

Nume și grupă:

Sisteme de Operare

30 mai 2011

Timp de lucru: 60 de minute

Notă: Toate răspunsurile trebuie justificate

1. În urma unui apel `fork`, atât procesul copil cât și procesul părinte apelează o funcție non-reentrantă. De ce nu reprezintă acest lucru o problemă?

2. În ce situație două adrese virtuale ale aceluiași proces accesează aceeași adresă fizică?

3. În ce context instrucțiunea:

```
a = 0x42
```

declanșează o operație de *swap in*?

4. Dați exemplu de situație în care un proces are deschise mai mulți descriptori de fișier decât există fișiere (*inode-uri*) în sistem.

5. Fie următorii timpi:

- **t1**: timpul de schimbare între două procese;
- **t2**: timpul de schimbare de context între două thread-uri cu implementare user (*user level threads*);
- **t3**: timpul de schimbare de context între două thread-uri cu implementare kernel (*kernel level threads*).

Sortați crescător (de la mic la mare) cei trei timpi și justificați sortarea.

6. Fie o operație sincronă non-blocantă (`read`) și o operație asincronă (`aio_read`), apelate ca mai jos (pseudocod):

```
write(handle, buf, BUFSIZ);
aio_write(handle buf, BUFSIZ);
```

Câți octeți vor fi scriși în `handle-ul dat`, în fiecare caz, după încheierea fiecărei operații inițiate de instrucțiunile de mai sus?

7. De ce, în cazul programului `ping`, prima operație realizată în funcția `main` este crearea unui socket raw?

8. De ce o bună parte din *flush*-urile de TLB au loc imediat după primirea unei întreruperi de ceas?

9. De ce, în general, un apel `mmap` nu declanșează o operație de înlocuire de pagină, indiferent cât spațiu se solicită să fie alocat?

10. Două procese P1 și P2 comunică printr-un mecanism de cozi de mesaje unidirecționale. Se folosesc două astfel de cozi (M1 și M2) pentru comunicație bidirecțională. Cozile nu există inițial.

Procesul P1 folosește M1 pentru a trimite un mesaj către P2, iar P2 folosește M2 pentru a trimite răspunsul. După aceasta, comunicația se încheie și cozile sunt distruse.

Descrieți secvența de operații de forma *create*, *open*, *read*, *write*, *close*, *destroy* pe care trebuie să le execute fiecare proces. Ce proces trebuie să pornească primul?

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:.....