

**SISTEME DE OPERARE**  
**EXAMEN FINAL**  
**7 FEBRUARIE 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 120 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. (5 pcte) Pentru fiecare termen, alegeti cea mai potrivita definitie:

Termen	Definitie
(i) Race condition	(a) Resursele necesare unui proces sunt detinute de altul si invers
(ii) Deadlock	(b) Cel mult un proces acceseaza datele partajate
(iii) Sectiune critica	(c) Alternare a accesului care produce rezultate deterministe
(iv) Excludere mutuala	(d) Acces intretesut care produce rezultate nedeterministe
(v) Sincronizare	(e) Cod care acceseaza date partajate
	(f) Evenimente care apar intr-o ordine bine specificata

2. (9 pcte) Concepte fundamentale in gestiunea memoriei

- (5 pcte) Dati definitii pentru: (a) adresa virtuala; (b) adresa fizica; (c) spatiu de adrese virtuale; (d) Present (valid) bit; (e) pagina vs frame
- (4 pcte) Ce este segmentarea memoriei? Enumerati trei diferente intre paginare si segmentare.

3. (10 pcte) Gestiunea memoriei

- (5 pcte) Descrieti operatia unui TLB (Translation Look-aside Buffer). Care sunt implicatiile folosirii TLB asupra context-switching-ului si ce solutii se pot folosi?
- (5 pcte) Ce este o tabela inversata de pagini si cand se foloseste? Ce se intampla cand exista pagini partajate in sistem (mentionati si solutii posibile)?



4. (12 pte) Controlul proceselor/thread-urilor

- (2 pte) Enumerați principalele stări în care se poate afla un proces în execuție și explicați în ce condiții se face tranziția între aceste stări? (O diagramă adnotată corespunzător e suficientă).
- (4 pte) Care e semantica apelului sistem Unix `vfork` și de ce era necesar? Care sunt limitările de utilizare (sau precauțiile necesare) atunci când se folosește `vfork`?
- (2 pte) Enumerați două avantaje ale thread-urilor kernel față de procese
- (2 pte) Enumerați două avantaje ale thread-urilor user față de cele kernel
- (2 pte) Enumerați două dezavantaje ale thread-urilor user rezolvate de soluții cum ar fi scheduler activations

5. (10 pte) Stocarea datelor

- (2 pte) Ce este un *i-node* Unix și la ce folosește?
- (2 pte) Ce este un *v-node* și la ce folosește?
- (6 pte) Un disk are 40 de cilindri. Fie o secvență de citiri care implică cilindrii 11, 1, 36, 16, 34, 9 și 12, în această ordine. Comparați timpul de căutare (*seek time*) al următorilor algoritmi: SSTF, SCAN (algoritmul liftului), C-LOOK. Exprimați timpul de căutare în termenii numărului de cilindri săriți de bratul de citire al discului pentru a satisface secvența de accese.

6. (8 pte) Un PC are 128 MB de RAM și folosește un microprocesor care implementează un spațiu de adrese pe 43 de biți, are pagini de 8 KB și folosește PTE-uri (intrări în tabela de pagini) de 8 bytes. Un program care rulează pe acest computer are 10 KB de text și date încărcate la baza spațiului de adrese și 5 KB de stivă la capatul superior al spațiului de adrese.

- Cat de mare ar fi o tabela liniară de pagini pentru a mapa acest program în RAM?
- Trasați o diagramă (o schema) care mapează acest program în memoria RAM folosind o tabela de pagini pe 3 niveluri care folosește un număr egal de biți de adresa pentru fiecare nivel. Cat spațiu ocupa această tabela de pagini?
- Descrieți o posibilă implementare pentru tabela de pagini de la punctul (a) și calculați spațiul de memorie fizică necesar stocării acestei tabele.

7. (16 pte) Funcția `int ticket(int *sequencer)` incrementează întregul referit de `sequencer` și întoarce valoarea incrementată.

```
int ticket(int *sequencer) {
    *sequencer += 1;
    return *sequencer;
}
```

Folosind primitivele de sincronizare de la punctele (a) și (b), scrieți versiuni ale funcției `ticket` care incrementează `atomic *sequencer` (valoarea referențiată de adresa de memorie care constituie operandul funcției) atunci când funcția rulează pe un sistem multiprocesor. În



fiecare caz, definiți semantica primitivei (felul în care operează) și specificați ce valoare întoarce.

(a) (4 pte) Test-and-Set: `int TAS(int *location);`

(b) (4 pte) Load-Linked: `int LL(int *location);` Store-Conditional: `int SC(int value, int *location);`

(c) (8 pte) Sistemul de operare SOS (Strange OS) oferă programatorului un singur tip de primitivă de sincronizare: *event counters* (contoare de evenimente). Un contor de evenimente este o variabilă de tip întreg  $E$  nedescrescătoare asupra căreia se pot executa următoarele operații:

**READ(E)** – citește valoarea lui  $E$

**ADVANCE(E)** – incrementează atomic valoarea lui  $E$

**AWAIT(E, v)** – blochează procesul apelant până când  $E$  ajunge la valoarea  $v$

Folosiți funcția **ticket** cu semantica de mai sus și contoare de evenimente pentru a implementa semafoare, adică furnizați (pseudo)cod pentru structura de date folosită de semafor și pentru operațiile **down** și **up**.

8. (14 pte) O aplicație multimedia conține trei thread-uri de timp real:  $T_v$  afișează stream-ul video iar  $T_s$  și  $T_d$  reprezintă canalele stâng și respectiv drept ale sursei de sunet stereo. Caracteristicile de timp în milisecunde ale acestor thread-uri sunt:

Thread	Perioada	Țimp de calcul
$T_v$	100	50
$T_s$	10	2
$T_d$	10	2

(a) (3 pte) Se pot planifica aceste thread-uri conform algoritmului Rate Monotonic (considerați ca  $\ln 2 = 0,69$ )? Comentati rezultatul obținut (explicați scurt rezultatul, nu va marginiti să răspundeți sec da/nu).

(b) (3 pte) Se pot planifica aceste thread-uri folosind algoritmul Earliest Deadline First? Comentati rezultatul obținut.

(d) (8 pte) Fie două thread-uri de timp real cu parametrii de timp exprimați în milisecunde ca în tabela de mai jos.

Thread	Țimp de sosire	Deadline	Țimp de calcul
$T_1$	0	17	10
$T_2$	2	10	5

Presupunând o cuantă de timp de 1 milisecundă, trasați diagrame de timp pentru execuția acestor thread-uri conform algoritmilor de planificare Earliest Deadline First, respectiv Least Slack First (planifica primul thread-ul cu cel mai mic **slack**, unde **slack** este durata maximă de timp cu care un thread poate fi întârziat fără să piardă deadline-ul).

9. (16 pcte) Biroul de Taxe si Impozite al unei primarii are un singur ghiseu la care un functionar se ocupa de incasarea taxelor si impozitelor datorate de contribuabili. In birou nu este permisa prezenta a mai mult de  $n$  contribuabili in asteptare la un moment dat. Daca nu exista nici un contribuabil doritor sa-si plateasca taxele, functionarul doarme pe scaunul sau. Cand apare un contribuabil, el trebuie sa-l trezeasca pe functionar daca doarme. Daca un contribuabil intra in birou si exista deja  $n$  contribuabili in asteptare, el paraseste incinta. Altfel, daca functionarul incaseaza taxe de la alta persoana, contribuabilul ramane in birou in asteptare. Programati o solutie care modeleaza comportamentul functionarului, respectiv al contribuabililor, folosind semafoare astfel incat sa nu apara race conditions.

10. (5, 8, 9, 10 sau 12 pcte, **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 – 6. 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 – 6 si in conditiile enuntate anterior.