

Милош Утвић

Информатички практикум 2

Софтвер, 1. део

Програм и програмске структуре

Степен аутоматизације обраде

- ▶ Ако упоредимо средства која се користе за извршавање неке радње (на пример, рачунање или прање веша), разликујемо **ручна, полуаутоматска и аутоматска** средства.
- ▶ Ручна средства помажу човеку да изврши радњу, али фактички све ради само човек (нпр. ручно прање чарапа, рачунање помоћу рачунаљке, тј. абакуса).
- ▶ Полуаутоматска средства могу сама да обаве појединачне кораке радње, али човек мора да зада сваки корак посебно: нпр. током рачунања помоћу калкулатора (дигитрона), човек задаје податке и операције, а дигитрон обавља рачун.

Програм

- ▶ Појам *програм* се среће на разним местима (програм културне манифестације, програм наставе, програм машине за прање веша, рачунарски програм, итд.).
- ▶ Средства за аутоматску обраду података омогућавају човеку да:
 - ▶ пре почетка обраде, поред података који треба да се обраде, зада и конкретне кораке обраде, њихов прецизан редослед и услове под којима се они извршавају.
 - ▶ током саме обраде човек не мора да интервенише, већ средство самостално изводи појединачне кораке обраде и провером услова доноси одлуке да ли неке кораке треба прескочити или поновити.
- ▶ Спецификација корака обраде које аутоматско средство треба да изврши одређеним редоследом и под одређеним условима назива се **програм**.

Терминологија

- ▶ У рачунарству се најмањи елементи (кораци) рачунарског програма који могу самостално да се изврше именују различито у зависности од тога на ком нивоу се посматра извршавање програма:

- ▶ На нивоу програмског језика (Pascal, C, C++, Java, C#, Python) се користи термин **наредба**, тј. програм је скуп наредби. Нпр. наредба доделе (за рачунање процентуалног износа):

`procenat = deo / celina;`

- ▶ На нивоу оперативног система (DOS, Windows, Linux, Macintosh) се користи термин **команда**, нпр. за приказ података о текстуалним датотекама у текућем каталогу (DOS, Windows Command Prompt)

`dir *.txt`

- ▶ На нивоу процесора се програм третира као скуп (машинских) **инструкција**

ADD X

Програмска структура



Пример:

- Улаз: унос оцена из свих предмета на крају студија за неког студента
- Обрада: израчунавање просечне оцене студента
- Излаз: исписивање просечне оцене студента на екран, у датотеку, штампање дипломе студента итд.

Редослед извршавања корака програма (наредби, команди, инструкција)

- ▶ У највећем броју случајева кораци програма се извршавају оним редоследом којим су записани у програму, тј. сваки корак се извршава после завршетка претходног корака, а онда се прелази на следећи корак.
- ▶ Таква структура програма се назива **линеарна структура**.
- ▶ Међутим, постоје и структуре програма као што су разграната и циклична, код којих редослед извршавања инструкција одступа од њиховог редоследа у запису програма.

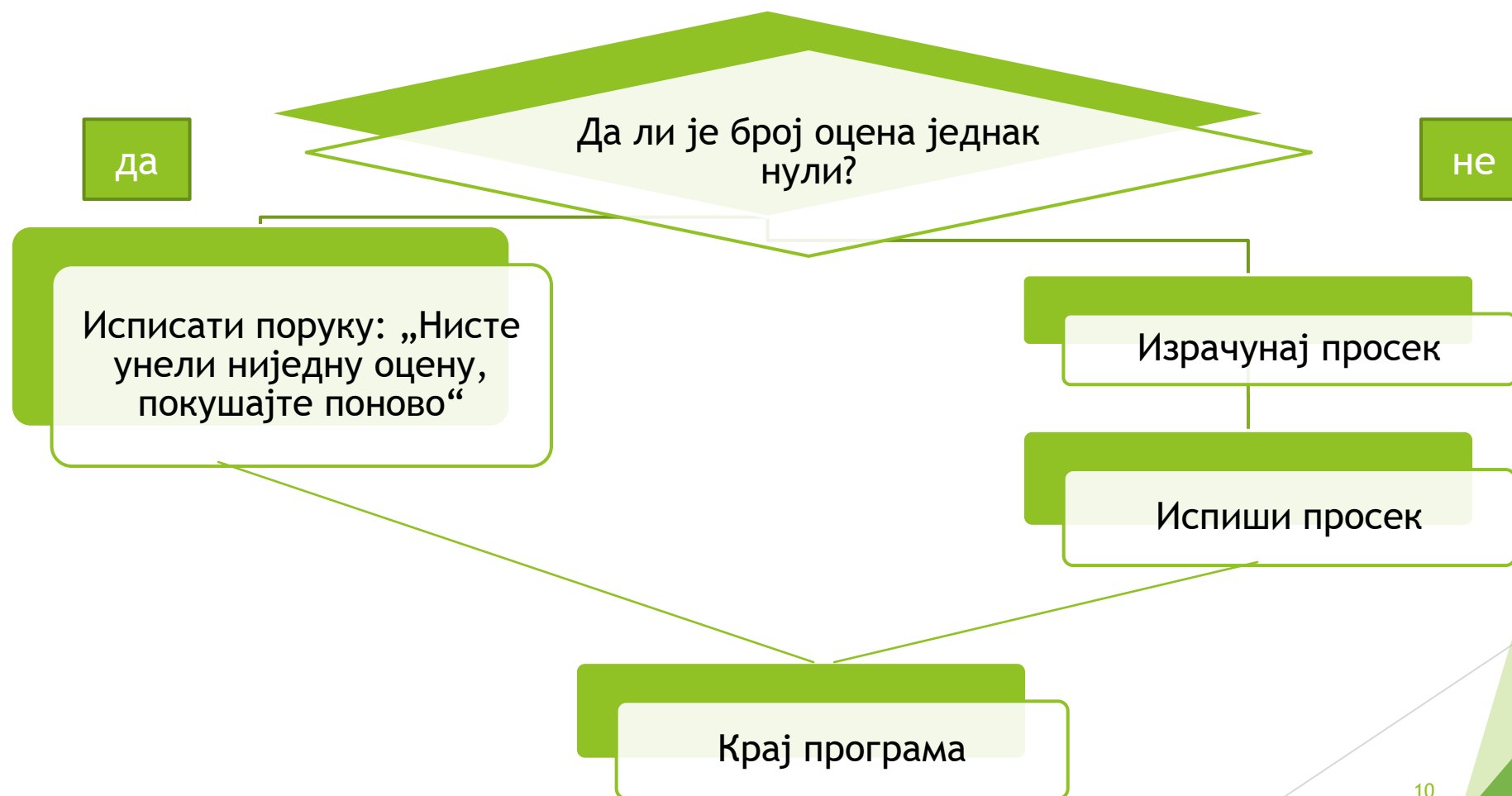
Линеарна структура (пример)



Разграната структура

- ▶ Понекад даље радње програма зависе од вредности добијене у неком кораку програма.
- ▶ Пример: када корисник програма уноси оцене, оне морају бити одређене вредности (5-10), тако да треба увести проверу да ли корисник уноси тражене вредности. Такође, ако корисник не унесе ниједну оцену, треба спречити програм да рачуна просек јер ће се у том случају просек свести на израз 0/0.

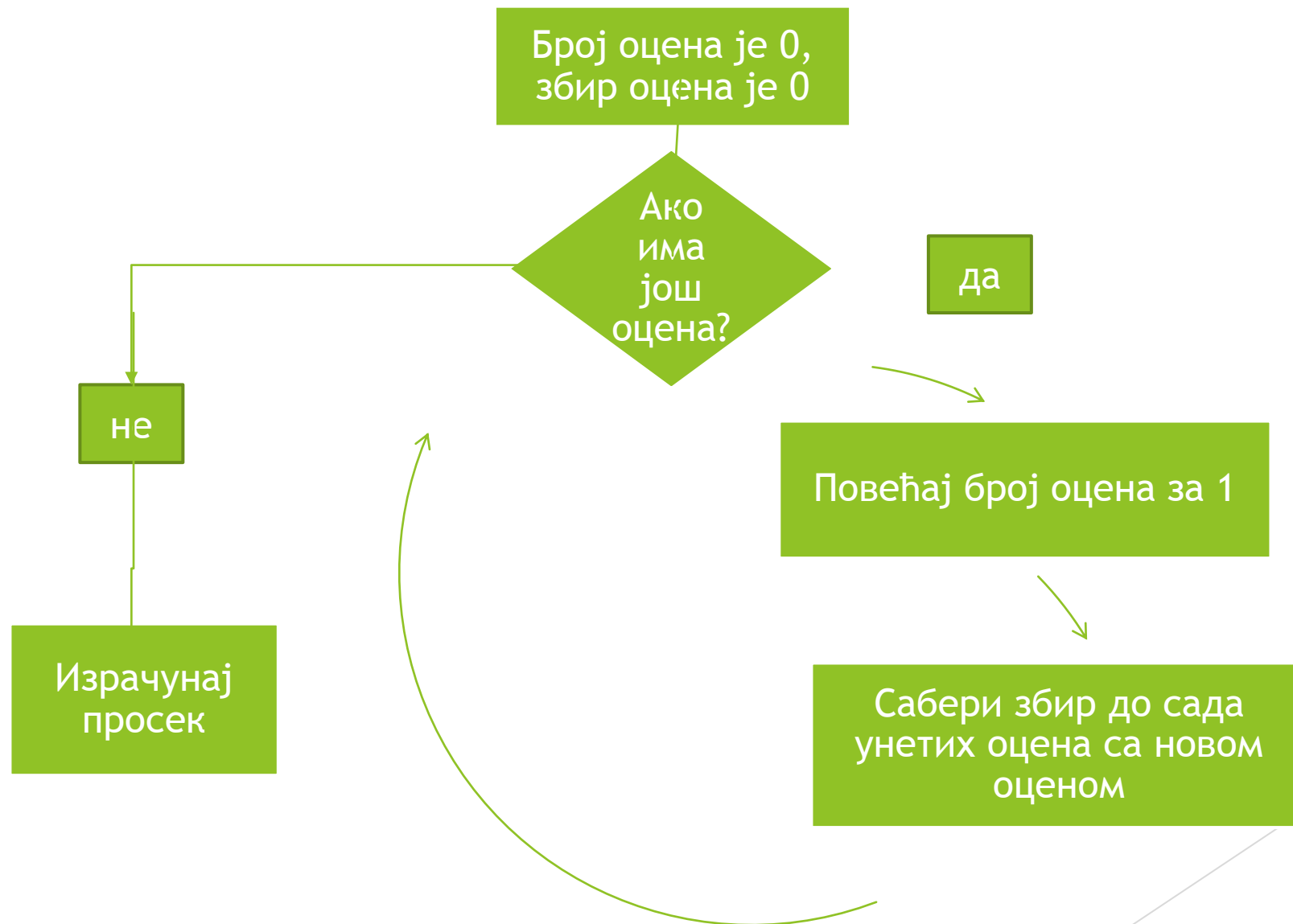
Разграната структура (пример)



Циклична структура (петља)

- ▶ Понекад је неопходно понављати одређени низ корака програма више пута, односно док је неки задати услов испуњен.
- ▶ Пример: ако се оцене студента уносе једна по једна, онда при сваком уносу треба урадити исти низ корака:
 - ▶ Повећати број оцена за један (пребројавање оцена)
 - ▶ Додати унету оцену на збир до тада унетих оцена.
- ▶ На почетку је збир оцена и број оцена једнак нули.
- ▶ Поступак се понавља све док има још оцена које треба унети.

Пример цикличне структуре (петље)



Упоредни преглед структуре делова програма

- ▶ Код линеарне структуре се сваки корак извршава тачно једном.
- ▶ Код разгранате структуре неки делови програма се извршавају највише једном, тј. једном или ниједном.
- ▶ Код цикличне структуре неки делови програма се могу извршавати једном, ниједном или више пута.
- ▶ Програм може садржати делове који користе сваку од наведене три структуре.

Софтвер

Терминологија

- ▶ Хардвер је постојао као термин у енглеском и пре електронских рачунара и означавао је (у слободном преводу) „гвожђурију“ (енгл. hardware store = „гвожђара“).
- ▶ Енглески појам software је скован као нека врста „супротности“ за hardware. Појам hardware је рашчлањен на придев hard („тврд“) и суфикс ware који се користи да означи некакво груписање и одговара српским суфиксима за збирне именице (отуда и превод „гвожђурија“). Комбиновањем придева soft („мек“), као супротности за hard, и суфикса ware настао је појам software. Ова игра речи је непреводива на српски језик (упркос покушајима типа „тврдотворина“ и „мекотворина“, в. Референце), па су из тог разлога оба термина транскрибована на српски и користе се као интернационализми хардвер и софтвер.

Дефиниција

- И док хардвер означава „гвожђурију“, тј. разне физичке делове рачунарског система са којима смо се упознали у оквиру курса Информатички практикум 1, софтвер се користи као збирна именица да означи све рачунарске програме (енгл. computer program). У том смислу, софтвер пре треба да се схвати као допуна хардверу, а не као његова супротност.

Врсте софтвера

- ▶ Софтвер се може класификовати по различитим критеријумима. У оквиру курса Информатички практикум 2 разматраћемо само поделе по:
 - ▶ намени:
 - ▶ апликативни софтвер (енгл. application software)
 - ▶ системски софтвер (енгл. system software)
 - ▶ злонамерни или малициозни софтвер (енгл. malicious software, malware).
 - ▶ софтверској лиценци.

Референце

1. *Oksfordski rečnik računarstva*. Beograd : Nolit, 1990.

Апликативни софтвер

Дефиниција

- ▶ Апликативни софтвер је добио име по енгл. појму application („примена“) којим се у енглеском језику означава рачунарски програм са специфичном наменом који решава неки корисников проблем (на пример, графичко пројектовање). С друге стране, и системски софтвер има важну примену, али више када је у питању функционисање рачунарског система, тако да га корисници често узимају здраво за готово, тј. подразумевају да је то саставни део рачунарског система (на пример, унос и приказ текста на екрану).

Примери апликативног софтвера (1)

- ▶ Програми за обраду и форматирање текста (енгл. word processors):
 - ▶ Microsoft Office Word
 - ▶ Apache OpenOffice Writer (verzija prilagođena za srpski zove se Pisac)
 - ▶ LibreOffice Writer
 - ▶ За ширу листу видети на пример https://en.wikipedia.org/wiki/List_of_word_processors
 - ▶ Notepad и Notepad++ не припадају овој класи софтвера зато што немају могућност форматирања делова текста, само производе и чувају „чист, обичан“ текст (енгл. plain text).

Примери апликативног софтвера (2)

- ▶ Програми за табеларна израчунавања (енгл. spreadsheet software):
 - ▶ Microsoft Office Excel
 - ▶ Apache OpenOffice Calc
 - ▶ LibreOffice Calc
 - ▶ За ширу листу видети на пример https://en.wikipedia.org/wiki/List_of_spreadsheet_software

Примери апликативног софтвера (3)

- ▶ Системи за управљање базама података (енгл. Database Management Systems, скр. DBMS):
 - ▶ Microsoft Office Access
 - ▶ Microsoft SQL Server
 - ▶ IBM DB2
 - ▶ Oracle
 - ▶ MySQL
 - ▶ PostgreSQL (Postgres)
 - ▶ MarkLogic
 - ▶ MongoDB
 - ▶ За ширу листу видети на пример https://en.wikipedia.org/wiki/Lists_of_database_management_systems

Примери апликативног софтвера (4)

- ▶ Програми за преглед слика (енгл. image viewers):
 - ▶ ACDSee
 - ▶ IrfanView
 - ▶ Nomacs
 - ▶ Eye of Gnome
 - ▶ За ширу листу видети на пример https://en.wikipedia.org/wiki/Comparison_of_image_viewers
- ▶ Типови слике:
 - ▶ растерска слика (слика као скуп пиксела); уређивање на нивоу пиксела, тако на пример ради MS Paint)
 - ▶ векторска слика (слика као скуп објеката, тј. геометријских фигура: тачака, правих линија, полигона, кривих); уређивање на нивоу објеката са могућношћу да се лако изабере објекат и промене његова својства.

Примери апликативног софтвера (5)

- ▶ Програми за обраду слика (енгл. image processing):
 - ▶ уређивање растерских слика (енгл. raster graphics editors):
 - ▶ Adobe PhotoShop
 - ▶ GIMP (GNU Image Manipulation Program)
 - ▶ ImageMagick
 - ▶ За ширу листу видети на пример https://en.wikipedia.org/wiki/Comparison_of_raster_graphics_editors
 - ▶ уређивање векторских слика
 - ▶ Adobe Illustrator
 - ▶ Apache OpenOffice Draw
 - ▶ Corel Draw
 - ▶ Inkscape
 - ▶ За ширу листу видети на пример https://en.wikipedia.org/wiki/Comparison_of_vector_graphics_editors

Примери апликативног софтвера (6)

- ▶ Програми за обраду звука (енгл. audio editing)
 - ▶ Soundforge
 - ▶ Audacity
 - ▶ За ширу листу видети на пример https://en.wikipedia.org/wiki/Comparison_of_digital_audio_editors
- ▶ Програми за обраду видео-записа (енгл. video editing software)
 - ▶ Ffmpeg
 - ▶ VirtualDub
 - ▶ За ширу листу видети на пример https://en.wikipedia.org/wiki/List_of_video_editing_software

Примери апликативног софтвера (7)

- ▶ Програми за репродукцију звука и видео-записа су обично обједињени у један програм (енгл. media player software):
 - ▶ GOM Player
 - ▶ Media Player Classic
 - ▶ Mplayer
 - ▶ PowerDVD
 - ▶ VLC
 - ▶ Winamp
 - ▶ За ширу листу видети на пример https://en.wikipedia.org/wiki/Comparison_of_video_player_software

Системски софтвер: оперативни системи

Врсте системског софтвера

- ▶ Системски софтвер је добио назив по томе што брине о функционисању рачунарског система. Сама подела између апликативног и системског софтвера није строга, тако да временом неки рачунарски програм, који се до јуче убрајао у апликативни софтвер, може да постане толико неопходан као подршка корисницима, да на крају постане саставни део системског софтвера. Слично важи и за врсте системског софтвера:
 - ▶ фирмвер (енгл. firmware)
 - ▶ оперативни системи (енгл. Operating systems, скр. OS, тј. ОС)
 - ▶ драјвери (енгл. drivers)
 - ▶ услужни софтвер (енгл. accessories)
 - ▶ језички процесори (енгл.)
- ▶ па тако може да се говори о брисању границе између драјвера и оперативних система, тј. драјвери се данас третирају као део оперативних система. Такође, уобичајено је да се куповином ОС-а добију и примерци услужног софтвера као саставног дела ОС-а (нпр. Notepad, Paint итд.).

Оперативни систем (ОС). Примери

- ▶ Microsoft Windows (3x, 9x, NT, 2000, XP, Vista, 7, 8x, 10)
- ▶ Apple Mac OS
- ▶ UNIX i njegovi derivati:
 - ▶ Linux distribucije:
 - ▶ Debian
 - ▶ Slackware
 - ▶ Ubuntu
 - ▶ За ширу листу видети
https://en.wikipedia.org/wiki/Comparison_of_Linux_distributions
 - ▶ FreeBSD

Оперативни системи за мобилне телефоне. Примери

- ▶ Android (заснован на језгру Linux-a)
- ▶ Apple iOS
- ▶ За ширу листу видети
https://en.wikipedia.org/wiki/Mobile_operating_system

Терминологија

- ▶ Уређаји рачунарског система и програми којима систем располаже се једним именом називају **ресурси**. Према томе, рачунарски систем располаже:
 - ▶ хардверским и
 - ▶ софтверским ресурсима.

Улоге оперативног система

▶ Оперативни систем:

- ▶ управља хардверским ресурсима (уређајима), тестира исправност, надгледа их и обезбеђује њихов несметани и ефикасни рад;
- ▶ посредује између корисника и рачунара (на пример, када корисник жели да види податке у некој датотеци, да покрене неки програм);
- ▶ посредује између рачунарских програма (софтвера) и уређаја рачунарског система (на пример, комуникација између програма са отвореним документом који треба да се одштампа и штампача);
- ▶ посредује у комуникацији између различитих рачунара (мрежна комуникација)
- ▶ брине о сигурности система (вишекориснички рад на истом рачунару, детекција злонамерног софтвера).

Управљање хардверским ресурсима

- ▶ Између вишеструких улога које оперативни систем обавља током управљања хардверским ресурсима размотрићемо следеће:
 - ▶ управљање процесором (енгл. processor management),
 - ▶ управљање радном (примарном, главном, оперативном, унутрашњом) меморијом (енгл. memory management),
 - ▶ управљање уређајима (у оквиру те теме биће обрађени фирмвер и драјвери),
 - ▶ систем датотека и каталога (енгл. file system).

Управљање процесором (multitasking)

- ▶ Процес (енгл. task) је програм који се извршава. Ако покренемо четири примерка истог програма (нпр. Notepad), у питању је један те исти програм, али четири различита процеса. Сваки од тих процеса има своје посебно место у радној меморији рачунара (тзв. контекст процеса).
- ▶ Једноставни оперативни системи омогућавају да се у сваком тренуку извршава један једини процес, тј. док један процес не заврши рад, други процес не може започети са радом.
- ▶ Данашњи оперативни системи омогућавају рад више процеса истовремено (енгл. multitasking). Тако, док слушате музику на рачунару, можете истовремено да уређујете неки документ и покренете његово штампање.

Управљање процесором (time sharing)

- ▶ Поставља се питање: како функционише multitasking ако процесор (који има само једно језгро) у једном тренутку може да изврши само једну инструкцију?
- ▶ Постоје разни рецепти за решење тог проблема, један од најједноставнијих је рад у расподељеном времену (енгл. time sharing).
- ▶ Идеја да се време процесора подели на толико ситне временске интервале који се додељују програмима наизменично. Додељени временски интервали су, с једне стране, довољно дуги да се током њих може извршити више инструкција неког процеса, а с друге стране толико кратки да корисник не може да примети да процесор наизменично извршава процесе, већ има утисак да се они одвијају истовремено.

Управљање процесима

- ▶ Често постоји потреба да корисник тренутно оконча рад неког процеса који се „отео контроли“ и кога оперативни систем не може сам да прекине. Из тог разлога оперативни системи обично располажу системским програмом који нуди приказ активних процеса и ресурса који користе, као и могућност прекида њиховог рада, на пример:
 - ▶ Task Manager (управљач процесима), Windows.
 - ▶ команде top (приказ процеса) и kill (прекид процеса), Linux.

Управљање процесима (dead lock)

- ▶ Један од задатака оперативног система је расподела ресурса процесима на захтев. При томе оперативни систем мора да води рачуна да не дође до ситуација када расподела не може да се обави, односно када рачунарски систем не може да настави са радом. Типичан пример такве ситуације је узајамно закључавање (енгл. deadlock), када два или више процеса желе исте ресурсе, при чему је сваки од њих већ добио неке, али не све жељене ресурсе, тако да ниједан не може да настави са радом јер чека на остале да ослободи ресурсе који му недостају (Дијкстрин пример „Филозофи на вечери“

https://en.wikipedia.org/wiki/Dining_philosophers_problem)

Управљање радном меморијом

- ▶ Са оперативним системом који подржава multitasking, корисници су склони да покрену више процеса и да их наизменично користе, тако да ниједан није завршио са радом и заузима део радне меморије. Ако систем не може да додели меморију на захтев јер је комплетна меморија заузета (искоришћена), систем не може да настави са радом јер чека да се део меморије ослободи што се неће десити само од себе.
- ▶ Старије верзије оперативних система су обично биле немоћне у таквом случају, а појединци су развијали апликативне програме који би упозоравали корисника да се критични тренутак нестанка меморије ближи и предлагали да се меморија ослободи насилним прекидом рада појединих процеса.

Виртуелна меморија

- ▶ Свака меморија је једна врста коначног складишта за податке и процесе (тј. програме који се извршавају). Самим тим, она се не може непрекидно попуњавати новим подацима и новим програмима. Данашњи оперативни системи проблем недостатка расположиве радне меморије решавају користећи виртуелну меморију (енгл. virtual memory, swap).
- ▶ Идеја је да се део спољне меморије (диска, HDD, SSD) третира као део радне меморије, тј. да се процеси који се не користе привремено изместе из радне меморије на спољну меморију (диск).

Управљање уређајима: фирмвер и драјвери

- ▶ Системски софтвер најнижег нивоа представљају два типа софтвера:
 - ▶ фирмвер (енгл. firmware) и
 - ▶ драјвери (енгл. device drivers) или управљачи уређајима, погонски програми.
- ▶ Сличности:
 - ▶ и један и други тип софтвера је задужен да обезбеди функционалност и контролу хардверских уређаја, тј. да пусти уређај у рад (тј. погон, одатле проф. Парезановић зове драјвере погонашима) и да га надзире.
 - ▶ обично оба типа софтвера праве и дистрибуирају произвођачи уређаја.

Специфичности фирмвера

- ▶ Фирмвер је уграђен у сам уређај и непосредно задужен за његову иницијализацију (довођење у почетно радно стање) и контролу даљег рада уређаја.
- ▶ Самим тим је фирмвер је зависан од уређаја и фактички је неприменљив на уређај другог типа ма како сродни били.
- ▶ Фирмвер се обично чува у некој трајној меморији (енгл. non volatile, persistent memory) као што је на пример ROM, EPROM или флеш меморија (не на USB-флешу). Самим тим је могуће ажурирати фирмвер неких хардверских уређаја (ако се не чува у ROM-у) како би се унапредио рад уређаја.

Примена фирмвера

- ▶ Фирмвер није ограничен на примену само у рачунарским системима, већ се у пракси користи у готово свим електронским уређајима, било да су у питању:
 - ▶ ТВ монитори и даљински управљачи,
 - ▶ мобилни телефони,
 - ▶ аутомобили,
 - ▶ веш машине,
 - ▶ дигиталне камере
 - ▶ матичне плоче (BIOS), дискови (HDD, SSD), мрежни уређаји, штампачи и разни други периферијски уређаји рачунарског система итд.

Специфичности драјвера

- ▶ Драјвери су физички одвојени од уређаја на које се односе, обично се чувају на спољној меморији на којој се физички налази и сам оперативни систем. Уређаји су међусобно повезани магистралама и картицама те их и драјвери користе како би комуницирали са уређајима.
- ▶ Назив драјвера упућује на то да су они **управљачи уређајима**, тј. да управљају њиховим радом.
- ▶ Драјвери имају улогу посредника између оперативног система с једне стране и фирмвера и самог уређаја са друге стране. Оперативни систем користи драјвере за комуникацију са фирмвером и уређајем (тј. драјвери преводе поруке које ОС разуме у поруке које фирмвер и уређај разумеју и обрнуто). Преко драјвера ОС контролише уређаје и добија информације о њиховом тренутном статусу.⁴⁴

Однос оперативних система и драјвера

- ▶ Иако се оперативни системи и драјвери третирају као две различите врсте системског софтвера, чињеница је да постоје драјвери који су саставни део оперативних система. Наиме, произвођачи оперативних система, у жељи да смање зависност ОС-а од конкретних уређаја, испоручују као део ОС-а тзв. **генеричке** драјвере, тј. драјвере који се могу користити за сродне уређаје различитих произвођача (нпр. генерички драјвер за веб-камерицу независан од конкретног произвођача веб-камерица). Самим тим се често и не прави разлика између ОС-а и драјвера, већ се драјвери третирају као подскуп програма који чине ОС.

Резервисање (алокација) меморијског простора на спољашњим меморијама

- ▶ Већ смо се упознали са јединицама меморијског капацитета (бит, бајт) и на примеру процесора и радне меморије видели да се приликом преноса и обраде података користе веће јединице (процесорска реч, односно меморијски регистар).
- ▶ Оперативни систем такође користи веће посебне јединице за резервацију (алокацију) меморијског простора на спољашњим меморијама (HDD и SSD дисковима, USB-флешу, меморијским картицама итд.) познате под називом **кластери** (енгл. cluster, allocation unit).

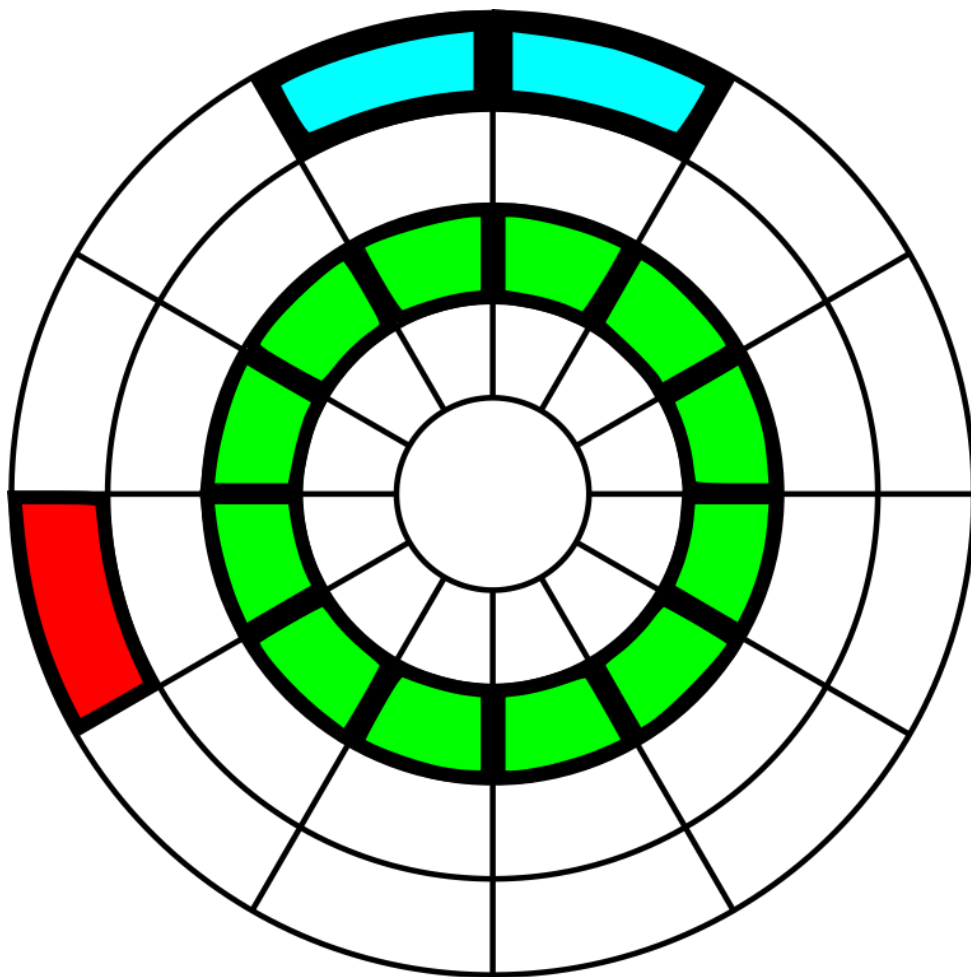
Пример: кластери на HDD-у (1): стазе и сектори

- ▶ У оквиру ИП1 смо видели да се HDD састоји од више диск-плоча на којима се записују подаци. Записи података на плочи су распоређени на концентричним круговима (тачније, прстеновима) који се називају **стазе** (енгл. tracks). Стазе су премазане феромагнетним материјалом и подељене на ситне делове који могу да се намагнетишу на један од два могућа начина (два различита смера намагнетисања, односно поларитета). Тиме сваки намагнетисани део добија улогу бинарне меморијске ћелије која може бити у једном од два могућа стања означена као 0 или 1.
- ▶ Стазе су подељене на делове (нпр, величине 512 бајта) који се називају сектори. **Сектори** (енгл. sectors) су најмањи делови HDD-а којима се може приступити по адреси, па се упис и читање са HDD-а врши по секторима. Адреса сектора се састоји од редног броја плоче на којој се налази сектор и редног броја сектора на плочи.

Пример: кластери на HDD-у (2)

- ▶ Кластер је најмања јединица спољне меморије коју оперативни систем може да резервише за датотеку. Самим тим, датотека је заправо позитиван цео број кластера.
- ▶ Кластер се састоји од једног или више сектора (обично суседних). Величина кластера зависи од верзије оперативног система. На пример, ако је сектор величине 512 бајта, а кластер садржи осам сектора, величина кластера биће $8 \times 512 \text{ B} = 4096 \text{ B} = 4 \text{ KiB}$.

Стазе, сектори и кластери на HDD-у



стаза



сектор



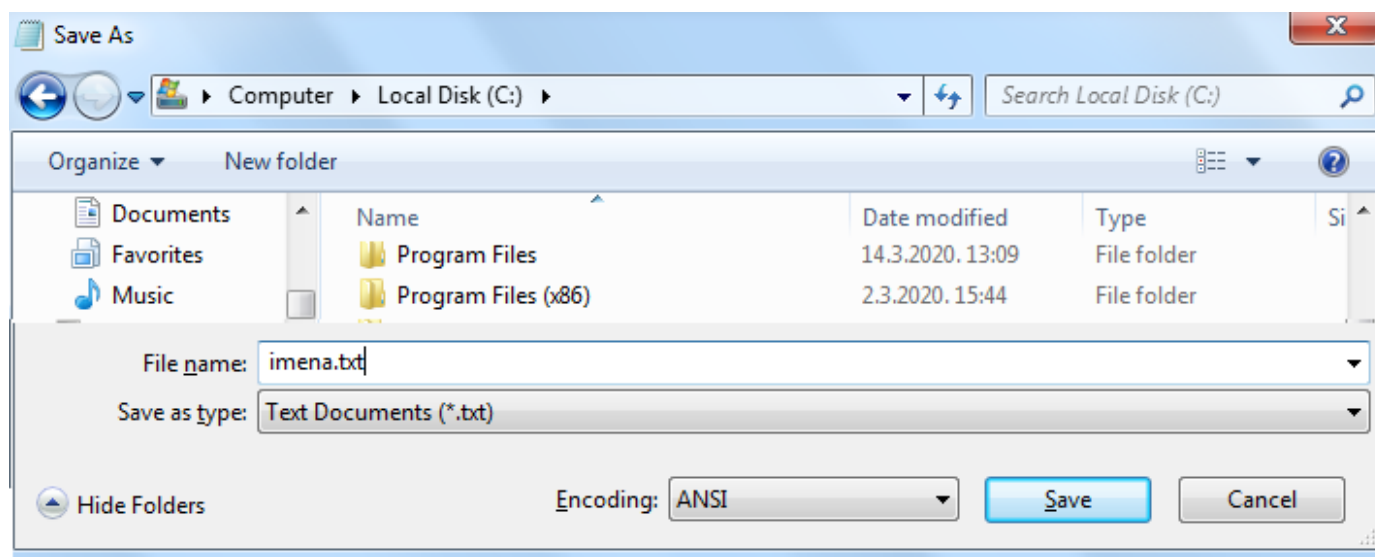
кластер

Експеримент

- ▶ Отворите Notepad и откуцајте у два реда латиницом два имена користећи само слова енглеске абецедe, на пример:

Ivana

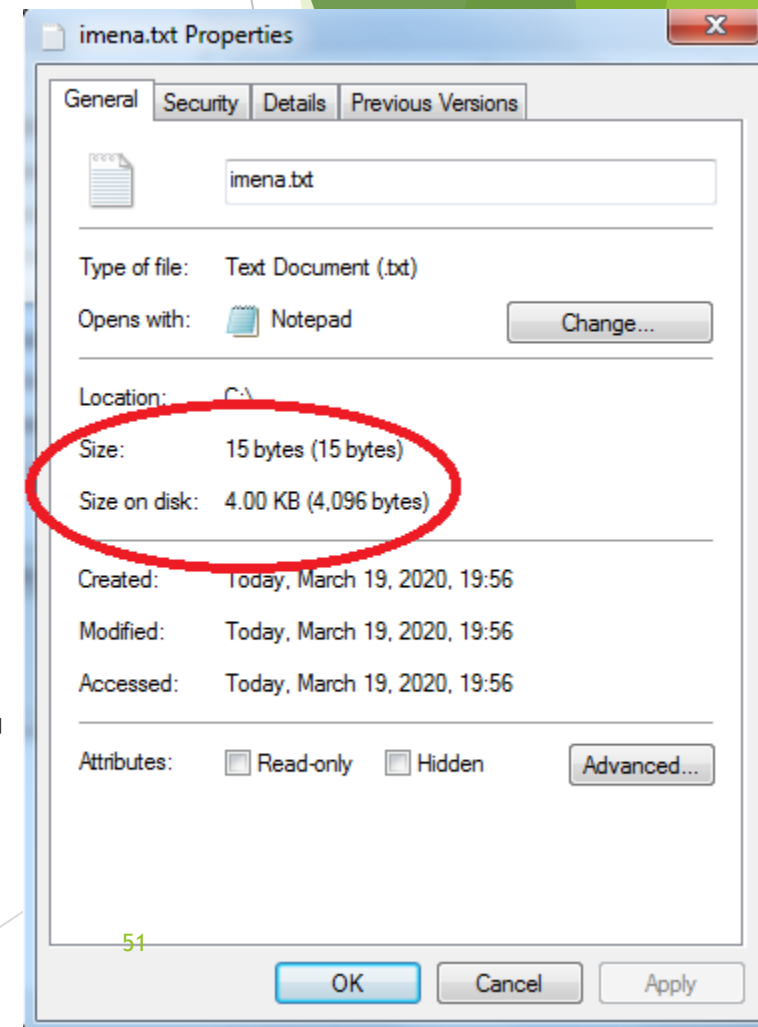
Milica



- ▶ Сачувајте текст као датотеку (нпр. imena.txt) на свом диску. Колика је величина ове датотеке? Подразумевано кодирање текста (Encoding: ANSI) које користи Notepad троши по један бајт за свако енглеско слово и два бајта за сваки прелазак у нови ред притиском на тастер Enter. Према томе, очекивано би било да величина ове датотеке буде $5 + 2 + 6 + 2 = 15$ бајтова.

Резултат експеримента

- ▶ Међутим, када погледамо системски дијалог Properties који, између осталих својстава датотеке, показује и величину датотеке видимо да је величина датотеке заиста 15 бајтова, али да на диску датотека заузима 4 KiB = 4096 бајтова.
- ▶ Пошто знамо да датотека мора да заузме целобројан број кластера (дакле, бар један), резултат експеримента доказује да је величина кластера на овом диску 4 KiB и да је датотека заузела цео кластер, а да при том користи мање од 1% његовог капацитета. Дакле, није исплативо правити мале датотеке на диску попут imena.txt јер траће простор.



Форматирање ниског нивоа

- ▶ Диск се пре употребе мора форматирати:
 - ▶ Прво форматирање или форматирање ниског нивоа (енгл. Low level formatting) врши сам произвођач како би се на диску дефинисале стазе и сектори и омогућио упис података. На овом нивоу само фирмвер и драјвери могу да обављају операције уписа на диск, односно читања са диска.

Партиционисање диска

- ▶ Корисник мора, пре другог форматирања диска, дефинисати логичке уређаје, односно партиције (енгл. partition, лат. pars, partis = део) на диску. Цео диск може бити једна партиција, али је уобичајено да се диск подели (логички, не физички) на две или више партиција које ОС третира као посебне уређаје.
- ▶ **НАПОМЕНА:** прича о партицијама излази из оквира овог курса, тако да не можемо овде прецизно објаснити шта су примарне, проширене (енгл. extended) и логичке партиције, односно шта су логички уређаји (енгл. volume, logical drive) и која је суптилна разлика између логичких партиција и логичких уређаја. Ради једноставности излагања, пошто најчешће једна логичка партиција садржи један логички уређај, нећемо правити разлику између њих и зваћемо их просто партиција или логички уређај.

Computer Management

File Action View Help

Computer Management console tree:

- System Tools
 - Task Scheduler
 - Event Viewer
 - Shared Folders
 - Local Users and Groups
 - Performance and Monitoring
 - Device Manager
 - Storage
 - Disk Management
 - Services and Applications

Volume	Layout	Type	File System	Status	Capacity	Free Space	% Free	Fault Tolerance	Overhead
(C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Primary Partition)	211.28 GB	148.16 GB	70 %	No	0%
Data (D:)	Simple	Basic	NTFS	Healthy (Primary Partition)	217.12 GB	150.04 GB	69 %	No	0%
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary Partition)	100 MB	72 MB	72 %	No	0%

Disk 0: Basic, 465.76 GB

Volume	Layout	Type	File System	Status	Capacity	Free Space	% Free	Fault Tolerance	Overhead
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary Partition)	100 MB	72 MB	72 %	No	0%
(C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Primary Partition)	211.28 GB	148.16 GB	70 %	No	0%
Data (D:)	Simple	Basic	NTFS	Healthy (Primary Partition)	217.12 GB	150.04 GB	69 %	No	0%
Unallocated									37.26 GB

CD-ROM 0: DVD (E:), No Media

CD-ROM 1: DVD (H:), No Media

Legend: ■ Unallocated ■ Primary partition

Device Manager

File Action View Help

Device Manager tree:

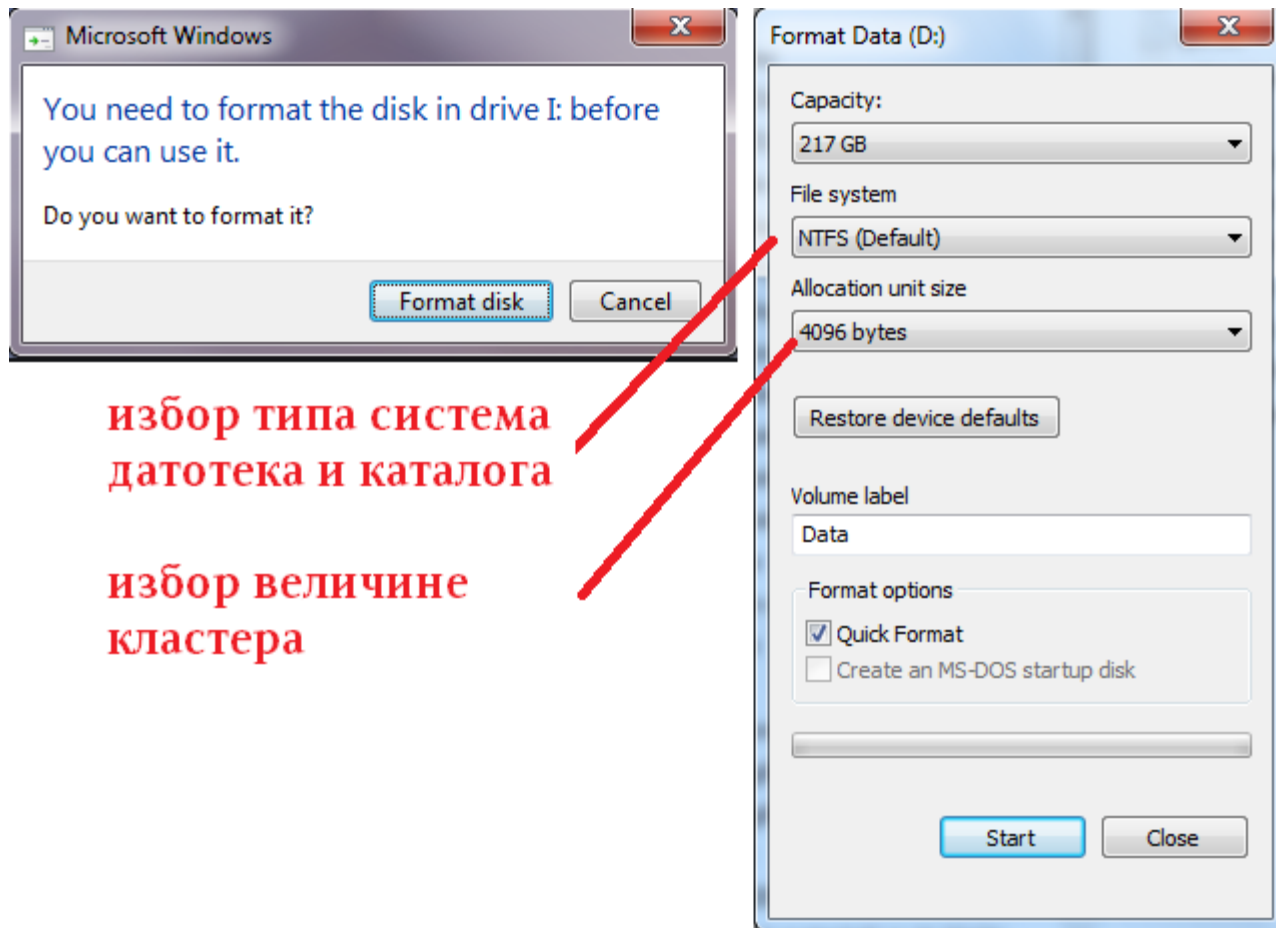
- brucelee2018
 - Batteries
 - Computer
 - Disk drives
 - Samsung SSD 860 EVO 500GB ATA Device
 - Display adapters
 - ATI Mobility Radeon HD 4570
 - DVD/CD-ROM drives
 - Human Interface Devices
 - USB Input Device
 - IDE ATA/ATAPI controllers
 - IEEE 1394 Bus host controllers
 - Imaging devices
 - Keyboards
 - Mice and other pointing devices

► Један SSD и две партиције (C: и D:)

Форматирање високог нивоа

- ▶ Друго форматирање врши оперативни систем и то не целог диска, већ сваке појединачне партиције посебно како би могао да је користи као спољну меморију. Том приликом се на свакој партицији посебно дефинишу:
 - ▶ величина, адреса и статус кластера,
 - ▶ систем датотека и каталога партиције.
- ▶ Форматирање високог нивоа могу да обаве и крајњи корисници; на пример, када прикључе USB-флеш меморију коју ОС не може да прочита, неретко ОС избаци поруку којом нуди корисницима да форматирају уређај. Међутим, ако не желите да уништите све податке, најбоље је покушати прикључивање још пар пута, укључујући и прикључивање флеша на другим рачунарима, пре него што се одлучите на овај драстични корак.

Пример форматирања високог нивоа



**избор типа система
датотека и каталога**

**избор величине
кластера**

О физичкој организацији датотечног система, кластерима и табели алокације датотека

► Прочитати онлајн материјал Цветане Крстев

[http://poincare.matf.bg.ac.rs/~cvetana/Nastava/Materijal/
Softver_v1.pdf](http://poincare.matf.bg.ac.rs/~cvetana/Nastava/Materijal/Softver_v1.pdf)

од 4. стране (пасус „Deo operativnog sistema koji obavlja jedan od najvažnijih poslova je datotečki sistem“) до наслова *Услужни програми* на 10. страни.

Логичка организација система датотека и каталога

- Систем датотека и каталога (енгл. file system, скр. FS) дефинише на који начин ће датотеке бити организоване на диску и како ће се та информација чувати. Постоји више типова FS-а од којих се неки употребљавају само на одређеним уређајима. С друге стране, на истом HDD-, SSD-, USB-флеш уређају различите партиције могу користити различите типове FS-а: NTFS, FAT32, exFAT, ext3 итд.

Метаподаци датотеке

- ▶ Већ смо видели да је датотека (енгл. file) заправо позитиван цео број кластера (један или више).
- ▶ Ради лакшег коришћења датотека, неопходно је да се о свакој од њих чувају одређене информацијама којима се по потреби може лако приступити:
 - ▶ назив и тип датотеке,
 - ▶ датум креирања и последње модификације,
 - ▶ величина датотеке,
 - ▶ ко је аутор, односно власник датотеке,
 - ▶ које операције су дозвољне над датотеком (читање, измена садржаја, извршавање) и ко има право да их обавља (само власник, одређена група корисника, свако).

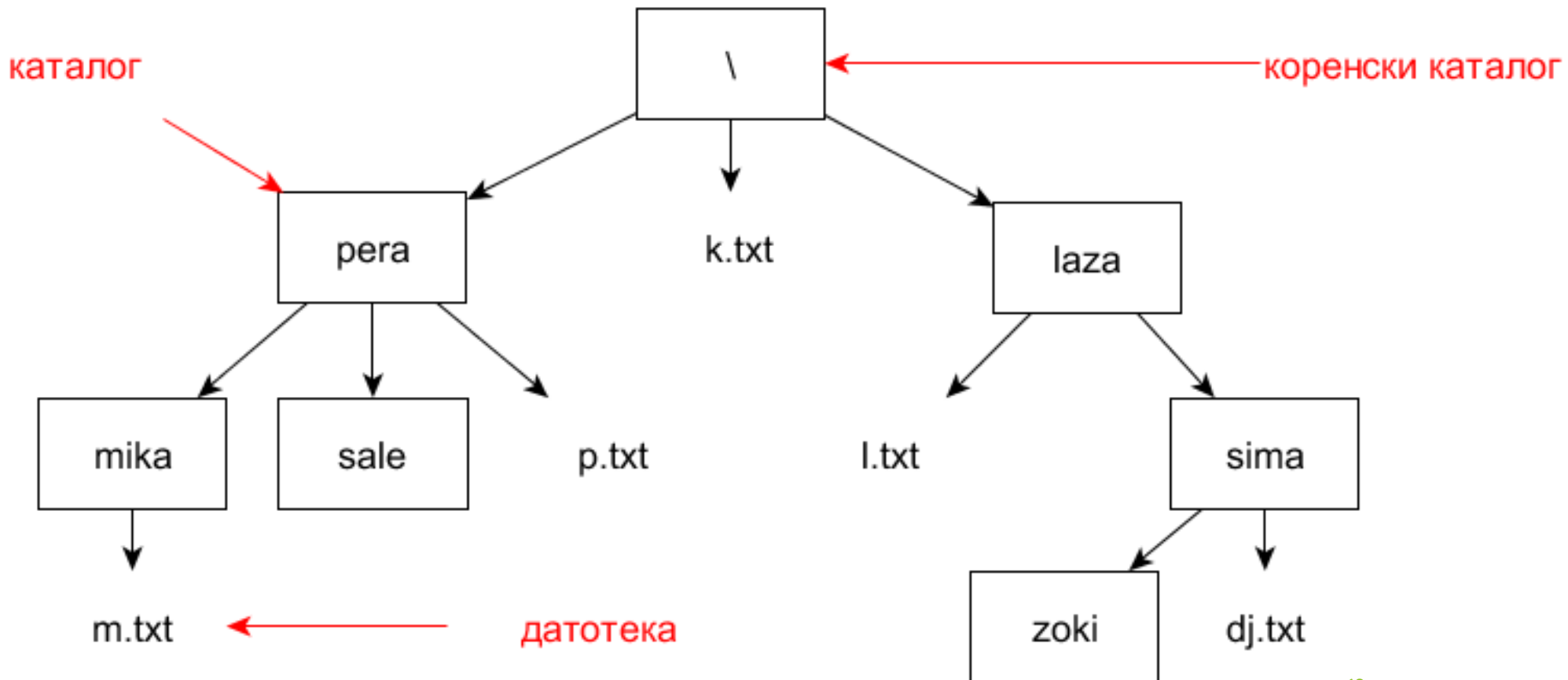
Каталози

- ▶ Најједноставнији каталог (енгл. folder, directory) је специјална датотека која је заправо листа информација (метаподатака) о другим датотекама, при чему се метаподаци о једном примерку одређене датотеке морају налазити у тачно једном каталогу.
- ▶ Каталог може бити и празан, а исто тако може, поред метаподатака о датотекама, да садржи и метаподатке о другим каталозима за које се користи израз поткаталог (енгл. subfolder). И за метаподатке о примерку одређеног поткаталога важи да се морају налазити у тачно једном каталогу.
- ▶ Програми као што је My Computer приказују садржаје каталога, тј. листу информација о датотекама и поткаталозима. С друге стране, корисник има доживљај да су саме датотеке и поткаталози садржај каталога иако то у стварности није тако, већ су то само одговарајући метаподаци.

Коренски каталог и хијерархија

- ▶ Тек направљени систем датотека и каталога партиције на почетку садржи један једини празан каталог – **коренски каталог** (енгл. root folder, root directory). То је једини каталог који није поткаталог ниједног другог каталога на партицији. Ознака коренског каталога зависи од оперативног система: Windows користи обрнуту косу црту \ (енгл. backslash), док Unix и његови деривати (Linux, FreeBSD), MacOS означавају коренски каталог (обичном) косом цртом / (енгл. slash).
- ▶ Сваки накнадно направљени каталог или датотека морају се наћи у оквиру неког претходно постојећег каталога на истој партицији. На тај начин се систем датотека и каталога партиције може представити једном хијерархијском структуром у облику дрвета („породичног стабла) где гране повезују каталог са датотекама и поткаталозима чије метаподатке садржи. Дрво се црта тако да је коренски каталог на врху, а „гранање“ се врши наниже. Ако постоји грана од чвора дрвета А (каталог) наниже ка чвору дрвета Б (поткаталог или датотека) у значењу да каталог А садржи поткаталог или датотеку Б, уобичајено је да се каже да је „А отац од Б“ или да је „Б дете од А“, односно да је „А родитељски (надређени) каталог за Б“.

Систем датотека и каталога уређаја (Windows) као дрвоидна структура („породично стабло“)



Адресирање уређаја

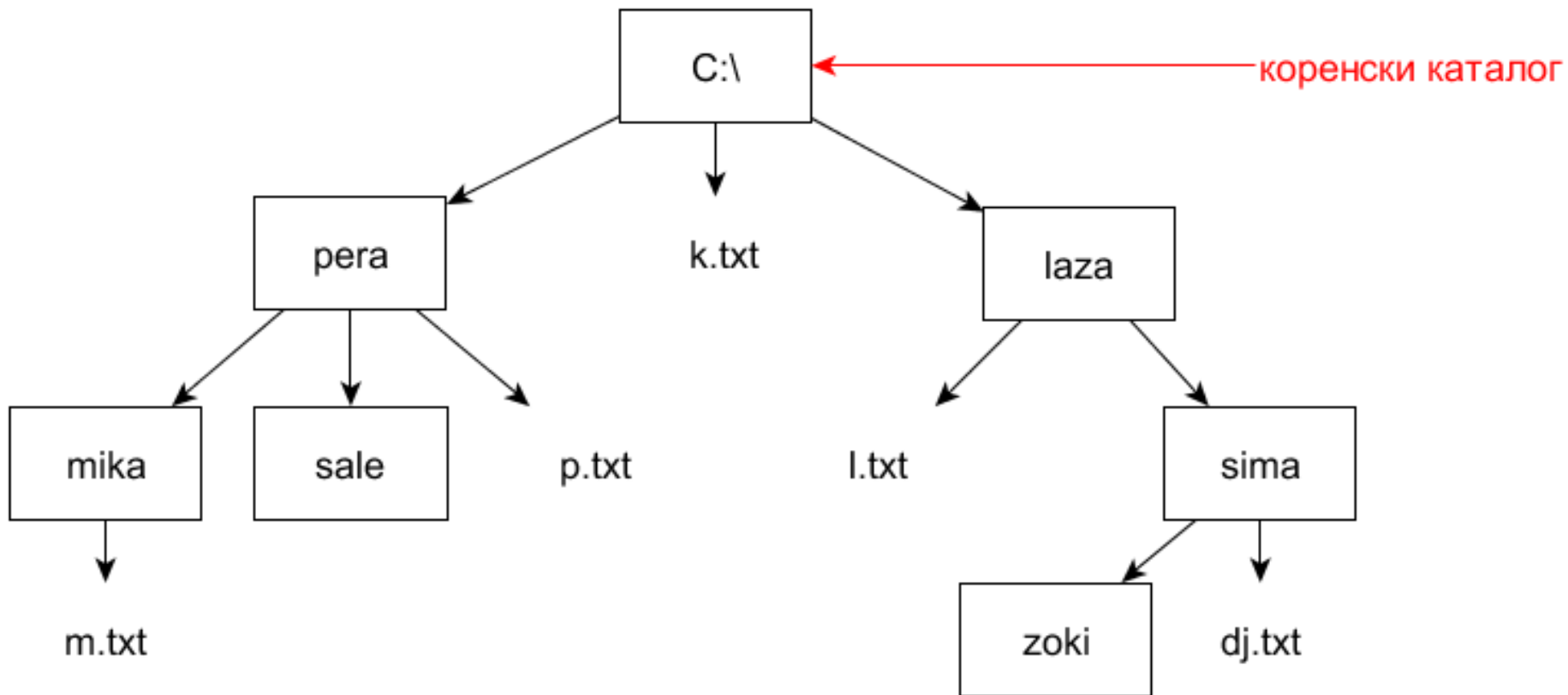
- ▶ Неки оперативни системи, (UNIX и деривати, Mac OS), адресирају уређаје на исти начин као датотеке и каталогe.
- ▶ Оперативни систем MS Windows пак користи слова енглеске абецеде праћене двотачком за адресирање уређаја (A:, B:, C:, D: итд.) при чему важи следеће:
 - ▶ слова A: и B: су унапред заузета за дискетне уређаје који се више не користе (A: за уређај намењен дискетама од 3,5 инча, B: за уређај намењен дискетама од 5,25 инча).
 - ▶ уобичајено је да партиције уређаја на коме је оперативни систем (обично HDD, SSD) добију следећа слова (C: ...), а онда следе остали уређаји са секундарним меморијама (додатни HDD, SSD, уређај за DVD, USB-флеш итд.).

Пример адресирања уређаја (MS Windows)

- ▶ Претпоставимо да се у систему налазе два HDD-а и да се најпре њиховим партицијама додељују слова уређаја. Ако при диск има 3 партиције, а други само две, адресе уређаја би биле расподељене на следећи начин:
- ▶ A: дискетни уређај за дискете од 3,5 инча;
- ▶ B: дискетни уређај за дискете од 5,25 инча;
- ▶ C: прва партиција првог HDD-а;
- ▶ D: друга партиција првог HDD-а;
- ▶ E: трећа партиција првог HDD-а;
- ▶ F: прва партиција другог HDD-а;
- ▶ G: друга партиција другог HDD-а;
- ▶ H: следећи уређај...

Апсолутно адресирање датотека и каталога

- ▶ Апсолутна адреса датотеке или каталога се састоји из два дела:
 - ▶ адресе уређаја на коме се датотека или каталог налази и
 - ▶ путање од коренског каталога уређаја до саме датотеке или каталога. Путања је састављена од назива свих каталога у које треба ући (почев од коренског) да би се дошло до циља, укључујући и име циљног каталога, односно датотеке. Називи каталога (и евентуално циљне датотеке) су раздвојене обрнутом косом цртом (енгл. `backslash`). Обрнута коса црта се користи и као ознака коренског каталога на самом почетку путање.



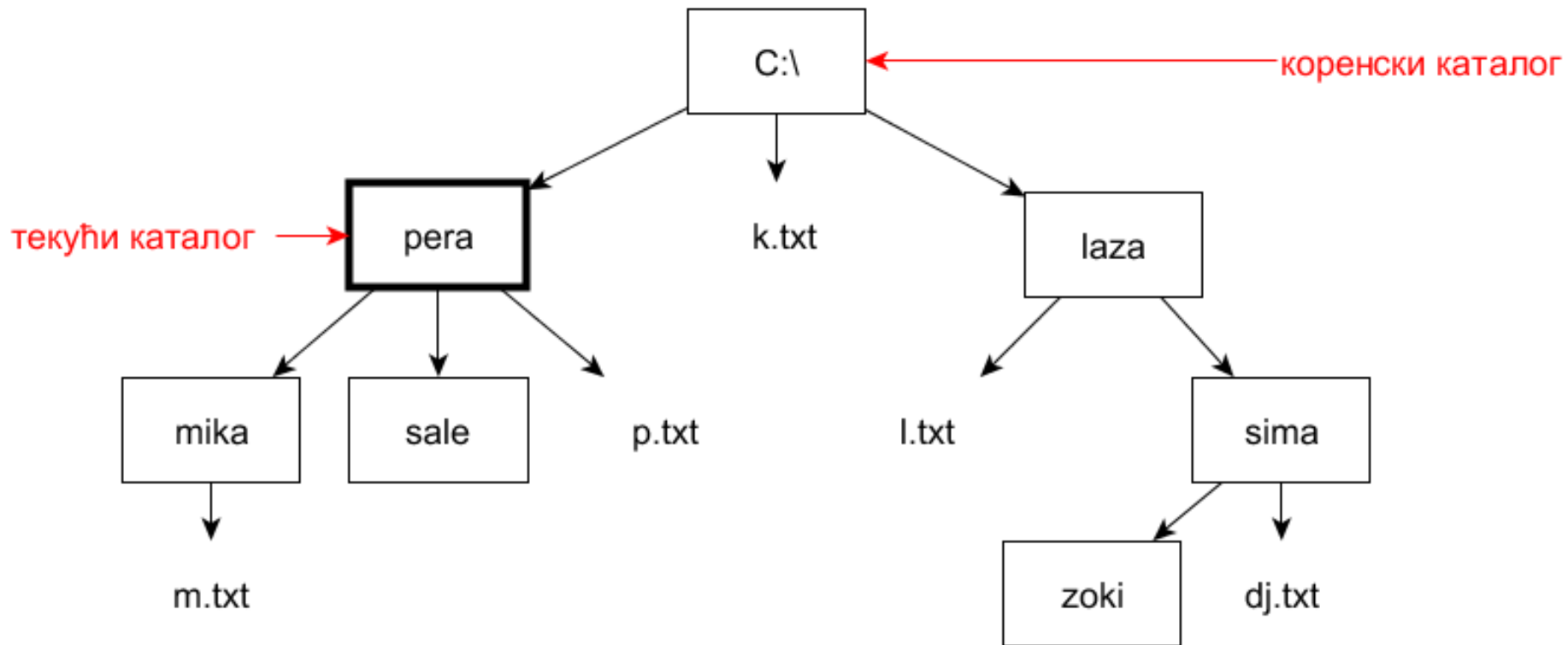
Датотека или каталог	Апсолутна адреса (црвено је обојена ознака коренског каталога)
Коренски каталог прве партиције првог HDD-а	C:\
zoki	C:\laza\sima\zoki
m.txt	C:\pera\mika\m.txt

Релативно адресирање датотека и каталога

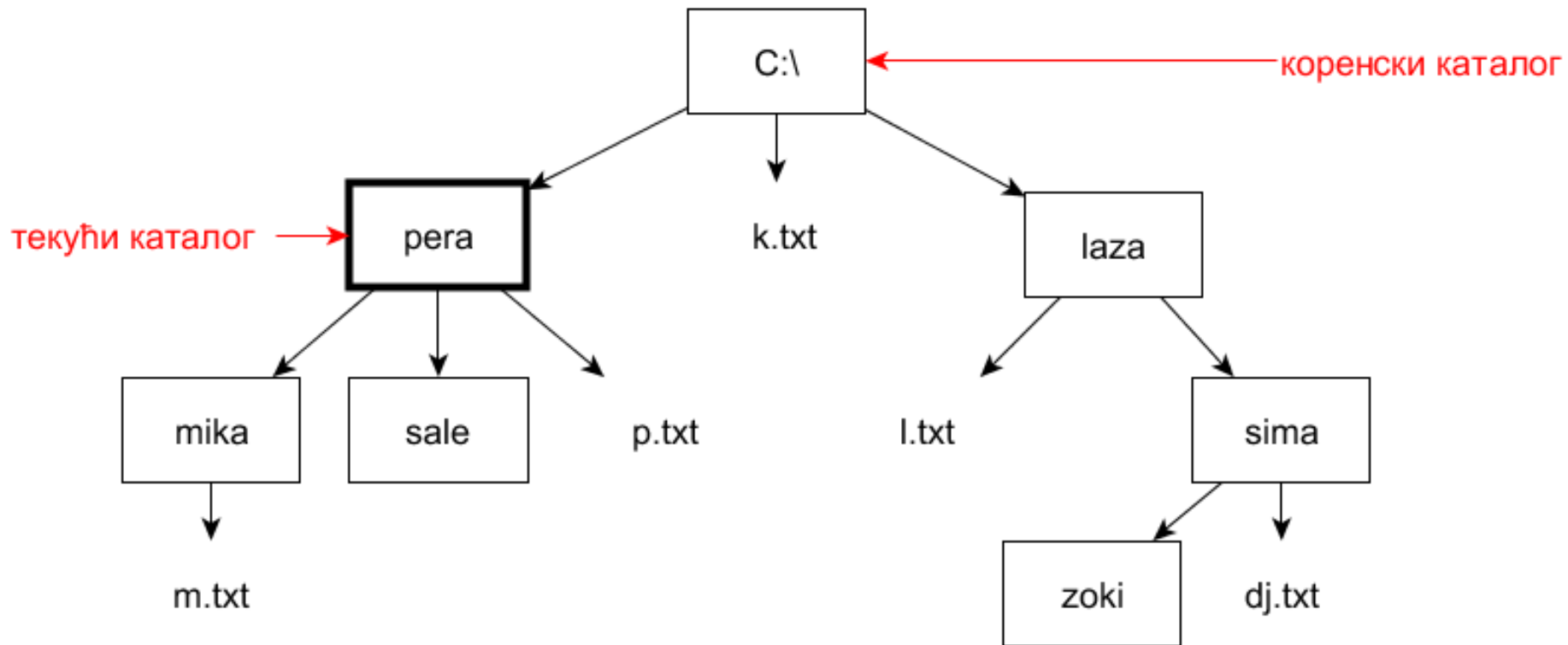
- ▶ Овде ћемо размотрити само релативно адресирање датотека и каталога у односу на текући каталог, тј. каталог у коме се корисник тренутно налази (на пример, када у програму My Computer анализира садржај неког каталога).
- ▶ За разлику од апсолутне адресе која представља путању од коренског каталога до циља, релативна адреса представља (најкраћу) путању од текућег каталога до циља. Самим тим, релативна адреса није јединствена и нема смисла уколико је текући каталог непознат.

Правила релативног адресирања

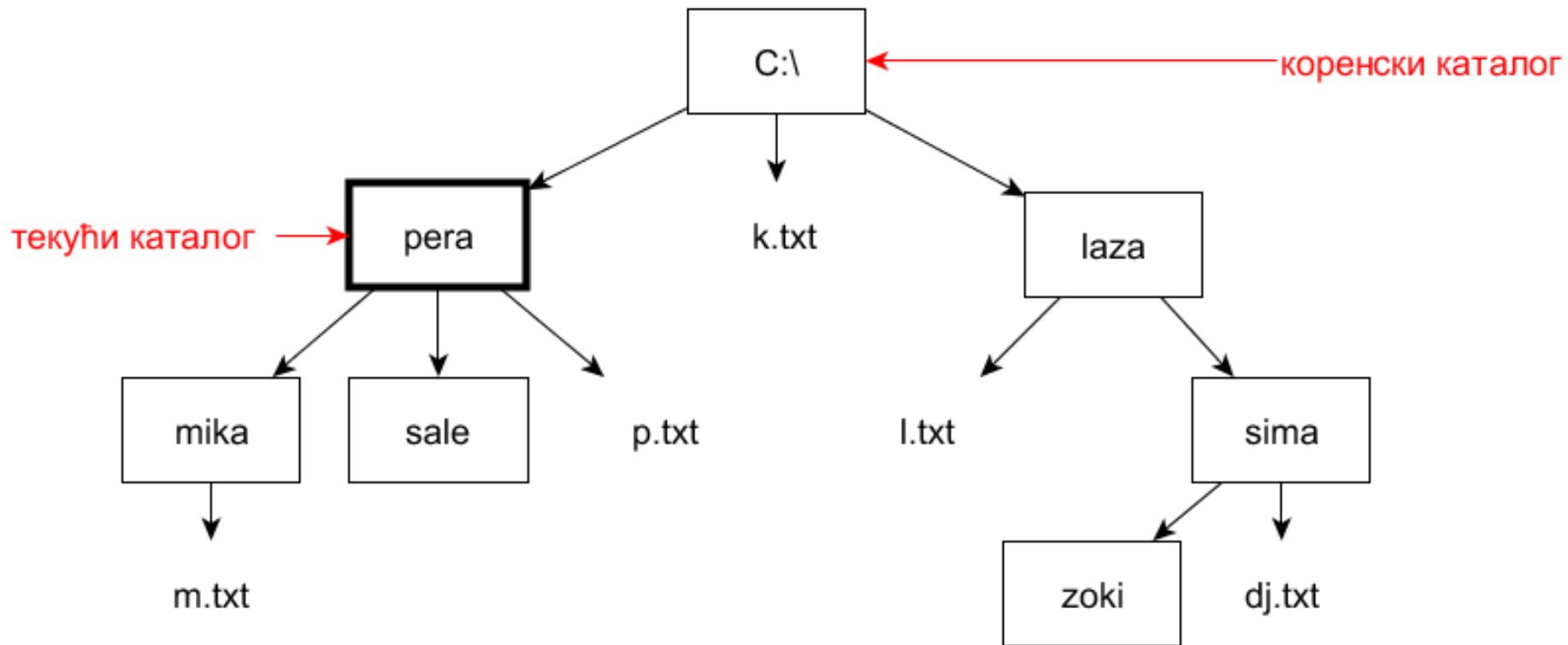
- ▶ Путањом у релативној адреси се не описује где се налазимо у дрвету већ куда треба ићи да би се стигло до циљног каталога или датотеке. Из тог разлога је погрешно наводити назив текућег каталога на почетку путање. Једини изузетак од овог правила је навођење тачке на почетку као адресе текућег каталога, али је то непотребно и може да се помеша са претходним правилом, те тај изузетак треба избегавати. Дакле, наводи се само куда треба ићи.
- ▶ У случају да се крећемо низ дрво, тј. прелазимо од чвора ка неком његовом детету, неопходно је навести име тог детета јер један чвор може имати више деце и самим тим кретање наниже није једнозначно одређено.
- ▶ Када се крећемо навише, тј. од чвора детета ка чвору родитељу, наводе се искључиво две тачке по хоризонтали (..), тј. никад се не наводи име родитељског чвора. Наиме, у дрвету сваки чвор (изузев корена) има тачно једног родитеља, те се само на један начин може ићи навише. С друге стране, навођење имена чвора родитеља у адреси би се погрешно протумачило као кретање наниже уместо навише.



Датотека или каталог	Релативна адреса у односу на текући каталог (pera)
pera	.
p.txt	p.txt
mika	mika
m.txt	mika\m.txt
коренски каталог \	..



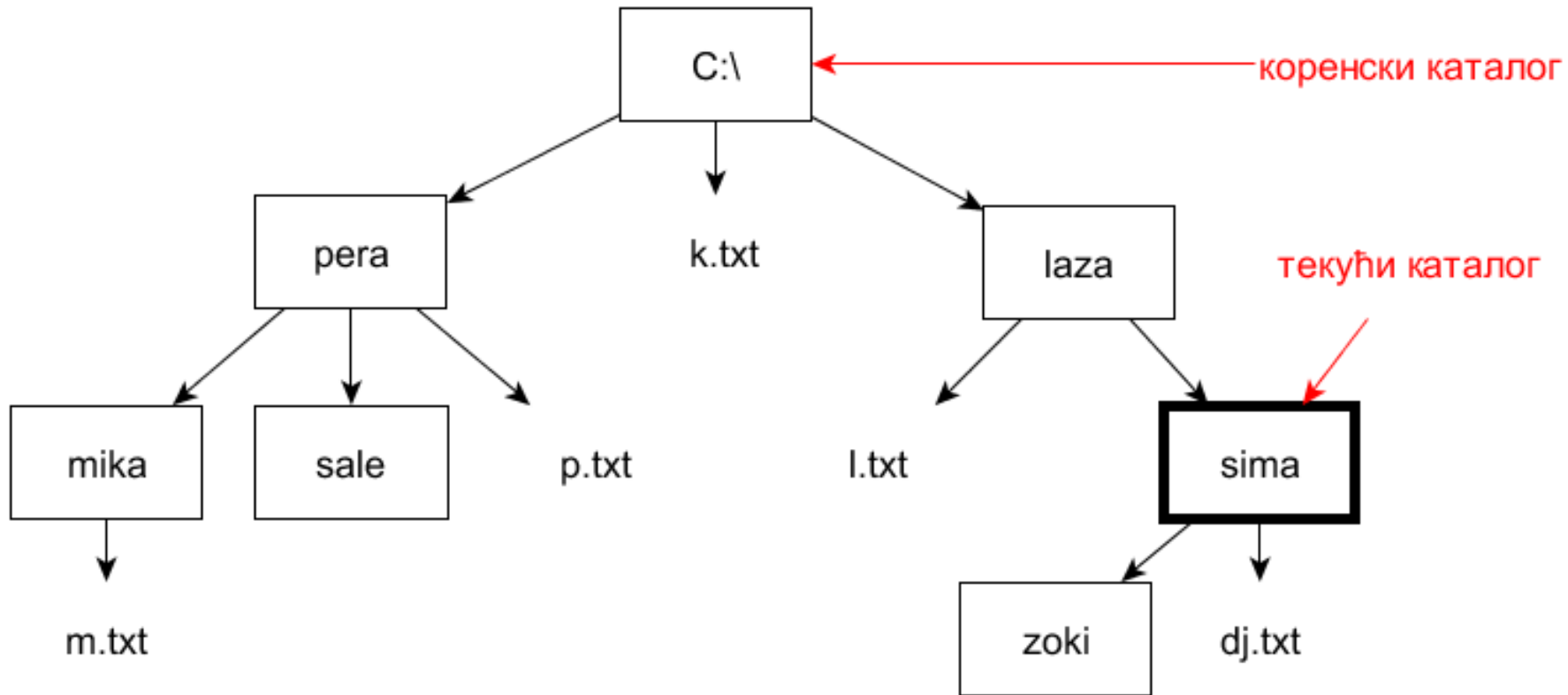
Датотека или каталог	Релативна адреса у односу на текући каталог (pera)
k.txt	..\k.txt
laza	..\laza
l.txt	..\laza\l.txt
sima	..\laza\sima
dj.txt	..\laza\sima\dj.txt



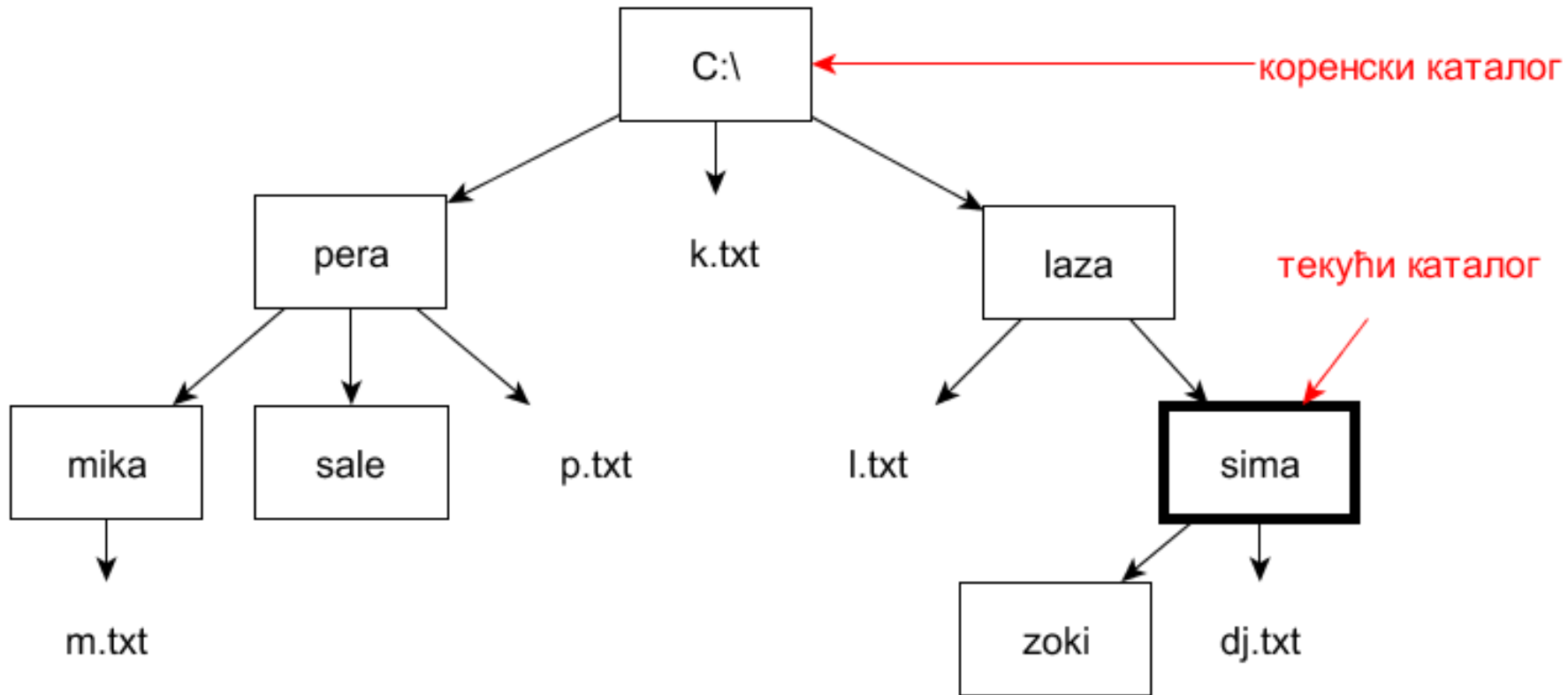
Датотека или каталог	Релативна адреса у односу на текући каталог (pera)
sale	sale
zoki	..\laza\sima\zoki

Релативне адресе и промена текућег каталога

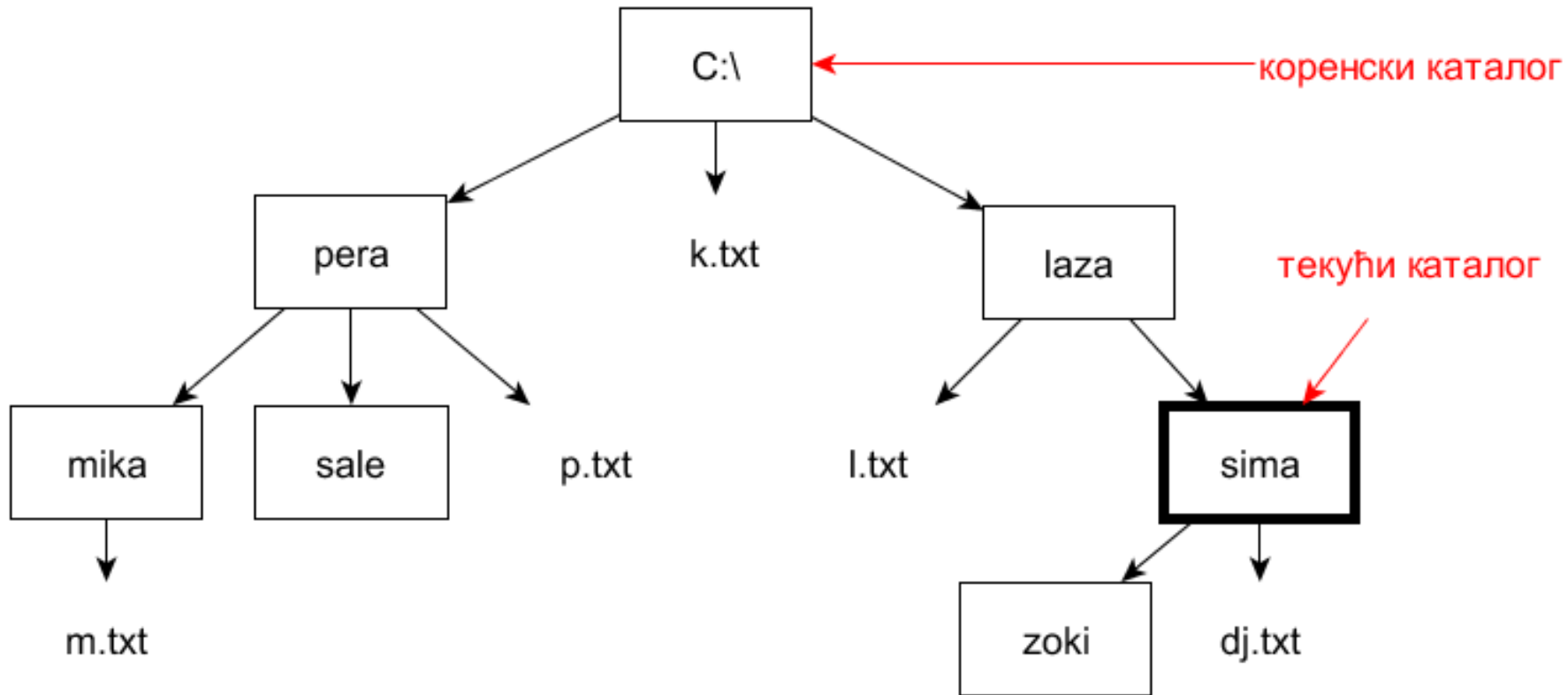
- ▶ Релативна адреса нема смисла ако се не зна шта је текући каталог.
- ▶ Променом текућег каталога мењају се и релативне адресе свих датотека и каталога. [Претходни пример](#) (текући каталог `pera`) и [следећи пример](#) (текући каталог `sima`) управо илуструју како се релативне адресе истих датотека и каталога мењају са променом текућег каталога.



Датотека или каталог	Релативна адреса у односу на текући каталог (sima)
sima	.
zoki	zoki
dj.txt	dj.txt
laza	..
l.txt	..\l.txt



Датотека или каталог	Релативна адреса у односу на текући каталог (sima)
коренски каталог \	..\..\
k.txt	..\..\k.txt
pera	..\..\pera
sale	..\..\pera\sale
p.txt	..\..\pera\p.txt



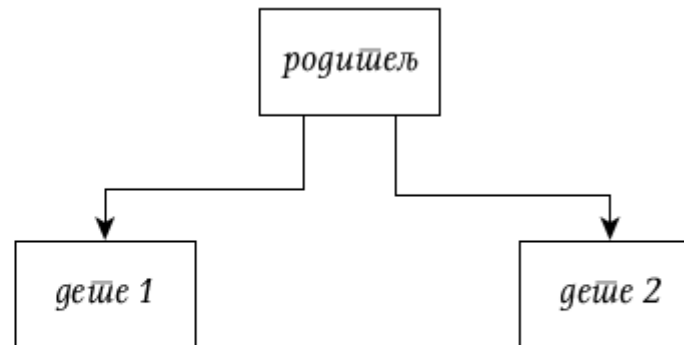
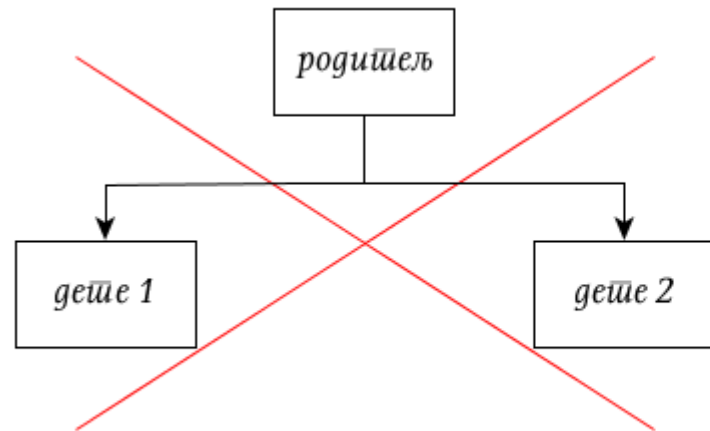
Датотека или каталог	Релативна адреса у односу на текући каталог (sima)
mika	..\..\pera\mika
m.txt	..\..\pera\mika\m.txt

Правила чијим непоштовањем се праве најчешће грешке (1)

- ▶ Никад се не наводи име текућег каталога у релативној адреси.
- ▶ Никад се не наводи име родитељског каталога у релативној адреси.
- ▶ Имена каталога се наводе у релативној адреси само када идемо наниже по дрвету.

Правила чијим непоштовањем се праве најчешће грешке (2)

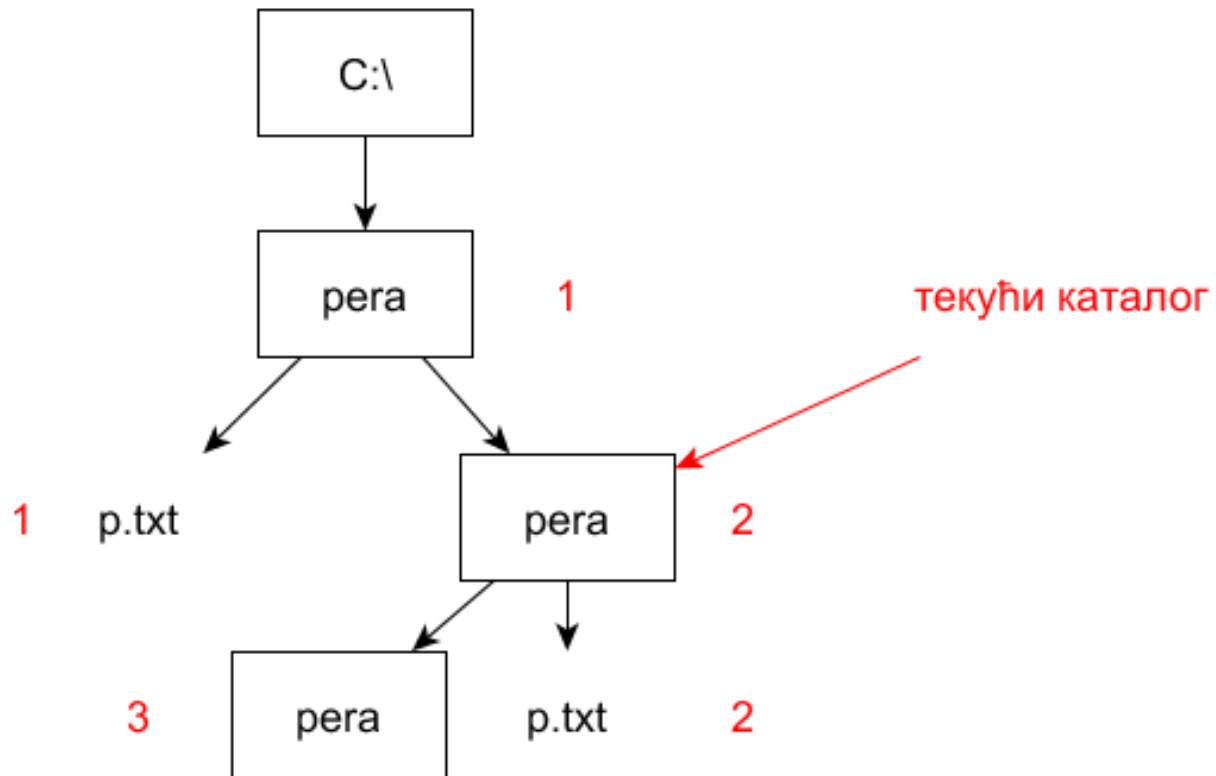
- ▶ Гране које спајају браћу (сестре) са родитељским чвором су одвојене, тако да једини пут од једног братског (сестринског) чвора до другог води преко родитеља.



Примери за вежбу

- ▶ Следећи три примера могу да послуже као вежба да ли је релативно адресирање савладано или не.
- ▶ Решења су дата само за први пример, преостале примере пробајте да урадите сами.

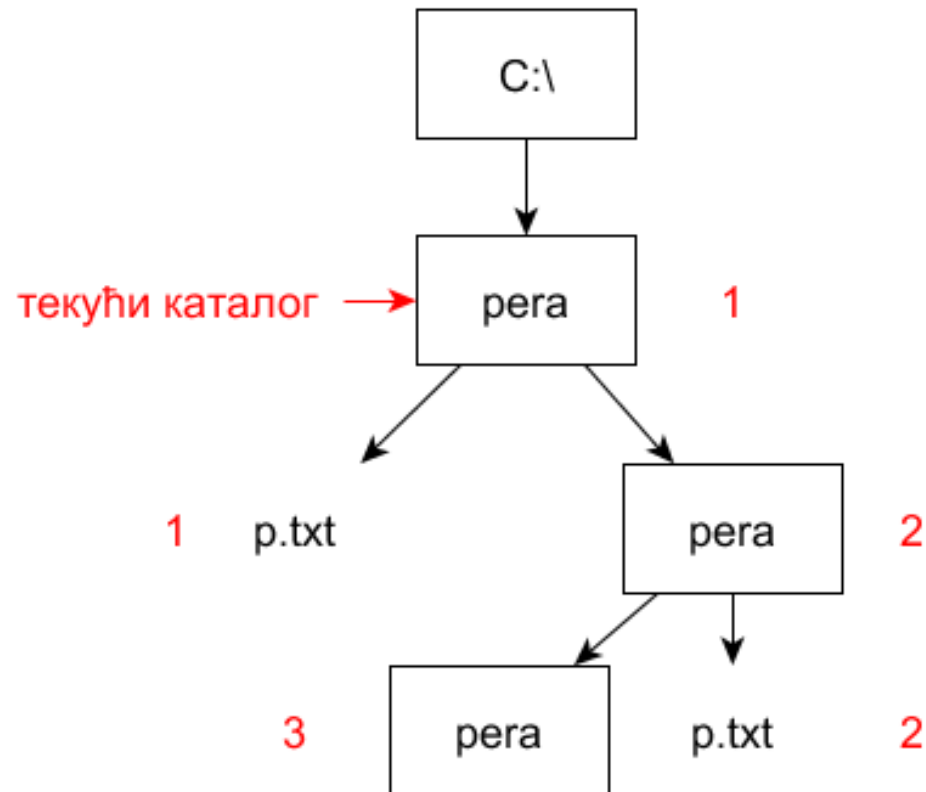
Вежба 1



Датотека или каталог	Релативна адреса у односу на текући каталог (pera , означен са 2)
коренски каталог \	..\..
pera (1)	..
pera (2)	.
pera (3)	pera
p.txt (1)	..\p.txt
p.txt (2)	p.txt

Вежба 2

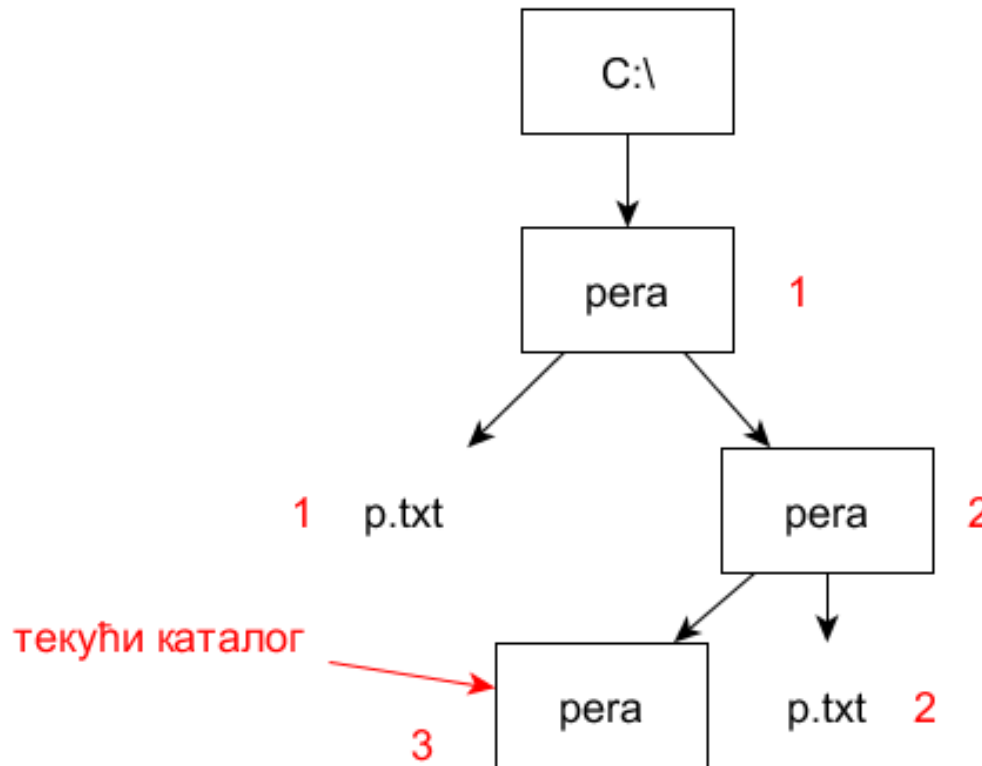
Покушајте да следеће адресе урадите сами.



Датотека или каталог	Релативна адреса у односу на текући каталог (pera, означен са 1)
коренски каталог \	
pera (1)	
pera (2)	
pera (3)	
p.txt (1)	
p.txt (2)	

Вежба 3

Покушајте да следеће адресе урадите сами.



Датотека или каталог	Релативна адреса у односу на текући каталог (pera, означен са 3)
коренски каталог \	
pera (1)	
pera (2)	
pera (3)	
p.txt (1)	
p.txt (2)	

Фрагментација

- ▶ Приликом попуњавања празне радне меморије процесима, односно празне измењиве спољне меморије датотекама и каталозима, у почетку се простор искоришћава редом и додељује као низ узастопних меморијских ћелија, односно кластера. Међутим, временом процеси престају са радом или буду измештени у виртуелну меморију, а истовремено почињу са радом неки други, па се у радној меморији појављују наизменично заузети и слободни делови. Слично, на измењивим спољним меморијама се мења садржај, тј. једне датотеке и каталози се бришу, а други настају, па се и у низу кластера наизменично смењују заузети и слободни поднизови. У оба случаја се меморија „распарчава“ и више није могуће доделити непрекидни меморијски простор процесу, датотеци или каталогу, већ им се додељују више физички одвојених „парчића“ (фрагмената) слободног простора. Сама појава се назива **фрагментација**.

Дефрагментација

- ▶ Фрагментација утиче на перформансе рачунарског система јер временом постаје све теже пронаћи довољан број фрагмената слободног простора за нови процес, датотеку или каталог, а на спољној меморији као што је HDD, губи се време на серијско проналажење свих кластера датотеке или поткаталога који могу бити разбацани по диску, а додатно се троше механички делови који контролишу главу за читање.
- ▶ У случају HDD-а се проблем фрагментације решава тако што се посебним софтвером садржај заузетих кластера преписује тако да датотеке заузимају суседне кластере и заузимају просто редом, као на почетку када је диск био празан. Поменути процес преписивања се назива дефрагментација.
- ▶ У случају SSD-а дефрагментација је непотребна јер се сваком блоку приступа директно по адреси.

Чување информација о партицијама

- ▶ На сваком уређају спољне меморије (HDD, SSD, USB флеш меморија) одваја се један део меморије који чува:
 - ▶ партициону табелу уређаја, тј. листу адреса на којима почињу појединачне партиције уређаја, као и величине тих партиција.

Basic Input Output System (BIOS)

- ▶ Основни систем за улаз и излаз на персоналним рачунарима (ПР), познатији по енглеском акрониму BIOS, је фирмвер који се налази се на матичној (родитељској) плочи рачунара у оквиру посебне меморије (раније ROM, данас EEPROM или флеш меморија, тако да BIOS може да се ажурира).
- ▶ Када се укључи ПР, први софтвер који процесор извршава је BIOS, тј. увек се у бројач инструкција упише адреса прве инструкције BIOS-а.
- ▶ На почетку рада BIOS проверава који све уређаји постоје у рачунарском систему, омогућава да почну са радом (иницијализује их) и тестира њихову исправност. Према томе, BIOS садржи основне програме за:
 - ▶ тестирање радне меморије,
 - ▶ тестирање уређаја са секундарним меморијама (HDD, SDD, уређаји за оптичке дискове типа DVD и слично, USB-флеш итд.) и
 - ▶ тестирање и контролу основних периферијских уређаја — тастатуре и монитора.

„Подизање“ оперативног система (bootstrapping, boot)

- ▶ Приликом тестирања уређаја са секундарним меморијама, BIOS покушава да пронађе партицију на којој је инсталиран оперативни систем, а затим језгро ОС-а (енгл. kernel) учитава у радну меморију и њему предаје контролу, тј. надаље језгро оперативног система управља рачунарским системом. Тај процес се назива „подизање“ ОС-а (енгл. bootstrapping, boot = „гајка на чизми“) по угледу на приче барона Минхаузена:
https://upload.wikimedia.org/wikipedia/commons/3/3b/Muenchhausen_Herrfurth_7_500x789.jpg
- ▶ BIOS има одвојену додатну меморију на матичној плочи у којој се записују параметри које BIOS користи сваки пут када се укључи рачунар. Та меморија се назива CMOS. CMOS чува информације о спољним меморијама које су саставни део система (HDD, уређаји за оптичке дискове типа DVD RW и сл.), омогућава подешавање системског времена (пре инсталације ОС-а), као и промену редоследа по коме ће BIOS тражити ОС на уређајима са спољним меморијама (енгл. boot sequence).

Boot sequence и системско време

- ▶ Пре него што се систем инсталира, обично се подеси у CMOS-у да BIOS најпре тражи ОС на оној спољној меморији на којој се налази инсталациони диск или флеш, а на крају се редослед промени тако да BIOS најпре тражи ОС на оном уређају на коме је ОС инсталиран чиме се штеди време подизања ОС-а.
- ▶ CMOS се раније чувао у брзој електронској меморији која је губила садржај ако нема електричног напајања, па се на матичној плочи користила литијумска батерија да чува садржај CMOS-а када рачунар не ради. Данас се CMOS чува у EEPROM-меморији, а батерија се користи да чува подешавања системског часовника (када се потроши, системско време се обично приказује као 1. јануар 1900. године, што упозорава корисника да треба да је замени).

Начини на које се ОС учитава и активира

- ▶ Описани начин на који BIOS проналази, учитава оперативни систем у радну меморију и активира га је пример аутоматског пуњења ОС-а. Поред аутоматског пуњења постоје још и
 - ▶ **полуаутоматско пуњење** (приликом пуњења је потребна интервенција и контрола особе која надгледа систем - оператера; чест случај у великим индустријским системима).
 - ▶ **резидентни ОС** (оперативни систем је већ „напуњен“, тј. све време је присутан у некој трајној меморији (ROM, EEPROM) која је уједно и радна меморија, тако да ОС почиње да ради истовремено са системом и не захтева пуњење).

Више оперативних система на истом рачунару

- ▶ С обзиром да се приликом поделе на партиције само једна партиција може прогласити за активну (тј. за партицију са које ће се учитати оперативни систем), поставља се питање да ли је могуће и како инсталирати више оперативних система на истом рачунару, тако да се на почетку рада рачунара изабере жељени систем?
- ▶ Одговор је да, могуће је инсталирати више оперативних система на истом рачунару, али на различитим партицијама секундарних меморија. Проблем избора ОС-а на почетку рада рачунара решава софтвер који управља подизањем рачунара (енгл. boot manager). Пример таквог софтвера је GRUB.
- ▶ Свака партиција са инталираним оперативним системом има свој програм за учитавање ОС-а или „пуњач“ (енгл. loader), а софтвер попут GRUB-а одржава њихов списак са адресама. Када корисник покрене рачунар и BIOS потражи ОС, покренуће GRUB. GRUB ће онда понудити кориснику листу инсталираних оперативних система и на основу направљеног избора покренуће одговарајући програм за учитавање ОС-а.

GRUB

```
GNU GRUB version 2.00-19ubuntu2

Linux Mint 16 Cinnamon 64-bit, 3.11.0-12-generic (/dev/sda2)
Linux Mint 16 Cinnamon 64-bit, 3.11.0-12-generic (/dev/sda2)
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)
Windows 8 (loader) (on /dev/sda1)
```

Горња слика је преузета 14. марта 2020. године са адресе <https://www.techmesto.com/set-windows-as-default-in-linux-dual-boot/> Илуструје листу инсталираних оперативних система и њихових „пуњача“, међу којима су Linux Mint 16 и Windows 8.

Виртуелизација

- ▶ Неки други пут.

Злонамерни софтвер

Дефиниција

- ▶ Пре него што размотримо остале врсте системског софтвера (пре свега услужни софтвер), неопходно је да опишемо злонамерни или малициозни софтвер (енгл. malicious software, malware).
- ▶ Под злонамерним софтвером се подразумева скуп програма чији је циљ да ометају или прекину функционисање рачунарског система, односно да нанесу штету и то:
 - ▶ изазивањем квара на хардверским деловима рачунара,
 - ▶ модификацијом софтвера,
 - ▶ крађом или уништавањем података,
 - ▶ шпијунирањем, уцењивањем, застрашивањем
 - ▶ крађом идентитета корисника,
 - ▶ досађивањем нежељеним информацијама (рекламирањем).

Инфективни софтвер

- ▶ Посебно је опасан инфективни софтвер као подврста злонамерног софтвера која је у стању да се преноси са једног рачунарског система на други, налик инфективним болестима које се преносе са једног живог бића на друго.
- ▶ Пренос се најчешће обавља преко секундарних меморија (екстерни HDD или SSD, USB флеш меморија, некада дискете) које се користе за копирање података са једног рачунара на други. Такође, уколико је један рачунар у рачунарској мрежи „заражен“ инфективним софтвером, врло је вероватно да ће се заразити и остали рачунари. Интернет је такође значајан извор заразе у случају када се преузима садржај непознатог порекла, поготово софтвер.
- ▶ Две најважније врсте инфективног софтвера су вируси и црви.

Вируси

- ▶ Назив су добили по вирусима у природи који су у стању да направе своју копију унутар живог организма (домаћина) и да тако шире заразу на друге организме. Рачунарски вируси су пак програми чијим се извршавањем копира неки њихов део и уграђује у друге датотеке, пре свега програме. Самим тим, извршавањем „заражених“ програма се „зараза“ шири јер се наставља копирање и уграђивање „зараженог“ скупа инструкција у друге датотеке.

Црви

- ▶ Црви представљају злонамерни софтвер који има способност да самог себе копира (репликација) и да се затим копира на све рачунаре који су повезани са „зараженим“ рачунаром. При том, за разлику од вируса, црви не уграђује своје инструкције у друге датотеке.
- ▶ Црви пре свега нападају рачунарске мреже, успоравају мрежни саобраћај и заузимају слободни простор на меморијским уређајима.

Остале врсте злонамерног софтвера

- ▶ тројанци (лажно представљање)
- ▶ бомбе (временски темпиране акције)
- ▶ adware (досађивање рекламама)
- ▶ spyware (шпијунирање, крађа података, лозинки)
- ▶ scareware (заstraшивање)
- ▶ ransomware (уцењивање, нпр. откуп за откључавање рачунара, односно неког његовог уређаја)
- ▶ rootkits (скривени софтвер који модификује ОС и омогућава преузимање контроле са удаљеног рачунара)
- ▶ backdoors (избегавање идентификације којом се спречава приступ неовлашћеним корисницима, омогућавање приступа уљезима)

Софтверске лиценце

Софтвер као интелектуална својина

- ▶ Софтвер се данас третира као интелектуална својина која подлеже ауторским правима. Власници ауторских права над неким софтвером (који не морају бити аутори тог истог софтвера) одређују услове под којима корисници могу да:
 - ▶ извршавају софтвер;
 - ▶ анализирају како је софтвер написан (увидом у изворни кôд);
 - ▶ модификују софтвер;
 - ▶ праве копије софтвера;
 - ▶ дистрибуирају како оригинални софтвер, тако и његове модификације (деривате).
- ▶ Правна формулација наведених услова представља софтверску лиценцу којом се корисницима дају или укидају дозволе за извршавање, анализу, модификацију, копирање и дистрибуцију софтвера. Према томе, пре него што се софтвер први пут користи, односно набави, неопходно је упознати се са његовом лиценцом.

Литература

- ▶ Ова тема је детаљно обрађена у раду:

Цветана Крстев, „Улога и место слободног софтвера у библиотекама и јавном сектору“, У: Зборник радова са међународне научне конференције “Етика у науци и култури”, Септембер 25-27, 2009, Београд, Србија, ур. Александра Вранеш и Љиљана Марковић, стр. 233-247, Филолошки факултет, Универзитет у Београду, ISBN 978-86-86419-97-2, 2010.)

<http://poincare.matf.bg.ac.rs/~cvetana/Nastava/Materijal/Krstev-Etika2009.pdf>

- ▶ Из тог разлога треба прочитати цео чланак, а у наставку следи допуна: објашњење појмова којих у чланку нема, као и осврт на однос пиратског и слободног софтвера.

Дељени софтвер (донаторска верзија)

- ▶ За разлику од бесплатног власничког софтвера (енгл. freeware, скр. FW, на пример Skype, Adobe Acrobat Reader, Winamp, Google Chrome), дељени софтвер (енгл. shareware, скр. SW) се само бесплатно дели, али зато очекује некакву новчану надокнаду.
- ▶ Новчана надокнада не мора бити обавезна; у том случају се ради о софтверу који је на граници између FW и SW и који апелује на корисникову савест да, ако је задовољан програмом и користи га, уплати власнику ауторских права новчану донацију. Отуда и назив донаторска верзија софтвера (енгл. donationware) за ту подврсту дељеног софтвера.

Дељени софтвер (демо-верзија)

- Софтверске лиценце неретко наглашавају да „купац софтвер користи на сопствену одговорност“, тј. продавац не сноси никакву одговорност ако купац због коришћења софтвера изгуби податке или хардверски уређај рачунарског система престане са радом. С друге стране, различите софтверске фирме производе програме који имају исту намену и поставља се питање који од њих одабрати. Уобичајено је да се кориснику понуди демонстрациона (скр. демо-) верзија софтвера (енгл. demonstration software). Постоји неколико сценарија дељења демо-верзије софтвера.

Дељени софтвер (пробна и „обогалјена/осакаћена“ верзија)

- ▶ Један сценарио подразумева да корисник бесплатно преузме пробну верзију софтвера (енгл. trialware) коју може да користи неко ограничено време (нпр. 15-30 дана) или ограничен број пута (броји се свако покретање програма) уз сагласност власника ауторских права, а по истеку тог времена су могући различити исходи:
 - ▶ програм је и даље функционалан, али има ознаку да „није регистрован“, тј. да није купљен и даље коришћење је нелегално;
 - ▶ програм функционише, али корисника стално ометају поруке да треба да га региструје; поруке могу бити ретке, тако да програм може даље да се користи (нелегално), али могу бити и толико честе да обесмишљавају даље коришћење програма.

Пробна и „обогалјена/осакаћена“ верзија софтвера (2)

- ▶ програм делимично или сасвим престане да функционише и захтева некакву лозинку (кључ) коју корисник може да добије од власника софтвера тек када га купи. У случају да програм само делимично престане да функционише, он се заправо претвара у тзв. „обогалјену/осакаћену“ верзију софтвера (енгл. crippleware).
- ▶ Други сценарио подразумева да се, у циљу популаризације софтвера, понекад одвојено деле „осакаћена“ верзија софтвера као бесплатно издање софтвера са ограниченим могућностима и тзв. „потпуна“ или „професионална“ верзија програма чије се преузимање плаћа.

Пиратерија и пиратски софтвер

- ▶ Пиратерија је термин који је постојао још пре рачунарства и у првобитном значењу је представљао разбојништво (пљачку) бродова. Гусарење је сличан термин, али означава легализовану пљачку бродова туђих држава.
- ▶ У савремено доба, неовлашћено коришћење, копирање, модификовање и дистрибуција (продаја или бесплатно дељење) туђих производа се такође назива пиратерија. Са овим проблемом се посебно сусрећу музичка, филмска и софтверска индустрија, па чак и издавачи књига.
- ▶ Под изразом **пиратски софтвер** се подразумева софтвер који се неовлашћено користи, копира, модификује и дистрибуира (продаје или дели).
- ▶ Интернет је страховито допринео развоју пиратерије, али су и закони о заштити ауторских права (посебно оних у САД) допринели гашењу појединих сајтова који су подржавали размену пиратског материјала (на пример, Megaupload).

Слободни софтвер као алтернатива

- ▶ У доба када је све више слободног софтвера као алтернативе за познате, али и скупе власничке програме, неразумно је и ризично користити пиратске верзије софтвера. Ево неколико примера слободног софтвера и бесплатног власничког софтвера као алтернативе популарних власничких програма:
- ▶ Notepad++ (обрада текста без форматирања);
- ▶ OpenOffice, LibreOffice (алтернатива програмима из пакета Microsoft Office):
- ▶ 7zip, IZArc (алтернатива програмима за компресију и декомпресију података WinZip, WinRAR и сл.);
- ▶ mplayer, MPC-HC, VLC за репродукцију аудио и видео материјала;
- ▶ IrfanView (преглед и једноставна обрада слика);
- ▶ GIMP (обрада растерске графике, права замена за Adobe Photoshop);
- ▶ Inkscape (обрада векторске графике, права замена за CorelDraw);
- ▶ Многи други корисни апликативни програми: yed Graph Editor, JabRef,¹⁰⁶ MikTeX, TeXStudio, calibre, SharpDevelop, Unitex, MySQL, R, XAMPP, ...