



# Operativni sistemi

## Uvod i pregled operativnih sistema

**Prof. dr Dragan Stojanović**

Katedra za računarstvo  
Univerzitet u Nišu, Elektronski fakultet

# Literatura

- ✿ *Operating Systems: Internals and Design Principles*, 9<sup>th</sup> edition – 2017, Pearson Education Inc. (5<sup>th</sup> edition - 2005, 6<sup>th</sup> edition - 2008, 7<sup>th</sup> edition - 2012, 8<sup>th</sup> edition – 2014)

- ✦ <http://williamstallings.com/OperatingSystems/>
- ✦ <http://williamstallings.com/OperatingSystems/OS9e-Student/>

- ✿ Poglavlje 2: Pregled operativnog sistema

# Operativni sistem (OS)

- ✿ **Operativni sistem** predstavlja organizovanu kolekciju programa koji upravlja izvršavanjem aplikativnih programa i služi kao interfejs između aplikativnih/sistemskih programa i hardvera računara
- ✿ Ciljevi i funkcije operativnog sistema:
  - ✦ **Pogodnost** - Da omogući lako i pogodno korišćenje računara
  - ✦ **Efikasnost** - Da obezbedi efikasno korišćenje i upravljanje resursima računara
  - ✦ **Mogućnost razvoja** - Da obezbedi osnovu za efikasan razvoj, testiranje i uvođenje novih funkcija sistema

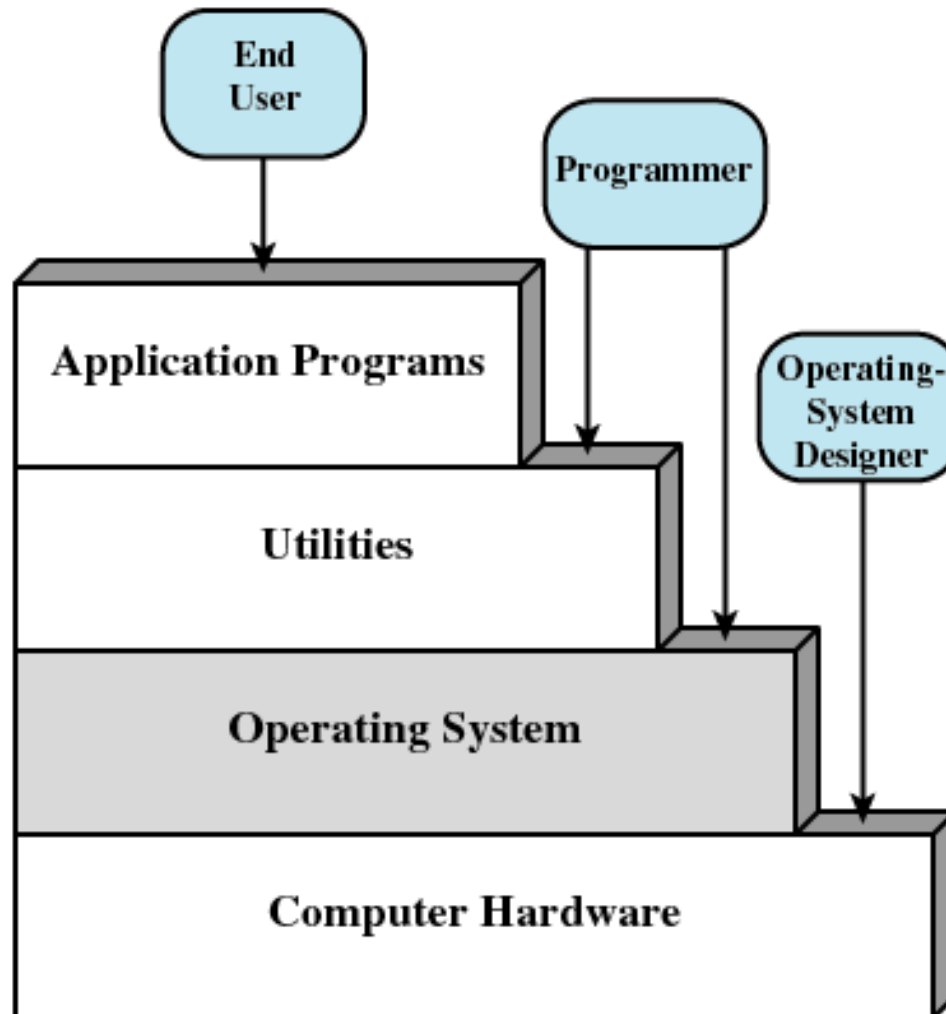


# OS kao interfejs između korisnika i računara

- ✿ Korisnik nije zainteresovan za detalje računarskog hardvera, već vidi računarski sistem kao skup aplikacija
- ✿ Korisnik interaguje sa OS-om pomoću komandi komandnog jezika (shell-a), ili preko grafičkog korisničkog interfejsa (GUI)
- ✿ Programer (*software developer*) pristupa računarskom sistemu i razvija aplikacije pomoću sistemskog softvera
  - ✦ Sistemski softver čine sistemski programi neophodni za razvoj i izvršenje aplikacija na računaru (uslužni programi) i upravljanje resursima računarskog sistema
  - ✦ Najvažniji sistemski softver je **operativni sistem**
  - ✦ OS sakriva detalje hardvera od programera i obezbeđuje mu jednostavan i prikladan interfejs za korišćenje računarskog sistema



# OS kao interfejs između korisnika i računara

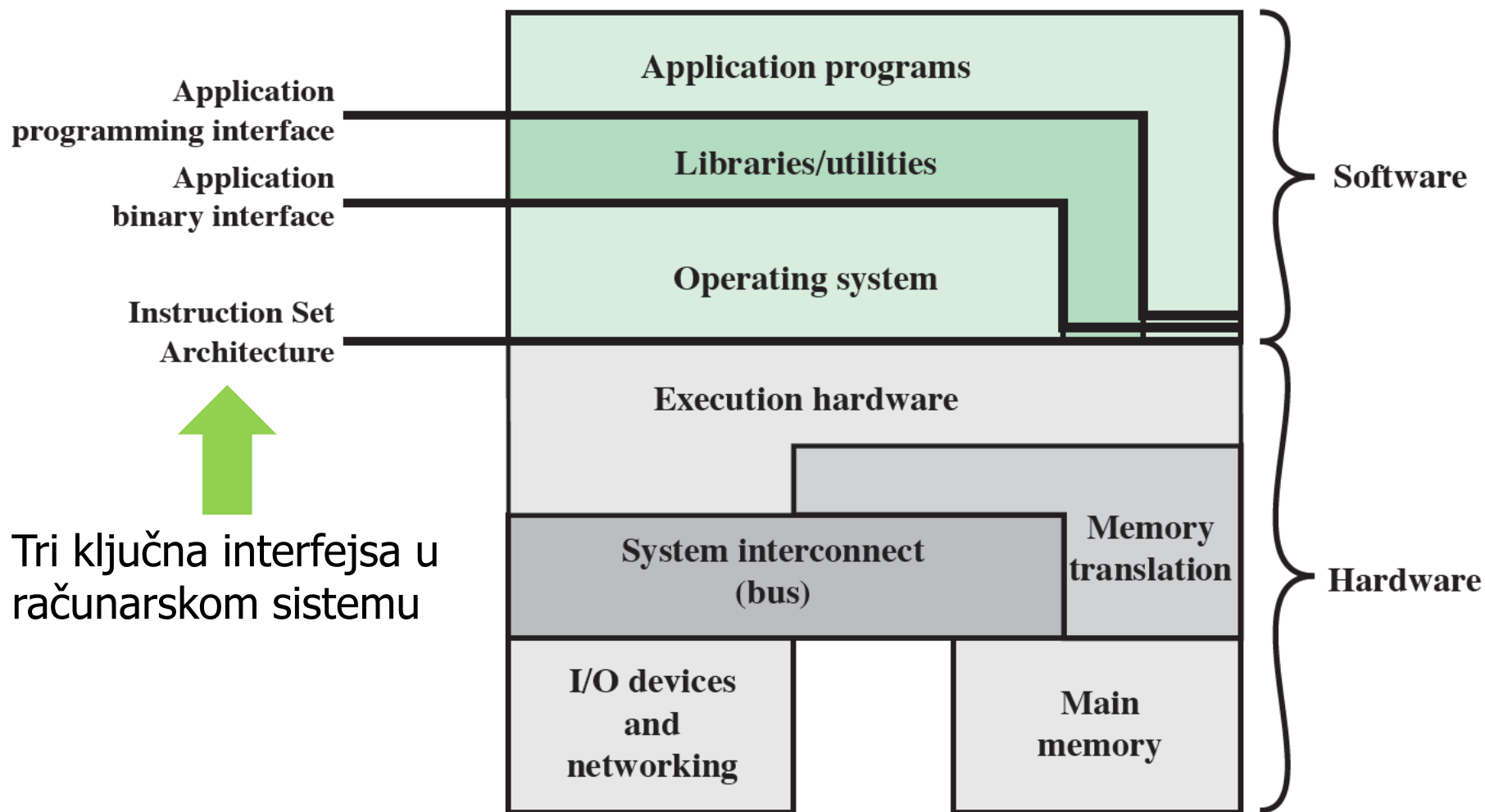


Uvod i pregled operativnih sistema

Operativni sistemi



# Struktura hardvera i softvera računara



# Servisi operativnog sistema

Operativni sistem obezbeđuje servise u sledećim domenima:

- ❖ Razvoj programa
- ❖ Izvršavanje programa
- ❖ Pristup U/I uređajima
- ❖ Kontrolisan pristup datotekama
- ❖ Pristup sistemu i upravljanje pristupom sistemskim resursima
- ❖ Otkrivanje grešaka i odgovor na greške
- ❖ Obračun korišćenja resursa sistema i nadgledanje performansi

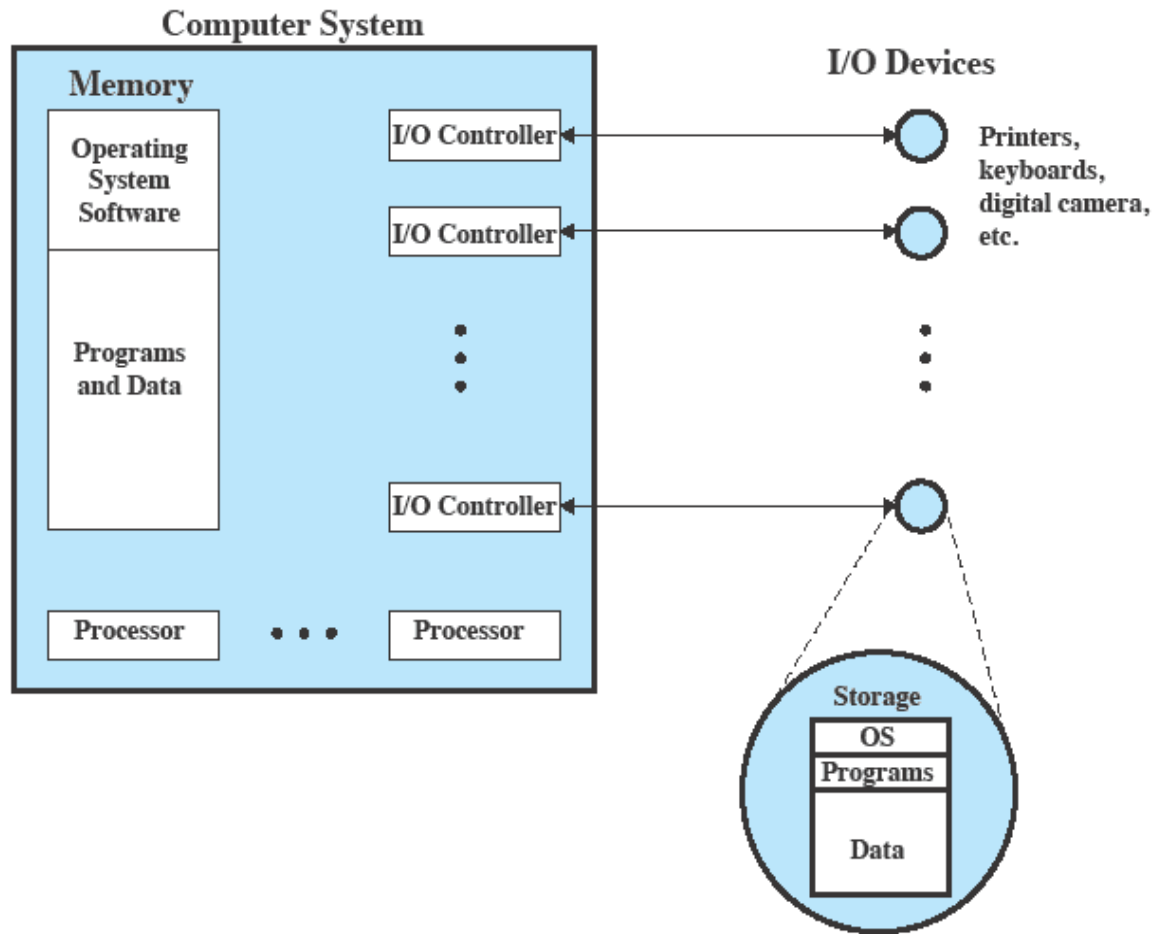
# OS kao menadžer resursa

- ✿ Operativni sistem upravlja **resursima računarskog sistema**, a to su hardverski uređaji (procesor, memorija, štampač, disk, kamera,...) ili strukture podataka (datoteka, semafor, slog u bazi podataka, bafer poruka,...) koji su na raspolaganju korisnicima i programima.
- ✿ Operativni sistem je softver, skup programa koji se izvršavaju na procesoru.
- ✿ OS se povremeno odriče izvršavanja na procesoru, i „predaje“ procesor na izvršavanje korisničkog programa.
- ✿ Kada se desi neki događaj u računarskom sistemu (prekid, trap), operativni sistem „preuzima procesor“ i izvršava se na procesoru da bi obavio odgovarajuće upravljačke funkcije nad resursima.



# OS kao upravljač resursa

- Deo OS se nalazi u glavnoj memoriji
  - Kernel** (jezgro) - sadrži najbitnije funkcije u okviru OS
  - Delovi OS koji se trenutno koriste
- Ostatak memorije sadrži korisničke programe i podatke
- OS upravlja dodelom procesora, memorije, U/I,... korisničkim programima



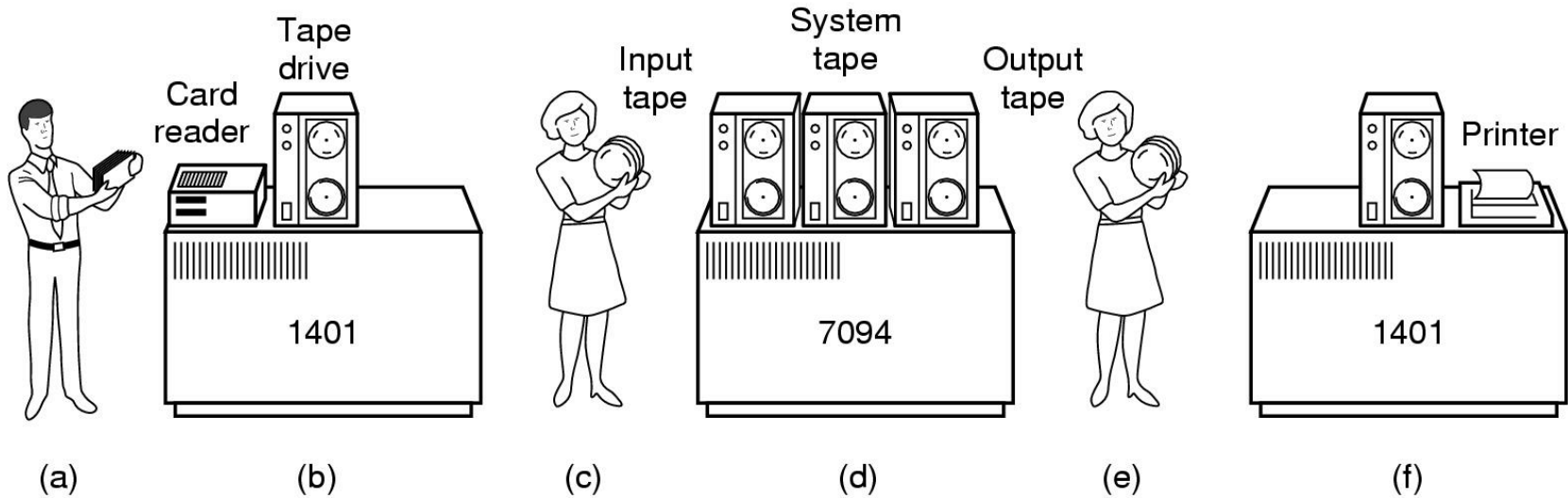
# Lakoća evolucije OS

- ✿ Operativni sistemi moraju biti sposobni da evoluiraju tokom vremena iz sledećih razloga:
  - ✦ Nadogradnja hardvera i razvoj novih tipova hardvera
  - ✦ Novi servisi: kao odgovor na zahteve korisnika ili sistemskih administratora
  - ✦ Korekcija grešaka
- ✿ Operativni sistem mora biti modularne strukture sa jasno definisanim interfejsima između modula i dobro dokumentovan

# Razvoj operativnih sistema

- ✚ **Serijska obrada** (1945 – 1955)
  - ✚ Vakumske cevi, bušene kartice, mašinski jezik
  - ✚ **Nema OS-a**, programeri su pristupali direktno hardveru
- ✚ Jednostavni **sistemi paketne obrade** (*batch systems*) (1955 – 1965)
  - ✚ Tranzistori, mainframe računari, asemblerski jezik, FORTRAN, COBOL
  - ✚ **Monitor** – jednostavan OS (IBSYS – IBM OS za 7090/7094 računare)
- ✚ Multiprogramirani sistemi paketne obrade
- ✚ Sistemi sa deljenjem vremena (*time sharing*) (1965-1980)
  - ✚ Integrisana kola, mini računari i radne stanice, C, UNIX
  - ✚ **Multiprogramiranje, timesharing**
  - ✚ IBM System/360, *Compatible Time-Sharing System* (CTSS), UNIX,...
- ✚ Personalni računari (1980 – danas)
  - ✚ LSI/VLSI, mikroprocesori, personalni računari (PC), mikroračunari
  - ✚ Windows, Apple Mac OS, UNIX, Linux, ...
- ✚ Distribuirani, paralelni, mobilni računari (1990 - danas)
  - ✚ Multiprocesorski sistemi, distribuirani sistemi, sistemi za rad u realnom vremenu, mobilni računari (pametni telefoni, tableti)
- ✚ *Cloud computing*, Sveprisutno računarstvo, IoT (*Internet of Things*),...

# Sistemi paketne obrade (batch sistemi)



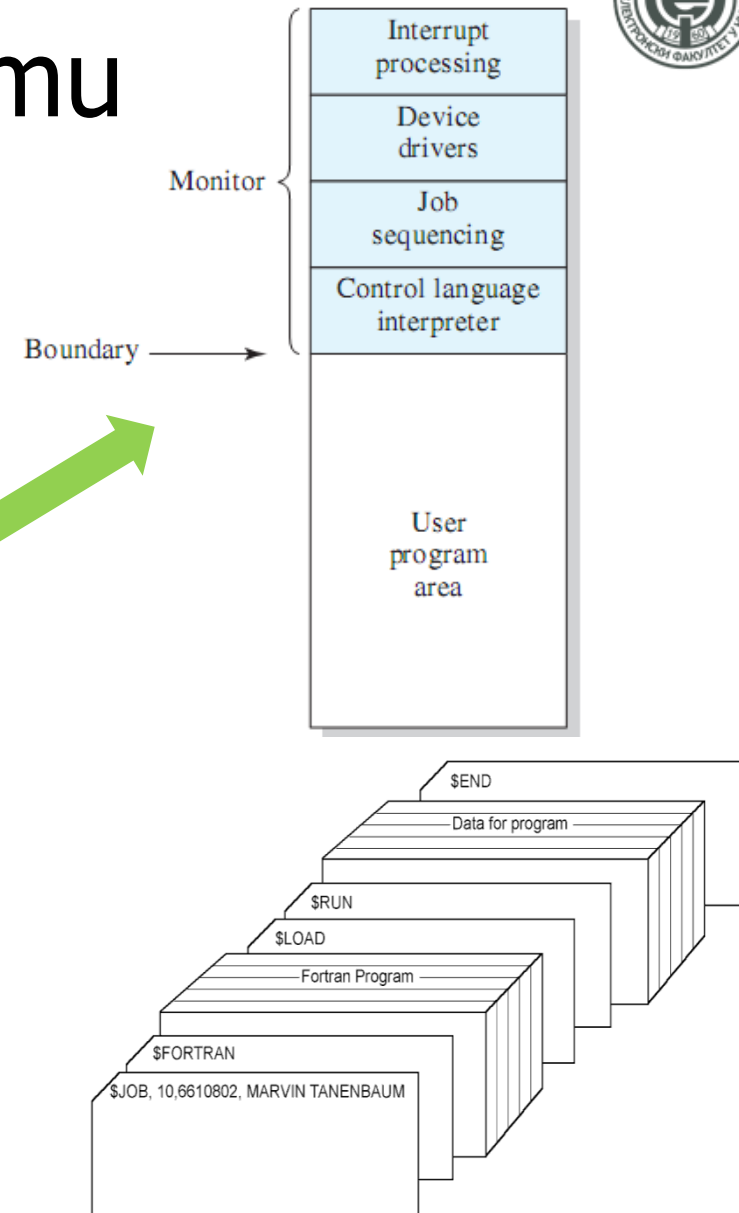
## Stari mainframe računari

*Modern Operating Systems,  
Tanenbaum, 2014*

- ❑ Bušene kartice se unose u IBM 1401
- ❑ Čitanje sa kartica i snimanje na magnetnu traku
- ❑ Postavljanje trake na IBM 7094 koji obavlja obradu i rezultat snima na magnetnu traku (IBSYS operativni sistem)
- ❑ Postavljanje trake na IBM 1401 i štampanje

# Monitor u *batch* sistemu

- Struktura tipičnog posla (*job*) zadatog bušenim karticama
- Operativni sistem - **Monitor**
  - FMS (Fortran Monitor System)
  - IBSYS (IBM-ov OS za 7094 računar)
- Monitor** je stalno smešten u glavnoj memoriji dostupan za izvršavanje (rezidentni monitor)
- Čita sa ulaznog uređaja jedan po jedan posao (*job*), smešta instrukcije i podatke u korisnički deo memorije i startuje izvršenje posla na procesoru.
- Po završetku, monitor učitava i izvršava sledeći posao (*job*).
- Instrukcije se monitoru zadaju preko *Job Control Language* (JCL)

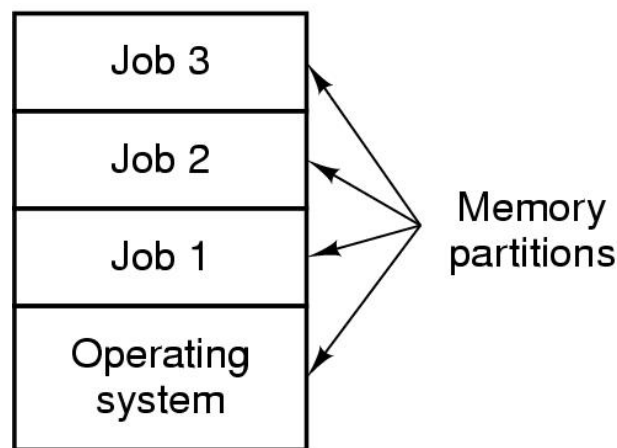


# Monitor – operativni sistem

- ✿ **Monitor** (OS sa paketnom obradom) je računarski program koji je smešten u deo glavne memorije i naizmenično se izvršava sa korisničkim programima
- ✿ Poželjna svojstva hardvera:
  - ✦ Zaštita memorije koju zauzima monitor
  - ✦ Tajmer
  - ✦ Privilegovane mašinske instrukcije – može ih izvršiti samo monitor
  - ✦ Prekidi
- ✿ Dva režima rada
  - ✦ **Kernel režim** (mod)
    - Monitor (operativni sistem) se izvršava u kernel modu
  - ✦ **Korisnički režim** (mod)
    - Korisnički programi se izvršavaju u korisničkom modu, koriste samo podskup iz skupa instrukcija i samo neke mogućnosti HW (generalno, instrukcije za U/I i zaštitu memorije su zabranjene u korisničkom modu)
    - Za ostalo korisnički programi pozivaju funkcije (servise) OS-a

# Multiprogramski sistem paketne obrade

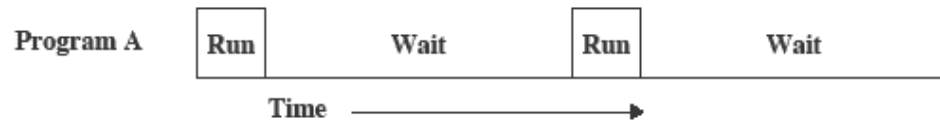
- ❁ **Multiprogramiranje** (*multitasking*) – Operativni sistem istovremeno smešta u memoriju više poslova; u jednom trenutku samo jedan od poslova se izvršava na CPU, ukoliko se blokira izvršenjem U/I operacije (npr. čitanje podataka sa diska), aktivira se planiranje poslova
- ❁ **Planiranje poslova** (*Job scheduling*) – OS mora da iz skupa svih poslova izabere one koji će biti smešteni u memoriju i odrediti jedan koji će se izvršavati - planiranje CPU (CPU scheduling)
- ❁ **Dodatna svojstva hardvera**
  - ❑ U/I prekidi i DMA
  - ❑ Upravljanje memorijom
- ❁ **Operativni sistemi:**
  - ❑ OS/360
  - ❑ MULTICS
  - ❑ UNIX (System V, BSD)



Uvod i pregled operativnih sistema

# Multiprogramiranje

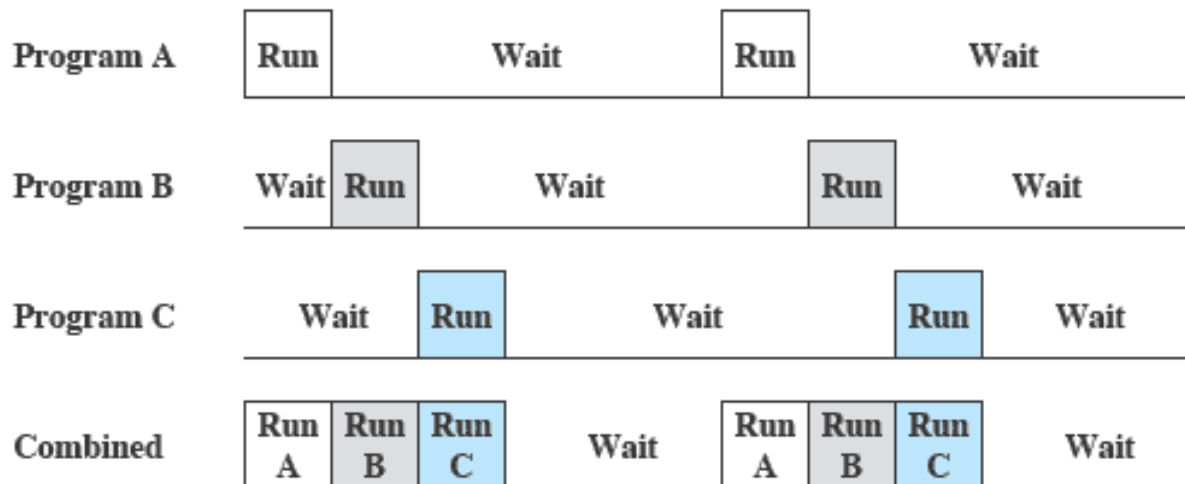
- Monoprogramiranje - CPU mora da čeka dok se ne završi U/I instrukcija



|                           |                             |
|---------------------------|-----------------------------|
| Read one record from file | 15 $\mu$ s                  |
| Execute 100 instructions  | 1 $\mu$ s                   |
| Write one record to file  | 15 $\mu$ s                  |
| <b>TOTAL</b>              | <b>31 <math>\mu</math>s</b> |

$$\text{Percent CPU Utilization} = \frac{1}{31} = 0.032 = 3.2\%$$

- Multiprogramiranje sa tri aktivirana programa





# Primer multiprogramiranja

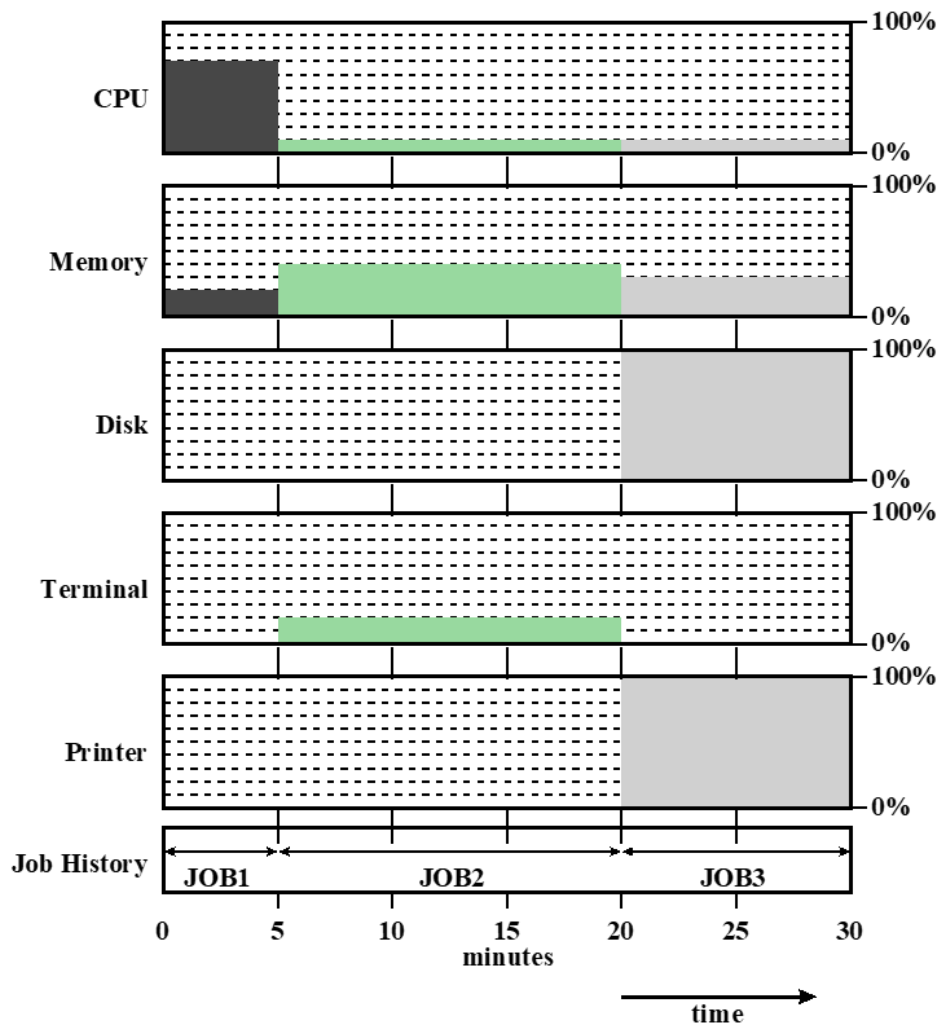
## Primeri atributa za izvršavanje programa (250MB memorije)

|                 | JOB1          | JOB2      | JOB3      |
|-----------------|---------------|-----------|-----------|
| Type of job     | Heavy compute | Heavy I/O | Heavy I/O |
| Duration        | 5 min         | 15 min    | 10 min    |
| Memory required | 50 M          | 100 M     | 75 M      |
| Need disk?      | No            | No        | Yes       |
| Need terminal?  | No            | Yes       | No        |
| Need printer?   | No            | No        | Yes       |

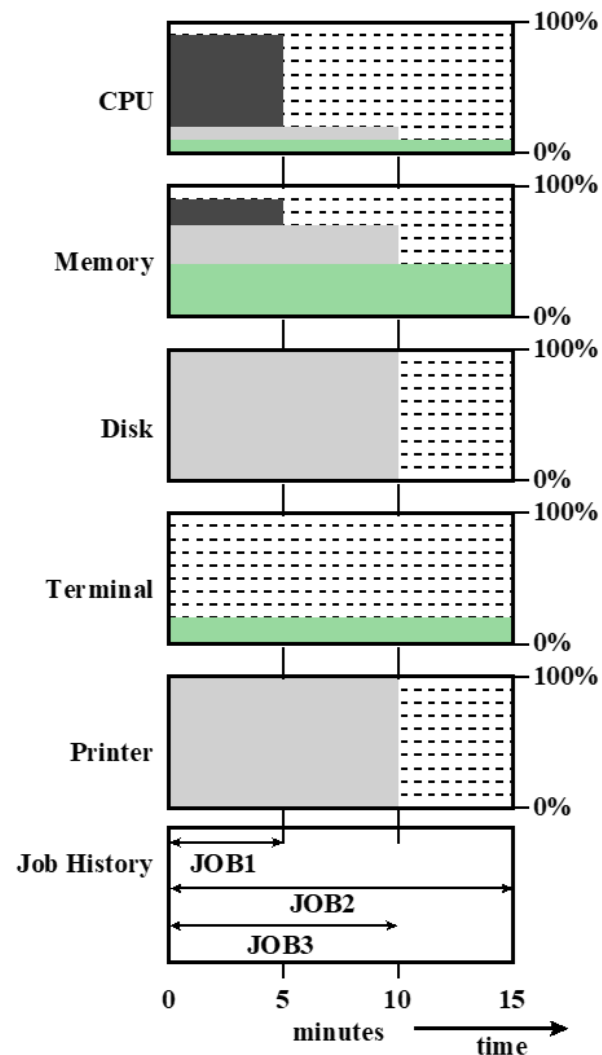
## Efekti multiprogramiranja na iskorišćenje resursa

|                    | Uniprogramming | Multiprogramming |
|--------------------|----------------|------------------|
| Processor use      | 20%            | 40%              |
| Memory use         | 33%            | 67%              |
| Disk use           | 33%            | 67%              |
| Printer use        | 33%            | 67%              |
| Elapsed time       | 30 min         | 15 min           |
| Throughput         | 6 jobs/hr      | 12 jobs/hr       |
| Mean response time | 18 min         | 10 min           |

# Histogram iskorišćenosti resursa



(a) Uniprogramming



(b) Multiprogramming

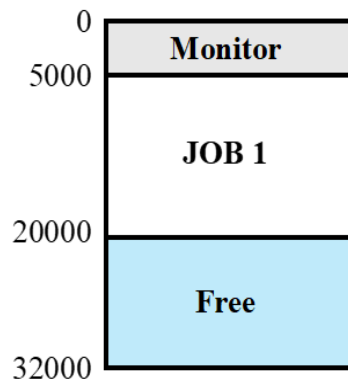
# Sistemi sa deljenjem vremena

- ✿ ***Time-sharing*** – procesorsko vreme je podeljeno između više korisnika
- ✿ Koristi multiprogramiranje za višekorisnički rad pri čemu svaki korisnik pristupa sistemu interaktivno putem terminala
- ✿ Svakom korisničkom programu se dodeljuje po jedan mali vremenski period (deo procesorskog vremena) za izvršavanje, pre nego što se pređe na drugi program
- ✿ Jedan od prvih *time-sharing* OS je CTSS (*Compatible Time-Sharing System*) razvijen 1961-te na MIT za IBM 709, a kasnije prenet na IBM 7094
  - ✦ Računar sa glavnom memorijom od 32000 36-bitnih reči, pri čemu monitor zauzima 5000 reči

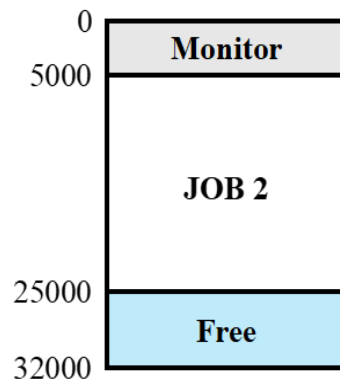


# Primer rada CTSS

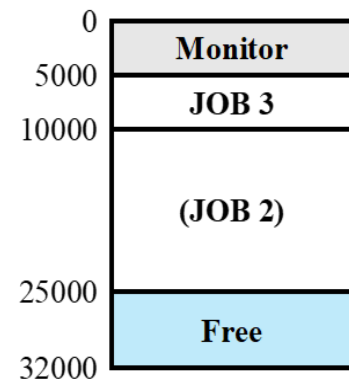
- JOB1: 15000
- JOB2: 20000
- JOB3: 5000
- JOB4: 10000



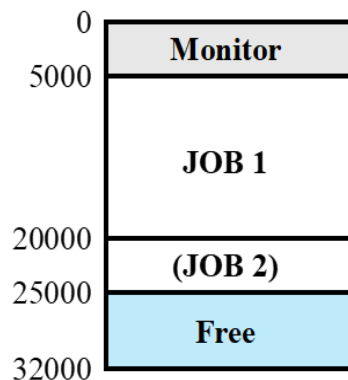
(a)



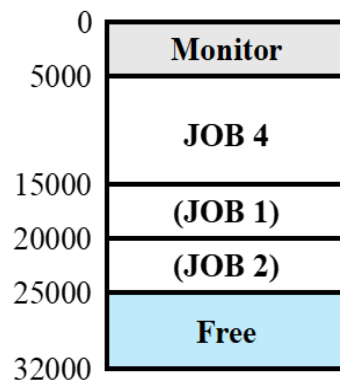
(b)



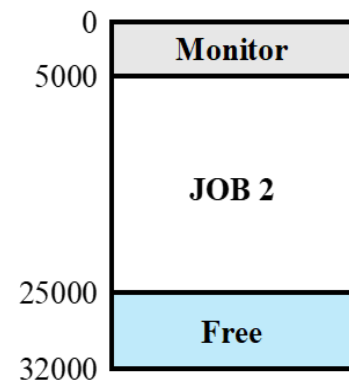
(c)



(d)



(e)



(f)

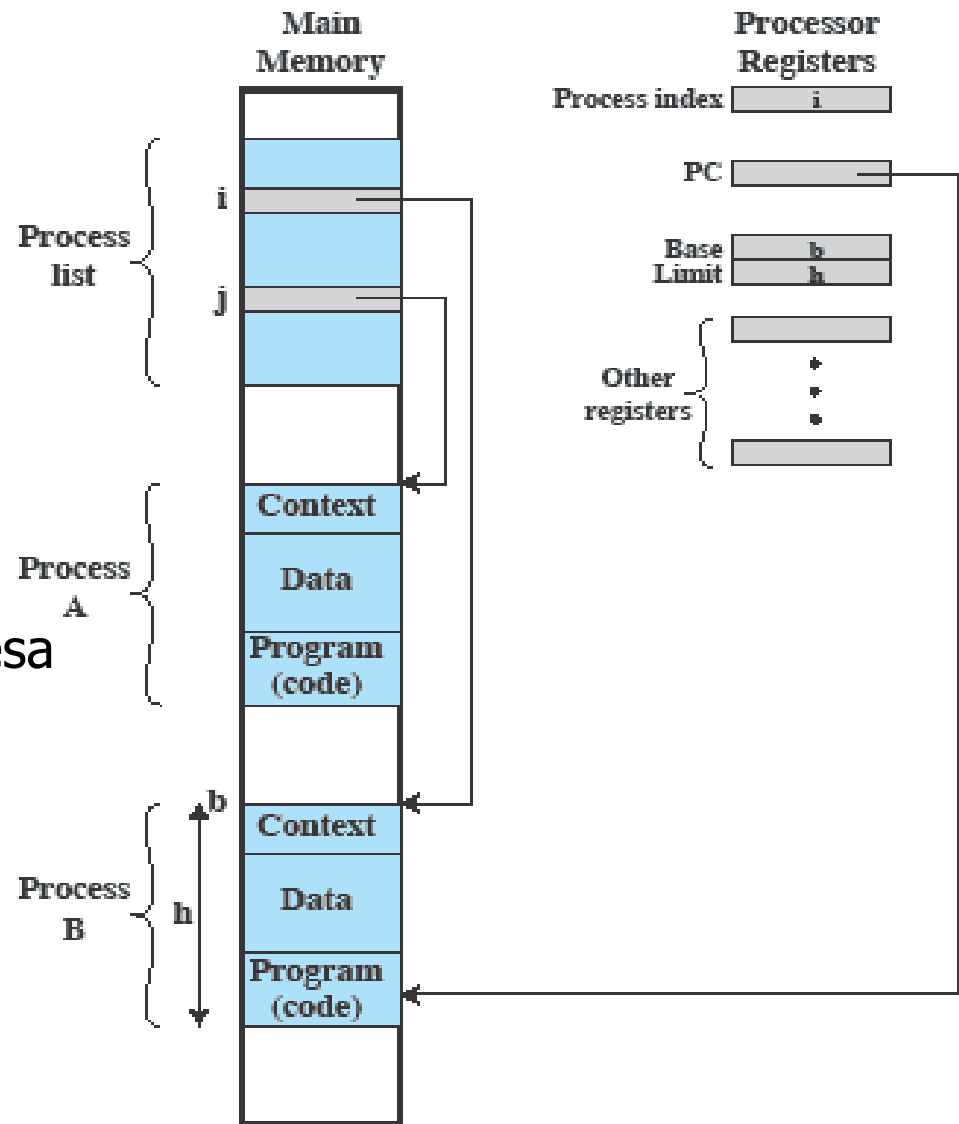


# Glavna dostignuća u razvoju OS

- ❖ Procesi
- ❖ Upravljanje memorijom
- ❖ Zaštita i bezbednost informacija
- ❖ Planiranje i upravljanje resursima
- ❖ Struktura sistema

# Procesi

- Proces je program u izvršenju
- Proces se sastoji od tri komponente
  - Izvršni program
  - Podaci koji se obrađuju u programu
  - Kontekst izvršenja** procesa
- Tipična implementacija procesa prikazana na slici



# Upravljanje memorijom

## ✿ Osnovne odgovornosti OS

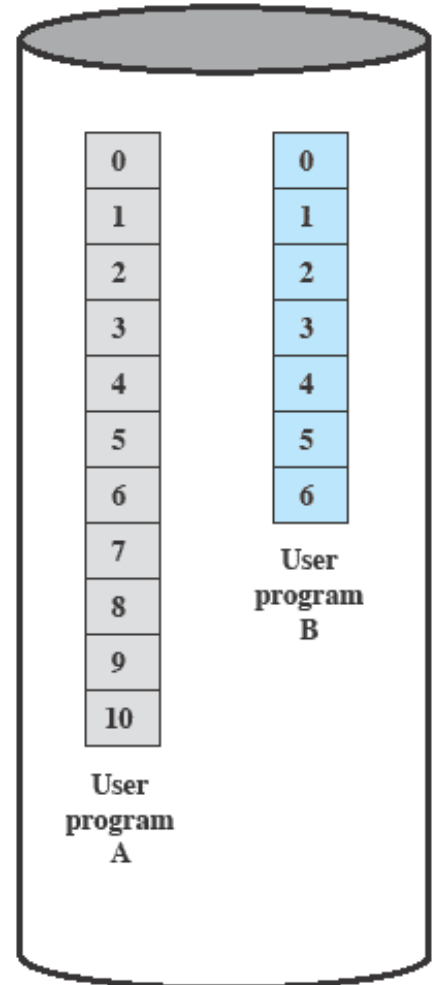
- ✦ Izolacija procesa
- ✦ Automatsko dodeljivanje i upravljanje
- ✦ Podrška za modularno programiranje
- ✦ Zaštita i kontrola pristupa
- ✦ Korišćenje dugotrajne memorije

## ✿ Koncept virtuelne memorije i *file* sistema

- ✦ Virtuelna adresa
- ✦ Realna (fizička) adresa u glavnoj memoriji

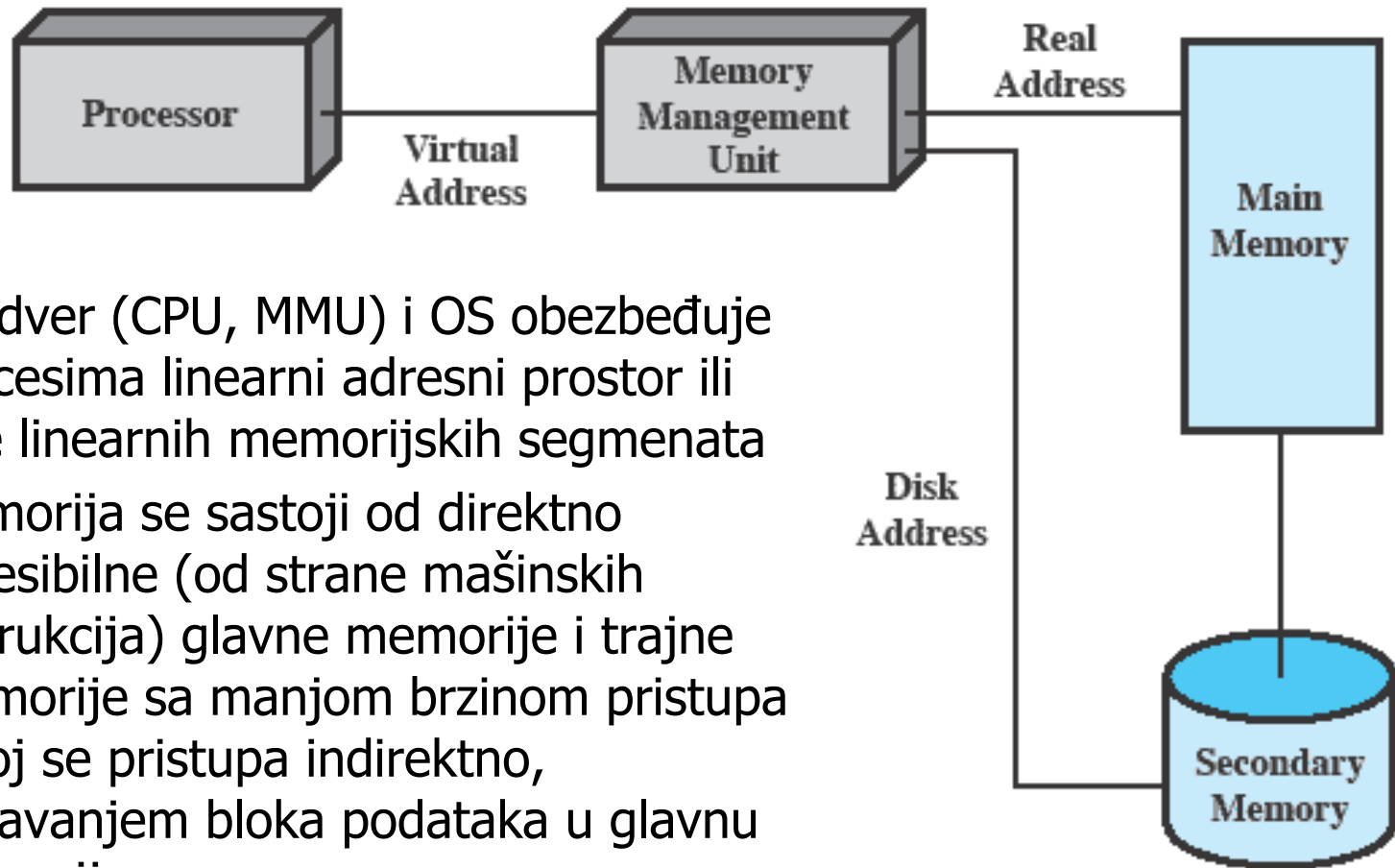
|     |     |     |     |
|-----|-----|-----|-----|
| A.1 |     |     |     |
|     | A.0 | A.2 |     |
|     | A.5 |     |     |
| B.0 | B.1 | B.2 | B.3 |
|     |     |     |     |
|     |     |     |     |
|     |     |     |     |
|     |     | A.7 |     |
|     | A.9 |     |     |
|     |     |     |     |
|     |     | A.8 |     |
|     |     |     |     |
|     |     |     |     |
|     |     |     |     |
|     | B.5 | B.6 |     |
|     |     |     |     |

Main Memory



Disk

# Adresiranje virtuelne memorije



- ✚ Hardver (CPU, MMU) i OS obezbeđuje procesima linearni adresni prostor ili više linearnih memorijskih segmenata
- ✚ Memorija se sastoji od direktno adresibilne (od strane mašinskih instrukcija) glavne memorije i trajne memorije sa manjom brzinom pristupa kojoj se pristupa indirektno, učitavanjem bloka podataka u glavnu memoriju



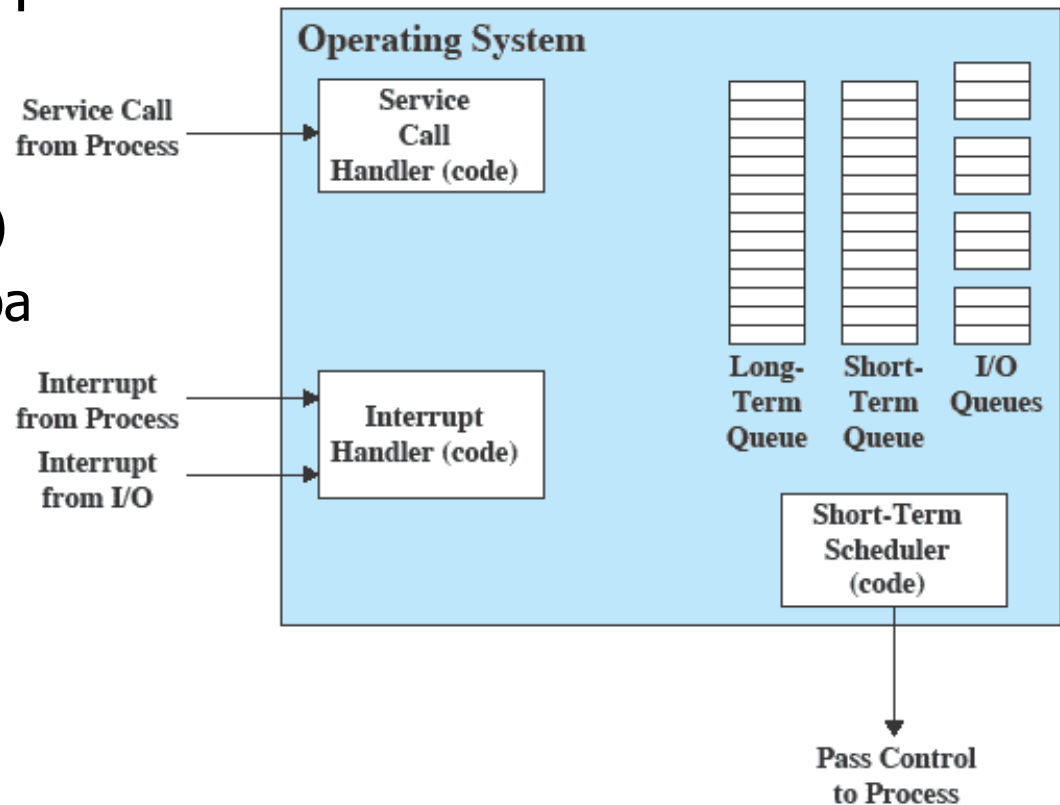
# Zaštita informacija i bezbednost

- ✚ Raspoloživost
  - ✚ Zaštita sistema od prekida funkcionisanja
- ✚ Poverljivost (tajnost)
  - ✚ Zaštita podataka od neovlašćenog pristupa
- ✚ Integritet podataka
  - ✚ Zaštita podataka od neautorizovane modifikacije
- ✚ Autentičnost
  - ✚ Pogodna verifikacija identiteta korisnika i validnosti poruka i podataka

# Raspoređivanje resursa i upravljanje

Strategija raspoređivanja i dodele resursa mora da uključi 3 faktora:

- ❖ Nepristrasnost (pravičnost)
- ❖ Različitost odgovora - Treba napraviti razliku između različitih klasa procesa
- ❖ Efikasnost
  - Maksimizovanje propusne moći (*throughput*),
  - Minimizovanje vremena odziva (*response time*) i
  - Opsluživanje što više korisnika (*time-sharing*)



Ključni elementi OS za multiprogramiranje



# Pravci razvoja savremenih OS

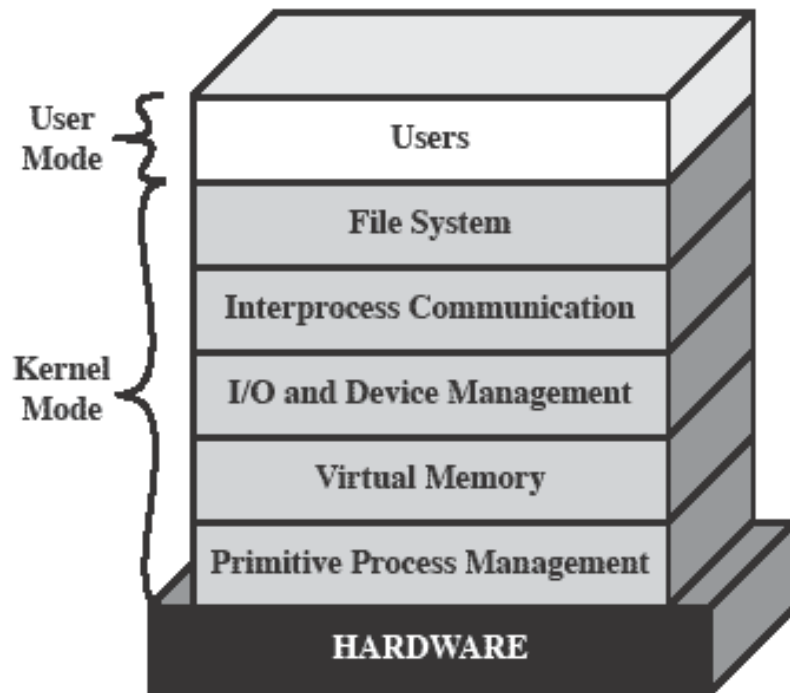
- ❖ Mikrokernel arhitektura
- ❖ Višenitna obrada (*Multithreading*)
- ❖ Paralelne računarske arhitekture
  - ❖ Simetrično multiprocesiranje (*symmetric multiprocessing-SMP*) –*multicore* arhitekture
- ❖ Distribuirani operativni sistemi (*cloud, cluster*) i real-time embedded OS (IoT)
- ❖ Objektno-orijentisani dizajn

# Arhitektura operativnih sistema

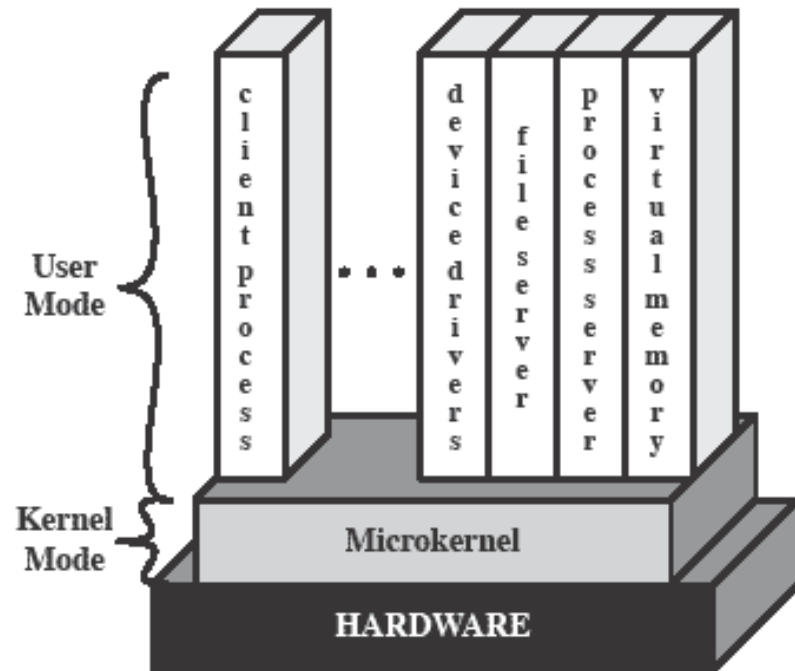
|                                   |                                   |
|-----------------------------------|-----------------------------------|
| GUI & komandni jezik              | Upravljanje zaštitom i sigurnošću |
| Upravljanje datotekama            |                                   |
| Upravljanje mrežnom komunikacijom |                                   |
| Upravljanje U/I uređajima         |                                   |
| Upravljanje memorijom             |                                   |
| Upravljanje procesima i nitima    |                                   |
| Hardver računara                  |                                   |

# Arhitektura OS kernela

## Monolitna (slojevita) i mikrokernel arhitektura



(a) Layered kernel



(b) Microkernel

# Monolitna arhitektura OS

- Operativni sistem je kolekcija procedura. Pri čemu svaka može pozivati svaku poznajući njen interfejs (skup parametara i rezultat) i svaka procedura može pristupati deljivim podacima i strukturama podataka OS

- Prednosti:

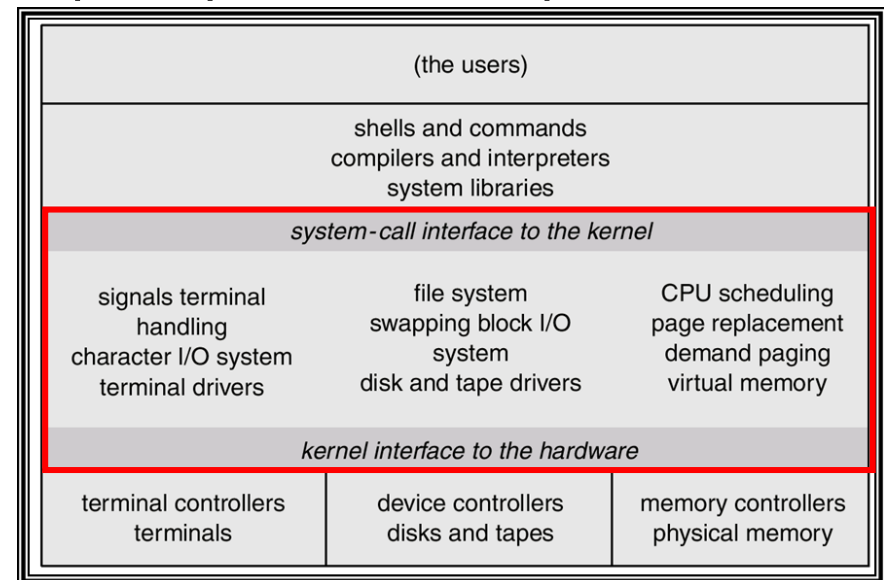
- Performanse i visok nivo zaštite od pristupa korisničkih procesa

- Nedostaci:

- Loša proširljivost, održavanje
  - Loša zaštita između komponenti kernela

- UNIX - OS sadrži dva dela

- Sistemske programi i kernel



# Mikrokernel arhitektura OS

- ✿ Samo kritični OS procesi se izvršavaju u režimu kernela, npr. pristup U/I uređajima i U/I drajveri
- ✿ Ostale funkcije OS implementirane su kao servisi koji se izvršavaju u korisničkom režimu
- ✿ Prednosti
  - ✦ Jednostavno proširenje OS jer dodavanje novih servera ne zahteva modifikaciju kernela
  - ✦ Jednostavno portovanje OS sa jedne na drugu hardversku platformu
  - ✦ Pošto se svi serverski procesi izvršavaju u korisničkom modu, greška u nekom od njih ne uzrokuje pad OS
- ✿ Primeri:
  - ✦ Mach (*Carnegie Mellon University*, sredina 1980-ih)
  - ✦ Tru64UNIX (ranije Digital UNIX)
  - ✦ Apple macOS (ranije MacOS X) - Mach kernel + deo BSD kernela
  - ✦ QNX, MINIX



# Višenitna obrada (*Multithreading*)

- ✿ Proces je podeljen u **niti** koje mogu da se izvršavaju konkurentno (paralelno)
  - ✦ **Nit** (*thread*)
    - Jedinica izvršenja koja se može planirati i rasporediti za izvršenje
    - Izvršava se sekvencijalno i može biti prekinuta i ponovo nastavljena
  - ✦ **Proces** je skup jedne ili više niti i pridruženih sistemskih resursa, poput memorije koja sadrži kod i podatke, otvorenih datoteka, i U/I uređaja
- ✿ **Višenitnost** je korisna u aplikacijama koje obavljaju više suštinski nezavisnih zadataka koji ne moraju serijski da se izvršavaju
  - ✦ Primer: Web server koji prihvata i opslužuje zahteve klijenata

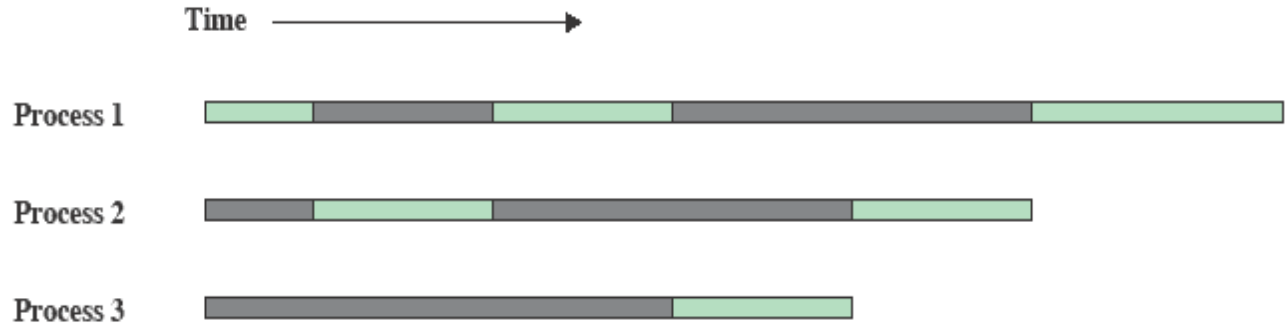


# Simetrično multiprocessing (SMP)

- ✿ Postoji više procesora u sistemu
- ✿ Ovi procesori dele istu glavnu memoriju i U/I resurse
- ✿ Svi procesori mogu izvršavati iste funkcije
- ✿ Prednosti:
  - ✦ Performanse
  - ✦ Raspoloživost u slučaju otkaza jednog procesora
  - ✦ Inkrementalno povećanje performansi dodavanjem dodatnih procesora
  - ✦ Skaliranje – može postojati više računarskih konfiguracija sa različitim brojem procesora sa različitom cenom i performansama

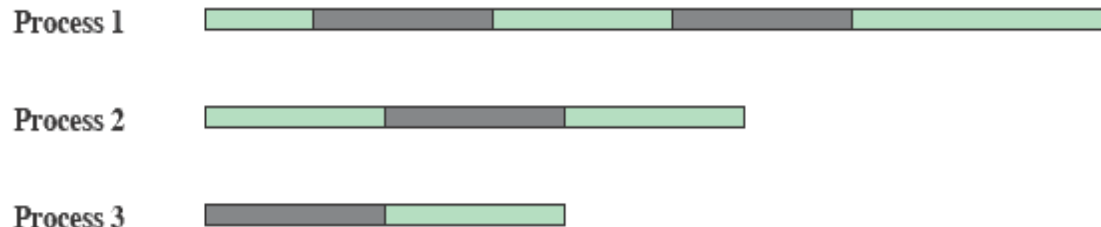
# Simetrično multiprocessing (SMP)

## • Multiprogramiranje (jedan procesor)



(a) Interleaving (multiprogramming, one processor)

## • Multiprocesiranje (dva procesora)



(b) Interleaving and overlapping (multiprocessing; two processors)

Blocked Running

# Sistemske pozivi

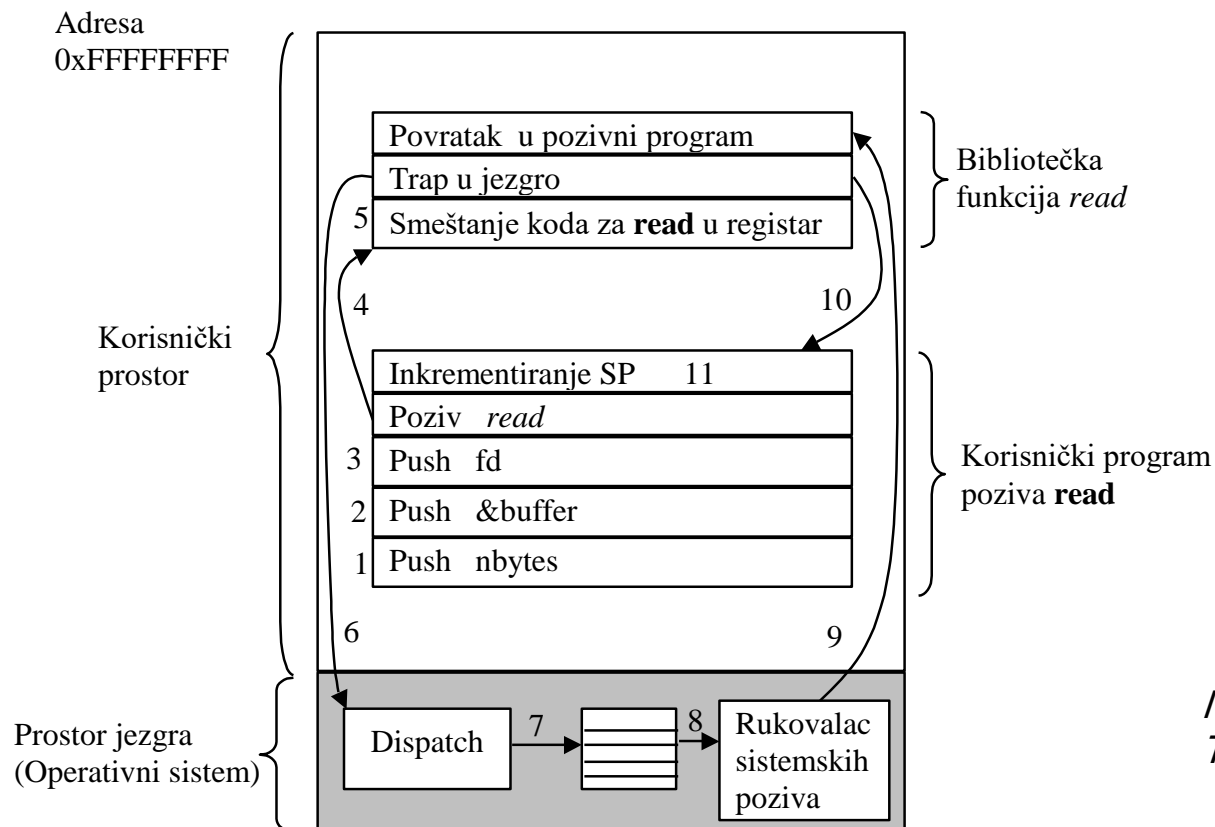
- ✿ **Sistemske pozivi** (*system calls*) obezbeđuju interfejs između aplikativnih/sistemskih programa i operativnog sistema
- ✿ Omogućuju pristup funkcijama operativnog sistema od strane korisničkih programa
  - ✦ Unix/Linux – POSIX.1 (POSIX.1-2017) - IEEE Std 1003.1-2017
  - ✦ Windows - Windows API (*Application Programming Interface*)
- ✿ Sistemske poziv se obavlja u okviru korisničkog programa pozivom funkcije iz standardne biblioteke za odgovarajući programski jezik (API).
- ✿ U okviru ove funkcije se argumenti smeštaju na **stek**, i poziva **trap** instrukcija čiji je argument kôd sistemskog poziva.
- ✿ **Trap** instrukcija izaziva softverski prekid, OS čuva stanje prekinutog procesa, prelazi u mod kernela i poziva funkciju kernela (rutinu, *system call handler*) koja implementira sistemske poziv
- ✿ Postoje sistemske pozivi za upravljanje procesima, memorijom, datotekama, U/I uređajima, mrežnom komunikacijom, za dobijanje informacija o radu sistema, upravljanje GUI (Windows), itd.



# Izvršenje sistemskog poziva *read*

```
#include <unistd.h>
```

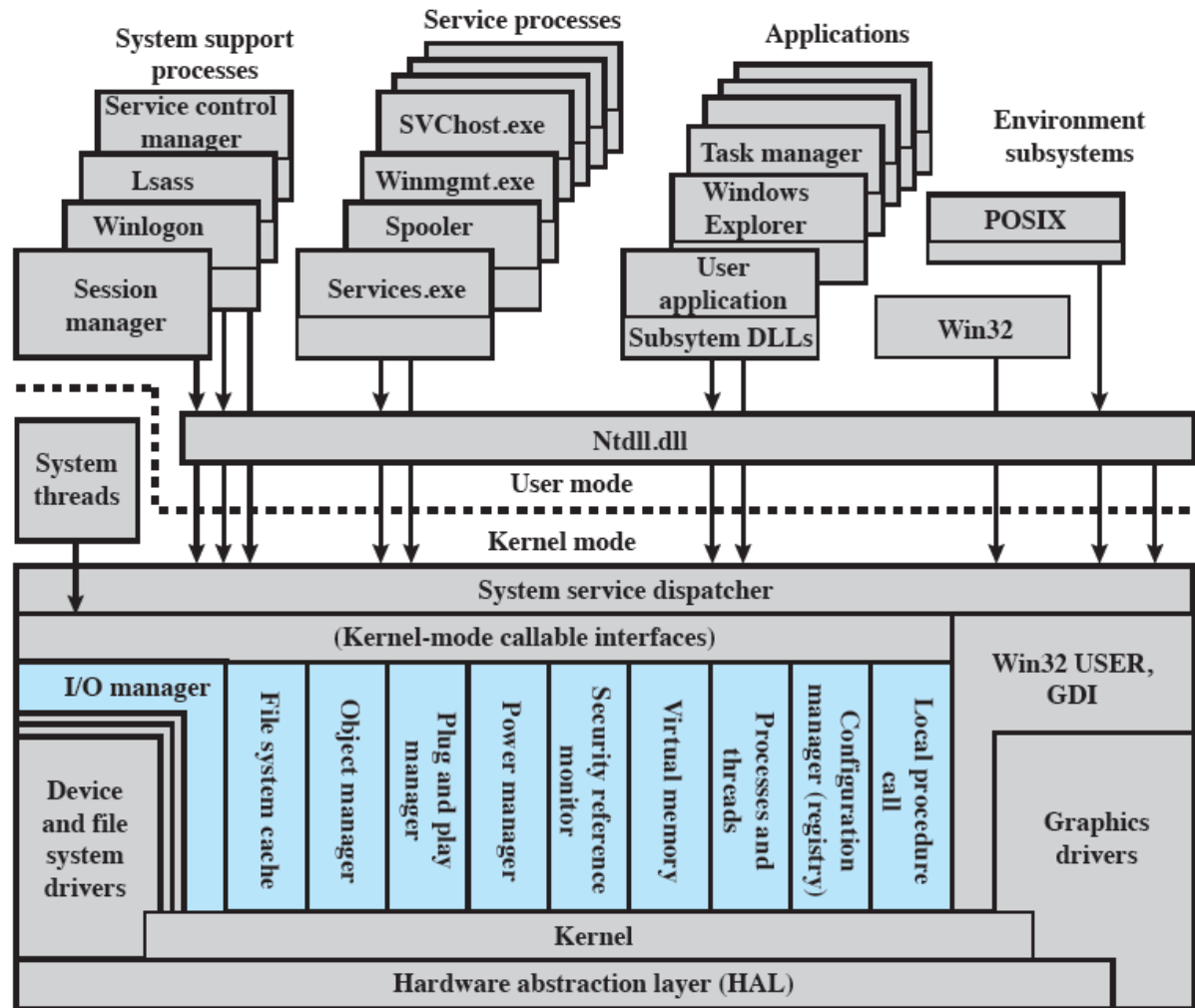
```
ssize_t read(int fd, void *buffer, size_t nbytes);
```



*Modern Operating Systems,  
Tanenbaum, 2014*



# Microsoft Windows



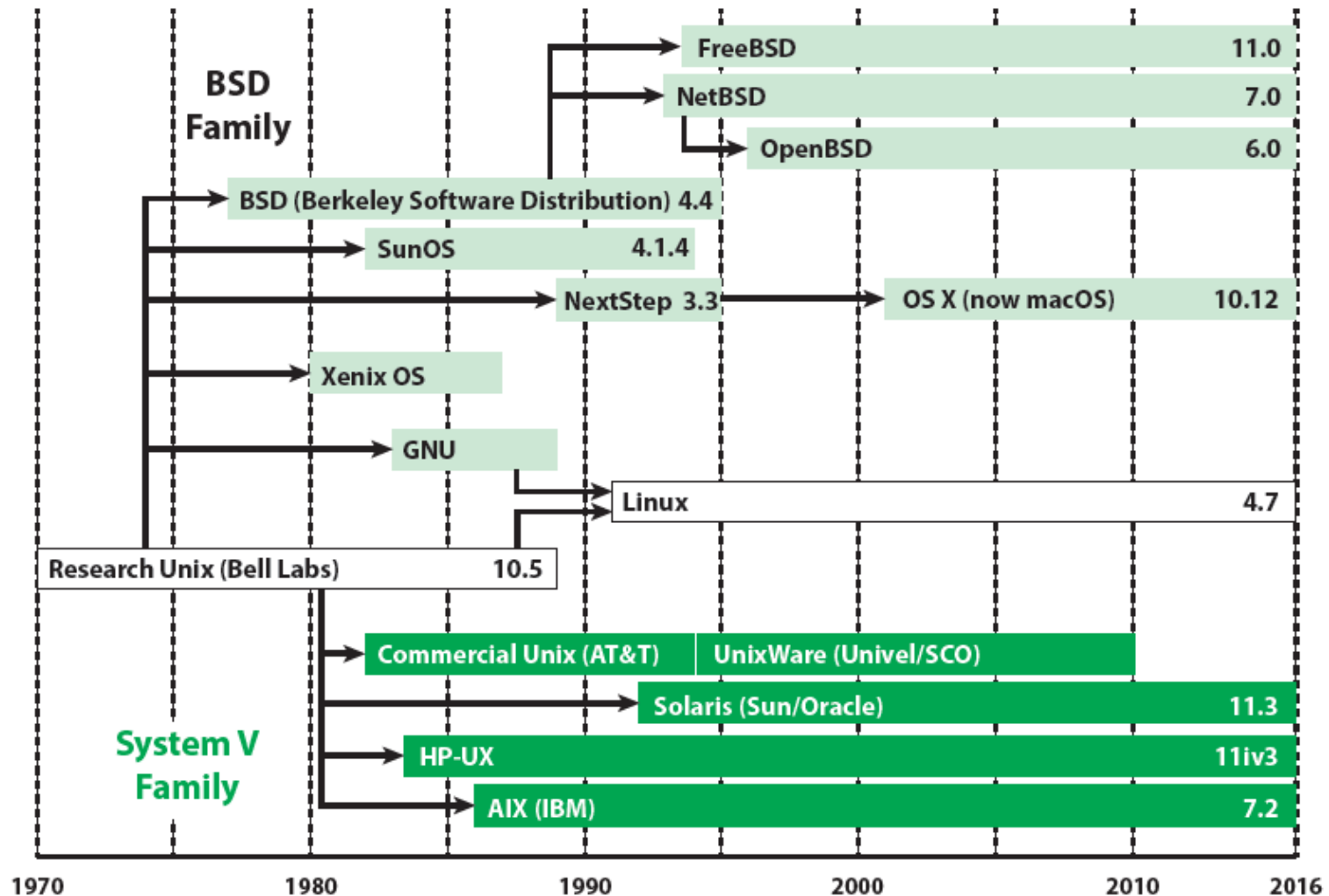
Lsass = local security authentication server  
POSIX = portable operating system interface  
GDI = graphics device interface  
DLL = dynamic link libraries

Colored area indicates Executive

## Uvod i pregled operativnih sistema

### Operativni sistemi

# UNIX familija





# UNIX

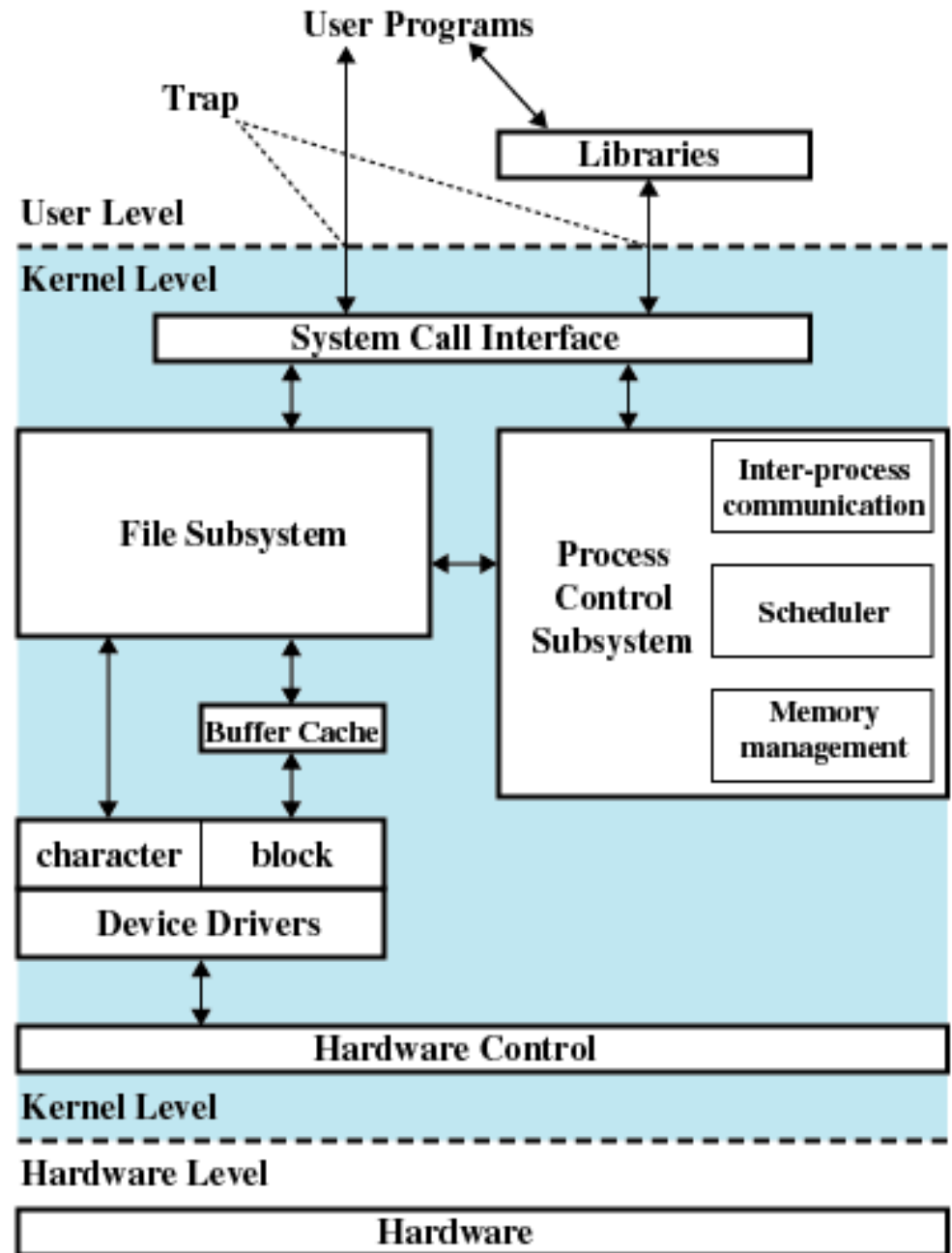
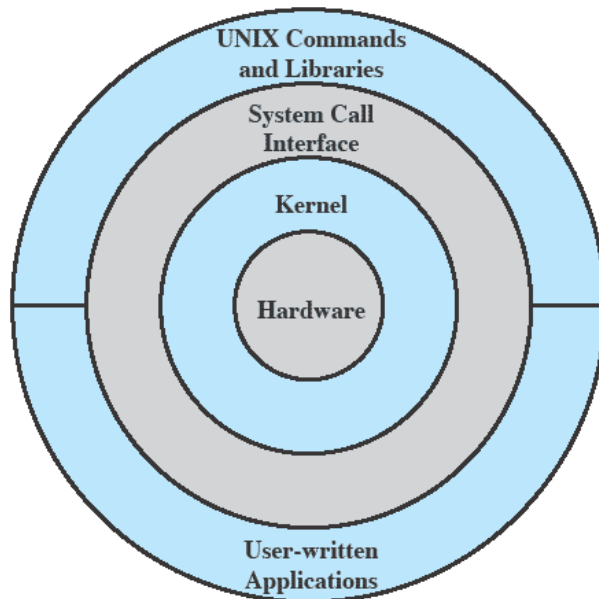


## Savremeni Unix

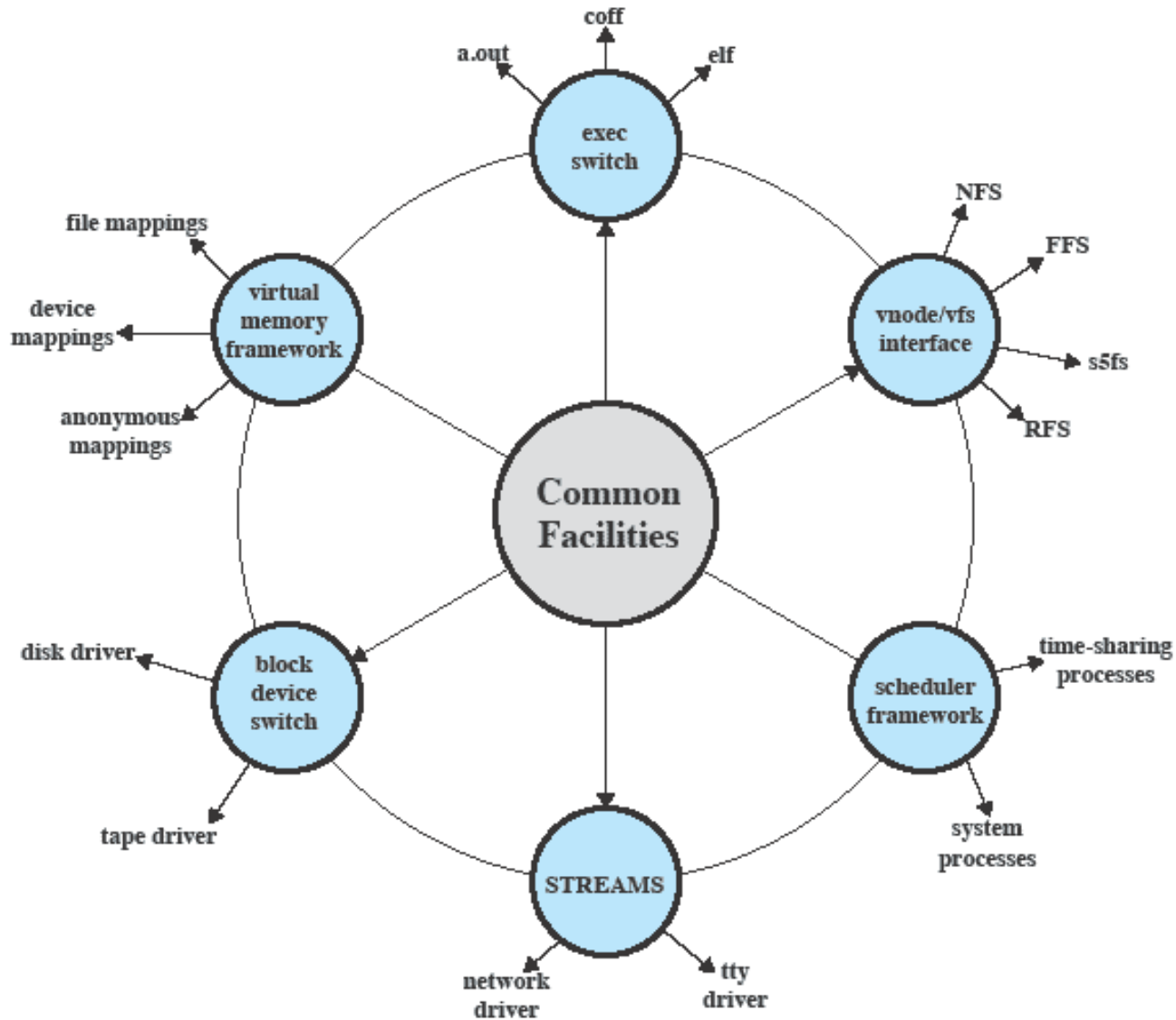
- ❖ System V R4 (SVR4)
- ❖ Solaris 10
- ❖ 4.4 BSD & FreeBSD



## Tradicionalni UNIX kernel



# Savremeni UNIX kernel



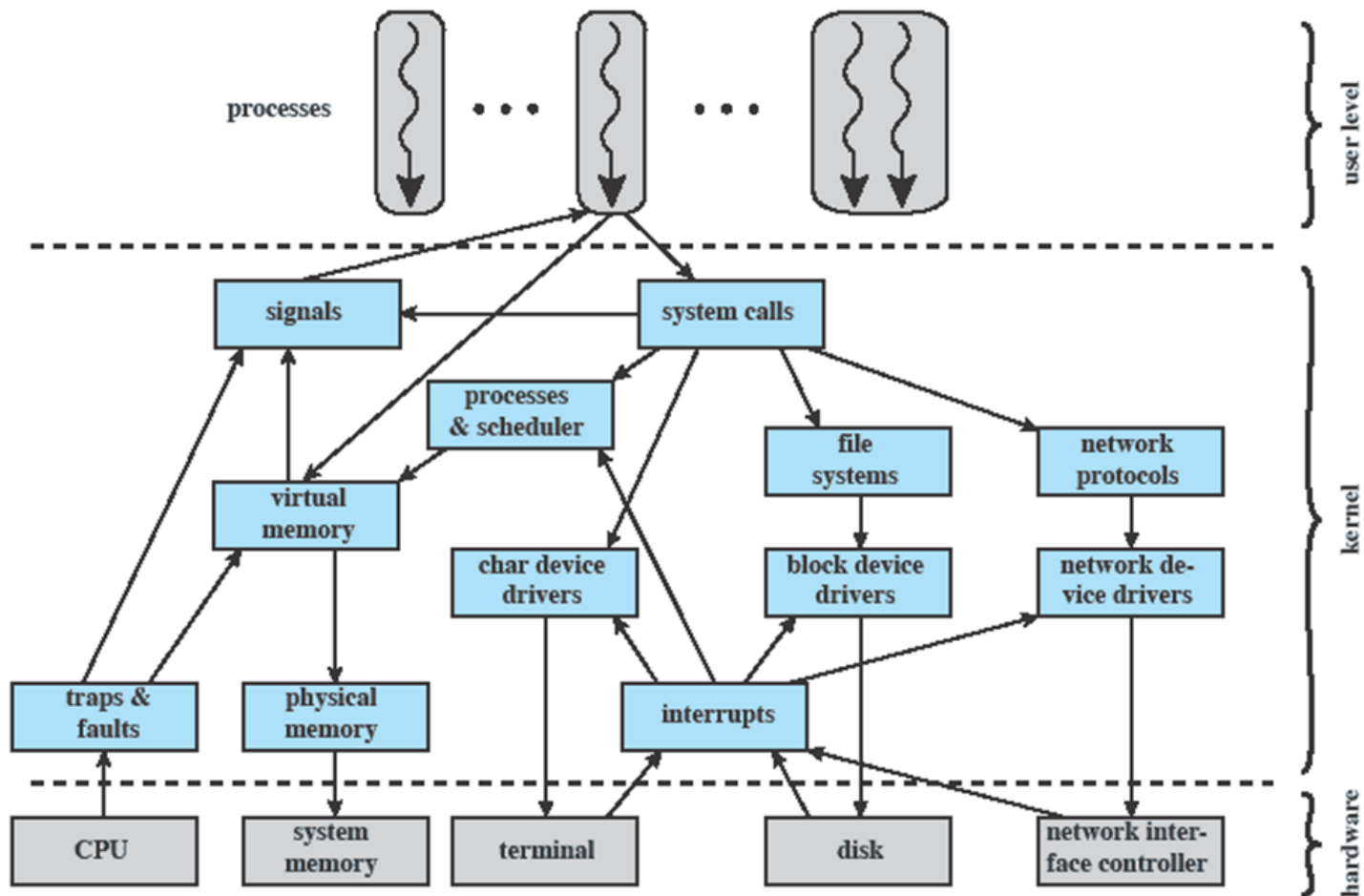
Uvod i pregled operativnih sistema

Operativni sistemi



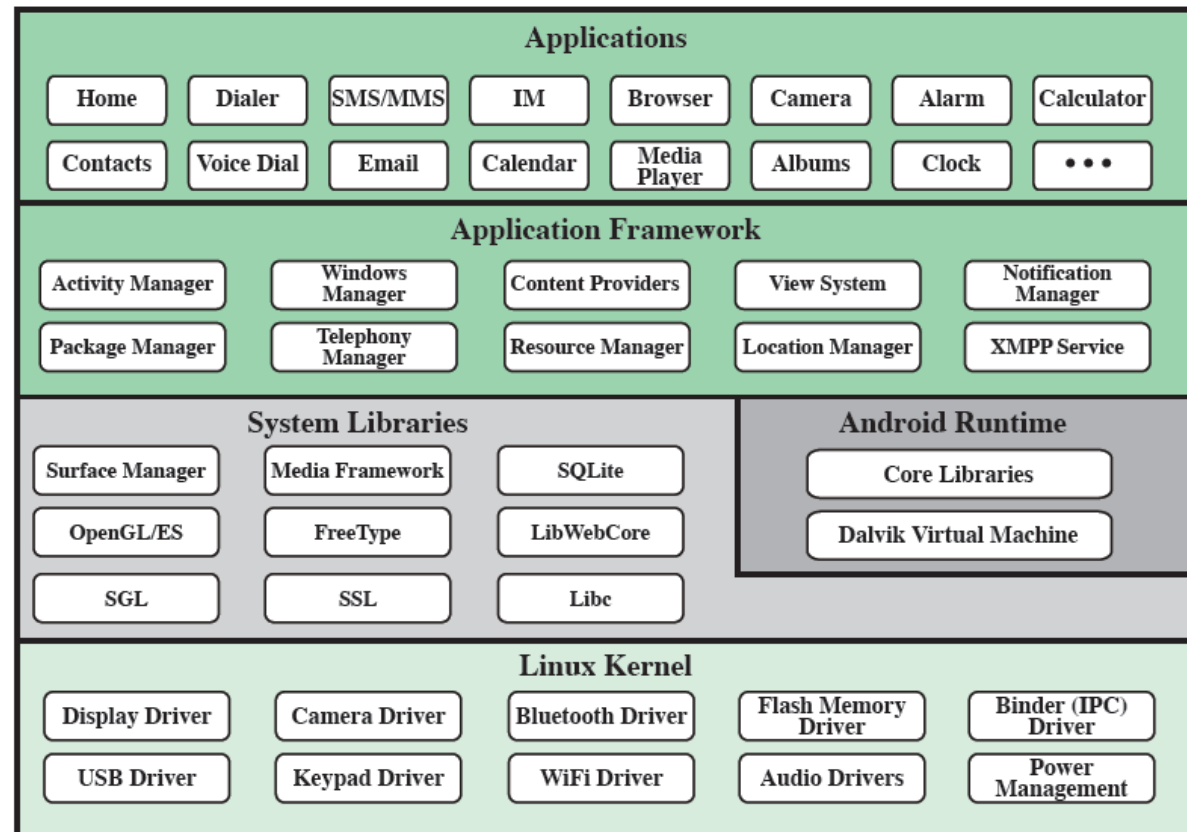
# Linux

## ⚙ Komponente Linux kernela



# Android

## Android arhitektura



Implementation:

Applications, Application Framework: Java

System Libraries, Android Runtime: C and C++

Linux Kernel: C

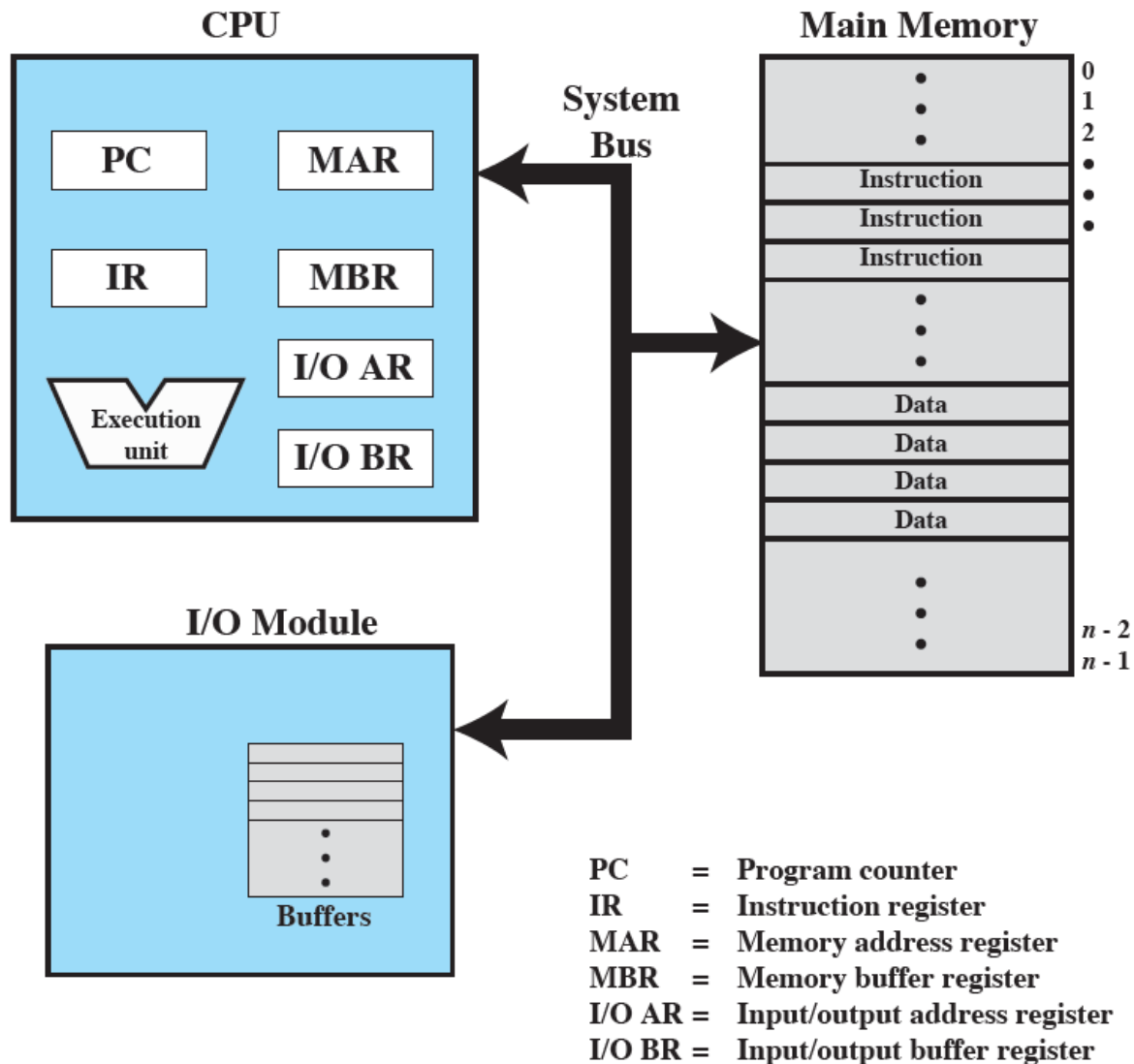
**Uvod i pregled operativnih sistema**  
Operativni sistemi



# **PREGLED RAČUNARSKOG SISTEMA (Poglavlje 1)**

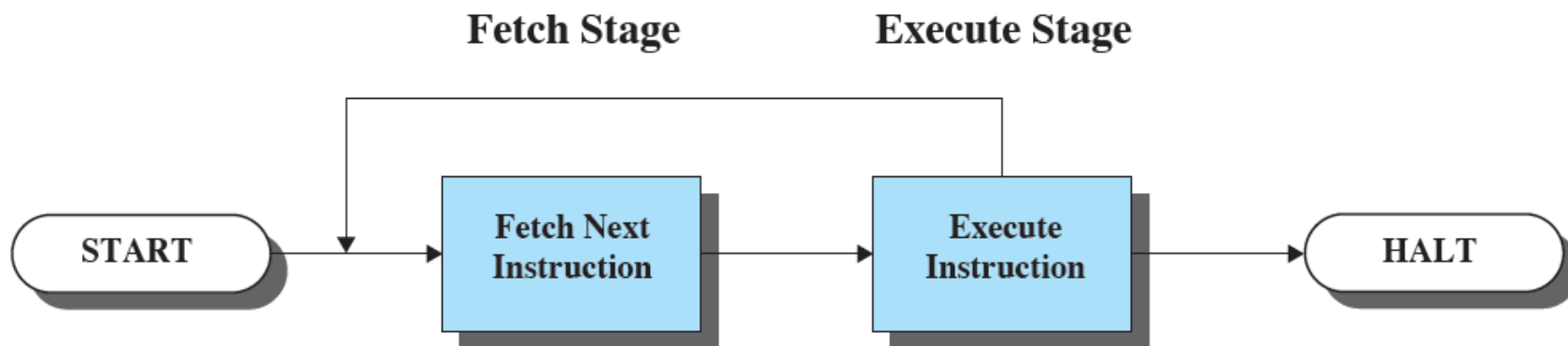
**Uvod i pregled operativnih sistema**  
Operativni sistemi

# Komponente računara

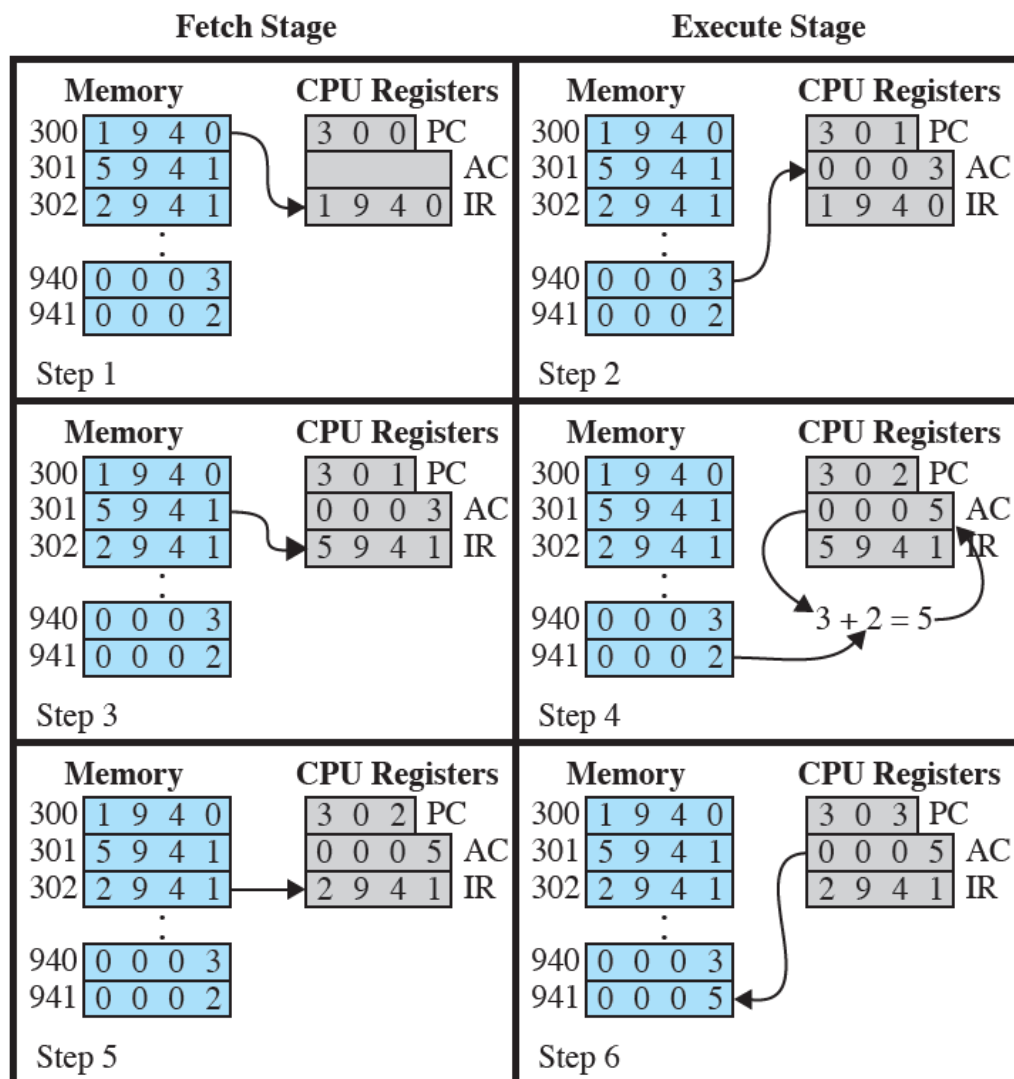


# Ciklus izvršenje instrukcija

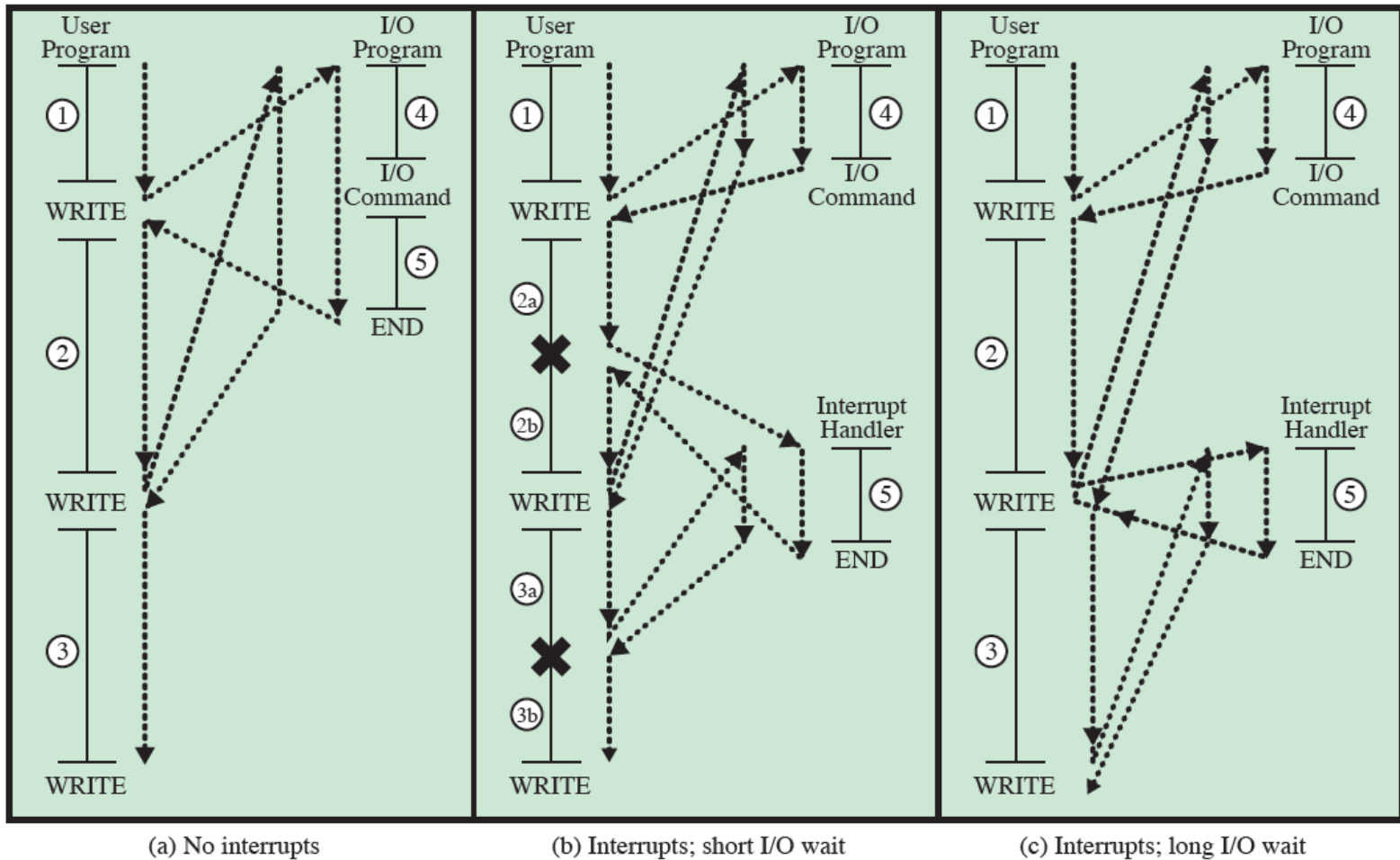
Fetch – Decode – Execute



# Primer izvršenja programa

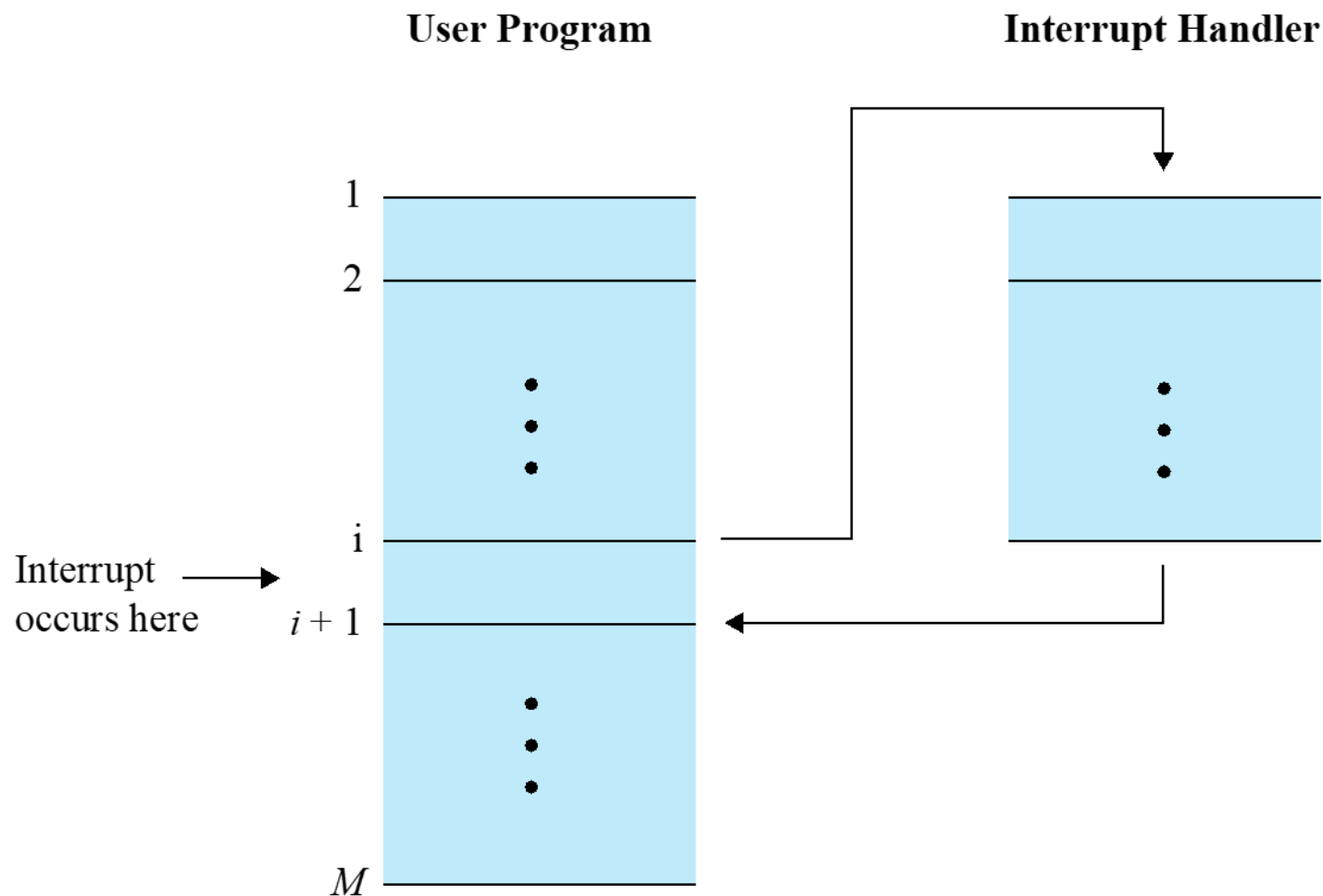


# Tok izvršenja programa - sa i bez prekida



✖ = interrupt occurs during course of execution of user program

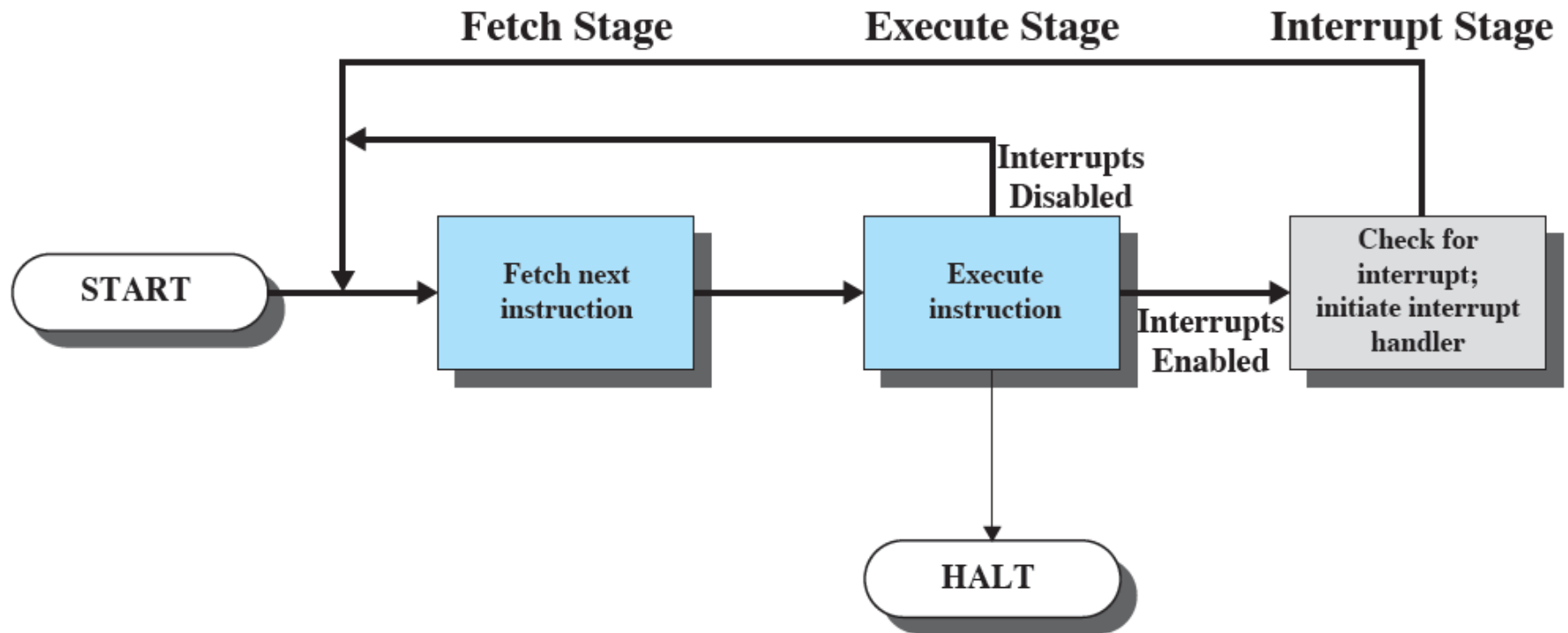
# Prenos upravljanja usled prekida



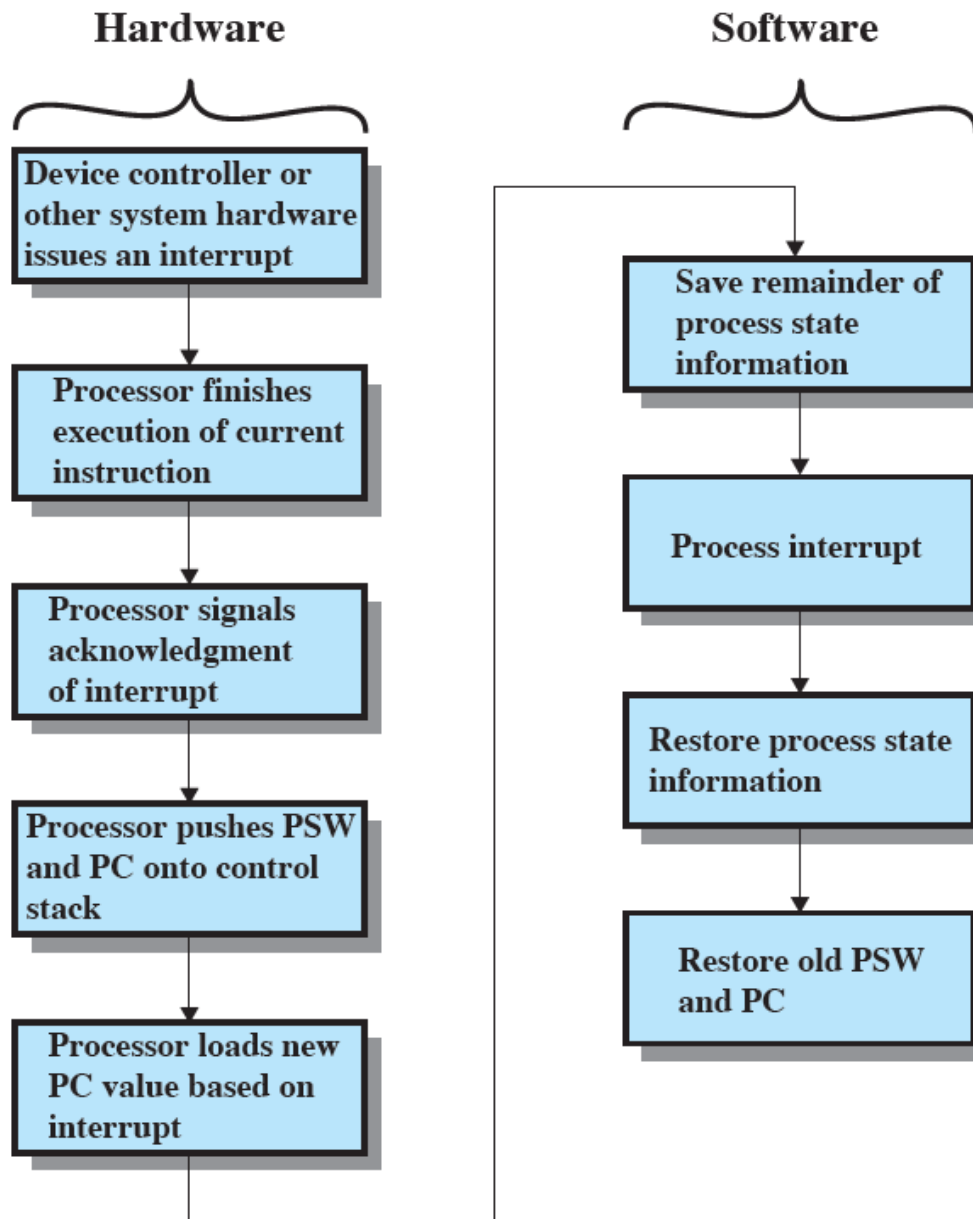




# Ciklus izvršenja instrukcija - sa prekidom

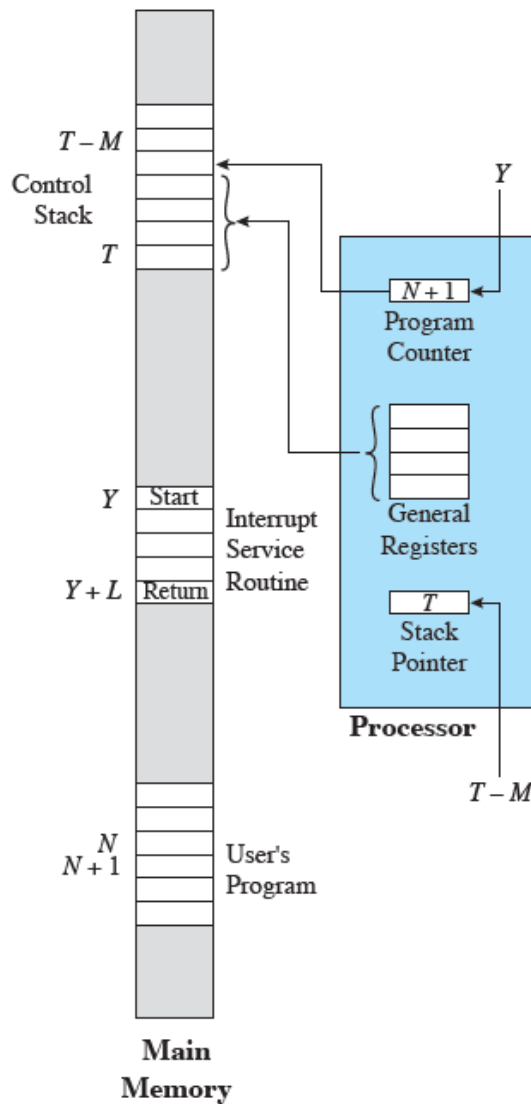


# Jednostavna obrada prekida

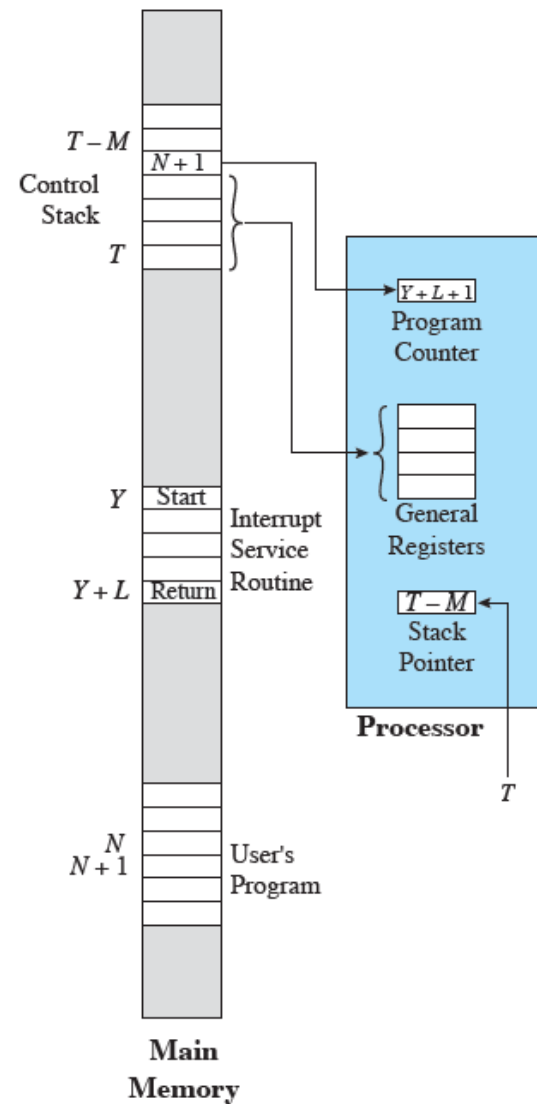


# Prekidi

➊ Promene u memoriji i registrima tokom prekida



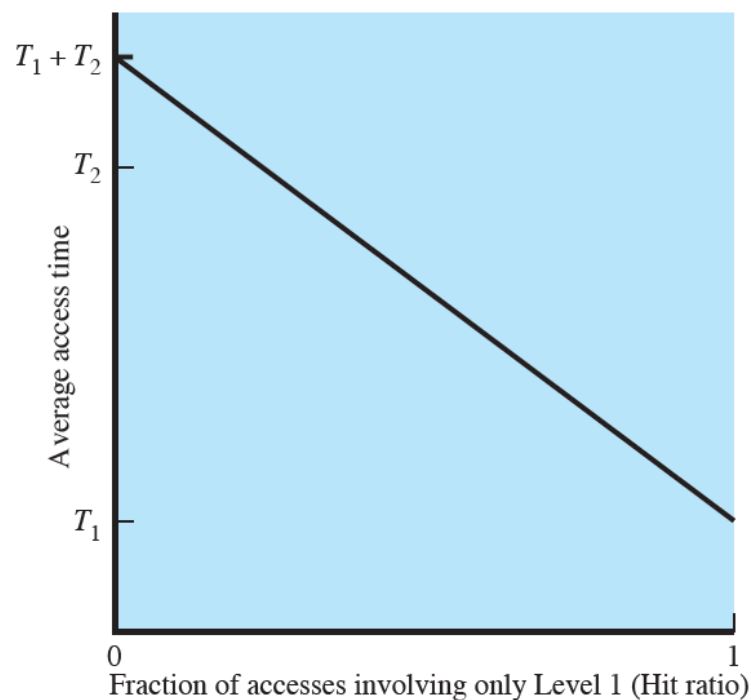
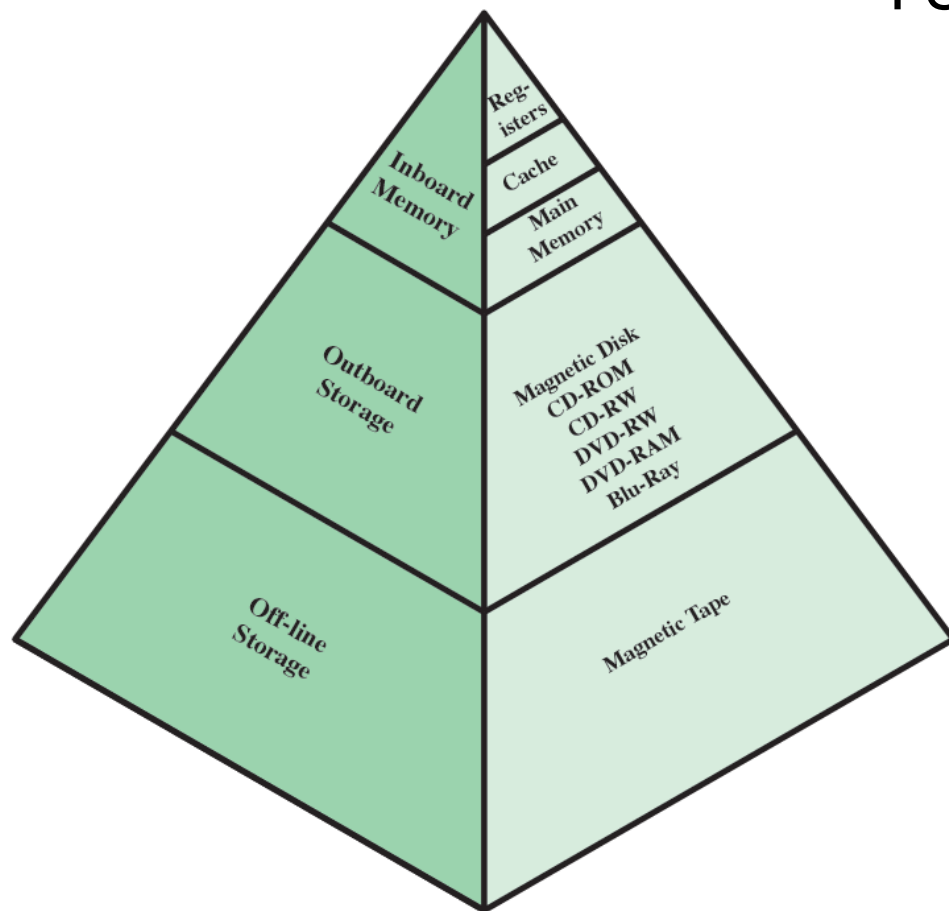
(a) Interrupt occurs after instruction at location  $N$



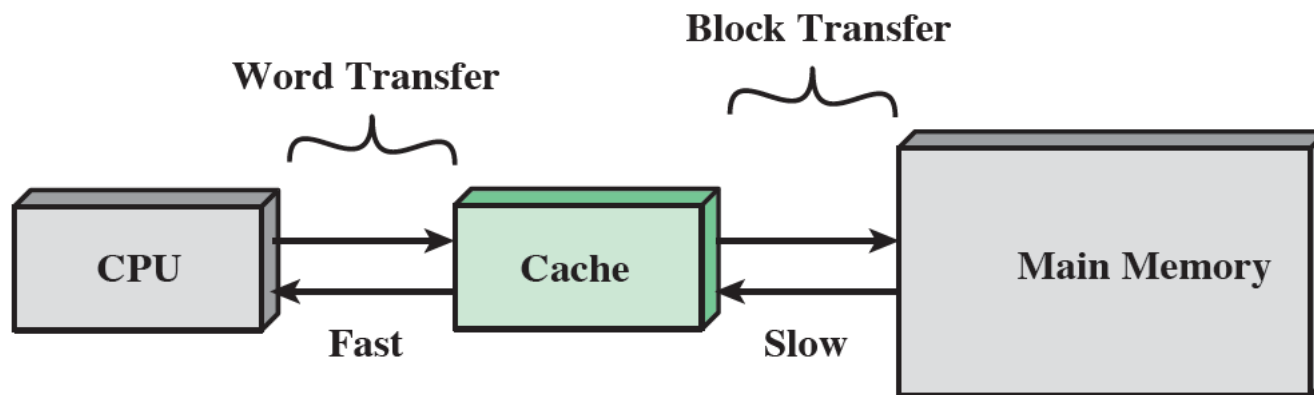
(b) Return from interrupt

# Memorijska hijerarhija

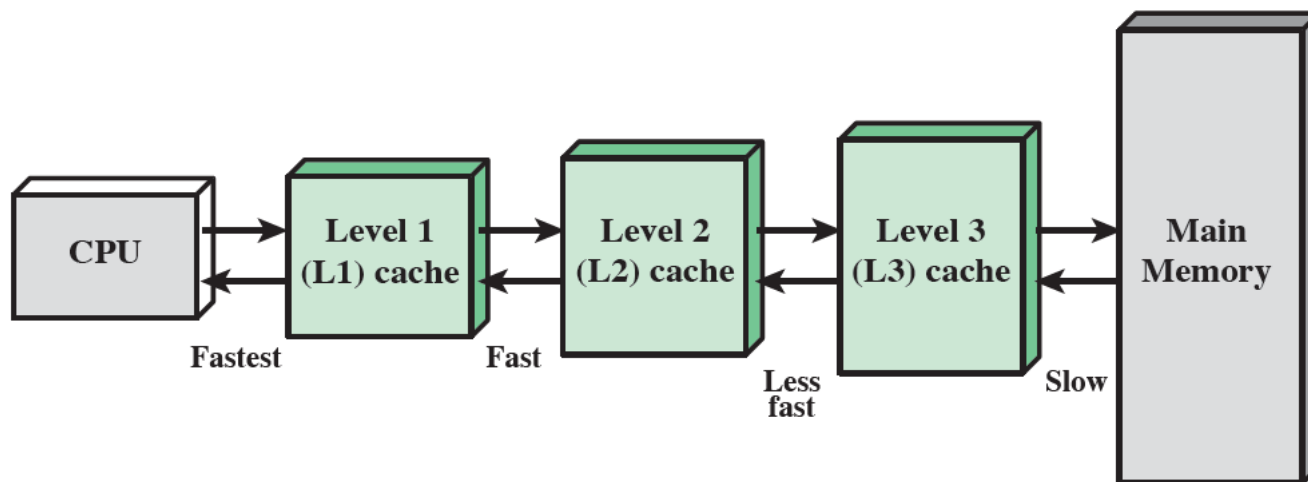
## Performanse memorije u 2 nivoa



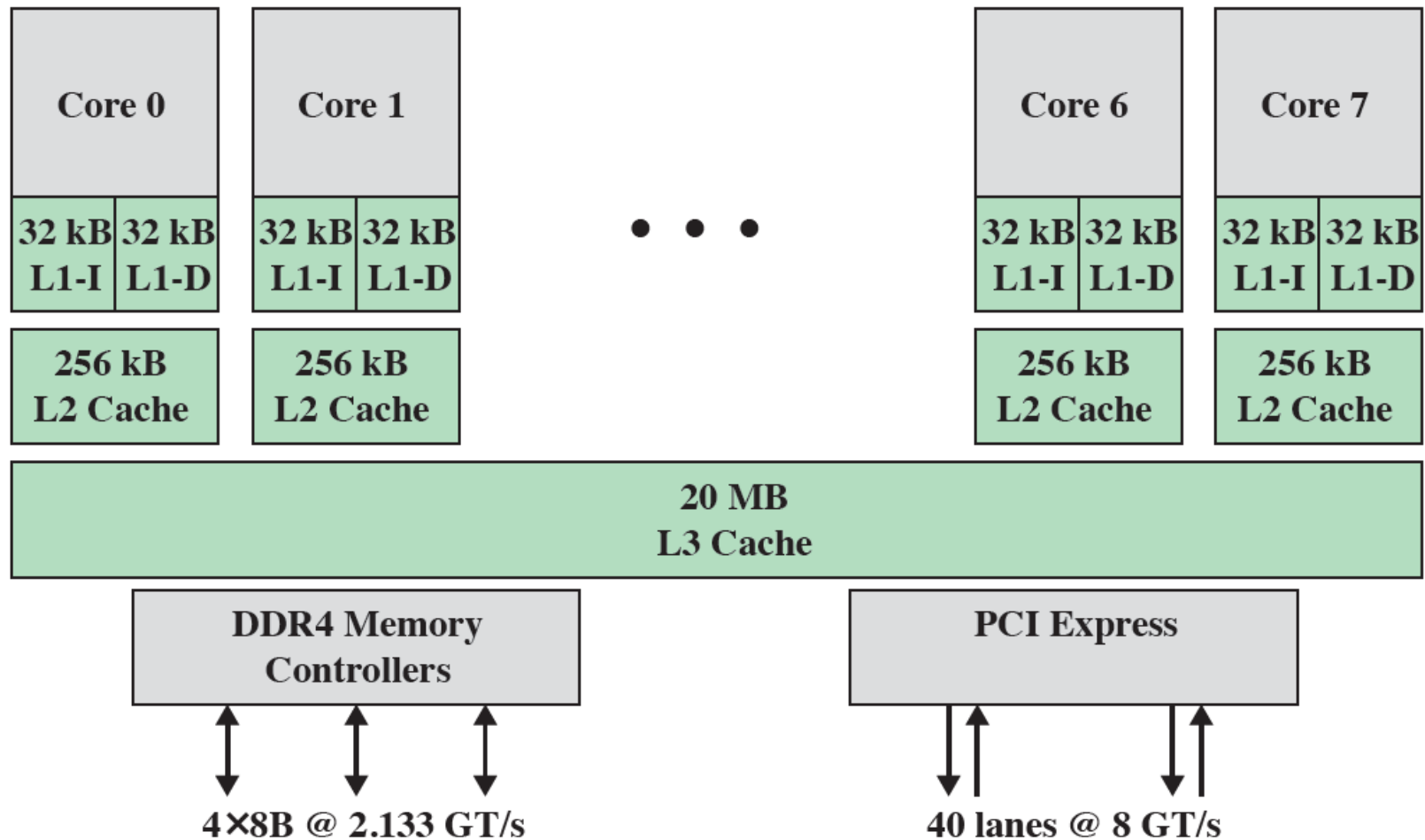
# Keš memorija



(a) Single cache



# Intel Core i7-5960X – blok dijagram





# Domaći zadatak

## ❁ Praktikum iz Sistemskog softvera

- I. UNIX/Linux - Poglavlje 1. Operativni sistem UNIX/Linux
- II. Windows - Poglavlje 1. Operativni sistem Windows 2000

## ❁ Pročitati poglavlja:

- 1. Pregled računarskog sistema
- 2.7 Pregled Microsoft-ovog Windowsa
- 2.8 Tradicionalni sistemi Unix
- 2.9 Savremeni sistemi Unix
- 2.10 Linux

## ❁ *Student resources*

- ❁ <http://williamstallings.com/OperatingSystems/OS9e-Student/>

## ❁ *Animations*

- <http://williamstallings.com/OS/Animation/Animations.html>
- <https://www.uttyler.edu/files/cosc3355/animations/>