Sisteme de operare

22 iunie 2010

Timp de lucru: 90 de minute

NOTĂ: toate răspunsurile trebuie justificate

1. Care dintre secțiunile de memorie de mai jos sunt proprii unui proces dar nu unui program/executabil? Justificați.
text, rodata, data, bss, heap, stack

2. Precizați o situație în care accesarea unei adrese virtuale valide produce page fault, fără a produce segmentation fault.

3. Presupunem că avem 3 page frames la dispoziție, toate inițial goale. Se realizează următorul şir de accese (numerele reprezintă pagini virtuale): 3 2 1 0 3 2 4 3 2 1 0 4. Câte page faulturi vor rezulta în urma folosirii algoritmului FIFO? Dar dacă se mărește numărul de page frames la 4?

4. Se consideră următoarea schemă de segmentare :

Segment	Base	Length
0	100	1000
1	1200	250
2	1800	300
3	2200	500
4	3000	800

Care dintre următoarele reprezintă adrese logice valide? Adresele sunt de forma (segment, offset) .
a. (0, 820)
b. (1, 430)
c. (2, 13)

5. Dați exemplu de situație în care operația lock(&mutex) conduce la invocarea scheduler-ului și un exemplu de situație în care nu conduce la invocarea scheduler-ului.

6. Un sistem de fișiere dispune de un bitmap pentru inode-uri (un bit specifică folosirea sau nu a unui inode) de 32KB. Câte symlink-uri pot fi create? (este suficient ordinul de mărime şi justificarea răspunsului)

7. Se dă următoarea secvență de execuție :

Thread A	Thread B
work_a1	work_b1
work_a2	work_b2

Realizați sincronizarea celor 2 fire de execuție folosind semafoare astfel încât work_a1 să se execute înainte de work_b2 şi work_b1 să se execute înainte de work_a2. Folosiți primitivele: **sem_init(sem_t *sem, int count), sem_up(sem_t *sem), sem_down(sem_t *sem)**.

8. Într-o aplicație multi-threaded există două threaduri. Primul execută o funcție CPU intensive, iar celălalt execută preponderent operații I/O. De ce folosirea implementării threadurilor la nivel user nu este de dorit?

9. De ce, înainte de a realiza un apel exec(), e recomandat să se închidă toate fișierele de care nu are nevoie procesul copil?

10. Care este numărul minim de apeluri de sistem generate de următoarea secvență de pseudocod? (toate apelurile de funcții se întorc cu succes)

```
acquire_mutex(&m);
write(fd, "abcd", 4);
free(p);
release_mutex(&m);
```

11. De ce nu se poate implementa un mecanism de memorie partajată pentru un sistem cu paginare inversată?

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: .....