &
      \g(10pt) \r \\\
```

bude následovat tabulka o třech sloupcích, z nichž první je zarovnaný na levý okraj, druhý je středovaný a třetí je zarovnaný na pravý okraj. Přitom mezi sloupci bude vždy mezera 20pt (zde bychom mohli místo `\g(10pt) & \g(10pt)` napsat např. `\g(20pt)&`, ale první způsob bude nutné použít v případě, že mezi sloupci bude svislá čára, která potom bude oddělena 10pt od obou sloupců). Tato jednoduchá tabulka

První řádek	Druhá část prvního řádku	Třetí část
Druhý řádek	Druhá část druhého řádku	Třetí část druhého řádku
Třetí řádek	Druhá část	Třetí část

může být tedy např. napsána takto (příkaz `\\` v textu tabulky značí přechod na nový řádek)

```
\table
\tss \l \g(10pt) & \g(10pt) \c \g(10pt) & \g(10pt) \r \\\
První řádek & Druhá část prvního řádku & Třetí část \\\
Druhý řádek & Druhá část druhého řádku & Třetí část
druhého řádku \\\
Třetí řádek & Druhá část & Třetí část
\endtable
```

Tuto tabulku můžeme nyní „vylepšit“ např. následujícím způsobem:

To je středovaný nápis	Druhý a třetí sloupec	
První řádek	Druhá část prvního řádku	Třetí část
Druhý řádek	Druhá část druhého řádku	Třetí část druhého řádku
Třetí řádek	Druhá část	Třetí část

Tato tabulka byla zapsána příkazy

```
\boxedtables
\sides
\table
\tss \g(10pt) "\l \g(9pt) & \g(9pt) \c \g(9pt) &
      \g(9pt) \r \g(10pt) \\\
\hl
```

```

\c\bf To je středovaný nápis |
\xx2\c\bf Druhý a třetí sloupec \\\
\hl
První řádek | Druhá část prvního řádku | Třetí část \\\
\hl
Druhý řádek | Druhá část druhého řádku | Třetí část
druhého řádku \\\
\hl
Třetí řádek | Druhá část | Třetí část\\
\hl
\endtable

```

Všimneme si zde několika věcí, především nových příkazů. Příkaz `\boxedtables` způsobí, že mezerování mezi řádky této tabulky, která je rámovaná, bude jiné než u tabulky, která rámovaná není (přitom je ovšem možné vzdálenosti mezi řádky měnit i jinými způsoby a to dokonce pro každý řádek zvlášť). Příkaz `\hl` (horizontal line) vyrábí vodorovné čáry v tabulce, příkaz `|` čáry svislé (příkaz `|` nahrazuje v tomto případě příkaz `&`). Přitom na okrajích tabulky nemusíme příkaz `|` psát, neboť jsme před tabulkou napsali příkaz `\sides`. Všimneme si, že zde jsme příkaz `\g` použili (na rozdíl od předchozí tabulky) i přímo za `\tss` a před `\\`. Je to proto, aby okrajové svislé čáry byly oddělené od textu tabulky (o uvedenou vzdálenost). Příkazy `\xx2\c` způsobí, že nápis „Druhý a třetí sloupec“ bude napsán přes dva sloupce a zároveň středován (vzhledem k těmto dvěma sloupcům). Zajímavý je příkaz `"\l` v řádku `\tss`. Kdybychom zde použili pouze `\l`, byly by poslední tři řádky stále zarovnané na levý okraj řádku prvního (mluvím stále o prvním sloupci). Při použití `"\l` jsou řádky opět zarovnané na levý okraj, ale první řádka je středována (neboť je zde příkaz `\c`) vzhledem k zbylým řádkům *jako celku* (mimochodem, v tabulkách \LaTeX u není něco takového tak úplně snadné).

Mimo příkazy `\l`, `\c`, `\r` je možné v řádku `\tss` použít též příkazy `\L`, `\C`, `\R` s jedním parametrem. Např. příkaz `\L{\bf}` (v řádku `\tss`) způsobí, že řádky příslušného sloupce budou zarovnané na levý okraj a budou tištěny tučně. Další typ sloupce je možné deklarovat příkazy `\p{<rozměr>}` nebo `\P{<rozměr>}{...}`. Oba tyto příkazy umožní psát text do odstavce uvedeného rozměru. Druhým parametrem příkazu `\P` je navíc možné určit typ písma v tomto odstavci podobně jako příkazy `\L`, `\C`, `\R`.

Jednotlivé sloupce v jednom řádku je přitom možné vůči sobě vertikálně různě zarovnávat. Možnosti jsou `\t` (na vrchol), `\m` (na střed), `\b` (na dolní okraj). Tyto příkazy se zapisují do zvláštního řádku, který začíná příkazem `\sss` (Side SpecificationS) a který je umístěn za řádek `\tss` (a který je opět ukončen `\`).

Když např. příkazy `\A`, `\B` budou obsahovat text objevující se v prvním resp. druhém sloupci, pak tabulku

Ukázka krátkého odstavce uvnitř tabulky. Šířka tohoto odstavce je 42 mm	Zde je další ukázka krátkého odstavce uvnitř tabulky. Šířka tohoto odstavce je stejná jako šířka odstavce umístěného nalevo. Má však více řádků, aby bylo patrné zarovnávání textu v levé části.
Ukázka krátkého odstavce uvnitř tabulky. Šířka tohoto odstavce je 42 mm	Zde je další ukázka krátkého odstavce uvnitř tabulky. Šířka tohoto odstavce je stejná jako šířka odstavce umístěného nalevo. Má však více řádků, aby bylo patrné zarovnávání textu v levé části.
Ukázka krátkého odstavce uvnitř tabulky. Šířka tohoto odstavce je 42 mm	Zde je další ukázka krátkého odstavce uvnitř tabulky. Šířka tohoto odstavce je stejná jako šířka odstavce umístěného nalevo. Má však více řádků, aby bylo patrné zarovnávání textu v levé části.

je možné zapsat takto:

```
\sides
\table
\tss \g(10pt) \p(42mm) \g(10pt) & \g(10pt) \p(42mm)
      \g(10pt) \\\
\sss \t \m \b \\\
\hl
\ex(3pt)
\A | \B \\\
\ex(2pt) \hl \ex(3pt)
\A | \B \\\
```

```

\ex(2pt) \hl \ex(3pt)
\A | \B \\
\ex(2pt) \hl
\endtable

```

Mimo to, že je možné definovat sloupec jako odstavec, je také možné v rámci sloupce „do jednoho řádku napsat více řádků“ pomocí příkazů `\= a \+`, přičemž `\+` značí přechod na nový řádek uvnitř daného řádku. Tak např. tabulku

První řádek	Druhá část prvního řádku	Třetí část prvního řádku
Druhý řádek	Druhá část druhého řádku	Třetí část druhého řádku
Třetí řádek	Třetí část třetího řádku	Třetí část třetího řádku

zapišeme příkazy

```

\sides
\table
\tss \g(5pt) \l \g(10pt) & \g(10pt) \c \g(10pt) &
      \g(15pt) \r \g(5pt) \\
\sss \t \m \b \\
\hl
\ex(3pt)
\= První řádek \= | \= Druhá část\+prvního řádku\= |
\= Třetí část\+prvního\+řádku\= \\
\ex(2pt) \hl \ex(3pt)
\= Druhý řádek \= | \= Druhá část\+druhého řádku\= |
\= Třetí část\+druhého\+řádku\= \\
\ex(2pt) \hl \ex(3pt)
\= Třetí řádek \= | \= Třetí část\+třetího řádku\= |
\= Třetí část\+třetího\+řádku\= \\
\ex(2pt) \hl
\endtable

```

Příkaz `\ex(3pt)` zde zvětšuje mezerování řádků a zároveň o příslušnou délku „natahuje“ svislé čáry (pro mezerování mezi řádky lze např. též použít příkaz `\vs(...)`, který však neovlivní délku svislých linek — existují ale ještě další možnosti jak ovlivnit mezerování řádek).

Podobně jako je možné psát nápisy přes dva nebo více sloupců, je možné mít části tabulky, které jsou „nataženy“ přes dva nebo více řádků. Místo příkazu `\xx<číslo>` použijeme příkaz `\yy<číslo>`. Tabulka

První dva řádky	Druhá část prvního řádku	Třetí část prvního řádku
	Druhá část druhého řádku	Druhé dva řádky
Třetí řádek	Druhá část třetího řádku	

je generována příkazy

```

\sides
\table
\tss \g(10pt) \c \g(10pt) & \g(10pt) \c \g(10pt) &
      \g(10pt) \c \g(10pt) \\
\hl \ex(3pt)
\yy2 \m
\= První\+dva\+řádky \= |
\= Druhá část\+prvního řádku \= |
\=Třetí část\+prvního\+řádku\= \\
\ex(2pt)
\hlyy
\ex(3pt)
| \= Druhá část\+druhého řádku \= |
\yy2 \m \= Druhé\+dva\+řádky \= \\
\ex(2pt)
\hlyy
\ex(4pt)
\= Třetí řádek \= | \= Druhá část\+třetího řádku \= | \\
\ex(2pt)
\hl

```

`\endtable`

Zde mimo již známé příkazy se objevuje příkaz `\hlyy`. Tento příkaz použijeme místo příkazu `\hl`, když chceme, aby vodorovná čára mezi první a druhou řádkou neprotnula zdvojený první sloupec (nápís „První dva řádky“). Na tomto místě jsme mohli místo příkazu `\hlyy` použít také příkaz `\hls ... \l`. Zmiňme se zde o příkazech ke generování vodorovných čar. Již dříve jsme použili příkaz `\hl` a před chvílí příkaz `\hlyy`. Další je příkaz `\hL(<rozměr>)`, kterým lze generovat vodorovnou čáru opět o délce šířky tabulky, ale o určené tloušťce (příkazem `\hl` bývá generována čára o tloušťce 0.4pt).

Tak třeba příkazy

```
\table
\tss \g(3pt) "\l \g(9pt) & \g(9pt) \c \g(9pt) &
      \g(9pt) \r \g(3pt) \l
\hl(1pt) \vs(4pt)
\c\bf To je středovaný nápis &
\xx2\c\bf Druhý a třetí sloupec \l
\vs(4pt)
\hL(.6pt) \vs(2pt) \hL(.6pt)
\vs(4pt)
První řádek & Druhá část prvního řádku & Třetí část \l
\vs(4pt) \hL(.6pt) \vs(4pt)
Druhý řádek & Druhá část druhého řádku & Třetí část
druhého řádku \l
\vs(4pt) \hL(.6pt) \vs(4pt)
Třetí řádek & Druhá část & Třetí část\l
\vs(4pt) \hL(1pt)
\endtable
```

lze generovat tabulku

To je středovaný nápis		Druhý a třetí sloupec	
První řádek	Druhá část prvního řádku	Třetí část	
Druhý řádek	Druhá část druhého řádku	Třetí část druhého řádku	
Třetí řádek	Druhá část	Třetí část	

Příkazy `\hl` a `\hL` generují linky na šířku tabulky. Obecnější je příkaz `\hls`, který umožňuje kontrolovat vodorovné čáry v každém sloupci zvlášť. Příkaz `\hls` (Horizontal LineS) bude mít nějaké parametry, které jsou pro jednotlivé sloupce oddělené symbolem `&` a příkaz je ukončen `\\`. Tak třeba příkazem `\hls .4pt & 1pt & 2pt \\` budou generovány čáry silné 0.4pt v prvním sloupci, 1pt ve druhém a 2pt ve třetím sloupci. Tabulka

<u>První řádek</u>	<u>Druhá část prvního řádku</u>	<u>Třetí část</u>
Druhý řádek	Druhá část druhého řádku	Třetí část druhého řádku
Třetí řádek	Druhá část	Třetí část

byla generována příkazy

```
\table
\tss \g(3pt) \l \g(10pt) & \g(10pt) \c \g(10pt) &
      \g(10pt) \r \g(3pt) \\
První řádek & Druhá část prvního řádku & Třetí část \\
\hls .4pt & 1pt & 2pt \\
Druhý řádek & Druhá část druhého řádku & Třetí část
druhého řádku \\
Třetí řádek & Druhá část & Třetí část \\
\hl
\endtable
```

Místo běžné tloušťky 0.4pt je možné krátce napsat `*` a pokud bude údaj o tloušťce vynechán, nebude v daném sloupci generovaná žádná čára (ekvivalentní údaj 0pt). Všimneme si, že v tabulce byly generovány čáry pouze na šířku sloupců, ale již nikoli přes mezery mezi sloupci. Chceme-li čáru „natahnout“ přes tyto mezery vlevo nebo vpravo, použijeme dodatečných parametrů `l`, `r` (ještě další parametry mohou tuto čáru naopak zase o přesně stanovenou délku zkrátit atd.). Např. při použití `\hls ... & *:l;r & ... \\` bude čára (o tloušťce 0.4pt) „natažena“ vlevo i vpravo přes mezery kolem tohoto sloupce; jelikož tato kombinace se používá často, lze `*:l;r` nahradit zkratkou `+`, tj. psát `\hls ... & + & ... \\` Tak třeba pomocí příkazů

```
\table
\tss \g(3pt) \l \rt4{\g(8pt)&\g(8pt)\c} \g(3pt) \\
```

```

\hl \vs(2pt)
& \xx2\c Úloha 1 & \xx2\c Úloha 2 \\
\vs(2pt)
\hls & + & *:1 & *:0pt;r & + \\
\vs(2pt)
$n$ & Metoda 1 & Metoda 2 & Metoda 3 & Metoda 4\\
\vs(2pt) \hl \vs(2pt)
3 & 5.490 & 5.490 & 4.235 & 5.490 \\
5 & 2.278 & 2.478 & 2.918 & 2.278 \\
7 & 3.329 & 2.908 & 2.701 & 2.584 \\
\vs(2pt) \hl
\endtable

```

dostaneme tabulku

n	Úloha 1		Úloha 2	
	Metoda 1	Metoda 2	Metoda 3	Metoda 4
3	5.490	5.490	4.235	5.490
5	2.278	2.478	2.918	2.278
7	3.329	2.908	2.701	2.584

Chceme-li měnit tloušťku svislých čar, můžeme použít příkaz `\l{<rozměr>}`. Použijeme-li tento příkaz v řádku `\tss` místo znaku `&`, bude svislá čára mezi příslušnými sloupci (generovaná příkazem `|`) mít vždy uvedenou tloušťku. Chceme-li měnit sílu svislé čáry pouze v některých řádcích, pak v řádku `\tss` použijeme normálně znak `&`, ale příkaz `\l{...}` použijeme v příslušném řádku místo `|`. K jemnému posunu čáry vpravo či vlevo lze použít příležitostný druhý argument v příkazu `\l{...}`. Tak třeba tabulku

To je středovaný nápis	Druhý a třetí sloupec	
První řádek	Druhá část prvního řádku	Třetí část
Druhý řádek	Druhá část druhého řádku	Třetí část druhého řádku
Třetí řádek	Druhá část	Třetí část

lze generovat příkazy

```

\sides
\table
\tss \g(5pt) "\l \g(9pt) & \g(9pt) \c \g(9pt) &
\g(9pt) \r \g(5pt) \\

```

```

\hl
\c\bf To je středovaný nápis |
\xx2\c\bf Druhý a třetí sloupec \\\
\hl
První řádek | Druhá část prvního řádku | Třetí část \\\
\hls + | *;2.6pt:+ & + \\\
Druhý řádek \|( *;2.6pt) Druhá část druhého řádku
\|(3pt) Třetí část druhého řádku \\\
\hls * | 3pt:+ & + \\\
Třetí řádek | Druhá část | Třetí část\\\
\hl
\endtable

```

Příkazy `\xx`, `\yy`, o kterých jsme mluvili dříve, je možné kombinovat a můžeme dostat třeba následující tabulku

Dvojitý řádek a sloupec		Třetí část prvního řádku
		Třetí část druhého řádku
Třetí řádek	Druhá část třetího řádku	Třetí část třetího řádku

Tato tabulka byla generována příkazy

```

\sides
\table
\tss \g(5pt) \c \g(10pt) & \g(10pt) \c \g(10pt) &
      \g(10pt) \c \g(5pt) \\\
\sss \t \m \b \\\
\hl
\exs(5pt) & | \\\
\xx2 \yy2 \m \c \bf
\= Dvojitý řádek\+a sloupec\= |
\= Třetí část\+prvního\+řádku\= \\\

```

```

\exs(5pt) & | \\
\hls & & + \\
& | \= \hd(11pt;2pt) Třetí část\+druhého\+řádku\= \\
\exs(5pt) & | \\
\hl
\= Třetí řádek \= | \= Druhá část\+třetího řádku\= |
\= \hd(11pt;2pt) Třetí část\+třetího\+řádku\= \\
\ex(3pt)
\hl
\endtable

```

Jiný příklad je tabulka generovaná příkazy

```

\sides
\table
\tss \rt6{\g(5pt)\c\g(5pt)&} \g(5pt)\c\g(5pt) \\
\sss \b\b\b\b\m\b \\
\cf\bfbf
\hl
\yy4 \m
Skupina A |
\xx6 \c
Skupina B \\
\hlyy
| \xx4 \c
Podskupina BA | \yy3 \m Podsk. BB |
\yy3 \m Podsk. BC \\
\hlyy
| \xx2 \c Metoda 1 | \xx2 \c Metoda 2 | | \\
\hlyy
\exa(3pt)
| \= Úloha 1\= | \= Úloha 2\+(Dini)\= |
\=Úloha 1\= | \=Úloha 2\+(Dini)\= | | \\
\exb(1pt) \hl
\endtable

```

Tabulka vypadá takto:

Skupina A	Skupina B						Podsk. BB	Podsk. BC
	Podskupina BA							
	Metoda 1		Metoda 2					
	Úloha 1	Úloha 2 (Dini)	Úloha 1	Úloha 2 (Dini)				

Tvorba tabulek v $\mathcal{L}_\mathcal{A}\mathcal{S}$ - \TeX u má mnoho dalších možností, o kterých jsem se zde nezmínil, jako je třeba možnost nastavení pevné šířky tabulky (např. na šířku stránky), možnost sloupců, které budou v `math`-modu (textovém nebo `display`), skládání větších tabulek z několika menších atd. atd.

Miroslav Dont

A přece se (jakžtakž) točí ...!

Na poslední valné hromadě $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{G}$ u se na mne o přestávce jednání obrátil jeden kolega (jeho jméno bohužel neznám) s následujícím zajímavým problémem: Lze zjistit skutečnou délku posledního řádku v odstavci?

Jako možnou aplikaci uváděl (poněkud svérázný) způsob sazby odstavců, kdy odsazení (`\parindent`) je vždy rovno právě délce posledního řádku v předchozím odstavci. Taková sazba by možná byla dobrá pro rychločtení, neboť se tím zřejmě zmenšuje počet očních fixací.

Od boku jsem vystřelil následující postup: (1) pomocí `\lastbox` získat poslední řádek a ten pak vybalit pomocí `\unhbox`. Dotyčný kolega mne přesvědčil, že to nefunguje. Několik dalších návrhů, jako třeba iterovat parametr `\parfillskip`, také neprošlo.

K rozhovoru se připojil Jiří Zlatuška a vyslovil názor, že to v \TeX u vůbec nejde. Navrhl též řešení, spočívající ve čtení dotyčného rozměru z DVI souboru. Tím by se určitě délka řádku zjistit dala, ale nebylo by ji možno použít *on-line*, například pro řešení výše zmíněné aplikace.