

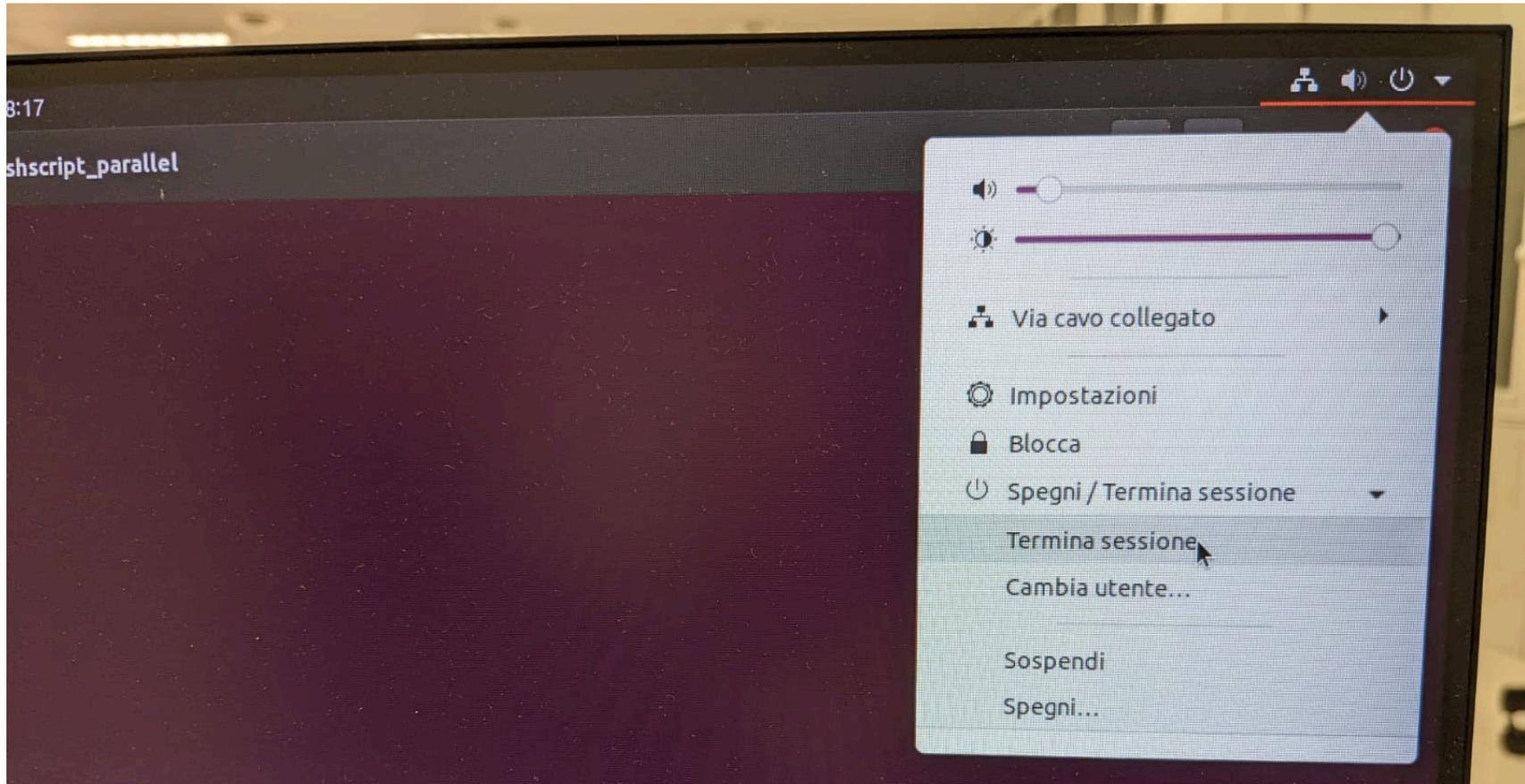
=Corso di Sistemi Operativi

Corso estinto di Sistemi Operativi e Reti - Modulo Sistemi Operativi

Prova scritta - Marzo 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? 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

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

-> Qualsiasi altro nome di file verrà IGNORATO <-

Punto 1.

Introduci il concetto di “accesso tamarro”. Quando si richiede una pista in modalità “tamarra”, la squadra richiedente viene sempre accodata **come la prima** della lista di attesa. Prevedi tuttavia un meccanismo a tua scelta che consenta alle squadre standard di poter giocare allorché restano in attesa troppo a lungo per via di continui accessi di squadre tamarre. Dovranno dunque esserci tre modalità di accesso alla pista: le due modalità preesistenti, accesso gentile e accesso standard; e la nuova modalità di accesso, detta appunto “modalità tamarra”.

Punto 2.

Implementa un meccanismo di visualizzazione periodica dello stato di occupazione delle piste, secondo una strategia a tua scelta.

Punto 3.

Introduci la possibilità di condividere una pista tra due squadre. Due squadre possono condividere la stessa pista (posto che siano disponibili palle a sufficienza per tutti i giocatori), se entrambe hanno indicato, attraverso un apposito parametro, di dare il consenso alla condivisione di piste. Sicuramente avrai vari dubbi su come realizzare questo punto: dovrai essere tu, **in autonomia**, a scegliere per il meglio.

SALVA SPESSO

ESERCIZIO 2 - LINGUAGGI DI SCRIPTING

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

Scrivere uno script Perl chiamato **analizza_file.pl** che si comporti come segue:

1. Stampi su **STDOUT** i nomi di tutti i file **di testo** (`.txt`) presenti nella cartella corrente che contengano almeno 20 righe.
2. Stampi su **STDOUT** il nome del file di testo (`.txt`) più grande presente nella cartella corrente, indicando anche il numero totale di righe e di parole contenute al suo interno.
3. Crei un file chiamato `righe_random.txt` contenente **10 righe scelte casualmente** da tutti i file `.txt` presenti nella cartella corrente. Se ci sono meno di 10 righe totali tra tutti i file, deve scrivere tutte le righe disponibili.

Per il punto 3, prendi in considerazione di usare il comando `shuf`