Универзитет у Београду

Електротехнички факултет



Конфигурабилни преводилац и симулатор архитектура рачунара

Мастер рад

|  |  |
| --- | --- |
| Ментор: | Кандидат: |
| доц. др Милош Цветановић | Бојан Јелача 3106/2014 |

Београд, Август 2016.

Садржај

[Садржај 2](#_Toc459123324)

[1. Увод 4](#_Toc459123325)

[2. Преводилац и симулатор - дефиниције и значења 5](#_Toc459123326)

[2.1. Преводилац 5](#_Toc459123327)

[2.1.1. Лексичка анализа 5](#_Toc459123331)

[2.1.2. Синтаксна анализа 6](#_Toc459123332)

[2.1.3. Семантичка анализа 6](#_Toc459123333)

[2.1.4. Генерисање кода 6](#_Toc459123334)

[2.2. Симулатор 6](#_Toc459123335)

[3. Преглед постојећих решења 7](#_Toc459123336)

[3.1. Превиодиоци 7](#_Toc459123337)

[3.1.1. А-0 систем 7](#_Toc459123338)

[3.1.2. Турбо Паскал 7](#_Toc459123339)

[3.1.3. ГНУ колекција преводилаца 8](#_Toc459123340)

[3.1.4. LLVM 8](#_Toc459123341)

[3.1.5. Visual Studio 9](#_Toc459123342)

[3.2. Симулатори 10](#_Toc459123343)

[3.2.1. Proteus Design Suite 10](#_Toc459123345)

[4. Преглед функционалности 12](#_Toc459123346)

[4.1. Дефинисање архитектуре рачунарских компоненти 12](#_Toc459123347)

[4.1.1. Дефинисање архитектуре меморије 12](#_Toc459123348)

[4.1.2. Дефинисање архитектуре компоненти општег типа 14](#_Toc459123349)

[4.2. Дефинисање рада рачунарских компоненти 15](#_Toc459123350)

[4.3. Дефинисање изгледа рачунарских компоненти 15](#_Toc459123351)

[1. Стил писања текста 16](#_Toc459123352)

[4.4. Организација текста тезе 16](#_Toc459123353)

[4.5. Стил писања, скраћенице, преводи са енглеског 16](#_Toc459123354)

[4.6. Слике, табеле, променљиве у тексту 17](#_Toc459123355)

[4.7. Мерне јединице 19](#_Toc459123356)

[4.8. Програмски код 19](#_Toc459123357)

[4.9. Референце 20](#_Toc459123358)

[4.10. Списак скраћеница, слика, табела 20](#_Toc459123359)

[4.11. Захвалница 20](#_Toc459123360)

[5. Форматирање текста 21](#_Toc459123361)

[5.1. Форматирање текста и стилови у алату *Microsoft Word* 21](#_Toc459123362)

[5.2. Форматирање насловне стране 22](#_Toc459123363)

[5.3. Форматирање текста тезе 23](#_Toc459123364)

[5.3.1. Хијерархија наслова текстуалних целина 23](#_Toc459123365)

[5.3.2. Форматирање текста у оквиру текстуалних целина 24](#_Toc459123366)

[5.3.3. Слике и табеле 24](#_Toc459123367)

[5.3.4. Једначине 24](#_Toc459123368)

[5.3.5. Променљиве 24](#_Toc459123369)

[5.3.6. Набрајање 24](#_Toc459123370)

[5.4. Листа референци (списак литературе) 25](#_Toc459123371)

[5.5. Списак скраћеница, слика и табела 26](#_Toc459123372)

[5.6. Прилози 26](#_Toc459123373)

[5.7. Садржај 26](#_Toc459123374)

[6. Литература 28](#_Toc459123375)

[7. Списак скраћеница 29](#_Toc459123376)

[8. Списак слика 30](#_Toc459123377)

[9. Списак табела 31](#_Toc459123378)

[Подешавање стилова текста за *Word* 2003 32](#_Toc459123379)

[Подешавање изгледа странице 32](#_Toc459123380)

[Форматирање основног текста и наслова хијерархијски уређених текстуалних целина 32](#_Toc459123381)

[Основни текст 34](#_Toc459123382)

[Наслов поглавља 36](#_Toc459123383)

[Наслов потпоглавља 38](#_Toc459123384)

[Наслов одељка 39](#_Toc459123385)

[Наслов пододељка 39](#_Toc459123386)

[Слике 40](#_Toc459123387)

[Позиционирање слике 40](#_Toc459123388)

[Назив и нумерација слике 41](#_Toc459123389)

[Табеле 41](#_Toc459123390)

[Позиционирање и изглед табеле 41](#_Toc459123391)

[Назив и нумерација табеле 41](#_Toc459123392)

[Једначине 41](#_Toc459123393)

[Набрајање 42](#_Toc459123394)

[Садржај 46](#_Toc459123395)

[Списак литературе 47](#_Toc459123396)

[Форматирање наслова за садржај, списак литературе (скраћеница, слика, табела) 47](#_Toc459123397)

[Наслови хијерархијски уређених целина у прилозима централном тексту тезе 48](#_Toc459123398)

[Први ниво наслова у прилозима 48](#_Toc459123399)

[Други ниво наслова у прилозима 48](#_Toc459123400)

[Трећи ниво наслова у прилозима 49](#_Toc459123401)

[Четврти ниво наслова у прилозима 49](#_Toc459123402)

1. Увод

Тема овог рада је самостална израда софтверске апликације чија је сврха да кориснику омогући дефинисање произвољног рачунарског система за који онда добија преводилац и симулатор.

У поглављу 2 је дат кратак теоријски осврт на основне елементе теме рада. Дате су кратке дефиниције и теоријска објашњења појмова *преводилац* и *симулатор*.

У области преводилаца постоје бројна, већ развијена решења. Исто важи и за област симулатора. Међутим, оно што ово решење издваја од осталих јесте могућност да кориник сам дефинише целокупан систем, као и да на различите начине прати извршавање кода на свом систему. Детаљнији преглед постојећих решења и разлике реализованог решења у односу на постојећа дати су у поглављу 3.

Сам рад представља надоградњу дипломског рада !Наслов рада! који је кориснику омогућавао да дефинише само архитектуру рачунара. Апликација која је предмет и циљ овог рада омогућава кориснику да, поред архитектуре рачунара, дефинише и архитектуре осталих рачунарских компоненти, начине на који компоненте међусобно интерагују, као и изглед сваке од компоненти. Након што то уради, корисник добија преводилац за језик дефинисане архитектуре рачунара и симулатор на коме може да покреће програмски код. Приликом покретања кода, корисник може да прати стања свих компоненти система, да прати вредности сигнала и магистрала које повезују компоненте, да дефинише фреквенцију рада система итд. Детаљан преглед свих корисничких функционалности заједно са корисничким упуством дат је у поглављу 4.

Основна намена апликације је едукативна и експериментална. У едукативном смислу, апликација је намењена наставницима како би лакше пренели знање студентима везано за неку конкретну архитектуру рачунарског система, као и како би омогућили студентима лакше разумевање теоријских основа рачунарских система. Такође, едукативна сврха апликације је и обучавање студената за пројектовање рачунарских система. У експерименталном смислу, апликација се може примењивати за тестирање нових идеја у пројектовању рачунарских система, пре него што се она хардверски реализују, како би се одмах виделе основне предности и мане идеја. О овој теми биће више речи у поглављу 5.

Апликација је реализована коришћењем програмског језика Ц# (енг. *С#*), радног оквира *.NET 4.5* компаније Мајкрософт (енг. *Microsoft*). За реализацију графичког интерфејса коришћено је окружење *Windows forms.* За реализацију функционалног дела апликације, коришћено је неколико спољашњих библиотека. Детаљан опис целокупне реализације апликације са примерима програмског кода налази се у поглављима !Списак поглавља!.

У последњем поглављу, изведен је закључен и дате смернице како се реализована апликација може проширивати и надограђивати.

1. Преводилац и симулатор - дефиниције и значења

У овом поглављу дат је кратак теоријски осврт на термине *преводилац* и *симулатор* у домену рачунарске технике. Дефинисана су њихова значења, наведене намене и истакнуте разлике.

Иако се ова два појма често спомињу и реализују заједно, суштински су то два потпуно различита и независна термина. Стога, преводилац може да постоји потпуно независно од симулатора, и обрнуто. Имајући то у виду, софтверску реализацију преводиоца је могуће извести потпуно независно и за постојање преводиоца није неопходно постојање симулатора. Наравно, важи и обрнуто - реализацију симулатора је могуће извести потпуно независно и за постојање симулатора није неопходно постојање преводиоца.

* 1. Преводилац

Преводилац представља компјутерски програм или низ програма који трансформише код написан у неком програмском језику у машински код који се након тога може извршавати на циљној машини. Улаз преводиоца је датотека или скуп датотека које садрже код написан у програмском језико. Преводилац као излаз даје датотеку која се може извршити на циљној машини.

Постоји неколико фаза превођења кода написаног у програмском језику у машински код:

* лексичка анализа
* синтаксна анализа
* семантичка анализа
* генерисање кода

2. 1. 1. Лексичка анализа

Лексичка анализа представља анализирање програмског кода током којег се код трансформише у листу токена. Сваки токен представља низ карактера коме се додаје специфично значење. Тако генерисани токени представљају почетно стање за синтаксну анализу. Програм или део програма који обавља лексичку анализу обично се назива лексер.

Током лексичке анализе, програмски код пролази кроз низ провера на основу предефинисаног скупа правила, односно граматике програмског језика. На основу граматике, лексер, на основу граматике, групише програмски код на тај начин креирајући токене. Тако креирани токени су делови програмског кода који представљају смислену целину.

У случају да се у оквиру лексичке анализе пронађу делови кода који ни на који начин не могу да се уклопе у граматику програмског језика, лексер обично прекида рад преводиоца. Типичне грешке које се у овој фази могу открити су употреба непостојећих кључних речи, употреба непостојећих симболаи слично.

* + 1. Синтаксна анализа

Друга фаза превођења програмског кода је синтаксна анализа или парсирање. Програм или део програма који извршава синтаксну анализу се најчешће назива парсер.

У овој фази, низ токена добијен од лексера се трансформише у синтаксно стабло, које представља улаз семантичке анализе. Типична репрезентација синтаксног стабла је стабло чији сваки чвор дефинише операцију коју треба извршити, а деца тог чвора представљају аргументе операције.

Током синтаксне анализе могу пронаћи грешке као што су некомплетне наредбе, погрешан распоред симбола операција и операнада и слично. Уобичајено је да се, у случају појаве синтаксни грешака не прелази на фазу семантичке анализе, већ се превођење зауставља.

* + 1. Семантичка анализа

У процесу семантичке анализе користи се синтаксно стабло да би се проверила семантичка доследност програмског кода са дефинисаним програмским језиком. Такође, прикупљају се информације о типовима које су од користи за генерисање кода.

Битан део семантичке анализе је проверавање типове. У овом делу, преводилац проверава да се тип сваког операнда поклапа са операцијом. На пример, многи програмски језици захтевају да индекс низа буде цео број. У случају да се као индекс низа не користи цео број.

* + 1. Генерисање кода

Последња фаза превођења је генерисање кода. У овој фази се на основу информација из синтаксног стабла и знања о машинском језику генерише машински код који се онда може извршавати на циљној машини. Тако генерисани код се уписује у меморију и спреман је за извршавање.

* 1. Симулатор

Симулатор је рачунарски програм који симулира извршавање програма за друге рачунаре или рачунарске системе.

Основна сврха симулатора јесте евалуација и валидација неке новодефинисане архитуктере рачунара или рачунарског система. Такође, симулатори се могу користити и у процесима обучавања будућих корисника неког рачунарског система.

1. Преглед постојећих решења

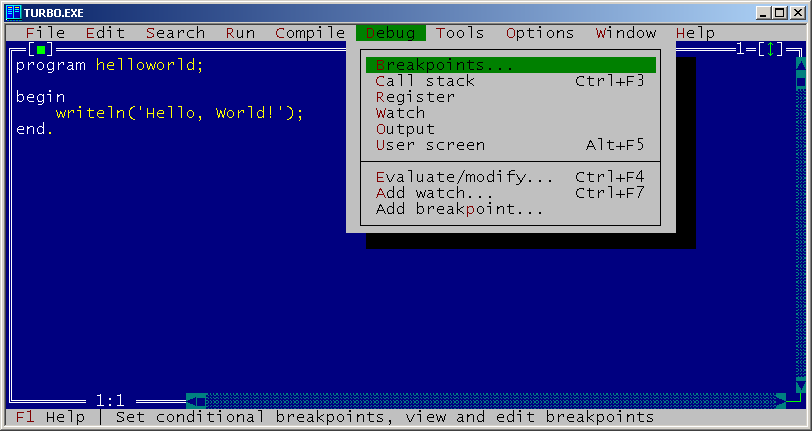
Постоји велики број рачунарских преводилаца и симулатора. У овом поглављу дат је кратак преглед неколико историјски најважнијих преводилаца и симулатора.

* 1. Превиодиоци
     1. А-0 систем

А-0 систем (аритметички језик - верзија 0) написан је од стране америчке научнице Грејс Хопер 1951. године. Циљни рачунар био је *UNIVAC I*. Ово је први преводилац икада развијен. Чешћа намена овог система је било уписивање програма у меморију и повезивање програма. Програми су се састојали од потпрограма и аргумената за потпрограме и по томе је језик највише личио на асемблерске језике. Овај преводилац је тако дефинисане програме преводио директно у машински код.

* + 1. Турбо Паскал

Турбо Паскал је развојни систем који укључује преводилац и графичко развојно окружење за програмски језик Паскал. Развила га је компанија Борланд 1983. године. Овај алат је значајан јер је један од првих који је укључивао графичко развојно окружење, које је данас уобичајени део сваког озбиљног система за развој софтвера. На наредној слици приказан је изглед графичког окружења алата Турбо Паскал.



Слика 3.1.1. Графичко окружење алата Турбо Паскал

* + 1. ГНУ колекција преводилаца

ГНУ колекција преводилаца је систем преводилаца развијен од стране компаније *GNU Project* који подржава различите програмске језике. Настао је 1987. године. Систем је написан у програмском језику Ц, а касније верзије су написане у језику Ц++. Одиграо је веома битну улогу у развоју бесплатног софтвера.

Иницијална верзија је подржавала само језик Ц, а врло брзо развијена је и верзија за Ц++. Данас подржава велики број језика, међу којима:

* Ц
* Ц++
* Објектни Ц
* Објектни Ц++
* Фортран
* Јава
* Ада
* *Go*

Подржава још већи број архитектура, од којих су најбитније:

* *ARM*
* *AVR*
* *SPARC*
* *x86*
* *x64*
* *MIPS*
* *Intel i960*
  + 1. LLVM

*LLVM додати у табелу скраћеница (Low Level Virtual Machine)* јe колекција модуларних и вишеструко употребљивих преводилаца. Писан је у језику Ц++ и подржава оптимизацију у време превођења, повезивања и извршавања. Настао је 2003. године на Универзитету Илиној.

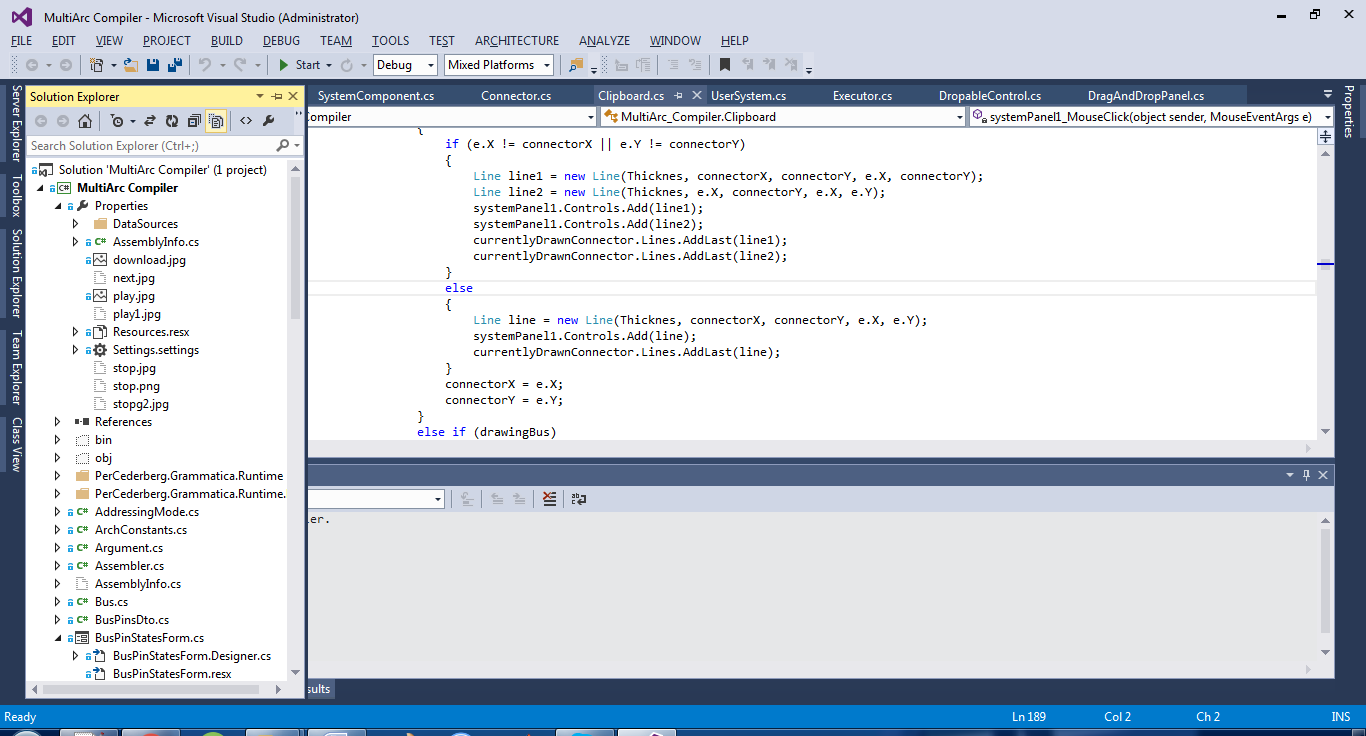
Између осталих, *LLVM* подржава следеће језике:

* *ActionScript*
* Ада
* Ц#
* Делфи
* Фортран
* Јава
* Пајтон
* Руби
* Скала
  + 1. Visual Studio

*Visual Studio* је интегрисано развојно окружење развијено од стране компаније Мајкрософт. Превасходна намена овог окружења је развој апликација за оперативни систем *Microsoft Windows*, као веб сајтова, веб апликација и веб сервиса. Прва верзија објављена је 1997. године, а тренутно је актуелна верзија 2015.

Укључује едитор који подржава аутоматско завршавање наредби као и аутоматско преуређивање кода. Подржава дебаговање на нивоу програмског кода и на нивоу машинског кода. Такође, укључује многе уграђене графичке алате за развој корисничког интерфејса, веб дизајн, дијаграме класа и дизајн шема база података.

На слици 3.1.2. приказан је типичан изглед окружења *Visual Studio*.



Слика 3.1.2. Типичан изглед екрана у алату *Visual Studio*

Ово окружење подржава различите програмске језике као што су:

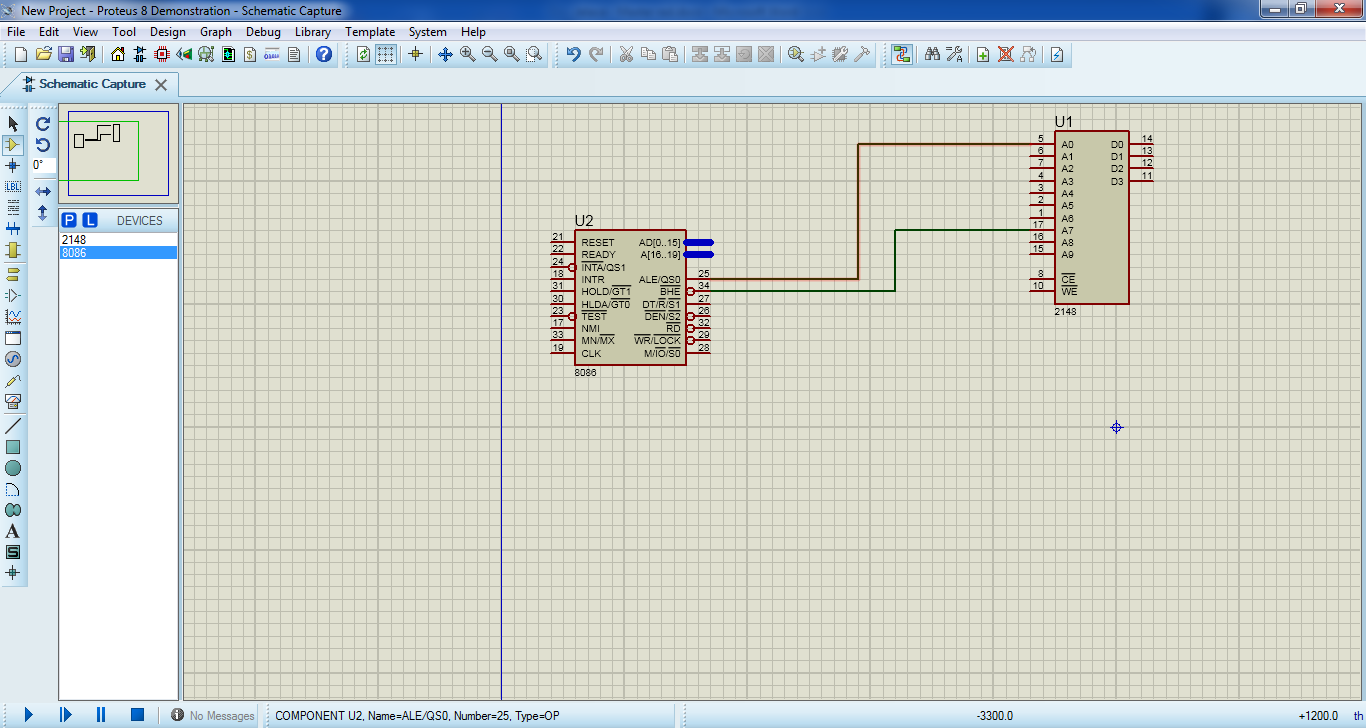
* Ц
* Ц++
* *Visual Basic*
* Ц#
* Ф#
* *XML*
* *HTML*
* *JavaScript*
* *CSS*
  1. Симулатори

У овом делу пажња неће бити посвећена прегледу већег броја симулатора развијених кроз исторују, јер је њихов број заиста огроман. Исто тако, сви симулатори развијани до сада су имали најразличитије намене, од симулирања рада авиона све до симулирања рада специфичног рачинарског система. Уместо тога, пажња ће бити посвећена окружењу *Proteus Design Suite*, развијеног од стране британске компаније *Labcenter Electronics Ltd*, које је било један од основних узора апликације која је тема овог рада.

* 1. 1. Proteus Design Suite

*Proteus Design Suite* је алат који је развила британска компанија *Labcenter Electronics Ltd.* 1988. године. Алат се покреће на оперативном систему *Microsoft Windows.*

Ово је алат за аутоматизацију електронског дизајна који укључује дефинисање шеме система, симулацију рада микроконтролера и бројне модуле штампаних плоча. На слици 3.2.1. приказан је део алата *Proteus Design Suite* за дефинисање шеме система.



Слика 3.2.1. Дефинисање шеме система у алату *Proteus Design Suite*

Симулација рада микроконтролера се рализује тако што као улаз добије датотеку са машинским кодом. Након тога, симулација се обавља паралелно са било којом електронском компонентом повезаном било аналогним, било дигиталним везама. То омогућује да се користи за широк спектар прототипова из области:

* контрола мотора
* контрола температуре
* дизајн корисничког интерфејса

Пошто не захтева постојање никаквог хардвера, има веома широку примену као алат за учење и вежбање.

Покрива широк спектар технологија:

* *Microchip Technologies*
* *Atmel AVR*
* *Arduino*
* *NXP*
* *Texas Instruments*

Алат *Proteus Design Suite* подржава и дизајн штампаних плоча. То је могуће урадити тако што се прво дефинише унутрашња шема штампане плоче, а затим и спољашњи изглед и пинови.

1. Преглед функционалности

У овом поглављу дат је детаљан преглед свих функционалности које апликација која је предмет овог рада укључује, заједно са упутством за реализовање ових функционалности. Пошто је део функционалности развијен као тема дипломског рада !наслов рада! у овом поглављу пажња се посвећује искључиво функционалностима које нису биле део дипломског рада.

Функционалности које ова апликација додаје на функционалности апликације која је развијена као тема дипломског рада могу се поделити на:

* дефинисање архитектуре рачунарских компоненти
* дефинисање рада рачунарских компоненти
* дефинисање изгледа рачунарских компоненти
* пројектовање рачунарског система
* симулирање рада рачунарског система
  1. Дефинисање архитектуре рачунарских компоненти

Апликација разликује три типа рачунарских компоненти:

* процесор
* меморија
* компоненте општег типа

Процесор је централна компонента рачунарског система која може покретати машински код добијен превођењем програмског кода. Овом типу компоненте се неће посвећивати већа пажња, пошто је детаљно описан у дипломском раду.

У наредним одељцима, пажња ће бити посвећена меморији и компонентама општег типа.

* + 1. Дефинисање архитектуре меморије

Меморија је компонента чија је намена складиштење веће количине података. Она не може извршавати програмски код и нема регистарски фајл, већ само меморијске локације. Корисник може дефинисати протокол по коме се подаци дохватају из меморије и протокол по коме се подаци уписују у меморију. Детаљан опис дефинисања овог протокола дат је у потпоглављу 4.2.

Услов за рализацију функционалности дефинисања архитектуре меморије је претходно отворен пројекат.

Да би дефинисао архитектуру меморије, корисник треба из главног менија апликације да изабере опцију *Architecture*, из менија који се отвори опцију *Load* и затим опцију *Memory*. Све ово се може остварити и истовременим притиском тастера *CTRL* и *М* на тастатури. Након тога, отвара се дијалог за избор датотеке у којој се налази опис архитектуре меморије.

Уколико већ није, корисник може ван апликације да дефинише датотеку која описује архитектуру меморије. Ради се о датотеци написаној у *XML* језику чији је детаљан опис дат у наставку текста.

Цела датотека треба да садржи један главни *XML* чвор *memory.*Унутар овог чвора налазе се чворови који дефинишу својства меморије:

* **Чвор *name* -** Вредност овог чвора дефинише име меморијске компоненте
* **Чвор *filename* -** Вредност овог чвора дефинише локацију и име датотеке у којој се налази код који дефинише рад меморије, о чему ће више речи бити у потпоглављу 4.2. Уколико се овај чвор не наведе, узима се датотека са именом истим као име меморијске компоненте, са екстензијом *.cs* и смешта у директоријум */Data/Memories* унутар пројектног фолдера.
* **Чвор *size* -** Вредност овог чвора дефинише укупну величину меморије у бајтовима
* **Чвор *au*** *-* Вредност овог чвора дефинише величину адресне јединице
* **Чвор *rom\_start* -** Вредност овог чвора дефинише почетну адресу РОМ меморије
* **Чвор *rom\_end* -** Вредност овог чвора дефинише крајњу адресу РОМ меморије
* **Чвор *ram\_start*** *-* Вредност овог чвора дефинише почетну адресу РАМ меморије
* **Чвор *ram\_end* -** Вредност овог чвора дефинише крајњу адресу РАМ меморије
* **Чвор *init\_file*** **-** Вредност овог чвора дефинише локацију датотеке у којој се налази иницијални садржај меморије који корисник може да дефинише
* **Чвор *storage\_file*** **-** Вредност овог чвора дефинише локацију датотеке у којој се чува садржај меморије током извршавања програма
* **Чвор *dimensions* -** Вредност овог чвора дефинише величину визуелне репрезентације меморије. Више о овоме биће у потпоглављу 4.3.
* **Чвор *ports* -** Вредност овог чвора дефинише портове које меморија садржи. Овај чвор има своје подчворове од којих сваки представља назив порта. Подчворови сваког овог чвора дефинишу његова својства:
  + **Чвор *name* -** Вредност овог подчвора дефинише име порта
  + **Чвор *side* -** Вредност овог подчвора дефинише страну на којој се налази порт. Може имати вредности *left, rigth, up* и *down.*
  + **Чвор *type* -** Вредност овог подчвора дефинише тип порта. Може имати вредности *in, out* и *inout.*
  + **Чвор *number* -** Вредност овог подчвора дефинише величину порта у битовима, тј. број пинова порта.
* **Чвор *design* -** Овај чвор се користи при дефинисању изгледа меморије. О овоме ће више речи бити у потпоглављу 4.3.

Након што је учитао датотеку која описује архитектуру меморије, апликација сама генерише датотеку специфицирану чвором *filename*, уколико она већ није генерисана. Ова датотека дефинише код који описује рад меморије. Више речи о томе биће у потпоглављу 4.2.

* + 1. Дефинисање архитектуре компоненти општег типа

Компоненте општег типа су компоненте чија је намена обављање посла који корисник дефинише. Оне не могу извршавати програмски код и немају меморијске локације, већ само регистарски фајл. Детаљнији опис дефинисања посла који компонента обавља дат је у потпоглављу 4.2.

Услов за рализацију функционалности дефинисања архитектуре компоненти општег типа је претходно отворен пројекат.

Да би дефинисао архитектуру компоненте општег типа, корисник треба из главног менија апликације да изабере опцију *Architecture*, из менија који се отвори опцију *Load* и затим опцију *Other Component*. Све ово се може остварити и истовременим притиском тастера *CTRL* и *T* на тастатури. Након тога, отвара се дијалог за избор датотеке у којој се налази опис архитектуре компоненте општег типа.

Уколико већ није, корисник може ван апликације да дефинише датотеку која описује архитектуру компоненте општег типа. Ради се о датотеци написаној у *XML* језику чији је детаљан опис дат у наставку текста.

Цела датотека треба да садржи један главни *XML* чвор *othercomponent.*Унутар овог чвора налазе се чворови који дефинишу својства компоненте општег типа:

* **Чвор *name* -** Вредност овог чвора дефинише име компоненте
* **Чвор *filename* -** Вредност овог чвора дефинише локацију и име датотеке у којој се налази код који дефинише рад компоненте општег типа, о чему ће више речи бити у потпоглављу 4.2.Уколико се овај чвор не наведе, узима се датотека са именом истим као име компоненте, са екстензијом *.cs* и смешта у директоријум */Data/Other* унутар пројектног фолдера.
* **Чвор *dimensions* -** Вредност овог чвора дефинише величину визуелне репрезентације компоненте. Више детаља о овоме биће изнето у потпоглављу 4.3.
* **Чвор *registers* -** Вредност овог чвора дефинише регистре које компонента садржи. Сваки подчвор представља дефиницију једног регистра. Његови подчворови дефиншу својства регистра:
  + **Чвор *size* -** Вредност овог подчвора дефинише величину регистра у битовима.
  + **Чвор *name* -** Вредност овог подчвора дефинише име регистра
  + **Чвор *value* -** Вредност овог подчвора дефинише почетну вредност регистра
  + **Чвор *part* -** Ово је посебан тип подчвора који служи за дефинисање делова регистра, односто подрегистара. Сваки подчвор овог типа дефинише један део регистра који се може користити у програмском коду.Дефинисање дела неког регистра се обавља кроз подчворове подчвора *part*:
    - **Чвор *start* -** Овај подчвор дефинише бит од кога део регистра почиње
    - **Чвор *end* -** Овај подчвор дефинише бит на којем се завршава део регистра
    - **Чвор *name* -** Овај подчвор дефинише име, односно псеудоним дела регистра који се може користити у програмском коду
* **Чвор *ports* -** Вредност овог чвора дефинише портове које компонента садржи. Овај чвор има своје подчворове од којих сваки представља назив порта. Подчворови сваког овог чвора дефинишу његова својства:
  + **Чвор *name* -** Вредност овог подчвора дефинише име порта
  + **Чвор *side* -** Вредност овог подчвора дефинише страну на којој се налази порт. Може имати вредности *left, right, up* и *down.*
  + **Чвор *type* -** Вредност овог подчвора дефинише тип порта. Може имати вредности *in, out* и *inout.*
  + **Чвор *number* -** Вредност овог подчвора дефинише величину порта у битовима, тј. број пинова порта.
* **Чвор *design* -** Овај чвор се користи при дефинисању изгледа компоненте. О овоме ће више речи бити у потпоглављу 4.3.

Након што је учитао датотеку која описује архитектуру меморије, апликација сама генерише датотеку специфицирану чвором *filename*, уколико она већ није генерисана. Ова датотека дефинише код који описује рад меморије. Више речи о томе биће у потпоглављу 4.2.

* 1. Дефинисање рада рачунарских компоненти

Да би корисник могао да дефинише рад неке рачунарске компоненте, треба да отвори пројекат и да учита архитектуру те компоненте.

Након што учита архитектуру компоненте, у директоријуму *Data/Memories* или *Data/Other* отвори датотеку са именом компоненте и суфиксом *.cs*, ако није другачије специфицирао својством *filename.* У овој датотеци, постоји предефинисана метода *Cycle,* која као аргумент прихвата компоненту чији рад се дефинише. У случају да се ради о меморији, компонента ће бити типа *Memory,* ау супротном ће бити типа *OtherComponent*. Дефинисање рада процесора је описано у дипломском раду и овде се томе неће посвећивати пажња.

Корисник може да измени тело ове методе и тиме дефинише скуп наредби које компонента циклично обавља. Како би се кориснику олакшао посао, објекат компоненте који је прослеђен као аргумент пружа следеће методе:

* ***GetPort* -** Ова метода прихвата аргумент типа стринг и враћа порт са именом истим као прослеђени аргумент, уколико такав порт постоји
* ***GetPin* -** Ова метода прихвата аргумент типа стринг и враћа пин са именом истим као прослеђени аргумент, уколико такав пин постоји
* ***GetAllPorts* -** Ова метода враћа низ свих портова компоненте
* ***GetAllPins* -** Ова метода враћа низ свих пинова компоненте
* ***Wait*** **-** Ова метода реализована је на три начина. Уколико корисник проследи аргументе типа стринг и цео број, онда ова метода блокира извршавање док порт са именом истим као први аргумент не добије вредност другог аргумента. У случају да корисник проследи аргументе типа стринг и *PinValue*, метода блокира извршавање док пин са именом истим као први аргумент не добије вредност другог аргумента, која може бити из опсега *TRUE, FALSE* или *HIGHZ.*Ако корисник проследи један целобројни аргумент, метода блокира извршавање онолико откуцаја системског сата колико је специфицирано артументом методе
* ***WaitForRisingEdge* -** Ова метода прихвата аргумент типа стринг који представља име пина. Метода блокира извршавање док се не појави узлазна ивица на пину специфицираном са аргументом методе, ако такав постоји
* ***WaitForFallingEdge* -** Ова метода прихвата аргумент типа стринг који представља име пина. Метода блокира извршавање док се не појави силазна ивица на пину специфицираном са аргументом методе, ако такав постоји
* **Дохватање податка са меморијске локације** *-* Уколико се дефинише рад меморије, корисник може прочитати вредност са било које локације у меморији. То може учинити са *memory[address],* где је*memory* аргумент прослеђен методи *Cycle*, а *address* представља адресу меморијске локације са које се жели дохватити податак
* ***GetRegister* -** Ова метода прихвата стринг као аргумент и враћа регистар специфициран аргументом, уколико такав постоји. Може се позвати само у случају да се дефинише рад компоненте општег типа.

Након што је дефинисао рад компоненте, корисник треба да поново учита архитектуру те компоненте, на начин описан у потпоглављу 4.1, да би апликација поново учитала код који описује рад компоненте. Алтернативни начин за остваривање овога је избором опције *Architecture* из главног менија, а затим кликом на опцију *Recompile.* То се може остварити и притиском тастера *F3* на тастатури.

* 1. Дефинисање изгледа рачунарских компоненти

Кориснику апликације пружена је могућност да дефинише визуелну репрезентацију сваке компоненте. Услов за реализацију ове функционалности је отворен пројекат и дефинисана архитектура компоненте.

Дефисање изгледа компоненте може се остварити на три начина:

* **подразумевани начин** - корисник дефинише само димензије и позиције портова
* **кроз графички интерфејс** - корисник користи графички интерфејс специјално за намену дефинисања изгледа компоненте
* **писањем наредби у коду** - корисник дефинише код који се извршава при исцртавању компоненте
  + 1. Подразумевани начин дефинисања изгледа рачунарске компоненте

За овај начин, већина својстава је предефинисана.

Корисник треба да, при дефиницији архитектуре компоненте (као што је описано у потпоглављу 4.1), дефинише локацију сваког порта. Поред тога, корисник треба да дефинише чвор *dimensions*, којим су обухваћена два подчвора:

* **Чвор *height* -** Вредност овог подчвора дефинише висину компоненте у пикселима
* **Чвор *width* -** Вредност овог подчвора дефинише ширину компоненте у пикселима

Када корисник на овај начин дефинише позиције портова и димензије компоненте, компонента ће се аутоматски исцртати. Компонента ће бити беле боје, са црним оквирима и портовима. Уколико се, при дефиницији система, о чему ће више речи бити у потпоглављу 4.4, изабере овако дефинисана компонента, њени оквири ће постати зелени.

* + 1. Дефинисање изгледа рачунарске компоненте кроз графички интерфејс
  1. Пројектовање рачунарског система

1. Стил писања текста

У овоме поглављу биће дате смернице везане за стил писања текста тезе, начин организације текста у поглавља, израду слика и табела, организацију прилога основном тексту тезе и слично.

* 1. Организација текста тезе

Основни текст тезе организован је у хијерархијски уређене целине. Разликоваћемо четири нивоа хијерархије: поглавља, потпоглавља, одељци и пододељци. Основни текст тезе треба да садржи увод, централна поглавља и закључак. Поред основног текста, теза мора имати и садржај, списак референци (литературе), списак скраћеница, списак слика и списак табела, а може садржати и прилоге у виду програмског кода или неких додатних објашњења. Поред текста, теза типично садржи и известан број слика и табела, како у основном тексту, тако и у прилозима.

Тезу треба отпочети уводом, у коме је неопходно објаснити у вези са којом облашћу је тема тезе, какви су претходни резултати истраживања у тој области и осврнути се на допринос који доноси дата теза. Последњи део увода треба да садржи информације о организацији саме тезе – броју поглавља од којих се она састоји и кратак осврт на садржај сваког од тих поглавља. Тезу треба завршити закључком, у коме ће се направити резиме доприноса тезе и најважнијих закључака.

Све текстуалне целине (поглавља, потпоглавља, итд) треба да имају структуру есеја. Препоручљиво је да свака текстуална целина отпочне уводним параграфом, које ће са бар по једном реченицом најавити све делове те целине. Такође, поглавље би требало завршити параграфом у коме ће се резимирати најважнији закључци поглавља.

* 1. Стил писања, скраћенице, преводи са енглеског

Треба водити рачуна да стил писања буде јасан и конзистентан. Стога треба користити краће реченице, а дуже реченице је боље поделити на неколико краћих. Сваки део текста мора бити јасан писцу. Недопустиво је писати нешто што самом аутору није јасно.

Приликом писања основног текста тезе, текст треба поделити у параграфе. Параграф представља скуп реченица које представљају једну логичку целину. Параграф не сме да буде прекратак (на пример, само једна реченица). Уобичајено, параграф не би смео да садржи мање од три реченице.

Треба водити рачуна и о граматичкој исправности текста. Текст треба да буде писан формално, а никако се не сме писати у првом лицу једнине. У оним ситуацијама у којима аутор жели да нагласи сопствени допринос, може се користити прво лице множине. Друго лице и једнине и множине такође се не сме користити у тексту.

Реченице са енглеског треба преводити по смислу, а не дословно. Проверити да ли за неке појмове већ постоје устаљени термини на српском језику. Уколико то није случај, треба по смислу превести енглески термин за тај појам, и користити тај исти превод у целом тексту тезе. Недопустиво је истовремено користити два различита нова превода за исти појам. Уколико се у српском одомаћено користи енглески термин за неки појам, треба га писати онако како се изговара, „по Вуку“. На пример, могу се користити термини као веб страна, свич итд. Ако је тешко наћи добар превод или енглески термин није устаљен у оквиру српског језика, енглески оригинал се мора писати *курзивом*. Никако не треба мешати српски и енглески, нпр. не треба писати „у *Microsoft Word*-у“, већ „у алату *Microsoft Word*“.

Треба водити рачуна да се сви појмови, променљиве и скраћенице дефинишу пре него што се користе. У ситуацији када се за неки појам могу користити два термина на српском језику, пожељно је одабрати један који ће се користити током писања тезе. Скраћенице је приликом првог појављивања неопходно објаснити у загради. На пример, TCP (*Transmission Control Protocol*), ARP (*Address Resolution Protocol*),…. Треба уводити променљиве за све вредности које се помињу у тексту. Нпр. „процесор ће издвојити првих *pb*=8 бита“. Ово олакшава касније позивање на променљиву и спречава мешање појмова који имају исте вредности.

Уколико аутор одлучи да неку врсту објекта или појма пише масним словима, или подвуче, то мора чинити конзистентно за све објекте те врсте или појма. На пример, аутор може да одлучи да функције из програма које се описују у тексту тезе пише масним словима, али онда треба да пише масним словима све такве функције, а не неке да пише масним словима, неке да подвуче, а неке да остави у обичном фонту.

* 1. Слике, табеле, променљиве у тексту

Све слике, графици и табеле морају бити урађене на исти начин. Текст који се појављује на сликама треба бити на српском језику уколико је то могуће. Слике цртати у адекватном алату. На пример, *Microsoft Visio* је погодан за представљање схема, дијаграма токова и сл. Избор алата се препушта студентима и менторима, при чему је важно да слике у раду буду јасне и обезбеђују квалитетан приказ у одштамапном примерку рада.

У случају да постоји више слика међусобно повезаних (на пример, дијаграм тока неког алгоритма који не стане на једну страну), неопходно је кроз називе слика јасно назначити о ком делу слике се ради (на пример, Слика 1.1 - Назив слике (први део), Слика 1.2 - Назив слике (други део),...). При томе, у случају потребе треба обезбедити и повезнице на самим сликама које олакшавају само тумачење слика (на пример, ако је у питању дијаграм тока, обезбедити маркере за линије које прелазе између слика, тако да се недвосмислено зна која линија на наредној слици је наставак линије са претходне слике).

Графици су врста слика, те треба да имају исти формат као и остале слике у тексту. Обавезно треба навести називе оса, а пожељно је ставити и наслов графика. У случају да је на графику приказано више линија потребно је приказати и легенду која омогућава разликовање линија на графику, односно тумачење графика. Величина фонта мора бити таква да су све ознаке лако читљиве. Водити рачуна о томе да се криве на графику не смеју разликовати само по боји, већ и по изгледу линије, јер разлика мора бити видљива и када се мастер рад одштампа црно-бело. Слика 3.3.1. представља пример неправилног приказа графика пошто би у црно-белом приказу легенда графика била неупотребљива и график се не би могао правилно тумачити. Слика 3.3.2. показује коректно приказан график где су маркери коришћени за додатно разликовање линија поред боје. Легенда графика са слике 3.3.2. би се и у црно-белом приказу могла тумачити и могле би се разликовати линије на графику. Уместо маркера (или уз маркере) се може користити и различит тип приказа линије, на пример, пуна линија, испрекидана линија и сл.

nekorektnaslika

Слика 3.3.1. Неправилан приказ графика

korektnaslika

Слика 3.3.2. Правилан приказ графика

Слике и табеле се нумеришу у оквиру потпоглавља, нпр. прва табела у потпоглављу 2.3. има редни број 2.3.1. Свака слика и табела морају бити описане и прокоментарисане у тексту, макар и кратко. Свака табела и слика морају имати назив. Не може се оставити само нумерација слике/табеле без назива, нпр. да се напише само Слика 2.1.

Променљиве се у тексту пишу *курзивом*. Све променљиве треба дефинисати приликом првог појављивања у једначини. Списак променљивих и њихове дефиниције треба да буду дати у другом прилогу документа.

Табела 3.3.1. представља образац за изглед табеле.

Табела 3.3.1. Образац за изглед табеле

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **величина 1** | **величина 2** | **величина 3** | **величина 4** | **величина 5** | **величина 6** | **величина 7** | **величина 8** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

* 1. Мерне јединице

Приликом писања резултата мерења и сл., у приказу резултата треба користити основне и изведене SI јединице мерених/приказиваних величина [5]. При томе SI јединице писати латиницом. На пример, ознаку за центиметар треба писати cm, а не цм. Избегавати коришћење енглеског термина *per*, тј. не треба писати *p* уместо косе црте. На пример, уколико је приказивани проток података сто мегабита у секунди треба користити ознаку Mb/s уместо Mbps ознаке која се користи у литератури на енглеском језику.

* 1. Програмски код

Уколико се уз тезу прилаже програмски код, треба да буде јасно описана структура пројекта и позиција приложеног кода унутар комплетног пројекта. Такође, код треба да буде изузетно добро коментарисан. Програмски код се прилаже у прилогу. Делови кода се могу наводити и у самом тексту рада (централним поглављима), али ти делови кода не би требали да буду дужи од пола стране.

Променљиве у програму треба да имају формат ImePromenljive и морају бити конзистентне. Пример конзистентно именованих променљивих биле би AdrMaleTabele и AdrVelikeTabele, а неконзистентне би биле AdresaMaleTabele и AdrVelikeTabele. Променљиве треба да имају интуитивне, али и што је могуће краће називе. У тексту тезе променљиве из кода треба писати *курзивом*. Стил писања променљивих може бити и другачији у зависности од врсте програмског језика (на пример, неки програмски језици разликују мала и велика слова, а неки не), па кандидат у договору са ментором може да дефинише стил писања променљивих у програму и на другачији начин.

Аутор може по жељи извршити форматирање текста програмског кода ради боље прегледности, али при томе мора поштовати одређена правила. Форматирање мора бити конзистентно, односно исти начин форматирања мора се применити на све објекте исте врсте у тексту програмског кода. Ако се део програмског кода ради ефикаснијег објашњења убацује у тезу, пожељно је користити одговарајуће алате који ће очувати форматирање текста програмског кода. На пример, ако се користи едитор Notepad++ може се урадити следеће:

Фајл који садржи програмски код треба да се отвори у едитору Notepad++.

Селектује се жељени део програмског кода.

У главном (хоризонталном) менију се изабере опција *Plugins->NppExport->Copy RTF to Clipboard*.

На жељено место у тези се уради *Paste* копираног дела програмског кода.

На овај начин ће форматирање кода из едитора бити очувано, нпр. боје кључних речи ће бити очуване. Наравно, могу се користити и други алати за адекватно убацивање програмског кода у текст тезе.

* 1. Референце

Сви делови текста, слике, табеле и др. који су преузети из неког извора морају бити референцирани. Референца се наводи као број у угластој загради, где број представља редни број извора са списка литературе која је наведена на крају рада. У случају преузете слике и табеле, референца се наводи на крају назива слике/табеле. У случају преузете реченице, референцу навести на крају преузете реченице, при чему ако је реченица дословно преузета онда је потребно дословно преузети текст ставити под наводнике. У случају преузетог дела текста референцу навести на крају последње реченице преузетог дела, при чему ако је текст дословно преузиман онда је потребно преузети текст ставити под наводнике. У случају позивања на неки извор, на одговарајуће место у реченици навести референцу (нпр. У тези је коришћен алгоритам предложен у [2]). Све референце које су наведене у списку литературе, морају имати бар једно позивање у тексту тезе. Изузетак представља индиректно коришћена литература (уџбеници, приручници и сл.) која је коришћена приликом рада на тези, али из које није ништа директно преузето у самом раду. Али, чак и у том случају није лоше да се у неком од уводних поглавља напомене која је помоћна литература коришћена приликом рада на тези и да се на том месту позову референце.

* 1. Списак скраћеница, слика, табела

Иза списка коришћене литературе, треба навести списак скраћеница. Скраћенице наводити у азбучном или абецедном редоследу, при чему се прво пише скраћеница, а онда и њено објашњење. Ако се користе само скраћенице које воде порекло из енглеског језика препорука је да се користи абецедни редослед.

Иза списка скраћеница треба навести списак слика, а потом и списак табела. У случају списка слика прво се наводи нумерација и назив слике, а потом и страница појављивања слике. На исти начин се наводе и табеле. У темплејту су дефинисани стилови за означавање назива слике и табеле, па је могуће аутоматски креирати списак слика и табела.

* 1. Захвалница

У случају да кандидат жели да напише захвалницу (или посвету), онда се она ставља на страну која се смешта између наслова тезе и садржаја. Захвалница не би смела да буде дужа од једне стране, а за сам формат захвалнице користити исти формат као за параграф основног текста тезе.

1. Форматирање текста

Формат наслова, поднаслова, текста, ознака слика и референци који се користи у овом документу представља образац по коме треба извршити форматирање текста тезе. Приликом писања је неопходно водити рачуна о томе да се стил текста увек прецизно дефинише (основни текст, наслов одговарајућег нивоа, ознака слике и табеле). На овај начин омогућава се употреба неких функција које постоје у алату *Microsoft* *Word*, попут аутоматског прављења садржаја, на пример.

Овај документ и у њему дефинисане стилове треба користити као образац. Стилови се могу наћи преко поља *Formatting Toolbar* у верзији *MS Word 2003*, или преко *Home->Styles* (кликом на стрелицу) у верзији *MS Word 2007*. Препоручује се да овај документ сачувате у два фајла, нпр. **teza.docx** и **uputstvo.docx**. У фајлу **teza.docx** унесите своје податке на насловну страну, а остали део упутства осим садржаја избришите, и затим почните писање тезе користећи припремљене стилове према упутству из фајла **uputstvo.docx**.

Према потреби могу се и ручно извршити подешавања на начин који ће бити описан у прилогу А. Ово се првенствено односи на оне случајеве када дође до поремећаја у дефиницији стилова текста који су у употреби. Не треба самостално мењати параметре без преке потребе. Уколико се нека промена ипак врши, она треба да важи за све инстанце текста одређеног стила. На пример, уколико се мења размак између слике и текста објашњења испод ње, онда исти размак мора бити између свих слика и текста који их објашњава.

У овом поглављу ће прво бити објашњено шта се подразумева под форматирањем текста. Затим ће бити дата листа стилова који су на располагању у овом документу и објашњено где и како их треба користити. Детаљније упутство о томе како се може ручно извршити подешавање странице, параметара текста и дефинисање одговарајућих стилова биће дато у прилогу А.

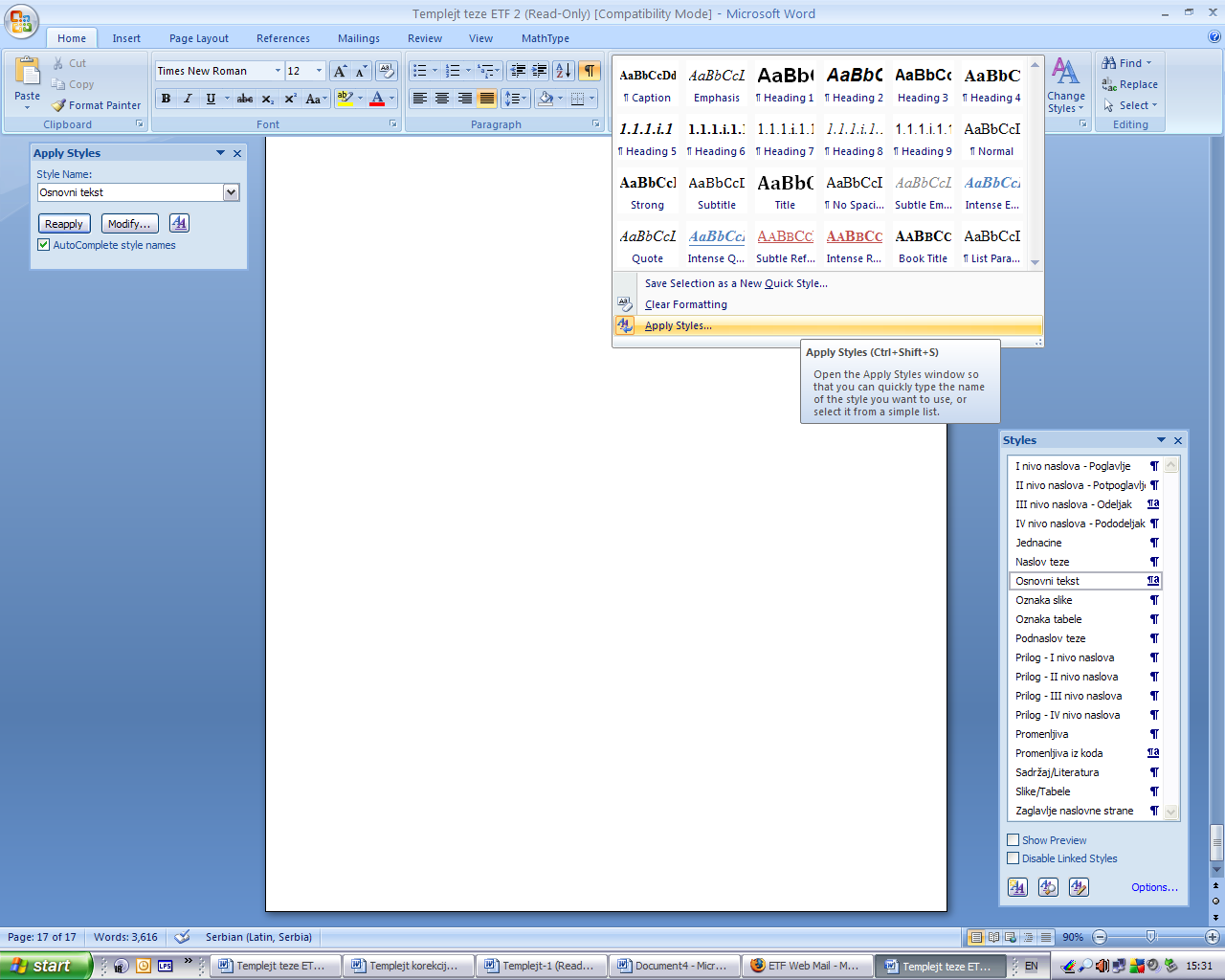
* 1. Форматирање текста и стилови у алату *Microsoft Word*

Под форматирањем текста подразумева се задавање параметара који дефинишу изглед текста (на пример: тип и величина фонта, начин поравнавања текста, итд). У алату *MS Word* је могуће селектовати неки део текста, а затим извршити подешавање његових параметара, коришћењем опција за форматирање расположивих у менију *Format* или помоћу неке од иконица у оквиру поља *Formatting Toolbar* за *MS Word 2003*, или коришћењем неке од опција из менија *Home->Font* и *Home->Paragraph* за *MS Word 2007.* На овај начин, параметри текста се могу подешавати појединачно и независно једни од других. Међутим, могуће је и истовремено подесити више параметара текста, задавањем стила текста. На слици 4.1.1. се види део поља *Formatting Toolbar* *MS Word 2003* у коме се подешава стил текста – у овом случају подешен је стил Osnovni tekst. Кликом на стрелицу са десне стране овог поља отвара се падајући мени у коме су понуђени стилови расположиви у тексту. Одабиром неког од понуђених стилова, дефинише се стил за цео параграф у коме се налази курсор.



Слика 4.1.1. Одабирање стила текста у алату *MS Word 2003*.

У *MS Word 2007* стил се бира одабиром опције *Home->Style*, у оквиру које је неопходно или кликнути на стрелицу у доњем десном углу (испод поља *Change Styles*), чиме се отвара прозор са листом расположивих стилова – десно на слици 4.1.2, или кликом на стрелицу у доњем десном углу *Quick Styles* листе (лево од поља *Change Styles*), и одабиром опције *Apply Styles*, чиме се отвара прозор налик пољу постојећем у *MS Word 2003* – у горњем левом углу на слици 4.1.2.



Слика 4.1.2. Одабирање стила текста у алату *MS Word 2007*.

* 1. Форматирање насловне стране

Насловна страна садржи назив установе у заглављу (Универзитет у Београду у првом реду заглавља, Електротехнички факултет у другом реду заглавља), лого установе, наслов тезе, поднаслов тезе (у коме је специфицирано да је у питању мастер рад), име кандидата и име ментора, као и место и датум (годину и месец) израде тезе.

Насловна страна овог документа представља образац који треба употребити. Заглавље је форматирано стилом Zaglavlje naslovne strane. Лого факултета позициониран је стилом Slike/Tabele. За наслов и поднаслов тезе коришћени су стилови Naslov teze и Podnaslov teze. Стил Vreme predaje teze употребљен је за форматирање датума на дну странице. Име кандидата и ментора писани су у табели без ивица.

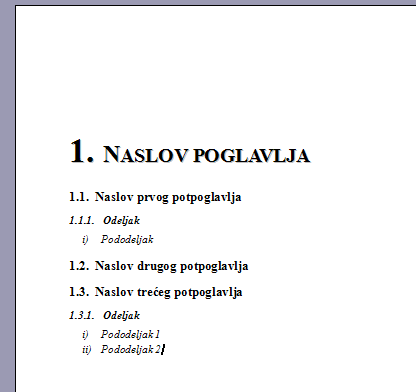
Аутор тезе треба да на место постојећег унесе назив своје тезе. Затим, треба да унесе своје име у поље табеле испод текста „Кандидат“, као и одговарајући датум на дну странице. Такође, аутор треба да у поље табеле испод текста „Ментор“ унесе име ментора и његову титулу (ако рад има два ментора онда се ментори наводе један испод другог).

* 1. Форматирање текста тезе

У овоме поглављу биће објашњено како се форматира централни текст тезе, док ће форматирање додатака, листе референци (литературе), садржаја и сл. бити објашњено касније.

* + 1. Хијерархија наслова текстуалних целина

У поглављу 2. било је речи о организацији текста тезе у хијерархијски уређене целине, односно поглавља, потпоглавља, одељке и пододељке. За наслове ових текстуалних целина користе се стилови I nivo naslova – Poglavlje, II nivo naslova – Potpoglavlje, III nivo naslova – Odeljak и IV nivo naslova – Pododeljak. Да би се аплицирао стил на наслов у *MS Word* *2003*, курсор се постави на откуцани наслов и из поља *Formatting Toolbar* се бира одговарајући стил наслова. У *MS Word* *2007*, се такође курсор постави на наслов, а затим се на *Home->Styles* изабере одговарајући стил наслова. Слика 4.3.1 приказује хијерархију наслова текстуалних целина.



Слика 4.3.1. Хијерархија наслова текстуалних целина.

* + 1. Форматирање текста у оквиру текстуалних целина

Текст у оквиру ових целина пише се стилом Osnovni tekst. Као што је раније напоменуто, курсор се постави на одређени текст и изабере се овај стил.

* + 1. Слике и табеле

У оквиру текста тезе могу се наћи и слике и табеле.

Позиционирање слика и табела

Слике и табеле треба позиционирати централно на страници, при чему оне не смеју да прелазе ширину текста. У сврхе позиционирања слике или табеле користити стил Slike/Tabele, тако што се маркира слика или табела и зада овај стил на већ описани начин.

Означавање слика

Ознаке слика садрже редни број и назив слике, а пишу се испод слика. Слике треба нумерисати према броју поглавља и потпоглавља у коме се налазе, а текст позиционирати централно испод слике. На пример, слика 3.3.1. је прва слика у потпоглављу 3.3. За форматирање нумерације и објашњења испод слике користити стил Oznaka slike. Поставити курсор на текст објашњења и одабрати овај стил.

Означавање табела

Ознаке табела садрже редни број и назив табеле, а пишу се изнад табела. Као и код слика, редни број табеле треба да садржи број поглавља и потпоглавља. Користити стил Oznaka tabele.

* + 1. Једначине

Једначине у тексту морају бити нумерисане. Користити нумерацију са десне стране и укључити број поглавља и потпоглавља у нумерацију.

 (3.3.1)

Нумерисане једначине прво позиционирати централно на страници маркирањем једначине а затим коришћењем стила Jednacine, чиме се дефинише и размак између једначине и текста испод и изнад ње. Затим додати нумерацију, одмах уз једначину, са њене десне стране. Потом додати размак помоћу таб тастера са обе стране једначине, чиме ће се нумерација пребацити на десни крај реда, а једначина ће остати центрирана.

* + 1. Променљиве

Променљиве у тексту пишу се стилом Promenljiva, а уколико је неопходно може се користити и *Equation Editor* за њихово писање*.* За индексе се мора користити одговарајућа иконица. Пошто су називи променљивих из програмског кода обично дугачки, њих треба писати стилом Promenljiva из кода, код кога је размак међу карактерима повећан, ради лакше читљивости.

* + 1. Набрајање

За набрајање користити неки од понуђених стилова за набрајање. Стил Nabrajanje користи графичке ознаке за разликовање нивоа у набрајању:

Ово је први ниво у стилу Nabrajanje. Курсор треба поставити на почетак параграфа који се жели форматирати овим стилом, а затим одабрати стил Nabrajanje из листе понуђених стилова.

* + Ово је други ниво набрајања у стилу Nabrajanje. Курсор треба поставити на почетак параграфа који се жели форматирати овим стилом, одабрати стил Nabrajanje из листе понуђених стилова, а затим притиснути *Таb* како би се променио ниво у набрајању.
    - Ово је трећи ниво набрајања у стилу Nabrajanje. Курсор треба поставити на почетак параграфа који се жели форматирати овим стилом, одабрати стил Nabrajanje из листе понуђених стилова, а затим притиснути *Таb* два пута, како би се променио ниво у набрајању.

Код стила Numerisano nabrajanje у употреби су бројеви и слова:

Ово је први ниво у стилу Numerisano nabrajanje. Курсор треба поставити на почетак параграфа који се жели форматирати овим стилом, а затим одабрати стил Numerisano nabrajanje из листе понуђених стилова. У случају да нумерација не почне од јединице (што се дешава уколико је негде у претходном тексту већ вршено набрајање на овај начин), селектовати број који се појавио, кликнути десним тастером миша и одабрати опцију *Restart numbering*.

* 1. Ово је други ниво набрајања у стилу Numerisano nabrajanje. Курсор треба поставити на почетак параграфа који се жели форматирати овим стилом, одабрати стил Numerisano nabrajanje из листе понуђених стилова, а затим притиснути *Тab* како би се променио ниво у набрајању.
     1. Ово је трећи ниво набрајања у стилу Numerisano nabrajanje. Курсор треба поставити на почетак параграфа који се жели форматирати овим стилом, одабрати стил Numerisano nabrajanje из листе понуђених стилова, а затим притиснути *Tab* два пута, како би се променио ниво у набрајању.

Стилове за набрајање користити у случајевима када треба укратко и прегледно извршити неку класификацију, не и када треба дати детаљна објашњења ставки које су набројане (дужа од неколико реченица). Уколико постоји потреба за детаљнијим објашњењима, треба размотрити другачију организацију текста – може се, на пример, посветити цео одељак класификацији, па се свака ставка онда може детаљно објаснити у оквиру одговарајућег пододељка.

* 1. Листа референци (списак литературе)

Теза мора садржати листу референци (списак коришћене литературе), која се пише на крају централног текста тезе, а пре евентуалних прилога. Наслов листе референци форматирати стилом Sadrzaj/Literatura, а референце у оквиру листе форматирати стилом Referenca. Референце треба нумерисати према редоследу појављивања у тексту. У тексту се на референце позива бројем у угластој загради.

Уколико се референцира књига, наводе се прво имена аутора, затим назив књиге *курзивом*, издавач и година издавања. Пожељно је навести и бројеве страна на којима се налази садржај који се цитира. У примеру који је дат овде, референца [1] је књига.

Уколико се ради о раду у часопису – референца [2], наводе се прво имена аутора, назив рада под наводницима, име часописа *курзивом*, број часописа, странице на којима се рад налази и година издавања.

За рад са конференције, као што је то референца [3], треба навести имена аутора, назив рада под наводницима, назив конференције (односно зборника радова са конференције) *курзивом* и годину одржавања конференције.

За документ који је доступан путем Интернета [4] наводе се имена аутора ако су позната, назив документа, веб адреса са које се он може преузети и датум када је последњи пут кандидат посетио дотичну веб адресу.

* 1. Списак скраћеница, слика и табела

Списак скраћеница се наводи иза списка коришћене литературе. Скраћенице навести у абецедном или азбучном редоследу. Ако се користе само скраћенице које воде порекло из енглеског језика препорука је да се користи абецедни редослед. Списак скраћеница се наводи у табели. Скраћенице се наводе тако што се у првој колони наведе скраћеница, а потом у другој колони оригиналан назив од кога је потекла скраћеница.

Списак слика и табела се наводи иза списка скраћеница. Списак слика и табела се аутоматски креира. Важно је да се користи стил Oznaka slike за нумерацију и назив слике, односно Oznaka tabele за нумерацију и назив табеле, пошто се аутоматски креиране листе заснивају на наведеним стиловима. Десним кликом миша на листу слика/табела и одабиром опције *Update Field* се освежава списак слика/табела. Свака промена у називима слика и табела захтева употребу наведене опције за освежавање списка. У додатку А.7 је објашњено како креирати списак слика, односно табела уколико се грешком избрише списак слика/табела који већ постоји у овом документу.

За форматирање наслова списка скраћеница, слика и табела се користи стил Sadrzaj/Literatura.

* 1. Прилози

У прилозима се може дати програмски код или нека додатна теоријска појашњења, чије би излагање у централном тексту тезе представљало дигресију, а није увек неопходно за разумевање текста тезе. Текст прилога може бити организован у хијерархијски уређене целине, као и текст поглавља. За форматирање наслова прилога користити стил Prilog – I nivo naslova, а у случају потребе на располагању су и стилови Prilog – II nivo naslova, Prilog – III nivo naslova, Prilog – IV nivo naslova.

* 1. Садржај

Уколико се стилови свих наслова у тексту прецизно дефинишу, могуће је аутоматски убацити садржај у *Word* документ. На крају писања, пошто постоје сва поглавља, списак литературе и прилози, треба извршити ажурирање садржаја датог у овом документу, тако што се селектује комплетан текст садржаја, кликне десним дугметом миша и одабере опција *Update Field*. Свака накнадна промена текста захтеваће ново ажурирање садржаја.

Уколико сте случајно избрисали постојећи садржај, нови садржај може се направити према упутству датом у додатку А.7.

5. Примена апликације

1. Литература

D. Gookin, *Word 2003 for Dummies*, Wiley, 2003.

M. Kodialam, T. V. Lakshman, and S. Sengupta, "Traffic-Oblivious Routing for Guaranteed Bandwidth Performance," *Communications Magazine*, 45(4), pp. 46-51, Apr. 2007.

M. Antić, A. Smiljanić, "Oblivious Routing Scheme Using Load Balancing Over Shortest Paths", *Proc. of IEEE International Conference on Communications*, 2008.

*Understanding Styles in Microsoft Word* [Online]. Available: <http://www.addbalance.com/usersguide/styles.htm> (09.02.2016.)

М. Гојгић, "Прописи из области метрологије," *Електропривреда*, 58(2): 84-93, Април-Јун 2006. (<http://www.eps.rs/Elektroprivreda/2-2006.pdf>, 09.02.2016.)

1. Списак скраћеница

|  |  |
| --- | --- |
| ARP | *Address Resolution Protocol* |
| TCP | *Transmission Control Protocol* |

1. Списак слика

[Слика 3.3.1. Неправилан приказ графика 3](#_Toc445721835)

[Слика 3.3.2. Правилан приказ графика 3](#_Toc445721836)

[Слика 4.1.1. Одабирање стила текста у алату *MS Word 2003*. 3](#_Toc445721837)

[Слика 4.1.2. Одабирање стила текста у алату *MS Word 2007*. 3](#_Toc445721838)

[Слика 4.3.1. Хијерархија наслова текстуалних целина. 3](#_Toc445721839)

[Слика А.1.1. Подешавање маргина на страници. 3](#_Toc445721840)

[Слика А.2.1. Прозор са листом стилова текста. 3](#_Toc445721841)

[Слика А.2.2. Прозор за подешавање параметара стила текста. 3](#_Toc445721842)

[Слика А.2.3. Дефинисање стила Osnovni tekst. 3](#_Toc445721843)

[Слика А.2.4. Параметри за параграф основног текста. 3](#_Toc445721844)

[Слика А.2.5. Одабир језика стила. 3](#_Toc445721845)

[Слика А.2.6. Подешавања фонтa и параграфa за стил I nivo naslova – Poglavlje. 3](#_Toc445721846)

[Слика А.2.7. Одабир типа нумерације поглавља. 3](#_Toc445721847)

[Слика А.2.8. Подешавање изгледа нумерације. 3](#_Toc445721848)

[Слика А.2.9. Подешавање фонта за нумерацију поглавља. 3](#_Toc445721849)

[Слика А.2.10. Подешавања нумерације за наслов потпоглавља. 3](#_Toc445721850)

[Слика А.2.11. Подешавања нумерације за наслов пододељка. 3](#_Toc445721851)

[Слика А.5.1. Подешавање позиција за таб. 3](#_Toc445721852)

[Слика А.6.1. Одабир типа нумерације за стил Nabrajanje (лево) и стил Numerisano nabrajanje (десно) 3](#_Toc445721853)

[Слика А.6.2. Параметри за први ниво набрајања стила Nabrajanje 3](#_Toc445721854)

[Слика А.6.3. Параметри за други ниво набрајања стила Nabrajanje 3](#_Toc445721855)

[Слика А.6.4. Параметри за трећи ниво набрајања стила Nabrajanje 3](#_Toc445721856)

[Слика А.6.5. Параметри за први ниво набрајања стила Numerisano nabrajanje 3](#_Toc445721857)

[Слика А.6.6. Параметри за други ниво набрајања стила Numerisano nabrajanje 3](#_Toc445721858)

[Слика А.6.7. Параметри за трећи ниво набрајања стила Numerisano nabrajanje 3](#_Toc445721859)

[Слика А.7.1. Подешавања за садржај. 3](#_Toc445721860)

[Слика А.8.1. Подешавање нумерације за списак литературе. 3](#_Toc445721861)

[Слика А.10.1. Нумерација првог нивоа наслова у прилозима. 3](#_Toc445721862)

[Слика А.10.2. Нумерација другог нивоа наслова у прилозима. 3](#_Toc445721863)

[Слика А.10.3. Нумерација трећег нивоа наслова у прилозима. 3](#_Toc445721864)

[Слика А.10.4. Нумерација четвртог нивоа наслова у прилозима. 3](#_Toc445721865)

1. Списак табела

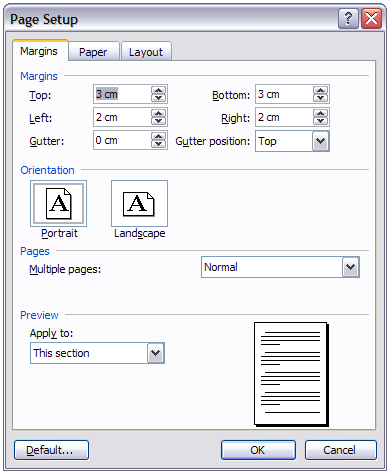
[Табела 3.3.1. Образац за изглед табеле 3](#_Toc445721866)

Подешавање стилова текста за *Word* 2003

У овом прилогу биће дато детаљно упутство како се стилови коришћени у овом документу могу дефинисати у алату *Microsoft Word 2003*.

Подешавање изгледа странице

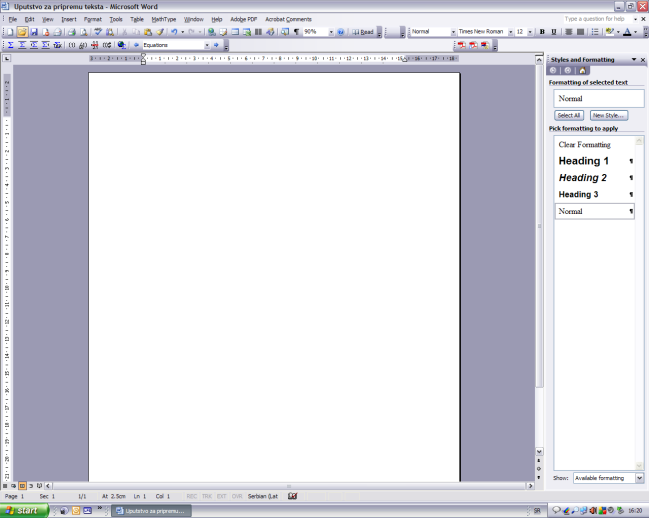
Ширина маргина се подешава опцијом *File->Page Setup*. Вредности за горњу и доњу маргину треба подесити на 3 cm, а за бочне маргине на 2 cm, као на слици А.1.1. У секцији *Paper* подесити да је величина листа А4, а у секцији *Layout* поставити вредност *New page* за *Section start*.



Слика А.1.1. Подешавање маргина на страници.

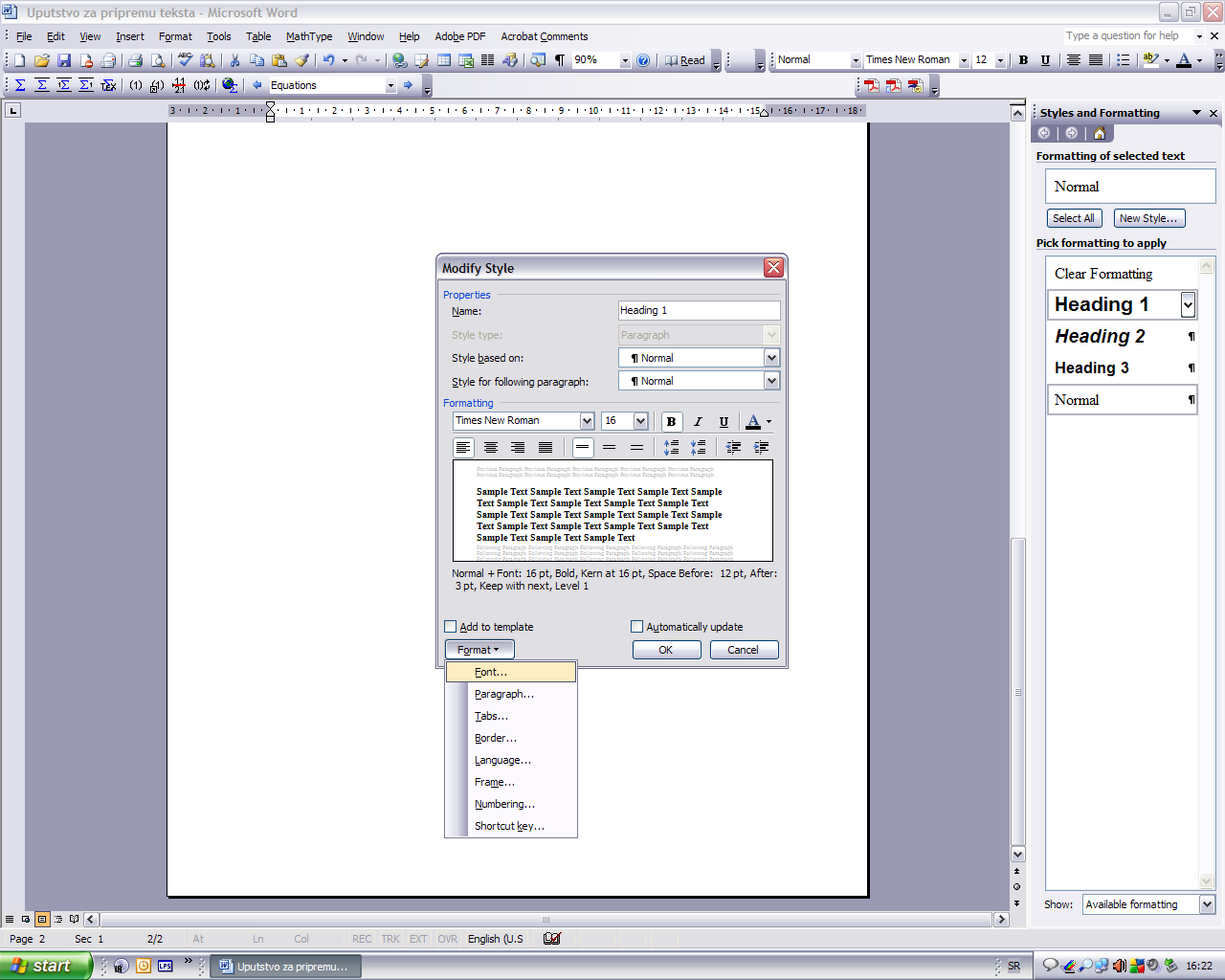
Форматирање основног текста и наслова хијерархијски уређених текстуалних целина

У *MS Word* се форматирање стилова текста одабиром опције *Format->Styles and Formatting*. Тиме се отвара прозор са десне стране екрана у коме је могуће подешавати изглед и величину фонта, размак између наслова и текста, поравнање параграфа, нумерацију и слично – слика А.2.1.



Слика А.2.1. Прозор са листом стилова текста.

Кликом левим тастером миша на стрелицу која се налази поред неког стила текста у листи отвара се падајући мени. Одабиром опције *Modify* у овом менију отвара се прозор за подешавање параметара тог стила текста – слика А.2.2.

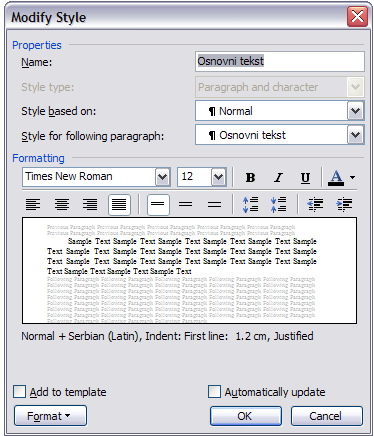


Слика А.2.2. Прозор за подешавање параметара стила текста.

У *MS Word* су почетни стилови дефинисани темплејтом *Normal.dot* и у почетку су на располагању стилови као на слици А.2.2. Они се могу мењати, како би се постигао жељени формат свих инстанци текста одређеног стила. Могуће је и дефинисати нове стилове, одабиром опције *New Style*, која се налази непосредно изнад листе стилова.

Основни текст

За писање основног текста дефинисаћемо стил Osnovni tekst. Овај стил базиран је на стилу *Normal*, који је на располагању у оквиру *Normal.dot* темплејта. Подесити и стил наредног параграфа на Osnovni tekst, као на слици А.2.3 (подешавањем овог поља дефинише се који стил ће аутоматски бити примењен на нови параграф, након стискања тастера *Enter*).



Слика А.2.3. Дефинисање стила Osnovni tekst.

Пошто је дефинисан нови стил, може се приступити подешавању његових параметара, одабиром опције *Format* у доњем левом углу прозора на слици А.2.3. Потребно је подесити следеће параметре:

Фонт

Основни тексттезе пише се фонтом *Times New Roman*, величине 12 pt.

Параграф

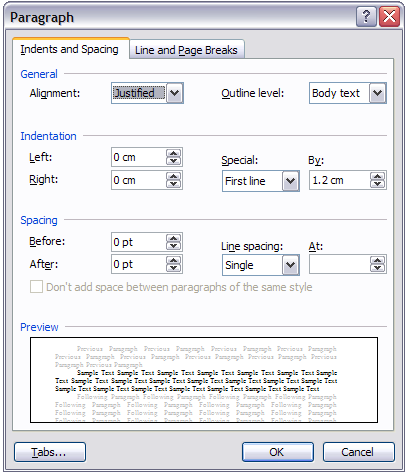
Параметрима за параграф дефинише се начин поравнавања текста у оквиру параграфа, евентуално увлачење прве линије параграфа, као и размаци испод и изнад параграфа текста форматираног датим стилом. За параграф стила Osnovni tekst неопходно је подесити следеће параметре:

*General*: *Alignment*: Justified, *Outline level*: Body text

*Indentation: Left:* 0 cm, *Right:* 0 cm, *Special:* First line, *By*: 1.2 cm

*Spacing: Before:* 0 pt, *After:* 0 pt, *Line spacing:* Single

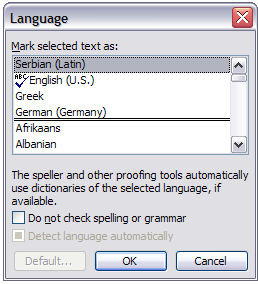
На слици А.2.4. дат је пример прозора у коме се подешавају параметри за параграф. Параметри су подешени за случај стила Osnovni tekst.



Слика А.2.4. Параметри за параграф основног текста.

Језик

Међу понуђеним језицима одабрати *Serbian(Latin)* или *Serbian(Latin, Serbia)*. Тиме ће се спречити да *MS Word* аутоматски проверавајући спелинг (а очекујући енглески, као што је подешено у *Normal.dot* темплејту) означи све српске речи као некоректно написане.



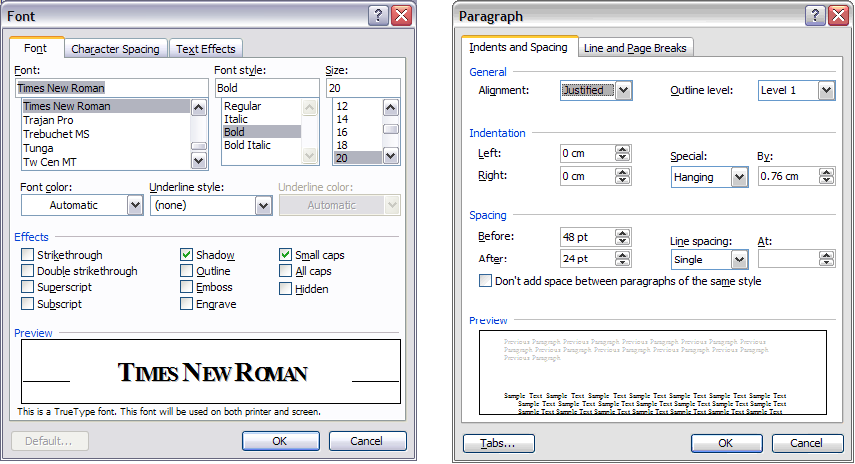
Слика А.2.5. Одабир језика стила.

Наслов поглавља

Наслов поглавља пише се стилом I nivo naslova – Poglavlje. Овај стил заснован је на стилу *Heading 1* који је на располагању у оквиру *Normal.dot*. Пасус који следи после наслова поглавља треба да је писан стилом Osnovni tekst.

Фонт

Наслов поглавља пише се **масним словима**,фонтом *Times New Roman*, величине 20 pt, са укљученим ефектима *small caps* и *shadow.*



Слика А.2.6. Подешавања фонтa и параграфa за стил I nivo naslova – Poglavlje.

Параграф

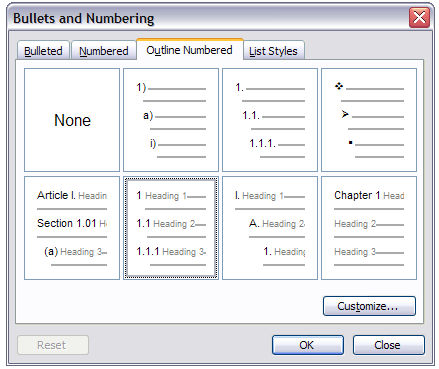
*General*: *Alignment*: Justified, *Outline level*: Level 1

*Indentation*: *Left*: 0 cm, *Right*: 0 cm, *Special*: Hanging, *By*: 0.76 cm

*Spacing*: *Before*: 48 pt, *After*: 24 pt, *Line spacing*: Single

Нумерација

У оквиру поља *Outline Numbered* одабрати нумерацију као на слици А.2.7. Затим кликом на поље *Customize* отворити прозор за подешавање начина нумерације.



Слика А.2.7. Одабир типа нумерације поглавља.

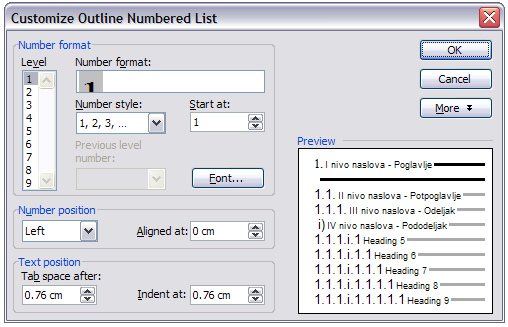
Подесити следеће:

*Number format*: 1. (додати тачку после броја)

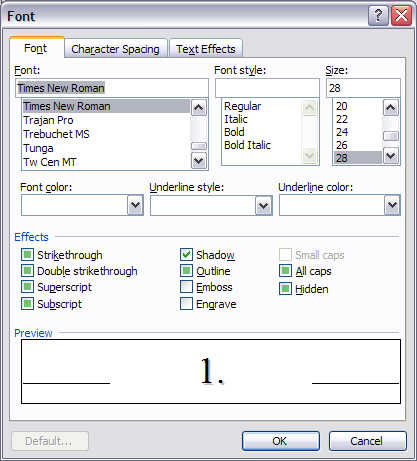
*Number position*: Left, *Alligned at*: 0 cm

*Text position*: *Tab space after*: 0.76 cm, *Indent at*: 0.76 cm.

Кликом на поље *Font* отвара се прозор за подешавање фонта. Одабрати *Times New Roman*, величина фонта 28 pt, са ефектом *shadow*, као на слици А.2.9.



Слика А.2.8. Подешавање изгледа нумерације.



Слика А.2.9. Подешавање фонта за нумерацију поглавља.

Језик

Језик и у овом случају подесити на *Serbian(Latin)* или *Serbian(Latin, Serbia)*.

Наслов потпоглавља

Наслов потпоглавља пише се стилом II nivo naslova – Potpoglavlje. Овај стил заснован је на стилу *Heading 2*, а у параграфу после њега треба да се користи стил Osnovni tekst.

Фонт

Наслов потпоглавља пише се **масним словима**,фонтом *Times New Roman*, величине 14 pt.

Параграф

*General*: *Alignment*: Justified, *Outline level*: Level 2

*Indentation*: *Left*: 0 cm, *Right*: 0 cm, *Special*: Hanging, *By*: 1 cm

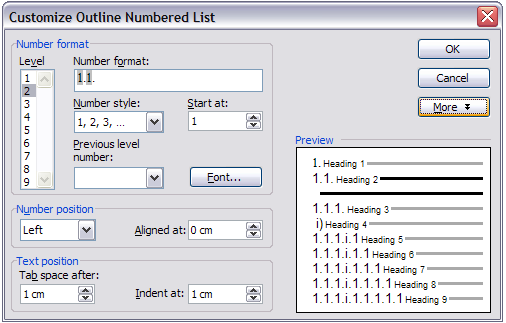
*Spacing*: *Before*: 12 pt, *After*: 8 pt, *Line spacing*: Single

Нумерација

Уколико се ручно врши подешавање, неопходно је додати тачку након броја 1.1. у пољу *Number Format* и подесити следеће параметре:

*Text position*: *Tab space after*: 1 cm, *Indent at*: 1 cm.

Остали параметри ће бити подешени на основу подешавања за наслов поглавља.



Слика А.2.10. Подешавања нумерације за наслов потпоглавља.

Језик

Подразумевани језик и у овом случају подесити на *Serbian(Latin)* или *Serbian(Latin, Serbia)*.

Наслов одељка

Наслов одељка пише се стилом III nivo naslova – Odeljak. Овај стил заснован је на стилу *Heading 3*, а у параграфу после њега треба да се користи стил Osnovni tekst.

Фонт

Наслов одељка пише се **масним словима**, *курзив*фонтом *Times New Roman*, величине 12 pt.

Параграф

*General*: *Alignment*: Justified, *Outline level*: Level 3

*Indentation*: *Left*: 0 cm, *Right*: 0 cm, *Special*: Hanging, *By*: 1.27 cm

*Spacing*: *Befor*e: 12 pt, *After*: 6 pt, *Line spacing*: Single

Нумерација

Додати тачку у пољу *Number Format* после броја 1.1.1. Поставити параметре

*Text position:* *Tab space after:* 1.27 cm, *Indent at:* 1.27 cm.

Језик

Подразумевани језик и у овом случају подесити на *Serbian(Latin)* или *Serbian(Latin, Serbia)*.

Наслов пододељка

Наслов пододељка пише се стилом IV nivo naslova – Pododeljak. Овај стил заснован је на стилу *Heading 4*, а у параграфу после њега треба да се користи стил Osnovni tekst.

Фонт

Наслов одељка пише се *курзив* фонтом *Times New Roman*, величине 12 pt.

Параграф

General: Alignment: Justified, Outline level: Level 4

Indentation: Left: 0.5 cm, Right: 0 cm, Special: Hanging, By: 0.7 cm

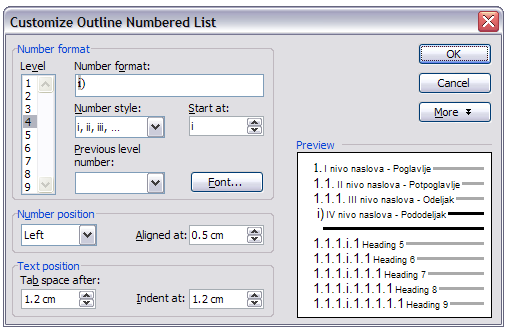
Spacing: Before: 3 pt, After: 3 pt, Line spacing: Single

Нумерација

*Number format*: *Number format*: i), *Number style*: i, ii, iii, ... *Start at*: i

*Number position*: Left, *Aligned at*: 0.5 cm

*Text position*: *Tab space after*: 1.2 cm, *Indent at*: 1.2 cm.



Слика А.2.11. Подешавања нумерације за наслов пододељка.

Језик

Подразумевани језик и у овом случају подесити на *Serbian(Latin)* или *Serbian(Latin, Serbia)*.

Слике

Позиционирање слике

Слике треба да буду центриране на страници. Размак изнад и испод слике треба да износи 8 pt. За правилно позиционирање слике и прављење одговарајућег размака користи се стил Slike/Tabele. Овај стил заснован је на претходно дефинисаном стилу Osnovni tekst, за следећи параграф користи се стил Osnovni tekst, а неопходно је подесити следеће параметре за параграф:

General: Alignment: Centered, Outline level: Body text

Indentation: Left: 0 cm, Right: 0 cm, Special: (none)

Spacing: Before: 8 pt, After: 8 pt, Line spacing: Single

Назив и нумерација слике

Редни број и назив слике пишу се стилом Oznaka slike. Овај стил заснован је на стилу *Caption* из *Normal.dot*. Следећи параграф треба да је писан стилом Osnovni tekst. Потребно је подесити следеће параметре:

Фонт

Подесити фонт на величину 10 pt, **bold**.

Параграф

*General: Alignment:* Centered*, Outline level:* Body text

*Indentation: Left:* 0 cm, *Right:* 0 cm, *Special:* (none)

*Spacing: Before:* 6 pt, *After:* 8 pt, *Line spacing:* Single

Језик

Подразумевани језик је као и до сада *Serbian(Latin)* или *Serbian(Latin, Serbia)*.

Табеле

Позиционирање и изглед табеле

Табеле треба центрирати на страници, и не смеју да прелазе ширину текста. Користити стил Slike/Tabele.

Назив и нумерација табеле

Редни број и назив табеле пишу стилом Oznaka tabele. Овај стил заснован је на стилу *Caption* из *Normal.dot*. Следећи параграф треба да је писан стилом Osnovni tekst.

Фонт

Подесити фонт на величину 10 pt, **bold**.

Параграф

*General: Alignment:* Centered*, Outline level:* Body text

*Indentation: Left:* 0 cm, *Right:* 0 cm, *Special:* (none)

*Spacing: Before:* 8 pt, *After:* 6 pt, *Line spacing:* Single

Једначине

Једначине се пишу стилом Jednacine, заснованим на стилу *Normal* из *Normal.dot*. Подесити следеће параметре:

Параграф

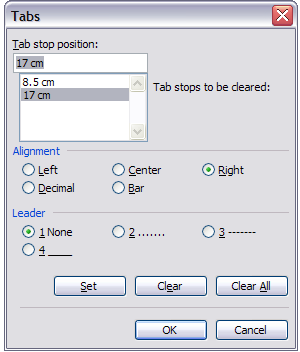
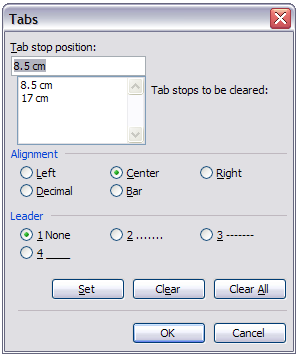
*General: Alignment:* Centered*, Outline level:* Body text

*Indentation: Left:* 0 cm, *Right:* 0 cm, *Special:* (none)

*Spacing: Before:* 3 pt, *After:* 3 pt, *Line spacing:* Single

Таб

Додати зауставне позиције за таб на 8.5 cm и 17 cm. У почетку нема дефинисаних позиција. У поље у коме је на слици А.5.1. лево маркирана вредност 8.5 cm уписати вредност зауставне позиције. За зауставну позицију на 8.5 cm подесити остале параметре као на слици лево (*Alignment*: Center, *Leader*: None), а затим одабрати опцију *Set*. За зауставну позицију на 17 cm подесити остале параметре као на слици А.5.1. десно (*Allignment*: Right, *Leader*: None). Затим одабрати опцију *Set*.



Слика А.5.1. Подешавање позиција за таб.

Набрајање

За набрајање се користе стилови Nabrajanje и Numerisano nabrajanje, заснованим на претходно дефинисаном стилу Osnovni tekst. Следећи параграф треба да је писан стилом Osnovni tekst. Подесити следеће параметре:

Параграф

*General*: *Alignment*: Justified, *Outline level*: Body text

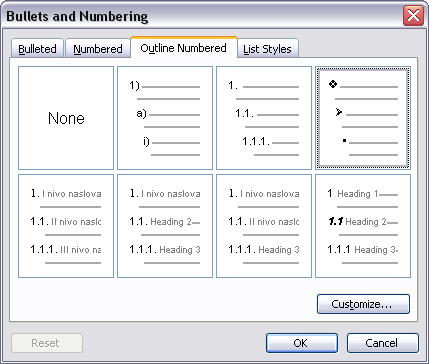
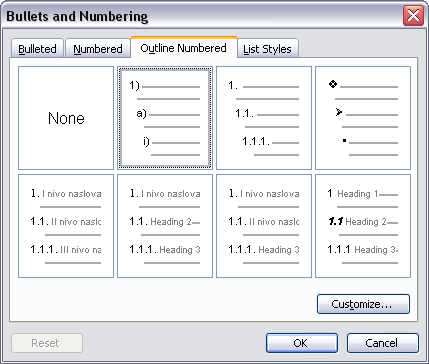
*Indentation: Left:* 1.2 cm, *Right:* 0 cm, *Special:* Hanging, *By*: 0.6 cm

*Spacing: Before:* 1 pt, *After:* 1 pt, *Line spacing:* Single

Нумерација

У оквиру сваког од понуђених стилова дефинисана су по три нивоа набрајања.

За стил Nabrajanje потребно је одабрати опцију *Outline numbered*, као на слици А.6.1. лево, а за стил Numerisano nabrajanje као на слици А.6.1. десно. Затим треба подесити параметре за различите нивое набрајања према упутству које следи.

Слика А.6.1. Одабир типа нумерације за стил Nabrajanje (лево) и стил Numerisano nabrajanje (десно)

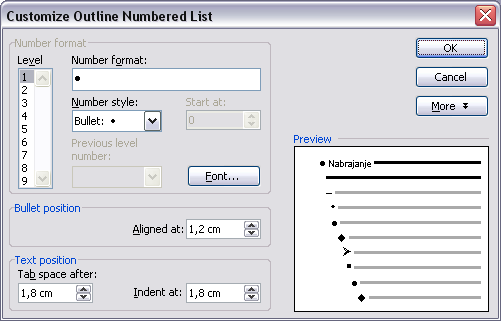
За стил Nabrajanje параметри су следећи:

Први ниво набрајања – слика А.6.2:

*Number format*: *Number format*: •, *Number style*: Bullet •

*Bullet position*: *Aligned at*: 1.2 cm

*Text position*: *Tab space after*: 1.8 cm, *Indent at*: 1.8 cm.



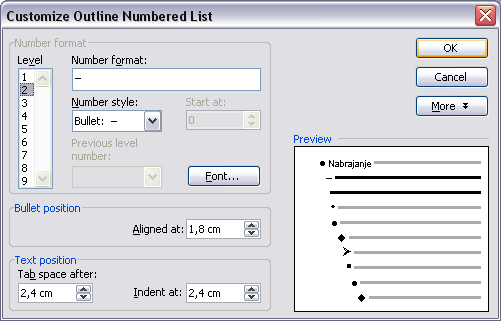
Слика А.6.2. Параметри за први ниво набрајања стила Nabrajanje

Други ниво набрајања – слика А.6.3:

*Number format*: *Number format*: – , *Number style*: Bullet –

*Bullet position*: *Aligned at*: 1.8 cm

*Text position*: *Tab space after*: 2.4 cm, *Indent at*: 2.4 cm.



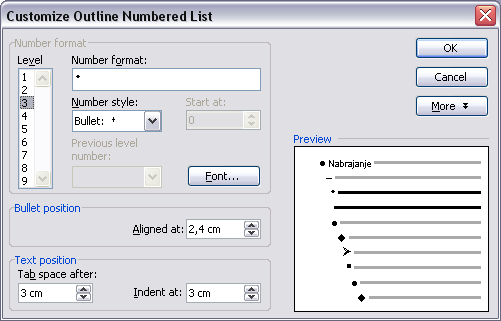
Слика А.6.3. Параметри за други ниво набрајања стила Nabrajanje

Трећи ниво набрајања – слика А.6.4:

*Number format*: *Number format*: ♦ , *Number style*: Bullet ♦

*Bullet position*: *Aligned at*: 2.4 cm

*Text position*: *Tab space after*: 3 cm, *Indent at*: 3 cm.



Слика А.6.4. Параметри за трећи ниво набрајања стила Nabrajanje

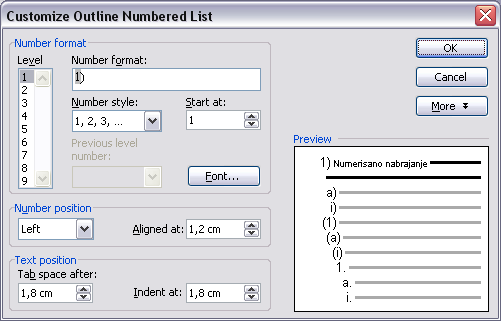
За стил Numerisano nabrajanje параметри су:

1. Први ниво набрајања – слика А.6.5:

*Number format*: *Number format*: 1), *Number style*: 1,2,3,...

*Number position*: Left, *Aligned at*: 1.2 cm

*Text position*: *Tab space after*: 1.8 cm, *Indent at*: 1.8 cm.



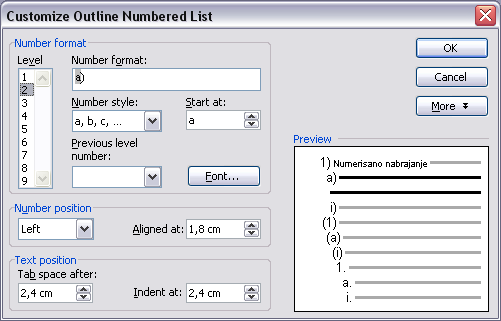
Слика А.6.5. Параметри за први ниво набрајања стила Numerisano nabrajanje

1. Други ниво набрајања – слика А.6.6:

*Number format*: *Number format*: a), *Number style*: a,b,c,...

*Number position*: Left, *Aligned at*: 1.8 cm

*Text position*: *Tab space after*: 2.4 cm, *Indent at*: 2.4 cm.



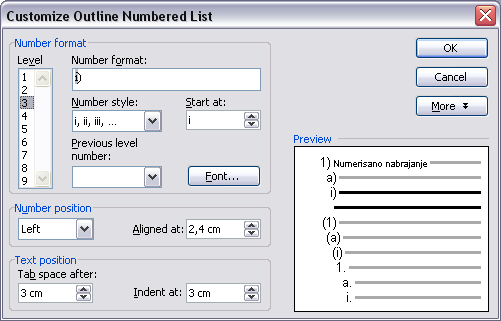
Слика А.6.6. Параметри за други ниво набрајања стила Numerisano nabrajanje

1. Трећи ниво набрајања – слика А.6.7:

*Number format*: *Number format*: i), *Number style*: i,ii,iii,...

*Number position*: Left, *Aligned at*: 2.4 cm

*Text position*: *Tab space after*: 3 cm, *Indent at*: 3 cm.



Слика А.6.7. Параметри за трећи ниво набрајања стила Numerisano nabrajanje

Садржај

Садржај се убацује одабиром опције *Insert->Reference->Index and Tables*. У прозору који се на овај начин отвори одабрати секцију *Table of Contents*, као на слици А.7.1.



Слика А.7.1. Подешавања за садржај.

У оквиру ове секције подешава се број нивоа наслова који се појављују у садржају. Такође, са листе понуђених формата садржаја може се одабрати и формат садржаја који ће се користити. Одабрати формат *Formal* и подесити број нивоа који се појављују у садржају на три:

*Formats*: Formal, *Show Levels*: 3

Одабиром опције *Table of Figures* која се може видети на слици А.7.1 се може аутоматски креирати листа слика, односно табела. У *Options* подешавањима је потребно изабрати стил Oznaka slike за листу слика, односно Oznaka tabele за листу табела.

Списак литературе

Списак литературе форматира се стилом Referenca. Овај стил заснован је на стилу Основни текст, параграф који следи треба да је такође форматиран стилом Referenca. Остала подешавања су следећа:

Параграф

*General*: *Alignment*: Justified, *Outline level*: Body text

*Indentation*: *Left*: 0.63 cm, *Right*: 0 cm, *Special*: Hanging, *By*: 0.63 cm

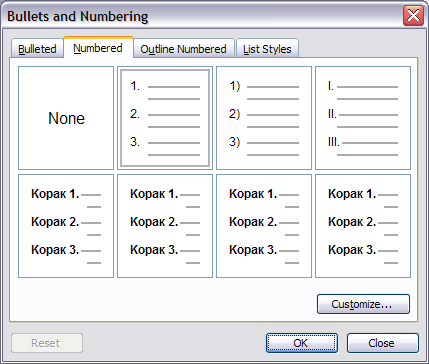
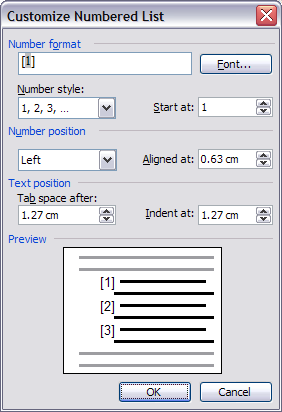
*Spacing*: *Before*: 0 pt, *After*: 0 pt, *Line spacing*: Single

Нумерација

У оквиру опције *Numbered* одабрати нумерацију као на слици А.8.1. лево (1, 2, 3) и кликнути на *Customize*. У прозору који се отвори додати угласту заграду око броја 1 у пољу *Number format*, као на слици А.8.1. десно. Остале параметре подесити на следеће вредности:

*Number position*: Left, *Aligned at*: 0.63 cm.

*Text position*: *Tab space after*: 1.27 cm, *Indent at*: 1.27 cm.

Слика А.8.1. Подешавање нумерације за списак литературе.

Форматирање наслова за садржај, списак литературе (скраћеница, слика, табела)

За наслов садржаја, списка литературе, списка скраћеница, списка слика и списка табела користи се стил Sadrzaj/Literatura. Овај стил заснован је на претходно дефинисаном стилу I nivo naslova – Poglavlje, а после наслова следи параграф писан стилом Osnovni tekst. Треба променити вредности параметара за параграф и нумерацију у односу на стил I nivo naslova – Poglavlje на следеће вредности:

Параграф

*General*: *Alignment*: Justified, *Outline level*: Level 1

*Indentation*: *Left*: 0 cm, *Right*: 0 cm, *Special*: (none)

*Spacing*: *Before*: 48 pt, *After*: 24 pt, *Line spacing*: Single

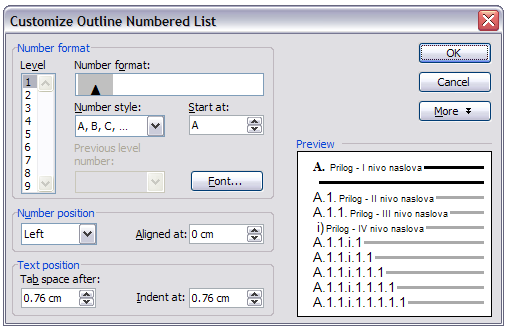
Нумерација

У пољу *Outline Numbered* у прозору на слици А.2.7. одабрати опцију *None*, односно укинути нумерацију.

Наслови хијерархијски уређених целина у прилозима централном тексту тезе

Први ниво наслова у прилозима

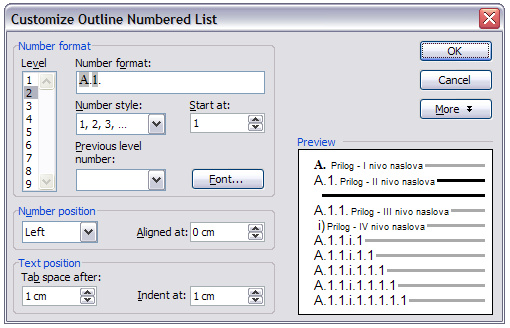
Прилози централном тексту тезе имају истру структуру као и поглавља тезе, једино је први ниво наслова у овом случају обележен словима. Стога је прво потребно дефинисати стил Prilog – I nivo naslova, на основу стила *Heading 1* и са потпуно истим подешавањима као што су она за стил I nivo naslova – Poglavlje, осим када је нумерација у питању. Код нумерације уместо броја 1 у поље *Number format* треба унети слово А и тачку иза њега, као на слици А.10.1.



Слика А.10.1. Нумерација првог нивоа наслова у прилозима.

Други ниво наслова у прилозима

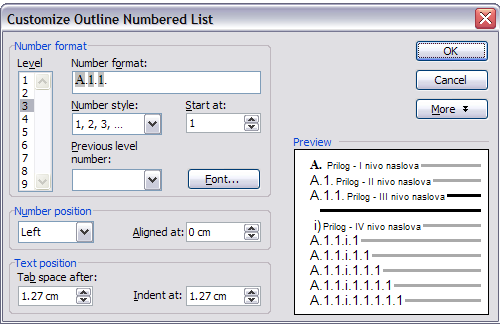
За други ниво наслова у прилозима дефинише се стил Prilog – II nivo naslova. Овај стил заснован је на стилу Prilog – I nivo naslova. Ово је јако битно, јер у супротном може доћи до проблема у нумерацији потпоглавља и одељака у прилозима. Сва остала подешавања су у овом случају као за стил II nivo naslova – Potpoglavlje, осим нумерације која има формат са слике А.10.2, где је у поље *Number format* уписано А.1.



Слика А.10.2. Нумерација другог нивоа наслова у прилозима.

Трећи ниво наслова у прилозима

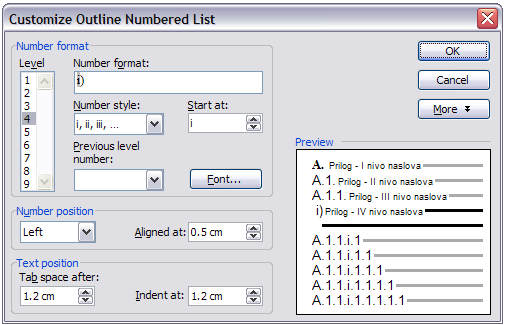
За трећи ниво наслова у прилозима дефинише се стил Prilog – III nivo naslova. Овај стил заснован је на стилу Prilog – II nivo naslova. Сва остала подешавања су у овом случају као за стил III nivo naslova – Odeljak, осим нумерације која има формат са слике А.10.3, где је у поље *Number format* уписано А.1.1.

.

Слика А.10.3. Нумерација трећег нивоа наслова у прилозима.

Четврти ниво наслова у прилозима

За четврти ниво наслова у прилозима дефинише се стил Prilog – IV nivo naslova. Овај стил заснован је на стилу Prilog – III nivo naslova. Сва остала подешавања су у овом случају као за стил IV nivo naslova – Pododeljak, укључујући и нумерацију која има формат са слике А.10.4.



Слика А.10.4. Нумерација четвртог нивоа наслова у прилозима.