

NON SPEGNERE IL PC A FINE ESAME

Corso di Sistemi Operativi e Reti

Prova scritta di NOVEMBRE 2018

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

NON SPEGNERE IL PC A FINE ESAME

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

Si progetti una struttura dati thread-safe chiamata `MonitoredVector`. Un `MonitoredVector` è una collezione di N oggetti di tipo generico T con specifici metodi pubblici:

`void lock(int i)` . Mette un lock sull'elemento in posizione i .

`void unlock(int i)` . Rilascia un lock precedentemente preso sull'elemento in posizione i . Deve essere generata una eccezione se l'indice i non dovesse essere realmente soggetto a locking.

`void lock(int[] V)` . Mette un lock su tutti gli elementi del `MonitoredVector` il cui indice è elencato nel vettore V . Ad esempio se $V[0] = 1$ e $V[1] = 3$, bisogna bloccare l'elemento 1 e l'elemento 3 del `MonitoredVector`. Nel caso, attende finchè tutti gli indici specificati non sia liberi da lock.

`void waitForLock()` . Si blocca in attesa che non ci sia nessun lock su nessun elemento del `MonitoredVector`.

I metodi richiesti devono essere implementati garantendo la necessaria thread safety; non sono ammesse situazioni di deadlock; è opportuno migliorare l'accessibilità concorrente alla struttura dati ed evitare 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 DEVO USARE? **JAVA 7 O SUCCESSIVO**

Il linguaggio da utilizzare per l'implementazione è Java. È consentito usare qualsiasi funzione di libreria di Java 7 o successivi.

MA IL MAIN() LO DEVO SCRIVERE? E I THREAD DI PROVA? **SOLO PER FARE IL TUO DEBUG**

Non è esplicitamente richiesto di scrivere un `main()` o di implementare esplicitamente del codice di prova, anche se lo si suggerisce 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 `lsuf` nei sistemi Linux è utilizzato per mostrare all'utente quali file sono aperti in quali processi. Si noti che, in particolare la colonna `COMMAND` riporta il nome di un processo e la colonna `SIZE/OFF` riporta l'occupazione del processo in RAM.

```
# lsuf
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE NAME
init	1	root	cwd	DIR	253,0	4096	2 /
init	1	root	rtd	DIR	253,0	4096	2 /
init	1	root	txt	REG	253,0	145180	147164 /sbin/init
init	1	root	mem	REG	253,0	1889704	190149 /lib/libc-2.12.so
init	1	root	0u	CHR	1,3	0t0	3764 /dev/null
init	1	root	1u	CHR	1,3	0t0	3764 /dev/null
init	1	root	2u	CHR	1,3	0t0	3764 /dev/null
init	1	root	3r	FIFO	0,8	0t0	8449 pipe
init	1	root	4w	FIFO	0,8	0t0	8449 pipe
init	1	root	5r	DIR	0,10	0	1 inotify
init	1	root	6r	DIR	0,10	0	1 inotify
init	1	root	7u	unix	0xc1513880	0t0	8450 socket

Si scriva uno script in Perl dal nome `listFiles.pl` che estenda l'uso del comando `lsuf` come descritto di seguito. Lo script potrà essere eseguito nel seguente modo:

```
./lsuf.pl [execution_mode] [process] [backup_file]
```

1. Il parametro `[execution_mode]` è **sempre obbligatorio** e può essere `-r` o `-b`.
 - a. **Caso `-r`:** In questo caso è obbligatoria soltanto l'opzione `[process]`. Lo script dovrà eseguire il comando `lsuf` a runtime e salvarne il risultato in un file di testo dal nome `lsuf.out`. Il valore di `[process]` indicherà il nome di un singolo processo da isolare e salvare all'interno del file in output. Nel caso in cui `[process]` sia uguale a `"-1"` tutti i processi dovranno essere salvati nel file di output. **(si veda esempio esecuzione 3)**
 - b. **Caso `-b`:** In questo caso sono obbligatori gli argomenti `[process]` e `[backup_file]`. Nel caso in cui il parametro `[process]` sia uguale a `"-1"`, lo script dovrà leggere un file indicato tramite il parametro `[backup_file]`, raggruppare tutti i processi con lo stesso nome e mostrare la somma della colonna `SIZE/OFF`, in ordine di **valore della somma crescente**, per ogni processo. Se `[process]` fosse settato con il nome di un processo particolare, allora bisognerebbe calcolare e mostrare in output solo la `SIZE` totale occupata dal quel singolo processo.

NON SPEGNERE IL PC A FINE ESAME

ESEMPI (gli esempi 1 e 2 sono basati sul file backup.bk fornito. L'esempio 3 è indicativo rispetto a un output di lsof non fornito):

Esecuzione 1:

```
[execution_mode] = -b
[process] = chrome
[backup_file] = backup.bk
francesco@francesco:~/Scrivania/Novembre2018$ ./lsof.pl -b chrome backup.bk
chrome --> 368424531
_
```

Esecuzione 2:

```
[execution_mode] = -b
[process] = -1
[backup_file] = backup.bk
francesco@francesco:~/Scrivania/Novembre2018$ ./lsof.pl -b -1 backup.bk
gvfsd-dns --> 12982838
gvfsd-net --> 15388594
cinnamon- --> 84176413
evolution --> 125088652
dconf\x20 --> 162109329
gdbus --> 175092167
gmain --> 175092175
sandbox_i --> 208420632
chrome --> 368424531
```

Esecuzione 3:

```
[execution_mode] = -r
[process] = -1
[backup_file] = NON DEVE ESSERE PRESENTE
```

```
francesco@francesco:~/Scrivania/Novembre2018$ ./lsof.pl -r -1
francesco@francesco:~/Scrivania/Novembre2018$ ls
backup.bk  lsof.out  lsof.pl
francesco@francesco:~/Scrivania/Novembre2018$ cat lsof.out | head -n5
COMMAND  PID  TID      USER  FD      TYPE      DEVICE  SIZE/OFF      NODE NAME
systemd  1    1        root  cwd      unknown   -        -              /proc/1/cwd (readlink: Permission denied)
systemd  1    1        root  rtd      unknown   -        -              /proc/1/root (readlink: Permission denied)
systemd  1    1        root  txt      unknown   -        -              /proc/1/exe (readlink: Permission denied)
systemd  1    1        root  NOFD     -        -        -              /proc/1/fd (opendir: Permission denied)
```

NON SPEGNERE IL PC A FINE ESAME