

Corso di Sistemi Operativi e Reti

Corso di Sistemi Operativi

Prova scritta - Settembre 2022

ISTRUZIONI PER CHI È IN PRESENZA:

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

~~ISTRUZIONI PER CHI SI TROVA ONLINE:~~

- ~~1. Questo file contiene il testo che ti è stato dato ieri, incluso il codice;~~
- ~~2. Mantieni a tutto schermo~~ questo file per tutta la durata della prova; puoi scorrere liberamente tra le sue pagine, ma non puoi cambiare applicazione;
- ~~3. Firma~~ preliminarmente il foglio che userai per la consegna con nome cognome e matricola;
- ~~4. Svolgi~~ il compito; puoi usare solo carta, penna e il tuo cervello;
- ~~5. Aiutati~~ con i numeri di linea per indicare le eventuali modifiche che vorresti fare al codice che ti è stato dato.
- ~~6. Alla scadenza~~ termina *immediatamente* di scrivere, e attendi di essere chiamato, pena l'esclusione dalla prova;
- ~~7. Quando è il tuo turno~~ mostra il foglio ben visibile in webcam, e poi metti una foto dello stesso foglio in una chat privata Microsoft Teams con il prof.

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 nuove strutture dati, o estendendo quelle preesistenti laddove si ritenga necessario, risolvendo eventuali ambiguità. Si può cambiare il codice dei metodi esistenti dove serve.

POSSO CAMBIARE IL PROTOTIPO DEI METODI RICHIESTI O DI QUELLI ESISTENTI? **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. Analogamente, i metodi esistenti possono essere modificati nel loro codice, ma non se ne deve cambiare il risultato finale o il significato.*

CHE LINGUAGGIO POSSO USARE? **PYTHON 3.X**

Il linguaggio da utilizzare per l'implementazione è Python 3.6 o successivo. Ricorda che l'operatore di formattazione `f` (esempio, `f"Ciao sono la stringa {testo}"`) è disponibile solo dalla versione 3.6 di Python in poi, ma può essere sostituito con `"Ciao sono la stringa %s" % testo`

POSSO CONSENTIRE SITUAZIONI DI RACE CONDITION NEL MIO CODICE? **NO**

POSSO CONSENTIRE SITUAZIONI DI DEADLOCK NEL MIO CODICE? **NO**

POSSO CONSENTIRE ALTRE SITUAZIONI DI BLOCCO TOTALE NEL MIO CODICE, TIPO NESTED LOCKOUT, LIVELOCK O ALTRO? **NO**

POSSO CONSENTIRE SITUAZIONI DI STARVATION NEL MIO CODICE? **SI, tranne quando ti viene chiesto esplicitamente di rimuoverle**

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

E' obbligatorio implementare esplicitamente del codice di prova oppure modificare il codice di prova pre-esistente, e accertarsi che giri senza errori prima della consegna.

ESERCIZIO 1 - PROGRAMMAZIONE MULTITHREADED

Punto 1

Arricchisci la classe `TorreInCostruzione` con il metodo `waitForStrato(S : int)`. Tale metodo pone in attesa il thread chiamante fintantoché gli operai non completano lo strato `S`. Se `S` è maggiore dell'altezza finale `H` della torre, arrotonda `S` ad `H`.

Punto 2

Nota che il codice fornito stampa periodicamente l'intero array che rappresenta la Torre. Rimuovi questa stampa e introduci un Thread che, sfruttando il metodo `waitForStrato`, stampi periodicamente, a passi arrotondati al 10%, quanta percentuale della Torre è stata correntemente realizzata. Decidi tu come gestire gli arrotondamenti tra il numero di strati conclusi e la percentuale di completamento. L'output prodotto da questo thread deve essere una sequenza di linee che dicono "10% completato" - "20% completato" - "30% completato" ... ecc.

Punto 3

Introduci il metodo `aggiungiStratoUrgente(s : str)`. Tale metodo può essere invocato durante la costruzione di una Torre, e istanzia due nuovi Operai capaci di aggiungere il carattere `s` (con un ritardo di 100ms tra una posa e l'altra) a una `TorreInCostruzione`. I due nuovi operai dovranno lavorare insieme per aggiungere, non appena sia completo lo strato in corso, un nuovo strato fatto di `s` alla `TorreInCostruzione`, sospendendo la normale alternanza tra Mattonatori e Cementatori. I due nuovi thread dovranno poi terminare al completamento dello strato aggiuntivo, mentre la costruzione della torre deve riprendere normalmente. L'altezza finale della Torre sarà aumentata da `H` ad `H+1`.

SALVA SPESSO

ESERCIZIO 2, TURNO 1 - PERL

Si scriva uno script Perl dal nome `booking.pl` che analizzi la disponibilità delle stanze di un hotel ed estrapoli alcune informazioni. Le disponibilità sono codificate all'interno di alcuni file di testo a loro volta compressi in un archivio in formato `.tar.gz`. Lo script dovrà essere eseguito nel seguente modo

```
./booking.pl PATH
```

Il parametro PATH è obbligatorio e dovrà rappresentare il percorso alla cartella contenente un file chiamato `Book.tar.gz`.

Lo script dovrà estrarre il contenuto del file `Book.tar.gz` e analizzare i file in esso contenuti. Ciascun file è denominato come `mese_anno.txt`. Ad esempio, il file `giugno_22.txt` potrebbe contenere il seguente testo:

L	M	Me	G	V	S	D
01	02	03	04	05	06	07
18	14	18	21	22	15	10
08	09	10	11	12	13	14
7	7	9	14	12	1	5
15	16	17	18	19	20	21
15	6	22	3	7	20	1
22	23	24	25	26	27	28
12	14	1	4	19	0	10
29	30	-	-	-	-	-
17	21	-	-	-	-	-

Mentre il file **luglio_22.txt** potrebbe contenere il testo:

L	M	Me	G	V	S	D
-	-	01	02	03	04	05
-	-	3	7	5	7	19
06	07	08	09	10	11	12
1	7	14	7	21	4	8
13	14	15	16	17	18	19
19	0	7	15	21	16	1
20	21	22	23	24	25	26
22	1	17	1	21	4	15
27	28	29	30	31	-	-
4	15	2	3	21	-	-

La prima riga di ogni file contenente uno schema mensile indica il giorno della settimana: L->Lunedì, M->Martedì, Me->Mercoledì, G->Giovedì, V->Venerdì, S->Sabato, D->Domenica.

Dalla riga successiva si alternano righe che contengono giorni e righe che contengono la disponibilità di camere nel giorno corrispondente: ad esempio, per giorno 1 giugno risultano 18 camere disponibili; giorno 2 giugno risultano 14 camere libere; giorno 3 giugno risultano 18 camere libere; per giorno 16 giugno risultano 6 camere libere e così via.

Una volta avviato, lo script rimane in attesa di richieste da standard input e termina quando viene inserita la parola **END**.

Le possibili richieste formulabili da standard input sono le seguenti:

- **-n mese**

con questa richiesta bisogna produrre in `STDOUT` il numero di camere disponibili per ciascun giorno della settimana per quel mese. La stampa sarà ordinata in ordine crescente di camere e, a parità, verranno ordinati i giorni della settimana in ordine alfabetico inverso.

Esempio:

La richiesta:

-n luglio

produrrà in STDOUT la stringa:

Martedì 23

Sabato 31

Giovedì 33

Mercoledì 43

Domenica 43

Lunedì 46

Venerdì 86

- **-m**

con questa richiesta bisogna produrre un file **log.txt** in cui si indica il numero totale di giorni con 0 camere disponibili secondo i file a disposizione

Esempio:

Nei mesi analizzati, ci sono stati 2 giorni con 0 camere disponibili.