

II kolokvijum

IME i PREZIME
BROJ INDEKSA

PRAVILA.

I deo kolokvijuma se radi 15 minuta. Nije dozvoljeno korišćenje računara, niti materijala sa vežbi. POENA: 7
II deo kolokvijuma se radi isključivo na računaru. Vreme izrade - 60 minuta. Dozvoljeno je koristiti odštampane man strane za IPC f-je, signal, fork. POENA: 12

I deo - 7 poena

1. Kada proces prelazi iz aktivnog u ready stanje? Navesti bar 3 slučaja.

1 poen

2. Opisati semantiku operacije izlaska iz kritičnog regiona, pri čemu je pristup kritičnom regionu kontrolisan jednim objektom tipa semafor.

1 poen

3. Šta se dešava pri izvršavanju ovog koda

2 poena

```
... /* include ...*/  
main(){ if (fork() == 0) { printf("Consumer");}  
        else {  
            fork();  
            printf("Producer");  
        }  
}
```

Koliko procesa će biti generisano? Obležite nekim imenom/brojem svaki proces, prokomentarišite u kakvom su odnosu (koji je proces kojem roditelj ili dete) i kakav izlaz svaki od njih daje.

4. Pseudo kodom opisati kako bi se upotrebom semafora mogao iskontrolisati pristup zajedničkom resursu. Postoje dva tipa procesa Producer i Consumer. Resurs sadrži 2 slota. Producer funkcijom `upisi(x)`, puni samo jedan (slobodan) slot. Oba slota moraju da budu napunjena da bi čitanje moglo da započne, pri čemu Consumer f-jom `citaj()` preuzima vrednosti iz oba slota. Nije dozvoljeno novo čitanje, dok se nove vrednosti ne upisu, niti dva punjenja oba slota za redom (Dakle, redosled mora da bude Producer, Producer, Consumer) Imenovati semafore. Objasniti im namenu. Navesti početne vrednosti.

3 poena

PRODUCER

```
x: slot;  
while true do  
begin
```

```
    upisi(x);
```

```
end;
```

CONSUMER

```
x: array[1..2] of slot;  
while true do  
begin
```

```
    citaj(x);
```

```
end;
```

II deo - 9 poena

zadatak za 9 poena

Napisati program koji simulira problem iz 4-tog zadatka. Fork-ovati.

zadatak za 5 poena

Napisati jedan 1 serverski i 1 klijentski program, koji komuniciraju upotrebom redova za poruke.

Klijent:

3 koraka:

1. šalje ID objekta koji koristi
2. čeka potvrdu (poruku) od servera da može da se odjavi
3. uklanja svoj red

Server:

Među svojim podacima sadrži niz (neka se zove PRIJAVLJENI) u kojem se nalaze msqid klijenata.

Beskonačna petlja, jedna iteracija:

1. čeka poruke od 3 klijenta iz kojih čita msqid i ime i upisuje u PRIJAVLJENI
2. nakon toga svima šalje poruku da mogu da se odjave
3. ako je došlo do prekida pre slanja poruka klijentima o odjavljivanju, sighandler treba to slanje da obavi pa nakon toga da ukloni serverov objekat