

POPRAVNI II kolokvijuma

IME i PREZIME
BROJ INDEKSA

PRAVILA.

I deo kolokvijuma se radi 30 minuta. Nije dozvoljeno korišćenje računara, niti materijala sa vežbi. POENA: 9

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: 10

I deo - 9 poena

1. Da li se može desiti da na pola obrade jednog interapta, procesor započne obradu novog? Čime je to regulisano?

1.5 poena

2. Šta sadrže procesne liste? Koliko ih ima i koje su? Ko ih održava?

1.5 poena

3. Šta se dešava pri izvršavanju ovog koda

2 poena

```
... /* include ...*/  
main(){  
    int i=1;  
    if (fork() > 0) printf("%d\n",++i);  
    fork();  
    printf("%d\n",++i);  
}
```

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

4. Šta se dešava ako pri izvršavanju ovog koda (odgovarajućem) procesu bude poslat signal ^C

1 poen

```
#include <signal.h>
void my_handler (int sig);
int main(void) {
    signal (SIGINT, my_handler);
    sleep(3);
    signal (SIGINT, SIG_DFL);
    sleep(3);
    return 0;
}
void my_handler (int sig) {
    printf ("Handler %d\n", sig);
    exit(1);
}
```

5. Pseudo kodom opisati kako bi se upotrebom semafora mogao iskontrolisati pristup zajedničkom resursu. Postoje tri procesa Producer, Consumer1 i Consumer2. 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 Consumer1 i Consumer 2 f-jom citaj() preuzimaju vrednosti iz oba slota odjednom. Nije dozvoljeno novo čitanje, dok se nove vrednosti ne upisu, niti dva uzastopna upisa dok oba ;ita;a ne pro;itaju vrednosti (Dakle, redosled mora da bude Producer, Producer, oba Consumer-a). Imenovati semafore. Objasniti im namenu. Navesti početne vrednosti.

3 poena

| PRODUCER | CONSUMER1 | CONSUMER2 |
|---------------|-------------------------|-------------------------|
| x: slot; | x: array[1..2] of slot; | x: array[1..2] of slot; |
| while true do | while true do | while true do |
| begin | begin | begin |
| upisi(x); | citaj(x); | citaj(x); |
| end; | end; | end; |

zadatak za 10 poena

Napisati program koji implementira sledeci kod:

| | |
|--|--|
| Sluzbenik int spremio=0; while true { P(mutex); spremam(++spremio); V(mutex); V(posiljka); } | Kurir int poslao=0; while true { P _s (posiljka); P(mutex); saljem(++poslao); V(mutex); end; |
|--|--|

F-ja **spremam()**

ispisuje poruku Spremam!
sleep par sekundi
ispisuje poruku Spremio /-tu posiljku!

F-ja **saljem()**

ispisuje poruku Preuzio /-ti paket!
sleep par sekundi
ispisuje poruku Ukupno poslao /*5 posiljki!

zadatak za 10 poena

Napisati jedan 1 serverski i 2 klijentska programa (sender i receiver), koji komuniciraju upotrebom redova za poruke.

Klijent - receiver:

1. serveru u poruci tipa 1 šalje ID objekta koji koristi i komentar da je receiver
2. čeka poruku od servera, a kada je dobije, ispiše njen sadržaj i
3. uklanja svoj red

Klijent - sender:

1. serveru u poruci tipa 1 šalje ID objekta koji koristi i komentar da je sender
2. sa standardnog ulaza čita jednu poruku i šalje je serveru u poruci tipa 2, a zatim
3. uklanja svoj red i završava sa radom

Server:

Među svojim podacima sadrži nizove: senderi i receiveri u kojem se nalaze msqid klijenata.

Beskonačna petlja, jedna iteracija:

1. skida 6 poruka tipa 1 i pakuje podatke u odgovarajuće nizove, ako nema isti broj sender-a i receiver-a ispisuje komentar, uklanja svoj IPC objekat i završava sa radom
2. ukoliko je bio isti broj sender-a i receiver-a onda server tri puta:
skida po jednu poruku od sender-a i šalje jednom receiver-u, prvu poruku prvom receiver-u u nizu, drugu drugom ...)
3. uklanja svoj objekat i završava sa radom
4. regulisati: ako je došlo do prekida pre slanja poruka klijetima o odjavljivanju, sighandler treba to slanje da obavi pa nakon toga da ukloni serverov objekat

zadatak za 5 poena

Napisati 2 programa (sender i receiver), koji komuniciraju upotrebom redova za poruke.

Regulisati: ako je došlo do prekida pre kraja rada receiverslanja poruka klijetima o odjavljivanju, sighandler treba to slanje da obavi pa nakon toga da ukloni serverov objekat

Klijent - receiver:

1. prima poruku od sender-a ispisuje je na ekranu, a zatim
2. uklanja svoj red
3. regulisati: ako je došlo do prekida pre stizanja poruke, ukloniti red

Klijent - sender:

1. sa standardnog ulaza čita jednu poruku i šalje je receiver-u, a zatim
2. uklanja svoj red i završava sa radom