



# OPERACIJSKI SUSTAVI

Ponavljjanje 1.4,5,6

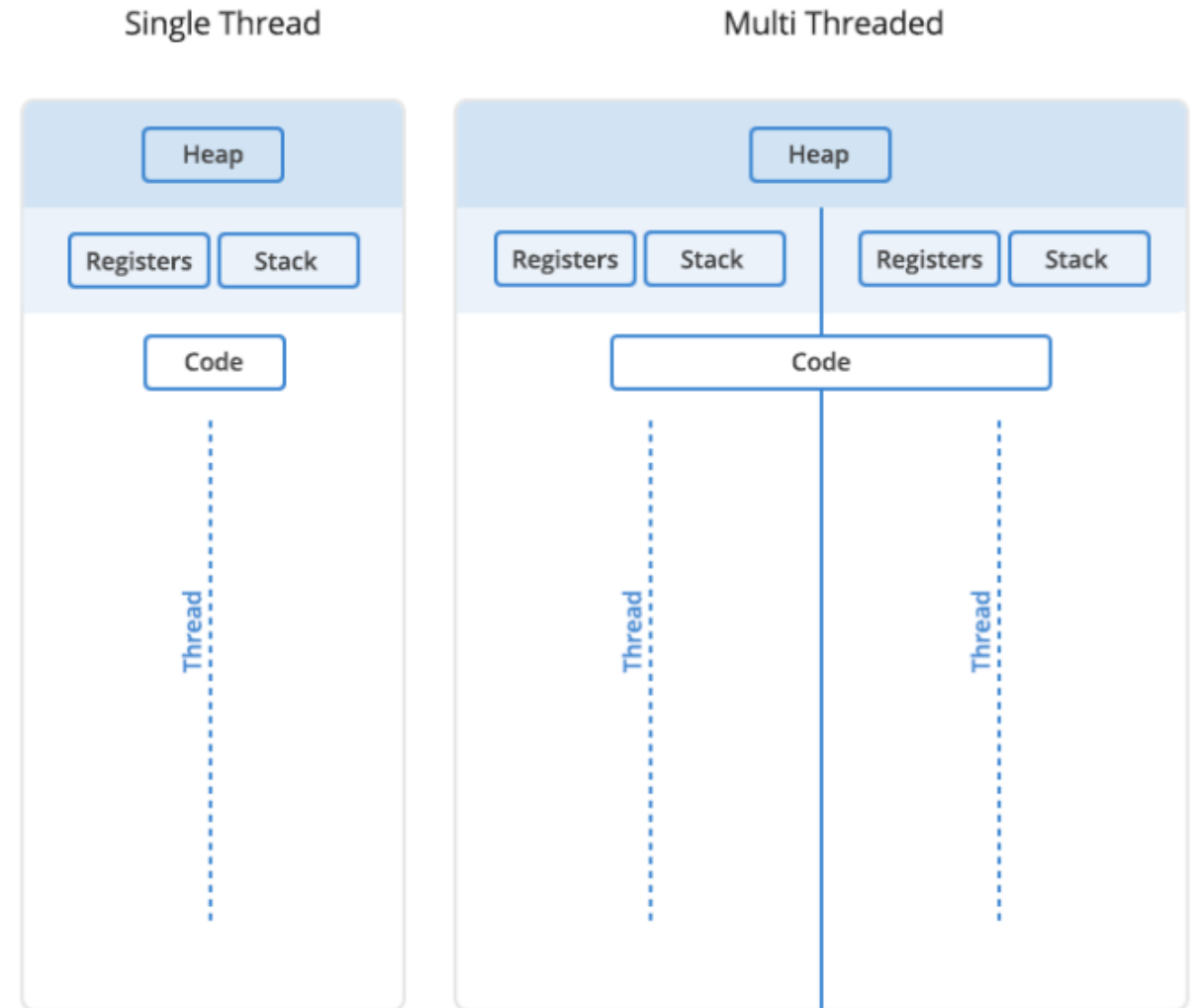
Memorija

Disk

Ugrađeni operacijski sustavi

# Podsjetnik: Što je process!

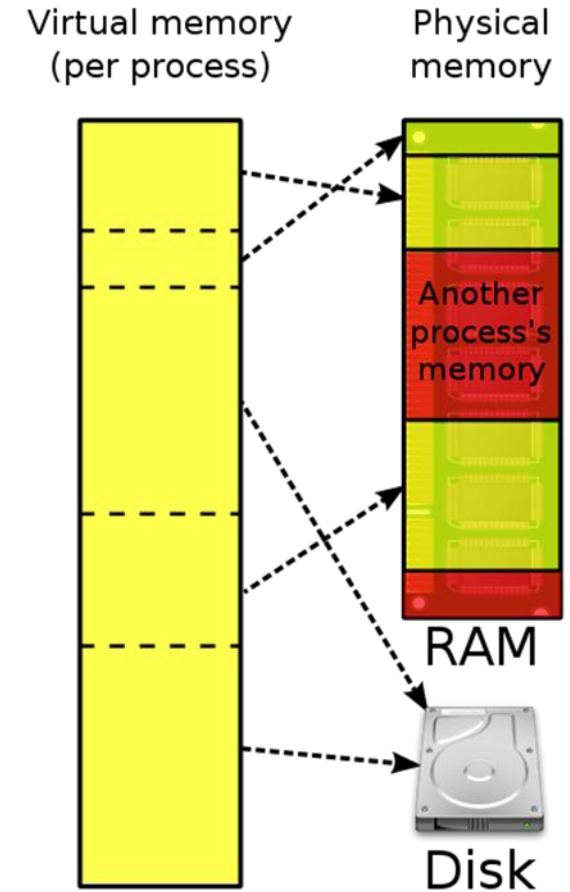
- Heap/Varijabilni dio memorije procesa dijele ga sve dretve (threads)
- Svaka dretva ima svoj registar i stog
- Svaka dretva (*treba je znati napisati*) ima svoj dio koda koji je zajednički unutar procesa
- Svaki proces ima svoj identitet – PID (Process ID)
- Svaka dretva ima svoji ID unutar svog procesa
- Proces se izvršava unutar **jednog** CPUa (ne može se dijeliti na više CPU)
- **Hyper-threading** (1 fizički CPU podijeljen u više virtualnih jezgri)
  - Hyper-threading – OS „vidi” više CPUova



# Ishod 4: Memorija i strainčenje

# Virtual memory

- Operativni sustav **rezervira** dio memorije za svaki proces
- Operativni sustav **mapira** memorijske adrese (virtualne adrese) koje aplikacija koristi na fizičke adrese
- Operacijski sustav grupirat će podatke u blokove – koji se nazivaju **stranica**
- Operativni sustav će uravnotežiti ono što se nalazi u RAM-u pomoću **algoritama** zamjene stranica
- Blok više stranica u RAM-u je **segment** (ili okvir ili **stranice**)

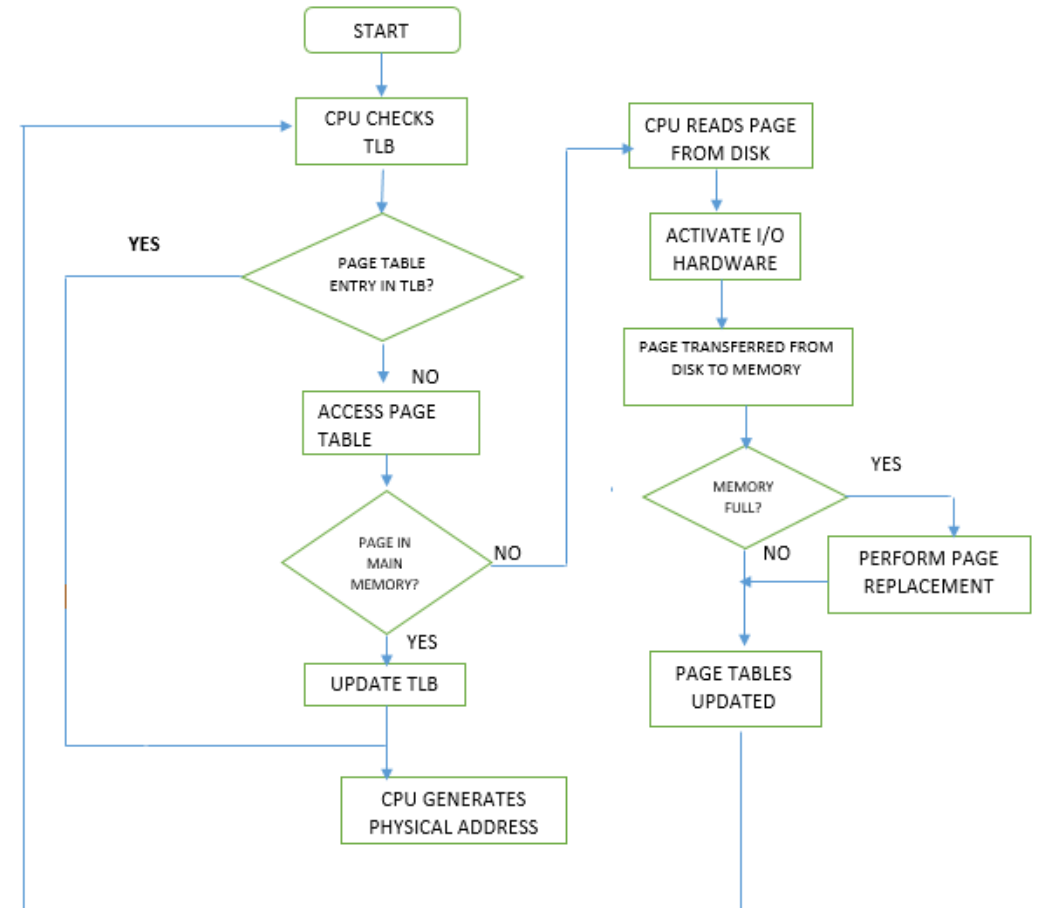


# Tehnike upravljanja memorijom

- **Fiksno particioniranje**
  - Glavna memorija podijeljena je na brojne **statičke** particije u vrijeme stvaranja sustava.
- **Dinamičko particioniranje**
  - Particije se stvaraju dinamički, tako da se svaki proces učitava u particiju potpuno **iste** veličine kao i taj proces.
- **Jednostavna stranična stranica**
  - Glavna memorija podijeljena je u nekoliko okvira **jednake** veličine.
- **Jednostavna segmentacija**
  - Svaki proces je podijeljen u niz segmenata. Proces preuzima podatke učitavanjem svih njegovih segmenata u **dinamičke** particije koje ne moraju biti susjedne.
- **Stranično povezivanje s virtualnom memorijom**
  - Kao i kod jednostavne stranične stranice, osim što nije potrebno učitavati sve stranice procesa.
  - Nerezidentne stranice koje su potrebne automatski se unose kasnije.
- **Segmentacija virtualne memorije**
  - Kao i kod jednostavne segmentacije, osim što nije potrebno učitavati sve segmente procesa.
  - Nerezidentni segmenti koji su potrebni automatski se unose kasnije.

# Međuspremnik prevođenja (TLB) Translation Lookaside Buffer

- Cache – predmemorija
  - Možda ćemo nešto trebati staviti u RAM, pa zašto da tražim u cijeloj virtualnoj memoriji



# Zadatak 1. FIFO

- Stranice se postavljaju kao vezani popis; stranica se uklanja s kraja dok je nova stranica postavljena na početku
- Zadan je sljedeći referentni niz: 0,2,1,1,6,4,0,1,0,3,1,2,1, 1, broj referenci je 12.
  - Rezervirane su 3 stranice/segmenta u memoriji
  - Rezervirane su 4 stranice/segmenta u memoriji

# Rješenje 1

- 1.a (3 segmenta)

Korak Step	Ulaz IN	Što/What? Hit/Miss	Izlaz OUT	Memorija RAM		
1	0	Miss		0		
2	2	Miss		0		
3	1	Miss		0	1	
4	6	Miss		0	1	6
5	4	Miss	0	2	6	4
6	0	Miss	2	6	4	0
7	1	Miss	6	4	0	1
8	0	Hit		4	0	1
9	3	Miss	4	0	1	3
10	1	Hit		0	1	3
11	2	Miss	0	1	3	2
12	1	Hit		1	3	2

Broj učitavanja: 3  
Broj pogreški: 9 (75%)

- 1.b (4 segmenta)

Korak Step	Ulaz IN	Što/What? Hit/Miss	Izlaz OUT	Memorija RAM			
1	0	Miss		0			
2	2	Miss		0	2		
3	1	Miss		0	2	1	
4	6	Miss		0	2	1	6
5	4	Miss	0	2	1	6	4
6	0	Miss	2	1	6	4	0
7	1	Hit		1	6	4	0
8	0	Hit		1	6	4	0
9	3	Miss	1	6	4	0	3
10	1	Miss	6	4	0	3	1
11	2	Miss	4	0	3	1	2
12	1	Hit		0	3	1	2

Broj učitavanja: 3  
Broj pogreški: 9 (75%)



# LRU vs. LFU

- LRU – Least recently used
  - Zamjeni se stranica kojoj najduže vrijeme nije pristupano u prošlosti
  - *“Vjerojatno ako nešto nismo dugo koristili nam neće trebati sutra”*
- LFU – Least Frequently used
  - Zamjeni se stranica koja se najrijeđe koristila u prošlosti (*“stranica koja je u prošlosti imala najmanje pregleda/učitavanja”*)

# Zadatak 2. LRU/LFU

- Niz: 3, 2, 1, 0, 3, 2, 4, 3, 2, 1, 0, 4, 3, 2
- Prikažite sva stanja straničenja LRU algoritma ako je kapacitet memorije:
  - a) 3.
  - b) 4
- Prikažite sva stanja straničenja LFU algoritma ako je kapacitet memorije:
  - a) 3
  - b) 4

# Rješenje 2a,b

- 2a (LRU, s=3)

Korak Step	Ulaz IN	Što/What? Hit/Miss	Izlaz OUT	Memorija RAM		
1	3	Miss		3		
2	2	Miss		3	2	
3	1	Miss		3	2	1
4	0	Miss	3	0	2	1
5	3	Miss	2	0	3	1
6	2	Miss	1	0	3	2
7	4	Miss	0	4	3	2
8	3	Hit		4	3	2
9	2	Hit		4	3	2
10	1	Miss	4	1	3	2
11	0	Miss	3	1	0	2
12	4	Miss	2	1	0	4
13	3	Miss	1	3	0	4
14	2	Miss	0	3	2	4

Number of Hits: 2  
Number of Miss: 12 (86%)

- 2b (LRU, s=4)

Korak Step	Ulaz IN	Što/What? Hit/Miss	Izlaz OUT	Memorija RAM			
1	3	Miss		3			
2	2	Miss		3	2		
3	1	Miss		3	2	1	
4	0	Miss		3	2	1	0
5	3	Hit		3	2	1	0
6	2	Hit		3	2	1	0
7	4	Miss	1	3	2	4	0
8	3	Hit		3	2	4	0
9	2	Hit		3	2	4	0
10	1	Miss	0	3	2	4	1
11	0	Miss	4	3	2	0	1
12	4	Miss	3	4	2	0	1
13	3	Miss	2	4	3	0	1
14	2	Miss	1	4	3	0	2

Number of Hits: 4  
Number of Miss: 10 (71%)

# Rješenje 2c,d

- 2c (LFU, s=3)

Korak Step	Ulaz IN	Što/What? Hit/Miss	Izlaz OUT	Memorija RAM		
1	3	miss		3		
2	2	miss		3	2	
3	1	miss		3	2	1
4	0	miss	3	0	2	1
5	3	miss	2	0	3	1
6	2	miss	1	0	3	2
7	4	miss	0	4	3	2
8	3	hit		4	3	2
9	2	hit		4	3	2
10	1	miss	4	1	3	2
11	0	miss	3	1	0	2
12	4	miss	2	1	0	4
13	3	miss	1	3	0	4
14	2	miss	0	3	2	4

Number of Hits: 2  
Number of Miss: 12 (86%)

- 2d (LFU, s=4)

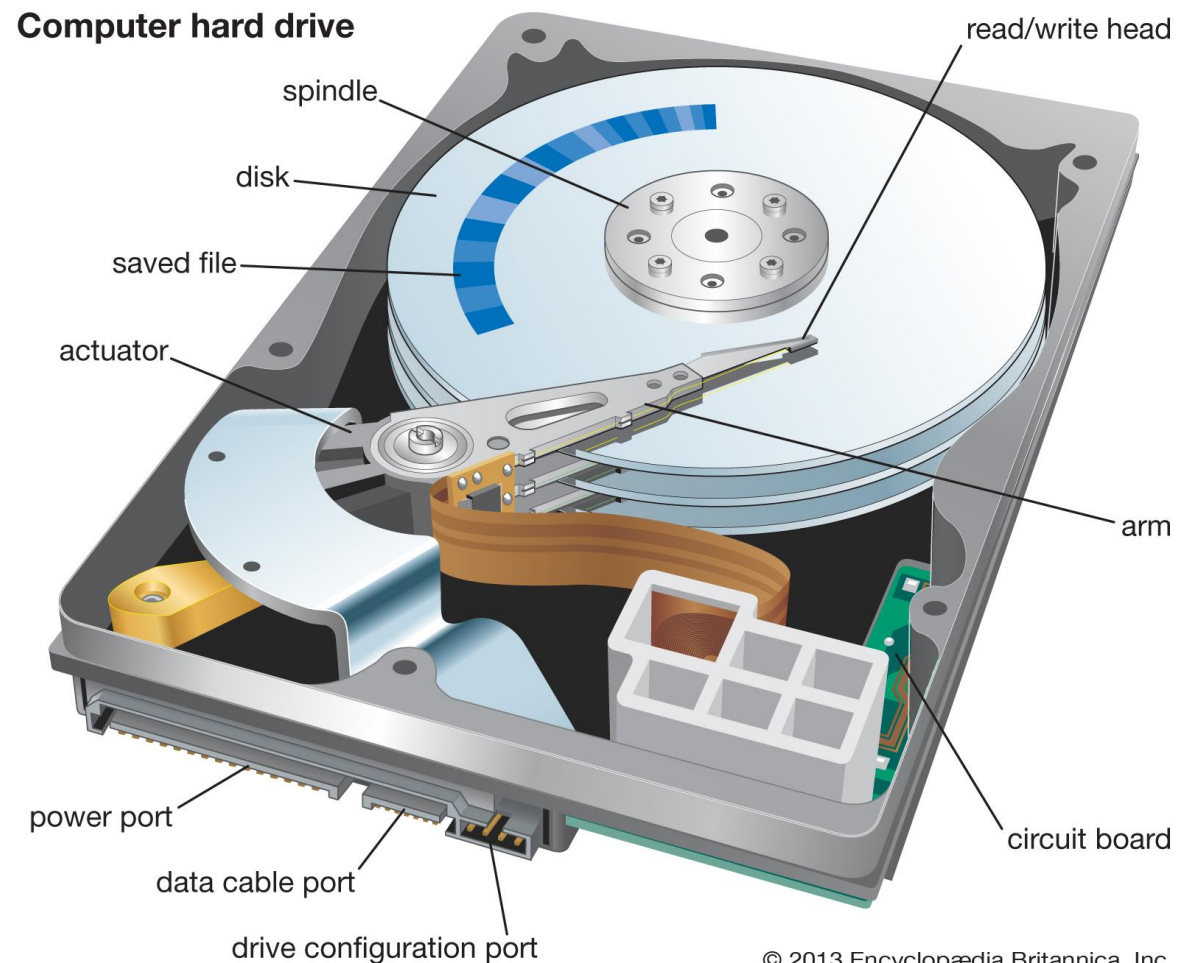
Korak Step	Ulaz IN	Što/What? Hit/Miss	Izlaz OUT	Memorija RAM			
1	3	Miss		3			
2	2	Miss		3	2		
3	1	Miss		3	2	1	
4	0	Miss		3	2	1	0
5	3	Hit		3	2	1	0
6	2	Hit		3	2	1	0
7	4	Miss	1	3	2	4	0
8	3	Hit		3	2	4	0
9	2	Hit		3	2	4	0
10	1	Miss	0	3	2	4	1
11	0	Miss	4	3	2	0	1
12	4	Miss	3	4	2	0	1
13	3	Miss	0	4	2	3	1
14	2	Hit		4	2	3	1

Number of Hits: 5  
Number of Miss: 9 (64%)

# Ishod 5: Disk

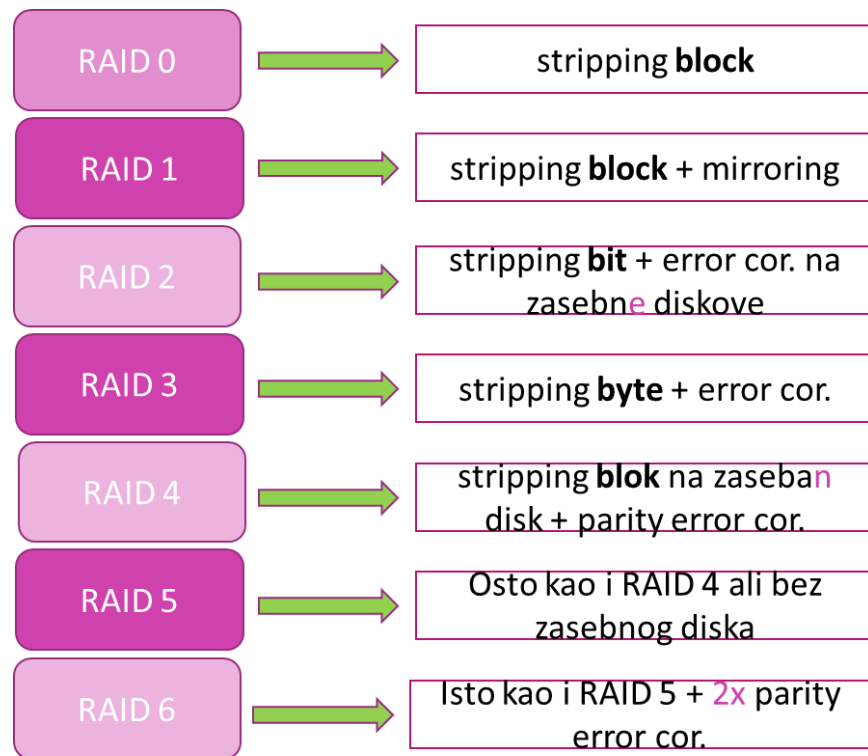
# Djelovi diska

- Jedna ili više okruglih ploča presvučenih magnetskim materijalom
  - koje se vrte **konstantnom brzinom** s mehanizmom magnetskih glava
    - koje se mogu pomicati približno radijalno iznad ploča
- Upravljačkog sklopa, spremnika, sučelja prema elektromehaničkom dijelu
- Sučelje prema sabirnici računala



© 2013 Encyclopædia Britannica, Inc.

# Usporedba



Raid	Min.Disks	Storage Efficiency %	Cost	Read Performance	Write Performance	Write Penalty
0	2	100	Low	Very good for both random and sequential read.	Very good	No
1	2	50	High	Good, better than a single disk.	Good, slower than a single disk, as every write must be committed to all disks.	Moderate
3	3	$(n-1)*100/n$ Where n = number of disks	Moderate	Good for random reads and very good for sequential reads.	Poor to fair for small random writes, good for large, sequential writes.	High
4	3	$(n-1)*100/n$ Where n = number of disks	Moderate	Very good for random reads. Good to very good for sequential writes.	Poor to fair random writes, fair to good for sequential writes.	High
5	3	$(n-1)*100/n$ Where n = number of disks	Moderate	Very good for random reads. Good for sequential reads.	Fair for random writes, slower due to parity overhead, fair to good for sequential writes.	High
6	4	$(n-2)*100/n$ Where n = number of disks	Moderate but more than RAID 5	Very good for random reads. Good for sequential reads.	Good for small, random writes (has write penalty).	Very high
1+0 and 0+1	4	50	High	Very good.	Good	Moderate

# Algoritmi raspoređivanje poslova diska:

- FCFS
- SSTF
- SCAN (dizalo)
- LOOK
  - (podvarijanta dizala koja ne ide do krajnjih sektora)
- C-SCAN (jednosmjerno dizalo)
- C-LOOK
  - (podvarijanta jednosmjernog dizala koja ne ide do krajnjih sektora)
- **Seek distance** – vrijeme pretraživanja
  - vrijeme potrebno za pomicanje ruke (*arm*) diska na određenu stazu gdje se podaci trebaju čitati ili pisati.
- Referentni string - Struktura podataka
  - lista brojeva sektora kojima različite zadaće pristupaju.



# Zašto koristimo algoritme?

- Višestruki I/O zahtjevi mogu stizati od strane različitih procesa i samo jedan I/O zahtjev može istovremeno poslužiti diskovni kontroler. Stoga drugi I/O zahtjevi moraju čekati u redu čekanja i trebaju biti zakazani.
- Dva ili više zahtjeva mogu biti udaljeni jedan od drugog pa mogu rezultirati većim pomakom kraka diska.
- Tvrdi diskovi su jedan od najsporijih dijelova računalnog sustava i stoga im je potrebno pristupati na učinkovit način.

# Zadatak

- Referentni niz: 1, 2, 3, 4, 12, 11, 12, 11, 10, 5, 35, 11, 10, 1, 35, 30, 29, 12, 13
- Prvi sektor je 0, zadnji sektor je sektor 50
- Glava diska je na sektoru: 12 i glava se kreće u lijevo (prema padajućem sektoru)
- Upotrijebiti algoritme:
  - A) FIFO
  - B) SSTF,
  - C) SCAN
  - D) LOOK
  - E) C-SCAN
  - F) C-LOOK
- Za svaki algoritam napraviti tablicu sa koracima i izračunati seek distance.

# Rješenja:

- A) FIFO = 147 ...
- B) SSTF = 51 ...
- C) SCAN = 87 ili 47
- D) LOOK = 72 ili 35
- E) C-SCAN = 99 ili 125
- F) C-LOOK = 67 ...

*Napomena: Možda su neka rješenja kriva!?*



# **Ishod 6: RTOS + algoritmi, Multimedijalni Operacijski sustavi, Ugrađeni Operacijski sustavi**

# Vrste Operacijskih sustava

Batch OS

Dizajnirani za učinkovitu obradu velikih količina podataka i automatizaciju zadataka koji se ponavljaju kao što su obrada plaća, generiranje izvješća i obrada podataka



mainframe

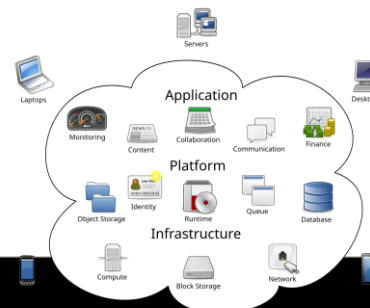
Multiprograming OS

Dopuštaju višestrukim programima da rade istovremeno dijeleći CPU i druge systemske resurse



Distributed OS

Upravljaju resursima i koordiniraju zadatke na više međusobno povezanih računala i čvorova.



Embeded OS

Računalni čipovi ugrađeni u veće uređaje ili strojeve



Potrošačka elektronika

Real Time OS

Koriste se u vremenski kritičnim okruženjima u kojima su važni pouzdanost i rokovi

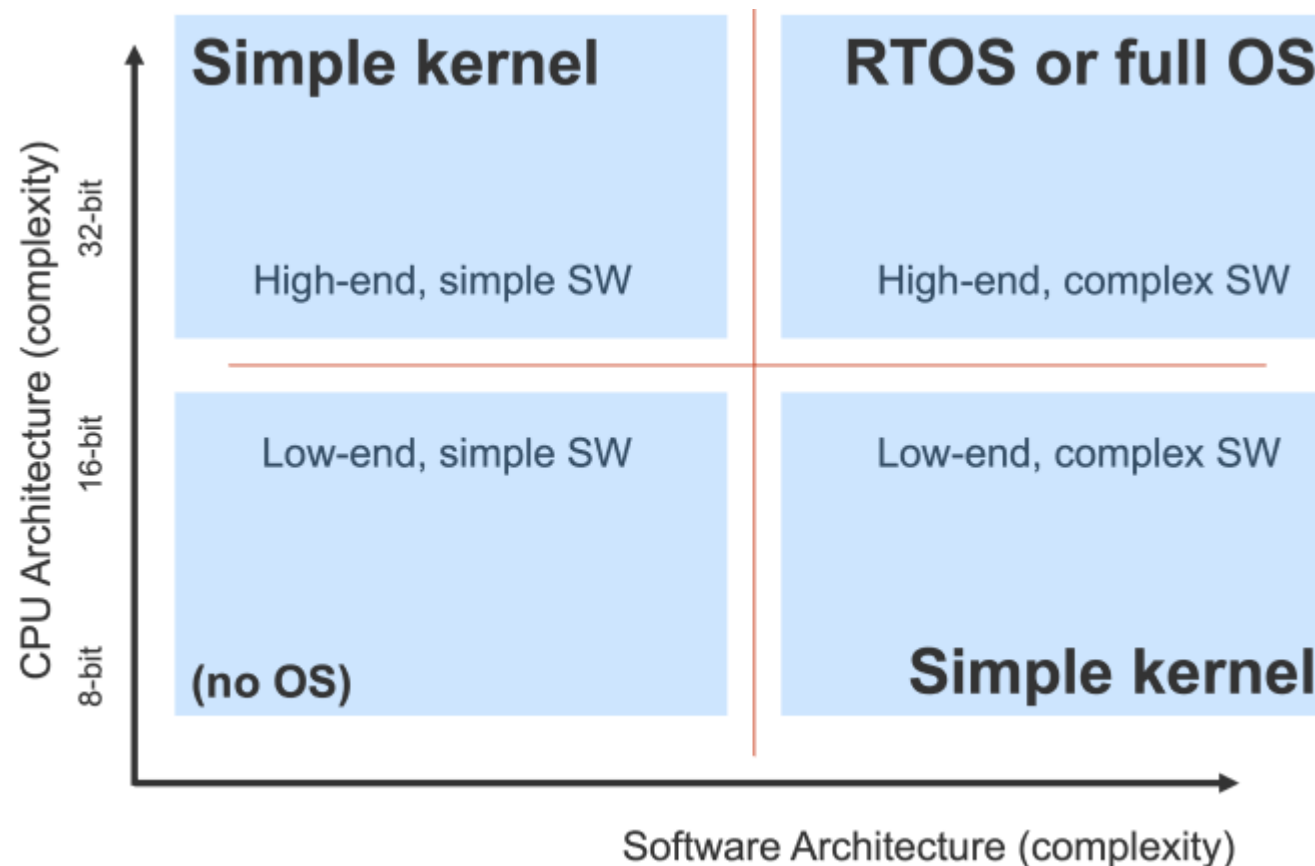


# Ugrađeni (**embedded**) operacijski sustavi

- Koriste upotrebu elektronike i softvera unutar proizvoda koji ima **određenu** funkciju ili skup funkcija, za razliku od računala opće namjene
- U većini slučajeva imaju interakciju sa korisnikom i senzorom
- Imaju svoj ili prenamijenjeni OS + svoju aplikaciju
- Npr. stroj za pranje suđa, robo-usisavač, automobil (radio, sustav za kontrolu motora, kočenja, izgaranja...), video kamera, dizalo/lift, dron, printer, blagajna u dućanu, bankomat, mrežni usmjeritelji/preklopnici...
- Primjeri ugrađenih OS:
  - $\mu$ Clinux
  - TinyOS
  - Windows CE
  - Symbian...

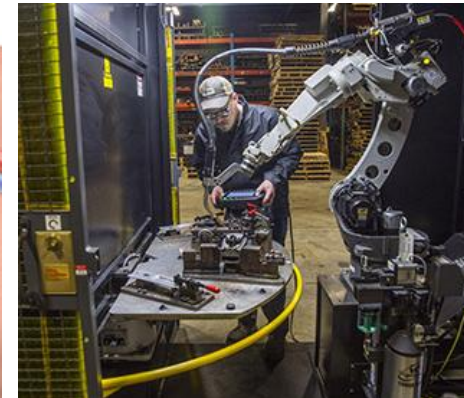
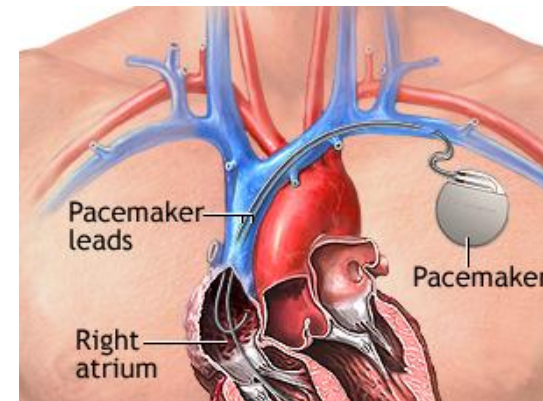
# Arhitektura tipova OSa

- Koji OS mi treba?
- Gdje mogu nabaviti OS koji mi treba?
  - **Napiši ga sam**
  - Koristi Open-source
  - Kupi



# Real-time operating system - ROTS

- Operativni sustav za *real-time* (stvarno-vremenske) aplikacije
- .hr stručni prijevod: „Multimedijski operacijski sustav”
- Primjeri:
  - Sustav za kontrolu leta (*Airline traffic control systems*)
  - Sustav za rezervaciju karata
  - Srčani stimulator srca (*pacemaker*)
  - Mrežni multimedijski sustavi (IPTV)
  - Roboti (npr. u auto industriji)
  - Vojni sustavi
  - ...



Izvor: Google



# Karakteristike

## Prednosti:

- Visoki stupanj iskorištenja
- Fokus na aplikaciji
- Error-free
- Alokacija memorije

## Nedostaci:

- Limitiran broj zadataka
- Zahtjevni hardware (\$\$\$\$)
- Koriste kompleksne algoritme
- Zahtijevaju zasebne driver-e (u assembleru)

Komercijalni OS: pSOS (Portable Software On Silicon), Versatile Real-Time Executive (VRTX), RT Linux, LynxOS...

# Zadatak 3.

- A – 10ms svakih 40ms
  - B – 5ms svakih 30ms
  - C – 5ms svakih 20ms
- 
- a) Izračunaj prioritete i izvedivost (1 bod od 15)
  - b) RMS algoritam (3 boda od 15)
  - c) EDF algoritam (4 boda od 15)

# Rješenje 3a:

- Rješivost:

- $\frac{10}{40} + \frac{5}{30} + \frac{5}{20} \leq 1$

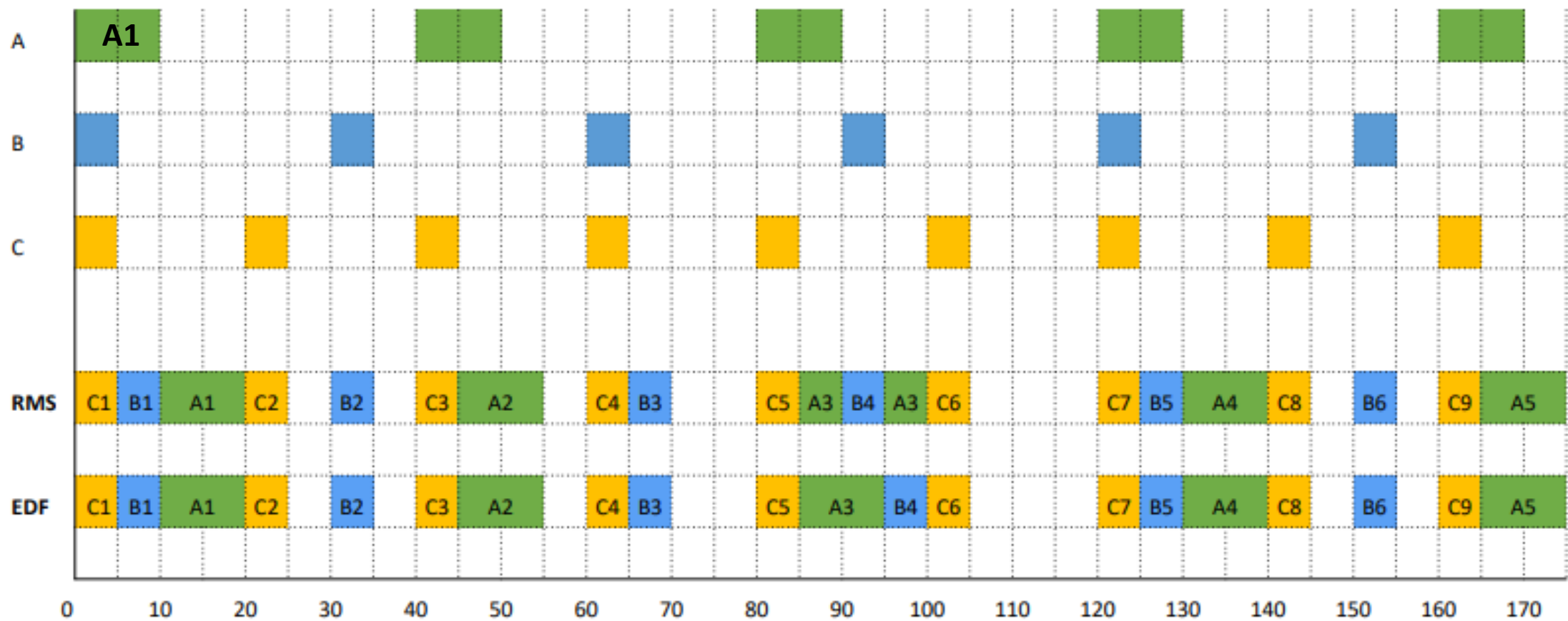
- $0,6667 \leq 1$  – zadatak je rješiv

- Prioriteti:

- A – 10ms svakih 40ms, prioritet:  $1/40\text{ms} = 25$
  - B – 5ms svakih 30ms, prioritet:  $1/30\text{ms} = 33$
  - C – 5ms svakih 20ms, prioritet:  $1/20\text{ms} = 50$

- Redoslijed prioriteta: C, B, A

# Rješenje 3b, 3c.



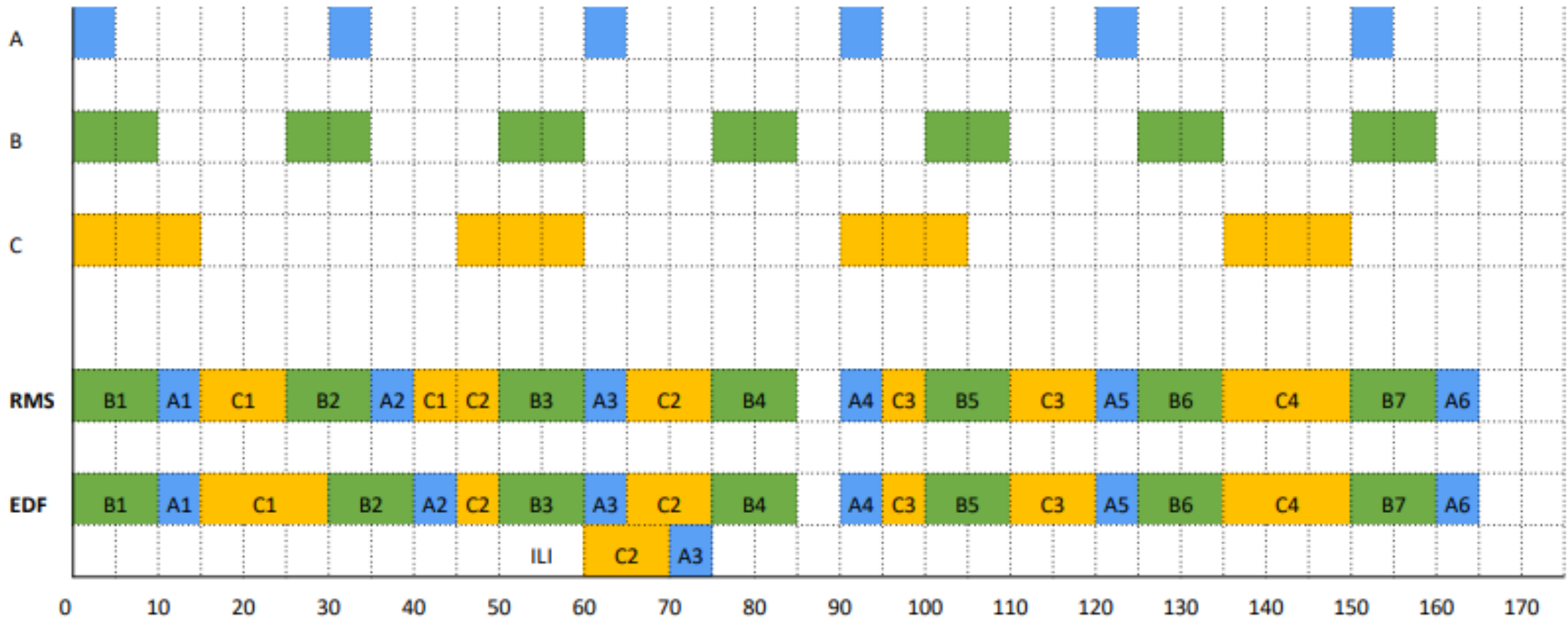
# Zadatak 4.

- A – 5ms svakih 30ms
  - B – 10ms svakih 25ms
  - C – 15ms svakih 45ms
- 
- Izračunaj prioritete i izvedivost (1 bod od 15)
  - RMS algoritam (3 boda od 15)
  - EDF algoritam (4 boda od 15)

# Rješenje 4a.

- A – 5ms svakih 30ms, prioritet 33
- B – 10ms svakih 25ms, prioritet 40
- C – 15ms svakih 45ms, prioritet 22
  
- Izvodljivost:
  - $\frac{5}{30} + \frac{10}{25} + \frac{15}{45} \leq 1$
  - $0,9 \leq 1$ , zadatak je rješiv

# Rješenje 4b, 4c



# Priprema za među-ispit i ispit

- Što je proces, što je dretva?
  - Raspodjela vremena u CPU
  - Stanja procesa
  - *Multi-proces vs. Multi-threading vs. Multi-processor*
- Prekidi
  - Tko uzrokuje i kako se obrađuju
- Procesor
  - Jedna jezgra vs. više jezgri
  - Jedan procesor vs. Vise-procesora
  - Četiri pristupa za planiranje višeprocorskih dretvi i dodjelu procesorskog vremena
  - Tri klasifikacije višeprocorskih sustava
- Memorija
  - Što je to stranica i koje je veličine
  - Planiranje memorije i stranica
  - Među-spremnik ili TLB
- Disk
  - FAT vs NTFS
  - RAID
  - Među-spremnik (cache)
- Ugrađeni/embedded OS
  - Gdje i zašto se koriste

+ **algoritmi** koji kažu što će se čitati/obraditi **prvo** u jedinici vremena (jer se tako računalo **ubrzava**)



# Hvala na pažnji!

Sretno na ispitu! 😊

...i vidimo se sljedeće godine na:

## OPERATING SYSTEM



For this you need

**PATIENCE**



For this you need

**MONEY**



For this you need

**SKILLS**