

Tema TPM

Rosca Alexandru-David, Octavian Regatun

Noiembrie 2024

1 Exercițiul 1

Avem secvența de execuție de mai jos.

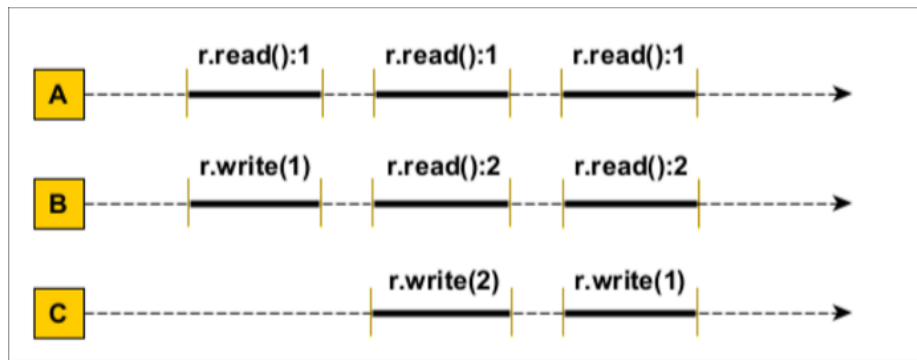


Figure 1: Secvența de Execuție

1.1 Liniarizabilitatea

Este aceasta liniarizabila?

Da, secvența de execuție este liniarizabilă deoarece există o ordine secvențială în care toate operațiile asupra variabilei `r` par să se fi desfășurat instantaneu, fiecare la un moment specific din această ordine, iar valorile obținute la citiri reflectă scrierile anterioare corespunzătoare la care ne așteptăm.

Acest concept se poate observa grafic în exemplul de mai jos care ilustrează o secvență liniarizabilă.

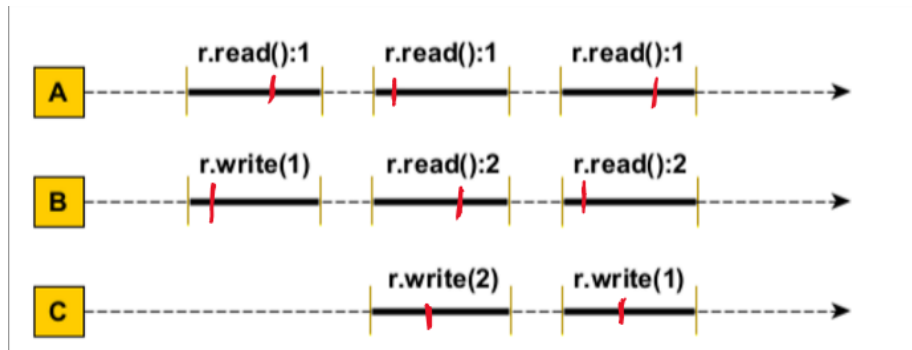


Figure 2: Exemplu Liniarizabilitate

Vom explica acum de ce figura de mai sus demonstrează faptul că secvența de execuție este liniarizabilă, arătând istoria secvenței de execuție și împărțind fiecare secțiune transversal și analizând fiecare caz pentru a determina valorile de intrare, intermediare și de ieșire posibile.

1.1.1 Secțiunea 1

În prima secțiune, observăm următoarele comportamente:

- Se intră cu valoarea $r = 0$ în secțiune, deci ca Thread-ul A să citească 1, cineva trebuie să scrie această valoare în r , lucru pe care poate să-l facă Thread-ul B, lucru marcat mai jos ca atare.
- Indiferent de momentul în care are loc scrierea, variabila r va avea valoarea 1 la finalul acestei secțiuni

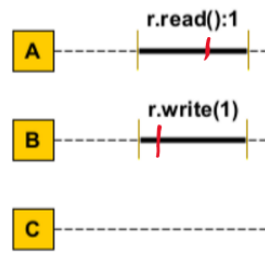


Figure 3: Secțiunea 1

1.1.2 Secțiunea 2

În secțiunea a doua, analizăm următoarele comportamente:

- Se intră cu valoarea $r = 1$ în secțiune, deci Thread-ul A poate să citească 1 direct, totuși Thread-ul B are nevoie să citească 2 deci cineva trebuie să scrie această valoare în r , lucru pe care poate să-l facă Thread-ul C.
- Indiferent de momentul în care are loc scrierea din partea Thread-ului C, variabila r va avea valoarea 2 la finalul acestei secțiuni

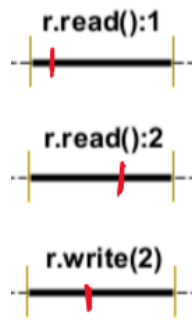


Figure 4: Secțiunea 2

1.1.3 Secțiunea 3

În secțiunea a treia, comportamentele observate sunt următoarele:

- Se intră cu valoarea $r = 2$ în secțiune, deci Thread-ul B poate să citească 2 direct de la început, totuși Thread-ul A are nevoie să citească 1 deci cineva trebuie să scrie această valoare în r , lucru pe care poate să-l facă Thread-ul C.
- Indiferent de momentul în care are loc scrierea din partea Thread-ului C, variabila r va avea valoarea 1 la finalul acestei secțiuni



Figure 5: Secțiunea 3

1.1.4 Istoria Secvenței de Executie

Pentru a demonstra ca această secvență este liniarizabilă, putem ordona operațiile astfel încât acestea să reflecte o ordine secvențială realistă a lor:

- B scrie valoarea 1 — Secțiunea 1 — r.write(1)
- A citește valoarea 1 — Secțiunea 1 — r.read(1)
- A citește valoarea 1 — Secțiunea 2 — r.read(1)
- C scrie valoarea 2 — Secțiunea 2 — r.write(2)
- B citește valoarea 2 — Secțiunea 2 — r.read(2)
- B citește valoarea 2 — Secțiunea 3 — r.read(2)
- C scrie valoarea 1 — Secțiunea 3 — r.write(1)
- A citește valoarea 1 — Secțiunea 3 — r.read(1)

1.2 Consistența Secvențială

Este secvența consistentă secvențial?

Da, secvența de execuție este consistentă secvențial deoarece liniarizabilitatea este o formă mai strictă a consistenței secvențiale.

Prin definiție, dacă o secvență este liniarizabilă, aceasta respectă ordinea secvențială și, în plus, asociază fiecărei operații un moment "instantaneu", ceea ce reprezintă o constrângere în plus.

Încât secvența este liniarizabilă ea verifică pe lângă toate cerințele de consistență secvențială și altele în plus.

Consistența secvențială permite o ordine secvențială fără a necesita instanțietate de care vorbeam anterior, în timp ce liniarizabilitatea impune atât respectarea ordinii cât și un punct specific în timp în care fiecare operație pare să se execute.