Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Operativni sistemi 2
Nastavnik: prof. dr Dragan Milićev
Odsek: Softversko inženjerstvo, Računarska tehnika i informatika
Kolokvijum: Integralni, februar 2021.
Datum: 11. 2. 2021.

Integralni kolokvijum iz Operativnih sistema 2

Kandidat: _____

Broj indeksa: _____ *E-mail:*_____

Kolokvijum traje 2 sata. Dozvoljeno je korišćenje literature.

Zadatak 1	_____/10	*Zadatak 3*	_____/10
Zadatak 2	_____/10	*Zadatak 4*	_____/10

Ukupno: _____/40 = _____%

Napomena: Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je uokviri (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Ocenjivanje unutar potpitanja je po sistemu "sve ili ništa", odnosno nema parcijalnih poena. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**.

1. (10 poena)

Napisati na programskom jeziku Java kod servera koji za korisnike izvršava izračunavanja na grafičkim karticama. Server ima N kartica. Svaka kartica je povezana sa dve susedne odgovarajućim kablovima i tako povezane čine prsten. Korisnik može da traži M kartica za svoje izvršavanje (gde je $M \leq N$). Server treba da pokrene izvršavanje na M susednih kartica (na lancu od M kartica). Kartice su nedeljiv resurs: dok se ne završi prethodno izračunavanje, sledeće ne sme da se pokrene. Izračunavanja za nekoliko korisnika može da se radi paralelno, ali samo na disjunktnim skupovima kartica. Da bi sprečio izgladnjivanje, server treba da pokreće izračunavanja samo onim redom kojim su zahtevi pristigli.

Server pokreće izračunavanja na karticama pomoću metode:

```
String runExecutionOnCards(String commands, int[] cards);
```

Ova metoda prima komande kao jedan string koji šalje korisnik. Rezultat izvršavanja je string koji se šalje korisniku kao odgovor. U tim stringovima mogu da se nađu bilo koji znaci (čak i prelazak u novi red), osim znakova `#` i `@`. Drugi parametar metode je niz indeksa kartica na kojima treba da se pokrene izvršavanje. Indeksi kartica u sistemu su od 0 do N–1, gde su kartice sa susednim indeksima i fizički susedi (kartica 0 je povezana sa karticama broj 1 i N–1, kartica 1 je povezana sa karticama 0 i 2 itd).

Korisnik šalje serveru samo komande za kartice i to na koliko susednih kartica te komande treba da se izvršavaju. Format poruka je potrebno osmisliti u okviru rešenja. Server osluškuje zahteve korisnika na portu 5555.

Rešenje:

2. (10 poena)

U nekom sistemu resursi se identifikuju neoznačenim celim brojevima. Funkcija:

```
int alloc (unsigned resource);
```

atomično zauzima resurs sa datim identifikatorom. Njen poziv je sinhroni, neblokirajući: ukoliko je resurs uspešno zauzet, ona odmah vraća 1, a ako nije, odmah vraća 0, bez suspenzije pozivaoca; u slučaju bilo kakve greške, vraća negativnu vrednost. Funkcija `release(unsigned)` oslobađa resurs sa zadatim identifikatorom.

a)(5) Realizovati funkciju `allocate` koja ima isti potpis kao i `alloc`, ali koja kontrolu pozivaocu vraća tek kada je resurs uspešno zauzet ili ako se prilikom zauzimanja dogodila greška. Ako resurs ne može da bude zauzet odmah, pozivajući tok kontrole se uspavljuje neko slučajno vreme, a onda se ponovo pokušava zauzimanje resursa. Na raspolaganju su sledeće funkcije:

`void sleep (unsigned long sleep_time)`: uspavljuje pozivaoca na zadato vreme u μs;
`unsigned long rnd (unsigned long n)`: vraća slučajan ceo broj u ospegu 0..*n*-1.

Koji problem ima ovakav pristup alokaciji resursa?

b)(5) Korišćenjem funkcije realizovane pod a), realizovati funkciju:

```
int allocate (unsigned res[], size_t n);
```

koja u jednom pozivu zauzima čitav niz resursa dimenzije *n* (sve ili ništa), ali tako da se spreči mrtva/živa blokada.

Rešenje:

3. (10 poena)

Neki sistem upravlja stranicama rezervisanim za procese na sledeći način. Sve ove stranice zauzimaju kontinualan niz susednih okvira fizičke memorije. Za evidenciju okvira služi niz `frames` struktura tipa `FrameDesc` datog dole; element *i* ovog niza odgovara okviru broj *i* u navedenoj oblasti memorije. Strukture `FrameDesc` koje odgovaraju zauzetim okvirima (onim u kojima se nalaze učitane stranice) povezane su u jednostruko ulančanu kružnu listu uređenju po redosledu učitavanja stranica u odgovarajući okvir, a sistem primenjuje algoritam časovnika/davanja nove šanse (*clock*, *second chance*) zamene stranica; ova kružna lista je uvek neprazna. U strukturi `FrameDesc` polje `next` sadrži indeks elementa u nizu koji je sledeći u ovoj kružnoj listi, a polje `pgDesc` je pokazivač na deskriptor stranice u PMT koja se nalazi u odgovarajućem okviru (*null* ako je okvir slobodan). U deskriptoru stranice najniži bit je bit referenciranosti. Globalna promenljiva `clockHand` je „kazaljka“ ovog algoritma i sadrži indeks elementa u nizu na koji kazaljka ukazuje.

Implementirati funkciju `clock` koja treba da pomeri kazaljku (ili je ostavi na istom mestu) na mesto koje ukazuje na okvir sa stranicom koja je žrtva za izbacivanje prilikom zamene stranice.

```
struct FrameDesc {
  unsigned long next, *pgDesc;
};
extern struct FrameDesc frames[];
extern unsigned long clockHand;
void clock ();
```

Rešenje:

4. (10 poena)

Napisati *Bash Shell* skriptu koja prima kao jedini parametar ime fajla. Sadržaj datog fajla se koristi za generisanje C/C++ koda. Generisani kod skripta treba da odštampa na standardnom izlazu. U svakoj liniji se nalaze informacije o jednom podatku. Linija ne sadrži blanko znakove, a sva slova u liniji su mala. Podatak ima dva imena odvojena tačkom, veličinu izraženu u bitima i adresu datu u heksadecimalnom obliku. Za svaki podatak treba generisati C/C++ kod koji pravi jednu promenljivu veličine u bitima kolika je veličina podatka, promenljivu koristi za čitanje vrednosti podatka na osnovu imena i adrese i tu pročitanu vrednost ispiše zajedno sa imenom podatka. Izgenerisani poziv funkcije `printf` treba da sadrži informaciju o veličini podatka u bajtovima. Tačan format ulaznog fajla i izlaza zaključiti na osnovu datog primera. U slučaju neispravnih parametara potrebno je ispisati grešku i prekinuti izvršavanje skripte. Sadržaj u datom fajlu smatrati ispravnim i smatrati da će izgenerisani kod na osnovu ulaznog fajla biti ispravan.

Primer ulaznog fajla:

```
register(ra.ctrl,8,0x00)
register(ra.status1,16,0x01)
register(rb.data,64,0x03)
register(r1.inter,8,0x0b)
```

Primer izlaza za dati ulaz:

```
{uint8_t __reg;
read_reg("ra", "ctrl", 0x00, &__reg);
printf("ra.ctrl=0x"PRIx8"\n", __reg);}
{uint16_t __reg;
read_reg("ra", "status1", 0x01, &__reg);
printf("ra.status1=0x"PRIx16"\n", __reg);}
{uint64_t __reg;
read_reg("rb", "data", 0x03, &__reg);
printf("rb.data=0x"PRIx64"\n", __reg);}
{uint8_t __reg;
read_reg("r1", "inter", 0x0b, &__reg);
printf("r1.inter=0x"PRIx8"\n", __reg);}
```

R
e
š
e
n
j