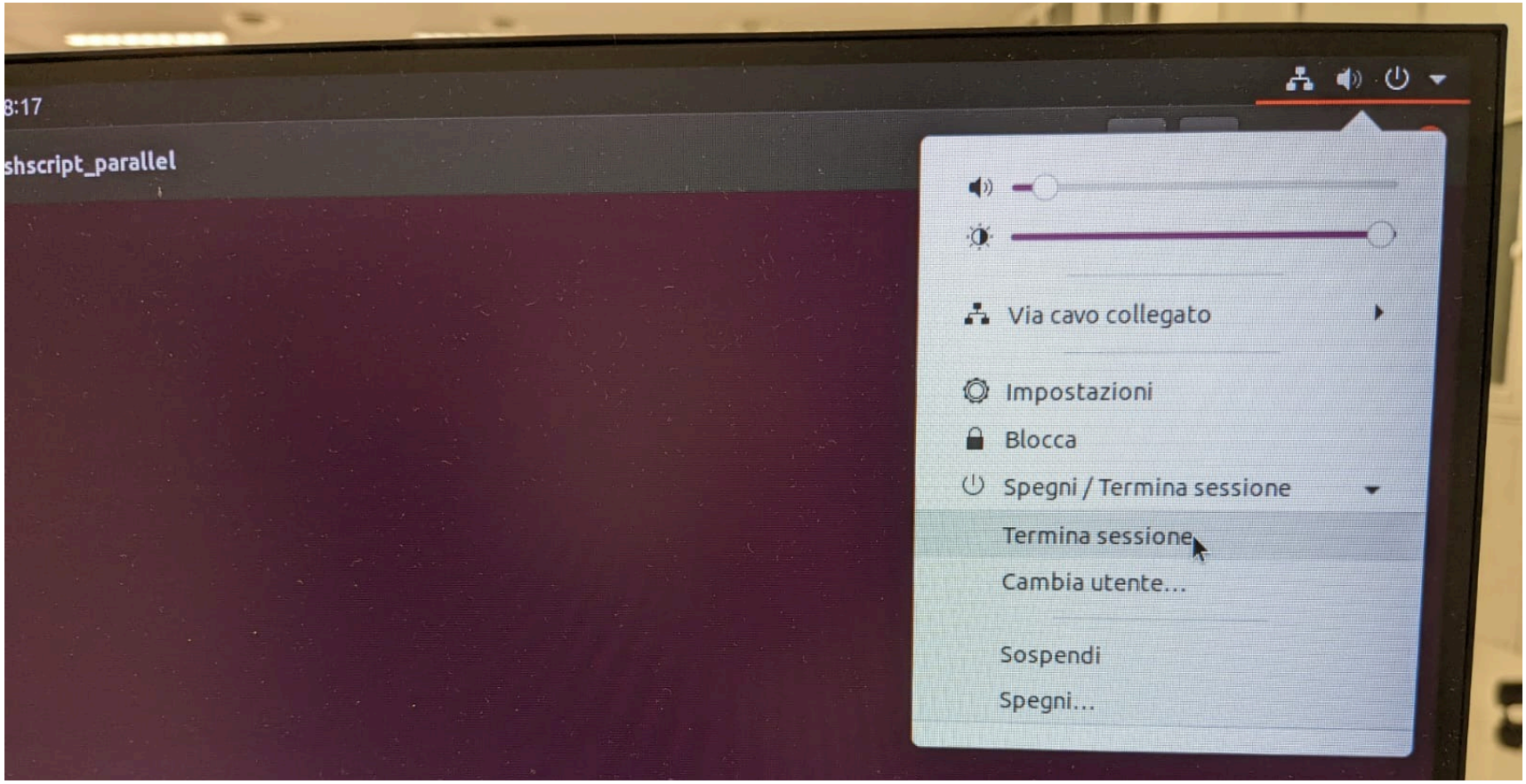
Corso di Sistemi Operativi

Corso estinto di Sistemi Operativi e Reti - Modulo Sistemi Operativi

Prova scritta - Luglio 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, risolvendo eventuali ambiguità. Si può cambiare il codice dei metodi esistenti dove serve.

POSSO CAMBIARE IL PROTOTIPO DEI METODI RICHIESTI O DI QUELLI ESISTENTI? Si, 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? 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 ⅔ 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.
Utilizza il FairLock per implementare il seguente scenario: 3 contatori il cui accesso a ciascuno dei quali, sia per le operazioni di incremento che per quelle di decremento, è da proteggere con un proprio rispettivo FairLock; 10 thread che accedono per leggere e scrivere i tre contatori in maniera del tutto casuale e in tempi casuali. Questo scenario va implementato definendo una classe che chiamerai `Scenario_semplice`. Il costruttore di tale classe dovrà creare i 3 contatori e avviare i thread che lavorano su di essi.

Punto 2.
Utilizza il FairLock con gestione della starvation per implementare il seguente scenario: 1 Fairlock posto a protezione di un numero intero; 40 thread che accedono per leggere e scrivere il numero intero in maniera del tutto casuale e in tempi casuali; alcuni di questi thread usano l'accesso urgente al lock. Definisci una classe che chiamerai `Scenario_intermedio`. Il costruttore di tale classe dovrà creare il numero intero e avviare i thread che lavorano su di esso. Un thread S di gestione della starvation modifica lo starvation control cercando di adattarlo alla situazione attuale del fairlock. Puoi progettare S come preferisci ed aggiungere a FairLock tutte le estensioni che ritieni necessarie.

Punto 3.
Modifica FairLock trasformandolo in un PriorityLock: ogni thread ha priorità di accesso al lock in proporzione

inversa al numero di accessi precedenti. In generale, un thread che ha acceduto *n* volte, deve essere preferito a un thread che ha acceduto *m* volte, se $n < m$. Ad esempio, un thread che non ha mai usato un certo FairLock deve essere preferito rispetto ai thread che hanno acceduto tre volte, mentre i thread che hanno acceduto tre volte devono essere prioritizzati rispetto a quelli che hanno acceduto attualmente cinque volte.

SALVA SPESSO

ESERCIZIO 2 - LINGUAGGI DI SCRIPTING

(Punteggio minimo richiesto 18/30. Pesa per ⅓ 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 `luglio-25.pl` allegato dovrebbe ricevere tre argomenti da linea di comando:

- il nome di un comando Unix `CM`
- una stringa di ricerca `S`
- un numero intero `I`

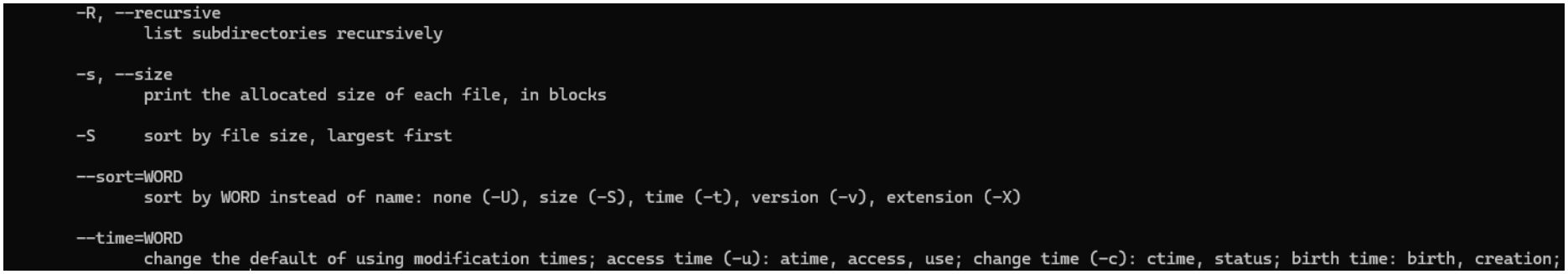
e dovrebbe eseguire le seguenti operazioni:

- estrarre la documentazione relativa al comando specificato.
- cercare la stringa di ricerca `S` nella documentazione estraendo, per ogni occorrenza, anche `I` righe prima e dopo la stringa trovata, ottenendo un certo blocco di testo.
- identificare, tra tutti i blocchi trovati, il blocco più lungo (cioè quello con il maggior numero di righe consecutive).
- stampare a schermo solo questo blocco. In caso di più blocchi con la stessa lunghezza massima, lo script dovrebbe stampare il primo tra questi.

Ad esempio, se invocato come

```
perl luglio-25.pl ls size 3
```

lo script dovrebbe stampare



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.***