

NON SPEGNERE IL PC A FINE ESAME

Corso di Sistemi Operativi e Reti

Prova scritta di GENNAIO 2020

PRIMA PARTE

ISTRUZIONI

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.

SALVA SPESSO il tuo lavoro

SALVA SPESSO

ESERCIZIO 1 (Programmazione multithread. Punti: 0-20)

Si deve progettare una struttura dati thread-safe, detta `RoundRobinLock`. Un `RoundRobinLock` può essere usato per gestire l'accesso a un unico microfono in una tavola rotonda di N partecipanti. Ogni partecipante può prendere il microfono se questo è libero, mentre si pone in attesa se il microfono è occupato; inoltre il microfono stesso viene ceduto alla propria destra al primo dei partecipanti che ha chiesto la parola.

Un altro possibile uso del `RoundRobinLock` è quando ci sono N gruppi distinti di fruitori di una risorsa, che però non possono usare la risorsa contemporaneamente. Si immagini una ciotola di cibo per animali, dalla quale possono mangiare contemporaneamente tutti i gatti, tutti i cani, tutte le galline, ma non gatti, cani e galline contemporaneamente.

I metodi di cui deve essere dotata la struttura dati sono:

```
__init__(self, N : int)
```

Costruisce un `RoundRobinLock` da N gruppi di partecipanti.

```
acquire(self, id : int)
```

Prova ad acquisire il `RoundRobinLock`, dichiarando un `id` partecipante che può andare da 0 a $N-1$. Il lock può essere acquisito subito se questo è libero, o anche se è occupato da altri partecipanti che hanno dichiarato lo stesso `id` del chiamante. In tutti gli altri casi è necessario porsi in attesa bloccante finché non arriva il proprio turno.

```
release(self, id : int)
```

Rilascia il proprio accesso al `RoundRobinLock` dichiarando il proprio `id` partecipante. Se, grazie all'operazione di rilascio appena effettuata, non ci sono più partecipanti con lo stesso `id` del chiamante ad occupare il `RoundRobinLock`, è necessario garantire che il turno di accesso sia garantito, in ordine di priorità, ai partecipanti con `id` consecutivi, cominciando dai partecipanti in attesa con identificativo $(id+1) \% N$, e passando eventualmente ai successivi `id` in attesa.

Le strutture dati devono essere implementate garantendo la necessaria thread safety; non è ammesso progettare strutture dati con potenziali situazioni di deadlock; è opportuno migliorare l'accessibilità concorrente alle strutture dati ed evitare, se presenti, situazioni di starvation.

NON SPEGNERE IL PC A FINE ESAME

CI SONO DEI PUNTI AMBIGUI NELLA TRACCIA? **COMPLETA TU**

È parte integrante di questo esercizio completare le specifiche date nei punti non esplicitamente definiti, introducendo o estendendo tutte le strutture dati laddove si ritenga necessario, e risolvendo eventuali ambiguità.

POSSO CAMBIARE IL PROTOTIPO DEI METODI RICHIESTI? **NO**

Non è consentito modificare il prototipo dei metodi se questo è stato fornito. Potete aggiungere qualsivoglia campo e metodo di servizio, e qualsivoglia classe ausiliaria, ma NON variare l'interfaccia dei metodi pubblici già specificati.

CHE LINGUAGGIO POSSO USARE? **PYTHON 3.X; oppure JAVA 7 o successivo**

Il linguaggio da utilizzare per l'implementazione è Python 3, o in alternativa, Java. È consentito usare qualsiasi funzione di libreria di Python 3.X o di Java 7 e versioni successive.

MA IL MAIN() LO DEVO SCRIVERE? E I THREAD DI PROVA? **SI**

Sebbene non saranno oggetto di valutazione, è obbligatorio scrivere un `main()` e implementare esplicitamente del codice di prova **funzionante**, che è comunque necessario per testare il proprio codice prima della consegna.

SALVA SPESSO