

Nume și grupă:

Sisteme de Operare

26 mai 2012

Timp de lucru: 60 de minute

Notă: Toate răspunsurile trebuie justificate

1. Fie următoarea secvență de cod:

```
char *a;
int pid, i = 0;
a = malloc(1024 * 4096);

/* TODO: start */
...
/* TODO: stop */

pid = fork();
switch (pid) {
    case -1:
        exit(EXIT_FAILURE);
    case 0:
        /* page fault start */
        for (i = 0; i < 1024 * 4096; i++)
            a[i]++;
        /* page fault stop */
        break;
    default:
        break;
}
```

Completați, dacă este posibil, 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 */`. Motivați alegerea făcută.

2. O bibliotecă de lucru cu liste simplu înlănțuite folosește un *mutex* global pentru a asigura accesul exclusiv la elementele listei. Care este neajunsul acestei implementări? Este posibilă o implementare care folosește câte un *mutex* pentru fiecare element? De ce?

3. Atunci când alegem dimensiunea memoriei *swap* ce aspecte ale hardware-ului trebuie luate în considerare și de ce: numărul de procesoare, arhitectura CPU (32/64 biți), dimensiunea spațiului de stocare, dimensiunea memoriei?

4. Pe un sistem cu arhitectură pe 64 de biți, cu un singur procesor, există următoarele latențe:

- schimbarea registrelor – 10 ns
- flush TLB – 20 ns
- schimbarea tabelii de pagini – 100 ns
- schimbarea tabelii de descriptori de fișier – 100 ns

Considerând că nu există alte latențe, cât va dura o schimbare de context între două *thread-uri* ale aceluiași proces și de ce?

5. Un programator dorește să folosească o funcție *thread-safe* pentru calculul factorialilor unor numere într-un program *multi-threaded*. Care din implementările de mai jos este *thread-safe*? Care este preferabilă și de ce?

<pre>unsigned long fact1 (unsigned long n){ unsigned long i, ret = 1; for(i = 1; i <= n; i++) ret = i * ret; return ret; }</pre>	<pre>unsigned long fact2 (unsigned long n){ if (n == 1) return n; else return n * fact2(n - 1); }</pre>
---	---

6. Fie două fișiere *a* și *b*, aflate pe aceeași partiție *ext3* și două procese *P1* și *P2* care rulează simultan. *P1* efectuează scrieri în fișierul *a*, iar *P2* efectuează citiri din fișierul *b*. Știind că *inode-ul* fișierului *a* este diferit de *inode-ul* fișierului *b*, în ce situație este necesară sincronizarea celor două procese?

7. Fie afirmația "Pe orice partiție *ext3* există mai multe *dentry-uri* decât *inode-uri*." Precizați și justificați valoarea de adevăr a afirmației.

8. Dați un exemplu de situație în care un apel *read asincron non-blocant aio_read(handle, buf, BUFSIZ)*; se va întoarce cu valoarea *BUFSIZ* (toți octeții solicitați vor fi citați până la întoarcerea apelului).

9. Un utilizator dorește să descarce 20 de fișiere mari, de peste 10 GB. Descărcarea secvențială a fișierelor îi indică utilizatorului o viteză de descărcare constantă, egală cu lățimea de bandă a plăcii de rețea. În cazul inițierii tuturor descărcărilor simultan, suma vitezelor de descărcare reprezintă mai puțin de 10% din valoarea anterioară. Considerând că sistemul are resurse hardware suficiente (număr de core-uri, cantitate de RAM, spațiu pe disc, etc.) care este o cauză probabilă pentru comportamentul de mai sus?

10. Un administrator de sistem folosește un sistem gazdă cu arhitectură pe 32 de biți care susține o topologie de mașini virtuale. După o perioadă, topologia trebuie mutată pe un sistem cu arhitectura pe 64 de biți. Care soluție de virtualizare necesită cel mai mic efort de portare dintre VMware Workstation și LXC? Ignorați efortul de portare a sistemului de operare gazdă, a hipervizorului și a VMM-ului.

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:.....