

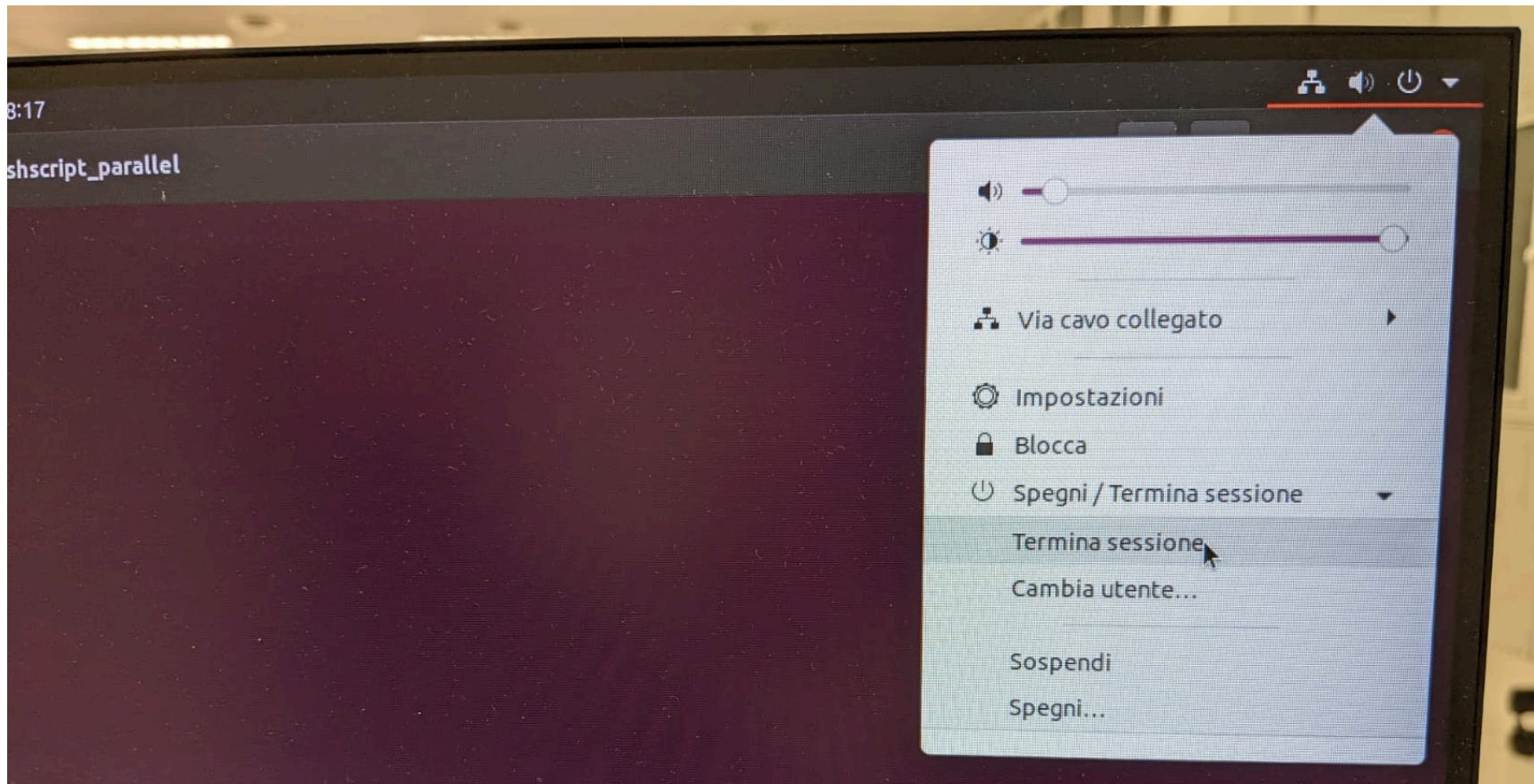
Corso di Sistemi Operativi

Corso estinto di Sistemi Operativi e Reti - Modulo Sistemi Operativi

Prova scritta - Settembre 2024

ISTRUZIONI:

1. **Rinomina** subito 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 senza spazi**; 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 fai **“Termina Sessione/Logout”**, ma lascia la postazione mantenendo il PC acceso.
5. **E’ tua diretta responsabilità** garantire l’integrità del tuo elaborato, anche in caso di assenza di corrente. **Salva spesso il tuo lavoro**



e NON spegnere il PC.

SALVA SPESSO

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

Il codice che scriverai è pensato per essere usato come un modulo di libreria. Un modulo di libreria potrebbe essere usato da altri programmatori, i quali si aspettano di trovare una specifica interfaccia.

Non è quindi consentito modificare il prototipo dei metodi pubblici di una classe se questi sono stati forniti. Puoi aggiungere qualsivoglia campo e metodo privato, 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 l'interfaccia, 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 il codice di prova oppure modificare il codice di prova pre-esistente, e accertarsi che giri senza errori prima della consegna.

ESERCIZIO 1 - PROGRAMMAZIONE MULTITHREADED

(Punteggio minimo richiesto 18/30. Pesa per $\frac{2}{3}$ del voto finale)

Svolgi tutti i punti della traccia in un unico file che chiamerai `SOLUZIONE.py`

Punto 1

Aggiungi un metodo `cambiaVerso` che modifica la direzione di scorrimento del nastro. Inizialmente il nastro scorre in senso orario, ma se qualche thread invoca `cambiaVerso`, ogni successiva operazione di `shift` invocata da qualsiasi altro thread dovrà muovere il nastro in senso antiorario. In generale, ogni invocazione di `cambiaVerso` deve alternare la direzione di scorrimento tra senso antiorario e orario.

Punto 2

Disattiva le stampe che trovi nel codice, e introduci un meccanismo di stampa periodica dello stato del buffer. Puoi scegliere il formato della visualizzazione a tuo piacimento.

Punto 3

Introduci la possibilità di poter riservare delle posizioni addizionali per dei cuochi (non solo la posizione zero). Nelle posizioni riservate a cuochi, non deve essere più possibile invocare `get` o `getList`, ma solo `put` e `putList`. Per esempio l'utilizzatore della classe `RunningSushiBuffer` potrebbe decidere di riservare anche lo slot 5 per le operazioni dei cuochi. Dovrai dunque prevedere la possibilità di aggiungere e togliere posti riservati.

Sei completamente libero di decidere come realizzare questo meccanismo.

Punto 4

Progetta e realizza uno scenario in cui i nastri trasportatori sono due (A e B). A e B girano in direzioni opposte, i cuochi producono piatti che vengono posti sia su A che su B, mentre i clienti devono avere la possibilità di prelevare cibo indifferentemente da uno dei due nastri.

SALVA SPESSO

ESERCIZIO 2 - LINGUAGGI DI SCRIPTING

(Punteggio minimo richiesto 18/30. Pesa per $\frac{1}{3}$ del voto finale)

Scrivi uno script Perl chiamato `check_permissions.pl` che esegua un controllo sui permessi dei file in una directory il cui path viene passato come parametro su linea di comando allo script. Nel dettaglio, lo script dovrà:

1. ricevere come parametro sulla linea di comando il percorso di una cartella C;
2. scansionare il contenuto della cartella C, verificando che
 - per ogni file F contenuto in C:
 - i. il proprietario di F possa leggere e scrivere F;
 - ii. i membri del gruppo di F possano solo leggere F;
 - iii. tutti gli altri utenti possano solo leggere F;
 - per ogni sottocartella D all'interno di C:
 - i. il proprietario di D possa leggere, scrivere ed eseguire
 - ii. i membri del gruppo di D possano leggere ed eseguire
 - iii. tutti gli altri utenti possano leggere ed eseguire
3. Se lo script trova file o directory con permessi non conformi, deve correggerli e riportare l'azione intrapresa in un file di log chiamato `permissions.log` (dovrà cioè scrivere sul file il comando shell necessario ad impostare correttamente i permessi). Qualora l'operazione dovesse sollevare un errore, anche quest'ultimo dovrà essere riportato nel file di log. Il file dovrebbe contenere blocchi di testo strutturati come segue (SOLO per quei file/cartelle che abbiano richiesto una correzione dei permessi):

- `nome_file (OPPURE nome_cartella)`
linea di comando shell usata per correggere i permessi
eventuale errore sollevato dal comando **OPPURE** la stringa `OPERAZIONE OK` qualora non ci fossero errori

Puoi creare una cartella di test lanciando lo script `makeTestFolder.py` che trovi insieme alla traccia. Lo script creerà una cartella con contenuti e attributi casuali all'interno della cartella da cui lancerai lo script