

Nume și grupă:

## Sisteme de Operare

15 iunie 2015

Timp de lucru: 90 de minute

**Notă:** Toate răspunsurile trebuie justificate


1. **(7 puncte)** De ce procesele I/O intensive primesc, în general, o prioritate mai mare decât procesele CPU intensive?
2. **(7 puncte)** De ce considerăm apelul `chroot()` o formă de sandboxing?
3. **(7 puncte)** Ce se întâmplă în cazul operației `down()` pe un semafor?
4. **(7 puncte)** De ce schimbarea de context între două user level threads ale aceluiași proces este mai rapidă decât schimbarea de context între două kernel level threads ale aceluiași proces?
5. **(7 puncte)** Ce conține o intrare în TLB (*Translation Lookaside Buffer*)?
6. **(10 puncte)** Pe un sistem cu suport DEP (*Data Execution Prevention*), un atacator urmărește să realizeze un apel `mprotect(..., PROT_EXEC, ...)` pe un set de pagini aparținând stivei. Ce urmărește atacatorul cu un astfel de apel? Cum îl va ajuta în continuarea atacului?
7. **(10 puncte)** Un programator măsoară durata celor două instrucțiuni de mai jos, realizate consecutiv:  

```
p = mmap(NULL, PAGE_SIZE, PROT_READ | PROT_WRITE, MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);
p[0] = 'a';
```

În urma măsurătorii observă că prima instrucțiune (`mmap`) durează 900 de nanosecunde iar a doua durează 3000 de nanosecunde (adică mai mult decât prima). Ce putem spune despre overhead-ul de timp cauzat de un apel de sistem față de overhead-ul de timp cauzat de page fault handler?

8. **(10 puncte)** Într-un program se realizează apelul:

```
fd = open("/home/student/myfile", O_RDONLY);
```

Apelul reușește. Stim că `/home/student/myfile` este un o legătură simbolică (*symbolic link*). Câte inode-uri sunt citite în cadrul apelului `open()` de mai sus?

9. **(10 puncte)** De ce, în general, pornirea unui proces durează mai mult la prima sa pornire decât la următoarele porniri?
10. **(10 puncte)** În ce situație atât apelul `send()` pe transmițător, cât și apelul `recv()` pe receptor (în aceeași conexiune TCP) sunt simultan blocați?
11. **(25 puncte)** Se propune implementarea unui profiler de aplicații multithreading cu obiectivul de a obține informații legate de profilul de performanță al acestor aplicații.
  - a. Ce informații/metrici sunt utile de a fi furnizate de profiler dezvoltatorului aplicației? În ce formă sunt utile să fie furnizate acestea dezvoltatorului (numere, medii, grafice, tabele)? **(7 puncte)**

- b.** Ce facilități trebuie să ofere sistemul de operare și hardware-ul pentru funcționarea profilelui? **(6 puncte)**
- c.** Cum va accesa profilerul facilitățile oferite de sistemul de operare și hardware (apeluri de sistem, memorie partajată, sisteme de fișiere, dispozitive virtuale etc.)? **(5 puncte)**
- d.** Ce facilități/metrici vor fi de interes pentru dezvoltatorul unei aplicații de transcoding (conversie video)? Cum va fi folosit profilerul de către dezvoltator pentru a profila/optimiza aplicația de transcoding? **(7 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:**.....