Sisteme de operare

20 iunie 2010

Timp de lucru: 90 de minute

NOTĂ: toate răspunsurile trebuie justificate

1. Câte inode-uri va folosi un hard-link către un fișier aflat pe un sistem de fișiere diferit?

2. Fie următoarea secvență de comenzi:

```
touch a.txt
ln a.txt b.txt
ln -s b.txt c.txt
```

Comanda ln fără opțiuni creează hard link-uri, iar comanda ln cu opțiunea -s creează symbolic link-uri. Câte inode-uri, respectiv dentry-uri vor fi create în urma rulării comenzilor de mai sus?

3. Un proces execută secvența următoare în două situații diferite:

```
a = malloc(5000);
memset(a, 0, 5000);
```

Într-una din situații rezultă două page fault-uri, iar în alta trei page fault-uri. Explicați acest comportament.

4. Un program citește un fișier de pe disc, operație care durează T1. Imediat după prima rulare, se execută din nou programul și durează T2. T2 este semnificativ mai mic decât T1. Cum explicați?

5. Care proces este părintele proceselor zombie?

6. Precizați o situație în care accesarea unei adrese virtuale valide produce segmentation fault.

7. Este utilă folosirea "canary value" (stack smashing protection) în cadrul funcției de mai jos? Justificați.

```
void f(char *msg)
{
        char *buffer = malloc(10);
        strcpy(buffer, msg);
}

...
f("supercalifragilisticexpialidocious");
...
```

8. Un sistem S1 folosește segmentare. Timpul de translatare a unei adrese virtuale într-o adresă fizică este T1. Un sistem S2 folosește paginare, iar timpul de translatare este T2. Care dintre timpii T1 și T2 este mai mare?

9. Ce se întâmplă cu sistemul de bază (host) în cazul în care apare o eroare fatală la nivelul nucleului:
a) unei mașini virtuale VMware Workstation;
b) unui container OpenVZ.

10. Fie următoarele două secvențe de programe

```
       /* S1 */
fd = open(“a.txt”, O_RDWR | O_CREAT, 0644);

pid = fork();
if (pid == 0) {
        write(fd, "a", 1);
        close(fd);
        exit(EXIT_SUCCESS);
}

wait(&status);
write(fd, "b", 1);
```

```
        /* S2 */
void *thread_handler(void *arg)
{
        write(fd, "a", 1);
        close(fd);
        return NULL;
}

fd = open(“a.txt”, O_RDWR | O_CREAT, 0644);
pthread_create(&tid, thread_handler, NULL);
pthread_join(&tid, NULL);
write(fd, "b", 1);
```

În cazul secvenței S2 apelul `write(fd, "b", 1);` se întoarce cu eroarea EBADF. Care este explicația? De ce în primul caz nu se întâmplă același lucru?

11. Fie următoarea secvență de operații:

```
for (i = 0; i < N; i++)
        a[i] = 1;
```

Secvența este rulată pe două sisteme diferite care nu dispun de TLB sau memorie cache. Pe un sistem au loc N de accese la memorie iar pe un alt sistem 2*N accese. Secvența este identică și rulată în aceleași condiții (același program) pe ambele sisteme. Cu ce diferă cele două sisteme?

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: .....