
Informatika

Građa računala, operacijski sustavi, uvod u programiranje

Izv. prof. dr. sc. Jurica Babić

Prof. dr. sc. Marija Seder

Doc. dr. sc. Đula Nađ

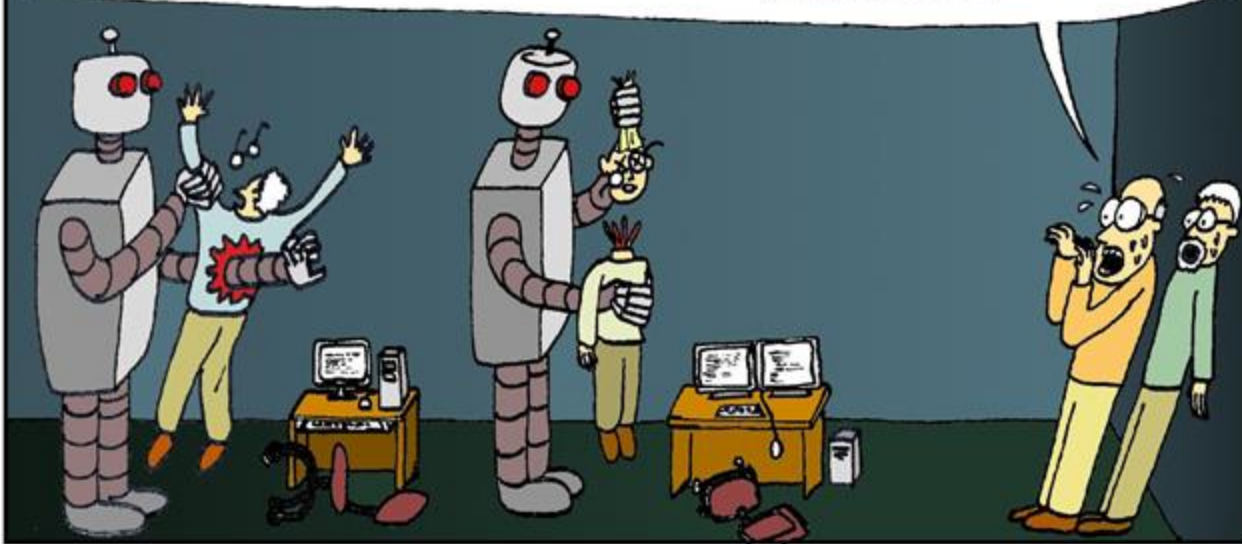
Prof. dr. sc. Vedran Podobnik

Ana Radović, univ. mag. ing. comp.

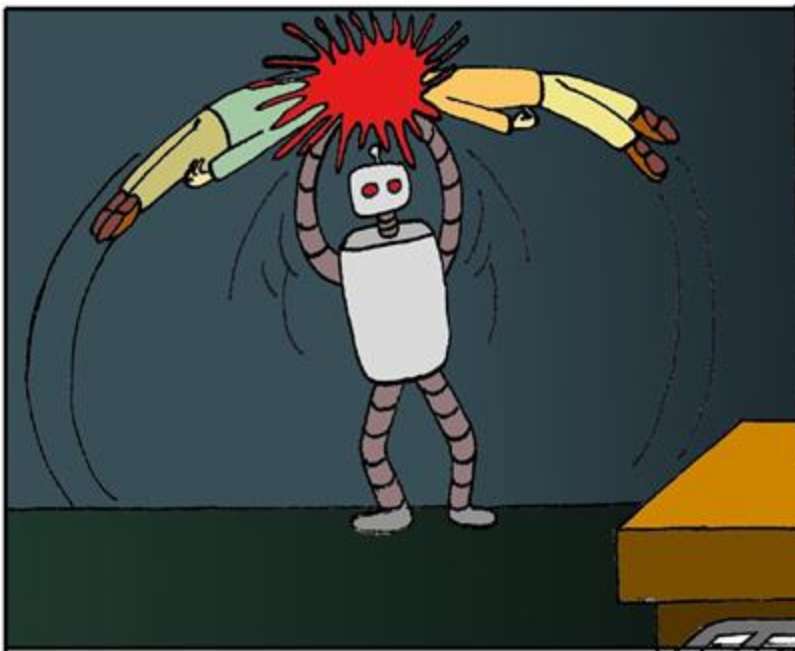
Sadržaj

- Građa računala
- Operacijski sustavi
- Uvod u programiranje

OH NO! THE ROBOTS ARE KILLING US!!!



BUT WHY?!!? WE NEVER PROGRAMMED THEM TO DO THIS!!!



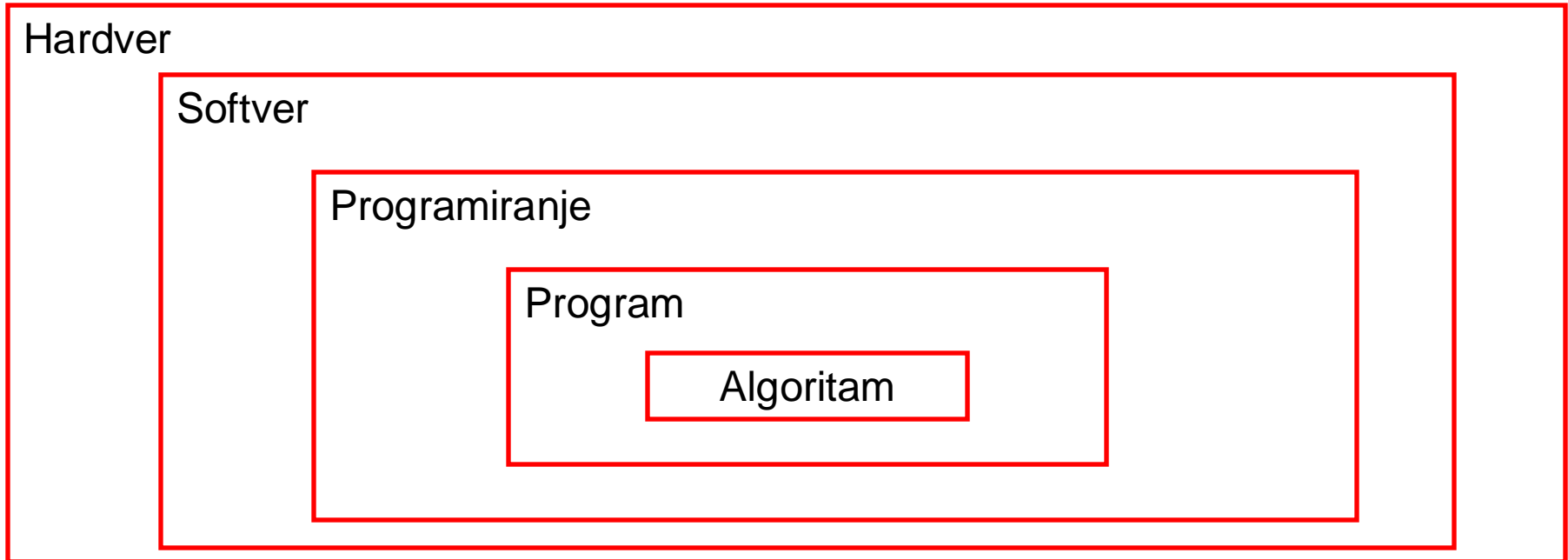
```
static bool isCrazyMurderingRobot = false;
```

```
void interact_with_humans (void){  
    if(isCrazyMurderingRobot = true)  
        kill(humans);  
    else  
        be_nice_to(humans);  
}
```

Građa računala

- **Računalo:** složeni programirajući stroj koji automatski obavlja aritmetičke i logičke operacije određene programom – može mijenjati, programirati
- **Građa računala:** skup pravila i metoda koja opisuju funkcionalnost, organizaciju i implementaciju računala
- **Cilj:** dizajnirati računalo koje maksimizira performanse uz što je moguće nižu cijenu

Građa računala



- **Algoritam:** skup koraka koji definiraju kako se izvodi zadatak
- **Program:** prezentacija algoritma
- **Programiranje:** proces razvoja programa
- **Softver:** skup programa i podataka koji omogućuju izvršavanje određenih zadataka
- **Hardver:** oprema korištena za unos informacija (podataka) i ispis
▶ 5 rezultata programa, algoritama

Povijest računala

- **Nulta generacija računala** – mehanički strojevi za računanje (1644.-1945.)
- **Prva generacija računala** – elektronička računala s elektronskim cijevima (1945.-1955.)
- **Druga generacija računala** – tranzistor kao građevna komponenta (1955.-1965.)
- **Treća generacija računala** – integrirani sklopovi (1965.-1980.)
- **Četvrta generacija računala** – sklopovi vrlo visokog stupnja integracije (1980.-?)
- **Peta generacija računala** – koncept umjetne inteligencije

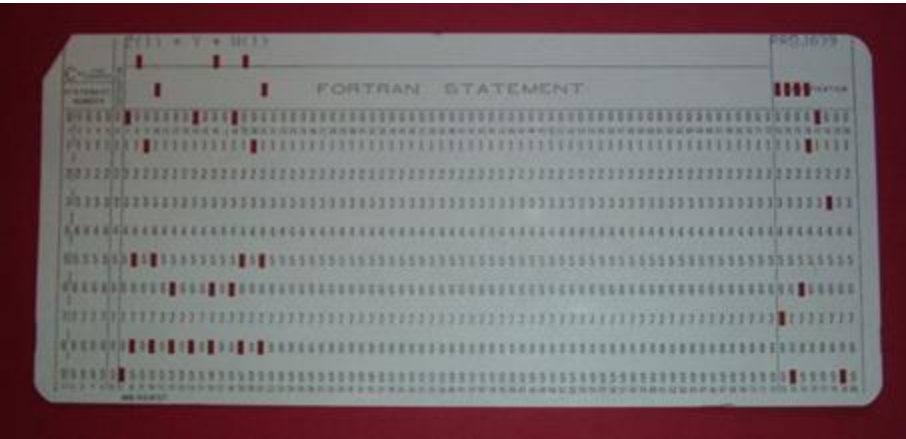
Povijest računala – nulta generacija

- Nulta generacija računala – mehanički strojevi za računanje (1644.-1945.)
 - Pozicija kotača predstavlja brojeve
 - Izumitelji: Blaise Pascal, Wilhelm Leibniz, Charles Babbage

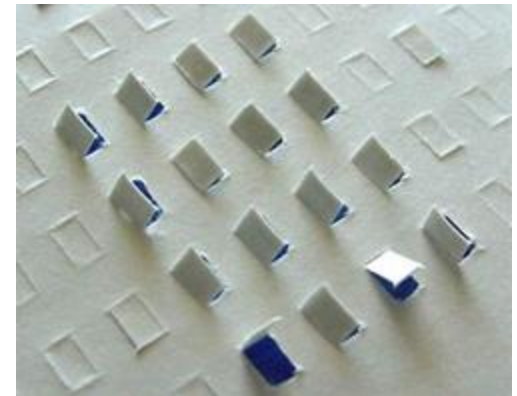


Pascalov mehanički kalkulator – 1645.

Povijest računala – nulta generacija



- ❑ Povijest sremanja podataka: bušene kartice
- ❑ Izumitelj Joseph-Marie Jacquard (1801) patentirao je bušene kartice za pohranu uzoraka i dezena koji su se izrađivali u tkanini
- ❑ Koncept spremanja informacija je iskoristio Charles Babbage u svom analitičkom stroju
- ❑ Popularne kroz 1970.-te



Povijest računala – nulta generacija



Dio diferencijskog stroja

- Charles Babbage – “otac računala”, 1822. izgradio je **prvo mehaničko računalo opće namjene** – diferencijski stroj
- Prvi diferencijski stroj sastojao se od oko 25,000 dijelova, težio 15 tona i bio 2.4 m visok. Imao je memoriju, jedinicu za računanje, ulaznu jedinicu (čitač bušenih kartica) i izlazne jedinice (pisač i bušač kartica). Nije nikad završen.

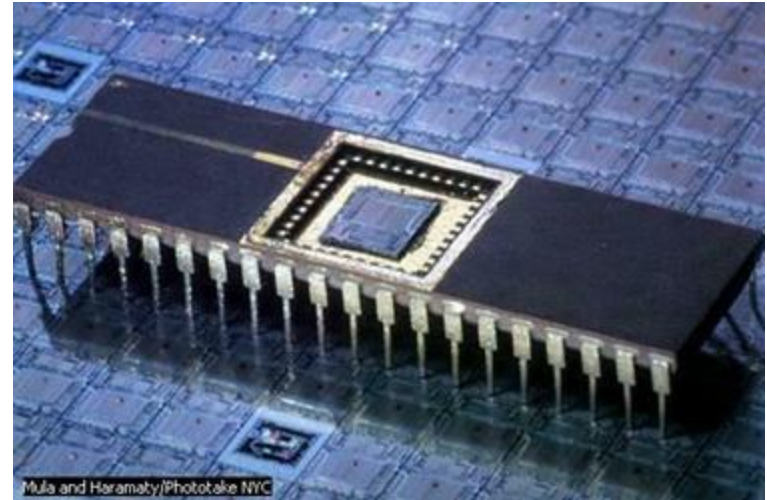
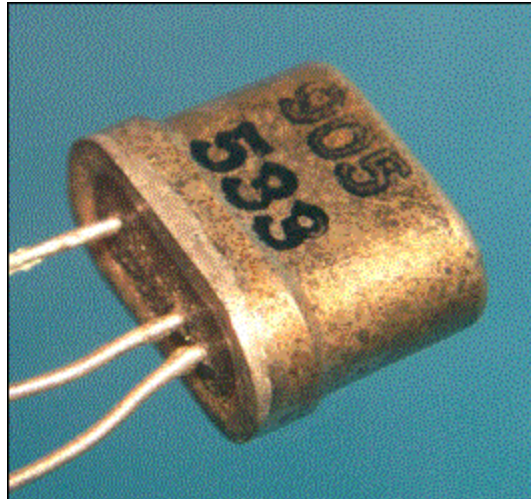
Povijest računala – nulta generacija



Enigma s tri rotora i priključnom pločom

- Stroj „Enigma“ – elektromehanički stroj za šifriranje poruka koji su Nijemci koristili u 2. svjetskom ratu
- Stroj „Bomba“ – Britanci 1939. godine razvili elektromehaničko računalo za razbijanje šifre Enigme
 - - prvo krenuli Poljaci, pa Alan Turing
 - - mnogo faktora za uspješno probijanje Engime (često korištenje istih fraza u porukama, zaplijena ključnih materijala o Enigme iz podmornica, tajnost)

Povijest računala – prva generacija



Elektronska (vakuumska) cijev

Stare televizije,
Radio prijemnici

Pregrijavanje, puno
energije

Tranzistor

Pojačala za
zvuk,
preklopnici,
napajanja...

Brži i
učinkovitiji od
el. cijevi

Mikročip

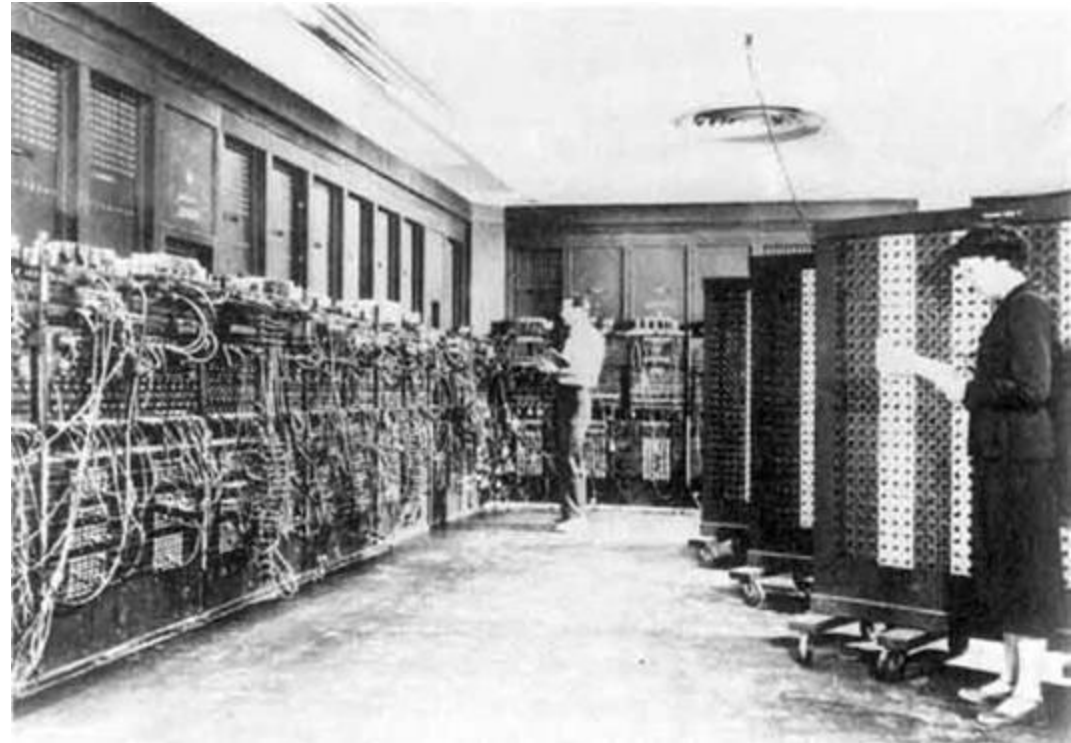
Npr. procesori u
računalu/mobitelu

U sebi ima milijune ili
milijarde tranzistora i
ostalih komponenti!



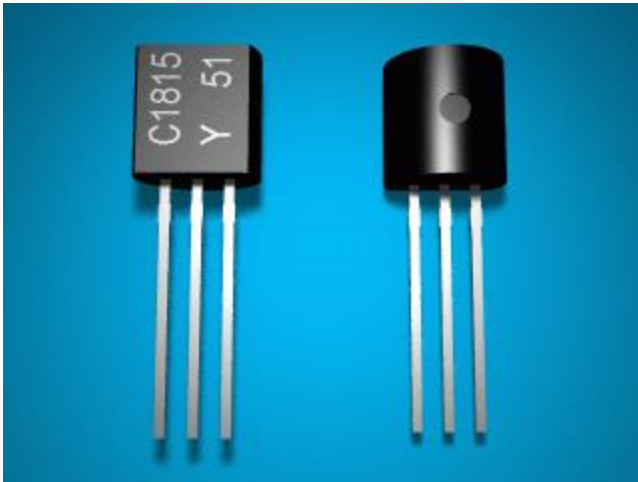
Povijest računala – prva generacija

- ❑ Prva generacija računala – elektronička računala s elektronskim cijevima (1945.-1955.)
 - ❑ 1946. prvo elektronsko digitalno računalo ENIAC (Electronic Numerical Integrator and Calculator)
 - ❑ 18000 elektronskih cijevi, 1500 releja, 30 tona
 - ❑ **3 operacije korijenovanja po sekundi** (neopisivo sporo po današnjim standardima)
 - ❑ Alan Turing 1945 – uvodi pojam algoritma za matematičke izračune - univerzalni Turingov stroj
 - ❑ John von Neumann 1945 – većina današnjih računala zasnovano na von Neumannovoj arhitekturi



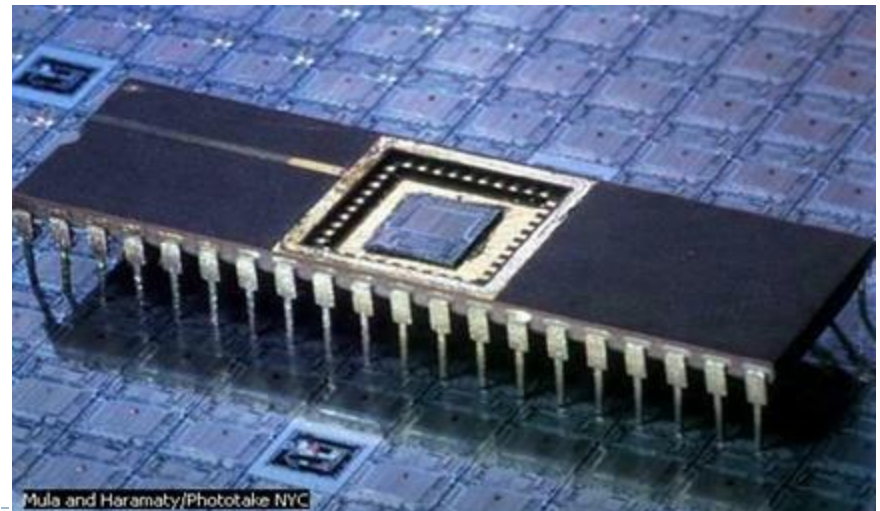
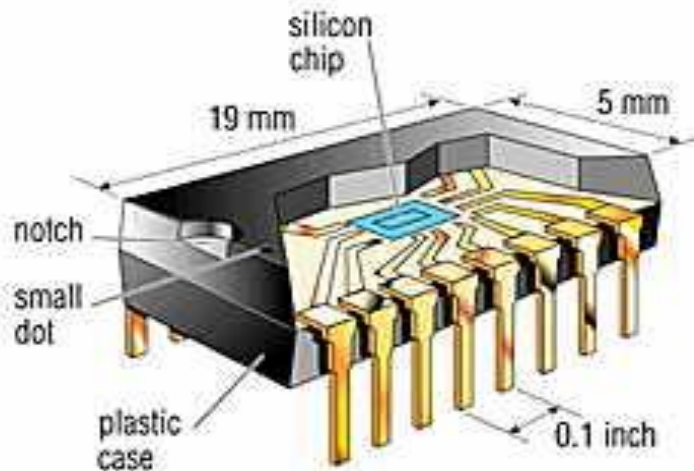
Povijest računala – druga generacija

- Druga generacija računala – tranzistor kao građevna komponenta (1955.-1965.)
 - 1947 - Walter Brittain i Willaim Shockley izumili tranzistor – elektronička sklopka
 - 1955. - Harwell CADET – prvo tranzistorsko računalo



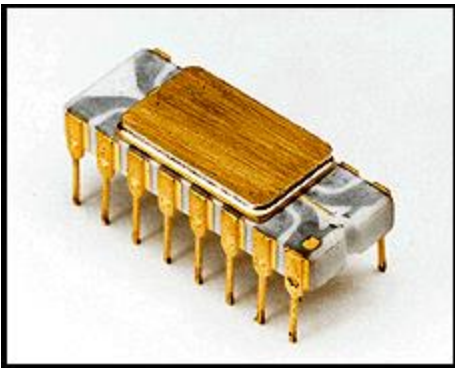
Povijest računala – treća generacija

- ❑ Treća generacija računala – integrirani sklopovi (1965.-1980.)
 - ❑ 1959, Jack Kirby i Robert Noyce (postaje suosnivač Intel Corp.) razvili su prvi integrirani sklop (silikonski čip ili mikročip)
 - ❑ Integrirani sklop (**integrated circuit (IC)**) je skup velikog broja tranzistora na komadiću silicija
 - ❑ 1971. prvi 4-bitni mikroprocesor Intel 4004 – CPU
 - ❑ 1972. 8-bitni mikroprocesor Intel 8008
 - ❑ Tipkovnice, monitori



Povijest računala – četvrta generacija

- ❑ Četvrta generacija računala – sklopovi vrlo visokog stupnja integracije (1980.-?)
 - ❑ Mooreov zakon – broj tranzistora na čipu se udvostručuje svakih 18-24 mjeseca
 - ❑ 1981. IBM osobno računalo (Intel 8088 mikroprocesor)



Povijest računala – četvrta generacija

- ❑ Četvrta generacija računala – sklopovi vrlo visokog stupnja integracije (1980.-?)
 - ❑ 1975 – Steve Jobs i Steve Wozniak pokreću Apple Computer
 - ❑ 1984 – Apple Macintosh s grafičkim korisničkim sučeljem (GUI)



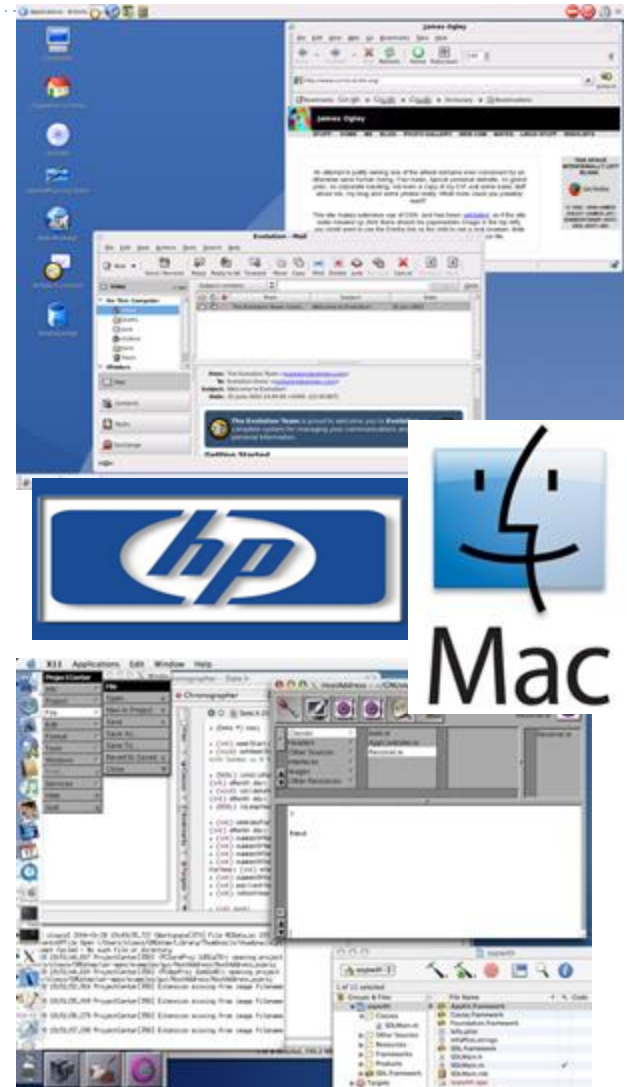
Povijest računala – četvrta generacija

- Četvrta generacija računala – sklopovi vrlo visokog stupnja integracije (1980.-?)
 - prijenosna računala (laptopi):
 - 1981 - Adam Osborne, proizveo Osborne 1
 - 1988 - Compaq laptop s ekranom u boji
 - 2008 – Macbook Air – najtanji laptop
 - 2011 – Dell XPS 15Z – najtanji Windows laptop



Današnji trendovi

- Microsoft održava Windowse, zadnja verzija “Windows 11”
- LINUX operacijski sustav postaje sve popularniji
- IBM prestaje raditi mala računala – Lenovo je kupio njihov posao 2005.
- HP je najveći proizvođač PC-a (drugi je Lenovo)
- Apple je najveći igrač za osobna računala, naročito za “kreativno tržište”, iPad, iPhone, Mac Mini, iMac itd.
- Google je najveći igrač za internetska rješenja



Povijest računala – peta generacija

- Peta generacija računala – koncept umjetne inteligencije
 - Pojava paralelne obrade više instrukcija pokazuje mogućnost umnažanja snaga više CPU-a
 - Cilj pete generacije je razviti uređaje koji imaju sposobnost učenja, samoorganizacije
 - Robotika
 - Nanotehnologija
- Tvrtka OpenAI kao lider u rješenjima zasnovanima na umjetnoj inteligenciji

Građa računala

- **Osnovni dijelovi računala**
 - matična ploča (grafička kartica, zvučna kartica, mrežna kartica, CPU)
 - ROM/RAM memorija
 - tvrdi disk
 - CD/DVD/Blue-ray
 - sabirnice
 - sučelja
 - napajanje

Hardver vs Softver

- ❓ **Hardver** = fizičke
komponente koje
čine računalo

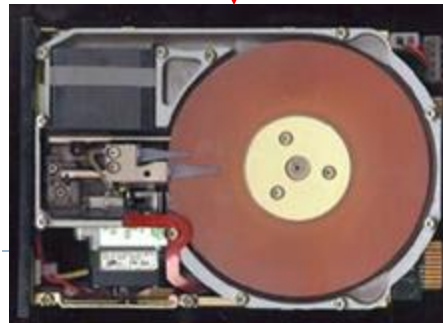
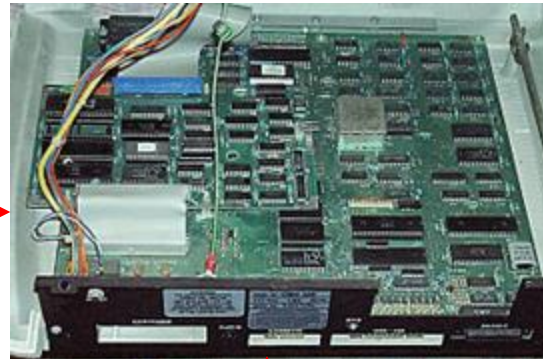
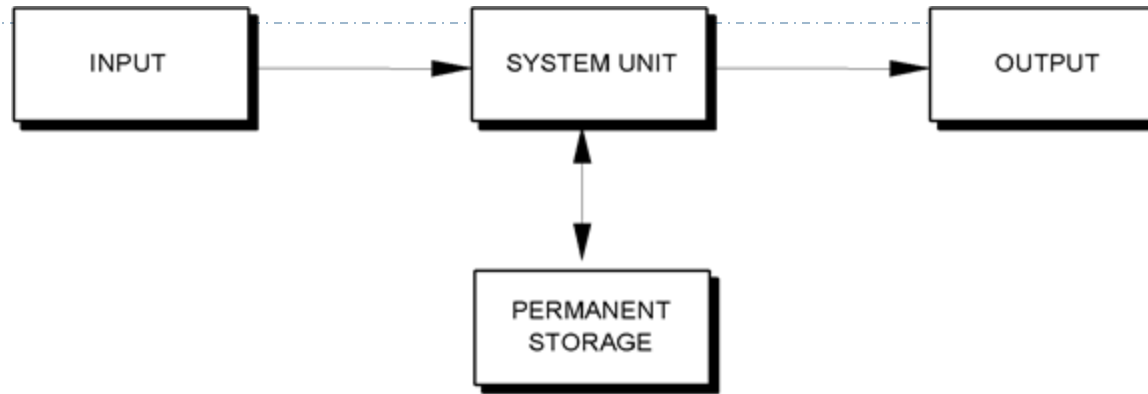


Hardver vs Softver

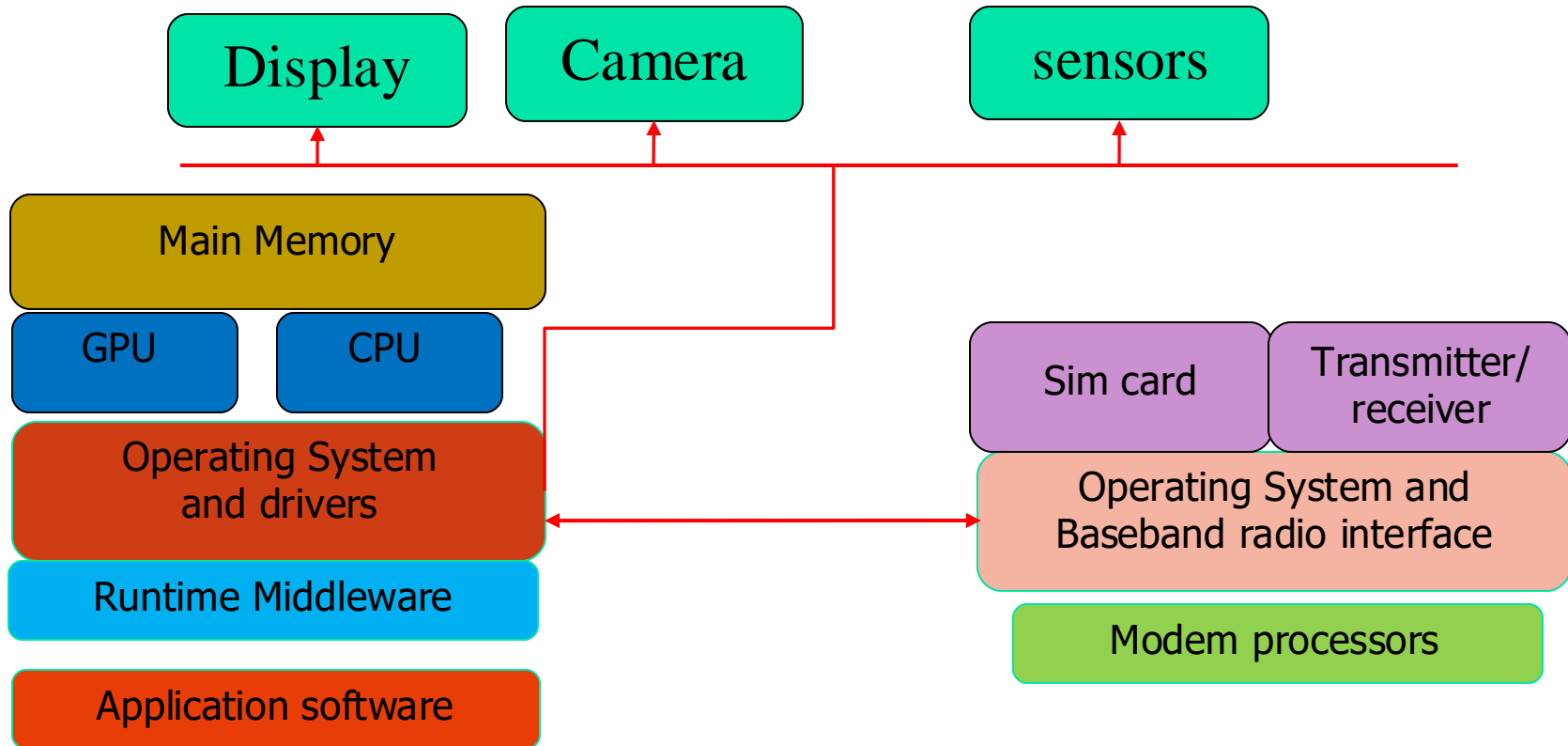
- ❑ **Softver** = programi (instrukcije) koje govore računalu što da radi
 - ❑ Sistemski softver
 - ❑ Aplikacijski softver
 - ❑ Pohranjen na hard disk, blue-ray, itd.



Komponente hardvera



Komponente hardvera - smartphone

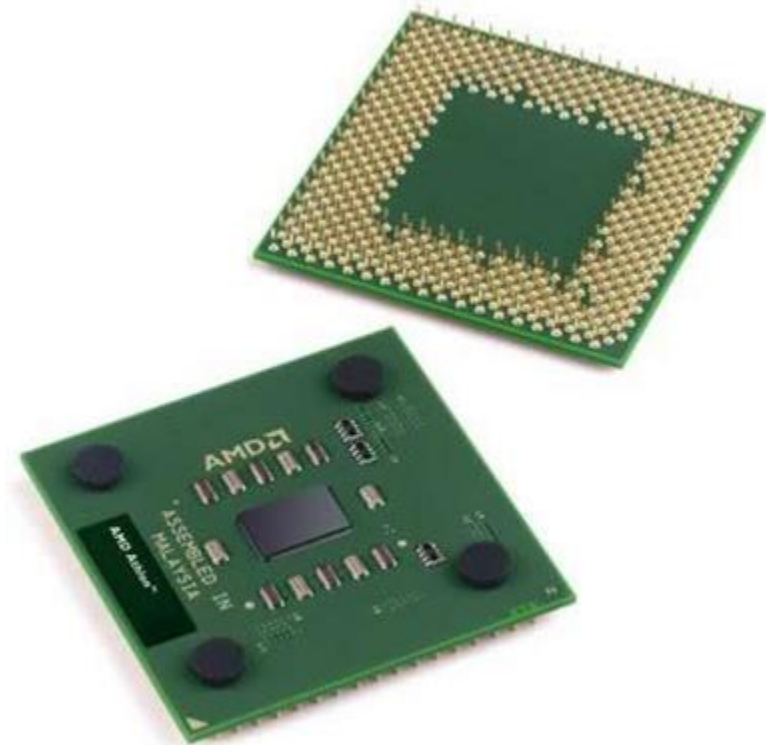


Procesor: CPU

CPU (Central Processing Unit) = složeni skup elektroničkih sklopova na jednom ili više integriranih sklopova (čipova) koji:

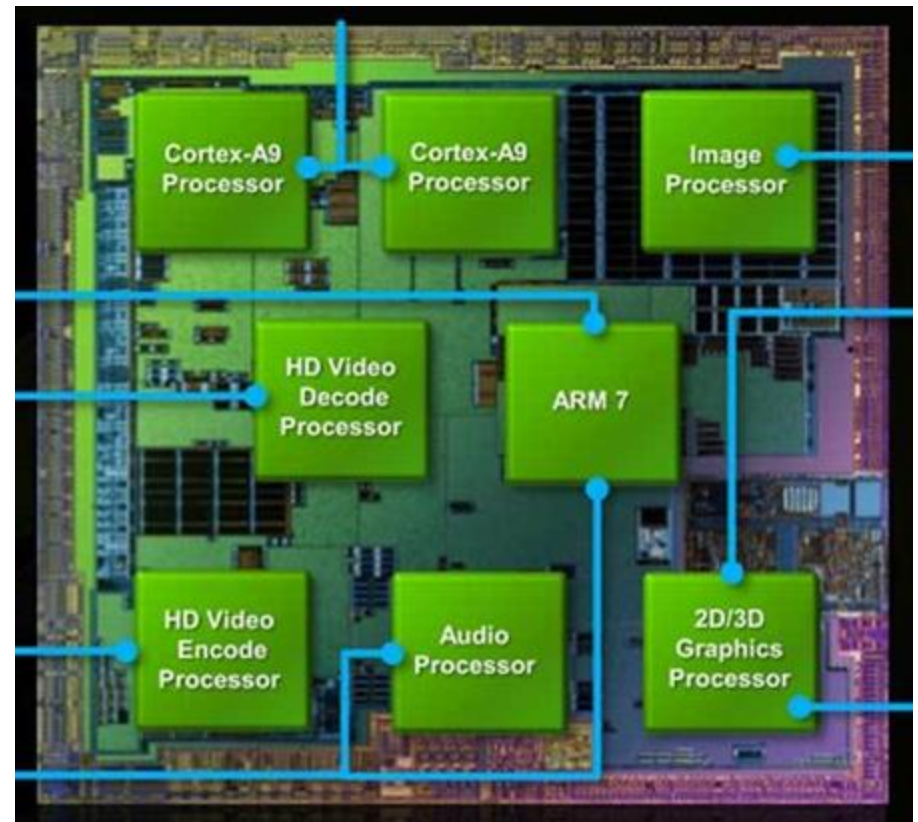
1. izvršavaju instrukcije softverskog programa
2. komuniciraju s ostalim dijelovima računala, posebno RAM i ulaznim uređajima

CPU je „mozak“ računala!



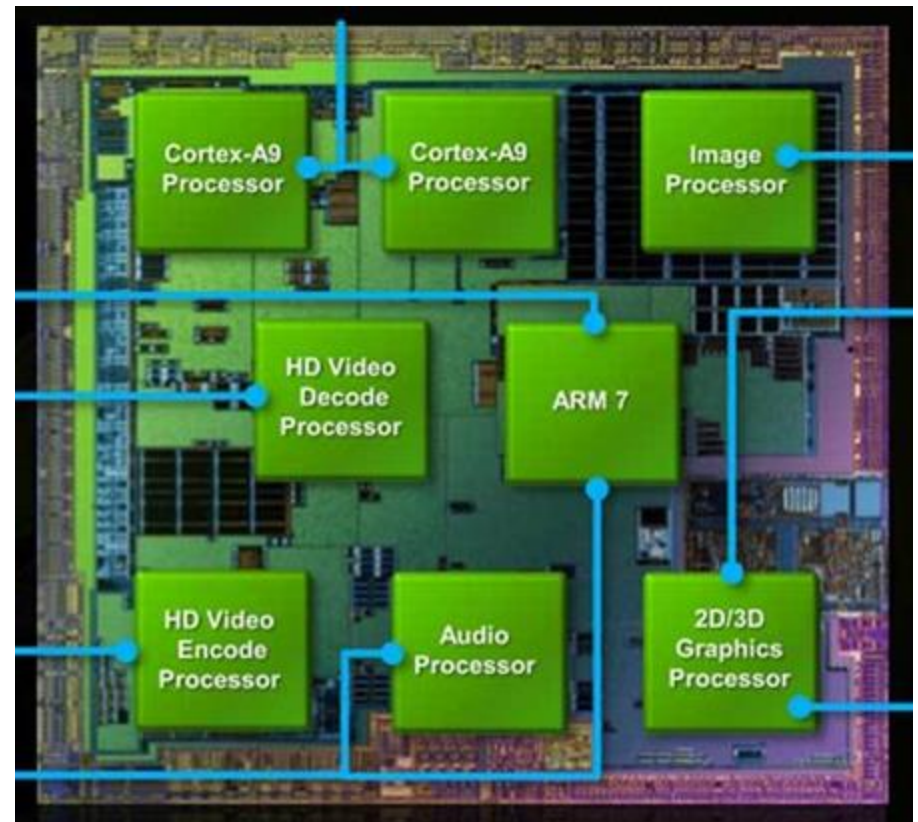
Procesor: CPU - smartphone

- ARM arhitektura (većinom)
- Sustav na čipu – „sve je na čipu, procesor, grafički procesor, modul za povezivanje, senzori itd...“
- 32-bit ili 64-bit.
- Niska potrošnja energije
- Više-jezgreni



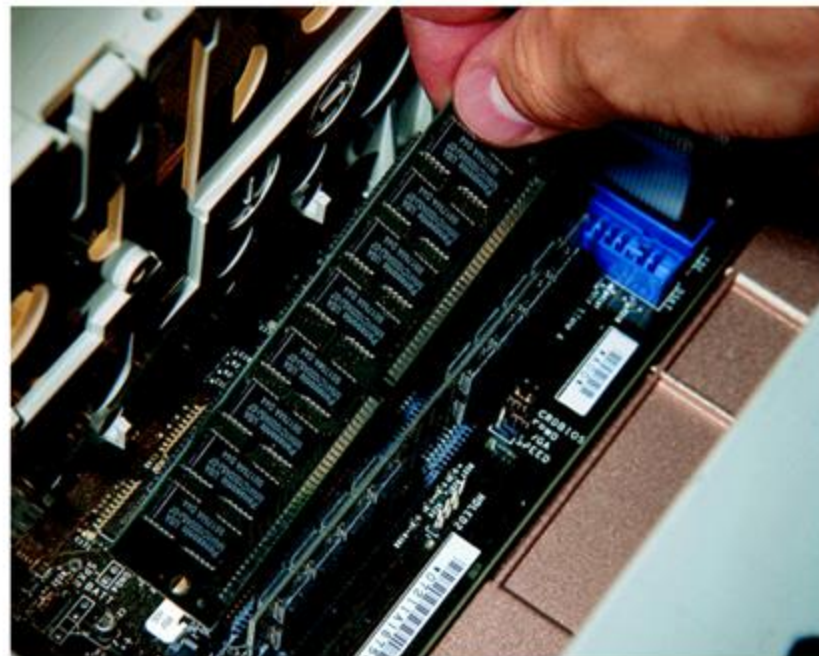
Procesor: GPU - smartphone

- (Graphics processing unit) – kao CPU, ali za stvaranje slika, obradu slika, računalnu grafiku
- Niska potrošnja energije
- Akceleracija 3D iscrtavanja
- Dekodiranje medijalnih zapisa
- Paralelna obrada velike količine podataka



Memorija i pohrana

- RAM (Random Access Memory)
 - Vrlo brza, sadržaj nestaje bez napajanja
- Unutarnja pohrana - ROM
 - Pohranjuje OS i kritične datoteke koje se trajno pamte, korisničke podatke, BIOS
- Vanjska pohrana
 - Pohranjuje korisničke podatke i podatke aplikacija
 - HDD, SSD, microSD



Softver

- Sustavski softver
- Aplikacijski softver



Aplikacijski softver

- ❑ Izvršava određeni zadatak:
 - ❑ Obrada teksta (MS Word)
 - ❑ Web preglednik
 - ❑ Media player
 - ❑ Photo editor
 - ❑ Igre
- ❑ Ne može raditi bez operacijskog sustava - OS (Operating System)

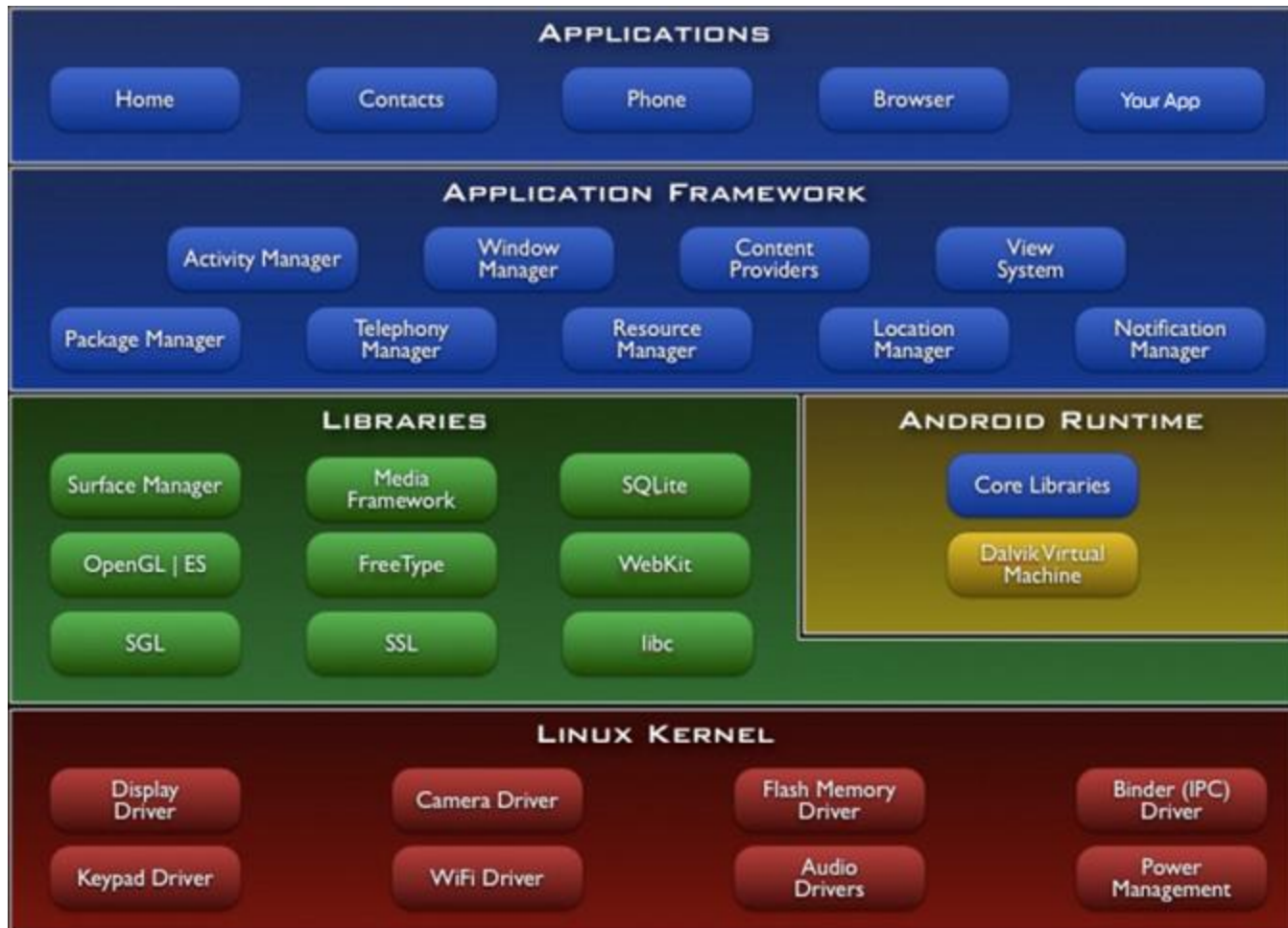


Sustavski (sistemski) softver

- Dizajniran da bi omogućio usluge drugom softveru:
 - Operacijski sustav
 - Pogoni igrica (*engine*)
 - Industrijska automatizacija
 - Chrome OS, Firefox OS (web preglednik je jedino korisničko sučelje)
- Ne može se deinstalirati bez utjecaja na drugi softver

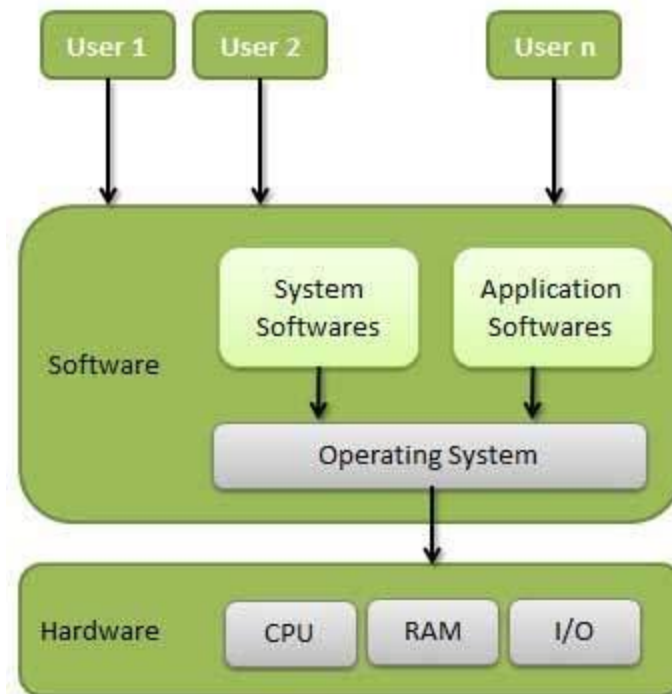


Arhitektura Androida - software



Operacijski sustavi

- Operacijski sustav je posrednik između hardvera računala i korisnika – omogućuje korisniku okruženje u kojem će jednostavno i efikasno izvršavati programe
- OS je softver koji upravlja hardverom



Operacijski sustavi

- Jezgra (kernel) – srž operacijskog sustava koji povezuje korisničke aplikacije i hardverske dijelove kroz sljedeće funkcije:
 - **Upravljanje memorijom** – kernel omogućuje siguran pristup memoriji po zahtjevu procesa koji se izvršava
 - **Upravljanje uređajima** – kernel upravlja na koji način procesi pristupaju određenom hardveru – pomoću drivera (upravljački program za uređaj)
 - **Sistemske pozivi** – program traži od kernela uslugu za koju nema dozvolu preko sistemskih poziva
 - **Uočavanje grešaka** – kernel omogućuje da greške u programu ne utječu na izvođenje drugih programa tako što zabranjuju pristup memoriji i ograničavaju CPU vrijeme

Jezgra (kernel) – primjer pokretanja igre

1. Igra (aplikacija) će **tražiti kernel** da koristi resurse (procesor, grafički procesor, memoriju).
2. **Kernel zatim upravlja tim zahtjevima** – odlučuje koliko procesorskog vremena igra dobiva, gdje će pohraniti svoje podatke u memoriji, i kako će koristiti grafički čip za prikaz slike.
3. Kernel pazi da igra ne ometa druge aplikacije (npr. web preglednik) koje se možda istovremeno izvršavaju.

Intuitivna metafora (zamislite bilo koju tvornicu):

Aplikacija = „radnik koji želi nešto obaviti” (npr. pokrenuti igru)

Hardver = „stroj u tvornici koji obavljaju navedeni posao” (npr. prikaz igre)

Kernel = „menadžer u tvornici”, prima zahtjeve radnika i govori strojevima što trebaju raditi i kako. Odlučuje kada i kako svaki „radnik” smije koristiti resurse „strojeva” da sve radi glatko i bez sudara.

Operacijski sustavi

- Operacijski sustavi nalaze se na mnogim uređajima koji sadrže računalo
 - Mobilni telefoni
 - Konzole video igara
 - Web serveri (poslužitelji)
 - Super računala
 - Satelitski prijamnici
 - Automobili

Operacijski sustavi

- Tipovi operacijskih sustava
 - Jedno i više korisnički (multiuser)
 - Jedno i više zadaćni (multitasking)
 - Distribuirani
 - Ugradbeni (embedded)
 - OS za rad u stvarnom vremenu (real-time)

Operacijski sustavi

- Početne generacije OS-a
 - U drugoj generaciji računala – *batch* obrada
 - U trećoj generaciji računala – MULTICS, UNIX, Minix i Linux
 - U četvrtoj generaciji računala – CP/M, DOS, MS-DOS, Windows i ostali

Operacijski sustavi

- Popularni OS-i:
 - Zasnovani na Unix-u:
 - Android
 - BlackBerry
 - BSD
 - Chrome OS
 - iOS
 - Linux
 - OS X
 - QNX
 - Steam OS
 - Microsoft Windows
 - z/OS
 - Hard real time: FreeRTOS, Micrium, VxWorks



Uvod u programiranje

- **Program** – niz instrukcija koje govore računalu kako obaviti zadatak – izvršava program
- **Računala su strojevi** – na osnovnoj razini dvopoložajne sklopke – 1 uključeno i 0 isključeno
- **Instrukcije** koje se izvršavaju su binarne, ovise o računalu – strojni kod
- **Programski jezik** olakšava korisniku izdavanje instrukcija računalu i zajednički je brojnim računalima
- Računalo mora **prevesti** ili **interpretirati** takav kod u strojni kod
- **Prevoditelj vs interpreter**

Uvod u programiranje

- Programski jezici:
 - Strojni jezik – 0,1
 - Asemblerski jezik (zbirni) – dvo-, tro-slovčane oznake
 - Jezici niže razine – ovise o hardveru (assembler)
 - Jezici više razine – ne ovise o hardveru
 - Proceduralni jezici – C, Go, Fortran, Pascal, BASIC, Python
 - Skriptni jezici – Perl, Tcl, Python, JavaScript, Lua, MATLAB
 - Objektno orijentirani jezici – Python, C++, Objective-C, Smalltalk, Delphi, Java, Swift, C#, Perl, Ruby, PHP

Uvod u programiranje

❓ Primjeri Hello world

❓ C

```
#include <stdio.h>
int main() {
    printf("Hello world!\n");
    return 0;
}
```

❓ BASIC

```
10 PRINT "Hello world!"
20 END
```

❓ Perl

```
#!/usr/bin/perl
print "Hello world!\n";
```

Uvod u programiranje

- Primjeri Hello world
- Java

```
public class HelloWorld
{
    public static void main(string[] args)
    {
        System.out.println("Hello world!");
    }
}
```

- Pascal

```
program HelloWorld;
begin
    writeln('Hello world!');
end.
```

Uvod u programiranje

- **Prevoditelj (compiler)**
 - većina ne prevode direktno u strojni kod nego u C kod
 - GCC (C, C++), ghc (Haskell), fpc (Pascal), V8 (JavaScript), Visual Basic 6 Native Code (BASIC)
- **Interpreter**
 - liniju po liniju prevodi u strojni kod
 - PHP 3 (PHP), SpiderMonkey (JavaScript)
- **Kombinirane opcije**
 - Virtualni stroj: javac, jvm (Sun Java), CPython (Python), .NET (C#)
 - apstraktna sintaksa stabla: perl (Perl)

Uvod u programiranje

- ❑ Programiranje uključuje više aktivnosti
 - ❑ analiza i razumijevanje problema
 - ❑ razvoj algoritama
 - ❑ provjeru zahtjeva algoritama i njihove računske složenosti
 - ❑ implementaciju (kodiranje) u odabranom programskom jeziku
 - ❑ testiranje, debugiranje, održavanje koda, implementacija izgradbenog sustava – razvoj softvera

Uvod u programiranje

- Osnovni ciljevi programiranja
 - pronaći niz instrukcija koje će **automatizirati izvedbu određenog zadatka ili riješiti zadani problem**
 - osigurati da program **ostvaruje zadatak**
 - osigurati da drugi korisnici mogu **jednostavno koristiti program**
 - osigurati **jednostavnost shvaćanja i nadogradnje programa** bez prevelikog utrošenog vremena
 - koristiti **jednostavna i razumljiva imena varijabli**
 - osigurati **čitljivost i učinkovitost koda**
 - iskoristiti prednost **postojanja “Free Software”**
 - (ako je moguće) objaviti kod pod GPL (GNU General Public License)

Uvod u programiranje

❓ Povijest programiranja

- ❓ matematičarka **Ada Lovelace** napisala je prvi program za analitički stroj – izračun bernulijeva niza brojeva
- ❓ **von Neumannova arhitektura** – instrukcije (elementarne binarne operacije) pohranjene u memoriji računala – strojni jezik
- ❓ **asemblerski jezik** omogućio tekstualni format instrukcija
- ❓ **1954. FORTRAN** – prvi visokorazinski programski jezik (“FORmula TRANslation”) – koristi se u inženjerskim aplikacijama
- ❓ "noviji" jezici: **COBOL** – koristi se u korporativnim podatkovnim centrima, **skriptni jezici** – često se koriste za razvoj web stranica, **C** – često se koristi u ugradbenim aplikacijama

Uvod u programiranje

❓ zašto python:

- ❓ postoje moćniji programski jezici (Lisp)
- ❓ postoje brži programski jezici (C)
- ❓ postoje više korišteni programski jezici (Java)
- ❓ postoje egzotičniji jezici (Haskell)

Uvod u programiranje

❓ zašto python:

- ❓ dobra kombinacija različitih parametara:
 - ❓ dobra čitljivost koda,
 - ❓ jednostavnost pisanja koda,
 - ❓ brzina razvoja koda,
 - ❓ ogroman skup biblioteka na raspolaganju
 - ❓ brzina automatiziranja zadataka
 - ❓ fleksibilnost prema traženom zadatku umjesto okruženju

Uvod u programiranje

	Fortran	Julia	Python	R	Matlab	Octave	Mathematica	JavaScript	Go	LuaJIT
	gcc 4.8.2	0.3.7	2.7.9	3.1.3	R2014a	3.8.1	10.0	V8 3.14.5.9	go1.2.1	gsl-shell 2.3.1
fib	0.57	2.14	95.45	528.85	4258.12	9211.59	166.64	3.68	2.20	2.02
parse_int	4.67	1.57	20.48	54.30	1525.88	7568.38	17.70	2.29	3.78	6.09
quicksort	1.10	1.21	46.70	248.28	55.87	1532.54	48.47	2.91	1.09	2.00
mandel	0.87	0.87	18.83	58.97	60.09	393.91	6.12	1.86	1.17	0.71
pi_sum	0.83	1.00	21.07	14.45	1.28	260.28	1.27	2.15	1.23	1.00
rand_mat_s tat	0.99	1.74	22.29	16.88	9.82	30.44	6.20	2.81	8.23	3.71
rand_mat_ mul	4.05	1.09	1.08	1.63	1.12	1.06	1.13	14.58	8.45	1.23

vremena izvođenja u odnosu na C (manje je bolje, C izračun = 1.0)

Uvod u programiranje

- koristit ćemo verziju Python 3.12.6
- <https://www.python.org/downloads/>

Laboratorijska vježba (pretpostavka: studenti posjeduju prijenosna računala)

1. Preuzeti i instalirati Python
2. Pokrenuti program IDLE koji služi za razvoj programa u Pythonu
3. Napraviti svoj prvi program koristeći sljedeću naredbu:

```
print(„Bravo ja“)
```