

Model examen dat in 2019

1. (a) Ce este un proces zombie?
(b) Scrieti o secventa scurta de cod si aratati cand un process devine zombie (fara main)

2. Fie urmatoarea secventa

```
for( i = 0; i < 3; i++){  
    fork();  
    pthread_create();  
    fork();  
}
```

- (a) (5p) Cate procese si fire de executie sunt create? La numarare, ce presupunere ati facut legata de fork()?
- (b) (5p) Desenati arborescenta proceselor si firelor de executie create. Etichetati cu P procesele si cu T firele de executie.

3. Considerati problema filosofilor si solutia propusa mai jos pentru $n \in \mathbb{N}$ filosofi asezati la masa.
do {

```
    wait(chopstick[i]);  
    wait(chopstick[(i+1)%n]);  
    /* ... */  
    signal(chopstick[i]);  
    signal(chopstick[(i+1)%n]);  
} while (true);
```

Aceasta solutie permite aparitia fenomenului de *deadlock*.

- (a) (5p) Modificati solutia radicand asimetric betisoarele; filosofii impari ridica intai betisorul din stanga, cei pari pe cel din dreapta. Aratati ca nu mai apare fenomenul.
 - (b) (10 p) Aratati daca noua solutie satisface cele trei proprietati: exclusivitate mutuala, progresul si timp finit de asteptare.
4. Fie o matrice $A \in \mathbb{N}^{10 \times 10}$ tinuta contiguu in memorie pe linii si fie un system in care avem 3 frame-uri disponibile. In acest sistem intr-o pagina incap 10 intregi, iar programele P1 si P2 de mai jos incap in totalitate intr-o pagina.
P1:
for(i = 0; i < 10; i++)
 for(j = 0; j < 10; j++)
 A[i][j] = 0;
P2:
for(j = 0; j < 5; j++)
 for(i = 0; i < 5; i++)
 A[i][j] = 0;
(a) (5p) Cum arata programul si datele repartizate pe pagini?
(b) (5p) Folosind algoritmul LRU, care este programul efficient? De ce?
(c) (5p) Cum arata diagramele Gantt pentru P1 si P2?