

Nume și grupă:

Sisteme de Operare

13 septembrie 2014

Timp de lucru: 60 de minute

Notă: Toate răspunsurile trebuie justificate

- (7 puncte)** Câte tabele de pagini se găsesc la un moment dat într-un sistem de operare?
- (7 puncte)** De ce overhead-ul unui apel de sistem (oricât de simplu ar fi apelul de sistem) este, în majoritatea cazurilor, semnificativ mai mare decât overhead-ul unui apel de bibliotecă? Presupunem că vorbim de un apel de bibliotecă care nu face în spate apel de sistem
- (7 puncte)** În ce situație o operație de tip `down()` pe un semafor blochează thread-ul curent și în ce situație nu îl blochează?
- (7 puncte)** De ce fiecare thread al unui proces dispune de o stivă proprie?
- (7 puncte)** Un proces CPU intensive este pornit și rulează timp de 30 de minute. La încheierea sa, se observă că acesta a consumat doar 20 de secunde timp de procesor. Cum se explică acest lucru?
- (10 puncte)** În general, nu se pot crea hard link-uri la directoare. Cu toate acestea numărul de link-uri aferente unui director diferă între directoare diferite; putem observa acest lucru prin rularea comenzii `stat` pe diverse directoare: `/`, `/home`, `/usr/lib`. De ce diferă numărul de link-uri între directoare?
- (10 puncte)** De ce codul dintr-un shellcode se încheie, în general, cu o instrucțiune pentru realizarea unui apel de sistem (de forma `int 0x80`)?
- (10 puncte)** Un utilizator rulează, în terminal, **de mai multe ori** cele două comenzi de mai jos (rulăm de mai multe ori ca să fie informațiile citit cache-uite și să nu afecteze rezultatul):

```
find /usr > find.out
find /usr
```

Utilizatorul observă că prima comandă (cu redirectare în fișier) durează semnificativ mai puțin decât a doua (fără redirectare, care afișează pe terminal). De ce?

- (10 punct)** Care este avantajul unui server web care folosește mai multe procese pentru servirea cererilor față de unul care folosește mai multe thread-uri?
- (10 puncte)** Un executabil are zona de cod de 1MB. Cu toate acestea, un proces creat din acest executabil ocupă, pe parcursul rulării, maxim 100KB de memorie RAM. Cum explicați?
- (15 puncte)** Dorim să implementăm o rețea peer-to-peer. Rețeaua va pune preț mai mult decât orice pe disponibilitatea conținutului (*availability*). Fiecare peer va rezerva un spațiu dat pe hard disk-ul propriu pentru fișiere care nu sunt ale sale dar care ne propunem să fie disponibile.

Perioada de timp cât un peer este activ și lățimea sa de bandă sunt factori în stabilirea nivelului de implicare (*involvement*) al acestui peer. Un peer implicat va putea descărca mai rapid date de la alți peeri; fiecare peer își controlează banda de upload și acordă mai mult peerilor implicați.

Rețeaua este folosită pentru transferuri de fișiere mici (muzică, documentație, mici fișiere video). Transferul se realizează doar între un peer și alt peer.

Care vor fi primitivele protocolului de comunicare între peeri?

Cum ați reține, în cadrul rețelei peer-to-peer, informațiile despre implicarea unui peer (involvement), pentru a fi accesate de peeri? Motivați alegerea.

Ce facilități veți folosi pentru o performanță cât mai bună a aplicației specifică unui peer (pentru transferul fișierelor pe rețea, stocarea fișierelor)?

Ce facilități oferă rețeaua pentru a asigura disponibilitatea conținutului? Adică fiecare fișier să fie stocat în cât mai multe locuri.

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:.....