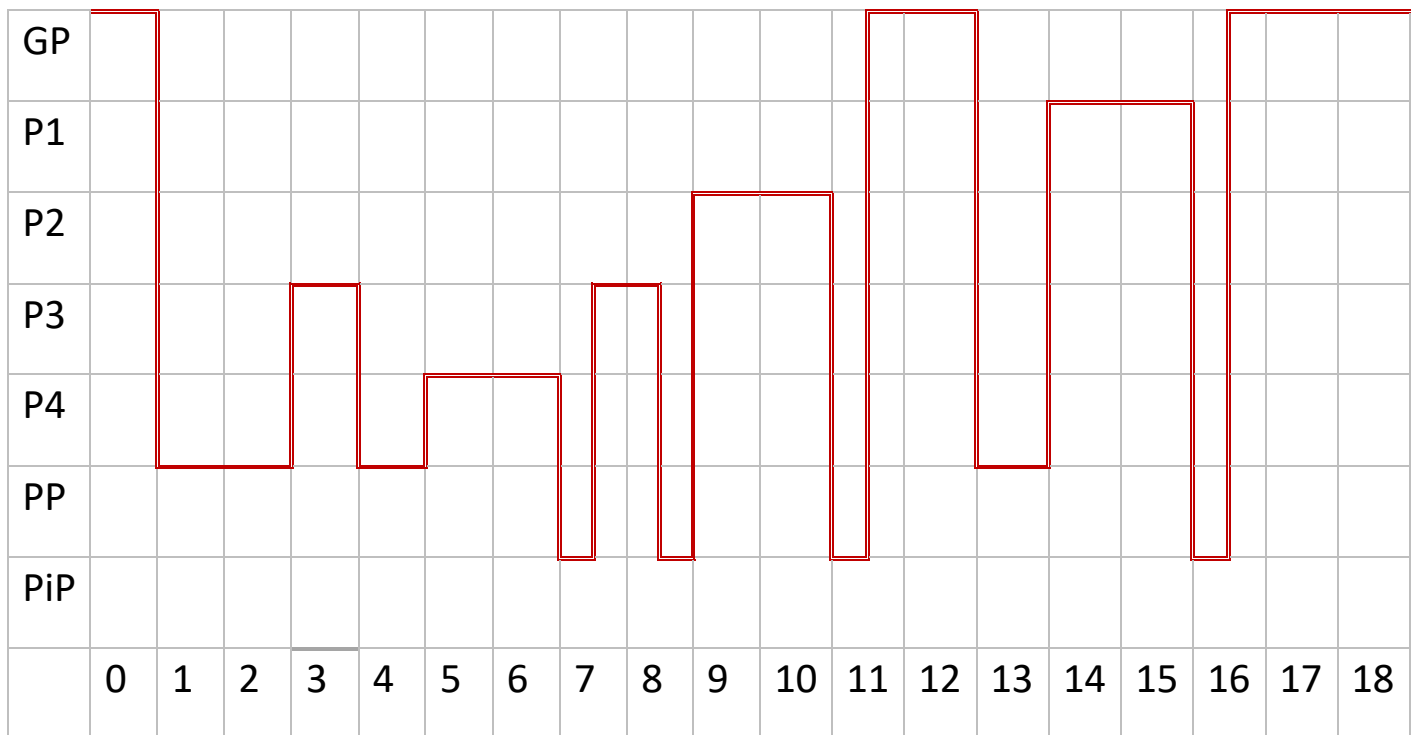


1. Zaokružiti jedan od ponuđenih dijelova teksta odvojenih znakom „/“, nadopisati potreban tekst na crte „\_\_\_“
- (1) Nužni uvjeti za potpuni zastoj jesu da u sustavu bude minimalno/maksimalno 2 dretva/e/i te minimalno/maksimalno 2 semafor/a.
- (2) U sustavu s poslužiteljem, opterećenje od 100% moguće je jedino u determinističkom/nederminističkom slučaju.
- (3) Sustav će bolje (brže) reagirati na korisničke zahtjeve ako sustav koristi: raspoređivanje prema redu prispjeca / raspoređivanje podjelom vremena
- (4) Littleovo pravilo je iskazano formulom  
$$\rho = \frac{a}{b} \quad / \quad \bar{n} = \alpha \bar{T} \quad / \quad n = \frac{m}{2} \quad / \quad T_R = \frac{1}{\omega}$$
- (5) Sklopovska potpora nije potrebna ako se koristi sljedeći način upravljanja spremnikom: STATIČKI (za dinamički način koristi se zbrajalo, a za straničenje TLB)
- (6) Prosječno rotacijsko kašnjenje iznosi:  $T_R/2$
- (7) U opisniku datoteke pamte se neka svojstva datoteke. Od vremenskih svojstava to su zapisana vremena: vrijeme stvaranja, vrijeme modifikacije, vrijeme zadnje pristupa
2. U nekom sustavu bez sklopa za prihvrat prekida, ali uz programsku potporu za obradu prema prioritetu, javljaju se prekidi P3 u trenutku  $t=0$  ms, P2 u  $t=1$  ms, prekid P4 u  $t=3$  ms te prekid P1 u  $t=12$  ms. Prioritet prekida određen je brojem (P4 ima najveći prioritet). Grafički prikazati aktivnosti procesora u glavnom programu (GP), procedurama za obradu prekida ( $P_i$ ) koje traju 2ms te procedurama za prihvrat prekida (PP) koje traju 1 ms i povratak iz prekida (PiP) koje traju 0,5 ms.



3. U promatranom trenutku stanje sustava je sljedeće: dretva 7 je aktivna, dretve 5, 3 i 1 su u redu pripravnih dretvi te dretve 4 i 2 su u redu općeg semafora S. Svi redovi organizirani su po prioritetu gdje veći broj dretve označava prioritet. Ako tada dretva 7 pozove jezgrinu funkciju PostaviOSem(S), kako će izgledati struktura podataka jezgre nakon poziva?

Red Aktivna D: 7

Red Pripravne D: 5,4,3,1

Red OSEM(S): 2

OSEM[1].v = 0

4. Neki sustav upravljanja treba ostvariti korištenjem tri dretve *dohvati* te jednom dretvom *obradi*. Svaka dretva dohvati provjerava svoj dio sustava funkcijom *provjera(id)*, gdje *id* predstavlja identifikacijski broj dretve koja tu funkciju poziva (1, 2 ili 3). Kada funkcija *provjera(id)* vrati nulu tada taj dio sustava ne treba dodatnu pažnju te dotična dretva *dohvati* spava 5 sekundi prije ponovne provjere. U protivnom, kada je povratna vrijednost veća od nule, dretvi dohvati ta vrijednost predstavlja prioritet otkrivenog događaja te to treba dojaviti dretvi *obradi*. Dretva dohvati potom čeka potvrdu da je njen događaj obrađen prije nego li ponavlja provjeru. Dretva *obradi* čeka da joj netko dojadi potrebu za obradom. Kad joj se dojadi ona među dojavljenim događajima odabire događaj najvećeg prioriteta te ga obrađuje funkcijom *obradi\_događaj(prio)*. Po završetku obrade jednog

događaja, dretva obradi ide na sljedeći ako takav postoji ili čeka na dojavu novog. Napisati pseudokod dretvi korištenjem semafora.

<pre>dretva dohvati (id) {     ponavljaj {         prio = provjera(id)         ako je prio == 0 tada {             spavaj(5)         } inače {             ...             događaj[id] = prio             ...         }     } }</pre>	<pre>dretva obradi(id) {     ....     ponavljaj {         ...         id = 0         ako je događaj[2] &gt; događaj[id]             tada { id = 2 }         ako je događaj[3] &gt; događaj[id]             tada { id = 3 }         ako je događaj[id] &gt; 0 tada {             ....             obradi događaj (događaj[id])             ....             događaj[id] = 0             ....         } inače {             ....         }         ....     } }</pre>
---	---

<pre>dretva dohvati (id) {     ponavljaj {         prio = provjera(id)         ako je prio == 0 tada {             spavaj(5)         } inače {             MUTEX_LOCK(&amp;M)             događaj[id] = prio             COND_SIGNAL(&amp;M, NOVO)             COND_WAIT(&amp;M, STARI[ID])             MUTEX_UNLOCK(&amp;M)         }     } }</pre>	<pre>dretva obradi(id) {     MUTEX_LOCK(&amp;M)     ponavljaj {         ...         id = 0         ako je događaj[2] &gt; događaj[id]             tada { id = 2 }         ako je događaj[3] &gt; događaj[id]             tada { id = 3 }         ako je događaj[id] &gt; 0 tada {             MUTEX_UNLOCK(&amp;M)             obradi događaj (događaj[id])             MUTEX_LOCK(&amp;M)             događaj[id] = 0             COND_SIGNAL(&amp;M, STARI[ID])         } inače {             COND_WAIT(&amp;M, NOVI)         }         MUTEX_UNLOCK(&amp;M)     } }</pre>
--	--

5. U nekom sustavu poslužitelj radi s 75% opterećenjem te je prosječno zadržavanje poslova u sustavu dvije sekunde. Kada bi poslužitelj zamijenili novim pet puta jačim (po svim kriterijima) te kada bi na taj poslužitelj preusmjerili još tri puta toliko poslova prosječno istih svojstava (ukupno bi tada poslužitelj primao četiri puta više nego stari poslužitelj), koliko bi tada bilo prosječno zadržavanje poslova u sustavu? Pretpostaviti Poissonovu razdiobu dolazaka i eksponencijalnu razdiobu za trajanja obrade.

$$\rho = \frac{\alpha}{\beta} \bar{T} = \frac{1}{\beta - \alpha} \quad \rho = 75\% = 0.75, \bar{T} = 2s, \bar{\beta} = 5\beta, \bar{\alpha} = 4\alpha$$

$$0.75 = \frac{\alpha}{\beta} \xrightarrow{\text{iz toga}} \alpha = 0.75\beta \quad 2 = \frac{1}{\beta - \alpha} \xrightarrow{\text{iz toga}} 2 = \frac{1}{0.25\beta} \xrightarrow{\text{itd.}} \beta = 2$$

$$\alpha = 1.5, \quad \bar{\alpha} = 6, \quad \bar{\beta} = 10$$

$$\bar{\bar{T}} = \frac{1}{10 - 6} = 0.25s$$

6. Neki sustav koristi raspoređivanje prema prioritetu kao osnovnom kriteriju te prema podijeli vremena kao dodatnom koji se koristi kad prvi kriterij ne daje samo jednu dretvu (opisan kao SCHED\_RR). U nekom trenutku  $t_0$  u sustavu se nalaze dretve A, B, C, D i E. Dretva A ima najveći prioritet 5, slijede dretva B i C s prioritetom 4 te dretve D i E s prioritetom 3. Dretva A treba još 30 ms za završetak, dretva B 50 ms, dretva C 20 ms te dretve D i E po 30 ms. Prikazati stanje procesora od trenutka  $t_0$  pa dok sve dretve ne završe s radom ako je kvant vremena  $T_q = 10 \text{ ms}$  (kada se koristi).

E																
D	E	E	E	E												
C	D	D	D	D	E											
B	C	B	C	B	D	E	D	E	D	E						
A	B	C	B	C	B	D	E	D	E	D	E					
0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160

$$T_q = 10ms$$

	Prioritet	Trajanje
A	5	30
B	4	50

C	4	20
D	3	30
E	3	30

7. U sustavu koji raspolaže s 20 stranica za programe pokreću se redom procesi: P1 koji ukupno treba 10 stranica, P2 koji treba 4 stranice, P3 koji treba 12 stranica, P4 koji treba 6 stranica, P5 koji treba 8 te P6 koji treba 10. Svi su procesi već pripremljeni na pomoćnom spremniku (disku). Upravljanje okvirima je ostvareno tako da se niti jednom zahtjevu neće dati više od 8 okvira, a ako nema dovoljno mjesta daje se minimalno 4 okvira. Nadalje, kad se neki zahtjev ne može poslužiti s minimalno 4 okvira tada se procesima koji zauzimaju više od 4 okvira oduzimaju okviri. Pri oduzimanju okvira odabire se proces s najviše okvira i njemu se oduzima najstariji okvir. Potom će se postupak ponavljati dok se ne dobiju svi potrebni okviri ili se više ne može ništa nikome oduzeti. Prikazati stanje sustava do (pokušaja) posluživanja svih procesa.
- P1 => 10 stranica, P2 => 4 stranice, P3 => 12 stranica, P4 => 6 stranica, P5 => 8 stranica, P6 => 10 stranica (min: 4 stranice, max: 8 stranica)

1	1	1	1	1	1	1	1	1											
2	1	1	1	1	1	1	1	1	2	2	2	2							
3	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	3	3	3
4	4	4	1	1	1	1	1	1	2	2	2	2	4	4	3	3	3	3	3
5	4	4	5	5	1	1	1	1	2	2	2	2	4	4	5	5	3	3	3
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

8. Neki disk ima 6 površina na kojima su zapisani podaci. Na svakoj stazi nalazi se 2048 sektora, svaki sektor veličine 512 B. Disk se okreće s 6000 okr/min. Pri čitanju disk čita cijelu stazu u interni spremnik te prenosi potrebne sektore u radni spremnik brzinom 4.096 Gbit/s. Za vrijeme prijenosa ne može čitati s diska, ali može se paralelno pomicati glavu. Datoteka veličine 25 MiB kompaktno je smještena na disku. Koliko traje čitanje te datoteke u radni spremnik, ako je vrijeme traženja staze  $T_{seek} = 5 \text{ ms}$  te vrijeme postavljanja na susjednu stazu  $T_1 = 0.5 \text{ ms}$ .

$$T_R = \frac{1}{\omega} = \frac{1}{6000 \frac{\text{okr}}{\text{min}} \rightarrow 100 \frac{\text{okr}}{\text{s}}} = 0.01 \text{ s} \quad \frac{T_R}{2} = 0.005 \text{ s} = 5 \text{ ms}$$

2048 za \* 0.5kB (512B) = 1MB → 25 staza potrebno za smještaj podatka

1. Staza

$T_{seek} = 5 \text{ ms}$
$\frac{T_R}{2} = 5 \text{ ms}$
$T_{čitanje} = 10 \text{ ms}$
$T_{prijenos} = \frac{1024 * 1024 * 8}{4.096 * 10^9} = 2.048 \text{ ms}$

2. Staza → i idućih 23 staze

$\frac{T_R}{2} = 5$
$T_{čitanje} = 10 \text{ ms}$
$T_{prijenos} = \frac{1024 * 1024 * 8}{4.096 * 10^9} = 2.048 \text{ ms}$
S obzirom da je vrijeme $T_1 < T_{prijenos}$ → ne uzimamo vrijeme $T_1$ iz razloga što dok traje prijenos pozicioniramo glavu na iduću stazu

$$\begin{aligned} T_{uk} &= 1. \text{ staza} + (2 - 25 \text{ staze}) \\ &= (5 + 5 + 10 + 2.048) + 24 * (5 + 10 + 2.048) \\ &= 431.2 \text{ ms} \end{aligned}$$

9. Neka datoteka veličine 25 MiB kompaktno je smještena na disku koji koristi NTFS particiju s veličinom bloka od 4 KiB. Prvi blok datoteke nalazi se na disku u bloku rednog broja 123456.

a) Navesti dio opisnika datoteke koji opisuje njen smještaj.

VCN	LCN	#
0	123456	6400

b) Ako se pri radu diska pokvario blok diska 123600 koji blok datoteke će biti izgubljen (tj. trebati će ga dohvatiti iz kopije u arhivi)?

$$123600 - \text{LCN} + 1 = 145. \text{ blok u datoteci}$$