

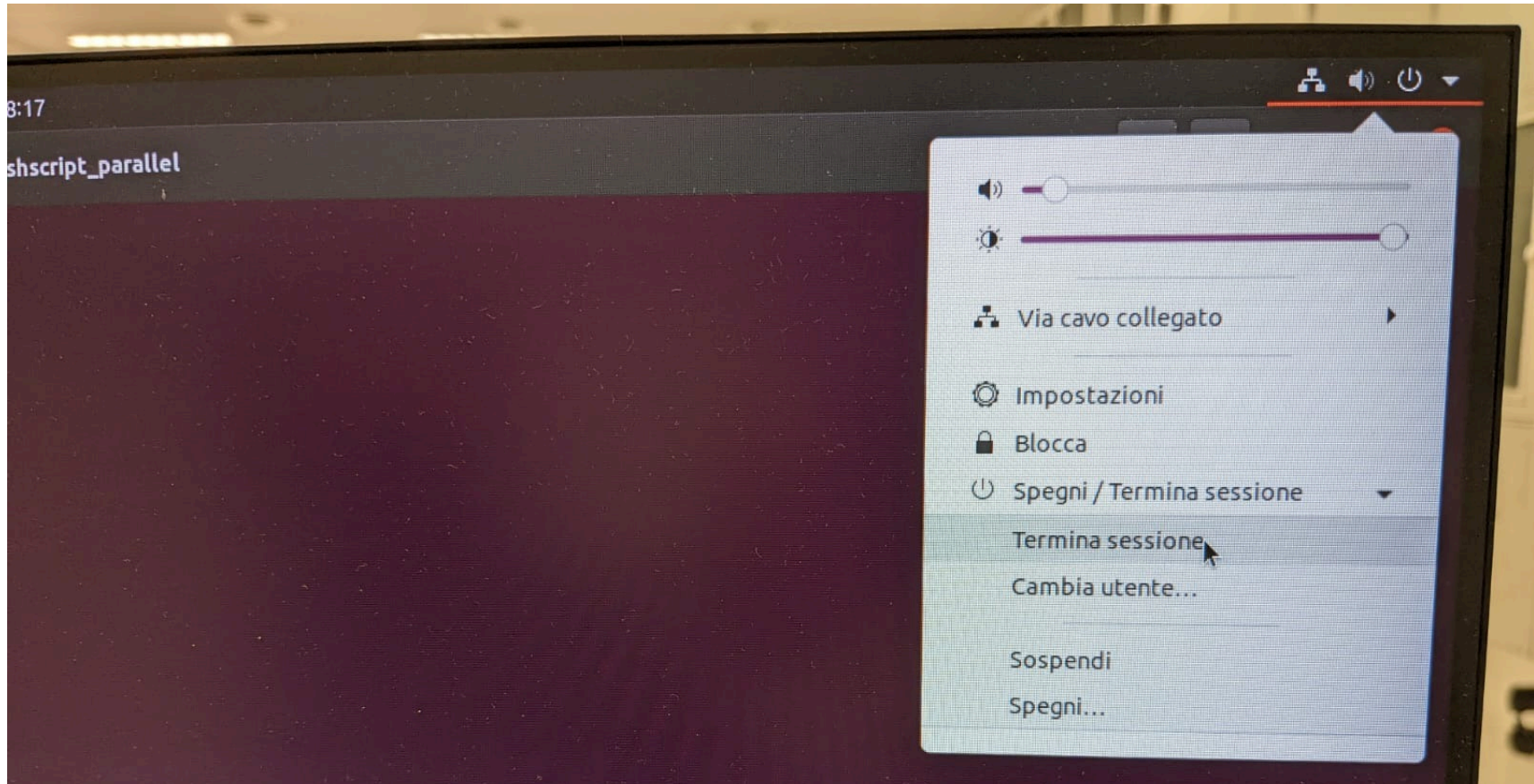
# Corso di Sistemi Operativi e Reti

## Corso di Sistemi Operativi

Prova scritta - Marzo 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**

*Non è consentito modificare il prototipo dei metodi pubblici di una classe se questi sono stati forniti. Potete 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)

Estendi il meccanismo del gioco delle sedie in almeno due dei modi che ti propongo:

-Un “Campionato” è costituito da 5 round del gioco delle sedie. Il partecipante che perde in un round non può partecipare al successivo. Nel gioco ci sono almeno 6 partecipanti. Scrivi tutto il codice necessario a gestire un intero Campionato e dichiararne il vincitore;

-Ogni partita del gioco delle sedie dura T secondi. La durata T della partita deve poter essere scelta prima dell’avvio della partita; ogni partecipante può cambiare sedia nell’arco della durata della partita (ma è vietato occupare due sedie contemporaneamente). A ogni sedia corrisponde un punteggio pari al proprio indice: ad esempio la sedia 1 vale 1 punto, la sedia 5 vale 5 punti, ecc. Vince il partecipante che si siede sulla sedia di punteggio maggiore.

-Le partite sono a squadre di tre partecipanti. La squadra che si ritrova con un componente che non è riuscito a sedersi è quella che perde complessivamente.

Questo è tutto. Come sempre, è tuo specifico compito decidere cosa fare per tutti i dettagli che non sono stati espressamente definiti. Ricorda che tutti metodi richiesti sono da intendersi pubblici e devono essere tutti thread-safe.

# SALVA SPESSO

## ESERCIZIO 2 - LINGUAGGI DI SCRIPTING

(Punteggio minimo richiesto 18/30. Pesa per 1/3 del voto finale)

Si scriva uno script chiamato `gruppo_utente.pl` che riceve come parametro da linea di comando una stringa, e che verifica che tale stringa corrisponda al nome di un utente; in caso affermativo, lo script deve stampare su STDOUT gli id e il nome dei gruppi di appartenenza dell'utente. Se la stringa non corrisponde ad un nome utente, lo script stamperà "Utente non trovato".

Esempio.

Se lo script fosse invocato come

```
./gruppo_utente.pl denise
```

e `denise` corrispondesse ad un nome utente, appartenente ai gruppi chiamati `denise`, `sudo`, `admin`, corrispondenti rispettivamente agli id `100`, `0`, `23`, l'output atteso deve essere

```
GID  Nome Gruppo
100  denise
0    sudo
23   admin
```

N.B. si consiglia l'utilizzo del comando `groups` e del file `/etc/group`.