

Nume și grupă:

## Sisteme de Operare

29 mai 2017

Timp de lucru: 90 de minute

**Notă:** Toate răspunsurile trebuie justificate


- 1. (7 puncte)** În ce situație un apel `down()` pe semafor nu blochează procesul apelant?
- 2. (7 puncte)** Un proces realizează într-o secvență de timp 3 schimbări de context voluntare și 200 de schimbări de context nevoluntare. Ce putem spune despre acest proces?
- 3. (7 puncte)** Un proces transmitător folosește 10 apeluri `send()` pentru a trimite 10 pachete a către 100 de octeți. Care este numărul **minim** de apeluri `recv()` pe partea receptorului pentru a primi toate datele transmitătorului?
- 4. (7 puncte)** O bibliotecă de transcoding (translatare dintr-un format în alt format) are suport de multithreading. De ce, în general, se recomandă ca numărul de thread-uri să fie egal cu numărul de core-uri?
- 5. (7 puncte)** De ce pentru placa de rețea poate avea sens să folosim, în anumite situații, polling în loc de întreruperi, dar nu are sens pentru hard disk?
- 6. (10 puncte)** Într-un sistem de fișiere un director are 100 de intrări și ocupă pe disc 4KB. În același sistem de fișiere un director are 50 de intrări (mai puține) și ocupă pe disc 8KB (mai mult). Cum explicați că un director cu număr mai mic de intrări ocupă mai mult spațiu pe disc?
- 7. (10 puncte)** Un atacator proiectează o aplicație malware în care ascunde o vulnerabilitate de tip buffer overflow și un apel `system("/bin/bash")`. De ce atacatorul va putea exploata vulnerabilitatea și obține un shell prin rularea aplicației și pe un sistem cu suport de ASLR (*Address Space Layout Randomization*)?
- 8. (10 puncte)** Un proces solicită sistemului de operare să ruleze cu cerințe real time. De ce sistemul va evita folosirea spațiului de swap pentru acest proces?
- 9. (10 puncte)** În cadrul spațiului de adrese ale unui același proces, două adrese virtuale distințe pot referi aceeași adresă fizică (prin mecanismul de memorie virtuală și prin maparea a două pagini virtuale la aceeași pagină fizică). Care este un caz de utilizare pentru această situație?
- 10. (10 puncte)** Un sistem folosește sandboxing. Un proces poate să-și stabilească un sandbox propriu printr-un apel `apply_sandbox(sandbox_profile)`, unde `sandbox_profile` este un profil de sandbox cu reguli. Cea mai importantă regulă este cea care împiedică procesul să invoce apelul de sistem `open()` pentru deschiderea unei fișiere. De ce este utilă această regulă și în ce moment se apelează `apply_sandbox(sandbox_profile)`?
- 11. (25 puncte)** O companie realizează o implementare de API low-level portabil peste funcționalitatea de bază dată de diverse sisteme de operare (creare de procese, thread-uri, sincronizare etc.): Windows, Linux, macOS. Obiectivul este să fie ușor de creat aplicații portabile și eficiente.
  - a. Ce va conține API-ul expus de implementare? **(5 puncte)**
  - b. Pentru o funcție dată expusă de API (la alegerea voastră), schițați modul de implementare.

**(8 puncte)**

- c. Cum veți trata cazurile în care nu există echivalent între două sisteme de operare? **(5 puncte)**
- d. Implementarea se dorește a fi eficientă. Cum veți măsura eficiența? **(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:**.....