NON SPEGNERE IL PC A FINE ESAME

Corso di Sistemi Operativi e Reti

Prova scritta di GENNAIO 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)

Un `RunningSushiBuffer` è un particolare tipo di buffer circolare thread-safe con `N` slot in cui è possibile inserire elementi di tipo `T` in posizione 0 ed estrarne da tutte le altre posizioni (da `1` a `N-1`). Inoltre, gli elementi del `RunningSushiBuffer` possono essere arbitrariamente ruotati di `X` posizioni in senso orario (Si veda la figura 1).

I metodi che devono essere implementati sono i seguenti:

`void put(T t)`. Si pone in attesa bloccante se la posizione 0 è occupata. Inserisce l'elemento t nel buffer in posizione 0, non appena la posizione si rende libera.

`T get(int i)`. Si pone in attesa bloccante se la posizione `i` è libera. Rimuove e restituisce l'elemento in posizione `i` quando questo si rende disponibile. Il metodo lancia un'eccezione se `i<1` oppure `i>N-1`, dove `N` è la dimensione del buffer.

`void shift(int j)`. Ruota di `j` posizioni il buffer circolare in senso orario. La rotazione è da intendersi come una rinumerazione degli indici dove ogni elemento accessibile in generica posizione `X` viene passato in posizione `(X+j) mod N`, dove `N` è la dimensione del buffer. Ad esempio se `j=1`, l'elemento in posizione 0 deve passato in posizione 1, l'elemento 1 deve passato in posizione 2, mentre l'elemento N-1 deve passare in posizione `(N-1+1)%N = 0`

`void shift()`. Ruota il buffer di 1 posizione in senso orario, secondo le specifiche di cui sopra.

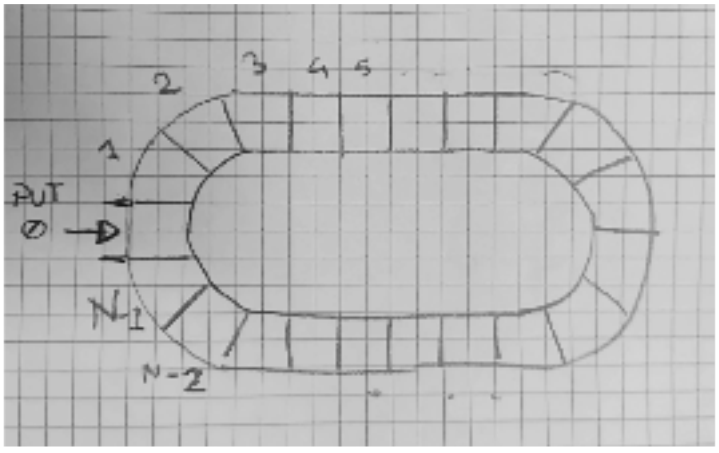

Figura 1

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.

F

Figura 2: Una vera Running Sushi Conveyor Belt. L’immagine è fornita a solo scopo di intuizione. Attenersi alle specifiche date.

NON SPEGNERE IL PC A FINE ESAME

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

Il file `dump.log` contiene alcune informazioni riguardanti le connessioni di rete in entrata e uscita.

Una riga di esempio del file è così composta:

```
TIMESTAMP IP IP_SORGENTE.PORTA_SORGENTE > IP_DESTINAZIONE.PORTA_DESTINAZIONE:
PROTOCOLLO, altri_parametri
```

############################## Esempio Reale ##############################

```
14:25:51.932550 IP 160.97.62.90.62536 > 224.0.0.252.5355: UDP, length 21
14:26:10.171155 IP 23.6.123.119.443 > 192.168.0.100.44664: Flags [.], ack
1, win 990, options [nop,nop,TS val 254276428 ecr 3274039351], length 0
```

Lo scopo dell'esercizio è quello di creare uno script Perl dal nome `dump.pl` che prenderà in input il nome del file di log (in questo caso `dump.log`) e 2 interi positivi S ed F, con S ≤ F ed F < 24.
Lo script dovrà essere quindi richiamato nel seguente modo:

```
./dump.pl dump.log 14 15
```

dove:

Nome File = dump.log
S = 14
F = 15
ed S ≤ F

Lo script dovrà essere in grado di rintracciare tutte le connessioni avvenute tra le ore **S** e le ore **F**.

Esempio:

Nel caso in cui lo script sia invocato con S = 14 ed F = 15, solo le connessioni avvenute tra le 14 e le 15 andranno prese in considerazione e tutte le altre andranno tralasciate

```
14:25:53.325453 IP 192.168.0.100.47172 > 216.58.205.200.443: UDP, length 1350 --> OK
15:34:07.952457 IP 31.13.86.36.443 > 192.168.0.100.54946: Flags [P.], seq 165408:165546, ack
20306, win 425, options [nop,nop,TS val 3121993885 ecr 78163997], length 138 --> OK
16:34:08.136165 IP 192.168.0.100.49736 > 216.58.205.195.443: UDP, length 41 --> NO
```

NON SPEGNERE IL PC A FINE ESAME

e infine produrre in output 2 file di testo che chiameremo **udp.log** e **flags.log** e che rispettino le specifiche riportate di seguito.

Il primo file (udp.log) conterrà il TIMESTAMP seguito da `IP_SORGENTE.PORTA > IP_DESTINAZIONE.PORTA` di tutte le connessioni di rete che avranno usato il protocollo UDP ordinate cronologicamente.
Alla fine del file è necessario stampare il numero totale delle connessioni di cui si è tenuto traccia (vedi esempio).

Esempio del formato file udp.log (Nota l'ordine cronologico per data/ora):

```
14:25:51.932550 --> 160.97.62.90.62536 > 224.0.0.252.5355
14:25:52.895087 --> 172.217.23.106.443 > 192.168.0.100.41997
```

Totale: 2

Il secondo file (flags.log) conterrà il TIMESTAMP seguito da `IP_SORGENTE.PORTA > IP_DESTINAZIONE.PORTA` di tutte le connessioni di rete che avranno usato un protocollo diverso da UDP ordinate in ordine cronologico inverso.
Alla fine del file è necessario stampare il numero totale delle connessioni di cui si è tenuto traccia (vedi esempio).

Esempio del formato file flags.log (Nota l'ordine cronologico per data/ora):

```
14:26:10.171155 --> 23.6.123.119.443 > 192.168.0.100.44664
14:25:53.357586 --> 23.6.123.119.52152 > 192.168.0.100160.97.62.1.443
```

Totale: 2