

L^AT_EX: Sistem za pripremu dokumenata

Vjeran Hari, Diana Šimić

Sadržaj

1 Polazni pojmovi	2
1.1 Što je L ^A T _E X	2
1.2 Ulazna datoteka (<i>input file</i>)	2
1.3 Stil dokumenta	2
1.4 Kako pokrenuti L ^A T _E X	3
1.5 Specijalni znakovi	3
1.6 Odvajanje prazninama	5
1.7 Komentari u ulaznoj datoteci	6
1.8 Izgled i veličina pisma	7
1.9 Kreiranje specijalnih znakova	9
2 Slaganje okolina	9
2.1 Okolina za centriranje teksta	10
2.2 Lijevo i desno pozicioniranje teksta	10
2.3 Okoline za pravljenje lista	11
2.3.1 Obična lista	11
2.3.2 Lista s nabranjanjem	11
2.3.3 Opisna lista	13
2.3.4 Imenovana lista	14
2.4 Okoline za citiranje	15
2.5 Okolina za pisanje poezije	16
2.6 Okolina za doslovni prikaz ulazne datoteke	18
3 Slaganje matematičkog teksta	19
3.1 Jednostavni matematički tekst	19
3.1.1 Slova, jednostavni izrazi i jednakosti	20
3.1.2 Matematički simboli	22
3.2 Složeniji matematički tekst	22
3.2.1 “Array” okolina	22
3.2.2 Višelinijnske jednadžbe	24
3.2.3 Specijalni efekti	25
4 Tabele	26
4.1 “Tabbing” okolina	26
4.2 “Tabular” okolina	29
4.3 “Table” i “Figure” okoline	30

5	Fusnote i reference	31
5.1	Fusnote	31
5.2	Brojači i ukrižene reference	31
6	Posebne operacije	34
6.1	Definiranje makro naredbi — naredbe “newcommand” i “renewcommand”	34
6.2	Naredbe za definiranje okolina “newenvironment” i “newtheorem”	35
A	Osnove rada računala	36
A.1	Dijelovi računala	36
A.2	Programska podrška	38
A.3	Osnove operacionog sustava MS-DOS	41
A.4	Upotreba programa \LaTeX pod MS-DOSom	44
B	Podsjetnik za Norton Editor	44
C	Tablice posebnih i matematičkih oznaka	47
D	Tablica ASCII znakova	47

1 Polazni pojmovi

1.1 Što je \LaTeX

Sistem za slaganje teksta \LaTeX je posebna verzija programa za slaganje teksta \TeX Donalda Knutha. \TeX je vrlo složeni program, a \LaTeX dodaje \TeX u niz naredbi koje pojednostavljaju pripremanje dokumenata i omogućavaju korisniku da se usredotoči na sadržaj i željeni oblik teksta, a ne na naredbe za slaganje. Međutim pri korištenju \LaTeX a ostaje nam i dalje mogućnost korištenja i svih naredbi \TeX a.

1.2 Ulazna datoteka (*input file*)

Tiskani dokument načinjen pomoću \LaTeX a ima svoju osnovu u ulaznoj datoteci koja sadrži tekst dokumenta i \LaTeX ove naredbe za slaganje teksta. Ulazna datoteka se priprema pomoću bilo kojeg programa za obradu teksta koji sprema (odnosno ima mogućnostspremiti) datoteku u ASCII formatu. Neki programi za obradu teksta zovu ASCII datoteku “tekst datoteka” ili “neformatirana” datoteka (*plain, plain text, nonformatted, . . .*). Ime ulazne datoteke mora imati nastavak “.tex”, npr. *doc.tex* ili *xx.tex*. Zbog čestog spominjanja, u daljem tekstu ćemo ulaznu datoteku kraće zvati ulaz. Npr. rečenica “Na ulazu ćemo pisati . . .” će značiti “U ulaznoj datoteci ćemo pisati . . .”.

1.3 Stil dokumenta

Prvi redak ulazne datoteke mora započeti naredbom `\documentstyle` koja odabire jedan od četiri stila dokumenta: *article* (članak), *book* (knjiga), *report* (izvještaj) i *letter* (pismo). Ako npr. želimo koristiti stil *article* prvi redak na ulazu će biti `\documentstyle{article}`. Za druge stilove će se `{article}` zamijeniti s `{book}`, `{report}` ili `{letter}`. Uz odabrani stil, dokument će biti složen prema njegovim specifičnim pravilima.

Prije tipkanja bilo kakvog teksta koji ulazi u dokument moramo iza naredbe `\documentstyle` napisati još

```
\begin{document}
```

a nakon ispisa zadnjeg retka dokumenta moramo još napisati

```
\end{document}
```

Ove dvije naredbe ukazuju programu \LaTeX na početak i kraj obrade (tj. slaganja) dokumenta.

Najlakše je složiti najobičniji tekst. Njega treba natipkati na standardan način (uz iznimku specijalnih slova kao što su ć, č, đ, š, i ž) i smjestiti između `\begin{document}` i `\end{document}`.

1.4 Kako pokrenuti \LaTeX

Neka je naziv ulazne datoteke *dat1.tex*. \LaTeX pokrećemo pisanjem (u okruženju operativnog sistema)

```
latex dat1 <return>
```

Kada se prvi puta pokreće \LaTeX na nekoj ulaznoj datoteci stvaraju se pomoćne datoteke s nastavcima “.aux” i “.log” te datoteka s nastavkom “.dvi” (u gornjem primjeru: *dat1.aux*, *dat1.log* i *dat1.dvi*). U datoteci s nastavkom “.aux” nalaze se korisne informacije vezane uz reference i labele koje se koriste u dokumentu. U datoteci s nastavkom “.log” (ili “.lis”) su pohranjene sve poruke koje \LaTeX sistem daje za vrijeme obrade, a koje su vidljive i na ekranu u toku slaganja teksta. Nakon što se obrada datoteke završi, kao glavni rezultat slaganja dobijemo datoteku s nastavkom “.dvi” (pokrata za *device independent file*). Ova datoteka sadrži elemente slike (tj. konačnog izgleda) dokumenta i pomoću posebnog programa (tzv. *preview program*) možemo konačni izgled dokumenta vidjeti na ekranu. Također pomoću posebnog programa možemo dokument ispisati na pisaču. Kao što ulaznu datoteku kraće zovemo ulaz tako ćemo pod izlazom podrazumijevati konačni izgled dokumenta kakav bi izgledao ispisan na papir ili prikazan *preview programom* na ekranu.

1.5 Specijalni znakovi

- Komandni znakovi

Već smo rekli da se osim teksta samog dokumenta u ulaznoj datoteci nalaze i \LaTeX ove naredbe za slaganje teksta. Stoga treba obratiti pažnju na slijedeće “komandne znakove”:

`\` tzv. *backslash* koji se koristi kao početak većine \LaTeX ovih naredbi

`$` označava početak i kraj matematičkog načina (dijela teksta u kojem se koriste posebni matematički simboli)

`&` odvađa stupce u tabelama

`%` služi za komentare unutar ulazne datoteke jer tekst koji slijedi `%` u nekom retku \LaTeX ignorira

1.6 Odvajanje prazninama

- Razmaci između riječi i rečenica

U ulaznoj datoteci između riječi pišemo jedno ili više praznih mjesta (bjelina). Više bjelina između riječi neće na izlazu dati više nego jednu bjelinu. Da bi na izlazu dobili između riječi jednu bjelinu više, moramo na ulazu utipkati `_` (gdje znak `_` iza znaka `\` označava jednu bjelinu). Npr. ako utipkamo ‘`‘Molim_dodajte_mi_olovku’`’, dobit ćemo: “Molim dodajte mi olovku”. Ako hoćemo trostruke bjeline zamijenit ćemo `_` sa `__`. Za bolju kontrolu horizontalnog pomaka stoji nam na raspolaganju naredba `\hspace{n}`, gdje `n` specificira udaljenost. Npr. ako na ulazu pišemo

```
Danas \hspace{18mm} je lijepi \hspace{1cm} dan.
```

dobit ćemo

```
Danas           je lijepi           dan.
```

Kao mjeru udaljenosti možemo uzeti mm (milimetar), cm (centimetar), in (inch), pt (“point” — štamparska točka), pc (pica), ex (širina slova x) i em (širina slova M). Pri tom vrijedi odnos $1\text{in} = 25.4\text{mm} = 72.27\text{pt}$. Argument “`n`” u `\hspace{n}` može biti i negativan i tada će pomak biti u lijevu stranu. Npr. u rečenici

```
Danas \hspace{-3mm} je lijepi\hspace{1cm} dan.
```

dobit ćemo

```
Danasje lijepi           dan.
```

Riječi se mogu pomaknuti istovremeno prema lijevoj i desnoj margini pomoću naredbe `\hfill`. Npr. redak

```
lijevo \hfill desno
```

će dati

```
lijevo                                                    desno
```

Kad se miješaju kosi i normalni (tj. ne kosi) znakovi dodaje se malo bjeline da prijelaz bude ljepši, npr. ako na ulazu pišemo

```
Bila je to {\em velika uvreda} pa joj nije oprostio.
```

dobit ćemo

```
Bila je to velika uvreda pa joj nije oprostio.
```

dok drugi način

```
Bila je to {\em velika uvreda\/} pa joj nije oprostio.
```

daje

```
Bila je to velika uvreda pa joj nije oprostio.
```

Komanda `\/` dodaje malo bjeline nakon kosog znaka dok ne dodaje ništa nakon normalnog znaka.

Nakon točke obično se pri slaganju teksta ostavlja nešto veća praznina koja označava kraj rečenice. \LaTeX pretpostavlja da točka završava rečenicu osim ako slijedi iza velikog slova. To uglavnom daje dobre rezultate osim kod izuzetaka, na primjer kod “itd.” U takvom slučaju mora se nakon točke nadopisati `\` (“`itd.\`”) i to tako da `\` slijedi odmah nakon

točke (dakle ne ‘ ‘itd. \ ’’ već ‘ ‘itd.\ ’’). U rijetkim slučajevima točka koja završava rečenicu slijedi iza velikog slova. Da bi L^AT_EX u rekli da ta točka završava rečenicu, ispred točke se napiše \@. Na primjer ako želimo na izlazu dobiti

Rimljani pišu $I + I = II$. Zaista!

na ulazu ćemo napisati

Rimljani pi\v su $I + I = II$ \@. Zaista!

Kad točka koja završava rečenicu slijedi iza desne zagrade ili navodnika, slovoslagari obično dodaju malo praznine. U takvim slučajevima treba Texu malo pomoći. Dodatni razmak se obično stavlja i iza ?, ! i : . Za malo razmaka možete upotrijebiti naredbu \,. Ako želite spriječiti da se dvije riječi razdvajaju u dva retka utipkajte ~ između njih, npr. Mr~Jones, Poglavlje~{12} će na izlazu dati Mr Jones, Poglavlje 12 i neće razdvajati susjedne riječi u dva retka. Sam znak ~ proizvodi na izlazu jednu bjelinu.

- Razmaci između redaka i odlomaka

Prelazak u novi red (bez uvlačenja tj. indentacije) se može nametnuti naredbom \\. Za malo više vertikalnog razmaka može se koristiti i \\[n] gdje “n” označava razmak kao u \hspace{n} naredbi. Prelaz u novi red na ulazu ne znači isto na izlazu pa na ulazu ne treba cijepati riječi koje su se našle blizu desne margine. Međutim prazni redak na ulazu znači početak novog odlomka, koji normalno započinje s uvlačenjem. Više praznih redaka na ulazu neće proizvesti više praznih redaka na izlazu već će samo započeti novi odlomak. Dodatni vertikalni razmak ćete proizvesti naredbom \vspace{n} gdje “n” ima isti smisao kao u \hspace{n} naredbi. Ova naredba ne smije se koristiti unutar odlomka jer se inače mogu dogoditi čudne stvari. Ako na ulazu napišemo:

Ju\v cer je bio lijepi dan.

\vspace{3ex}

Ivana je zato i\v sla u grad.

\vspace{-2ex}

Danas je ru\v zan dan.

dobit ćemo na izlazu:

Jučer je bio lijepi dan.

Ivana je zato išla u grad.
Danas je ružan dan.

Uvlačenje retka može se nametnuti naredbom \indent i spriječiti naredbom \noindent. Uvijek, iza svake naredbe (ako ne završava s }) treba ostaviti prazno mjesto.

1.7 Komentari u ulaznoj datoteci

Ako želimo da se neki tekst pojavljuje u ulaznoj datoteci a ne u izlaznoj datoteci koristimo znak %. Svaki tekst koji se nalazi u jednom retku ulazne datoteke iza znaka % L^AT_EX će ignorirati pa se neće pojaviti u izlaznoj datoteci. Npr. ako u ulaznoj datoteci napišemo

```
% Sada slijedi ne\ v sto va\ v zno.
Za vrijeme posljednjeg rata % koji je bio grozan
stradalo je oko 200000 ljudi %vjerojatno i 300000
uglavnom civila.
```

```
dobit ćemo na izlazu ispis
```

```
Za vrijeme posljednjeg rata stradalo je oko 200000 ljudi uglavnom civila.
```

Ako dijelite ulaznu datoteku sa drugim osobama (npr. pišete zajednički rad pa datoteku šaljete pomoću elektronske pošte) vaši komentari će ih obavijestiti o vašim dvojbama ili razjašnjenjima koji nisu za javnost tj. za izlaznu datoteku.

1.8 Izgled i veličina pisma

Većinu teksta obično slažemo koristeći tzv. *roman* pismo. No ponekad želimo dio teksta naglasiti, pa tada upotrijebimo naredbu `\em` koju smo već vidjeli u primjerima u poglavlju 1.6. Ovom naredbom naredili smo \LaTeX u da umjesto uobičajenog pisma koristi *italic* pismo. U \LaTeX u imamo na raspolaganju šest pisama:

podebljano pismo	<code>{\bf podebljano pismo}</code>
<i>italic pismo</i>	<code>{\it italic pismo}</code>
glatko pismo	<code>{\sf glatko pismo}</code>
<i>ukošeno pismo</i>	<code>{\sl uko\ v seno pismo}</code>
UMANJENA VELIKA SLOVA	<code>{\sc umanjena velika slova}</code>
kao stroj za pisanje	<code>{\tt kao stroj za pisanje}</code>

Pritom drugi stupac predstavlja odgovarajući sadržaj u ulaznoj datoteci. Za izbor običnog pisma unutar okoline koja je složena drugim pismom koristimo naredbu `\rm`. Dakle, ako jedan ili više znakova (npr. slovo, riječ, rečenicu, odlomak ili više odlomaka) želimo napisati drugim pismom, jednostavno ispred prvog znaka napišemo zgradu `{` iza koje odmah slijedi komanda za vrstu pisma, a nakon zadnjeg znaka napišemo `}`. Npr. rečenicu:

Pomoću \LaTeX a možemo pisati **podebljanim**, **glatkim**, *italic* i *ukošenim* slovima, a posebno su zanimljiva UMANJENA VELIKA SLOVA ili slova sa starih pisacih strojeva

smo na ulazu napisali (do na točku koja završava rečenicu) na slijedeći način:

```
Pomo\ ' cu \LaTeX a mo\ v zemo pisati {\bf podebljanim}, {\sf glatkim},
{\it italic} i {\sl uko\ v senim} slovima, a posebno su zanimljiva
{\sc umanjena velika slova} ili
{\tt slova sa starih pisa\ ' cih strojeva}.
```

Istu rečenicu možemo napisati i ovako:

```
Pomo\ ' cu \LaTeX a mo\ v zemo pisati \bf podebljanim, \sf glatkim,
\it italic \rm i \sl uko\ v senim slovima, \rm a posebno su zanimljiva
\sc umanjena velika slova \rm ili
\tt slova sa starih pisa\ ' cih strojeva.\rm
```

Naime, komanda `\bf` prebacuje obično (roman) pismo u podebljano, zatim `\sf` prebacuje u glatko, pa `\it` u italic i tako dalje. Na kraju se pomoću komande `\rm` vraćamo u obično pismo. Doseg svake od ovih komandi je najmanja okolina² u kojoj su definirani (u našem slučaju to je cijeli dokument) ili do slijedeće komande za vrstu pisma.

Osnovna veličina pisma u L^AT_EXu je 10pt. Ako želimo drugu veličinu možemo ju zadati npr. kao opciju naredbe `\documentstyle` i to u obliku `\documentstyle[12pt]{article}` ili `\documentstyle [11pt]{article}`. Tada će bazična veličina znakova unutar cijelog dokumenta biti 12pt odnosno 11pt. Ako želimo u istom dokumentu koristiti slova različitih veličina možemo to postići naredbama za promjenu veličine pisma:

gen	<code>\tiny</code>	gen	<code>\normalsize</code>	gen	<code>\LARGE</code>
gen	<code>\scriptsize</code>	gen	<code>\large</code>	gen	<code>\huge</code>
gen	<code>\footnotesize</code>	gen	<code>\Large</code>	gen	<code>\Huge</code>
gen	<code>\small</code>				

Normalna veličina dobije se naredbom `\normalsize`, naredbom `\footnotesize` dobije se veličina slova u fusnotama, a veličina znakova u gornjim i donjim indeksima dobije se naredbom `\scriptsize`. Ove naredbe automatski postavljaju izgled pisma na *roman*, pa ako želimo drugi izgled pisma, moramo staviti naredbu za vrstu pisma *iza* naredbe za promjenu veličine. Npr. ako na izlazu želimo dobiti: “Ubodi ose znaju biti **jako bolni**”, na ulazu ćemo napisati: `Ubodi ose znaju biti {\large\bf jako bolni}`.

Koje će biti prave veličine znakova zavisi o bazičnoj veličini znakova zadanoj u naredbi `\documentstyle` i o naredbama za promjenu veličine pisma unutar dokumenta. Slijedeća tabela pokazuje pripadne stvarne veličine znakova na izlazu.

	10pt	11pt Opcija	12pt Opcija
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Ova tabela vrijedi za *roman* pismo. Za ostala pisma ne možemo automatski koristiti sve bazične veličine znakova. Koji kombinacije su dozvoljene zavisi o računalu na kojem radimo. Slijedeća tabela nam pokazuje kombinacije koje postoje na većini distribucija L^AT_EXa u računalima. Pritom slova P odnosno X označavaju da pripadna kombinacija postoji odnosno ne postoji. Slovo Z označava da dotična kombinacija ne postoji automatski na raspolaganju, ali se do nje može doći na zahtjev, pomoću `\newfont` komande, koja će biti objašnjena u slijedećoj točki.

²vidi slijedeću točku o okolinama

	<code>\it</code>	<code>\bf</code>	<code>\sl</code>	<code>\sf</code>	<code>\sc</code>	<code>\tt</code>
5pt	Z	Z	X	X	X	X
6pt	X	Z	X	X	X	X
7pt	P	Z	X	X	X	X
8pt	P	Z	Z	Z	Z	Z
9pt	P	P	Z	Z	Z	P
10pt	P	P	P	P	Z	P
11pt	P	P	P	P	Z	P
12pt	P	P	P	P	Z	P
14pt	Z	P	Z	Z	Z	Z
17pt	Z	P	Z	Z	Z	Z
20pt	Z	Z	Z	Z	Z	Z
25pt	X	Z	X	X	X	X

Uočimo, jedino za `\sc` pismo moramo koristiti `\newfont` komandu i za standardne bazične veličine znakova od 10pt, 11pt i 12pt.

1.9 Kreiranje specijalnih znakova

Kadkad nam zatrebaju neki specijalni (nematematički) simboli poput Ξ , ffl , \emptyset , \cdot . Ti znakovi se mogu kreirati na više načina. Jedan od njih je pomoću komande `\symbol{}`, pri čemu se unutar vitičastih zagrada napiše prvo jednostruki apostrof ' i zatim oktalni broj (tj. broj zapisan u brojevnom sustavu s bazom 8) znaka u ASCII kodnom sustavu. Npr. `\symbol{'100}` će dati @. Ako se koristi znak " umjesto ', tada treba napisati heksadecimalni (tj. u bazi 16) redni broj znaka u ASCII kodnom sustavu. Npr. `\symbol{"40}` će dati opet @. Kako smo mi ipak navikli na decimalni brojevni sustav, najjednostavnije je napisati unutar vitičastih zagrada samo decimalni broj bez apostrofa. Dakle umjesto `\symbol{'100}` možemo pisati `\symbol{64}`. U tabeli 14 u dodatku D je prikazan ASCII kodni sustav, pri čemu su redni brojevi znakova dani u decimalnom sustavu. Da bi u komandi `\symbol{}` lakše koristili oktalne i heksadecimalne brojeve u tabeli 15 su prikazani brojevi u rasponu 1-128 u decimalnom, oktalnom i heksadecimalnom sustavu.

Ako dokument zahtijeva često pojavljivanje nekog specijalnog znaka može se stvoriti nova komanda pomoću \LaTeX ove `\newcommand` komande. Najbolje mjesto za pisanje takve "makro" komande je između `\documentstyle` i `\begin{document}` komandi, jer će tada komanda vrijediti za cijeli dokument (inače bi vrijedila tek unutar najmanje okoline u kojoj je definirana i to tek od mjesta gdje je definirana). Npr. za kreiranje komande koja generira " trebamo u napisati

```
\newcommand{\la}{\char 92}
```

i zatim u dokumentu, na mjestu gdje želimo da se pojavi znak " napisati `\la`. U definiciji makro naredbe mogli smo umjesto `{\char 92}` napisati `{\char '134}` odnosno `{\char "5C}`.

2 Slaganje okolina

\LaTeX radi u sistemu organiziranom po okolinama. To znači da se ulazni neobrađeni tekst dijeli na manje cjeline koje su zasebno složene prema danim pravilima. Cijela ulazna datoteka je jedna cjelina koja je podvrgnuta pravilima naznačenim naredbom `\documentstyle` i naredbama u preambuli — dijelu ulazne datoteke između naredbi `\documentstyle` i `\begin{document}`. Ostale cjeline (okoline) nalaze se između naredbi `\begin{document}` i `\end{document}`. Tekst

2.3 Okoline za pravljenje lista

L^AT_EX podržava tri vrste lista:

- Obična lista (kao ova)
- Lista s numeracijom
- Opisna lista

2.3.1 Obična lista

Ova lista započinje s `\begin{itemize}` i završava s `\end{itemize}`. Gornja lista spada u ovu vrstu i načinjena je pomoću slijedećih naredbi:

```
\begin{itemize}
\item Obična lista (kao ova)
\item Lista s nabranjanjem
\item Opisna lista
\end{itemize}
```

2.3.2 Lista s nabranjanjem

Ova okolina automatski broji svaki član liste. Ako u ulaznoj datoteci zamijenimo naredbe `\begin{itemize}` i `\end{itemize}` s `\begin{enumerate}` i `\end{enumerate}` dobit ćemo na izlazu

1. Obična lista (kao ova)
2. Lista s nabranjanjem
3. Opisna lista

Uočimo na slijedećem primjeru kako se ova lista ugnježđuje.

1. Carstvo *Phyta*
 - (a) Koljeno *Euglenophyta*
 - (b) Koljeno *Chlorophyta*
 - i. Razred **Chlorophyceæ**
 - ii. Razred **Charophyceæ**
 - (c) Koljeno *Chrysophyta*
 - i. Razred **Xanthophyceæ**
 - ii. Razred **Chrysophyceæ**
 - iii. Razred **Bacillariophyceæ**
2. Carstvo *Animalia*
 - (a) Koljeno *Protozoa*
 - (b) Koljeno *Porifera*
 - (c) Koljeno *Mesozoa*

- (d) Koljeno *Arthropoda*
- i. Razred **Arachnida**
 - ii. Razred **Crustacea**
 - iii. Razred **Insecta**
 - A. Red ODONATA
 - B. Red ORTHOPTERA
 - C. Red LEPIDOPTERA

Ova lista je napisana u ulaznoj datoteci na slijedeći način (uvlačenje je korišteno zbog preglednosti):

```

\begin{enumerate}
  \item Carstvo {\em Phyta}
  \begin{enumerate}
    \item Koljeno {\em Euglenophyta}
    \item Koljeno {\em Chlorophyta}
    \begin{enumerate}
      \item Razred {\bf Chlorophyce\ae}
      \item Razred {\bf Charophyce\ae}
    \end{enumerate}
    \item Koljeno {\em Chrysophyta}
    \begin{enumerate}
      \item Razred {\bf Xanthophyce\ae}
      \item Razred {\bf Chrysophyce\ae}
      \item Razred {\bf Bacillariophyce\ae}
    \end{enumerate}
  \end{enumerate}
  \item Carstvo {\em Animalia}
  \begin{enumerate}
    \item Koljeno {\em Protozoa}
    \item Koljeno {\em Porifera}
    \item Koljeno {\em Mesozoa}
    \item Koljeno {\em Arthropoda}
    \begin{enumerate}
      \item Razred {\bf Arachnida}
      \item Razred {\bf Crustacea}
      \item Razred {\bf Insecta}
      \begin{enumerate}
        \item Red {\sc odonata}
        \item Red {\sc orthoptera}
        \item Red {\sc lepidoptera}
      \end{enumerate}
    \end{enumerate}
  \end{enumerate}
\end{enumerate}

```

L^AT_EX dozvoljava da se u dubinu ide do četiri nivoa.

2.3.3 Opisna lista

Ova okolina umjesto brojenja članova liste koristi opisne oznake. Stoga odlomci liste imaju pridružene oznake — kao naljepnice. Npr. ova lista

Tropske kišne šume Na kopnenim područjima koja leže blizu ekvatora temperatura samo malo varira cijele godine i tamo gdje kišne padavine iznose 225 cm ili više, nalaze se velike tropske kišne šume.

Travnjaci Sjeverno i južno od tropskih kišnih šuma nalaze se velika područja kopna zvana travnjaci.

Pustinje Gdje oborine na kopnenim područjima iznose manje od 25 cm godišnje nalaze se pustinje.

Bjelogorične šume U srednjim širinama, gdje padavine prelaze 75 cm godišnje, nalazimo bjelogorične šume.

Tajga Sjeverno od bjelogoričnih šuma i sjevernih travnjaka i pustinja nalazi se širok pojas poznat pod imenom tajga ili sjeverna crnogorična šuma.

Tundra Na sjevernim granicama tajge drveće postaje rastom manje i postepeno prelazi u golemo golo područje poznato pod imenom tundra.

je rezultirala iz slijedećeg niza naredbi u ulaznoj datoteci

```
\begin{description}
\item[Tropske kišne šume] Na kopnenim područjima
koja leže blizu ekvatora temperatura samo malo varira
cijele godine i tamo gdje kišne padavine iznose 225~cm
ili više, nalaze se velike tropske kišne šume.
\item[Travnjaci] Sjeverno i južno od tropskih
kišnih šuma nalaze se velika područja kopna
zvana travnjaci.
\item[Pustinje] Gdje oborine na kopnenim područjima
iznose manje od 25~cm godišnje nalaze se pustinje.
\item[Bjelogorične šume] U srednjim širinama, gdje
padavine prelaze 75~cm godišnje, nalazimo
bjelogorične šume.
\item[Tajga] Sjeverno od bjelogoričnih šuma i
sjevernih travnjaka i pustinja nalazi se širok pojas
poznat pod imenom tajga ili sjeverna crnogorična
šuma.
\item[Tundra] Na sjevernim granicama tajge drveće
postaje rastom manje i postepeno prelazi u golemo golo
područje poznato pod imenom tundra.
\end{description}
```

Uočimo kako se koriste uglate zagrade za navođenje oznake odlomka. Tekst unutar zagrada se automatski naglašava, osim ako se ne zahtijeva drugačije. Npr. `\item[{\it Travnjaci}]` bi oznaku odlomka ispisala u kurzivu.

2.3.4 Imenovana lista

Kadkad bismo željeli tekst odlomka dovoljno udaljiti od oznaka odlomaka, toliko da su sve oznake odvojene od odlomaka. Ili obrnuto, možda bismo željeli odlomke poravnati odmah do lijevog ruba. Takvu listu ćemo zvati imenovana lista. U preambuli treba prvo napisati:

```
\newcommand{\namelistlabel}[1]{\mbox{#1}\hfill}
\newenvironment{namelist}[1]{%
\begin{list}{
  {
    \let\makelabel\namelistlabel
    \settowidth{\labelwidth}{#1}
    \setlength{\leftmargin}{1.1\labelwidth}
  }
}{%
\end{list}}
```

a unutar dokumenta započeti listu naredbom `\begin{namelist}{n}` i završiti s `\end{namelist}` gdje “n” označava tekst čija duljina određuje razmak svih odlomaka do lijeve margine. Npr. ako na ulazu napišemo

```
\begin{namelist}{Tropske ki\v snexx}
\item[Tropske ki\v sne \v sume] Na kopnenim ...
\item[Travnjaci] Sjeverno i ju\v zno od tropskih ...
\item[Pustinje] Gdje oborine na kopnenim podru\v cjima ...
\item[Bjelogori\v cne \v sume] U srednjim \v sirinama ...
\item[Tajga] Sjeverno od bjelogori\v cnih \v suma i ...
\item[Tundra] Na sjevernim granicama tajge drve\v ce ...
\end{namelist}
```

dobit ćemo na izlazu

Tropske kišne šume Na kopnenim područjima koja leže blizu ekvatora temperatura samo malo varira cijele godine i tamo gdje kišne padavine iznose 225 cm ili više, nalaze se velike tropske kišne šume.

Travnjaci Sjeverno i južno od tropskih kišnih suma nalaze se velika područja kopna zvana travnjaci.

Pustinje Gdje oborine na kopnenim područjima iznose manje od 25 cm godišnje nalaze se pustinje.

Bjelogorične šume U srednjim širinama, gdje padavine prelaze 75 cm godišnje, nalazimo bjelogorične šume.

Tajga Sjeverno od bjelogoričnih šuma i sjevernih travnjaka i pustinja nalazi se širok pojas poznat pod imenom tajga ili sjeverna crnogorična šuma.

Tundra Na sjevernim granicama tajge drveće postaje rastom manje i postepeno prelazi u golemo golo područje poznato pod imenom tundra.

Da smo umjesto `{Tropske kišnexx}` napisali `{Tajgax}` i umjesto `\item[...]` koristili `\item[{\bf ...}]`, dobili bismo

Tropske kišne šume Na kopnenim područjima koja leže blizu ekvatora temperatura samo malo varira cijele godine i tamo gdje kišne padavine iznose 225 cm ili više, nalaze se velike tropske kišne šume.

Travnjaci Sjeverno i južno od tropskih kišnih suma nalaze se velika područja kopna zvana travnjaci.

Pustinje Gdje oborine na kopnenim područjima iznose manje od 25 cm godišnje nalaze se pustinje.

Bjelogorične šume U srednjim širinama, gdje padavine prelaze 75 cm godišnje, nalazimo bjelogorične šume.

Tajga Sjeverno od bjelogoričnih šuma i sjevernih travnjaka i pustinja nalazi se širok pojas poznat pod imenom tajga ili sjeverna crnogorična šuma.

Tundra Na sjevernim granicama tajge drveće postaje rastom manje i postepeno prelazi u golemo golo područje poznato pod imenom tundra.

Konačno, da smo listu započeli s `\begin{namelist}{}{}` dobili bismo

Tropske kišne šume Na kopnenim područjima koja leže blizu ekvatora temperatura samo malo varira cijele godine i tamo gdje kišne padavine iznose 225 cm ili više, nalaze se velike tropske kišne šume.

Travnjaci Sjeverno i južno od tropskih kišnih suma nalaze se velika područja kopna zvana travnjaci.

Pustinje Gdje oborine na kopnenim područjima iznose manje od 25 cm godišnje nalaze se pustinje.

Bjelogorične šume U srednjim širinama, gdje padavine prelaze 75 cm godišnje, nalazimo bjelogorične šume.

Tajga Sjeverno od bjelogoričnih šuma i sjevernih travnjaka i pustinja nalazi se širok pojas poznat pod imenom tajga ili sjeverna crnogorična šuma.

Tundra Na sjevernim granicama tajge drveće postaje rastom manje i postepeno prelazi u golemo golo područje poznato pod imenom tundra.

2.4 Okoline za citiranje

Kad želimo citirati tekst iz neke knjige, članka itd. koristit ćemo *quote* ili *quotation* okoline. Razlika je u tome što *quote* ne uvlači prvi redak svakog odlomka dok *quotation* baš to čini. Evo primjera. U knizi Heaven and Hell Emanuel Swedenborg piše:

1. In *Matthew 24*, the Lord is talking to His disciples about the end of the age, or the last time of the church. In concluding His predictions about the church's successive states of love and faith, He says,

Immediately after the misery of those days the sun will be darkened and the moon will not give her light, and the stars will fall from heaven, and the powers of the heaven will be shaken. And the sign of the Son of man will appear in heaven, and then all the tribes of the earth will mourn; and they will see the Son of man coming in the clouds of heaven with power and great glory. And He will send His angels with a trumpet and a great voice, and they will gather together His elect from the four winds, from the one end of the heavens to the other.

People who take these words literally can only believe that all these things are going to happen literally, just as described, in some final time called Last Judgement.

U ovom primjeru smo dva puta koristili naredbu `\begin{quote}`, drugi put smo iza te naredbe dodali naredbe `\small \it` za korištenje manjih slova u kurzivu. Doseg tih naredbi je unutar najmanje okoline u kojoj su definirane. To je opće pravilo u L^AT_EXu da naredba definirana unutar neke okoline ima doseg tek unutar nje.

Kao primjer citiranja pomoću okoline quotation koristimo jedan drugi izvod iz iste knjige.

The sun which will be darkened refers to the Lord in respect to love, the moon to the Lord in respect to faith. The stars refer to insights of what is good and true (or of love and faith), the sign of the Son of man in heaven to Devine truth becoming visible, and the coming of the Lord in the clouds of heaven with power and glory to his presence in the Word and to revelation. The clouds refer to the literal meaning of the Word, the glory to the Word's inner meaning, and the angels with trumpet and large voice to heaven, where devine truth come from.

On this basis we can determine what these words of the Lord would have us understand, namely that the Lord is going to open the Word in respect to its inner meaning at the end of the church , when there is no more love and therefore no faith; and that he is going to uncover arcana of heaven.

2.5 Okolina za pisanje poezije

Za poeziju se koristi *verse* okolina. Slijedeća pjesma

TOMISLAV

Tomislav, davni kralj Hrvata
 jutros snjegovima mrsi put
 i pod svoj skut golubove skriva.
 Tomislav zagrebački gost
 s dragih poljana i njiva.

Njegov je korak mjeden
 njegov je pogled leden.

A pahulje lete, lete
 I vjetar okolo mete
 praveći piruete.

A kad noćne svjetiljke planu,
 — laku noć — bljesnu danu,
 Tomislav tu negdje bdije
 I srebrno se smije.

Milan Taritaš

je dobivena utipkavanjem

```

\hspace{4em}
{\large \bf TOMISLAV}

\vspace{1em}

\begin{verse}
Tomislav, davni kralj Hrvata\\
jutros snjegovima mrsi put\\
i pod svoj skut golubove skriva.\\
Tomislav zagreba\{c}ki gost\\
s dragih poljana i njiva.

\vspace{1em}

Njegov je korak mjeden\\
njegov je pogled leden.

\vspace{1em}

A pahulje lete, lete\\
I vjetar okolo mete\\
prave\{c}i piruete.

\vspace{1em}

A kad no\{c}ne svjetiljke planu,\\
--- laku no\{c} --- bljesnu danu,\\
Tomislav tu negdje bdije\\
I srebrno se smije.
\end{verse}

\vspace{1em}

\hspace{10em}
{\small Milan Tarita\v{s} }

```

Ako želimo veća slova možemo npr. u naslovu, umjesto `\large` staviti `\Large`, dok unutar *verse* okoline (dakle iza `\begin{verse}`) stavimo `\large`. Izlaz izgleda ovako

TOMISLAV

Tomislav, davni kralj Hrvata
jutros snjegovima mrsi put

i pod svoj skut golubove skriva.
Tomislav zagrebački gost
s dragih poljana i njiva.

Njegov je korak mjeden
njegov je pogled leden.

A pahulje lete, lete
I vjetar okolo mete
praveći piruete.

A kad noćne svjetiljke planu,
— laku noć — bljesnu danu,
Tomislav tu negdje bdije
I srebrno se smije.

Milan Taritaš

Primijetimo da smo za razdvajanje kitica koristili `\vspace` naredbu. Količinu vertikalnog pomaka najbolje sami ocijenimo.

2.6 Okolina za doslovni prikaz ulazne datoteke

Okolina *verbatim* prikazuje tekst točno onako kako se pojavljuje u ulaznoj datoteci. Na izlazu se koristi romanski stil slova, a prikazuju se svi znakovi uključujući nove linije. Čak se i `\` znak pojavljuje na izlazu. Jedini izuzetak je `\end{verbatim}` naredba koja samo završava ovu okolinu. Na primjer, ako na izlazu želimo dobiti

```
{\it Kiklop\}
      je jedan od
{\bf najboljih} romana koje
      sam dosad \v{c}itao.
treba samo utipkati
\begin{verbatim}
{\it Kiklop\}
      je jedan od
{\bf najboljih} romana koje
      sam dosad \v{c}itao.
```

`\end{verbatim}`

Ako želimo da se na izlazu ispiše `\end{verbatim}`, dovoljno je iza naredbe `\end{verbatim}` napisati `\verb|\end{verbatim}|`. Sve što je unutar para `| |` se slaže u doslovnom (verbatim) načinu. Zapravo bilo koja dva jednaka znaka (osim znaka `*`) se mogu koristiti kao granične točke za tekst između njih. Na primjer

```

\verb-Kiklop je najbolji roman ...-
\verb&Kiklop je najbolji roman ...&
\verb#Kiklop je najbolji roman ...#
\verb+Kiklop je najbolji roman ...+

```

Znak `|` je pogodan jer se ne koristi često u normalnom tekstu. Unutar graničnih točaka se stavlja tekst u obliku koji želimo na izlazu. Međutim cijela `\verb` naredba se mora nalaziti unutar jednog retka ulazne datoteke.

Kada se koristi `verbatim` okolina treba naredbe za njen početak i kraj `\begin{verbatim}` i `\end{verbatim}` natipkati od prvog znaka u retku (dakle posve ulijevo). U protivnom će se pojaviti dodatni prazan redak na izlazu.

3 Slaganje matematičkog teksta

Jedna od važnijih odlika \LaTeX a je snaga i jednostavnost slaganja matematičkog teksta.

3.1 Jednostavni matematički tekst

U \LaTeX u se matematički tekst može slagati na tri načina. Oni su opisani sa tri matematičke okoline.

math okolina koja slaže formule kao npr. $\sum_{i=1}^n x_i$ unutar retka u tekstu. Ova okolina započinje s `$` i završava istim znakom. Umjesto znaka `$` kao graničnike možemo još koristiti par zagrada `\(, \)`, ili naredbe `\begin{math}` i `\end{math}`. Obično se koristi graničnik `$`, jer ga je lako utipkati. Međutim za duže formule preporučaju se drugi graničnici jer se `$` lakše zaboravi, pa je onda teže ispravljati greške.

displaymath okolina koja centrira formulu u novom retku. Npr.

$$\sum_{i=1}^n x_i.$$

Ova okolina započinje (završava) naredbama `\[` (odnosno `\]`) ili `\begin{displaymath}` (`\end{displaymath}`). Kao graničnici se mogu koristiti i dvostruki dolarski znaci `$$`.

equation okolina, koja je posve jednaka okolini `displaymath`, ali stavlja broj relacije koji se automatski povećava za jedan unutar svakog poglavlja knjige (kod stila *book* i *report*) ili u cijelom tekstu članka (kod stila *article*). Broj se stavlja uz desnu (lijevu) marginu na neparnim (parnim) stranicama. Kod stila *article* stavlja se uvijek desno. Npr.

$$\sum_{i=1}^n x_i. \tag{1}$$

Ovu okolinu započinjemo naredbom `\begin{equation}` a završavamo s `\end{equation}`.

Ako želimo da se formula u *displaymath* i *equation* okolini lijevo pozicionira treba dodati opciju *fleqn* u naredbi `\documentstyle`, npr.

```
\documentstyle[12pt,fleqn]{article}
```

3.1.1 Slova, jednostavni izrazi i jednakosti

L^AT_EX omogućava jednostavno slaganje donjih i gornjih indeksa, razlomaka, korijena, imena funkcija i matematičkih simbola. Općenito, slova u matematičkom načinu (tj. matematičkoj okolini) su nakošena i malo više razdvojena nego u romanskom stilu. Ona se razlikuju od slova u *italic* i *ukošenom* pismu. Uočimo razliku u riječima: *razlika*, *razlika* i *razlika* koje su dobijene utipkavanjem `\em razlika`, `\sl razlika` i `$razlika$`. Ako želimo neki tekst unutar matematičke okoline napisati u romanskom pismu moramo ga napisati unutar vitičastih zagrada `\mbox{ }` komande. Brojevi se međutim pojavljuju u romanskom stilu. Praznine u matematičkom načinu L^AT_EX zanemaruje. Ako nam je potrebna praznina u matematičkoj okolini moramo ju nametnuti naredbom `_`. Ako želimo neko drugo pismo npr. *italic*, možemo koristiti `\mbox{\em }` komandu pri čemu tekst pišemo između `\em` komande i desne vitičaste zagrade. Komande za mijenjanje pisma kao npr. `\em` rade i direktno u matematičkoj okolini, ali one mijenjaju stil samo za slova i znamenke, dakle ne mijenjaju ostale znakove. Npr. ako na ulazu napišemo `\(\bf 289\sqrt{x^2} \div z*t \)` dobit ćemo $289\sqrt{x^2} \div z * t$. Spomenimo da T_EX tretira mala grčka slova kao znakove, a velika kao slova. Uočimo razliku u primjeru: $\Pi \times \Sigma$, $\pi \times \sigma$ koji je dobijen pisanjem `\$ \bf \Pi \times \Sigma`, `\$ \bf \pi \times \sigma` na ulazu. Jasno, doseg `\$ \bf $` komande je unutar matematičke okoline u kojoj je definirana, dakle između znakova `$ $`.

Postoje dvije komande za mijenjanje pisma koje rade isključivo u matematičkoj okolini: `\cal` koja producira kaligrafska slova (samo velika) i `\mit` koja stvara matematička italic slova (standardna slova za matematičku okolinu). Pritom će se `\mit` rijetko kada koristiti, a `\cal` češće, npr. u matematici za imena funkcija i opratora: $\mathcal{F}, \mathcal{G}, \mathcal{O}$.

Ako želimo podebljani ispis, ne samo slova i znamenaka nego i ostalih znakova u matematičkoj okolini, možemo koristiti `\boldmath` komandu. One se ne smije pisati u matematičkoj okolini (zato se unutar nje koristi `\mbox{\boldmath $ $}`). Ipak ova komanda ne podebljava donje i gornje indekse, znakove `+` `:` `;` `!` `?` `(` `)` `[` `]` i graničnike promjenjive visine koji počinju s `\left` i završavaju sa `\right`. Npr. pisanjem `\boldmath $A+b-\alpha$` se dobije $A + b - \alpha$.

Spomenimo još da se mali (i još manji) znakovi u matematičkoj okolini mogu dobiti pisanjem `\scriptstyle` (`\scriptscriptstyle`) komandi. Npr. `\scriptscriptstyle A+b-\alpha` će dati $A+b-\alpha$.

Evo sada primjera kako se dobivaju donji i gornji indeksi u matematičkoj okolini.

- Gornje i donje indekse slažemo pomoću znakova `_` i `^`:

Izlaz	Ulaz	Izlaz	Ulaz
x_2	<code>\\$x_2\\$</code>	x_2y_3	<code>\\$x_2y_3\\$</code>
$y = z_{3b}$	<code>\\$y=z_{3b}\\$</code>	x^3	<code>\\$x^3\\$</code>
x^3y^2	<code>\\$x^3y^2\\$</code>	5^x	<code>\\$5^x\\$</code>
${}_3Y^2$	<code>\\$_3Y^2\\$</code>	x^{y^3}	<code>\\$x^{\{y^3\}}\\$</code>
$x_{\frac{3}{2}}$	<code>\\$x_{\frac{3}{2}}\\$</code>	$x_{\frac{2}{3}}$	<code>\\$x_{\frac{2}{3}}\\$</code>
$a_{ij}^{(k)}$	<code>\\$a_{ij}^{\{(k)\}}\\$</code>		
$(X_{a,b,c}^{[i,j]})^2$	<code>\\$(X_{a,b,c}^{\{[i,j]\}})^2\\$</code>		

$$A_{(ijk)}^{\{p,q\}}(m;n;p) \quad \$A_{\{(ijk)\}^{\{p,q\}}}(m;n;p)\$$$

- **Razlomke** slažemo pomoću znaka / ili naredbe `\frac`:

Izlaz	Ulaz
$3/4$	<code>\\$3/4\\$</code>
$(3/7) * 12.8$	<code>\\$(3/7)*12.8\\$</code>
$\frac{13}{125}$	<code>\\$\frac{13}{125}\\$</code>
$\frac{a}{b} = \frac{m+n}{x+y}$	<code>\\$\frac{a}{b}=\frac{m+n}{x+y}\\$</code>
$\frac{a}{1+x} + \frac{1}{2}$	<code>\\$\frac{a}{1+x}+\frac{1}{2}\\$</code>

Naredba `\frac` bi se uglavnom trebala koristiti u *displaymath* i *equation* okolini zbog boljeg izgleda. Evo par složenijih razlomaka:

Izlaz	Ulaz
$x = \frac{y^3 + z/5}{y^2 + 8}$	<code>\[x = \frac{y^3 + z/5}{y^2 + 8} \]</code>
$x = y + \frac{x^3}{z_2} - 4^n$	<code>\[x = y + \frac{x^3}{z_2} - 4^n \]</code>
$z = \frac{x^{\frac{3}{n-2}}}{1-x}$	<code>\[z = \frac{x^{\{\frac{3}{n-2}\}}}{1-x}\]</code>

- **Korijene** slažemo pomoću naredbe `\sqrt`:

Izlaz	Ulaz	Izlaz	Ulaz
$\sqrt{2}$	<code>\\$\sqrt{2}\\$</code>	$\sqrt[3]{n}$	<code>\\$\sqrt[n]{3}\\$</code>
$\sqrt{n+19}$	<code>\\$\sqrt{n+19}\\$</code>	$\sqrt{n^3 + \sqrt{y_2}}$	<code>\\$\sqrt{n^3+\sqrt{y_2}}\\$</code>

- **Točkice** kao oznaka za ponavljanje mogu biti horizontalne, okomite i kose. Horizontalne mogu biti donje ... i centralne ... Prve se postižu pisanjem naredbe `\ldots` koju možemo upotrebljavati i u matematičkom načinu i u običnom tekstovnom načinu (koji još zovemo i LR način, što dolazi od “left-right” tj. način u kojem se tekst slaže s lijeva na desno). Ostale naredbe smijemo koristiti samo u matematičkom načinu.

Tip	Izlaz	Ulaz
Donje	y_1, \dots, y_n	<code>\\$y_1,\ldots ,y_n\\$</code>
Centralne	$1 + 2 + \dots + 13$	<code>\\$1+2+\cdots + 13\\$</code>
Vertikalne	\vdots	<code>\\$\vdots\\$</code>
Dijagonalne	\ddots	<code>\\$\ddots\\$</code>

3.1.2 Matematički simboli

Matematičke simbole koje \LaTeX može ispisati smijemo koristiti samo u matematičkom načinu. Npr. da bismo na izlazu dobili α moramo napisati \alpha . Sve \LaTeX ove matematičke simbole možete naći u dodatku C, tablice 5–13, str. 48–50. Znakovi u tablici 10 biti će manji ako ih pišete u *math* okolini, a veći ako ih koristite u okolinama *displaymath* ili *equation*. Ako želite prekrižiti (negirati ili nijekati) bilo koji simbol stavljanjem kose crte preko simbola, to se postiže naredbom \not koju ispišete ispred naredbe za simbol koji želite prekrižiti. Npr. ulazni tekst $\text{\mbox{ako je}}\ y^2\text{\not=} 4\ \text{\mbox{tada slijedi}}\ y\text{\not=} 2$ će na izlazu dati ako je $y^2 \neq 4$ tada slijedi $y \neq 2$.

\LaTeX obično dobro razdvaja znakove ali koji puta želimo sami razdvojiti znakove i simbole. To možemo u matematičkom načinu postići pomoću naredbi za horizontalno pomicanje: \backslash , (tanki razmak), \backslash (srednji razmak), \backslash (debeli razmak), i $\backslash!$ (negativni tanki razmak). Naredba za negativni razmak se ponaša kao backspace (pomak unazad) naredba. Pritom se \backslash , može koristiti u svakom načinu (ne samo matematičkom).

Imena funkcija, koja upisujemo pomoću odgovarajućih naredbi, pojavljuju se u romanskom stilu iako su pisana u matematičkom načinu (okolini). Popis funkcija koje su unaprijed zadane nalazi se u dodatku C u tablici 11 na str. 50. Za funkcije za koje ne postoje \LaTeX naredbe možemo koristiti naredbu $\text{\mbox{}}\{}$ npr. $\text{\mbox{sh}}\backslash: x$. Neke od funkcija kao npr. \lim , \inf , \sup , \liminf , \limsup , \max , \min dozvoljavaju donje indekse koji se u “display” (tj. *displaymath*, *equation*, *eqnarray*) okolinama na izlazu ispisuju ispod imena funkcija. Unutar odlomka, dakle u *math* okolini taj indeks je izvučen više gore i udesno kako bi uzeo što manje vertikalnog prostora. Evo primjera: $\lim_{n \rightarrow \infty} x = 0$ se dobije tipkajući (unutar odlomka) $\backslash \backslash \text{\lim}_{n \rightarrow \infty} x = 0$. U *displaymath* okolini bi to izgledalo ovako:

$$\lim_{n \rightarrow \infty} x = 0 \quad \backslash [\backslash \lim_{n \rightarrow \infty} x = 0 \backslash]$$

Postoje matematički simboli (vidi tabelu 10 na str. 49) koji se prikazuju u dvije veličine ovisno o tome da li se nalaze ili ne u display okolinama. Ovisno o tome oni također donje, ali i gornje indekse ispisuju na različite načine. Evo primjera

$$\sum_{k=1}^n |x_k|^2 = \prod_{i=1}^p \int_0^{\infty} G_i dx \quad \backslash [\backslash \sum_{k=1}^n \mid x_k \mid ^2 = \backslash \prod_{i=1}^p \int_0^{\infty} \{\text{\cal G}\}_i dx \backslash]$$

dok će ista matematička relacija u *math* okolini poprimiti oblik $\sum_{k=1}^n |x_k|^2 = \prod_{i=1}^p \int_0^{\infty} g_i dx$. Ako unutar odlomka tj. *math* okoline iznimno želimo oblik koji se javlja u display okolinama tada moramo napisati \displaystyle , a samu formulu ispred desnog \ znaka. Slično, ako zbog nekog razloga želimo u display okolinama pisati formule sa zahtjevom da one na izlazu izgledaju kao u *math* okolini onda to forsiramo pišući komandu \textstyle na početku okoline.

3.2 Složeniji matematički tekst

3.2.1 “Array” okolina

Array okolina omogućava slaganje jedno- i dvo-dimenzionalnih polja. Za svaki stupac se može naznačiti da li će sadržaj biti lijevo, desno ili centralno pozicioniran. To se postiže pisanjem jednog od slova: l, r, c. Retci se razdvajaju naredbom $\backslash\backslash$, osim zadnjeg retka u kojem se $\backslash\backslash$

izostavlja. Stupci se razdvajaju znakom &. Taj znak se izostavlja iza zadnjeg stupca. Evo primjera:

$$\begin{array}{ccc} a & 14 & c \\ d-3 & e & f \\ g & h & \lambda \end{array}$$

```

\[\ \begin{array}{ccc}
a & & 14 & & c \\
d-3 & & e & & f \\
g & & h & & \lambda \end{array} \]

```

Ili npr.

$$\det \begin{vmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{vmatrix} > 0$$

se postiže tipkanjem

```

\[\ \det \left|
\begin{array}{cccccc}
a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\
a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\
a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn}
\end{array}
\right| > 0 \]

```

Vertikalne crte ispred i iza matrice dobili smo korištenjem naredbe za međaše `\left|` i `\right|`. Smisao ovih naredbi je da se vertikalna crta | isteže sve dok ne bude jednaka visini one konstrukcije koja je naznačena između `\left|` i `\right|`. Umjesto | možemo koristiti npr. razne zagrade, razne vertikalne strelice i sl. (vidi dodatak C, tablica 12, str. 50.) Npr.

$$\left(\begin{array}{c} \left[\begin{array}{cc} a & b \\ d & d \\ e & f \end{array} \right] \\ x \\ y \end{array} \right) = \left(\begin{array}{c} \mathbf{X} \\ x \\ y \end{array} \right)$$

```

\[\ \left( \begin{array}{c}
\left[ \begin{array}{cc}
a & b \\
d & d \\
e & f
\end{array} \right] \\
x \\
y
\end{array} \right) = \left( \begin{array}{c}
\mathbf{X} \\
x \\
y
\end{array} \right) \]

```

Oba međaša ne moraju biti jednaka, ali se moraju koristiti u paru. Ako želimo imati samo jedan međaš moramo umjesto drugog međaša upotrijebiti `\left.` ili `\right.`. Evo primjera

$$F(x) = \begin{cases} 0 & \text{ako je } x \leq 0 \\ x & \text{ako je } 0 \leq x \leq 1 \\ 1 & \text{ako je } x \geq 1 \end{cases}$$

```

\[\ F(x)=\left\{ \begin{array}{l}
0 & \text{ako je } x \leq 0 \\
x & \text{ako je } 0 \leq x \leq 1 \\
1 & \text{ako je } x \geq 1
\end{array} \right. \]

```

Uočimo razliku ako parametre lr zamijenimo s ll

$$F(x) = \begin{cases} 0 & \text{ako je } x \leq 0 \\ x & \text{ako je } 0 \leq x \leq 1 \\ 1 & \text{ako je } x \geq 1 \end{cases}$$

```
\[ F(x)=\left\{ \begin{array}{l} 0 \\ x \\ 1 \end{array} \right. \begin{array}{l} \text{ako je } x \leq 0 \\ \text{ako je } 0 \leq x \leq 1 \\ \text{ako je } x \geq 1 \end{array} \]
```

Vrlo su korisne i naredbe `\multicolumn`, kao i naredbe za poravnavanje prema nekom znaku i pisanje graničnih linija. U slijedećem primjeru koristimo decimalnu točku kao ishodište za poravnavanje.

123.45
99988.111
4.1234
na pr.
783.5

```
\[ \begin{array}{|r@{.}l|} \hline 123 & 45 \\ 99988 & 111 \\ 4 & 1234 \\ \multicolumn{2}{|c|}{\mbox{na pr.}} \\ 783 & 5 \\ \hline \end{array} \]
```

Uočimo da smo mogli koristiti bilo koji drugi znak umjesto točke za poravnavanje. Svaka okomita crta u naredbi `\begin{array}` ispisuje vertikalnu liniju na tom mjestu kroz cijelu visinu polja. Slično `\hline` ispisuje horizontalnu crtu na mjestu gdje je navedena. Ako `\hline` slijedi iza naredbe `\\` onda se linija povlači ispod podataka u prethodnom retku. Naredba `\multicolumn{c}{c}` omogućuje da se ulazni tekst pruži kroz više stupaca. Prvi parametar označava kroz koliko stupaca se tekst pruža, drugi pokazuje kako je pozicioniran (l, c, r), dok je treći parametar sam tekst. Konačno, naredba `\mbox{}` osigurava da se tekst složi u LR načinu (obični tekstualni način). To je potrebno jer okolina *array* radi samo unutar matematičkog načina pa bi bez `\mbox{}` slova bila ukošena i drugačije razmaknuta.

3.2.2 Višelinjske jednadžbe

Jednadžbe i matematički izrazi mogu biti tako dugački da se moraju protezati u više redaka. U tu svrhu se koristi okolina *eqnarray*. Ona radi kao polje s tri stupca razdvojena znakovima `&`. Za razliku od polja koje moramo pisati unutar neke matematičke okoline (u matematičkom načinu), okolina *eqnarray* sama stvara matematički način. Evo jednostavnog primjera:

$$x = a - b - c - d - e - f - g$$

```
\begin{eqnarray*} x & = & a-b-c-d \\ & & e-f-g \end{eqnarray*}
```

Primijetimo da `\mbox{}` zauzima malo mjesta ispred znaka `-` koji je stoga više desno pozicioniran. Ako maknemo zvijezdicu u `\begin{eqnarray*}` i `\end{eqnarray*}` komandama svaki redak dobiva broj jednadžbe koji je pozicioniran uz desnu marginu. Ako u nekom od tih redaka ne želimo broj, moramo ispred komande `\\` napisati komandu `\nonumber`. Evo još jednog načina kako lomiti dugu jednadžbu u više redaka:

$$x + y + v + w + z + x_1 + y_1 + v_1 = a - b - c - d - e - f - g - a_1 - b_1$$

```
\begin{eqnarray*} \lefteqn{x+y+v+w+z+x_1+y_1+v_1=} \\ & & a-b-c-d-e-f-g \\ \end{eqnarray*}
```

Kao i u prijašnjem primjeru dva znaka `&` nameću pojavljivanje prvog znaka u drugom retku dva stupca u desno. \LaTeX određuje širinu svakog stupca prema najdužem tekstu koji se u tom stupcu pojavljuje. Naredbom `\lefteqn{}` odredili smo da dužinu teksta unutar znakova `{}` \LaTeX zanemaruje. Naredbom `\hspace{}` možemo pomicati redove u desno ili u lijevo. Evo još jednog složenijeg primjera

$$\begin{aligned} \Upsilon_{ij} = & \max \left\{ \sum_{t=2u+1}^{n'} \left(|c_{2j-1,t}^{(2i-1,t)}|^2 + |c_{2j,t}^{(2i-1,t)}|^2 \right), \sum_{t=2u+1}^{n'} \left(|c_{2j-1,t}^{(2i,t)}|^2 + |c_{2j,t}^{(2i,t)}|^2 \right) \right\} \\ & \cdot \sum_{t=2u+1}^{n'} \left(\sin^2 \phi_{2i-1,t} + \sin^2 \phi_{2i,t} \right) + \\ & + \max \left\{ \sum_{t=2u+1}^{n'} \left(|c_{2i-1,t}^{(2j-1,t)}|^2 + |c_{2i,t}^{(2j-1,t)}|^2 \right), \sum_{t=2u+1}^{n'} \left(|c_{2i-1,t}^{(2j,t)}|^2 + |c_{2i,t}^{(2j,t)}|^2 \right) \right\} \\ & \cdot \sum_{t=2u+1}^{n'} \left(\sin^2 \phi_{2j-1,t} + \sin^2 \phi_{2j,t} \right). \end{aligned} \quad (2)$$

To smo dobili utipkavajući

```
begin{eqnarray} \lefteqn{
\Upsilon_{ij} = \max \left\{ \sum_{t=2u+1}^{n'} \left( |c_{2j-1,t}^{(2i-1,t)}|^2 + |c_{2j,t}^{(2i-1,t)}|^2 \right), \sum_{t=2u+1}^{n'} \left( |c_{2j-1,t}^{(2i,t)}|^2 + |c_{2j,t}^{(2i,t)}|^2 \right) \right\}
\cdot \sum_{t=2u+1}^{n'} \left( \sin^2 \phi_{2i-1,t} + \sin^2 \phi_{2i,t} \right) +
& + \max \left\{ \sum_{t=2u+1}^{n'} \left( |c_{2i-1,t}^{(2j-1,t)}|^2 + |c_{2i,t}^{(2j-1,t)}|^2 \right), \sum_{t=2u+1}^{n'} \left( |c_{2i-1,t}^{(2j,t)}|^2 + |c_{2i,t}^{(2j,t)}|^2 \right) \right\}
\cdot \sum_{t=2u+1}^{n'} \left( \sin^2 \phi_{2j-1,t} + \sin^2 \phi_{2j,t} \right).
\end{eqnarray}
```

Iz ovog primjera vidljivo je da naredbe za produžene graničnike `\left` i `\right` možemo koristiti i za uobičajene zagrade. Nužno je da se ove naredbe pojavljuju u paru, jer \LaTeX određuje njihovu visinu prema visini teksta koji smo upisali između dva graničnika.

3.2.3 Specijalni efekti

Matematičke akcente dobivamo pomoću naredbi kao što su `\hat{}`, `\tilde{}` i sl. (vidi dodatak C, tablica 13, str. 50.) Npr. \hat{c} daje \hat{c} , dok \tilde{c} daje \tilde{c} . Posebno je važna naredba `\stackrel{\rightarrow}{+}`. Npr. utipkavanjem $\overrightarrow{+}$ možemo dobiti simbol $\overrightarrow{+}$. Iznad i ispod formula možemo povući linije pomoću naredbi `\overline{}` i `\underline{}`. Npr.

$$\overline{\overline{x^2 + z + \sqrt{x}}}, \underline{2x^3} \text{ je podvučen}$$

dobijemo utipkavanjem

```
\[
\overline{\overline{x}^2 + z + \sqrt{\overline{x}}} ,
\underline{2\ :x} \mbox{ je } \underline{\mbox{podvu}\v{c}en}
\]
```

Na sličan način pomoću naredbi `\overbrace{}` i `\underbrace{}` dobivamo gornje i donje vitičaste zagrade, npr.

$$\overbrace{x^2 + z + \underbrace{\sqrt{x} + y + \sqrt{w}}}$$

dobijemo utipkavanjem

```
\[
\overbrace{x^2 + \overbrace{z + \sqrt{x}}} +
\underbrace{y + \sqrt{w}}
\]
```

4 Tabele

Tehničke knjige i članci često sadrže mnoštvo detaljnih brojeanih podataka. Takvi podaci se najbolje prikazuju u tabelarnom obliku. \LaTeX omogućava nekoliko oblika tabela. Najjednostavnija je *tabbing* okolina koja radi slično kao što radi “tab” tipka na tastaturi. Složenija je *tabular* okolina koja omogućava da se dodaju linije i “kutije” oko teksta. Riječ “kutija” (engl. “box”) označava pravokutnik u kojem se može nalaziti jedan ili više redaka teksta.

4.1 “Tabbing” okolina

Ova okolina se najbolje može razumjeti gledajući primjere.

<i>Krvna grupa</i>	<i>Antigeni</i>	<i>Antitijela</i>	
O	ništa	a, b	<code>\begin{tabbing}</code>
A	A	b	<code>{\it Krvna grupa} \= {\it Antigeni} \=</code>
B	B	a	<code>{\it Antitijela} \\ O \> ni\sx ta \> a, b \\ A \> A \> b \\ B \> B \> a \\ AB \> A, B \> ni\sx ta</code>
AB	A, B	ništa	<code>\end{tabbing}</code>

Uočimo da su “tab” pozicije definirane u prvom retku pomoću kontrolnog niza znakova `\=`. Podaci za prvi stupac se pišu lijevo pozicionirano, tj. iza znaka `\`. Svakom slijedećem stupcu prethodi kontrolni niz `\>`. Dakle naredba `\>` igra ulogu sličnu onoj tipke *tab* na tastaturi kada koristimo neki program za obradu teksta; ona kaže \LaTeX u da napreduje do slijedeće tab pozicije. Primijetimo da svaki redak na ulazu osim zadnjeg završava naredbom `\` za prelazak u novi red. Sjetimo se da jedna ili tisuću praznina u ulaznoj datoteci znači jednu prazninu na izlazu. Kod tabela vrijedi i dodatno pravilo da se paraznine iza (ali ne i one ispred) `\=` i `\>` ignoriraju.

Postoje još dva načina pisanja ove tabele. Svaki omogućava bolju kontrolu širine stupaca. Prvi način traži da se za svaki stupac upiše onoliko znakova “x” koliko želimo pri čemu svaki znak x proširi stupac za njegovu širinu na izlazu (ovisno o izboru veličine znakova u naredbi `\documentstyle`). Na kraju linije se piše naredba `\kill` koja \LaTeX u kaže da više nema tab pozicija. Također, linija koja završava sa `\kill` se ne ispisuje na izlazu. Evo primjera:

<i>Krvna grupa</i>	<i>Antigeni</i>	<i>Antitijela</i>	<code>\begin{tabbing}</code>
O	ništa	a, b	<code>xxxxxxxxxxxxx\=xxxxxxxxxxxxx\=\kill</code>
A	A	b	<code>{\it Krvna grupa} \>{\it Antigeni} \></code>
B	B	a	<code>{\it Antitijela} \\ 0 \> ni\sx ta \> a, b \\ A \> A \> b \\ B \> B \> a \\ AB \> A, B \> ni\sx ta</code>
AB	A, B	ništa	<code>\end{tabbing}</code>

Uočimo da sada i pred naslove u tabeli moramo staviti naredbu `\>` koja kaže $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u gdje da ih stavi. Kad ocjenjujete širinu svakog stupca osigurajte dovoljno mjesta tako da se svi podaci u stupcu dobro smjeste. Ako se ne predvidi dovoljno mjesta kao u slijedećem primjeru mogu se dva teksta preklapati. Slijedeći primjer također pokazuje kako se unutar tabele širine stupaca mogu redefinirati.

<i>Krvna grupa</i>	<i>Antigeni</i>	<i>Antitijela</i>	<code>\begin{tabbing}</code>
O	ništa	a, b	<code>xxxxxx\=xxxxxx\=\kill</code>
A	A	b	<code>{\it Krvna grupa} \> {\it Antigeni} \></code>
B	B	a	<code>{\it Antitijela} \\ 0 \> ni\sx ta \> a, b \\ A \> A \> b \\ xxxxxxxxxxxxxx\=xxxxxxxxxx\=\kill</code>
AB	A, B	ništa	<code>{\it Krvna grupa} \> {\it Antigeni} \></code>
			<code>{\it Antitijela} \\ B \> B \> a \\ AB \> A, B \> ni\sx ta</code>
			<code>\end{tabbing}</code>

Drugi je način da se uključe naredbe za širinu razmaka:

<i>Krvna grupa</i>	<i>Antigeni</i>	<i>Antitijela</i>	<code>\begin{tabbing} \hspace{1in}\=</code>
O	ništa	a, b	<code>\hspace{3cm}\=\hspace{10em}\kill</code>
A	A	b	<code>{\it Krvna grupa} \>{\it Antigeni}</code>
B	B	a	<code>\>{\it Antitijela} \\ 0 \> ni\sx ta \> a, b \\ A \> A \> b \\ B \> B \> a \\ AB \> A, B \> ni\sx ta</code>
AB	A, B	ništa	<code>\end{tabbing}</code>

Sve deklaracije koje se načine unutar `tabbing` okoline su lokalne u odnosu na onaj dio tabele u kojem su definirani. Točnije, doseg deklaracije je do slijedećeg znaka `\=`, `\>`, `\\` ili `\kill`. Evo primjera

			<code>\begin{tabbing} \hspace{1in}\=</code>
			<code>\hspace{3cm}\=\hspace{10em}\kill</code>
<i>Krvna grupa</i>	Antigeni	Antitijela	<code>\it Krvna grupa \> Antigeni</code>
O	ništa	a, b	<code>\> \bf Antitijela \\\</code>
A	A	b	<code>\Large 0 \> ni\sx ta \> a, b \\\</code>
B	B	a	<code>A \>\tiny A \> b \\\</code>
AB	A, B	ništa	<code>B \> \bf B \> a \\\</code>
			<code>AB \> A, \bf B \> ni\sx ta</code>
			<code>\end{tabbing}</code>

Ako je vaša tabela dugačka ili vrlo složena bit će bolje koristiti okolinu *tabular*. Ipak i za okolinu *tabbing* postoje dodatne naredbe koje se mogu korisno upotrijebiti. One su prikazane u slijedećoj tabeli.

Naredba	Posljedica
<code>\></code>	Slijedeća tab pozicija
<code>\<</code>	Prethodna tab pozicija
<code>\‘</code>	Prema desnoj margini
<code>\’</code>	Desno pozicionirano u prethodnom stupcu
<code>\=</code>	Definira novu tab poziciju na tekućoj poziciji
<code>\\</code>	Započinje novi redak
<code>\kill</code>	Pamti definicije tab pozicija bez ispisa teksta i započinje novi redak
<code>\+</code>	Definira novu lijevu marginu jednu tab poziciju nadesno
<code>\-</code>	Definira novu lijevu marginu jednu tab poziciju nalijevo
<code>\pushtabs</code>	Spremi tekuće tab pozicije u <i>tabbing</i> okolini
<code>\poptabs</code>	Ponovo definira tab pozicije spremljene pomoću <code>\poptabs</code>

Ako želite unutar okoline *tabbing* koristiti akcente koje u običnom tekstovnom načinu dobivate naredbama `\=`, `\’` i `\‘`, umjesto njih koristite naredbe `\a=`, `\a’` i `\a‘`, respektivno.

Slijedeći primjer pokazuje kako se neke od tih “*tabbing* naredbi” mogu koristiti. Ulaz

```
\begin{tabbing}
xxxxxxxxxxxxxxxx\=                \kill
Broj      \> A-123 \‘ B-555        \\\
1         \> 21                    \+   \\\
          19 \’ 55                 \\\
\< 2      \> 12                    \-   \\\
3         \> 12                    \‘ nije primjenjiva \\\
\pushtabs
Podaci za 4 nisu dovr\sx eni      \\\
\poptabs
5         \> 777
\end{tabbing}
```

će dati izlaz

Broj	A-123	B-555
1	21	
	19 55	
2		
3	12	nije primjenjiva

Podaci za 4 nisu dovršeni

5 777

Vidimo da su korištene složene naredbe za tabulaciju. Ako zahtijevate tabele iole višeg stupnja složenosti trebali biste koristiti okolinu *tabular*.

4.2 “Tabular” okolina

Okolina *tabular* omogućava slaganje složenih tabela i slika. Ona vrlo jednostavno crta okvirne linije i linije unutar tabele. Kao okolina dosta je slična matematičkoj *array* okolini. Glavna je razlika da se njeni elementi obrađuju u bilo kojem, tj. ne samo u matematičkom načinu.

Slijedeći primjer pokazuje komponente zraka:

KOMPONENTA	UDAHNUTI ZRAK	IZDAHNUTI ZRAK	ZRAK U MJEHURIĆIMA
Dušik	78,08%	27,26%	78,36%
Kisik	20,94%	16,30%	15,20%
Ugljični dioksid	0,04%	4,50%	5,50%
Argon i ostali plinovi	0,94%	0,94%	0,94%

Evo kako se to postiglo

```
\begin{center}
\begin{tabular}{|l|r|r|r|} \hline
& {\bf UDAHNU} & {\bf IZDAHNU} & {\bf ZRAK U} \\
{\bf KOMPONENTA} & {\bf ZRAK} & & \\
& {\bf ZRAK} & {\bf MJEHURI} & \\
& & & {\bf CY IMA} \\
Dušik & 78,08% & 27,26% & 78,36% \\
Kisik & 20,94% & 16,30% & 15,20% \\
Ugljični dioksid & 0,04% & 4,50% & 5,50% \\
Argon i ostali plinovi & 0,94% & 0,94% & 0,94% \\
\hline
\end{tabular}
\end{center}
```

Ovdje je okolina *center* horizontalno centrirala tabelu. Okomite crte | u vitičastim zagradama nakon početne naredbe `\begin{tabular}` određuju položaj okomitih linija u tabeli. Parametar l označava da se podaci smještavaju lijevo unutar stupca. Slično, parametri r i c služe za desno odnosno centralno pozicioniranje unutar stupca. Naredba `\hline` crta vodoravnu liniju na mjestu gdje je napisana; ako slijedi naredbu `\\` nacrtat će se linija ispod podataka u tom retku.

L^AT_EX dozvoljava korištenje ostalih naredbi kao npr. naslovi stupaca, i varijabilna širina stupaca. Evo primjera

<i>Krvna grupa</i>	<i>Genotip</i>	<i>Genotipska frekvencija</i>
O	aa	r^2
A	AA AO	p^2 $2pr$
B	BB BO	q^2 $2qr$
AB	AB	$2pq$

koji je složen tipkanjem

Naredbe za smještanje tabele/slike neće uvijek besprijekorno raditi. \LaTeX obrađuje dokument po stranicama. Ako navedemo “donji” tj. “b” položaj za tabelu na stranici na kojoj dužina postojećeg teksta i visina tabele zajedno premašuju dopuštenu visinu stranice, tabela će se smjestiti prema sljedećoj naredbi. Višestruke naredbe (npr. “bht”) su dozvoljene. Uglatu zagradu s naredbama za smještanje možemo i izostaviti, tada \LaTeX smješta tabelu/sliku kao da su upisane naredbe “tbp”.

Svaka tabela ili slika se automatski prebrojava kad je uključena u *table* ili *figure* okolinu — unutar cijelog dokumenta ako se koristi *article* stil, odnosno unutar svakog poglavlja ako se koristi *book* ili *report* stil. Naslov tabele/slike možemo zadati pomoću naredbe `\caption{ }`. Ako je ona napisana prije (poslije) same tabele u ulaznoj datoteci onda naslov ide iznad (ispod) tabele. U primjeru tablice 1 ta je naredba napisana na početku. Na kraju spomenimo da je tablica 1 ispisana u ulaznoj datoteci između rečenica *One su dane u sljedećoj tabeli.* i *Naredbe za smještanje tabele/slike neće uvijek besprijekorno raditi.* U ulaznoj datoteci, na tome mjestu smo napisali:

```
\begin{table}[htp]
\caption{Naredbe za smještanje tabele i slika} \ref{tbl:smjestaj}
\begin{center}
\begin{tabular}{|c|c|} \hline
Naredba & Mjesto smještanja tabele \\ \hline
b & dno stranice \\ h & teksta \\ t & vrh stranice \\ p & odvojena pluta na stranici \\ \hline
\end{tabular}
\end{center}
\end{table}
```

5 Fusnote i reference

5.1 Fusnote

Da bismo napravili fusnotu³ na ovoj stranici natipkali smo sljedeći tekst:

Da bismo napravili fusnotu `\footnote{Engleski se piše footnote}` na ovoj

Dakle se tekst fusnote umeće unutar teksta, a započinje naredbom `\footnote` iza koje je unutar vitičastih zagrada napisan sadržaj fusnote. Brojač fusnota se automatski povećava za jedan pri svakom nailasku na naredbu `\footnote`, ali to ipak ovisi o stilu dokumenta. Kod stila *article* se brojač uzastopce povećava unutar cijelog dokumenta. Kod *report* i *book* stila svakim nailaskom na novo poglavlje (tj. na naredbu `\chapter`) se brojač stavlja na nulu.

Važno je zapamtiti jednu stvar: kad napravite fusnotu, njen broj će se pojaviti točno na mjestu gdje se nalazi znak `\` u naredbi `\footnote`. To znači da znak `\` moramo postaviti točno iza riječi ili znaka interpunkcije povrh kojeg želimo da se broj pojavi. Dakle, bez praznine između, jer će inače u konačnom dokumentu biti praznina prije broja fusnote.

5.2 Brojači i ukrižene reference

\LaTeX koristi više brojača. Oni su vezani uz neke okoline ili naredbe. Brojači imaju nazive prema odnosnim okolinama ili naredbama. Evo njihove liste:

³Engleski se piše footnote

part	paragraf	figure	enumi
chapter	subparagraf	table	enumii
section	page	footnote	enumiii
subsection	equation	mpfootnote	enumiv
subsubsection			

Brojači enumi, enumii, . . . prebrojava ju različite razine okoline enumerate. Brojač mpfootnote broji fusnote unutar okoline *minipage*.

Brojači su potrebni da bismo se mogli pozivati na neke dijelove teksta. Npr. česte su rečenice u stručnom tekstu kao ova: “. . . vidi relaciju (8) i Crtež 5”. Nije dobro da se brojevi 8/5 pojave u ulaznoj datoteci jer ako naknadno dodamo (ili oduzmemo) jednu relaciju/crtež moramo sve kasnije pripadne brojeve relacija/crteža mijenjati. Umjesto toga toj relaciji odnosno crtežu se pridružuje ključ, tj. bilo kakav niz znakova (po našem izboru). Kasnije (ili možda prije u tekstu) mi se u ulaznoj datoteci pozivamo na taj ključ, a ne na broj koji će L^AT_EX generirati. Ključ se definira naredbom `\label{ }`, a poziva se naredbama `\ref{ }` i `\pageref{ }`. Unutar vitičastih zagrada napišemo ključ. Evo jednog primjera. Prvo pogledajmo dva odlomka teksta na izlazu.

.....

3 Matematičke osnove

U Pitagorinom⁴ poučku koji glasi

$$c^2 = a^2 + b^2 \quad (8)$$

c je duljina hipotenuze a a i b su duljine kateta.

.....

.....

O Pitagorinu Poučku (8) sam Pitagora (vidi fusnotu 4) nije mnogo razmišljao. Uostalom pročitajte Odjeljak 3 na stranici 32 u ovoj skripti.

.....

U ulaznoj datoteci smo napisali slijedeće:

```
.\dotfill .
```

```
\section{Matemati\cx ke osnove} \label{ch:mat osn}
U Pitagorinom\footnote{\label{fn:pit} Pitagora je \v{z}ivio\ldots}
pou\v{c}ku koji glasi
\begin{equation} \label{pit}
c^2 = a^2 + b^2
\end{equation}
$c$ je duljina hipotenuze a $a$ i $b$ su duljine kateta.
```

```
.\dotfill .
```

```
\vspace{1em}
```

⁴ Pitagora je živio. . .

.\dotfill .

0 Pitagorinu Pou\{c}ku (\ref{pit}) sam Pitagora (vidi fusnotu~\ref{fn:pit}) nije mnogo razmi\{s}ljao. Uostalom pro\{c}itajte Odjeljak~\ref{ch:mat osn} na stranici~\pageref{ch:mat osn} u ovoj skripti.

.\dotfill .

Rekli smo, ključ se sastoji od bilo kakvog niza znakova. Pritom se razlikuju mala i velika slova. naredbu `\label{ }` stavljamo unutar one okoline na koju ćemo se kasnije pozivati. Najbolje mjesto je odmah iza početka okoline. Brojač `equation` broji za okoline `equation` i `equanarray` tj. tretira ih kao istu okolinu. Spomenimo još naredbu `\pageref` koja ispisuje broj stranice na kojoj je definiran pripadni ključ. Kod referiranja je uputno koristiti tildu kako se referirani broj ne bi odvojio od prethodne riječi na kraju retka.

Naredba `\setcounter` postavlja brojač na određenu vrijednost, a naredba `\addtocounter` povisuje/snizuje vrijednost brojača za danu vrijednost. Ako na ulazu napišemo:

```
\setcounter{footnote}{6} Jer\footnote{\ldots} se \addtocounter{footnote}{-3}
broja\ v ci\footnote{\ldots} pove\ v caju za jedan svaki put kad se koristi
pripadna okolina, definirajte ih prije upotrebe okoline za jedan manje od broja
kojeg \ v zelite pridru\ v ziti broja\ v cu.
```

dobit ćemo na izlazu:

Jer⁷ se brojači⁵ povećaju za jedan svaki put kad se koristi pripadna okolina, definirajte ih prije upotrebe okoline za jedan manje od broja kojeg želite pridružiti brojaču.

Dok se ostali brojači prije prvog korištenja pripadne okoline definiraju nulom, brojač okoline `page` započinje s vrijednošću 1. Taj brojač se povećava za jedan kod svakog prelaza na novu stranicu, ali nakon prelaska na novu stranicu, pa ima uvijek vrijednost tekuće stranice.

Tekuća vrijednost brojača se može ispisati (velikim ili malim) arapskim, rimskim i grčkim brojevima. Na raspolaganju su naredbe `\arabic{page}`, `\roman{page}`, `\Roman{page}`, `\alph{page}` i `\Alph{page}`, pri čemu smo kao brojač uzeli `page`. Za generiranje izlaznog oblika broja \LaTeX koristi naredbu koja ima prefiks `the` ispred imena brojača. Npr. ako želimo da se brojevi sekcija u dokumentu ispisuju arapskim brojevima, koristit ćemo `\renewcommand` na slijedeći način:

```
\renewcommand{\thesection}{\arabic{section}}
```

Na taj način zapravo `\thesubsection` naredba postaje isto što i `\arabic{section}` naredba. Kod brojača `subsection`, `subsubsection`, itd. često želimo imati oznake kao **2.3 Šume**, **2.3.1 Bjelogorica** gdje broj **2** dolazi od brojača okoline `section`, **3** od brojača okoline `subsection`, a **1** od brojača okoline `subsubsection`. To se za okolinu `\subsection` postiže naredbama

```
\renewcommand{\thesection}{\arabic{section}}
\renewcommand{\thesubsection}{\thesection.\arabic{subsection}}.
```

⁷...

⁵...

Ako se ovaj način ispisivanja brojača okoline *subsection* koristi u cijelom dokumentu, prirodno je smjestiti ove `\renewcommand` naredbe u preambulu. O `\renewcommand` naredbama vidi slijedeću točku. Napomenimo još da se na slični način mogu generirati izlazni brojevi za *equation* okolinu.

6 Posebne operacije

6.1 Definiranje makro naredbi — naredbe “newcommand” i “renewcommand”

Ako nam se u dokumentu često pojavljuje određeni tekst ili komanda možemo koristiti pokratu koju definiramo naredbom `\newcommand`. Npr. ako se često pojavljuje naziv časopisa *Journal of Theoretical Biology* možemo u preambuli napisati:

```
\newcommand{\JTB}{\em Journal of Theoretical Biology\}
```

U tekstu dokumenta ćemo umjesto pisanja

```
{\em Journal of Theoretical Biology\}
```

koristiti `\JTB`. Dakle upotreba je jednostavna: argument u prvoj vitičastoj zagradi predstavlja tzv. makro naredbu dok drugi argument predstavlja tekst koji će se umetnuti umjesto makro naredbe.

U našem jeziku zgodno je umjesto č š ž Č Š Ž koristiti `\cx \sx \zx \CX \SX \ZX`, a umjesto ć i Ć pisati `\cy \CY`. Uočimo da se iza korištene makro naredbe mora nalaziti jedno prazno mjesto da \LaTeX zna gdje je kraj makro naredbe. Npr. rečenicu “Čakovčić je pametan.” smo napisali u ulaznoj datoteci ovako: ‘`\CX akov\cx i\cy\ je pametan.`’. Uočimo da iza `\cy` slijedi i znak `\`. To je zato jer \LaTeX dvije praznine shvaća kao jednu, pa bi izostalo prazno mjesto između riječi. Možda je još bolje koristiti direktno \TeX ove naredbe `\v` i `\'` pomoću kojih gornju rečenicu napišemo ovako: ‘`\v Cakov\v ci\ ' c je pametan.`’.

Isto tako `\newcommand` se može koristiti kao pokrata za neke duže naredbe. Npr. umjesto `\begin{equation}` možemo pisati `\be`, dok umjesto `\end{equation}` možemo pisati `\ee`. Ili npr. ako nam se često javlja grčko slovo γ možemo ga redefinirati makro naredbom

```
\newcommand{\ga}{\$\gamma$}
```

ako se javlja u običnom tekstu ili naredbom

```
\newcommand{\ga}{\gamma}
```

ako se uglavnom javlja u matematičkoj okolini (npr. u formulama). U makro definicijama možemo koristiti i promjenjive parametre. Npr. ako nam se često javljaju matricni elementi tipa $x_{ij}^{[k]}$, možemo definirati:

```
\newcommand{\mat}[3]{\{#\1\}^{\{#\2\}}_{\{#\3\}}}
```

Vidimo da naredba `\newcommand` započinje imenom makro naredbe koju definiramo, zatim se pojavljuje broj parametara u uglatoj zagradi (maksimalno 9 parametara) i konačno tekst kojim \LaTeX zamjenjuje makro naredbu u tekstu, pri čemu umjesto parametara upotrebljavamo oznake “# n” za n-ti parametar. $a_{48}^{[9]}$ možemo napisati koristeći `\mat{a}{9}{48}`.

Ako želimo mijenjati postojeću naredbu koristimo naredbu `\renewcommand{ }{ }`. U prvoj zagradi pišemo naredbu koju hoćemo promijeniti, a u drugoj opet tekst kojim ju zamjenjujemo. Npr. ako u preambuli napišemo

```
\renewcommand{\bf}{\it}
```

\LaTeX će prije slaganja teksta svako pojavljivanje `\bf` zamijeniti s `\it`.

6.2 Naredbe za definiranje okolina “newenvironment” i “newtheorem”

Pomoću naredbe `\newenvironment` možemo definirati vlastite okoline. Evo primjera. Ako pišemo poeziju i želimo za stihove koristiti posebni tip slova možemo to načiniti ovako. U preambuli napišemo

```
\newenvironment{mpoe}{\begin{verse} \em}{\end{verse}}
```

U tekstu okolina započne s `\begin{mpoe}` a završimo s `\end{mpoe}`. U vitičastim zagradama iza naredbe `\newenvironment` slijedi naziv nove okoline, a zatim naredbe koje L^AT_EX treba izvršiti na početku odnosno na kraju okoline.

Matematičari, logičari, filozofi i drugi vole svoje aksiome, definicije, teoreme, primjere i sl. napisati na poseban način. Za to se koriste posebne okoline. One se definiraju u preambuli pomoću naredbe `\newtheorem`. Ako u preambuli napišemo

```
\newtheorem{teorem}{\bf Teorem}
```

U ulaznoj datoteci ćemo napisati recimo

```
\begin{teorem}
```

Ako je $a=b$ i $b=c$ onda je $a=c$.

```
\end{teorem}
```

```
\begin{teorem}
```

Ako je $a < b$ i $b < c$ onda je $a < c$.

```
\end{teorem}
```

Dobit ćemo

Teorem 6.1 *Ako je $a = b$ i $b = c$ onda je $a = c$.*

Teorem 6.2 *Ako je $a < b$ i $b < c$ onda je $a < c$.*

Svaka okolina definirana pomoću naredbe `\newtheorem` ima svoj brojač. Ako želimo da se brojač vrati na nulu iza svakog poglavlja napisat ćemo

```
\newtheorem{teorem}{\bf Teorem}[chapter]
```

Ako upotrebljavamo *article* stil možemo zamijeniti `chapter` recimo sa `section`. Ako želimo da se teoremi, definicije, leme, propozicije, posljedice, primjeri i slične okoline broje istim brojačem tada u preambuli ovako pišemo:

```
\newtheorem{teorem}{\bf Teorem}[section]
```

```
\newtheorem{de}[teorem]{\bf Definicija}
```

```
\newtheorem{lema}[teorem]{\bf Lema}
```

```
\newtheorem{prop}[teorem]{\bf Propozicija}
```

```
\newtheorem{prim}[teorem]{\bf Primjer}
```

```
\newtheorem{poslj}[teorem]{\bf Posljedica}
```

Kada u tekstu koristimo neku od okolina definiranih pomoću naredbe `\newtheorem` imamo još jednu mogućnost. Evo primjera. Neka su okoline definirane kao gore. Ako napišemo:

```
\begin{poslj}[Banach]
```

Ako je $x > y$ i $y > z$ tada je $x > z$.

```
\end{poslj}
```

dobit ćemo

Posljedica 6.3 (Banach) *Ako je $x > y$ i $y > z$ tada je $x > z$.*

A Osnove rada računala

Elektroničko računalo je uređaj ili sustav koji može automatski izvršavati niz operacija zadanih na unaprijed određeni način. Najčešće primjenjivane operacije su obrada podataka i računanje. Niz naredbi kojim su određene operacije koje računalo provodi nazivamo program. Prvo računalo opće namjene bilo je ENIAC — *Electronic Numerical Integrator and Calculator*. Konstruirali su ga John W. Mauchly i J. Prosper Eckert Jr. na University of Pennsylvania's Moore School u vremenu od 1943–46, a služilo je za izvođenje složenih balističkih proračuna.

Svi podaci koje računalo obrađuje numerički su kodirani u binarnom brojevnom sustavu. Dok je za zapisivanje broja u dekadskom brojevnom sustavu potrebno deset različitih znakova koji predstavljaju znamenke broja, u binarnom sustavu koristimo samo dva znaka 0 i 1. Položaji koji u dekadskom zapisu odgovaraju potencijama broja 10: 1, 10, 100, 1000 itd. u binarnom zapisu odgovaraju potencijama broja 2: 1, 2, 4, 8 itd. Tako je binarni broj 1101001 jednak dekadskom $2^6 + 2^5 + 2^3 + 2^0 = 64 + 32 + 8 + 1 = 105$. Na taj način moguće je sve brojeve prikazati nizom elektroničkih komponenti koje mogu biti u dva stanja - provodljivom (što znači da kroz njih može teći električna struja) ili neprovodljivom (što znači da se ponašaju kao izolatori i ne provode električnu struju). Uz takav prikaz brojeva moguće je pomoću elektroničkih komponenti vršiti operacije računanja i uspoređivanja na kojima se zasniva rad elektroničkih računala. Time smo riješili zapisivanje brojeva, međutim računala često koristimo za obradu nenumeričkih podataka. Da bismo prikazali slova i ostale znakove u računalu, moramo ih kodirati korištenjem brojeva. Postoji više sustava za kodiranje slova i brojeva pomoću binarnih kodova od kojih su najpoznatiji ASCII i EBCDIC. Osobna računala na kojima ćemo mi raditi predstavljaju znakove u ASCII kodu. Riječ ASCII je pokratak za American Standard Code for Information Interchange (američki standardni kod za razmjenu informacija). Ova j kod uveden je u upotrebu 1963. godine i vrlo je raširen. Znakovi su predstavljeni sedmeroznamenastim binarnim brojevima od 0 do 127. Moguće je prikazati 128 različitih znakova, ali prva 32 koda služe za upravljanje uređajima kojima se šalje tekst, pa se stvarno koristi samo 96 kodova kojima su predstavljena velika i mala slova engleske abecede, znakovi interpunkcije i neki znakovi računskih operacija i brojevnih odnosa. U novije vrijeme koristi se takozvani prošireni ASCII standard koji koristi osmeroznamenaste binarne brojeve za prikaz znakova. Na taj način moguće je prikazati 256 različitih znakova. U proširenom ASCII skupu prvih 128 znakova podudara se s onima u osnovnom skupu, a slijedećih 128 znakova nije standardizirano, već se razlikuje od uređaja do uređaja. U računarskom nazivlju jednu binarnu znamenku nazivamo bit (od engl. binary digit = binarna znamenka). Bit je ujedno i mjera za najmanju količinu informacije u teoriji informacija. Osmeroznamenasti binarni broj (kojim je predstavljen jedan znak u ASCII kodu) zovemo byte.

A.1 Dijelovi računala

Osnovni dijelovi računala su centralna procesna jedinica, centralna memorija, vanjska memorija i ulazno-izlazni uređaji. Ti dijelovi čine elektroničku osnovu rada računala. Elektroničke i mehaničke dijelove računala nazivamo prema engleskom nazivu *hardware*.

Centralna procesna jedinica je najvažniji dio računala. To je onaj dio koji direktno izvršava naredbe programa. Centralne procesne jedinice nekih računala sastoje se iz više elektroničkih komponenti, a u suvremenim mikroracunlima cijela centralna procesna jedinica nalazi se u jednom čipu koji zovemo mikroprocesor. Snaga procesora mjeri se brzinom rada i veličinom podataka koje procesor može obraditi izvršavanjem jedne naredbe. Brzinu rada procesora možemo mjeriti na više načina — vremenom trajanja najjednostavnije operacije koju mikroprocesor može izvršiti (osnovni ciklus mikroprocesora, npr. 200 nanosekundi), frekvencijom osnovnog ciklusa

(npr. 5 MHz) ili brojem složenijih računskih operacija koje mikroprocesor izvrši u sekundi (tj. brojem FLOP-ova u sekundi, što dolazi od floating-point operation, npr. 100000 FLOP-ova). Veličina podatka koji mikroprocesor može obraditi izvršavanjem jedne naredbe mjeri se u bitima, pa tako govorimo o 8-bitnim, 16-bitnim i 32-bitnim mikroprocesorima.

Danas u svijetu ima više proizvođača mikroprocesora npr. Intel Corp., Zilog, Motorola, Texas Instruments, DEC, Bell Lab i dr. Mi ćemo vježbe održati na računalima tipa PC. U PC računala ugrađeni su međusobno kompatibilni Intelovi mikroprocesori od kojih noviji i brži mikroprocesori mogu izvršavati sve naredbe svojih prethodnika, uz cijeli niz novih naredbi. Tako je npr. u prva PC računala i kasnija PC XT računala ugrađen mikroprocesor INTEL 8088 8-bitni mikroprocesor s osnovnim ciklusom od 4,77 MHz. Novija AT i PS/2 računala imaju ugrađen 16-bitni mikroprocesor 8086 ili 80286 istog proizvođača s osnovnim ciklusom 6 do 16 MHz, a još novija računala 32-bitni mikroprocesor 80386 osnovnog ciklusa i do 30 MHz, po kojem ova računala i nazivamo PC-386. INTEL je razvio i mikroprocesor 80486, osnovnog ciklusa i do 66 MHz koji se ugrađuje u računala PC-486.

Centralna memorija je dio računala u kojem su u toku rada računala pohranjeni programi koje računalo izvršava i podaci koje obrađuje. Centralnu memoriju računala možemo podijeliti u dva tipa tzv. RAM i ROM. RAM (od engl. random access memory) je memorija čiji sadržaj centralna procesna jedinica može čitati ali i mijenjati. U RAM-u su pohranjeni podaci i izvršne naredbe tj. program. Isključivanjem računala cijeli sadržaj RAM-a se briše. ROM (od engl. read only memory) je memorija iz koje je moguće jedino čitati, tj. čiji sadržaj se ne može mijenjati. Tu su pohranjeni programi koji služe za početno pokretanje računala, tj. koji sa vanjske memorije učitavaju i pokreću operacioni sustav. Sadržaj ROM memorije ne briše se isključivanjem računala. U obični ROM sadržaj se zapisuje u toku proizvodnje i nije ga kasnije moguće mijenjati. Postoje i razne vrste memorijskih čipova čiji sadržaj možemo posebnim postupkom izbrisati (npr. EPROM - ultraljubičastom svjetlosti, EEROM - posebnim električnim signalima) i zatim u posebnim uređajima prepisati drugim podacima ili programima. Najmanja jedinica informacije u centralnoj memoriji koju centralna procesna jedinica može pročitati ili zapisati obično je 1 byte. Mjesto u centralnoj memoriji sa kojeg centralna procesna jedinica čita ili na koje piše zovemo adresom. Adresa je redni broj byta u memoriji i kreće se od 0 pa do najvećeg broja koji centralna procesna jedinica može obraditi. Prema tome količina centralne memorije ugrađene u računalo ograničena je kapacitetom centralne procesne jedinice. Kako je maksimalni binarni broj koji centralna procesna jedinica može obraditi ograničen potencijom broja 2, i količina memorije obično se izražava potencijama broja 2. Ipak, mi smo previše navikli na dekadski sustav, pa je $2^{10} = 1024$, što je blizu 1000, izabrana kao osnova za veću izvedenu jedinicu. 1024 byta nazivamo kilobyte i označavamo Kb, analogno izvedenoj jedinici 1000 puta većoj od osnovne kod ostalih mjera, a $2^{20} = 1048576$ byta nazivamo megabyte i označavamo Mb. U PC računala obično je ugrađeno između 256 Kb i 4 Mb RAM-a.

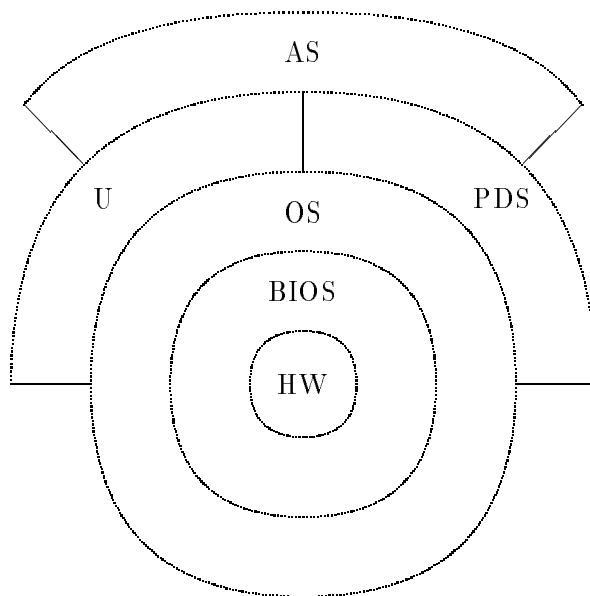
Vanjska memorija su uređaji velikog kapaciteta koji služe za čuvanje datoteka s podacima i programima. Najčešće korišteni uređaji su čitač/pisač magnetne trake i disketni i disk pogoni. U novije vrijeme često se koriste kazetni uređaji tzv. streameri za čuvanje sigurnosnih kopija podataka. Najsuremeniji uređaji za pohranjivanje velike količine podataka su pogoni optičkih diskova. Zbog vrlo velike količine podataka koji na njih stanu, te zbog nemogućnosti brisanja podataka, optički diskovi se najviše koriste za trajno pohranjivanje arhivskih podataka, kao i za distribuciju velikih količina podataka kao npr. enciklopedija i sekundarnih znanstvenih publikacija. Magnetne trake, diskete, tvrde diskove, optičke diskove i kazete na kojima se čuvaju podaci nazivamo medijima za pohranjivanje podataka.

Ulazno/izlazni uređaji su uređaji putem kojih računalo komunicira s korisnikom. Putem ulazno/izlaznih uređaja računalo učitava podatke kao i izvorne programe, te predaje rezultate

obrade korisniku, a također i predaje upravljačke naredbe drugim uređajima koji su na računalo spojeni (npr. kod računala koji služe za automatsko upravljanje strojevima i sl.). Povijesni su ulazno/izlazni uređaji čitač i pisač bušene papirne trake i kartica, koje danas nalazimo samo u muzejima. Suvremena računala komuniciraju s okolinom putem tastature, miša, digitalizatora slike i zvuka, kugle, elektronske olovke, video kamere i niza drugih uređaja za unos podataka iz okoline u računalo, te video monitora, raznih vrsta pisača i crtača i drugih uređaja za prikazivanje rezultata rada računala.

A.2 Programska podrška

Da bi računalo moglo raditi nužne su naredbe koje upravljaju njegovim radom. Kao što smo već rekli niz naredbi kojima upravljamo radom računala nazivamo program. U računarskoj terminologiji programe skupnim imenom nazivamo engleskom riječi *software*. Sve programe na računalu možemo promatrati u hijerarhiji programa, koju šematski možemo prikazati koncentričnim krugovima (slika 1). Na najnižem nivou ove hijerarhije nalaze se osnovni ulazno-izlazni programi (BIOS = basic input/output system). Na slijedećem nivou nalazi se operacioni sustav (OS = operating system). Zatim slijede razni uslužni programi (U = utilities) i razvojni sustav (PDS = program development system). Na posljednjem nivou nalaze se korisnički programi (AS = application software).



Slika 1: Hijerarhija programske podrške na računalima (HW - hardware, BIOS - osnovni ulazno-izlazni sustav, OS - operacioni sustav U - uslužni programi, PDS - razvojni sustav, AS - korisnički programi)

BIOS čine podprogrami koji omogućavaju drugim programima komuniciranje sa svim dijelovima računala, kao i s ulazno-izlaznim uređajima. Iako svako računalo ima specifični BIOS prilagođen njegovoj građi, ipak možemo govoriti i o standardizaciji, u smislu istog načina pozivanja podprograma BIOS-a i prenošenja parametara. Na PC računalima ovi programi smješteni su u ROM ugrađen u računalo.

Operacioni sustav direktno komunicira s korisnikom, te omogućava pozivanje odnosno učitavanje drugih programa s vanjskih memorijskih jedinica i njihovo pokretanje, te posreduje između korisnika odnosno korisničkih programa i osnovnog ulazno-izlaznog sustava. Na raznim računalima nalazimo razne operacione sustave. Dok na super računalima i velikim računarskim sustavima obično nalazimo specijalne operacione sustave, na mini i osobnim računalima sve više se pojavljuju tzv. standardni operacioni sustavi. Standardni pri tome znači da su u komunikaciji s korisnikom odnosno ulazno-izlaznim uređajima jednaki, dok je sam program u mašinskom kodu za svako računalo posebno pisan. Od takvih operacionih sustava najpoznatiji su UNIX na mini računalima koji je postao de facto standard za mini računala, te MS-DOS na PC računalima. Važno je spomenuti još i CP/M koji je prvi standardizirani jednokorisnički operacioni sustav na malim računalima, te OS/2 za PC računala koji nije tako opće prihvaćen kao MS-DOS. Na vježbama ćemo se služiti PC računalima koji rade pod MS-DOS-om.

Uslužni programi, kao što im i samo ime kaže, pružaju niz usluga korisnicima računala, kao npr. inicijalizaciju medija za pohranjivanje podataka (disketa i diskova), kopiranje podataka, održavanje sustava datoteka na tvrdom disku itd.

Razvojni sustav služi profesionalnim sistem analitičarima, programerima i koderima za razvoj korisničkih programa. Sastoji se od editora - programa za unos i ispravljanje podataka i programa u izvornom obliku i raznih programskih jezika te sustava za pomoć pri pisanju, ispravljanju i testiranju programa.

Programske jezike možemo podijeliti u nekoliko grupa po raznim kriterijima. Prva takva podjela je na niže i više programske jezike. Ova podjela odnosi se na sličnost naredbi programskog jezika i strojnih naredbi koje centralna procesna jedinica jedino razumije. Niži programski jezici su oni kod kojih naredbe programskog jezika točno odgovaraju naredbama centralne procesne jedinice. Takvi jezici samo su zapis riječima (tzv. mnemonicima) binarnog koda koji upravlja centralnom procesnom jedinicom. Takve jezike nazivamo zajedničkim imenom *Assembleri*. Naravno, assemblera ima isto toliko koliko i različitih procesora, a i za iste procesore često na tržištu možemo naći assemblere raznih proizvođača. Kod viših programskih jezika pojedine naredbe ne odgovaraju osnovnim naredbama procesora, već su bliže našem ljudskom načinu zadavanja algoritma i rješavanja problema. Prema specifičnosti problema za čije su rješavanje namijenjeni postoji i cijeli niz različitih programskih jezika. Od najpoznatijih nabrojat ćemo neke:

Jezici opće namjene

- | | |
|--------|---|
| ALGOL | <u>al</u> gorithmic <u>l</u> anguage — naziv obitelji programskih jezika opće namjene velike važnosti za razvoj programskih jezika. Prvi jezik ove obitelji stvoren je 1958. godine, a kasnije verzije nazivane su prema godini u kojoj su nastale ALGOL60 i ALGOL68. Jezik je mnogo češće primjenjivan u Evropi, dok u SAD nije uspio istisnuti uvriježeni FORTRAN. Mnogo je važnije što je rad na ALGOL-u indirektno doveo do razvoja Pascala, u kojem nalazimo mnogo strukturnih sličnosti s ALGOL-om. |
| BASIC | <u>b</u> eginner's <u>a</u> ll-purpose <u>s</u> ymbolic <u>i</u> nstruction <u>c</u> ode. Vrlo jednostavan jezik opće namjene razvijen sredinom 60-tih. Vrlo je čest na malim računalima, ali nažalost nestandardiziran. |
| C | jezik koji je prvenstveno razvijen za programiranje sistemskih programa u okviru UNIX operacionog sustava, ali se danas sve više koristi i u drugim okruženjima i za ostale namjene. |
| Pascal | programski jezik opće namjene razvijen s namjenom za podučavanje programiranja. Izrazito je čiste logičke strukture. Ovaj jezik vrlo je popularan na Sveučilištima. |

PL/I jezik opće namjene koji je zamišljen kao spoj mogućnosti programskih jezika namijenjenih numeričkim obradama i obradama baza podataka. Razvijen je u IBM-u gdje su se nadali da će PL/I zamijeniti COBOL, FORTRAN i ALGOL. Ovaj jezik nije općenito prihvaćen među ostalim proizvođačima računarske opreme.

Jezici namijenjeni numeričkim obradama

FORTRAN formula translation - vrlo popularan programski jezik namijenjen za izvođenje znanstvenih proračuna. Prvu verziju FORTRAN-a napravio je IBM 1956. godine. U novije vrijeme jezik je standardiziran te je danas općenito u upotrebi verzija FORTRAN77.

Jezici namijenjeni obradi baza podataka

COBOL common business-oriented language - jezik za knjigovodstvene primjene nastao 60-tih godina. Potpuno je standardiziran, a primjenjuje se na cijelom nizu mini i velikih računala. U novije vrijeme postoji i na jačim osobnim računalima.

dBase III+ noviji jezik za obradu baza podataka. Posebno je razvijen za primjenu na osobnim računalima.

ORACLE jedan od novijih jezika za obradu baza podataka

Jezici koji se primjenjuju u području umjetne inteligencije

LISP list processing — jezik za obradu lista. To je najstariji jezik namijenjen razvoju programa umjetne inteligencije.

PROLOG noviji jezik baziran na obradi pravila zaključivanja.

Jezici za obrade u realnom vremenu

Ada jezik za obrade u realnom vremenu — npr. kontroliranje raznih uređaja. Razvijen je u SAD na narudžbu Ministarstva obrane 1980. godine, a nazvan imenom Auguste Ade Lovelace, koju kao asistenta Charlesa Babbage-a smatramo prvim programerom.

Druga podjela jezika je na prevodiocce (compilere) i tumače (interpretere). Prevodioci su programi koji tzv. izvorni kod — tj. tekst programa pisanog u višem programskom jeziku, prevode u strojni kod koji procesor može direktno učitati u memoriju računala i izvršiti. Tumači su programi koji čitaju jednu po jednu naredbu pisanu u višem programskom jeziku i zatim sami izvršavaju operacije specificirane tom naredbom. Mnogi programski jezici postoje u oba oblika. Prevodioci imaju prednost što se prevedeni program znatno brže izvršava, a moguće je vršiti i optimizaciju s obzirom na prostor koji program zauzima u centralnoj memoriji ili brzinu izvođenja programa. Mana prevodilaca je što prilikom ispravljanja programa moramo ponoviti cijeli ciklus ispravljanja teksta u editoru i prevođenja programa, što kod većih programa može trajati vrlo dugo. S druge strane tumači vrlo sporo izvršavaju programe, ali je ispravljanje znatno jednostavnije jer nije potrebno prolaziti kroz ciklus prevođenja u strojni kod. Danas se sve češće pojavljuju jezici u kombinaciji prevodioca i tumača, koji omogućavaju razvoj i ispravljanje programa uz upotrebu tumača, a konačni program se zbog efikasnosti prevodi prevodiocem i krajnji korisnik ima na raspolaganju prevedeni program.

U grupu korisničkih programa ulaze npr. programi za obradu teksta, razni statistički paketi, razni računovodstveni programi, programi za projektiranje pomoću računala i dr. Od ovih programa vama će najčešće biti potrebni programi za obradu teksta i statistički paketi.

Programi za obradu teksta u potpunosti zamjenjuju pisaču mašinu i daktilografa. Korištenje takvih programa omogućava autoru samostalno ispravljanje i prepravljavanje teksta, a konačni ispis automatski se formatizira. Od programa za obradu teksta na PC računalima najpoznatiji su WordPerfect, Wordstar, Word, Multimate, a ima i niz drugih. Uz ove programe postoji i niz programa za slaganje teksta pomoću računala, kao što su Ventura Publisher i sl. Jedan od takvih programskih paketa je i \LaTeX tj. \TeX .

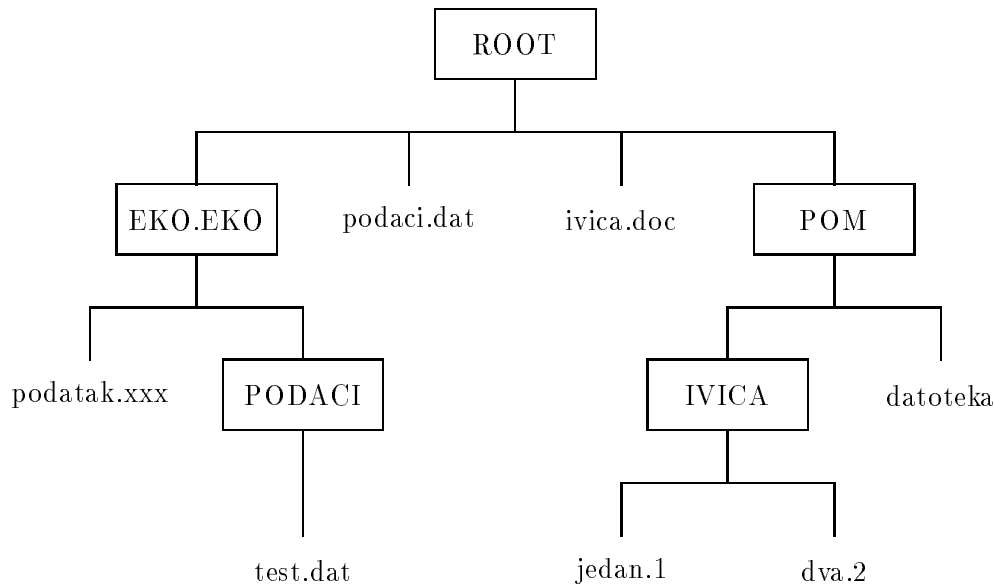
Statistički paketi su složeni programski sustavi koji omogućavaju provođenje statističkih obrada bez poznavanja programskih jezika i programiranja. Od statističkih paketa na PC računalima poznati su SPSS/PC, Statgraphics, Microstat, SysStat, SAS/PC, a ima i drugih.

A.3 Osnove operacionog sustava MS-DOS

Većina PC računala radi pod operacionim sustavom MS-DOS. Naziv mu dolazi od Microsoft disk operating system. Prva riječ označava proizvođača — Microsoft, a daljnje riječi upućuju da se radi o operacionom sustavu čiji rad je vezan uz vanjsku memoriju na disketnom ili disk pogonu. Da bismo mogli upoznati osnove ovog operacionog sustava moramo prvo upoznati organizaciju podataka na disku ili disketi.

Svi podaci ili programi koji su pohranjeni u vanjskoj memoriji pohranjuju se u obliku datoteke. Datoteka ili engleski “file” je niz znakova pohranjenih na disku, a koji čine jednu cjelinu i imenovani su nekim imenom. Ime datoteke sastoji se od dva dijela, prvi dio je osnovno ime koje ima najviše 8 znakova (tj. slova ili brojeva), a drugi dio je nastavak ili “extension” i ima najviše 3 znaka. Uobičajeno je pri pisanju osnovno ime datoteke odvojiti od nastavka točkom, pa tako možemo imati datoteke “xxxx.yyy” ili “234drzzz.ax”. Iako datoteke možemo imenovati kako god želimo, ipak zbog lakšeg snalaženja na računalu postoje neki standardni nastavci koji označavaju tip datoteke. Nije preporučljivo svoje datoteke imenovati nazivima koji se koriste za datoteke operacionog sustava. Uobičajeni su nastavci “exe” i “com” za izvršne programe. Svi izvršni (strojni) programi imaju ove nastavke, kako oni kupljeni, tako i oni koje smo dobili pomoću programskog prevodioca. Nastavak “bat” rezerviran je za tzv. “batch” programe koji se sastoje od niza naredbi operacionog sustava koje želimo izvršiti uvijek na isti način. Uobičajeni nastavci za tekst datoteke programa za obradu teksta je “doc” od engleskog “document”, a datotekama s podacima najčešće dajemo nastavak “dat”. Neki programi imaju posebni format zapisa podataka, pa tako pripremljenim datotekama sami daju nastavak specifičan za taj program. Ulazna datoteka za \LaTeX mora imati nastavak “tex”, a za FORTRAN prevodilac, ovisno o proizvođaču, može imati nastavak “for” ili “f77”.

Disk na PC računalu može biti vrlo velikog kapaciteta i sadržavati i nekoliko tisuća datoteka. Kada bismo trebali naći neku datoteku na neorganiziranom disku od nekoliko tisuća datoteka mogli bismo izgubiti mnogo vremena, a moglo bi nam se i dogoditi da više datoteka nazovemo istim imenom, pa tako prilikom stvaranja jedne datoteke izbrišemo sadržaj druge. Ovo je posebno opasno kada više osoba koristi isto PC računalo. Zato je disk organiziran u obliku hijerarhije tzv. direktorija. Direktorij možemo shvatiti kao posebnu datoteku koja sadrži popis datoteka i informacija o njihovoj veličini i položaju na disku (kao indeks časopisa koji je i sam časopis s informacijama o nazivu i sadržaju drugih časopisa, ali i mjestu gdje ih možemo naručiti). Osnovni direktorij diska nazivamo engleskim nazivom “root” (ili korijen). Svaki direktorij može osim običnih datoteka sadržavati i druge direktorije. Kada bismo nacrtali strukturu diska s direktorijima dobili bismo obrnuto stablo s korijenom na vrhu (slika 2).



Slika 2: Grafički prikaz strukture direktorija na disku. (imena direktorija su uokvirena i napisana velikim, a imena datoteka malim slovima)

Osim toga na PC računalu može biti i više disk i disketnih pogona. Diskove i disketne pogone označavamo pod MS-DOS-om slovima engleske abecede A, B, C, D, E itd. iza kojih slijedi dvotočka. Sam operacioni sustav ne razaznaje velika od malih slova pa je dopušteno pisanje npr. "A:" ili "d:". Direktoriji su i sami datoteke, pa ime direktorija mora zadovoljavati pravila imenovanja datoteka, uključujući i nastavak. Kada želimo pozvati neku određenu datoteku moramo navesti na kojem se disku nalazi, u kojem direktoriju i kako se zove. Pri tom ako se datoteka nalazi u direktoriju unutar nekog drugog direktorija moramo navesti oba direktorija. Nazive direktorija odvajamo od specifikacije diska, naziva datoteke i međusobno kosom crtom nagnutom u lijevo "\" (za razliku od običnog znaka "/"). Npr. datoteka "test.dat" na disku "d:" u direktoriju "podaci" koji se nalazi unutar direktorija "ivan" poziva se s "d:\ivan\podaci\test.dat". Pri tom se prvi dio punog naziva "d:\ivan\podaci" naziva put ili engleski "path" jer određuje put kojim možemo doći do datoteke.

Kada bismo uvijek morali navoditi puni naziv datoteka s kojima radimo morali bismo jako puno pisati. Kako bismo izbjegli pisanje dugih naziva, u MS-DOS-u postoji mogućnost da neki direktorij izaberemo kao tzv. radni disk i direktorij, odnosno disk i direktorij koji se podrazumijeva (engl. *default*). Nakon što to učinimo, poziv datoteke samo imenom i nastavkom podrazumijevati će datoteku tog imena u radnom direktoriju na radnom disku.

Kada operacionom sustavu zadamo neku naredbu, računalo ju neće izvršiti dok unos ne završimo tipkom ENTER koja označava da smo naredbu u potpunosti unijeli. Da bismo zadali radni disk dovoljno je utipkati slovo koje određuje taj disk i dvotočku, te naravno ENTER. Koji disk je radni disk možemo odrediti na prvi pogled jer nam se računalo obično javlja slovom koje određuje radni disk - ovako: **A>** ili **C>**. Radni direktorij možemo zadati naredbom "CD" ili duže "CHDIR" (što dolazi od engleskog "change directory" tj. promijeni direktorij), nakon koje slijedi naziv direktorija koji će se ubuduće podrazumijevati. Iako MS-DOS ne razlikuje velika od malih slova pri upisivanju naredbi i imena direktorija i datoteka, mi ćemo pri pisanju naredbi

dogovorno pisati velikim slovima tekst naredbe koji morate upisati doslovno kako piše, malim slovima opis teksta koji možete mijenjati ovisno o potrebi, a kurzivom primjer naziva kakav može stajati na tom mjestu. Naredba za promjenu direktorija glasi “CD ime-direktorija” npr. “CD *ivan*”. Ako željeni direktorij nije u sadašnjem radnom direktoriju moramo ispred njegovog imena staviti “\”, npr. “CD *ivan*”. Želimo li se vratiti u root direktorij dovoljno je utipkati “CD \”.

Naredbom “DIR” možemo vidjeti listu svih datoteka koje se nalaze u radnom direktoriju. Želimo li vidjeti sadržaj nekog drugog direktorija možemo iza naredbe dopisati odgovarajući put, npr. “DIR *c:\pctex\latex*”.

Pojedine datoteke možemo kopirati na drugi disk ili u drugi direktorij naredbom “COPY ime-datoteke-1 ime-datoteke-2”. Iza naredbe “COPY” slijede naziv datoteke koju kopiramo i mjesto i naziv datoteke u koju želimo kopirati. Ako ciljna datoteka već ne postoji MS-DOS će napraviti novu. Npr. datoteku “test.tex” na disku “c:” i u direktoriju “pctex\latex\tekst” kopiramo na disk “d:” u direktorij “tex” i datoteku “moj.tex” naredbom:

```
COPY c:\pctex\latex\tekst\test.tex d:\tex\moj.tex
```

koju naravno moramo završiti pritiskom na tipku ENTER.

Datoteke možemo i brisati naredbom “DEL” iza koje slijedi puni naziv datoteke. Npr. već spomenutu datoteku “test.dat” možemo izbrisati na slijedeći način:

```
DEL c:\pctex\latex\tekst\test.tex
```

ali pri tom treba jako paziti. I najmanja greška u utipkavanju ove naredbe može vas i vaše kolege koštati izgubljenog teksta za čije utipkavanje je utrošeno mnogo vremena i truda.

Program pohranjen na disku pozivamo jednostavno utipkavanjem njegovog imena (bez nastavka “exe” ili “com” - operacioni sustav sam traži datoteku odgovarajućeg nastavka). Operacioni sustav potom će učitati program sa diska u centralnu memoriju i započeti izvršavanje programa. Ukoliko program nije u radnom direktoriju na radnom disku može se dogoditi da operacioni sustav javi kako program nije našao. U tom slučaju treba pogledati naredbom “DIR” da li se program uopće nalazi u trenutno odabranom direktoriju i eventualno pomoću “CD” naredbe preći u direktorij u kojem se program nalazi. Slično se može dogoditi i s datotekama, pa ako naredbom “DIR” ne nađemo neku datoteku treba pogledati u neki drugi direktorij ili na drugi disk prije nego ustvrdimo da datoteke nema na disku.

Evo nekoliko primjera korištenja navedenih naredbi MS-DOS-a

C:	prelazak na disk c:
A:	prelazak na disk a:
CD \ <i>microsta</i>	prelazak u direktorij <i>microsta</i> na trenutačnom radnom disku
CD	povratak u root direktorij
DIR	ispis sadržaja radnog direktorija
DIR <i>prog*.dat</i>	ispis dijela sadržaja radnog direktorija, i to svih datoteka čije ime počinje na “prog”, i koje imaju nastavak “dat”
DIR <i>d:\podaci</i>	ispis sadržaja direktorija “podaci” na disku “d:”
COPY <i>podaci.dat a:</i>	kopiranje datoteke “podaci.dat” u radnom direktoriju u datoteku istog imena na disketu u pogonu “a:”
COPY <i>d:\podaci\ivica.dat</i>	kopiranje datoteke “ivica.dat” iz

direktorija “podaci” na disku “d:” u
datoteku istog imena u radnom direktorij

A.4 Upotreba programa L^AT_EX pod MS-DOSom

Kao što je već prije rečeno L^AT_EX pozivamo jednostavno upisujući u okruženju operacionog sustava:

L^AT_EX ime-datoteke

gdje je “ime-datoteke” naziv datoteke u kojoj se nalazi naš tekst, a koja mora imati nastavak *tex*. Npr. ako se naš tekst nalazi u datoteci *doc.tex*, tekst ćemo složiti pozivanjem programa:

L^AT_EX doc

L^AT_EX će kao rezultat napraviti datoteku *doc.dvi*, u kojoj će biti složen tekst u obliku koji ne ovisi o uređaju na koji se ispisuje. Ako želimo tekst vidjeti na ekranu, pozvat ćemo program za prikaz teksta:

VIEW doc

koji će prikazati izgled složenog teksta iz datoteke *doc.dvi* na ekranu. Za ispis na laserski pisač upotrijebiti ćemo program *dvihp*, koji će napraviti datoteku *doc.hp*:

DVIHP doc

Datoteku *doc.hp* koju smo dobili pomoću programa *dvihp* možemo ispisati na laserskom pisaču naredbom:

COPY *doc.hp* /B PRN:

Važno je uočiti opciju /B u ovoj naredbi, jer bez nje nećete moći ispisati svoj tekst.

B Podsjetnik za Norton Editor

Norton Editor je jedan od najjednostavnijih ekranskih editora, tj. programa za upis i uređivanje običnih tekstovnih (ASCII) datoteka na PC računalima. Pozivamo ga naredbom:

NE ime-datoteke

gdje je kao i u prethodnom poglavlju velikim slovima ispisan tekst koji moramo doslovno prepisati, a malim opis polja čiji sadržaj mijenjamo ovisno o našim potrebama. Npr. želimo li unijeti ili ispraviti tekst u datoteci “primjer.tex” u trenutačnom direktoriju pozvat ćemo Norton Editor naredbom “NE *primjer.tex*” gdje je tekst koji stoji kao primjer za moguće “ime-datoteke” napisan kurzivom. Ako datoteka pod navedenim imenom ne postoji, Norton Editor će napraviti novu datoteku s tim imenom. Drugi oblik poziva programa Norton Editor je:

NE ime-datoteke-1 ime-datoteke-2

Ovaj oblik poziva programa daje isti rezultat kao da smo utipkali:

COPY ime-datoteke-1 ime-datoteke-2

NE ime-datoteke-2

Prema tome prva datoteka mora postojati, a ako druga i postoji preko nje će se upisati sadržaj prve datoteke i zatim otvoriti za ispravljanje. Prije upotrebe drugog oblika poziva programa treba zato uvijek provjeriti da li već postoji datoteka takvog imena.

Kad pozovemo Norton Editor na ekranu se prvo pojavi naslovna strana s nazivom programa i proizvođača s uputom da stisnemo bilo koju tipku, a potom se pojavi početak sadržaja datoteke

(ili prazni ekran, ako je datoteka prazna). Na dnu ekrana nalazi se redak u kojem piše u kojem retku i stupcu datoteke se nalazimo, ime datoteke koju uređujemo, te još neke obavijesti — da li smo u načinu za umetanje (*Insert*) ili pretipkavanje (*Replace*), da li je uključeno automatsko prelaženje u novi redak ($WW=nn$, gdje je nn duljina retka) ili ne ($WW=Off$) i da li je uključeno automatsko uvlačenje novih redaka (*Indent*). Na ekranu se nalazi i jedan poseban znak koji treperi. To je tzv. kurzor koji pokazuje gdje se u tekstu trenutačno nalazimo, odnosno gdje će se pojaviti slijedeći utipkani znak. U slijedećim tablicama prikazane su naredbe za uređivanje teksta.

NAREDBE ZA POMICANJE KURZORA

←	- Pomakni kurzor lijevo
→	- Pomakni kurzor desno
↑	- Pomakni kurzor gore
↓	- Pomakni kurzor dolje
Ctrl + ←	- Pomakni kurzor za riječ lijevo
Ctrl + →	- Pomakni kurzor za riječ desno
Home	- Postavi kurzor na početak retka
End	- Postavi kurzor na kraj retka
PgUp	- Pomakni kurzor za jedan ekran gore
PgDn	- Pomakni kurzor za jedan ekran dolje

NAREDBE ZA BRISANJE

⇐	- Znak lijevo od kurzora
Del	- Znak na kojem je kurzor
Ctrl + W	- Riječ lijevo
Alt + W	- Riječ desno
Ctrl + L	- Do početka retka
Alt + L	- Do kraja retka
Alt + K	- Cijeli redak
Ctrl + U	- Vрати izbrisano

NAREDBE ZA RAD S DATOTEKAMA

F3	E	- Spremi i završi
F3	S	- Spremi i nastavi rad
F3	Q	- Odustani
F3	N	- Otvori novu datoteku
F3	X	- Prijedi u novi prozor
F3	W	- Zapiši tekst do kurzora u datoteku
F3	L	- Učitaj još teksta iz datoteke
F3	A	- Dodaj sadržaj datoteke
F3	C	- Zatvori izlaznu datoteku

NAREDBE ZA RAD S BLOKOVIMA

F4	S	- Postavi oznaku bloka
F4	R	- Izbriši oznake bloka
F4	D	- Izbriši blok
F4	C	- Kopiraj blok
F4	W	- Kopiraj blok iz prozora
F4	M	- Preseli blok
F4	L	- Označi redak
F4	E	- Označi do kraja retka
F4	F	- Nađi oznaku bloka

UPRAVLJANJE OBLIKOVANJEM TEKSTA

F5	L	- Zadađ dužinu retka
F5	W	- (U/Is)ključi automatski prelazak u novi red
F5	F	- Složi odlomak
F5	T	- Zadađ razmak za tabulatore
F5	C	- Izbor vrste kurzora
F5	D	- Izbor boje ekrana
F5	I	- (U/Is)ključi uvlačenje redaka
F5	S	- Spremi trenutačnu konfiguraciju editora

NAREDBE ZA ISPISIVANJE NA PISAČU

F7	P	- Ispiši sadržaj memorije
F7	B	- Ispiši blok
F7	E	- Pomakni papir na početak stranice
F7	S	- Zadađ dužinu stranice
F7	M	- Zadađ lijevu marginu

NAREDBE ZA PRETRAŽIVANJE

Alt	+	F	- Traži niz znakova unaprijed
Ctrl	+	F	- Traži niz znakova unazad
Alt	+	C	- Nastavi tražiti unaprijed
Ctrl	+	C	- Nastavi tražiti unazad

DODATNE NAREDBE

F1		- Pomoć	
F2		- Status Editora	
F9		- Privremeni izlaz u DOS	
Ins		- Uključivanje umetanja teksta	
F6	Ins	- Uključivanje prepisivanja teksta	
F6	G	- Postavljanje na zadani redak	
F6	M	- Nađi odgovarajuću zagradu	
F6	C	- Sažeti ispis	
Ctrl	+	P	- Umetni kontrolni znak
Alt	+	V	- Zamijeni velika i mala slova do kraja retka

Ako želimo automatski zamijeniti neki tekst drugim možemo upotrijebiti produženi oblik naredbe za traženje. Za pretraživanje unaprijed nakon što utipkamo

Alt

 +

F

 unesemo

tekst koji želimo naći, potom opet utipkamo `Alt` + `F` i zatim tekst kojim nađeni tekst želimo zamijeniti. Na kraju pritisnemo tipku `ENTER` ili `ESC`. Tipka `ESC` znači da se pri traženju zanemari veličina slova, a tipka `ENTER` da se ona uzme u obzir. Želimo li potražiti znak za prelazak u novi redak možemo utipkati zajedno `CTRL+ENTER` u okviru teksta koji tražimo ili zamjenjujemo. Kod svakog nađenog niza znakova možemo birati postupak:

- Y - zamijeni
- N - nemoj zamijeniti
- * - zamijeni dalje bez pitanja

RAZMAKNICA - prekini pretraživanje

Alt	+	C	- nastavi pretraživanje unaprijed
Ctrl	+	C	- nastavi pretraživanje unatrag

C Tablice posebnih i matematičkih oznaka

Tablica 2: Akcenti (samo za LR način)

ò	\`{o}	õ	\~{o}	ö	\v{o}	ø	\c{o}
ó	\' {o}	ō	\={o}	ö	\H{o}	ø	\d{o}
ô	\^{o}	ò	\. {o}	ôo	\t{oo}	ø	\b{o}
ö	\" {o}	ö	\u{o}				

Tablica 3: Strani znakovi (samo za LR način)

œ	\oe	â	\aa	ø	\o
Œ	\OE	Å	\AA	Ø	\O
æ	\ae	ł	\l	ı	?‘
Æ	\AE	Ł	\L	ı	!‘
ß	\ss				

D Tablica ASCII znakova

Tablica 4: Strani znakovi (za sve načine)

†	\dag	§	\S	©	\copyright
‡	\ddag	¶	\P	£	\pounds

Tablica 5: Grčka slova (samo matematički način)

<i>Mala slova</i>							
α	<code>\alpha</code>	θ	<code>\thetaeta</code>	o	<code>o</code>	τ	<code>\tauau</code>
β	<code>\betaeta</code>	ϑ	<code>\varthetaeta</code>	π	<code>\pi</code>	υ	<code>\upsilonlon</code>
γ	<code>\gammamma</code>	ι	<code>\iotaota</code>	ϖ	<code>\varpi</code>	ϕ	<code>\phi</code>
δ	<code>\deltaelta</code>	κ	<code>\kappaappa</code>	ρ	<code>\rho</code>	φ	<code>\varphi</code>
ϵ	<code>\epsilonpsilon</code>	λ	<code>\lambdambda</code>	ϱ	<code>\varrho</code>	χ	<code>\chi</code>
ε	<code>\varepsilonpsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ψ	<code>\psi</code>
ζ	<code>\zetaeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>	ω	<code>\omega</code>
η	<code>\etaeta</code>	ξ	<code>\xi</code>				
<i>Velika slova</i>							
A	<code>A</code>	H	<code>H</code>	N	<code>N</code>	T	<code>T</code>
B	<code>B</code>	Θ	<code>\Theta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>
Γ	<code>\Gamma</code>	I	<code>I</code>	O	<code>O</code>	Φ	<code>\Phi</code>
Δ	<code>\Delta</code>	K	<code>K</code>	Π	<code>\Pi</code>	X	<code>X</code>
E	<code>E</code>	Λ	<code>\Lambda</code>	P	<code>P</code>	Ψ	<code>\Psi</code>
Z	<code>Z</code>	M	<code>M</code>	Σ	<code>\Sigma</code>	Ω	<code>\Omega</code>

Tablica 6: Znakovi binarnih operacija (samo matematički način)

\pm	<code>\pm</code>	\cap	<code>\cap</code>	\diamond	<code>\diamond</code>	\oplus	<code>\oplus</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\triangle	<code>\bigtriangleup</code>	\ominus	<code>\ominus</code>
\times	<code>\times</code>	\uplus	<code>\uplus</code>	∇	<code>\bigtriangledown</code>	\otimes	<code>\otimes</code>
\div	<code>\div</code>	\sqcap	<code>\sqcap</code>	\triangleleft	<code>\triangleleft</code>	\oslash	<code>\oslash</code>
$*$	<code>\ast</code>	\sqcup	<code>\sqcup</code>	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\star	<code>\star</code>	\vee	<code>\vee</code>	\triangleleft	<code>\lhd</code>	\bigcirc	<code>\bigcirc</code>
\circ	<code>\circ</code>	\wedge	<code>\wedge</code>	\triangleright	<code>\rhd</code>	\dagger	<code>\dagger</code>
\bullet	<code>\bullet</code>	\setminus	<code>\setminus</code>	\triangleleft	<code>\unlhd</code>	\ddagger	<code>\ddagger</code>
\cdot	<code>\cdot</code>	\wr	<code>\wr</code>	\triangleleft	<code>\unrhd</code>	\amalg	<code>\amalg</code>

Tablica 7: Znakovi relacija (samo matematički način)

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>	\models	<code>\models</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>	\perp	<code>\perp</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>	\mid	<code>\mid</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\asymp	<code>\asymp</code>	\parallel	<code>\parallel</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>	\bowtie	<code>\bowtie</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>	\Join	<code>\Join</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\neq	<code>\neq</code>	\smile	<code>\smile</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\doteq	<code>\doteq</code>	\frown	<code>\frown</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\propto	<code>\propto</code>		
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>				

Tablica 8: Strelice (samo matematički način)

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>
\Rrightarrow	<code>\Rrightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>
\hookleftarrow	<code>\hookleftarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\leadsto	<code>\leadsto</code>
\uparrow	<code>\uparrow</code>	\downarrow	<code>\downarrow</code>
\Uparrow	<code>\Uparrow</code>	\Downarrow	<code>\Downarrow</code>
\updownarrow	<code>\updownarrow</code>	\mapsto	<code>\mapsto</code>
\Updownarrow	<code>\Updownarrow</code>	\longmapsto	<code>\longmapsto</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>

Tablica 9: Razne oznake (samo matematički način)

\aleph	<code>\aleph</code>	$'$	<code>\prime</code>	\forall	<code>\forall</code>	<code>\forall</code>	<code>\forall</code>	∞	<code>\infty</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\square	<code>\Box</code>	\square	<code>\Box</code>
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\diamond	<code>\Diamond</code>	\triangle	<code>\triangle</code>
\jmath	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>	\triangle	<code>\triangle</code>	\clubsuit	<code>\clubsuit</code>
ℓ	<code>\ell</code>	\top	<code>\top</code>	\natural	<code>\natural</code>	\clubsuit	<code>\clubsuit</code>	\diamondsuit	<code>\diamondsuit</code>
\wp	<code>\wp</code>	\perp	<code>\bot</code>	\sharp	<code>\sharp</code>	\diamond	<code>\diamond</code>	\heartsuit	<code>\heartsuit</code>
\Re	<code>\Re</code>	\parallel	<code>\parallel</code>	\backslash	<code>\backslash</code>	\heartsuit	<code>\heartsuit</code>	\spadesuit	<code>\spadesuit</code>
\Im	<code>\Im</code>	\angle	<code>\angle</code>	∂	<code>\partial</code>	\spadesuit	<code>\spadesuit</code>		
\mho	<code>\mho</code>								

Tablica 10: Znakovi u dvije veličine (samo matematički način)

Σ	\sum	<code>\sum</code>	\cap	\bigcap	<code>\bigcap</code>	\odot	\bigodot	<code>\bigodot</code>
Π	\prod	<code>\prod</code>	\cup	\bigcup	<code>\bigcup</code>	\otimes	\bigotimes	<code>\bigotimes</code>
\amalg	\coprod	<code>\coprod</code>	\sqcup	\bigsqcup	<code>\bigsqcup</code>	\oplus	\bigoplus	<code>\bigoplus</code>
\int	\int	<code>\int</code>	\vee	\bigvee	<code>\bigvee</code>	\uplus	\biguplus	<code>\biguplus</code>
\oint	\oint	<code>\oint</code>	\wedge	\bigwedge	<code>\bigwedge</code>			

Tablica 11: Funkcije nalik na logaritam (samo matematički način)

<code>\arccos</code>	<code>\cos</code>	<code>\csc</code>	<code>\exp</code>	<code>\ker</code>	<code>\limsup</code>	<code>\min</code>	<code>\sinh</code>
<code>\arcsin</code>	<code>\cosh</code>	<code>\deg</code>	<code>\gcd</code>	<code>\lg</code>	<code>\ln</code>	<code>\Pr</code>	<code>\sup</code>
<code>\arctan</code>	<code>\cot</code>	<code>\det</code>	<code>\hom</code>	<code>\lim</code>	<code>\log</code>	<code>\sec</code>	<code>\tan</code>
<code>\arg</code>	<code>\coth</code>	<code>\dim</code>	<code>\inf</code>	<code>\liminf</code>	<code>\max</code>	<code>\sin</code>	<code>\tanh</code>

Tablica 12: Graničnici (samo matematički način)

<code>(</code>	<code>(</code>	<code>)</code>	<code>)</code>	<code>↑</code>	<code>\uparrow</code>
<code>[</code>	<code>[</code>	<code>]</code>	<code>]</code>	<code>↓</code>	<code>\downarrow</code>
<code>{</code>	<code>\{</code>	<code>}</code>	<code>\}</code>	<code>↕</code>	<code>\updownarrow</code>
<code>⌊</code>	<code>\lfloor</code>	<code>⌋</code>	<code>\rfloor</code>	<code>↑</code>	<code>\Uparrow</code>
<code>⌈</code>	<code>\lceil</code>	<code>⌋</code>	<code>\rceil</code>	<code>↓</code>	<code>\Downarrow</code>
<code>⟨</code>	<code>\langle</code>	<code>⟩</code>	<code>\rangle</code>	<code>↕</code>	<code>\Updownarrow</code>
<code>/</code>	<code>/</code>	<code>\</code>	<code>\backslash</code>		
<code> </code>	<code> </code>	<code> </code>	<code>\ </code>		

Tablica 13: Akcenti za matematički način

<code>â</code>	<code>\hat{a}</code>	<code>á</code>	<code>\acute{a}</code>	<code>ā</code>	<code>\bar{a}</code>	<code>ȧ</code>	<code>\dot{a}</code>
<code>ă</code>	<code>\check{a}</code>	<code>à</code>	<code>\grave{a}</code>	<code>ā</code>	<code>\vec{a}</code>	<code>ä</code>	<code>\ddot{a}</code>
<code>ã</code>	<code>\breve{a}</code>	<code>ã</code>	<code>\tilde{a}</code>				

Tablica 14: Tablica ASCII znakova

1	Δ	2	Θ	3	Λ	4	Ξ	5	Π	6	Σ	7	Υ	8	Φ
9	Ψ	10	Ω	11	ff	12	fi	13	fl	14	ffi	15	ffl	16	ı
17	j	18	`	19	'	20	˘	21	˙	22	-	23	°	24	,
25	ß	26	æ	27	œ	28	ø	29	Æ	30	Œ	31	Ø	32	ˆ
33	!	34	”	35	#	36	\$	37	%	38	&	39	'	40	(
41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0
49	1	50	2	51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	ı	61	=	62	ı	63	?	64	@
65	A	66	B	67	C	68	D	69	E	70	F	71	G	72	H
73	I	74	J	75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X
89	Y	90	Z	91	[92	“	93]	94	^	95	˘	96	‘
97	a	98	b	99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n	111	o	112	p
113	q	114	r	115	s	116	t	117	u	118	v	119	w	120	x
121	y	122	z	123	-	124	—	125	”	126	˘	127	”	128	

Tablica 15: Tablica brojeva u tri brojevna sustava

D	O	H	D	O	H	D	O	H	D	O	H	D	O	H
1	1	1	2	2	2	3	3	3	4	4	4	5	5	5
6	6	6	7	7	7	8	10	8	9	11	9	10	12	A
11	13	B	12	14	C	13	15	D	14	16	E	15	17	F
16	20	10	17	21	11	18	22	12	19	23	13	20	24	14
21	25	15	22	26	16	23	27	17	24	30	18	25	31	19
26	32	1A	27	33	1B	28	34	1C	29	35	1D	30	36	1E
31	37	1F	32	40	20	33	41	21	34	42	22	35	43	23
36	44	24	37	45	25	38	46	26	39	47	27	40	50	28
41	51	29	42	52	2A	43	53	2B	44	54	2C	45	55	2D
46	56	2E	47	57	2F	48	60	30	49	61	31	50	62	32
51	63	33	52	64	34	53	65	35	54	66	36	55	67	37
56	70	38	57	71	39	58	72	3A	59	73	3B	60	74	3C
61	75	3D	62	76	3E	63	77	3F	64	100	40	65	101	41
66	102	42	67	103	43	68	104	44	69	105	45	70	106	46
71	107	47	72	110	48	73	111	49	74	112	4A	75	113	4B
76	114	4C	77	115	4D	78	116	4E	79	117	4F	80	120	50
81	121	51	82	122	52	83	123	53	84	124	54	85	125	55
86	126	56	87	127	57	88	130	58	89	131	59	90	132	5A
91	133	5B	92	134	5C	93	135	5D	94	136	5E	95	137	5F
96	140	60	97	141	61	98	142	62	99	143	63	100	144	64
101	145	65	102	146	66	103	147	67	104	150	68	105	151	69
106	152	6A	107	153	6B	108	154	6C	109	155	6D	110	156	6E
111	157	6F	112	160	70	113	161	71	114	162	72	115	163	73
116	164	74	117	165	75	118	166	76	119	167	77	120	170	78
121	171	79	122	172	7A	123	173	7B	124	174	7C	125	175	7D
126	176	7E	127	177	7F	128	200	80	129	201	81	130	202	82

D=decimalni, O=oktalni, H=heksadecimalni