

---

---

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 2013.

*Datum:* 30.8.2013.

*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 blokove veličine `chunkSize` znakova pozivom operacije `put()`; znakovi se uzimaju iz bafera pozivaoca na koji ukazuje argument `buffer`, a stavljaju u trenutni „izlazni“ bafer od dva interna bafera veličine `size` znakova. Potrošač uzima znak po znak iz trenutnog „ulaznog“ bafera pozivom operacije `get()`. Kada obojica završe sa svojim baferom, baferi zamenjuju 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* buffer);
    char get ();
private:
    ...
};
```

Rešenje:

## 2. (10 poena) Interfejs fajl sistema

Dat je izvod iz originalne dokumentacije za Unix/Linux sistemski poziv za otvaranje fajla:

```
int open(const char *pathname, int flags, mode_t mode);
```

The argument **flags** must include one of the following access modes: **O\_RDONLY**, **O\_WRONLY**, or **O\_RDWR**. These request opening the file read-only, write-only, or read/write, respectively.

In addition, zero or more file creation flags and file status flags can be bitwise-or'd in **flags**. The file creation flags are **O\_CLOEXEC**, **O\_CREAT**, **O\_DIRECTORY**, **O\_EXCL**, **O\_NOCTTY**, **O\_NOFOLLOW**, **O\_TRUNC**, and **O\_TTY\_INIT**...

**O\_CREAT** If the file does not exist it will be created. The owner (user ID) of the file is set to the effective user ID of the process. The group ownership (group ID) is set to the effective group ID of the process...

**mode** specifies the permissions to use in case a new file is created. This argument must be supplied when **O\_CREAT** is specified in **flags**.

The following symbolic constants are provided for **mode**:

<b>S_IRWXU</b>	0x0700 user (file owner) has read, write and execute permission
<b>S_IRUSR</b>	0x0400 user has read permission
<b>S_IWUSR</b>	0x0200 user has write permission
<b>S_IXUSR</b>	0x0100 user has execute permission
<b>S_IRWXG</b>	0x0070 group has read, write and execute permission
<b>S_IRGRP</b>	0x0040 group has read permission
<b>S_IWGRP</b>	0x0020 group has write permission
<b>S_IXGRP</b>	0x0010 group has execute permission
<b>S_IRWXO</b>	0x0007 others have read, write and execute permission
<b>S_IROTH</b>	0x0004 others have read permission
<b>S_IWOTH</b>	0x0002 others have write permission
<b>S_IXOTH</b>	0x0001 others have execute permission

a)(5) Napisati kod za sistemski poziv koji će kreirati fajl „test.txt“ u tekućem direktorijumu pozivajućeg procesa (označava se sa `.`) i otvara ga za čitanje i upis, tako da vlasnik tog fajla ima prava na čitanje i upis, članovi grupe prava samo na čitanje, a ostali nikakva prava.

Odgovor:

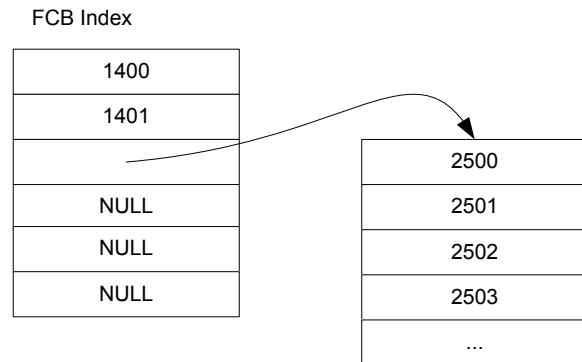
b)(5) Ukoliko takav identičan poziv izvrši neki drugi proces (nakon što je poziv iz a) uspešno izvršen) koji se izvršava u ime drugog korisnika koji nije član iste grupe kao korisnik u čije ime je izvršen prvi poziv, da li će taj drugi poziv uspeti? Obrazložiti.

Odgovor:

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

Neki fajl sistem primenjuje kombinovanu tehniku indeksirane alokacije sadržaja fajla. U FCB fajla nalaze se dva ulaza koji predstavljaju indeks nultog nivoa (direktni pokazivači na dva prva bloka sadržaja fajla) i još četiri ulaza koji ukazuju na blokove sa indeksima prvog nivoa.

Na slici je prikazan deo FCB nekog fajla. Blok je veličine 512B, a broj bloka (svaki ulaz u indeksnom bloku) zauzima 4 bajta. Bajtovi sadržaja fajla se broje počev od 0.



U kom bloku na disku se nalazi bajt sadržaja ovog fajla sa datim rednim brojem (pored konačnog odgovora, dati i celu računicu):

a)(3) 1000 (decimalno)?

Odgovor: \_\_\_\_\_

Postupak i obrazloženje:

b)(3) 2570 (decimalno)?

Odgovor: \_\_\_\_\_

Postupak i obrazloženje:

c)(4) Kolika je maksimalna veličina fajla koju dozvoljava ovaj sistem?

Odgovor: \_\_\_\_\_

Postupak i obrazloženje:

Rešenje: