

Laborator 4

Intr-un sistem exista 10 operatori care se afla in starea idle. Orice operator poate executa 2 task-uri, T1 sau T2. Numarul maxim de operatori care pot executa simultan task-ul T1 este 4, iar numarul maxim de operatori care pot executa simultan T2 este 6. Un operator care este idle, poate incepe executia lui T1 (startT1) sau poate incepe executia lui T2 (startT2). Dupa terminarea executiei lui T1 (endT1) (sau T2 - endT2), operatorul revine in starea idle.

Calculati invariantii si aratati ca:

P1: Numarul total de operatori (care executa un task la un moment dat sau sunt in idle) este 10.

P2: Numarul maxim de operatori care executa T1 este 4, iar numarul maxim de operatori care executa T2 este 6

Un birou de procesare a cererilor. Intr-o zi sunt programate 10 persoane care pot depune cereri (programarea nu se face pe ore)

- Un functionar: "ready", "working" sau "idle" (initial este "ready")
- O persoana (care are programare) depune o cerere (intr-un buffer C)
- Daca este in "ready", functionarul preia o cerere depusa si trece in "working". Functionarul va termina procesarea cererii, revenind in "ready"
- Daca este in ready si nu exista cereri depuse, functionarul poate trece in "idle".
- Daca functionarul este in "idle", functionarul poate reveni oricand in "ready"

Folosind invariantii, sa se arate ca:

P1: functionarul poate fi exact in una din stările "ready", "working" sau "idle" la un moment dat

P2: daca la un moment dat functionarul poate trece in "idle" atunci nu exista cereri depuse in acel moment

Fie un sistem cu un sender (S) si un receiver (R). S trimite mesaje formate din 5 biti catre R. S este initial intr-o stare S_ready. Daca S este in starea S_ready, atunci S poate trimite un bit, pe care il plaseaza intr-un buffer B, ajungand intr-o stare S_done. De aici, poate reveni in starea S_ready. **Daca S este in starea S_done si a trimis cei 5 biti care formeaza un mesaj, S trebuie sa treaca intr-o stare S_wait** (asteapta confirmare pentru mesaj).

R este initial intr-o stare R_ready. Din aceasta stare, daca in buffer exista un mesaj complet, poate prelua mesajul si ajunge intr-o stare R_done. Din aceasta stare poate trimite o confirmare catre S (intr-un buffer C), revenind in starea R_ready.

S, daca este in starea S_wait si exista o confirmare de la R, poate prelua confirmarea revenind in starea initiala.

Aratati ca:

- P1: S poate fi exact in una din starile S_ready, S_done, S_wait la un moment dat.
R poate fi exact in una din starile R_ready, R_done la un moment dat
- P2: Daca S a transmis toti bitii dintr-un mesaj si este in S_done, S nu poate reveni in starea S_ready
- P3: nr maxim de biti din B la un moment dat este 5, nr maxim de mesaje de confirmare din C este 1

P1, P2, P3 participanti la protocol, initial sunt toti in starea ready.

Daca P1 este in stare ready (P1r), trimite mesaj (plasat intr-un buffer m1) si trece intr-o stare waiting (P1w). P2, daca este in starea ready (P2r) preia mesajul de la P1, trece in stare "done" (P2d) si ulterior face forward la mesaj catre P3 (mesajul trimis intr-un buffer m2), trecand intr-o stare waiting (P2w), in care asteapta confirmare de la P3 ca a primit mesajul. P3, daca este in ready (P3r), preia mesajul de la P2 si trece intr-o stare "done" (P3d). Din aceasta stare poate trimite o confirmare catre P2 (plasata intr-un buffer c2), revenind in ready. P2 primește confirmarea de la P3, trecand intr-o stare P2_done_conf, dupa care trimite confirmarea catre P1 (intr-un buffer c1), trecand inapoi in ready. P1, daca este in waiting si are o confirmare in bufferul c1, preia confirmarea si revine in ready.

P1) Daca P1 e in starea ready, atunci si ceilalti sunt in starea ready;

P2) Daca P1 e in ready, bufferele sunt goale (si cele de mesaje si cele de confirmari)

P3) Daca P3 nu a receptionat mesajul de la P2, P1 nu poate trimite un nou mesaj