

Nume și grupă:

Sisteme de Operare

15 iunie 2017

Timp de lucru: 90 de minute

Notă: Toate răspunsurile trebuie justificate

1. (7 puncte) Un programator dorește să transforme programul său într-un daemon Linux. Care sunt pașii obligatorii pe care trebuie să îi parcurgă?

2. (7 puncte) Presupunând ca apelul `fork()` se execută cu succes, ce valoare va fi afișată de următorul program, și de câte ori:

```
int i = 10;
if (fork() == 0)
    i++;
else {
    sleep(1);
    i++;
    printf("%d", i);
}
```

3. (7 puncte) Care este cel mai rapid mod de a transfera o cantitate mare de date între două procese pe același sistem de operare? Argumentați răspunsul.

4. (7 puncte) Explicați de ce un inode nu poate avea hard links pe două sisteme de fișiere diferite.

5. (7 puncte) Dați exemplu de situație în care un proces generează page fault care nu duce la terminarea sa.

6. (10 puncte) Un programator instalează un handler pentru a prinde semnalul `SIGSEGV`. Care din următoarele acțiuni vor funcționa corect atunci când sunt implementate în handler-ul de semnal?

- a) afișarea unui mesaj de eroare la consolă
 - b) `execve()` pentru a reporni procesul afectat
 - c) citirea memoriei procesului și detectarea adresei problematice
- Argumentați răspunsul.

7. (10 puncte) Cum se explică faptul că două sisteme Linux conectate cu un link de 10Gbps pot transfera UDP la o viteză de maximum 5Gbps și TCP la 10Gbps?

8. (10 puncte) Un sistem de fișiere FAT folosește blocuri de 4KB. Indicați, în cazul cel mai defavorabil, pentru un fișier ce conține 1000 de blocuri de date, numărul de blocuri de metadate care trebuie citite pentru a citi întreg fișierul.

9. (10 puncte) Cum putem reduce riscul de exploatare pentru executabilele care au bitul `setuid` setat?

10. (10 puncte) Explicați cum putem ataca o vulnerabilitate de tip *stack buffer overflow* pe un sistem Linux 32 biți care folosește SSP (*Stack Smashing Protector*, numit și suport de *canary value*), dar are dezactivate DEP (*Data Execution Prevention*) și ASLR (*Address Space Layout Randomization*).

11. (25 puncte) Un programator dorește să scrie un program care testează viteza calculatorului său însă știe să folosească doar apelurile de sistem `open()` și `read()`. Programatorul folosește utilitarul `/usr/bin/time` pentru a măsura timpul de execuție total al programului său. Ajutați programatorul să scrie cod care:

- a. Măsoară viteza de citire din memoria cache a procesorului, presupunând că există un singur nivel. **(5 puncte)**
- b. Măsoară viteza de citire din memoria principală. **(5 puncte)**
- c. Măsoară viteza de citire de pe disc. **(5 puncte)**
- d. Estimează dimensiunea memoriei cache. **(5 puncte)**
- e. Estimează dimensiunea memoriei principale. **(5 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:.....