

Nume și grupă:

Sisteme de Operare

12 iunie 2018

Timp de lucru: 90 de minute

Notă: Toate răspunsurile trebuie justificate

- (7 puncte) Prezentați un avantaj și un dezavantaj al paginării ierarhice.
- (7 puncte) Care este mecanismul prin care sistemele de fișiere moderne micșorează timpul de verificare al integrității la bootare? Susțineți răspunsul printr-un exemplu.
- (7 puncte) Care este avantajul rulării serviciilor folosind utilizatori diferiți (de exemplu, serverul web este rulat sub utilizatorul httpd, iar serviciul de email sub utilizatorul postfix)?
- (7 puncte) Se dă un sistem cu un singur CPU (un singur core) ce are în permanență 10 procese în starea READY. Considerați acest sistem ca fiind unul încărcat (load 100%)? Explicați.
- (7 puncte) Completați secvența de cod de mai jos astfel încât *Examen SO 2018* să fie scris într-un fișier numit `SO.txt`. Nu aveți voie să folosiți niciun apel de scriere în fișier (`write`, `fprintf`, etc.).

```
int main()
{
    ...
    printf("Examen SO 2018");
}
```

- (10 puncte) Se dau 2 fișiere de cod C:

<p>fis1.c:</p> <pre>int a[10] = {10}; int main() { int i; for (i = 0; i < 10; i++) { a[0]++; } }</pre>	<p>fis2.c:</p> <pre>int a[10] = {0}; int main() { int i; for (i = 0; i < 10; i++) { a[0]++; } }</pre>
--	---

Compilăm fiecare program și obținem câte un fișier executabil pentru fiecare: `fis1` și `fis2`. De ce fișierul executabil `fis1` este mai mare decât `fis2`?

- (10 puncte) Care este motivul pentru care următoarea secvență de cod nu afișează nimic?

```
int main()
{
    int *a = 0;
    printf("Examen SO 2018");
    *a = 0;
}
```

8. (10 puncte) Precizați două cazuri în care se realizează un context switch. Care dintre cele două este mai lent și de ce?

9. (10 puncte) Care va fi valoarea afișată de următorul cod?

```
int t;
int tfunc()
{
    int *s;
    s = &t;
    (*s)++;
}
int main()
{
    pthread_t t1, t2;
    pthread_create(&t1, NULL, tfunc, NULL);
    pthread_create(&t2, NULL, tfunc, NULL);
    t++;
    printf("%d", t);
}
```

10. (10 puncte) Având o aplicație client-server ce comunică peste un canal având throughput de 1MBps (1 mega byte pe secundă), clientul trimite folosind un singur apel `send` 20MB (mega bytes), iar apelul se întoarce cu succes fără a bloca, notificând clientul că cei 20MB s-au trimis. De ce a fost posibil acest lucru în contextul în care într-o secundă putem trimite doar 1 mega byte?

11. (25 puncte) Dorim proiectarea unui nou server web care să poată susține un număr mare de cereri pe care platforma Moodle a facultății noastre (cs.curs.pub.ro) îl are. Analizați implementarea (avantaje și dezavantaje cu exemple care să le susțină) folosind următoarele mecanisme:

a. Analizați implementarea (avantaje și dezavantaje cu exemple care să le susțină) folosind doar procese. (5 puncte)

b. Cum ați realiza sincronizarea între procese? (5 puncte)

c. Analizați implementarea (avantaje și dezavantaje cu exemple care să le susțină) folosind doar thread-uri. (5 puncte)

d. La ce ar fi bun un model care se folosește și de procese și de thread-uri în același timp? (5 puncte)

e. Ce mecanisme de comunicare între procese ați putea folosi? Numiți cel puțin două. (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:.....