Prvi kolokvij iz predmeta *Operacijski sustavi*

4. svibnja 2023. Student _____

1. (1) Zašto OS omogućuje prenosivost programa na računala koja imaju dosta različito sklopovlje, ali kompatibilan procesor. Također, zašto korisnici koji rade na jednom računalu mogu bez "podučavanja" raditi i na drugom s istim OS-om?

   OS programi koriste preko sučelja za programe (API) a korisnici preko sučelja za korisnike (GUI)

   Sučelje OS-a je jednako bez obzira na sklopovlje, tako da i na različitom sklopovlju programi i korisnici na jednak način koriste OS.

2. (1) Objasniti razliku između pojmova "program" i "dretva".

   program - niz instrukcija (na disku, u memoriji) koji kaže kako nešto napraviti

   dretva - niz instrukcija u izvođenju, vezanih (poredanih) vremenom izvođenja - kako se program izvodi u konkretnom slučaju

3. (2) Ulazno-izlazne naprave se mogu koristiti mehanizmima radnog čekanja, prekidima te napravama s izravnim pristupom spremniku (DMA). Navesti svojstva (dobra i loša) svakog od navedenih mehanizama.

   radno čekanje – dobra svojstva: jednostavno sklopovlje (jednostavan procesor, pristupni sklop)

   radno čekanje – loša svojstva: neproduktivan rad

   prekidi – dobra svojstva: produktivan rad i brza reakcija

   prekidi – loša svojstva: treba sklop, kućanski poslovi (neproduktivan rad)

   DMA – dobra svojstva: efikasan rad s napravama koje daju puno podataka

   DMA – loša svojstva: treba napredniji sklop

4. (1) Kada se mogu dva zadatka Z_i i Z_j definirana svojim domenama D_i i D_j te kodomenama K_i i K_j pustiti da rade paralelno?

   $(D_i \cap K_j) \cup (D_j \cap K_i) \cup (K_i \cap K_j) = \emptyset$

5. (1) Koji je glavni nedostatak algoritama međusobnog isključivanja pokazanih u 4. poglavlju (Dekker, Lamport, uz pomoć Ispitaj_i_postavi())?

   radno čekanje

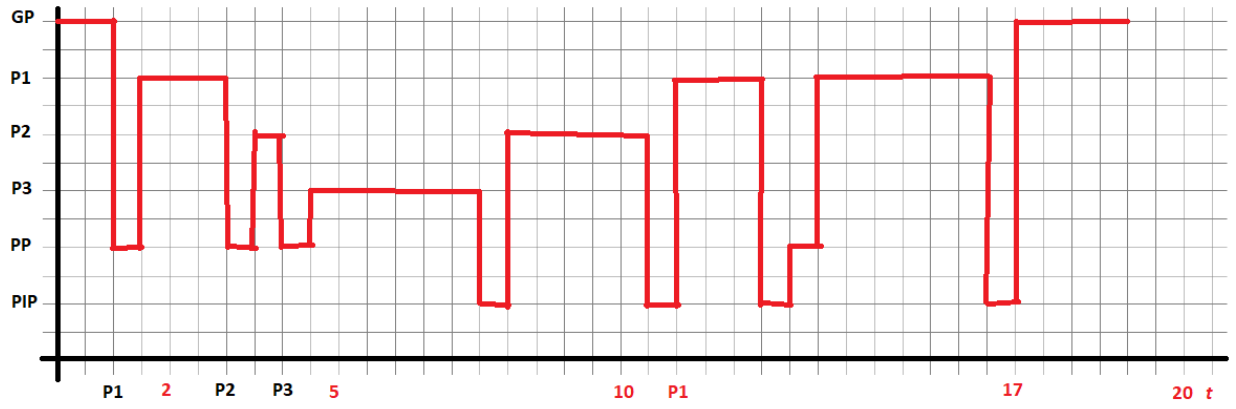
6. (1) Navesti stanja u kojima se dretva može naći (u jednostavnom modelu jezgre)?

   aktivno, pripravno, blokirano [, pasivno]

7. (1) Navesti minimalne uvjete za nastanak potpunog zastoja.

   1. bar dvije dretve i bar dva sredstva koje te dretve obje koriste
   2. u svakom trenutku sredstvo smije koristiti samo jedna dretva (međusobno isključivo)
   3. dretvi se sredstvo ne može oduzeti, ona ga sama otpušta kada joj više ne treba
   4. dretva drži dodijeljeno sredstvo dok traži dodatno sredstvo

8. (3) U nekom sustavu sa sklopom za prihvat prekida javljaju se prekidi: P_1 u 1. i 11., P_2 u 3. te P_3 u 4. ms. Prioritet prekida je određen brojem (P_3 ima najveći prioritet). Procedura za prihvat 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 prihvat prekida (PP) i povratak iz prekida (PiP).



9. (2) U nekom sustavu procesor radi na frekvenciji 10 MHz i pri svakoj instrukciji treba jedan sabirnički ciklus (procesorski/sabirnički ciklus traje 1/(10 MHz) = 100 ns). Neka vanjska jedinica generira podatak svakih 100 μs. Nakon prijenosa 300 podataka treba ih obraditi, za što je potrebno dodatnih 400 sabirničkih ciklusa. Koji postotak procesorskog vremena (tj. sabirničkih ciklusa) se troši na tu napravu ako se za prijenos podataka koristi:
   a) (1) mehanizam prekida, uz 200 sabirničkih ciklusa za obradu pojedinog prekida,
   b) (1) sklop s izravnim pristupom spremniku?

   jedan ciklus/period = 300 podataka => 300*100μs = 30 ms

   a)

   200 sab.cik. * 300 + 400 = 60400 sab. ciklusa = 60400*100 ns = 6,04 ms

   6,04 / 30 = 0,2013 = 20,13 %

   b)

   300 + 400 = 700 sab. ciklusa = 700*100 ns = 70 μs = 0,07 ms

   0,07 / 30 = 0,00233 = 0,233 %

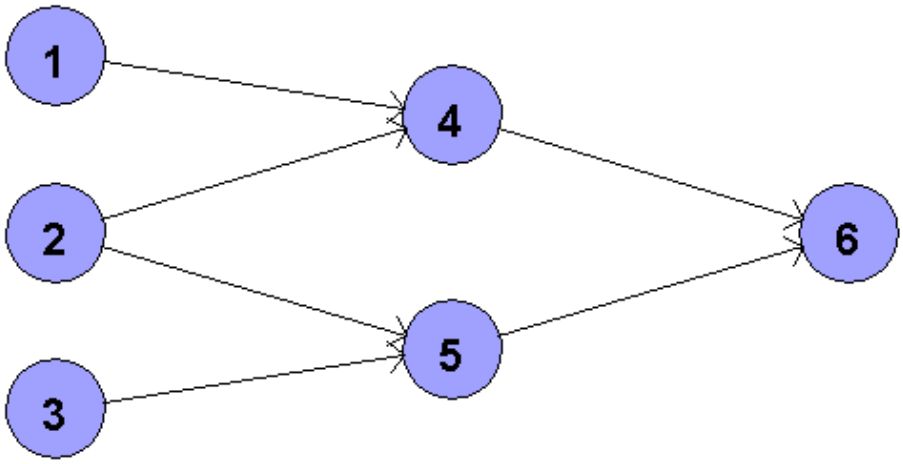
   U prethodnom rješenju nije uračunata obrada prekida koji DMA sklop generira nakon prijenosa 300 podataka jer nije zadano to trajanje. Ako bi pretpostavili 200 sab. cikl. kao i u a) slučaju, onda bi rješenje bilo:

   300 + 200 + 400 = 900 sab. ciklusa = 900*100 ns = 90 μs = 0,09 ms

   0,09 / 30 = 0,003 = 0,3 %

10. (2) 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 zadane tablicom. Odrediti maksimalno paralelni sustav zadataka uzimajući u obzir njihove domene i kodomene te međusobni odnos u lancu.

	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
M_1	D	D	D			
M_2	K			K		D
M_3		K		D	D	K
M_4			K		K	



11. (4) Stanje nekog sustava u početnom stanju je prikazano u donjoj tablici. 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. `Postavi_OSEM(1)` 2. `Otkucaj_sata()` 3. `Započni_UI(1)` 4. `Odgodi(7)`. Prikazati stanje sustava nakon svakog od tih poziva.

	poč.st.	nakon 1.	nakon 2.	nakon 3.	nakon 4.
Aktivna_D	3	3	4	3	2
Pripravne_D	1	2 1	3 2 1	2 1	1
OSEM[1]	2	-	-	-	-
Odgođene_D	4^1 5^6	4^1 5^6	5^6	5^6	5^6 3^1
UI[1]	-	-	-	4	4
Idući jezgr. funkcija	Postavi OSEM(1)	Otkucaj sata()	Započni UI(1)	Odgodi(7)	

12. (5) Tri dretve tipa A dohvaćaju nove poslove sa `p=dohvati()` (nezavisno) te ih potom predaju na obradu dretvama tipa B preko zajedničke varijable `novi_posao`. Svaka dretva tipa B obrađuje svoj dio posla sa `obradi_dio_posla()` (izravno koristeći posao u varijabli `novi_posao`). Kad su sve dretve B gotove (njih šest) onda se može pozvati (od strane samo jedne dretve B) `pohrani()` i tek tada primiti novi posao (neka dretva A može staviti svoj posao u `novi_posao`). Napisati pseudokod za dretve A i B koristeći binarne i/ili opće semafore za sinkronizaciju. Navesti početne vrijedosti varijabli i semafora.

```
dretva A
{
  ponavljaj {
    p = dohvati()

    ČekajBSEM(stavi)

    novi_posao = p

    za i = 1 do 6
      PostaviOSEM(radi)
  }
  do zauvijek
}
```

```
dretva B
{
  ponavljaj {

    ČekajOSEM(radi)

    obradi_dio_posla()

    ČekajBSEM(KO)
    br++
    ako je br == 6 {
      pohrani()
      br = 0
      PostaviBSEM(stavi)
    }
    PostaviBSEM(KO)

  }
  do zauvijek
}
```

Početne vrijednosti:

```
BSEM[stavi].v = 1
BSEM[KO].v = 1
OSEM[radi].v = 0
br = 0
```

13. (3) Jedna dretva proizvođača i jedna dretva potrošača komuniciraju preko međuspremnika M prema kodu.

```
proizvođač
{
  ponavljaj {
    P = stvori poruku()
    Čekaj_OSEM(2)
    M[ULAZ] = P
    ULAZ = (ULAZ + 1) MOD N
    Postavi_OSEM(1)
  }
  do zauvijek
}
```

```
potrošač
{
  ponavljaj {
    Čekaj_OSEM(1)
    R = M[IZLAZ]
    IZLAZ = (IZLAZ + 1) MOD N
    Postavi_OSEM(2)
    obradi poruku(R)
  }
  do zauvijek
}
```

Ako je u nekom trenutku u međuspremniku X nepročitanih poruka (X < N), najstarija nepročitana poruka na mjestu 3, koje vrijednosti imaju opći semafori 1 i 2 te varijable ULAZ i IZLAZ u tom trenutku?

`OSEM[1].v = X` `OSEM[2].v = N-X` `ULAZ = (3+X) MOD N` `IZLAZ = 3`