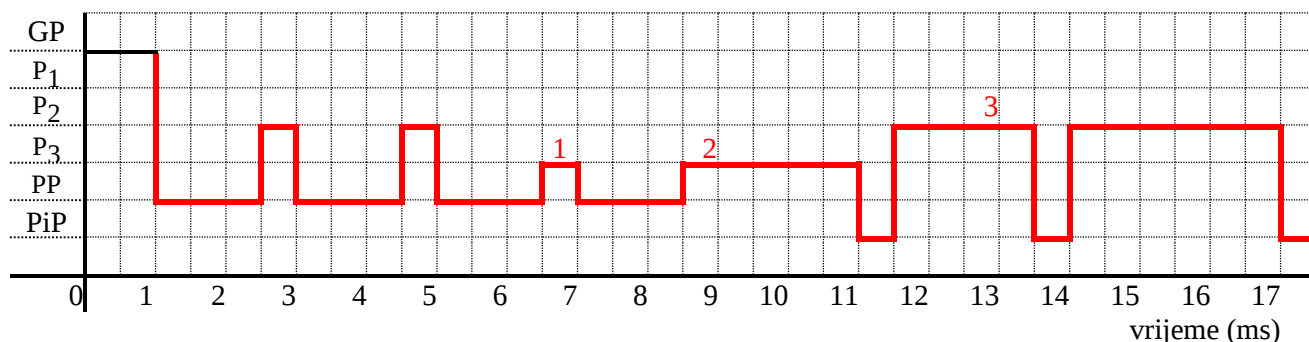


ZEMRIS, 25. 4. 2025.	Ime i prezime	JMBAG
OS, međuispit, grupa B		

Napomena: Postupak za sve računske zadatke treba biti na ovim papirima, košuljici ili dodatnim papirima koje priložiti. Na obrascu za rješenja označiti rješenja za zadatke 1-26. Točno rješenje nosi 1 bod, netočno -0,25. Ako je odgovor E onda obavezno na obrascu navesti točno rješenje (koje nije već ponuđeno). Zadatak 27 rješavati na ovim papirima (zadnjoj stranici).

Z1-3 U nekom sustavu bez sklopa za prihvata prekida ali s programskom potporom za prihvata prekida prema prioritetu javljaju se prekidi: P2 u trenucima $t=1$ ms i $t=7$ ms, P1 u $t=3$ ms, te prekid P3 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, proceduri za prihvata prekida (PP) koja traje 1,5 ms te proceduri za povratak iz prekida (PiP) koja traje 1 ms, do trenutka $t=17,5$.



1. U trenutku $t = 6,75$ ms izvodi se:

- a) P1 b) P2 c) P3 d) PP e) nešto drugo

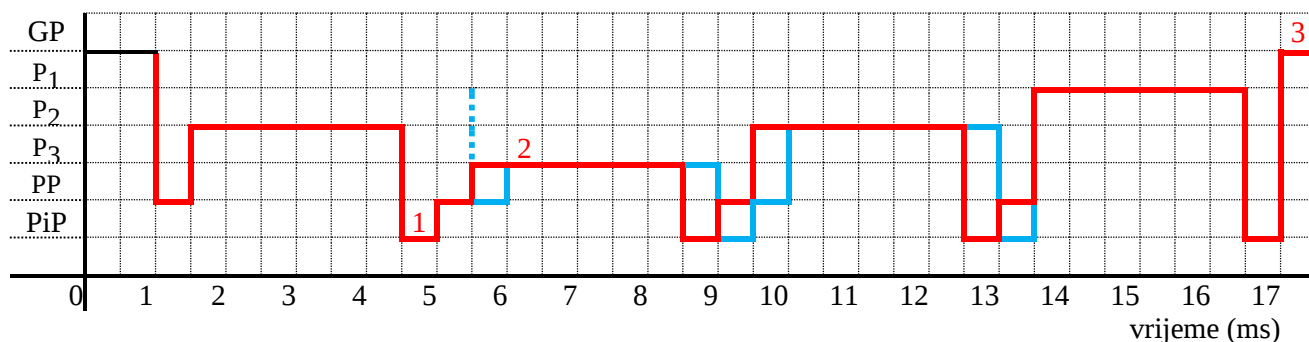
2. U trenutku $t = 9$ ms polje OZNAKA_ČEKANJA[] (ili K_Z[]) ima vrijednost (binarno - bit za P1 je lijevo):

- a) 0 0 0 b) 1 0 0 c) 1 1 0 d) 1 1 1 e) nešto drugo
 $P_1 P_2 P_3$ $P_1 P_2 P_3$ $P_1 P_2 P_3$ $P_1 P_2 P_3$

3. U trenutku $t = 13$ ms izvodi se:

- a) GP b) P1 c) P2 d) P3 e) nešto drugo

Z4-6 U nekom sustavu sa sklopom za prihvat prekida prema prioritetu javljaju se prekidi kao i u Z1-3 (P2 u $t=1$ ms i $t=7$ ms, P1 u $t=3$ ms, te P3 u $t=5$ ms). Prioritet prekida određen je brojem (P3 ima najveći prioritet). Grafički prikazati aktivnosti procesora u glavnom programu, procedurama za obradu prekida koje traju 3 ms, proceduri za prihvat prekida koja traje 0,5 ms i proceduri za povratak iz prekida koja traje 0,5 ms, do $t=17,5$ ms.



4. U trenutku $t = 4,75$ ms izvodi se:

- a) PP b) PiP c) P1 d) P2 e) nešto drugo

5. U trenutku $t = 6,25$ ms registar sklopa K_Z (kopije zastavica) ima vrijednost (binarno - bit za P1 je lijevo):

- a) 0 0 0 b) 1 0 0 c) 1 1 0 d) 1 1 1 e) nešto drugo
 $P_1 P_2 P_3$ $P_1 P_2 P_3$ $P_1 P_2 P_3$ $P_1 P_2 P_3$

6. U trenutku $t = 17,25$ ms izvodi se:

- a) GP b) P1 c) P2 d) P3 e) nešto drugo

7. Neka naprava koristi sklop za izravni pristup spremniku (DMA). Prosječno u sekundi prenosi 10000 podataka (svaki preko zasebnog sabirničkog ciklusa). Koji postotak sabirničkih ciklusa treba navedena naprava ako sabirnica radi na 10 MHz? Zanimariti trajanje obrade prekida nakon prijenosa bloka podataka.

- a) 0,01 % b) 0,1 % c) 0,001 % d) 1 % e) nešto drugo

Z8-13 Sustav zadataka je zadan u obliku lanca (kada se zadaci izvode ovim redom rezultat će biti ispravan): $Z1 \rightarrow Z2 \rightarrow Z3 \rightarrow Z4 \rightarrow Z5 \rightarrow Z6 \rightarrow Z7 \rightarrow Z8$. Zadaci imaju domene (D) i kodomene (K) prema tablici.

	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8
M1	D	D	K		K		D	
M2	K		D		D			K
M3		K		D		K		D
M4				K	D		K	
M5						D		

- a) Odrediti maksimalno paralelni sustav zadataka, uzimajući u obzir njihov međusobni odnos u lancu i domene i kodomene prema tablici.
b) Sinkronizirati sustav zadataka binarnim semaforima (proširiti $Tx \rightarrow Tx'$ pozivima sa semaforima), naznačiti početne vrijednosti svih semafora.

8. Koliko se najviše zadataka može izvoditi paralelno?

- a) 1 b) 2 c) 3 d) 4 e) nešto drugo

9. Koliko neposrednih prethodnika ima zadatak Z6?

- a) 1 b) 2 c) 3 d) 4 e) nešto drugo

10. Koliko neposrednih sljedbenika ima zadatak Z6?

- a) 1 b) 2 c) 3 d) 4 e) nešto drugo

11. Koliko je binarnih semafora potrebno za sinkronizaciju?

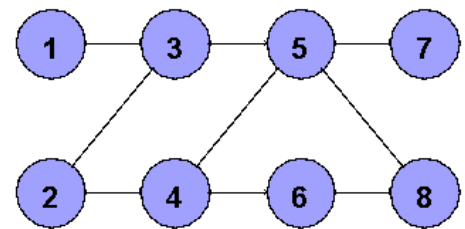
- a) 10 b) 9 c) 7 d) 4 e) nešto drugo

12. Koje su početne vrijednosti binarnih semafora na koje čeka Z3?

- a) 0 b) 1 c) 2 d) 3 e) nešto drugo

13. Koliko ČekajBSEM (IspitatiBSEM) poziva je potrebno u T7?

- a) 1 b) 2 c) 3 d) 0 e) nešto drugo



Z14-17 U promatranom trenutku stanje sustava („Početno stanje“ u tablici) je sljedeće: dretva 7 je aktivna; dretve 5, 3 i 1 su u redu pripremljenih dretvi, dretva 4 je u redu općeg semafora S te dretve 6 i 2 u redu odgođenih – dretvi 6 treba još 9 otkucaja sata da se od tamo makne a dretvi 2 još 15 otkucaja nakon micanja dretve 6. Svi redovi organizirani su po prioritetu gdje veći broj dretve označava veći prioritet (osim odgođenih i redova naprava). Pozivi jezgrinih funkcija koji se tada događaju su redom, jedan nakon drugoga zadani u tablici. Pokazati stanje sustava nakon svakog od ovih poziva (svaki idući poziv odnosi se na prethodno stanje, ne početno).

	Početno stanje	Nakon Otkucaj_sata()	Nakon Zakasni(5)	Nakon Prekid_UI(1)	Nakon PostaviOSEM(S)
Aktivna_D	7	7	5	5	5
Pripravne_D	5 3 1	5 3 1	3 1	4 3 1	4 3 1
Odgođene_D	6 ⁹ 2 ¹⁵	6 ⁸ 2 ¹⁵	7 ⁵ 6 ³ 2 ¹⁵	7 ⁵ 6 ³ 2 ¹⁵	7 ⁵ 6 ³ 2 ¹⁵
OSEM[S]	-(.v=0)	-(.v=0)	-(.v=0)	-(.v=0)	-(.v=1)
Red UI[1]	4	4	4	-	-

14. Koja je vrijednost u zadano_kašnjenje u opisniku dretve koja poziva Zakasni(5) nakon tog poziva?

- a) 4 b) 5 c) 9 d) 14 e) nešto drugo

15. Koje sve dretve neće biti aktivne tijekom ovog isječka vremena, za vrijeme svih ovih poziva?

- a) 1, 3 b) 2, 6 c) 1, 2, 3, 6 d) 1, 2, 3, 4, 6 e) nešto drugo

16. Koja je vrijednost semafora OSEM[S] nakon poziva PostaviOSEM(S)?

- a) 0 b) 1 c) -1 d) 2 e) nešto drugo

17. Koja je dretva aktivna nakon poziva Prekid_UI(1)?

- a) 3 b) 4 c) 5 d) 7 e) nešto drugo

Z18-20 U nekom sustavu se nalazi šest proizvođača i pet potrošača koji komuniciraju preko zajedničkog međuspremnik veličine 20 podataka algoritmom prikazanim na predavanjima.

18. Koliko je minimalno semafora potrebno za sinkronizaciju (za maksimalno paralelni rad)?
a) 1 opći + 1 binarni b) 1 opći + 2 binarna c) 2 opća + 1 binarni d) 2 opća + 2 binarna e) nešto drugo
19. Ako je u međuspremniku 15 nepročitanih poruka, koja je trenutna vrijednost semafora kojeg ispituju potrošači prije pristupa međuspremniku?
a) 15 b) 5 c) 20 d) 0 e) nešto drugo
20. Ako je u međuspremniku 5 nepročitanih poruka, koja je trenutna vrijednost semafora kojeg ispituju proizvođači prije pristupa međuspremniku?
a) 5 b) 15 c) 20 d) 0 e) nešto drugo
21. Kažemo da OS „olakšava“ korištenje računala. Koja od slijedećih izjava nije istinita?
a) korisnici koji znaju raditi na jednom računalu, znat će raditi i na drugom ako je i tamo isti OS
b) programi koji rade na jednom računalu, radit će i na drugom (različitom) koje ima isti OS (i tip procesora)
c) računalo na kome je instaliran neki OS je lakše za prenositi s jednog mjesta na drugo
d) programi ne moraju imati ugrađene upravljačke programe za naprave već ih koriste preko OS-a
e) nešto drugo
22. Najbrže će raditi (s najmanje dodatnog posla hipervizora) virtualna računala u virtualizaciji:
a) tipa 3 (virtualizacija na razini dretve)
b) tipa 2 (virtualizacija na razini programa)
c) tipa 1 (virtualizacija na razini OS-a)
d) tipa 0 (virtualizacija na razini sklopovlja)
e) nešto drugo
23. Pri zamjeni jedne dretve drugom moramo spremati kontekst prve dretve i obnoviti kontekst druge. Što sve spada u taj kontekst dretve?
a) svi registri procesora, opisnik dretve
b) svi registri procesora, stog dretve
c) svi registri procesora
d) svi registri procesora, stog dretve, opisnik dretve
e) nešto drugo
24. U sustavu koji upravlja s nekoliko aktivnosti, neka naprava generira blokove podataka. Najprikladniji način upravljanja tom napravom jest:
a) radno čekanje b) prekidi c) korištenjem sklopa za izravni pristup spremniku
d) svi su jednako dobri e) nešto drugo
25. Dvije su dretve (i i j) zavisne (ne smiju se izvoditi paralelno) ako i samo ako vrijedi (D -domena, K -kodomena):
a) $(D_i \cap D_j) \cup (K_i \cap K_j) \neq \emptyset$
b) $(D_i \cap D_j) \cup (D_i \cap K_j) \cup (K_i \cap D_j) \neq \emptyset$
c) $(D_i \cap K_j) \cup (K_i \cap D_j) \cup (K_i \cap K_j) \neq \emptyset$
d) $(D_i \cap D_j) \cup (D_i \cap K_j) \cup (K_i \cap D_j) \cup (K_i \cap K_j) \neq \emptyset$
e) nešto drugo
26. Na jednoprocesorskom računalu u jezgru OS-a ulazi se prekidom jer:
a) jer posluživanje naprava možemo raditi samo mehanizmom prekida
b) jer je manje kućanskog posla od izravnog poziva funkcije
c) u obradi prekida nas nitko ne može prekinuti (obrada prekida == kritični odsječak)
d) je to brže od izravnog poziva funkcije
e) nešto drugo

27. (4) U nekom sustavu nasumice se stvaraju dretve s početnom funkcijom `dretva(tip)` (`tip` je 0 ili 1). Dretve u svom radu prvo generiraju jedan znak (funkcijom `gen()` koju mogu pozivati i paralelno). Potom ga stavljaju u međuspremnik `M` (kapaciteta jednog znaka) na sljedeći način: ako je u `M` malo slovo (što svaka dretva može provjeriti uvjetom `test(M)==0`), idući znak u `M` stavlja dretva s `tip=0`; ako je u `M` veliko slovo (`test(M)==1`) idući znak u `M` stavlja dretva s `tip=1`; inače, ako je neki drugi znak u `M` (`test(M)==2`) onda bilo koja dretva može staviti znak u `M`. Nakon stavljanja znaka u `M` dretva mora pozvati `pohrani(M, tip)` (također u kritičnom odsječku). Početno se u `M` nalazi znak `+`. Sinkronizirati dretve binarnim i/ili općim semaforima te po potrebi dodatnim varijablama. Navesti početne vrijednosti svih korištenih semafora i varijabli. Radno čekanje nije dozvoljeno. *Preporuka: prvo riješiti na pomoćnim papirima pa onda prepisati rješenje ovdje.*

```
dretva(tip)
{
    znak = gen()
    ...
    M = znak
    pohrani(M, tip)
    ...
}
```

Jedno rješenje (slično „ping-pong: barem 2 ping prije svakog ponga“ koji smo radili na predavanju)

```
dretva(tip) {
    znak = gen()
    ČekajBSEM(KO)
    dok je (test(M) == 1 - tip) {          //ili: (test(M)==1 I tip==0) ILI
        PostaviBSEM(KO)                   //      (test(M)==0 I tip==1)
        ČekajBSEM(tip)
        ČekajBSEM(KO)
    }
    M = znak
    pohrani(M, id)
    ako je (test(M) == 2)                  // ili uvijek postavi semafore 0 i 1
        PostaviBSEM(0)                   // ionako oni provjeravaju što je u M
        PostaviBSEM(1)                   //
    inače                                  //
        PostaviBSEM(test(M))             //
    PostaviBSEM(KO)
}
Početne vrijednosti:
M='+' – zadano
BSEM[KO].v = 1
BSEM[0].v = 0      moraju biti dva semafora inače se može dogoditi
BSEM[1].v = 0      radno čekanje; nije greška ni .v=1
```

Drugo rješenje (sličnije „pušačima“)

```
dretva(tip) {
    znak = gen()
    postavio = 0 //lokalna varijabla
    ponavlja {
        ČekajBSEM(tip)
        ČekajBSEM(KO)
        ako je (test(M)==tip ili test(M)==2) { //ili test(M) != 1 - tip
            M=znak
            pohrani(M, tip)
            PostaviBSEM(0)           //ili s „ako je“ kao i gore
            PostaviBSEM(1)           //
            postavio=1               //ili znak = 0 ili -1 i sl.
        }
        PostaviBSEM(KO)
    }
    dok je postavio == 0             //ili znak != 0
}
Početne vrijednosti:
M='+' – zadano
BSEM[KO].v = 1
BSEM[0].v = 1      mora biti 1 u ovom rješenju
BSEM[1].v = 1
```

Može rješenje i sa zasebnim kodovima za `dretva(0){}` i `dretva(1){}`

Može rješenje i s rekurzivnim pozivom umjesto petlje (ali takva rješenja ne preporučamo).