

II kolokvijum

IME i PREZIME
BROJ INDEKSA

PRAVILA.

I deo kolokvijuma se radi 15 minuta. Nije dozvoljeno korišćenje računara, niti materijala sa vežbi.

II deo kolokvijuma se radi isključivo na računaru. Vreme izrade - 60 minuta. Dozvoljeno je koristiti odštampane man strane za IPC f-je, signal, fork.

I deo

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

1 poen

2. Ko obavlja obaveštavanje blokiranog procesa da je uzrok blokiranja uklonjen, tj. da je ulazak u kritični region moguć (signalizacija)? Opisati postupak signalizacije na primeru semafora.

2 poena

3. Šta se dešava pri izvršavanju ovog koda

2 poena

```
... /* include ...*/  
main(){ if (fork() > 0) {printf("Prvi\n");}  
        else {  
            if (fork() == 0) printf("Drugi\n");  
            printf("Treci\n");  
        }  
    }
```

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);
    printf ("Catching SIGINT\n");
    sleep(3);
    signal (SIGINT, SIG_IGN);
    printf ("Hello!\n");
    return 0;
}
void my_handler (int sig) {
    printf ("I got SIGINT, number %d\n", sig);
}
```

5. Pseudo kodom opisati kako bi se upotrebom semafora mogao kontrolisati pristup zajedničkom resursu - Igraonica. Postoje dva tipa procesa Vaspitacica i Dete.

3 poena

Jedna vaspitačica može da čuva do 5oro dece. Deca ne mogu da se igraju ako vaspitačica nije tu ili ih nema dovoljno.

Ne proveravati slučaj koji bi mogao u realnom primeru da se desi da vaspitacica ne može da izađe ako ima dece koje neće imati ko da čuva.

Imenovati semafore. Objasniti im namenu. Navesti početne vrednosti.

Vaspitacica

Dete

Cuvam();

IgramSe();

zadatak za 10 poena

Napisati program koji simulira problem iz 5-tog zadatka. Napisati posebne programe za Vaspitacicu i Dete.

cuvam()

ispisuje poruku Ulazi jedna vaspitacica.

sleep 20 sekundi

ispisuje poruku Izazi jedna vaspitacica.

igramse()

ispisuje poruku Dete uslo!

ispisuje poruku sleep par sekundi

ispisuje poruku Dete izaslo!

zadatak za 8 poena Simulacija Chat-a

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

Chat-klijent:

1. šalje ID objekta koji koristi serveru
2. čeka poruku od servera o svom rednom broju
3. u beskonačnoj petlji sledeće
 - ako je 1. prijavljen on šalje svoj redni broj i tekst svoje poruke (učitati sa standardnog ulaza) serveru (server tu poruku prosleđuje 2. prijavljenom i čeka od njega poruku sa odgovorom na tekst 1. klijenta), prima odgovor od servera
 - ako je 2. prijavljen sačeka poruku od servera (poruka 1. klijenta), a zatim šalje svoj redni broj i tekst (čiji će tekst server proslediti 1. klijentu)Ako se u odgovoru servera nalazi tekst "odjava" klijent se odjavljuje (ovakvu poruku dobija u slučaju da se drugi klijent odjavio).
4. ako klijent primi SIGINT signal pre uklanjanja svog reda šalje poruku serveru sa tekстом "odjava" a zatim završava sa radom.

Server:

Prima prijave 2 klijenta i šalje im odgovor u kojem se nalazi odgovarajući redni broj.

Beskonačna petlja, jedna iteracija:

1. čeka poruku od klijenta iz koje čita redni broj i analizira tekst
 - a. ako je tekst "odjava" šalje istu poruku drugom klijentu, uklanja svoj red i završava sa radom
 - b. ako tekst nije "odjava" tekst poruke šalje drugom klijentu
2. ako primi SIGINT signal šalje poruke klijentima sa tekстом "odjava", uklanja svoj red i završava sa radom

zadatak za 5 poena

Chat-klijent:

1. šalje ID objekta koji koristi serveru
U beskonačnoj petlji
2. čeka poruke koje sadrže neki tekst koji treba da ispiše na ekranu
3. ukoliko je tekst poruke "odjava" uklanja svoj red

Server:

Čeka poruke od dva klijente koji treba da pošalju ID svog reda za poruku.

Beskonačna petlja, jedna iteracija:

1. sa standardnog ulaza čita tekst i prosleđuje ga klijentima
2. ukoliko je tekst poruke "odjava" uklanja svoj red i završava sa radom
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