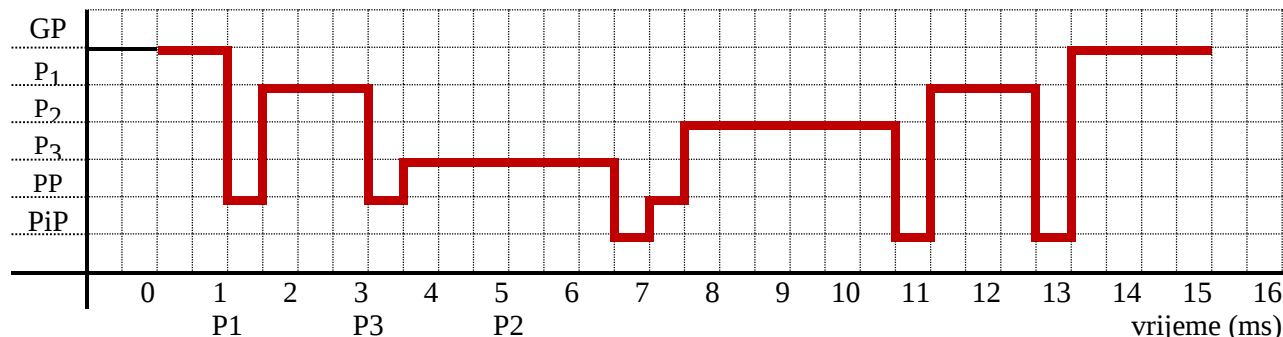


ZEMRIS, 21. 6. 2024.	Ime i prezime	JMBAG
Operacijski sustavi, završni ispit		

Napomena: Pisati čitko! Nečitki odgovori su krivi.

1. (8) U nekom sustavu sa sklopom za prihvat prekida javljaju se prekidi: P1 u trenutku $t=1$ ms, P3 u $t=3$ ms, te prekid P2 u $t=5$ ms. Prioritet prekida određen je brojem (P3 ima najveći prioritet). Grafički prikazati aktivnosti procesora u glavnom programu (GP), procedurama za obradu prekida (P_i) koje traju 3 ms te procedurama za prihvat prekida (PP) koje traju 0,5 ms i povratak iz prekida (P_iP) koje traju 0,5 ms.



Koja je vrijednost u registar kopija zastavica (KZ) u trenutku $t = 6$ ms?

KZ($t=6$) = 010 (P2 čeka početak obrade, zastavica za P1 i P3 su obrisane jer su prihvaćeni)

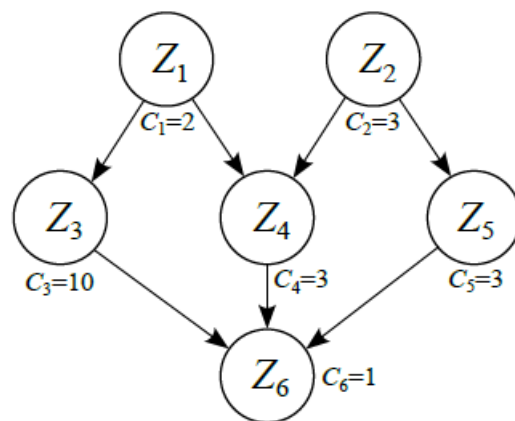
2. (6) Neka naprava koristi sklop za izravni pristup spremniku (DMA). Prosječno u sekundi prenosi 4000 podataka (svaki preko zasebnog sabirničkog ciklusa). Nakon što se prenese 1000 podataka sklop izaziva prekid. Obrada takvog prekida potroši 250 sabirničkih ciklusa. Koji postotak sabirničkih ciklusa treba navedena naprava ako sabirnica radi na 50 MHz?

u sekundi treba 4000 sab.ciklusa za prijenos te $4000/1000 * 250$ za prekide = 5000 ukupno
 $5000 / 50\,000\,000 * 100\% = 0,01\%$

3. (8) U promatranom trenutku stanje sustava je sljedeće: dretva 7 je aktivna; dretve 5, 3 i 1 su u redu pripravnih dretvi, dretva 4 je u redu općeg semafora S te dretva 6 je u redu odgođenih – treba još 5 otkucaja sata da se od tamo makne. Svi redovi organizirani su po prioritetu gdje veći broj dretve označava veći prioritet (osim odgođenih i redova naprava). Pozivi koji se tada događaju su redom, jedan nakon drugoga: PostaviOSEM(S), ZapočniUI(1), Odgodi(7) te Prekid_sata(). Pokazati stanje sustava nakon svakog od ovih poziva.

	Početno stanje	Nakon PostaviOSEM(S)	Nakon ZapočniUI(1)	Nakon Odgodi(7)	Nakon Prekid_sata()
Aktivna_D	7	7	5	4	4
Pripravne_D	5 3 1	5 4 3 1	4 3 1	3 1	3 1
Odgođene_D	6 ⁵	6 ⁵	6 ⁵	6 ⁵ 5 ²	6 ⁴ 5 ²
OSEM[S]	4 (.v=0)	- (.v=0)	- (.v=0)	- (.v=0)	- (.v=0)
Red UI[1]	-	-	7	7	7

4. Zadan je sustav zadataka prema slici desno. Uz svaki zadatak zadano je i koliko procesorskog vremena treba za izvođenje (C_n). Sustav treba sinkronizirati općim semaforima (OSEM).



- a) (6) Navesti potrebne semafore i njihove početne vrijednosti te (na košuljicu) napisati pseudokod za zadatak uz pretpostavku da je koristan posao zadatka Z_i označen s T_i .

```

Z1: T1; PostaviOSEM(1); PostaviOSEM(2);
Z2: T2; PostaviOSEM(2); PostaviOSEM(3);
Z3: ČekajOSEM(1); T3; PostaviOSEM(4);
Z4: ČekajOSEM(2); ČekajOSEM(2); T4; PostaviOSEM(4);
Z5: ČekajOSEM(3); T5; PostaviOSEM(4);
Z6: ČekajOSEM(4); ČekajOSEM(4); ČekajOSEM(4); T4;
  
```

- b) (2) Koliko će trajati izvođenje sustava na dvoprocesorskom sustavu?

{Z1,Z2},{Z2,Z3},{Z3,Z4},{Z3,Z5},{Z3,-},{Z6,-}
 0-2 2-3 3-6 6-9 9-12 12-13 => **13**

5. U nekom sustavu se nalazi sedam proizvođača i tri potrošača koji komuniciraju preko zajedničkog međuspremnik kodom prikazanim na predavanjima.

- a) (1) Koliko je minimalno semafora potrebno za sinkronizaciju (za maksimalno dozvoljeni paralelni rad)?

2 opća (broje puna i prazna mjesta u ms) + 2 binarna (za pristup ms)

- b) (1) Ako je u međuspremniku veličine deset podataka sedam nepročitanih, koja je trenutna vrijednost semafora kojeg ispituju proizvođači prije pristupa međuspremniku?

tri su mjesta prazna => OSEM[1].v = 3

6. (6) U nekom sustavu se izvode dretve 1a, 1b, 2a, 2b, 3a, 3b. Sve se raspoređuju prema prioritetu kao primarnom kriteriju te podjelom vremena kao sekundarnom uz kvant $T_q=1$ (način SCHED_RR). Prioritet dretvi određen je njenim indeksom - 3a i 3b imaju najveći prioritet 3. Dretve se javljaju u sustavu u trenutcima: $(t, D) = \{(0, 2a), (1, 1a), (2, 2b), (3, 3a), (5, 1b), (9, 3b)\}$. Pokazati izvođenje dretvi do $t=20$ ukoliko svaka dretva treba četiri jedinice vremena za svoje izvođenje. U trenucima kad se više događaja događa istovremeno proizvoljno odabrati redoslijed.

Pripravne dretve (nije potrebno pisati)																					
							1b	1b				1b	1b	1b	1b						
					1a	1a	1a	1a	1b	1b	1a	1a	1a	1a	1b						
				1a	2b	2b	2b	2b	1a	1a	2b	2b	2b	2b	1a	1b	1b				
		1a	2a	2a	2a	2a	2a	2b	2a	2a	2a	2a	2a	2a	2b	1a	1a	1b	1a	1b	1a
Aktivna	2a	2a	2b	3a	3a	3a	3a	2a	2b	3b	3b	3b	3b	2a	2b	2b	1a	1b	1a	1b	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

7. (3) Za neki proces zadana je tablica prevođenja (desno). Ako je veličina stranice 256 B koju će fizičku adresu generirati sklop za logičku adresu 0x0281?

iz veličine stranice vidimo da treba 8 bita za odmak: zadnje dvije heksa znamenke se kopiraju, a one prije označavaju adresu stranice/okvira

0x0281 => stranica = 0x02, odmak = 0x81

stranica 0x02=2 je u okviru 0x23 (prema tablici)

pretvorba 0x0281 => 0x2381

str.	okvir	bit pris.
0	0x16	1
1	-	0
2	0x23	1
3	0x48	1
4	-	0

8. (2) Neki današnji 64-bitovni procesi kod straničenja koriste 48-bitovne logičke adrese, veličinu stranice od 4 KB, veličinu zapisa (retka) u tablici prevođenja (jednog opisnika stranice) od 64 bita te četverorazinsku tablicu prevođenja. Neki mali proces treba 10 KB kompaktnog radnog spremnika (tri uzastopne stranice). Koliko će dodatnog prostora OS trebati za tablicu prevođenja (koja sadrži minimalno potrebno podataka)?

minimalno treba po jedna stranica u svakoj razini (u svakoj stranici samo jedan upisnik za iduću razinu osim u zadnjoj gdje se koriste tri za tri stranice procesa): 4 x 4 KB = 16 KB

9. (6) U sustavu koji koristi dinamičko upravljanje spremnikom s 8 MB spremnika javljaju se zahtjevi za pokretanjem procesa te završetka prema redu: +P4, +P2, -P4, +P1, +P3, -P2, +P2, -P3, +P4, -P2, +P5. Plus ispred oznake procesa +PX označava zahtjev za pokretanje procesa PX koji treba X MB radnog spremnika, dok -PX označava kraj izvođenja tog procesa. Prikazati stanje spremnika za zadane zahtjeve sve do kraja ili do trenutka kada se zahtjev ne može poslužiti (nema dovoljno velike rupe).

zahtjev	radni spremnik							
+P4	4	4	4	4				
+P2	4	4	4	4	2	2		
-P4					2	2		
+P1					2	2	1	
+P3	3	3	3		2	2	1	
-P2	3	3	3				1	
+P2	3	3	3	2	2		1	
-P3				2	2		1	
+P4	nema mjesta							
-P2								
+P5								

Pri posluživanju zahtjeva se uvijek korisni najmanja rupa koja je dovoljno velika

10. (4) Pokazati primjenu strategija LRU i OPT nad nizom zahtjeva za stranicama: 1 2 8 1 3 4 8 1 5 6 7 3 1 8 ako su na raspolaganju četiri okvira.

		1	2	8	1	3	4	8	1	5	6	7	3	1	8
LRU	-	(1)	1	1	0	1	1		0	1	1	1	(3)	3	3
	-	-	(2)	2		2	(4)			4	(6)	6	6	6	(8)
	-	-	-	(8)		8	8	0		8	8	(7)	7	7	7
	-	-	-	-		(3)	3			(5)	5	5	5	(1)	1
OPT	-	(1)	1	1	0	1	1		0	1	1	1		0	
	-	-	(2)	2		2	(4)			(5)	(6)	(7)			
	-	-	-	(8)		8	8	0		8	8	8			0
	-	-	-	-		(3)	3			3	3	3	0		

11. Neka datoteka pohranjena je na disku koji koristi NTFS. U opisniku te datoteke nalazi se tablica desno.

VCN	LCN	#
1	4000	3000
3001	40000	5000

a) (2) Ako je blok velik 4 KB, koliko je velika datoteka?

$$(\Sigma\# = 3000 + 4000) * 4 \text{ KB} = 28000 \text{ KB}$$

b) (3) U kojem bloku diska se nalazi 5432. blok datoteke?

$$\text{očito je u drugom dijelu (veći od 3000): } 5432 - 3001 = X - 40000 \Rightarrow X = 42431$$

12. (3) Usporediti poredati RAID 1, RAID 5 i RAID 6 u tri aspekta:

najveći kapacitet: **RAID 5, RAID 6, RAID 1**

najbrži rad: **RAID 5, RAID 6, RAID 1 (RAID 1 je brz samo u čitanju)**

otpornost na kvarove: **RAID 6, RAID 1, RAID 5**

13. (9 bodova) Problem sličan problemu pet filozofa pojavljuje se u „Restoranu na kraju svemira“. Tamo za istim stolom sjede osobe s različitim brojem ruku i različitim običajima hranjenja. Pretpostaviti da se na sredini stola nalazi 20 vilica i da svaka osoba koja jede mora u svakoj ruci imati jednu vilicu kad jede. Neke osobe glasno jedu što ometa neke druge osobe koje ne mogu istodobno jesti uz taj zvuk. Stoga osobe koje glasno jedu neće započeti jesti ako neka druga osoba kojoj to smeta već jede, ali i osoba kojoj smeta glasno jedenje neće početi jesti ako neka druga tada glasno jede (pričekati će da završi). Nakon što se najede osoba opere vilice i vraća ih na sredinu stola. Simulirati osobe dretvama i sinkronizirati ih monitorima. Pretpostaviti da su parametri za dretvu broj ruku (*br_ruku*), oznaka (0/1) da li osoba jede glasno (*glasno_jede*) te oznaka (0/1) smeta li osobi glasno hranjenje drugih osoba (*smeta_glasno*). Napisati početne vrijednosti korištenih varijabli. Nije potrebno rješavati problem izgladnjivanja, ali ako postoji u rješenju navesti scenarij njegove pojave.

```

dretva osoba(br_vilica, glasno_jede, smeta_glasno)
{
    ponavlja {
        razmišlja;

        uđi_u_monitor(m)
        dok je (vilica_na_stolu < br_vilica)
            čekaj_u_redu_uvjeta(red_vilice, m)
        vilica_na_stolu -= br_vilica

        dok je (glasno_jede == ISTINA I br_smeta_glasno > 0) ILI
            (smeta_glasno == ISTINA I br_glasno_jedu > 0)
            čekaj_u_redu_uvjeta(glasno, m)

        ako je (glasno_jede == ISTINA)
            br_glasno_jedu++
        ako je (smeta_glasno == ISTINA)
            br_smeta_glasno++
        izađi_iz_monitora(m)

        jede;

        uđi_u_monitor(m)
        vilica_na_stolu += br_vilica
        oslobodi_sve_iz_reda_uvjeta(red_vilice)
        ako je (glasno_jede == ISTINA)
            br_glasno_jedu--
        ako je (smeta_glasno == ISTINA)
            br_smeta_glasno--
        oslobodi_sve_iz_reda_uvjeta(glasno)
        izađi_iz_monitora(m)
    }
}

globalne varijable s početnim vrijednostima:
vilica_na_stolu = 20
br_glasno_jedu = 0
br_smeta_glasno = 0

```

kraće:

```

dretva osoba(br_vilica, glasno_jede, smeta_glasno) //0/1 za *jede/*smeta
{
    ponavlja {
        razmišlja;

        uđi_u_monitor(m)
        dok je (vilica_na_stolu < br_vilica) ||
            (glasno_jede I br_smeta_glasno) || (smeta_glasno I br_glasno_jedu)
            čekaj_u_redu_uvjeta(red, m)
        vilica_na_stolu -= br_vilica
        br_glasno_jedu += glasno_jede
        br_smeta_glasno += smeta_glasno
        izađi_iz_monitora(m)

        jede;

        uđi_u_monitor(m)
        vilica_na_stolu += br_vilica
        br_glasno_jedu -= glasno_jede
        br_smeta_glasno -= smeta_glasno
        oslobodi_sve_iz_reda_uvjeta(red)
        izađi_iz_monitora(m)
    }
}

```