

NON SPEGNERE IL PC A FINE ESAME

Corso di Sistemi Operativi e Reti

Seconda prova scritta di LUGLIO 2019

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;
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

NON SPEGNERE IL PC A FINE ESAME

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

Si deve implementare un lock a contatore (detto anche in letteratura *semaforo*). Un lock a contatore, che chiameremo `CLock`, è dotato di un certo numero di permessi di accesso che può essere aumentato o diminuito a seconda delle circostanze. E' inoltre possibile impostare opzionalmente un numero massimo di permessi disponibili che non può essere mai superato. I metodi che deve avere un `CLock` sono:

```
CLock(permessi : int).
```

Costruisce un nuovo `CLock` con numero iniziale di permessi pari a `permessi`. Inizialmente non c'è limite sul numero massimo di permessi.

```
acquire().
```

Rimane in attesa bloccante se il numero di permessi disponibili è zero; non appena il numero di permessi disponibili diventa almeno pari a 1, decrementa il numero di permessi disponibili di 1 ed esce.

```
release().
```

Incrementa il numero di permessi disponibili di 1 ed esce. Se è stato impostato un limite massimo tramite il metodo `limita`, allora bisogna verificare che i permessi disponibili non eccedano il valore massimo raggiungibile. Se i permessi disponibili sono già pari a quelli massimi disponibili, allora `release` esce senza modificare i permessi attualmente disponibili.

```
limita(n : int)
```

Limita a `n` il numero massimo di permessi disponibili che si possono raggiungere. Se il parametro `n=0`, bisogna impostare il `CLock` per non avere nessun limite (esattamente come alle impostazioni iniziali).

```
getPermessi() -> int
```

Restituisce il numero attuale di permessi ancora disponibili.

```
getPermessiMax() -> int
```

Restituisce il numero massimo di permessi che si può raggiungere. Se non c'è limite sui permessi allora si restituisce 0.

NON SPEGNERE IL PC A FINE ESAME

ESEMPIO:

Supponiamo di avere il seguente spezzone di codice:

```
1. C = new CLock(3);    print(C.getPermessi());
2. C.limita(5);        print(C.getPermessi());
3. C.release();        print(C.getPermessi());
4. C.release();        print(C.getPermessi());
5. C.release();        print(C.getPermessi());
6. C.acquire();        print(C.getPermessi());
```

L'output che si dovrebbe vedere in console in corrispondenza di ciascuna print è:

Linea 1: 3

Linea 2: 3

Linea 3: 4

Linea 4: 5

Linea 5: 5

Linea 6: 4

Si noti che la terza invocazione di `C.release()` non aumenta i permessi disponibili poiché è stato toccato il valore massimo di 5.

Tutti i metodi richiesti per devono essere implementati garantendo la necessaria thread safety; non sono ammesse situazioni di deadlock; è opportuno migliorare l'accessibilità concorrente alle strutture dati ed evitare, se presenti, situazioni di starvation.

Non è consentito servirsi della classe `threading.Semaphore` di Python o della classe `java.util.concurrent.Semaphore` di Java.

NON SPEGNERE IL PC A FINE ESAME

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, che è comunque necessario per testare il proprio codice prima della consegna.

NON SPEGNERE IL PC A FINE ESAME

ESERCIZIO 2 (Linguaggi di scripting. Punti 0-10)

Il comando `lshw` riassume la configurazione dell'hardware presente nella propria macchina. Di ogni periferica viene riportato, tra le varie informazioni, il nome del produttore (**vendor** o **product**), una breve descrizione dell'hardware (**description**) e altri dettagli utili quali driver, firmware e versione prodotto.

Si scriva uno script perl dal nome `lshardware` che estenda le funzionalità del comando `lshw`. Di seguito è riportata la sinossi del comando da implementare:

```
./lshardware [-s|-b] nome_file
```

Lo script, a seconda delle opzioni inserite in fase di esecuzione, eseguirà una delle seguenti operazioni:

1. `./lshardware -s nome_file`: lo script esegue il comando `lshw` e salva tutto il suo output all'interno del file `nome_file`;
2. `./lshardware -b nome_file`: lo script legge da input il file `nome_file`, che contiene del testo nello stesso formato dell'output del comando `lshw`, ed effettua le seguenti operazioni sul suo contenuto:
 - a. Conta e stampa in ordine decrescente il numero dei dispositivi forniti per ogni **vendor** (a parità di numero di dispositivi, è necessario stampare il vendor in ordine alfabetico);
 - b. Per ogni **vendor**, stampa in STDOUT la **description** dei dispositivi forniti. La stampa dovrà essere ordinata alfabeticamente sul nome del vendor.

Di seguito è riportato un esempio di output del comando da implementare:

```
vendor: Dell Inc.  
num_devices: 3  
vendor: Intel Corp.  
num_devices: 3  
vendor: NVIDIA Corporation  
num_devices: 2  
  
vendor: Dell Inc.  
descriptions:  
    Portable Computer  
    Motherboard  
    BIOS  
vendor: Intel Corp.  
descriptions:  
    VGA compatible controller  
    Audio device  
    CPU  
vendor: Samsung  
descriptions:  
    ATA Disk_1  
    ATA Disk_2
```