
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 2014.

Datum: 29.8.2014.

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

Dat je neki sekvencijalni, blokovski orijentisani ulazni uređaj sa koga se blok znakova veličine `BlockSize` učitava na zadatu adresu funkcijom:

```
void readBlock(char* addr);
```

Od ovog uređaja napraviti apstrakciju sekvencijalnog, znakovno orijentisanog ulaznog uređaja (ulazni tok), odnosno realizovati funkciju koja učitava znak po znak sa tog uređaja:

```
char getchar();
```

Ignorirati sve greške.

Rešenje:

2. (10 poena) Interfejs fajl sistema

U nekom fajl sistemu postoje sledeći sistemski pozivi za osnovne operacije sa fajlom:

```
int open (const char *pathname, int flags, mode_t mode);
int close (int fhandle);
int read (int fhandle, byte* buffer, unsigned long size);
int write (int fhandle, byte* buffer, unsigned long size);
```

Sve ove operacije u slučaju greške vraćaju negativnu vrednost sa kodom greške. U slučaju uspeha, operacija otvaranja fajla vraća „ručku“ fajla (engl. *file handle*), a ostale vraćaju 0.

Realizovati objektno orijentisani „omotač“ ovog interfejsa, odnosno implementirati apstrakciju fajla kao klasu sa sledećim interfejsom:

```
class File {
public:
    File (const char *pathname, int flags, mode_t mode) throw Exception;
    ~File () throw Exception;

    void read (byte* buffer, unsigned long size) throw Exception;
    void write (byte* buffer, unsigned long size) throw Exception;
};
```

Prilikom kreiranja objekta ove klase treba implicitno otvoriti fajl, a prilikom uništavanja objekta treba implicitno zatvoriti fajl. U slučaju greške, sve ove operacije treba da podignu izuzetak definisanog tipa `Exception`. Instance ovog tipa (klase) mogu se inicijalizovati celobrojnim kodom greške koju vraćaju sistemski pozivi.

Rešenje:

3. (10 poena) Implementacija fajl sistema

Neki fajl sistem primenjuje kombinovanu tehniku indeksirane alokacije sadržaja fajla. U FCB fajla polje `singleIndex` predstavlja direktni indeks kao niz od `singleIndexSize` elemenata, pri čemu svaki element sadrži broj fizičkog bloka sa sadržajem fajla (ili *null*, ako je neiskorišćen). Ako veličina sadržaja fajla preraste veličinu podržanu ovim indeksom, naredni blokovi sadržaja fajla (preko ove veličine) indeksiraju se indeksom u dva nivoa. Za te potrebe, polje `dblIndex` u FCB sadrži `DblIndex0Size` elemenata, pri čemu svaki element sadrži broj bloka sa indeksom drugog nivoa (ili *null*, ako je neiskorišćen). Blokovi sa indeksom drugog nivoa sadrže najviše `DblIndex1Size` ulaza sa brojevima blokova sa sadržajem fajla. Veličina bloka je `BlockSize`.

Realizovati funkciju `getFilePBlockNo()` datu dole, koja se koristi u implementaciji modula za organizaciju fajla i koja treba da vrati broj fizičkog bloka na disku za dati fajl i dati redni broj bajta sadržaja tog fajla (počev od 0). Dati redni broj bajta je već proveren pre poziva ove funkcije, tako da sigurno adresira bajt unutar stvarne veličine sadržaja fajla (ne prekoračuje ga).

```
PBlock getFilePBlockNo (FCB* fcb, unsigned long byte);
```

Na raspolaganju je funkcija koja vraća pokazivač na sadržaj bloka na disku sa datim brojem, učitano u keš:

```
void* getDiskBlock (PBlock blkNo);
```

Rešenje: