

Nume și grupă:

Sisteme de Operare

10 iunie 2016

Timp de lucru: 90 de minute

Notă: Toate răspunsurile trebuie justificate

1. (7 puncte) Fie următoarea secțiune de cod:

```
char a[128];
...
a[900] = '\0';
```

Intruțiunea de atribuire nu rezultă în excepție de memorie (*segmentation fault*). Acest lucru se întâmplă, însă în momentul folosirii instrucțiunii `a[901] = '\0'`. Cum explicați?

2. (7 puncte) În ce situație un apel de tipul `write()` modifică atât cursorul de fișier cât și dimensiunea fișierului?

3. (7 puncte) Precizați și justificați o metrică numerică pentru nivelul de productivitate al unui planificator de procese.

4. (7 puncte) De ce operația de tipul `seek()` **nu** are sens pe dispozitive de tip caracter și pe socketi, ci doar pe dispozitive de tip bloc?

5. (7 puncte) De ce este de preferat să folosim spinlock-uri pentru regiuni critice mici și mutexuri pentru regiuni critice mari?

6. (10 puncte) Fișierele executabile conțin o secțiune numită PLT (sau *function stubs*) prin care se intermediază apelul funcțiilor de bibliotecă. Atacurile de tipul *ret-to-plt* urmăresc să folosească PLT pentru apelul funcțiilor de bibliotecă. De ce sunt de interes aceste atacuri pe sisteme care folosesc DEP (*Data Execution Prevention*) și ASLR (*Address Space Layout Randomization*)?

7. (10 puncte) Fișierul `/etc/shadow` conține rezumate hash (funcții de tip one-way) ale parolilor. De ce se urmărește ca funcțiile care calculează rezumatul hash al unei parole să fie costisitoare din punctul de vedere al timpului: o rulare de funcție (care calculează un hash pentru o parolă) să dureze, să nu fie obținut rezultatul instant?

8. (10 puncte) *Thrashing* este o problemă a unui mecanism de stocare în care componentele acestuia (pagini, intrări) sunt foarte des schimbate, ducând la o performanță scăzută; în loc să fie folosite, acele componente sunt foarte des schimbate. Cum are loc *TLB thrashing*?

9. (10 puncte) Un kernel preemptiv este unul în care poate întrerupe un proces de pe un procesor atunci când îi expiră cuanta sau este un alt proces prioritar, iar procesul rulează în spațiul kernel (*kernel space*). Un kernel nepreemptiv poate întrerupe procesul în acele condiții doar dacă rulează în spațiul utilizator (*user space*). Un kernel preemptiv este mai util pentru interactivitate sau pentru productivitate?

10. (10 puncte) `a.txt` și `b.txt` sunt două link-uri (nume) la același inode. Inode-ul are contorul de link-uri egal cu 2. Pe un sistem dat, după operația `mv a.txt /boot/` contorul de link-uri ajunge la 1 pentru inode. Cum explicați?

11. (25 puncte) Vi se propune să implementați un profiler pentru aplicații multithreaded la nivelul unui sistem de operare dat. Profiler-ul trebuie să ofere informații despre consumul de memorie, costul de execuție la nivelul funcțiilor, zone cu durată mai mare la nivelul unui program. Indicați deciziile de proiectare și implementare a profiler-ului:

a. Ce suport aveți nevoie din partea sistemului de operare/hardware-ului pentru a putea rula profiler-ul? **(6 puncte)**

b. Cum veți colecta informațiile necesare pentru un proces/program dat? Cum va arăta modul de rulare a profiler-ului? Ce opțiuni (de forma celor în linia de comandă) va avea? **(7 puncte)**

c. Ce funcționalități trebuie să ofere profiler-ul pentru a oferi informații utile pentru profilarea thread-urilor multiple? **(6 puncte)**

d. Cum puteți folosi informațiile oferite de profiler pentru a mări eficiența programului? Ce urmăriți în rezultatele profiler-ului și ce decizii veți lua de modificare a programului pentru a fi mai eficient? **(6 puncte)**

În conformitate cu ghidul de etică al Departamentului 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:.....