

Corso di Sistemi Operativi e Reti

Prova scritta del 16 SETTEMBRE 2020

ESERCIZIO 1 - PROGRAMMAZIONE MULTITHREAD

ISTRUZIONI PER CHI È IN PRESENZA:

1. **Rinomina** la cartella chiamata "Cognome-Nome-Matricola" che hai trovato sul Desktop e in cui hai trovato questa traccia, sostituendo "Cognome" "Nome" e "Matricola" con i tuoi dati personali e **lasciando i trattini**; se hai un doppio nome oppure un doppio cognome dovrai chiamare la cartella come in questo esempio:
 - a. DeLuca-MarcoGiovanni-199999
2. **Carica** tutto il materiale didattico che vorrai usare sul Desktop; puoi farlo solo nei primi 5 minuti della prova;
3. **Svolgi** il compito; lascia tutto il sorgente che hai prodotto nella cartella di cui al punto 1;
4. Quando hai finito lascia la postazione *facendo logout*,

senza spegnere il PC.

ISTRUZIONI PER CHI SI TROVA ONLINE:

1. **Questo file contiene il testo che ti è stato dato ieri, incluso il codice;**
2. **Mantieni a tutto schermo** questo file per tutta la durata della prova; puoi scorrere liberamente tra le sue pagine, ma non puoi cambiare applicazione;
3. **Firma** preliminarmente il foglio che userai per la consegna con nome cognome e matricola;
4. **Svolgi** il compito; puoi usare solo carta, penna e il tuo cervello;
5. **Alla scadenza** termina *immediatamente* di scrivere, e attendi di essere chiamato, pena l'esclusione dalla prova;
6. **Quando è il tuo turno** mostra il foglio ben visibile in webcam, e poi metti una foto dello stesso foglio in una chat privata Microsoft Teams con il prof.

ESERCIZIO 1 - QUESITO DA RISOLVERE

Introduci nella classe `DatoCondiviso` una terza tipologia di accesso al dato che si aggiunge alla modalità lettura e alla modalità scrittura. Chiameremo tale modalità T. E' necessario dunque implementare i metodi di acquisizione e rilascio in modalità T, rispettando le seguenti regole:

`acquireTLock(self)`: per acquisire il lock in modalità T, possono essere presenti al massimo 3 lettori attualmente in possesso dell'accesso in lettura, e non bisogna trovarsi nel caso in cui gli ultimi due thread ad avere acquisito il lock sono stati entrambi scrittori. Se le condizioni di acquisizione del lock non sono soddisfatte bisogna porsi in attesa e acquisire il lock quando tali condizioni sono verificate. Non ci sono vincoli al numero di acquisizioni contemporanee del lock in modalità T. E' possibile acquisire il lock in modalità T anche in contemporanea a uno scrittore e viceversa.

`releaseTLock(self)`: rilascia il lock in modalità T che è stato precedentemente acquisito. Quando un lock in modalità T viene rilasciato, è necessario che il prossimo thread che acquisisce il lock sia uno scrittore. Non deve essere possibile l'accesso in modalità T o in modalità lettura finchè questo non si verifica.

MATERIALE DIDATTICO

Il codice fornito implementa una classe che incapsula un dato accessibile con la tradizionale modalità Read/Write lock. Un meccanismo di controllo della starvation impone che uno scrittore non attenda più di 5 operazioni di locking in lettura prima di ottenere accesso in scrittura. I metodi offerti dalla classe `DatoCondiviso` sono:

`acquireReadLock(self):`

Acquisisce il lock in lettura sul dato condiviso, secondo le normali politiche di accesso in lettura, ma con controllo della starvation.

`releaseReadLock(self):`

Rilascia il lock in lettura sul dato condiviso, secondo le normali politiche di accesso in lettura.

`acquireWriteLock(self):`

Acquisisce il lock in scrittura sul dato condiviso, secondo le normali politiche di accesso in scrittura, ma con controllo della starvation.

`releaseWriteLock(self):`

Rilascia il lock in scrittura sul dato condiviso, secondo le normali politiche di accesso in scrittura.

`getDato(self):` Restituisce il valore del dato condiviso.

`setDato(self, v):` Imposta a v il valore del dato condiviso.