

U tablici **zaokružiti** brojeve zadataka koji su rješavani – priložiti ovaj papir s rješenjima.

Zad:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	∑	
Bod:																				

Napomena: PISATI ČITKO: NEČITLJIVO = KRIVO!

1. (2) Navesti slojeve računalnog sustava.

korisnik <--> programi <--> operacijski sustav <--> sklopovlje

2. (2) Opisati upravljanje ulazno-izlaznim napravama pomoću sklopova s izravnim pristupom spremniku.

DMA sklop treba programirati. On potom prenosi podatke od/prema vanjskoj napravi korištenjem sabirnice koju zatraži i dobiva od procesora.

3. (2) Koju mogućnost nudi saznanje da su dva zadatka nezavisna?

Nezavisni zadaci se mogu izvoditi proizvoljnim redoslijedom pa i paralelno.

4. (2) Opisati strukturu podataka za ostvarenje kašnjenja (u jednostavnom modelu jezgre).

Red Odgođene\_D je složen prema vremenima aktiviranja: prva u redu je ona će se najprije "probuditi" i nastaviti s radom.

U opisniku dretve nalazi se brojač koliko još otkućaja sata dretvu treba odgoditi. Ako dretva nije prva u listi, onda se taj broj nadodaje na odgodu prethodne dretve.

5. (2) Kako treba proširiti jezgrine funkcije za višeprocorske sustave (gdje korištenje prekida kao mehanizma poziva jezgrinih funkcija nije dovoljna zaštita)?

dodatnom sinkronizacijom metodom radnog čekanja

6. (2) Opisati raspoređivače SCHED\_FIFO i SCHED\_RR.

Oba raspoređivača su prioritetni raspoređivači:

- prvo se raspoređuje dretva najveća prioriteta

Tek kada odabir nije jedinstven (ima više dretvi istog najvećeg prioriteta) tada se koristi drugi kriterij:

- kod SCHED\_FIFO - uzima se prva takva dretva i izvodi do njena završetka ili blokiranja

- kod SCHED\_RR - uzima se prva takva dretva i izvodi do njena završetka ili blokiranja ili dok ne istekne kvant vremena kada se miče na kraj reda (za taj prioritet)

7. (2) Opisati način pretvorbe logičke adrese u fizičku kod straničenja (može i samo grafički).

Logička adresa dijeli se na: adresu stranice + odmak od početka stranice

Fizička adresa dijeli se na: adresu okvira + odmak od početka stranice

Pretvorba se odvija korištenjem sklopa i tablice prevođenja koja sadrži informaciju gdje se nalazi pojedina stranice u kojem okviru).



11. (3) Obrada svakog zahtjeva u nekom web poslužitelju odvija se korištenjem nekoliko dretvi. Dretva obradi najprije dohvaća zahtjev. Potom signalizira dretvama dohvati\_stranicu, dohvati\_zaglavlje i dohvati\_oglase da obave svoj dio posla te potom kad su one gotove, dretva obradi sklapa odgovor kojeg vraća klijentu. Nepotpuni pseudokod dretvi zadan je u nastavku.

```

obradi() {
    ponavljaaj {
        zahtjev = čekaj_zahtjev()
        "pošalji signale"
        "čekaj signale"
        odgovor = sklopi_odgovor()
        vrati(odgovor)
    }
}

dohvati_*(klijent) {
    ponavljaaj {
        "čekaj signal"
        napravi_potrebno
        "pošalji signal -> obradi"
    }
}

```

Zamijeniti dijelove označene navodnicima kodom za sinkronizaciju. Koristiti semafore ili monitore. Navesti početne vrijednosti korištenih varijabli (i semafora ako se oni koriste).

S monitorima:

```

obradi() {
    mutex_lock(m)
    ponavljaaj {
        zahtjev = čekaj_zahtjev()
        gotovo = 0
        cond_broadcast(radi)
        while(gotovo < 3)
            cond_wait(sklopi)
        odgovor = sklopi_odgovor()
        vrati(odgovor)
    }
    mutex_unlock(m)
}

dohvati_*(klijent) {
    mutex_lock(m)
    ponavljaaj {
        cond_wait(radi)
        mutex_unlock(m)
        napravi_potrebno
        mutex_lock(m)
        gotovo++
        cond_signal(sklopi)
    }
    mutex_unlock(m)
}

```

Sa semaforima:

```

obradi() {
    ponavljaaj {
        zahtjev = čekaj_zahtjev()
        PostaviOSEM(1)
        PostaviOSEM(2)
        PostaviOSEM(3)
        ČekajOSEM(4)
        ČekajOSEM(4)
        ČekajOSEM(4)
        odgovor = sklopi_odgovor()
        vrati(odgovor)
    }
}

dohvati_*(klijent) {
    ponavljaaj {
        ČekajOSEM(i) //i=1/2/3
        napravi_potrebno
        PostaviOSEM(4)
    }
    mutex_unlock(m)
}

```

12. (3) Poslovi  $P_1$ ,  $P_2$  i  $P_3$  se javljaju periodički svakih 200 ms. Unutar svake periode  $P_1$  se javlja 20 ms prije  $P_2$  koji se javlja 10 ms prije  $P_3$ . Obrada za  $P_1$  traje 50 ms, za  $P_2$  30 ms te za  $P_3$  20 ms. Izračunati:
- prosječan broj dolazaka novih poslova u jedinici vremena ( $\alpha$ )
  - prosječno trajanje obrade ( $1/\beta$ )
  - prosječan broj poslova u sustavu ( $\bar{n}$ )
  - prosječno zadržavanje poslova u sustavu ( $\bar{T}$ ).

```
2: 33
1: 222333
P:1111122233-----
```

```
alfa= 3 / 200 ms = 3/0,2 = 15 1/s
1/beta = (50+30+20)/3 = 100/3 = 33,3 ms
T1=50 ms, T2=60 ms, T3=70 ms, T = (50+60+70)/3 = 180/3 = 60 ms
n = alfa*T = 15 * 0,06 = 0,9
```

13. (3) Zbog preopterećenja nekog poslužitelja koji je radio s 90 % opterećenjem, trećina zahtjeva koje je on dobivao je preusmjerena drugdje. Ako je prije prosječno vrijeme zadržavanja zahtjeva u tom poslužitelju bilo 1 s, koliko će biti nakon ovog preusmjeravanja (samo za zahtjeve u ovom poslužitelju)?

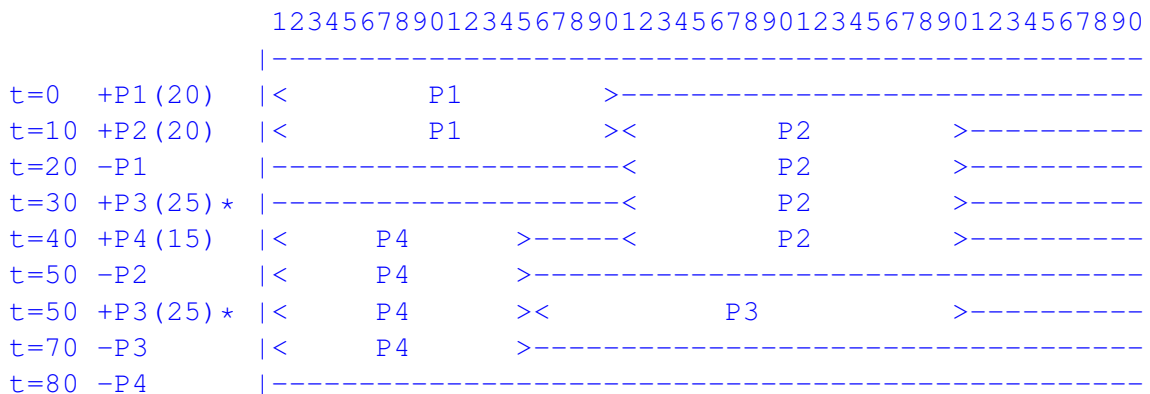
```
rho=0.9
alfa'=2/3 alfa
T=1
```

```
rho, T => alfa=9, beta=10 => alfa' = 6
T'=1/(10-6) = 1/4 = 0,25 s
```

14. (3) Dretve A, B, C i D javljaju se u trenucima 0. ms, 3. ms, 5. ms, 10. ms, respektivno. Trajanje rada svake dretve je po 5 ms. Ukoliko se koristi raspoređivanje podjelom vremena uz kvant vremena  $T_q = 2$  ms, pokazati rad raspoređivača nad zadanim skupom dretvi.

```
3:
2:          C B          D B B C C D
1:          B A A C B B C C D D B B C D
P:| A | A | B |A| C | B | C | D |B|C| D |D|
t:0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
  +   +   +   -   +           - -   -
  A   B   C   A   D           B C   D
```

15. (3) U višeprocorskom sustavu koji koristi dinamičko upravljanje spremnikom s 50 MB spremničkog prostora za programe javljaju se sljedeći događaji:
- u  $t=0$  ms zahtjev za pokretanjem  $P_1$  koji treba 20 MB i 20 ms procesorskog vremena
  - u  $t=10$  ms zahtjev za pokretanjem  $P_2$  koji treba 20 MB i 40 ms procesorskog vremena
  - u  $t=30$  ms zahtjev za pokretanjem  $P_3$  koji treba 25 MB i 20 ms procesorskog vremena
  - u  $t=40$  ms zahtjev za pokretanjem  $P_4$  koji treba 15 MB i 40 ms procesorskog vremena.
- Ako se neki zahtjev ne može ispuniti u trenutku postavljanja, on će biti na čekanju do trenutka kada se može ispuniti. Prikazati rad sustava i stanje radnog spremnika pri posluživanju navedenih zahtjeva (dok se svi ne posluže!).



16. Nad matricama  $A[N, N]$  i  $B[N, M]$  obavljaju se statističke analize u nekoliko koraka.
- (1) Za svaki redak  $i$  matrice  $A$  izračunava se statistika i zapisuje u redak  $i$  matrice  $B$ .
  - (1) Za cijelu matricu  $A$  se izračunava statistika korištenjem samo matrice  $B$  (jednim prolazom slijedno kroz matricu  $B$ ).
  - (2) Svaka dva različita retka matrice  $A$  se međusobno uspoređuju (rezultat se zapisuje u te retke matrice  $A$ ).

Program koji radi navedeno se izvodi u sustavu sa straničenjem uz veličinu stranice  $N$  riječi (pretpostaviti da je  $N = 10 \cdot M$  i  $N > 10$ ). Koliko će promašaja izazvati navedeni program ukoliko za matrice  $A$  i  $B$  na raspolaganju postoje tri okvira te strategija izbacivanja stranice je LRU? Zanemariti promašaje zbog dohvata ostalih podataka i instrukcija.

1. statistika za 10 redaka od A stane u 10 redaka od B => 1 stranica  
neka je: - prvih  $N$  stranica matrica A  
- idućih  $N/10$  matrica B  
zahtjevi idu redom:  
1  $N+1$  2  $N+1$  ... 10  $N+1$  11  $N+2$  12  $N+2$  ....  $N$   $N+N/10$   
svi  $N+x$  su pogotci, osim prvog te ukupno imamo:  
-  $N$  promašaja zbog matrice A te  $N/10$  zbog B =  $N + N/10$

2. obrada B:  $N/10$  promašaja

3. usporedba svakog retka sa svakim:

za  $i = 1$  do  $N-1$   
za  $j = i+1$  do  $N$   
uspoređi redak  $i/j$

kad je  $i=x$ : radimo s retkom  $x$  (dohvat=promašaj) te zasebno za svaki idući ( $N-x$  promašaja)  
ukupno:  $N+(N-1)+\dots = N \cdot (N+1) / 2$

UKUPNO:  $N+N/10 + N/10 + N \cdot (N+1) / 2$  ( $=N+N/5+N \cdot N/2+N/2= N \cdot N/2+N \cdot 17/10$ )

17. (3) Neki disk ima 4 površine od kojih svaka ima 512 staza, svaka staza ima 4096 sektora, a svaki sektor je velik 4 kB. Disk se okreće s 6000 okr/min. Disk radi na načelu da pročita cijelu stazu u interni spremnik te ju prenosi u radni spremnik – prijenos traje 5 ms. Prosječno postavljanje glave iznosi 10 ms, a premještanje na susjednu stazu 1 ms. Na tom disku kompaktno je pohranjena datoteka veličine 160 MB. Koliko će trajati čitanje cijele datoteke u radni spremnik (uz zanemarenje svih ostalih operacija, tj. pretpostavke da je opisnik datoteke već učitana u radni spremnik te da nema drugih zahtjeva prema disku)?

```
staza = 4096*4 kB = 16 MB
datoteka = 160 MB => 10 staza => 2 puna cilindra + 2 staze na 3.
6000/60 = 100 okr/s => TR=10 ms, TR/2=5 ms
```

```
t = T_seek
  + (TR/2+TR+TP) + (TR/2+TR+TP) + (TR/2+TR+TP) + (TR/2+TR+max(TP, T1))
  + (TR/2+TR+TP) + (TR/2+TR+TP) + (TR/2+TR+TP) + (TR/2+TR+max(TP, T1))
  + (TR/2+TR+TP) + (TR/2+TR+TP)
```

```
max(T_P, T1) = T_P
```

```
t = T_seek + 10*(TR/2+TR+TP) = 10+10*(5+10+5) = 10+10*20 = 210 ms
```

18. (3) Neka datoteka je pohranjena kompaktno po dijelovima u blokovima:

- a) 1. dio: 536–587
- b) 2. dio: 144–255
- c) 3. dio: 442–549.

Skicirati organizaciju kazaljki za tu datoteku, ako se radi o datotečnom sustavu UNIX (inode) koji koristi 10+3 kazaljki u opisniku, uz veličinu blokova od 2 kB i veličinu kazaljke od 64 bita. Nije potrebno pisati vrijednosti kazaljki, već samo gdje su one raspoređene.

Nenamjernom greškom se prvi i treći blok datoteke preklapaju: očito takva datoteka ne postoji. Međutim, to ne utječe na rješenje.

```
536--587 => 52
144--255 => 112
442--549 => 108
ukupno   => 272 bloka
```

u bloku 2 kB stane 2 kB/ 64 bita = 2048/8 = 256 kazaljki

```
272 = 10 + 256 + 6
      1-10 11. 12.
```

Kazaljke:

```
1-10 11 12
| +-[blok s 256 kaz.: koristi se samo prva]
|   +-[blok s 256 kaz.: koristi se prvih šest]
|
| +-[blok s 256 kaz: sve se koriste]
```