

Ispit iz predmeta Operativni sistemi 1

Ime i prezime: _____

Broj indeksa: _____ Broj poena: _____/30

Ispit traje 90 minuta. Nije dozvoljeno korišćenje literature.

1.(3) Navedeno je nekoliko funkcija standardne biblioteke jezika C (*libc*). Za svaku od njih navesti da li u svojoj implementaciji sigurno sadrži sistemski poziv (upisati „Da“) ili najverovatnije ne sadrži sistemski poziv (iako bi teorijski i to moglo da bude deo funkcionalnosti jezgra operativnog sistema, za tim nema nikakve potrebe, pa se iz praktičnih razloga ne radi; upisati „Ne“).

Funkcija	Opis	Sadrži sist. poziv
<code>void* memcpy(void* dest, const void* src, size_t count)</code>	Kopira niz bajtova u memoriji sa jednog mesta na drugo.	
<code>void exit (int status)</code>	Završava izvršavanje programa.	
<code>int getchar(void)</code>	Učitava znak sa standardnog ulaza.	
<code>char* strchr(const char *str, int c)</code>	Traži prvu pojavu znaka u datom nizu znakova.	

2.(3) Precizno objasniti zašto klasičan linker svoj posao obavlja u dva prolaza (a ne može samo u jednom). Obrazloženje ilustrovati primerom.

Odgovor:

3.(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:

4.(3) Šta je DLL i čemu služi? Objasniti osnovni motiv za postojanje ovog koncepta.

Odgovor:

5.(3) Šta radi procesorska instrukcija tipa *test-and-set* i za šta se ona tačno koristi u operativnim sistemima?

Odgovor:

6.(3) Precizno objasniti ceo mehanizam u sistemima nalik sistemu Unix kojim se postiže to da sistemskim pozivom *kill* jedan proces gasi drugi proces (šta *tačno* radi ovaj sistemski poziv i kako dovodi do gašenja procesa).

Odgovor:

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

```
shared var flag1 = false, flag2 = false;
process P1
begin
  loop
    flag1 := true;
    while flag2 = true do null end;
    <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:

8.(3) Precizno objasniti šta radi sledeći fragment programa:

```
int fd = open("./mydoc.txt", O_CREAT|O_TRUNC|O_WRONLY);
if (fd<0) handle_error("Cannot open data file.");

int pid = fork ();

if (pid==0) {
  dup2(fd,1);
  execlp("./myprog.a", NULL);
  handle_error("Cannot open exe file.");
}
```

Odgovor:

9.(3) Precizno objasniti šta radi sledeća komanda: `chmod g=x, u+rwx, o-wx .././frd`

Odgovor:

10.(3) Precizno objasniti šta je i zašto najveći nedostatak ulančane organizacije fajla kod proizvoljnog pristupa fajlu. Zašto taj problem ne postoji kod FAT organizacije?

O
d
g
o
v
o