

Ispit iz predmeta Operativni sistemi 1

Ime i prezime: _____

Broj indeksa: _____

Broj poena: _____/30

Ispit traje 1,5 sat. Nije dozvoljeno korišćenje literature.

1.(3) Ako se nad sledećim programom kreira jedan proces, koliko će ukupno procesa biti kreirano (uključujući i taj jedan početni), pod pretpostavkom da su svi sistemski pozivi uspeli?

```
void main {  
    for (int i=0; (i<3) && fork(); i++);  
}
```

Odgovor:

2.(3) Šta je problem sledeće implementacije kritične sekcije uposlenim čekanjem?

```
process P1  
begin  
    loop  
        while flag2 = true do null end; (* Busy wait *)  
        flag1 := true;  
        <critical section>                (* Critical section *)  
        flag1 := false;                  (* Exit protocol *)  
        <non-critical section>  
    end  
end P1;
```

```
process P2  
begin  
    loop  
        while flag1 = true do null end; (* Busy wait *)  
        flag2 := true;  
        <critical section>                (* Critical section *)  
        flag2 := false;                  (* Exit protocol *)  
        <non-critical section>  
    end  
end P2;
```

Odgovor:

3.(3) Šta je osnovna razlika između tehnike dinamičkog učitavanja i tehnike preklopa (*overlays*)?

Odgovor:

4.(3) Ukratko objasniti zašto je kod segmentne organizacije virtuelne memorije obavezna provera prekoračenja granice segmenta prilikom svakog adresiranja, a kod stanične organizacije ta provera ne postoji.

Odgovor:

5.(3) U nekom sistemu postoje sledeći sistemski pozivi:

```
int async_write (const char* buffer);  
void wait (int request_id);
```

Operacija `async_write` asinhrono zadaje operaciju izlaza datog niza znakova na neki izlazni uređaj i vraća interni sistemski identifikator tog zahteva (veći od 0), odnosno kod greške (manji od nula). Operacija `wait` suspenduje pozivajući proces sve dok operacija sa datim identifikatorom nije završena. Korišćenjem ovih sistemskih poziva, realizovati sinhroni izlaz (funkcija treba da vraća 0 u slučaju uspeha, a kod greške u suprotnom):

```
int sync_write (char* buffer);
```

Rešenje:

6.(3) Kakav problem može nastati kod ulančane alokacije fajla ako se greškom promeni sadržaj bloka u kom je smešten deo sadržaja nekog fajla?

Odgovor:

7.(3) Ukratko objasniti šta je inkrementalni, a šta totalni bekap (*backup*) fajl sistema i navesti primer operativnog sistema koji ima ugrađenu podršku za inkrementalni bekap.

Odgovor: