

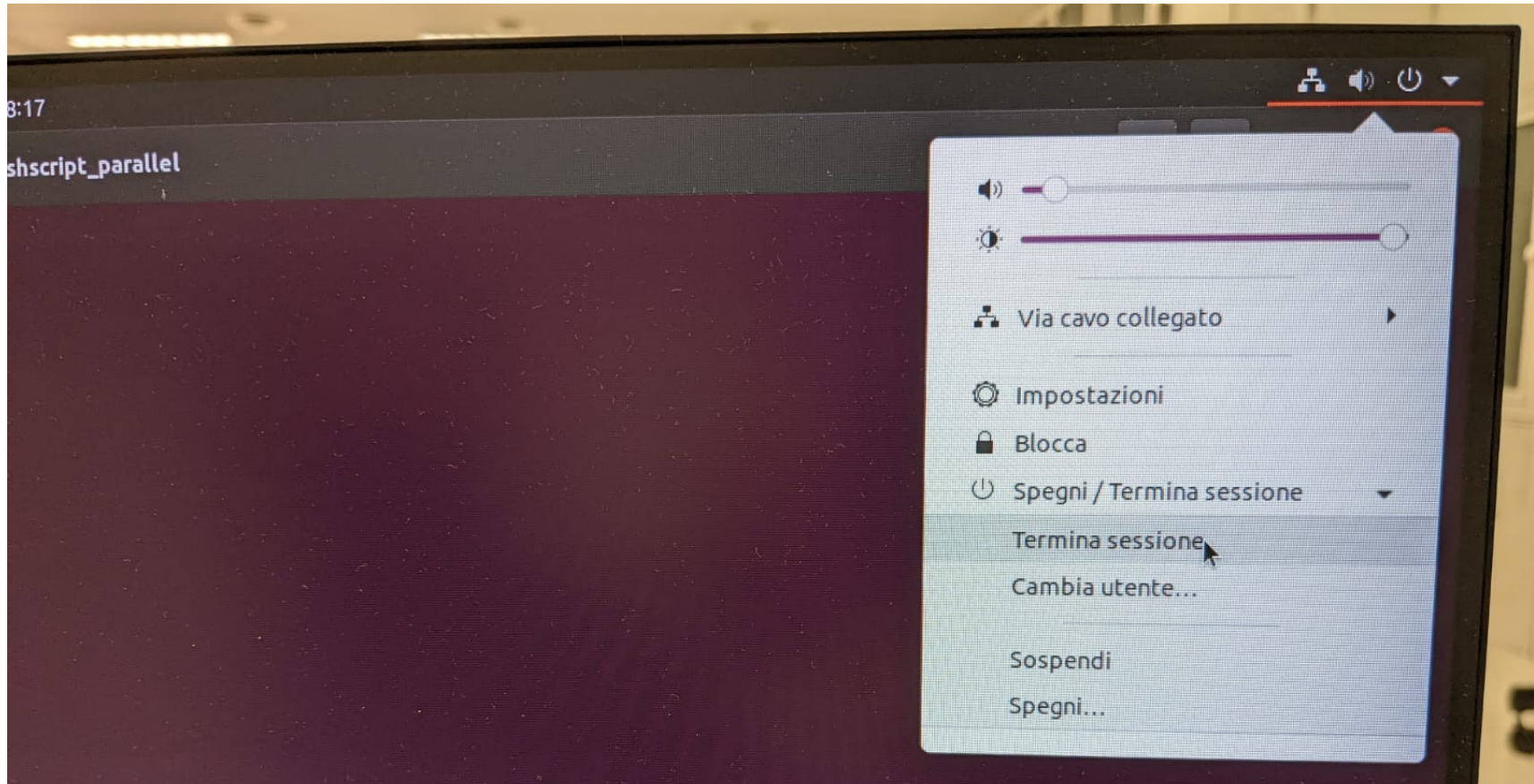
Corso di Sistemi Operativi e Reti

Corso di Sistemi Operativi

Prova scritta - 29 Giugno 2023

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”**, lascia la postazione.
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

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 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)

Punto 1. Implementa il metodo `miglioraPosizione(self, r, c)`. Se esiste un elemento `C` attualmente posizionato sulla colonna `c` in posizione `r`, questo metodo prova a spostare `C` su una delle colonne adiacenti, ma solo qualora su una di tali colonne adiacenti sia conveniente spostarsi (poichè la fila sarebbe più corta per chi si sposta). Se l'elemento predetto non esiste (e cioè la posizione `(r,c)` risulta vuota), il metodo esce senza far nulla.

Punto 2. Introduci la figura dell' "Uomo invisibile al barista". L'Uomo invisibile al barista è un elemento presente in coda che viene sistematicamente ignorato dal barista e che viene servito solo quando ci sono solo uomini invisibili al barista in prima fila. Un elemento su 10, scelto a caso, ha la sfortuna di essere "Invisibile al barista" senza saperlo.

Punto 3. Supponi che un certo elemento `E` sia stato precedentemente inserito usando il metodo `put`, e di voler sapere se infine `E` è stato estratto dal barista. A tal scopo progetta il metodo `attendiServizio(self, E)`. Tale metodo deve comportarsi nel seguente modo: se `E` risulta già accodato sul Bancone, ci si pone in attesa bloccante, finchè `E` non viene estratto, per poi uscire restituendo `True`. Se `E` non risulta già accodato al momento dell'invocazione del metodo `attendiServizio`, bisogna uscire restituendo `False`.

SALVA SPESSO

ESERCIZIO 2 - LINGUAGGI DI SCRIPTING

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

Scrivere una linea di comando shell che restituisca l'identificativo del processo che ha utilizzato per più tempo la CPU (si sconsiglia l'utilizzo del comando *top* a causa della presenza di caratteri speciali che rendono complicato il parsing delle stringhe).

Per la consegna, la linea di comando prodotta dovrà essere scritta su un file chiamato *Soluzione_Esercizio_2*.