

Corso di Sistemi Operativi e Reti

Prova scritta di LUGLIO 2019

ISTRUZIONI

1. **Rinomina** la cartella chiamata "CognomeNomeMatricola" che hai trovato sul Desktop e in cui hai trovato questa traccia, sostituendo "Cognome" "Nome" e "Matricola" con i tuoi dati personali;
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

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

Bisogna implementare una `BlockingQueue` FIFO con i tradizionali metodi di inserimento e rimozione (`put` = inserimento bloccante; `get` = prelievo bloccante), modificata per fornire alcune funzioni aggiuntive. Chiameremo questa nuova tipologia di coda `TimedBlockingQueue`. Le funzioni aggiuntive sono le seguenti:

```
timedPut(e, timeout).
```

Inserisce l'elemento `e` all'interno della `TimedBlockingQueue`. Qualora la coda sia piena, si attende finché non si libera un posto secondo la disciplina usuale. Dal momento dell'inserimento in poi, `timedPut` termina ed esce. In seguito, l'elemento `e` può rimanere nella coda per `timeout` secondi. Qualora trascorressero più di `timeout` secondi senza che `e` venga prelevato, allora `e` viene cancellato dalla coda.

```
waitFor(e).
```

Se l'elemento `e` è inserito nella coda, attende finché `e` non viene estratto da qualche thread, oppure finché non scade un eventuale suo corrispondente `timeout`. Restituisce `true` se l'elemento è stato estratto correttamente, oppure `false` se l'elemento risulta eliminato per scadenza del `timeout`. Viene inoltre restituito `false` immediatamente anche se l'elemento `e` non è correttamente presente nella coda.

ESEMPI (Si supponga di lavorare con la coda `Q`):

1. Supponiamo di inserire l'oggetto `O` all'interno di `Q` con `timeout` di 0.5 secondi, mediante la chiamata `Q.timedPut(O, 0.5)`, che inserisce `O` immediatamente e termina (*prima dello scadere degli 0.5 secondi*). Durante questo intervallo di tempo `O` potrebbe essere normalmente prelevato attraverso una chiamata al metodo `get`. Trascorsi gli 0.5 secondi `O` verrà invece eliminato dalla coda e non sarà più prelevabile.
2. Supponiamo che il thread `T1` inserisca l'oggetto `O` con la chiamata `Q.timedPut(O, 1.0)`. Successivamente il thread `T1` invoca `Q.waitFor(O)`. Quest'ultima chiamata si blocca: un thread `T2` potrebbe estrarre `O`, invocando il metodo `Q.get()` in meno di un secondo, e in tal caso la chiamata a `Q.waitFor(O)` uscirebbe restituendo `true`. Se invece `O` dovesse restare inserito in coda per più di un secondo senza essere estratto, `O` dovrebbe essere eliminato dalla coda e `T1` dovrebbe uscire dalla chiamata `Q.waitFor(O)`. Quest'ultima chiamata restituirà in tal caso `false`.

NON SPEGNERE IL PC A FINE ESAME

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 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

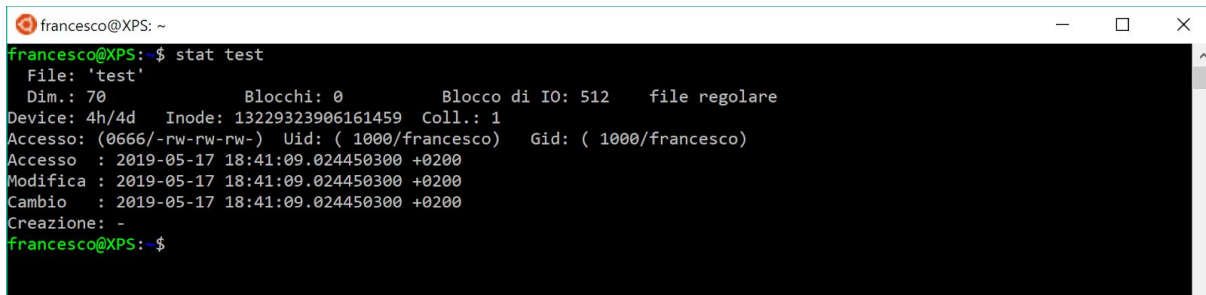
ESERCIZIO 2 (Linguaggi di scripting. Punti 0-10)

Nei sistemi Unix/Linux, il comando `stat` mostra informazioni dettagliate riguardo un determinato file o filesystem. Si scriva uno script Perl dal nome `stat.pl` in grado di estendere le funzionalità del comando shell `stat`. In particolare, lo script prenderà da linea di comando una serie di opzioni (non obbligatorie) e il nome di una directory (obbligatoriamente).

Sinossi del comando da implementare:

```
./stat.pl directoryName [OPTIONS]
```

Output del comando `stat` standard impartito su un file di prova chiamato 'test':



```
francesco@XPS: ~  
francesco@XPS:~$ stat test  
File: 'test'  
Dim.: 70          Blocchi: 0          Blocco di IO: 512   file regolare  
Device: 4h/4d   Inode: 13229323906161459  Coll.: 1  
Accesso: (0666/-rw-rw-rw-)  Uid: ( 1000/francesco)  Gid: ( 1000/francesco)  
Accesso  : 2019-05-17 18:41:09.024450300 +0200  
Modifica : 2019-05-17 18:41:09.024450300 +0200  
Cambio   : 2019-05-17 18:41:09.024450300 +0200  
Creazione: -  
francesco@XPS:~$
```

Lo script dovrà operare nel seguente modo:

1. nel caso in cui non ci dovessero essere opzioni in input, lo script mostrerà in standard output la somma dei parametri `Dim.`, `Blocchi` di tutti i file contenuti nella cartella `directoryName`. Infine, saranno riportati anche i nomi dei file con dimensione minima e massima (seguiti dalle rispettive dimensioni);
2. se si inserisce l'opzione `-b` dovrà essere effettuata soltanto la somma dei `Blocchi` seguita dai nomi dei file con minore e maggiore numero di blocchi occupati;
3. se si inserisce l'opzione `-t="tipoFile"` bisognerà effettuare la stessa operazione del punto 1 ma filtrando per i file specificati con l'opzione `-t`; se, ad esempio, usassimo l'opzione `-t="file regolare"` dovremo aggregare ed effettuare le somme solo sui file testuali (le informazioni sul tipo di file sono specificate subito dopo il parametro `Blocco di IO`);
4. NON SONO AMMESSE ALTRE OPZIONI