

L^AT_EX 2_ε ZA AUTORE

ALEKSANDAR SAMARDŽIĆ
GORAN NENADIĆ
PREDRAG JANIČIĆ

BEOGRAD, 2003.

Autori:

mr Aleksandar Samardžić, Matematički fakultet Univerziteta u Beogradu
mr Goran Nenadić, Matematički fakultet Univerziteta u Beogradu i
Department of Computation, UMIST, Manchester
dr Predrag Janičić, Matematički fakultet Univerziteta u Beogradu

Recenzenti:

doc. dr Nenad Mitić, Matematički fakultet Univerziteta u Beogradu
doc. dr Zoran Petrović, Matematički fakultet Univerziteta u Beogradu
mr Vladimir Filipović, Matematički fakultet Univerziteta u Beogradu

Ovaj materijal je objavljen uz dozvolu autora i namenjen je isključivo
za ličnu upotrebu. Distribucija u komercijalne svrhe nije dozvoljena.

Predgovor

Priprema za štampu tekstova koji sadrže „puno matematike“ i visoko su zahtevni sa stanovišta grafičkog izgleda dugo je podrazumevala ili puno tipografskog i tehničkog znanja ili loše rezultate. Pojavom sistema \TeX i \LaTeX omogućeno je da sami autori na prirodan i jednostavan način pripremaju takve tekstove. Zahvaljujući tome, tokom prethodnih godina, ovi sistemi su postali *de facto* standard za pripremu teksta u akademskoj matematičkoj i računarskoj, ali i u inženjerskoj zajednici. Većina knjiga i časopisa u ovim oblastima pripremljena je upotrebom ovih sistema, a oni za sve veći broj autora postaju nezaobilazan alat i u svakodnevnom radu.

Knjiga *$\LaTeX 2_{\epsilon}$ za autore* zamišljena je kao priručnik namenjen i početnicima i iskusnim korisnicima sistema \LaTeX . Naravno, od nje ne treba očekivati odgovore na sva pitanja u vezi sa \LaTeX -om, ali se nadamo da ona pokriva većinu najčešćih potreba. Za temeljnije upoznavanje sa sistemima \TeX i \LaTeX svakako preporučujemo, između ostalih, „klasične“ knjige: Knutov *\TeX Book* [19] i Lamportov *\LaTeX : A Document Preparation System* [21], kao i *The \LaTeX Companion* [12].

Iako inicijalno planirana kao nova verzija priručnika *Osnovi \LaTeX -a* (autora Predraga Janičića i Gorana Nenadića, izdanje VEDES, Beograd, 1995), knjiga *$\LaTeX 2_{\epsilon}$ za autore* gotovo da ne sadrži njegove tragove. Osim što ima drugačiju organizaciju i obim materijala, ova knjiga pokriva (i) karakteristike i mogućnosti nove verzije \LaTeX -a ($\LaTeX 2_{\epsilon}$).

Knjiga je organizovana u deset glava i tri dodatka. Uvodna glava (glava 1) razmatra osnovne elemente pripreme za štampu upotrebom sistema \LaTeX . Glave 2 i 3 opisuju kreiranje i zadavanje strukture \LaTeX -dokumenta, kao i dodatna formatiranja (npr. numeraciju strana, poravnavanja, prorde i sl.). Načini upotrebe različitih fontova obrađeni su u glavi 4, a pisanje matematičkih formula u glavi 5. Rad sa tabelama, slikama i grafikom opisan je u glavi 6, a međusobna referisanja, pisanja fusnota i kreiranje bibliografije i indeksa u glavi 7. Definiisanje novih komandi i načini za proširivanje \LaTeX -a navedeni su u glavi 8, a višejezičko prilagođavanje u glavi 9. Kreiranje slajdova i prezentacija je tema kojom se bavi glava 10. Dodatak A daje opis upotrebe sistema \LaTeX , dok dodatak B sadrži primere koji su korišćeni i na koje se poziva u knjizi. Konačno, u dodatku C navedene su liste komandnih sekvenci i specijalnih simbola.

Da bi se razlikovale od ostatka teksta, komande \LaTeX -a su ispisane slovima pisaae mašine, a grafički izgled primera izdvojen je simbolima [] i simbolima []

[]]. Primeri koji su navedeni na kraju knjige dostupni su u elektronskom obliku na Internet adresi www.matf.bg.ac.rs/latex2e. Sa iste strane mogu se autorima poslati komentari na knjigu, koji će uvek biti dobrodošli.

Zahvalni smo recenzentima, našim kolegama sa Matematičkog fakulteta u Beogradu, dr Nenadu Mitiću, dr Zoranu Petroviću i mr Vladimiru Filipoviću koji su pažljivim čitanjem i brojnim korisnim sugestijama doprineli kvalitetu knjige. Zahvaljujemo i svima ostalima koji su svojim pitanjima i predlozima uticali na knjigu.

Želimo da naglasimo da je pisanje ove knjige autorima predstavljalo vrlo prijatan zadatak. I naslov ove knjige predstavlja svesnu igru reči: osim autorima tekstova koji pišu u L^AT_EX-u, knjiga je namenjena i sopstvenim autorima koji su u nju — kao u podsetnik — zabeležili upravo ono što oni, kao korisnici L^AT_EX-a, najčešće koriste. Knjiga već ispunjava drugu namenu, a nadamo se da će njeni čitaoci smatrati da je korisna i za prvu.

Autori

Beograd, novembar 2003.

Predgovor elektronskom izdanju

Deset godina nakon objavljivanja štampanog izdanja ove knjige, za njom i dalje postoji interesovanje. Kako bi bila lakše dostupna što širem krugu korisnika, odlučili smo (uz saglasnost izdavača štampanog izdanja iz 2003. godine) da bude slobodno dostupna na Internetu.

Ovo, elektronsko izdanje razlikuje se neznatno od štampanog izdanja iz 2003.: glava 10 sada je uglavnom posvećena paketu `beamer` (koji je u međuvremenu postao dominantan paket za kreiranje prezentacija), umesto paketu `prospcr`.

Zahvaljujemo se svim čitaocima na komentarima i drago nam je ako je ova knjiga doprinela popularnosti L^AT_EX-a u našoj sredini. Ispravke i komentari na tekst mogu se uputiti elektronskom poštom na adresu latex2e@matf.bg.ac.rs. Ažurna verzija ove knjige, kao i arhiva sa primerima, su raspoložive na Internet adresi www.matf.bg.ac.rs/latex2e.

Autori

Beograd, novembar 2013.

Sadržaj

| | | |
|----------|--|-----------|
| 1 | Priprema za štampu i L^AT_EX | 1 |
| 1.1 | Računarska priprema teksta za štampu | 1 |
| 1.2 | T _E X i L ^A T _E X | 4 |
| 1.3 | Funkcionisanje T _E X-a | 7 |
| 1.4 | Označavanje teksta — komande u L ^A T _E X-u | 9 |
| 1.5 | Upotreba L ^A T _E X-a: jedan primer | 11 |
| 2 | Struktura L^AT_EX dokumenta | 15 |
| 2.1 | Preambula | 15 |
| 2.2 | Tekst dokumenta | 17 |
| 2.3 | Primer klase | 21 |
| 3 | Formatiranje teksta | 23 |
| 3.1 | Definisanje izgleda strane | 23 |
| 3.2 | Prelom redova, pasusa i strana | 27 |
| 3.3 | Proredi u tekstu | 29 |
| 3.4 | Boksovi | 31 |
| 3.5 | Poravnavanja teksta | 32 |
| 3.6 | Formatiranje listi | 33 |
| 3.7 | Doslovni, neformatirani ispis teksta | 34 |
| 4 | Fontovi | 36 |
| 4.1 | Tekući i osnovni font | 37 |
| 4.2 | Osnovni skup veličina slova | 37 |
| 4.3 | Osnovni skup vrsta fontova | 39 |
| 4.4 | Kodiranje | 44 |
| 4.5 | Promena osnovnog skupa vrsta fontova | 45 |
| 4.6 | Korišćenje dodatnih fontova | 46 |
| 4.7 | Specijalni simboli | 47 |
| 5 | Formatiranje formula i teorema | 50 |
| 5.1 | Formule | 50 |
| 5.2 | Matematički simboli | 51 |
| 5.3 | Složeni objekti | 53 |

| | | |
|-----------|---|------------|
| 5.4 | Fontovi u matematičkom režimu rada | 58 |
| 5.5 | Teoreme i slično | 59 |
| 6 | Formatiranje tabela i slika | 60 |
| 6.1 | Okruženje za tabele | 60 |
| 6.2 | Okruženje za crteže | 61 |
| 6.3 | Uključivanje slika u dokumente | 63 |
| 6.4 | Pozicioniranje pokretnih objekata | 65 |
| 7 | Referisanja u \LaTeX-u | 68 |
| 7.1 | Referisanja na delove teksta i objekte | 68 |
| 7.2 | Fusnote | 69 |
| 7.3 | Kreiranje indeksa | 70 |
| 7.4 | Kreiranje bibliografije | 71 |
| 8 | Proširivanje \LaTeX-a | 76 |
| 8.1 | Kreiranje novih komandi | 76 |
| 8.2 | Kreiranje novih okruženja | 79 |
| 8.3 | Kreiranje novih paketa | 79 |
| 9 | Višejezička podrška | 81 |
| 9.1 | Paket za višejezičku podršku | 81 |
| 9.2 | Pisanje na srpskom jeziku | 83 |
| 10 | Kreiranje slajdova i prezentacija | 87 |
| 10.1 | Osnovna podrška za kreiranje prezentacija | 87 |
| 10.2 | Napredna sredstva za kreiranje prezentacija | 87 |
| A | Korišćenje \LaTeX-a | 91 |
| A.1 | \LaTeX distribucije | 91 |
| A.2 | \LaTeX dokumentacija | 92 |
| A.3 | Obrada \LaTeX datoteka | 92 |
| A.4 | Obrada DVI datoteka | 93 |
| A.5 | Obrada POSTSCRIPT datoteka | 94 |
| B | Primeri | 95 |
| C | Tabele komandi | 145 |
| C.1 | Specijalni simboli i dijakritici | 145 |
| C.2 | Matematički simboli | 146 |

Glava 1

Priprema za štampu i L^AT_EX

U uvodnom poglavlju navodene su osnovne karakteristike pripreme za štampu upotrebom sistema L^AT_EX. Osim poređenja sa ostalim sistemima, naveden je i kratak istorijski osvrt na razvoj T_EX-a i L^AT_EX-a, kao i razlike između njih. Takođe, predstavljene su i osnove funkcionisanja L^AT_EX-a, načini zapisa komandi i obeležavanja teksta. Na kraju glave dat je i jedan jednostavan primer i ukratko opisan način upotrebe sistema L^AT_EX.

1.1 Računarska priprema teksta za štampu

Unos i priprema teksta za štampu danas se skoro u potpunosti odvija uz pomoć računara. U današnje vreme je, takođe, potpuno uobičajeno da autor sâm obavlja veliki deo, a neretko i kompletnu pripremu za štampu. Programi koji u tome pomažu, tzv. *tekst-procesori* (ili procesori teksta), oblikuju uneti tekst poravnavajući margine, smanjuju ili povećavaju razmake koji se pojavljuju između reči, štampaju neke delove teksta većim ili manjim slovima ili, pak, različitim pismima.

Slično tradicionalnoj pripremi za štampu, u i računarskoj pripremi za štampu srećemo dva pristupa. Prvi pristup podrazumeva da se autor prevashodno usredredi na sadržaj teksta, a da grafičko slaganje u potpunosti prepusti specijalizovanim programima koji igraju ulogu profesionalnih slovoslagача. Ovaj pristup nazivamo *procesivnim*. Drugi pristup (tzv. *interaktivno-grafički*) podrazumeva upotrebu računara kao savremene i sofisticirane pisaće mašine, uz pomoć koje autor, uglavnom samostalno, grafički oblikuje svoj tekst tokom unosa.

Kod interaktivno-grafičkih sistema, korisnik na ekranu računara dobija približni izgled svoga teksta još u trenutku dok ga unosi u, slično korišćenju klasične pisaće mašine. Ovako funkcionišu mnogi današnjih komercijalni tekst-procesori (npr. MICROSOFT WORD, MACWRITE, WORDPERFECT i sl.). U njih su integrisani *uređivači teksta* (editori, programi kojima se unosi i pamti tekst) i *formateri* (programi koji oblikuju tekst za štampu). Najveći deo intervencija nad izgledom teksta realizuje se interaktivno uz pomoć raznih ikona i komandi

koje se zadaju ili mišem ili pomoću tastature, a efekat komandi je odmah vidljiv. Izgled koji tekst ima na ekranu u principu odgovara onome što će biti dobijeno prilikom štampanja, zbog čega se ovi sistemi često nazivaju i WYSIWYG (od engleskog *what-you-see-is-what-you-get*, ono što vidiš, to i dobijaš). Međutim, često se za ovakve sisteme navodi i da ono što vidiš, to je sve što dobijaš (engleski *what-you-see-is-all-you've-got*), pri čemu se aludira na to da je, u principu, kod ove vrste tekst-procesora naglasak samo na izgledu tj. grafičkoj strukturi teksta. Međutim, čak i tada, najčešće je reč o lokalnom izgledu koji retko uzima u obzir celinu grafičkog izgleda dokumenta.

Za razliku od interaktivno-grafičkih sistema koji su se pojavili i stekli popularnost tek poslednjih decenija, procesivni sistemi su nešto stariji i ne pripadaju klasi WYSIWYG sistema. Među njima su najpoznatije razne verzije TROFF-a i T_EX-a.¹ Kod procesivnih sistema, faza unosa i opisa teksta striktno je odvojena od faze u kojoj se vrše grafičko slaganje teksta i vizuelizacija. Unos teksta se realizuje korišćenjem proizvoljnog editora (nezavisno od tekst-procesora), dok se priprema za štampu i vizulizacija realizuju posebnim programima koji uzimaju u obzir celokupnost izgleda svake strane pripremanog teksta. Na taj način se dosledno sledi praksa tradicionalne pripreme za štampu: autor teksta je odgovoran za pripremu i opis sadržaja, dok se realizacija grafičkog oblikovanja prepušta proverenim majstorima. Dok se u slučaju interaktivno-grafičkih sistema korisnik pretežno koncentriše na izgled teksta, kod procesivnih sistema naglasak je na logičkoj strukturi, budući da u toku unosa autor opisuje svoj tekst, i ne brine kako on trenutno izgleda na ekranu.

U oba slučaja, tekst-procesori kao model zapisa interno koriste tzv. obeležene tekstove. U obeleženom tekstu se, osim zapisa samih karaktera koji čine tekst, nalaze i dodatne informacije (tzv. *obeležja*) kojima se naznačavaju određene karakteristike delova teksta. Kod interaktivno-grafičkih sistema obeležja se prevashodno vezuju za grafičke karakteristike (npr. koji deo teksta treba predstaviti krupnijim slovima, koji deo ćirilicom a koji nekim drugim pismom, koji deo treba da bude centriran i slično). Ta obeležja nisu vidljiva ni dostupna korisniku, i obično nastaju kao posledica određene akcije nad tekstom. Kod procesivnih sistema, obeležja se unose eksplicitno u tekst, i najčešće označavaju logičke celine u tekstu (npr. naslov, iskaz teoreme, oznaka formule i slično), a njihovo prikazivanje se definiše jedinstveno za sve instance koje odgovaraju tom logičkom obeležju. Navedimo jedan primer. Ako se pri opisu nekog programskog paketa njegove *komande* žele ispisati kurzivnih pismom, u interaktivnom sistemu će se primenjivati akcija za ispis kurzivom nad svakom pojavom *komande* u tekstu (što će biti automatski zabeleženo odgovarajućim obeležjem). S druge strane, u slučaju procesivnog sistema, korisnik će svaku pojavu *komande* eksplicitno obeležiti kao takvu, a onda će definisati da se svaka *komanda* štampa kurzivnim pismom. Ova razlika je naročito bitna u slučaju da se odluka vezana za ispis *komandi* kasnije promeni, tako da se one ne ispusuju kurzivnim stilom nego, na primer, bez-serifnim slovima. U prvom slučaju, korisnik će morati da

¹TROFF je deo standardne kolekcije UNIX programa koji se koriste za obradu teksta. Osim njega koriste se i NROFF i ROFF. O T_EX-u i njegovim verzijama biće više reči kasnije u ovoj knjizi.

potraži svaku pojavu *komande* i promeni joj grafička svojstva, dok će u drugom slučaju biti dovoljno promeniti definiciju koja određuje način štampanja *komandi*.

Izbor između procesivnog i interaktivno-grafičkog pristupa nije samo pitanje ličnog opredeljenja. On svakako zavisi od vrste dokumenata koji se tipično pripremaju za štampu, njihove dužine i složenosti. Ispis kratkih poruka jednostavnog teksta često zahteva manje vremena i truda (uz sličan grafički rezultat) ukoliko se koriste interaktivni sistemi. Slično, priprema multimedijalnih dokumenata se često oslanja na interaktivne sisteme. S druge strane, svako ko je pripremao matematički ili tehnički tekst u nekom interaktivnom sistemu, zna koliko je vremena potrebno da bi svaka formula izgledala baš onako kako je autor zamislio, zbog čega je utisak da procesivni sistemi imaju veću popularnost u naučnim i tehničkim krugovima. Dok kod interaktivnih sistema korisnik manje-više sâm slaže formule, kod procesivnih sistema se izgled formule (i ostalog teksta) generiše automatski iz njenog opisa. Slično, procesivni sistemi će se verovatno pokazati kao bolji izbor pri pripremi vrlo dugačkih i složenih tekstova, za mnoštvom referenci i međusobnih ukazivanja, čak i ukoliko nema komplikovanih formula. Na primer, ukoliko se u interaktivnom sistemu u dugačkom dokumentu izvrši neka promena na samom početku dokumenta, može se desiti da lančana promena značajno poremeti izgled sledećih strana (budući da ovakvi sistemi najčešće vode računa samo o lokalnom grafičkom izgledu), pa će korisnik pre štampanja morati da proveri i možda doradi izgled svake strane. Konačno, rad sa dokumentima koji će biti doradivani ili menjani na više (različitih) sistema je lakši upotrebom procesivnih sistema.²

Ipak, treba naglasiti da su danas razlike između ova dva pristupa sve manje. U interaktivno-grafičkim sistemima postoje tzv. *stilovi* kojima se može definisati niz grafičkih karakteristika za određene klase objekata, i koji se, potom, mogu primeniti na odgovarajuće delove teksta. Takođe, moguće je da se, do nekog nivoa, označi i logička struktura dokumenta, izgeneriše sadržaj, unesu referisanja i slično. S druge strane, procesivni sistemi su sve češće pojačani programskim okruženjima koja omogućavaju da se dokument, u obliku pripremljenom za štampu, pojavljuje paralelno sa unosom teksta (npr. u zasebnom prozoru). Ako je nekada brzina računara bila problem da se to obavi u realnom vremenu, onda taj problem sve više iščezava, čime se delimično razrešava problem izvesne nečitljivosti unetog teksta pre njegove konačne vizuelizacije.

Sumirajući razlike između navedena dva pristupa, možemo reći da procesivni sistemi podržavaju sistematsku i logičku pripremu za štampu, i da ih je pogodnije koristiti u slučaju složenijih, dužih i grafički zahtevnijih tekstova, kao i tekstova koji će možda kasnije biti ponovo korišćeni ili menjani. Oni omogućavaju da se problemi vizuelizacije potisnu u drugi plan tokom pisanja i da se autor, dok unosi tekst, u potpunosti usredsredi na sadržaj onoga što piše. Dodatno, iako je proces vizuelizacije odvojen, procesivni sistemi omogućavaju potpunu i neposrednu kontrolu i nad procesom pripreme za štampu (npr. ekspli-

²Konkretan primer za ovu tvrdnju je i ova knjiga (pripremljena jednim procesivnim sistemom), pri čijoj je izradi jedan od autora koristio operativni sistem WINDOWS, drugi operativni sistem UNIX, a treći je, prema raspoloživom okruženju, koristio oba.

citnim definisanjem parametara strane, razmaka redova, slova i sl.), sve do nivoa koji štampani dokument može učiniti ne samo grafički neskladnim, nego i potpuno nečitljivim (npr. ukoliko su razmak između redova definiše tako da redovi preklapaju). S druge strane, interaktivno-grafički sistemi često onemogućavaju korisniku da proizvede nečitljiv dokument ograničavanjem skupa komandi raspoloživih za grafičko oblikovanje (što, jasno, može da ograniči i kontrolu nad procesom pripreme za štampu).

Udobnost u radu je ranije često bila isticana kao značajna prednost interaktivno-grafičkih sistema, ali su pojave novih procesivnih radnih okruženja (videti dodatak A) svakako doprinele da se ta prednost smanji, ako ne i potpuno izgubi. Možda je tačno da procesivni sistemi zahtevaju malo više vremena i strpljenja pri ovladavanju komandama na samom početku, ali se taj „zaostatak“ vrlo brzi nadoknadi kasnijim sistematičnim i prilagodljivim pristupom pripremi teksta za štampanje.

Na kraju, navedimo jos dve značajne razlike. Tekstovi pripremljeni za neki od procesivnih sistema su, u principu, nezavisni od platforme na kojoj se mogu vizuelizovati i/ili ponovo koristiti. Oni se najčešće oslanjaju na međunarodne, javno dostupne i prihvaćene standarde, a ne na „standarde“ koje su razvile i koriste određene kompanije, što je slučaj sa većinom interaktivnih sistema. Dodatno, procesivni sistemi su po pravilu javno dobro i besplatno su dostupni, dok je većina interaktivnih sistema komercijalna.

1.2 T_EX i L^AT_EX

T_EX je sistem za pripremu teksta za štampu koji je kasnih sedamdesetih godina prošlog veka osmislio Donald Knut.³ Budući da je i sâm imao puno problema pri pisanju knjiga korišćenjem tada postojećih sistema, njegov cilj je bio da napravi „sistem za obradu teksta namenjen pisanju lepih knjiga — pogotovo knjiga koje sadrže puno matematike“. Njegova ideja je bila da tradicionalni proces pripreme za štampu digitalizuje, i da napravi sistem koji će omogućiti visok kvalitet s jedne strane, a jednostavnost i prirodnost sa druge. Knut je svoj sistem opisao u knjizi *The T_EXbook* [19], koja predstavlja jednu od najobimnijih i najpreciznijih dokumentacija koje su napisane za programske pakete ove veličine.

T_EX pripada klasi procesivnih sistema: za neposredan unos i opis teksta koristi se bilo koji editor, dok se za pretvaranje teksta u grafički odraz koristi odgovarajući program, koji igra ulogu vrhunskog elektronskog slovoslagača.

³Donald Knut je, bez sumnje, jedno od najznačajnijih imena računarstva dvadesetog veka. Dobitnik je prestižne Tjuringove nagrade za doprinos razvoju računarstva (1974. godine), pre svega za rezultate u oblasti analize algoritama i dizajna programskih jezika. Životnim delima mu se smatraju sistem T_EX i višetomna, još nezavršena knjiga *Umetnost programiranja* (Donald Knuth, *The Art of Computer Programming*, Reading, Massachusetts: Addison-Wesley, Vol. 1–3), koju priprema od 1962, i na kojoj i sada, kao penzionisani profesor Stanford univerziteta (Kalifornija), radi (četvrti i peti deo se očekuju do 2007. odnosno 2010. godine). Više informacija o Knutovim aktivnostima dostupno je na Internet strani: www-cs-staff.stanford.edu/~knuth.

Konačnom skladnom grafičkom izgledu je posvećena naročita pažnja, budući je Knut u \TeX ugradio vrhunske tradicionalne tipografske tehnike i podršku najvišim tipografskim zahtevima.

\TeX danas predstavlja nezamenjiv alat pri pisanju matematičke, računarske i tehničke literature. Iako se možda i može raspravljati o (ne)komfortu koji pruža (ili zahteva) \TeX , izvesno je da je on dobio bitku protiv drugih sistema za obradu teksta u naučnim krugovima, pa sada i njegova rasprostranjenost postaje faktor od izuzetne važnosti za svakoga ko namerava da koristi \TeX . Na primer, mnogi časopisi (ne samo matematički, računarski i tehnički, već i oni koji se bave psihologijom, lingvistikom i sl.) zahtevaju (ili bar strogo preporučuju) pripremu teksta upotrebom \TeX -a, smatrajući da je \TeX *lingua franca* za pripremu naučnih radova za štampu. \TeX se ne koristi samo zbog svoje lakoće i izražajnosti pri opisu tekstova, već i zbog prenosivosti: tekstovi pripremljeni u \TeX -u su, u principu, nezavisni od platforme na kojoj se procesiraju, vizuelizuju i ponovo koriste.

\TeX podržava sistematičan pristup pripremi za štampu, pogotovo dužih i grafički komplikovanih tekstova. Pripremanjem teksta pomoću \TeX -a, precizno se opisuju objekti koji treba da budu formatirani, pri čemu su ti opisi najčešće sasvim prirodni i jednostavni. Takvi opisi olakšavaju pisanje pre svega matematičkih ili tehničkih tekstova, jer korisnik samo opisuje šta želi (i treba) da dobije, a sistemu ostavlja da to i uradi. Svako ko je pisao matematičke tekstove korišćenjem nekog drugog sistema za pripremu teksta iskusio je mnoštvo problema (npr. skladan izgled teksta i zapisa teorema i definicija, zapis formula — pogotovo razlomaka, integrala, izraza sa indeksima, konjugovanih vrednosti, matrica i sl.). U slučaju \TeX -a, sistem preuzima brigu o mnogim takvim stvarima za koje nije neophodan eksplicitan opis, pa tako, na primer, u matematičkim izrazima automatski podešava veličine određenih simbola (npr. simbola za uniju, integral i sl.) ili razmak između njih, što rezultuje skladnim izgledom odštampanog teksta, a korisniku znatno olakšava rad. Osim za matematičke formule, \TeX nudi i široku podršku i za referisanja u tekstu, generisanje indeksa i bibliografija, uključivanje grafike i slično.

Kao i kod svakog procesivnog sistema, izgled teksta se ne dobija za vreme njegovog unosa, već se tekst tokom unosa opisuje, a zatim se po potrebi grafički vizuelizuje. Upotreba \TeX -a se zasnova na korišćenju specifičnog jezika (odnosno skupa komandi) za opis teksta, pomoću kojih se opisuje ono što se želi dobiti odštampano. Opisani tekst se dalje obrađuje specifičnim programom (tzv. \TeX procesorom) koji igra ulogu profesionalnog slovoslagača, i koji, u skladu sa opisom, formira grafičku „sliku“ teksta. Ta slika se dalje može vizuelizovati na proizvoljnom medijumu (ekranu, štampaču i sl.), slično kao što se i tradicionalne štamparske ploče mogu preslikati na različite medijume.

Logika \TeX -a je u osnovi bliska logici programiranja: \TeX je, u stvari, programski jezik koji služi za strukturiranje i grafičko opisivanje teksta. On dopušta proširivanje i prilagođavanje pojedinačnim specifičnim potrebama, što, pre svega, podrazumeva mogućnost da se — korišćenjem osnovnog skupa komandi \TeX -a — formira skup sopstvenih definicija, odnosno *makroa*, koji će omogućavati efikasno i udobno korišćenje \TeX -a, kao i formiranje odgovarajućih

biblioteka takvih makroa koji se potom mogu ponovo koristiti. Za poslednjih dvadesetak godina napravljeno je mnoštvo korisnih javno dostupnih biblioteka makroa. Na primer, Društvo matematičara Amerike je kreiralo skup makroa koji je nazvan $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$, a slično su i mnoga druga strukovna udruženja definisala svoje makroe (npr. $\text{Ling}\TeX$, $\text{Astro}\TeX$, $\text{PHYST}\TeX$).

Jednu od opštih biblioteka kreirao je Lesli Lamport⁴ i nazvao je \LaTeX . \LaTeX je, dakle, sistem za pripremu za štampu nastao na osnovama \TeX -a. On se oslanja na \TeX , koji u osnovi i realizuje pripremu za štampu, tj. slaganje teksta, dok \LaTeX obezbeđuje skup specifičnih klasa dokumenata koje omogućavaju znatno udobniji rad. Budući da je reč o skupu makroa, \LaTeX opisi se automatski prevode u odgovarajuće \TeX komande upotrebom odgovarajućeg \LaTeX procesora. \LaTeX je detaljno opisan u [12].

Osnovna razlika između \TeX -a i \LaTeX -a je u sledećem: upotrebom osnovnog, „čistog“ \TeX -a (tzv. *plain* \TeX), autor ima mogućnost (ali i obavezu) da potpuno opiše izgleda svog teksta, uključujući, na primer, i definisanje širine i dužine teksta na strani, tipove slova i njihove veličine, razmake koji se ostavljaju iza i ispred naslova, kao i mnoge druge tehničke detalje. Iako omogućava potpunu kontrolu, to često može da bude zamorno i da rezultira neskladnim izgledom teksta. Makroi \LaTeX -a omogućavaju da se mnogi standardni zadaci (npr. definisanje dužine i širine teksta na strani A4 formata) obave automatski izborom i navođenjem odgovarajućeg „stila“ (npr. `a4paper`). Na taj način se \LaTeX -u prepušta slaganje dokumenta u skladu sa predefinisanim tipografskim standardima za datu klasu. Naravno, autor u svakom trenutku može da predefiniše neka ili sva podrazumevana formatiranja koje za njega obavlja \LaTeX . Međutim, treba naglasiti da su stilovi i makroi ugrađeni u \LaTeX nastali kao rezultat vrlo ozbiljnog i stručnog rada desetina stručnjaka iz oblasti tipografije, i da nije realno očekivati da će prosečan korisnik \LaTeX -a biti često u prilici da kreativno poboljša grafički izgled svog dokumenta menjanjem postojećih standarda. Štaviše, ovakav pristup pomaže korisniku da izbegne kreiranje grafički potpuno neprihvatljivih tekstova, na primer, biranjem neskladne veličine i tipa slova za naslov svog dokumenta ili za ispis naslova poglavlja. Opšta je preporuka da autor treba da se, koliko god je to moguće, oslanja na ugrađene standarde i da se uzdrži od menjanja osnovnih makroa i definicija. To, naravno, nikako ne znači da autori ne treba da definišu svoje makroe i komande koji pokrivaju njihove specifične potrebe.

Prva verzija \LaTeX -a (poznata kao \LaTeX 2.09) u upotrebi je od 1985, a tekuća verzija (poznata pod imenom \LaTeX 2_ε) od 1994. Ova knjiga predstavlja vodič za efikasno i pravilno pripremanje tekstova za štampu korišćenjem sistema \LaTeX 2_ε (mada pokriva i rad sa verzijom \LaTeX 2.09). Svakako, budući da je \LaTeX 2_ε vrlo kompleksan sistem i da pruža mnoštvo mogućnosti, za detaljnije upoznavanje sa ovim sistemom preporučljivo je pročitati neku od „klasičnih“

⁴Osim po kreiranju \LaTeX -a, matematičar Lesli Lamport poznat je i po značajnim rezultatima u oblasti paralelnog procesiranja i distribuiranih sistema. Radio je kao istraživač u velikim računarskim kompanijama (DEC/Compaq, SRI International), a trenutno je angažovan u kompaniji Microsoft. Više informacija o Lamportu dostupno je na Internet strani: <http://research.microsoft.com/users/lamport/>.

knjiga o $\text{T}_{\text{E}}\text{X}$ -u i/ili $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -u (npr. [19], [12]), ili neku od knjiga o $\text{T}_{\text{E}}\text{X}$ -u odnosno $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -u koje su izašle na našem jeziku, od kojih su mnoge navedene u bibliografiji.

Budući da korisnik o tome ne mora da vodi računa, u tekstu najčešće nije eksplicitno naglašavano šta je izvorna $\text{T}_{\text{E}}\text{X}$ komanda, a šta $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ makro (koje ćemo takođe zvati i $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ komandama), kao ni koje komande pripadaju kojoj verziji $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -a. Korisnik može, u istom dokumentu, koristiti i kombinovati (gotovo sve) komande iz svakog od ovih paketa. Ipak, ova knjiga prati i preporučuje pristup karakterističan za najnoviju verziju $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -a. U tom smislu, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ će se uvek odnositi na verziju $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$, dok će se — u slučaju potrebe — stvari karakteristične za $\text{T}_{\text{E}}\text{X}$ ili $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2.09$ eksplicitno naglašavati.

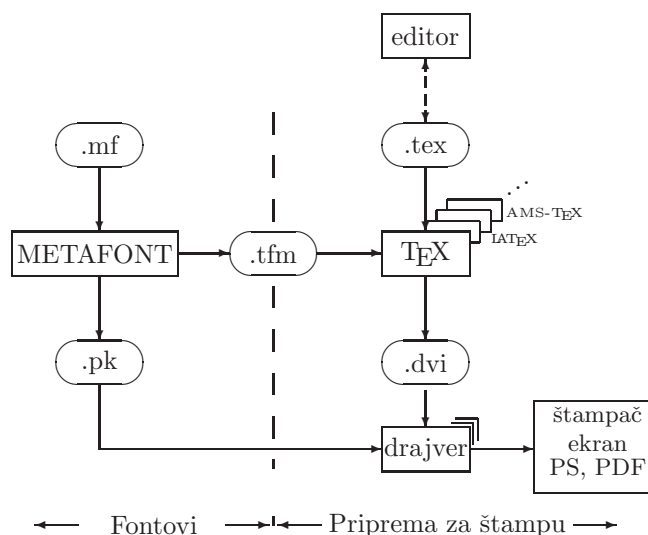
Na kraju ovog poglavlja spomenimo i nešto o samim imenima $\text{T}_{\text{E}}\text{X}$ i $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Lamport navodi da je — kad je reč o upotrebi ovih sistema — izgovor imena jedna od najtežih stvari, i to ne samo za početnike. Iz tog razloga gotovo sve knjige o $\text{T}_{\text{E}}\text{X}$ -u i $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -u počinju davanjem rimujućih primera koji bi korisnike naveli na pravilan (ili željeni) izgovor. Naime, ime $\text{T}_{\text{E}}\text{X}$ izvedeno je od grčke reči $\tau\epsilon\chi\nu\iota\sigma$, koja označava umetnost, umeće, veštinu, i izgovara se „teh“ (ili ređe „tek“), a ne „teks“ ili bilo kako drugačije. Otuda, čitanjem $\text{T}_{\text{E}}\text{X}$ -a kao da je ispisan ćirilicom dobiće se pravilan izgovor. U svakom slučaju, da bi se naznačila razlika između reči $\text{T}_{\text{E}}\text{X}$ i reči TeX i TEX (koje se obično izgovaraju kao „teks“ i označavaju druge, najčešće komercijalne proizvode), preporučuje se da se za zapis imena Knutovog sistema koristi odgovarajući logo ($\text{T}_{\text{E}}\text{X}$), a da se — kada to nije moguće — ime zapisuje kao TeX ili sa slovom E spuštenim u odnosu na druga dva slova (npr. $T_{\text{E}}X$).

Slično, za referisanje na $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ treba koristiti odgovarajući logo, ili ga zapisivati kao LaTeX ili $L^{\text{A}}T_{\text{E}}X$. Što se izgovora tiče, u našoj sredini uobičajen izgovor je „lateh“, mada se (ponegde u engleskom govornom području) koristiti i „lej-teh“, sa naglaskom bilo na prvom bilo na drugom slogu.

1.3 Funkcionisanje $\text{T}_{\text{E}}\text{X}$ -a

Slaganje teksta u $\text{T}_{\text{E}}\text{X}$ -u odvija se pomoću dve osnovne komponente koje obuhvataju pripremu za štampu i podršku za fontove. Slika 1.1 ilustruje strukturu i komponente $\text{T}_{\text{E}}\text{X}$ sistema.

Kao što je navedeno ranije, tekst se opisuje korišćenjem proizvoljnog editora i, uobičajeno, čuva u datoteci sa ekstenzijom tex . Od opisa teksta i pridruženih biblioteka makroa (npr. biblioteke $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -makroa), $\text{T}_{\text{E}}\text{X}$ generiše tzv. DVI (eng. *De Vice Independent*) grafičku sliku teksta, koja se čuva u datoteci sa ekstenzijom dvi . Slično klasičnoj pripremi za štampu u kojoj se koriste olovne pločice, grafičku sliku (sadržanu u DVI datoteci) čine nizovi dvodimenzionalnih pravougaonika koji odgovaraju objektima koji čine tekst (npr. slova, cifre, linije i slično). DVI datoteke sadrže kompletan opis izgleda teksta (niz pravougaonika sa dimenzijama i sadržajem), koji je nezavisan od medijuma na kome će dokument biti vizuelizovan. Pri generisanju ovih pravougaonika, $\text{T}_{\text{E}}\text{X}$ -u su neophodne dimenzije i oblik objekta koji će biti smešteni u svaki od pravougao-

Slika 1.1: Struktura i osnovne komponente $\text{T}_{\text{E}}\text{X}$ -a

nika: na primer, pravougaonik koji odgovara slovu i je, u principu, uži i duži od pravougaonika koji se koristi za slovo u . Osim toga, shodno najfinijoj tradiciji slovoslagačkog zanata, pojedine kombinacije slova (npr. fi , ff , fl) mogu biti predstavljene jednim pravougaonikom (takve kombinacije zovu se ligaturama, videti poglavlje 4.7). Veličine pravougaonika svakako zavise od tipa fonta (tj. grafičkog izgleda slova) koji se koristi. Iz tog razloga, pri generisanju DVI datoteke, $\text{T}_{\text{E}}\text{X}$ koristi tzv. metričke opise fontova koji su upotrebljeni u dokumentu. Ti opisi se nalaze u datotekama sa ekstenzijom `tfm` (eng. *TeX font metric*).

Prosečan korisnik $\text{T}_{\text{E}}\text{X}$ -a obično koristi neke od velikog broja postojećih fontova koji su javno dostupni (npr. Computer Modern Roman ili Adobe Helvetica), i prisutni na svim $\text{T}_{\text{E}}\text{X}$ sistemima. Dodatni fontovi se mogu dizajnirati korišćenjem paketa METAFONT, koji je takođe osmislio Donald Knut, a koji je detaljno opisan u njegovoj knjizi *The METAFONTbook* [20]. Iako se METAFONT tipično automatski instalira uz sâm $\text{T}_{\text{E}}\text{X}$, njegova upotreba, kao i kreiranje novih fontova daleko prevazilazi okvire ove knjige. Za upotrebu nekih od postojećih nestandardnih fontova, dovoljno je pribaviti odgovarajuću MF (METAFONT) datoteku, odakle se automatski po potrebi generišu TFM i PK datoteke. Dok se TFM datoteke koriste pri kreiranju DVI datoteka, drajveri koriste PK (eng. *packed font*) datoteke da bi prikazali sadržaj teksta na konkretnom tipu ekrana ili štampača.

DVI datoteka se može vizuelizovati na prozvoljnom uređaju za koji je dostupan neki od programa (drajvera) za vizuelizaciju DVI datoteka, kao i skup odgovarajućih PK fontova. Uz pomoć odgovarajućeg drajvera, DVI datoteka se može prikazati na ekranu, prevesti u neki prenosivi format za štampanje (npr. PDF ili PS) ili poslati direktno na štampač (videti detaljnije u dodatku A).

1.4 Označavanje teksta — komande u \LaTeX -u

Priprema teksta u \LaTeX -u podrazumeva da se određeni delovi teksta (npr. naslovi, definicije, formule i slično) označe (obeležje) da bi bili pravilno pripremljeni za štampu (tj. složeni). Za označavanje se koriste \LaTeX komandne sekvence ili, kraće, \LaTeX komande (koje su, zapravo, \TeX komande i makroi). Same komande se, naravno, ne prikazuju u konačnoj (vizuelizovanoj) verziji dokumenta, već se koriste da ukažu \LaTeX procesoru na delove teksta koji treba da imaju određeni grafički izgled.

Komande počinju znakom `\` za kojim sledi niz slova koji \LaTeX treba da prepozna kao ime postojeće komandne sekvence. Na primer, komande \LaTeX -a su `\section`, `\it`, `\LaTeX`, itd. Pri identifikovanju komandi u tekstu, \LaTeX podrazumeva da je ime komande maksimalan niz slova koja se nalaze iza znaka `\` (sve do pojave prvog neslovnog karaktera). Na primer, u zapisu `a \in S`, \LaTeX bi prepoznao komandu `in` (koja služi za ispis matematičkog simbola \in), dok bi u zapisu `a \inS` pokušao da prepozna komandu `\inS`, pa, ako takva komanda ne postoji, došlo bi do greške. Za razdvajanje komande od ostatka teksta najčešće se koriste blanko simbol (tj. razmak) ili vitičaste prazne zagrade (`{}`), koje označavaju tzv. prazan simbol. Prethodna komanda bi se, tako, mogla ispravno zapisati kao `a \in S` ili kao `a \in{}S`. Ukoliko u tekstu posle komande prirodno sledi neslovni karakter koji ne zahteva razmak, onda se on može koristiti i za označavanje kraja komande. Na primer, \LaTeX će ispravno prepoznati komandu `\sin` u zapisu `\sin(x)`, kao i komandu `\LaTeX` u zapisu `\LaTeX-a`.

Napomenimo da upotreba blanko simbola ili bilo kog drugog simbola za razdvajanje komande od ostatka teksta ne znači da će na tom mestu u štampanom tekstu biti napravljen odgovarajući razmak. Blanko simbol posle komande samo označava njen kraj, a dodatni razmak se neće praviti osim ukoliko to nije predviđeno tekstom koji sledi ili tipom same komande (npr. neke matematičke komande obavezno generišu razmak). Tako bi isti efekat (ispis teksta \LaTeX a) imale komande `\LaTeX a` i `\LaTeX{}a`. Konačno, napomenimo da pri upotrebi komandi treba voditi računa da li se one pišu velikim i/ili malim slovima. Na primer, komande `\LaTeX`, `\Latex` i `\latex` potencijalno mogu imati različita dejstva.

Komande mogu označavati zapis pojedinih specijalnih simbola (npr. \LaTeX logo), imati dejstvo na određeni deo teksta (npr. ispis kurzivnim pismom), davati određene instrukcije \LaTeX procesoru (npr. prelaz na novu stranu) i sl. Komande mogu imati i svoje argumente. Na primer, u slučaju kad deluju na određeni deo teksta, taj tekst se obično navodi kao argument. Na primer,

komanda `\emph` ima kao argument deo teksta koji će biti ispisan naglašeno (npr. `\emph{naglašen tekst}`), dok se naslov glave prosleđuje kao argument komandi `\section` (npr. `\section{Naslov}`). Uobičajeno je da se obavezni argumenti komandi pišu unutar vitičastih `{ }` zagrada, dok se opcioni argumenti navode u uglastim `[]` zagradama. Neke jednostavne \LaTeX komande prikazane su u tabeli 1.1.

| komanda | namena | primer upotrebe | izgled |
|--------------------------|--------------------|----------------------------------|----------------------|
| <code>\TeX</code> | TeX logo | <code>\TeX</code> | TeX |
| <code>\LaTeX</code> | \LaTeX logo | <code>\LaTeX</code> | \LaTeX |
| <code>\emph</code> | naglašen tekst | <code>\emph{naglašeno}</code> | <i>naglašeno</i> |
| <code>\textbf</code> | podebljan tekst | <code>\textbf{podebljano}</code> | podebljano |
| <code>\textit</code> | kurzivni tekst | <code>\textit{kurziv}</code> | <i>kurziv</i> |
| <code>\large</code> | krupnija slova | <code>{\large krupnije}</code> | krupnije |
| <code>\small</code> | sitnija slova | <code>{\small sitnije}</code> | sitnije |
| <code>\underline</code> | podvlačenje teksta | <code>\underline{podvuci}</code> | podvuci |
| <code>\title</code> | naslov dokumenta | <code>\title{Naslov}</code> | Naslov |
| <code>\section</code> | ново poglavlje | <code>\section{Naslov}</code> | 1. Naslov |
| <code>\subsection</code> | ново potpoglavlje | <code>\subsection{Naslov}</code> | 1.1 Naslov |
| <code>\footnote</code> | fusnota | <code>\footnote{fusnota}</code> | ¹ fusnota |
| <code>_</code> | indeks dole | <code>\$x_{i+1}\$</code> | x_{i+1} |
| <code>^</code> | indeks gore | <code>\$x^{j+1}\$</code> | x^{j+1} |
| <code>\in</code> | relacija pripada | <code>\$x \in S\$</code> | $x \in S$ |
| <code>\forall</code> | relacija pripada | <code>forall{x}</code> | $\forall x$ |
| <code>\newpage</code> | nova strana | <code>\newpage</code> | |

Tabela 1.1: Primer nekih osnovnih komandi \LaTeX -a

| okruženje | namena |
|-------------------------|------------------------|
| <code>center</code> | centriranje teksta |
| <code>flushright</code> | poravnavanje u desno |
| <code>flushleft</code> | poravnavanje u levo |
| <code>quote</code> | za izdvajanje teksta |
| <code>tabular</code> | za kreiranje tabela |
| <code>itemize</code> | za nenumerisane liste |
| <code>enumerate</code> | za numerisane liste |
| <code>verbatim</code> | za neformatirani ispis |

Tabela 1.2: Primer nekih osnovnih okruženja u \LaTeX -u

Pored komandi, instrukcije se \LaTeX -u mogu zadavati i preko tzv. *okruženja*. Okruženja menjaju način na koji \LaTeX formatira deo teksta koji je naveden u okviru njih. Početak okruženja označava se komandnom sekvencom oblika `\begin{okruženje}`, a kraj komandom `\end{okruženje}`. Na pri-

mer, okruženje `center` može da se koristi za ispisivanje centriranog teksta, a `thebibliography` za ispis formatiranog spiska bibliografije. Okruženja takođe mogu imati argumente koji se navode u nastavku odgovarajuće `\begin` naredbe. Neka okruženja prikazana su u tabeli 1.2.

U glavama koje slede opisane su mnoge komande i okruženja koje obezbeđuje sâm \LaTeX , kao i način za definisanje novih komandi i okruženja. Nove komande su najčešće vezane za opis određenih logičkih objekata u tekstu (npr. iskazi teorema, definicije, vektori i sl.).

Pored komandi koje treba da proizvedu neko dejstvo u izlaznom tekstu, \LaTeX podržava i komandu (tzv. *komentar*) koja omogućava da se delovi teksta označe tako da budu ignorisani tokom obrade \LaTeX procesorom. Komentari se označavaju znakom za procenat (%): deo teksta zapisan od znaka % do kraja reda smatra se komentarom, i ne uzima se u obzir pri slaganju teksta, ali se može koristiti da bi se zabeležilo dodatno objašnjenje o samom tekstu (npr. podaci o autoru, datum izmene dokumenta, i sl.) ili formatiranju (npr. objašnjenje neke novouvedene komande, ili naznaka početka glave, poglavlja, bibliografije i sl.). Pored toga, komentari su vrlo korisni za označavanje delova koji se privremeno neće prikazivati na izlazu (npr. nedovršeni delovi teksta) ili za delove koji nisu ispravno obeleženi \LaTeX komandama (da bi se sprečilo da \LaTeX procesor javlja greške). Pisanje komentara može često biti od velike pomoći pri nastavku rada na dokumentu posle duže pauze.

Osim `\`, `{`, `}` i `%`, još nekoliko simbola (`&`, `$`, `#`, `_`, `~`, `^`) ima specijalno dejstvo u \LaTeX -u. Zapis takvih simbola se dobija pomoću jednostavnih komandi koje su nabrojane u tabeli 4.9 (strana 48).

1.5 Upotreba \LaTeX -a: jedan primer

Na kraju uvodnog poglavlja dajemo primer jednostavne ulazne datoteke (slika 1.2, strana 13) i ilustrujemo osnovne principe korišćenja \LaTeX -a. Ulazna datoteka se može kreirati proizvoljnim editorom, ponajbolje onim koji direktno podržava upotrebu naših specifičnih slova (kao što su slova sa dijakriticima). Danas gotovo svi operativni sistemi poseduju takve editore, a podrška za naša slova se može dobiti izborom odgovarajućeg kodiranja.⁵

Dokument kreiran u \LaTeX -u ima dva dela: u prvom se navode njegove opšte karakteristike (npr. tip dokumenta, pomoćni paketi koji se koriste, razne definicije i slično), dok se u drugom nalazi opis samog teksta. Prvi deo (tzv. *preambula*) počinje komandom `\documentclass{klasa}` koja zadaje jednu od predefinisanih klasa u skladu sa kojom se dokument kreira. Na primer, standardne klase su `article` (za članke i radove), ili `book` (za knjige). Sâm

⁵Odgovarajuća standardna kodiranja za srpski jezik su *ISO 8859-2* za latinični tekst (poznato i kao *Latin2*), odnosno *ISO 8859-5* za ćirilične karaktere, i njih treba izabrati u opcijama korišćenog editora. Ova kodiranja su često i podrazumevano dostupna pod UNIX sistemima. Operativni system WINDOWS ima sopstvena kodiranja koja se aktiviraju „promenom“ tastature na *Serbian (Latin)* ili *Serbian (Cyrillic)*. Za detalje videti glavu 9. Ukoliko naša specifična slova ipak nisu raspoloživa, mogu se koristiti odgovarajuće \LaTeX -komande (videti tabelu 9.2, strana 83).

tekst dokumenta počinje komandom `\begin{document}` i završava se komandom `\end{document}`. Dakle, \LaTeX datoteke imaju sledeći oblik:

```

\documentclass{klasa}

...

\begin{document}

...

\end{document}

```

U primeru koji je naveden na strani 13, tip (odnosno klasa) dokumenta je `article`. U preambuli je navedeno da se koristi pomoćni paket `inputenc` (sa opcijom za kodiranje `latin2`). O ovom paketu biće više reči u glavi 9.

Tekst dokumenta iz primera na slici 1.2 sadrži dva poglavlja obeležena komandama `\section`, koje sadrže naslove odgovarajućih celina. Naglasimo da korisnik u ulaznoj datoteci sâm prelama redove (ili to za njega čini editor), dok će \LaTeX stvarni izgled redova odrediti prilikom procesiranja (nezavisno od toga kako su redovi prelomljeni u ulaznoj datoteci u okviru jednog pasusa). Novi pasus u dokumentu označava se ostavljanjem jednog praznog reda u ulaznoj datoteci, dok se forsirani prelazak u novi red (bez počinjanja novog pasusa) dobija komandom `\\`.

Pored komandi koje ispisuju odgovarajuće logoe⁶, u prvom pasusu se koristi i komanda `\emph` koja označava deo teksta koji će biti ispisan *naglašeno* u odnosu na ostatak teksta, dok komanda `\textbf` u drugom pasusu označava deo teksta koji će biti ispisan **podebljano**. Slično, komandom `\underline` (u poslednjem pasusu) obeležen je deo teksta koji će biti podvučen. U svim slučajevima, odgovarajući delovi teksta se pojavljuju kao argumenti komandi. Dodatno, drugi pasus je izdvojen u okruženje `center` radi centriranog formatiranja. Tabele 1.1 i 1.2 daju primere još nekih često korišćenih osnovnih komandi i okruženja u \LaTeX -u, koji su dovoljni za kreiranje jednostavnih datoteka.

U drugom poglavlju primera na strani 13 ilustrovan je zapis jedne jednostavne matematičke formule. Formule se zapisuju izdvajanjem u tzv. matematičko okruženje uokvireno znacima `$$`. Komande `\forall`, `\geq`, `\in` i `\varepsilon` ispisuju redom simbole \forall , \geq , \in i ε , dok „komande“ `^` i `_` služe za ispis indeksa koji se navode kao argumenti.

⁶U skladu sa onim što je navedeno ranije u poglavlju 1.4, zapise logoa smo razdvojili od ostalog teksta upotrebom praznog simbola (`{}`) da bismo sprečili neželjeno spajanje sa sledećom rečju. Naime, blanko simbol u zapisu `\TeX` je koristi se samo da razdvoji komandu `\TeX` od teksta koji sledi, pa bi rezultat ispisa bio \TeX e. Upotreba praznog simbola razrešava ovaj problem.

```
% ----- datoteka: primer.tex

% ----- tip dokumenta je 'article' (npr. za radove, članke)
\documentclass{article}

% ----- pomoćni paket 'inputenc' za latin-2 kodiranje
\usepackage[latin2]{inputenc}

% ----- početak teksta
\begin{document}

% ----- Poglavlje 1
\section{\TeX{} i \LaTeX{}

\TeX{} je sistem za pripremu teksta za štampu koji je kasnih
sedamdesetih godina prošlog veka kreirao Donald Knut, sa ciljem
da napravi \emph{sistem za obradu teksta namenjen pisanju lepih
knjiga, pogotovo knjiga koje sadrže puno matematike}.

\begin{center}
\LaTeX{} je sistem za pripremu za štampu nastao na osnovama
\TeX-a. \\\ On definiše skup specifičnih \textbf{klasa
dokumenata} koje se koriste za formatiranje tekstova.
\end{center}

% ----- Poglavlje 2:
\section{Formatiranje teksta}

Pripremanjem teksta u \TeX-formatu potpuno precizno se
opisuje na koji će način on biti složen, pri čemu
su ti opisi najčešće savim prirodni i jednostavni.

\underline{Matematičke formule} se zapisuju izdvajanjem
pomoću simbola  $\$$  na sledeći način:


$$\forall x \in S_{\{1\}}: x^2 \geq \epsilon$$


% ----- kraj teksta

\end{document}
```

Slika 1.2: Primer ulazne datoteke

Dokument koji je zapisan u ulaznoj datoteci potrebno je, zatim, obraditi \LaTeX procesorom, komadom oblika

```
latex primer.tex
```

gde je `primer.tex` datoteka u kojoj je dokument sačuvan. Kao rezultat uspešnog procesiranja dokumenta, \LaTeX kreira datoteku `primer.dvi`, koja sadrži grafički opis dokumenta u DVI formatu. Ukoliko dođe do greške u ulaznoj datoteci (npr. neispravno obeležen tekst ili pogrešno uneta komanda), potrebno je — posle otklanjanja propusta — datoteku ponovo procesirati (otklanjanje greški je objašnjeno u dodatku [A](#)). Dokument u DVI obliku može se pomoću odgovarajućeg programa prikazati na ekranu, ili se može konvertovati u format pogodan za štampanje (videti takođe dodatak [A](#)). Za prikazivanje DVI datoteka na ekranu postoji veći broj programa: na primer, pod UNIX-om se u ovu svrhu može koristiti program `xdvi`, dok se pod WINDOWS-om može koristiti program `dviwin`. Primer sa slike [1.2](#) može se, u štampanoj verziji, videti u dodatku [B](#) (strana [96](#)).

Glava 2

Struktura L^AT_EX dokumenta

Umesto da od korisnika očekuje (ili da mu prepusti) da detaljno opisuje grafički izgled svog dokumenta (kao što je to slučaj u samom T_EX-u), L^AT_EX omogućava upotrebu određenih klasa dokumenata, na osnovu kojih će biti izgenerisan odgovarajući izgled teksta u skladu sa opšteprihvaćenim tipografskim standardima. Na taj način ne samo da se korisnik oslobađa mnogih tehničkih detalja, nego se i sprečava da napravi tipografske propuste u formatiranju svog dokumenta.

L^AT_EX podržava više standardnih tipova dokumenata sa odgovarajućom logičkom strukturom, koja može da podrazumeva organizaciju dokumenata u glave, poglavlja, pasuse i sl. Precizna struktura i odgovarajući način formatiranja dokumenata opisan je tzv. *klasama*. Klasa definiše opštu strukturu i izgled teksta, njegove elemente i osobine.

Pri kreiranju dokumenta navodi se klasa u skladu sa kojom će se formatirati tekst. Autor definiše logičku strukturu teksta, a na osnovu nje L^AT_EX određuje izgled teksta. Formatiranje definisano nekom klasom može biti prilagođeno potrebama specifičnim za neke tekstove (ili za neke podtipove dokumenata). Standardne klase se mogu modifikovati, a može se kreirati i potpuno nova klasa.

Kao što je navedeno u poglavlju 1.5, L^AT_EX-dokument se sastoji od preambule, koja definiše klasu i opšti izgled teksta, i samog teksta dokumenta.

2.1 Preambula

Preambula počinje naredbom koja zadaje jednu od postojećih *klasa* u skladu sa kojom se dokument kreira. Ova komanda mora biti prva komanda u dokumentu, a njen oblik je sledeći:

```
\documentclass[opcije]{klasa}
```

L^AT_EX obezbeđuje nekoliko standardnih klasa. Na primer, za formatiranje članaka i radova koristi se klasa `article`, dok se klasa `proc` koristi za formatiranje zbornika. Za pisanje knjiga koristi se klasa `book`, a klasa `letter` za

pisanje pisama. Različiti tipovi izveštaja mogu se formatirati u skladu sa klasom `report`, a prezentacije u skladu sa klasom `slides`.

Standardne klase mogu biti prilagođene deklarisanjem opcija u okviru parametra `opcije`. Ovaj parametar može biti i izostavljen, a u slučaju kada postoji više opcija, one se razdvajaju zarezima. Neke od standardnih opcija definišu:

- veličinu slova u osnovnom tekstu (npr. slova veličine `10pt`, `11pt` ili `12pt` tačaka; podrazumevana vrednost zavisi od klase);
- broj kolona u tekstu (za formatiranje teksta u jednu (`onecolumn`) ili dve kolone (`twocolumn`); podrazumevana vrednost je `onecolumn`);
- jednostrano ili dvostrano formatiranje teksta (opcije `oneside` za jednostrano i `twoside` za dvostrano slaganje; podrazumevana vrednost za klasu `book` je `twoside`, a za ostale standardne klase je `oneside`);
- kreiranje zasebne naslovne strane (opcija `titlepage` podrazumeva kreiranje zasebne naslovne strane, dok `notitlepage` ne izdvaja naslov na zasebnu stranu; za klasu `article` podrazumevana vrednost je `notitlepage`, a za klase `report` i `book` podrazumevana vrednost je `titlepage`);
- veličinu papira (npr. `a4paper`, `a5paper`, `b5paper`, `letterpaper`; podrazumevana vrednost zavisi od klase).

Na primer komanda `\documentclass[twocolumn,11pt]{book}` deklarise upotrebu klase `book` sa slovima veličine `11pt`, pri čemu će tekst biti složen u dve kolone. Naglasimo da korisnik ne treba da definiše veličinu strane, širinu kolona i sl., budući da se te vrednosti preuzimaju iz standardne klase (u ovom slučaju iz klase `book`).

Svaka od klasa definisana je odgovarajućom datotekom sa ekstenzijom `cls`. Na primer, parametri klase `article` definisani su u datoteci `article.cls`. Osim standardnih klasa, postoji i veliki broj klasa koje su kreirala razna udruženja, organizacije ili pojedinci, bilo modifikovanjem postojećih klasa ili definisanjem novih. Za upotrebu bilo koje nestandardne klase potrebno je navesti njeno ime u `\documentclass` komandi, a odgovarajuću `cls` datoteku smestiti u radni direktorijum (ili u odgovarajući sistemski direktorijum). Direktno kreiranje ili izmena datoteke klase nije preporučljiva, već se, umesto toga, ukoliko je potrebno prilagođavanje klase, preporučuje kreiranje odgovarajućih paketa (videti poglavlje 8).

Napomenimo da su u verziji \LaTeX 2.09 umesto klasa korišćeni tzv. stilovi (definisani odgovarajućim `sty` datotekama), i da je, shodno tome, u preambuli korišćena odgovarajuća `\documentstyle[opcije]{stil}` komanda. Da bi se „stara“ \LaTeX 2.09 datoteka prevela u novi, \LaTeX 2 ϵ format, u većini slučajeva je dovoljno zameniti `\documentstyle` komandu komandom `\documentclass`, pod uslovom da je raspoloživa i odgovarajuća `cls` datoteka.

Osim komandi definisanih standardnim klasama, za formatiranje teksta mogu se koristiti i različiti dodatni paketi koji proširuju skup i/ili funkcionalnost \LaTeX

komandi. Paketi se uključuju neposredno iza komande `\documentclass` naredbom oblika

```
\usepackage [opcije]{paket}
```

Navodimo nekoliko često korišćenih paketa, od kojih će neki biti opisani detaljnije kasnije u ovoj knjizi:

- `makeidx` — za kreiranje indeksa;
- `fancyheadings` — za kreiranje složenijih zaglavlja;
- `longtable` — za kreiranje velikih tabela;
- `epsfig` — za uključivanje slika u EPS (Encapsulated PostScript) formatu;
- `fontenc` — za podršku različitim kodiranjima fontova;
- `babel` — za višejezičko okruženje;
- `amstex` — za upotrebu prilagođenih $\mathcal{A}\mathcal{M}\mathcal{S}$ - \TeX komandi.

Slično klasama, i u radu sa paketima moguće je navesti opcije koje dodatno određuju raspoloživost komandi iz paketa. Na primer, komanda

```
\usepackage [english,serbian]{babel}
```

označava da će paket `babel` biti korišćen sa opcijama `english` i `serbian` koje uključuju podršku za engleski i srpski jezik (videti glavu 9).

Osim paketa, u preambuli se mogu navesti i definicije novih komandi i vrednosti raznih parametara koji se žele predefinisati (npr. veličina margine), o čemu će biti reči u glavi 3. Ukoliko korisnik redovno koristi određene predefinisane parametre, može ih izdvojiti u zasebnu datoteku (npr. `definicije.tex`), koju će komandom `\include{definicije.tex}` uključivati u preambulu svakog svog dokumenta.

2.2 Tekst dokumenta

Početak samog teksta dokument označava se komandom `\begin{document}`, a kraj komandom `\end{document}`. Tekst naveden iza naredbe `\end{document}` se ignoriše. \LaTeX ne zahteva da se ceo tekst dokumenta nalazi u jednoj datoteci, već se, naročito u slučaju dužih tekstova, preporučuje podela (npr. po celinama, tj. glavama ili poglavljima) u zasebne datoteke. Te datoteke se u glavni dokument mogu uključiti komandom `\include`. Ovom komandom postiže se veća preglednost \LaTeX -dokumenta, i olakšava upravljanje tekućom verzijom (budući da se određeni delovi dokumenta mogu lako isključiti iz procesiranja „stavljanjem pod komentar“; slično se može koristiti i par komandi `\iffalse` i `\fi`: tekst zapisan između njih biće ignorisan)¹. Na primer:

¹Za alternativni način „komentaranja“ delova teksta videti i poglavlje 8.

```

\documentclass{article}
\include{moje_definicije.tex}

\begin{document}
\include{poglavlje1.tex}
\include{poglavlje2.tex}
\include{poglavlje3} % .tex se podrazumeva
%\include{poglavlje4.tex} - još nije gotovo
\end{document}

```

U zavisnosti od klase, gotovo svaki dokument sadrži naslov i niz odgovarajućih tekstuelnih celina (npr. glave, poglavlja, i sl.). Osnovna ideja je da logička podela dokumenta na celine bude zadatak autora, a da se \LaTeX -u prepusti da iz te strukture složi najskladniju grafičku sliku teksta. U sledećim poglavljima izložićemo standardni način organizacije strukture dokumenata, kao i osnovne principe kojih se \LaTeX drži pri slaganju dokumenata koji pripadaju standardnim klasama.

Naslov dokumenta

Naslov celog dokumenta može se automatski formatirati upotrebom komande `\maketitle`, koja se navodi na željenom mestu u tekstu dokumenta (ne nužno na početku). Pre ove komande moraju se definisati atributi naslova komandama

```

\title{Naslov dokumenta}
\author{Ime Prezime, Institucija}
\date{datum}

```

Na osnovu vrednosti ovih atributa, komandom `\maketitle` izgenerisaće se skladan izgled naslova, koji će, pored samog teksta naslova, uključiti ime autora i datum. Ukoliko se želi izostavljanje nekog od ovih atributa pri formatiranju naslova, argument odgovarajuće komande treba ostaviti praznim (na primer `\date{}`)². Ukoliko dokument ima više autora, njihova imena treba razdvojiti komandom `\and` ili oznakom za prelom reda (`\\`). Prelom reda može se koristiti i za razdvajanje linija koje odgovaraju jednom autoru (npr. za razdvajanje imena i adrese). Dodatno, komandom `\thanks{tekst}` može se kreirati specijalna fusnota koja sadrži komentar koji se odnosi na naslov ili autora (npr. zahvalnica, napomena i sl.). Ova komanda se navodi u okviru argumenata neke od navedenih komandi. Dajemo jedan primer definisanja atributa naslova:

```

\title{Priprema teksta za štampu\thanks{Radna verzija}}
\author{Dunja Mitrović \\ Univerzitet u Beogradu \and
        Jovan Jovanović \\ Univerzitet u Novom Sadu \and
        Petar Petrović \\ Univerzitet Crne Gore}
\date{}

```

²Ako se komanda `\date{datum}` potpuno izostavi, biće prikazan tekući sistemski datum.

Ove komande se mogu navesti ili u preambuli ili u samom tekstu, dok se `\maketitle` navodi u okviru teksta na mestu na kome se želi naslov. U klasama `book` i `report` naslov dokumenta će uvek biti izdvojen na zasebnu stranu. Naslovna strana neće biti numerisana, a broj sledeće strane biće 1. U klasi `article`, tekst dokumenta koji sledi iza komande `\maketitle` ispisuje se neposredno iza formatiranog naslova, a naslov je moguće izdvojiti i na zasebnu stranu navođenjem opcije `titlepage` u komandi `\documentclass`.

Primer [B.20](#) (strana 143)

Sadržaj i izgled zasebne naslovne strane korisnik može i samostalno da kreira korišćenjem okruženja `\begin{titlepage}` i `\end{titlepage}`. U tom slučaju, korisnik mora da vodi računa o formatiranju svih detalja te strane (npr. veličina slova, raspored redova na strani, i sl.). Ovako kreirana strana nije numerisana, a broj sledeće strane u dokumentu je 1.

Definisanje strukture teksta

\LaTeX omogućava i podstiče opisivanje hijerarhijske *logičke strukture* teksta, odnosno organizovanje teksta u logičke celine (npr. glave, poglavlja, pasuse). Različitim standardnim klasama dokumenata odgovaraju različite logičke strukture. Na primer, u klasi `book` najviša celina je glava, dok su dokumenti klase `article` organizovani u poglavlja.

Označavanje početka nove celine teksta vrši se odgovarajućom komandom čiji je argument naslov te celine. Zadana struktura teksta određuje njegov grafički izgled (što, između ostalog, uključuje odgovarajuće formatiranje i numerisanje naslova celina), omogućava automatsko generisanje sadržaja (tj. formatiranog spiska naslova celina), referisanje na pojedine celine itd. U tabeli [2.1](#) navedene su raspoložive komande za označavanje celina u klasama `article`, `book` i `report` (pri čemu u klasi `article` nije raspoloživa komanda `\chapter`). Komande su navedene u hijerarhijskom poretku: prva je najviša u hijerarhiji, dok svaka sledeća označava celinu koja je direktna podcelina celine određene prethodnom komandom. Svaka celina može sadržati više svojih direktnih podcelina. Komande navedene u tabeli [2.1](#) generišu naslove celina u odgovarajućem obliku na mestu na kome je komanda zadata. Pri tome, naslov nove glave (i njen tekst) uvek će počinjati na novoj neparnoj strani u generisanom dokumentu.

| celina | komanda |
|-----------------|---|
| glava | <code>\chapter{Naslov glave}</code> |
| poglavlje | <code>\section{Naslov poglavlja}</code> |
| potpoglavlja | <code>\subsection{Naslov potpoglavlja}</code> |
| potpotpoglavlja | <code>\subsubsection{Naslov potpotpoglavlja}</code> |
| pasus | <code>\paragraph{Naslov pasusa}</code> |
| potpasus | <code>\subparagraph{Naslov potpasusa}</code> |

Tabela 2.1: Komande za označavanje celina

Pri formatiranju naslova celina, osim samog naslova generiše se i odgovarajućí redni broj celine. Sve celine u dokumentu se automatski numerišu u okviru svoje nadceline.³ Na primer, treće potpoglavlje četvrtog poglavlja druge glave biće označeno brojem **2.4.3**. Označavanje celina se realizuje korišćenjem brojača od kojih je svaki pridružen jednom tipu celine. Vrednosti brojača se automatski ažuriraju uvođenjem novih celina. Eksplicitna promena vrednosti brojača moguća je korišćenjem komande `\setcounter{tip_celine}{broj}`. Ova komanda eksplicitno postavlja vrednost brojača za izabrani tip celine, ne utiče na brojače nadcelina, a sve brojače podcelina postavlja na 0. Na primer, ako se želi da dato poglavlje bude označeno brojem **10** bez obzira na tekuću vrednost brojača, onda pre naredbe `\section{Naslov desetog poglavlja}` treba navesti komandu `\setcounter{section}{9}`. Ova naredba ne uzima u obzir oznaku prethodnog poglavlja, a (eventualno) sledeće poglavlje će imati oznaku **11**. Naredbe ovog tipa omogućavaju, između ostalog, lakšu pripremu delova dokumenta, a mogu se koristiti u preambuli i/ili u okviru teksta.

Napomenimo da je moguće i izostavljanje numeracije celine korišćenjem odgovarajuće komande za označavanje celine kojoj je dodat simbol `*` (na primer, `\section*{Naslov nenumerisanog poglavlja}`). Ova komanda ne menja nijedan od brojača celina, a odgovarajuća celina se neće pojaviti u sadržaju.

U klasama `article`, `book` i `report` raspoloživa je i komanda `\part` kojom se hijerarhijski najviše celine u tekstu grupišu u delove (eng. `part`) numerisane rimskim brojevima. Ovi brojevi se ne pojavljuju u oznakama podcelina. Komanda `\part` ne menja nijedan od brojača ostalih celina, a naslovi delova će se pojaviti u sadržaju. U klasama `book` i `report` naslovi delova biće ispisani na novim zasebnim neparnim stranama, a naredne, parne, strane biće ostavljene praznim.

U formatiranim naslovima delova i glava osim rednih brojeva biće ispisane i reči **Part** odnosno **Chapter**, koje mogu biti prilagođene drugim jezicima (ili potrebama) na način opisan u poglavlju 8.1, ili upotrebom paketa `babel` sa izborom odgovarajućeg jezika (na primer, za ispis pomenutih naslova na srpskom jeziku (**Deo**, **Glava**, itd.), dovoljno je u preambuli dokumenta navesti komandu `\usepackage[serbian]{babel}`, koja je detaljnije opisana u glavi 9).

Početak dela dokumenta u kome se nalaze dodaci označava se komandom `\appendix`. Celine na najvišem hijerarhijskom nivou biće od te pozicije označavane velikim slovima abecede. Dodatno, u klasi `book` naslovima glava smeštenih u dodatku biće dodata reč **Appendix** (umesto **Chapter**), što se takođe može prilagoditi na prethodno pomenute načine.

U klasi `book` raspoložive su i komande `\frontmatter`, `\mainmatter` i `\backmatter`, koje mogu biti korisne za podelu knjige u tri dela: uvodni, glavni i završni. One prilagođavaju naslove celina i numeraciju strana pravilima za formatiranje knjiga. Na primer, komanda `\frontmatter`, kojom započinje uvodni deo knjige, aktivira numeraciju strana rimskim brojevima. Ovaj deo uobičajeno obuhvata predgovor, sadržaj, liste tabela i slika i sl. Komanda `\mainmatter`

³Redni brojevi će biti prikazani samo za celine do nivoa određenog izabranom klasom. Na primer, u klasi `book` označavaju se celine do nivoa potpoglavlja.

označava početak glavnog dela i obično se navodi neposredno pre prve glave knjige. Ova komanda vraća numeraciju na arapske brojeve i postavlja broj stranica na 0. Spisak literature (bibliografija), indeks i slično navode se iza komande `\backmatter`. U standardnim klasama ova komanda nema vidljivih efekata.

U klasama `article` i `report` raspoloživa je i celina za formatiranje sažetka (tj. rezimea, apstrakta). Sažetak se kreira na željenom mestu navođenjem teksta sažetka između para komandi `\begin{abstract}` i `\end{abstract}`. Tekst sažetka biće ispisan fontom odgovarajuće veličine i naslovljen (bez numeracije) predefinisanim rečju **Abstract**, koja se takođe može promeniti na načine opisane ranije.

Automatski formatiran sadržaj strukturiranog dokumenta (tj. dokumenta sa označenim celinama) može se dobiti naredbom `\tableofcontents`. Lista naslova numerisanih celina i odgovarajući redni brojevi strana automatski će biti izgenerisani na poziciji na kojoj je ova komanda navedena. Ova lista će u dokumentima standardnih tipova biti naslovljena rečju **Contents**, koja može biti prilagođenja odgovarajućem jeziku njegovim navođenjem u opciji paketa `babel`, ili promenjena na način opisan u glavi 8.1. Naglasimo da u slučaju automatskog generisanja sadržaja dokument treba dva puta obraditi \LaTeX procesorom. U prvom razlazu \LaTeX izdvađa listu numerisanih celina i pripadajućih brojeva strana u privremenu datoteku sa ekstenzijom `toc`, dok se pri drugom procesiranju ta lista ugrađuje na odgovarajuće mesto u dokumentu.

Primer B.2 (strana 98)

2.3 Primer klase

Ovde dajemo primer klase `letter`, koja može da se koristi za formatiranje pisama. Iako ova klasa prirodno podrazumeva nešto drugačiju strukturu dokumenata od ostalih standardnih klasa, predstavimo je ovde kao primer klase zbog njene jednostavnosti. Da bi se dokument formatirao kao pismo, pored odgovarajuće `\documentclass{letter}` komande, u preambuli je potrebno navesti vrednosti koje definišu:

| | | |
|-------------------|----------|------------------------------|
| ime pošiljaoca | komandom | <code>\name{...}</code> |
| adresu pošiljaoca | komandom | <code>\address{...}</code> |
| datum slanja | komandom | <code>\date{...}</code> |
| potpis pošiljaoca | komandom | <code>\signature{...}</code> |

Ukoliko se želi izostaviti neki od ovih podataka, navode se prazne vitičaste zagrade (`{ }`). Navođenjem komande `\signature{...}` u preambuli obezbeđuje se potpis za kraj pisma, a ukoliko se ona ne navede, za potpis se koristi argument komande `\name{...}`.

Jedan dokument može da sadrži više pisama od kojih se svako piše unutar okruženja `letter`:

```
\begin{letter}{primalac}
```

```
...
\end{letter}
```

Argument `primalac` sadrži ime i adresu primaoca pisma, a može biti i prazan. Svako pismo biće formatirano na zasebnoj strani, ali sa istim zaglavljem koje je definisano komandama navedenim u preambuli. Obraćanje primaocu može se navesti u okviru naredbe `\opening{...}`, a pismo se zaključuje tekстом sadržanim u komandi `\closing{...}`.

Argument komande `\cc{...}` sadrži spisak primalaca kopija pisma, a argument komande `\encl{...}` spisak priloga. Komadnom `\ps` se označava početak post-skriptuma (pri čemu se ne ispisuje uobičajeno *P.S.*). Komandom `\makelabels` koja se navodi u preambuli generiše se posebna strana sa adresama svih primalaca.

Ovde navodimo jedan primer koji se u štampanoj verziji može videti u datku [B](#) (strana [100](#)).

```
\documentclass{letter}
\usepackage[latin2]{inputenc}
\name{Petar Petrović}
\address{Matematički fakultet \\ Studentski trg 16 \\
         11000 Beograd}

\begin{document}

\begin{letter}{Prirodno-matematički fakultet \\
             Univerzitet u Kragujevcu}

\opening{Poštovane kolege,}

Kao što vam je poznato, \LaTeX\ podržava više standardnih
tipova dokumenata sa odgovarajućom logičkom strukturom, koja
može da podrazumeva organizaciju dokumenata u glave,
poglavlja, pasuse i sl. Precizna struktura i odgovarajući
način formatiranja dokumenata opisan je tzv. \em klasama.
Klasa definiše opštu strukturu i izgled teksta, njegove
elemente i osobine.

\closing{Srdačno,}

\cc{Univerzitet u Nišu \\ Univerzitet Crne Gore}
\encl{Spisak predloga \\ Lista elektronskih adresa}
\end{letter}

\end{document}
```

Glava 3

Formatiranje teksta

Pored predefinisane formatiranja koje je zadato raznim parametrima izabrane \LaTeX klase, korisnik ima na raspolaganju i obilje mogućnosti za dodatno formatiranje izgleda teksta. Na primer, parametri formatiranja koji definišu poravnanja pasusa ili prorede u tekstu mogu se eksplicitno promeniti ukoliko je to potrebno. U ovoj glavi su razmotrene komande kojima se definiše izgled strane, prelamaju i poravnavaju redovi i strane, uvode liste i boksovi, i slično. Konačno, na kraju glave prikazano je i okruženje za tzv. doslovni ispis teksta, odnosno za prikazivanje teksta bez formatiranja.

3.1 Definisane izgleda strane

Svaku stranu čini zaglavlje, tekst strane i dno strane. Dok tekst strane služi za ispis samog dokumenta, zaglavlje i dno se koriste za ispis odgovarajućih informacija na svakoj od strana (npr. redni broj strane, imena autora, naslov glave ili poglavlja, i sl.). Njihove karakteristike su određene izabranom klasom, a mogu biti i promenjene.

Dužina i širina teksta na strani, kao i margine definisani su izabranom klasom i eventualnim opcijama (npr. `a4paper`, `b5paper`), a definisane veličine važe za ceo dokument. One se mogu i eksplicitno promeniti dodeljivanjem vrednosti odgovarajućim parametrima komandom `\setlength`, koja se navodi u preambuli. Vrednosti parametara zadaju se u jedinicama dužine¹. Parametrom `\textheight` eksplicitno se zadaje dužina, a parametrom `\textwidth` širina teksta na strani. Za eksplicitno menjanje vrednosti gornje margine koristi se parametar `\topmargin`, za levu marginu parnih strana parametar `\evensidemargin`,

¹Parametri koji u \LaTeX komandama označavaju dužinu mogu se zadati u različitim jedinicama: milimetrima (`mm`), centimetrima (`cm`) i inčima (`in = 2.54 cm`), kao i štamparskim (printerskim) tačkama (`pt`, printer point) i osnovnim širinama (`em`) i visinama (`ex`) slova u tekućem fontu. Štamparska tačka je fiksna jedinica koja se tradicionalno koristi za opis veličine slova: jednom centimetru odgovara 28.45 pt. S druge strane, dužina od `1em` jednaka je širini slova „M“, a dužina od `1ex` jednaka je visini slova „x“ u tekućem fontu; veličina `1em` pogodna je za definisanje dužina horizontalnih, a veličina `1ex` za definisanje dužina vertikalnih objekata.

a za levu marginu neparnih `\oddsidemargin`. Vrednosti parametara za margine dodaju se na predefinisane vrednosti, pa mogu biti i negativne. Veličine donje i desne margine automatski se podešavaju u skladu sa dužinom i širinom teksta. Promena navedenih parametara ilustrovana je sledećim primerima:

```
\setlength{\textheight}{21cm}
\setlength{\textwidth}{150mm}
\setlength{\topmargin}{-2cm}
\setlength{\evensidemargin}{13mm}
\setlength{\oddsidemargin}{12mm}
```

Slika 3.1 ilustruje parametre koji se mogu koristiti za definisanje izgleda strane. U slučaju formatiranja teksta u dve kolone (kada je opcija `twocolumn` navedena u komandi `\documentclass`), pored navedenih parametara moguće je menjati i parametar `\columnsep`, koji određuje razmak između kolona.

Visina zaglavlja odnosno dna strane određena je parametrima `\headheight` odnosno `\footheight`, dok su njihove širine definisane vrednošću parametra `\textwidth`.

Pored teksta koji se zapisuje u okviru strane, kraći tekst se može ispisati i na marginama komandom

```
\marginpar{tekst}
```

Osim podrazumevane vertikalne orijentacije papira (engl. portrait), strane se mogu formatirati i u horizontalnoj (engl. landscape) orijentaciji. Za tu svrhu se može koristiti opcija `landscape` u `\documentclass` komandi, ili paket `portland`.

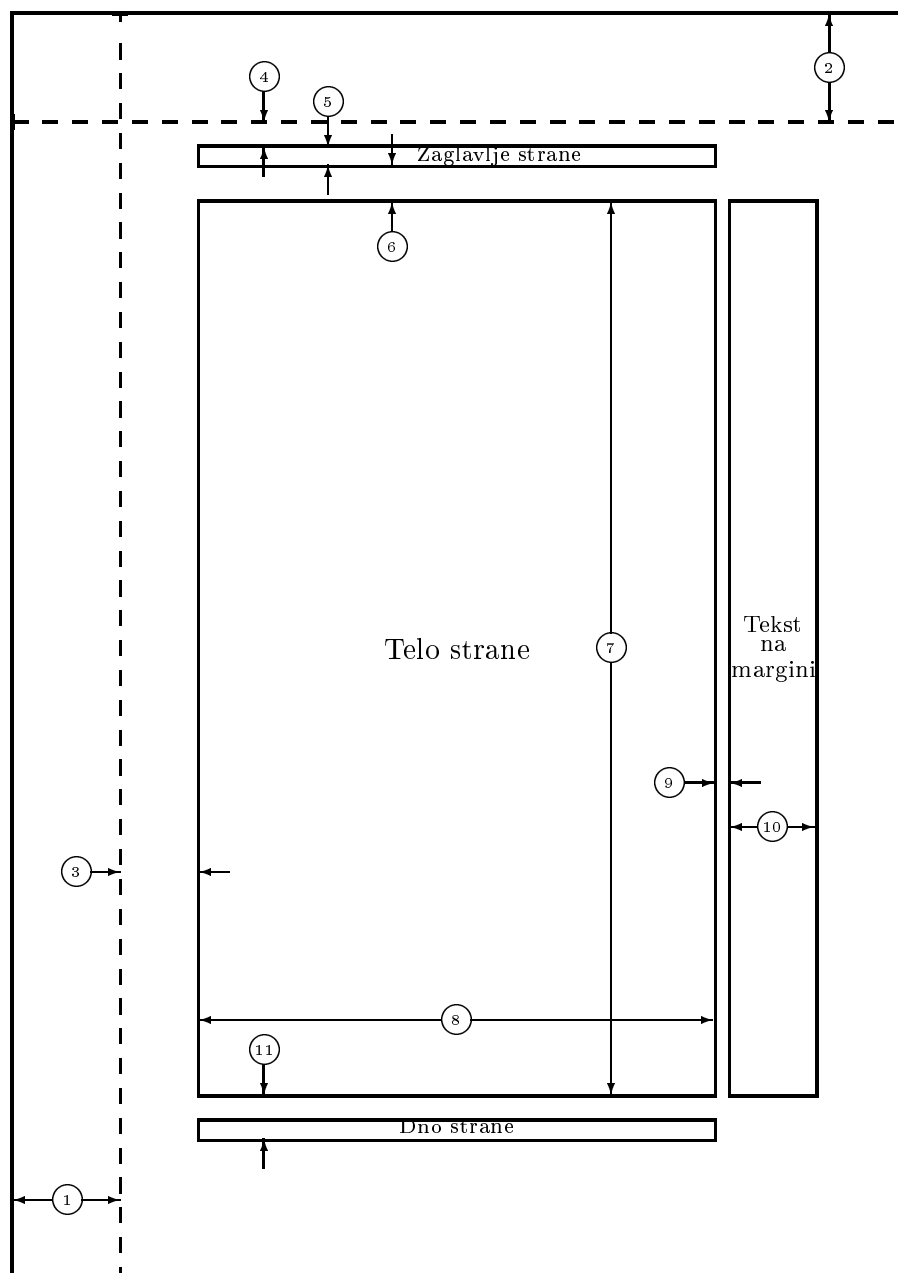
Numeracija strana se vrši automatski, a redni brojevi strana mogu biti prikazani u zaglavlju ili u dnu strane. Na primer, podrazumevani tip numeracije za klasu `article` je numeracija arapskim brojevima na sredini dna strane, a zaglavlje je prazno. Izgled zaglavlja i dna strane može se izabrati komandom `\pagestyle{opcija}` gde opcija ima jednu od sledećih vrednosti:

`empty` — zaglavlje i dno strane su prazni;

`plain` — dno strane sadrži centiranu numeraciju strane, a zaglavlje je prazno (ovo je podrazumevana vrednost za klase `article` i `report`);

`headings` — dno strane je prazno, a zaglavlje sadrži numeraciju strane i naslov odgovarajuće celine. Na primer, u klasi `article`, navodi se naslov poglavlja koji je tekuće na početku strane; ako je izabrana opcija `twoside`, onda se na parnim stranama navodi naslov poglavlja koje je tekuće na početku strane, a na neparnim naslov potpoglavlja koje je tekuće na kraju strane; u klasi `book` na parnim stranama se navodi naslov tekuće glave, a na neparnim naslov poglavlja koje je tekuće na kraju strane;

`myheadings` — dno strane je prazno, a zaglavlje sadrži numeraciju i tekst koji se definiše komandom `\markboth{levo zaglavlje}{desno zaglavlje}`.



| | | | |
|----|------------------------------------|----|--|
| 1 | <code>1 inč + \hoffset</code> | 2 | <code>1 inč + \voffset</code> |
| 3 | <code>\oddsidemargin = 53pt</code> | 4 | <code>\topmargin = 17pt</code> |
| 5 | <code>\headheight = 12pt</code> | 6 | <code>\headsep = 25pt</code> |
| 7 | <code>\textheight = 598pt</code> | 8 | <code>\textwidth = 345pt</code> |
| 9 | <code>\marginparsep = 11pt</code> | 10 | <code>\marginparwidth = 57pt</code> |
| 11 | <code>\footskip = 30pt</code> | | <code>\marginparpush = 5pt (nije prikazano)</code> |
| | <code>\hoffset = 0pt</code> | | <code>\voffset = 0pt</code> |
| | <code>\paperwidth = 597pt</code> | | <code>\paperheight = 845pt</code> |

Slika 3.1: Parametri za definisanje izgleda strane
(navedene vrednosti su podrazumevane za stranu formata A4 u klasi `article`)

Ukoliko je izabrana opcija `twoside`, onda se u prvoj zagradi navodi tekst za zaglavlja na parnim (tj. levim), a u drugoj tekst za zaglavlja na neparnim (tj. desnim) stranama. Ako je izabrana opcija `oneside`, onda se prvi argument komande `\markboth` ignoriše, a umesto nje se može koristiti i komanda `\markright{zaglavlje}` čiji je (jedini) argument tekst zaglavlja za sve strane. Ukoliko se ovaj stil koristi u klasama `book` i `report`, počeci celina tipa glava (`chapter`) i tipa deo (`part`) poništavaju dejstvo komandi `markboth` i `markright`, tj. zaglavlja treba definisati zasebno u svakoj od tih celina. Naglasimo da zaglavlje leve strane određuje poslednja `\markboth` komanda pre kraja te strane, a zaglavlje desne strane određuje prva `\markboth` (odnosno `\markright`) komanda na toj strani ako takva komanda postoji ili, u suprotnom, poslednja `\markboth` komanda pre te strane.² Komande `\markboth` i `\markright` mogu se koristiti i u stilu `headings`, pri čemu se njima zamenjuje automatski formatirano zaglavlje strane.

Za fleksibilniji izgled zaglavlja i dne strane mogu se koristiti opcije paketa `fancyheadings` (navođenjem komande `\usepackage{fancyheadings}` u preambuli). Na ovaj način, pored standardnih opcija, postaju dostupne i opcije `fancy` i `fancyplain`. Korišćenjem ovog paketa, zaglavlje i dno strane sadrže po tri komponente (levu, centralnu i desnu) čiji se sadržaj definiše komandama oblika:

```
\lhead[A2]{A1}
\chead[B2]{B1}
\rhead[C2]{C1}
\lfoot[D2]{D1}
\cfoot[E2]{E1}
\rfoot[F2]{F1}
```

Ukoliko je izabrana opcija `fancy`, ove komande imaju efekat ilustrovan slikom 3.2. U navedenim komandama, argument u uglastim zagradama je opcioni i predstavlja sadržaj komponenti za parne (tj. leve) strane, a argument u vitičastim sadržaj komponenti za neparne (tj. desne) strane. Ako se opcioni argument izostavi (u nekoj od komandi) koristiće se vrednost drugog (obaveznog) argumenta. U argumentima se, osim „običnog“ teksta, mogu pojaviti i komande `\thepage` (redni broj tekuće strane), `\thesection` (oznaka tekućeg poglavlja), `\thechapter` (oznaka tekuće glave), `\date` (za tekući datum), itd. Na primer, zaglavlja koja sadrže redni broj strane i redni broj poglavlja (na levim) odnosno odeljka (na desnim stranama) mogu se definisati komandama:

```
\lhead[\thepage]{\thechapter}
\rhead[\thesection]{\thepage}
```

²Na primer, ako bi se u dokumentu klase `article` sa izabranom opcijom `twoside`, na početku svakog poglavlja navela komanda `\markboth` sa naslovom poglavlja kao prvim argumentom (dok drugi nije bitan), a na početku svakog potpoglavlja komanda `\markright` sa naslovom potpoglavlja kao argumentom, dobio bi se efekat označavanja zaglavlja analogan korišćenju stila `headings`.

| | | | | | | |
|--------------|----|----|--|--------------|----|----|
| A2 | B2 | C2 | | A1 | B1 | C1 |
| tekst strane | | | | tekst strane | | |
| D2 | E2 | F2 | | D1 | E1 | F1 |

Slika 3.2: Prikaz izgleda parne (leve) i neparne (desne) strane dobijen korišćenjem paketa `fancyheadings`

Zaglavlje strane može biti podvučeno, a dno strane nadvučeno linijama različitih debljina. Podrazumevana debljina linije za podvlačenje zaglavlja je 0.4pt, a za nadvlačenje dna strane je 0pt (tj. nema nadvlačenja), što se može promeniti komandama tipa:

```
\setlength{\headrulewidth}{0.6pt}
\setlength{\footrulewidth}{0.2pt}
```

Opcija `fancyplain` ne zahteva definisanje sadržaja komponenti zaglavlja i dna strane, već one imaju vrednosti predefinisane u skladu sa izabranom klasom (zaglavlje strane sadrži naslove celina slično kao u opciji `headings`, a dno strane sadrži numeraciju kao u opciji `plain`). Napomenimo da paket `fancyheadings` pruža i druge mogućnosti za definisanje izgleda strane.

U tekstu može da se pojavi više `\pagestyle{opcija}` komandi sa različitim opcijama, pri čemu svaka ima dejstvo počevši od pozicije na kojoj je navedena. Ukoliko se želi da samo tekuća strana ima specifičan izgled, koristi se komanda `\thispagestyle{opcija}`.

Iako se strane u dokumentu automatski numerišu, eksplicitno postavljanje rednog broja strane može da se postigne upotrebom komande `\setcounter{page}{broj}`. Izbor tipa zapisa brojeva kojima se strane numerišu može se zadati komandom `\pagenumbering{tip_brojeva}` (pri čemu ova komanda resetuje brojač strana), gde `tip_brojeva` može imati jednu od sledećih vrednosti: `arabic` (za arapske brojeve, npr. 1, 2, 3, ...), `roman` (za rimske brojeve zapisane malim slovima, npr. i, ii, iii, ...); `Roman` (za rimske brojeve zapisane velikim slovima, npr. I, II, III, ...), `alph` (za mala slova abecede, npr. a, b, c, ...) i `Alph` (za velika slova abecede, npr. A, B, C, ...).

Komande koje se odnose na izgled strane i numeraciju, mogu se navesti ili u preambuli (kada deluju na ceo dokument) ili u samom tekstu (kada imaju dejstvo od te pozicije).

3.2 Prelom redova, pasusa i strana

Na osnovu (pre)definisanih veličina strane i margina, \LaTeX automatski „prelama“ redove, pasuse i strane, podrazumevano poravnavajući tekst na sve margine. Na primer, bez obzira na broj reči u jednom redu koji je unet u editoru, u

formatiranom redu biće tačno onoliko reči koliko može i treba da stane. To se postiže prenošenjem viška reči u novi red i podešavanjem razmaka između reči u okviru reda, kao i dodatnim prelamanjem reči kada je to potrebno. Slično, poravnavanje pasusa u okviru strane realizuje se podešavanjem odgovarajućih proreda između njih.

Osim povećanja razmaka između reči, one se i prelamaју ukoliko je to potrebno da red ne bi izgledao poluprazan ili prepun. Prelamanje se podrazumevano realizuje na osnovu pravila i obrazaca definisanih za engleski jezik, a odgovarajuća pravila i obrasce moguće je definisati i za ostale jezike (što izlazi iz okvira ove knjige). Ukoliko neku reč ne može zadovoljavajuće da prelomi, \LaTeX je može ostaviti na kraju reda i izvan desne margine (tzv. prepuni redovi, eng. *overflow box*). U takvim slučajevima, moguće je definisati pojedinačna pravila za prelamanje problematičnih reči. Mesta na kojima se neka reč u tekstu može ispravno prelomiti označavaju se komandom `\-` (npr. `pro\ble\ma\tič\no`, `naj\bo\lje`, `geo\met\rij\ski`). Ova komanda se odnosi samo na datu pojavu reči, dok se isti efekat za sva pojavljivanja navedene reči postiže komandom `\hyphenation`. Argumenti ove komande su reči u kojima su mesta mogućih prelamanja označena simbolom `-` (npr. `\hyphenation{geo-met-rij-ski}`).³ Parametrima `lefthyphenmin` i `righthyphenmin` može se definisati minimalni broj slova reči koja ostaju u tekućem, odnosno prenose se u sledeći red. Na primer, u skladu sa pravopisnim pravilima za naš jezik, drugom parametru treba dodeliti vrednost 2 komandom `\righthyphenmin 2`.

Komandom `\sloppy` prelamanje reči svodi se na minimum. Time se mogu izbeći prepuni redovi, ali razmak između reči u tom slučaju može biti prevelik. Razumno je koristiti ovu komandu onda kada nema potpunog rešenja za prelamanje reči i ako su ispravan prelom reči i poravnata desna margina važniji od sveukupnog izgleda dokumenta. Dodatno, da bi se sprečilo bilo kakvo prelamanje u okviru nekog niza reči, mogu se koristiti i tzv. boksovi (vidi poglavlje 3.4). Sa druge strane, komandom `\fussy` može se „vratiti“ na podrazumevani režim rada, odnosno ostavljanje reči koje prelaze na desnu marginu u slučajevima kada dati red ne može biti složen kako treba.

Osim automatskog prelamanja reda, eksplicitni prelazak u novi red (kada je potrebno) može se postići navođenjem tzv. terminatora reda. Terminatori reda su komande koje za efekat imaju prelazak u novi red. Na primer, komandama `\\` i `\newline` postiže se безусловni prelazak u novi red, pri čemu se tekući red ne poravnava na desnu marginu (red se „tvrdo“ prelama i ne uvode se dodatni razmaci između reči). Komande `\linebreak[n]`, odnosno `\nolinebreak[n]` sugerišu (ali ne garantuju) prelamanje odnosno neprelamanje reda na navedenom mestu, ali se red, u slučaju prelamanja, poravnava i na desnu marginu (povećavanjem razmaka između reči). Vrednost opcionog parametra `n` može biti 0, 1, 2, 3 ili 4: što je vrednost veća, to je manji stepen poštovanja podrazumevanih pravila za prelom reda (drugim rečima, `\linebreak[4]` u principu odgovara komandi `\newline`, osim što će red dodatno biti i desno poravnat).

³Na žalost, u komandi `\hyphenation` mogu se navoditi samo reči koje se zapisuju isključivo slovima engleskog alfabeta.

Ove komande su pogodne kao ispomoć \LaTeX -u u slučajevima kada red ne može da logično prelomi. Pored komandi navedenih u ovim pasusu, terminatori reda su i prazan red, oznaka za početak pasusa, komande kojima se označavaju počeci celina, itd.

Primer [B.4](#) (strana [102](#))

Prelom strana se vrši automatski, dok se bezuslovni prelazak na novu stranu može zahtevati navođenjem naredbe `\newpage`, koja „tvrdo“ prelama stranu na navedenom mestu (sadržaj strane se ne „razvlači“ tako da pokrije celu stranu). Naredbom `\pagebreak[n]`, odnosno `\nopagebreak[n]` sugeriše se (ali ne garantuje) prelamanje odnosno neprelamanje strane na navedenom mestu, pri čemu se — u slučaju preloma — sadržaj strane „poravnava“ i na gornju i na donju marginu. Slično odgovarajućim komandama za (ne)prelamanje redova, vrednost opcionog parametra `n` može biti 0, 1, 2, 3 ili 4: što je vrednost veća, to je manji stepen poštovanja podrazumevanih pravila za prelom strane (drugim rečima, `\pagebreak[4]` u principu odgovara komandi `\newpage`, osim što će strana biti dodatno poravnata na gornju i donju marginu). Tekuća strana može se prelomiti i komandama `\clearpage` i `\cleardoublepage` koje su detaljnije opisane u poglavlju [6.4](#).

Da bi se sprečilo prelamanje strane na neželjenom mestu, tekuća strana se može i eksplicitno produžiti (do određene mere) navođenjem komande `\enlargethispage{veličina}`. Parametar `veličina` određuje veličinu produženja strane i može biti zadat u jedinicama dužine ili u parametrima koji označavaju razmake u tekstu (npr. `\baselineskip`, `\parskip`, itd; videti poglavlje [3.3](#)). Na primer, komanda `\enlargethispage{\baselineskip}` omogućava produžavanje strane za jedan red. Komanda `\enlargethispage*{veličina}` dodatno pokušava da tekst tekuće strane sažme koliko je to moguće, i najčešće je prati komanda `\pagebreak`.

3.3 Proredi u tekstu

Proredi u tekstu mogu biti proredi između reči (u okviru reda), između redova (u okviru pasusa), i između pasusa (u okviru strane). Osnovne vrednosti za prorede su definisane odgovarajućim parametrima, ali korisnik može zahtevati i prorede zadatih veličina.

\LaTeX se automatski stara o podešavanju razmaka između reči (ako se tekst poravnava na obe margine). Proizvoljan broj blanko simbola između reči u ulaznoj datoteci tretira se kao jedan razmak, kao što više praznih redova ima isto dejstvo kao jedan prazan red. Dakle, dodatni razmak ili prored u redu ne može se dobiti navođenjem niza blanko simbola u editoru, niti se dodatni vertikalni razmak može napraviti ostavljanjem više praznih redova.

Horizontalni razmak na datoj poziciji u redu može se zahtevati naredbom `\hspace{veličina}`, gde se `veličina` zadaje u proizvoljnoj jedinici dužine (npr. `\hspace{0.5cm}`). Pomak može biti i negativan i služi za primicanje teksta. Razmaci na kraju reda automatski se uklanjaju, a da bi se to sprečilo koristi se komanda `\hspace*{veličina}`. Horizontalni razmaci se mogu dobiti i pomoću

komandi `\quad` (generiše razmak od `1em`, tj. prored koji odgovara širini slova „M“ u tekućem fontu) i `\qqquad` (razmak dva puta veći od `\quad`). Komanda `_` (blanko simbol neposredno sledi iza `\` simbola) generiše jedan blanko simbol srednje širine. Razmaci se takođe mogu dobiti i pomoću boksova (videti poglavlje 3.4), kao i komandom `\indent` (videti kasnije).

Primer B.4 (strana 102)

Prostor do kraja tekućeg reda može se popuniti praznim prostorom komandom `\hfill`, dok komanda `\dotfill` dopunjava tekući red tačkama. Preostalu slobodnu širinu tekućeg reda (do desne margine) vraća sekvenca `\stretch{1}`. Ona može, na primer, biti korisna za označavanje kraja dokaza odgovarajućim simbolom (npr. simbolom \square) navođenjem komande oblika:

```
\hspace*{\stretch{1}}$\Box$
```

\LaTeX podrazumevano tretira tačku (.) kao kraj rečenice, i pravi dodatni razmak u skladu sa pravopisnim pravilima. Da bi se taj razmak izbegao u zapisu godina ili u skraćenicama (u kojima se koristi tačka), iza tačke treba otkucati karakter `~` (na primer, npr. `~2003.~godina`). Dodatno, ovako obeležen razmak ne može biti iskorišćen za prelom reda. Potpuno isključivanje pravila „dodatnog razmaka iza tačke“ može se zahtevati komandom `\frenchspacing`.

Novi pasus u tekstu dobija se ostavljanjem bar jednog praznog reda ili navođenjem komande `\par`. Početak pasusa se podrazumevano uvlači za veličinu definisanu parametrom `\parindent`. Ukoliko se ne želi uvlačenje prve reči prvog reda u pasusu, treba navesti komandu `\noindent` na početku pasusa. S druge strane, komandom `\indent` pravi se horizontalni razmak veličine `\parindent`.

Osnovni razmak između redova u okviru pasusa definisan je parametrom `\baselineskip`, koji se može promeniti komandom `\setlength` ili komandom `\renewcommand` (videti glavu 8). Promena ima uticaja od mesta navođenja pa do eventualne nove takve naredbe. Međutim, \LaTeX može automatski resetovati ovu vrednost u slučaju promene fonta. Zbog toga se ne preporučuje menjanje ovog parametra, već parametra `\baselinestretch` koji skalira vrednost osnovnog razmaka. Podrazumevana vrednost za `\baselinestretch` je 1 (jednostruki prored), ali se može promeniti komandom `\renewcommand`. Na primer, komandom `\renewcommand{\baselinestretch}{2}` dobija se dvostruki prored u (čitavom) dokumentu.

Osnovni razmak između pasusa definisan je parametrom `\parskip` čija je predefinisana vrednost 0 (između pasusa se podrazumevano ne pravi dodatni razmak). Dodatni razmak se može dobiti promenom vrednosti ovog parametra (npr. navođenjem komande `\parskip=\baselineskip` dobijaće se po jedan prazan red između pasusa).

Naredbama `\smallskip`, `\medskip`, `\bigskip` dobija se, redom, mali, srednji odnosno veliki vertikalni prored u tekstu. Proizvoljan vertikalni prored se može dobiti naredbom `\vspace{veličina}`, gde se veličina zadaje u proizvoljnoj jedinici dužine (npr. `\vspace{15mm}`, `\vspace{\baselineskip}`). Pomak može biti i negativan i služi za primicanje teksta. Razmaci na kraju strane automatski se uklanjaju, a da bi se to sprečilo koristi se komanda `\vspace*{veličina}`.

Primeri B.4 (strana 102), B.5 (strana 104)

3.4 Boksovi

L^AT_EX omogućava formiranje tzv. *boksova* ili kutija (eng. box) u koje je moguće smestiti i više reči, a koji će se, sa stanovništa slaganja teksta, tretirati kao jedan znak. Otuda, sadržaj boksova se ne može prelomiti između redova ili između stana. Na sličan način se tretiraju i tabele (videti poglavlje 6.1).

Najjednostavniji boks može se kreirati komandom `\mbox{...}`. U ovom slučaju, širina boksa zavisi od širine njegovog argumenta. Boksovi fiksne veličine mogu se kreirati komandama oblika:

```
\makebox[veličina][ravnanje]{sadržaj}
```

gde *veličina* označava širinu boksa (npr. 10mm), a *ravnanje* može biti l ili r, i označava pomeranje sadržaja boksa ulevo (eng. left), odnosno udesno (eng. right). Ako se drugi par zagrada izostavi sadržaj je centriran. Parametar *sadržaj* predstavlja tekst koji će se nalaziti u boks. Ovakvi boksovi se mogu koristiti i za pravljenje horizontalnih proreda u tekstu (navođenjem praznog boksa, npr. `\makebox[10mm]{}`). Uokvireni boks dobija se naredbom oblika

```
\framebox[veličina][ravnanje]{sadržaj}
```

Ukoliko se želi grupisanje nekoliko redova u jedan boks, koriste se tzv. *parboksovi* koji se dobijaju naredbom

```
\parbox[pozicija][visina][tip][širina]{sadržaj}
```

gde *širina* označava veličinu parboksa (npr. 100pt), a *sadržaj* će biti formatiran u potrebnom broju redova u zavisnosti od širine parboksa. Argumenti *pozicija*, *visina* i *tip* su opcioni, pri čemu su prva dva ili oba navedena ili oba izostavljena. Pri tome

- *pozicija* određuje da li će boks biti vertikalno poravnat u odnosu na tekući red teksta po svom gornjem, najvišem redu (vrednost t, od *top*), po svom donjem, najnižem redu (vrednost b, od *bottom*) ili po svom srednjem redu (vrednost c, od *center*), što je podrazumevana vrednost.
- *visina* zadaje visinu boksa (u bilo kojim jedinicama dužine, npr. 15mm), a podrazumevana vrednost je visina tekućeg teksta.
- *tip* određuje poziciju teksta u okviru boksa. On može da ima jednu od sledećih vrednosti: t (od eng. top, — gornja linija boksa je poravnata sa osnovnom linijom teksta), c (od eng. center — tekst je vertikalno centriran, podrazumevana vrednost), b (od eng. bottom) — donja linija boksa je poravnata sa osnovnom linijom teksta), s (od eng. stretch vertically — tekst je „razvučen“ vertikalno (pri čemu je neophodno da tekst sadrži razmake koji to dopuštaju).

Ukoliko je tekst prevelik za parboks, preporučuje se korišćenje *mini-strane* koja se definiše okruženjem oblika

```
\begin{minipage}[pozicija]{širina}
sadržaj mini-strane
\end{minipage}
```

Parametri `pozicija` i `širina` imaju iste uloge kao i u naredbi `\parbox`. Mini-stranu je moguće formatirati analogno običnoj strani.

Primer [B.4](#) (strana [102](#))

3.5 Poravnavanja teksta

Osim teksta poravnatog na sve margine, može se eksplicitno zahtevati i drugačije formatiranje teksta. Na primer, delovi teksta mogu se formatirati bez prelamanja reči i bez podešavanja razmaka između njih:

levo poravnato — upotrebom okruženja `flushleft`:

```
\begin{flushleft}
tekst poravnat ulevo
\end{flushleft}
```

desno poravnato — upotrebom okruženja `flushright`:

```
\begin{flushright}
tekst poravnat udesno
\end{flushright}
```

centrirano — upotrebom okruženja `center`:

```
\begin{center}
tekst koji se centrira
\end{center}
```

Tekst naveden u ovim okruženjima biće formatiran kao novi pasus (sa podrazumevanim uvlačenjem prvog reda, videti poglavlje [3.3](#)) i razdvojen dodatnim proredima od ostatka teksta. Za odgovarajuća poravnavanja bez umetanja dodatnih proreda koriste se komande `\raggedright`, `\raggedleft` i `\centering`. Ove komande imaju efekta od mesta navođenja do kraja teksta (ili do naredne komande istog tipa), dok se formatiranje samo dela teksta postiže tako što se on piše unutar vitičastih zagrada sa odgovarajućom komandom na početku i praznim redom (ili bilo kojim drugim terminatorom reda) na kraju, pre zatvorene zagrade.

Komanda oblika `\centerline{centrirana linija}` može da se koristi za centriranje jedne linije teksta. Ova komanda ne uvodi dodatne prorede, ali joj mora neposredno prethoditi prazan red (ili bilo koji drugi terminator reda).

Delovi teksta mogu biti izdvojeno formatirani upotrebom sledećih okruženja:

```
\begin{quote}
tekst
\end{quote}
```

```

\begin{quotation}
tekst
\end{quotation}

```

Tekst naveden između parova ovih komandi biće poravnat i levo i desno, pri čemu je uvlačenje povećano u odnosu na obe margine. Pri tome, u okruženju `\quote` prvi red pasusa neće biti dodatno uvučen.

Primer B.5 (strana 104)

Dodatno uvlačenje celog pasusa u odnosu na margine definisano je parametrima `\leftskip` i `\rightskip`, koji imaju podrazumevane vrednosti 0cm. Ove vrednosti se mogu promeniti odgovarajućim `\setlength` komandama. Postavljene vrednosti ovih parametara imaju efekta samo na cele pasuse i to od mesta navođenja do kraja teksta (ili do naredne komande istog tipa).

3.6 Formatiranje listi

Postoji više načina za formatiranje nizova stavki (tj. listi) koji se koriste za različita nabranjanja:

opšta lista — upotrebom okruženja `itemize`:

```

\begin{itemize}
\item[oznaka prve stavke] Prva stavka
\item[oznaka druge stavke] Druga stavka
. . .
\end{itemize}

```

numerisana lista — upotrebom okruženja `enumerate`:

```

\begin{enumerate}
\item[oznaka prve stavke] Prva numerisana stavka
\item[oznaka druge stavke] Druga numerisana stavka
. . .
\end{enumerate}

```

opisna lista — upotrebom okruženja `description`:

```

\begin{description}
\item[oznaka prve stavke] Prva stavka
\item[oznaka druge stavke] Druga stavka
. . .
\end{description}

```

Oznake stavki koje se navode uz komandu `\item` su opcione u svim tipovima listi i ako se ne navedu koriste se podrazumevane vrednosti. U okviru navedenih okruženja može se koristiti novo nabranjanje koje daje ugnježdene liste. Dozvoljeno je korišćenje ugnježdavanja do četvrtog nivoa, a podrazumevane oznake stavki su prilagođene nivou liste kojoj ta stavka pripada.

Ukoliko se izostavi opciona oznaka stavki, u okruženju `itemize` se koriste podrazumevane vrednosti i to: `•` za prvi nivo, `–` za drugi nivo, `*` za treći nivo i

· za četvrti nivo, dok se u okruženju `enumerate` koriste arapski redni brojevi za prvi nivo, mala slova engleskog alfabeta za drugi nivo, rimski brojevi zapisani malim slovima za treći nivo i velika slova engleskog alfabeta za četvrti nivo. Za okruženje `description` podrazumevana oznaka je prazna, a ukoliko se navede, ispisuje se podebljanim (odnosno `\bfseries`) slovima (ukoliko nije eksplicitno navedeno drugačije).

Tekst naveden u okviru stavki se izdvaja u odnosu na ostatak teksta dodatnim proredima i dodatnim uvlačenjem u odnosu na levu marginu. U okruženjima `itemize` i `enumerate` oznake stavki se poravnavaju udesno (pri čemu duže oznake mogu da pređu levu marginu), a u okruženju `description` ulevo (pa je ono pogodnije za liste sa dužim oznakama stavki).

Primer [B.4](#) (strana [102](#))

3.7 Doslovni, neformatirani ispis teksta

U pripremi određenih delova teksta pogodno je izbeći uobičajeno formatiranje koje karakteriše podešavanje margina, proreda i uvlačenja, zamenjivanje višestrukih blanko simbola jednim razmakom, višestrukih praznih redova jednim praznim redom itd. Na primer, delovi programskog kôda su, zbog „nazubljenja“, čitljiviji ukoliko njihov ispis u potpunosti odgovara izgledu u tekst-editoru. Slično, i zapisivanje primera upotrebe samih \LaTeX komandi u nekom tekstu zahteva specifično formatiranje. U tu svrhu postoje komande koje omogućavaju ignorisanje \LaTeX formatiranja (i \LaTeX komandi), kao i prikaz dela teksta onako kako je zapisan u ulaznoj datoteci, tj. komande koje omogućavaju doslovni, neformatirani ispis teksta. Za takav ispis koristi se familija fontova `ttfamily` (koja podrazumevano ima vrednost `cmmt`, odnosno „slova pisaće mašine“, videti glavu [4](#)).

Za doslovni ispis kraćih delova teksta koristi se komanda oblika `\verb|doslovno prikazan tekst|`. Par simbola `|` označava početak i kraj teksta koji treba doslovno ispisati. Na primer, komanda `\verb|\LaTeX|` ispisace `\LaTeX`. Umesto simbola `|` kao graničnik se može koristiti bilo koji simbol različit od slovnih karaktera i od karaktera `*`. Na primer, komanda

```
\verb/if (a || b) { c++; d++; }/
```

daje sledeći rezultat:

```
[
  if (a || b) { c++; d++; }
]
```

Komanda `\verb*` razlikuje se od prethodne po tome što vidno označava sve blanko simbole. Na primer,

```
\verb*|if a < x^.v then x=x^.levo|
```


daje sledeći ispis:

```
[
  if_□a_□<_□x^ .v_□then_□x=x^ .levo
]
```

Napomenimo da se ove komande ne mogu koristiti u okviru argumenata drugih komandi (npr. u komandi za ispis fusnota).

Za neformatirani ispis većih delova teksta koristi se okruženje `verbatim`, u okviru koga će biti prikazani svi novi redovi i horizontalni i vertikalni razmaci:

```
\begin{verbatim}
tekst
\end{verbatim}
```

Na primer, ovo okruženje je pogodno za doslovni ispis programskog kôda, ali i \LaTeX datoteka (jer se ignoriše dejstvo \LaTeX komandi).⁴ Ukoliko je potrebno da se izbegne sãmo uobičajeno automatsko formatiranje a da \LaTeX komande ipak imaju dejstvo (na primer, komande za zapis slova sa akcentima), može se koristiti paket `alltt` kojim je definisano sledeće okruženje:

```
\begin{alltt}
tekst
\end{alltt}
```

U tekstu navedenom u okviru ovog okruženja dejstvo imaju \LaTeX komande koje počinju simbolom `\`, a vitičaste zagrade imaju uobičajene uloge. Ukoliko je potrebno formatirati matematičku formulu, koristi se okruženje `\(. . .\)` ili `\[. . .\]` (umesto `$. . . $` odnosno `$$. . . $$`, videti glavu 5.1).

Familija `ttfamily` koja se koristi za doslovni, neformatirani ispis teksta, može se promeniti na način opisan u poglavlju 4.5. Naglasimo da se samim prelaskom na familiju `ttfamily` ne može postići efekat doslovnog ispisa, već samo efekat promena fonta.

Primer B.4 (strana 102)

⁴Okruženje `verbatim` može da stvara problem ukoliko u tekstu ima `tab`-simbola, u okviru boksova i u nekim drugim situacijama. Za doslovni ispis koji nema ove nedostatke i pruža više mogućnosti preporučuju se drugi paketi, kao na primer paket `moreverb` (sa okruženjima `verbatimtab`, `listing` i drugim), ili paket `program` (za formatiranje programskog kôda).

Glava 4

Fontovi

Font je uređen skup grafičkih znakova koji imaju ujednačen izgled i metričke karakteristike. Izbor tzv. osnovnih fontova zasnovan je na izabranoj klasi (i navedenim opcijama), pri čemu je u tekstu moguće koristiti i dodatne fontove. U \LaTeX -u svaki font ima sledeće atribute:

kodiranje (eng. encoding) — redosled karaktera u fontu (npr. `OT1` — osnovno latinično kodiranje, `T1` — prošireno latinično kodiranje, `OT2` — osnovno ćirilično kodiranje, `OMS` — kodiranje matematičkih simbola, itd.);

familija (eng. family) — kolekcija grafičkih karakteristika koja određuje srodne fontove (npr. Knutove familije `cmr` — Computer Modern Roman, `cms` — Computer Modern Sans Serif, `cmtt` — Computer Modern Typewriter, familije `ptm` — Adobe Times, `phv` — Adobe Helvetica, `wncyr` — tzv. Vašingtonska ćirilica, itd.);

serija (eng. series) — širina i debljina slova (npr. `m` — slova srednje debljine, `b` — podebljana slova, `bx` — podebljana široka slova, `sb` — polupodebljana slova, `c` — kondenzovana slova, itd.);

oblik (eng. shape) — oblik slova (npr. `n` — normalna, uspravna slova, `it` — kurzivna, italik slova), `sl` — nagnuta slova, `sc` — sitna velika slova, itd.);

veličina (eng. size) — veličina slova (npr. `10pt`).

Svaki \LaTeX font je specifikovan vrednostima navedenih atributa. Na primer, specifikacija `OT1 cmr m n 10pt` odgovara fontu koji je označen sa `cmr10` i koji je podrazumevani font u \LaTeX dokumentima. Na svakoj poziciji u dokumentu aktivan je jedan, tzv. *tekući* font. Tekući font je inicijalno, na početku dokumenta, jednak tzv. *osnovnom*, *podrazumevanom* fontu.

4.1 Tekući i osnovni font

Na početku dokumenta, tzv. *osnovni*, odnosno *podrazumevani* font služi za inicijalizaciju tekućeg fonta, pri čemu je veličina slova zadana klasom dokumenta ili eventualno navedenim opcijama. Atributi osnovnog fonta (kodiranje, familija, serija i oblik) određeni su redom vrednostima parametara `\encodingdefault`, `\familydefault`, `\seriesdefault` i `\shapedefault`. Osnovni font na bilo kojoj poziciji u tekstu dokumenta može postati tekući font primenom komande `\normalfont`, pri čemu se zadržava tekuća veličina. Osnovni font koristi se i na svim pozicijama na kojima je implicitno primenjena komanda `\normalfont`, kao, na primer, u ispisivanju naslova celina u standardnim klasama ili pri upotrebi L^AT_EX 2.09 komandi `\bf`, `\it`, `\sl` (o kojima će biti reči u daljem tekstu).

Atributi tekućeg fonta čuvaju se u odgovarajućim internim parametrima. Njihovo direktno, neposredno menjanje se ne preporučuje i ovde ih nećemo navoditi. Vrednosti atributa tekućeg fonta mogu se posredno promeniti sledećim komandama niskog nivoa:

```
\fontencoding{kodiranje}
\fontfamily{familija}
\fontseries{serija}
\fontshape{oblik}
\fontsize{veličina}{osnovni razmak izmedju redova}
```

Svakom od navednih komandi menja se samo dati atribut tekućeg fonta¹ i on postaje aktivan tek nakon komande `\selectfont` (ostali atributi zadržavaju tekuće vrednosti). Za istovremenu promenu svih atributa tekućeg fonta (sem veličine) koristi se naredba `\usefont` oblika:

```
\usefont{kodiranje}{familija}{serija}{oblik}
```

Da bi se obezbedili prenosivost, uniformnost i udobniji rad, definisan je i osnovni skup fontova. Osnovni skup fontova određen je osnovnim skupom veličina i osnovnim skupom vrsta slova (vrstu slova, odnosno vrstu fonta, karakterišu atributi familija, serija i oblik).

4.2 Osnovni skup veličina slova

Osnovna veličina fonta, odnosno slova i drugih simbola u tekstu² određena je odgovarajućom opcijom (10pt, 11pt ili 12pt) u `\documentclass` naredbi, odnosno sâmom klasom ukoliko je ta opcija izostavljena (na primer, za klasu `article` osnovna veličina slova je 10pt). Osnovna veličina određuje tekuću veličinu slova na početku dokumenta. Delove teksta moguće je pisati i drugim

¹Promena tekućeg kodiranja i/ili familije zahteva postojanje odgovarajuće datoteke za definisanje fonta (tzv. *font definition*, `fd` datoteka); o ovim datotekama biće reči u poglavljima 4.4 i 4.5.

²Veličina slova se tradicionalno izražava u printerskim tačkama (pt). Jednom centimetru odgovara 28.45 printerskih tačka.

veličinama slova iz tzv. osnovnog skupa veličina slova (neposredno je raspoloživo deset veličina). Naredbe za promenu tekuće veličine slova se koriste tako što se deo teksta piše unutar vitičastih zagrada (`{, }`) sa odgovarajućom naredbom na početku. Osnovna veličina slova odgovara komandi `\normalsize`. Ilustracija odgovarajućih komandi data je u tabeli 4.1, a veličine slova u zavisnosti od osnovne veličine date su u tabeli 4.2.

| komanda | izgled slova |
|---------------------------------|--------------|
| <code>{\tiny Aa}</code> | Aa |
| <code>{\scriptsize Aa}</code> | Aa |
| <code>{\footnotesize Aa}</code> | Aa |
| <code>{\small Aa}</code> | Aa |
| <code>{\normalsize Aa}</code> | Aa |
| <code>{\large Aa}</code> | Aa |
| <code>{\Large Aa}</code> | Aa |
| <code>{\LARGE Aa}</code> | Aa |
| <code>{\huge Aa}</code> | Aa |
| <code>{\Huge Aa}</code> | Aa |

Tabela 4.1: Standardne komande za promenu tekuće veličine slova

| opcija | 10pt | 11pt | 12pt |
|----------------------------|------|------|------|
| <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 |

Tabela 4.2: Veličine slova u zavisnosti od osnovne veličine

Naredbe za promenu tekuće veličine slova mogu se koristiti i bez zagrada, pri čemu izabrana veličina važi do kraja teksta ili do pojave nove naredbe te vrste. Za vraćanje na osnovnu veličinu slova koristi se naredba `\normalsize`. Na primer, tekst

U ispisu ovoga teksta koriste se `\scriptsize scriptsize`,
`\Large Large` i `\normalsize normalsize` slova.

daje sledeći rezultat:

```
[
  U ispisu ovoga teksta koriste se scriptsize, Large i normalsize slova.
]
```

4.3 Osnovni skup vrsta fontova

Pod vrstom slova (odnosno pod vrstom fonta) podrazumevamo osobine koje karakterišu atributi familija, serija i oblik. Kao i veličina, osnovne vrste slova (tj. vrednosti navedenih atributa) specifikovane su izabranom klasom dokumenta, pri čemu se delovi teksta u dokumentu mogu pisati i različitim vrstama slova.

Pri pripremi dokumenta inicijalno je na raspolaganju fiksiran skup vrsta fontova čiji su atributi određeni izabranom klasom dokumenta. Taj skup je određen kombinovanjem sledećih vrednosti atributa (u tabeli 4.3 su date njihove podrazumevane vrednosti za klasu `article`):

tri familije koje se označavaju sa: `rmfamily`, `sffamily` i `ttfamily`. Tekuće vrednosti ovih familija čuvaju se u sledećim parametrima: `\rmdefault`, `\sfdefault`, `\ttdefault`. U standardnim klasama ove familije su, podrazumevano, `cmr`, `cmss` i `cmtt`.

dve serije označene sa: `mdseries` i `bfseries`. Njihove vrednosti se čuvaju redom u parametrima `\mddefault` i `\bfdefault`. U standardnim klasama, ovi parametri imaju vrednost `m` (slova srednje, uobičajene debljine) i `bx` (podebljana proširena slova).

četiri oblika označena sa: `upshape`, `itshape`, `slshape` i `scshape`. Njihove vrednosti čuvaju se u sledećim parametrima: `\updefault`, `\itdefault`, `\sldefault`, `\scdefault`. U standardnim klasama, ovi parametri imaju, redom, vrednosti `n` (normalna, uspravna slova), `it` (italik, kurzivna slova), `sl` (nagnuta slova), `sc` (sitna velika slova).

Primitimo da su, u principu, inicijalno dostupne $3 \cdot 2 \cdot 4 = 24$ vrste fontova (naravno, ne računajući njihove razne veličine). Njih zovemo *osnovnim skupom vrsta fontova*. Svaki font iz ovog skupa se jednostavno može postaviti za tekući. Pri tome, ukoliko font sa traženim atributima nije raspoloživ u datoj instalaciji, \LaTeX procesor ne prijavljuje grešku, već primenjuje mehanizam za supstituciju fontova³. Osnovna zamisao je da tekući font uvek uzima vrednosti iz standardnog osnovnog skupa, a da se samo u specifičnim situacijama vrše promene

³ Mehanizam supstitucije definiše postupak izbora vrednosti atributa fonta kojim će biti zamenjen font koji ne postoji. Određivanje alternativnog fonta se vrši pokušajem zamene redom atributa oblik, serija, familija dok se ne pronađe prvi odgovarajući font (pri čemu je kodiranje fiksno). Ukoliko se takav font ne pronađe, koristi se tzv. *error font*. Svi ovi postupci zamena definisani su u `fd` datotekama, a mogu se preinačiti korišćenjem komandi `\DeclareFontSubstitution` i `\DeclareErrorFont`.

| parametar | atribut | vrednost za klasu <code>article</code> |
|-------------------------|----------|--|
| <code>\rmdefault</code> | familija | <code>cmr</code> |
| <code>\sfdefault</code> | familija | <code>cmss</code> |
| <code>\ttdefault</code> | familija | <code>cmtt</code> |
| <code>\mddefault</code> | serija | <code>m</code> |
| <code>\bfdefault</code> | serija | <code>bx</code> |
| <code>\updefault</code> | oblik | <code>n</code> |
| <code>\itdefault</code> | oblik | <code>it</code> |
| <code>\sldefault</code> | oblik | <code>sl</code> |
| <code>\scdefault</code> | oblik | <code>sc</code> |

Tabela 4.3: Parametri raspoloživih fontova za klasu `article`

osnovnog skupa vrsta fontova, kao i korišćenje fontova van tog skupa. Ilustracija podrazumevanog osnovnog skupa vrsta fontova za klasu `article` data je u tabeli 4.4. Ova tabela ilustruje i mehanizam zamene nepostojećih fontova (u vrstama koje specificuju fontove koji nisu raspoloživi).

Vrednosti parametara `\rmdefault`, `\mddefault`, `\updefault` (koje su u standardnim klasama jednake `cmr`, `m` i `n`) se inicijalno, na početku dokumenta, dodeljuju parametarima `\familydefault`, `\seriesdefault` i `\shapedefault` koji određuju osnovni font. O promenama podrazumevanih vrednosti svih ovih parametara videti poglavlje 4.5.

Tekuća vrsta slova eksplicitno se menja komandama koje su date u tabeli 4.5. Svaka od ovih komandi menja samo odgovarajući atribut tekućeg fonta, a ostali atributi ostaju nepromenjeni. Promena tekućeg fonta na neki iz osnovnog skupa postiže se navođenjem dela teksta unutar vitičastih zagrada sa odgovarajućom komandom na početku. Na primer, tekst

```
U ispisu ovoga teksta koriste se {\itshape kurzivna},
{\bfseries podebljana}, {\itshape\bfseries kurzivna
podebljana slova}, {\itshape\bfseries\large large
kurzivna podebljana slova} i {\sffamily slova
{\slshape sans serif} familije}.
```

daje sledeći rezultat:

```
[
U ispisu ovoga teksta koriste se kurzivna, podebljana, kurzivna podebljana slova, large kurzivna podebljana slova i slova sans serif familije.
]
```

Navedeni primer pokazuje i da je moguće kombinovati promene više različitih atributa tekućeg fonta (pri čemu nije bitan poredak odgovarajućih komandi).

Da bi se pojednostavilo korišćenje navedenih komandi uvedene se ekvivalentne komande (navedene u tabeli 4.6) uniformnog oblika koje ne sadrže tip atributa koji se menja. Ove komande imaju dejstvo na tekst koji se navodi kao argument.

| familija | serija | oblik | vrsta slova |
|----------------------------|--------------------------|-------------------------|---------------------------------|
| <code>rmfamily=cmr</code> | <code>mdseries=m</code> | <code>upshape=n</code> | CMR medium normal |
| <code>rmfamily=cmr</code> | <code>mdseries=m</code> | <code>itshape=it</code> | <i>CMR medium italic</i> |
| <code>rmfamily=cmr</code> | <code>mdseries=m</code> | <code>slshape=sl</code> | <i>CMR medium slanted</i> |
| <code>rmfamily=cmr</code> | <code>mdseries=m</code> | <code>scshape=sc</code> | CMR MEDIUM SMALL CAPS |
| <code>rmfamily=cmr</code> | <code>bfseries=bx</code> | <code>upshape=n</code> | CMR bold normal |
| <code>rmfamily=cmr</code> | <code>bfseries=bx</code> | <code>itshape=it</code> | <i>CMR bold italic</i> |
| <code>rmfamily=cmr</code> | <code>bfseries=bx</code> | <code>slshape=sl</code> | <i>CMR bold slanted</i> |
| <code>rmfamily=cmr</code> | <code>bfseries=bx</code> | <code>scshape=sc</code> | CMR bold small caps |
| <code>sffamily=cmss</code> | <code>mdseries=m</code> | <code>upshape=n</code> | CMSS medium normal |
| <code>sffamily=cmss</code> | <code>mdseries=m</code> | <code>itshape=it</code> | <i>CMSS medium italic</i> |
| <code>sffamily=cmss</code> | <code>mdseries=m</code> | <code>slshape=sl</code> | <i>CMSS medium slanted</i> |
| <code>sffamily=cmss</code> | <code>mdseries=m</code> | <code>scshape=sc</code> | CMSS MEDIUM SMALL CAPS |
| <code>sffamily=cmss</code> | <code>bfseries=bx</code> | <code>upshape=n</code> | CMSS bold normal |
| <code>sffamily=cmss</code> | <code>bfseries=bx</code> | <code>itshape=it</code> | <i>CMSS bold italic</i> |
| <code>sffamily=cmss</code> | <code>bfseries=bx</code> | <code>slshape=sl</code> | <i>CMSS bold slanted</i> |
| <code>sffamily=cmss</code> | <code>bfseries=bx</code> | <code>scshape=sc</code> | CMSS bold small caps |
| <code>ttfamily=cmtt</code> | <code>mdseries=m</code> | <code>upshape=n</code> | CMTT medium normal |
| <code>ttfamily=cmtt</code> | <code>mdseries=m</code> | <code>itshape=it</code> | <i>CMTT medium italic</i> |
| <code>ttfamily=cmtt</code> | <code>mdseries=m</code> | <code>slshape=sl</code> | <i>CMTT medium slanted</i> |
| <code>ttfamily=cmtt</code> | <code>mdseries=m</code> | <code>scshape=sc</code> | CMTT MEDIUM SMALL CAPS |
| <code>ttfamily=cmtt</code> | <code>bfseries=bx</code> | <code>upshape=n</code> | CMTT bold normal |
| <code>ttfamily=cmtt</code> | <code>bfseries=bx</code> | <code>itshape=it</code> | <i>CMTT bold italic</i> |
| <code>ttfamily=cmtt</code> | <code>bfseries=bx</code> | <code>slshape=sl</code> | <i>CMTT bold slanted</i> |
| <code>ttfamily=cmtt</code> | <code>bfseries=bx</code> | <code>scshape=sc</code> | CMTT bold small caps |

Tabela 4.4: Ilustracija osnovnog skupa vrsta fontova (u kodiranju OT1) za klasu `article` (u imenima vrste slova, radi kraćeg zapisa, piše *bold* umesto *bold extended*)

Slično kao i naredbe za promenu tekuće veličine slova, naredbe za promenu tekuće vrste slova mogu se koristiti i bez zagrada, pri čemu izabrana vrsta važi do kraja teksta ili do pojave nove naredbe istog tipa. Ipak, preporučljivo je tekst koji se piše posebnom veličinom ili vrstom slova pisati između vitičastih zagrada, kako zbog bolje preglednosti, tako i zbog bolje kontrole dejstva odgovarajućih naredbi.

Komande navedene u tabeli 4.5 mogu se opisati na osnovu komandi niskog nivoa za promenu tekućeg fonta (videti poglavlje 4.1) što je prikazano u tabeli 4.7.

Umesto naredbi datih u tabelama 4.5 i 4.6 mogu se koristiti i jednostavnije komande koje su nasleđene iz verzije \LaTeX 2.09 (i zadržane u \LaTeX 2_ε zbog kompatibilnosti). Ilustracija tih komandi i ekvivalentni nizovi \LaTeX 2_ε komandi dati su u tabeli 4.8 (navedeni efekti se odnose na klasu, odnosno stil `article` pod pretpostavkom da nije menjan osnovni skup vrsta fontova (videti poglavlje 4.5)). Naglasimo da te komande menjaju sve atribute tekućeg fonta, pa se njima ne može postići kombinovanje promena različitih atributa tekućeg fonta.

| komanda | atribut | postavlja vrednost atributa na parametar |
|-----------------------------|--|---|
| <code>{\rmfamily }</code> | familija | <code>\rmdefault</code> |
| <code>{\sffamily }</code> | familija | <code>\sfdefault</code> |
| <code>{\ttfamily }</code> | familija | <code>\ttdefault</code> |
| <code>{\mdseries }</code> | serija | <code>\mddefault</code> |
| <code>{\bfseries }</code> | serija | <code>\bfdefault</code> |
| <code>{\upshape }</code> | oblik | <code>\updefault</code> |
| <code>{\itshape }</code> | oblik | <code>\itdefault</code> |
| <code>{\slshape }</code> | oblik | <code>\sldefault</code> |
| <code>{\scshape }</code> | oblik | <code>\scdefault</code> |
| <code>{\normalfont }</code> | kodiranje familija serija oblik | <code>\encodingdefault</code> <code>\familydefault</code> <code>\seriesdefault</code> <code>\shapedefault</code> |

Tabela 4.5: Standardne komande za promenu tekuće vrste slova

| komanda | ekvivalent |
|-------------------------------|--------------------------------|
| <code>\textrm{...}</code> | <code>{\rmfamily ...}</code> |
| <code>\textsf{...}</code> | <code>{\sffamily ...}</code> |
| <code>\texttt{...}</code> | <code>{\ttfamily ...}</code> |
| <code>\textmd{...}</code> | <code>{\mdseries ...}</code> |
| <code>\textbf{...}</code> | <code>{\bfseries ...}</code> |
| <code>\textup{...}</code> | <code>{\upshape ...}</code> |
| <code>\textit{...}</code> | <code>{\itshape ...}</code> |
| <code>\textsl{...}</code> | <code>{\slshape ...}</code> |
| <code>\textsc{...}</code> | <code>{\scshape ...}</code> |
| <code>\textnormal{...}</code> | <code>{\normalfont ...}</code> |

Tabela 4.6: Uniformne komande za promenu tekuće vrsta slova

Na primer, tekst

U ispisu ovoga teksta koriste se `\it` kurzivna,
`\bf` podebljana i `\sl` nagnuta `\rm` slova.

daje sledeći rezultat:

```
[
  U ispisu ovoga teksta koriste se kurzivna, podebljana i nagnuta slova.
]
```

Naglašavanje (eng. *emphasis*) dela teksta promenom oblika slova ali bez eksplicitnog navođenja željenog oblika, realizuje se korišćenjem naredbe `\em`. Na primer, tekst

| komanda | ekvivalent |
|-----------------------------|---|
| <code>{\rmfamily }</code> | <code>\fontfamily{\rmdefault}\selectfont</code> |
| <code>{\sffamily }</code> | <code>\fontfamily{\sfdefault}\selectfont</code> |
| <code>{\ttfamily }</code> | <code>\fontfamily{\ttdefault}\selectfont</code> |
| <code>{\mdseries }</code> | <code>\fontseries{\mddefault}\selectfont</code> |
| <code>{\bfseries }</code> | <code>\fontseries{\bfdefault}\selectfont</code> |
| <code>{\upshape }</code> | <code>\fontshape{\updefault}\selectfont</code> |
| <code>{\itshape }</code> | <code>\fontshape{\itdefault}\selectfont</code> |
| <code>{\slshape }</code> | <code>\fontshape{\sldefault}\selectfont</code> |
| <code>{\scshape }</code> | <code>\fontshape{\scdefault}\selectfont</code> |
| <code>{\normalfont }</code> | <code>\usefont{\encodingdefault}{\familydefault}{\seriesdefault}{\shapedefault}\selectfont</code> |

Tabela 4.7: Odnos komandi visokog i niskog nivoa za promenu tekućeg fonta

| komanda | $\text{\LaTeX} 2_{\epsilon}$ ekvivalent | vrsta slova |
|------------------|---|-------------------------|
| <code>\rm</code> | <code>\normalfont\rmfamily</code> | roman |
| <code>\sf</code> | <code>\normalfont\sffamily</code> | slova bez serifa |
| <code>\tt</code> | <code>\normalfont\ttfamily</code> | slova pisaće mašine |
| <code>\bf</code> | <code>\normalfont\bfseries</code> | podebljana slova |
| <code>\it</code> | <code>\normalfont\itshape</code> | <i>kurziv (italik)</i> |
| <code>\sl</code> | <code>\normalfont\slshape</code> | <i>nagnuta slova</i> |
| <code>\sc</code> | <code>\normalfont\scshape</code> | SITNA VELIKA SLOVA |

Tabela 4.8: \LaTeX 2.09 komande i $\text{\LaTeX} 2_{\epsilon}$ ekvivalenti za promenu tekuće vrste slova

U ispisu ovoga teksta koriste se `{\em istaknuta slova}`, `{\bfseries podebljana {\em istaknuta slova}}`, `{\sl nagnuta {\em istaknuta} slova}`, kao i `{\em dvostruko {\em istaknuta} slova}`.

daje sledeći rezultat:

```
[
  U ispisu ovoga teksta koriste se istaknuta slova, podebljana istaknuta slova, nagnuta istaknuta slova, kao i dvostruko istaknuta slova.
]
```

Komanda `\emph` ima isto dejstvo, s tim što se tekst koji se naglašava navodi kao njen argument (u okviru vitičastih zagrada). Komanda za naglašavanje menja samo atribut koji definiše oblik tekućeg fonta izabranog teksta i to na sledeći način:

```
slshape → upshape
itshape → upshape
upshape → itshape
```

scshape \rightarrow itshape

Primer B.6 (strana 106)

4.4 Kodiranje

Pod kodiranjem fonta⁴ podrazumeva se redosled karaktera u fontu (npr. OT1 (Knutovo, osnovno latinično kodiranje), T1 („Cork“, prošireno latinično kodiranje), OT2 (osnovno ćirilično kodiranje — Washington University Cyrillic encoding), T2 (skup ćiriličnih kodiranja), OMS (kodiranje matematičkih simbola) itd.). Razlog uvođenja različitih kodiranja je omogućavanje korišćenja raznih specifičnih simbola i alfabeta (npr. grčkog i ćiriličnih). Kodiranje OT1 je namenjeno pisanju tekstova na engleskom jeziku, dok je kodiranje T1 prošireno dodatnim simbolima za druge latinične alfabete (npr. ł, œ, đ, Đ).

Podrška upotrebi različitih kodiranja je jedno od značajnih unapređenja sistema $\text{\LaTeX} 2_{\epsilon}$: $\text{\LaTeX} 2.09$ je podržavao samo OT1 kodiranje, dok je u $\text{\LaTeX} 2_{\epsilon}$ ugrađena podrška i za OT1 i za T1, a mogu se koristiti i dodatna kodiranja.

Osnovno kodiranje u dokumentu je određeno parametrom `\encodingdefault` koji u standardnim klasama ima podrazumevanu vrednost OT1. Tekuće kodiranje je moguće promeniti komandom poput `\fontencoding{T1}` iza koje, ukoliko nije u preambuli, treba da sledi komanda `\selectfont`. Ako je T1 kodiranje aktivirano na opisani način, onda su, na primer, dostupne komande `\dj` i `\DJ` za slova đ i Đ (koje nisu dostupne u osnovnom, OT1 kodiranju).

Podrška za kodiranja OT1 i T1 je ugrađena u $\text{\LaTeX} 2_{\epsilon}$, dok je za ostala kodiranja potrebno koristiti paket `fontenc` (sa opcijama koje ukazuju na izabrana kodiranja). Sledeći primer ilustruje kako je moguće u jednom dokumentu koristiti latiničnu abecedu (`cmr` familiju) i grčki alfabet:

```
\documentclass{article}
\usepackage[LGR,T1]{fontenc}

\begin{document}

Ime \TeX{} izvedeno je iz grčke reči
{\fontencoding{LGR}\selectfont τεχνικος}.

\end{document}
```

```
[
Ime \TeX{} izvedeno je iz grčke reči τεχνικος.
]
```

U prethodnom primeru, kodiranje T1 je osnovno jer je navedeno kao poslednja opcija paketa `fontenc`. Na sličan način moguće je koristiti i ćirilične fontove (više o korišćenju ćirilice videti u glavi 9).

⁴U ovoj glavi pod „kodiranjem“ se uvek misli na „kodiranje fonta“. O kodiranju ulaznog teksta videti u glavi 9.

Naglasimo da je za svaki par kodiranje-familija potrebna odgovarajuća datoteka za definisanje fonta (engl. *font definition*, `fd` datoteka); na primer, za upotrebu grčkih slova u kodiranju LGR potrebna je datoteka `LGRcmr.fd`, za upotrebu ćirilične familije `wncyr` sa kodiranjem OT2 datoteka `OT2wncyr.fd`, dok je datoteka `T1cmr.fd` deo standardne instalacije.

Na izloženi način menja se samo tekuće kodiranje, bez uticaja na osnovno (podrazumevano) kodiranje i na osnovni raspoloživi skup fontova. Zbog toga bi, na primer, naslovi celina koji bi se pojavili u „grčkom“ delu prethodnog primera, bili ispisani upotrebom podrazumevanog, T1 kodiranja, čak i ako je, na primer, grčka familija izabrana za osnovnu i/ili tekuću. Slično, i korišćenje \LaTeX 2.09 komande `bf` (i srodnih) dovelo bi do upotrebe osnovnog a ne tekućeg kodiranja, što može dovesti do neodgovarajućih promena tekućeg fonta. Osnovno kodiranje menja se komandom oblika⁵:

```
\renewcommand{\encodingdefault}{OT2}
```

(iza koje, ukoliko nije u preambuli, treba da sledi komanda `\selectfont`).

4.5 Promena osnovnog skupa vrsta fontova

Osnovni skup vrsta fontova u dokumentu (koji je definisan izabranom klasom) može biti promenjen. Kao što je rečeno u poglavlju 4.3, taj skup određen je vrednostima parametara koji su navedeni u tabeli 4.3. Te vrednosti mogu biti promenjene komandama tipa:

```
\renewcommand{ime_parametra}{ime_ili_vrednost_parametara}
```

Na primer, komandom `\renewcommand{\rmdefault}{wncyr}`, ćirilična familija `wncyr` postaje osnovna familija dokumenta.⁶ Naglasimo da su komande ovog tipa deklarativne, pa imaju efekat tek nakon komande `\selectfont` (koja postavlja tekući font na osnovu tekućih vrednosti atributa) ili nakon komande kojom se aktivira odgovarajući atribut (čak i ako je taj atribut bio aktivan i pre komande `\renewcommand`). (Izuzetak od ovog pravila čini jedino komanda `\renewcommand{\rmdefault}{...}` koja je navedena u preambuli i za koju se odgovarajuća familija implicitno aktivira na početku dokumenta.)

Vrednost parametra može biti postavljena na konstantnu vrednost ili na vrednost nekog drugog parametra istog tipa. Na primer, efekat komande

```
\renewcommand{\itdefault}{\sldefault}
```

je da vrednost `\itdefault` ukazuje na vrednost parametra `\sldefault` i svaka promena ovog drugog će se direktno odraziti i na vrednost prvog. Sa druge

⁵ U primeru se koristi komanda `\renewcommand` kojom se menja postojeća definicija za neku komandu. Više o komandi `\renewcommand` videti u poglavlju 8.1.

⁶ Postavljanje određene familije za jednu od tri raspoložive zahteva postojanje odgovarajuće (tzv. *font definition*, `fd`) datoteke za njen opis. Na primer, datoteka `OT1cmr.fd` sadrži opis `cmr` familije, a `OT2wncyr.fd` ćirilične `wncyr` familije.

strane, komanda `\renewcommand{\itdefault}{sl}` postavlja vrednost parametra `\itdefault` na oblik `sl` (i na nju ne utiču promene vrednosti parametra `\sldefault`).

Promene osnovnog skupa vrsta fontova ilustrovane su sledećim primerom:

```
\documentclass[10pt]{article}

\begin{document}

{\tt Ovaj tekst ispisan je podrazumevanom
{\itshape tt familijom} i podrazumevanim {\it it oblikom}.}

\renewcommand{\ttdefault}{cmss}
\renewcommand{\itdefault}{sl}

{\tt Ovaj tekst ispisan je promenjenom
{\itshape tt familijom} i promenjenim {\it it oblikom}.}
\end{document}
```

```
[
Ovaj tekst ispisan je podrazumevanom tt familijom i podrazumeva-
nim it oblikom.
Ovaj tekst ispisan je promenjenom tt familijom i promenjenim it oblikom.
]
```

Navedeni primer ilustruje i različit efekat komandi `\it` i `\itshape`.

4.6 Korišćenje dodatnih fontova

\LaTeX daje mogućnost uvođenja i korišćenja bilo kojeg od raspoloživih fontova, uključujući i fontove izvan osnovnog skupa fontova. Tekući font, odnosno pojedine njegove atribute, moguće je promeniti komandama koje su navedene u poglavlju 4.1, na primer:

```
\fontencoding{T1}
\fontfamily{wncyr}
\usefont{T1}{wncyr}{bx}{sl}
\usefont{OT1}{\ttdefault}{m}{\itdefault}
```

U komandama koje se odnose na familiju potrebno je imati odgovarajuću `fd` datoteku. U tom slučaju, postaje raspoloživ čitav skup fontova određen datom familijom, te se pojedinačni atributi (npr. veličina, serija, oblik) tekućeg fonta mogu menjati na uobičajen način.

Dodatni font (pod navedenim imenom) sa fiksiranim vrednostima svih atributa može se definisati (čak i ako ne postoji odgovarajuća `fd` datoteka) komandama tipa:

```

\newfont{\novi_font}{cmr10 scaled\magstep1}
\newfont{\cirilica}{wncyr9 scaled\magstep1}
\newfont{\cir_bf}{wncyb9 scaled\magstep1}
\newfont{\adobe}{ptm8 scaled\magstep2}
\newfont{\cmff}{cmff10 scaled\magstep2}
\font\NoviFont=cmr10 scaled 1750

```

koje se navode u preambuli. Naglasimo da se ovakvim definicijama uvodi tačno po jedan dodatni font (tj. font sa fiksiranim vrednostima svih atributa, uključujući i veličinu). Promena fonta u okviru teksta u font specifikovan nekom ovakvom definicijom, realizuje se slično korišćenju naredbi za promenu vrste ili veličine slova:

```

\documentclass[10pt]{article}

\newfont{\cmff}{cmff10 scaled\magstep2}

\begin{document}

Ovaj tekst je ispisan {\cmff CMFF slovima \v{c}ija
{\large veli\v{c}ina} {\bfseries ne mo\v{z}e} biti menjana}.

\end{document}

```

```

[
Ovaj tekst je ispisan CMFF slovima čija veličina ne može biti me-
njana.
]

```

Naglasimo da standardne komande za promenu veličine i vrste slova poništavaju efekte naredbi za korišćenje dodatnih fontova.

4.7 Specijalni simboli

Simboli koji imaju specijalno značenje u L^AT_EX-u (npr. \, {, %, itd.), kao i specifična slova u različitim alfabetima zapisuju se komandama od kojih su neke date u tabeli 4.9.

Slova sa dijakriticima mogu se dobiti komandama datim u tabeli C.2 u dodatku C.

Za slova koja su definisana u nekom kodiranju mogu se kreirati komande koje omogućavaju korišćenje tih slova lokalnim promenama tekućeg kodiranja. Na primer, komande za slova đ i Đ mogu se definisati i koristiti u OT1 kodiranju na osnovu komandi koje postoje u T1 kodiranju:

```

\newcommand\DJ{{\fontencoding{T1}\selectfont\dj}}
\newcommand\DJj{{\fontencoding{T1}\selectfont\DJj}}

```

| komanda | simbol |
|-------------------------|--------------------|
| <code>\backslash</code> | <code>\</code> |
| <code>\\$</code> | <code>\$</code> |
| <code>\&</code> | <code>&</code> |
| <code>\%</code> | <code>%</code> |
| <code>\#</code> | <code>#</code> |
| <code>\{</code> | <code>{</code> |
| <code>\}</code> | <code>}</code> |
| <code>\-</code> | <code>-</code> |

Tabela 4.9: Zapisi nekih specijalnih simbola

Sa ovakvom definicijom, u tekstu se slova đ i Đ zapisuju komandama `\dJ{}` i `\Dj{}` (primetimo da su ove oznake morale da budu promenjene u odnosu na komande definisane za T1 kodiranje, kako bi se izbegla rekurzivna definicija). Više o slovima srpskog pisma i o obradi tekstova na srpskom jeziku videti u glavi 9.

Za neke specijalne simbole bez dijakritika postoje definisane komande (npr. komanda `\oe` za œ, `\o` za ø, `\copyright` za ©, `\ss` za ß, `\S` za §, itd.), čiji je pregled dat u tabeli C.1 u dodatku. Za neke specijalne simbole potrebno je učitati posebne pakete (na primer, simbol □ dobija se komandom `\Box` ako je učitan paket `latexsym.sty`).

Simboli se mogu zapisivati i direktnim referisanjem na njihov kôd u aktivnom kodiranju komandom `\symbol`. Kôd simbola može se navesti u dekadnom, oktalnom (kada mu prethodni apostrof), ili heksadekadnom sistemu (kada mu prethodi navodnik). Na primer, u podrazumevanom OT1 kodiranju, komandama `\symbol{27}` i `\symbol{"1B}` dobija se simbol œ, a komandama `\symbol{28}` i `\symbol{'34}` dobija se simbol ø.

L^AT_EX podržava i tradicionalno tipografsko pravilo korišćenja tzv. *ligatura*. Ligatura predstavlja kombinaciju više simbola koja se tretira i prikazuje kao jedinstven objekat radi boljeg estetskog efekta. U tipografiji je, na primer, uobičajeno da se slova „f“ i „i“ štampaju sa smanjenim razmakom kao u reči

fizika

što izgleda vizuelno prihvatljivije nego

fizika.

L^AT_EX procesor (bez eksplicitnog zahteva korisnika) prepoznaje niz slova koji odgovara ligaturi i zamenjuje ga jednim grafičkim simbolom (npr. u osnovnoj `cmr` familiji fontova, ligaturi „fi“ odgovara simbol `\symbol{12}`). Prikaz ligature na nekom mestu moguće je izbeći razdvajanjem simbola koji je čine separatorom `{}` ili `\-` (npr. `f{}izika`).

U različitim familijama fontova postoje različite ligature. Na primer, u osnovnoj, `cmr` familiji fontova, podržane su i sledeće ligature „fi“, „ff“, „fl“, „ffi“, „ffi“. Tehnika ligatura je u nekim familijama fontova iskorišćena i za zapis specifičnih simbola. Na primer, u ćiriličnoj `wncyr` familiji fontova i kodiranju

OT2, niz slova `nj` se tretira kao ligatura `њ`, pri čemu se u rečima u kojima treba izbeći upotrebu ove ligature koristi separator `{}` (kao u `in{ }jektivno`). Pregled ostalih ligatura u OT2 kodiranju fonta dat je u tabeli 9.3.

Ligaturama su realizovane i rastavne crtice različitih dužina, kao i dvostruki navodnici. Jedna crtica (`-`) daje najkraću povlaku (koristi se za višesložne reči, kao u *matematičko-programerski*), dve crtice (`--`) povlaku srednje dužine (npr. za razdvajanje brojeva, kao u *na stranama 123–126*), a tri crtice (`---`) daju dugačku povlaku (koristi se za razdvajanje umetnute rečenice). Za dvostruke navodnike koriste se, po srpskom pravopisu, ligature `,` i `‘ ‘` (`,primer‘ ‘` daje „primer“), a po engleskom pravopisu, ligature `‘ ‘` i `’ ’` (`‘ ‘example’ ’` daje “example”).

Za simbole koji se ne mogu dobiti korišćenjem standardnih dijakritika i nisu definisani ni u jednom raspoloživom kodiranju moguće je definisati nove komande (videti poglavlje 8.1). Na primer, slova `đ` i `Đ` (u kodiranju OT1) moguće je definisati na sledeći način:

```
\newcommand\d{d\kern-0.4em\symbol{22}\kern-0.1em}
\newcommand\D{D\kern-0.8em\raise0.2ex\hbox{--}\kern0.3em}
```

Ove definicije⁷ uvode komande kojima se preko njihovih argumenata ispisuje odgovarajuća horizontalna crta na potrebnoj visini. Slova `đ` i `Đ` se onda mogu dobiti komandama `{\d}` i `{\D}` (ili `\d{}` i `\D{}`).

⁷Komande koje su upotrebljene u ovim definicijama su naredbe osnovnog (*plain*) T_EX-a i njihov opis izlazi iz okvira ove knjige.

Glava 5

Formatiranje formula i teorema

U $\text{T}_\text{E}_\text{X}$ -u/ $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$ -u postoje dva režima rada — jedan za formatiranje „običnog teksta“ i jedan za formatiranje matematičkih formula. Delovi teksta označeni kao formule tretiraju se drugačije od ostatka teksta i simboli koji se u njima pojavljuju ispisuju se upotrebom specijalnih, matematičkih, fontova pri čemu su podrazumevani razmaci između simbola definisani na poseban način. U ovom delu knjige govori se o formatiranju teksta u matematičkom režimu, kao i generalno o formatiranjima potrebnim za pisanje matematičkih tekstova.

5.1 Formule

Matematičke formule mogu biti formatirane u okviru tekućeg reda ili izdvojene u odnosu na ostatak teksta. Formule zapisane između simbolâ $\langle \text{ i } \rangle$ ili, kraće, između para simbola $\$$ formatiraju se u okviru tekućeg reda. Formatiranje formule u zasebnom, centriranom i izdvojenom redu postiže se njenim zapisivanjem između simbolâ $\langle [\text{ i }] \rangle$, između para simbolâ $\$ \$$ ili u okruženju $\langle \text{begin}\{\text{displaymath}\} \text{ i } \text{end}\{\text{displaymath}\} \rangle$. Ako se koristi $\langle \text{documentclass} \text{ opcija } \text{fleqno} \rangle$, onda se ovako formatirane formule poravnavanju uz levu marginu. Formule se mogu zapisati i u okruženju $\langle \text{begin}\{\text{equation}\} \text{ i } \text{end}\{\text{equation}\} \rangle$. Ovo okruženje izdvaja formulu od ostatka teksta u zaseban red, centrira je i automatski numeriše.

Redni broj (izdvojene) formule se podrazumevano ispisuje uz desnu marginu. Ukoliko se u $\langle \text{documentclass} \text{ komandi navede opcija } \text{leqno} \rangle$, redni broj formule se ispisuje uz levu marginu. Numerisanje formula se vrši na nivou celina koji je određen izabranom klasom (npr. u klasi `book` numerisanje se vrši na nivou glava, dok je u klasi `article` numeracija na nivou celog dokumenta). Eksplicitna promena rednog broja koji označava formule vrši se komandom $\langle \text{setcounter} \rangle$. Na primer, $\langle \text{setcounter}\{\text{equation}\}\{100\} \rangle$ postavlja brojač na 100 i sledeća numerisana formula biti obeležena rednim brojem 101. Dodatno, za označavanje

formula zapisanih između parova simbolâ $$$$ mogu se koristiti i komande oblika $\backslash eqno\{oznaka\}$ i $\backslash leqno\{oznaka\}$. Komanda ovog tipa se navodi na samom kraju zapisa formule, a formula će biti označena sa desne, odnosno sa leve strane tekстом koji je naveden kao njen argument.

Razmaci u okviru formula automatski se podešavaju na specifičan način. Blanko simboli ne utiču na razmake, čak i kada se koriste kao separatori komandi. Na primer, svaki od zapisa

```
$x y + \sin x$,
$xy+\sin x$,
$x y+\sin{x}$ i
$xy +\sin{x}$
```

daje rezultat $xy + \sin x$. Naglasimo da zapis $\$xy+\sin x\$$ nije ispravan, jer između komande $\backslash sin$ i simbola x ne postoji separator. Podrazumevani razmaci se mogu i eksplicitno menjati: komande \backslash , $\backslash :$ $\backslash ;$ uvode manji, srednji i veći dodatni razmak, dok komanda $\backslash !$ podrazumevani razmak smanjuje. Dodatno, za veće razmake mogu se koristiti i komande $\backslash enskip$, $\backslash quad$ i $\backslash qquad$. Podrazumevani font u matematičkom modu je specijalni italik font (koji se ne primenjuje na oznake funkcija i operatora, kao npr. u $\sin x$).

Paket `amstex` proširuje L^AT_EX-a mnoštvom korisnih komandi za formatiranje matematičkih formula.¹

Primer [B.7](#) (strana 108)

5.2 Matematički simboli

Za zapis matematičkih simbola koji označavaju relacije, operacije, funkcije i slično, koriste se komande koje asociraju na značenje, ime ili izgled odgovarajućeg simbola. Na primer, simbol \emptyset se zapisuje komandom $\backslash emptyset$, simbol \int komandom $\backslash int$, simbol \leq komandom $\backslash leq$ (od engleskog *less or equal*), simbol \vee komandom $\backslash vee$ itd. Negacija bilo kog relacijskog simbola dobija se komandom $\backslash not$ (npr. $\backslash not\approx$ daje $\not\approx$). Grčka slova mogu se zapisati komandama datim u dodatku, u tabeli [C.5](#) (npr. slova α , β , γ , Γ , Ω zapisuju se komandama $\backslash alpha$, $\backslash beta$, $\backslash gamma$, $\backslash Gamma$, $\backslash Omega$). Primeri relacijskih i funkcijskih simbola dati su u tabeli [5.1](#) (detaljniji spisak ovih i drugih matematičkih simbola dat je u dodatku [C](#)). Postoje paketi koji definišu dodatne matematičke simbole (npr. `latexsym` i `amssymb`).

Sledeći primeri ilustruju korišćenje nekih matematičkih simbola:

```
$$\forall \varepsilon > 0(\exists \delta)(\forall x \in A)
(|x-a| < \delta \Rightarrow |f(x)-f(a)| < \varepsilon)$$
```

¹Na primer, paket `amstex` uvodi komandu $\backslash operatorname$ kojom se označava da je neki niz simbola oznaka funkcije/operatora i da treba da bude ispisan uspravnim, a ne italik fontom (npr. $\backslash operatorname{tg}$).

| komanda | simbol |
|------------------------------|-------------------|
| <code>\leq</code> | \leq |
| <code>\geq</code> | \geq |
| <code>\in</code> | \in |
| <code>\wedge</code> | \wedge |
| <code>\vee</code> | \vee |
| <code>\Rightarrow</code> | \Rightarrow |
| <code>\Leftrightarrow</code> | \Leftrightarrow |
| <code>\sin</code> | \sin |
| <code>\cos</code> | \cos |

Tabela 5.1: Zapisi oznaka nekih relacijskih i funkcijskih simbola

[$(\forall \varepsilon > 0)(\exists \delta)(\forall x \in A)(|x - a| < \delta \Rightarrow |f(x) - f(a)| < \varepsilon)$]

[`$$AB \parallel DE \wedge AC \parallel DF \wedge \alpha \cong \delta \Rightarrow \triangle ABC \cong \triangle DEF$$`]

[$AB \parallel DE \wedge AC \parallel DF \wedge \alpha \cong \delta \Rightarrow \triangle ABC \cong \triangle DEF$]

[]

U matematičkom tekstu (a i inače) vitičaste zagrade se zapisuju kao `\{ i \}`. Kada je potrebno da se u matematičkom tekstu veličina zagrada prilagođava kontekstu, onda se koriste komande `\left` i `\right`. Ovako zapisane zagrade se uvek moraju pojavljivati u parovima, ali mogu biti i različitog tipa, pri čemu se pod tipovima podrazumevaju male, uglaste, vitičaste, vertikalne i „prazne“ zagrade (koje se označavaju tačkom). Leve zagrade se zapisuju komandama

`\left(, \left[, \left\{, \left|, \left.`

a desne komandama

`\right(, \right[, \right\{, \right|, \right.`

Kao razdvajajući simboli čija se veličina prilagođava kontekstu, mogu se koristiti i simboli navedeni u tabelama [C.12](#) i [C.13](#).

Tri tačke (koje se tretiraju kao jedan znak) zapisuju se komandama `\ldots` ili `\dots` (horizontalne tri tačke na osnovnoj liniji — ...), `\cdots` (centrirane horizontalne tri tačke — ...), `\vdots` (vertikalne horizontalne tri tačke — ⋮), `\ddots` (dijagonalne horizontalne tri tačke — ⋱).

5.3 Složeni objekti

Pojedinačni matematički simboli se mogu kombinovati u složene objekte, pri čemu se njihove veličine, pozicije i međusobni odnosi automatski određuju u skladu sa kontekstom.

Donji indeksi, kao i donje granice suma, proizvoda, integrala, unija i slično, zapisuju se iza simbola $_$ i u okviru zagrada $\{ \}$ (koje mogu biti izostavljene ukoliko se indeks sastoji od samo jednog znaka). Gornji indeksi, stepeni i gornje granice zapisuju se analogno, iza simbola $^$. Na bilo kojoj poziciji u okviru matematičke formule može da se javi najviše jedan donji i najviše jedan gornji indeks. Svaki indeks može biti i složen objekat, pri čemu se grupisanje simbola određuje vitičastim zagradama. Upotrebu indeksa ilustruju sledeći primeri:

$$\begin{aligned} & \{\mathbf{a}_{\mathbf{b}}^{\mathbf{c}} = \mathbf{a}_{\mathbf{b}}^{\mathbf{c}} = \mathbf{a}^{\mathbf{c}}_{\mathbf{b}} \neq \{\mathbf{a}_{\mathbf{b}}\}^{\mathbf{c}} \neq \\ & \{\mathbf{a}^{\mathbf{c}}\}_{\mathbf{b}} \neq \mathbf{a}_{\{\mathbf{b}^{\mathbf{c}}\}} \neq \mathbf{a}^{\{\mathbf{c}_{\mathbf{b}}\}} \} \end{aligned}$$

[]

$$a_b^c = a_b^c = a_b^c \neq a_b^c \neq a_b^c \neq a_{bc} \neq a^{cb}$$

[]

$$\int_0^1 \int_{\pi}^{2\pi} f(x,y) dx dy$$

[]

$$\int_0^1 \int_{\pi}^{2\pi} f(x,y) dx dy$$

[]

$$\sum_{i=1}^n a_n \cos nx$$

[]

$$\sum_{i=1}^n a_n \cos nx$$

[]

$$\prod_{i=1}^1 \prod_{j=i+1}^{2003} (i^j - j^i)$$

$$\prod_{i=1}^1 \prod_{j=i+1}^{2003} (i^j - j^i)$$

`$$\lim_{n \rightarrow \infty} a_n$$`

$$\lim_{n \rightarrow \infty} a_n$$

`$$\max_{x \in A} f(x)$$`

$$\max_{x \in A} f(x)$$

`$$\bigcup_{i=1}^{\infty} \{\mathcal{F}_i\}$$`

$$\bigcup_{i=1}^{\infty} \mathcal{F}_i$$

Donja i gornja horizontalna vitičasta zagrada zapisuju se redom komandama `\underbrace` i `\overbrace`, kao što je ilustrirano sledećim primerom:

`$$\overbrace{a \dots a}^n \underbrace{a \dots a}_{vwx} a \dots a \overbrace{b \dots b}^n \overbrace{c \dots c}^n$$`

$$\overbrace{a \dots a \dots a}^n \underbrace{a \dots a}_{vwx} a \overbrace{b \dots b}^n \overbrace{c \dots c}^n$$

Za oznake kao što su „nadvučeno“, „podvučeno“, „kapa“, „vektor“ i sl. koriste se sledeće komande (videti dodatak C):

| | |
|-----------------------------------|------------------------|
| <code>\bar{a}</code> | \bar{a} |
| <code>\hat{a}</code> | \hat{a} |
| <code>\vec{a}</code> | \vec{a} |
| <code>\tilde{a}</code> | \tilde{a} |
| <code>\overline{a+b}</code> | $\overline{a+b}$ |
| <code>\underline{a+b}</code> | $\underline{a+b}$ |
| <code>\widehat{a+b}</code> | $\widehat{a+b}$ |
| <code>\widetilde{a+b}</code> | $\widetilde{a+b}$ |
| <code>\overrightarrow{a+b}</code> | $\overrightarrow{a+b}$ |

U okviru matematičkog teksta često se javlja i potreba za pisanjem jednog znaka iznad drugog. U tu svrhu se koristi komanda `\stackrel`, i to ilustruju sledeći primeri:

$$\stackrel{\text{def}}{=}$$

$$\stackrel{\rightarrow}{AB}$$

Za oznaku razlomka koristi se komanda `\frac` (od engleskog *fraction*). Iza komande `\frac`, u vitičastim zagradama navode se redom brojilac i imenilac. Na primer,

$$\sin^2 x = \frac{\tan^2 x}{1 + \frac{1}{\cot^2 x}}$$

Kao što se vidi iz primera, imenilac ili brojilac mogu ponovo biti razlomci (ili neki drugi matematički izrazi).

Za oznaku n -tog korena koristi se komanda `\sqrt[n]`. Ako se parametar `[n]` ne navede, dobija se oznaka za kvadratni koren:

```
$$a = \sqrt{b+c} + \sqrt[3]{b+c}$$
```

$$a = \sqrt{b+c} + \sqrt[3]{b+c}$$

Matrice, determinante i drugi matematički zapisi slične strukture zapisuju se u okruženju `\begin{array}` i `\end{array}` (eng. *array* — polje, niz, red). Ovakve strukture opisuju se u skladu sa pravilima koji važe u L^AT_EX-u za sva polja i tabele (pojedinačni elementi razdvajaju se simbolom `&` a vrste komandom `\\`; više detalja videti u poglavlju 6.1). U zapisivanju zagrada koriste se pravila opisana u poglavlju 5.2. Na primer, zapis

```
$$\left(\begin{array}{ccc}
a & b & c \\
d & f & g \\
e & h & i
\end{array}\right)$$
```

daje:

$$\left(\begin{array}{ccc} a & b & c \\ d & f & g \\ e & h & i \end{array} \right)$$

Slično matricama, zapisuju se i formule složenije strukture, što ilustruju sledeća tri primera:

```
$$\prod_{\substack{i,j=1 \\ i \neq j}}^{2003} (i^j - j^i) = 0$$
```

$$\prod_{\substack{i,j=1 \\ i \neq j}}^{2003} (i^j - j^i) = 0$$

```

$$u=\lim_{
\begin{array}{l}
x \rightarrow a \\
y \rightarrow b
\end{array}}
f(x,y)$$

```

$$u = \lim_{\substack{x \rightarrow a \\ y \rightarrow b}} f(x, y)$$

```

$$|x|=\left\{
\begin{array}{l}
x \text{ za } x \geq 0 \\
-x \text{ inače}
\end{array}
\right.
$$

```

$$|x| = \begin{cases} x & \text{za } x \geq 0 \\ -x & \text{inače} \end{cases}$$

Višelinijske formule (kod kojih je potrebno specifično poravnavanje i označavanje po linijama) zapisuju se slično poljima, u okruženju `\begin{eqnarray}` i `\end{eqnarray}` i to ilustruje sledeći primer:

```

$$\begin{eqnarray}
S_{\{1\}}(n) & = & 1+2 + \dots + (n-1) + n, \\
S_{\{2\}}(n) & = & 1^2+2^2+\dots \\
& & \dots + (n-1)^2+n^2.
\end{eqnarray}$$

```

$$S_1(n) = 1 + 2 + \dots + (n - 1) + n, \quad (5.1)$$

$$S_2(n) = 1^2 + 2^2 + \dots + (n - 1)^2 + n^2. \quad (5.2)$$

Sve linije ovako zapisane formule biće označene rednim brojevima (u nizu označenih formula) osim linija koje se završavaju komandom `\nonumber`. Poravnavanje linija se vrši po simbolima `&`, a krajevi pojedinačnih linija označavaju se simbolima za prelazak u novi red (simbolima `\\`). Komande `\begin{eqnarray*}` i `\end{eqnarray*}` imaju istu funkciju, s tom razlikom što se formule ne označavaju.²

5.4 Fontovi u matematičkom režimu rada

U matematičkom modu tekući font se može menjati komandama `\mathrm`, `\mathbf`, `\mathit`, `\mathcal` (ili `{\cal ...}`) redom za uspravna, podebljana italik i tzv. „kaligrafska“ slova (ove komande imaju dejstvo na tekst koji je naveden kao argument, u okviru vitičastih zagrada). Prilikom korišćenja bilo kojeg od ovih fontova važe pravila o razmacima u okviru matematičkog teksta. Font u okviru matematičkih formula nije neophodno eksplicitno navoditi — podrazumevani font je font koji se dobija komandom `\mathit` (ovaj font je sličan, ali ne i jednak običnom italik fontu). Može se omogućiti i korišćenje nekog dodatnog fonta u matematičkom režimu. Na primer, komandom

```
\DeclareMathAlphabet{\mathgot}{U}{euf}{m}{n}
```

navedenom u preambuli, omogućava se korišćenje `euf` familije fontova (tzv. gotski font ili gotica) u matematičkom režimu. Uz navedenu komandu, tekst

```
$$A_I \times \mathcal{B}_J = \mathit{A}_I \times \mathit{B}_J$$
```

daje sledeći rezultat:

$$A_I \times \mathcal{B}_J = \mathit{A}_I \times \mathit{B}_J$$

U matematičkom modu moguće je pisati i običan tekst (za koji ne važe pravila formatiranja matematičkih formula) korišćenjem komande `\mbox{...}`. Tekst naveden kao argument ove komande ispisuje se fontom koji je bio tekući pre početka formule i sa razmacima uobičajenim u običnom, ne-matematičkom tekstu. Za pisanje običnog teksta u matematičkom modu mogu se koristiti i komande `\textrm`, `\textsf`, `\texttt`, `\textbf`, `\textit`, `\textsl`, `\textsc` (kojima se bira vrsta slova za običan tekst).

²Višelinjske formule mogu biti zapisane i u obliku obične tabele, ali tada ne mogu biti označene, a i svi elementi moraju biti posebno označeni kao matematički delovi teksta.

5.5 Teoreme i slično

\LaTeX olakšava pisanje matematičkih tekstova i na taj način što korisnik može da mu prepusti označavanje definicija, teorema, lema i drugih sličnih celina. Označavanje takvih delova teksta vrši po celinama u okviru dokumenta. Koje će to celine biti bira korisnik u preambuli dokumenta.

Za deklarisanje teorema, lema, definicija i sličnih celina koristi se komanda `\newtheorem` čiji su parametri korisnikova interna oznaka za tako izdvojenu celinu, zatim tekst koji će biti ispisivan na početku takvih celina i, opciono, celine dokumenta po kojima će biti označavani ovako izdvojeni delovi teksta. Na primer, komandom:

```
\newtheorem{definicija}{Definicija}[section]
```

postize se da tekst koji je u dokumentu zapisan u okruženju `\begin{definicija}` i `\end{definicija}` bude ispisivan drugačijim fontom od ostatka teksta, da na njegovom početku piše **Definicija** i da nosi redni broj u okviru tekućeg poglavlja (npr. ako je u pitanju treća definicija u okviru drugog poglavlja definicija će nositi oznaku **2.3**). Referisanja (ukazivanja) na ovakve celine vrše se u skladu sa opštim pravilima (opisanim u glavi 7).

Paket `theorem` pruža dodatne mogućnosti za formatiranje teorema i sličnih celina.

Primer [B.7](#) (strana [108](#))

Glava 6

Formatiranje tabela i slika

L^AT_EX pruža različite mogućnosti za formatiranje i uključivanje tabela odnosno slika u dokument. U ovoj glavi biće razmotrena L^AT_EX okruženja za tabele i crteže, zatim uključivanje slika u L^AT_EX dokumente, kao i pozicioniranje ovakvih objekata u tekstu.

6.1 Okruženje za tabele

L^AT_EX omogućava jednostavno kreiranje tabela. Tabela se kreira unutar okruženja `tabular` koje počinje komandom `\begin{tabular}{kolone}` i završava se komandom `\end{tabular}`, a između kojih se stavlja sadržaj tabele. Parametar `kolone` opisuje izgled tabele: za svaku kolonu navodi se jedno od slova `l`, `r` ili `c` i ona označavaju da li se sadržaj te kolone ravna ulevo (`l`), udesno (`r`) ili se centrira (`c`), kao i način razdvajanja kolona (`|` za razdvajanje vertikalnim linijama, a blanko znak za razdvajanje prazninom). Tako, na primer, opis `\begin{tabular}{||l|rc||}` definiše tabelu sa tri kolone koja je spolja uokvirena sa po dve linije. Pri tome, prva i druga kolona su razdvojene linijom, a druga i treća nisu; sadržaj prve kolone biće poravnat uz levu ivicu, druga će biti centrirana, a sadržaj treće biće poravnat udesno.

Kraj vrste u tabeli se označava sa `\\`, a horizontalna linija se dobija naredbom `\hline`. Elementi pojedinih vrsta se razdvajaju znakom `&` i mora ih biti koliko i kolona (odnosno koliko i slova `l`, `r` ili `c` u parametru `kolone`). Neki elementi mogu da budu prazni (na primer, `pera & & mika \\`). Tabela se tretira kao jedan znak i tako se određuje njena pozicija (o fleksibilnijem pozicioniranju tabela videti poglavlje 6.4).

Kreiranje polja u tabeli koje obuhvata više kolona postiže se naredbom oblika:

```
\multicolumn{broj}{pozicija}{sadržaj}
```

gde `broj` predstavlja broj polja nad kojima se formira zajedničko polje, `pozicija` može biti `l`, `r` ili `c`, a `sadržaj` je sadržaj polja. Naredba `\cline{n-m}` iscrta

horizontalnu liniju samo od n -te do m -te kolone i služi za „podvlačenje“ zajedničkih polja.

Primer [B.8](#) (strana 110)

6.2 Okruženje za crteže

\LaTeX preko okruženja `picture` omogućava direktno kreiranje jednostavnih crteža, koji mogu sadržati linije, strelice, krugove i druge objekte. Koordinate tačaka, kao i sve ostale mere u okruženju za crtanje, izražene su u određenim jedinicama dužine. Jedinica dužine je definisana vrednošću `\unitlength`, koja se može promeniti `\setlength` komandom, kao na primer:

```
\setlength{\unitlength}{1mm}
```

Nije preporučljivo menjati jedinicu dužine unutar okruženja.

Okruženje za crtanje počinje komandom oblika:

```
\begin{picture}(širina,visina)(x0,y0)
```

Prvi par koordinata je obavezni argument i određuje širinu i visinu crteža. U pitanju su nominalne vrednosti kojima se \LaTeX -u stavlja do znanja koliko prostora treba da rezerviše za crtež — crtanje objekata izvan ovih granica (pa čak i izvan granica strane) neće, međutim, ovim biti sprečeno. Drugi par koordinata je opcioni argument i omogućava promenu koordinatnog početka. Inicijalno je koordinatni početak u donjem levom uglu crteža, a preko ovog argumenta se može pomeriti u proizvoljnu tačku. Okruženje se završava komandom `\end{picture}`.

Objekti se u okviru slike pozicioniraju komandom `\put(x,y){objekat}`. Prvi argument (x,y) je par koordinata koji određuje gde se na crtežu postavlja takozvana referentna tačka objekta. Drugi argument, `objekat`, predstavlja neku od komandi za crtanje.

Linije se crtaju komandom `\line` oblika `\line(dx,dy){dužina}` gde dx i dy određuju dužinu linije duž x odnosno y ose. Vrednosti dx i dy , koje određuju nagib linije, moraju biti celi, uzajamno prosti brojevi u intervalu $[-6, 6]$. Parametar `dužina` predstavlja dužinu projekcije linije na x osu. Referentna tačka za liniju predstavlja početnu tačku linije.

Komandom `\vector` crta se linija koja ima strelicu na jednom kraju. Sintaksa ove komande je identična sintaksi `\line` komande, s tim što ovde vrednosti za nagib moraju biti u intervalu $[-4, 4]$.

Pravougaonici se u okruženju za crtanje zadaju komandom `\framebox` o kojoj je već bilo reči u poglavlju [3.4](#). Treba voditi računa da su, u okviru okruženja za crtanje, dimenzije ovih objekata izražene u `\unitlength` jedinicama. Referentna tačka za pravougaonik je donji levi ugao pravougaonika.

Za crtanje kruga koristi se komanda `circle` čiji je oblik:

```
\circle[*]{poluprečnik}
```

Referentna tačka za krug je centar kruga. Varijanta komande `\circle*` crta popunjeni krug. Poluprečnik kruga može biti najviše 40pt za nepopunjeni, odnosno 15pt za popunjeni krug (ili ekvivalent u drugim jedinicama).

Komanda `\oval` služi za crtanje pravougaonika sa zaobljenim uglovima (tako zvanih ovala). Sintaksa komande je:

```
\oval(širina,visina)[deo]
```

Referentna tačka je centar ovala. Opcioni argument `deo` omogućava da se nacrtaju samo jedna polovina ili jedna četvrtina ovala i može imati vrednosti: `t` — za gornju polovinu, `b` — za donju polovinu, `r` — za desnu polovinu i `l` — za levu polovinu, dok se za crtanje četvrtine pravougaonika može koristiti kombinacija odgovarajuća dva slova.

U komandi `\put` može stajati i običan tekst i tada se on ispisuje na zadatoj poziciji. Tako bi, na primer, komanda:

```
\put(50,50){Zdravo}
```

ispisala reč `Zdravo` počev od tačke sa koordinatama (50, 50). Treba, međutim, imati u vidu da se `\unitlength` jedinice ne odnose na veličinu slova, već se tekst uvek ispisuje tekućim fontom.

Primer B.9 (strana 112)

Kako se može videti, okruženje za crtanje nameće veliki broj ograničenja, pa je stoga pogodno samo za najprostije crteže. Vremenom su razvijeni paketi koji prevazilaze neka od ovih ograničenja (na primer, paket `epic`), zatim paketi koji omogućavaju crtanje složenijih objekata (na primer, paket `bezier`), pa čak i paketi koji donose u \LaTeX dobar deo mogućnosti `POSTSCRIPT` [16] jezika za crtanje (na primer, paket `pstricks`). Treba pomenuti i programe `GCLC` odnosno `WinGCLC` [9], koji omogućavaju eksplicitno opisivanje složenih geometrijskih slika. Podrška za snimanje crteža u \LaTeX format je vremenom takođe ugrađena i u neke programe za crtanje, tako da se crteži napravljeni ovim programima mogu direktno uključiti u \LaTeX dokument. Primeri ovakvih programa su `gnuplot` [11] za crtanje grafika matematičkih funkcija i `xfig` [29] za vektorsko crtanje opšte namene. Ovi programi omogućavaju da se crteži kreiraju ili koristeći komande za crtanje visokog nivoa (kao u slučaju `gnuplot-a`) ili potpuno interaktivno (kao u slučaju `xfig-a`), a zatim pri snimanju te crteže prevode u odgovarajuće \LaTeX komande za crtanje koje koriste okruženje za crtanje ili neki od pomenutih dodatnih paketa. Na taj način, crteži se mogu kreirati znatno komfornije nego da se radi direktno u \LaTeX -u, ali ipak neretko su rezultati nezadovoljavajući, zato što \LaTeX nema dovoljno dobru ugrađenu podršku za crtanje.

Iz ovih razloga je češće slučaj da se crteži (kao i rasterske slike) unose u \LaTeX dokument u originalnom formatu, za šta \LaTeX pruža solidnu podršku i o tome će biti reči u nastavku.

6.3 Uključivanje slika u dokumente

\TeX , odnosno \LaTeX , sami po sebi nemaju ugrađenu podršku za uključivanje slika. Umesto toga, može se iskoristiti mogućnost da se \TeX komandom `\special` prenesu određene komande programu koji prikazuje ili obrađuje DVI datoteke. Ovim komandama se može naznačiti koju sliku treba uključiti u dokument, te gde i na koji način je postaviti. Vremenom su programi za rad sa DVI datotekama napredovali tako da prepoznaju jedan broj formata slika.

S obzirom na to da se u najvećem broju slučajeva \TeX dokument preko odgovarajuće DVI datoteke prevodi u POSTSCRIPT format radi štampanja, podrška za uključivanje slika u EPS (*encapsulated* POSTSCRIPT) formatu u \TeX dokumente je najbolje razvijena. Dodatni razlog što je podrška za ovaj format dobro razvijena jeste što on uključuje informaciju o fizičkim dimenzijama slike. Ta informacija je neophodna \LaTeX procesoru radi uključivanja slike u dokument, a vrlo mali broj formata za slike predviđa postojanje te informacije.¹

\LaTeX je tokom vremena obezbeđivao određene pakete koji su omogućavali da se izbegne potreba direktnog uključivanja komandi specifičnih za programe za rad sa DVI datotekama u dokumente radi uključivanja EPS slika. Rezultat evolucije ovih paketa je paket `graphicx` čije korišćenje danas predstavlja preporučeni način za uključivanje slika u dokumente, pa će u skladu sa tim i biti predstavljen u ovom tekstu.

EPS format je podskup POSTSCRIPT-a, a POSTSCRIPT predstavlja kompletan programski jezik za opis strane koja se prosleđuje štampaču. EPS format je znatno pojednostavljen u odnosu na POSTSCRIPT. Takođe, EPS datoteka obavezno mora u zaglavlju da ima specifikovanu veličinu uokvirujućeg pravouganika (takozvani *bounding box*) koji obuhvata sve objekte na slici. Već je pomenuto da je ovo važan preduslov da bi \TeX mogao da izvrši uključivanje slike u dokument. Jedinice u kojima se izražavaju širina i visina *bounding box*-a su takozvani POSTSCRIPT *point*-i.² Informacije o veličini *bounding box*-a omogućavaju \LaTeX -u da odvoji odgovarajući prostor na strani za sliku.

EPS slike se mogu dobiti iz odgovarajućih POSTSCRIPT datoteka tako što se zasebno izračuna i upiše u zaglavlje veličina *bounding box*-a i proveriti da slika ne sadrži zabranjene POSTSCRIPT komande.³ Mnogo češće se, međutim, EPS slike kreiraju direktno programima za crtanje. Svi bolji programi za vektorsko crtanje (na primer `xfig`, Adobe *Illustrator* [13], `CorelDRAW` [4], `AutoCAD` [3]) odnosno za rastersko crtanje (na primer `gimp` [10], Adobe *Photoshop* [25]) te programi za konverziju grafičkih formata (na primer `imagemagick` [14]) podržavaju snimanje slika u EPS format koje se onda mogu uključivati u \LaTeX dokumente.

Za uključivanje slika u dokumente koristi se komanda `\includegraphics` iz paketa `graphicx`. Sintaksa ove komande je sledeća:

```
\includegraphics[opcije]{datoteka.eps}
```

¹Ovo je posebno karakteristično za formate rasterskih slika. Kod ovih formata, veličina slike se obično izražava u pikselima, koji nemaju fiksne fizičke dimenzije.

²Ove jedinice iznose tačno $\frac{1}{72}$ inča i nešto su veće od \TeX *point*-a čija je veličina $\frac{1}{72.27}$ inča.

³Jedan način da se to uradi je da iskoristi program `pstoepsi`.

Ova komanda ne podrazumeva početak novog pasusa, tako da se može iskoristiti i za uključivanje slika i u oviru reda. Komanda podržava veći broj opcija, od kojih su neke nabrojane u tabeli 6.1. Opcije se mogu navoditi proizvoljnim redom i razdvajaju se na uobičajeni način zarezima; sve opcije iz tabele, kao i veći broj ostalih opcija, zadaju se u parovima `ime=vrednost`.

| opcija | značenje |
|---------------------|--|
| <code>height</code> | visina slike |
| <code>width</code> | širina slike |
| <code>angle</code> | ugao rotacije (u stepenima) originalne slike |
| <code>origin</code> | tačka oko koje se vrši rotacija |

Tabela 6.1: Opcije komande `\includegraphics`.

Sve dimenzije u tabeli se mogu navesti u bilo kojim \LaTeX jedinicama ili parametrima. Od svih pobrojanih opcija najčešće se zadaje širina slike. Prilikom zadavanja širine zgodno je iskoristiti vrednost parametra `\textwidth`, koja predstavlja širinu teksta na strani. Tako bi se, na primer, komandom:

```
\includegraphics[width=0.8\textwidth]{fig7.eps}
```

navelo da se slika `fig7.eps` prostire na 80% širine teksta i slika bi bila proporcionalno skalirana. Radi izbegavanja neuniformnog skaliranja slike, preporučeno je uvek zadavati samo jednu dimenziju slike — u tom slučaju slika se automatski skalira tako da odnos originalne širine i visine slike bude očuvan.

Često se u slikama koje se uključuju u \LaTeX dokumente javlja potreba za ispisom matematičkih i drugih specijalnih simbola koji postoje u \LaTeX -u. U tu svrhu se može koristiti paket `psfrag`, koji omogućava uključivanje proizvoljnog \LaTeX teksta na mesto postojećeg običnog teksta u EPS datotekama.

Treba pomenuti da je ponekad izgled rasterskih slika uključenih u \LaTeX tekstove u programima za pregled dokumenata nezadovoljavajući. Razlog tome je što takvi programi obično ne prikazuju dokument u prirodnoj veličini i što nemaju ugrađen neki sofisticirani algoritam za skaliranje slike. Kada se dokument štampa na papiru onda obično nema ovih problema. Takođe, štampači obično imaju znatno veću rezoluciju od rezolucije ekrana, tako da konačni rezultat daleko bolje izgleda odštampan. Vektorske slike, zbog mogućnosti skaliranja elemenata slike, obično izgledaju zadovoljavajuće i na ekranu i na papiru.

Primer B.10 (strana 114)

Da bi se slika u formatu koji nije EPS uključila u \LaTeX dokument, obično je najbolje da se konvertuje u EPS format. U tu svrhu se, ako program u kome je slika kreirana ne podržava snimanje u EPS format, može iskoristiti neki od programa za konvertovanje grafičkih formata opšte namene kakav je ranije pomenuti `imagemagick` program. Za neke grafičke formate postoje namenski programi za konverziju u EPS format koji mogu imati nekih prednosti u odnosu na konvertor opšte namene. Tako, na primer, za konvertovanje JPEG [2] formata u EPS format postoji namenski program `jpeg2ps` [18] koji koristi mogućnosti direktnog rada sa JPEG slikama u novijim POSTSCRIPT interpretatorima. Na ovaj način

se dobija znatno kraća EPS datoteka nego u slučaju kad se konverzija vrši, na primer, programom `imagemagick`. Prema tome, pre pristupanja konverziji vredi pažljivo proučiti sve relevantne informacije koje se odnose na originalni format.

Ako se iz nekog razloga želi da se konverzija u EPS format ili bilo kakva druga manipulacija datoteke sa slikom izvrši tokom \LaTeX procesiranja, onda se može iskoristiti `\DeclareGraphicsRule` komanda iz `graphicx` paketa. Ovom komandom se za datoteke sa datom ekstenzijom u argumentu `\includegraphics` komande specifikuje koja operacija treba nad njima da se izvrši i kako dalje da se tretiraju. Komanda međutim nije prenosiva i radi samo na operativnim sistemima koji podržavaju mehanizam takozvanih cevi (na primer, UNIX).

6.4 Pozicioniranje pokretnih objekata

Uključivanje slika ili tabela u \LaTeX dokumente može dovesti do problema prilikom slaganja teksta. Naime, kad ovi objekti ne mogu da stanu na tekuću stranu, \LaTeX procesor ih mora u celini preneti na sledeću. Ako želi da to izbegne, autor mora ručno da podešava poziciju objekta, i to obično svaki put kada nešto izmeni u tekstu koji mu prethodi. U cilju automatizovanja tog postupka, \LaTeX obezbeđuje okruženja `figure` i `table` koja vode računa o pozicioniranju objekata tako da se uvek dobije najbolji mogući izgled dokumenta.

Korišćenjem pomenutih okruženja pozicija slike odnosno tabele u tekstu više nije fiksirana, zbog čega se onda ovakvi objekti nazivaju pokretnim objektima.⁴ U skladu sa tim, u pratećem tekstu ne treba upotrebljavati formulacije tipa „sledeća slika“ ili „tabela koja sledi“ već se isključivo na ovakve objekte treba pozivati pomoću referenci (koje im se mogu pridruživati na način opisan u poglavlju 7.1). Okruženja za pokretne objekte počinju komandom:

```
\begin{figure}[pozicija]
```

za slike, odnosno komandom:

```
\begin{table}[pozicija]
```

za tabele. Opcionim argumentom `pozicija` mogu se \LaTeX -u preneti određene sugestije u pogledu pozicioniranja slike ili tabele. Vrednost ovog argumenta može biti bilo koja kombinacija slova `h`, `t`, `b` ili `p`, pri čemu `h` (od *here*) označava pozicioniranje objekta na mesto na kojem se u tekstu javlja okruženje `figure` ili `table`, `t` (od *top*) na vrh strane, `b` (od *bottom*) na dno strane i `p` (od *page*) na posebnu stranu (koja može eventualno da sadrži druge pokretne objekte). Podrazumevana vrednost ovog opcionog argumenta je `tbp`. Redosled slova nije bitan, već se pozicioniranje objekta uvek pokušava fiksni redom `h-t-b-p`, naravno uzimajući u obzir samo pozicije određene onim slovima koja su navedena u opcionom argumentu. U argumentu se može javiti i karakter `!` i u tom slučaju \LaTeX pokušava da pozicionira objekte na način koji je korisnik zadao, čak i ako

⁴Zapravo, unutar ovih okruženja ne moraju da budu slike odnosno tabele već se tu može naći bilo kakav sadržaj, ali je s obzirom na automatsko označavanje koje \LaTeX primenjuje na ova okruženja pogodno da to zaista budu slike odnosno tabele.

se to kosi sa njegovim pravilima o postizanju dobrog izgleda strane. Ovu opciju treba izbegavati, a ako se zadaje opcioni argument onda je dobro da on sadrži što veći broj slova kako bi \LaTeX procesoru bilo ostavljeno više mogućnosti u pogledu pozicioniranja. Treba imati na umu da se ovim argumentom procesoru samo sugeriše kako da pozicionira objekat. Ako nije u mogućnosti da postavi objekat na traženi način, \LaTeX će ove sugestije ignorisati. Opcije za zadavanje određenih sugestija u pogledu pozicioniranja pokretnih objekata treba koristiti umereno, u skladu sa opštim pravilom da se slaganje teksta u što većoj meri prepušta \LaTeX -u.

Da bi se razumeo raspored pokretnih objekata u određenim situacijama, korisno je poznavati osnove algoritma koji \LaTeX koristi za raspoređivanje. \LaTeX , naime, najpre nastoji da pozicionira svaki pokretni objekat na način na koji je autor zadao. Ako objekat ne može da stane na tekuću stranu, onda se on stavlja u takozvani red slika odnosno red tabela, a strana se popunjava tekстом koji sledi iza objekta. Kada započne procesiranje naredne strane, \LaTeX pokušava da napravi posebnu stranu samo od pokretnih objekata. Ako nema dovoljno objekata za takvu stranu, \LaTeX onda obrađuje jedan po jedan objekat kao da se svaki od njih upravo pojavio u tekstu. Pri tome, \LaTeX ponovo u pozicioniranju nastoji da sledi sve sugestije koje je autor eventualno zadao.⁵ Ako se i na novoj strani javi neki pokretni objekat, on se stavlja na kraj odgovarajućeg reda i obrađuje na isti način kao ostali objekti u redu.

U nekim situacijama, pokretni objekti se nagomilavaju u redovima i ne bivaju prikazani sve do kraja dokumenta. Tada se može iskoristiti komanda `\clearpage` kojom se naređuje da se red isprazni počev od naredne strane. Komandom `\clearpage`, međutim, odmah počinje nova strana, tako da tekuća strana može ostati nepopunjena. U takvim slučajevima može se iskoristiti komanda `\afterpage` iz paketa `afterpage`. Naime, komande koje se nalaze u argumentu ove komande se izvršavaju kada započne procesiranje strane koja sledi stranu na kojoj se nalazi `\afterpage` komanda. Na taj način, ako se izda komanda `\afterpage{\clearpage}`, tekuća strana će biti potpuno popunjena, a potom će na narednoj strani početi pražnjenje redova pokretnih objekata.

Komanda `\cleardoublepage` radi isto što i komanda `\clearpage`, s tim što u dvostrano formatiranom dokumentu vrši prelazak na novu neparnu stranu, ostavljajući jednu praznu stranu ukoliko je potrebno.

Unutar okruženja `figure`, odnosno `table` moguće je komandom `\caption` zadati naslov slike, odnosno tabele. Ova komanda ima jedan obavezni argument i to je tekst koji će biti ispisan uz sliku odnosno tabelu. Komanda ima kao opcioni argument tekst koji će biti ispisan za dati pokretni objekat u listi slika, odnosno tabela. Lista slika se može generisati komandom `\listoffigures`, a lista tabela komandom `\listoftables`. Ove komande su slične komandi `\tableofcontents` za kreiranje sadržaja dokumenta.

Za horizontalno poravnanje pokretnog objekta može se iskoristiti neka od komandi `\raggedleft`, `\centering` ili `\raggedright` za poravnanje ulevo, horizontalno centriranje ili poravnanje udesno. Tako bi, na primer, slika iz dato-

⁵Osim, naravno, *here* specifikatora koga više nije moguće primeniti.

teke `fig7.eps` mogla biti uključena u dokument kao centrirana i uz zadavanje ostalih pomenutih parametara na sledeći način:

```
\begin{figure}
\centering
\includegraphics[width=0.8\textwidth]{fig7.eps}
\caption{Primer slike}
\label{slike:primer}
\end{figure}
```

Svi primeri u ovom delu teksta se odnose na rad sa slikama, a sa tabelama se postupa na potpuno analogan način.

Kao naslov slike odnosno tabele, \LaTeX prvo ispisuje reč *Figure* odnosno *Table*, zatim redni broj (zasebna numeracija slika i tabela se automatski generiše i održava), potom dvotačku i na kraju nisku koja je zadata komandom `\caption`. Da bi se reči *Figure* odnosno *Table* promenile, može se iskoristiti jedan od metoda opisanih u poglavljima 8.1 odnosno 9.1).

Oko pokretnog objekta se može nacrtati okvir komandom `\fbox`. Ova komanda je identična komandi `\mbox`, osim što crta okvir oko sadržaja koji se nalazi u boksu. Ukoliko se želi da se i naslov kao i sam objekat nalaze u okviru, onda je najbolje staviti ih u jednu mini stranu, a tu mini stranu opet unutar komande `\fbox`. Razlog za ovo je što \LaTeX radi u različitim režimima sa pokretnim objektom odnosno naslovom. Tako bi, na primer, okvir oko slike (i naslova) u prethodnom primeru mogao da se doda na sledeći način:

```
\begin{figure}
\centering
\fbox{\begin{minipage}{4in}
\centering
\includegraphics[width=\textwidth]{fig7.eps}
\caption{Primer slike}
\label{slike:primer}
\end{minipage} }
\end{figure}
```

Komanda `\includegraphics` kojom se vrši uključivanje slika, odnosno okruženje `tabular` koje služi za opisivanje tabele, ne podrazumevaju početak novog pasusa. Na taj način, moguće je tretirati više slika odnosno tabela poređanih horizontalno kao jedan pokretni objekat. Horizontalno poravnanje se može podešavati komandom `\hspace`. Ukoliko se želi podešavati vertikalno poravnanje onda svaku komandu `\includegraphics` odnosno okruženje `tabular` treba staviti u posebnu mini stranu. Ako se pored njih u svaku mini stranu stavi i komanda `\caption`, onda se dobija više odvojenih pokretnih objekata poređanih jedan pored drugog (ali još uvek unutar jednog pokretnog objekta).

Primer [B.11](#) (strana 116)

Glava 7

Referisanja u L^AT_EX-u

U dužim tekstovima često su potrebna referisanja (ukazivanja) na određene delove teksta. L^AT_EX ima odličnu podršku za referisanja na delove teksta i objekte, kao i za kreiranje indeksa i bibliografije. Pri tome, L^AT_EX automatski održava brojeve referenci i na taj način znatno olakšava postupak referisanja.

7.1 Referisanja na delove teksta i objekte

Za obeležavanje mesta u dokumentu na koje se može referisati služi komanda `\label`. Ova komanda je oblika:

```
\label{oznaka}
```

pri čemu je *oznaka* proizvoljna niska karaktera kojom će dato mesto u tekstu biti referisano. Uobičajena je praksa, radi lakšeg snalaženja sa referencama, da se ova niska sastoji od dve reči razdvojene dvotačkom. Pri tome, prva reč obično označava tip objekta koji će biti referisan, a druga reč predstavlja jedinstveno ime za taj objekat. Tako bi, na primer, neke oznake u ovom preporučenom formatu mogle biti `poglavlje:uvod`, `teorema:pitagorina` ili `tabela:temperature`.

Komanda `\label` može da označi bilo koju jedinicu teksta koja ima automatski pridružen redni broj. Ova komanda se uvek odnosi na poslednju takvu jedinicu teksta koja se javila pre `\label` komande. Zato je ovu komandu preporučeno navoditi na samom početku jedinice teksta na koju se komanda odnosi, na primer odmah iza `\section` ili `\subsection` komande. Reference na pokretne objekte (na primer slike i tabele) se postavljaju na isti način, zadavanjem komande `\label` unutar odgovarajućeg okruženja. Obavezno je, međutim, da se komanda `\label` navede iza komande `\caption`.

Označeni deo dokumenta može se referisati komandama `\ref` ili `\pageref`. Sintaksa ovih komandi je `\ref{oznaka}`, odnosno `\pageref{oznaka}`, gde je *oznaka* niska karaktera kojom je u odgovarajućoj `\label` komandi označen deo

dokumenta koji se referiše. Komanda `\ref` uključuje broj poglavlja, potpoglavlja, slike, tabele ili teoreme u okviru koje stoji odgovarajuća `\label` komanda. Komanda `\pageref` uključuje broj strane na kojoj stoji odgovarajuća `\label` komanda.

Tako bi se, na primer, reference na ovo poglavlje u tekstu mogle napraviti na sledeći način:

```
\label{poglavlje:referisanja}
Videti poglavlje \ref{poglavlje:referisanja} na strani
\pageref{poglavlje:referisanja}.
```

i rezultat bi bio:

```
[
  Videti poglavlje 7.1 na strani 69.
]
```

Slično kao što je rečeno u vezi sa komandom `\tableofcontents` (koja, zajedno sa drugim sličnim komandama za generisanje sadržaja, kao što su `\listoftables` ili `\listoffigures`, takođe predstavlja vid referisanja), kada se koriste komande `\label` i `\ref` dokument se mora dvaput procesirati \LaTeX -om. U prvom prolazu određuje se na koje delove teksta se odnose pojedinačne `\label` komande i ove informacije se upisuju u datoteku sa ekstenzijom `aux`, da bi se u drugom prolazu odgovarajuće oznake upisale u dokument na mestima pojavljivanja `\ref` komandi.

7.2 Fusnote

Fusnote se navode komandom `\footnote` oblika:

```
\footnote[broj]{tekst}
```

Argument `tekst` predstavlja tekst koji će biti ispisan u fusnoti. Opcioni argument `broj` omogućava da se neposredno upiše broj fusnote koji bi inače \LaTeX dodelio. Ovu komandu uvek treba stavljati neposredno iza reči (sintagme, rečenice) na koju se fusnota odnosi. \LaTeX ispisuje svaku fusnotu na dnu strane na kojoj se javlja odgovarajuće referisanje. Fusnote su od običnog teksta odvojene jednom horizontalnom linijom.

Na primer, sledeća \LaTeX sekvenca:

```
Latinično pismo, koje je danas najraspostranjenije u svetu,
\footnote{Ovo pismo koristi oko 80% stanovnika naše planete.}
razvili su Feničani.
```

bi rezultovala uključenom fusnotom (koja se može videti na dnu strane):

```
[
  Latinično pismo, koje je danas najraspostranjenije u svetu,1 razvili su Feničani.
]
```

¹Ovo pismo koristi oko 80% stanovnika naše planete.

[]

Komanda `\thanks`, koja predstavlja specijalnu vrstu fusnota, opisana je u poglavlju 2.2.

7.3 Kreiranje indeksa

Postojanje kvalitetnog indeksa znatno uvećava upotrebljivost svakog dokumenta. \LaTeX , zajedno sa pomoćnim programom `makeindex`, omogućava jednostavno kreiranje i održavanje indeksa. Podrška za indekse u \LaTeX -u je implementirana u vidu paketa `makeidx` i njega je potrebno uključiti u preambulu komandom `\usepackage`. Takođe, u preambulu treba staviti i `\makeindex` komandu kojom se \LaTeX -u stavlja da znanja da prilikom procesiranja teksta treba da generiše informacije neophodne za indeksiranje.

Pojave termina, odnosno mesta u tekstu za koje se želi da se nađu u indeksu označavaju se `\index` komandom oblika:

```
\index{ključ}
```

pri čemu se `ključ` odnosi na stavku koja će se pojaviti u indeksu. Označene stavke će biti izlistane u rastućem alfabetskom redosledu sa pripadajućim brojem strane. Postoji nekoliko varijanti za zadavanje ključa:

- Ako se kao ključ zada običan termin, onda se taj termin i stavlja u indeks.
- Ako se kao ključ navedu dva termina razdvojena karakterom `!`, onda se u indeks stavlja obe stavke, ali tako da je drugi termin podstavka prvog. Tako, na primer, ako se postavljanje stavke u indeksu zada u tekstu komandom `\index{motori!mlazni}`, onda će stavka „mlazni“ biti upisana u indeks kao podstavka stavke „motori“.
- Ako se kao ključ navedu dva termina razdvojena `@` karakterom, prvi termin određuje gde će u indeksu (u smislu sortiranja) biti postavljena odgovarajuća stavka, a šta će tamo biti ispisano određeno je drugim terminom. Drugi termin može sadržati i \LaTeX komande za formatiranje teksta, što je naročito zgodno za formule. Tako bi se, na primer, komandom `\index{latex@LaTeX{}}` formirala stavka u indeksu koja bi bila sortirana prema terminu „latex“, ali koja bi bila ispisana kao „ \LaTeX “.

Moguće je i kombinovanje `!` i `@` karaktera u komandi `\index`. U svakoj od varijanti biće automatski generisan spisak termina (na osnovu ključa) praćenih brojevima strana na kojima se nalaze odgovarajuće `\index` komande.

Za ispis formatiranog indeksa na određenom mestu u tekstu služi komanda `\printindex`. Kao naslov dela teksta koji sadrži indeks ispisuje se reč *Index*. Ovaj naslov može se zameniti na jedan od načina opisanih u poglavljima 8.1 odnosno 9.1.

Kada \LaTeX procesira dokument koji uključuje `makeidx` paket i `\makeindex` komandu u preambuli, svaka pojava `\index` komande u tekstu dovodi do upisivanja odgovarajuće stavke praćene brojem strane u datoteci koja ima isto ime kao datoteka koju \LaTeX procesira i ekstenziju `idx`. Ovu datoteku zatim treba procesirati programom koji se u većini \LaTeX distribucija zove `makeindex` i koji se pokreće sa:

```
makeindex datoteka.idx
```

Program `makeindex` sortira stavke iz `idx` datoteke u novu datoteku sa ekstenzijom `ind`. Kada se dokument ponovo procesira \LaTeX -om, sortirani indeks se uključuje u dokument na mestu gde je u njemu zadata komanda `\printindex`. Dakle, radi generisanja indeksa, potrebno je dokument dva puta procesirati \LaTeX -om i između tih procesiranja pokrenuti program `makeindex`.

Primer [B.12](#) (strana [118](#))

7.4 Kreiranje bibliografije

\LaTeX formatira spisak korišćenih bibliografskih jedinica i omogućava olakšano referisanje u tekstu pomoću imena (takozvanih ključeva), koji se definišu za svaku bibliografsku jedinicu. Bibliografija se u \LaTeX -u može kreirati korišćenjem okruženja `thebibliography`. Komanda kojom se započinje ovo okruženje je:

```
\begin{thebibliography}{reč}
```

pri čemu `reč` određuje maksimalnu širinu i oblik oznake u bibliografiji. Tako, na primer, ako se koristi podrazumevani način označavanja bibliografskih jedinica rednim brojevima i ako se zna da će ukupan broj bibliografskih jedinica biti manji od 100, onda se za `reč` može staviti `99`. Ako se oznake bibliografskih jedinica neposredno označavaju, na primer tako da se sastoje od dva slova i dva broja, onda se za `reč` može staviti, na primer, `NN99` (ovde je `N` odabrano kao relativno široko slovo). Okruženje `thebibliography` se završava odgovarajućom `\end` komandom oblika `\end{thebibliography}`.

Svaka stavka u bibliografiji se navodi komandom `\bibitem` oblika:

```
\bibitem[oznaka]{ključ}
```

Opcioni argument `oznaka` predstavlja nisku karaktera koja će stajati kao oznaka bibliografske jedinice, na mestima gde je citirana u tekstu. Ova niska ne bi trebalo da bude šira od reči navedene u komandi kojom je započeto `thebibliography` okruženje. Obavezni argument `ključ` je proizvoljna niska kojom se u tekstumuže referisati na odgovarajuću bibliografsku jedinicu. Iza `\bibitem` komande kao običan tekst piše se opis bibliografske jedinice.

\LaTeX automatski u bibliografiji ispisuje svaku bibliografsku jedinicu, zajedno sa odgovarajućom oznakom u uglastim zagradama. Kao naslov dela teksta u kome se navodi bibliografija, \LaTeX ispisuje reč *Bibliography*. Naslov se može

promeniti jednim od metoda opisanim u poglavljima 8.1 odnosno 9.1. Bibliografija se ispisuje na onom mestu u tekstu na kome se nalazi `thebibliography` okruženje.

Radi citiranja jedne ili više bibliografskih jedinica, u tekstu se koristi komanda `\cite` oblika:

```
\cite[dodatak]{ključ1,ključ2,...}
```

\LaTeX zamenjuje ovu komandu oznakama odgovarajućih bibliografskih jedinica iz bibliografije. Oznake bibliografskih jedinica su predstavljene u uglastim zagradama, razdvojene zarezima (ukoliko ih ima više). Iza svih bibliografskih jedinica ispisuje se tekst koji je eventualno zadat opcionim argumentom `dodatak` komande `\cite`. Ovaj tekst se zarezom razdvaja od liste argumenata, pa se može upotrebiti, recimo, za navođenje konkretnog poglavlja ili strane iz date bibliografske jedinice.

Isto kao za obična referisanja, svaka izmena bibliografije zahteva da se dokument dva puta procesira \LaTeX -om pre nego što reference budu potpuno generisane i uključene u tekst.

Primer [B.13](#) (strana [120](#))

Za sistematičniji rad sa bibliografijama treba koristiti program `BIBTEX`. Ovaj program omogućava da se bibliografija generiše konsultovanjem jedne ili više datoteka sa podacima o bibliografskim jedinicama. Na osnovu ovako dobijenih podataka, \LaTeX i `BIBTEX` automatski generišu i sortiraju liste bibliografskih jedinica na koje se poziva u nekom dokumentu. Umesto u svakom \LaTeX dokumentu ponaosob, bibliografija se može čuvati u jednoj ili više namenskih bibliografskih datoteka. Izdvajanje podataka o bibliografskim jedinicama u zasebnu datoteku (ili više datoteka) omogućava da se lakše održavaju i ažuriraju ti podaci i izbegava se njihovo višestruko pojavljivanje u dokumentima što je neizbežno ako se radi sa okruženjem `thebibliography`. Usvojena ekstenzija za datoteke sa podacima o bibliografskim jedinicama je `bib`. Bibliografske jedinice se u ovim datotekama navode u formatu:

```
@vrsta (ključ,  
polje = vrednost,  
polje = vrednost,  
...  
polje = vrednost)
```

Karakter `@` označava da počinje opis nove bibliografske jedinice. Parametrom `vrsta` označen je tip bibliografske jedinice koji može biti neki od navedenih u tabeli [7.1](#). Parametrom `ključ` označena je niska kojom se u tekstu referiše bibliografska jedinica. Niz stavki oblika `polje = vrednost` služe za opis bibliografske jedinice. Vrste polja su navedene u tabeli [7.2](#). Vrednost svakog polja treba pisati pod navodnicima.

Za svaku vrstu bibliografskih jedinica propisano je koja su polja obavezna, a koja opciona. Tako, na primer, za bibliografsku jedinicu `BOOK` obavezna su polja `author` ili `editor`, zatim `title`, `publisher` i `year`, dok su za jedinicu `ARTICLE`

| vrsta | značenje |
|---------------|-----------------------------|
| ARTICLE | rad u časopisu |
| BOOK | knjiga |
| INPROCEEDINGS | rad u zborniku konferencije |
| MANUAL | tehnička dokumentacija |
| MASTERTHESIS | magistarska teza |
| PHDTHESIS | doktorska disertacija |
| UNPUBLISHED | neobjavljeni rad |

Tabela 7.1: Vrste bibliografskih jedinica koje podržava $\text{BIB}\text{T}_{\text{E}}\text{X}$

| polje | značenje |
|--------------|---|
| address | adresa izdavača |
| author | ime autora |
| editor | ime urednika |
| journal | ime časopisa |
| number | broj ili oznaka časopisa |
| organization | organizator konferencije |
| pages | strane časopisa na kojima se nalazi rad |
| publisher | ime izdavača |
| school | ustanova na kojoj je odbranjena teza |
| title | naslov rada |
| volume | tom časopisa |
| year | godina izdavanja |

Tabela 7.2: Vrste $\text{BIB}\text{T}_{\text{E}}\text{X}$ polja

obavezna polja `author`, `title`, `journal` i `year`. $\text{BIB}\text{T}_{\text{E}}\text{X}$ pri ispisu bibliografije uzima u obzir sadržaj obaveznih i onih opcionih polja koja su navedena. Polja sa nepoznatim imenom se ignorišu, što se može iskoristiti da se u datoteci čuvaju i dodatni podaci o bibliografskoj jedinici (na primer, *ISBN* broj za knjige).

Jedan primer zapisa u datoteci sa podacima o bibliografskim jedinicama bi bio:

```
@BOOK (knut86,
author = "Donald Knuth",
title = "The \TeX{}book",
publisher = "Addison-Wesley",
year = "1986",
isbn = "0-201-13447-0")
```

Zapis se odnosi na Knutovu knjigu *TEXbook* i može se videti da sadrži sva obavezna polja za bibliografsku jedinicu `BOOK`, ali i polje `isbn` koje će $\text{BIB}\text{T}_{\text{E}}\text{X}$ program ignorisati.

Program $\text{BIB}\text{T}_{\text{E}}\text{X}$ pri kreiranju bibliografije vrši određene zamene u tekstu koji je naveden u datoteci sa podacima o bibliografskim jedinicama. Tako, na

primer, ako rad ima više autora između svaka dva imena treba staviti veznik `and`. `BIBTEX` će u tom slučaju pretposlednje i poslednje ime razdvojiti veznikom *and*, a ostala imena zarezima. Ako se u polju `author` navede samo ime prvog autora i `and others`, onda će `BIBTEX` u bibliografiju upisati uobičajenu skraćenicu *et al.* nakon imena prvog autora. Da bi se izmenilo `BIBTEX` formatiranje, potrebno je promeniti datoteku u kojoj je definisan odgovarajući `BIBTEX` stil. Program `makebst` omogućava donekle automatizovanu izmenu datoteka sa stilovima u slučaju kada se bibliografija u potpunosti piše na jeziku koji nije engleski. Procedura može da bude dosta komplikovana, tako da je dodatne mogućnosti `BIBTEX`-a najbolje koristiti samo u slučajevima kada su sve reference na engleskom jeziku.

Ukoliko se bibliografije generišu programom `BIBTEX`, tada umesto korišćenja okruženja `thebibliography` na mestu na kome treba da bude ispisana bibliografija treba koristiti komande `\bibliographystyle` (za specifikovanje načina formatiranja referenci) i `\bibliography` (za navođenje imena jedne ili više datoteka sa podacima o bibliografskim jedinicama).

Komanda `\bibliographystyle` ima oblik:

```
\bibliographystyle{stil}
```

gde `stil` označava način na koji će biti označavane bibliografske jedinice u tekstu. Na raspolaganju je nekoliko standardnih stilova:

- `plain` stil obezbeđuje numeričko označavanje bibliografskih jedinica koje se u bibliografiji sortiraju prema alfabetskom redosledu,
- `unsrt` stil obezbeđuje numeričko označavanje, a bibliografske jedinice u bibliografiji su sortirane prema redosledu pojavljivanja u tekstu,
- `alpha` stil se odnosi na označavanje kombinacijom slova iz imena autora i godine publikovanja, dok se bibliografske jedinice u bibliografiji sortiraju alfabetski, kao kod `plain` stila.

Postoje i drugi, posebno definisani stilovi, koju su, kao i navedeni stilovi, opisani datotekama sa ekstenzijom `bst`.

Sintaksa komande `\bibliography` je:

```
\bibliography{lista}
```

pri čemu se `lista` odnosi na listu (zarezima razdvojenih) imena datoteka koje sadrže podatke o bibliografskim jedinicama. Komanda `\bibliography` umeće odgovarajući naslov i kreira listu referenci. Podrazumevani naslov se može promeniti na isti način kao kod okruženja `thebibliography`.

Radi ispravnog generisanja bibliografija i njenog uključivanja u `LATEX` dokumente potrebno je slediti određenu proceduru. Nakon što se dokument `ime.tex` procesira `LATEX`-om, treba pokrenuti program `bibtex` komandom:

```
bibtex ime
```

Na ovaj način će biti generisana datoteka `ime.bbl` koja sadrži bibliografiju u obliku koji može biti uključen u \LaTeX dokument. Tada treba tekst opet procesirati \LaTeX -om, pri čemu će bibliografija biti unešena u dokument. Na kraju treba tekst još jednom procesirati \LaTeX -om, da bi se reference u tekstu pravilno generisale.

Primer [B.14](#) (strana [122](#))

Glava 8

Proširivanje L^AT_EX-a

L^AT_EX predstavlja proširiv i prilagodljiv sistem i u ovom poglavlju će biti reči o dodavanju novih komandi, okruženja i paketa. Ove mogućnosti L^AT_EX-a pomažu kod skraćivanja zapisa često korišćenih komandi, definisanja novih i specijalnih klasa i paketa koji odgovaraju korisniku, prilagođavanja postojećih klasa i paketa i slično. Definicije novih komandi odnosno okruženja treba, radi preglednosti i lakšeg održavanja, navoditi u preambuli dokumenta.

8.1 Kreiranje novih komandi

Filozofija L^AT_EX-a podrazumeva da se objekti (koji će eventualno biti istaknuto prikazani) obeleže svojom logičkom grupom, a da se onda jednoobrazno definiše željeni grafički izgled za tu grupu. Drugim rečima, preporučeni pristup u L^AT_EX-u je da se tekst koji označava naslov poglavlja, tekst teoreme, zapis vektora, kod nekog programa i slično označe kao takvi, umesto da se neposredno svaki put definiše grafički izgled takvog teksta. Navedimo primer pisanja teksta teorema, pri čemu želimo da on bude ispisan kurzivnim pismom. Najjednostavniji način je da zadajemo komandu za upotrebu kurzivnog pisma `\textit` svaki put kad navodimo tekst teoreme. Međutim, ako je kasnije potrebno da se ispis promeni u obično pismo i da tekst teoreme započne niskom **Teorema** za kojom sledi redni broj teoreme, onda bismo morali svaku od teorema ručno da prepravimo. Filozofija L^AT_EX-a preporučuje drugačiji pristup: označiti tekst teorema tokom pisanja, a grafički izgled za sve teoreme definisati zasebno. Na taj način ne samo da je jednostavno promeniti grafički izgled pridružen odgovarajućoj klasi, nego je i tekst logički organizovan. Iz tako obeleženog teksta moguće je vrlo lako izdvojiti tekstove svih teorema (ili definicija, slika, tabela i slično), dok je iz teksta u kome su objekti samo grafički naglašeni (na primer kurzivnom pismom) to vrlo teško učiniti (budući da i drugi delovi teksta mogu biti ispisani kurzivom). Za podršku ovakvom načinu rada služi L^AT_EX mehanizam kreiranja novih komandi.

Komandom `\newcommand` uvodi se nova L^AT_EX komanda. Ova komanda ima

oblik:

```
\newcommand{ime}[broj_argumenata]{definicija}
```

Obavezni parametar `ime` predstavlja naziv nove komande (koji treba da počinje obrnutom kosom crtom), `broj_argumenata` je opcioni parametar i odnosi se na broj argumenata koje prima komanda, dok `definicija` određuje značenje nove komande i može sadržati bilo kakav tekst i/ili L^AT_EX komande. Ukoliko se opcioni argument izostavi, onda se podrazumeva da nova komanda nema argumenata.

U najjednostavnijem slučaju, nove komande se mogu koristiti kao skraćenice za neki deo teksta. Ako se, na primer, niska „Univerzitet u Beogradu“ često ponavlja u tekstu, onda je pogodno definisati komandu sa imenom recimo `\UBG` koja će zamenjivati ovaj tekst. To bi se moglo uraditi na sledeći način.

```
\newcommand{\UBG}{Univerzitet u Beogradu}
```

Po definisanju ove komande, svaka pojava komande `\UBG` u datom tekstu biće zamenjena niskom `Univerzitet u Beogradu`. Tako bi, na primer, rezultat procesiranja teksta:

```
\UBG{} je jedan od najvećih univerziteta u jugoistočnoj
Evropi. \UBG{} okuplja veliki broj instituta i fakulteta.
```

bio:

```
[
Univerzitet u Beogradu je jedan od najvećih univerziteta u jugoistočnoj
Evropi. Univerzitet u Beogradu okuplja veliki broj instituta i fakulteta.
]
```

Korišćenje argumenata omogućava kreiranje znatno upotrebljivijih i fleksibilnijih novih komandi. Argumenti se u definiciji nove komande referenciraju sa `#1`, `#2`, `#3` i tako redom.

Tako bi se, na primer, višelinijski komentari mogli uvesti i sledećom komandom:

```
\newcommand{\komentar}[1]{}
```

Ovde je definicija komande prazna, tako da se ono što se navede kao argument komande zapravo ne pojavljuje u generisanom dokumentu. Na taj način, postavljanjem dela teksta unutar komande `\komentar` može se postići da se taj deo teksta privremeno izbaci iz dokumenta.

Kao drugi primer, pretpostavimo da se u nekom tekstu često javlja potreba da se pojavljuju strane reči i da se želi, u zagradama, navesti njihov prevod na srpski u obliku „*table* (srp. sto)“. U cilju jednostavnijeg pisanja, mogla bi se definisati komanda oblika `\prevod` na sledeći način:

```
\newcommand{\prevod}[2]{\emph{#1} (srp. ~#2)}
```

Tada bi se `\emph{table}` (srp. *sto*) kraće zapisivalo `\prevod{table,sto}`. Primetimo da je, ukoliko kasnije promenimo stil ispisa (na primer, engleski oblik se ne piše kurzivnim pismom), dovoljno izmeniti samo definiciju komande, koja će se preneti na sva mesta na kojima je komanda korišćena.

Definisanje novih komandi naročito je pogodno za matematičke formule. Pretpostavimo da postoji potreba da se u nekom tekstu često koriste vektori oblika $[x_0 \ x_1 \ \dots \ x_{n-1}]$ ili $[y_0 \ y_1 \ \dots \ y_{m-1}]$. U tom slučaju može se definisati nova komanda `\vektor` sledećeg oblika:

```
\newcommand{\vektor}[2]{\[#1_{0}\ #1_{1}\ \ldots\ #1_{#2-1}]\}
```

Nakon unošenja gornje definicije, pomenuti vektori se mogu zapisati kratko `\vektor{x}{n}` odnosno `\vektor{y}{m}`.

Funkcionalnost identičnu L^AT_EX-ovoj komandi `\newcommand` nudi T_EX-ova komanda `\def`, pa se ona može koristiti umesto komande `\newcommand`. Komanda `\newcommand` je uvedena u L^AT_EX u skorije vreme, pa se u starijim dokumentima često sreće `\def` komanda.

L^AT_EX prihvata novu komandu definisanu komandom `\newcommand` samo ukoliko već ne postoji komanda sa istim imenom. Ako se ipak želi da nova definicija zameni staru, onda umesto komande `\newcommand` treba koristiti komandu `\renewcommand`, koja menja definiciju postojeće komande. Komanda `\renewcommand` je posebno pogodna za komande čija je svrha da čuvaju neke predefinisane vrednosti. Ova komanda se može iskoristiti i da se razni naslovi koje L^AT_EX automatski generiše promene tako da budu na željenom jeziku. Tako bi se, na primer, sledećim blokom komandi u preambuli dokumenta:

```
\renewcommand{\abstractname}{Apstrakt}
\renewcommand{\appendixname}{Dodatak}
\renewcommand{\bibname}{Literatura}
\renewcommand{\chaptername}{Glava}
\renewcommand{\contentsname}{Sadržaj}
\renewcommand{\enclname}{Prilozi}
\renewcommand{\figurename}{Slika}
\renewcommand{\indexname}{Indeks}
\renewcommand{\listfigurename}{Slike}
\renewcommand{\listtablename}{Tabele}
\renewcommand{\partname}{Deo}
\renewcommand{\prefacename}{Predgovor}
\renewcommand{\refname}{Literatura}
\renewcommand{\tablename}{Tabela}
```

postiglo da naslovi odgovarajućih elemenata teksta budu na srpskom jeziku umesto podrazumevanog engleskog.¹ Za sistematičnije pristupe rešavanju ovog problema videti poglavlje 9.1.

¹Treba voditi računa da nisu sve navedene komande definisane za svaki tip dokumenta, tako da za određene tipove dokumenata neke od gornjih komandi mogu generisati poruke o greškama.

8.2 Kreiranje novih okruženja

Na sličan način kao što se `\newcommand` komandom definišu nove \LaTeX komande, komandom `\newenvironment` mogu se definisati nova okruženja. Sintaksa ove komande je:

```
\newenvironment{ime}[broj_argumenata]{prolog}{epilog}
```

Obavezni parametar `ime` predstavlja ime novog okruženja, `broj_argumenata` je broj argumenata koji se prenosi novom okruženju, dok se `prolog` i `epilog` sastoje od komandi koje se procesiraju pre odnosno posle teksta navedenog unutar okruženja. Kao i kod komande `\newcommand`, zadavanje broj argumenata je opciono i ako taj broj nije naveden onda okruženje nema argumenata. Ako je broj argumenata naveden onda se, slično kao kod komande `\newcommand`, u `prolog`-u odnosno `epilog`-u argumenti mogu referisati sa `#1`, `#2`, `#3` i tako redom.

Ukoliko se, na primer, u nekom dokumentu na više mesta navodi tekst unutar `quote` okruženja i ukoliko se želi da se takav tekst dodatno istakne horizontalnom crtom na početku i na kraju, onda se umesto stalnog unošenja odgovarajućih komandi može definisati okruženje `naglasavanje` na sledeći način:

```
\newenvironment{naglasavanje}
{\par\rule{\textwidth}{1pt}\begin{quote}} % ovo je prolog
{\end{quote}\par\rule{\textwidth}{1pt}} % ovo je epilog
```

Komanda `\par` označava početak (ili kraj) pasusa. Komanda `\rule` služi za zadavanje horizontalnih odnosno vertikalnih linija (argumenti komande su dužina i debljina linije). Sada se naglašeni segmenti teksta mogu jednostavno navoditi kao:

```
\begin{naglasavanje}
Ispeci, pa reci.
\end{naglasavanje}
```

i biće postignut sledeći efekat:

Ispeci, pa reci.

8.3 Kreiranje novih paketa

Ukoliko se kreira veliki broj novih komandi, odnosno okruženja, koje se često koriste, pogodno je sve te definicije grupisati u poseban paket i izdvojiti u posebnu datoteku, a onda uključivati u dokument komandom `\usepackage`, kao i sve ostale pakete. Pored bolje organizacije dokumenta, izdvajanje definicija u poseban paket ima prednost i što se te definicije mogu lako koristiti i iz drugih

dokumenata, bez potrebe da se prepisuju u svaki novi dokument. Takođe se na ovaj način olakšava održavanje, a sa druge strane, ukoliko je to potrebno, neke od komandi ili okruženja se mogu ponovo definisati i u okviru dokumenta.

Na osnovu konvencije, datoteke koje sadrže pakete imaju ekstenziju `.sty`. Svaka ovakva datoteka treba da počne komandom `\ProvidesPackage` čija je sintaksa:

```
\ProvidesPackage{ime}
```

gde je `ime` ime paketa. Osim komande `\ProvidesPackage`, paket može da sadrži definicije komandi i okruženja. S obzirom na to da paketi u principu imaju znatno dužu upotrebnu vrednost od običnih dokumenata, preporučljivo je detaljno komentarisati njihov sadržaj.

Ukoliko bi se izdvojili komanda `\vektor` i okruženje `naglasavanje` pomenuti u prethodnim poglavljima u poseban paket, na primer, sa imenom `definicije`, onda bi sadržaj odgovarajuće datoteke `definicije.sty` bio:

```
% Paket definicije sadrži primer definicija.
\ProvidesPackage{definicije}

% Komanda \vektor služi za zapisivanje vektora.
\newcommand{\vektor}[2]{${\#1_{0}}\ #1_{1}\ \ldots\ #1_{#2-1}}$}

% Okruženje naglasavanje za ispis naglasenih delova teksta.
\newenvironment{naglasavanje}
{\par\rule{\textwidth}{1pt}\begin{quote}}
{\end{quote}\par\rule{\textwidth}{1pt}}
```

Definicije iz ovog paketa bi se potom učinile vidljivim u datom dokumentu tako što bi se u njegovu preambulu stavilo:

```
\usepackage{definicije}
```

Datoteke sa paketima se mogu smeštati u iste direktorijume sa dokumentima ili u odgovarajuće sistemske direktorijume.

Glava 9

Višejezička podrška

\TeX , a potom i \LaTeX , primarno su kreirani tako da slede američka tipografska pravila i pružaju podršku pisanju na engleskom jeziku. Vremenom je u izvesnoj meri dograđivana podrška i za pisanje na drugim jezicima. Danas se radi podrške upotrebi drugih jezika preporučuje korišćenje paketa `babel` koji će biti predstavljen u ovoj glavi. Takođe će biti razmotrene specifičnosti koje se odnose na pisanje na srpskom jeziku.

9.1 Paket za višejezičku podršku

Paket za višejezičku podršku `babel` se uključuje na uobičajeni način, komandom `\usepackage`. Kao opcije se mogu navesti oznake jednog ili više jezika koji će biti korišćeni u tekstu. Oznake za neke jezike su navedene u tabeli 9.1.

| oznaka | jezik |
|-----------------------|-------------|
| <code>serbian</code> | srpski |
| <code>english</code> | engleski |
| <code>croatian</code> | hrvatski |
| <code>french</code> | francuski |
| <code>german</code> | nemački |
| <code>greek</code> | grčki |
| <code>italian</code> | italijanski |
| <code>russian</code> | ruski |
| <code>spanish</code> | španski |

Tabela 9.1: Oznake za neke jezike u paketu `babel`

Tako bi se, na primer, podrška za srpski jezik uključila komandom:

```
\usepackage[serbian]{babel}
```

Pored navođenja u opcijama paketa `babel`, oznake za jezike se alternativno

moгу navesti i kao opcije klase dokumenta. U tom slučaju i ostali paketi mogu prepoznati ove opcije i prilagoditi se korišćenju datih jezika.

Jezička podrška koju donosi korišćenje paketa `babel` odnosi se pre svega na imena raznih delova teksta koje \LaTeX automatski generiše (na primer, predefinisana imena kojima se označavaju sadržaji, bibliografija, indeks i slično). Podrazumevane vrednosti ovih imena (koje su na engleskom jeziku) mogu se komandom `\renewcommand` zameniti odgovarajućim rečima jezika na kome se piše dokument na način opisan u poglavlju 8.1. Korišćenje paketa `babel` omogućava da se ovakve promene automatizuju i učine sistematičnim. Dalje, paket `babel` menja ispis datuma u slučajevima kada se taj ispis automatski generiše (kao na primer u komandi `\today` koja ispisuje tekući datum) tako da ovaj ispis bude na datom jeziku. Takođe se koriste pravila za hifenaciju (prelamanje reči na kraju reda) koja važe za dati jezik, ako su ona raspoloživa.

Paket `babel` obezbeđuje skroman skup komandi. Osnovna komanda je `\selectlanguage` i ima oblik:

```
\selectlanguage{jezik}
```

Opcija `jezik` se odnosi na oznaku jezika i to mora biti jedna od oznaka koja je navedena u preambuli dokumenta. Ovom komandom se označava da je tekst koji sledi napisan na datom jeziku i u skladu sa tim se primenjuju odgovarajuće konvencije. Ovo podešavanje je aktivno do pojave nove komande `\selectlanguage`, odnosno do kraja dokumenta. Okruženje `otherlanguage` funkcioniše na isti način kao komanda `\selectlanguage`, osim što se promena jezika odnosi samo na tekst unutar okruženja. Komanda kojom počinje ovo okruženje je oblika:

```
\begin{otherlanguage}{jezik}
```

gde `jezik` predstavlja oznaku jezika koji će biti korišćen u okruženju. Okruženje se završava komandom:

```
\end{otherlanguage}
```

Postoji i varijanta ovog okruženja `otherlanguage*`. Kod ove varijante se unutar okruženja automatski generisana imena ispisuju u jeziku koji važi izvan okruženja, dok se u svemu ostalom okruženje ponaša kao verzija bez zvezdice. Komanda `\language` predstavlja oznaku tekućeg jezika. Ova komanda se može iskoristiti na datom mestu u tekstu da se ta oznaka odštampa.

Dodatne komande, specifične za dati jezik, paket `babel` obično uvodi tako što redefiniše karakter " da bude specijalni karakter i onda te komande počinju ovim karakterom.

Podrška za razne jezike u paketu `babel` varira. Što se tiče srpskog jezika, ako se piše latinicom, paket `babel` podržava ispis svih automatski generisanih imena na srpskom jeziku, kao i ispis datuma. Definisane su i određene komande koje počinju karakterom ". Većina ovih komandi se odnosi na ispis naših slova sa dijakriticima ako se piše latiničnim pismom. Ono što nedostaje jeste podrška za srpsku hifenaciju; i pored određenih nastojanja da se obezbedi hifenacija za srpski, ona još uvek nije javno raspoloživa. Takođe, zbog postojanja padeža u srpskom jeziku, ispis datuma nije uvek korektan.

Osim paketa `babel`, postoji i dodatna podrška za rad sa određenim jezicima. Na primer, ispravnost zapisa pojedinih reči u \LaTeX dokumentima (takozvani *spell checking*) može se proveriti (posebnim) programom `ispell` [17].

Primer B.15 (strana 124)

9.2 Pisanje na srpskom jeziku

\LaTeX pruža solidnu podršku za pisanje na srpskom jeziku, pogotovu na latiničnom pismu. Postoji više načina na koji se u \LaTeX tekst mogu uneti naša slova. Već je pomenuto da \TeX fontovi sadrže naše karaktere i da se izborom odgovarajućeg kodiranja fonta (kodiranja fonta su opisana u poglavlju 4.4) ti karakteri mogu pravilno odštampati odnosno ispisati na ekranu.

Ukoliko se piše latiničnim pismom, jedan način za zapisivanje srpskih slova je da se koriste određeni simboli engleskog alfabeta kojima se dodaju odgovarajući dijakritici za srpska slova ć, č, š i ž. Ovakav način zapisivanja srpskih slova na latiničnom pismu je predstavljen tabelom 9.2. U skladu sa tim, sekvenca:

```
\'cilim, \v{c}i\v{c}ak, \v{s}al, \v{z}bun
```

bi dala rezultat:

```
[
  ćilim, čičak, šal, žbun
]
```

Na ovaj način nije moguće zapisati slovo đ, tako da se ono mora pisati kao `dj`.

| komanda | simbol |
|--------------------|--------|
| <code>\v{c}</code> | ć |
| <code>\v{C}</code> | Ć |
| <code>\v{s}</code> | š |
| <code>\v{S}</code> | Š |
| <code>\v{z}</code> | ž |
| <code>\v{Z}</code> | Ž |
| <code>\'c</code> | ć |
| <code>\'C</code> | Ć |

Tabela 9.2: Zapis srpskih latiničnih slova sa dijakriticima

Drugi način za zapisivanje srpskih latiničnih slova se sastoji od uključivanja paketa `babel` sa odgovarajućom opcijom za srpski jezik i potom korišćenja, u ovom paketu definisanih, komandi za zapisivanje srpskih slova koje počinju " karakterom. Tako se slova č, đ, š i ž mogu zapisivati komandama "c, "d, "s i "z. Ako je paket `babel` uključen na odgovarajući način, sekvenca:

```
"ci"cak, "dur"devak, "sal, "zbun
```

bi dala rezultat:

```
[
  čičak, đurđevak, šal, žbun
]
```

Na ovaj način nije moguće zapisati slovo ć, već se ono mora pisati na gore pomenuti način, kao `\'c`.

Oba pomenuta metoda podrazumevaju da tekst bude u takozvanom 7-bitnom kodiranju (kodiran sa 128 karaktera *ASCII* koda), odnosno da je tekst moguće uneti u bilo kakvom editoru. Ovo čini gornje metode dosta praktičnim, ali, sa druge strane, tekst u originalnom dokumentu ostaje nečitak i težak za održavanje. Stoga je znatno bolje rešenje da se tekst unosi u nekom 8-bitnom kodiranju, a da se potom iskoriste paketi `inputenc` i `fontenc` da bi se \LaTeX -u stavilo do znanja koje je kodiranje u pitanju, odnosno koje fontove da koristi.

Postoji veći broj definisanih 8-bitnih kodiranja za srpski jezik. Svako od tih kodiranja je nadskup *ASCII* kodiranja, pri čemu se kodovi za srpske karaktere nalaze u opsegu [128, 255]. Ono po čemu se kodiranja razlikuju jesu vrednosti kodova za pojedine karaktere.

Kao međunarodni standard *ISO 8859* [1] usvojena je familija kodiranja koja u pomenuti opseg smešta karaktere koji se javljaju u odgovarajućim grupama jezika. Za srpski jezik relevantna su kodiranja *ISO 8859-2* (poznato i pod imenom *Latin2*) koje propisuje kodove za latinične karaktere za srednje-evropske i istočno-evropske jezike, kao i *ISO 8859-5* koje propisuje kodove za ćirilične karaktere. Danas su široko rasprostranjeni editori koji omogućavaju unos teksta u nekom od *ISO 8859* kodiranja (na primer, `emacs` na UNIX-u). Na taj način, moguće je u \LaTeX dokument regularno uneti srpska slova. S obzirom na to da \LaTeX komande sadrže isključivo *ASCII* karaktere, izbor kodiranja nema nikakvog uticaja na njihov unos.

Kada se tekst unese na opisani način, potrebno je \LaTeX procesoru naznačiti kodiranje teksta. U tom cilju, treba uključiti paket `inputenc` na sledeći način:

```
\usepackage[kodiranje]{inputenc}
```

Ovde `kodiranje` predstavlja listu zareza razdvojenih oznaka kodiranja koja će biti korišćena u tekstu. U jednom tekstu se može koristiti više kodiranja, a jedino ograničenje je da se ne može menjati kodiranje unutar pasusa. Oznake za kodiranja *ISO 8859-2* odnosno *ISO 8859-5* su `latin2` odnosno `iso88595`.

Ukoliko se u tekstu koristi više kodiranja, onda se promena kodiranja naznačava komandom `\inputencoding`. Sintaksa ove komande je:

```
\inputencoding{kodiranje}
```

gde je `kodiranje` oznaka kodiranja za tekst koji sledi. Ovo kodiranje je aktivno do pojave nove komande `\inputencoding` ili do kraja dokumenta.

Osim *ISO 8859* familije kodiranja, paket `inputenc` podržava i brojna druga kodiranja. Kada se tekst unosi nekim WINDOWS editorom (na primer, editorom `WinEdit`), on može da bude snimljen u kodiranjima specifičnim za WINDOWS. Ova kodiranja se označavaju kao *CP1250* za latinične karaktere, odno-

sno *CP1251* za ćirilćne karaktere. U tom slućaju, dovoljno je kao opcije paketa `inputenc` navesti `cp1250`, odnosno `cp1251` i \LaTeX će prepoznati kodiranje. Ukoliko se tekst unosi nekim DOS editorom, onda su odgovarajuća kodiranja respektivno *CP852* i *CP855* za latinicu odnosno ćirilicu, a oznake kodiranja u `inputenc` paketu `cp852` odnosno `cp855`.

Pored tipa kodiranja ulaznog teksta, \LaTeX -u se mora naznaćiti i koji raspored karaktera (odnosno, koje kodiranje) da koristi za fontove. U tu svrhu se koristi paket `fontenc` o kojem je već bilo reći. Rasporedi koje treba navesti za srpska latinićna odnosno ćirilićna slova su `T1` odnosno `T2A`. Ukoliko se u dokumentu menja kodiranje ulaznog teksta, onda treba promeniti i raspored karaktera za font komandom `\fontencoding`, o kojoj je takođe ranije bilo reći.

Tipićeo, pri pisanju pod UNIX-om u preambulu dokumenta radi definisanja kodiranja teksta i kodiranja fonta treba navesti komande:

```
\usepackage[latin2]{inputenc}
\usepackage[T1]{fontenc}
```

ako se piše latinićnim pismom, odnosno komande:

```
\usepackage[iso88595]{inputenc}
\usepackage[T2A]{fontenc}
```

ako se piše ćirilićnim pismom. Dokument potom u editoru treba saćuvati u odgovarajućem 8-bitnim kodiranju i \LaTeX procesor će biti u stanju da ga pravilno procesira. Ekvivalentne komande za preambulu pri pisanju pod WINDOWS-om su:

```
\usepackage[cp1250]{inputenc}
\usepackage[T1]{fontenc}
```

za pisanje latinicom, odnosno:

```
\usepackage[cp1251]{inputenc}
\usepackage[T2A]{fontenc}
```

za pisanje ćirilicom.

Primer [B.16](#) (strana [126](#))

Sistem 8-bitnih kodiranja, mada dobro razraćen, ima odrećenih problema. Danas se kao novi standard u pogledu ove problematike sve više širi *Unicode* sistem kodiranja. U *Unicode*-u se karakteri kodiraju sa 16 bita tako da jedno kodiranje pokriva praktićeo sve karaktere koji su u upotrebi u svetu. Sa širenjem *Unicode* standarda, sve je više editora koji mogu da saćuvaju tekst u ovom kodiranju. U toku je rad na projektu (pod nazivom *Omega*) izmena \TeX -a i \LaTeX -a tako da mogu da prihvate i obrade *Unicode* tekst. U ovom trenutku mećuotim, ćitav sistem još nije upotrebljiv za pisanje na srpskom jeziku, tako da ovde neće biti detaljnije razmatran.

Pored standardnih kodiranja, za pisanje ćirilicom kod nas je vrlo rašireno rešenje da se tekst piše latinicom, a da se koristi `OT2` kodiranje fonta kod kojeg

su fontovi za većinu ćiriličnih slova postavljeni na ulazne kodove odgovarajućih latiničnih slova. Familija `wncyr` (takozvana Vašingtonska ćirilica) je veoma kvalitetna familija fontova u OT2 kodiranju. Prednosti ovakvog načina pisanja ćirilicom su što je tekst prenosiv (koristi se 7-bitno kodiranje), što je prebacivanje iz latinice u ćirilicu i obrnuto relativno jednostavno, kao i što je tekst predstavljen nezavisno od pisma. Sa druge strane, OT2 fontovi prikazuju određen broj naših slova pomoću ligatura, prema tabeli 9.3.

| ligatura | rezultujući simbol | primer zapisa | izgled primera |
|----------|--------------------|---------------|----------------|
| dj | ђ | djurdjevak | Ђурђевак |
| Dj | Ђ | Djordje | Ђорђе |
| zh | ж | mozhda | можда |
| Zh | Ж | Zhabljak | Жабљак |
| lj | љ | ljubav | љубав |
| Lj | Љ | Ljubljana | Љубљана |
| nj | њ | manji | мањи |
| Nj | Њ | Njegosh | Његош |
| ci | ћ | ciilim | ћилим |
| ci | Ћ | Ciiclevac | Ћићевац |
| ch | ч | machak | мачак |
| Ch | Ч | Chachak | Чачак |
| d2 | џ | d2ak | џак |
| D2 | Џ | D2ordan | Џордан |
| sh | ш | mashna | машна |
| Sh | Ш | Shabac | Шабац |

Tabela 9.3: Ligature u OT2 kodiranju fonta

Ove ligature nisu prisutne u latiničnim kodiranjima fontova, tako da pri konverziji iz ćirilice u latinicu ili obrnuto ipak treba izvršiti odgovarajuće supstitucije. Alternativno, mogu se u preambuli definisati komande za nabrojana slova koje će se u zavisnosti od toga da li tekst treba da bude prikazan ćiriličnim ili latiničnim fontom preslikati u odgovarajuće simbole.

Primer B.17 (strana 128)

Svi pomenuti načini pisanja na srpskom jeziku, kako latiničnim (pomoću dijakritika, pomoću komandi paketa `babel` ili izborom odgovarajućeg 8-bitnog kodiranja), tako i ćiriličnim pismom (pomoću ligatura ili izborom bilo kog od raspoloživih 8-bitnih kodiranja) su jednako valjani. Izbor treba uvek praviti u skladu sa specifičnim potrebama datog korisnika i očekivanim načinom distribucije dokumenta.

Primer B.18 (strana 130)

Glava 10

Kreiranje slajdova i prezentacija

\LaTeX se može koristiti i za kreiranje atraktivnih slajdova i prezentacija. U ovoj glavi biće razmotreni različiti načini na koje se može obaviti ovaj zadatak.

10.1 Osnovna podrška za kreiranje prezentacija

Osnovna podrška za kreiranje slajdova u \LaTeX -u postoji u vidu klase `slides`. Ova klasa podrazumeva korišćenje velikih, bez-serifnih slova čime se dobija dokument koji je pogodan za prikazivanje na projektoru. Pored ovoga, klasa `slides` ne pruža nikakvu dodatnu podršku za kreiranje prezentacija.

Primer [B.19](#) (strana [140](#))

10.2 Napredna sredstva za kreiranje prezentacija

Namenske komande koje olakšavaju rad sa slajdovima mogu se naći u dodatnim \LaTeX paketima. Postoji veći broj takvih paketa sa sličnim mogućnostima. Ovde će biti predstavljen paket `beamer` koji ima odličnu podršku za kreiranje dinamičkih prezentacija. Predviđeno je da se dokumenti kreirani uz pomoć ovog paketa prevode u PDF [\[15\]](#) format radi prikazivanja prezentacije na ekranu, pri čemu `beamer` nudi čitav niz efekata za postupni prikaz sadržaja, kao i za prelaz između dva slajda. Paket omogućava i generisanje slajdova za prikaz na projektoru, ili generisanje štampane verzije prezentacije. Ove i druge napredne mogućnosti paketa su detaljnije opisane u odgovarajućoj dokumentaciji.

Paket `beamer` sadrži definiciju istoimene klase dokumenata. \LaTeX dokumenti koji predstavljaju prezentacije kreirane korišćenjem ovog paketa treba da počnu sa:

```
\documentclass[opcije]{beamer}
```

Stil prezentacije je određen tzv. temom, koja se u preambuli dokumenta zadaje komandom:

```
\usetheme{tema}
```

Tema određuje boju pozadine i teksta, fontove kojima će biti ispisani naslovi ili običan tekst, grafiku koja će biti prikazana na svakom slajdu i tako dalje. Na raspolaganju je veliki broj podrazumevanih tema, koje su nazvane po gradovima, na primer *Antibes*, *Berlin*, *Copenhagen*, *Frankfurt*, *Madrid*, *Szeged* i *Warsaw*. Navedene teme u potpunosti određuju stil prezentacije, a pojedinačni aspekti prezentacije se mogu kontrolisati tzv. podtemama, koje se mogu svrstati u četiri kategorije navedene u tabeli 10.1. Još finija kontrola nad pojedinim aspektima prezentacije ostvaruje se komandama `\setbeamertheme`, `\setbeamerfont` i `\setbeamercolor`. Tako se na primer ikonice za navigaciju kroz prezentaciju, koje bivaju automatski generisane u svakoj *beamer* prezentaciji, eliminišu komandom:

```
\setbeamertheme{navigation symbols}{}
```

| komanda | značenje |
|--------------------------------------|---|
| <code>\useoutertheme{podtema}</code> | kontrolise dekoracije na slajdovima, kao na primer tekst i grafiku u zaglavlju slajda |
| <code>\useinnertheme{podtema}</code> | kontrolise izgled glavnog dela na slajdovima, kao na primer markere u listama, ili formatiranje teorema |
| <code>\usefonttheme{podtema}</code> | kontrolise fontove na slajdovima |
| <code>\usecolortheme{podtema}</code> | kontrolise boje na slajdovima |

Tabela 10.1: Vrste podtema u *beamer*-u

Klasa *beamer* redefiniše neke standardne \LaTeX komande. Komande koje se mogu navesti u preambuli dokumenta date su u tabeli 10.2. Komanda `\titlepage` na osnovu vrednosti zadatih u preambuli dokumenta kreira naslov prezentacije unutar datog slajda.

| komanda | značenje |
|------------------------|--------------------------------------|
| <code>title</code> | naslov prezentacije |
| <code>subtitle</code> | podnaslov prezentacije |
| <code>author</code> | autor (odnosno autori) prezentacije |
| <code>institute</code> | ime institucije sa koje dolazi autor |
| <code>date</code> | datum |

Tabela 10.2: Komande u preambuli prezentacije

Pojedinačni slajdovi u dokumentu se navode unutar okruženja `frame`. Ovo okruženje počinje komandom:

```
\begin{frame}{naslov}
```

a završava se komandom:

```
\end{frame}
```

Argument `naslov` u komandi kojom počinje okruženje predstavlja nisku koja će biti ispisana kao naslov slajda.

Različiti efekti prelaza sa jednog slajda na drugi mogu se postići stavljanjem odgovarajućih komandi unutar `frame` okruženja. Neke od tih komandi su `\transdissolve` (tekući slajd se preliva u naredni slajd), `\transwipe` (linija „briše“ ekran otkrivajući naredni slajd) ili `\transboxout` (naredni slajd se pomalja preko tekućeg počev od centralnog dela slajda prema ivicama). Podrazumevani efekat je da naredni slajd neposredno zamenjuje tekući slajd. Trajanje efekta se može precizirati `\transduration` komandom unutar `frame` okruženja.

Unutar okruženja `frame` mogu se koristiti sve \LaTeX komande za rad sa tekstom. Na slajdovima se često koristi okruženje `itemize` (videti poglavlje 3.6). Izgled takozvanih *bullet*-a, koji označavaju stavke liste na različitim nivoima hijerarhije, može se podešavati pomenutim komandama `\setbeamertemplate`, `\setbeamerfont` i `\setbeamercolor`.

Primer B.20 (strana 143)

Moguće je koristiti i animirane efekte za postupni prikaz sadržaja slajda. Ovakav efekat je najjednostavnije postići umetanjem komande `\pause` na jednom ili više mesta unutar `frame` okruženja. Sadržaj slajda tada tokom prezentacije biva prikazan inkrementalno, i to prvo samo deo slajda do mesta gde je umetnuta prva komanda `\pause`, zatim se komandom za prelazak na naredni slajd u prezentaciji prikazuje i deo slajda do mesta gde je umetnuta naredna komanda `\pause`, i tako dalje.

Preciznija kontrola nad ovim efektom se može postići `\onslide<lista>` komandom. Slajd se i u ovom slučaju prikazuje inkrementalno, te komanda za prelazak na naredni slajd u prezentaciji ovde aktivira deo po deo slajda. Brojevi u listi koja se zadaje u `\onslide` komandi označavaju u kom segmentu prikaza slajda će tekst koji sledi komandu biti vidljiv. Brojevi u listi se razdvajaju zarezima, a niz brojeva je moguće kraće zapisati u obliku `m-n`, gde je `m` prvi broj u nizu, a `n` zadnji. Ako se u `\onslide` komandi lista izostavi, onda će tekst koji sledi biti vidljiv u svakom segmentu prikaza datog slajda. Ukoliko je na primer sadržaj slajda definisan na sledeći način:

```
\begin{frame}{Suglasnici i samoglasnici}
\onslide
Suglasnici su: \\
\onslide<1,2,3>b \\
\onslide<2,3>c \\
\onslide<3>d
\onslide
```

```
Samoglasnici su: \\
\onslide<1-3>a \\
\onslide<2-3>e \\
\onslide<3>i
\end{frame}
```

tada će u prvom koraku prikaza slajda biti vidljiva slova *b* i *a*, u drugom koraku će biti dodata slova *c* i *e*, a u trećem koraku i slova *d* i *i*, dok će tekst „Suglasnici/samoglasnici su:“ biti vidljiv celo vreme prikaza slajda. Paket `beamer` proširuje i neke standardne $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ komande tako da podržavaju isti način specifikacije koraka tokom prikaza slajda u kojima će ispis komande biti vidljiv. Posebno je pogodno koristiti tu mogućnost u vezi sa `\item` komandom.

Dodatak A

Korišćenje L^AT_EX-a

Pored poznavanja pravila za označavanje teksta, korišćenje L^AT_EX-a pretpostavlja i nabavku i instalaciju odgovarajuće distribucije, a zatim i poznavanje korišćenja dokumentacije, samog L^AT_EX procesora, kao i pratećih programa.

A.1 L^AT_EX distribucije

T_EX i L^AT_EX predstavljaju kompleksne sisteme koji se sastoje od većeg broja programa, pratećih paketa i dokumentacije. Stoga je najbolje T_EX i L^AT_EX nabaviti i instalirati u obliku distribucije, odnosno zaokruženog paketa koji obedinjuje sve što je potrebno za rad.

Na UNIX-u se T_EX sa L^AT_EX-om tradicionalno isporučuje uz sam operativni sistem; u pitanju je najčešće T_ET_EX distribucija [27].

WINDOWS korisnicima je na raspolaganju veći broj distribucija. Ranije, pod DOS-om, E_MT_EX distribucija [8] je bila vrlo popularna. Pod WINDOWS-om, najviše korišćena je bila M_IK_TE_X distribucija [24].

U poslednje vreme, T_EX LIVE distribucija se nameće kao odličan izbor. Ova distribucija uključuje sve ključne T_EX programe i pakete, kao i veliki broj fontova, a održava se vrlo kompletnom i ažurnom, kako za WINDOWS tako i za UNIX sisteme.

Centralna Internet lokacija na kojoj se mogu pronaći razni sadržaji vezani za T_EX odnosno L^AT_EX, a pre svega distribucije, je *Comprehensive T_EX Archive Network*, odnosno skraćeno *CTAN* [5].

Distribucije obično ne uključuju nikakav editor. Za pisanje L^AT_EX dokumenta može poslužiti bilo kakav editor, ali je dobro odlučiti se za editor koji u određenoj meri prepoznaje L^AT_EX sintaksu i ima određenu podršku za pisanje na L^AT_EX-u i procesiranje dokumenta direktno iz editora, a takođe i podršku za snimanje teksta u 8-bitnom kodiranju. U tom smislu se pod UNIX-om može preporučiti *emacs* [7] editor, a pod WINDOWS-om *WinEdit* [28] editor.

L^AT_EX nije *WYSIWYG* (*What You See Is What You Get*) sistem, odnosno tokom pisanja dokument nije vidljiv u finalnom obliku, već ga je prethodno

potrebno procesirati i onda koristiti odgovarajući program za pregled da bi se videlo kako dokument izgleda. U poslednje vreme postoje određeni napori da se kreiraju okruženja koja bi omogućavala da se tokom pisanja dokument vidi u finalnom obliku, a sa druge strane podržavala sve \LaTeX komande za formatiranje. U tom smislu treba pomenuti LYX [23] odnosno (za WINDOWS) SCIENTIFIC WORD editor [26], a vreme će pokazati da li će ovakav način rada odneti prevagu nad načinom koji je sada dominantan.

A.2 \LaTeX dokumentacija

Pri instaliranju svake \LaTeX distribucije instalira se i odgovarajuća dokumentacija za programe i pakete koji sačinjavaju distribuciju. Ova dokumentacija je obično u vidu \TeX i \LaTeX datoteka koji se mogu procesirati na uobičajen način da bi se odgovarajući dokumenti dobili u čitljivijem obliku. Konvencija je da ovakvi dokumenti imaju ekstenziju `dtx` umesto standardne ekstenzije `tex`.

Dokumentacija za \LaTeX procesor i ostale prateće formate je takođe raspoloživa. Na UNIX-u je ta dokumentacija obično u vidu *man* strana, a na WINDOWS-u u *WinHelp* formatu. Ukoliko lokalna dokumentacija ne pruži odgovor na ono što se traži, onda svakako treba pokušati da se taj odgovor nađe na CTAN-u.

A.3 Obrada \LaTeX datoteka

Kada se dati dokument pripremi u skladu sa pravilima \LaTeX -a, može se procesirati komandom:

```
latex ime.tex
```

gde je `ime.tex` datoteka u kojoj je dokument sačuvan. Ekstenzija `tex` je podrazumevana, tako da se alternativno može koristiti i kraća komanda:

```
latex ime
```

Tokom procesiranja, \LaTeX procesor proverava da li postoje sintaksne greške u dokumentu. Ako naiđe na neku takvu grešku, procesor ispisuje odgovarajuću poruku o grešci, zatim ispisuje prompt `?` i zaustavlja se, očekujući od korisnika da preduzme neku akciju. Korisnik može pritisnuti taster `ENTER` u cilju ignorisanja greške. \LaTeX tada pokušava sam da se oporavi od greške i — ako u tome uspe — nastavlja sa procesiranjem, a u suprotnom ponovo ispisuje istu poruku o grešci. Korisnik može da pritisne i taster `h` (od *help*) za pomoć, zatim taster `r` za nastavak procesiranja bez obaziranja na tekuću i eventualne naredne greške, te taster `x` za prekid procesiranja.

Pored poruka o greškama, \LaTeX tokom procesiranja dokumenta ispisuje i razna upozorenja. Upozorenja ne zaustavljaju procesiranje, ali ukazuju na određene probleme pri ovom postupku. Obično se upozorenja odnose na nemogućnost slaganja teksta u skladu sa pravilima o dobrom izgledu dokumenta. Najčešće poruke koje se javljaju u vidu upozorenja su `Overfull \hbox...`, odnosno

`Underfull \hbox...` Prva poruka se odnosi na slučaj kada \LaTeX ne može da pronađe odgovarajući prelom za dati red teksta i onda ostavi poslednju reč u redu da prelazi desnu marginu. Uzrok ovog problema je najčešće nemogućnost prelamanja reči, pa se problem može otkloniti eksplicitnim definisanjem odgovarajućeg prelamanja, na način prezentiran u poglavlju 3.2. Sa druge strane, ukoliko \LaTeX uspešno prelomi red, ali tako da je razmak između reči preveliki po standardima za dobar izgled dokumenta, \LaTeX prijavljuje drugu navedenu poruku. I u ovom slučaju eksplicitno prelamanje reči (ovaj put prve u narednom redu) može pomoći da tekst bude složen kako treba. Alternativno, problemi sa prelamanjem se mogu rešiti primenom komande `\sloppy` koja je pomenuta u poglavlju 3.2.

Nakon uspešnog procesiranja datog dokumenta \LaTeX -om, u radnom direktorijumu biva kreiran veći broj datoteka. Datoteka sa ekstenzijom `log` sadrži upozorenja i poruke o greškama koje su prilikom procesiranja bile ispisane na ekranu. Datoteka sa ekstenzijom `aux` sadrži informacije prikupljene tokom procesiranja koje su potrebne procesoru da bi pri narednom procesiranju ispravno generisao reference i bibliografiju. Ukoliko dokument ima tabelu sadržaja ili indekse, onda se prilikom procesiranja generišu i druge pomoćne datoteke koje takođe služe za ispravno generisanje ovih delova teksta pri narednom procesiranju. Ključni rezultat procesiranja je datoteka `ime.dvi`, koja sadrži dokument u DVI formatu. Dokument u ovom obliku se može pomoću odgovarajućeg programa prikazati na ekranu, a može se i konvertovati u format pogodan za štampanje.

Za prikazivanje DVI datoteka na ekranu postoji veći broj programa. U nekim slučajevima takvi programi se isporučuju uz \LaTeX distribucije, a nekad su u pitanju namenski programi. Na UNIX-u se u ovu svrhu može iskoristiti program `xdvi` koji predstavlja deo *X Windows*-a, ili program `kdvi` koji je deo *KDE* grafičkog okruženja. Pod WINDOWS-om se može koristiti odgovarajući program iz raspoložive distribucije ili namenski program `dviwin` [6].

\LaTeX dokumenti se mogu prevesti i u HTML format radi potreba prezentacije ili publikovanja na Internetu. U tu svrhu se može iskoristiti program `latex2html` [22]. S obzirom na to da HTML format ima skromnu podršku za predstavljanje formula, `latex2html` prevodi formule u slike.

A.4 Obrada DVI datoteka

DVI datotke se u cilju štampanja obično prevode u POSTSCRIPT format, što se radi `dvips` programom koji dolazi uz sve \LaTeX distribucije. Takođe, DVI datoteke mogu se prevesti u PDF format odgovarajućim programom `dvipdf` koji se isporučuje uz sve \LaTeX distribucije. PDF format služi prevashodno za prikaz dokumenta na ekranu, s tim što su programi za pregled PDF dokumenata obično znatno komforniji od programa za pregled DVI dokumenata.

Za određene štampače koji ne podržavaju POSTSCRIPT format niti imaju POSTSCRIPT emulaciju, postoji mogućnost prevođenja DVI datoteka direktno u kontrolni jezik štampača. Posebno su dobro podržani HP štampači, pa tako postoji čitav niz programa `dvihp`, `dvilj`, `dvilj2p` i tako dalje, koji prevode

DVI datoteke u PCL jezik ovih štampača.

A.5 Obrada POSTSCRIPT datoteka

Postoji veliki broj programa kojima se mogu izvršiti određene transformacije generisanih POSTSCRIPT datoteka. Ti programi nisu standardizovani u \LaTeX distribucijama, ali kako su neki od njih veoma korisni vredi ih ovde ukratko pomenuti:

- Program `psbook` preuređuje redosled strana prilagođavajući dokument za štampanje i spajanje sa po četiri strane na svakom listu papira.
- Program `psmerge` omogućava spajanje više POSTSCRIPT dokumenata u jedan.
- Program `psnup` omogućava smeštanje više strana ulaznog POSTSCRIPT dokumenta na jednu stranu izlaznog dokumenta.
- Program `psresize` omogućava promenu veličine papira za dati POSTSCRIPT dokument.
- Program `psselect` omogućava kreiranje novog dokumenta od određenog skupa strana datog dokumenta.

Ukoliko se od \LaTeX dokumenta generiše POSTSCRIPT datoteka, onda se u dokumentu mogu koristiti i POSTSCRIPT fontovi. \LaTeX distribucije obično sadrže odgovarajuće pakete čijim se uključivanjem u dokument postiže zamena standardnih \LaTeX fontova odgovarajućim POSTSCRIPT fontovima. Pritom treba voditi računa o tome da ovi paketi vrše zamenu samo pojedinih familija fontova u dokumentu, kao i da ne postoje prateći POSTSCRIPT matematički fontovi za sve familije običnih POSTSCRIPT fontova. Jedna dobra kombinacija paketa koja u potpunosti zamenjuje \LaTeX fontove u dokumentu (uključujući i matematičke fontove) POSTSCRIPT fontovima bi bila:

```
\usepackage{mathptmx}  
\usepackage{helvet}  
\usepackage{courier}
```

Dodatak B

Primeri

Primer 1

```
% ----- datoteka: primer.tex

% ----- tip dokumenta je 'article' (npr. za radove, članke)
\documentclass{article}

% ----- pomoćni paket 'inputenc' za latin-2 kodiranje
\usepackage[latin2]{inputenc}

% ----- početak teksta
\begin{document}
% ----- Poglavlje 1
\section{\TeX{} i \LaTeX{}}

\TeX{} je sistem za pripremu teksta za štampu koji je kasnih
sedamdesetih godina prošlog veka kreirao Donald Knut, sa ciljem
da napravi {\em sistem za obradu teksta namenjen pisanju lepih
knjiga, pogotovo knjiga koje sadrže puno matematike}.

\begin{center}
\LaTeX{} je sistem za pripremu za štampu nastao na osnovama
\TeX-a. \ \ On definiše skup specifičnih \textbf{klasa
dokumenata} koje se koriste za formatiranje tekstova.
\end{center}

% ----- Poglavlje 2:
\section{Formatiranje teksta}

Pripremanjem teksta u \TeX-formatu potpuno precizno se
opisuje na koji će način on biti složen, pri čemu
su ti opisi najčešće savim prirodni i jednostavni.

\underline{Matematičke formule} se zapisuju izdvajanjem
pomoću simbola  $\$$  na sledeći način:


$$\forall x \in S_1: x^2 \geq \varepsilon$$


% ----- kraj teksta

\end{document}
```

1 \TeX i \LaTeX

\TeX je sistem za pripremu teksta za štampu koji je kasnih sedamdesetih godina prošlog veka kreirao Donald Knut, sa ciljem da napravi *sistem za obradu teksta namenjen pisanju lepih knjiga, pogotovo knjiga koje sadrže puno matematike*.

\LaTeX je sistem za pripremu za štampu nastao na osnovama \TeX -a. On definiše skup specifičnih **klasa dokumenata** koje se koriste za formatiranje tekstova.

2 Formatiranje teksta

Pripremanjem teksta u \TeX -formatu potpuno precizno se opisuje na koji će način on biti složen, pri čemu su ti opisi najčešće savim prirodni i jednostavni.

Matematičke formule se zapisuju izdvajanjem pomoću simbola $\text{\$}$ na sledeći način:

$$\forall x \in S_1 : x^2 \geq \varepsilon$$

Primer 2

```
\documentclass{article}
\usepackage[serbian]{babel}

\begin{document}

\section{Prvo poglavlje}

Tekst...

\subsection{Prvo potpoglavljje}

Tekst...

\subsection{Drugo potpoglavljje}

Tekst...

\section*{Nenumerisano poglavlje}

Tekst...

\subsection{Potpoglavljje nenumerisanog potpoglavljja}

Tekst...

\subsection*{Nenumerisano potpoglavljje}

Tekst...

\setcounter{section}{9}
\section{Poglavljje sa promenjenim rednim brojem}

Tekst...

\tableofcontents

\end{document}
```


1 Prvo poglavlje

Tekst...

1.1 Prvo potpoglavlje

Tekst...

1.2 Drugo potpoglavlje

Tekst...

Nenumerisano poglavlje

Tekst...

1.3 Potpoglavlje nenumerisanog potpoglavlja

Tekst...

Nenumerisano potpoglavlje

Tekst...

10 Poglavlje sa promenjenim rednim brojem

Tekst...

Sadržaj

| | |
|---|----------|
| 1 Prvo poglavlje | 1 |
| 1.1 Prvo potpoglavlje | 1 |
| 1.2 Drugo potpoglavlje | 1 |
| 1.3 Potpoglavlje nenumerisanog potpoglavlja | 1 |
| 10 Poglavlje sa promenjenim rednim brojem | 1 |

Primer 3

```
\documentclass{letter}
\usepackage[latin2]{inputenc}
\name{Petar Petrović}
\address{Matematički fakultet \\ Studentski trg 16 \\
         11000 Beograd}

\begin{document}

\begin{letter}{Prirodno-matematički fakultet \\
             Univerzitet u Kragujevcu}

\opening{Poštovane kolege,}

Kao što vam je poznato, \LaTeX\ podržava više standardnih
tipova dokumenata sa odgovarajućom logičkom strukturom, koja
može da podrazumeva organizaciju dokumenata u glave,
poglavlja, pasuse i sl. Precizna struktura i odgovarajući
način formatiranja dokumenata opisan je tzv. ~{\em klasama}.
Klasa definiše opštu strukturu i izgled teksta, njegove
elemente i osobine.

\closing{Srdačno,}

\cc{Univerzitet u Nišu \\ Univerzitet Crne Gore}
\encl{Spisak predloga \\ Lista elektronskih adresa}
\end{letter}

\end{document}
```

Matematički fakultet
Studentski trg 16
11000 Beograd

November 29, 2013

Prirodno-matematički fakultet
Univerzitet u Kragujevcu

Poštovane kolege,

Kao što vam je poznato, \LaTeX podržava više standardnih tipova dokumenata sa odgovarajućom logičkom strukturom, koja može da podrazumeva organizaciju dokumenata u glave, poglavlja, pasuse i sl. Precizna struktura i odgovarajući način formatiranja dokumenata opisan je tzv. *klasama*. Klasa definiše opštu strukturu i izgled teksta, njegove elemente i osobine.

Srdačno,

Petar Petrović

cc: Univerzitet u Nišu
Univerzitet Crne Gore

encl: Spisak predloga
Lista elektronskih adresa

Primer 4

```
\documentclass{article}
\usepackage[latin2]{inputenc}

\begin{document}

Strane u dokumentu se automatski numerišu, a eksplicitno
postavljanje rednog broja strane može se postići upotrebom
komande \linebreak[4] \verb|\setcounter{page}{broj}|. Komanda
\hspace{10mm} daje horizontalni razmak, a na\red\bom
\verb|\linebreak| možemo eksplicitno sugerisati prelamanje reda.

\bigskip
Doslovni zapis \framebox[2cm][r]{programa:}
\begin{minipage}[t]{60mm}
\begin{verbatim}
    int sum(int n)
    {
        int i, s = 0;
        for(i=1; i<=n; i++) s=s+i;
        return s;
    }
\end{verbatim}
\end{minipage}

\vspace{3\baselineskip}
\noindent Liste se kreiraju na sledeći način:
\begin{description}
\item[prva stavka] tekst...
\begin{itemize}
\item prva stavka
\item druga stavka
\end{itemize}
\item[druga stavka] tekst...
\begin{itemize}
\item[(a)] prva stavka
\item[(b)] druga stavka
\end{itemize}
\end{description}

\end{document}
```

Strane u dokumentu se automatski numerišu, a eksplicitno postavljanje rednog broja strane može se postići upotrebom komande `\setcounter{page}{broj}`. Komanda daje horizontalni razmak, a naredbom `\linebreak` možemo eksplicitno sugerisati prelamanje reda.

Doslovni zapis `programa:`

```
int sum(int n)
{
  int i, s = 0;
  for(i=1; i<=n; i++) s=s+i;
  return s;
}
```

Liste se kreiraju na sledeći način:

prva stavka tekst...

- prva stavka
- druga stavka

druga stavka tekst...

- (a) prva stavka
- (b) druga stavka
 - prva podstavka
 - druga podstavka

Primer 5

```
\documentclass{article}
\usepackage[latin2]{inputenc}
\usepackage[T1]{fontenc}

\begin{document}

Tekst dokumenta predefinirano se poravnava na obe margine
potrebnim povečavanjem razmaka između reči i prelamanjem reči, ali
se delovi teksta mogu formatirati i drugačije.

{\raggedright
Tekst dokumenta predefinirano se poravnava na obe margine
potrebnim povečavanjem razmaka između reči i prelamanjem reči, ali
se delovi teksta mogu formatirati i drugačije. \\
}

\begin{flushright}
Tekst dokumenta predefinirano se poravnava na obe margine
potrebnim povečavanjem razmaka između reči i prelamanjem reči, ali
se delovi teksta mogu formatirati i drugačije.
\end{flushright}

\begin{center}
Tekst dokumenta predefinirano se poravnava na obe margine
potrebnim povečavanjem razmaka između reči i prelamanjem reči, ali
se delovi teksta mogu formatirati i drugačije.
\end{center}

\vspace*{1cm}

\begin{quote}
Tekst dokumenta predefinirano se poravnava na obe margine
potrebnim povečavanjem razmaka između reči i prelamanjem reči, ali
se delovi teksta mogu formatirati i drugačije.
\end{quote}

\begin{quotation}
Tekst dokumenta predefinirano se poravnava na obe margine
potrebnim povečavanjem razmaka između reči i prelamanjem reči, ali
se delovi teksta mogu formatirati i drugačije.
\end{quotation}

\end{document}
```

Tekst dokumenta predefinisano se poravnava na obe margine potrebnim povečavanjem razmaka između reči i prelamanjem reči, ali se delovi teksta mogu formatirati i drugačije.

Tekst dokumenta predefinisano se poravnava na obe margine potrebnim povečavanjem razmaka između reči i prelamanjem reči, ali se delovi teksta mogu formatirati i drugačije.

Tekst dokumenta predefinisano se poravnava na obe margine potrebnim povečavanjem razmaka između reči i prelamanjem reči, ali se delovi teksta mogu formatirati i drugačije.

Tekst dokumenta predefinisano se poravnava na obe margine potrebnim povečavanjem razmaka između reči i prelamanjem reči, ali se delovi teksta mogu formatirati i drugačije.

Tekst dokumenta predefinisano se poravnava na obe margine potrebnim povečavanjem razmaka između reči i prelamanjem reči, ali se delovi teksta mogu formatirati i drugačije.

Tekst dokumenta predefinisano se poravnava na obe margine potrebnim povečavanjem razmaka između reči i prelamanjem reči, ali se delovi teksta mogu formatirati i drugačije.

Primer 6

```

\documentclass[12pt]{article}

\begin{document}

U ispisu ovoga teksta koriste se {\itshape kurzivna}, {\bfseries
podebljana}, {\itshape\bfseries kurzivna podebljana slova},
{\itshape\bfseries\large krupna kurzivna podebljana slova}
i {\sffamily slova {\slshape sans serif} familije}.

\vspace*{5mm}

{\Large
{\rmfamily
{\mdseries {\upshape Aa} {\itshape Aa} {\slshape Aa} {\scshape Aa}}
{\bfseries {\upshape Aa} {\itshape Aa} {\slshape Aa} {\scshape Aa}}
}

{\sffamily
{\mdseries {\upshape Aa} {\itshape Aa} {\slshape Aa} {\scshape Aa}}
{\bfseries {\upshape Aa} {\itshape Aa} {\slshape Aa} {\scshape Aa}}
}

{\ttfamily
{\mdseries {\upshape Aa} {\itshape Aa} {\slshape Aa} {\scshape Aa}}
{\bfseries {\upshape Aa} {\itshape Aa} {\slshape Aa} {\scshape Aa}}
}
}

\vspace*{5mm}

\texttt{\textit{tt-italik}}
\textit{\texttt{italik-tt}}

{\tt{\it tt-italik}}
{\it{\tt italik-tt}}
(\LaTeX{} 2.09 komande za promenu fonta ne mogu da kombinuju attribute)

\end{document}

```


U ispisu ovoga teksta koriste se *kurzivna*, **podebljana**, *kurzivna podebljana slova*, **krupna kurzivna podebljana slova** i slova *sans serif* familije.

Aa Aa Aa AA Aa Aa Aa Aa

Aa Aa Aa AA **Aa Aa Aa Aa**
Aa Aa Aa AA Aa Aa Aa Aa

tt-italik italic-tt

tt-italik italic-tt (\LaTeX 2.09 komande za promenu fonta ne mogu da kombinuju atribute)

Primer 7

```

\documentclass{article}
\usepackage[latin2]{inputenc}

\newtheorem{teorema}{Teorema}[section]

\begin{document}

\setcounter{equation}{500}

\section{Prvo poglavlje}

\begin{teorema}[Prva teorema]
Ovo je tekst prve teoreme.
\end{teorema}

\begin{teorema}[Druga teorema]
\label{thm:druga_teorema}
Ovo je tekst druge teoreme.
\end{teorema}

Ovo je primer formule zapisane u okviru teksta:

$$\int_{\pi}^{\infty} f(a,x)dx = 0 \ ;\ ;\ ;\ \mbox{za } a>0$$
,
ovo je primer izdvojene formule:

$$\int_{\pi}^{\infty} f(a,x)dx = 0 \ ;\ ;\ ;\ \mbox{za } a>0$$

a ovo je primer označene formule:
\begin{equation}
\int_{\pi}^{\infty} f(a,x)dx = 0 \ ;\ ;\ ;\ \mbox{za } a>0
\end{equation}

\section{Drugo poglavlje}

\begin{teorema}[Treća teorema]
Ovo je tekst treće teoreme: važi obrat teoreme
\ref{thm:druga_teorema}.
\end{teorema}

\end{document}

```

1 Prvo poglavlje

Teorema 1.1 (Prva teorema) *Ovo je tekst prve teoreme.*

Teorema 1.2 (Druga teorema) *Ovo je tekst druge teoreme.*

Ovo je primer formule zapisane u okviru teksta: $\int_{\pi}^{\infty} f(a, x)dx = 0$ za $a > 0$, ovo je primer izdvojene formule:

$$\int_{\pi}^{\infty} f(a, x)dx = 0 \quad \text{za } a > 0$$

a ovo je primer označene formule:

$$\int_{\pi}^{\infty} f(a, x)dx = 0 \quad \text{za } a > 0 \quad (501)$$

2 Drugo poglavlje

Teorema 2.1 (Treća teorema) *Ovo je tekst treće teoreme: važi obrat teoreme 1.2.*

Primer 8

```
\documentclass{article}

\begin{document}

% jednostavna tabela
\begin{tabular}{|l|r|r|}
\hline
Prezime i ime & Broj indeksa & Broj poena \\
\hline
Peri\`c Pera & 517/99 & 100 \\
Lazi\`c Laza & 521/99 & 85 \\
Miki\`c Mika & 509/99 & 75 \\
Jovi\`c Jova & 515/99 & 70 \\
\hline
\end{tabular}

\vspace*{1cm}

% komplikovana tabela
\begin{tabular}{|c|c||c|c|} \hline
\multicolumn{4}{|c|}{OSNOVNA PODELA} \\ \hline
\multicolumn{2}{|l|}{DEO 1} & \multicolumn{2}{|r|}{DEO 2} \\ \hline
deo 1.1 & deo 1.2 & deo 1.3 & deo 1.4 \\ \hline
deo 2.1 & \multicolumn{2}{|c|}{zajedni\`v{c}ki deo} & deo 2.4 \\ \hline
deo 3.1 & deo 3.2 & & \\ \cline{1-2}
deo 4.1 & deo 4.2 & deo 4.3 & deo 4.4 \\ \hline
\end{tabular}

\end{document}
```

| Prezime i ime | Broj indeksa | Broj poena |
|---------------|--------------|------------|
| Perić Pera | 517/99 | 100 |
| Lazić Laza | 521/99 | 85 |
| Mikić Mika | 509/99 | 75 |
| Jović Jova | 515/99 | 70 |

| OSNOVNA PODELA | | | |
|----------------|----------------|---------|---------|
| DEO 1 | | DEO 2 | |
| deo 1.1 | deo 1.2 | deo 1.3 | deo 1.4 |
| deo 2.1 | zajednički deo | | deo 2.4 |
| deo 3.1 | deo 3.2 | | |
| deo 4.1 | deo 4.2 | deo 4.3 | deo 4.4 |

Primer 9

```
\documentclass{article}

\begin{document}
% postavlja se jedinica duzine
\setlength{\unitlength}{1mm}

% kreira se crtez konacnog automata za prepoznavanje stringova
% oblika [a]?[bcd]+[e]*
\begin{picture}(100,80)

% stanja
\put(20,40){\circle{3}} \put(15,43){start} % pocetno stanje
\put(40,60){\oval(12,8)} \put(39,59){$1$} % stanje 1
\put(40,20){\oval(12,8)} \put(39,19){$2$} % stanje 2
\put(60,20){\oval(12,8)} \put(59,19){$3$} % stanje 3
\put(80,40){\circle*{3}} \put(80,43){stop} % završno stanje

% prelaz iz pocetnog stanja u stanje 1
\put(21,41){\vector(1,1){15}} \put(30,47){a}

% prelaz iz pocetnog stanja u stanje 2
\put(21,39){\vector(1,-1){15}} \put(30,31){bcd}

% prelaz iz stanja 1 u stanje 2
\put(40,56){\vector(0,-1){32}} \put(41,39){bcd}

% prelaz iz stanja 2 natrag u stanje 2
\put(40,16){\oval(6,4)[b]} \put(38,14){\vector(1,0){4}}
\put(37,10){bcd}

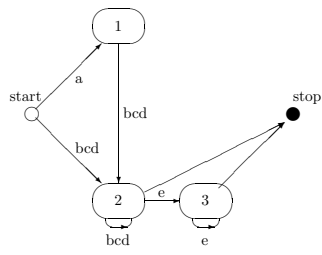
% prelaz iz stanja 2 u stanje 3
\put(46,20){\vector(1,0){8}} \put(49,21){e}

% prelaz iz stanja 2 u završno stanje
\put(46,22){\vector(2,1){32}}

% prelaz iz stanja 3 natrag u stanje 3
\put(60,16){\oval(6,4)[b]} \put(58,14){\vector(1,0){4}} \put(59,10){e}

% prelaz iz stanja 3 u završno stanje
\put(63,23){\vector(1,1){15}}

\end{picture}
\end{document}
```



Primer 10

```
\documentclass{article}
\usepackage{graphicx}

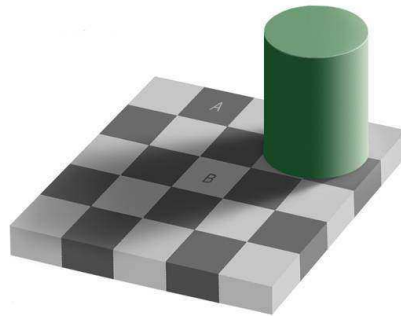
\begin{document}

Polja  $A$  i  $B$  na slici imaju istu boju (tj. jednake  $\{sc\}$  RGB
komponente).

% primer umetanja slike
\includegraphics[width=0.8\textwidth]{pr_pok_3.eps}

\end{document}
```


Polja *A* i *B* na slici imaju istu boju (tj. jednake RGB komponente).



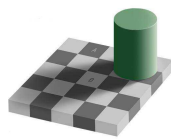
Primer 11

```
\documentclass{article}
\usepackage{graphicx}

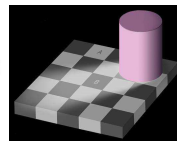
\begin{document}

% Primer umetanja slika jedne pored druge.
\renewcommand{\figurename}{Slika}
\begin{figure}
\centering
\begin{minipage}[t]{0.33\textwidth}
\centering
\includegraphics[width=4cm]{pr_po_4a.eps}
\caption{Prva slika}
\end{minipage}
\begin{minipage}[t]{0.33\textwidth}
\centering
\includegraphics[width=4cm]{pr_po_4b.eps}
\caption{Druga slika}
\end{minipage}
\end{figure}

\end{document}
```



Slika 1: Prva slika



Slika 2: Druga slika

Primer 12

```
\documentclass{article}
\usepackage{makeidx}
\renewcommand{\indexname}{Indeks}

\makeindex

\begin{document}

\Large

Ovo je tekst o programskom jeziku {\em C++}.

Ovde se govori o klasama\index{klase}\ldots

\newpage

Ovde o objektima\index{objekti}\ldots

\newpage

A ovde se pominju apstraktne klase\index{klase!apstraktne} i {\em
  friend} funkcije\index{friend@{\em friend} funkcije}\ldots

\newpage

\printindex

\end{document}
```

| | |
|--|---|
| <p>Ovo je tekst o programskom jeziku C++. Ovde se govori o klasama...</p> <p>1</p> | <p>Ovde o objektima...</p> <p>2</p> |
| <p>A ovde se pominju apstraktne klase i <i>friend</i> funkcije...</p> <p>3</p> | <p>Indeks <i>friend</i> funkcije, 3 klase, 1 apstraktne, 3 objekti, 2</p> <p>4</p> |

Primer 13

```
\documentclass{article}
\renewcommand{\refname}{Literatura}

\begin{document}

Osnovna literatura za \TeX{} je \cite{knut86}. Sistem takozvanih
makroa, na kojima se bazira \LaTeX{}, je opisan u \cite[poglavlje
20]{knut86}. \LaTeX{} je dobro opisan u \cite{lamp94, goos94}.

\begin{thebibliography}{9}
\bibitem{knut86} Donald Knuth, {\em The \TeX{}book},
  Addison-Wesley, 1986, ISBN 0-201-13447-0
\bibitem{lamp94} Leslie Lamport, {\em \LaTeX{}: A Document
  Preparation System}, Addison-Wesley, 1994, ISBN 0-201-52983-1
\bibitem{goos94} Michel Goosens, Frank Mittelbach, Alexander
  Samarin, {\em The \LaTeX{} Companion}, Addison-Wesley, 1994,
  ISBN 0-201-54199-8
\end{thebibliography}

\end{document}
```

Osnovna literatura za \TeX je [1]. Sistem takozvanih makroa, na kojima se bazira \LaTeX , je opisan u [1, poglavlje 20]. \LaTeX je dobro opisan u [2, 3].

Literatura

- [1] Donald Knuth, *The \TeX book*, Addison-Wesley, 1986, ISBN 0-201-13447-0
- [2] Leslie Lamport, *\LaTeX : A Document Preparation System*, Addison-Wesley, 1994, ISBN 0-201-52983-1
- [3] Michel Goossens, Frank Mittelbach, Alexander Samarin, *The \LaTeX Companion*, Addison-Wesley, 1994, ISBN 0-201-54199-8

Primer 14

```
\documentclass{article}
\renewcommand{\refname}{Literatura}

\begin{document}

Osnovna literatura za \TeX{} je \cite{knut86}. Sistem takozvanih
makroa, na kojima se bazira \LaTeX{}, je opisan u \cite[poglavlje
20]{knut86}. \LaTeX{} je dobro opisan u \cite{lamp94, goos94}.

\bibliographystyle{plain}
\bibliography{pr_ref_3}

% datoteka pr_ref_3.bib ima oblik:
%
% @BOOK (knut86,
% author = "Donald Knuth",
% title = "The \TeX{}book",
% publisher = "Addison-Wesley",
% year = "1986",
% isbn = "0-201-13447-0")
%
% @BOOK (lamp94,
% author = "Leslie Lamport",
% title = "\LaTeX{}: A Document Preparation System",
% publisher = "Addison-Wesley",
% year = "1994",
% isbn = "0-201-52983-1")
%
% @BOOK (goos94,
% author = "Michel Goosens and Frank Mittelbach and Alexander Samarin",
% title = "The \LaTeX{} Companion",
% publisher = "Addison-Wesley",
% year = "1994",
% isbn = "0-201-54199-8")

\end{document}
```


Osnovna literatura za \TeX je [2]. Sistem takozvanih makroa, na kojima se bazira \LaTeX , je opisan u [2, poglavlje 20]. \LaTeX je dobro opisan u [3, 1].

Literatura

- [1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The \LaTeX Companion*. Addison-Wesley, 1994.
- [2] Donald Knuth. *The \TeX book*. Addison-Wesley, 1986.
- [3] Leslie Lamport. *\LaTeX : A Document Preparation System*. Addison-Wesley, 1994.

Primer 15

```
\documentclass{article}
\usepackage[english,serbian]{babel}

\begin{document}

Danas je \today. Mo"zemo da pi"semo i na"sim latini"cnim
slovima\ldots

\selectlanguage{english}

Today is \today.

\end{document}
```

Danas je 29. novembar 2013. Možemo da pišemo i našim latiničnim slovima...
Today is November 29, 2013.

Primer 16

```
\documentclass{article}
\usepackage[T2A,T1]{fontenc}
\usepackage[iso88595,latin2]{inputenc}

\begin{document}

\inputencoding{latin2}
\fontencoding{T1}\selectfont

% pr_vi_2a je fajl u kome latinicom, u latin2 kodiranju, pise;
%   \{c}ilim, \{c}i\{c}ak, djurdjevak, \{s}al, \{z}bun
\input{pr_vi_2a}

\inputencoding{iso88595}
\fontencoding{T2A}\selectfont

% pr_vi_2b je fajl u kome cirilicom, u iso88595 kodiranju, pise:
%   \{c}ilim, \{c}i\{c}ak, djurdjevak, \{s}al, \{z}bun
\input{pr_vi_2b}

\end{document}
```

ćilim, ćićak, durdevak, šal, žbun
ћилим, чичак, ђурђевак, шал, жбун

Primer 17

```
\documentclass{article}
\usepackage[OT2,OT1]{fontenc}

\begin{document}

\'cilim, \v{c}i\v{c}ak, djurdjevak, \v{s}al, \v{z}bun

\fontfamily{wncyr}
\fontencoding{OT2}\selectfont

ciilim, chichak, djurdjevak, shal, zhbun

\end{document}
```

ćilim, ćićak, đurdjevak, sal, žbun
ћилим, чичак, ђурђеџак, шал, жбун

Primer 18

```
\documentclass[a4paper]{article}
\usepackage[OT1,OT2]{fontenc}
\usepackage{makeidx}

\newtheorem{hipoteza}{Hipoteza}

\renewcommand{\abstractname}{Apstrakt}
\renewcommand{\contentsname}{Sadržaj}
\renewcommand{\indexname}{Indeks}
\renewcommand{\refname}{Literatura}

\renewcommand{\rmdefault}{wncyr}
\newcommand{\Lat}{\fontencoding{OT1}\fontfamily{cmr}\selectfont}

\title{Problemi za milenijumsku nagradu\thanks{Ovaj tekst je
      zasnovan na D2on Baezovom opisu milenenijumskih problema
      ({\Lat
        {\texttt{http://math.ucr.edu/home/baez/twf.html}}}) i
      na tekstovima sa Internet lokacije Matematichkog
      instituta Klej ({\Lat
        {\texttt{http://www.claymath.org/}}}).} }
\author{Dragoslav S. Pavic1evic1 \\\
      Matematichki fakultet \\\
      Student\ -ski trg 16 \\\
      11000 Beograd } \date{}

\makeindex

\begin{document}

\maketitle

\begin{abstract}
  Na ulasku u novi milenijum, redakcija vodeclih svet\ -skih
  ma\ -te\ -ma\ -ti\ -cha\ -ra, pod okriljem Matematichkog
  instituta Kleja, izabrala je sedam znachajnih
  matematichkih problema koji do danas nisu resheni. Za
  reshenje svakog od njih pripremljena je nagrada od milion
  dolara. \index{dolar}
\end{abstract}

\setcounter{section}{-1}
```


\section{Uvod}

Na drugom medjunarodnom kongresu matematchara odrzhanom u Parizu 1900.~godine, David Hilbert ($\text{\LaTeX David Hilbert}$) odrzhao je znamenito predavanje o (tadashnjim) velikim otvorenim matematchkim problemima. U slicnom duhu, jedan vek kasnije, redakcija vodeclih svet\~skih matematchara, pod okriljem Matematchkog instituta Kleja ($\text{\LaTeX Clay Mathematics Institute of Cambridge, Massachusetts --- CMI}$), $\text{\index{Masachuset\~s}}$ izabrala je sedam ,,problema za milenijumsku nagradu“ --- sedam znacajnih matematchkih problema koji danas nisu reseni. Za resenje svakog od izabranih problema pripremljena je i novchana nagrada od 1000000\$. $\text{\index{dolar@\$}}$ U po\~glav\~lji\~ma od $\text{\ref{sec:P_NP}}$ do $\text{\ref{sec:Rimanova-hipoteza}}$ dat je kratak prikaz svakog od tih sedam problema.

\section{\LaTeX\textsc{\textmd{p=np}}{?}}
\label{sec:P_NP}

Ovo je najnoviji na listi sedam ,,milenijumskih problema“ i on pripada teorijskom rachunarstvu (problem je 1971.~godine formulisao Kuk (\LaTeX Cook) $\text{\cite{cook}}$). $\text{\index{Kuk}}$ Neki problem pripada klasi klasi $\text{\LaTeX\textsc{p}}$ ako postoji algoritam za njegovo reshavanje chi je je vreme izvrshavanja ograniceno nekim polinomom po duzhini ulaznih podataka. Problem pripada klasi $\text{\LaTeX\textsc{np}}$ ako postoji algoritam takav da je nakon primene niza takozvanih $\text{\LaTeX\textit{nd}}$ --izbora moguclie u polinomijalnom vremenu $\text{\underline{proveriti}}$ da li izabrano resenje zaista zadovoljava konkretnu instancu problema. $\text{\footnote{Vishe o klasama \LaTeX\textsc{p} i \LaTeX\textsc{np} i o \LaTeX\textsc{np}-kompletnosti videti, na primer, u \cite{garey-johnson}.}$ Ochigledno je da je $\text{\LaTeX\textsc{p}}\text{\subseq}\text{\LaTeX\textsc{np}}$, ali se josh uvek ne zna da li vazhi $\text{\LaTeX\textsc{p}}\text{\subseq}\text{\LaTeX\textsc{np}}$.

\section{Poenkareovo tvrdjenje}

$\text{\index{hipoteza!Poenkareova}}$ Sledecii problem formulisao je Poenkare (\LaTeX Poincar'e) 1904.~godine: da li je svaka trodimenziona mnogostrukost istog homotopskog tipa kao i trodimenziona sfera homeomorfna sa trodimenzionom sferom? Ana\~lo\~g\~oni Poenkareovog problema su potvrdno reseni za sve dimenzije vecie od tri i to je lepa ilustracija

činjenice da su mnogi problemi topologije teži za rešavanje u malim dimenzijama. `\index{topologija}`

`\section{Birk i Swinerton-Dajerovo tvrdjenje}`

Ovaj problem je u vezi sa eliptičkim krivama i, indirektno, sa teorijom brojeva. Eliptičke krive opisane su jednadžinama kao što je, na primer, $y^2 = x^3 + ax + b$. Svaka eliptička kriva određuje jednu Abelovu grupu i ona je konavno generisana tachakama sa racionalnim koordinatama. Pitanje je u kom slučaju kriva ima beskonavno mnogo tachaka sa racionalnim koordinatama. Birk (`{\Lat Birch}`) i Swinerton-Dajer (`{\Lat Swinerton-Dyer}`) su formulisali 1965. godine tvrdjenje koje uključuje takozvanu L -funkciju eliptičke krive. L -funkcija $L(s)$ odgovara broju rešenja odgovarajuće jednadžine po modulu p , gde je p prost broj. Tvrdi se da je $L(1) \neq 0$ ako i samo ako eliptička kriva ima beskonavno mnogo racionalnih tachaka. Rešenje ovog problema vodilo bi rešenju mnogih problema vezanih za diofantovske jednadžine.

`\section{Hodžovo tvrdjenje}`

Ovaj problem je u domenu algebarske geometrije i topologije. `\emph{Projektivni nesingularni algebarski varijetet}` je, pojednostavljeno, kompaktna glatka mnogostrukost opisana homogenim kompleksnim polinomijalnim jednakostima. Takav varijetet uvek ima parnu dimenziju, recimo $2n$. De Rhamova (`{\Lat De Rham}`) kohologija takvog varijeteta može se razložiti na delove $H^{p,q}$ oznachene parovima (p,q) celih brojeva između 0 i n , koristeći činjenicu da je svaka funkcija zbir holomorfne i antiholomorfne dela. Racionalni elementi unutar $H^{p,p}$ zovu se Hodžove (`{\Lat Hodge}`) forme. Na osnovu Poincaréove dualnosti, svaki zatvoreni analitički potprostor izabranog varijeteta definiše Hodžovu formu --- ovaj oblik Hodžove forme zove se `\emph{algebarski ciklus}`. Hodžovo tvrdjenje, formulisano 1950. godine glasi: svaka Hodžova forma je racionalna linearna kombinacija algebarskih ciklusa.

`\section{Postojanje i masena praznina za Jang-Milsovu teoriju}`

Jedan od najvećih problema moderne matematičke fizike je da li je standardni model fizike čestica matematički konzistentan. Ne zna se ni da li je „čista“ Jang-Milsova (`{\Lat Yang-Mills}`) teorija (nevezana za fermione) dobro

zasnovana kvantna teorija polja sa zheljenim svoj\stvi\ma. Kako bi se ovo pitanje preciziralo, formulisane su razlichite aksiome za kvantnu teoriju polja, kao, na primer, Haag-Kastler ($\text{\LaTeX Haag-Kastler}$) aksiome. Konstruktivistichka kvantna teorija polja is\pi\tu\je da li se mozhe izgraditi Jang-Milsova teorija tako da ona zadovoljava te aksiome. Shtavishe, potrebno je ispitati i postojanje Jang-Milsove teorije vezane za fermione, zajedno sa tzv. masenom prazninom.

$\text{\section{Postojanje i glatkost za jednachine Navijer-Stoksa}}$

Jednachine Navijer-Stoksa ($\text{\LaTeX Navier-Stokes}$) predstavljaju sistem di\fe\ren\ci\jal\nih jednachina koje opisuju tok viskozno fluida. Polje vektora koje opisuje tok fluida postaje veoma komplikovano u prisustvu turbulencija. Nije poznato da li postoji opshte reshenje skupa jednachina ili je moguće da se pojave singulariteti i da problem postane nedefinisan. Pojednostavljeno, ključni problem je odrediti da li reshenje postoji i ostaje glatko ili utvrditi uslove pod kojim to jeste sluchaj.

$\text{\section{Rimanova hipoteza}}$

$\text{\label{sec:Rimanova-hipoteza}}$

Rimanova (\LaTeX Riemann) zeta funkcija definishe se kao analitichko pro\shi\re\nje funkcije

$$\zeta(s) = \frac{1}{1^s} + \frac{1}{2^s} + \frac{1}{3^s} + \dots \quad \text{\;}; \text{\;}; \text{\;}; \text{Re}(s) > 1$$

Ona ima polove u celim brojevima manjim ili jednakim 1 . Zeta funk\ci\ja ima mnoshtvo nula u ,,kritichnoj traci“ gde je $\text{Re}(s)$ izmedju 0 i 1 . Godine 1859. Riman je postavio hipotezu $\text{\ref{hipoteza:Rimanova}}$. $\text{\index{hipoteza!Rimanova}}$

$\text{\begin{hipoteza}[Rimanova hipoteza]}$

$\text{\label{hipoteza:Rimanova}}$

Sve nule zeta funkcije za koje je $\text{Re}(s)$ izmedju 0 i 1 , imaju realni deo jednak $1/2$.

$\text{\end{hipoteza}}$

Ovo tvrdjenje ima veliki broj posledica, ukljuchujući one koji se odnose na distribuciju prostih brojeva. Do sada je za vishe od mi\li\jar\du brojeva u ,,kritichnoj traci“ utvrdjeno da imaju realni deo jednak $1/2$, ali Rimanova hipoteza ostaje i dalje nedokazana.

```
\section*{Zakljuchak}

U prethodnom tekstu ukratko su opisani ,milenijumski
problemi‘‘ (izabrani od strane Matematichkog instituta
Klej). Nadamo se da cile ovaj chlanak podstacili chitaoca da
se pozabavi navedenim problemima i time eventualno zaradi
nekoliko miliona dolara. \index{dolar@\$}

\paragraph{Zahvalnica}
Zahvaljujem kolegama Zharku Mijajloviclu, Zoranu
Pet\~{r}o\~{v}i\~{c}lu, Srdjanu Vukmiroviclu i Zoranu Staniclu na
dragocenim su\~{g}e\~{s}\~{t}i\~{j}a\~{m}a na radnu verziju ovog teksta.

{\Lat
\begin{thebibliography}{1}
\bibitem{cook} S.~A. Cook. The complexity of theorem proving
procedures. In \emph{Proceedings of the 3rd Annual ACM
Symposium on the Theory of Computation}, pages 151--158,
1971.
\bibitem{garey-johnson} M.~R. Garey and D.~S. Johnson.
\emph{Computers and Intractability}. W.~H.~Freeman, New
York, 1979.
\end{thebibliography}
}

\tableofcontents

\printindex

\end{document}
```

Проблеми за миленијумску награду*

Драгослав С. Павићевић
Математички факултет
Студентски трг 16
11000 Београд

Апстракт

На уласку у нови миленијум, редакција водећих светских математичара, под окриљем Математичког института Клеја, изабрала је седам значајних математичких проблема који до данас нису решени. За решење сваког од њих припремљена је награда од милион долара.

0 Увод

На другом међународном конгресу математичара одржаном у Паризу 1900. године, Давид Хилберт (David Hilbert) одржао је знаменито предавање о (тадашњим) великим отвореним математичким проблемима. У сличном духу, један век касније, редакција водећих светских математичара, под окриљем Математичког института Клеја (Clay Mathematics Institute of Cambridge, Massachusetts — CMI), изабрала је седам „проблема за миленијумску награду“ — седам значајних математичких проблема који до данас нису решени. За решење сваког од изабраних проблема припремљена је и новчана награда од 1000000\$. У поглављима од 1 до 7 дат је кратак приказ сваког од тих седам проблема.

1 $P=NP?$

Ово је најновији на листи седам „миленијумских проблема“ и он припада теоријском рачунарству (проблем је 1971. године формулисао Кук (Cook)[1]). Неки проблем припада класи P ако постоји алгоритам за његово решавање чије је време извршавања ограничено неким полиномом по дужини улазних података. Проблем припада класи NP ако постоји алгоритам такав да је након примене низа такозваних nd -избора могуће у полиномијалном времену проверити да ли изабрано решење заиста задовољава конкретну инстанцу проблема.¹ Очигледно је да је $P \subseteq NP$, али се још увек не зна да ли важи $P=NP$.

*Овај текст је заснован на Џон Баезовом опису миленијумских проблема (<http://math.ucr.edu/home/baez/twf.html>) и на текстовима са Интернет локације Математичког института Клеј (<http://www.claymath.org/>).

¹Више о класама P и NP и о NP -комплетности видети, на пример, у [2].

2 Поенкареово тврђење

Следећи проблем формулисао је Поенкаре (Poincaré) 1904. године: да ли је свака тродимензиона многострукост истог хомотопског типа као и тродимензиона сфера хомеоморфна са тродимензионом сфером? Аналогони Поенкареовог проблема су потврдно решени за све димензије веће од три и то је лепа илустрација чињенице да су многи проблеми топологије тежи за решавање у малим димензијама.

3 Бирк и Свинертон-Дајерово тврђење

Овај проблем је у вези са елиптичким кривама и, индиректно, са теоријом бројева. Елиптичке криве описане су једначинама као што је, на пример, $y^2 = x^3 + ax + b$. Свака елиптичка крива одређује једну Абелову групу и она је коначно генерисана тачакама са рационалним координатама. Питање је у ком случају крива има бесконачно много тачака са рационалним координатама. Бирк (Birch) и Свинертон-Дајер (Swinnerton-Dyer) су формулисали 1965. године тврђење које укључује такозвану L -функцију елиптичке криве. L -функција $L(s)$ одговара броју решења одговарајуће једначине по модулу p , где је p прост број. Тврди се да је $L(1) = 0$ ако и само ако елиптичка крива има бесконачно много рационалних тачака. Решење овог проблема водило би решењу многих проблема везаних за диофантовске једначине.

4 Хоцово тврђење

Овај проблем је у домену алгебарске геометрије и топологије. *Проективни несингуларни алгебарски варијетет* је, поједностављено, компактна глатка многострукост описана хомогеним комплексним полиномијалним једнакостима. Такав варијетет увек има парну димензију, рецимо $2n$. Де Рамова (De Rham) кохомологија таквог варијетета може се разложити на делове $H^{p,q}$ означене паровима (p, q) целих бројева између 0 и n , користећи чињеницу да је свака функција збир холоморфног и антихоломорфног дела. Рационални елементи унутар $H^{p,p}$ зову се Хоцове (Hodge) форме. На основу Поенкареове дуалности, сваки затворени аналитички потпростор изабраног варијетета дефинише Хоцову форму — овај облик Хоцове форме зове се *алгебарски циклус*. Хоцово тврђење, формулисано 1950. године гласи: свака Хоцова форма је рационална линеарна комбинација алгебарских циклуса.

5 Постојање и масена празнина за Јанг-Милсову теорију

Један од највећих проблема модерне математичке физике је да ли је стандардни модел физике честица математички конзистентан. Не зна се ни да ли је „чиста“ Јанг-Милсова (Yang-Mills) теорија (невезана

за фермионе) добро заснована квантна теорија поља са жељеним својствима. Како би се ово питање прецизирало, формулисане су различите аксиоме за квантну теорију поља, као, на пример, Хааг-Кастлер (Haag-Kastler) аксиоме. Конструктивистичка квантна теорија поља испитује да ли се може изградити Јанг-Милсова теорија тако да она задовољава те аксиоме. Штавише, потребно је испитати и постојање Јанг-Милсове теорије везане за фермионе, заједно са тзв. масеном празнином.

6 Постојање и глаткост за једначине Навијер-Стокса

Једначине Навијер-Стокса (Navier-Stokes) представљају систем диференцијалних једначина које описују ток вискозног флуида. Поље вектора које описује ток флуида постаје веома компликовано у присуству турбуленција. Није познато да ли постоји опште решење скупа једначина или је могуће да се појаве сингуларитети и да проблем постане недефинисан. Поједностављено, кључни проблем је одредити да ли решење постоји и остаје глатко или утврдити услове под којим то јесте случај.

7 Риманова хипотеза

Риманова (Riemann) зета функција дефинише се као аналитичко проширење функције

$$\zeta(s) = \frac{1}{1^s} + \frac{1}{2^s} + \frac{1}{3^s} + \dots, \operatorname{Re}(s) > 1$$

Она има полове у целим бројевима мањим или једнаким 1. Зета функција има мноштво нула у „критичној траци“ где је $\operatorname{Re}(s)$ између 0 и 1. Године 1859. Рيمان је поставио хипотезу 1.

Хипотеза 1 (Риманова хипотеза) Све нуле зета функције за које је $\operatorname{Re}(s)$ између 0 и 1, имају реални део једнак 1/2.

Ово тврђење има велики број последица, укључујући оне који се односе на дистрибуцију простих бројева. До сада је за више од милијарду бројева у „критичној траци“ утврђено да имају реални део једнак 1/2, али Риманова хипотеза остаје и даље недоказана.

Закључак

У претходном тексту укратко су описани „миленијумски проблеми“ (изабрани од стране Математичког института Клеј). Надамо се да ће овај чланак podstaћи читаоца да се позабави наведеним проблемима и тиме евентуално заради неколико милиона долара.

Захвалница Захваљујем колегама Жарку Мијајловићу, Зорану Петровићу, Срђану Вукировићу и Зорану Станићу на драгоценим сугестијама на радну верзију овог текста.

Литература

- [1] S. A. Cook. The complexity of theorem proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computation*, pages 151–158, 1971.
- [2] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman, New York, 1979.

Садржај

| | | |
|---|---|---|
| 0 | Увод | 1 |
| 1 | $P=NP?$ | 1 |
| 2 | Поенкареово тврђење | 2 |
| 3 | Бирк и Свинертон-Дајерово тврђење | 2 |
| 4 | Хофово тврђење | 2 |
| 5 | Постојање и масена празнина за Јанг-Милсову теорију | 2 |
| 6 | Постојање и глаткост за једначине Навијер-Стокса | 3 |
| 7 | Риманова хипотеза | 3 |

Индекс

долар, 1

§, 1, 3

хипотеза

 Поенкареова, 2

 Риманова, 3

Кук, 1

Масачусетс, 1

топологија, 2

Primer 19

```

\documentclass{slides}

\begin{document}

Cauchy-jev problem reda  $n$  je definisan diferencijalnom
jednačinom:
\begin{displaymath}
\mathbf{u}'(x)=\mathbf{f}(x,\mathbf{u}(x))
\end{displaymath}
uz početni uslov  $\mathbf{u}(x_0)=\mathbf{u}^{(0)}$ .
Ovde je:
\begin{displaymath}
\mathbf{u}(x)=\left[
\begin{array}{c}
u_0(x) \\
u_1(x) \\
\vdots \\
u_{n-1}(x)
\end{array}
\right]
\end{displaymath}
zatim:
\begin{displaymath}
\mathbf{f}(x,\mathbf{u}(x))=\left[
\begin{array}{c}
f_0(x,u_0,u_1,\dots,u_{n-1}) \\
f_1(x,u_0,u_1,\dots,u_{n-1}) \\
\vdots \\
f_{n-1}(x,u_0,u_1,\dots,u_{n-1})
\end{array}
\right]
\end{displaymath}
i:
\begin{displaymath}
\mathbf{u}^{(0)}=\left[
\begin{array}{c}
u_0^{(0)} \\
u_1^{(0)} \\
\vdots \\
u_{n-1}^{(0)}
\end{array}
\right]
\end{displaymath}

```

\end{document}

Cauchy-jev problem reda n je definisan diferencijalnom jednačinom:

$$\mathbf{u}'(x) = \mathbf{f}(x, \mathbf{u}(x))$$

uz početni uslov $\mathbf{u}(x_0) = \mathbf{u}^{(0)}$. Ovde je:

$$\mathbf{u}(x) = \begin{bmatrix} u_0(x) \\ u_1(x) \\ \dots \\ u_{n-1}(x) \end{bmatrix}$$

zatim:

$$\mathbf{f}(x, \mathbf{u}(x)) = \begin{bmatrix} f_0(x, u_0, u_1, \dots, u_{n-1}) \\ f_1(x, u_0, u_1, \dots, u_{n-1}) \\ \dots \\ f_{n-1}(x, u_0, u_1, \dots, u_{n-1}) \end{bmatrix}$$

i:

$$\mathbf{u}^{(0)} = \begin{bmatrix} u_0^{(0)} \\ u_1^{(0)} \\ \dots \\ u_{n-1}^{(0)} \end{bmatrix}$$

Primer 20

```
\documentclass{beamer}

\usetheme{Pittsburgh}
\usecolortheme{beetle}
\setbeamertemplate{navigation symbols}{}

\title{Osnove \LaTeX{-}a}
\author{Aleksandar \& Goran \& Predrag}
\institute{Univerzitet u Beogradu}
\date{}

\begin{document}

\begin{frame}
\titlepage
\end{frame}

\begin{frame}{Uvod}
\begin{itemize}
\item \LaTeX{} se bazira na \TeX{-}u
\item autor \TeX{-}a je Donald Knuth
\item autor \LaTeX{-}a je Leslie Lamport
\end{itemize}

Neke \LaTeX{} klase dokumenata su:
\begin{itemize}
\item article
\item book
\item letter
\item proc
\item report
\item slides
\end{itemize}
\end{frame}

\end{document}
```

Osnove \LaTeX -a

Aleksandar & Goran & Predrag

Univerzitet u Beogradu

Uvod

- \LaTeX se bazira na \TeX -u
- autor \TeX -a je Donald Knuth
- autor \LaTeX -a je Leslie Lamport

Neke \LaTeX klase dokumenata su:

- article
- book
- letter
- proc
- report
- slides

Dodatak C

Tabele komandi

C.1 Specijalni simboli i dijakritici

| | |
|-----------------------|-------------------------|
| † | <code>\dag</code> |
| § | <code>\S</code> |
| © | <code>\copyright</code> |
| ‡ | <code>\ddag</code> |
| ¶ | <code>\P</code> |
| £ | <code>\pounds</code> |
| œ | <code>\oe</code> |
| Œ | <code>\OE</code> |
| æ | <code>\AE</code> |
| Æ | <code>\AE</code> |
| ä | <code>\aa</code> |
| Å | <code>\AA</code> |
| ø | <code>\o</code> |
| Ø | <code>\O</code> |
| l | <code>\l</code> |
| L | <code>\E</code> |
| ß | <code>\ss</code> |
| ı | <code>?´</code> |
| ı | <code>!´</code> |
| \TeX | <code>\TeX</code> |
| \LaTeX | <code>\LaTeX</code> |
| $\LaTeX 2_{\epsilon}$ | <code>\LaTeXe</code> |

Tabela C.1: Zapisi specijalnih simbola

| | |
|----|----------|
| à | \`{a} |
| á | \'{a} |
| ä | \"{a} |
| ā | \={a} |
| ã | \~{a} |
| â | \^{a} |
| à | \cdot{a} |
| ä | \b{a} |
| ç | \c{a} |
| ä | \d{a} |
| â | \H{a} |
| âä | \t{aa} |
| ä | \u{a} |
| ã | \v{a} |

Tabela C.2: Zapisi dijakritičkih znakova

C.2 Matematički simboli

| | |
|-------------|-----------|
| \hat{a} | \hat{a} |
| \acute{a} | \acute{a} |
| \bar{a} | \bar{a} |
| \dot{a} | \dot{a} |
| \breve{a} | \breve{a} |
| \check{a} | \check{a} |
| \grave{a} | \grave{a} |
| \vec{a} | \vec{a} |
| \ddot{a} | \ddot{a} |
| \tilde{a} | \tilde{a} |

Tabela C.3: Zapisi dijakritičkih znakova u matematičkom modu

| | |
|------------------------|-----------------------------------|
| \widetilde{abc} | <code>\widetilde{abc}</code> |
| \widehat{abc} | <code>\widehat{abc}</code> |
| \overleftarrow{abc} | <code>\overleftarrow{abc}</code> |
| \overrightarrow{abc} | <code>\overrightarrow{abc}</code> |
| \overline{abc} | <code>\overline{abc}</code> |
| \underline{abc} | <code>\underline{abc}</code> |
| \overbrace{abc} | <code>\overbrace{abc}</code> |
| \underbrace{abc} | <code>\underbrace{abc}</code> |
| \sqrt{abc} | <code>\sqrt{abc}</code> |
| $\sqrt[n]{abc}$ | <code>\sqrt[n]{abc}</code> |
| f' | <code>f'</code> |
| f'' | <code>f''</code> |
| f''' | <code>f'''</code> |
| $\frac{abc}{xyz}$ | <code>\frac{abc}{xyz}</code> |

Tabela C.4: Simboli za označavanje niski slova

| | | | |
|---------------|--------------------------|------------|-----------------------|
| α | <code>\alpha</code> | A | A |
| β | <code>\beta</code> | B | B |
| γ | <code>\gamma</code> | Γ | <code>\Gamma</code> |
| δ | <code>\delta</code> | Δ | <code>\Delta</code> |
| ϵ | <code>\epsilon</code> | E | E |
| ε | <code>\varepsilon</code> | E | E |
| ζ | <code>\zeta</code> | Z | Z |
| η | <code>\eta</code> | H | H |
| θ | <code>\theta</code> | Θ | <code>\Theta</code> |
| ϑ | <code>\vartheta</code> | Θ | <code>\Theta</code> |
| ι | <code>\iota</code> | I | I |
| κ | <code>\kappa</code> | K | K |
| λ | <code>\lambda</code> | Λ | <code>\Lambda</code> |
| μ | <code>\mu</code> | M | M |
| ν | <code>\nu</code> | N | N |
| ξ | <code>\xi</code> | Ξ | <code>\Xi</code> |
| o | <code>o</code> | O | O |
| π | <code>\pi</code> | Π | <code>\Pi</code> |
| ϖ | <code>\varpi</code> | Π | <code>\Pi</code> |
| ρ | <code>\rho</code> | P | P |
| ϱ | <code>\varrho</code> | P | P |
| σ | <code>\sigma</code> | Σ | <code>\Sigma</code> |
| ς | <code>\varsigma</code> | Σ | <code>\Sigma</code> |
| τ | <code>\tau</code> | T | T |
| υ | <code>\upsilon</code> | Υ | <code>\Upsilon</code> |
| ϕ | <code>\phi</code> | Φ | <code>\Phi</code> |
| φ | <code>\varphi</code> | Φ | <code>\Phi</code> |
| χ | <code>\chi</code> | X | X |
| ψ | <code>\psi</code> | Ψ | <code>\Psi</code> |
| ω | <code>\omega</code> | Ω | <code>\Omega</code> |

Tabela C.5: Zapisi grčkih slova

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

Tabela C.6: Simboli relacija

| | | | |
|-----------------|----------------------------|------------------|-----------------------------|
| $+$ | <code>+</code> | $-$ | <code>-</code> |
| \pm | <code>\pm</code> | \mp | <code>\mp</code> |
| \times | <code>\times</code> | \div | <code>\div</code> |
| \cap | <code>\cap</code> | \cup | <code>\cup</code> |
| \sqcap | <code>\sqcap</code> | \sqcup | <code>\sqcup</code> |
| \oplus | <code>\oplus</code> | \ominus | <code>\ominus</code> |
| \otimes | <code>\otimes</code> | \oslash | <code>\oslash</code> |
| \odot | <code>\odot</code> | \bigcirc | <code>\bigcirc</code> |
| \triangleleft | <code>\triangleleft</code> | \triangleright | <code>\triangleright</code> |
| \triangleup | <code>\triangleup</code> | \triangledown | <code>\triangledown</code> |
| \diamond | <code>\diamond</code> | \circ | <code>\circ</code> |
| \bullet | <code>\bullet</code> | \cdot | <code>\cdot</code> |
| \uplus | <code>\uplus</code> | \amalg | <code>\amalg</code> |
| \ast | <code>\ast</code> | \star | <code>\star</code> |
| \setminus | <code>\setminus</code> | \wr | <code>\wr</code> |
| \vee | <code>\vee</code> | \wedge | <code>\wedge</code> |
| \dagger | <code>\dagger</code> | \ddagger | <code>\ddagger</code> |

Tabela C.7: Simboli binarnih operacija

| | | | |
|--------------------|-------------------------------|------------------------|-----------------------------------|
| \leftarrow | <code>\leftarrow</code> | \rightarrow | <code>\rightarrow</code> |
| \Leftarrow | <code>\Leftarrow</code> | \Rightarrow | <code>\Rightarrow</code> |
| \longleftarrow | <code>\longleftarrow</code> | \longrightarrow | <code>\longrightarrow</code> |
| \Lleftarrow | <code>\Lleftarrow</code> | \Longrightarrow | <code>\Longrightarrow</code> |
| \leftrightarrow | <code>\leftrightarrow</code> | \longleftrightarrow | <code>\longleftrightarrow</code> |
| \Leftrightarrow | <code>\Leftrightarrow</code> | \Llongleftrightarrow | <code>\Llongleftrightarrow</code> |
| \uparrow | <code>\uparrow</code> | \downarrow | <code>\downarrow</code> |
| \Uparrow | <code>\Uparrow</code> | \Downarrow | <code>\Downarrow</code> |
| \updownarrow | <code>\updownarrow</code> | \Updownarrow | <code>\Updownarrow</code> |
| \leftharpoonup | <code>\leftharpoonup</code> | \rightharpoonup | <code>\rightharpoonup</code> |
| \leftharpoondown | <code>\leftharpoondown</code> | \rightharpoondown | <code>\rightharpoondown</code> |
| \hookrightarrow | <code>\hookrightarrow</code> | \hookleftarrow | <code>\hookleftarrow</code> |
| \nearrow | <code>\nearrow</code> | \searrow | <code>\searrow</code> |
| \swarrow | <code>\swarrow</code> | \nwarrow | <code>\nwarrow</code> |
| \mapsto | <code>\mapsto</code> | \longmapsto | <code>\longmapsto</code> |

Tabela C.8: Simboli strelica

| | | | |
|-------------|------------------------|--------------|-------------------------|
| \sum | <code>\sum</code> | \prod | <code>\prod</code> |
| \bigcap | <code>\bigcap</code> | \bigcup | <code>\bigcup</code> |
| \bigodot | <code>\bigodot</code> | \bigotimes | <code>\bigotimes</code> |
| \bigoplus | <code>\bigoplus</code> | \biguplus | <code>\biguplus</code> |
| \coprod | <code>\coprod</code> | \bigsqcup | <code>\bigsqcup</code> |
| \bigvee | <code>\bigvee</code> | \bigwedge | <code>\bigwedge</code> |
| \int | <code>\int</code> | \oint | <code>\oint</code> |

Tabela C.9: Simboli promenljive veličine

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

Tabela C.10: Zapisi matematičkih funkcija koji se ispisuju normalnim, uspravnim slovima

| | | | |
|------------------------------------|--|--------------------------------------|--------------------------------------|
| \dots <code>\ldots</code> | \cdots <code>\cdots</code> | \vdots <code>\vdots</code> | \ddots <code>\ddots</code> |
| \forall <code>\forall</code> | \exists <code>\exists</code> | \top <code>\top</code> | \perp <code>\bot</code> |
| \neg <code>\neg</code> | \parallel <code>\parallel</code> | \angle <code>\angle</code> | \triangle <code>\triangle</code> |
| \Re <code>\Re</code> | \Im <code>\Im</code> | ∞ <code>\infty</code> | \emptyset <code>\emptyset</code> |
| \aleph <code>\aleph</code> | i <code>\imath</code> | j <code>\jmath</code> | \hbar <code>\hbar</code> |
| $'$ <code>\prime</code> | ∇ <code>\nabla</code> | ∂ <code>\partial</code> | ℓ <code>\ell</code> |
| \sharp <code>\sharp</code> | \natural <code>\natural</code> | \flat <code>\flat</code> | \surd <code>\surd</code> |
| \clubsuit <code>\clubsuit</code> | \diamondsuit <code>\diamondsuit</code> | \heartsuit <code>\heartsuit</code> | \spadesuit <code>\spadesuit</code> |
| \wp <code>\wp</code> | \backslash <code>\backslash</code> | | |

Tabela C.11: Pomoćni simboli

| | | | |
|--------------------------------|------------------------------------|--|--|
| $($ <code>(</code> | $)$ <code>)</code> | \uparrow <code>\uparrow</code> | \Uparrow <code>\Uparrow</code> |
| $[$ <code>[</code> | $]$ <code>]</code> | \downarrow <code>\downarrow</code> | \Downarrow <code>\Downarrow</code> |
| $\{$ <code>\{</code> | $\}$ <code>\}</code> | \updownarrow <code>\updownarrow</code> | \Updownarrow <code>\Updownarrow</code> |
| \lfloor <code>\lfloor</code> | \rfloor <code>\rfloor</code> | \lceil <code>\lceil</code> | \rceil <code>\rceil</code> |
| \langle <code>\langle</code> | \rangle <code>\rangle</code> | $/$ <code>/</code> | \backslash <code>\backslash</code> |
| $ $ <code> </code> | \parallel <code>\parallel</code> | | |

Tabela C.12: Razdvajajući simboli

| | | | |
|----------------------------------|----------------------------------|--------------------------------------|--------------------------------------|
| $\left($ <code>\left(</code> | $\right)$ <code>\right)</code> | $\left[$ <code>\left[</code> | $\right]$ <code>\right]</code> |
| \uparrow <code>\uparrow</code> | \Uparrow <code>\Uparrow</code> | \downarrow <code>\downarrow</code> | \Downarrow <code>\Downarrow</code> |

Tabela C.13: Veliki razdvajajući simboli

Bibliografija

- [1] ISO/IEC JTC 1 SC 2. *8-bit single-byte coded graphic character sets*. ISO/IEC, 1998.
- [2] ISO/IEC JTC 1 SC 29. JPEG 2000 *image coding system*. ISO/IEC, 2002.
- [3] www.autodesk.com.
- [4] www.corel.com.
- [5] www.ctan.org.
- [6] dviwin.keystone.gr.
- [7] www.gnu.org/software/emacs.
- [8] tug.ctan.org/tex-archive/systems/msdos/emtex.
- [9] www.matf.bg.ac.rs/~janicic/gclc.
- [10] www.gimp.org.
- [11] www.gnuplot.info.
- [12] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, 1994.
- [13] www.adobe.com/products/illustrator/main.html.
- [14] www.imagemagick.org.
- [15] Adobe Systems Inc. *Portable Document Format Reference Manual*. Addison-Wesley, 1993.
- [16] Adobe Systems Inc. *POSTSCRIPT Language Reference*. Addison-Wesley, 1999.
- [17] www.gnu.org/software/ispell.
- [18] www.pdflib.com/download/free-software/jpeg2ps.
- [19] Donald Knuth. *TeXBook*. Reading, Massachusetts: Addison-Wesley, 1984.

- [20] Donald Knuth. *The METAFONTBook*. Reading, Massachusetts: Addison-Wesley, 1986.
- [21] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison Wesley Professional, 1994.
- [22] www.latex2html.org.
- [23] www.lyx.org.
- [24] www.miktex.org.
- [25] www.adobe.com/products/photoshop/main.html.
- [26] www.mackichan.com.
- [27] www.tug.org/teTeX.
- [28] www.winedit.com.
- [29] www.xfig.org.

Gde je šta

`\(` , 50
`\)` , 50
`\-` , 28
`\[` , 50
`\]` , 50
`\\` , 28
`$` , 50
`$$` , 50
`{}` , 9
`~` , 30

abstract okruženje, 21
`\address`, 21
`\afterpage`, 66
afterpage paket, 66
alltt okruženje, 35
AMS-TeX, 6
`\and`, 18
`\appendix`, 20
`\author`, 18

babel paket, 81
`\backmatter`, 20
`\baselineskip`, 29, 30
`\baselinestretch`, 30
beamer paket, 87
`\begin{document}`, 12, 17
bezier paket, 62
`\bf`, 41
`\bfdefault`, 39
bfseries, 39
`\bibitem`, 71
bibliografija
 kreiranje, 71
`\bibliography`, 74
`\bibliographystyle`, 74
bibtex program, 74

`\bigskip`, 30
boksovi, 31
`\Box`, 48

`\cal`, 58
`\caption`, 66
`\cc`, 22
center okruženje, 32
`\centering`, 32
`\centerline`, 32
centriranje teksta, 32
`\chapter`, 19
`\circle`, 61
`\cite`, 72
`\cleardoublepage`, 66
`\clearpage`, 66
`\closing`, 22
cls datoteke, 16
`\columnsep`, 24
CTAN, 91

`\date`, 18, 21, 26
`\DeclareGraphicsRule`, 65
`\DeclareMathAlphabet`, 58
`\def`, 78
description okruženje, 33
dijakritici, 47
`\displaymath`, 50
`\DJ`, 44
`\dj`, 44
`\documentclass`, 12, 15
dokument
 klasa, 15
 preambula, 11, 15
 sadržaj, 21
 struktura teksta, 19
 tekst dokumenta, 11, 17

doslovni ispis, 34
\dotfill, 30
DVI drajver, 9
DVI datoteke, 7
dvostruku prored, 30

\em, 42
\emph, 43
empty opcija, 24
EM_T_EX, 91
\encl, 22
\encodingdefault, 37, 45
\end{document}, 12, 17
\enlargethispage, 29
enumerate okruženje, 33
epic paket, 62
equation, 50
\evensidemargin, 23

\familydefault, 37
fancyheadings paket, 26
\fbox, 67
\figure, 65
\figurename, 67
flushleft okruženje, 32
flushright okruženje, 32
font, 36
 atributi, 36, 37, 40, 41
 definicija, 44
 dodatni, 46
 familija, 36
 javno-dostupni, 8
 kodiranje, 36
 metrički opis, 8
 oblik, 36
 osnovni, 36, 37
 osnovni skup vrsta, 39, 45
 podrazumevani, 37
 promena tekućeg, 40
 serija, 36
 supstitucija, 39
 tekući, 36, 37, 40
 veličina, 36
\fontencoding, 37, 44
\fontfamily, 37
\fontseries, 37

\fontshape, 37
\fontsize, 37
\footheight, 24
\footnote, 69
\footnotesize, 38
formatiranje
 dve kolone, 16
 dvostrano, 16
 pisama, 21
 tabela i slika, 60
formule, 50
 numerisanje, 50
\framebox, 31, 61
\frenchspacing, 30
\frontmatter, 20
fusnota, 69
\fussy, 28

GCLC program, 62
gimp program, 63
gnuplot program, 62
graphicx paket, 63

\headheight, 24
headings opcija, 24
\hfill, 30
\hline, 60
\hspace, 29, 67
\Huge, 38
\huge, 38
\hyphenation, 28

\iffalse, 17
imagemagick program, 63
\include, 17
\includegraphics, 63
indeks
 kreiranje, 70
\indent, 30
\index, 70
\indexname, 70
inputenc paket, 84
\inputencoding, 84
\it, 41
\itdefault, 39
\item, 33

- itemize okruženje, 33
- itshape, 39

- jedinice dužine, 23
- jpeg2ps program, 64

- kôd simbola, 48
- Knut, D., 4
- kodiranje, 44
 - osnovno, 44, 45
 - tekuće, 45
- komanda
 - kreiranje nove, 76
- komenatari, 11, 17
- kreiranje sadržaja, 21

- \label, 68
- Lamport, L., 6
- landscape, 24
- \language, 82
- \LARGE, 38
- \Large, 38
- \large, 38
- \LaTeX , 6
 - \LaTeX 2.09, 6, 16
 - \LaTeX 2 ϵ , 6
 - argumenti komande, 9
 - distribucije, 91
 - dokument, 11
 - dokumentacija, 92
 - klasa dokumenta, 15
 - komande, 9
 - komentari, 11, 17
 - logo, 7, 9
 - makro, 6
 - obrada DVI datoteka, 93
 - obrada \LaTeX datoteka, 92
 - paketi, 16
 - podrška za srpski jezik, 83
 - preambula, 11, 15
 - proširivanje, 76
 - struktura dokumenta, 15
- \left, 52
- lefthyphenmin, 28
- \leftskip, 33
- letter klasa, 21

- ligature, 48
- \line, 61
- \linebreak, 28
- liste, 33
- \listoffigures, 66
- \listoftables, 66

- \mainmatter, 20
- \makebox, 31
- makebst program, 73
- makeidx paket, 70
- \makeindex, 70
- makeindex program, 70
- \makelabels, 22
- \maketitle, 18
- margine, 23
- \marginpar, 24
- \markboth, 24
- \markright, 24
- matematički simboli, 51
 - zagrade, 52
- \mathbf, 58
- \mathcal, 58
- \mathit, 58
- \mathrm, 58
- \mbox, 31
- \mddefault, 39
- mdseries, 39
- \medskip, 30
- METAFONT, 8
- MF datoteke, 8
- MIK \TeX , 91
- minipage okruženje, 31
- moreverb paket, 35
- \multicolumn, 60
- myheadings opcija, 24

- naglašavanje, 42
- \name, 21
- naslov, 18
 - argumenti, 18
 - naslovna strana, 16, 19
 - numerisanje, 19
- naslov celina, 19
- naslovna strana, 19
- neformatirani ispis, 34

`\newcommand`, 76
`\newenvironment`, 79
`\newline`, 28
`\newpage`, 29
`\newtheorem`, 59
`\noindent`, 30
`\nolinebreak`, 28
`\nopagebreak`, 29
`\normalfont`, 37
`\normalsize`, 38
notitlepage opcija, 16
numeracija
 formula, 50
 strana, 27
 tabela i slika, 67
 teorema, 59
numerisanje celina, 19

obeležja, 2
obrada POSTSCRIPT datoteka, 94
`\oddsidemargin`, 23
okruženje
 kreiranje novog, 79
onecolumn opcija, 16
oneside opcija, 16
`\onslide`, 89
`\opening`, 22
orijentacija papira
 horizontalna, 24
 vertikalna, 24
otherlanguage, 82
`\oval`, 62

`\pagebreak`, 29
`\pagenumbering`, 27
`\pageref`, 68
`\pagestyle`, 24
paket
 kreiranje novog, 79
paketi, 16
`\par`, 30
`\paragraph`, 19
`\parbox`, 31
`\parindent`, 30
`\parskip`, 29
`\part`, 20

`\pause`, 89
picture, 61
PK datoteke, 8
plain opcija, 24
pokretni objekti, 65
poravnanje, 27
 centrirano, 32
 desno, 32
 levo, 32
portrait, 24
prazan simbol, 9
preambula, 11, 15
prelom
 reda, 27--29
 strane, 27, 29
prezentacije, 87
`\printindex`, 70
prored, 29
 horizontalni, 29
 vertikalni, 30
`\ProvidesPackage`, 80
`\ps`, 22
psfrag paket, 64
pstricks paket, 62
`\put`, 61

quotation okruženje, 32
quote okruženje, 32

`\raggedleft`, 32
`\raggedright`, 32
rastavljanje reči, 28
razmak, 30
 između pasusa, 30
 između redova, 30
`\ref`, 68
referisanje, 68
 na delove teksta, 68
 na objekte, 68
`\renewcommand`, 78
`\right`, 52
righthyphenmin, 28
`\rightskip`, 33
`\rm`, 41
`\rmdefault`, 39
rmfamily, 39

`\rule`, 79

sažetak, 21

`\sc`, 41

`\scdefault`, 39

`\scriptsize`, 38

`scshape`, 39

`\section`, 19

`\selectfont`, 44

`\selectlanguage`, 82

`\seriesdefault`, 37

`\setcounter`, 19

`\setlength`, 23

`\sf`, 41

`\sfdefault`, 39

`sffamily`, 39

`\shapedefault`, 37

`\signature`, 21

skraćenice, 30

`\sl`, 41

slajdovi, 87

`\sldefault`, 39

slike

- uključivanje u dokumente, 63

`\sloppy`, 28

slova

- osnovni skup veličina, 37
- veličina, 37

`slshape`, 39

`\smallskip`, 30

`\special`, 63

specijalni simboli, 47

strana

- dno strane, 23, 24
- numeracija, 24
- veličina, 16
- zaglavlje strane, 23, 24

`\stretch`, 30

STY datoteke, 16

`\subparagraph`, 19

`\subsection`, 19

`\subsubsection`, 19

tabele, 60

`\table`, 65

`\tablename`, 67

`\tableofcontents`, 66

`\tableofcontents`, 21

tabular, 60

tekst

- doslovni ispis, 34
- neformatirani ispis, 34

tekst-procesori, 1

- interaktivno-grafički, 1
- obeležja, 2
- procesivni, 1

terminator reda, 28

TeX, 91

TeX, 4

- komande, 5
- logo, 7, 9
- procesor, 5

TeX LIVE, 91

`\textbf`, 40

`\textheight`, 23

`\textit`, 40

`\textmd`, 40

`\textnormal`, 40

`\textrm`, 40

`\textsc`, 40

`\textsf`, 40

`\textsl`, 40

`\texttt`, 40

`\textup`, 40

`\textwidth`, 23

TFM datoteke, 8

`\thanks`, 18

thebibliography, 71

`\thechapter`, 26

`\thepage`, 26

`\thesection`, 26

`\thispagestyle`, 27

`\tiny`, 38

`\title`, 18

titlepage okruženje, 19

titlepage opcija, 16

`\topmargin`, 23

`\tt`, 41

`\ttdefault`, 39

`ttfamily`, 39

twocolumn opcija, 16, 24

twoside opcija, 16, 24

`\underline`, 12
`\unitlength`, 61
`\updefault`, 39
`upshape`, 39
`\usefont`, 37
`\usepackage`, 17

`\vector`, 61
veličina strane, 16
`\verb`, 34
verbatim okruženje, 35
višejezička podrška, 81
`\vspace`, 30

xfig program, 62