

---

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 2015.

*Datum:* 4.9.2015.

*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

Na neki sekvencijalni, blokovski orijentisani izlazni uređaj identifikovan „ručkom“ handle se blok znakova veličine BlockSize ispisuje sa zadate adrese u memoriji funkcijom:

```
void writeBlock(IOHandle handle, char* addr);
```

Od ovog uređaja napraviti apstrakciju sekvencijalnog, znakovno orijentisanog izlaznog uređaja (izlazni tok), odnosno realizovati funkciju koja ispisuje znak po znak na taj uređaj:

```
void putchar(IOHandle handle, char);
```

Ignorirati sve greške.

Rešenje:

## 2. (10 poena) Interfejs fajl sistema

U nekom fajl sistemu postoje, između ostalih, i sledeći sistemski pozivi za osnovne operacije sa fajlom:

```
int fgetsize (int fhandle, unsigned long& size);
int fresize  (int fhandle, unsigned long newsize);
int fmoveto  (int fhandle, unsigned long offset);
int fwrite   (int fhandle, byte* buffer, unsigned long size);
```

Operacija `fgetsize` u izlazni argument `size` upisuje trenutnu veličinu sadržaja fajla, a operacija `fresize` menja veličinu sadržaja fajla na novu zadatu veličinu, koja može biti i veća i manja od trenutne; ukoliko je manja, sadržaj na kraju se odseca, a ukoliko je veća, sadržaj koji se dodaje je nedefinisan. Operacija `fmoveto` pomera kurzor na zadatu poziciju bajta (numeracija bajtova sadržaja fajla počinje od 0). Sve ove operacije u slučaju greške vraćaju negativnu vrednost sa kodom greške.

Realizovati operaciju koja proširuje sadržaj fajla datim sadržajem (dodaje ga na kraj):

```
int append (int fhandle, byte* buffer, unsigned long size);
```

Rešenje:

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

Neki fajl sistem primenjuje FAT za alokaciju sadržaja fajla. FAT je cela keširana u memoriji, na nju ukazuje pokazivač `fat`, i ima `FATSIZE` ulaza tipa `unsigned`. Prilikom ulančavanja blokova sa sadržajem fajla, `null` vrednost se označava vrednošću 0 u odgovarajućem ulazu u FAT, dok se slobodni blokovi ne ulančavaju posebno, već su njima odgovarajući ulazi u FAT označeni vrednostima `~0U` (sve jedinice binarno); blokovi broj 0 i broj `~0U` se ne koriste u fajl sistemu. U FCB polje `head` tipa `unsigned` sadrži broj prvog bloka sa sadržajem fajla (0 ako je sadržaj prazan).

Realizovati funkciju `extendFile()` datu dole, koja se koristi u implementaciji fajl sistema i koja treba da proširi sadržaj fajla za `by` blokova (da ih alokira i doda na kraj sadržaja fajla). Ova funkcija treba da vrati broj blokova kojim je stvarno proširen sadržaj fajla i koji može biti jednak `by`, ukoliko je u fajl sistemu bilo dovoljno slobodnog mesta, odnosno manji od te vrednosti (uključujući i 0), ukoliko nije bilo dovoljno slobodnih blokova.

```
unsigned extendFile (FCB* fcb, unsigned by);
```

Rešenje: