

**SISTEME DE OPERARE**  
**EXAMEN SESIUNE DE RESTANTE**  
**1 SEPTEMBRIE 2023**

**INSTRUCTIUNI**

- Va rog sa va scrieti numele pe fiecare foaie pe care o predati (inclusiv pe foile cu subiectele de examen pe care le veti returna impreuna cu celelalte foi)
- Examenul se tine fara documentatie pe masa si fara acces la echipamente electronice (telefon mobil, tableta, etc)
- Aveti 100 minute pentru a termina examenul. Abordati examenul cu inteligenta: daca nu stiti pe moment raspunsul la o intrebare, treceti la urmatoarea si reveniti mai tarziu; dati raspunsuri concise si evitati sa pierdeti vremea furnizand detalii irelevante sau care nu sunt solicitate.
- Primele 10 minute sunt destinate citirii subiectelor. In tot acest timp nu aveti voie sa va atingeti de ustensilele de scris. Nerespectarea acestei conditii se pedepseste cu iesirea din examen si pierderea punctajului aferent.

**SUBIECTE**

1. (4 pte) Structura sistemelor de operare
  - (a) (2 pte) Dati doua exemple de organizare a structurii sistemelor de operare.
  - (b) (2 pte) Explicati pe scurt diferentele dintre cele doua exemple de la punctul anterior.
2. (5 pte) Exceptii si intreruperi
  - (a) (2 pte) Ce sunt exceptiile? Dati doua exemple.
  - (b) (1 pct) Care e rolul intreruperilor in operarea sistemelor de calcul?
  - (c) (2 pte) Ce face sistemul de operare (kernelul) ca raspuns la o intrerupere hardware? Descrieti mecanismul de tratare a intreruperilor.
3. (6 pte) Sincronizarea proceselor.
  - (a) (3 pte) Ce este un monitor? Care sunt avantajele folosirii monitoarelor?
  - (b) (3 pte) Ce sunt variabilele conditie si cum se folosesc?
4. (5 pte) Deadlock.
  - (a) (4 pte) Care sunt conditiile necesare producerii deadlock-ului?
  - (b) (1 pct) Cate dintre conditiile de mai sus trebuie indeplinite pentru a se putea produce un deadlock?
5. (6 pte) Planificarea proceselor.
  - (a) (2 pte) Ce este inversiunea de prioritati?



- (b) (2 pct) Dati un exemplu de inversiune de prioritate.  
 (c) (2 pct) Prezentați pe scurt o soluție pentru rezolvarea problemei inversiunii de prioritate.
6. (10 pct) Alocare proporțională a resurselor.  
 (a) (2 pct) Descrieți algoritmul planificării de tip loterie (*Lottery Scheduling, LS*).  
 (b) (2 pct) Dati un exemplu de planificare LS a 5 procese care detin 4, 6, 2, 5 respectiv 1 tichete.  
 (c) (1 pct) Care este complexitatea algoritmului de planificare de tip loterie?  
 (d) (5 pct) Folosind exemplul de la punctul b), prezentați o soluție de îmbunătățire a algoritmului LS cu complexitate logaritmică.
7. (14 pct) Gestiunea memoriei.  
 (a) (2 pct) Descrieți mecanismul generic de suport hardware pentru protecția memoriei.  
 (b) (2 pct) Ce este swapping-ul și la ce ajută?  
 (c) (3 pct) Dati 3 exemple de algoritmi de alocare dinamică a memoriei cu partiții fixe (gestiunea "hole"-urilor care apar în asemenea sisteme). Explicați pe scurt (1 frază) cum funcționează cei trei algoritmi aleși.  
 (d) (2 pct) Ce este fragmentarea memoriei și de câte tipuri este? Explicați pe scurt.  
 (e) (3 pct) Cum se face transformarea adresei logice în adresa fizică în sistemele de gestiune a memoriei cu paginare? (E suficient să explicați mecanismul de suport hardware cu ajutorul unei scheme).  
 (f) (2 pct) Rezolva paginarea vreuna dintre problemele fragmentării memoriei? Explicați.
8. (14 pct) Memoria virtuală.  
 (a) (2 pct) Cum funcționează paginarea la cerere (*demang paging*)?  
 (b) (2 pct) Care sunt avantajele folosirii paginării la cerere?  
 (c) (4 pct) Ce este un *page-fault* și cum este tratat de către sistemul de operare (kernel)?  
 (d) (2 pct) Care este formula timpului efectiv de acces la memorie (*Effective Access Time, EAT*) în sisteme cu paginare la cerere, dându-se rata  $p$  ( $0 \leq p \leq 1$ ) a *page-fault*-urilor?  
 (e) (2 pct) Ce este tehnica de *Copy-On-Write (COW)* și la ce folosește?  
 (f) (2 pct) Cate tipuri de algoritmi de înlocuire a paginilor există în funcție de setul de frame-uri din care se alege victima (frame-ul care conține pagina ce va fi evacuată din memorie)? Explicați pe scurt consecințele alegerii fiecăruia dintre algoritmi.
9. (12 pct) Alocarea memoriei.  
 Kernelul unui sistem de operare folosește un alocator de memorie de tip *Buddy System* pentru a gestiona o memorie de 4 KB. Kernelul efectuează următoarea secvență de operații de alocare/dealocare de blocuri de memorie, dimensiunea blocurilor alocate fiind furnizată ca parametru al apelului și reprezentată în octeți:



```

A = allocate(53);
B = allocate(1536);
C = allocate(512);
D = allocate(1536);
E = allocate(512);
free(C);
free(A);
free(B);
free(E);
free(D);

```

La fiecare operatie de alocare/dealocare evidentiati vizual harta memoriei, reprezentand hasurat blocurile de memorie alocate la momentul respectiv si nehasurat blocurile libere (evidentiati in acelasi timp, separat si listele cu blocuri libere disponibile si dimensiunea lor). Harta memoriei trebuie adnotata cu adresele de inceput si sfarsit ale **tuturor** blocurilor, deopotriva a celor alocate cat si a celor nealocate (libere). Explicati pe scurt regulile de alocare si dealocare care se aplica precum si consecintele lor (spargerea in blocuri mai mici si respectiv compactarea in blocuri mai mari, daca e cazul).

10. (12 pcte) O aplicatie concurenta de filtrare de imagini ruleaza pe un multiprocesor simetric cu 4 procesoare pentru a filtra o imagine pe care o imparte in 4 cadrane, filtrarea fiecarui cadran fiind delegata unuia dintre procesoare.

(a) (2 pcte) Ce solutie ati folosi pentru implementarea aplicatiei, una care foloseste 4 procese sau una care foloseste 4 thread-uri? Explicati raspunsul.

(b) (4 pcte) Daca presupuneti in mod simplificator ca singura secventa de cod inherent seriala (i.e., neparalelizabila) din aplicatia concurenta este crearea proceselor/thread-urilor si daca presupunem ca raportul dintre timpul de creare a unui proces ( $T_{CP}$ ) si cel al unui thread ( $T_{CT}$ ) este  $K$ , explicati raspunsul anterior folosind *legea lui Amdahl* considerand ca aveti la dispozitie un numar  $n$  (teoretic) infinit de procesoare.

*Indicatii:* Considerati ca  $T_{CP}$  si  $T_{CT}$  sunt exprimate procentual, mai exact ele sunt procentele aferente crearii de procese/thread-uri din timpul serial total al aplicatiei care este considerat egal cu 1, ca in *legea lui Amdahl*; reflectati la valorile posibile ale lui  $K$ .

(c) (2 pcte) Presupuneti ca numarul de procesoare initial (4) creste la o valoare finita  $N$  (sa zicem multiplu de 4). Ce stipuleaza *legea lui Gustafson* in acest context, adica ce anume se poate face cu acest numar crescut de procesoare, in conditiile in care *legea lui Amdahl* ne spune care sunt limitariile calculului paralel pentru o problema data?

(d) (2 pcte) Folositi *legea lui Gustafson* pentru a argumenta raspunsul de la punctul a), folosind aceleasi presupuneri de la punctul b), si anume ca singura secventa de cod inherent seriala (i.e., neparalelizabila) din aplicatia concurenta este crearea proceselor/thread-urilor iar raportul dintre timpul de creare a unui proces ( $T_{CP}$ ) si cel al unui thread ( $T_{CT}$ ) este  $K$ . Indicatiile de la punctul b) raman valabile.



11. (12 pte) Considerati ca singurele surse de overhead de memorie dintr-un sistem de paginare a memoriei sunt numarul de octeti utilizati de tabela de pagini folosita si respectiv numarul de octeti irositi datorita fragmentarii. De asemenea, considerati ca aceste aspecte sunt singurele care contribuie la stabilirea dimensiunii optime a unei pagini in sistemul respectiv.
- (a) (4 pte) Data fiind dimensiunea medie unui program in octeti ca fiind o variabila  $S$ , numarul de octeti al unei intrari in tabela de pagini a procesului ca fiind  $E$  si respectiv  $P$  dimensiunea paginii sistemului exprimata in octeti, aflati dimensiunea optima  $P_{opt}$  a unei pagini pentru un sistem care ruleaza programe de 4MB si utilizeaza 8 bytes pentru stocarea unei intrari in tabela de pagini.
- (b) (8 pte) Presupuneti ca sistemul utilizeaza o tabela inversata de pagini (*inverted page table*) si are o memorie RAM de  $M$  octeti. Care este formula pentru determinarea dimensiunii optime  $P_{opt}$  a paginii in acest caz? Gasiti o solutie aproximativa pentru  $P_{opt}$  considerand ca pentru valori mari ale lui  $x$  ( $x > 1000$ , de ex.),  $\log(x) / x^2 = 0$ .
- Indicatie: determinati numarul de bytes folositi de campurile esentiale ale unei intrari in tabela inversata de pagini pentru a integra aceasta constrangere in modelul de optimizare.

12. (4, 5, 6, 10 sau 14 pte, **subiect optional**) Aproape intotdeauna exista cel putin un subiect al cursului pentru care ai sentimentul ca te-ai pregatit bine, dar nu apar intrebari referitoare la el in examen. Formulati un subiect care este potrivit pentru acest examen si raspundeti la el.

Acest subiect poate fi folosit pentru a inlocui **cel mult unul** dintre subiectele 1 – 8. Subiectul nu va fi punctat decat daca specificati ce subiect ati ales pentru a fi inlocuit si numarul de puncte pe care il solicitati, care trebuie sa fie egal cu numarul de puncte aferent subiectului inlocuit.

**N.B.** Acest subiect e **optional**, el nu se puncteaza decat in lipsa unuia dintre subiectele 1 – 8 si in conditiile enuntate anterior.