

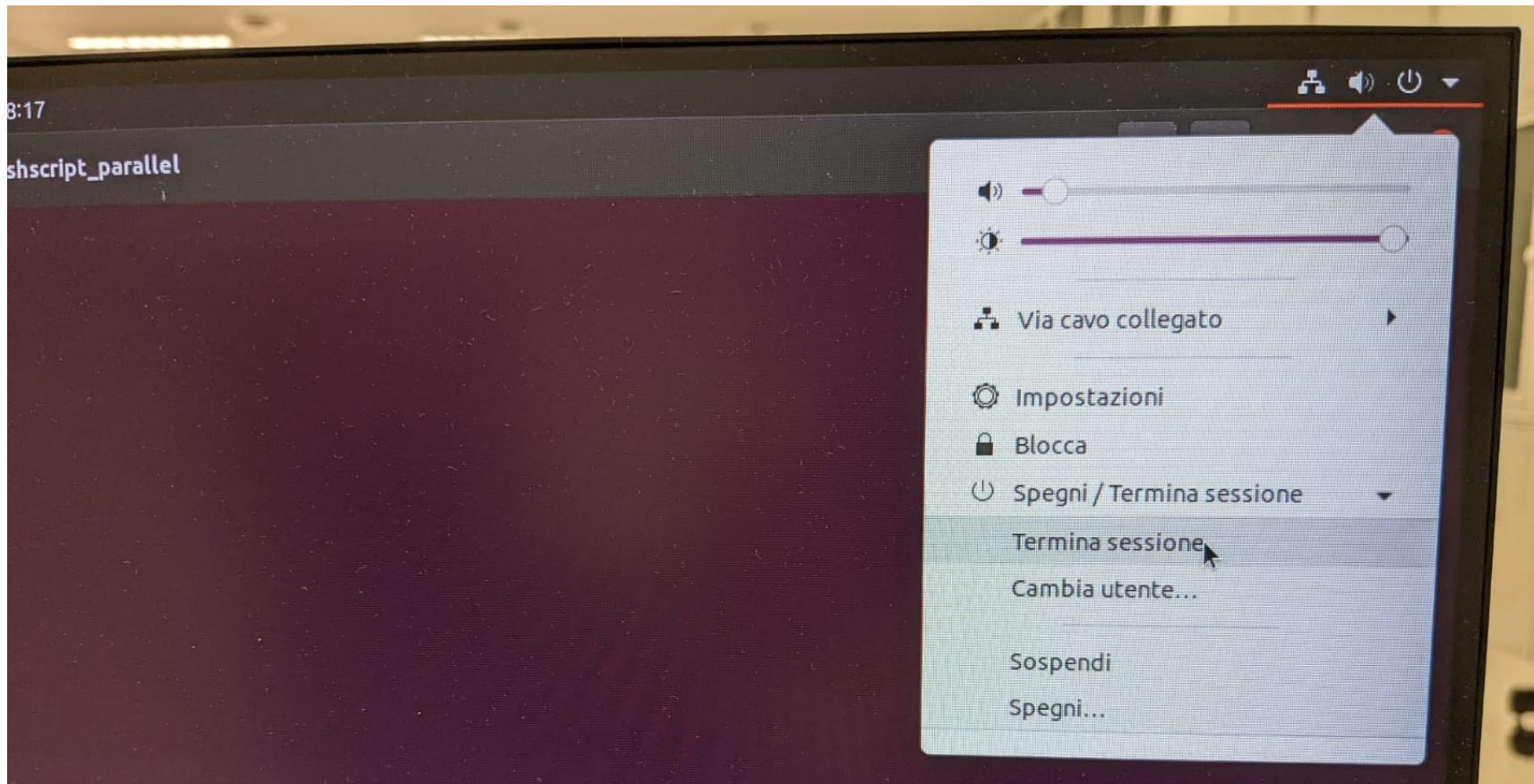
Corso di Sistemi Operativi

Corso estinto di Sistemi Operativi e Reti - Modulo Sistemi Operativi

Prova scritta - Settembre 2025

LEGGI ATTENTAMENTE LE 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? DECIDI TU COSA FARE

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, resolvendo eventuali ambiguità. Si può cambiare il codice dei metodi esistenti dove serve.

POSSO CAMBIARE IL PROTOTIPO DEI METODI RICHIESTI O DI QUELLI ESISTENTI? Sì, ma solo se li lasci retrocompatibili

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.

Se quindi modifichi il prototipo dei metodi pubblici di una classe, devi assicurarti che possano essere comunque utilizzati per come erano stati pensati prima delle tue modifiche. Puoi aggiungere qualsivoglia campo e metodo privato, e qualsivoglia classe ausiliaria. Analogamente, i metodi esistenti possono essere modificati nel loro codice, ma non se ne deve cambiare il risultato finale o il significato a meno che non ti venga richiesto esplicitamente.

CHE LINGUAGGIO POSSO USARE? PYTHON 3.X

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? Sì, tranne quando ti viene chiesto esplicitamente di rimuoverle

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

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

-> Qualsiasi altro nome di file verrà IGNORATO <-

Punto 1

Avrai notato che ogni magazzino prevede un unico lock globale per evitare race condition nell'aggiornamento degli oggetti presenti nel magazzino stesso. Questo impedisce ai thread lavoratori di accedere in parallelo alle varie tipologie di oggetti presenti nel magazzino. Modifica il codice per mitigare questo problema.

Punto 2

Modifica il codice di test in maniera da avere almeno 4 magazzini. Introduci poi il thread di tipo Fattorino. Un thread di tipo fattorino ripete il seguente ciclo di attività: *a)* seleziona un tipo di oggetto a caso T; *b)* individua il magazzino M che possiede più item di tipo T; *c)* preleva un oggetto di tipo T da M e lo sposta in un magazzino a caso. Se ritieni che il codice di Magazzino non sia adeguato a risolvere il punto 2, puoi estenderlo (garantendo la retro compatibilità).

Punto 3

Introduci la tipologia di thread Generatore. Un generatore G, ogni S secondi, aggiunge 5 item di tipo T al magazzino a cui è associato. S e T sono scelte in fase di costruzione di G.

SALVA SPESSO

ESERCIZIO 2 - LINGUAGGI DI SCRIPTING

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

NOTA BENE: In questo esercizio dovrei sistemare errori nello script che trovi nella cartella della traccia e non implementarne uno da zero. Leggi la traccia fino in fondo prima di procedere.

Lo script `settembre-25.pl` allegato dovrebbe fornire informazioni sull'utente che lo ha eseguito, producendo dunque in output:

- a) Il nome dell'utente che ha eseguito lo script.
- b) L'elenco dei PID di *tutti i processi in esecuzione* associati all'utente corrente.
- c) Lo spazio totale occupato su disco dalla home dell'utente, includendo in modo ricorsivo le sottocartelle. Lo spazio occupato dovrebbe essere espresso in forma "leggibile" (e.g., 1K 45M 21G).

Esempio:

Se lo script viene eseguito dall'utente `pinco`, l'output potrebbe essere il seguente:

punto a:

pinco

punto b:

516

575

576

punto c:

10.4G /home/pinco

Lo script allegato, però, presenta errori di sintassi e/o logici. Sistemalo per ottenere il comportamento sopra descritto. ***NON devi stravolgere il contenuto dello script: la sua struttura deve rimanere pressoché invariata.***