
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: Treći, januar 2022.
Datum: 20. 1. 2022.

Treći kolokvijum iz Operativnih sistema 2

Kandidat: _____

Broj indeksa: _____ *E-mail:* _____

Kolokvijum traje 1,5 sat. Dozvoljeno je korišćenje literature.

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

Ukupno: _____/30 = _____%

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) Upravljanje diskovima

Klasa `DiskScheduler`, čiji je interfejs dat dole, implementira algoritam *C-Look* raspoređivanja zahteva za operacije na disku. Zahtev za operaciju na disku predstavljen je strukturom `Req`, u kojoj postoje polja `next` za uvezivanje u listu i polje `block` koje sadrži broj bloka na disku na koji se zahtev odnosi. Blokovi su numerisani u skladu sa svojim položajem po cilindrima kako je uobičajeno (formalno, ako je broj bloka $b1$ manji od broja bloka $b2$, onda je broj cilindra $c1$ na kom je blok $b1$ manji ili jednak od broja cilindra $c2$ na kom je blok $b2$). Implementirati klasu `DiskScheduler` i sve njene operacije. Operacije ove klase imaju sledeće značenje:

- Operacijom `put` se nov zahtev ubacuje u red za raspoređivanje koji realizuje ova klasa.
- Operaciju `get` poziva ostatak kernela kada želi da dobije zahtev koji je na redu za obradu. Taj zahtev se tada ne izbacuje iz reda za raspoređivanje.
- Operaciju `remove` poziva ostatak kernela kada završi obradu datog zahteva koji je bio na redu za obradu, čime se taj zahtev izbacuje iz reda.

```
class DiskScheduler {
public:
    DiskScheduler ();

    void put (Req* r);
    Req* get ();
    void remove ();
};
```

Rešenje:

2. (10 poena) Operativni sistem Linux

Napisati *Bash* skriptu koja transformiše tekst u Java kod. Tekst se sastoji od proizvoljnog broja linija, gde svaka linija počinje rečima „set property“, nakon čega se u vitičastim zagradama nalazi ime opcije i na kraju se u vitičastim zagradama nalazi vrednost opcije. Primer jednog teksta je:

```
set property {property1} {500}  
set property {property2} {700}  
set property {property3} {value3}
```

Pretpostaviti da je format ulaznog teksta ispravan. Moguće je da se u tekstu pojavi isto ime opcije više od jednom. To treba tretirati kao grešku i u tom slučaju treba ispisati odgovarajuću poruku i prekinuti izvršavanje skripte.

Svaku liniju iz ulaznog teksta treba pretvoriti u poziv funkcije `setProperty` koja prima dva parametra tipa `string`. Primer izlaznog koda za dati ulazni tekst je:

```
setProperty("property1", "500");  
setProperty("property2", "700");  
setProperty("property3", "value3");
```

Ulazni tekst se nalazi u fajlu čija je putanja data kao argument skripte. U slučaju nedovoljnog broja parametra ili nepostojanja prava pristupa za čitanje tog fajla skriptu treba prekinuti i ispisati odgovarajuću poruku.

Rešenje:

3. (10 poena) Operativni sistem Linux

Napisati program koji pokreće dva procesa, *P1* i *P2*. Proces *P1* ciklično proizvodi vrednost x pomoću funkcije `getX`, a zatim koristi vrednost y pomoću funkcije `useY`. Proces *P2* ciklično čita vrednost x i proizvodi vrednost y pomoću funkcije `useXGetY`. Obezbediti da procesi čitaju sveže i ispravne vrednosti x i y . Deklaracije datih funkcija su:

```
int getX();  
void useY(int);  
int useXGetY(int);
```

Program napisati na programskom jeziku C koristeći imenovane cevi operativnog sistema Linux.

Rešenje: