

TEX

Lenart Bučar

8. marec 2017

Povzetek

Abstract

Kazalo

1	Uvod	1
2	T_EX	3
3	Izpeljave T_EXa	5
4	L^AT_EX	7
4.1	Matematično okolje	7
4.2	Okolja	7
4.2.1	Naštevanje	7
4.2.2	Razpredelnice	8
4.3	Paketi	8
4.3.1	Beamer	8
4.3.2	Lipsum	9
4.4	Oblikovanje besedila	9
4.4.1	Upoštevanje prelomov vrstic	9
4.4.2	Oblikovanje besedila v oblike	10
4.5	Posebnosti pri pisanju besedila	10
4.5.1	Ligature	10
4.5.2	Majhne velike črke (small caps)	11
4.6	Posebnosti pri pisanju besedila	11
4.6.1	Ligature	11
4.6.2	Majhne velike črke (small caps)	11
4.7	Navajanje virov	12

Poglavje 1

Uvod

T_EX je odprtokodno programsko okolje, ki je namenjeno pisanju besedila z matematičnimi enačbami, in omogoča pisanje z t. i. makri, ki so žapakirani”v obliki paketov, med katerimi je najznamenitejši L^AT_EX.

Glavni razvijalec T_EXa je ameriški računalničar ter programer, Donald Knuth, prvo različico programa pa je izdal leta 1978. Knuth se je odločil za razvijanje T_EXa zaradi pisanja svoje knjige *The Art of Computer Programming*. Program je bil na začetku napisan v programskem jeziku SAIL. Kasneje je Knuth prišel na zamisel, da bi pisal tako program kot dokumentacijo naenkrat. Jezik ki ga je uporabil se imenuje WEB, končni program pa je v Pascalu.

Ena izmed zelo značilnih stvari pri T_EXu je označevanje verzij. Od verzije 3 dobi vsaka posodobitev na koncu dodatno decimalno števko, tako da se vse bolj približuje številu π . Trenutno smo pri verziji 3.14159265, ki je izšla leta 2014.

Poglavje 2

T_EX

T_EX temelji na podlagi ukazov, ki se začnejo z znakom `\`. Ti ukazi (macro-ji) se razširijo v besedilo ali pa v druge ukaze. Poleg tega pa lahko uporabljamo še zanke ter če-potem-drugače (if-then-else) kontrolne strukture. Ukazi imajo lahko tudi določeno število neobveznih argumentov, ki se jih po klicu ukaza našteje znotraj oglatih oklepajev (`[in]`), obvezne argumente pa se našteje znotraj zavitih oklepajev (`{ in }`). S temi ukazi potem spreminjamo velikost besedila, obliko, font...

Ker je T_EX napisan tako dobro, teče skoraj na vseh operacijskih sistemih. Poleg tega pa Knuth ponuja denarno nagrado vsakemu, ki bi najdel hrošča (bug) v njem. Ta nagrada je na začetku znašale le 2.56 ameriških dolarjev, nato pa se je vsako leto podvojila, dokler se ni ustavila na trenutni vrednosti, ki znaša 372,6 dolarjev. Seveda pa nihče, ki je prejel to nagrado, ni čeka nikoli unovčil temveč ga je raje uokviril.

Poglavje 3

Izpeljave T_EXa

Na T_EXu temelji kar nekaj drugih paketov makrov, med katerimi so najznamenitejši:

- L^AT_EX (Lamport T_EX),
- ConT_EX_T,
- AMS-T_EX,
- JadeT_EX,
- PicT_EX,
- T_EXinfo in drugi.

Obstaja tudi kar nekaj razširitev, kot so BibT_EX za bibliografske podatke (kot je navajanje virov), PDFT_EX in drugi. T_EX in vse njegove razširitve so zastonj na voljo v CTANu. CTAN ali *Comprehensive T_EX Archive Network* (Popoln arhiv T_EXa) je omrežje, kamor uporabniki nalagajo svoje razširitve za T_EX. Trenutno ima okoli 5500 paketov, ki jih je naložilo približno 2500 uporabnikov. Ker je program odprtokoden, je ne samo mogoče, temveč tudi zaželeno, da ga ostali uporabniki predelajo ali celo izboljšajo, edini pogoj je, da se ga ne sme distribuirati pod imenom T_EX.

Poglavje 4

L^AT_EX

L^AT_EX je ena izmed najznamenitejših distribucij makrov T_EXa, namenjena je pa pisanju knjig, znanstvenih člankov...

4.1 Matematično okolje

Tako kot v T_EXu je tudi v L^AT_EXu zelo razvit matematični način vnašanja, ki ga med drugim uporablja tudi Wikipedia. Inicializira se ga ali z znakom \$ (T_EX, deluje tudi v L^AT_EXu) ali pa z \ (Samo L^AT_EX). V prvem primeru se ga na enak način tudi konča, v drugem pa se ga konča z \. Matematične formule lahko zapišemo v vrstici (inline, inicializira se z \$ ali \ () ali pa izven vrstice (inicializira se z \$\$ ali \[). Primer zapisa znotraj vrstice v tem načinu je $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$, primer izven nje pa:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

V matematičnem načinu lahko tudi z lahkoto zapišemo grške crke kot so π (Π), ϕ (Φ)...

4.2 Okolja

Tako T_EX kot L^AT_EX nam ponujata več različnih okolij, ki spremenijo marsikatero pravilo pisanja in s tem tudi končni izgled. Enega izmed njih smo že spoznali, to je matematično okolje, ki pa se mi zdi tako pomembno, da sem ga obravnaval posebej. Vsa okolja se začne z ukazom `\begin{ime okolja}`, konča pa z `\end{ime okolja}`.

4.2.1 Naštevaneje

Včasih želimo narediti seznam, in za to imamo na voljo tri okolja. Vsako se seveda obnaša malce drugače. Prvo je "itemize", pri katerem so stvari naštete z določenim simbolom pred njimi (privzeto je to pika), izgleda pa takole:

- Lorem
- Ipsum
- Dolor

Naslednje je "enumerate", ki zaporedno oštevilči predmete:

1. Sit
2. Amet
3. Consectetur

Zadnje okolje pa je "description", pri katerem imamo ime predmeta ter njegov opis:

Adipiscing: Elit

Integer: Tempor

Est: Ut Ux

4.2.2 Razpredelnice

Zelo pogosto, še posebej v strokovnih člankih, poročilih itd. moramo dodati kakšno razpredelnico. \LaTeX nam to omogoča z uporabo okolja "tabular". Pri njegovem začetku moramo navesti koliko stolpcev ter kakšna vrsta bo to. Na voljo imamo sledeče možnosti:

- c za sredisko poravnano besedilo,
- l za levo poravnavo besedila,
- r da desno poravnavo,
- p za obliko odstavka ter
- | za navpično črto.

Ko pišemo podatke v razpredelnico, se med polji pomikamo z znakom & , vrstico pa končamo z `\`. Če želimo, lahko z uporabo ukaza `\hline` še vodoravno razmejimo celice. Primer tabele bi izgledal takole:

```
\begin{tabular}{|l c||c|r|} (Najprej navpična črta, levo poravnano besedilo, dve sredinsko cen-
tirani polji razmejeni z dvojno črto ter desno poravnano polje)
\hline (Črta na začetku)
Lorem & Ipsum & Dolor & Sit \\\ (Tukaj ni vodoravne črte)
Amet & Consectetuer & Adipiscing & Elit \\\ \hline \hline (Tu pa imamo dve)
Ut & Purus & Elit & Vestibulum \\\ \hline

\end{tabular}
```

Rezultat pa bi bil takšen:

Lorem	Ipsum	Dolor	Sit
Amet	Consectetuer	Adipiscing	Elit
Ut	Purus	Elit	Vestibulum

4.3 Paketi

Seveda se lahko zgodi, da si pisec zaželi nek ukaz, ki ga \LaTeX sam po sebi ne ponuja. Sedaj ima na voljo dve možnosti. Lahko sam s pomočjo že obstoječih ukazov napiše novega, lahko pa pogleda ali je to že kdo naredil. Ker se ponavadi izkaže da je, lahko to kodo doda v svoj program z uporabo ukaza `\usepackage`. Glede na razširjenost programa, niti ni presenetljivo, da obstajajo paketi za skoraj vse. Od takih, ki omogočajo lažje pisanje znanstvenih enot, pa do takega, pri katerem s pomočjo enega samega ukaza dodamo "Lorem Ipsum" besedilo za zapolnjevanje prostora.

4.3.1 Beamer

\LaTeX ponuja tudi možnost izdelave predstavitve, z uporabo paketa beamer. Pri tem paketu, uporabnik definira kako naj izgleda posamezna prosojnica, kateri elementi naj se kdaj prikazujejo...

4.3.2 Lipsum

Verjetno vsi poznate besedilo, Lorem Ipsum, ki se uporablja za:

- zapolnjevanje prostora,
- predstavitev izgleda fonta in
- kjerkoli drugje, kjer se potrebuje neko besedilo, ki ali ni pomembno kakšno je, ali pa se ga pisec ta trenutek ne more spomniti.

Kliče se ga z ukazom `\lipsum`, v oglatih oklepajih pa lahko navedemo število odstavkov. Na primer če napičemo `\lipsum[1]` dobimo ven nekaj približno takega:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4.4 Oblikovanje besedila

L^AT_EX pri generiranju končnega besedila odstrani vse večkratne presledke, prehode v novo vrstico itd. Zato sta si besedili: “Lorem Ipsum” in “Lorem Ipsum” enaki. Po končanem generiranju namreč pri prvem dobimo Lorem Ipsum, pri drugem pa Lorem Ipsum. Če na primer vpišemo

```
L
o
r
e
m
I
p
s
u
m
```

se nam izpiše L o r e m I p s u m.

Lahko pa se nam zgodi, da želimo besedilo oblikovati točno na določen način.

4.4.1 Upoštevanje prelomov vrstic

Če želimo, da L^AT_EX upošteva prelome vrstic, lahko uporabimo ukaz `\obeylines`. Če torej recimo napišemo:

```
{\obeylines
Lorem Ipsum
Dolor Sit
Amet
}
```

Se nam izpiše:

Lorem Ipsum

Dolor Sit

Amet

Če pa tega ukaza ne bi uporabili, bi z istim vhodom dobili: Lorem Ipsum Dolor Sit Amet.

4.4.2 Oblikovanje besedila v oblike

Z uporabo ukaza `\parshape` lahko oblikujemo besedilo v poljubne oblike. Kličemo ga v naslednji obliki: `\parshape=n i_1 l_1 \dots` pri čemer je n število vrstic, i_n zamik posamezne vrstice, l_n pa njena širina. Če torej vnesemo `\parshape=16`

```
0.25\hsize 0.5\hsize
0.24\hsize 0.52\hsize
0.23\hsize 0.54\hsize
0.22\hsize 0.56\hsize
0.21\hsize 0.58\hsize
0.20\hsize 0.6\hsize
0.19\hsize 0.62\hsize
0.18\hsize 0.64\hsize
0.17\hsize 0.66\hsize
0.16\hsize 0.68\hsize
0.15\hsize 0.70\hsize
0.14\hsize 0.72\hsize
0.13\hsize 0.74\hsize
0.12\hsize 0.76\hsize
0.11\hsize 0.78\hsize
0.10\hsize 0.80\hsize
```

`\lipsum[1]` (za prvi odstavek besedila Lorem Ipsum), dobimo:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Seveda pa lahko na ta način oblikujemo kakršne koli oblike želimo.

4.5 Posebnosti pri pisanju besedila

4.5.1 Ligature

\LaTeX pri pisanju besedila kjer je mogoče, uporablja *ligature*. To pomeni, da ima za dve črki, ki se zelo pogosto pojavljata skupaj poseben znak. Primeri tega so:

fi

fl

ff

Če pa želimo, lahko tako, da med črki vrinemo {} prisilimo odstranitev ligature, kar izgleda takole:

fi

fl

ff

4.5.2 Majhne velike črke (small caps)

Če želimo neko stvar poudariti, ampak ne napisati z velikimi črkami, lahko uporabimo tako imenovane “majhne velike črke”. Po izgledu so enake velikim, s to razliko, da so pomanjšane.

4.6 Posebnosti pri pisanju besedila

4.6.1 Ligature

L^AT_EX pri pisanju besedila kjer je mogoče, uporablja *ligature*. To pomeni, da ima za dve črki, ki se zelo pogosto pojavljata skupaj poseben znak. Primeri tega so:

fi

fl

ff

Če pa želimo, lahko tako, da med črki vrinemo {} prisilimo odstranitev ligature, kar izgleda takole:

fi

fl

ff

4.6.2 Majhne velike črke (small caps)

Če želimo neko stvar poudariti, ampak ne napisati z velikimi črkami, lahko uporabimo tako imenovane “majhne velike črke”. Po izgledu so enake velikim, s to razliko, da so pomanjšane.

4.7 Navajanje virov

\LaTeX nam z pomočjo dodatka Bib \TeX omogoča preprosto navajanje virov. Potrebujemo le dodatno *.bib* datoteko, ki jo na zelenem mestu kličemo z `\bibliography{ime datoteke}`. Datoteka mora biti seveda pravilno oblikovana, da jo \LaTeX lahko prebere. Vsako delo, mora imeti svoj “odstavek”, ki pa izgleda takole:

```
@tip dela { interna oznaka, ki jo bomo uporabljali za citiranje,
  author = {},
  title = "",
  journal = "",
  volume = "",
  number = "",
  pages = "",
  year = "",
  DOI = ""
  publisher = "",
  year = "",
  url = "",
  note = "",
}
```