

---

---

Elektrotehnički fakultet u Beogradu  
Katedra za računarsku tehniku i informatiku

*Predmet:* Operativni sistemi 1 (SI2OS1, IR2OS1)

*Nastavnik:* prof. dr Dragan Milićev

*Odsek:* Softversko inženjerstvo, Računarska tehnika i informatika

*Kolokvijum:* Treći, septembar 2012.

*Datum:* 31.8.2012.

*Treći kolokvijum iz Operativnih sistema 1*

*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 = \_\_\_\_\_% = \_\_\_\_\_/10

**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) Ulaz/izlaz

Realizovati u potpunosti klasu `DoubleBuffer` čiji je interfejs dat. Ova klasa implementira dvostruki bafer. Proizvođač stavlja u bafer znak po znak pozivom operacije `put()`; znak se stavlja u trenutni „izlazni“ bafer od dva interna bafera veličine `size` znakova. Potrošač uzima blokove veličine `chunkSize` znakova iz trenutnog „ulaznog“ bafera pozivom operacije `get()`; znakovi se prepisuju u bafer pozivaoca na koji ukazuje argument `buffer`. Kada obojica završe sa svojim baferom, baferi zamjenjuju uloge. Proizvođač i potrošač su uporedne niti (ne treba ih realizovati), dok je sva potrebna sinhronizacija unutar klase `DoubleBuffer`. Pretpostaviti da je zadata veličina bafera u znakovima (argument `size` konstruktora) celobrojan umnožak zadate veličine bloka (argument `chunkSize`). Za sinhronizaciju koristiti semafore.

```
class DoubleBuffer {
public:
    DoubleBuffer (int size, int chunkSize);
    void put (char);
    void get (char* buffer);
private:
    ...
};
```

Rešenje:

## 2. (10 poena) Interfejs fajl sistema

Neki fajl sistem podržava strukture direktorijuma tipa acikličnog usmerenog grafa (DAG). Iz komandne linije mogu se izvršiti sledeće systemske komande:

<code>dir</code>	prikazuje sadržaj tekućeg direktorijuma
<code>cd &lt;dir&gt;</code>	menja tekući direktorijum
<code>alias &lt;file&gt; &lt;newname&gt;</code>	za postojeći fajl sa zadatom stazom <file> kreira novi ulaz u tekućem direktorijum pod datim novim imenom <newname>
<code>rm &lt;file&gt;</code>	briše ulaz sa zadatim imenom iz tekućeg direktorijuma; ukoliko je to poslednja referenca na fajl, briše se i sam fajl.

Zabeležena je sledeća sesija jednog korisnika:

```
> cd /home/docs
> dir
bar    <file>
foo    <file>
txt    <dir>
> cd /home/pics
> dir
jane   <file>
john   <file>
chld   <dir>
> alias /home/docs/bar foo
> rm john
> cd /home/docs
> rm bar
> cd /home/pics
> dir
```

a)(5) Napisati izlaz poslednje komande.

Odgovor:

b)(5) Nakon prikazane sekvence zadate su sledeće komande:

```
> rm foo
> cd /home/docs
> dir
```

Napisati izlaz poslednje komande.

Odgovor:

### 3. (10 poena) Implementacija fajl sistema

Neki fajl sistem primenjuje indeksirani pristup alokaciji prostora za sadržaj fajla, s tim da je indeks neograničen i organizovan kao jednostruko ulančana lista indeksnih blokova. Na prvi indeksni blok u listi ukazuje polje `index` u `FCB`. Svaki indeksni blok sadrži `NumOfEntries` ulaza tipa `BlkNo` koji ukazuju na blokove sa sadržajem fajla i još jedan ulaz (iza ovih) istog tipa koji ukazuje na sledeći indeksni blok u listi (vrednost 0 označava *null* ulaz). Na raspolaganju je funkcija za pristup blokovima diska kroz keš, koja vraća pokazivač na deo memorije u kome se nalazi traženi blok diska učitani u keš (vraća 0 u slučaju greške):

```
Byte* getDiskBlock (BlkNo block);
```

Realizovati funkciju `getFileBlock()` koja za dati fajl dohvata logički blok sa datim brojem. U slučaju prekoračenja veličine sadržaja fajla ili druge greške treba vratiti 0.

```
Byte* getFileBlock (FCB* file, unsigned int block);
```

Rešenje: