

---

---

Elektrotehnički fakultet u Beogradu  
Katedra za računarsku tehniku i informatiku

*Predmet:* Operativni sistemi 1  
*Nastavnik:* prof. dr Dragan Milićev  
*Odsek:* Računarska tehnika i informatika  
*Kolokvijum:* Prvi, maj 2022.  
*Datum:* 8. 5. 2022.

*Prvi 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 = \_\_\_\_\_ % = \_\_\_\_\_ /15

**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)

Neki sistem vodi evidenciju alociranih (deklarisanih) logičkih segmenata (regiona) za svaki proces u jednostruko ulančanoj listi elemenata tipa `RegionDesc` uređenih po početnoj virtuelnoj adresi segmenta (polje `addr`). Svaki logički segment opisuje jedan deskriptor tipa `RegionDesc` u kom polje `size` sadrži veličinu segmenta. Ova lista je uvek neprazna, jer kernel za svaki proces, prilikom njegovog pokretanja, alocira jedan logički segment na najnižim adresama virtuelnog adresnog prostora (počev od adrese 0) koji preslikava u svoj memorijski prostor.

```
struct RegionDesc {
    byte* addr; size_t size;
    RegionDesc* next;
};
```

```
void* createRegion (RegionDesc* phead, size_t sz);
```

Implementirati funkciju `createRegion` koja se koristi u implementaciji sistemskog poziva `mmap` sa parametrom `start_addr` jednakim `NULL`. Ova funkcija treba da alocira (deklariše) nov logički segment tražene veličine (`sz`) na bilo kom mestu u virtuelnom adresnom prostoru procesa u kom ima dovoljno prostora tražene veličine tako da se ne preklapa sa već alociranim segmentima i uključi ga u evidenciju segmenata procesa. U slučaju uspeha funkcija vraća početnu adresu alociranog segmenta, a u slučaju neuspeha vraća *null*. Najveća adresa u virtuelnom adresnom prostoru je `MAX_VADDR`. Argument `phead` je glava liste deskriptora segmenata datog procesa. Tip `byte` predstavlja celobrojni tip veličine jedne adresibilne jedinice. Ne treba poravnavati adrese i veličine segmenata. Dinamičku alokaciju prostora za potrebe struktura jezgra radi funkcija `kmalloc` koja ima isti potpis i dejstvo kao standardna C funkcija `malloc`.

Rešenje:

## 2. (10 poena)

U nekom sistemu adresibilna jedinica je bajt, virtuelni adresni prostor je veličine 4 GB, a fizički adresni prostor je veličine 1 GB. Sistem koristi segmentno-straničnu organizaciju memorije sa stranicom veličine 1 KB i segmentom maksimalne veličine 4 MB. Jedan ulaz u SMT zauzima dve 32-bitne reči; u nižoj je pokazivač (cela adresa) na PMT (ili 0 ako segment nije alocirano), a u višoj reči su prava pristupa (RWX) u najniža tri bita i maksimalan dozvoljen broj stranice u segmentu (*limit*) u bitima do njih; preostali biti se ne koriste. Jedan ulaz u PMT zauzima jednu 32-bitnu reč (koriste se najniži biti).

Da bi obradio sistemske pozive u kojima se neki parametar zadaje kao pokazivač (virtuelna adresa u adresnom prostoru pozivajućeg procesa), operativni sistem mora da konvertuje datu virtuelnu u fizičku adresu, jer se kod kernela izvršava u režimu bez preslikavanja adresa. Implementirati funkciju koja obavlja ovu konverziju (vraća *null* ako virtuelna adresa ne pripada alociranom delu virtuelnog adresnog prostora). Prvi parametar ove funkcije je pokazivač na SMT pozivajućeg procesa. Pretpostaviti da je `uint32` deklarisan kao neoznačeni 32-bitni celobrojni tip. Napisati sve potrebne deklaracije za tipove SMT i PMT.

```
void* v2pAddr (SMT pmt, void* vaddr);
```

Rešenje:

### 3. (10 poena)

Neki sistem koristi straničnu organizaciju memorije sa stranicom veličine 128 KB. Adresibilna jedinica je 16-bitna reč, virtuelni adresni prostor je veličine 8 GB, a fizički adresni prostor je veličine 2 GB. PMT je u jednom nivou. U deskriptoru stranice u PMT najviša dva bita koduju prava pristupa, dok je u najnižim bitima broj okvira; ako stranica nije alocirana, ceo deskriptor ima vrednost 0.

a)(3) Prikazati logičku strukturu virtuelne i fizičke adrese i označiti širinu svih polja.

b)(7) Implementirati funkciju `sharePages` koja se koristi u implementaciji sistemskog poziva kojim se zahteva logičko deljenje niza od `cnt` susednih stranica počev od stranice `pg1` procesa čija je PMT data parametrom `pmt1` sa nizom od `cnt` susednih stranica počev od stranice `pg2` procesa čija je PMT data parametrom `pmt2`, pri čemu su pomenute stranice drugog procesa već alocirane u okvire, a pomenute stranice prvog procesa nisu, već treba da se dele sa drugim. Ukoliko drugi proces nije alocirao sve pomenute svoje stranice ili je prvi proces već alocirao neku od svojih pomenutih stranica, ova funkcija treba da vrati grešku (negativnu vrednost). Tip `long` je širine 64 bita, tip `int` 32 bita, a tip `short` 16 bita.

```
typedef unsigned short ushort;
typedef ushort PMT[PMT_SIZE];

int sharePages (PMT pmt1, ushort pg1, PMT pmt2, ushort pg2, ushort cnt);
```

Rešenje: