

1. Un proces execută la un moment dat:

```
sigaction(SIGUSR1, &sa, NULL);
```

iar la un moment ulterior

```
write(fd, buffer, 4);
```

În care din situații este mai probabilă înlocuirea majorității intrărilor din TLB? Motivați. (Argumentele și modul de folosire a funcțiilor se presupun corecte.)

2. Care este limita de spațiu de swap pentru un sistem cu magistrala de adrese de 32 de biți cu spațiul de adrese împărțit 2GB/2GB (user/kernel). Dar pentru un sistem cu magistrala de adrese de 64 de biți?

3. O funcție signal-safe este o funcție care poate fi apelată fără restricții dintr-o rutină de tratare a unui semnal (signal handler). De ce nu este malloc o funcție signal-safe? Oferiți o secvență de cod pentru exemplificare.

4. Un program execută următoarea secvență de cod:

```
for (i = 0; i < BUFLLEN; i++)
```

```
    printf("%c", buf[i]);
```

iar altul

```
for (i = 0; i < BUFLLEN; i++)
```

```
    write(1, buf+i, 1);
```

Care secvență durează mai mult? De ce?

5. Un proces P se găsește în starea READY. Precizați și motivați două acțiuni care determină trecerea acestuia în starea RUNNING.

6. De ce obținerea ordonată/ierarhică a lock-urilor previne apariția deadlock-urilor, respectiv apariția fenomenului de starvation?

1. Un proces execută la un moment dat:

```
sigaction(SIGUSR1, &sa, NULL);
```

iar la un moment ulterior

```
write(fd, buffer, 4);
```

În care din situații este mai probabilă înlocuirea majorității intrărilor din TLB? Motivați. (Argumentele și modul de folosire a funcțiilor se presupun corecte.)

2. Care este limita de spațiu de swap pentru un sistem cu magistrala de adrese de 32 de biți cu spațiul de adrese împărțit 2GB/2GB (user/kernel). Dar pentru un sistem cu magistrala de adrese de 64 de biți?

3. O funcție signal-safe este o funcție care poate fi apelată fără restricții dintr-o rutină de tratare a unui semnal (signal handler). De ce nu este malloc o funcție signal-safe? Oferiți o secvență de cod pentru exemplificare.

4. Un program execută următoarea secvență de cod:

```
for (i = 0; i < BUFLLEN; i++)
```

```
    printf("%c", buf[i]);
```

iar altul

```
for (i = 0; i < BUFLLEN; i++)
```

```
    write(1, buf+i, 1);
```

Care secvență durează mai mult? De ce?

5. Un proces P se găsește în starea READY. Precizați și motivați două acțiuni care determină trecerea acestuia în starea RUNNING.

6. De ce obținerea ordonată/ierarhică a lock-urilor previne apariția deadlock-urilor, respectiv apariția fenomenului de starvation?

În conformitate cu ghidul de etică al Catedrei de Calculatoare, declar că nu am copiat la această lucrare. De asemenea, nu am ajutat și nu voi ajuta pe nimeni să copieze la această lucrare.

Nume:.....

Grupă:

Semnătură:

În conformitate cu ghidul de etică al Catedrei de Calculatoare, declar că nu am copiat la această lucrare. De asemenea, nu am ajutat și nu voi ajuta pe nimeni să copieze la această lucrare.

Nume:.....

Grupă:

Semnătură: