

Nume și grupă:

## Sisteme de Operare

5 septembrie 2018

Timp de lucru: 90 de minute

Notă: Toate răspunsurile trebuie justificate


- (7 puncte)** La ce este utilă paginarea ierarhică față de paginarea neierarhică?
- (7 puncte)** Pe un sistem de operare rulează două servicii: web și email. Cum putem proteja un serviciu în cazul unui atac asupra celuilalt?
- (7 puncte)** La ce este utilă jurnalizarea în cadrul sistemelor de fișiere?
- (7 puncte)** Se dă un sistem cu un singur procesor/core. Câte procese pot fi în starea **READY** la un moment dat? Dar în starea **RUNNING**?
- (7 puncte)** Scrieți secvența de cod care redirecțiază într-un fișier numit **eroare.txt** ieșirea de eroare standard.
- (10 puncte)** Se dau 2 fișiere de cod C:

```

fis1.c:
int a[1000] = {10,}; /* initialize first item to 10 */
int main()
{
    int i;
    for (i = 0; i < 1000; i++) {
        a[0]++;
    }
}
și

```

```

fis2.c:
int a[1000] = {0,}; /* initialize first item to 0 */
int main()
{
    int i;
    for (i = 0; i < 1000; i++) {
        a[i]++;
    }
}

```

Compilăm fiecare program și obținem câte 1 fișier executabil pentru fiecare: **fis1** și **fis2**. De ce fișierul executabil **fis1** este mai mare decât **fis2**?

- (10 puncte)** Un utilizator rulează secvența de comenzi de mai jos:

```

int *p, *q;
/* TODO */
*p = 1111;
printf("p: %p, q: %p, value_p: %d, value_q: %d\n", p, q, *p, *q);
*p = 2222;
printf("p: %p, q: %p, value_p: %d, value_q: %d\n", p, q, *p, *q);

```

Valorile afișate în cele două mesaje printf sunt, respectiv:

p: 0x80482132, q: 0x80729122, value\_p: 1111, value\_q: 1111  
p: 0x80482132, q: 0x80729122, value\_p: 2222, value\_q: 2222

Adică p și q, deși pointeri diferiți, referă aceeași valoare. Completați zona `/* TODO */` cu secvența de cod care conduce la acest comportament.

**8. (10 puncte)** Pentru a îmbunătăți accesul la date, în sistemele de calcul moderne au fost introduse două unități hardware: memoria cache și TLB. Explicați ce avantaje aduce fiecare.

**9. (10 puncte)** Care este diferența dintre cele 2 comenzi de mai jos:

```
ln examenso examen.so  
ln -s examen examen.so
```

Dați exemplu de un caz când sunt utile fiecare dintre ele.

**10. (10 puncte)** Având o aplicație client-server ce comunică peste un canal având throughput de 2MBps (2 mega bytes pe secundă), clientul trimite folosind un singur apel `send()` 20 de mega bytes, iar apelul se întoarce cu succes fără a se bloca, notificând clientul că s-au trimis doar 10MB. Care este motivul pentru care clientul a fost notificat că au ajuns doar 10MB? Toți acești 10MB au ajuns la server?

**11. (25 de puncte)** Dorim proiectarea unui sistem de votare distribuit pe mai multe noduri fizice. Un nod fizic poate conține unul sau mai mulți votanți.

- Descrieți o implementare optimă folosind thread-uri și procese? **(10 puncte)**
- Cum ați realiza sincronizarea între procese? **(10 puncte)**
- Este posibilă doar o implementare folosind thread-uri? Argumentați. **(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:**.....