

### Ispit iz predmeta Operativni sistemi 1

Ime i prezime: \_\_\_\_\_

Broj indeksa: \_\_\_\_\_ Broj poena: \_\_\_\_\_/30

*Ispit traje 90 minuta. Nije dozvoljeno korišćenje literature.*

**1.(3)** Šta je *vremenska raspodela (time sharing)* u interaktivnim sistemima i šta je bio motiv za njeno uvođenje?

Odgovor:

**2.(3)** Na assembleru procesora picoRISC napisati prevod za sledeću rekurzivnu C funkciju:

```
unsigned fibb (unsigned n) { return (n>1)?(fibb(n-1)+fibb(n-2)):1; }
```

Rešenje:

**3.(3)** Neki jednostavan multiprocesni operativni sistem organizuje memoriju particionisanjem. Virtuelna adresa je 16-bitna, deo fizičke memorije za smeštanje procesa je veličine 1 MB, adresibilna jedinica je bajt, a stepen multiprogramiranja je 16. Da li je za ovakvu organizaciju potrebna hardverska podrška u vidu registra *limit* i provere prekoračenja njegove vrednosti? Precizno obrazložiti odgovor.

Odgovor:

**4.(3)** Ukratko, ali precizno objasniti šta operativni sistem treba da uradi u sistemskom pozivu kojim proces traži preslikavanje zadatog simbola iz navedenog DLL-a u svoju virtuelnu adresu.

Odgovor:

**5.(3)** Korišćenjem sistemskih poziva *fork* i *exec* napisati funkciju `spawn` sa potpisom datim dole, koja treba da pokrene proces dete nad zadatim programom sa zadatim jednim argumentom. U slučaju bilo kakve greške, ova funkcija treba da vrati kod te greške, a u slučaju uspeha da vrati 0.

```
int spawn (const char* program_file, const char* arg);
```

**6.(3)** Navesti načine na koje tekući proces gubi procesor, ali ostaje spreman za izvršavanje.

Odgovor:

**7.(3)** Međusobno isključenje pristupa kritičnoj sekciji jezgra od strane više procesora izvedeno je pomoću hardverske instrukcije tipa *test\_and\_set* kao što je dato dole. Precizno objasniti šta je greška u ovoj implementaciji i ispraviti je.

```
static int locked = 0;
#define lock { while (locked); test_and_set(locked); }
#define unlock { locked = 0; }
```

Rešenje:

**8.(3)** Šta je dvostruko baferisanje? Objasniti pojam, njegovu namenu i primenu.

Odgovor:

**9.(3)** Šta je efekat sistemskog poziva za otvaranje fajla koji predstavlja simboličku vezu (*symbolic link*)? A komande za brisanje simboličke veze?

Odgovor:

**10.(3)** Neki fajl sistem primenjuje FAT za alokaciju sadržaja fajla. FAT je cela keširana u memoriji, na nju ukazuje pokazivač *fat*, a 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  $\sim 0U$  (sve jedinice binarno); blokovi broj 0 i broj  $\sim 0U$  se ne koriste u fajl sistemu. U FCB polje *head* tipa *unsigned* sadrži broj prvog bloka sa sadržajem fajla (0 ako je sadržaj prazan). Napisati kod kojim se oslobađaju svi blokovi sa sadržajem fajla sa datim FCB.

R  
e  
š  
e  
n  
j