



Универзитет “Св. Кирил и Методиј” во Скопје  
Факултет за електротехника и информациски технологии



## ПРОГРАМИРАЊЕ И АЛГОРИТМИ

# ВОВЕД ВО ПРОГРАМИРАЊЕ И АЛГОРИТМИ

- Програмски јазик C -



Програмирање и алгоритми

# ПОИМ ЗА КОМПЈУТЕР



# ШТО Е КОМПЈУТЕР

- Компјутер е електронски уред кој се управува со помош на програмирани софтверски инструкции.
- Компјутерот е способен да прима влезни податоци, да врши нивна обработка и да превзема логички акции.
- Генерално, терминот “компјутер” опишува збир на уреди кои функционираат во една заедничка целина со цел обавување на дадена функција.
- Историски терминот компјутер доаѓа од латинскиот збор **Computare** што во превод значи **пребројување**.
- Какви видови компјутери познаваме?

# ВИДОВИ КОМПЈУТЕРИ (Општа претстава)

## Персонален компјутер



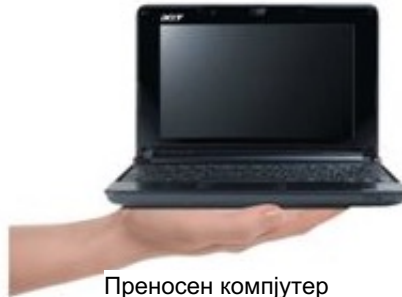
Статичен компјутер



Преносен компјутер



Таблет компјутер



Преносен компјутер



Таблет компјутер

## Паметен телефон



## Супер компјутер



# ВИДОВИ КОМПЈУТЕРИ (Широка претстава)



Компјутери во автомобилска индустрија



Индустриски компјутери

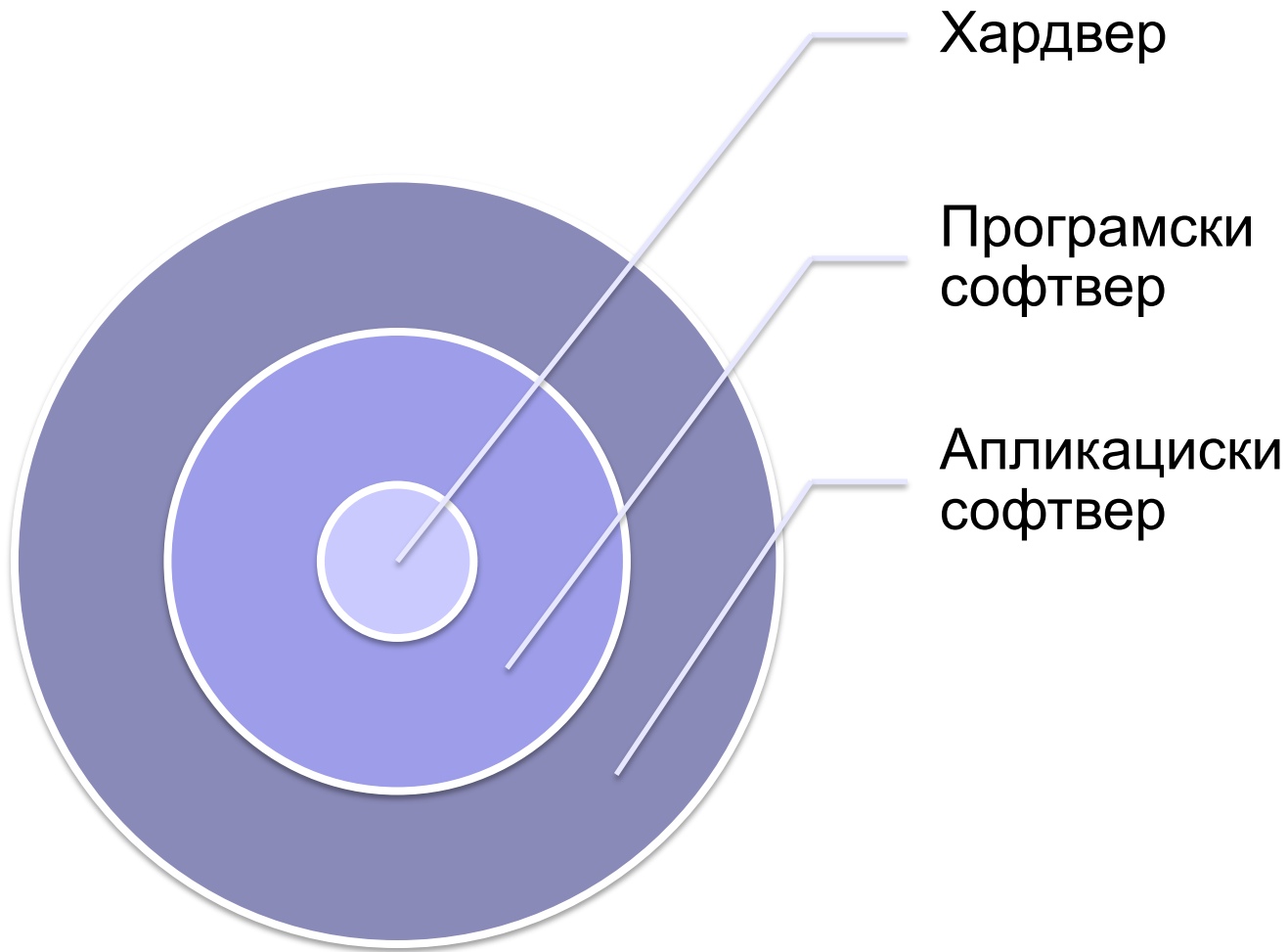


Вградливи микрокомпјутерски системи

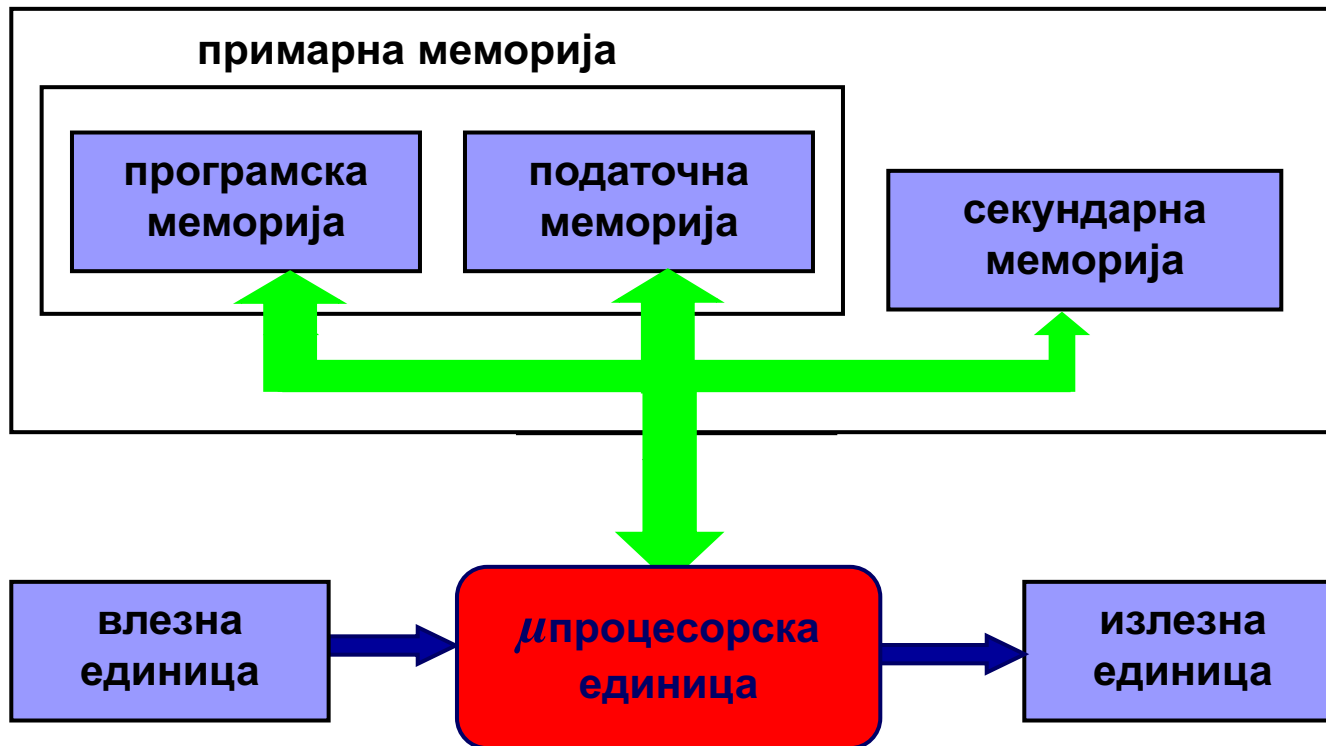


Компјутери во домаќинство

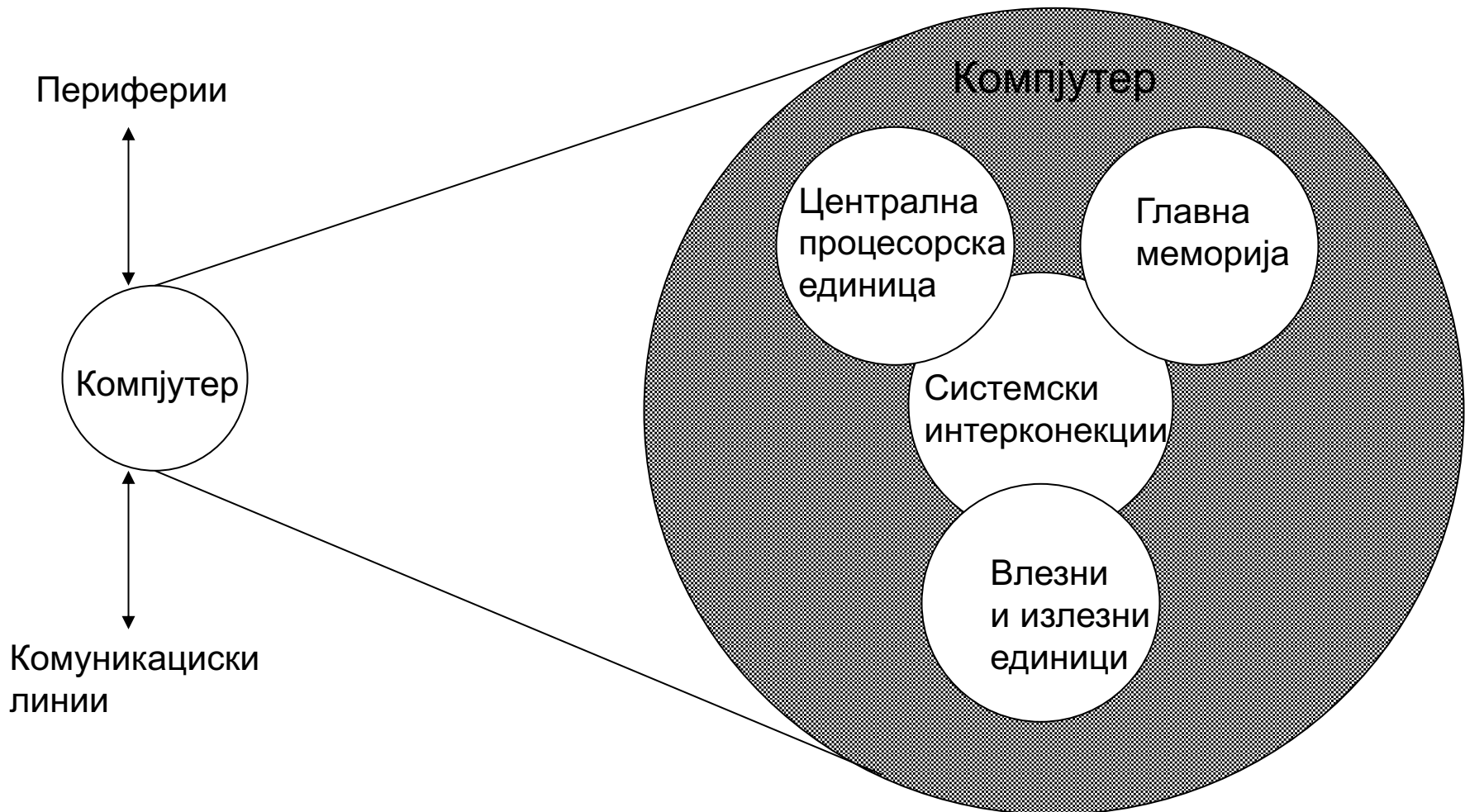
# ОРГАНИЗАЦИЈА НА КОМПЈУТЕР



# ОПШТА СТРУКТУРА

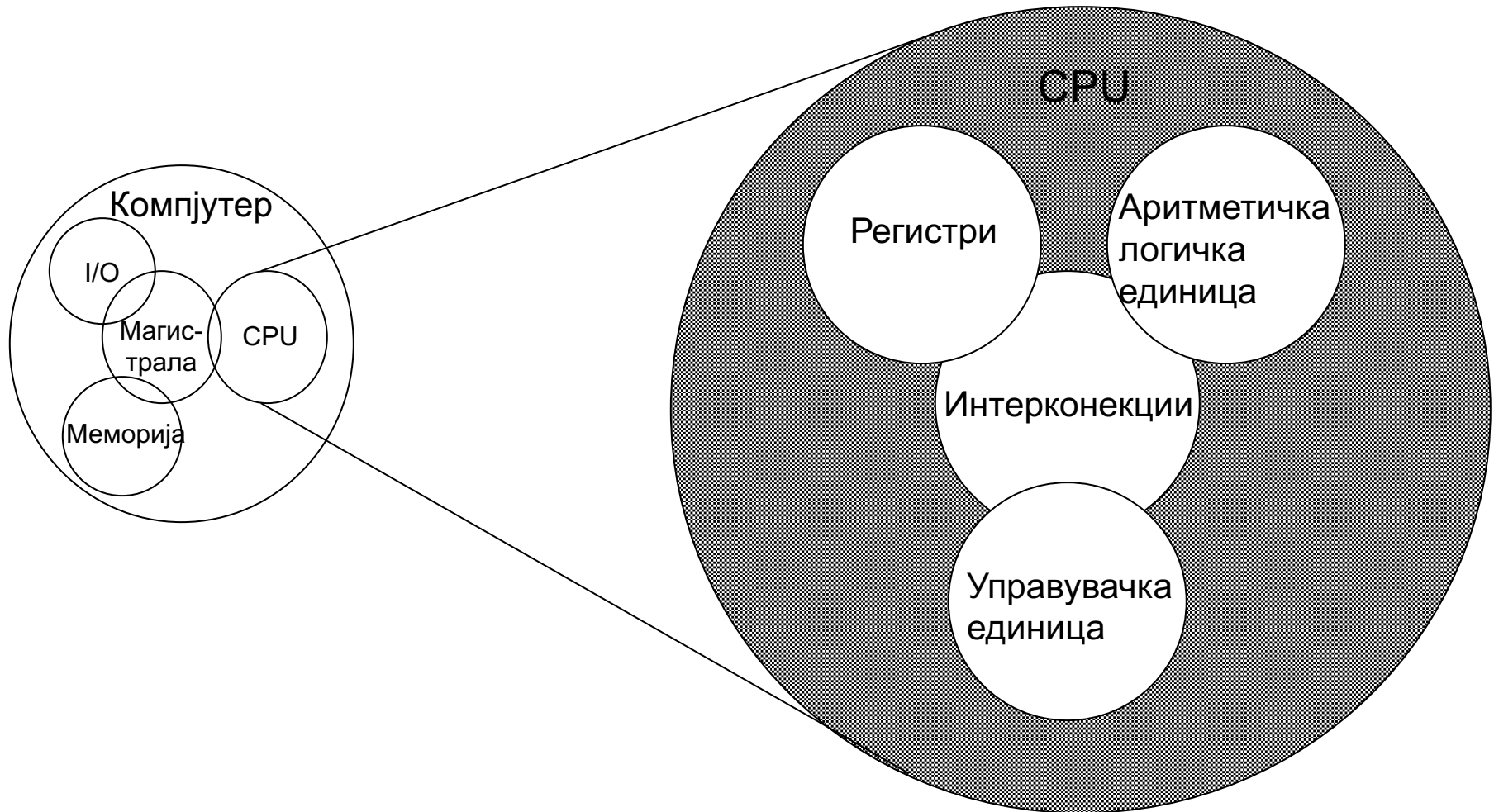


# СТРУКТУРА (високо ниво)





# Структура (CPU)

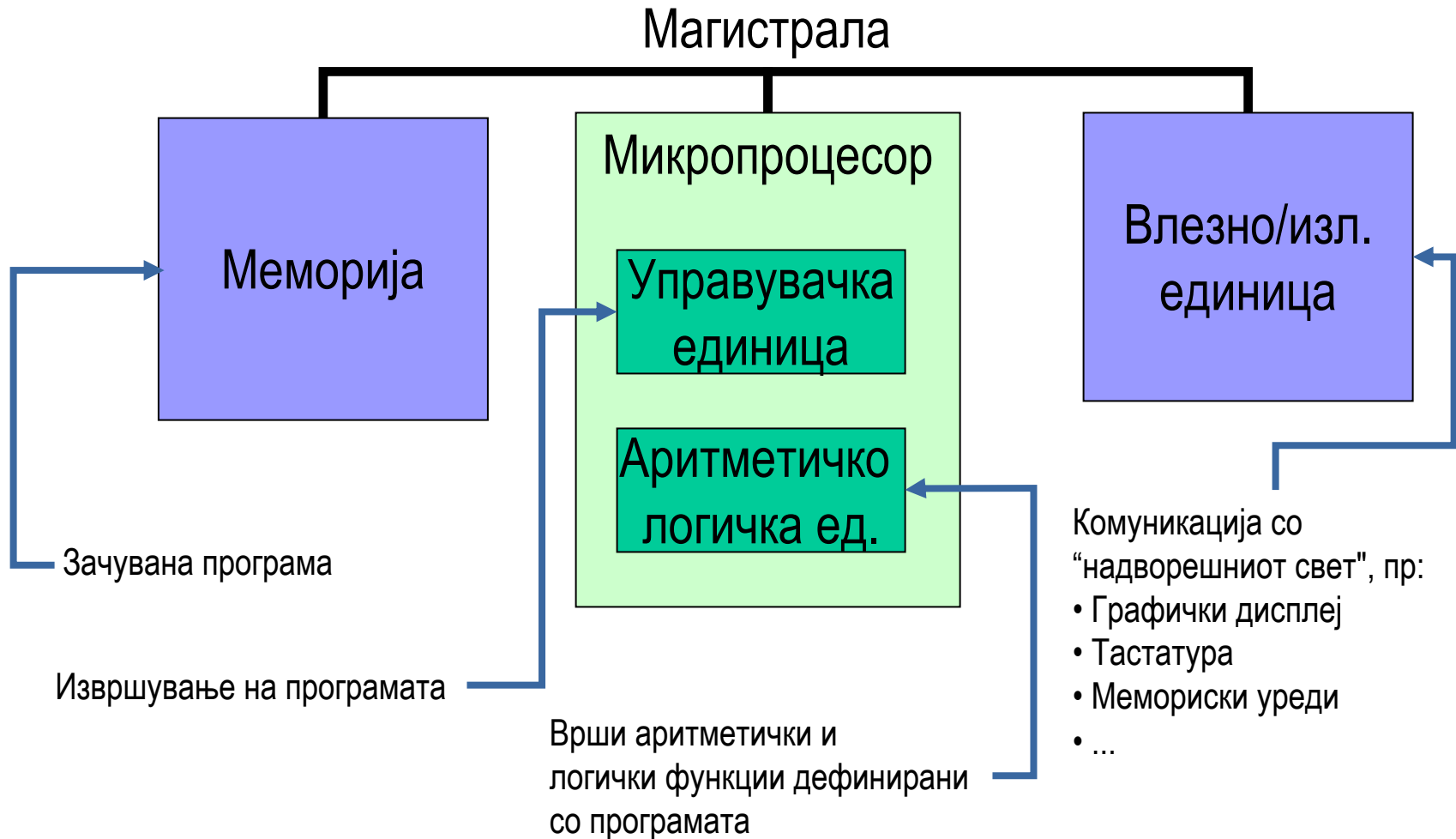




# VON NEUMANN АРХИТЕКТУРА

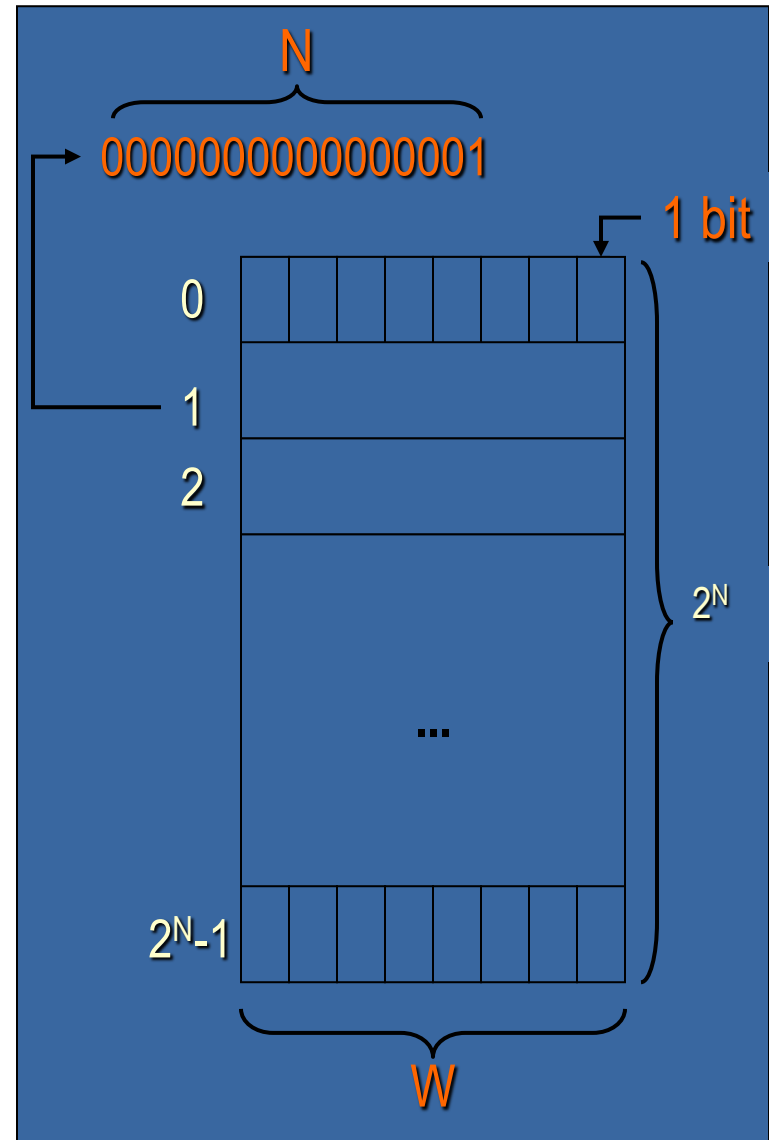
- Најголемиот број компјутерски системи се базираат на таканаречената **Von Neumann** архитектура
- Оваа архитектура се базира на следните три к-ки:
  - 1) Компјутерот содржи четири основни под-системи
    - Меморија
    - Аритметичко-логичка единица
    - Управувачка единица
    - Влезно/излезна единица
  - 2) Програмата е зачувана во меморијата на компјутерот
  - 3) Програмските инструкции се извршуваат редоследно (секвенцијално)

# VON NEUMANN ARХИТЕКТУРА



# МЕМОРИЈА

- Мора да се направи јасна разлика помеѓу:
  - Адреса на мемориска клетка
  - Содржина на мемориска клетка
- Должина на меморија  $W$ 
  - Колку бити содржи една мемориска клетка, вообичаено еден бајт (1 byte = 8 bits)
- Должина на адреса  $N$ 
  - Колку бити се користат за претставување на една адреса, го дефинира адресниот простор (максималната димензија на меморијата)
  - Доколку должината на адресата е  $N$  бити, тогаш адресниот простор е  $2^N$  (0, 1, 2, .. ,N-1)



# МЕМОРИЈА

## ■ Подредена низа од бајти

- Редниот број = мемориска адреса

## ■ Секој бајт = податок

## ■ Секој податок

- чист податок

ИЛИ

- програма (наредба)

Адреса  
(децимален облик)

Адреса  
(хекса-децимален облик)

$2^{32}-1$

FFFFFFFF

FFFFFFFE

FFFFFFFD

•

•

•

2

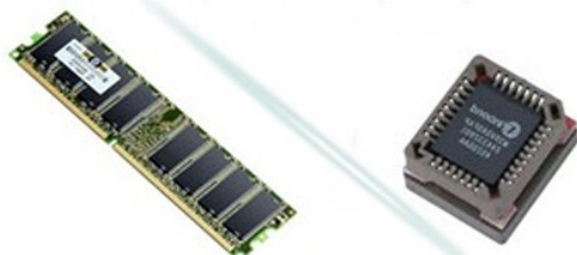
00000002

1

00000001

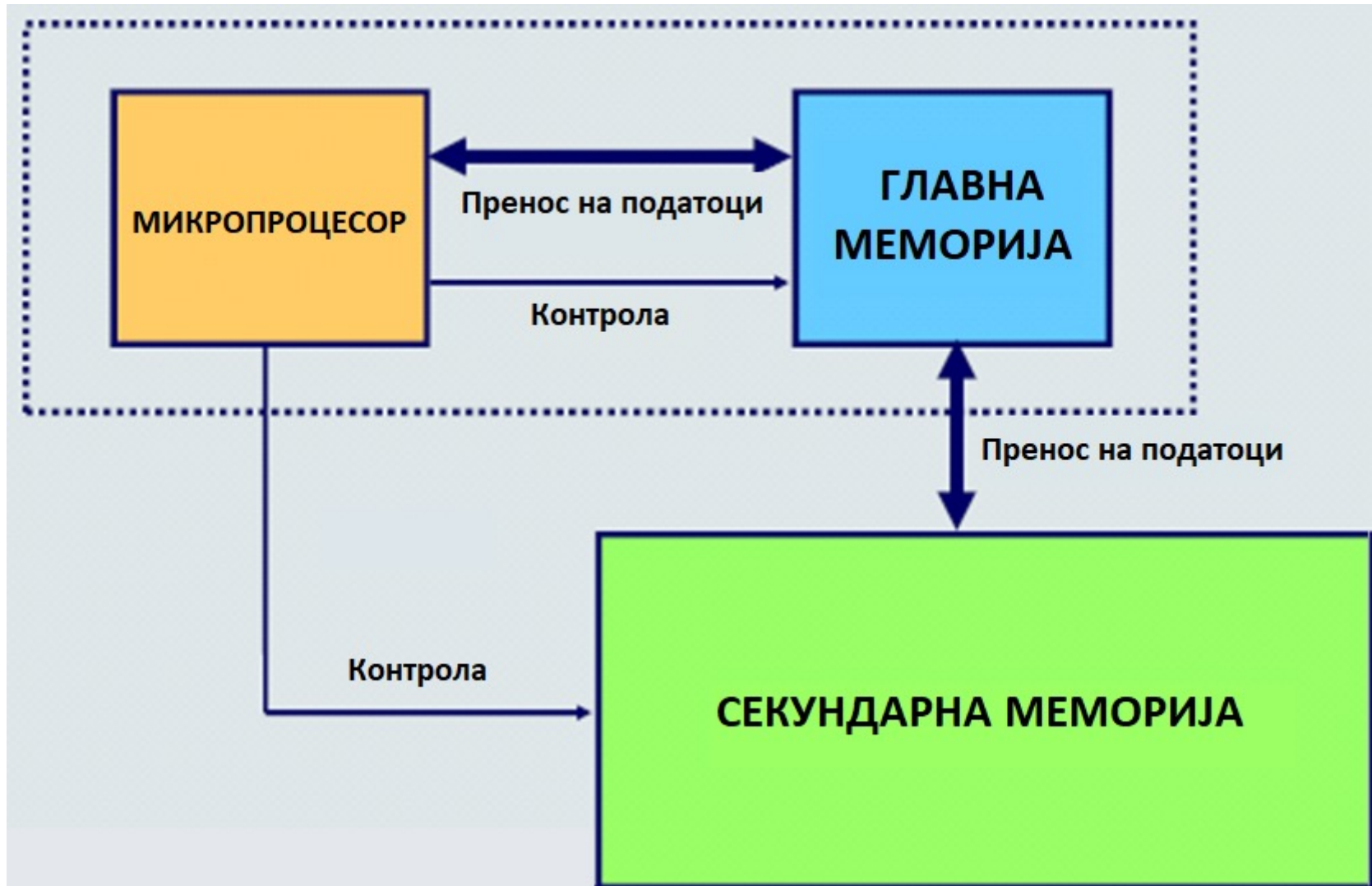
0

00000000



**RAM - ROM**

# ЛОКАЦИЈА НА МЕМОРИЈАТА





Програмирање и алгоритми

# АЛГОРИТМИ



## Што е алгоритам?

- Систематичен метод (приод) кон решавање на даден проблем
- Може да се разбере како секвенца на инструкции (чекори) кои е потребно да се извршат за да се реши одредена задача
- Чекорите на алгоритмот мора да бидат: **недвосмислени, јасни, едноставни, ефикасни и подредени во логичен редослед**
- Алгоритам е општ термин, на пример рецепт за правење храна е еден вид алгоритам
- **Компјутерски алгоритам** е алгоритам кој може да биде извршен од страна на компјутер

## Пример за алгоритам

- Пример – миење коса
  - Чекор 1: наводени ја косата
  - Чекор 2: стави шампон
  - Чекор 3: направи пена
  - Чекор 4: измиј
  - Чекор 5: исуши ја косата
- Што ако го промениме редоследот на чекорите 4 и 5?
  - Чекор 1: намокри ја косата
  - Чекор 2: стави шампон
  - Чекор 3: направи пена
  - Чекор 4: исуши ја косата
  - Чекор 5: измиј





# АЛГОРИТМИ



Програмирање и алгоритми

# ПОИМ ЗА ПРОГРАМИРАЊЕ

# ЗОШТО ПРОГРАМИРАЊЕ?

- Пишување софтвер (компјутерски програми) е опишување како нешто се прави. Чекори за извршување нешто.
- Пишувањето програма е предизвик исто како решавање сложувалка.
- Пишување програма може да биде слично на компонирање музика, градење кука, создавање ...
- Она што секој програмер бргу научува е дека „компјутерот е глупав“. Тој прави точно она што ќе му кажете, а што не мора да биде она што сте сакале.
- Програмирањето ќе Ве научи колку е значајна „јасноста во изразувањето“.

# ЗОШТО ПРОГРАМИРАЊЕ?

- Често се вели, дека некој всушност не го разбира проблемот, додека не успее проблемот да го објасни на некој друг. Осносно, некој навистина не разбрал нешто се додека не се обиде да го научи компјутерот, односно нештото да го изрази како алгоритам“ (Donald Knuth, во "*American Mathematical Monthly*," 81)
- It has often been said that a person does not really understand something until he teaches it to someone else. Actually a person does not really understand something until after teaching it to a computer, i.e., express it as an algorithm.“ (Donald Knuth, in "*American Mathematical Monthly*," 81)



# ПРОГРАМИРАЊЕ

- Едноставно кажување на компјутерот што треба да стори.
- **Програма**: детален и прецизен опис на некоја појава или процес кој е разбирлив и извршлив од страна на компјутер.



# ПРОГРАМИРАЊЕ

- Пишување програми...
- Што е тоа програма?
- **Програма** е низа на точно дефинирани инструкции напишани во некој **програмски јазик** со кои се извршуваат конкретни задачи.

# ПРОГРАМСКИ ЈАЗИЦИ

- Програмските јазици се вештачки јазици кои се употребуваат за да се контролираат машини, најчесто компјутери.

- Програмски јазици

Точно дефинирани системи од правила за опишување на задачите на компјутерот напишани во форма разбирлива и за компјутерот и за програмерот.



# ПОДЕЛБА НА ПРОГРАМСКИТЕ ЈАЗИЦИ

- Генерации на програмски јазици
- Јазици со општа и јазици со специјална намена
- Декларативни и императивни јазици
  - Како треба
    - операции - императивни јазици
  - Што треба
    - резултати - декларативни јазици





# ВИДОВИ ПРОГРАМСКИ ЈАЗИЦИ

- Машински јазици - множества инструкции напишани со 0 и 1
  - Природен јазик на компјутерите.
  - Макотрпно програмирање подложно на грешки.
  - Машински код тежок за разбирање и измени.
  - Програмирањето е долго и скапо.
  
- Јазици од првата генерација - асемблерски јазици
  - Едноставна модификација на машинските јазици.
  - Зависат од компјутерот на кој работат.



# ВИДОВИ ПРОГРАМСКИ ЈАЗИЦИ

- Јазици од втората и третата генерација - имплементациски јазици
  - Слични со говорните јазици.
  - Наредбите се послични на говорниот јазик,
  - Структурирано програмирање.
  - Јазици со општа намена.
  - Едноставност, униформност и преносливост на програмите.
- Јазици од четвртата и петтата генерација - апликативни јазици
- Јазици со специјална намена.

# ГЕНЕРАЦИИ НА ПРОГРАМСКИ ЈАЗИЦИ

- Компјутерска праисторија - програмирање со хардверски спојки.
- Нулта генерација (до 1954 година) - асемблер
- Прва генерација (1954-1958) - базирани на математички изрази
  - FORTRAN, ALGOL 58, Flowmatic
- Втора генерација (1959-1961)
  - FORTRAN II, ALGOL 60, COBOL, APL, LISP
- Трета генерација (1962-1970)
  - PL/I, ALGOL 68, PASCAL, SIMUAL, BASIC
- Четврта генерација (1970-1980)
  - Smalltalk, C, Prolog, Ada
- Петта генерација (1980- ) - објектно ориентирани јазици
  - Modula-3, C++, Object Pascal, Java



# Синтакса и семантика на програмските јазици

- Правила за пишување на програмите - **синтакса**.
- Пример:
  - Песот ја прескокна оградата.
  - Оградата прескокна песот ја.
- Синтаксата е студија за репрезентацијата, граматиката и структурата на еден јазик.
- Синтаксно точни јазици мора да бидат разбирливи за учесниците во разговорот.
- Значењето на синтаксички точна реченица - семантика.
- Пример:
  - Песот го гризна човекот.
  - Човекот го гризна песот.
- Семантика е студија за значењето, акциите, функциите и однесувањето на еден јазик.



# ПРОГРАМСКИ ПАРАДИГМИ

- **Компјутерите** се исклучително „глупави“ машини кои сè сфаќаат буквално и неспособни се да ги разберат двосмисленостите и недореченостите.
- Затоа **компјутерските програми** мора да бидат екстремно прецизни и недвосмислени. Токму заради ова, организацијата на овие детали е клучна за успешноста на една програма.
- Различните пристапи кон организацијата на овие детали ги претставува различните начини на програмирање (**програмски парадигми**).

---

парадигма – шаблон или шема која сугерира начин или постапка за решавање на проблеми



# СТРУКТУРИРАНО ПРОГРАМИРАЊЕ

- Структурирано програмирање е една од програмските парадигми
- Структурираното програмирање претставува техника за организација на програмата во логичка структура на хиерархиски модули.
- Големите рутини се разбиваат на помали модули.
- Употребата на наредбата за безусловен скок (GOTO) не се препорачува.
- Ова овозможува програмата да се разгледува по изолирани блокови меѓу кои интеракцијата е (релативно) контролирана.



# Основни елементи на структурираното програмирање

- Основни програмски структури:
- **Конкатенација** (секвенца од наредби кои се извршуваат по ред)
- **Селекција** (се избира и извршува една од повеќе понудени наредби, во зависност од состојбата на програмата)
- **Повторување** (се повторува група наредби се додека програмата не се најде во одредена состојба)



# Пишување, преведување, поврзување, извршување

- Изворна програма
- Преведувач (компајлер) е програма која програмите напишани во еден програмски јазик ги преведува во друг програмски јазик.
  - синтаксни грешки
- Објектна програма
- Извршна програма
- Интерпретер & Компајлер (преведувач)





# Пишување, преведување, поврзување, извршување

## ■ Фаза 1: Пишување програма

- ☐ Пишување во програма за уредување текст

## ■ Фази 2 и 3: Предпроцесирање и компајлирање

## ■ Фаза 4: Поврзување

- ☐ Посебен код-linker го поврзува објектниот код со кодот на потребните функции за да се добие извршна програма

## ■ Фаза 5: Вчитување

- ☐ Пред извршување програмата треба да се смести во главната меморија. Код-loader.

## ■ Фаза 6: Извршување

- ☐ Компјутерот под контрола на процесорот (CPU) ја извршува програмата, една по една инструкција во даден момент.



# КРАЈ