

### Ispit iz predmeta Operativni sistemi 1

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

Broj indeksa: \_\_\_\_\_ Broj poena: \_\_\_\_\_/21

*Ispit traje 1 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) u funkciji  $N$ , pod pretpostavkom da su svi sistemski pozivi uspeli?

```
const unsigned N = ...;
void f (unsigned n) {
    if (n==0) return;
    if (n%2) f(n-1);
    else if (!fork()) f(n-1);
}
void main { f(N); }
```

Odgovor:

**2.(3)** Šta je problem sledeće implementacije kritične sekcije uposlenim čekanjem (obratiti pažnju na to da procesi nisu sasvim simetrični)?

```
shared var flag1, flag2 : boolean = false;
```

```
process P1
begin
  loop
    while flag2 = true do null end;
    flag1 := true;
    <critical section>
    flag1 := false;
    <non-critical section>
  end
end P1;
```

```
process P2
begin
  loop
    flag2 := true;
    while flag1 = true do null end;
    <critical section>
    flag2 := false;
    <non-critical section>
  end
end P2;
```

Odgovor:

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

Odgovor:

**4.(3)** Virtuelni adresni prostor je veličine 1MB i organizovan je stranično, adresibilna jedinica je bajt, a veličina stranice je 256B. Neki proces je alocirao dva logička segmenta, oba su veličine po 16 stranica, jedan zauzima najniže, a drugi najviše adrese u adresnom prosturu. Stranice se učitavaju na zahtev, a trenutno nije učitana nijedna stranica ovog procesa. Posmatraju se tri date virtuelne adrese nezavisno (svaka za sebe kao da se izvrši sama u opisanom stranju); hardver je prilikom preslikavanja adrese generisao izuzetak zbog nemogućnosti preslikavanja (*page fault*). Navesti kako će operativni sistem obraditi svaki od ovih izuzetaka.

<i>Virtuelna adresa (hex)</i>	<i>Način obrade</i>
<b>004CA</b>	
<b>FF745</b>	
<b>F0745</b>	

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

```
int async_write (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:

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

Rešenje:

**6.(3)** Dat je rezultat komande `ls` za neki direktorijum na nekom sistemu nalik sistemu Unix. Korisnik Jane koji pripada grupi `staff` izvršava proces koji otvara fajl `WideTable.pdf` za upis. Da li će taj proces moći da otvori ovaj fajl?

```
$ ls -l
-rw-r--rw- 1 JohnDoe staff 549387 Jul 8 2019 WideTable.pdf
```

Odgovor:

**7.(3)** Navesti razlog zašto bi neki fajl sistem koristio klastere na disku različite veličine.

Odgovor: