

Problemă de examen: Sistem de programare pentru cabinete medicale

Context: La o clinică medicală se oferă 4 servicii care au loc în 4 cabinete diferite

1. Consultație oftalmologică (timp estimat: t_{oft})
2. Radiografie (timp estimat: t_r)
3. Consultație ortopedică (timp estimat: t_{ort})
4. Consultație orl (timp estimat: t_{orl})

Clinica are foarte mulți pacienți care vin și se pot înscrie pentru unul sau cel mult două servicii. Pacienții sunt preluați la intrare de mai mult asistenți $A(A=2)$ care adaugă pentru fiecare pacient un tichet de tip **{id, serviciul1, serviciul2}** într-o coadă globală (capacitate nelimitată), unde id este codul de identificare al clientului, iar serviciul1 și serviciul2 este un număr corespunzător unuia dintre serviciile 1, 2, 3, sau 4 (sau 0 pentru serviciul2 dacă nu se cere).

Pe zi se preiau 200 pacienți și se cere modelarea activității clinicii pe perioada unei zile (prin urmare pentru testare se vor genera random date pentru 200 de pacienți). Toți pacienții înscriși trebuie consultați.

Pentru fiecare serviciu, există o coadă specifică (capacitate nelimitată) formată prin tichete transferate din coada globală; există $W(W=3)$ threaduri care preiau înregistrări din coada globală și le adaugă în coada corespunzătoare serviciului.

Fiecare serviciu este gestionat de un thread dedicat. Fiecare thread al unui serviciu preia o înregistrare (id, serviciu1, serviciu2) din coada sa și, în urma consultației, se adaugă într-un fișier global de consultații efectuate o înregistrare de tip **(id, serviciu, timp)**; în plus dacă s-a cerut și al doilea serviciu atunci adaugă înregistrarea în coada corespunzătoare celui de-al doilea serviciu cerut.

Pentru a verifica activitatea din clinică, administratorul (1 thread) verifică periodic situația consultațiilor și a clienților care așteaptă. Această verificare presupune afișarea într-un fișier istoric(log) a următoarelor informații:

timp, nr_elemente_q1, nr_elemente_q2, nr_elemente_q3, nr_elemente_q4, nr_elemente_q)

Testare:

Pentru așteptarea condiționată, trebuie utilizat un mecanism de tip "wait-notify" (busy-waiting nu este acceptabil).

- $t_{oft} = 20\text{ms}$
- $t_r = 10\text{ms}$
- $t_{ort} = 40\text{ms}$
- $t_{orl} = 25\text{ms}$

Exemplu: id (1->1000), serviciu1 (1->4), serviciu2 (0->4, 0 înseamnă că nu există serviciu2)