



Operativni sistemi

- Uvod i pregled operativnih sistema -

Računarstvo i informatika

**Katedra za računarstvo
Elektronski fakultet u Nišu**



Softver

☀ Softver

- ☐ Sistemski softver
- ☐ Aplikativni softver

☀ Sistemski softver

- ☐ Sva programska sredstva neophodna za razvoj i izvršenje korisničkih i sistemskih programa na računaru i upravljanje resursima računarskog sistema
- ☐ Namenjen je svim korisnicima računarskog sistema

☀ Aplikativni softver

- ☐ Sva programska sredstva namenjena rešavanju konkretnih problema korisnika u određenom domenu



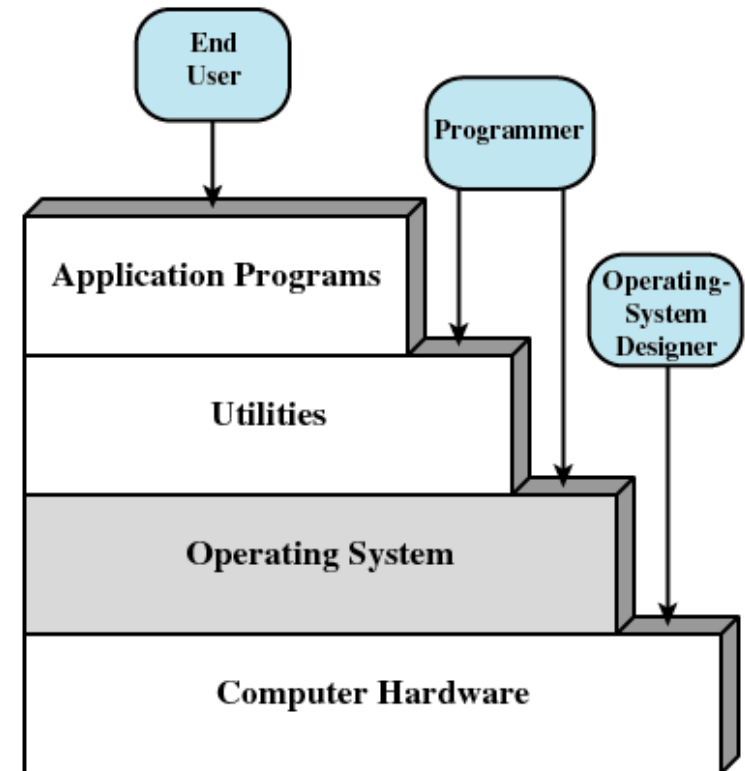
Operativni sistem (OS)

- ✿ OS predstavlja organizovanu kolekciju programa čije su osnovne funkcije i ciljevi:
 - ✦ **Pogodnost** - Da omogući lako i pogodno korišćenje računara
 - ✦ **Efikasnost** - Da obezbedi efikasno korišćenje i upravljanje resursima računara
 - ✦ **Mogućnost razvoja** - Da obezbedi osnovu za efikasno evoluiranje i razvoj novih funkcija sistema
- ✿ OS je interfejs između korisnika i HW računarskog sistema
 - ✦ Programi obično zahtevaju servise OS-a pomoću **sistemskih poziva** (*system call*)
 - ✦ Korisnici mogu direktno interagovati sa OS-om pomoću komandi komandnog jezika ili preko elemenata grafičkog korisničkog intrefejsa (GUI)



OS kao interfejs između korisnika i računara

- ✿ **Korisnik** vidi računarski sistem kao skup aplikacija
- ✿ **Programer** razvija aplikacije korišćenjem sistemskog softvera (pomoćnih programa i operativnog sistema). OS sakriva detalje hardvera od programera i obezbeđuje mu jednostavan i prikladan interfejs za korišćenje računarskog sistema
- ✿ OS omogućava aplikacionim programima pristup i korišćenje funkcija i servisa OS u pristupu hardverskim resursima





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 resursima
- ❖ Detekcija grešaka i odgovor na greške
- ❖ Evidencija korišćenja resursa sistema i nadgledanje performansi



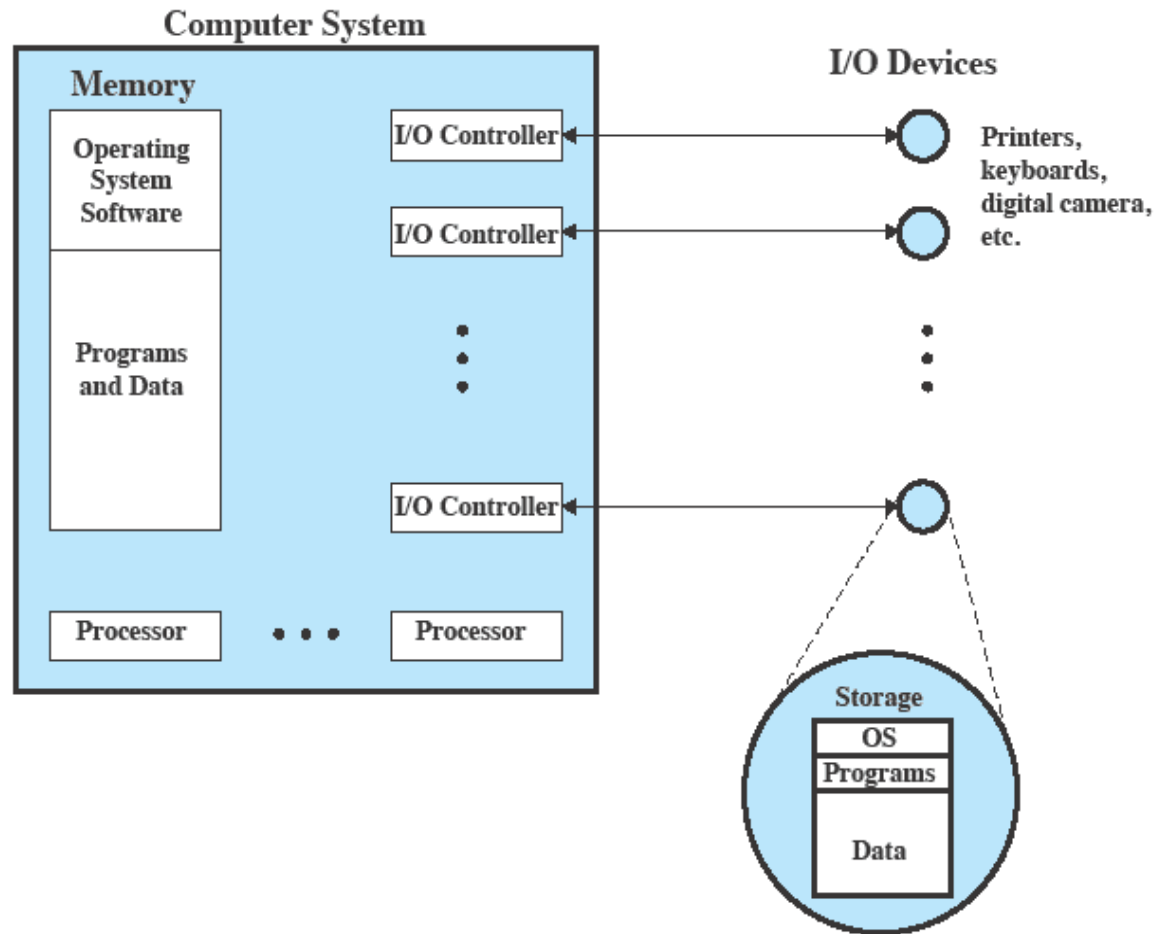
OS kao upravljač resursa

- OS upravlja resursima računarskog sistema
- **Resursi** 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
 - Apstrakcija resursa
 - OS prikriva detalje o tome kako HW radi tako što koristi **apstraktni model resursa**
 - Slični resursi imaju zajednički apstraktni model (npr. hard disk i USB flash)
 - Ovi modeli su specifični za OS
 - Npr. procesi su apstrakcija rada procesora, datoteke su apstrakcije sekundarnih memorijskih jedinica
 - Raspodela i zajedničko korišćenje resursa
 - Dodela i oslobađanje resursa
 - Evidentiranje korišćenja resursa
 - Obračun korišćenja resursa
 - Rešavanje konfliktnih zahteva za resursima od strane različitih programa i korisnika



OS kao upravljač resursa

- Deo OS se nalazi u glavnoj memoriji
 - Kernel** (jezgro) - sadrži najčešće korišćene funkcije u okviru OS
 - Delovi OS koji su trenutno u upotrebi
- Ostatak glavne memorije sadrži korisničke programe i podatke





Zajedničko korišćenje resursa

- ✿ Više programa koji se konkurentno izvršavaju mogu **deliti (zajednički koristiti) resurse**
- ✿ Postoje dva načina deljenja (multipleksiranja) resursa:
 - ✦ **U vremenu** – različiti programi ili korisnici koriste resurs naizmenično
 - Na primer, na računaru sa jednim CPU i više programa, operativni sistem dodeljuje CPU jednom programu, zatim drugom, itd., eventualno opet prvom (drugi primer: štampač)
 - Određivanje vremenske raspodele resursa, kome i koliko vremena će resurs biti dodeljen, je zadatak OS-a
 - ✦ **U prostoru** – svaki program ili korisnik dobija jedan deo resursa
 - Na primer, glavna memorija je obično podeljena na više trenutno aktivnih programa, a OS vodi računa o prostornoj raspodeli i zaštiti od pristupa tuđem delu memorijskog prostora
- ✿ Od prirode resursa zavisi način deljenja – u vremenu ili prostoru



Jednostavna evolucija OS

- ✿ Operativni sistemi moraju biti sposobni da evoluiraju tokom vremena iz sledećih razloga:
 - ✦ Unapređenje 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

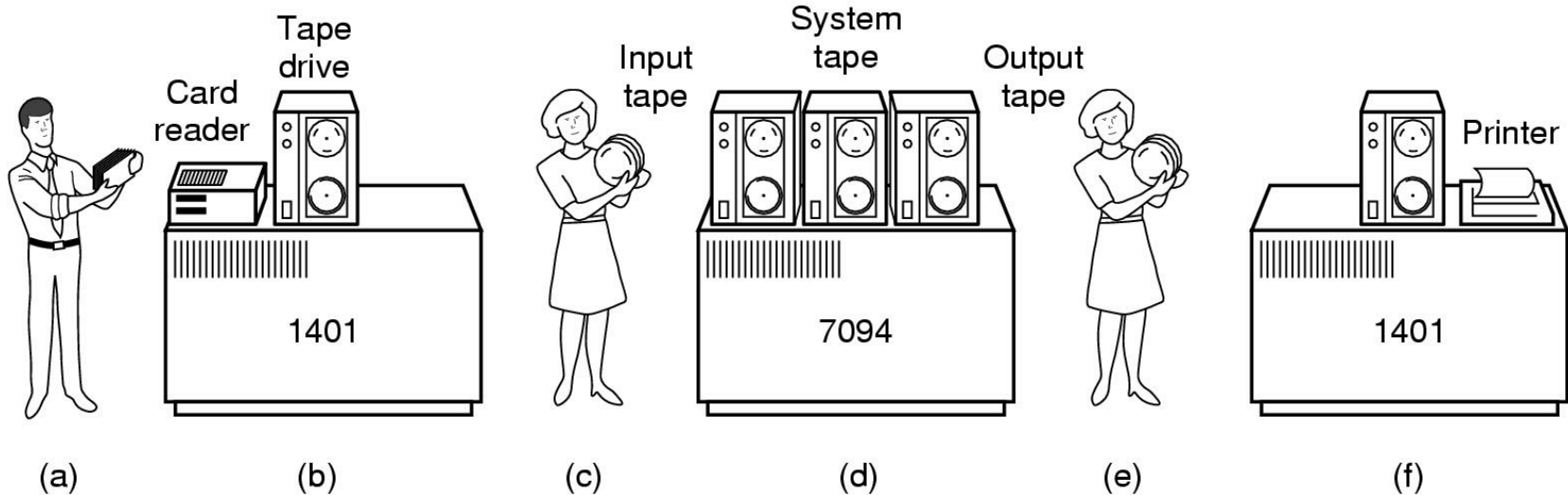


Istorija operativnih sistema

- ✿ Serijska obrada (1945 – 1955)
 - ✦ Vakumske cevi, bušene kartice, mašinski jezik
 - ✦ **Nema OS-a**, programeri su pristupali direktno hardveru
- ✿ Jednostavni paketni sistemi (*batch systems*) (1955 – 1965)
 - ✦ Tranzistori, mainframe računari, asemblerski jezik, FORTRAN, COBOL
 - ✦ **Monitor** – jednostavan OS (IBSYS – IBM OS za 7090/7094 računare)
- ✿ Multiprogramski *batch* sistemi
- ✿ Multiprogramski sistemi sa podelom 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), ugrađeni i sveprisutni računari



Sistemi paketne obrade (batch sistemi)



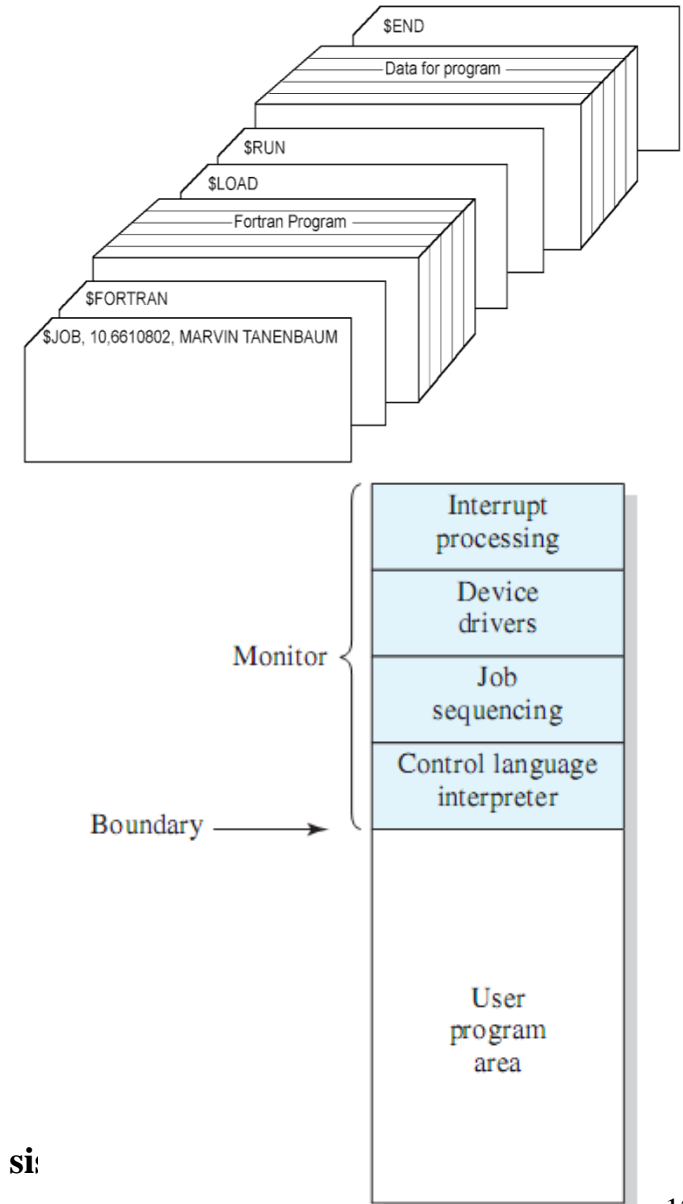
✿ Stari mainframe računari

- ✦ 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
- ✦ 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)
- Instrukcije se monitoru zadaju preko *Job Control Language* (JCL)
- Monitor** je stalno smešten u glavnoj memoriji i izvršava se u **kernel** modu
- Čita sa ulaznog uređaja jedan po jedan posao, smešta instrukcije i podatke u korisnički deo memorije i startuje izvršenje posla u korisničkom (**user**) modu.
- Po završetku, monitor učitava i izvršava sledeći posao.





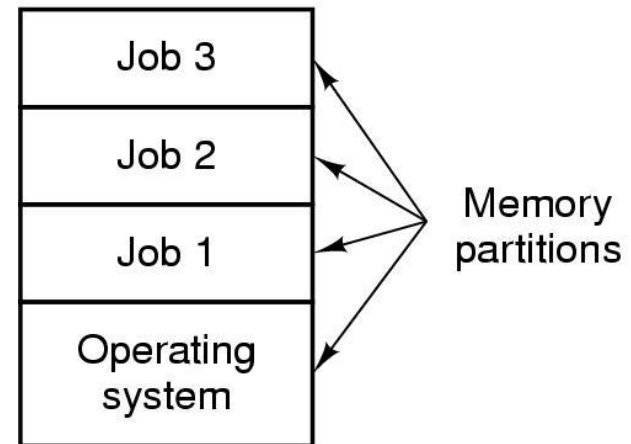
Monitor – operativni sistem

- ✿ Monitor (operativni sistem 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
 - ✦ Privilegiovane 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 paketni sistemi

- ❁ **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)
- ❁ U/I prekidi i DMA
- ❁ Upravljanje memorijom
- ❁ Operativni sistemi:
 - ❑ OS/360
 - ❑ MULTICS
 - ❑ UNIX (System V, BSD)



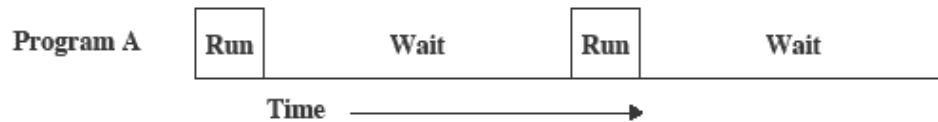
Uvod i pregled operativnih sistema

Operativni sistemi



Multiprogramiranje

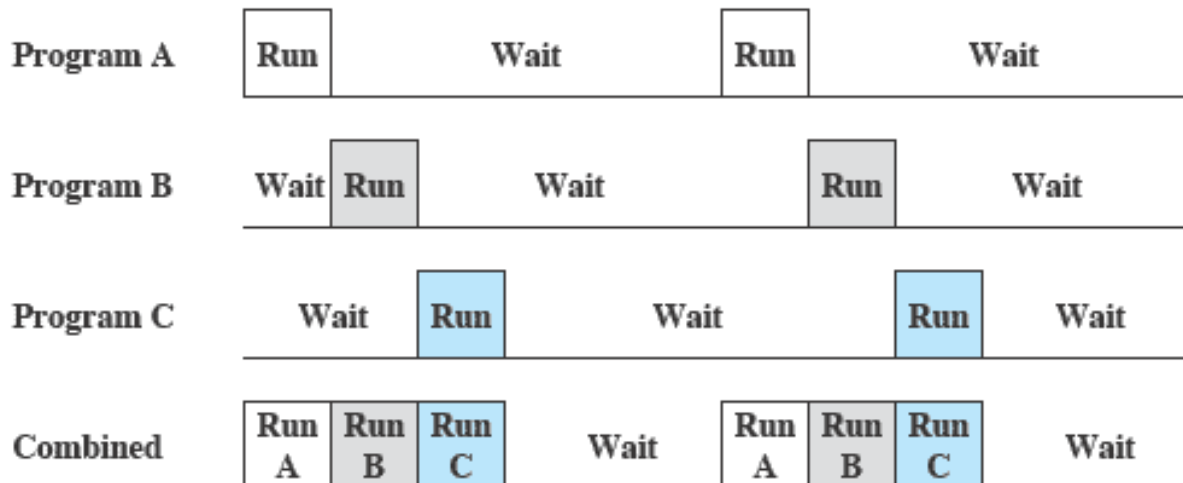
- Uniprogramiranje - CPU mora da čeka dok se ne završi U/I instrukcija



Read one record from file	15 μ s
Execute 100 instructions	1 μ s
Write one record to file	15 μ s
TOTAL	31 μ s

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

- Multiprogramiranje sa tri aktivirana programa





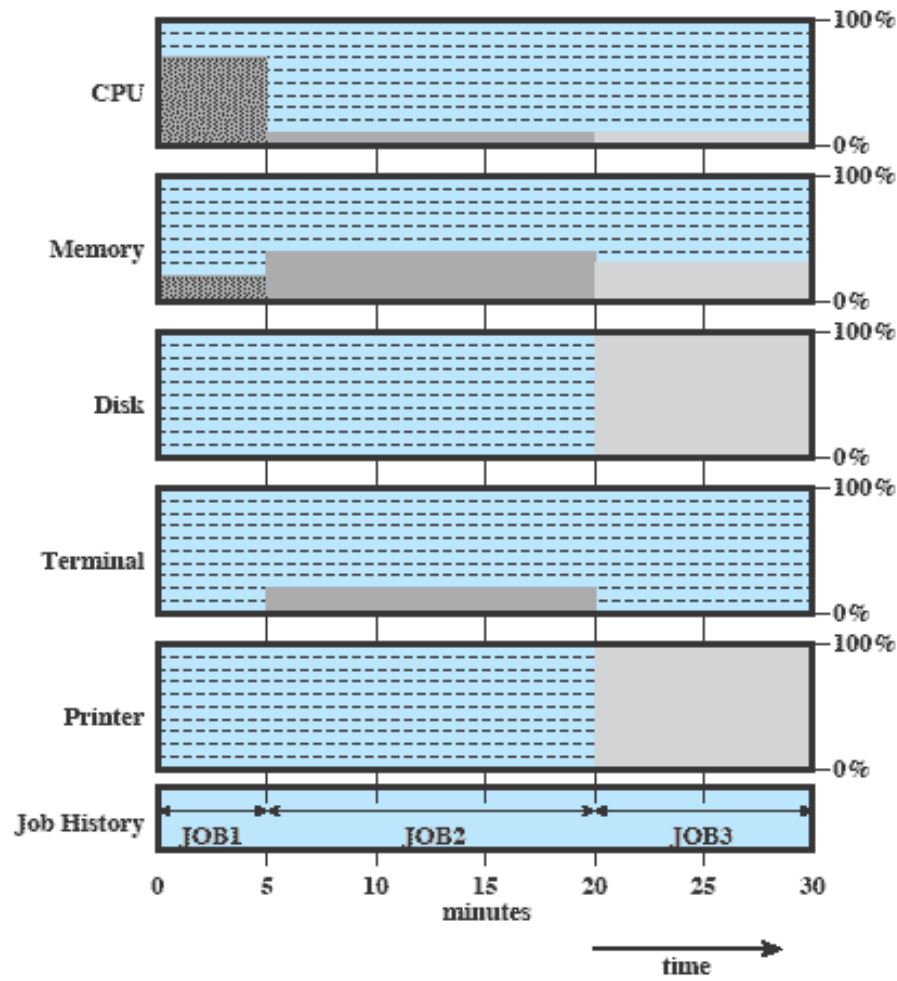
Primer multiprogramiranja

Table 2.1 Sample Program Execution Attributes

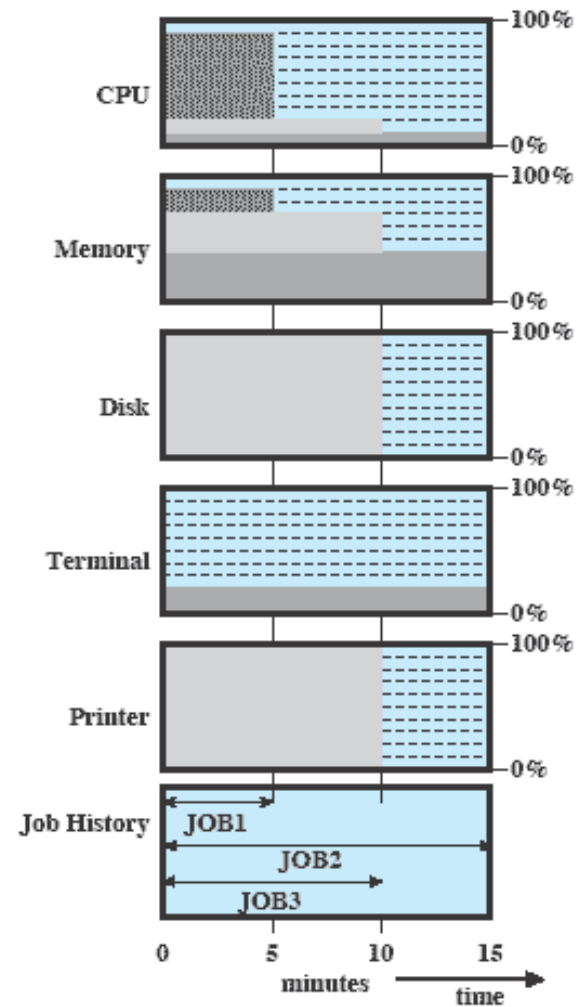
	JOB1	JOB2	JOB3
Type of job	Heavy compute	Heavy I/O	Heavy I/O
Duration	5 min	15 min	10 min
Memory required	50 M	100 M	75 M
Need disk?	No	No	Yes
Need terminal?	No	Yes	No
Need printer?	No	No	Yes



Histogram iskorišćenosti resursa



(a) Uniprogramming



(b) Multiprogramming

Uvod i pregled operativnih sistema

Operativni sistemi



Time-sharing sistemi

- ❖ **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. na MIT za IBM 709, a kasnije prenet na IBM 7094
 - ❖ Računar sa glavnom memorijom od 32000 36-bitnih reči, pri čemu monitor koristi 5000 reči



Primer rada CTSS

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

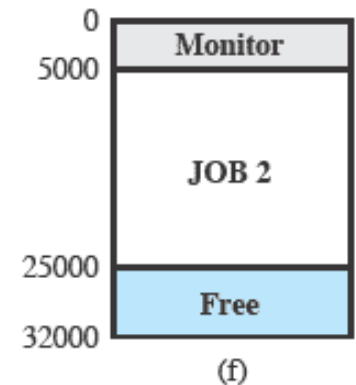
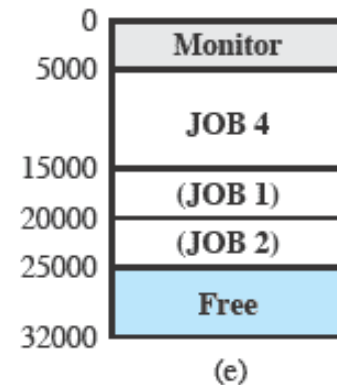
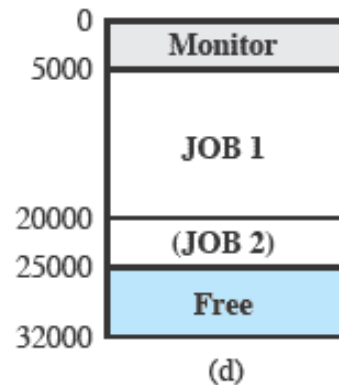
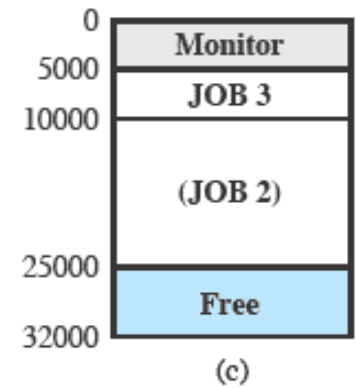
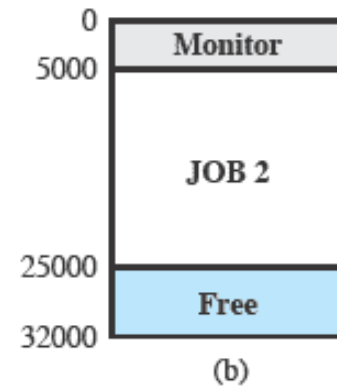
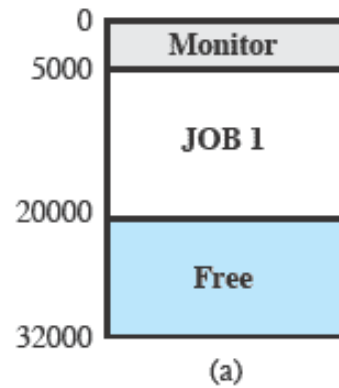


Figure 2.7 CTSS Operation



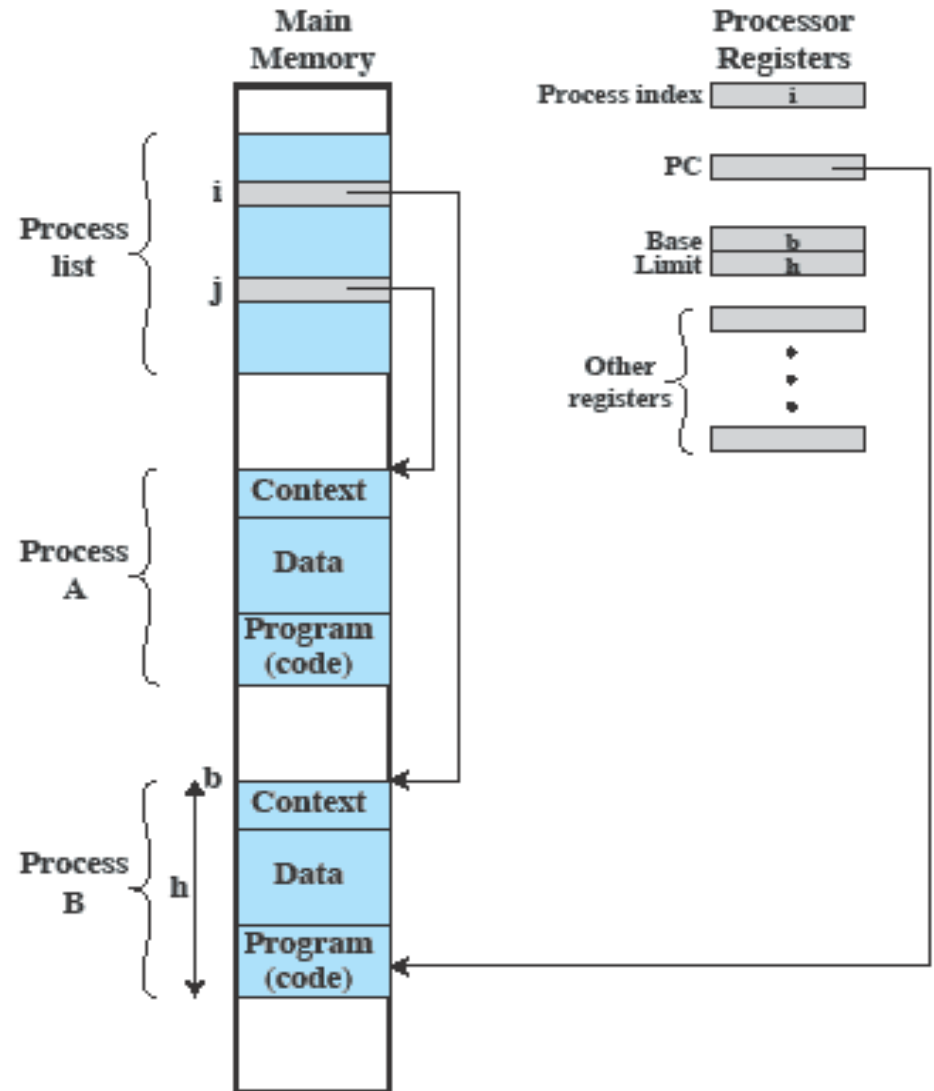
Glavni koncepti i dostignuća OS

- ❖ Procesi
- ❖ Upravljanje memorijom
- ❖ Zaštita i sigurnost 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
 - izvršni kontekst** (stanje procesa)
- Tipična implementacija procesa na slici



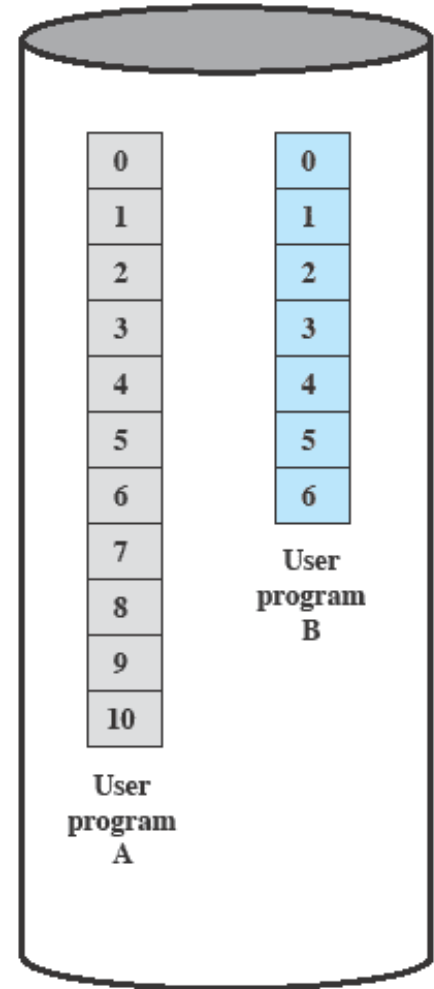


Upravljanje memorijom

- Osnovni zahtevi
 - ❑ Izolacija procesa
 - ❑ Automatska alokacija i upravljanje
 - ❑ Podrška za modularno programiranje
 - ❑ Zaštita i kontrola pristupa
 - ❑ Korišćenje trajne memorije
- Koncept virtuelne memorije (straničenje - *paging*) 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	

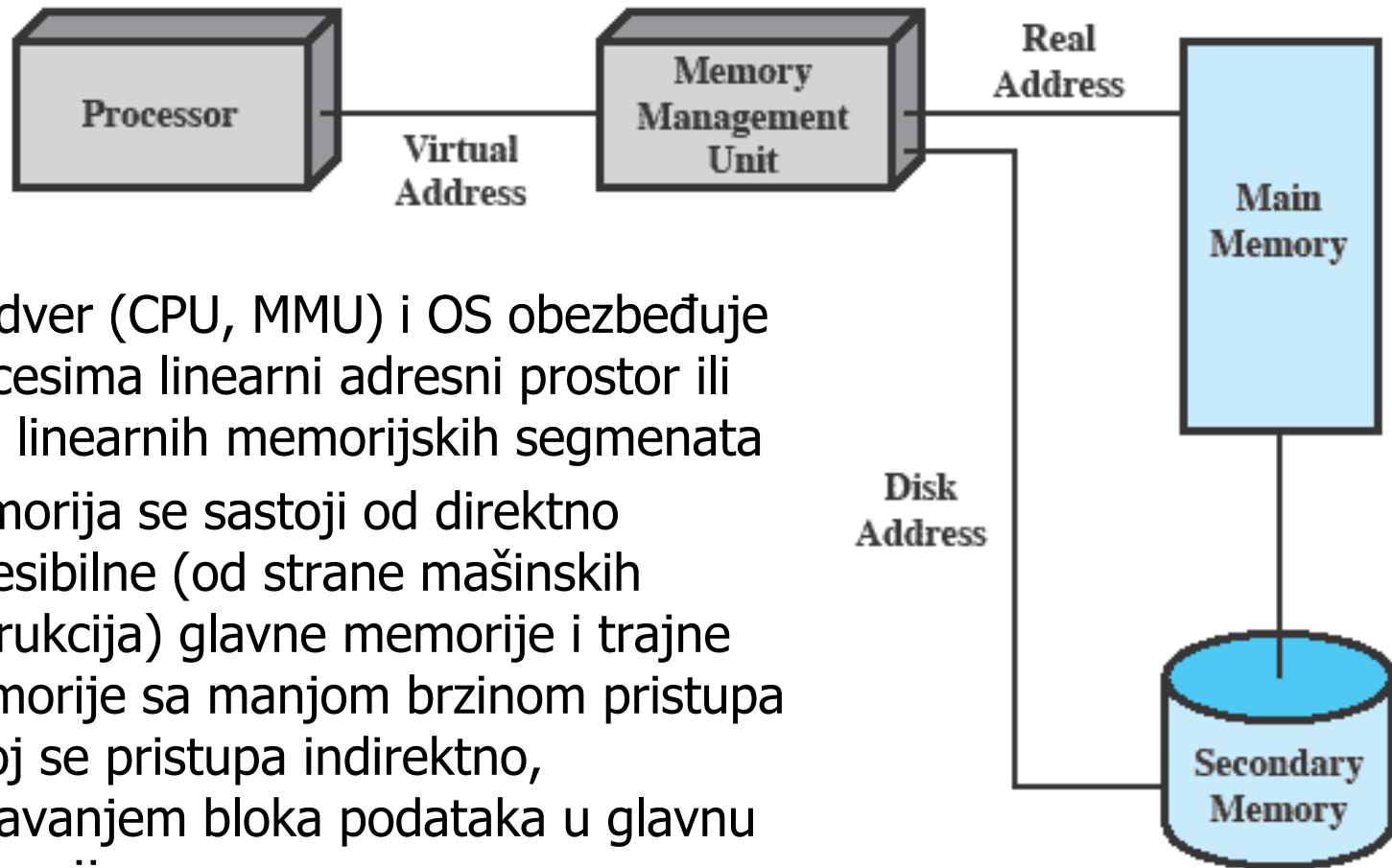
Main Memory



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



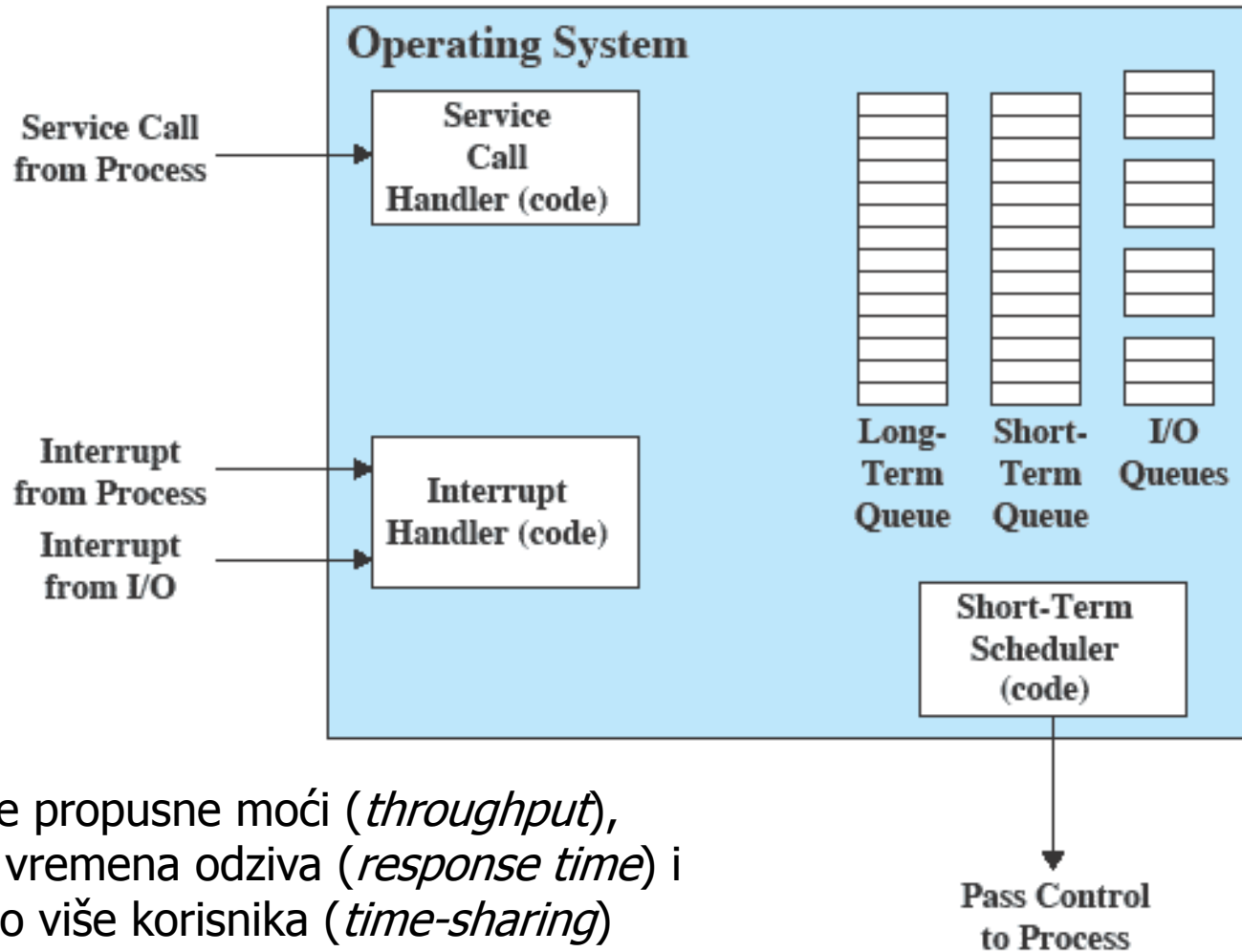
Zaštita i sigurnost informacija

- ✿ Dostupnost
 - ✦ Zaštita sistema od prekida funkcionisanja
- ✿ Tajnost
 - ✦ Zaštita podataka od neovlašćenog pristupa
- ✿ Integritet podataka
 - ✦ Zaštita podataka od neautorizovane modifikacije
- ✿ Autentikacija
 - ✦ Pogodna verifikacija identiteta korisnika i validnosti poruka i podataka



Planiranje i upravljanje resursima

- Strategija planiranja i dodele resursa mora da uključi 3 faktora:
 - Pravičnost
 - Treba napraviti razliku između različitih klasa procesa
 - Efikasnost
 - Maksimizovanje propusne moći (*throughput*), minimizovanje vremena odziva (*response time*) i opsluživanje što više korisnika (*time-sharing*)





Struktura sistema

- Operativni sistem treba da bude modularne strukture sa jasno definisanim intrerfejsima između modula
- Struktura operativnog sistema kao skup hijerarhijskih slojeva (nivoa)
 - Svaki sloj izvršava odgovarajući podskup funkcija
 - Svaki sloj izvršenje svojih funkcija zasniva na sledećem nižem sloju u strukturi i njegovom izvršenju primitivnijih funkcija
 - Ovim se problem rastavlja na određeni broj lakših pod-problema



Hijerarhija dizajna OS

Level	Name	Objects	Example Operations
13	Shell	User programming environment	Statements in shell language
12	User processes	User processes	Quit, kill, suspend, resume
11	Directories	Directories	Create, destroy, attach, detach, search, list
10	Devices	External devices, such as printers, displays, and keyboards	Open, close, read, write
9	File system	Files	Create, destroy, open, close, read, write
8	Communications	Pipes	Create, destroy, open, close, read, write
7	Virtual memory	Segments, pages	Read, write, fetch
6	Local secondary store	Blocks of data, device channels	Read, write, allocate, free
5	Primitive processes	Primitive processes, semaphores, ready list	Suspend, resume, wait, signal
4	Interrupts	Interrupt-handling programs	Invoke, mask, unmask, retry
3	Procedures	Procedures, call stack, display	Mark stack, call, return
2	Instruction set	Evaluation stack, microprogram interpreter, scalar and array data	Load, store, add, subtract, branch
1	Electronic circuits	Registers, gates, buses, etc.	Clear, transfer, activate, complement

HW



Hardverski slojevi

✿ Sloj 1

- ✦ Elektronska kola
- ✦ Objekti su registri, memorijske ćelije i logička kola
- ✦ Operacije su brisanje sadržaja registra ili čitanje iz memorijske lokacije

✿ Sloj 2

- ✦ Instrukcioni set procesora
- ✦ Operacije poput: add, subtract, load, store,...

✿ Sloj 3

- ✦ Dodaje koncept procedure ili potprograma, kao i call/return operacije

✿ Sloj 4

- ✦ Prekidi (*Interrupts*)



Koncepti vezani za multiprogramiranje

☉ Sloj 5

- ☒ Proces kao program u izvršavanju
- ☒ Suspendovanje i nastavljjanje procesa

☉ Sloj 6

- ☒ Sekundarni memorijski uređaji
- ☒ Transfer blokova podataka

☉ Sloj 7

- ☒ Kreiranje logičkog adresnog prostora za procese
- ☒ Organizovanje virtuelnog adresnog prostora u blokove



Upravljanje eksternim objektima

☉ Sloj 8

- ☐ Komunikacija informacijama i porukama između procesa

☉ Sloj 9

- ☐ Podrška za trajno memorisanje imenovanih datoteka

☉ Sloj 10

- ☐ Obezbeđuje pristup eksternim uređajima korišćenjem standardizovanih interfejsa

☉ Sloj 11

- ☐ Održava asocijaciju između eksternih i internih identifikatora u okviru direktorijuma

☉ Sloj 12

- ☐ Obezbeđuje potpunu funkcionalnost za podršku procesima

☉ Sloj 13

- ☐ Obezbeđuje interfejs korisnika prema operativnom sistemu



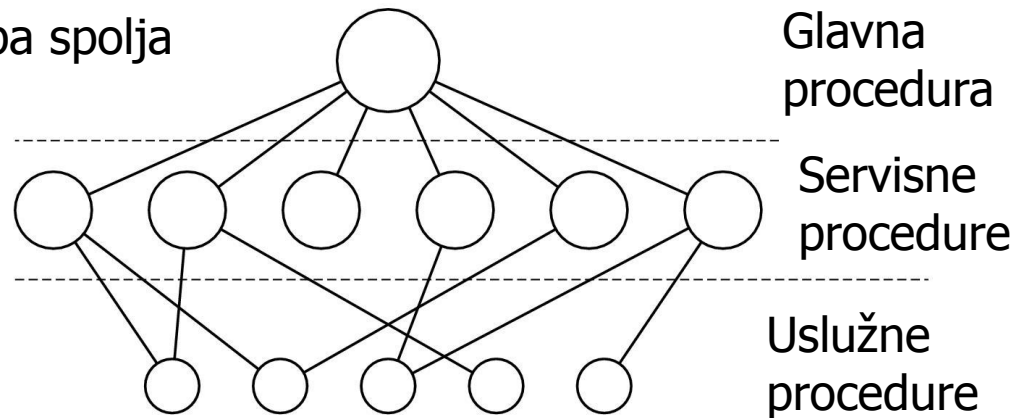
Savremeni operativni sistemi

- ✿ Karakteristike i funkcionalnost savremenih operativnih sistema:
 - ✦ Mikrokernel arhitektura
 - ✦ Višestruke niti (*Multithreading*)
 - ✦ Simetrično multiprocesiranje (*symmetric multiprocessing-SMP*) – *multicore* arhitekture
 - ✦ Distribuirani operativni sistemi
 - ✦ Objektno-orijentisani dizajn



Monolitna struktura

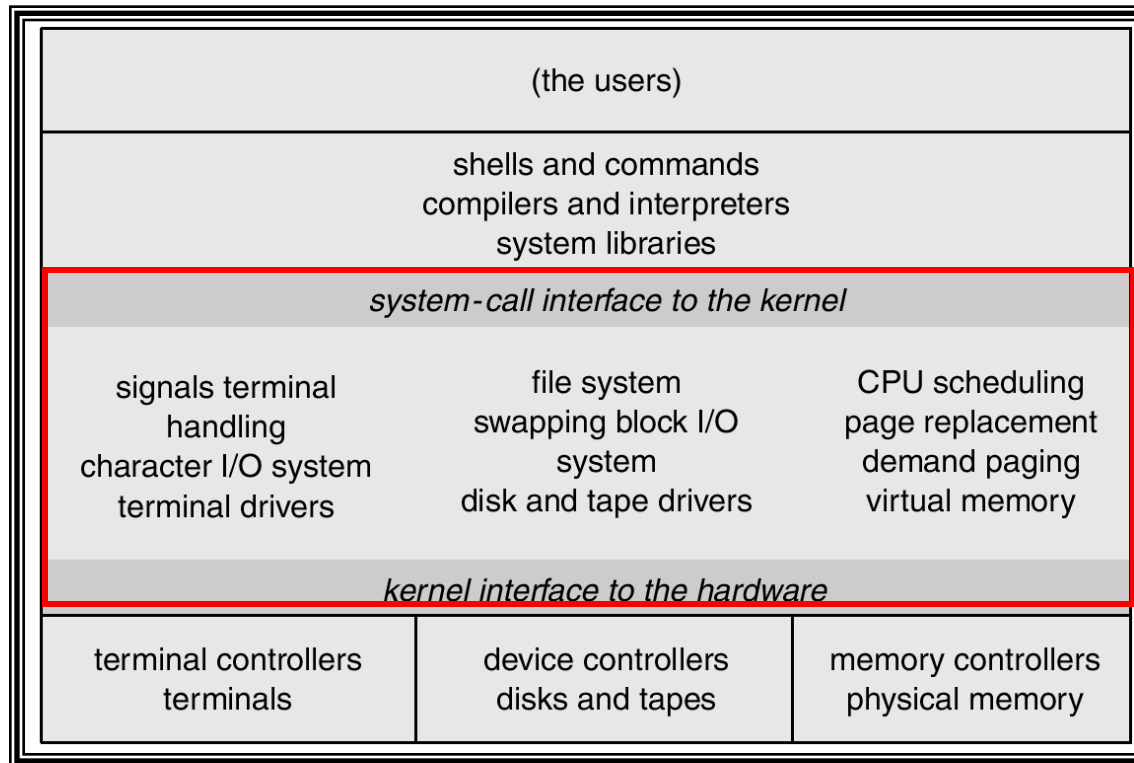
- Operativni sistem je kolekcija procedura. Pri čemu svaka može pozivati svaku poznajući njen interfejs (skup parametara i rezultat) i svaka procedura može pristupati deljivim podacima i strukturama podataka OS
- Jednostavna struktura: glavna, servisne i uslužne procedure
 - Servisne procedure izvršavaju sistemske pozive, a uslužne procedure pomažu jednoj ili više servisnih procedura
- Prednosti:
 - Performanse
 - Visok nivo zaštite od pristupa spolja
- Nedostaci:
 - Loša proširljivost
 - Komplikovana struktura
 - Loša zaštita između komponenti kernela





OS monolitne strukture

- MS DOS, UNIX, Windows (hibridna), Mac OS, Linux
- UNIX - OS sadrži dva dela: sistemski programi i kernel - sadrži sve od interfejsa prema hardverskim komponentama do interfejsa ka sistemskim i aplikativnim programima obezbeđenog u vidu sistemskih poziva





Slojevita struktura

- ✱ Operativni sistem je organizovan kao hijerarhija slojeva pri čemu svaki sloj OS obezbeđuje usluge slojevima iznad i koristi usluge slojeva ispod u hijerarhiji
- ✱ Sloj OS je implementacija apstraktnog objekta koji enkapsulira podatke i sadrži operacije za manipulaciju tim podacima. Te operacije mogu biti pozvane od operacija u slojevima iznad, a s druge strane one pozivaju operacije u slojevima ispod u hijerarhiji
- ✱ Prednost
 - ✱ Modularnost – obezbeđeno je jednostavno debugiranje i verifikacija operativnog sistema, kao i projektovanje i implementacija slojeva OS
- ✱ Nedostaci
 - ✱ Neophodna pažljiva definicija slojeva, jer jedan sloj može koristiti i pozivati samo funkcije slojeva ispod u hijerarhiji
 - ✱ Slaba efikasnost – pri sistemskom pozivu svaki sloj modifikuje parametre, obrađuje podatke i poziva funkcije nižih slojeva generišući dodatan rad i usporavajući izvršenje sistemskog poziva



OS slojevite strukture

- THE sistem – Dijkstra (1968) – Tehnički Univerzitet Ajdhoven, Holandija

Sloj	Funkcija
5	Operator
4	Korisnički programi
3	Upravljanje ulazom/izlazom
2	Komunikacija operatora i procesa
1	Upravljanje memorijom i dobošem
0	Dodela procesora i multiprogramiranje

- MULTICS

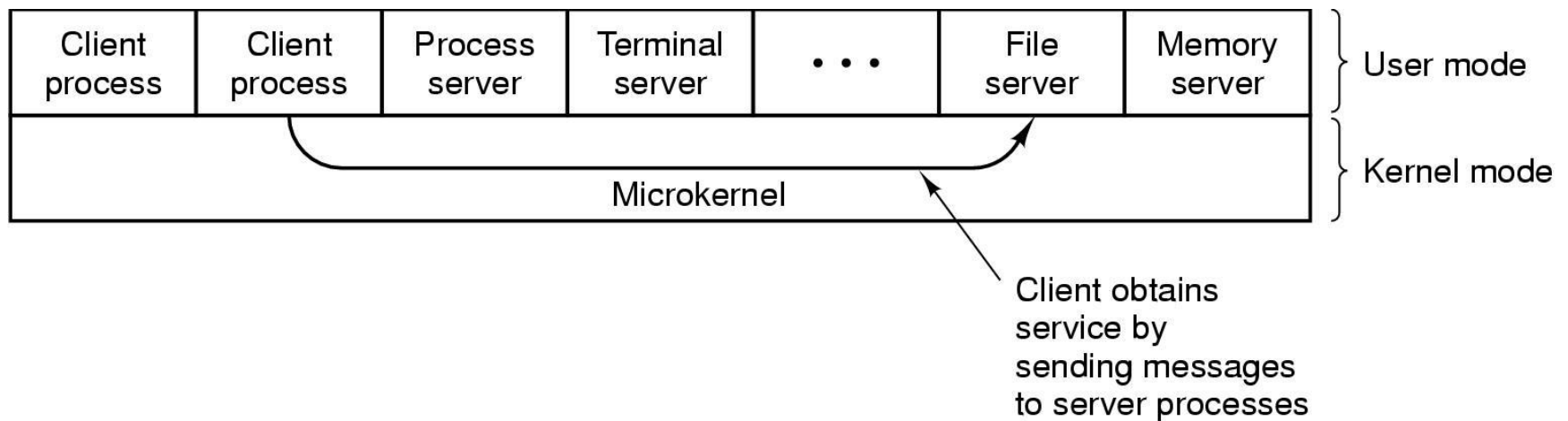
- struktura koncentričnih prstenova
- Procedura u spoljnom prstenu poziva proceduru u unutrašnjem prstenu nekom vrstom sistemskog poziva (provera i smeštanje parametara i TRAP instrukcija)

- OS/2 i prva verzija Windows NT



Mikrokernel arhitektura

- ❖ Struktuiranje operativnog sistema uklanjanjem svih ne značajnih komponenti iz kernela i njihovo implementiranje kao sistemskih ili programa na korisničkom nivou
- ❖ Osnovne funkcije kernela: kreiranje i uništenje procesa, planiranje procesa, sinhronizacija i komunikacija procesa, upravljanje memorijom
- ❖ Ostale funkcije OS implementirane su kao servisi koji se izvršavaju u korisničkom modu, dok je funkcija kernela da upravlja komunikacijom klijenta i servera (na primer, za čitanje bloka datoteke korisnički proces upućuje zahtev serverskom procesu)





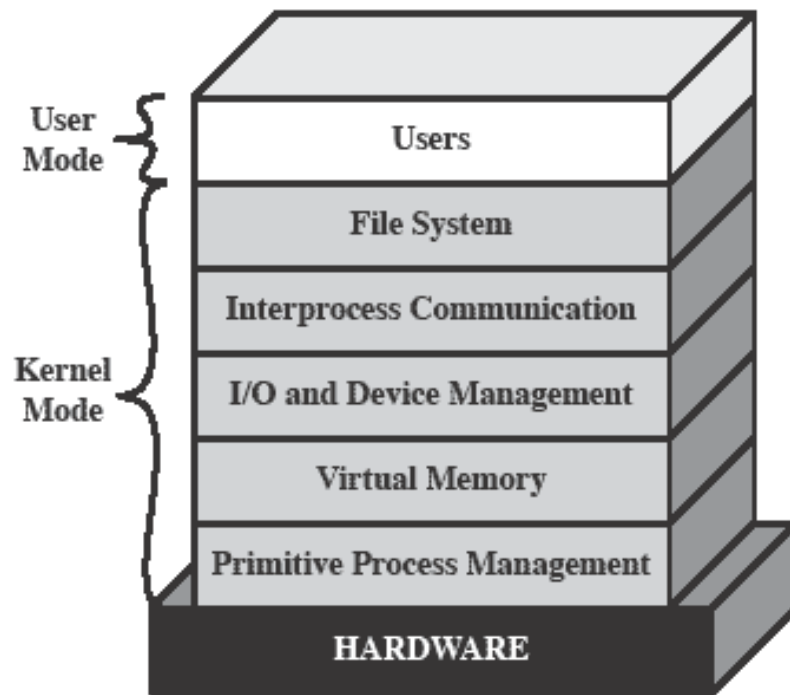
Mikrokernel arhitektura

- ✿ Samo kritični serverski procesi se izvršavaju u modu kernela
 - ✦ na primer, U/I dražveri i pristup U/I uređajima
- ✿ Prednosti
 - ✦ Jednostavno proširenje OS – dodavanje novih servisa ne zahteva modifikaciju kernela
 - ✦ Jednostavno portovanje OS sa jednog na drugi hardver
 - ✦ Pošto se svi serverski procesi izvršavaju u korisničkom modu, nemaju direktan pristup hardveru i pad nekog od njih ne uzrokuje pad OS
 - ✦ Može se prilagoditi radu na distribuiranim sistemima
- ✿ Primeri:
 - ✦ Mach (Carnegie Mellon University, sredina 1980-ih)
 - ✦ Tru64UNIX (ranije Digital UNIX)
 - ✦ Apple MacOS X (Mach kernel + deo BSD kernela)
 - ✦ QNX, MINIX
 - ✦ Windows NT/2000/XP (hibridna struktura)

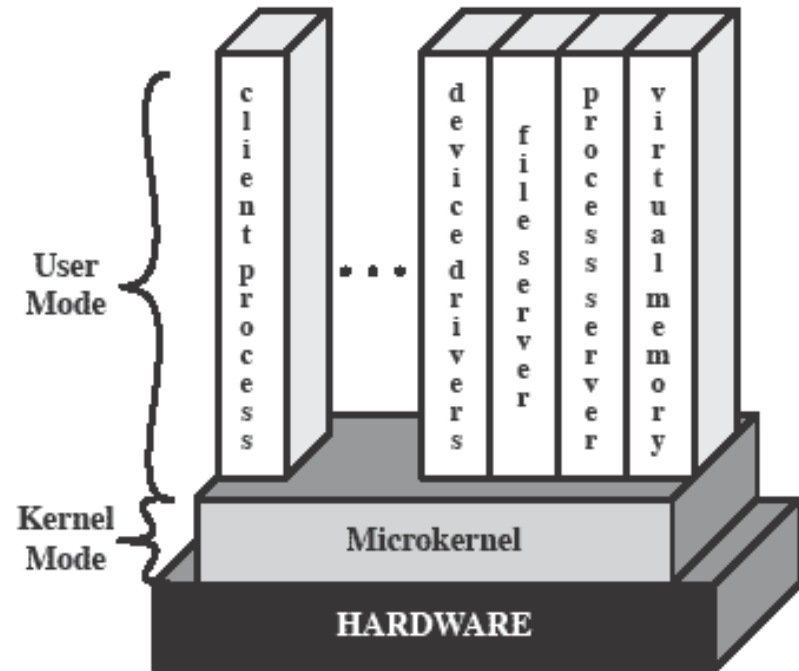


Arhitektura kernela (poglavlje 4.3)

🔗 Slojevita i mikrokernel arhitektura



(a) Layered kernel



(b) Microkernel



Višestruke niti (*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, otvorenihe 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



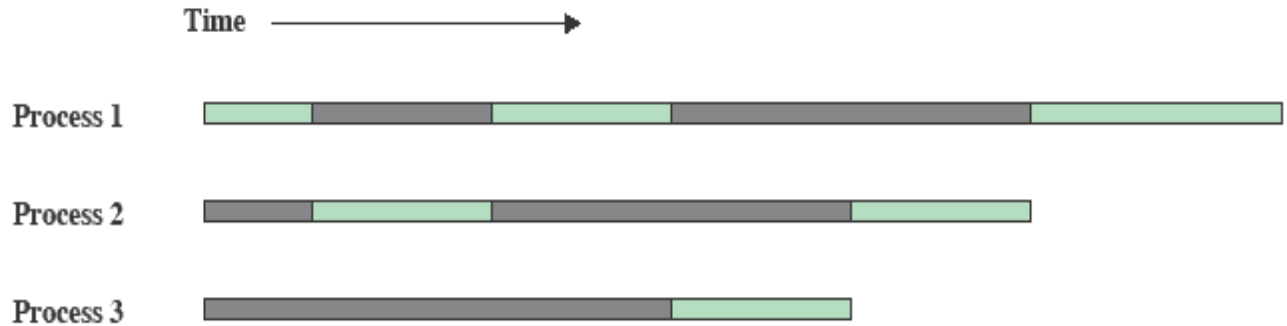
Simetrično multiprocesiranje (SMP)

- ✿ 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



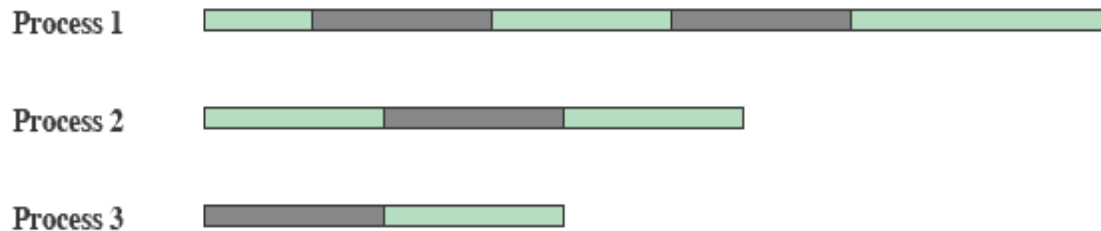
Simetrično multiprocessing (SMP)

• Multiprogramiranje (jedan procesor)



(a) Interleaving (multiprogramming, one processor)

• Multiprocesiranje (dva procesora)



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

Blocked Running

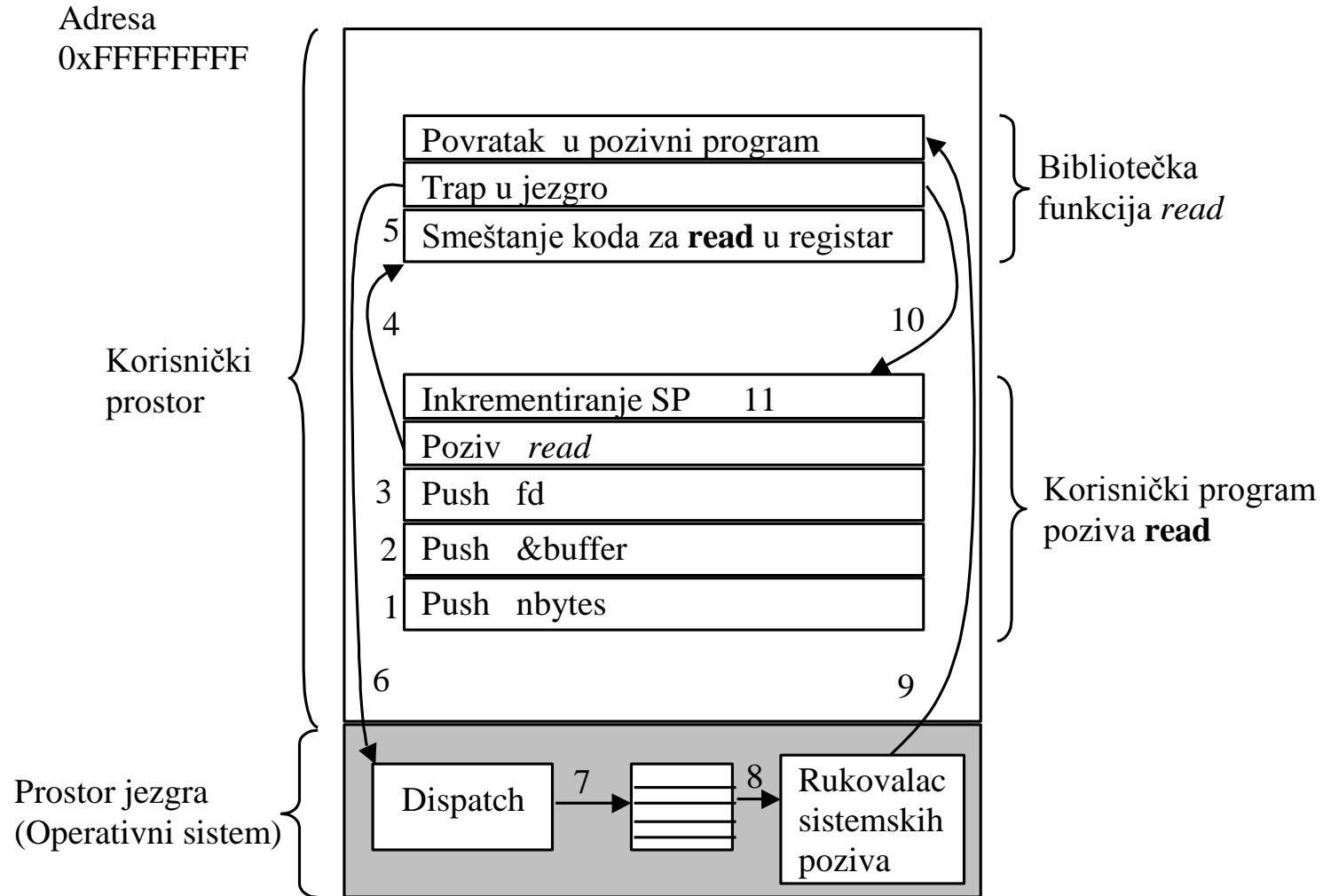


Sistemske pozivi

- ❖ 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 1003.1 standard
 - ❖ Windows - Win32 API (*Application Programming Interface*)
- ❖ Sistemske poziv se obavlja u okviru programa napisanog u programskom jeziku visokog nivoa pozivom funkcije iz standardne biblioteke uključene u prevodilac. U okviru ove funkcije se argumenti funkcije smeštaju na *stack*, i poziva **trap** instrukcija čiji je argument kod sistemskog poziva. Ova instrukcija izaziva softverski prekid, OS čuva stanje prekinutog procesa, prelazi u mod kernela i aktivira funkciju (rutinu, *system call handler*) kernela 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.

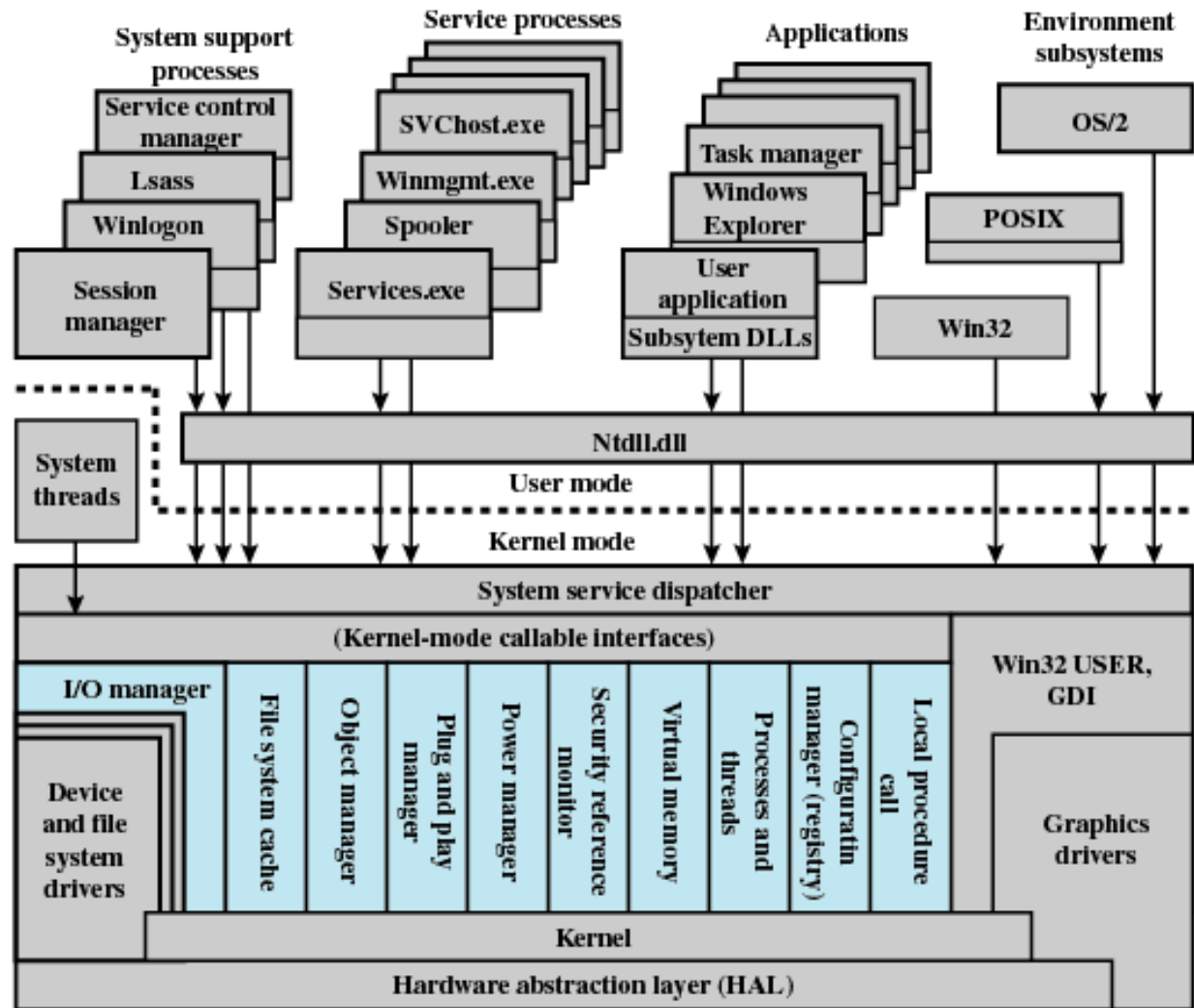


Izvršenje sistemskog poziva





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

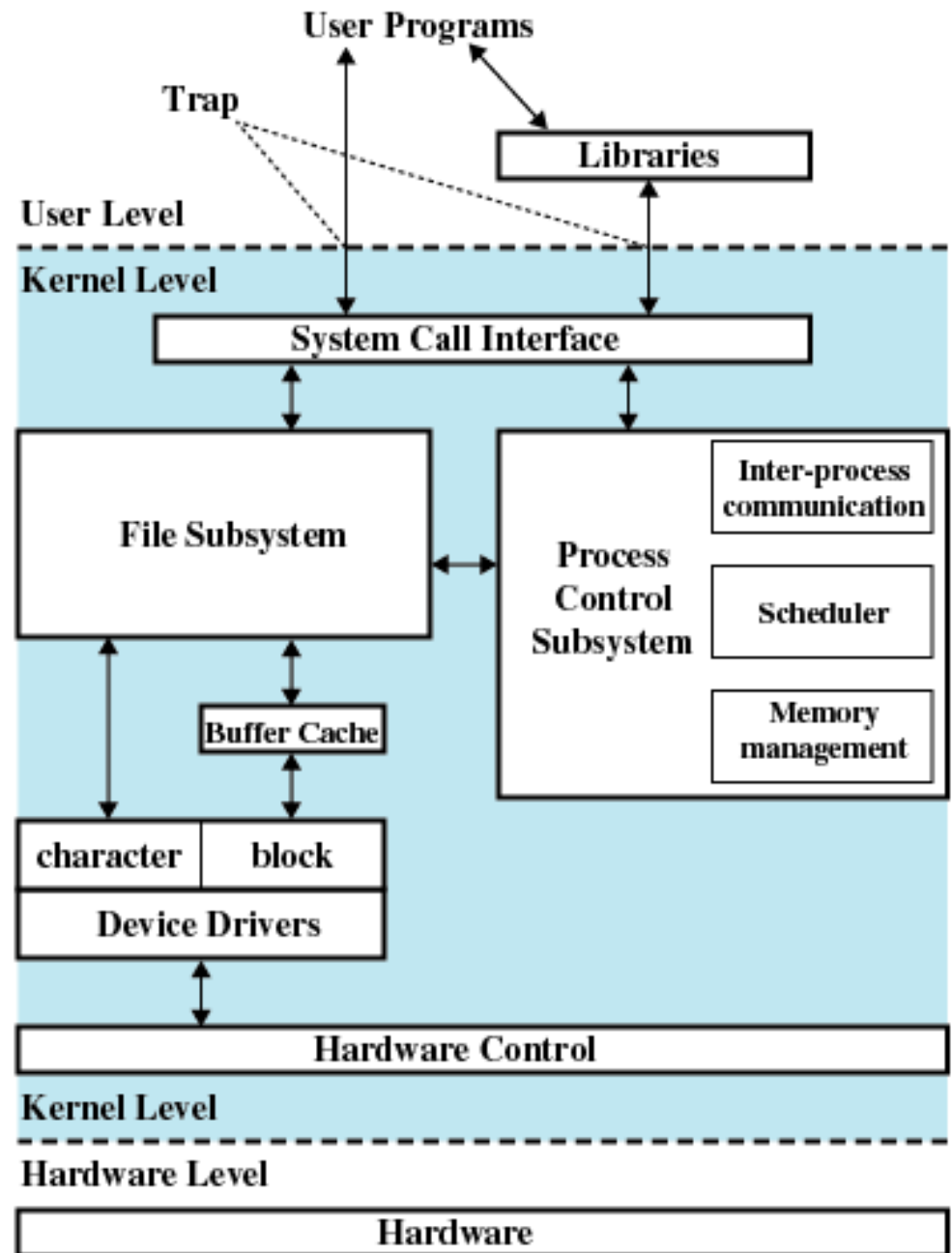
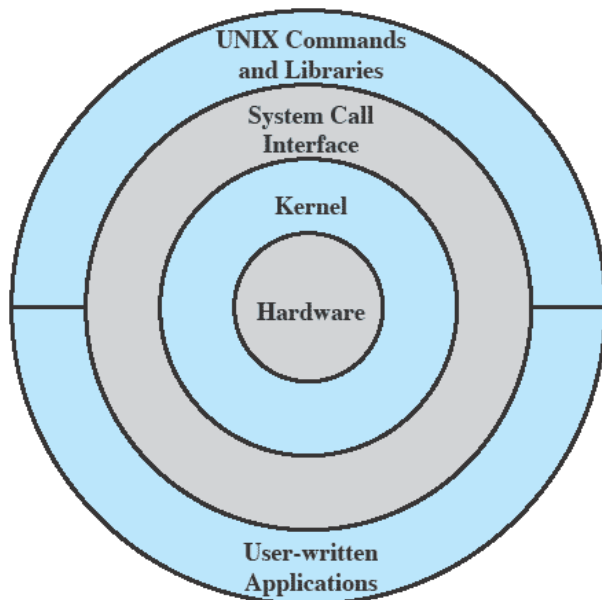


UNIX

Verzije

- ❖ System V R4 (SVR4)
- ❖ Solaris 10
- ❖ BSD
- ❖ Linux

Tradicionalni UNIX kernel

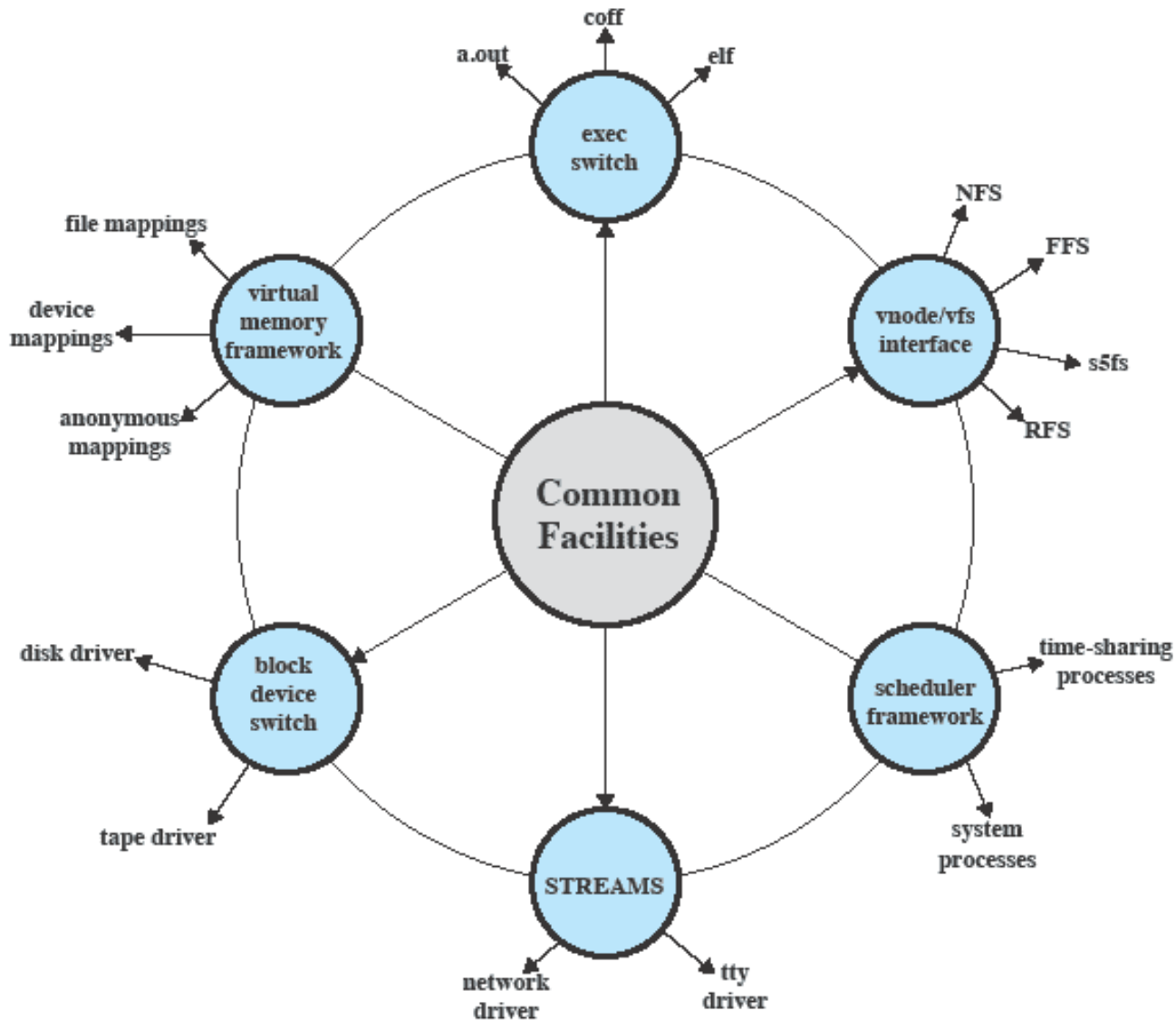


Uvod i pregled operativnih sistema

Operativni sistemi



Savremeni UNIX kernel

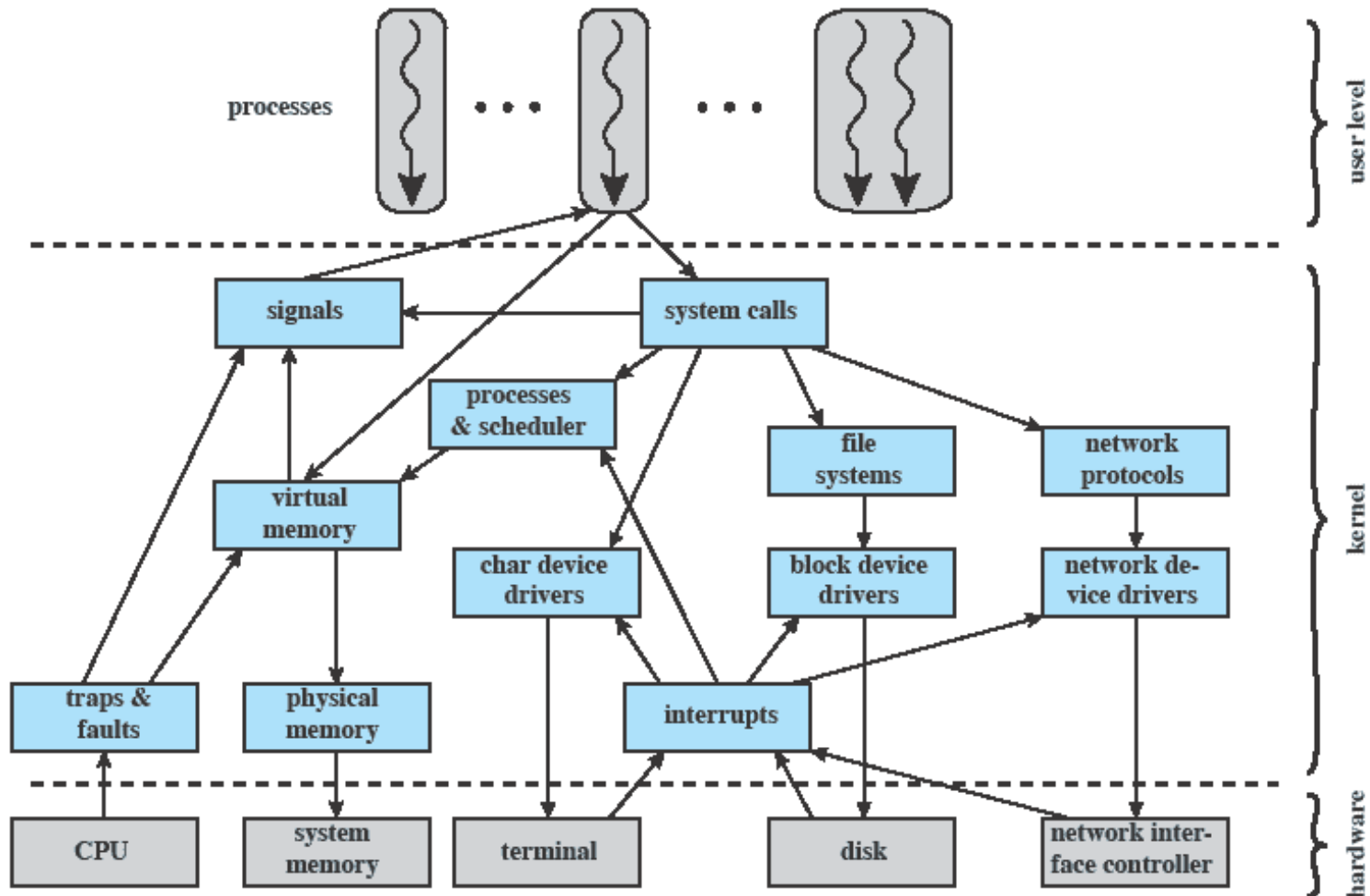


Uvod i pregled operativnih sistema

Operativni sistemi



Komponente Linux kernela





Domaći zadatak

- ✚ Pročitati i naučiti poglavlje iz knjige

2. Pregled operativnog sistema (2. Operating System Overview)

- ✚ Animacije funkcija i algoritama operativnog sistema

<http://williamstallings.com/OS-Animation/Animations.html>