

Nume și grupă:

Sisteme de Operare

27 mai 2011

Timp de lucru: 60 de minute

Notă: Toate răspunsurile trebuie justificate

1. O implementare de thread-uri în kernel (*kernel level threads*) respectiv o implementare de thread-uri în spațiul utilizator (*user level threads*) execută următoarea secvență (validă) de cod:

```
for (i = 0; i < 1024; i++)
    write(fd, "a", 1);
```

În cadrul căreia dintre cele două implementări vor fi generate mai multe apeluri de sistem?

2. De ce nu are sens operația de seek pe socket-uri?

3. Un sistem folosește pagini fizice (frame-uri) de 64KB. Precizați un avantaj, respectiv dezavantaj al folosirii unei astfel de dimensiuni de pagină fizică (frame), în locul unei pagini fizice (frame) de 4KB.

4. Un sistem dat folosește *copy-on-write*. Cu toate acestea se observă că timpul de creare a unui thread este semnificativ mai mic decât timpul de creare a unui proces. Care este principala cauză a acestei diferențe?

5. Fie următoarea secvență de cod:

```
char *a;
```

```
a = mmap(NULL, 1024 * 4096, PROT_READ | PROT_WRITE, MAP_ANONYMOUS | MAP_PRIVATE, -1, 0);
```

```
/* TODO: start */
```

```
...
```

```
/* TODO: stop */
```

```
pid = fork();
```

```
switch (pid) {
```

```
case -1:
```

```
    exit(EXIT_FAILURE);
```

```
case 0:
```

```
    int d = 0;
```

```
    /* page fault start */
```

```
    for (i = 0; i < 1024; i++)
```

```
        d += a[i*4096];
```

```
    /* page fault stop */
```

```
    break;
```

```
default:
```

```
    break;
```

```
}
```

Completați secțiunea marcată cu `/* TODO: start */` și `/* TODO: stop */` astfel încât să nu existe nici un page fault în secțiunea marcată cu `/* page fault start */` și `/* page fault stop */`.

6. Se dorește rezolvarea problemei filozofilor folosind doar monitoare. Care este numărul minim de monitoare necesar pentru rezolvarea problemei?

7. Pe un sistem cu suport OpenVZ rulează 10 containere. Pe un alt sistem rulează 10 mașini virtuale VMware. Toate mașinile virtuale/containererele rulează același sistem de operare. În care dintre cele două sisteme, planificatorul de procese al sistemului de bază (*host*-ului) lucrează cu mai multe procese?

8. Pe un sistem aflat în rulare, care dintre cozile READY, WAITING, RUNNING poate să nu conțină nici un proces?

9. De ce este considerat `TransmitFile` un apel *zero-copy*?

10. La o listare rezultă următorul output:

```
$ ls -lh /usr/  
[...]  
drwxr-xr-x  2 root  root      12K May 20 10:08 sbin  
drwxr-xr-x  6 root  root      4.0K May 14 08:52 src  
[...]
```

De ce directorul `sbin/` are dimensiunea (12K) mai mare decât a directorului `src/` (4K)?

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

Nume și grupă:

Semnătură:.....