

# LaTeX за начинаещи : професионално оформяне на страници достъпно за всички

Кики Новак

В чудесният свят на информатиката, малко програми са предизвиквали толкова полемика и противоречия като LaTeX (LaTeX), програма за текстообработка, широко разпространена в научните и университетски среди и непозната за обикновения потребител. Ако се позоваваме на предразсъдъците, които са често срещани в различни форуми, използването на LaTeX е строго резервирано за по-горе цитираната клиентела : това е програма направена от „учени“ за „учени“.

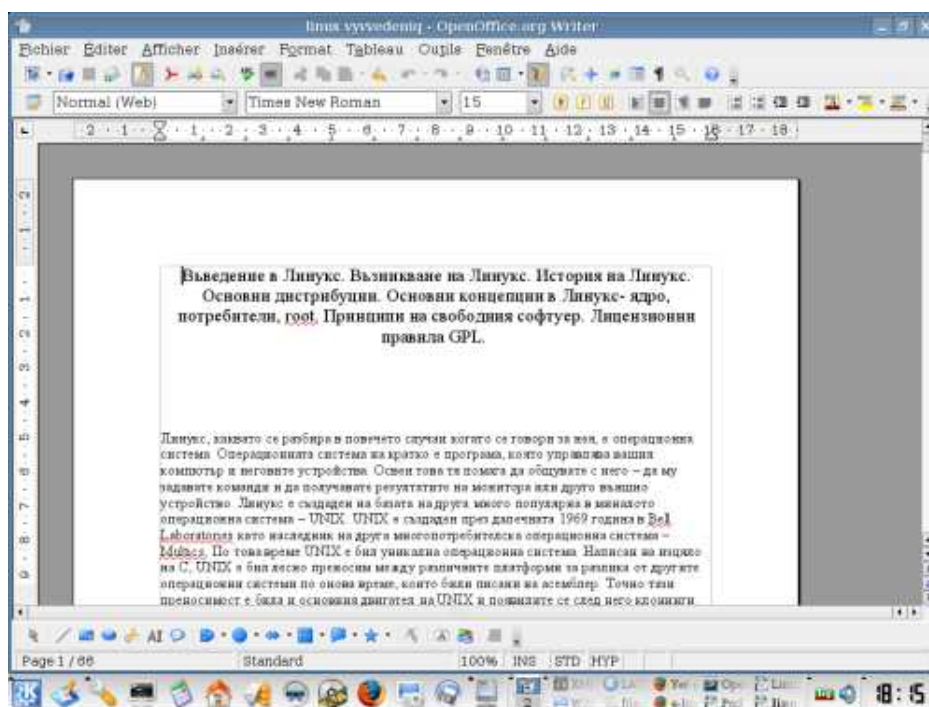
## Малко история преди да започнем...

TeX(което се произнася „Тек“ като „technique“ на английски), програмата, която е в основата на LaTeX, е разработена през 1977 от американският математик Доналд Кнът за писане на текстове, съдържащи математически формули. Кнът е бил недоволен от начина, по който статиите му са били публикувани от *American Mathematical Society*. TeX позволява на потребителите да извършват текстообработката на собствения им компютър, по начин достъпен дотогава само за издателските къщи. За тази цел, единственото условие е да се знаят известен брой команди : някакви си 300 базови команди и около 600 макроси. Първата версия на TeX е публикувана през 1982, и е подобрена през годините.

За да улесни използването на TeX, Лесли Лампорт разработва LaTeX, серия от макроси позволяваща текстообработка, използвайки серия от готови професионални формати. По този начин LaTeX е много по-лесен за използване от TeX, тъй като се нуждае единствено от данни, дефиниращи логичната структура на един текст, за да ги превърне след това в серия команди TeX, които ще осъществят „автоматичното оформяне“ на страницата. През 1994, LaTeX се преработва и обединява всички развили се дотогава версии. Новата версия се казва LaTeX2ε. Може да отбележим че номера на версия на TeX, клони към числото  $pi(3.14159...)$ , което без съмнение доказва доброто кодиране на програмата, тъй като TeX е напълно изчистена от грешки: още една причина за използването на LaTeX!

## 1 Въведение

Вярно е че TeX - в основата на LaTeX - е създадена от математик, разочарован от лошото оформяне на статиите му в специализираната преса. Вярно е също, че



Фигура 1: OpenOffice.org Writer

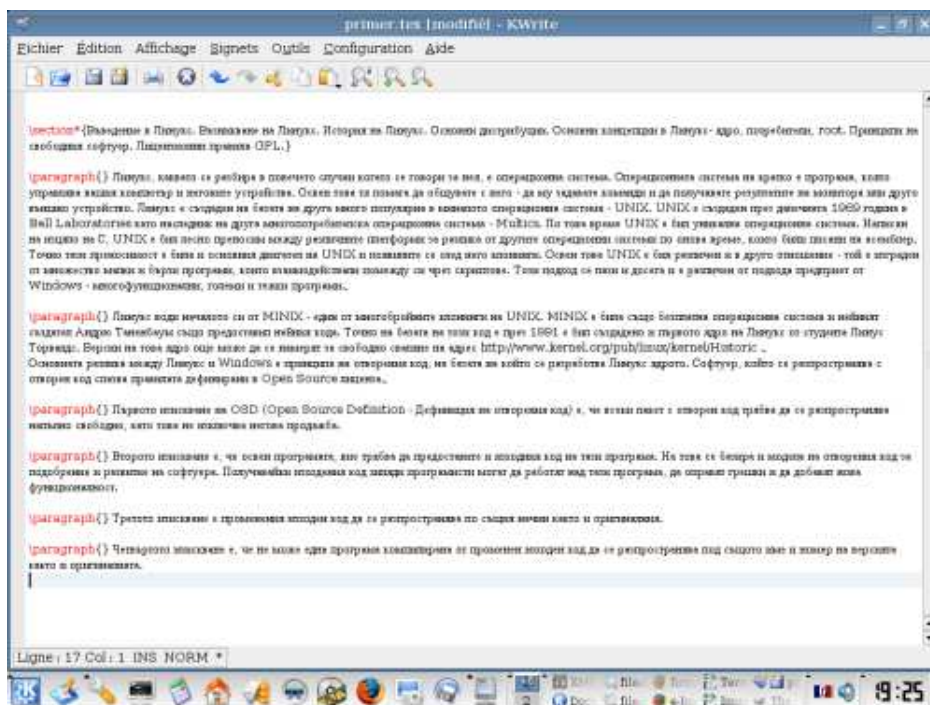
ЛаТеК съдържа хиляди функционалности, класирани в учебници, често използвани да подпират някой прозорец да не се блъска от вятъра.

И все пак, ЛаТеК има качествата да се превърне в предпочитаната програма на журналисти, писатели, студенти и други. Защото е достатъчно да се абстрахираме от всичките „изтънчени“ функционалности на тази програма, за да получим един инструмент за текстообработка колкото силен, толкова и лесен за използване, и способен да създава документи и текстове, на високо ниво, чието качество в голям мащаб надминава всичко, което може да се направи с най-добрия текст-редактор.

Тази статия предлага пълно въведение в „литературното“ използване на ЛаТеК. Тя ще ни позволи редактирането на всякакъв тип документи, статии, новели, романи, с общи думи всичко, което не използва графики и математически или химични формули, с качеството на издателска къща. Сега остава само да опитате.

## 2 Каква е разликата с един текст-редактор?

Един текст-редактор (*OpenOffice.org Writer, Abiword, Kword, Microsoft Word...*) работи на принципа WISIWIG : What You See Is What You Get ( = Това, което виждате е това, което ще получите). Или по-конкретно, това, което пиша на екрана е това, което ще излезе отпечатано.



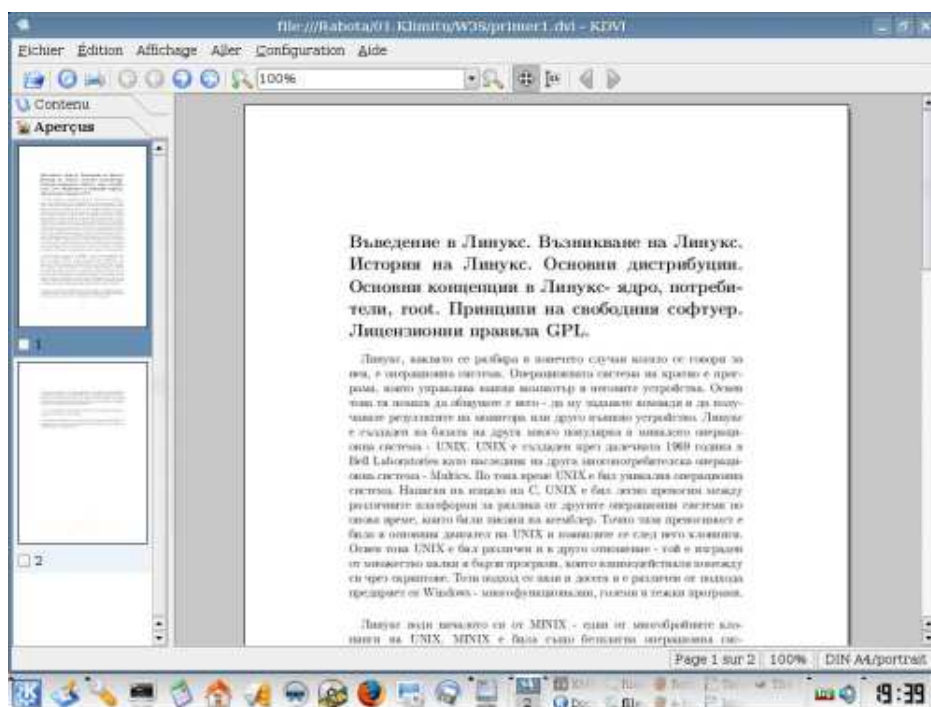
Фигура 2: Текстов редактор

За да напишете един текст (например глава на статия), голяма част от вас знаят процедурата. Избиране на шрифта (Times New Roman примерно), избиране на големината на шрифта на заглавието и на основния текст, избор на интервала между линиите, маркиране на съответния текст и натискане на бутона за удебелен шрифт, избор на центриране на текста и така нататък.

Да видим как изглежда текстообработката с LaTeX на същия текст. Тук трябва да минем през три етапа : редактиране, компилиране и визуализиране. На първо време, трябва да се създаде така наречения файл с изходен код с обикновен текстов редактор, както е показано на Фигура 2.

Противно на по-горния пример, този файл с изходен код не съдържа само текст, но и пояснения относно структурата на документа; тази основна разлика ще разберем с няколко примера, представени по-долу. След това файла с изходен код се компилира (ще разберем как) и резултата се показва в нов прозорец, както е показано на Фигура 3.

А сега поемете дълбоко въздух и се възхитете на разликата в качеството на двата текста.[...]. Продължаваме ли?



Фигура 3: Резултат Latex

### 3 Инсталиране на LaTeX

Пакетите LaTeX имат различни имена в зависимост от дистрибуциите:

- Debian : tetex-base, tetex-bin и tetex-doc.
- Mandrake : tetex и tetex-dvips. За предпочитане е да се инсталират незадължителните пакети tetex-latex, tetex-xdvi и tetex-doc.
- Red Hat : tetex, tetex-fonts и tetex-latex.
- SuSE : tetex и te-latex.
- Gentoo : emerge tetex (пакет dummy).

За останалите дистрибуции, консултирайте различните форуми или най-добре, нюз-групите *comp.text.tex*. Но не се притеснявайте, всички големи дистрибуции предлагат пакети LaTeX. И за да имате някаква идея за броя на потребителите на LaTeX (способни да ви помогнат при проблем) един малък пример : в американските университети LaTeX се използва много повече от Microsoft Word за редактирането на дипломни работи.

## 4 Първият ми документ ЛaTeX

*Сива е, скъпи приятелю, всяка теория!* казва Метистофелес на Фауст в епичната пиеса. Ето защо ще преминем директно на практическата част. Най-напред отворете един текстов редактор и създайте файла *test.tex*. Сред текстовите редактори, има известните Vim и Emacs, множество малки и лесни за работа редактори като nano, pico или ed и - last but not least - серия графични редактори като KWrite, KEdit, Kate или GEdit. Изборът е ваш, но не използвайте KWord, Abiword или OpenOffice.org Writer, които обработват текст, но не са редактори, защото може да имате няколко „приятни“ изненади.

Аз лично - и без да предизвиквам войни на религиозна основа ;о)-ви съветвам да използвате Vim, който предлага много специфични функционалности за редактирането на файловете ЛaTeX. Базите на редактора Vim се научават сравнително бързо, и за по-напредналото му използване можете да прочетете в множество самоучители. Освен това владението на Vim ще ви е от голяма полза в много други моменти, ако работите под Линукс.

Ето, на какво трябва да прилича вашия файл с изходен код *test.tex*:

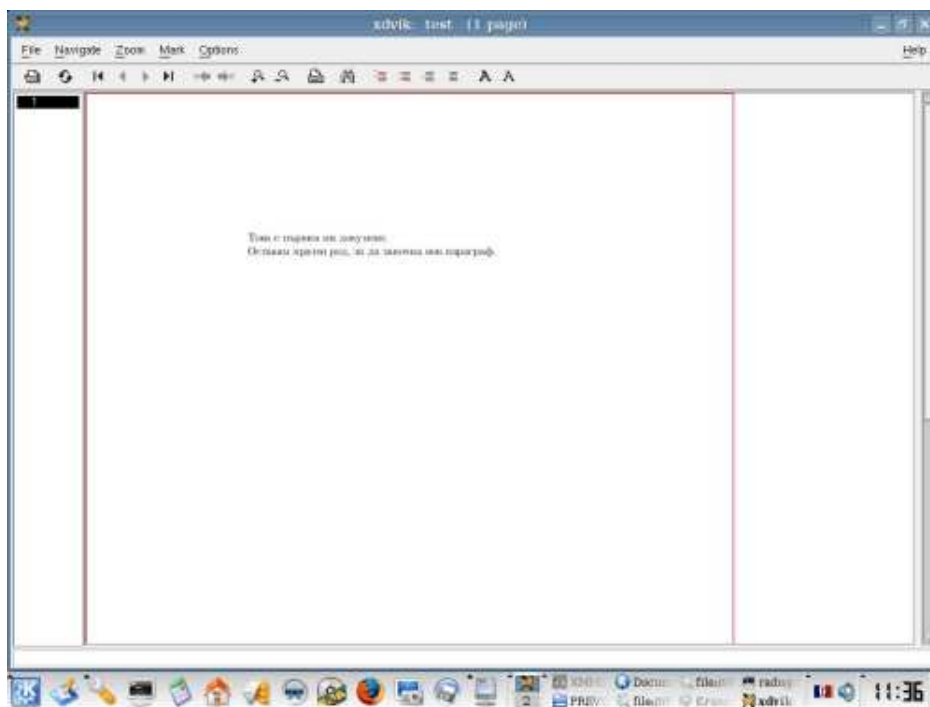
```
\documentclass{article}
\usepackage[T2A]{fontenc}
\usepackage[koi8-r]{inputenc}
\begin{document}
Това е първия ми документ.
```

```
Оставям празен ред, за да започна нов параграф.
\end{document}
```

Запишете вашия файл, затворете текстовия редактор и пуснете компилирането чрез командата `latex test.tex`, която пишете в една конзола (konsole, xterm, rxvt и други). Трябва да сте в същата директория, в която се намира и файла *test.tex*. Ще получите нещо, което прилича на следното:

```
kikinovac@ninthcircle:~$latex test.tex
```

```
This is pdfTeX, Version 3.141592-1.30.5-2.2 (Web2C 7.5.5)
entering extended mode
(./test.tex
LaTeX2e <2003/12/01>
Babel <v3.8d> and hyphenation patterns for american, french, german, ngerman,
bahasa, basque, bulgarian, catalan, croatian, czech, danish, dutch, esperanto, estonian,
finnish, greek, icelandic, irish, italian, latin, magyar, norsk, polish,
portuges, romanian, russian, serbian, slovak, slovene, spanish, swedish,
turkish, ukrainian, nohyphenation, loaded.
(/usr/share/texmf/tex/latex/base/article.cls
```



Фигура 4: Първият документ ЛaTeX

```
Document Class: article 2004/02/16 v1.4f Standard LaTeX document class
(/usr/share/texmf/tex/latex/base/size10.clo)
(/usr/share/texmf/tex/latex/base/fontenc.sty
(/usr/share/texmf/tex/latex/cyrillic/t2aenc.def)
(/usr/share/texmf/tex/latex/cyrillic/t2acmr.fd))
(/usr/share/texmf/tex/latex/base/inputenc.sty
(/usr/share/texmf/tex/latex/cyrillic/koi8-r.def))(. /test.aux)[1](. /test.aux)
Output written on test.dvi (1 page, 376 bytes).
Transcript written on test.log.
```

Сега проверете съдържанието на директорията, в която работите (с командата `ls` или в някой от файловите навигатори *mc*, *Nautilus* или *Konqueror*). Ако всичко е наред, ще намерите три новосъздадени файла : *test.aux*, *test.dvi* и *test.log*. Файлът, който ни интересува е *test.dvi* ("device independent"). Отново от конзолата, ще можем да визуализираме първия ни документ ЛaTeX, с помощта на командата:

```
kikinovac@ninthcircle:~$ xdvi test.tex
```

Не се притеснявайте от съобщения от рода на „kpath-sea: Running mktexfm...“, които се появяват в конзолата. Това са само различни шрифтове, които ЛaTeX създава, ако не ги намира в паметта (Фигура 4).

Прозореца `xdvi` позволява да се движите в документа с помощта на мишката, или използвайки копчетата на клавиатурата **n** (следваща страница), **p** (предишна

страница) и **q** (затваряне на прозореца `xdvi`).

Ако нещо не е станало, като предвидено, първо проверете дали сте инсталирали добре всички пакети, необходими за работенето на програмата ЛaTeX.

Следващата най-често срещана причина за грешка идва от файла с изходен код: достатъчно е да използвате различни скоби, като например `[ ]` вместо `{ }` или пък да забравите `\end{document}` на края на файла, и ЛaTeX показва серия от съобщения за грешка. В такъв случай, проверете съдържанието на файла `test.tex`, поправете евентуалните грешки, запишете промените и компилирайте отново с командата **latex test.tex**.

Не е нужно да затваряте прозореца `xdvi`. Ако сте прекомпилирали променения файл `test.tex` е достатъчно да цъкнете на бутона “Reread” или да натиснете **R**, да за опресните. Консултирайте ръководството на `xdvi` (с командата **man xdvi**), за да видите пълния списък на съществуващи команди.

За да можете да отпечатате първия ви документ ЛaTeX е нужно да го превърнете във файл PostScript с командата `dvips`:

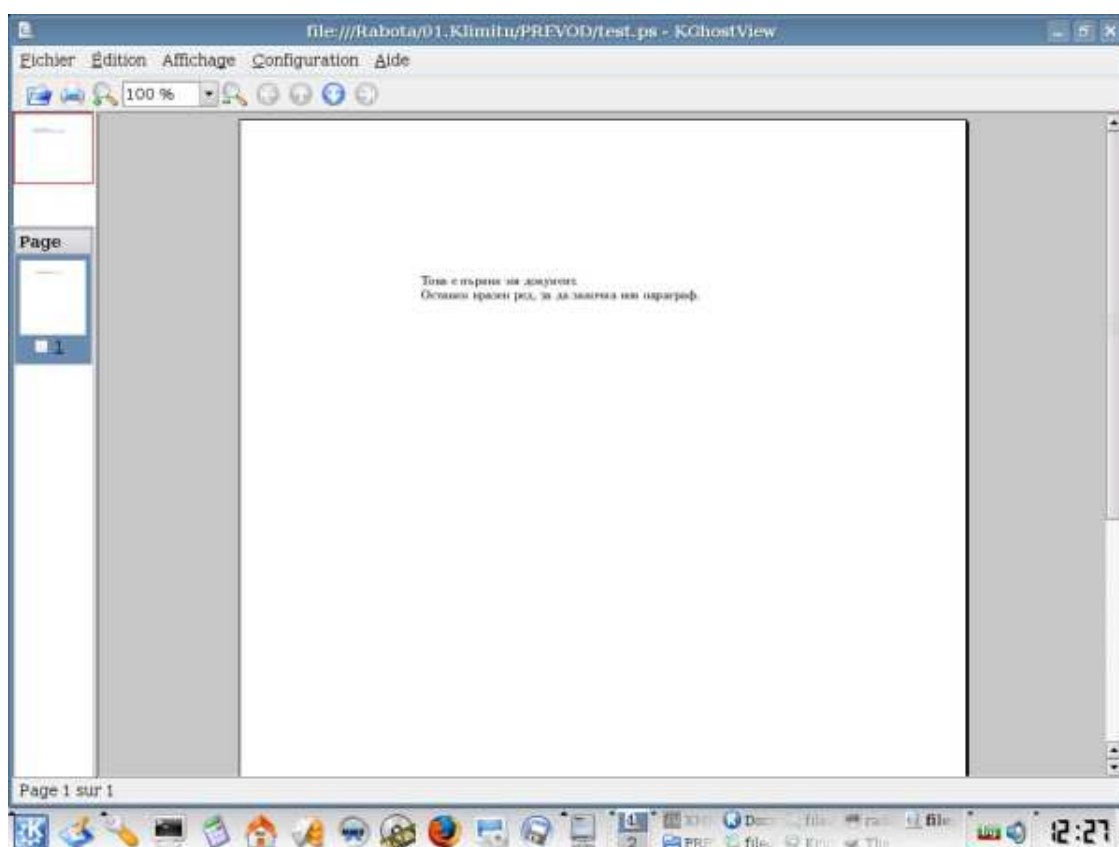
```
$ dvips test.dvi -o
This is dvips(k) 5.95b Copyright 2005 Radical Eye Software (www.radicleye.com)
' TeX output 2006.07.20:1045' -> test.ps
<tex.pro>. [1]
```

Незадължителният параметър **-o** предизвиква отпечатването на резултата „в един файл“. Този файл `test.ps` може да бъде отпечатан (с командата `lpr` например) или да се види с `gv`, `gnome-gv` или `kghostview` (Фигура 5).

## 5 Защо да е лесно, като може да е трудно?

Примерът, който бе описан по-горе съответства на „класическия“ метод на ЛaTeX: създаване на файл с изходен код във формат `.tex`, компилирането му с помощта на командата **latex**, евентуалната проверка на резултата с **xdvi**, и превръщането на файла `dvi` в *PostScript* файл с помощта на **dvips**.

Втори метод, много по-лесен от първия, е директното компилиране на файла с изходен код във формат PDF, чрез командата **pdflatex**. От файла `test.tex`, чрез командата **pdflatex test.tex** ще получите файл `test.pdf`, който можете да видите с помощта на една от по-горе споменатите Ghostview програми или с `acroread`, линукска версия на Acrobat Reader.



Фигура 5: PostScript файл



## 6 Малко теория

Току що създадохте първия си документ ЛaTeX, чиято структура съответства на тази на всички документи ЛaTeX:

Началото на документа (пояснителната част) - или всичко, което е между самото начало и командата `\begin { document }` - съдържа команди, дефиниращи структурата на документа, големината на шрифта, формата на листа, броя колони...Той трябва да съдържа поне една команда: тип на документа (`article`, `book`, `report`...които ще разгледаме подробно по-нататък)

Една команда е съставена от една обратна наклонена черта (`\`), с която се започва, име на командата, квадратни скоби `[ ]`, съдържащи евентуални параметри и къдрави скоби `{ }`, съдържащи параметри.

ЛaTeX е модулно цяло, чиито способности могат да бъдат допълнени чрез пояснения. Някои от тези пояснения, като `fontenc` и `inputenc`, са част от стандартната дистрибуция на ЛaTeX2ε.

Използвайте всеки път командите: `\usepackage[T2A]{fontenc}`, `\usepackage[koi8-r]{inputenc}` и `\usepackage[bulgarian]{babel}` в началото на вашия документ, за да дефинирате шифроването на документа и за да улесните вземането предвид от програмата на специалните знаци и ударенията, които ще можете тогава директно да пишете<sup>1</sup>.

Ако работите с латинската азбука, използвайте командите: `\usepackage[latin1]{inputenc}` и `\usepackage[T1]{fontenc}`. Ако пишете текст на френски, дефинирайте го като език на работа в началото, с помощта на командата: `\usepackage[frenchb]{babel}`. Внимание : **frenchb**, а не `french`.

Ако в началото на документа няма други команди и опции, ЛaTeX ще избере стойности по подразбиране за големината на листа, шрифта и белите полета, типа на шрифта и броя на колоните текст. Началото на документа свършва с командата: `\begin { document }`.

Между командите: `\begin { document }` и `\end { document }` се намира самия текст (или така наречената главна част на текста), както и команди, отнасящи се за текста на локално ниво:

- Емфази, тоест команди за получерен шрифт, курсивен шрифт и така нататък;
- Команди за промяна на шрифта;

---

<sup>1</sup>Тези команди могат да бъдат заменени с `\usepackage[T2A,OT1]{fontenc}`, `\usepackage[ot2enc]{inputenc}` и `\usepackage[bulgarian,english]{babel}` или с `\usepackage[T2A]{fontenc}`, `\usepackage[utf8]{inputenc}` и `\usepackage[bulgarian]{babel}`.

- Команди за промяна на полетата или центрирането на текста;
- Таблиците;
- Команди за нов параграф, и така нататък.

За да започнете нов параграф, оставете един празен ред в текста. ЛaTeX подравнява двустранно текста, така че не трябва да се притеснявате за евентуални пренасяния на думи. Използването на **babel** с опцията `bulgarian` ще осигури правилното пренасяне на думите.

## 7 Малко повече практика

Ето, че вече се сдобихме с необходимите знания за създаването на един „истински“ документ. Примерът, който следва ще ни позволи да се научим да пишем новела...или всякакъв друг текст, който не съдържа математически или химични формули, както и специални знаци, таблици, нито някаква йерархична подредба в глави и секции. Ако използвате свободното си време в писане, сега е момента да дадете на писанията си професионален вид. Ако ли не, винаги можете да се сдобиете с някоя новопубликувана новела по Интернет. Това е последното, което липсва в Мрежата.

Вземете първия ви документ ЛaTeX `test.tex` и изтрийте главната част на текста. Трябва да получите това:

```
\documentclass{article}
\usepackage[T2A]{fontenc}
\usepackage[koi8-r]{inputenc}
\begin{document}

\end{document}
```

Променете първия ред и сменете типа на документа, както и предефинираната големина на листа: `\documentclass[a4paper]{book}`.

След това, прибавете пояснения за заглавието и автора, като например:

```
\author{Ги дьо Мопасан}
\title{Щастieto}
\date{ }
```

Значението на първите два реда е очевидно. Командата `\date{ }`, използвана без пояснителен параметър, служи за изписване на днешната дата под заглавието.

Сега, копирайте и поставете през клипборда (или препишете) текста на вашата новела между командите: `\begin { document }` и `\end{document}` като внимавате да:

- Оставете празен ред всеки път, когато започвате нов параграф.
- За момента, избягвайте да използвате специални знаци като `#`, `$`, `%`, `^`, `&`, `_`, `{`, `}`, `~` и `\`. Това са знаци, които имат специфично значение в ЛaTeX. Ако ги напишете директно, може да очаквате изненади.
- Избягвайте двойните кавички(`"`), по-скоро използвайте само единични(`'`), или френски кавички `«и»`;
- Изберете сравнително дълъг текст: между пет и двадесет страници.

## 8 Блокове текст и пренасяне

Запишете текста под името *nouvelle.tex* и го компилирайте чрез командата **latex nouvelle.tex**. Ако сте избрали достатъчно дълъг текст, много е вероятно да получите множество съобщения за грешка, които приличат на това:

```
Overfull \hbox(1.83568pt too wide) in paragraph in lines 35-40 [ ] \ T2A /cmr/m/n/12
Раз-ли-ча-вах-ме раз-лич-ни-те слу-ча-и, пос-та-вах-ме гра-ни-ци, ци-ти-рах-ме при-
мерите ;
```

Какво означава това? За да се разбере това съобщение за грешка трябва да се знае, че ЛaTeX не само подравнява текста, но следи и подредането на думите на страницата, опитвайки се да избегне празните места, които могат да се срещнат в обработката на текста. Ако е нужно, ще се създаде пренасяне на дума, и сега разбираме защо е необходимо да се уточни езика, на който ще се пише: за да знае ЛaTeX как да пренесе думите.

Едно пренасяне на нов ред не винаги е възможно, или защото ЛaTeX не разпознава думата, или защото правилата за пренасяне го забраняват. За да се види това на нашия документ, използваме командата **xdvi nouvelle.dvi** и намираме изречението, което поставя проблем. Щраквайки с мишката върху текста (опитайте щракване с десния бутон и след това с левия, за да видите разликата) използваме лупата за уголемяване на `xdvi`.

В този случай, проблема е в думата „примерите“: виждаме, че тя леко излиза от полето с текст. ЛaTeX знае, че на български правилото за пренасяне забранява пренасянето на последната сричка, ако тя е определителен член. Но понякога сме принудени да го направим, и затова ще позволим пренасянето, пояснявайки го във файла с изходен код: `при\ -ме\ -ри\ -те`. Тази команда казва на ЛaTeX, че е възможно пренасяне на посочените места.

След тази малка корекция, записваме файла и го прекомпилираме (**latex nouvelle.tex**). Този път няма съобщения за грешка на посоченото по-горе място и текста е правилно подравнен в дясно. Остава само да повторим същото и за останалите думи, посочени от ЛaTeX като грешка.

Трябва да отбележим, че в някои случаи е трудно, и дори невъзможно да се намери задоволително разрешение. В такъв случай може да се използва командата `\sloppy`, за да е ЛaTeX по-малко прецизен и да остави празно място между думите в изречението или параграфа. Тогава ще се появи съобщение „Underfull \hbox“, но крайния резултат е въпреки всичко задоволителен.

Възможно е ЛaTeX да игнорира някои от използваните думи и да не знае как да ги пренесе на нов ред. В този случай използвайте командата `\hyphernation { списък с думи }` в началото на документа и избройте думите, от които ЛaTeX има нужда, като например:

```
\hyphernation { не-ра-бо-то-спо-собен не-ра-бо-то-спо-соб-ност }
```

## 9 Структурирайте документите

За разлика от обработването на текст WYSIWIG, с ЛaTeX може да структурирате документите много лесно. Тези от вас, които пишат доклади или дипломни работи особено ще оценят тази функция.

Преди да започнем с това, да разгледаме различните типове документи под ЛaTeX. В предните примери, ние вече използвахме някои, без да си дадем сметка за това: **article**, когато създадохме първия ни документ *test.tex* и **book** за *nouvelle.tex*. Ето основните типове документи предложени от ЛaTeX :

- **article**: Най-често използваният тип документ, много практичен за не много дълги статии. Първото ниво на подразделение е `\section` (ще разберете след малко).
- **book**: Тип документ създаден специално за писането на книги. Отпечатването е от двете страни на листа. Разделянето на отделни глави се прави с помощта на командата `\chapter`.
- **report**: Това е тип документ за по-дълги технически статии и доклади, с възможност отделните секции да се организират в глави с `\chapter`.

Сред останалите типове трябва да отбележим **letter** за писма и **slides** за представяне на доклади. Типът **memoir** е създаден специално за „литературни“ писания (но не само) и ще го разгледаме малко по-нататък.

Сега създайте един файл *section.tex* като този:

```
\documentclass{article}
\usepackage[T2A]{fontenc}
\usepackage[koi8-r]{inputenc}

\begin{document}

\section{Ниво 1 }

Това е първото ниво.

\section{Ниво 2}

Това е второто ниво.

\subsection{Ниво 3}
Това е 1-вото под-ниво.
\subsection{Ниво 4}
Това е 2-рото под-ниво.
\paragraph{Параграф 1} изглежда така.
\end{document}
```

Компилирайте файла (**latex sections.tex**) и го визуализирайте (**xdvi sections.dvi**) (Фигура 6).

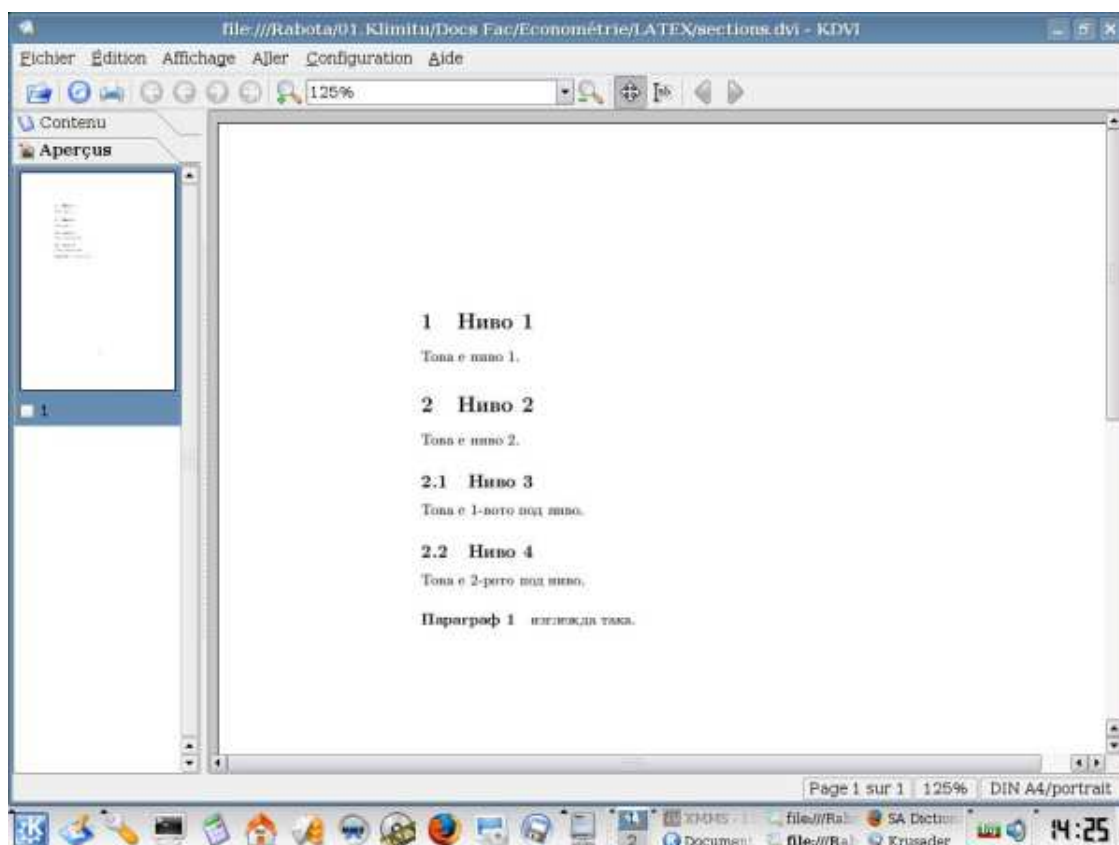
Сигурно сте приятно изненадани, че номерирането на различните секции, изборът на шрифта и т.н. се правят автоматично. Нещо, което ще ви позволи да се концентрирате повече върху същината на работата, без да се притеснявате за оформянето на текста на страницата.

## 10 Повече практика

*Използвай или ще изгубиш*, е известна английска поговорка, която означава, че ако не използваш новопридобитото познание рискуваш бързо да го изгубиш. Ето няколко практически съвети за да приложите всичко, научено до тук.

Изберете си някакъв текст в Интернет, който е разделен в секции. Опитайте се да го оформите, както направихме преди малко, използвайки типа документ **article**.

Сигурно още пазите дискетата с дипломната ви работа върху намалянето на артикулационната енергия в римския език? Или пък доктората ви върху изобразяването на птицата в средновековната литература? Или пък просто серията статии за свободния софтуер? Тогава, вземете част от същинската част на текста (без епиграфи, благодарности и библиография) и се опитайте да го оформите с ЛaTeX, използвайки типа документ **article** или **report**.



Фигура 6: Оформяне на документа

Изберете си някой публикуван в Мрежата роман, или пък просто някой роман, който събира прах в библиотеката ви. За момента използвайте само основната част на текста, както обяснихме по-горе и направете текстовото оформление с ЛaTeX, използвайки **book** и разделяйки текста на отделни глави с помощта на командата `\chapter{Заглавие на главата}` за главите на книгата, които имат заглавие и `\chapter{}` за тези, които нямат.

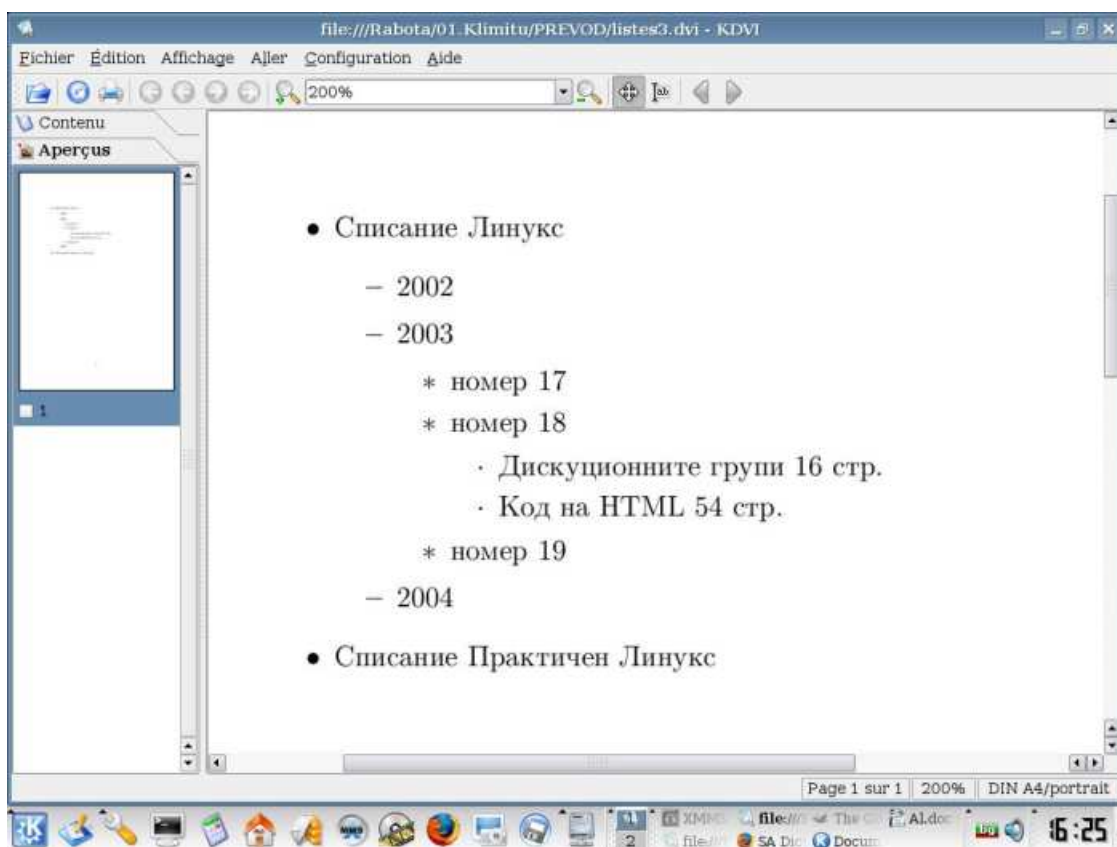
Възможно е в тези текстове да срещнете специални знаци като: #, \$, %, ^, &, \_, {, }, ~ и \. Можете да ги използвате слагайки една обратна наклонена черта пред тях (`\#`, `\$`, `\%`, `\^`, `\&`, `\_`, `\{`, `\}` и `\~`) с изключение на обратната наклонена черта, която ще получите чрез командата `\textbackslash`.

## 11 Правене на списъци

С ЛaTeX е възможно да се създават различни списъци: обикновени, изброявания и описания. За да разберем как става това, ще използваме както обикновено един пример. Създайте файл *listes.tex*, дефинирайте типа документ и началото на документа, както правихме досега и след това напишете следния списък :

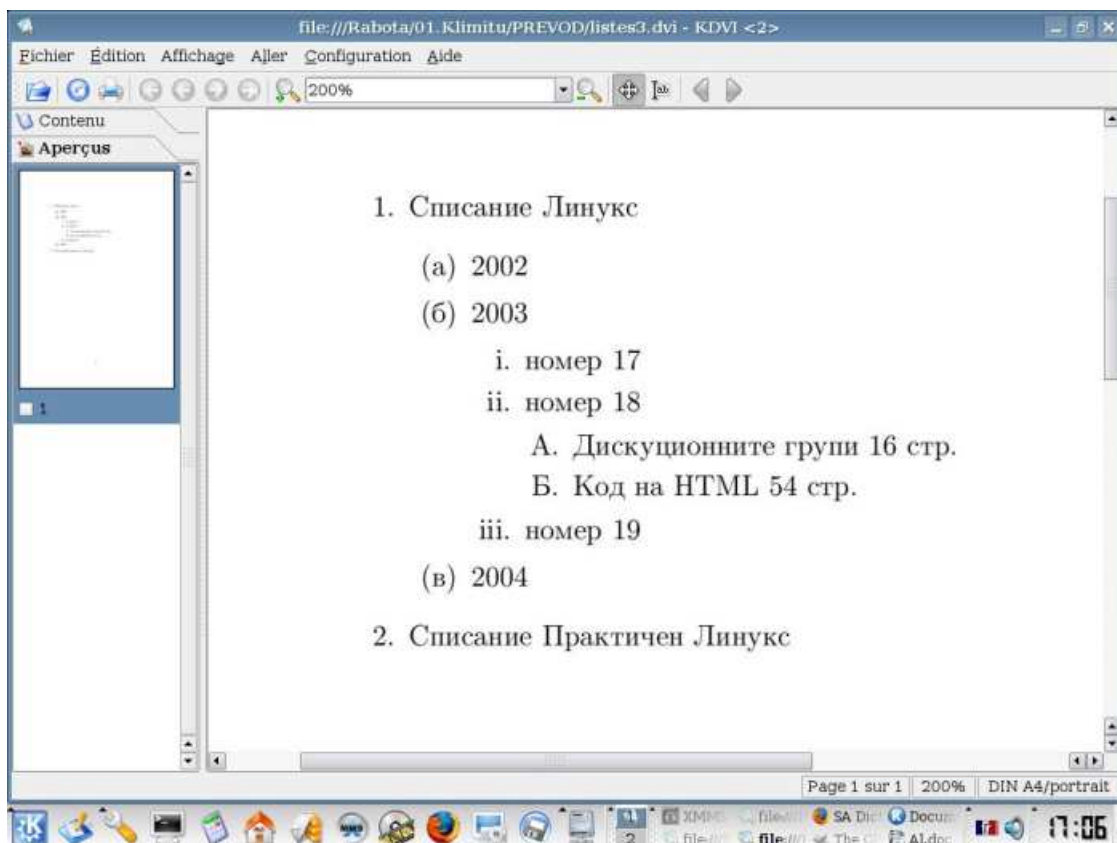
```
\begin{document}
  \begin{itemize}
    \item Списание Линукс
      \begin{itemize}
        \item 2002
        \item 2003
          \begin{itemize}
            \item номер 17
            \item номер 18
              \begin{itemize}
                \item Дискусионните групи 16 стр.
                \item Код на HTML 54 стр.
              \end{itemize}
            \end{itemize}
          \item номер 19
        \end{itemize}
      \item 2004
    \end{itemize}
  \item Списание Практичен Линукс
\end{itemize}
\end{document}
```

Както обикновено компилирайте файла с изходен код (**latex listes.tex**) и визуализирайте резултата (**xdvi listes.dvi**)(Фигура 7).



Фигура 7: Обикновен списък





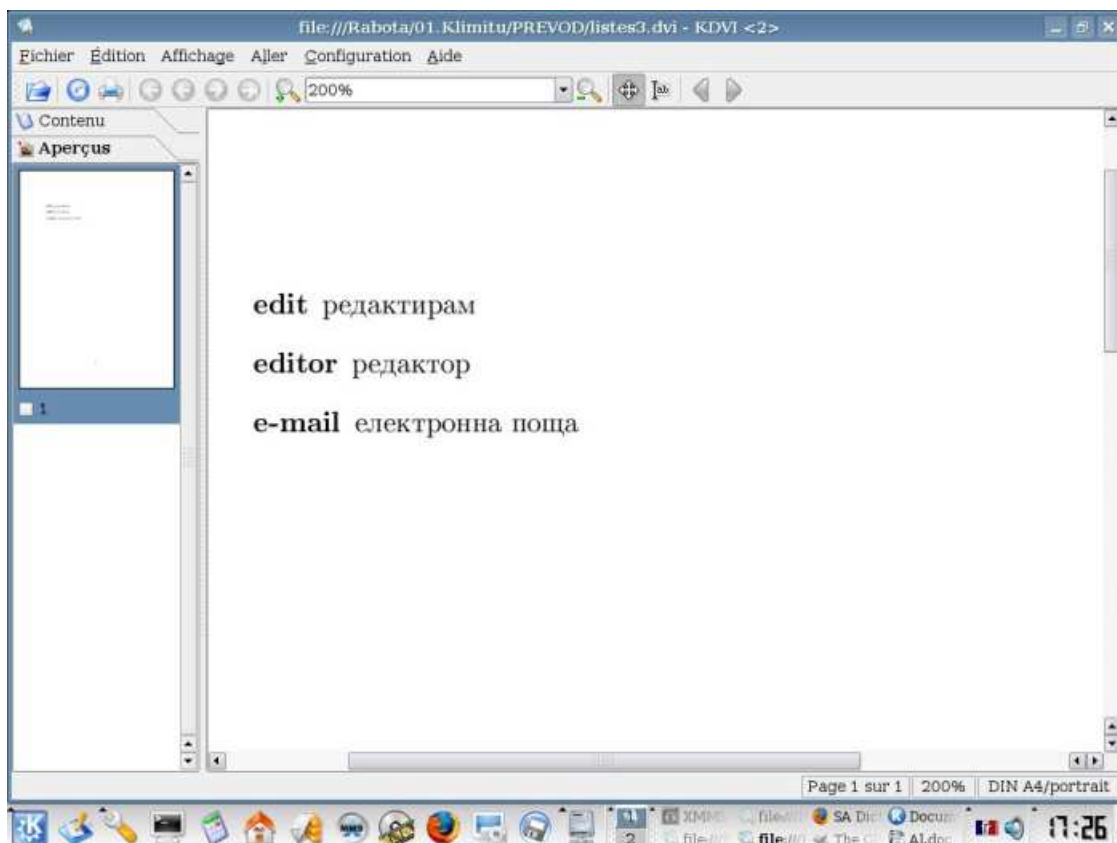
Фигура 8: Изброяване

Сега да си обясним значението на кода, който написахме преди малко и резултата, като направим резюме на случилото се:

- Един обикновен списък се намира между командите `\begin{itemize}` и `\end{itemize}`. В такъв случай казваме че списъка е в *itemize* - *среда*.
- Всеки елемент от списъка се въвежда с командата `\item`.
- Тези списъци могат да се застъпват до 4 под-нива. Желателно е да пишете кода както е показано в примера по-горе, за да може лесно да се ориентирате.

Списъците с изброяване работят на същия принцип. За да видите резултата заместете `itemize` с `enumerate` във вашия файл `listes.tex`. Запишете, компилирайте и вижте Фигура 8.

Тук номерирането започва от 1, и също е възможно да застъпвате нивата. В този случай второ ниво ще е с малки букви, трето с римски цифри, а четвърто с главни букви.



Фигура 9: Дефиниция

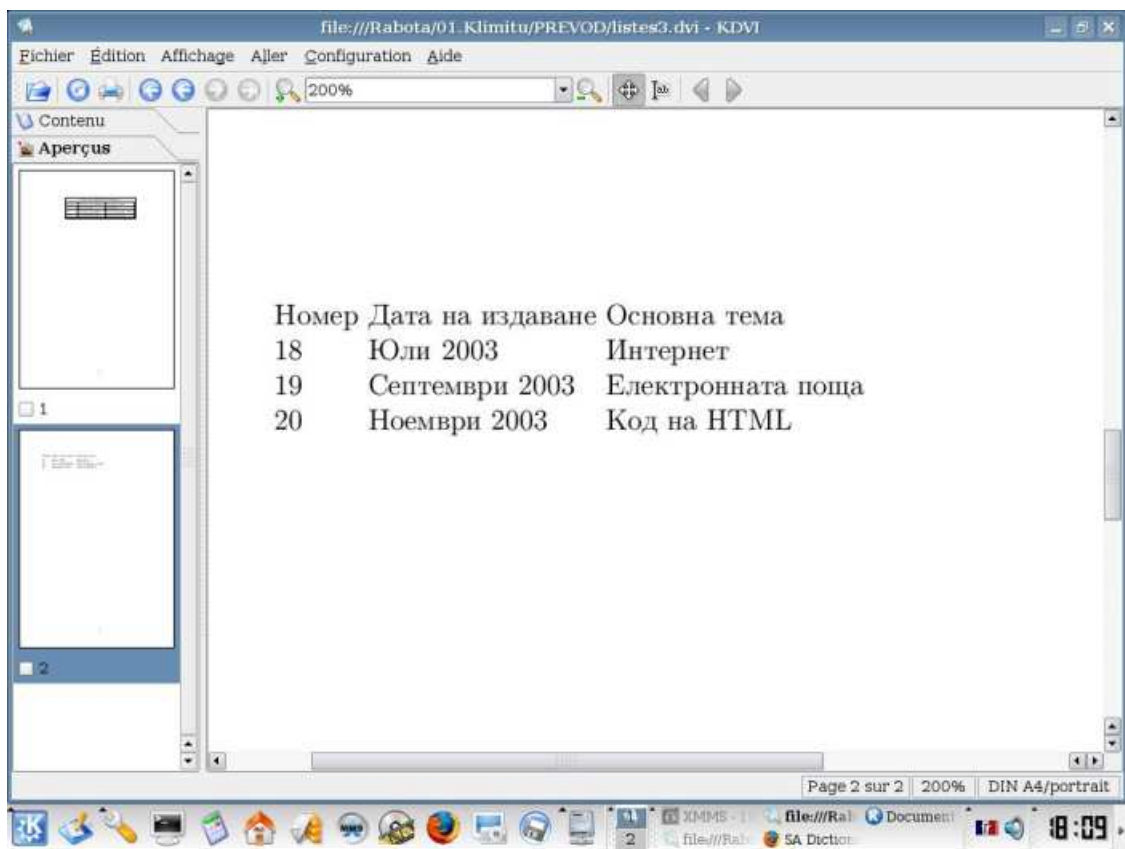
Що се отнася до средата **description**, както може да се досетим от името, тя се използва за описание, или за списъци с дефиниции каквито има в речниците (Фигура 9):

```
\begin{document}
  \begin{description}
    \item [edit] редактирам
    \item [editor] редактор
    \item [e-mail] електронна поща
  \end{description}
\end{document}
```

## 12 Табулатор и таблици

По време на компилирането ЛaTeX не взема предвид интервалите и празните места във файла с изходен код. Затова трябва да използваме специални команди. Средата, която ни го позволява е **tabbing** (Фигура 10):

- Създаваме първата линия на една таблица с помощта на командата `\=;`



Фигура 10: Tabbng

- Следващите линии се означават с командата `\>`;
- Края на една линия се означава с командата `\&`.

```

\begin{tabbing}

Номер \= Дата на издаване \= Основна тема \\
18    \> Юли 2003           \> Интернет \\
19    \> Юли 2003           \> Електронната поща \\
20    \> Юли 2003           \> Код на HTML \\

\end{tabbing}

```

Средата **tabbing** не позволява представянето на текст и данни под формата на таблица. Затова ЛaTeX предлага също средата **tabular**. Служим си с предния пример за да създадем таблица:

```

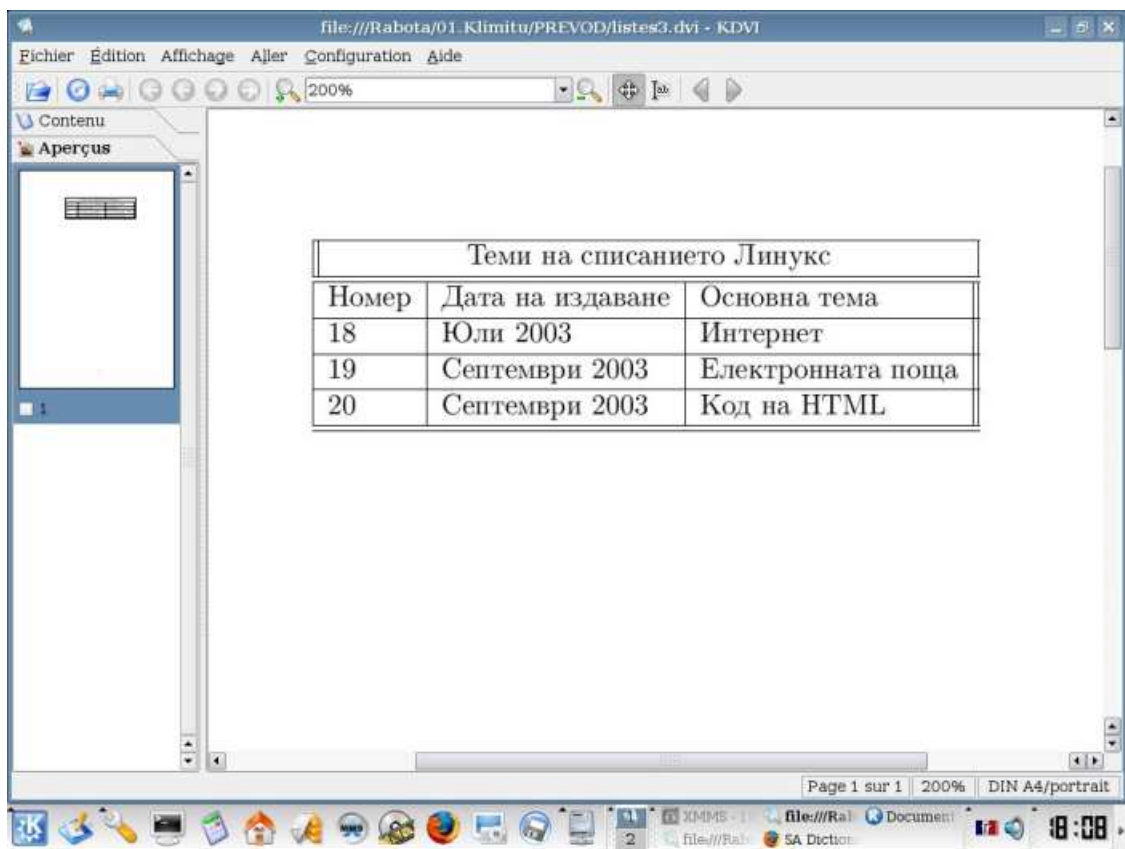
\begin{tabular}{t}{|l|l|}
\hline
\multicolumn{3}{|c|}{Теми на списанието Линукс} \\
\hline
\hline
Номер & Дата на издаване & Основна тема \\
\hline
18 & Юли 2003 & Интернет \\
\hline
19 & Септември 2003 & Електронната поща \\
\hline
20 & Септември 2003 & Код на HTML \\
\hline
\hline
\end{tabular}

```

Преди да обясним подробно таблицата, да се порадваме на резултата (Фигура 11).

Таблицата е в среда **tabular**, или по-ясно между командите `\begin {tabular }` [разположение] `{колони}` и `\end {tabular}`. Опцията `position` позволява да уточним разположението на таблицата по отношение на реда, на който се намираме:

- **t** (top) - горната част на таблицата да е на реда, на който се намираме;
- **b** (bottom) - долната част на таблицата да е на реда, на който се намираме;
- **c** (center) - за да разположим таблицата в средата.



Фигура 11: Tabular

Втория параметър `colones` определя броя на колоните, както и тяхното подреждане. Тя може да съдържа следните елементи:

- **r** - текста в колоната ще е подравнен в дясно;
- **l** - текста в колоната ще е подравнен в ляво;
- **c** - текста в колоната ще е двустранно подравнен (в средата);
- **|** - вертикален разделител.

В примера, който току що видяхме, `\begin{tabular}[t]{|l|l|l|}` дефинира таблица с три колони, в които текста ще се намира в ляво и ще е разделен с единична вертикална линия, а от външната страна таблицата ще е с две вертикални линии.

Таблицата, сама по себе си, може да съдържа следните елементи :

- **&** - разделя колоните една от друга;
- **\\** - край на една линия;
- **\hline** - хоризонтална линия по широчината на таблицата.

Широчината на колоните се определя от най-широката колона.

Командата `\multicolumn{брой}{подравняване на текста}{текст}` дефинира подравняването и съдържанието на текста, както и броя на колоните, в които ще е разположен. Тази команда се използва за поставяне на заглавие на таблицата.

## 13 Още повече практика

Сега, след като разбрахте принципа на работа с таблиците ...направете малка почивка. Може да измиете колата, да боядисате оградата в градината или пък да нахраните червеношийките, накацали по нея.

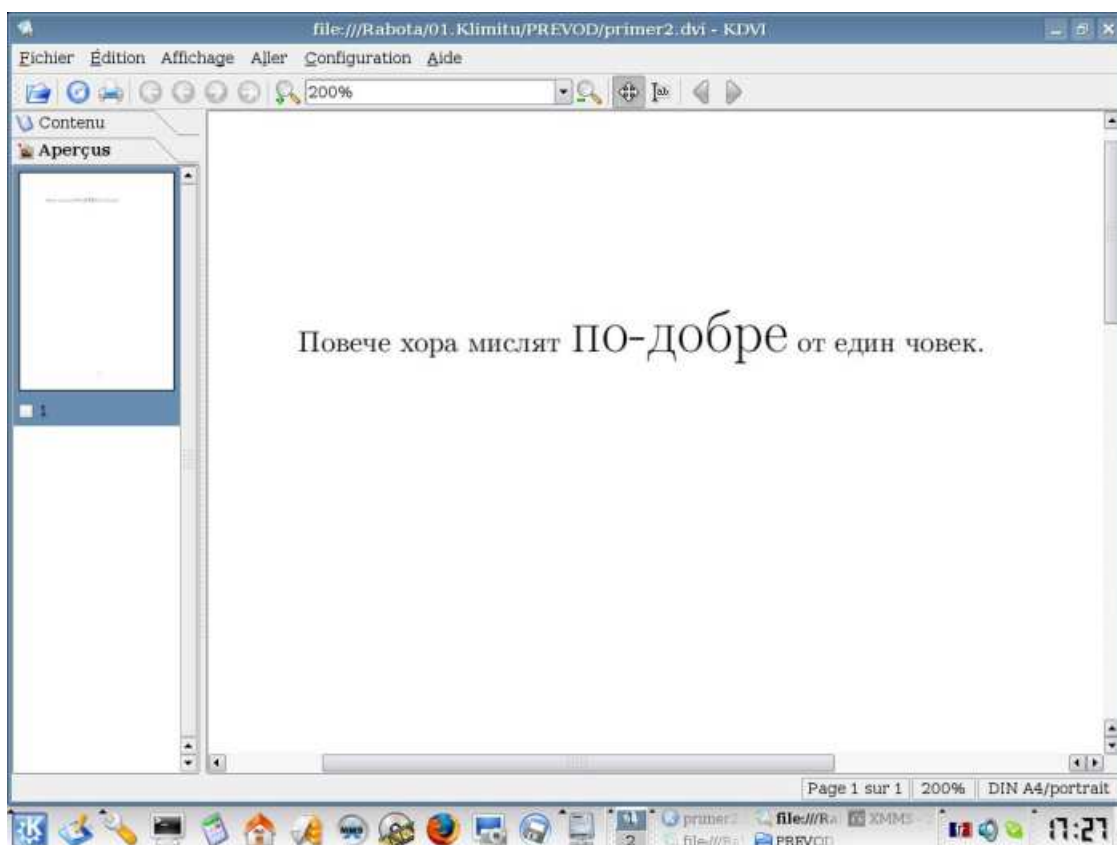
После, веднъж като се разведриете, може да се върнете към ЛaTeX. Намерете някоя таблица из документите ви, във вестника или в Интернет, и се опитайте да я направите, спазвайки правилата, които видяхме по-горе. Най-вече, не се страхувайте да експериментирате.

## 14 С характер!

Освен ако не е пояснено в опциите на `\documentclass`, ЛaTeX използва шрифта „Roman“ с големина 10 пункта<sup>2</sup> за всички документи. Това може да се промени както за думи, така за група думи, части от текста, или за целия документ.

---

<sup>2</sup>**пункт** = печатарска мерна единица. В Европа: 1 пункт на Дидо = ок.0.376мм, 1 мм = 0.266 пункта, 12 пункта = 1 цицер; в САЩ: 1 пункт = 1/72 инча = ок. 0.351 мм, 12 пункта = 1 пайк.



Фигура 12: Различна големина на текста

Смяната на големината на текста в самия документ се извършва с една обратна наклонена черта, следвана от големината на шрифта. Ако тази промяна се отнася само за една или няколко думи (Фигура 12), пишем командата и текста, за който тя се отнася в кърдрави скоби:

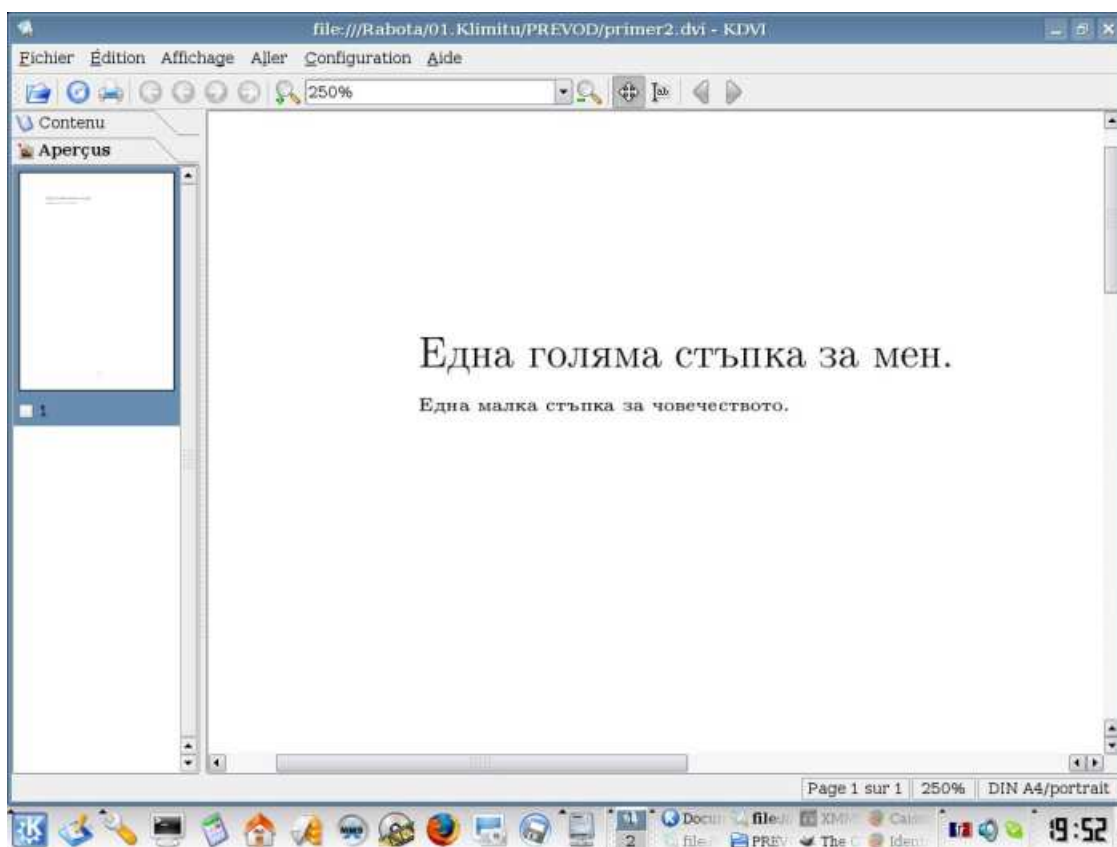
Повече хора мислят  $\{\backslash Huge\}$  по-добре от един човек.

Големината на текста може също да бъде дефинирана като среда (Фигура 13):

```
\begin{large}
Една голяма стъпка за мен.
\end{large}

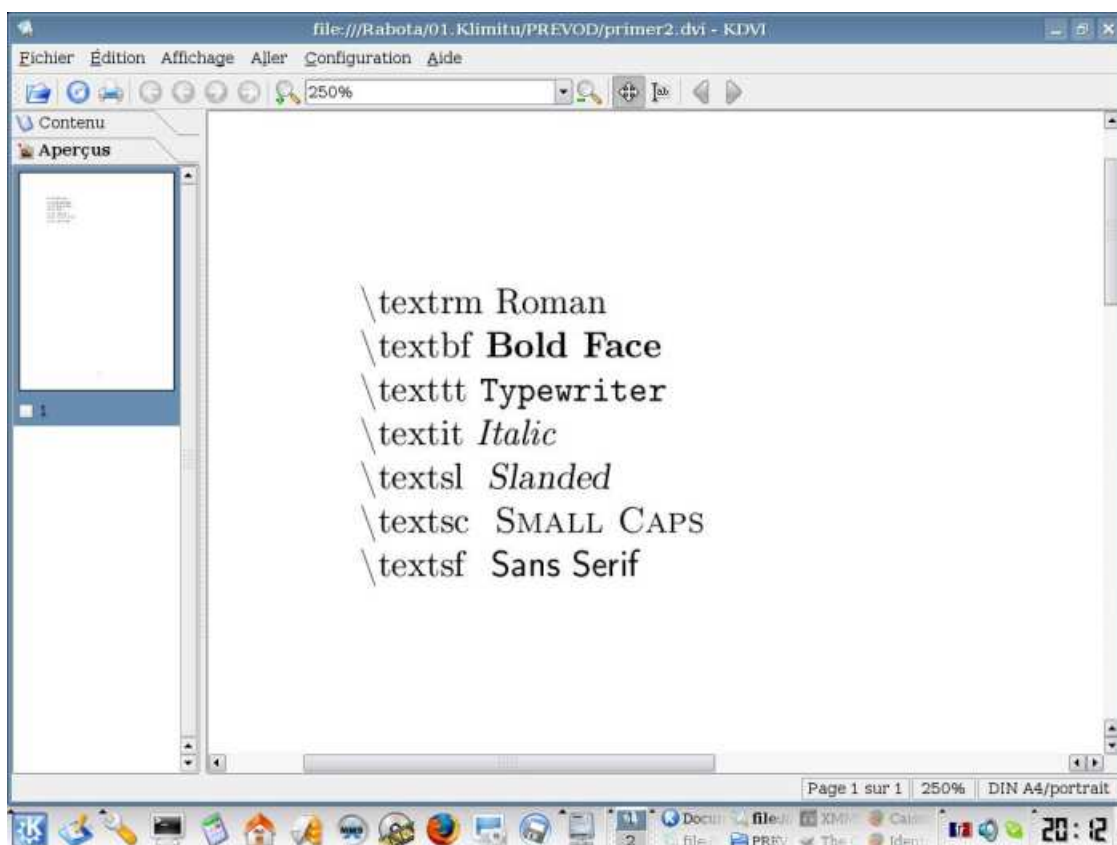
\begin{tiny}
Една малка стъпка за човечеството.
\end{tiny}
```

Възможните големина на текста са :  $\backslash tiny$  (миниатюрен),  $\backslash scriptsize$  (много малък),  $\backslash footnotesize$  (по-малък),  $\backslash small$  (малък),  $\backslash normalsize$  (нормален),  $\backslash large$  (голям),  $\backslash Large$  (по-голям),  $\backslash LARGE$  (много голям),  $\backslash Huge$  (огромен) и  $\backslash HUGE$  (гигантен).



Фигура 13: Различна големина на текста





Фигура 14: Различни видове шрифт

За да се дефинира големината на текста за целия документ, това трябва да се посочи в началото на документа. Може да избирате между 10, 11 и 12 пункта. Пояснете големината в скоби преди типът на документа, както е показано по-долу:

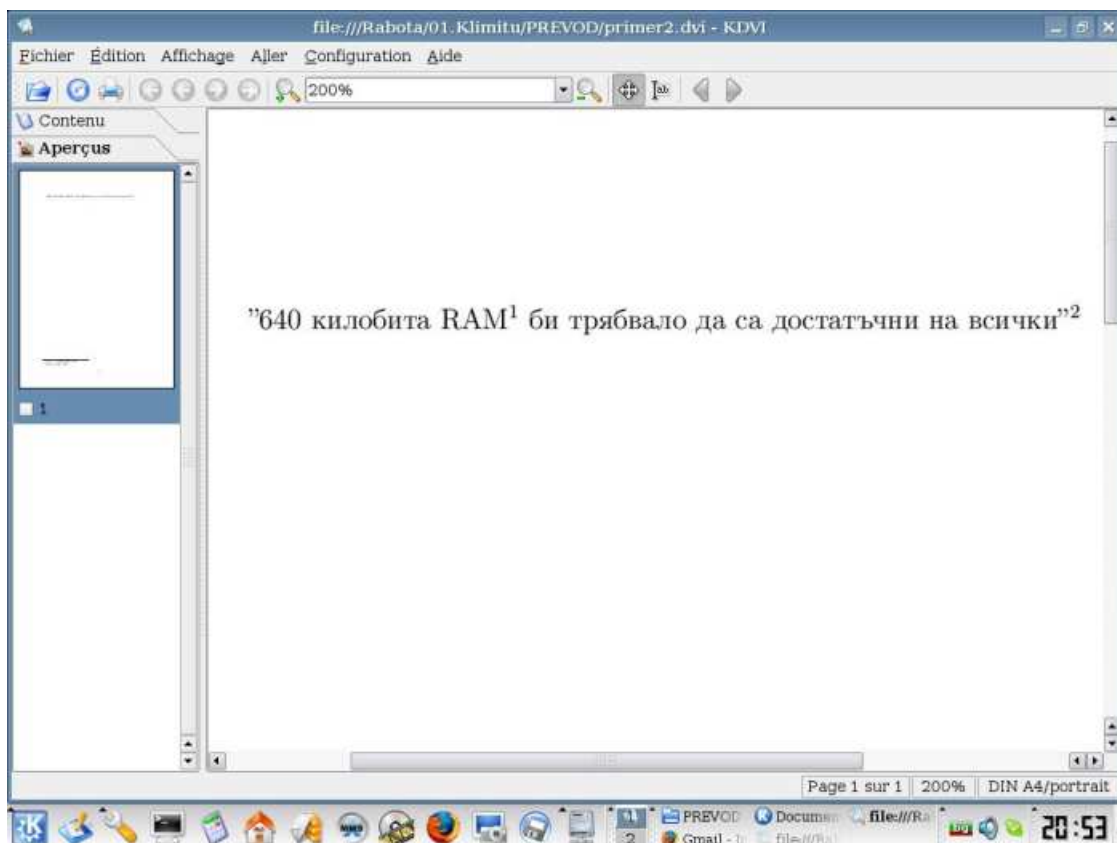
```
\documentclass[12pt]{article}
```

ЛаТеК предлага различни видове шрифт, между които може да избирате (Фигура 14).

За да наблегнете на някоя дума избягвайте да използвате курсивен шрифт, както може би сте свикнали с текстовите редактори. Командата под ЛаТеК за подчертаване на определена дума е `\emph{текст}`, с аргумент текста, който искате да подчертаете. По принцип, курсивния шрифт се използва за наблягане върху даден текст, освен ако текста не е вече курсивен. В този случай, ЛаТеК използва нормален шрифт.

## 15 Забележките

Забележките (към текст под линия) се дефинират чрез командата `\footnote {текст}`:



Фигура 15: Горната част на листа

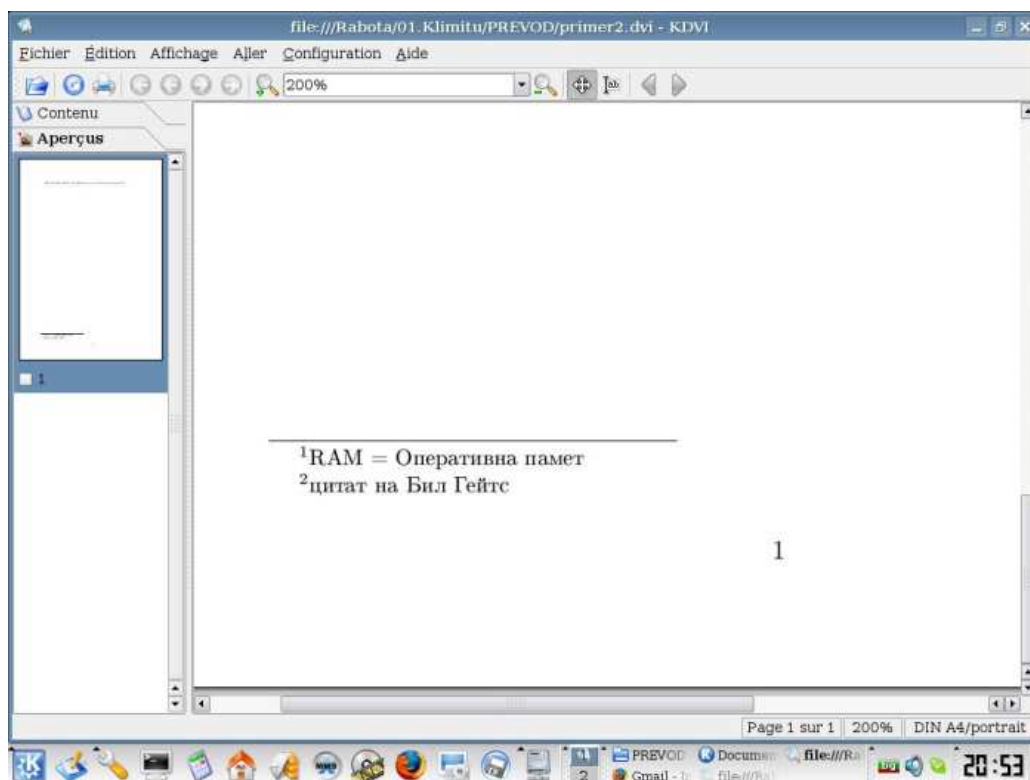
"640 килобита RAM\footnote {RAM = Оперативна памет} би трябвало да са достатъчни на всички"\footnote {цитат на Бил Гейтс}

Ако използвате типа документ **article**, забележката е в долната част на страницата, с най-малкия шрифт и номерирането е автоматично в целия документ (Фигура 16). Ако типа документ е **book** или **report**, номерирането на забележките започва от едно в началото на всяка нова глава. Можете да промените използвания шрифт на всяка забележка по следния начин :

\footnote {Може да използвате \textit{курсивен} и \textbf{удебелен} шрифт.}

## 16 Наслуки

Документите ЛaTeX позволяват дефинирането на кръстосани референции към определени части от текста, което много улеснява връзките към другите глави или определени страници. Една такава кръстосана референция се състои от два елемента: марка и референция.



Фигура 16: Забележка в долната част на листа

За да се дефинира една марка се използва командата `\label{име}`, където името е едно определено указание като например *пример3*, *нещо2*, *графика1* и така нататък.

Командата `\ref{име}` ще ви даде номера на съответната глава. Що се отнася до командата `\pageref{име}`, тя дава номера на съответната страница.

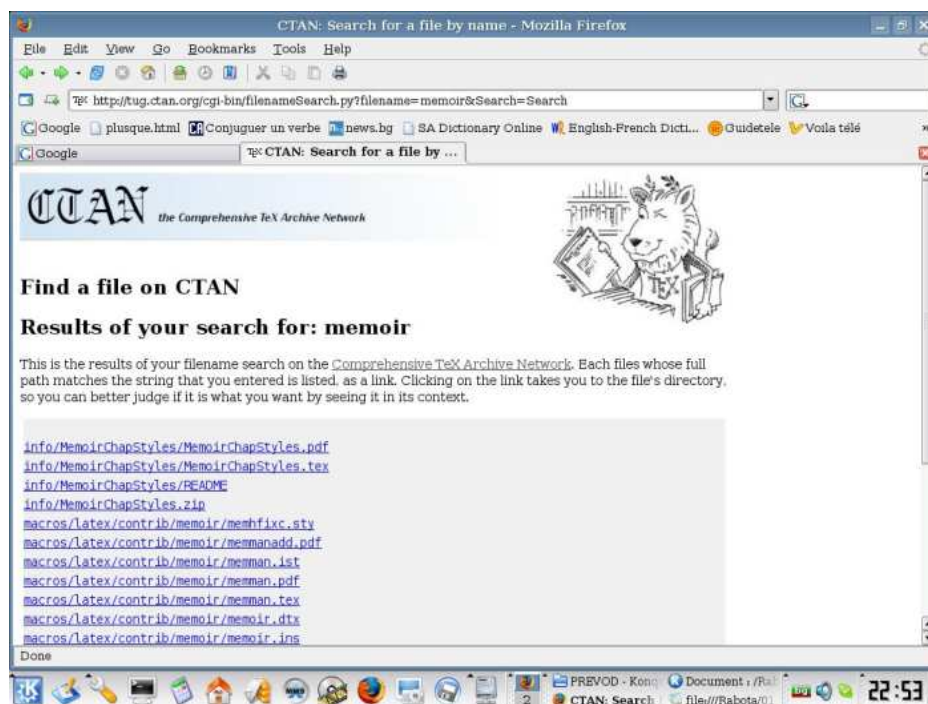
Един пример. Да използваме дипломата ни за фонетичното развитие на римския език (пример по-горе). Да приемем, че една от главите се казва :

```
\subsection {Лабиализирането на веларните съгласни през IIIти век\label{съгласни}}
```

Сега може да се реферанцирате към тази подглава и номер на страница, използвайки командите :

...по темата прочетете глава`\ref{съгласни}`, страница `\pageref{съгласни}`, относно лабиализирането на веларните съгласни през IIIти век.

Важно е да отбележим, че ако използвате кръстосани референции, без съмнение ще е нужно да компилирате два пъти вашия файл с изходен код. С първата компилация ЛaTeX създава помощен файл, за да може по-лесно да намира всички тези референции, които ще се появят чак при второто компилиране.



Фигура 17: «<http://tug.ctan.org/>»

Тази функция ще е много полезна на всички нещастни писатели на дипломни работи или докторати, които трудно се ориентират в документа си, след като вече са прибавили няколко допълнителни страници тук таме. Същото се отнася и за една друга практична функция на ЛaTeX : автоматичното създаване на съдържанието. Ако вземете какъвто и да е структуриран документ, който съдържа известен брой глави, подглави и секции, и след началото на документа напишете командата `\tableofcontents...`като го компилирате поне два пъти. Изненада?

## 17 Класата MEMOIR

През годините, ЛaTeX е доста подобрен и допълнен. Между тези подобрения, трябва да отбележим типа документ **memoir**, написан от Питър Уилсън.

Това е хибридна класа (тип документ) между **book**, **article** и **report**, която предлага известен брой допълнителни функции, много полезни при писането на докторати, новели, романи и други. Едно по-подробно представяне би било в повече за тази статия, но все пак кратко ще я разгледаме.

Класата **memoir** не се съдържа в стандартната конфигурация на ЛaTeX : трябва допълнително да се изтегли от Интернет. Това става като отидете на веб-страницата «<http://tug.ctan.org/search.html>» и напишете memoir в полето за търсене. Това ще ви отвори страницата с архивите на **memoir** (Фигура 17).

За да инсталирате класата **memoir**, трябва първо да изтеглите файловете *memoir.dtx*, *memoir.ins*, *mempatch.sty* и *memhfxc.sty*. Запишете ги във вашата директория на потребител или по-добре създайте една под-директория  $\tilde{\text{memoir}}$ . След като изтеглите файловете, отидете в директорията където се намират и изпълнете една след друга следните команди :

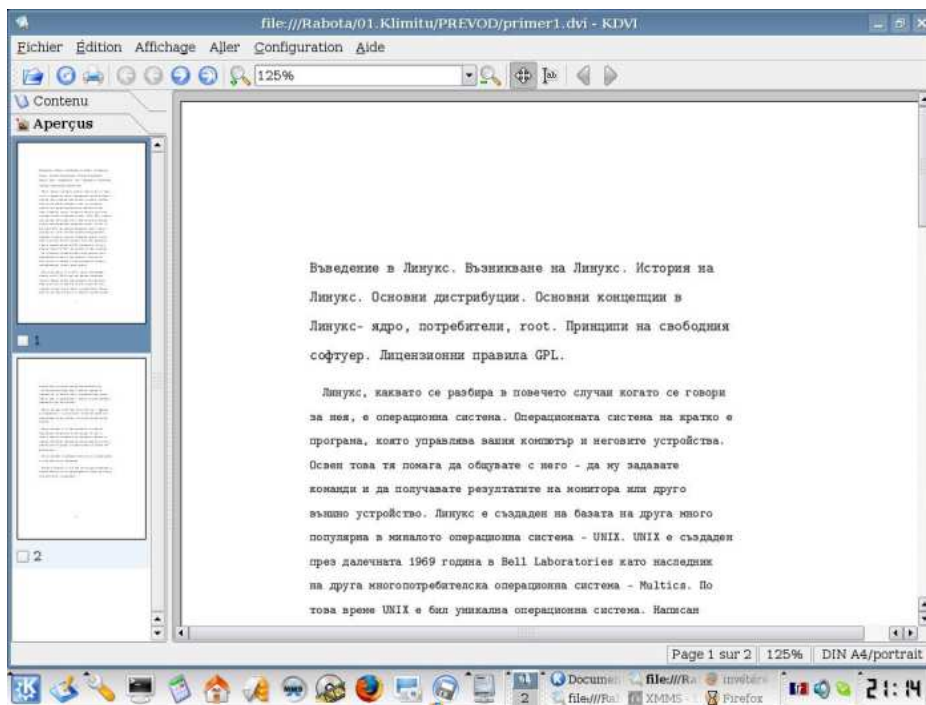
1. `\$ latex memoir.cls` (Тази команда създава `memoir.cls` и серия файлове `mem*.clo`)
2. `\$ latex memoir.dxt`
3. `\$ makeindex -s gind.ist memoir`
4. `\$ latex memoir.dxt`
5. За да видите упътването на английски : `\$ xdvi memoir.dvi` (Може евентуално да го отпечатате с помощта на командата, обяснена в началото на тази статия `dvips`.)
6. `\$ su`
7. `\# mkdir /usr/share/texmf / tex/latex/memoir`
8. `\# mv mem* /usr/share/texmf / tex/latex/memoir/` (Така слагаме всички тези файлове на място, където ЛaTeX лесно ще ги намери.)
9. `\# exhash` (С тази команда извършваме допълването с последните данни на библиотеките на ЛaTeX.)
10. Станете отново „начинаещ“ потребител.

Сега може да пробвате инсталирането. За да го направите изтеглете пълното ръководство на **memoir**, *memman.tex*, което е съставено в класата на документа **memoir**, и го компилирайте с командата **latex**.

Между практическите функции на **memoir** най-вече ни интересуват опциите на класата. Да си припомним, че **book**, **article** и **report** предлагат като големина на шрифта само 10, 11 и 12 пункта. Тук, може да дефинираме следните големина : 9, 10, 11, 12, 14 и 17 пункта. Пробвайте ги като използвате файла *nouvelle.tex* и дефинирайки типа документ, както следва :

```
\documentclass[a4paper, 17pt]{memoir}
```

Докато типа `book` ни задължаваше да „лъжем“ малко или много с командите за големината на шрифта като `\Large` или `\Huge`, за да избегнем „завъртулките“, **memoir** доста ни улеснява.



Фигура 18: Опцията ms

Както видяхме по-горе, съобщенията за грешка ни позволяват да набележим спасителната вратичка, в случай на нужда. Опцията `draft` (между другото на разположение и със стандартните типове документи на `LaTeX`) позволява да идентифицираме известните „overfull hbox“ с един поглед. Още веднъж използвайте файла `nouvelle.tex`, изтрийте всички пояснения за пренасянето на думите и го прекомпилирайте със следните опции :

```
\documentclass[a4paper,draft]{memoir}
```

Сега, всяка линия, която излиза от параграфа е отбелязана с черно каре, което позволява бързото идентифициране на евентуални проблеми с оформянето на страницата. Ако не желаете този тип предупреждение, заместете опцията `draft` с `final` или просто не слагайте нищо, защото това е опцията по подразбиране. Редовните потребители на `LaTeX` сигурно вече са срещали този проблем под една или друга форма. Тъкмо сте завършили оформянето на вашия документ, за да го изпратите на издателя ви и той ви го връща, искайки „файл Microsoft Word както всички останали“.

За да удовлетвори тези искания, Питър Уилсън е включил опцията `ms` (като `manuscript=ръкопис`). Използвайте за последен път файла `nouvelle.tex` и го прекомпилирайте с тази опция (Фигура 18).

```
\documentclass[a4paper,ms]{memoir}
```

За ваше най-голямо учудване, документът ви като че ли е написан на стара пишеща машина. Сега остава само да отпечатате този „ръчно направен“ документ и да го изпратите на издателя ви, уточнявайки, че не разполагате с Microsoft Word, но сте си направили труда да препишете романа си на старата си пишеща машина и за нещастие това е всичко, което сте могли да направите :o) Класата **memoir** предлага множество функции за създаването на писания от всякакъв тип, описани подробно в ръководството на дистрибуцията. Тя съдържа функциите на другите големи класи **book**, **article** и **report**.

Така можете да я използвате за редактирането на всичките ви писания. По този повод посетете страницата на Андриан Хийткот («<http://teaching.arts.usyd.edu.au/philosophy/1002/>») и изтеглете няколко от предложените текстове (върху най-различни теми като християнството, пънка и научната фантастика).

Освен интересното им съдържание, тези текстове дават идея за всичко, което може да бъде направено с ЛaTeX и с класата **memoir**.

## 18 За повече информация

Документацията и ръководствата за ЛaTeX е последното, което липсва. Но за един начинаещ съществува обратния проблем. Ако имате непредазливостта да потърсите „добри самоучители за начинаещи“ на `comp.text.tex` ще се сдобие с няколко хиляди страници, само като за начало. Затова ще ви препоръчам само едно ръководство, за да получите повече информация. Една малка книжка от 116 страници, която се казва „Не много кратко въведение в L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>“, написана от Тобиас Отикер, Юбер Партл, Ирен Хина и Даниел Флипо. Можете да я намерите на български на адрес «<http://tug.ctan.org/cgi-bin/ctanPackageInformation.py?id=lshort-bulgarian>», преведена от Стефка Караколева.

Препоръчвам ви го, защото това е един документ, който ще консултирате редовно.

Кики Новак  
<http://www.kikinovak.net>

Превод: Климентина Йорданова