



Operativni sistemi

- Upravljanje U/I i planiranje diska -

Prof. dr Dragan Stojanović

Katedra za računarstvo
Univerzitet u Nišu, Elektronski fakultet



Literatura

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

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

- ✿ **Poglavlje 11:** Upravljanje U/I i planiranje diska

Upravljanje U/I

- ✿ Upravljanje U/I predstavlja kompleksan i “najzamršeniji” deo dizajna i implementacije operativnog sistema
- ✿ Veoma je teško realizovati generalno, konzistentno rešenje za upravljanje U/I pošto postoji velika raznolikost U/I uređaja, kao i primena tih uređaja

Kategorije U/I uređaja

✿ Čitljivi za čoveka

- ✦ koriste se za komunikaciju sa korisnikom
- ✦ štampači
- ✦ video terminali
 - ekran
 - tastatura
 - miš

✿ Čitljivi za mašinu

- ✦ koriste se za komunikaciju sa elektronskom/računarskom opremom
- ✦ jedinice diska i trake
- ✦ senzori
- ✦ kontroleri
- ✦ aktuatori

✿ Komunikacioni

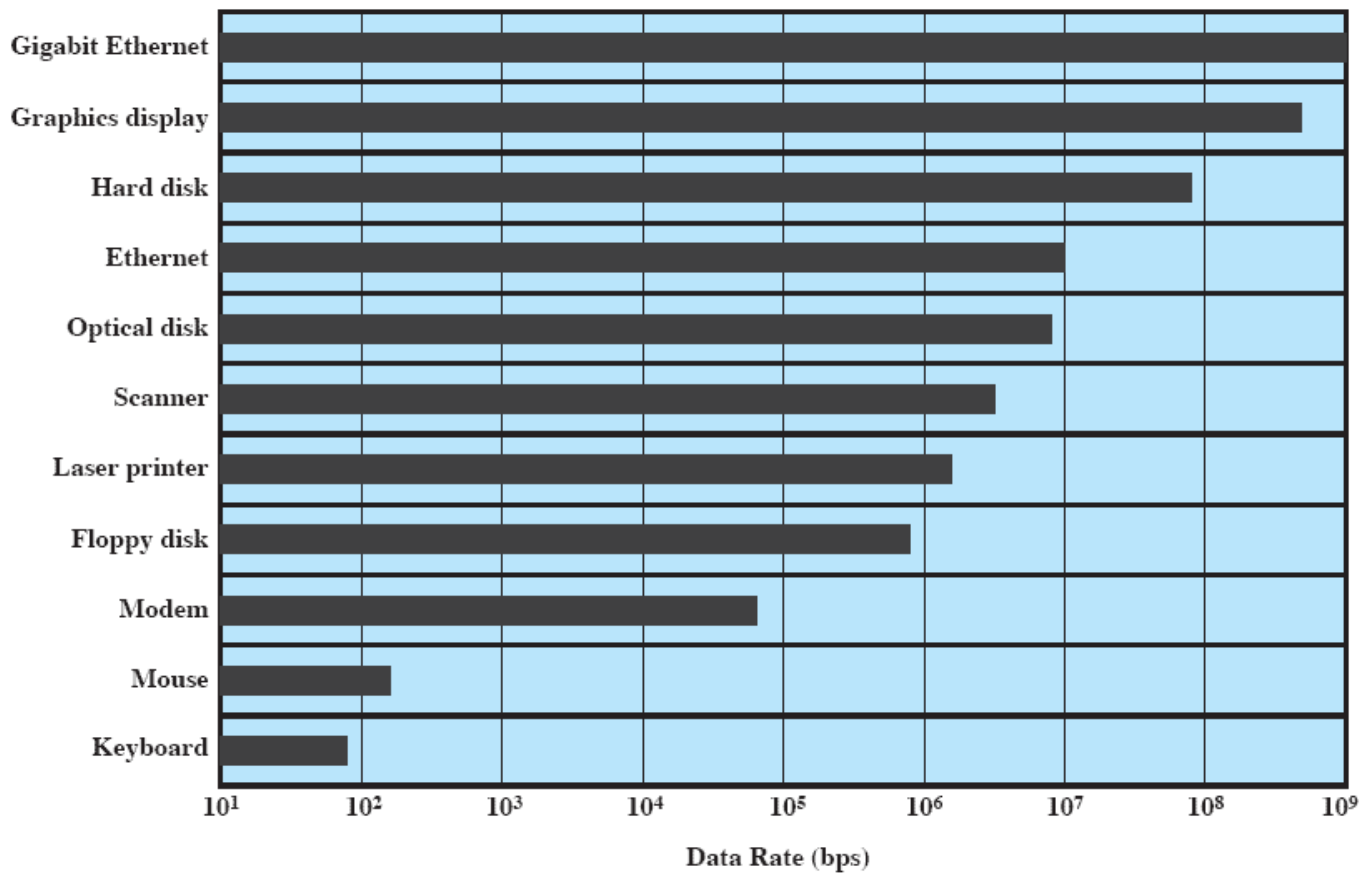
- ✦ koriste se za komunikaciju sa udaljenim uređajima
- ✦ mrežni adapteri
- ✦ modemi

Razlike među U/I uređajima

- ✿ Postoje velike razlike između klasa uređaja, čak i značajne razlike u okviru iste klase uređaja
 - ✦ Brzina prenosa podataka
 - ✦ Primena uređaja
 - ✦ Složenost upravljanja
 - ✦ Jedinica prenosa podataka – niz znakova (bajtova) ili blokovi
 - ✦ Reprezentacija podataka
 - ✦ Uslovi greške
- ✿ Ove različitosti otežavaju postizanje uniformnog i konzistentnog pristupa u upravljanju U/I uređajima u okviru operativnog sistema

Brzina prenosa podataka

Tipična brzina prenosa podataka U/I uređaja



Jedinica prenosa podataka

- ✿ **Prema jedinici prenosa podataka** U/I uređaji se mogu podeliti u dve kategorije:
 - ✦ blok uređaji
 - ✦ znakovni uređaji
- ✿ **Blok uređaji:** prenose podatke u blokovima fiksne veličine koji su adresibilni, npr. disk ili DVD
 - ✦ Tipična veličina bloka 512 bajta do 32 KB
 - ✦ Može se čitati i zapisivati svaki blok nezavisno od ostalih
 - ✦ Komande `read`, `write`, `seek`
- ✿ **Znakovni uređaji:** prihvataju i isporučuju tokove znakova (bajtova) koji nisu adresibilni
 - ✦ Npr. printeri, mrežni interfejsi, miš, tastatura, serijski portovi
 - ✦ Komande `get`, `put`
- ✿ Ima uređaja koji se ne mogu svrstati u ove dve kategorije
 - ✦ Na primer `clock` (tajmer)

Tehnike za izvođenje U/I

✿ Programirani U/I

- ✦ Procesor u korist procesa izdaje U/I komandu U/I modulu
- ✦ U/I modul izvršava U/I operaciju i u U/I statusni registar postavlja fleg da je operacija završena
- ✦ Proces koji je izdao U/I komandu je u radnom čekanju dok se U/I operacija ne završi i za to vreme proverava statusni registar

✿ U/I vođen prekidima

- ✦ Procesor u korist procesa izdaje U/I komandu U/I modulu
- ✦ Procesor nastavlja da izvršava naredne instrukcije
- ✦ U/I modul izvršava U/I komandu i šalje prekid kada je U/I komanda obrađena

✿ U/I korišćenjem DMA

- ✦ DMA modul upravlja razmenom podataka između glavne memorije i U/I modula
- ✦ Procesor šalje zahtev za prenos bloka podataka DMA modulu
- ✦ Procesor se prekida tek pošto se čitav blok prenese

Odnos među U/I tehnikama

- U mnogim računarskim sistemima DMA je dominantan oblik prenosa podataka koji mora biti podržan od strane operativnog sistema

	NE KORISTI PREKID	KORISTI PREKID
Prenos podataka između U/I uređaja i memorije ide preko procesora	Programirani U/I	Prekidima vođen U/I
Direktni prenos podataka između U/I uređaja i memorije		Direktni pristup memoriji (DMA)

Evolucija funkcije U/I

1. Procesor direktno upravlja U/I uređajima
 - Ovo se javlja kod jednostavnih uređaja kontrolisanih mikroprocesorima
2. Dodaje se kontroler ili U/I modul
 - Procesor koristi programirani U/I bez prekida. Na ovaj način procesor je odvojen od specifičnih detalja interfejsa perifernih uređaja
3. Ista konfiguracija kao i pod 2, ali se koriste prekidi.
 - Procesor ne troši vreme čekajući na završetak U/I operacije čime se povećava efikasnost
4. U/I modulu je omogućena direktna kontrola memorije preko DMA
 - Moguće je prebacivati blok podataka iz memorije ili u memoriju bez uključivanja procesora, osim na početku i kraju prenosa

Evolucija funkcije U/I (2)

5. U/I modul je proširen tako da predstavlja poseban procesor sa specijalnim skupom instrukcija prilagođenim U/I - **U/I kanal**
 - Centralni procesor (CPU) upućuje U/I procesor da izvrši U/I program koji se nalazi u glavnoj memoriji. U/I procesor pribavlja i izvršava ove instrukcije bez intervencije CPU. Ovim se omogućava da CPU specificira sekvencu U/I aktivnosti i da bude prekinut tek kada se celokupna sekvenca izvrši.
6. U/I modul poseduje sopstvenu lokalnu memoriju i u stvari je samostalni računar - **U/I procesor**
 - Sa ovom arhitekturom omogućava se upravljanje velikog skupa U/I uređaja uz minimalno angažovanje procesora

Direct Memory Access (DMA)

- ❖ Procesor prosleđuje U/I operaciju DMA modulu
- ❖ DMA modul vrši prenos podata direktno u ili iz memorije
- ❖ Kada završi prenos DMA modu šalje signal prekida procesoru

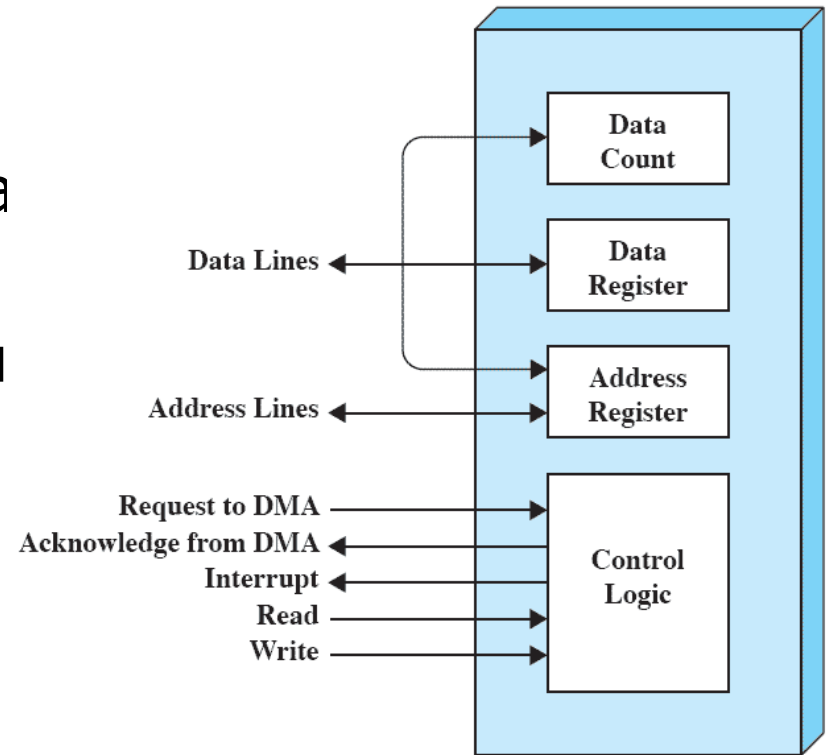
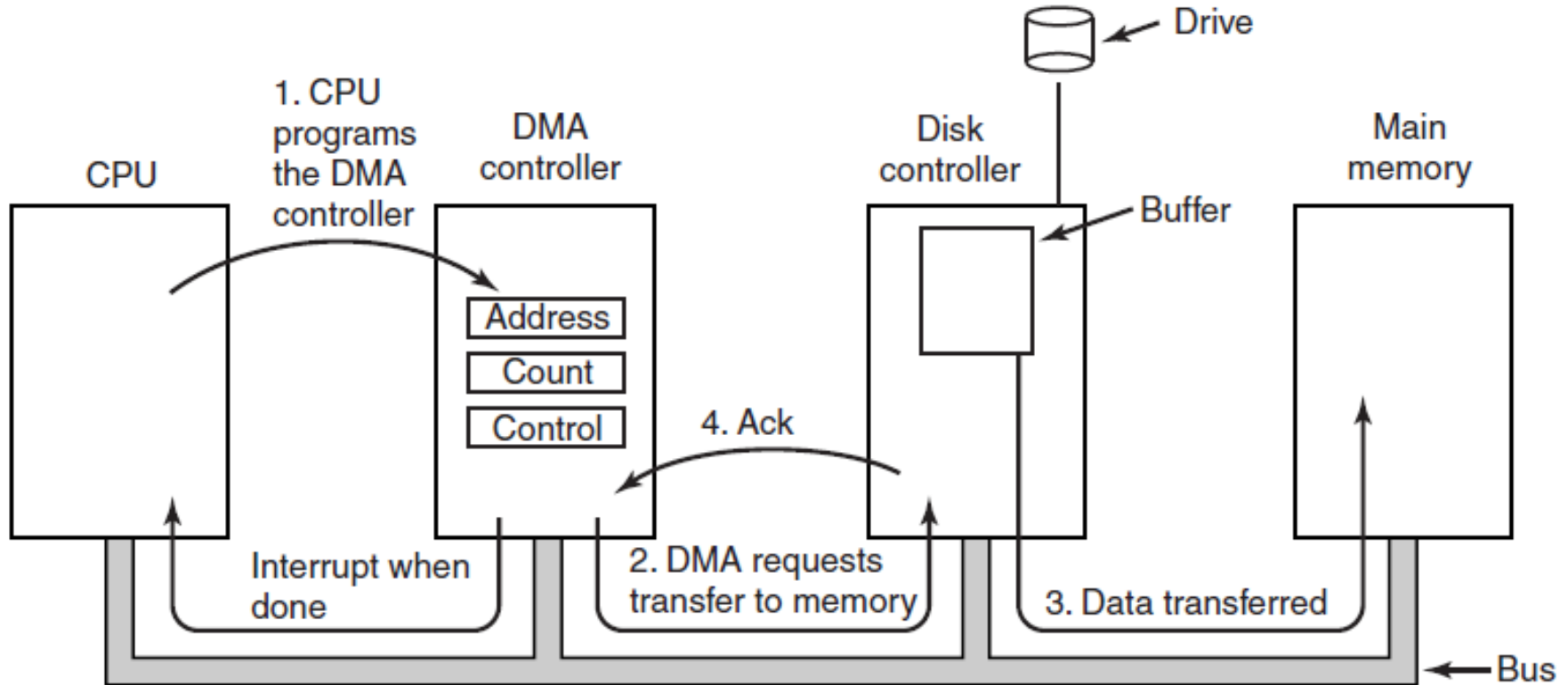


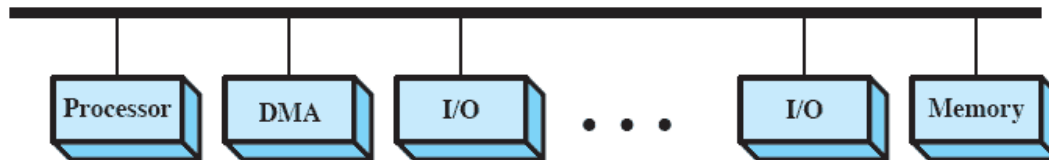
Figure 11.2 Typical DMA Block Diagram

Direct Memory Access (DMA)

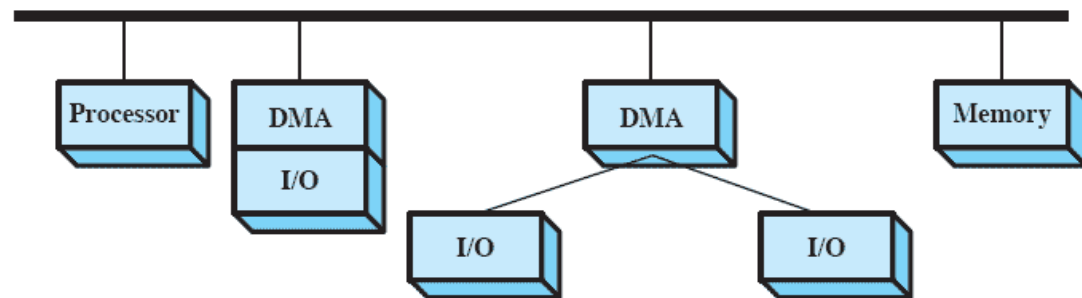


Tanenbaum, 2014

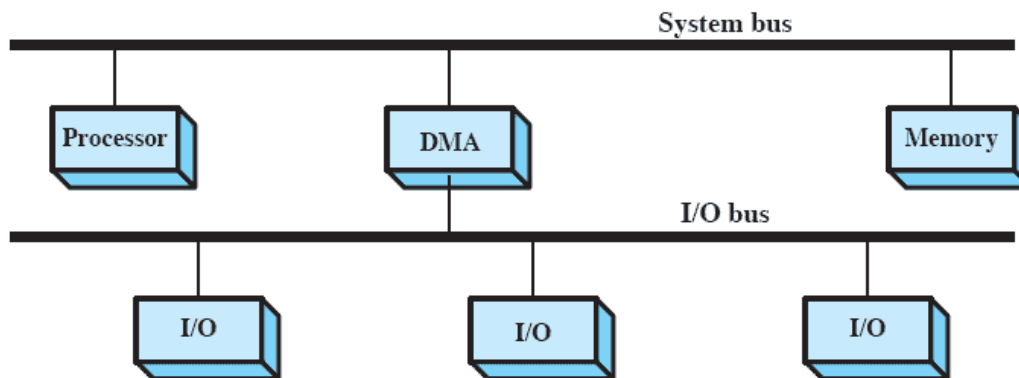
DMA konfiguracije



(a) Single-bus, detached DMA



(b) Single-bus, Integrated DMA-I/O



(c) I/O bus

Ciljevi U/I sistema - Efikasnost

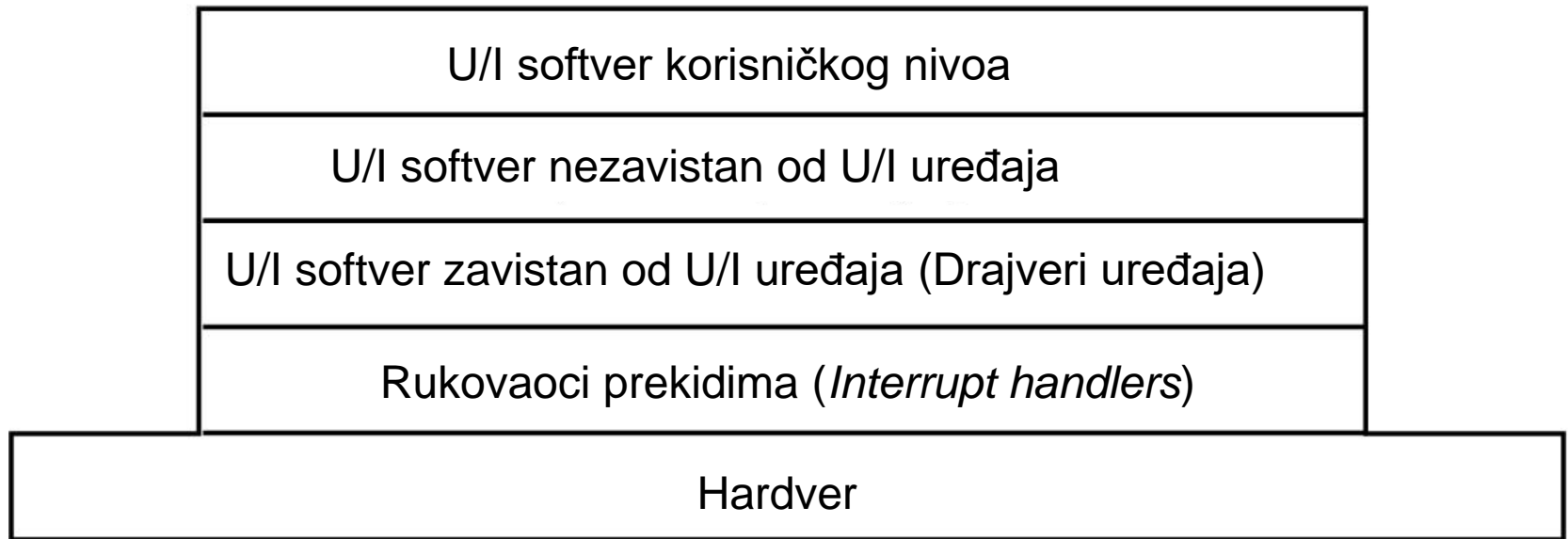
- ✱ Većina U/I uređaja je daleko sporija od glavne memorije i procesora
- ✱ Korišćenje multiprogramiranja omogućava da dok neki procesi čekaju na U/I, drugi procesi se izvršavaju
- ✱ Međutim U/I je daleko sporiji od procesora, tako da bi se obezbedila zaposlenost procesora mora se primeniti *swap*-ovanje (što je takođe U/I operacija)
- ✱ CPU se sve više oslobađa zadataka u vezi U/I –a i to prenosi na U/I modul (U/I kanal, U/I procesor), što poboljšava performanse,
 - ▣ Posebna pažnja je usmerena na U/I diska



Ciljevi U/I sistema - Generalnost

- ✿ Sa U/I uređajima treba raditi na jednoobrazan (uniforman, generalan) način
 - ▣ Procesi treba da vide na uniforman način U/I uređaje
 - ▣ OS treba na uniforman način da upravlja U/I uređajima

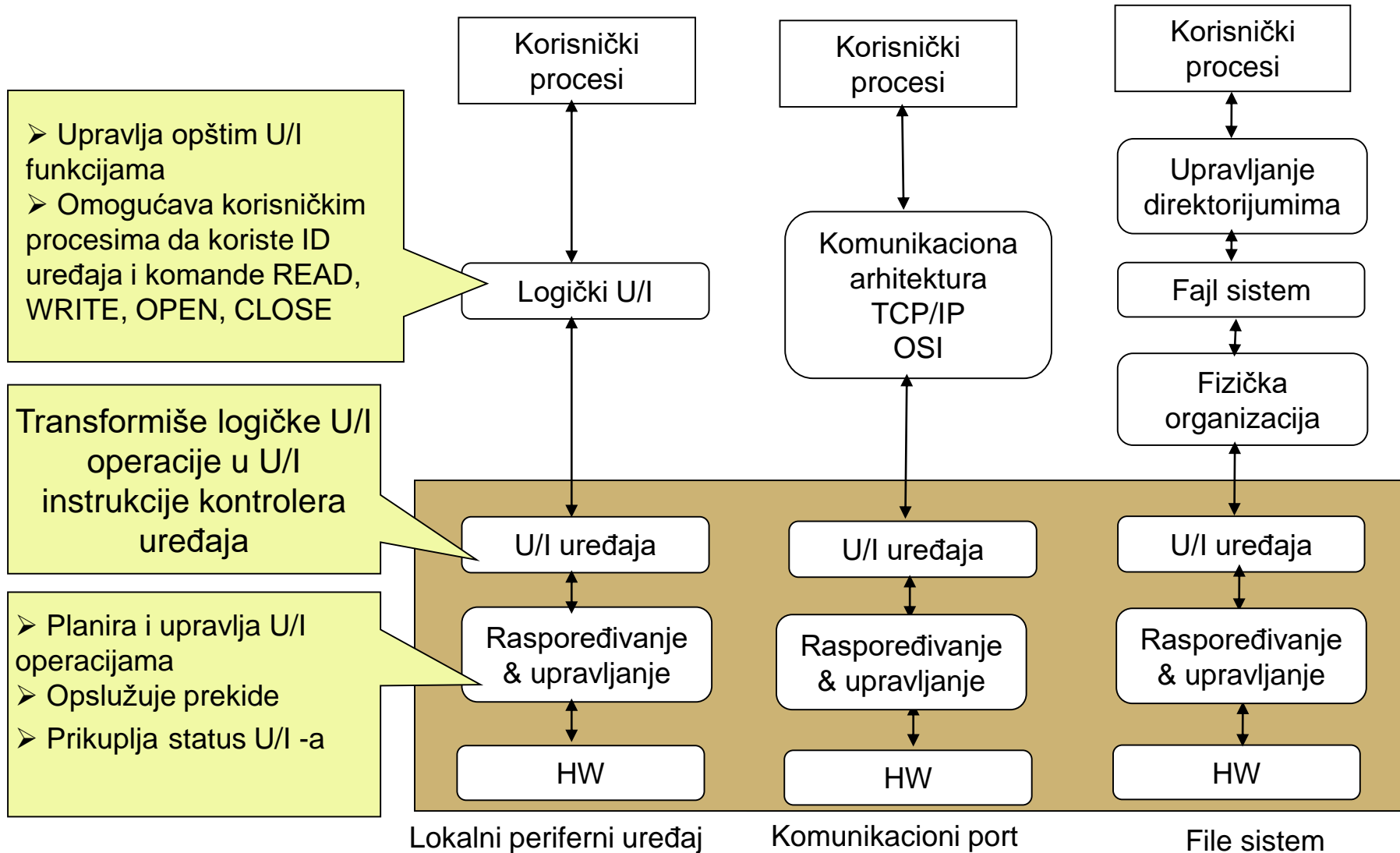
Slojevi U/I softvera



Tanenbaum, 2014

- ✿ Svaki sloj ima precizno definisanu funkciju i dobro definisan interfejs prema susednim slojevima
- ✿ Promene u jednom sloju se ne odražavaju na susedne slojeve

Modeli hijerarhijske organizacije U/I



Upravljanje U/I i planiranje diska

Operativni sistemi



Lokalni periferijski uređaji

- Logički U/I:
 - Upravlja uređajem kao logičkim resursom
- U/I uređaja:
 - Transformiše logičke U/I operacije u sekvencu U/I instrukcija kontrolera uređaja
- Planiranje i upravljanje:
 - Izvršava uređenje i planiranje U/I operacija i upravljanje operacijama. Rukovanje izuzecima i očitavanje i prijavljivanje U/I statusa se obavljaju na ovom sloju.



Komunikacioni port

- ❁ Identično kao u prethodnom slučaju samo što je logički U/I modul zamenjen komunikacionom arhitekturom
 - ❁ Ova arhitektura se sastoji od više slojeva
 - ❁ Primer, TCP/IP

File sistem

Upravljanje direktorijumima

- Na ovom nivou simbolička imena datoteka se prevode u identifikatore kojima se referenciraju datoteke, i obezbeđuje podrška za korisničke operacije nad direktorijumima, poput *add*, *delete*, *reorganize*

File sistem

- Obezbeđuje logičku strukturu datoteka, korisničke operacije, poput *open*, *close*, *read*, *write*, kao i prava pristupa

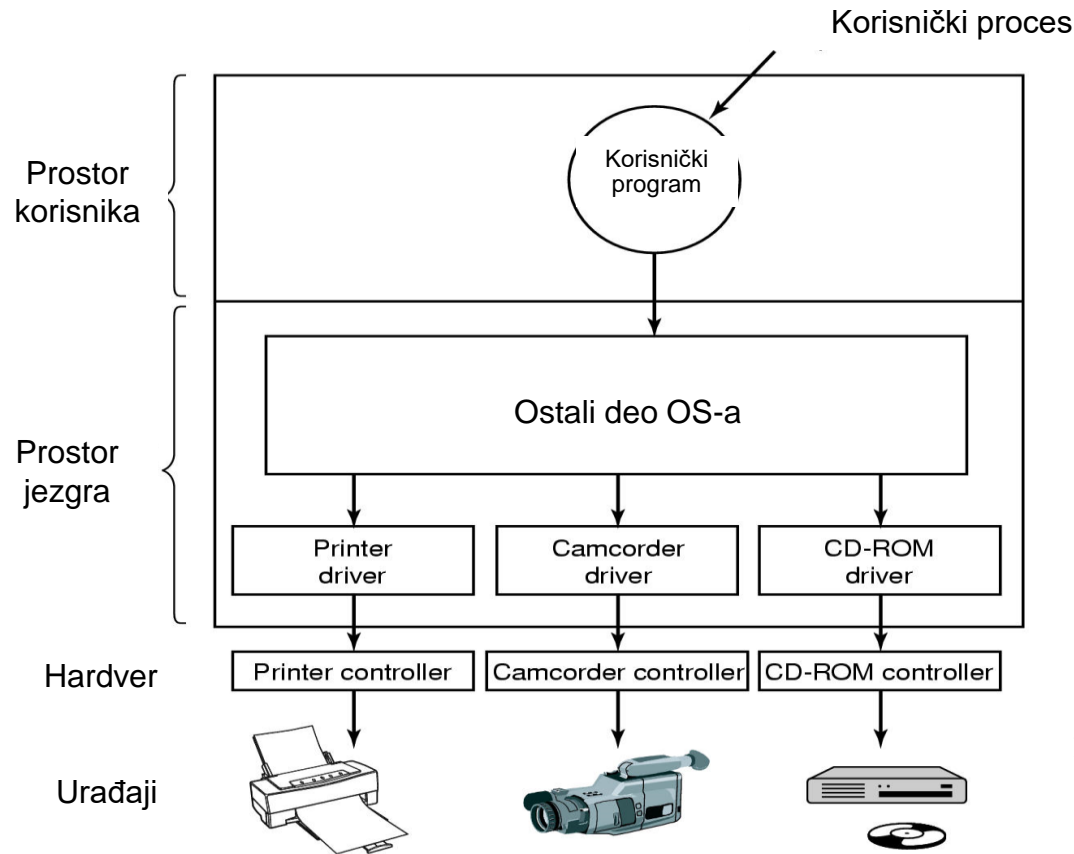
Fizička organizacija

- Logičke reference na datoteke se prevode u fizičke adrese na sekundarnoj memoriji, vodeći računa o fizičkoj strukturi medijuma

Drajveri uređaja (1)

- ✿ Za svaki uređaj koji se priključuje na računar potreban je specifičan program koji će njime upravljati - to je **drajver uređaja**
- ✿ Najčešće ga isporučuju proizvođači uz sam U/I uređaj
- ✿ Drajver uređaja je prilagođen konkretnom OS-u (Unix, Linux, MS Windows)
- ✿ Mnoge specifičnosti samog U/I uređaja su ugrađene u drajver uređaja
- ✿ Svaki drajver uređaja normalno rukuje jednim tipom U/I uređaja ili jednom klasom sličnih U/I uređaja
- ✿ Drajver uređaja je deo jezgra OS-a, pa može pristupati registrima uređaja
- ✿ Drajver uređaja ima:
 - ▣ dobro definisan interfejs ka ostalim delovima OS-a
 - ▣ precizno definisanu funkcionalnost

Drajveri uređaja (2)



Tanenbaum, 2014

- ✚ Komunikacija između drajvera i kontrolera uređaja ide preko sistemske magistrale

Drajveri uređaja (3)

Model aktivnosti drajvera

✚ **Stvarna kontrola U/I uređaja** obuhvata

- ✚ Određivanje sekvence komandi koje treba poslati U/I uređaju
- ✚ Upis komandi u registar uređaja
 - Nakon upisa svake komande drajver proverava da li kontroler uređaja prihvata tu komandu i da li je spreman da prihvati sledeću komandu
 - Neki kontroleri prihvataju komandu po komandu, dok neki imaju u memoriji bafer gde se može smestiti više komandi, odakle ih kontroler uzima na obradu bez podrške OS-a

✚ Nakon slanja svih komandi

- ✚ Drajver se blokira i čeka da ga prekid od U/I uređaja deblokira (ako obrada traje) ili
- ✚ Drajver se ne blokira (ako operacija ne traži kašnjenje)

✚ Nakon završetka operacije, drajver proverava da li je nastala greška

- ✚ Ako nije, prosleđuje podatke zahtevaocu U/I operacije
- ✚ Ako je nastala neka greška, vraća zahtevaocu kod greške
- ✚ Nakon toga bira sledeći zahtev za obradu, ako ih ima u redu zahteva, a ako ne on se blokira i čeka novi zahtev

U/I baferovanje

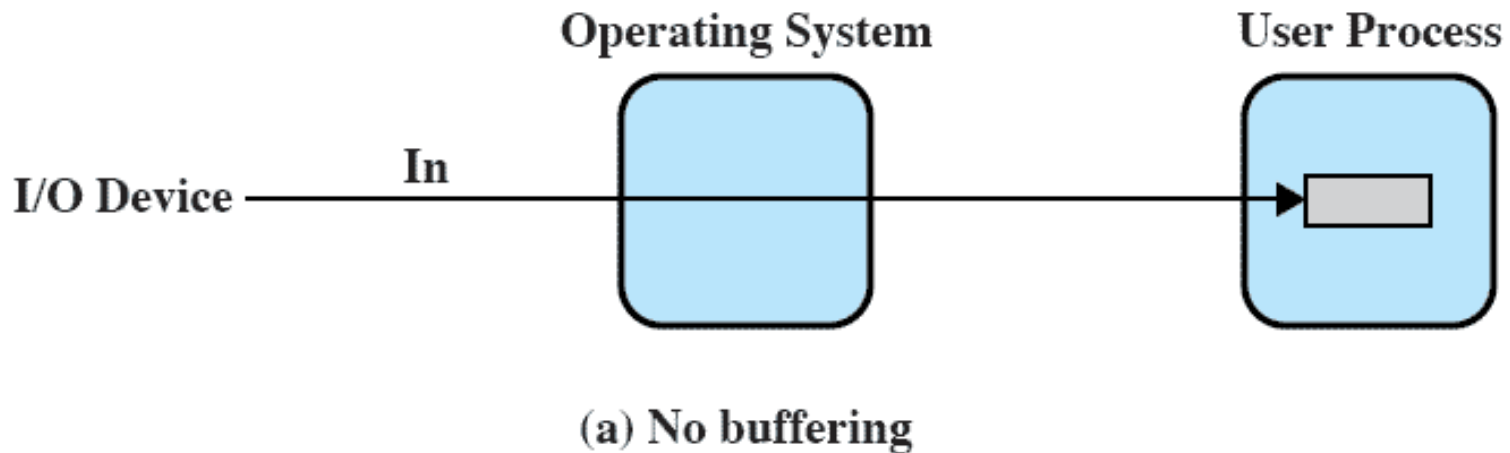
- ❁ Procesi moraju da čekaju dok se ne završi U/I i tek kada su podaci raspoloživi mogu da nastave sa izvršenjem
- ❁ Da bi se izbeglo uzajamno blokiranje neke stranice procesa moraju biti ("zaključane") u glavnoj memoriji tokom izvršavanja U/I
- ❁ Efikasnije je da se transfer ulaznih podataka obavi unapred u odnosu na zahtev, a da se transfer izlaznih podataka izvrši sa zakašnjenjem u odnosu na zahtev
- ❁ Bafer je oblast u memoriji gde se smeštaju podaci dok se prenose između U/I uređaja i memorijskog prostora procesa
 - ❏ Kada se ova U/I operacija izvodi može se garantovati da je zapisano stanje podataka u trenutku poziva U/I operacije

U/I baferovanje (2)

- ✿ Baferovanje orijentisano na blokove
 - ✦ Podaci su smešteni i prenose se u blokovima fiksne veličine
 - ✦ Transfer se obavlja pojedinačno, jedan blok u jednom trenutku, na osnovu reference bloka
 - ✦ Koristi se za sekundarne memorije, diskove, USB flash memorije, itd.
- ✿ Baferovanje orijentisano na tokove
 - ✦ Podaci se prenose u tokovima (stream) bajtova
 - ✦ Koristi se kod terminala, printera, komunikacionih portova, miša, tastature, itd

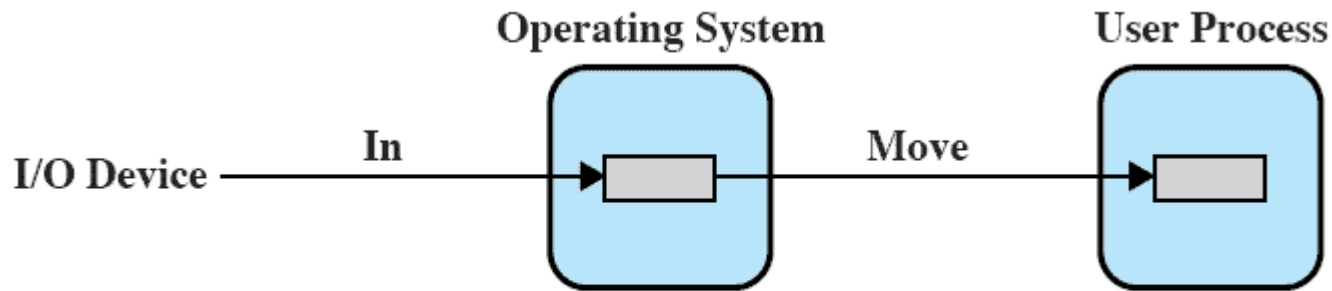
Bez U/I bafera

- Bez bafera OS direktno pristupa uređaju kada je neophodno



Jedan bafer

- Operativni sistem obezbeđuje jedan (pojedinačni) bafer u glavnoj memoriji za opsluživanje U/I zahteva

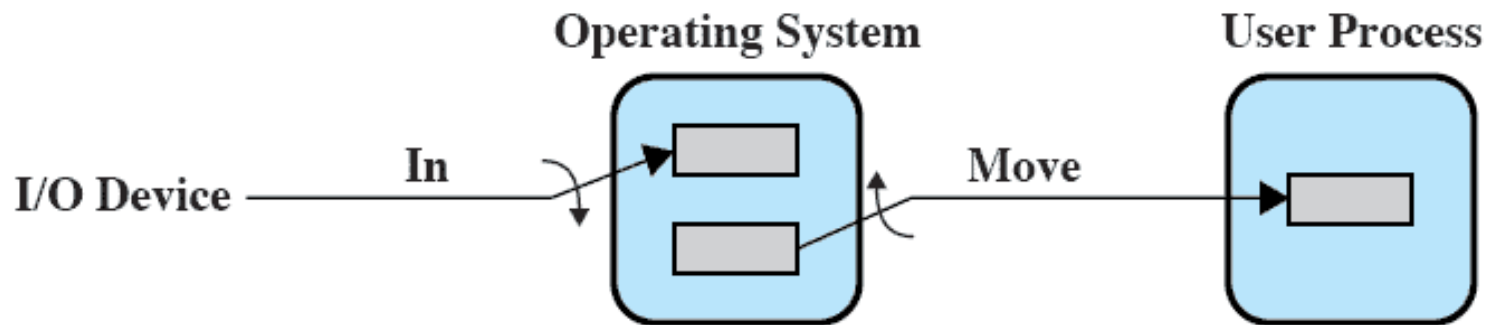


(b) Single buffering

- Blok orijentisani i tok orijentisani pojedinačni bafer

Dvostruki bafer

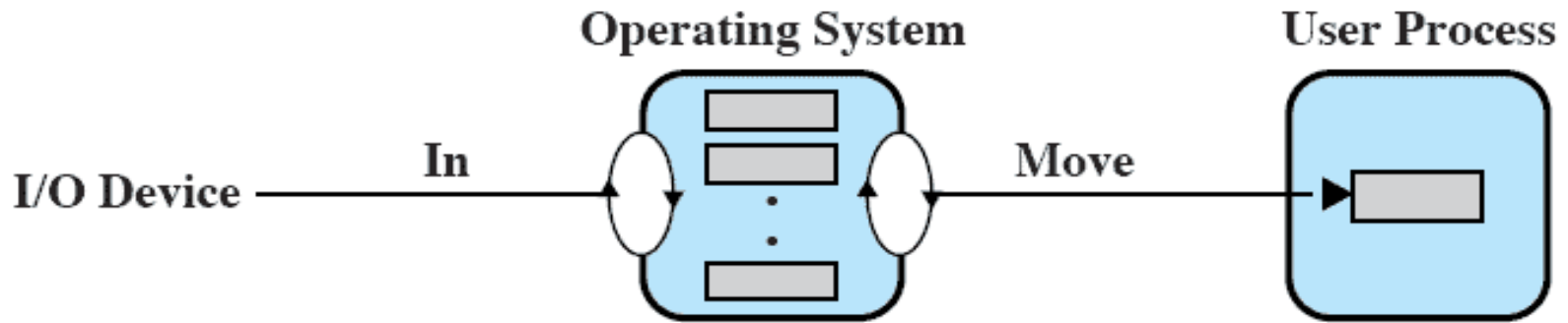
- Proces može prenesti podatke u ili iz jednog bafera dok operativni sistem prazni ili puni drugi bafer



(c) Double buffering

Cirkularni bafer

- ✿ Svaki bafer je jedinica u okviru cirkularnog bafera
 - ✦ Proizvođač-potrošač preko ograničenog kružnog bafera
 - ✦ Koristi se kada U/I operacija mora da uskladi brzinu sa procesom



(d) Circular buffering

Parametri performansi diska

- ✱ Da bi se nešto pročitalo ili zapisalo na disk potrebno je glavu diska pozicionirati na željenu trasu i na početak željenog sektora
- ✱ Vreme potrebno za čitanje i upis bloka određuju 3 parametra
 1. Vreme traženja (*Seek time*)
 - Vreme potrebno za pozicioniranje glave diska na željenu trasu
 2. Rotaciono kašnjenje (*Rotational delay* ili *latency*)
 - Vreme potrebno da bi se rotiranjem diska traženi sektor našao ispod glave diska
 3. Vreme prenosa (*Transfer time*)
 - Prenos se odvija tako što se sektor pomera iznad glave diska

Vreme pristupa (*access time*) = Vreme traženja + Rotaciono kašnjenje + Vreme prenosa

- ✱ Vreme prenosa celog fajla je brže kada je ceo fajl zapisan na istom cilindru i u susednim sektorima
- ✱ Kontrolu grešaka radi kontroler diska

Vreme disk transfera

⊕ Vreme transfera podataka sa diska



⊕ $T_a = T_s + 1/2r + b/rN$

- ⊗ r – rotaciona brzina diska u obrtajima u sekundi
- ⊗ b - broj bajtova koji se prenose
- ⊗ N – broj bajtova na stazi diska

Strategije raspoređivanja diska (1)

- ✿ Vreme traženja je razlog za razlike u performansama
- ✿ Za jedan disk postoji više U/I zahteva
- ✿ Ako se zahtevi biraju slučajno, to može dovesti do loših performansi diska
- ✿ Moguće strategije (politike) raspoređivanja
 - ✦ FIFO ili FCFS (First Come First Served)
 - ✦ Prioritet
 - ✦ LIFO
 - ✦ SSF (Shortest Seek First) ili SSTF (short service time first)
 - ✦ SCAN ili elevator
 - ✦ C-SCAN
 - ✦ N-step-SCAN
 - ✦ FSCAN

Strategije raspoređivanja diska (2)

- ✿ FIFO (prvi došao prvi uslužen)
 - ✦ Zahtevi se obrađuju sekvencijalno
 - ✦ Fer za sve procese
- ✿ Prioritet (prvo zahtev najvišeg prioriteta)
 - ✦ Cilj nije optimizacija korišćenja diska već ispunjenje nekih drugih ciljeva
 - ✦ Kraći batch poslovi mogu imati prioritet
 - ✦ Nudi dobro vreme odgovora kod interaktivnog režima
- ✿ LIFO (zadnji došao prvi uslužen)
 - ✦ Dobar za sisteme za obradu transakcija
 - Uređaj se daje poslednjem korisniku tako da se glava malo pomera
 - ✦ Moguće izgladnjivanje

Strategije raspoređivanja diska (3)

- ✿ SSF (Shortest Seek First) ili SSTF (Shortest Seek Time First)
 - ✦ Bira se U/I zahtev koji zahteva najmanje kretanje glave diska sa njegove trenutne pozicije
 - ✦ Uvek se bira minimalno vreme traženja
- ✿ SCAN ili elevator
 - ✦ Glava se pomera samo u jednom pravcu, zadovoljavajući zahteve sve dok ne dostigne poslednju trasu u tom pravcu
 - ✦ Tada se glava okreće i pomera se u obrnutom pravcu, nastavljajući da uslužuje zahteve na koje naiđe
- ✿ C-SCAN
 - ✦ Ograničava se skeniranje samo na jedan pravac
 - ✦ Kada se dostigne zadnja trasa u jednom pravcu ili kada više nema zahteva u tom pravcu, glava se vraća na drugi kraj diska i tada ponovo vrši skeniranje
- ✿ N-step-SCAN
 - ✦ Segmentira se red zahteva za diskom u podredove dužine N
 - ✦ Podredovi se obrađuju istovremeno korišćenjem SCAN
- ✿ FSCAN
 - ✦ Dva reda; jedan red je prazan za novi zahtev

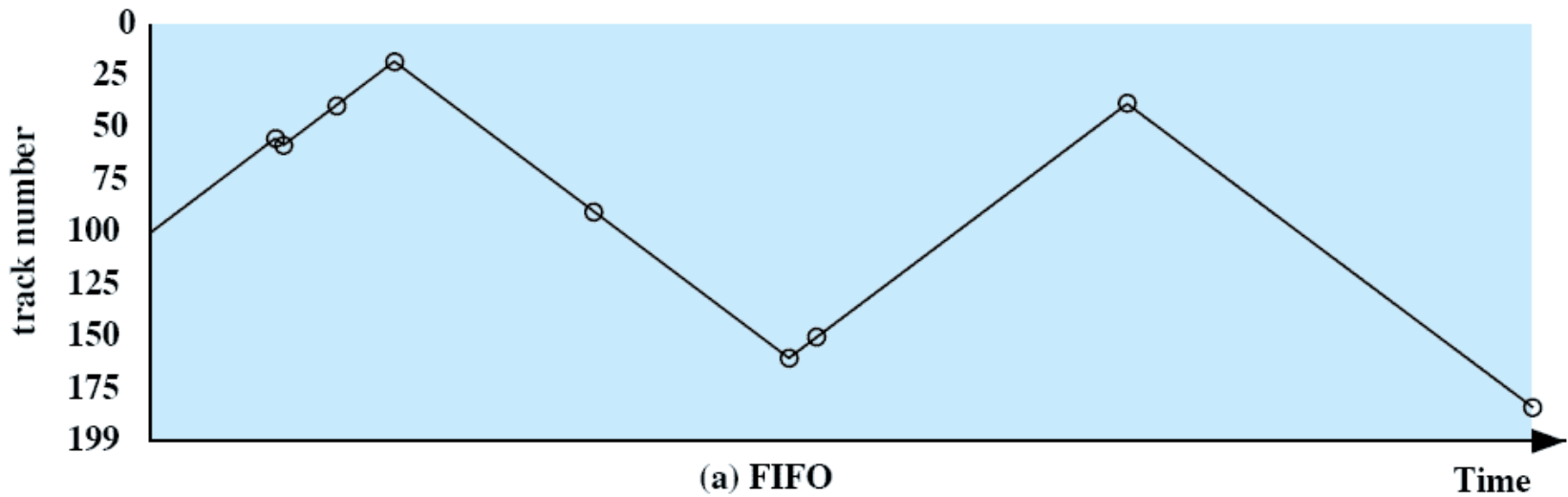


Algoritmi za planiranje diska

- ✿ Primer:
- ✿ Disk ima 200 staza
- ✿ Glava diska je inicijalno na stazi 100
- ✿ U redu zahteva su pristigli zahtevi za podacima na određenim stazama diska
 - ✦ 55, 58, 39, 18, 90, 160, 150, 38, 184

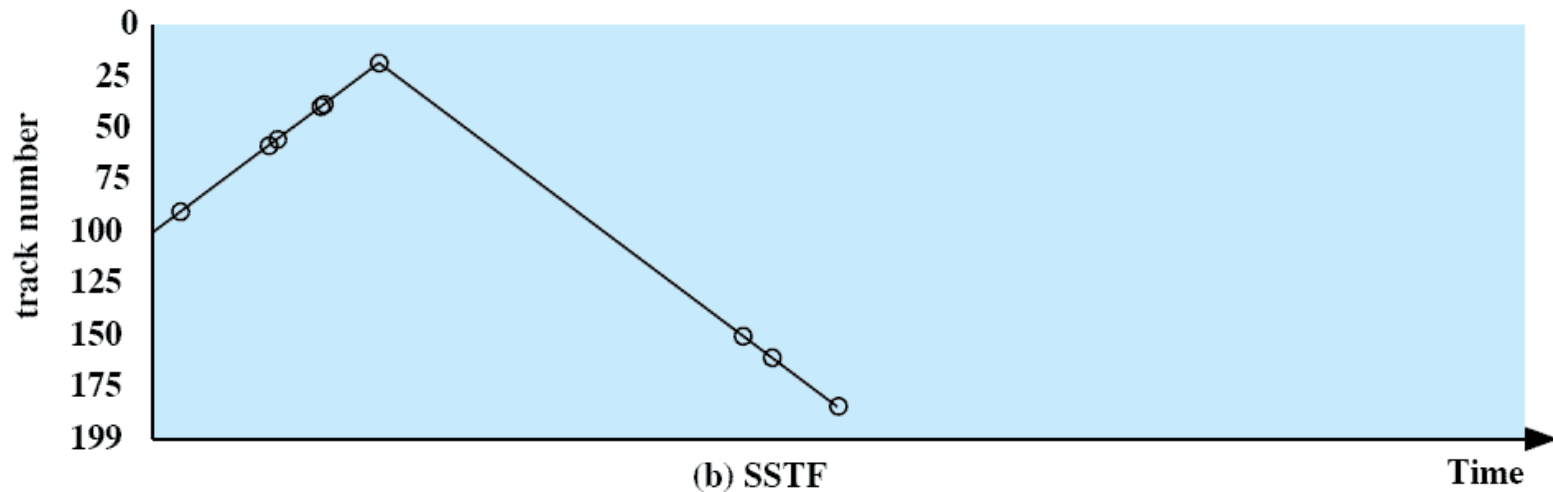


FIFO

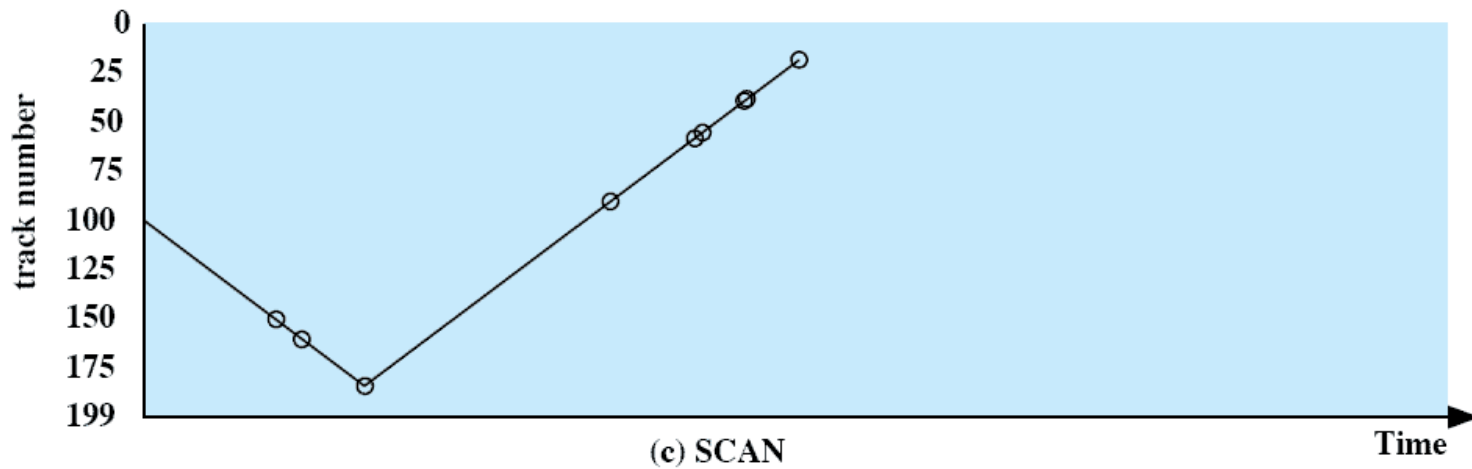




Shortest Service Time First

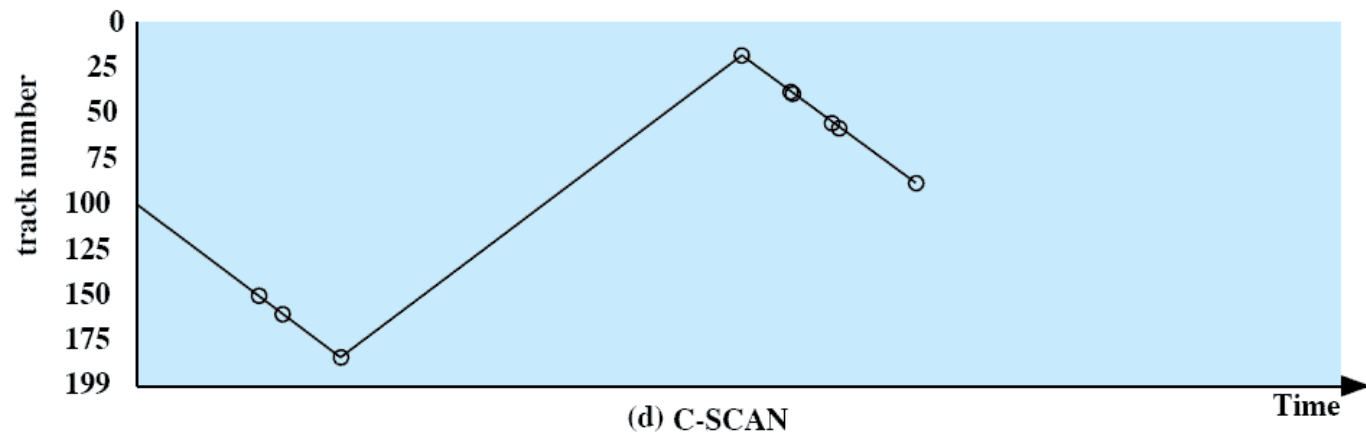


SCAN





C-SCAN



Uporedni pregled performansi

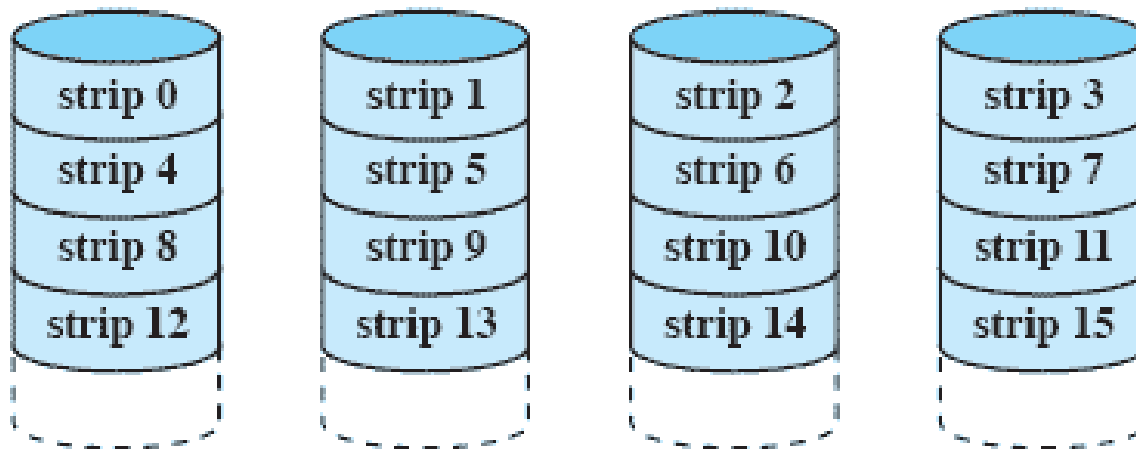
(a) FIFO (starting at track 100)		(b) SSTF (starting at track 100)		(c) SCAN (starting at track 100, in the direction of increasing track number)		(d) C-SCAN (starting at track 100, in the direction of increasing track number)	
Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed	Next track accessed	Number of tracks traversed
55	45	90	10	150	50	150	50
58	3	58	32	160	10	160	10
39	19	55	3	184	24	184	24
18	21	39	16	90	94	18	166
90	72	38	1	58	32	38	20
160	70	18	20	55	3	39	1
150	10	150	132	39	16	55	16
38	112	160	10	38	1	58	3
184	146	184	24	18	20	90	32
Average seek length	55.3	Average seek length	27.5	Average seek length	27.8	Average seek length	35.8

RAID

- ✿ *Redundant Array of Independent Disks*
- ✿ Skup fizičkih disk drajevova koje operativni sistem vidi kao jedan logički drajv
- ✿ Podaci su distribuirani po fizičkim drajvovima
- ✿ Redundantnost se koristi za smeštanje informacija o parnosti (kontrola i ispravka greške) čime se omogućuje oporavak nakon greške ili otkaza diska

RAID 0 – Podela u strip-ove

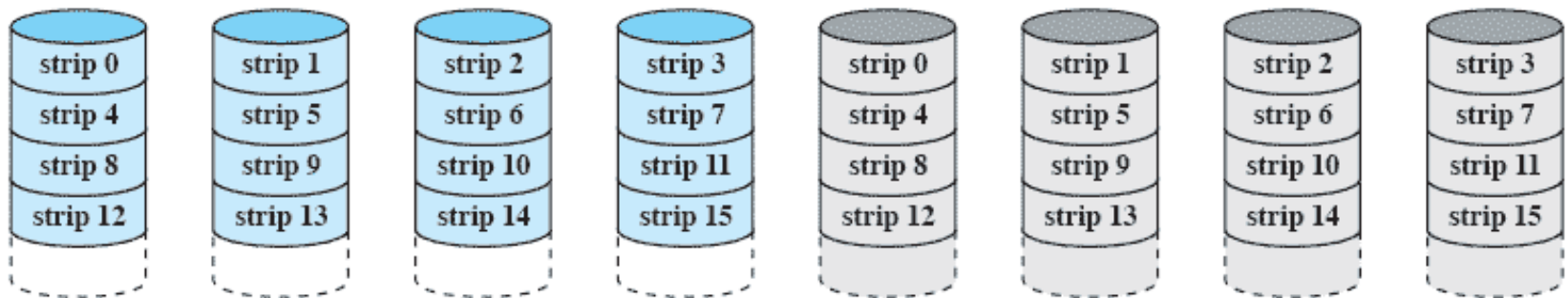
- ✚ Nije potpuni RAID – nema redundantnosti
- ✚ Otkaz diska ne može da se ispravi
- ✚ Veoma brz usled paralelnog čitanja/upisa



(a) RAID 0 (non-redundant)

RAID 1 - Mirroring

- ✿ Redundantnost kroz dupliciranje umesto kroz parnost
- ✿ Zahtev za čitanjem može biti obavljen paralelno
- ✿ Jednostavan oporavak usled otkaza diska

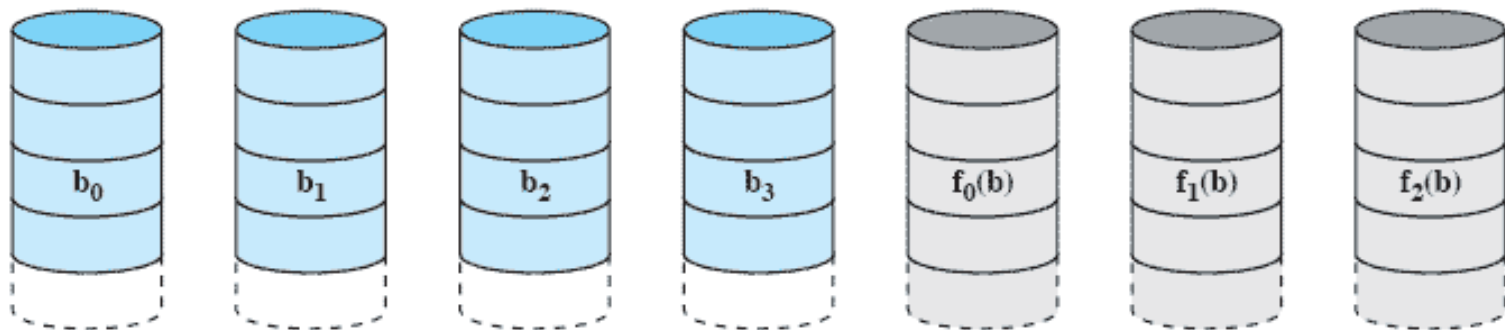


(b) RAID 1 (mirrored)

RAID 2

(Korišćenje Hamming koda)

- ✿ Sinhronizovano rotiranje diska
- ✿ Podela podatak (stripping) se koristi u ekstremno malim jedinicama (bajt, reč)
- ✿ Hamming kod se koristi za korigovanje grešaka na jednom bitu i detektovanje grešaka na dva bita

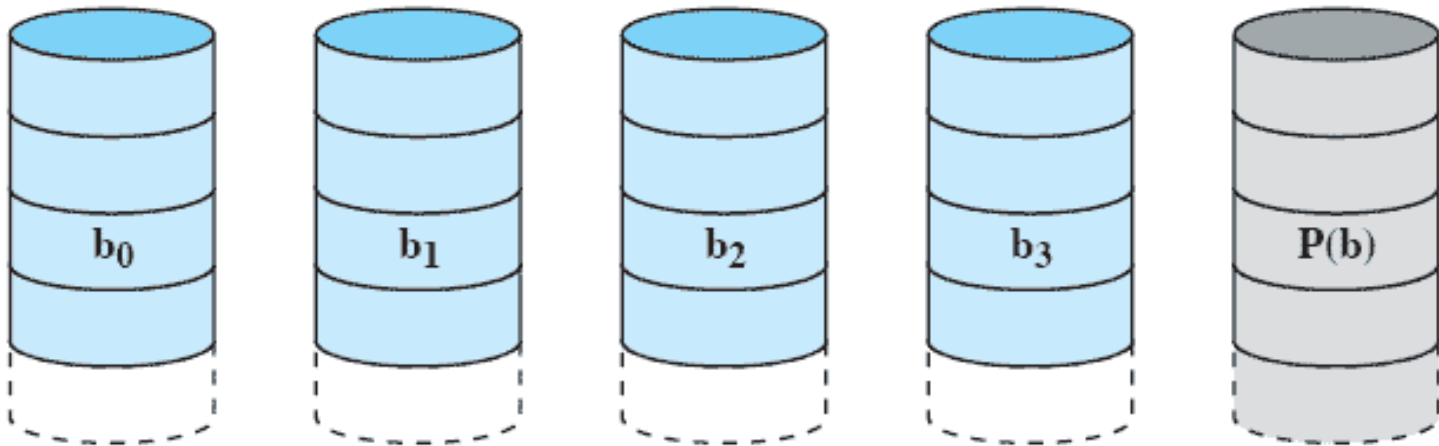


(c) RAID 2 (redundancy through Hamming code)

RAID 3

Bit parnosti

- Slično RAID-2 ali koristi bitove parnosti koji su smešteni na posebnom disku

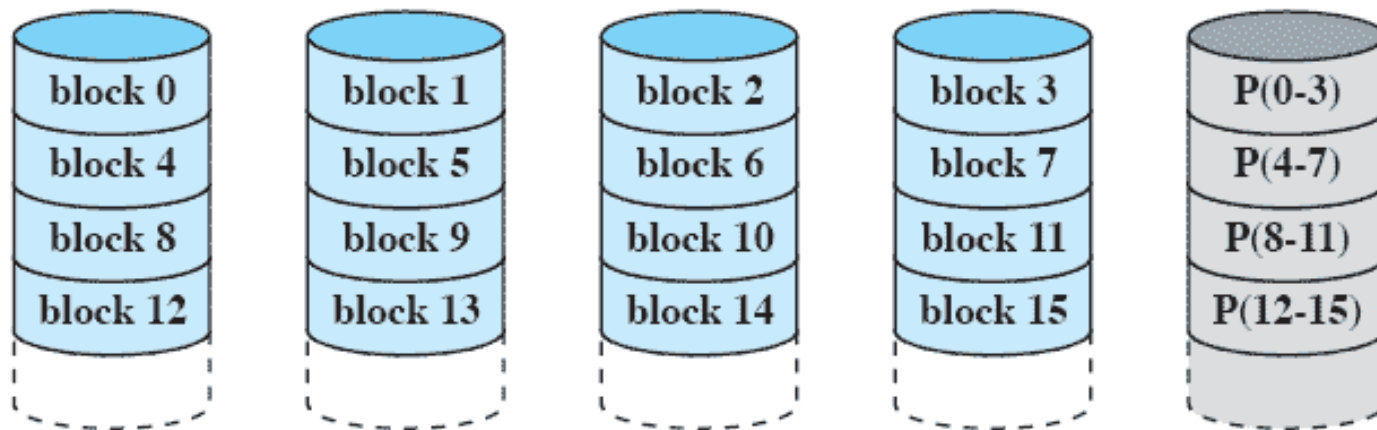


(d) RAID 3 (bit-interleaved parity)

RAID 4

Parnost na nivou bloka

- ✿ Parnost je izračunata nad odgovarajućim bitom po različitim blokovima i smeštena na poseban disk u odgovarajućem bloku

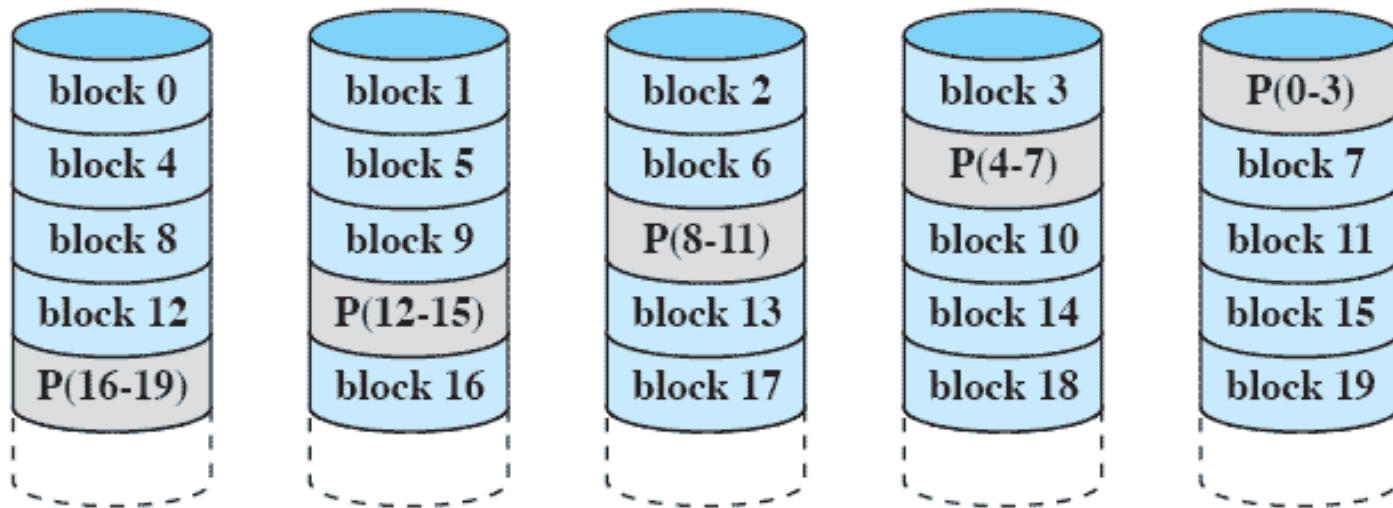


(e) RAID 4 (block-level parity)

RAID 5

Distribuirana parnost na nivou bloka

- Slično RAID-4 ali se bitovi parnosti distribuiraju po svim diskovima

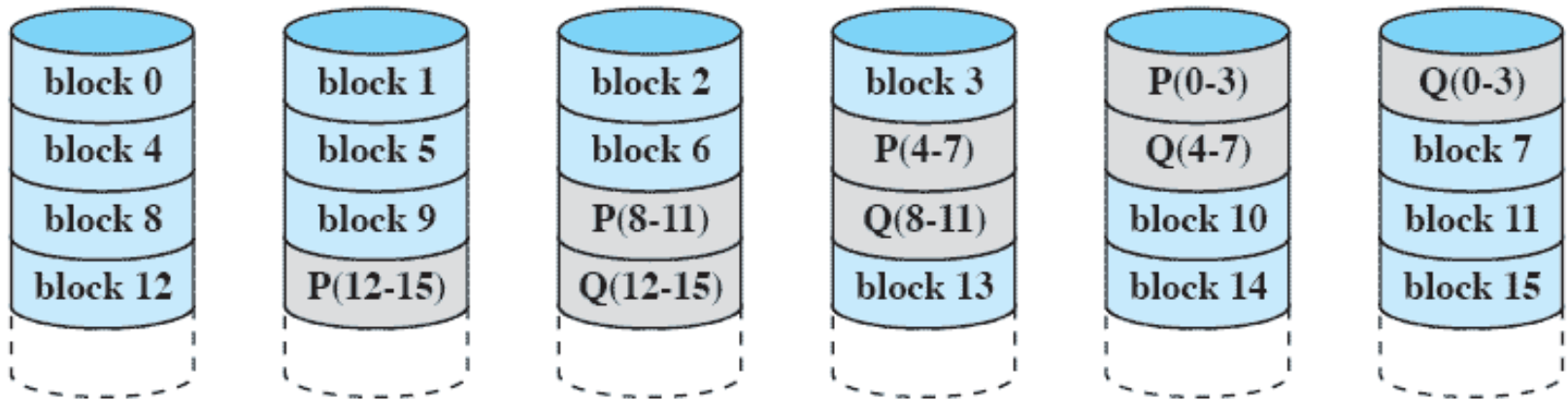


(f) RAID 5 (block-level distributed parity)

RAID 6

Dualna redundantnost

- Obavljaju se dva različita izračunavanja parnosti i smeštaju u posebne blokove na različitim diskovima
- Može se oporaviti od otkaza dva diska



(g) RAID 6 (dual redundancy)

Disk keš

- ✿ **Keš** je bafer u glavnoj memoriji gde se čuvaju kopije podataka sa diska (blokova diska)
 - ✦ Pristup kopijama podataka u kešu je mnogo efikasniji nego originalnim podacima
- ✿ **Primer U/I diska:**
 - ✦ Kad stigne U/I zahtev, OS prvo proverava da li su traženi podaci u kešu
 - ✦ Ako jesu, procesu se prosleđuju podaci iz memorije, bez pristupa disku
 - ✦ Ako nisu u kešu, OS ih učitava sa diska
 - ✦ Slično je i kod upisa na disk - U kešu se skupljaju podaci pre upisa na disk
- ✿ **Strategije zamene blokova u kešu**
 - ✦ LFU - *Least Frequently Used*
 - ✦ LRU - *Least Recently Used*

UNIX SVR4 I/O

- ✿ Svakom U/I uređaju je pridružena specijalna datoteka
 - ✦ Obezbeđuje jasan i uniforman interfejs ka uređajima za korisnike i procese
- ✿ Za pristup uređaju izdaju se zahtevi za čitanje i upis (*read*, *write*) na specijalnu datoteku
- ✿ Dva tipa U/I: baferovan i nebaferovan

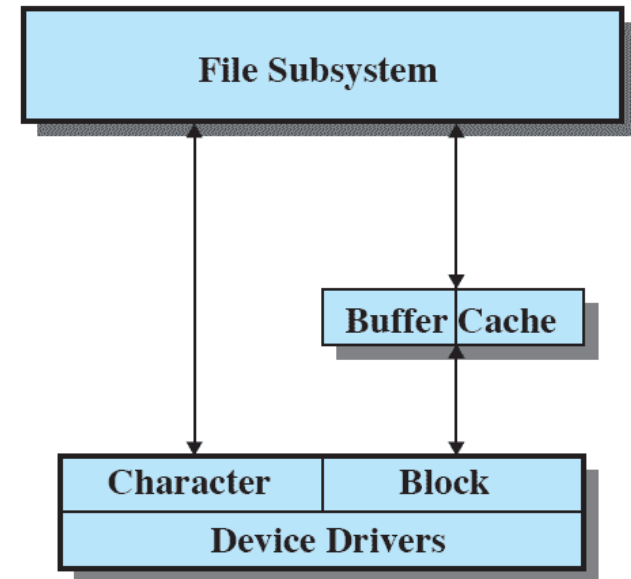


Figure 11.12 UNIX I/O Structure

Keširanje

Keš bafer - održavaju se tri liste

- ❑ Lista slobodnih
- ❑ Lista uređaja
- ❑ U/I red drajvera

Keš znakova

- ❑ Koristi se za znakovno orijentisane uređaje, terminale, štampače

Nebaferovan U/I - DMA

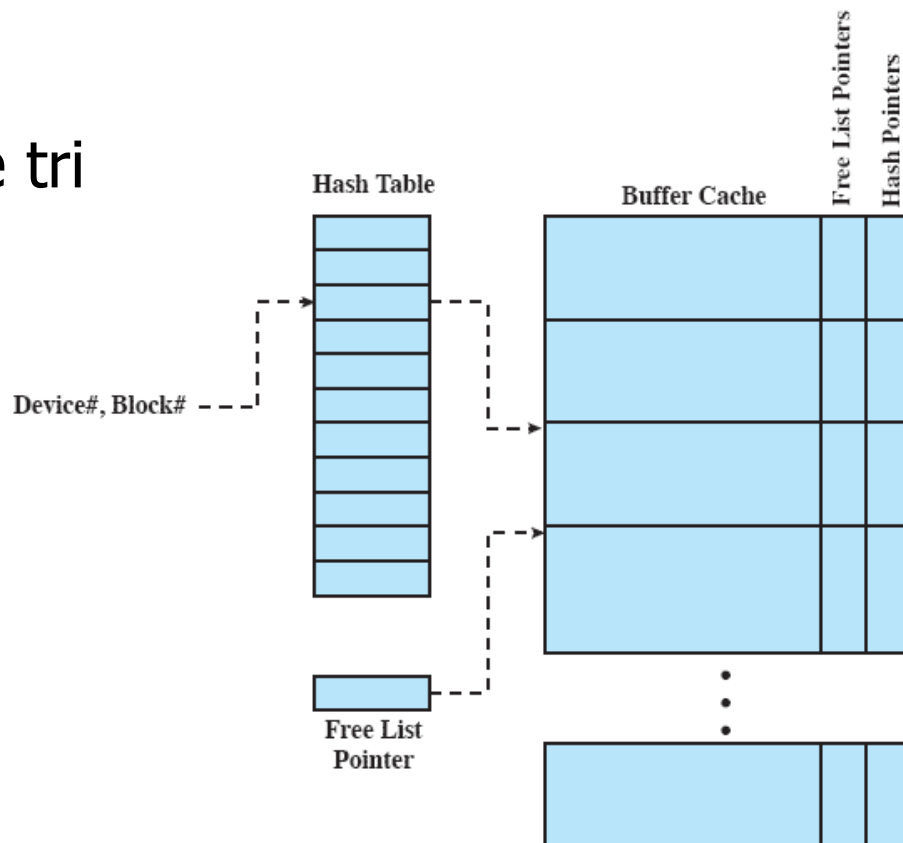


Figure 11.13 UNIX Buffer Cache Organization

Linux U/I

- Linux i UNIX imaju veoma slične mehanizme za upravljanje U/I
 - Linux kernel dodeljuje specijalnu datoteku svakom U/I drajveru uređaja
 - Razlikuju se blok, znakovni i mrežni uređaji
- Koristi Linux Elevator planer za planiranje diska – varijanta LOOK (SCAN) algoritma
 - U Linux 2.6 proširen sa dva dodatna algoritma: *Deadline* U/I planer (*Raspoređivač po roku*) i *anticipatory* U/I planer (*Raspoređivač sa predviđanjem*)
- Održava jedinstveni *page* keš za sav saobraćaj između diska i glavne memorije

Windows U/I

- Windows I/O manager - odgovaran za U/I u okviru Windows OS
- Obezbeđuje jedinstven interfejs za sve tipove drajvera uređaja
- Sastoji se od 4 modula
- Dva moda U/I operacija
 - Sinhroni
 - Asinhroni
- Podržani su RAID-0, 1, 5, 6

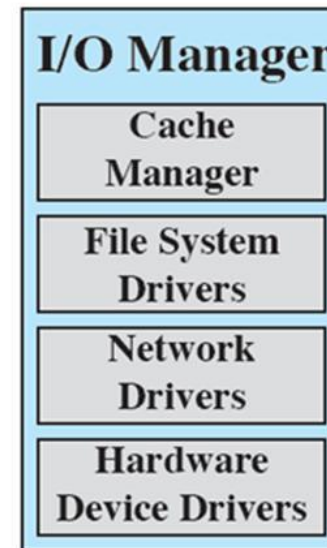


Figure 11.15 Windows I/O Manager



Domaći zadatak

☛ Poglavlje **11 Upravljanje U/I i planiranje diska**

- ☛ 11.13 Ključni pojmovi, kontrolna pitanja i problemi

☛ U/I animations

- ☛ <https://apps.uttyler.edu/Rainwater/COSC3355/Animations>
- ☛ Interrupt-Driven I/O Cycle
- ☛ The Life Cycle of an I/O Request
- ☛ Disk Scheduling Algorithms: FCFS, SSTF, SCAN