

# Prvi kolokvij iz predmeta *Operacijski sustavi*

6. svibnja 2022.

Student

1. (2) Navesti prednosti korištenja datoteke `Makefile` radi izgradnje nekog netrivialnog programa.

- ne treba svaki puta upisivati duge naredbe za prevođenje
- prevode se samo datoteke koje su mijanjene u međuvremenu

2. (2) Koja će instrukcija promijeniti stanje iz:

PC=0x1000, SP=0xF008, memorija:{F000:5000, F004:4000, F008:3000, F00C:2000}

u:

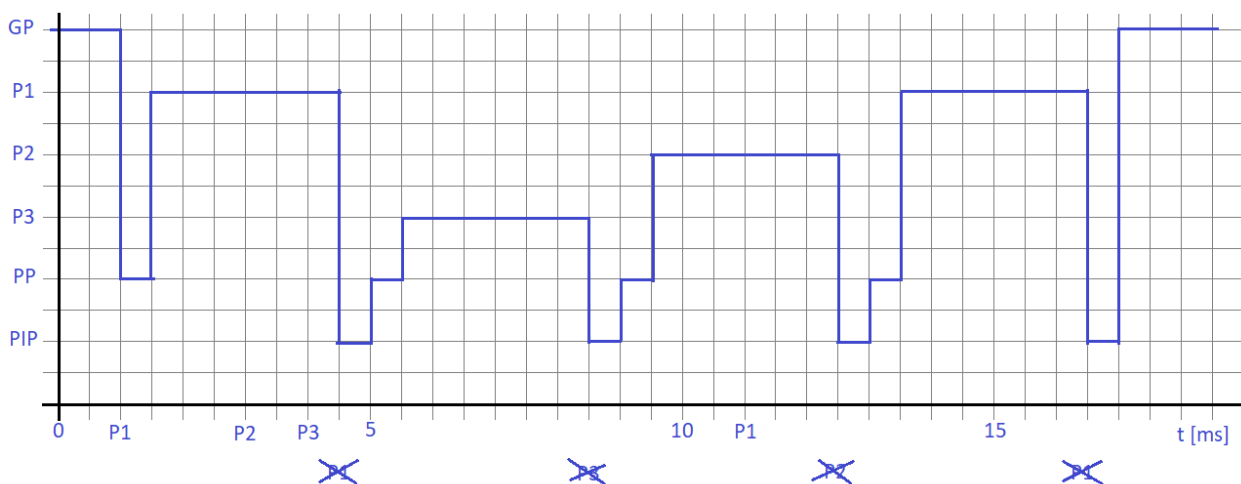
PC=0x8000, SP=0xF004, memorija:{F000:5000, F004:1004, F008:3000, F00C:2000}?

`CALL 0x8000`

3. (2) Pri prihvatu prekida, procesor mora obaviti:

1. zabrani daljnje prekidanje
2. prebaci se u prekidni način rada
3. na prekidni stog pohrani PC i RS
4. u PC stavi adresu prekidnog potprograma

4. (4) U nekom sustavu bez sklopa za prihvata prekida javljaju se prekidi:  $P_1$  u 1. i 11.,  $P_2$  u 3. te  $P_3$  u 4. ms. Prioritet prekida, koji određuje redoslijed provjere RS pristupnih sklopova u prekidnom potprogramu, određen je brojem ( $P_3$  ima najveći prioritet). Procedura za prihvata prekida (PP) traje 0,5 ms a procedura za povratak iz prekida (PiP) 0,5 ms. Obrada svakog prekida traje po 3 ms. Grafički prikazati aktivnosti procesora u glavnom programu (GP), procedurama za obradu prekida ( $P_i$ ) te procedurama za prihvata prekida (PP) i povratak iz prekida (PiP).



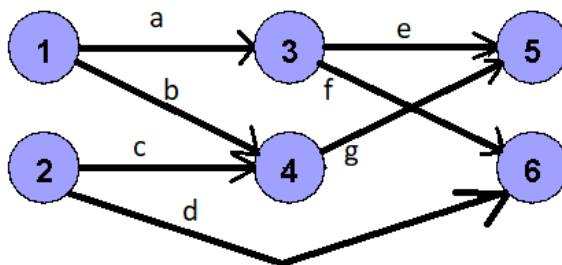
Rješenja koja su obrađivala prekide prema prioritetu su bodovana s 2 boda, ako su bila točna po tom načinu.

Međutim to u zadatku nije bilo zadano i na predavanju je naglašeno da će se samo onaj prvi način prihvata pitati u zadacima.

5. (5) Neki posao podijeljen je u šest zadataka  $Z_1$  do  $Z_6$ . Poznato je da će rezultat biti ispravan ako se zadaci izvode slijedno:  $Z_1 \rightarrow Z_2 \rightarrow Z_3 \rightarrow Z_4 \rightarrow Z_5 \rightarrow Z_6$ . Domene i kodomene zadataka su sljedeće:  $Z_1 : \{D_1 = M_1, K_1 = M_2\}$ ,  $Z_2 : \{D_2 = M_1, K_2 = M_3\}$ ,  $Z_3 : \{D_3 = M_2, K_3 = M_4\}$ ,  $Z_4 : \{D_4 = M_2, K_4 = M_1\}$ ,  $Z_5 : \{D_5 = M_3, K_5 = M_2\}$ ,  $Z_6 : \{D_6 = M_3, K_6 = M_4\}$ .

- a) (3) Odrediti maksimalno paralelni sustav zadataka (i prikazati ga usmjerenim grafom) uzimajući u obzir njihove domene i kodomene te međusobni odnos u lancu.
- b) (2) Sinkronizirati sustav semaforima (binarnim/općim, po izboru). Neka je tekst zadatka  $Z_i$  opisan sa  $T_i$ . Proširiti ga pozivima sa semaforima (prije i/ili poslije  $T_i$ ). Navesti početne vrijednosti semafora.

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



T1' : T1;  
 PostaviBSEM(a);  
 PostaviBSEM(b);

T2' : T2;  
 PostaviBSEM(c);  
 PostaviBSEM(d);

T3' : ČekajBSEM(a);  
 T3;  
 PostaviBSEM(e);  
 PostaviBSEM(f);

T4' : ČekajBSEM(b);  
 ČekajBSEM(c);  
 T4;  
 PostaviBSEM(g);

T5' : ČekajBSEM(e);  
 ČekajBSEM(g);  
 T5;

T6' : ČekajBSEM(d);  
 ČekajBSEM(f);  
 T6;

6. (2) Koje uvjete na algoritme međusobnog isključivanja ne zadovoljava sljedeći algoritam? Navesti i scenarije u kojima su uvjeti prekršeni.

```

uđi_u_KO(I) /* I je 0 ili 1 */   izađi_iz_KO(I)
{
    J = 1 - I
    ZASTAVICA[I] = 1
}

dok je (ZASTAVICA[J] != 0)
    ;
}

```

Početne vrijednosti:  
ZASTAVICA[0] = 0  
ZASTAVICA[1] = 0

u paralelnom radu obje dretve dižu zastavice i "zaglave" u petlji neispunjeni uvjet: 4. odluka se donosi u konačnom vremenu

7. (2) Kako se pozivaju jezgrine funkcije? Zašto baš tako?

prekidima: => procesor prelazi u privilegirani način rada,  
=> KO za jednoprocorske sustave

8. (2) Usporediti prednosti i nedostatke ostvarenja kritična odsječka pomoću algoritma koji koristi instrukciju Ispitaj\_i\_postavi ZASTAVICA (Test\_and\_set => TAS ZASTAVICA) te ostvarenje s binarnim semaforima (ČekajBSEM(KO), PostaviBSEM(KO)). Usporedbu napraviti za dva scenarija. U prvom se često događa da različite dretve paralelno traže ulaz u kritični odsječak (koji "dugo" traje). U drugom scenariju se vrlo rijetko događaju takve situacije iako se ulaz u KO traži puno češće (i traje "puno" kraće).

Oba ostvarenja su ispravna, razlika je samo u učinkovitosti  
1. scenarij: bolje je koristiti semafore, jer uz TAS dolazi do izražaja radno čekanje  
2. scenarij: bolje je koristiti TAS, je poziv jezgrinih funkcija uključuje dosta kućanskog posla (spremi kontekst, obnovi kontekst)

9. (4) Stanje nekog sustava u promatranom trenutku je sljedeće: dretva  $D_3$  je aktivna,  $D_1$  je u redu pripravnih,  $D_4$  u redu semafora OSEM[1], te  $D_2$  i  $D_5$  u redu odgođenih:  $D_2$  treba čekati pet otkucaja sata, a  $D_5$  18 otkucaja (ukupno). Red pripravnih dretvi je uređen prema prioritetu. Prioritet dretve određen je njenim indeksom – dretve s većim indeksom imaju veći prioritet ( $D_5$  ima najveći prioritet). U tom stanju dogodi se sljedeći niz poziva jezgrinih funkcija (svaki idući se događa neko kratko vrijeme nakon što je prethodni bio gotov): 1. Odgodi(9). 2. Postavi\_OSEM(1) 3. Započni\_UI(1) 4. Otkucaj\_sata()  
Prikazati stanje sustava nakon svakog od tih poziva.

	poč.st.	nakon 1.	nakon 2.	nakon 3.	nakon 4.
Aktivna_D	3	1	4	1	1
Pripravne_D	1	-	1	-	-
OSEM[1]	4	4	-(.v=0)	-(.v=0)	-(.v=0)
Odgođene_D	$2^5 \ 5^{13}$	$2^5 \ 3^4$	$2^5 \ 3^4$	$2^5 \ 3^4$	$2^4 \ 3^4$
		$5^9$	$5^9$	$5^9$	$5^9$
UI[1]	-	-	-	4	4
Iduća j.f.	Odgodi(9)	Postavi OSEM(1)	Započni UI(1)	Otkucaj sata()	

10. (5) Dretva *A* dohvaća novi posao sa  $p = \text{dohvati}()$  te ga potom stavlja u red sa  $\text{stavi}(p)$ . Dretve tipa *B* dohvaćaju poslove iz tog reda sa  $r = \text{uzmi}()$  i obrađuju sa  $\text{obradi}(r)$ . Zadan je pseudokod dretve *A*, koji uključuje i sinkronizaciju **binarnim semaforima**. Napisati pseudokod za dretve tipa *B*. Navesti početne vrijedosti varijabli i semafora.

```
dretva A {
    ponavljaaj {
        p = dohvati()
        ČekajBSEM(KO)
        stavi(p)
        br_poslova++
        PostaviBSEM(KO)
        PostaviBSEM(P)
    }
}
```

```
dretva B {
    ponavljaaj {
        ...
        r = uzmi()
        br_poslova--
        ...
        obradi(r)
    }
}
```

```
dretva B {
    ponavljaaj {
        ČekajBSEM(KO)
        dok je br_poslova == 0 radi
            PostaviBSEM(KO)
            ČekajBSEM(P)
            ČekajBSEM(KO)
        r = uzmi()
        br_poslova--
        PostaviBSEM(KO)
        obradi(r)
    }
}
```

Komentar: uobičajeno rješenje studenata je:

```
dretva B {
    ponavljaaj {
        ČekajBSEM(P)
        ČekajBSEM(KO)
        r = uzmi()
        br_poslova--
        PostaviBSEM(KO)
        obradi(r)
    }
}
```

Međutim, ovo rješenje nije ispravno!

Ispravno bi bilo samo kad bi semafor *P* bio opći a ne binarni.

Ipak, za ovakvo rješenje dodjeljeno je 2,5 bodova.