
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Operativni sistemi 2 (IR3OS2)
Nastavnik: prof. dr Dragan Milićev
Odsek: Računarska tehnika i informatika
Kolokvijum: Prvi, decembar 2017.
Datum: 2. 12. 2017.

Prvi 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 u okviru (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) Raspoređivanje procesa

U nekom sistemu klasa `Scheduler`, čija je delimična definicija data dole, realizuje raspoređivač spremnih procesa po prioritetu (engl. *priority scheduling*), tako da i operacija dodavanja novog spremnog procesa `put()` i operacija uzimanja spremnog procesa koji je na redu za izvršavanje `get()` imaju ograničeno vreme izvršavanja koje ne zavisi od broja spremnih procesa (kompleksnost $O(1)$ u odnosu na broj spremnih procesa). Postoji N redova spremnih procesa, pri čemu je N konfiguraciona konstanta. Red sa nižim indeksom ima viši prioritet (tj. red sa indeksom 0 je najvišeg prioriteta). Za izvršavanje se bira proces iz reda sa najvišim prioritetom, a unutar istog reda po *FCFS* redosledu.

Da bi sprečio izglednjivanje procesa, raspoređivač primenjuje tehniku starenja (engl. *aging*), tako što povremeno (npr. periodično) *svim* procesima koji su se tada zatekli u redu spremnih povećava prioritet za po jedan. (Pošto se ovo radi relativno retko, nije od prevelikog značaja što je neki proces nedavno, a neki ranije dospao u taj red.) Ovo izvršava operacija `age` klase `Scheduler`, koja takođe treba da ima kompleksnost $O(1)$ u odnosu na broj spremnih procesa.

U strukturi `PCB` postoji polje `next` kao pokazivač tipa `PCB*` koji služi za ulančavanje struktura `PCB` u jednostruke liste i celobrojno polje `priority` koje sadrži prioritet procesa. Ukoliko su svi redovi prazni, operacija `Scheduler::get` treba da vrati 0. Realizovati u potpunosti klasu `Scheduler`.

```
class Scheduler {
public:
    Scheduler ();
    PCB* get ();
    void put (PCB*);
    void age ();
private:
    static const int N;
    ...
};
```

Rešenje:

2. (10 poena) Meduprocesna komunikacija pomoću deljene promenljive

Korišćenjem klasičnih uslovnih promenljivih, napisati kod za monitor `Toggle` koji ima dve operacije, `flip` i `flop`, a koje pozivaoci mogu izvršavati strogo naizmenično.

Rešenje:

3. (10 poena) Međuprocesna komunikacija razmenom poruka

Na programskom jeziku Java implementirati server za čuvanje tekstualnih fajlova. Klijenti serveru šalju tekstualne fajlove i putanju gde ti fajlovi treba da se nalaze na serveru. Server treba da primi fajl, sačuva ga na zadatoj lokaciji i pošalje njegov sadržaj klijentu da bi klijent proverio da li je fajl uspešno sačuvan. Klijent javlja serveru da li se sadržaj poklapa. Ukoliko se ne poklapa, server treba da ukloni fajl koji je loše sačuvan. Server može istovremeno više klijenata da opslužuje. Ukoliko više klijenata želi da sačuva fajl na istoj putanji, obezbediti da operacija čuvanja i provere uspešnosti bude atomična. Protokol komunikacije sa klijentima osmisliti prema potrebi. Dozvoljena je upotreba koda sa vežbi (kod nije potrebno prepisivati, već samo precizno navesti koja klasa se koristi).

Rešenje: