Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Operativni sistemi 2 (SI3OS2, IR3OS2)
Nastavnik: prof. dr Dragan Milićev
Odsek: Softversko inženjerstvo, Računarska tehnika i informatika
Kolokvijum: Drugi, decembar 2017.
Datum: 2. 12. 2017.

Drugi kolokvijum iz Operativnih sistema 2

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 = _____%

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) Mrtva blokada

U nastavku je dato jedno rešenje problema filozofa koji večeraju (engl. *dining philosophers*) korišćenjem standardnh brojačkih semafora, u kome postoji mogućnost mrtve blokade (engl. *deadlock*). Bez uvođenja dodatnih semafora, modifikovati prikazano rešenje tako da mrtva blokada bude sprečena eliminacijom uslova kružnog čekanja (engl. *circular wait*).

```
var forks : array 0..4 of semaphore = 1;

task type Philosopher(i:int)
  var left, right : 0..4;
begin
  left := i; right := (i+1) mod 5;
  loop
    think;
    forks[left].wait;  // take the left fork
    forks[right].wait; // take the right fork
    eat;
    forks[left].signal;  // release the left fork;
    forks[right].signal; // release the right fork;
  end;
end;
```

Rešenje:

2. (10 poena) Upravljanje memorijom

Neki sistem koristi modifikovani algoritam časovnika/nove šanse (engl. *clock/second chance*), tako što se, umesto bita referenciranja, u obzir uzima brojač pridružen svakoj stranici. Prilikom obilaska kazaljke, za žrtvu se izabira stranica čiji je taj brojač nula; ako je brojač veći od nule, smanjuje se za jedan i stranici daje nova šansa. Operativni sistem povremeno, na periodične prekide, inkrementira brojač pridružen stranici na osnovu njenog bita referenciranja, s tim da ne dozvoljava prekoračenje (ako je brojač stigao do maksimalne vrednosti, takav i ostaje).

Sistem primenjuje straničenje u dva nivoa, virtuelna adresa je 64 bita, adresibilna jedinica je bajt, a smeštanje višebajtnih reči u memoriju je po šemi niži bajt-niža adresa (*little endian*). Svaki ulaz u PMT drugog nivoa, odnosno deskriptor stranice, sadrži dve 64-bitne reči. Nižu reč koristi hardver prilikom preslikavanja, dok je viša reč potpuno slobodna za korišćenje od strane operativnog sistema. Tu reč kernel koristi tako što u najniža 24 bita čuva nižih 24 bita virtuelne adrese iste ovakve (više) reči deskriptora sledeće stranice u kružnoj listi stranica uvezanih po FIFO principu za algoritam zamene stranica, u naredna 24 bita isti takav pokazivač na prethodnu u toj listi, a u najviših 16 bita smešta pomenuti brojač korišćenja stranice. Kernel preslikava svoj deo fizičke memorije u najviših 16 MB virtuelnog adresnog prostora svakog procesa, a prilikom izvršavanja kernel koda preslikavanje odgovara tekućem procesu (aktivan je memorijski kontekst tog procesa).

Primenjuje se globalna politika zamene stranica, a globalna promenljiva `clockHand` tipa `uint64*` predstavlja kazaljku za algoritam časovnika i ukazuje na drugu (višu) reč deskriptora stranice.

```
typedef unsigned long  uint64; // 64 bits
typedef unsigned int   uint32; // 32 bits
typedef unsigned short uint16; // 16 bits
typedef unsigned char  uint8;  // 8 bits
uint64* getVictim ();
```

Implementirati funkciju `getVictim` koja treba da vrati pokazivač na prvu (nižu) reč deskriptora stranice izabrane za zamenu.

Rešenje:

3. (10 poena) Upravljanje memorijom

Neki sistem koristi sistem parnjaka (engl. *buddy*) za alokaciju memorije. Adresibilna jedinica je bajt, a sistem parnjaka upravlja delom operativne memorije počev od adrese A00000h i veličine 16 stranica, sa stranicom od 4 KB kao najmanjom jedinicom alokacije.

Trenutno su slobodni sledeći blokovi memorije (data je adresa i veličina bloka u stranicama; sve vrednosti su heksadecimalne):
A06000/2, A08000/2, A0C000/4.

a)(2) Popuniti drugu kolonu sledeće tabele navodeći početne adrese slobodnih delova memorije odgovarajuće veličine koji su uvezani u listu u datom ulazu tabele, u skladu sa implementacijom strukture za potrebe ovog algoritma.

0	
1	
2	
3	
4	

b)(4) U stanju iz tačke a) izvršava se operacija alokacije bloka memorije veličine 4 KB. Popuniti tabelu za stanje strukture nakon izvršavanja te operacije alokacije.

0	
1	
2	
3	
4	

c)(4) U stanju iz tačke b) izvršava se operacija dealokacije dela memorije veličine 8 KB počev od adrese A0A000h. Popuniti tabelu za stanje strukture nakon izvršavanja te operacije dealokacije.

0	
1	
2	
3	
4	