

# AROS

Pitanja za 2. deo  
(bez zadataka)

by  
manjanebe & Perks  
2016.

**Napomena:**

*U skripti se nalaze pitanja koja su se ponavljala prethodnih godina na drugom kolokvijumu i rokovima iz AROS-a. Pitanja nisu sortirana po glavama. Topla preporuka svima je da prethodno pročitaju knjigu, koja je dobro napisana za razliku od knjige iz OIKT-a, a ovu skriptu koriste kako bi naučili pitanja za kolokvijum.*

### 1. Tipovi datoteka kod UNIX-a

Regularne datoteke, direktorijumi, specijalne datoteke i imenovane FIFO datoteke.

### 2. Vrste organizacije virtuelne memorije

- segmentna
- stranična
- segmentno-stranična

### 3. Šta je stranični prekid i kako se servisira?

Kada se u toku izvršavanja programa traži pristup adresi koja pripada strani koja nije u memoriji dolazi do prekida programa. Ova vrsta prekida naziva se stranični prekid.

1. Utvrditi da li je adresa kojoj se pristupa u memoriji. Ako jeste nastavi sa izvršavanjem instrukcije. U suprotnom idi na korak 2
2. Prekini izvršavanje programa
3. Nadji slobodan okvir u memoriji. Ako takav okvir ne postoji izbacijednu od strana, odnosno oslobodi jedan od okvira koji su dodeljeni programu
4. Pronadji na disku stranu kojoj se pristupa i upisi je u slobodan okvir
5. Azuriraj tabelu strana
6. Iniciraj izvršavanje instrukcije koja je izazvala prekid.

### 4. Šta je „trashing“

Stanje sistema kada se mnogo više vremena troši na obradu straničnih prekida nego na izvršavanje procesa. Rešava se smanjenjem stepena multiprogramiranja.

### 5. Osnovne funkcije savremenih operativnih sistema

Osnovne funkcije savremenih operativnih sistema su:

- upravljanje **procesima**;
- upravljanje **memorijom**;
- upravljanje **uredjajima**;
- upravljanje **podacima**;
- **zastita**;
- **komunikacija** sa drugim racunarima u mrezi;
- **upravljanje greskama i oporavak Sistema**

### 6. Sinhronizacija procesa

**Sinhronizacija procesa** znači da određen proces ne može ići dalje od unapred definisane tačke bez eksplicitnog signala koji on sam ne može da generiše.

### 7. Kada su dva procesa blokirana?

2 procesa su **blokirana** ako ni jedan od njih ne može da nastavi sa radom dok drugi proces ne nastavi sa radom.

**8. Šta je inverzija prioriteta procesa? Objasniti kako nastaje.**

Ako proces višeg prioriteta zahteva pristup resursu koji je zauzet od procesa nižeg prioriteta, tada je proces višeg prioriteta blokiran procesom nižeg prioriteta, sve dok proces nižeg prioriteta ne završi rad i oslobodi resurs. Na taj način proces nižeg prioriteta ima "viši" prioritet.

**9. Kako se može rešiti problem inverzije prioriteta procesa?**

Proces nižeg prioriteta koji pristupa datom resursu nasleđuje viši prioritet procesa koji zahteva pristup istom resursu, sve do završetka rada sa resursom. Na taj način se ubrzava izvršavanje procesa nižeg prioriteta. (Drugi procesi možda imaju veći prioritet od pomenutog procesa, pa proces višeg prioriteta, jer je njemu potreban resurs zauzet, može beskonačno da čeka) Kada proces nižeg prioriteta završi, njegov prioritet se vraća na originalnu vrednost. To je **protokol nasleđivanja prioriteta**.

**10. Šta su "stack" i "heap" i čemu služe?**

Stack memorija služi za memorisanje parametara prilikom poziva procedura i za lokalne promenljive. Drugim rečima, za podatke koji se dinamički dodeljuju po unapred definisanom redosledu(predictable order). Ovom memorijom upravlja kompajler, a ne korisnik.

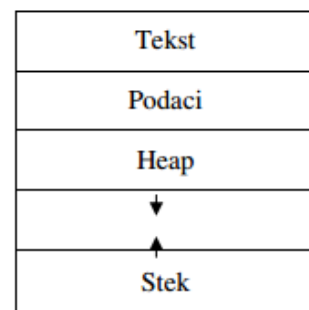
Heap memorija se koristi za proizvoljne strukture podataka, kao što su LinkedList itd. Drugim rečima, za podatke koji se dinamički dodeljuju na način koji nije unapred definisan(unpredictable manner). Ovom memorijom upravlja korisnik.

**11. Preslikavanje sadržaja**

Preslikavanje sadržaja se odnosi na preslikavanje memorijskih adresa u vrednosti (podatke) koje one sadrže, i vrši se u toku izvršenja programa.

**12. Grafički prikazati i objasniti adresni prostor procesa**

Svaki proces u toku izvršavanja zahteva memoriju i zato se svakom procesu pridružuje memorijski adresni prostor. U delu memorije koji je označen sa Tekst se nalaze instrukcije programa. Taj deo je fiksne velicine i instrukcije su u izvršnom obliku, tj. mašinske instrukcije programa koji se izvršava. U delu memorije koji je označen sa Podaci se nalaze predifinisane strukture podataka koje su inicijalizovane pre početka izvršavanja programa. Treći deo označen sa Heap se koristi za dinamičku dodelu memorije. Ovaj deo je promenljive veličine. U delu Stek se nalaze izvršni okviri funkcija koje su pozvane od strane datog programa. Povećanje i smanjenje veličine dela Stek je automatski.



**13. Objasniti šta je zaposleno čekanje ("busy wait").**

Zaposleno čekanje je situacija kada proces koji pokušava da izvrši P operaciju u situaciji kada je vrednost promenljive s jednaka nuli, troši vreme centralnog procesora, pri čemu ne može da napreduje dalje, a istovremeno onemogućava ostale procese da se izvršavaju.

**14. Šta je deskriptor datoteke?**

Prilikom otvaranja neke datoteke od strane date aplikacije, kernel vraća celobrojnu nenegativnu vrednost koja se zove deskriptor datoteke i ona na jedinstven način identifikuje tu datoteku za sve naredne operacije.

**15. Na koji način se operativni sistem "budi" iz stanja praznog hoda?**

Iz stanja praznog hoda operativni sistem se "budi":

- **prekidom** dobijenim od nekog hardverskog uređaja
- pojavom **izuzetka** od nekog korisničkog programa
- **sistemskim pozivom** iz nekog korisničkog programa

**16. Gde se koristi filesystem EXT2 i kakva je to struktura?**

ext2 je verzija linux sistema datoteka koja se zasniva na strukturi podataka inode, koji se naziva indeksni čvor. Struktura zauzima ili 64 ili 128B. Indeksni čvor postoji za svaku datoteku i sadrži najvažnije podatke o njoj. Struktura direktorijuma kod ext2 je hijerarhijska, počev od root direktorijuma (/).

ext3 je proširenje sistema ext2 i koristi se u Debian, Red Hat i drugim distribucijama linux-a. Osnovna razlika između ext2 i ext3 je to što je **dodat dnevnik transakcija**.

**17. U kojim stanjima mogu biti stranice u virtuelnoj memoriji?**

Stranice u virtuelnoj memoriji mogu biti u sledećim stanjima:

- **Preslikane u okvir fizičke memorije** (željeno stanje, sa njima se normalno radi ne dolazi do prekida kada program pozove adresu unutar stranice)
- **Inicijalno locirane na disku ili vraćene na disk** (prilikom poziva dolazi do straničnog prekida)
- **Ne koriste se** (dolazi do signal kill-a)

**18. Opisati "Best Fit" algoritam (algoritam najboljeg uklapanja).**

Kod "Best Fit" algoritma tabela slobodnog prostora uređuje se po rastućoj veličini slobodnog prostora. Ovaj algoritam se svodi na nalazjenje najmanjeg dovoljno velikog slobodnog memorijskog prostora u koji može da se smesti novi program.

**19. Opisati "First Fit" algoritam (algoritam prvog uklapanja)**

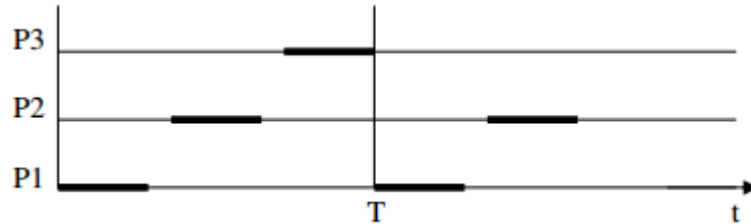
Kod "First Fit" algoritma operativni sistem vodi tabelu slobodnog prostora uređenu po rastućim adresama slobodnih delova memorije. Svodi se na pronalazjenje prvog slobodnog prostora, računajući od početka memorije, koji je dovoljno veliki da prihvati dati program.

**20. Logičke i primarne particije.**

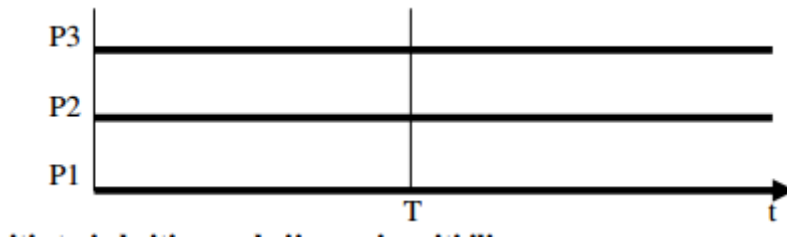
Primarne particije su one sa kojih je moguće podizanje operativnog sistema. Svaki disk mora da ima bar jednu primarnu particiju. Upotrebom više primarnih particija moguće je instalirati i koristiti više operativnih sistema na istom disku.

Logičke particije su particije čija je namena skladištenje podataka. Sa logičkih particija se ne može podizati operativni sistem.

**21. Prikazati kvaziparalelno izvršavanje programa na primeru programa P1, P2 i P3 pod uslovom da programi imaju isti prioritet izvršavanja i da operativni sistem radi u "time-sharing"-u.**



**22. Prikazati paralelno izvršavanje programa na primeru programa P1, P2 i P3.**



**23. Koji su potrebni uslovi za nastanak potpunog zastoja?**

- Medjusobno isključenje ("mutual exclusion")
- Posedovanje i cekanje ("hold-and-wait")
- Kružno cekanje ("circular wait")
- Nema prekidanja ("no preemption")

**24. Objasniti međusobno isključenje procesa prema kritičnoj sekciji.**

**Međusobno isključivanje procesa** u odnosu na kritične sekcije znači da se u bilo kom vremenskom intervalu može naći samo jedan proces u svojoj kritičnoj sekciji.

**25. Šta je kritična sekcija?**

Kritična sekcija je segment koda čije instrukcije mogu da uticu na druge niti (npr. azuriranje vrednosti zajednicke promenljive). Kada jedna nit izvršava kritičnu sekciju ni jedna druga nit ne sme da izvršava tu istu kritičnu sekciju.

**26. Šta je descriptor procesa (kontrolni blok procesa)?**

Kontrolni blok procesa (KBP) je skup podataka o aktivnom procesu o kojima OS void računa (ime procesa, identitet vlasnika, prioritet, PSW, oblast programa, oblast podataka, vrednost registara, logičko stanje itd.)

**27. Koje funkcije preslikavanja obuhvata upravljanje memorijom?**

Tipovi preslikavanja u upravljanju memorijom su: preslikavanje imena, preslikavanje adresa i preslikavanje sadržaja.

**28. Metode za dodelu prostora na disku.**

Tri glavne metode su:

- dodela susednih memorijskih lokacija
- dodela povezanih blokova fiksne veličine
- korišćenje šema sa indeksima

**29. Definirati operacije P i V u radu sa semaforima.**

$$P(s) = \begin{cases} s:=s-1 & \text{ako je } s>0 \\ \text{cekaj} & \text{u suprotnom} \end{cases}$$

$$V(s)=s:=s+1$$

**30. Koje podatke sadrži tabela stranica pored adrese okvira?**

- Tipična tabela stranica osim adrese okvira ima sledeće dodatne podatke:

- 1) bit koji pokazuje da li se stranica nalazi u operativnoj memoriji, tj. da li je stranici dodeljen okvir ili ne ("valid" ili "present" bit);
- 2) bit koji pokazuje da li je stranica u operativnoj memoriji modifikovana ili ne ("dirty: ili "modified" bit). Ako je stranica modifikovana tada mora biti upisana na disk;
- 3) bit koji pokazuje da li je stranica bila korišćena skoro ("referenced" ili "used" bit) ili ne;
- 4) dozvola pristupa koja označava da je stranica "read-only" ili je "read-write".
- 5) nekoliko bita namenjenih za stvarno adresiranje stranice u operativnu memoriju.

**31. Koje vrste planera postoje kod operativnih sistema?**

Kratkoročni i dugoročni.

Dugoročni planer selektuje procese koji su poslani na izvršavanje i prebacuje ih u red procesa spremnih za izvršavanje.

Kratkoročni planer selektuje proces iz reda procesa spremnih za izvršavanje i dodeljuje mu centralni procesor. Kratkoročni planer se veoma često poziva za razliku od dugoročnog.

**32. Šta je dispačer i koje su njegove funkcije?**

Dispačer je deo OS koji dodeljuje procesor procesu koji je izabran od strane kratkorocnog planera. Njegove funkcije su:

- promena konteksta
- prelazak u korisnicki nacin rada
- skok na odgovarajucu lokaciju unatar korisnickog programa radi ponovnog startovanja tog programa

Vreme koje je potrebno da dispačer zaustavi jedan proces I pokrene drugi naziva se dipečersko kašnjenje (dispatch latency)

**33. Kako je E. Dijkstra resio problem medjusobnog iskljucenja i sinhronizacije procesa?**

E Dijkstra je dao resenje problema medjusobnog iskljucenja i sinhronizacije procesa tako sto je uveo novi tip promenljive koja se naziva semfor i ciji je skup dozvoljenih vrednosti skup nenegativnih celih brojeva. Pritom je uve dve dozvoljene operacije nad promenljivim (P i V).

**34. Sistem se sastoji od dva procesa S i R koji dele zajednicki bafer kapaciteta C poruka. Proces S upisuje poruke u bafer, a proces R ih cita iz bafera. Koji uslovi moraju biti ispunjeni da bi sistem ispravno funkcionisao u situaciji kada relativne brzine pristupa baferu procesa S i R nisu unapred poznate?**

Da bi procesi S i R ispravno funkcionisali, u situaciji kada njihove relativne brzine nisu unapred poznate, neophodno je da budu ispunjeni sledeci uslovi:

- 1) posaljilac S ne sme da salje poruke kada je bafer pun, jer bi se ona izgubila;
- 2) primalac R ne sme da uzima poruku kada je bafer prazan, jer bi to mogao protumaciti kao signal da treba da prestane sa radom;
- 3) procesi ne smeju istovremeno pristupiti baferu;
- 4) vrednost promenljive N mora, u trenutku pristupa baferu bilo kog procesa, biti jednaka stvarnom broju poruka u baferu.

**35. U kom delu računara se nalaze procesi u stanju: 1) Blokiran/Suspendovan 2) Spreman/Suspendovan? Koja je razlika između ova dva stanja procesa?**

- 1) Proces je u spoljnoj memoriji I čeka na događaj
  - 2) Proces je u spoljnoj memoriji, ne čeka na događaje, ali će postati spreman za izvršavanje kada ga operativni sistem vrati u memoriju
- Proces koji je suspendovan, prestaje da se takmiči za resurse, oslobađaju se resursi

**36. U sistemu u kome postoji n procesa koji se izvršavaju na jednom centralnom procesoru broj mogućih načina za dodelu procesora je:**

**n!**

**37. Metode dodele prostora na disku koje koriste šeme sa indeksima, svakoj datoteci pridružuju tabelu indeksa. Koja su moguća rešenja za dodelu prostora na disku tabeli indeksa?**

Kao za obične datoteke:

- dodela susedne memorijske lokacije
- povezana lista blokova
- korišćenje indeksa u više nivoa

**38. Algoritmi planiranja kriterijumi**

1. Iskorišćenje CPU-a
2. Propusnost
3. Vreme provedeno u sistemu (turnaround time)
4. Vreme čekanja
5. Vreme odziva

**39. Objasniti šta je vreme provedeno u sistemu (turnaround time).**

Zbir vremena koje je process proveo čekajući da uđe u memoriju, vremena koje je proveo u radu procesa spremnih za izvršavanje i vremena potrošenog na procesorsko izvršavanje.

**40. Vrste algoritama.**

- **FCFS** First Come First Served
- **SJF** Shortest Job First
- **Priority Scheduling**
- **HPF** Highest Priority First
- **RR** Round-robin
- **SRTF** Shortest Remaining Time First
- **MLS** Multilevel Strategy

**41. Metode pristupa datotekama.**

- sekvencijalni pristup (kod editora i programskih prevodilaca)
- direktan ili relativni pristup (kod baza podataka)
  - indeksirani pristup (deo direktnog pristupa samo sa indeksima)

**42. Algoritmi za dodelu diska.**

FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK

**43. Šta je potrebno da se ostvari za upravljanje konkurentnim procesima?**

Upravljanje konkurentnim procesima usložnjava operativni sistem.

Potrebno je da:

1. Metod za deljenje vremena mora biti implementiran tako da omogući svakom od kreiranih procesa da dobije pristup sistemu;
2. Proces i sistemski resursi moraju da imaju zaštitu i moraju da budu zaštićeni međusobno;
3. Sistem ima ugrađene mehanizme unutar jezgra za prevenciju potpunog zastoja između procesa;



**44. Razlika između fizičke i virtuelne adrese.**

Virtuelna adresa je adresa u programu i nju generiše procesor, dok je fizička adresa adresa operativne memorije i puni se MAR.

**45. Vrste fragmentacije.**

Interna fragmentacija je deo memorije unutar regiona ili stranice koja je dodeljena datom procesu i ne koristi se od strane tog procesa. Interna fragmentacija može biti ako su u pitanju statičke particije.

Eksterna fragmentacija je neiskorišćena memorija između particija.

**46. Koji problem se javlja kod upravljanja memorijom pomoću dinamičkih particija?**

Osnovni problem je eksterna fragmentacija, jer zahteva dodatni hardver, usložnjava OS i zahteva više vremena za rad samog OS.

**47. Kada i kako se kod dinamičkih particija određuje veličina particije?**

**Kada** – u trenutku unošenja programa u memoriju.

**Kako** – Određuje se na osnovu veličine programa. Mora se definisati poseban algoritam za upravljanje smeštanjem (first fit, best fit, worst fit)

**48. Šta je TLB?**

TLB (Translation Lookaside Buffer) je asocijativna memorija koja se koristi u radu sa tabelama stranica. Sadrži broj stranica i broj okvira. Broj asocijativnih registra i TLB-u je između 8 i 2048. Ukoliko broj stranice nije u TLB-u, tada se mora pročitati tabela stranica sa diska da bi se dobio broj okvira.

Može se reći i da je TLB namenski keš za podatke iz tabele stranica koji služi da smanji broj pristupa disku prilikom pristupa određenoj adresi.

**49. Koja je razlika između prekida i izuzetaka?**

Prekid je sistemski poziv koji servisira asinhrono događaje koje dolaze van procesora, npr. Od I/O Uređaja, a izuzetak generiše sam procesor kada dođe do greške pri izvršavanju koda.

**50. Dat sistem ima p procesa. Svakom procesu je potrebno najviše m resursa datog tipa. Ukupno je raspoloživo r resursa. Koji uslov mora biti ispunjen da se ne bi pojavio deadlock?**

$$r \geq p(m - 1) + 1$$

r-raspoloživo resursa  
p-broj resursa  
m-potreban broj resursa

**51. Objasniti sta znaci potpuni zastoj(“deadlock”) ?**

To je situacija kada proces trajno ostaje u stanju cekanja, jer u isto vreme postoje drugi procesi koji su takodje u stanju cekanja, a zauzeli su resurse koje zahteva dati process i te resurse drze zauzetim. U takvoj situaciji dati proces nikada ne dobija zahtevane resurse.

## 52. Koja su moguća stanja procesa?

Moguća stanja procesa su sledeća:

- 1) NOV – proces je kreiran
- 2) IZVRŠAVA SE – instrukcije datog programa se izvršavaju;
- 3) ČEKA – proces čeka da se neki događaj dogodi;
- 4) SPREMAN – proces čeka da bude dodeljen procesoru
- 5) ZAVRŠEN – proces je završio izvršavanje.

## 53. Objasniti prelaske između mogućih stanja procesa.

- Nakon kreiranja procesa, proces se nalazi u stanju NOV. Kada se procesu dodele potrebni resursi, osim procesora, tada on prelazi u stanje SPREMAN. Nakon dodele procesora prelazi u stanje IZVRŠAVA SE. Po isteku dodeljenog kvantuma vremena proces prelazi iz stanja IZVRŠAVA SE u stanje SPREMAN. Ako u stanju IZVRŠAVA SE mora da sačeka neki događaj (npr. zavrsetak neke U/I aktivnosti), tada proces prelazi u stanje CEKA i o tome obavestava operativni sistem. Po završetku U/I aktivnosti proces iz stanja CEKA prelazi u stanje SPREMAN. Proces prelazi iz stanja SPREMAN u stanje IZVRŠAVA SE kada procesor postane raspoloživ i ako u toku izvršavanja procesa se završi izvršavanje svih programskih instrukcija tada proces prelazi u stanje ZAVRŠEN.

## 54. Objasniti šta je preslikavanje adresa.

Preslikavanje adresa se odnosi na preslikavanje programskih adresa u stvarne, fizičke memorijske adrese, i vrši se pomoću jedinice za upravljanje memorijom (MMU). U slučaju statickog povezivanja adresni prostor je linearan (sastoji se od niza kontinualnih adresa). U zavisnosti od načina upravljanja memorijom takav adresni prostor može se preslikovati u odgovarajući linearan memorijski prostor pa se preslikavanje adrese može opisati funkcijom:

$$a' = p + a$$

gde je

$a'$  – memorijska adresa,

$p$  – početna adresa programa u memoriji

$a$  – programska adresa

## 55. Objasniti FCFS?

Najjednostavniji logaritam, izvršava zahteve za dodelu diska onim redosledom kojim su nastali.

## 56. Objasniti SSTF?

Algoritam koji u datom trenutku vrši pokretanje glave diska uzeti u obzir zadate U/I zahteve. Osnovna ideja je minimiziranje vremena pozicioniranja u odnosu na trenutnu poziciju glave diska.

**57. Objasniti SCAN.**

Ovaj algoritam razresava problem "gladovanja" koji se može javiti kod SSTF algoritma. Problem se razresava tako što se nakon pokretanja glave diska dalje pokretanje se nastavlja u istom smeru, tj. najpre se servisiraju zahtevi pocev od najdalje spoljasnje staze ka unutrašnjoj stazi koja je krajnja, a zatim od najblize unutrašnje ka najdaljoj spoljasnjoj.

**58. Objasniti C-SCAN.**

Ovaj algoritam omogućuje uniformnija vremena odziva tako što se pokretanje glava diska uvek vrši u jednom smeru.

**59. Objasniti LOOK.**

Ovaj algoritam modifikuje SCAN algoritam, tako što zaustavlja kretanje glave u istom smeru ako nema više U/I zahteva u tom smeru.

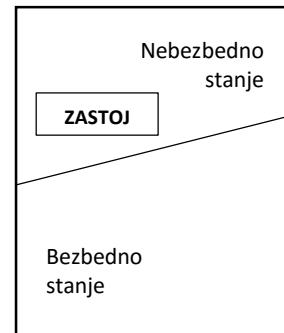
**60. Objasniti C-LOOK.**

Ovaj algoritam servisira zahteve pomeranjem glave diska od spoljasnjeg ka unutrašnjem cilindru dokle god ima zahteva, a zatim se vraća na spoljasnji najudaljeniji cilindar za koji postoji zahtev.

(Preporuka da se objašnjenje ova 4 algoritma pročita u knjizi, može da se desi da dođe zadatak sa njima)

**61. Grafički prikazati skup mogućih stanja sa stanovišta potpunog zastoja?**

Slika →



**62. Šta je poziv FLock kod UNIX-a?**

Flock (file lock) je sistemski poziv kod UNIX-a koji zaključava datu datoteku tako da ni jedan drugi proces ne može da je otvori u vreme kad je datoteka zauzeta.

**63. Objasniti razliku između sinhronih i asinhronih mehanizama procesa.**

**Sinhroni mehanizam** – ukoliko proces poziva drugi proces u vreme izvršavanja neke operacije, dolazi do blokiranja procesa jer mora da sačeka da se operacija završi.

**Asinhroni mehanizam** – u situaciji kada proces poziva drugi proces prilikom izvršavanja neke operacije nema čekanja, tj. proces ne biva blokiran.

**64. Šta je kernel?**

Kernel, jezgro ili nukleus operativnog sistema je deo OS koji sadrži najčešće korišćene funkcije i uvek se nalazi u operativnoj memoriji.

**65. Kako se može vršiti deljenje resursa?**

Deljenje resursa se može vršiti preko:

- **Vremenskog multipleksiranja** (resurs se dodeljuje na određeno vreme procesu)
- **Prostornog multipleksiranja** (resurs se deli na više delova)

**66. Objasniti rad semafora sa  $n \geq 2$  procesa korišćenjem mediska.**

Za više procesa koji imaju kritične sekcije jedan u odnosu na druge koristi se promenljiva medisk(međusobno isključenje) koja je tipa semafor.

```
var medisk: semaphore;  
begin  
  medisk:=1;  
  parbegin
```

..... svaki od procesa bi bio sledećeg oblika .....

```
  Pi : begin  
    repeat  
      P(medisk);  
      kritična_sekcija_i;  
      V(medisk);  
      ostale_naredbe_i;  
    until stop;  
  end;
```

```
.....  
  
  parend;  
end;
```

**67. Šta je lokalno, a šta globalno straničenje?**

**Globalno straničenje** – ukoliko su nakon straničnog prekida svi raspoloživi okviri kandidati za izbacivanje.

**Lokalno straničenje** – ukoliko su samo stranice u okvirima koje pripadaju datom procesu kandidati.

**68. Kolika je maksimalna veličina datoteka i particija kod FAT16 i FAT32?**

Kod FAT16 – maksimalna veličina datoteke i particije je 2GB

Kod FAT32 – maksimalna veličina datoteke je 4GB, a particije 2TB

**69. Šta sve može biti interfejs?**

- **GUI** (grafički korisnički interfejs)
- **Bilo koja korisnička aplikacija**
- **Interpreter komandi**

RANDOM PITANJA NA ZAOKRUŽIVANJE (ne zna se naslov pitanja, znaju se odgovori)

1. (Pitanje: „multi nešto“) odgovori: Hetch, Stack
2. Odgovori: Heap, Globalne promenljive
3. SJF izaziva gladovanje (ili SSTF ne zna se šta je bilo ponuđeno :D )