

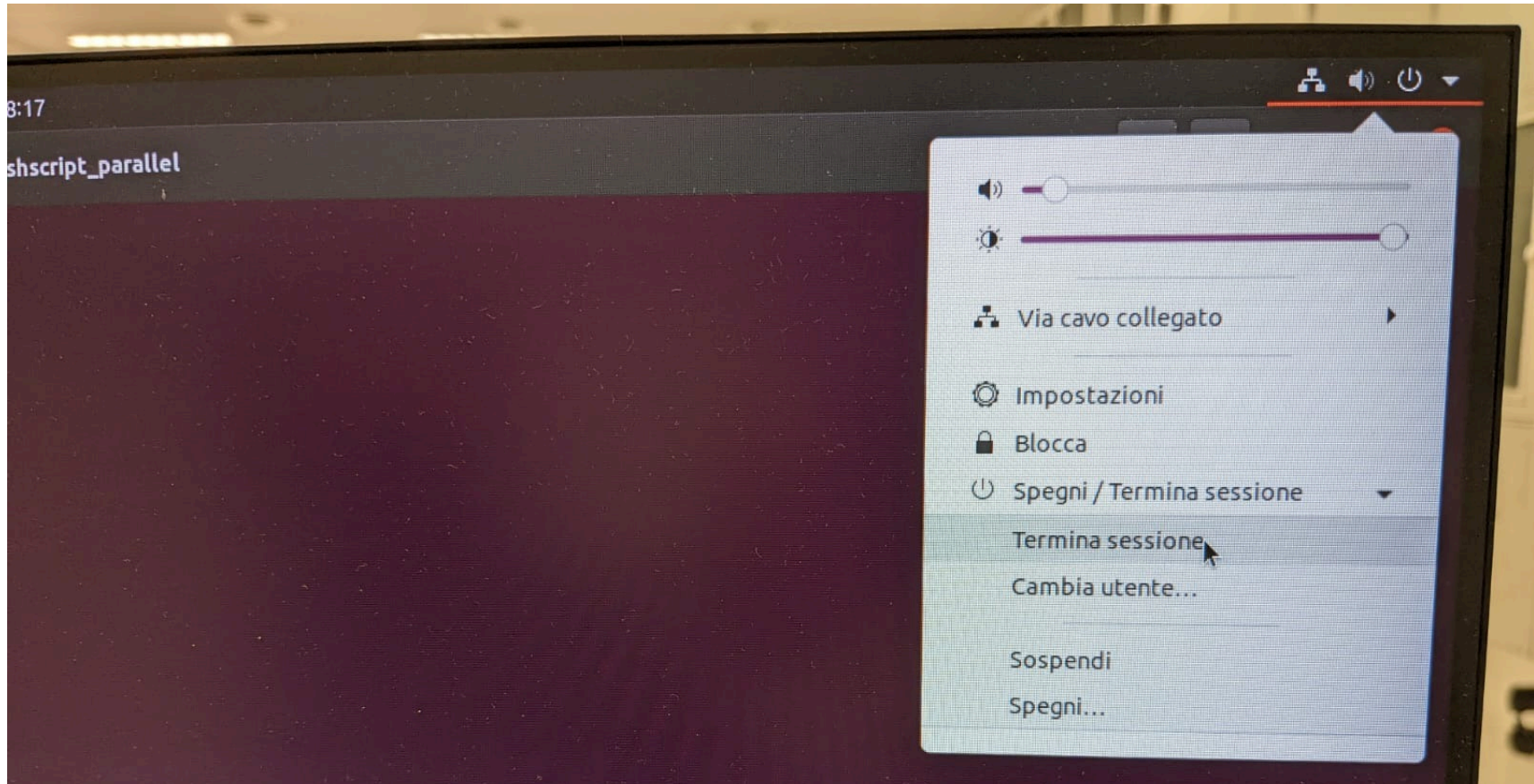
# Corso di Sistemi Operativi e Reti

## Corso di Sistemi Operativi

Prova scritta riservata agli studenti fuori corso - 21 Marzo 2023

### ISTRUZIONI PER CHI È IN PRESENZA:

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



**e NON spegnere il PC.**

**SALVA SPESSO**

## ~~ISTRUZIONI PER CHI SI TROVA ONLINE:~~

- ~~1. Questo file contiene il testo che ti è stato dato ieri, incluso il codice;~~
- ~~2. Mantieni a tutto schermo~~ questo file per tutta la durata della prova; puoi scorrere liberamente tra le sue pagine, ma non puoi cambiare applicazione;
- ~~3. Firma~~ preliminarmente il foglio che userai per la consegna con nome cognome e matricola;
- ~~4. Svolgi~~ il compito; puoi usare solo carta, penna e il tuo cervello;
- ~~5. Aiutati~~ con i numeri di linea per indicare le eventuali modifiche che vorresti fare al codice che ti è stato dato.
- ~~6. Alla scadenza~~ termina *immediatamente* di scrivere, e attendi di essere chiamato, pena l'esclusione dalla prova;
- ~~7. Quando è il tuo turno~~ mostra il foglio ben visibile in webcam, e poi metti una foto dello stesso foglio in una chat privata Microsoft Teams con il prof.

**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.** Fattorizza il codice esistente introducendo una classe `Sala` che possiede al suo interno una matrice quadrata di `PostoSafe`. Ricorda che “Fattorizzare” = “Riorganizzare”. Il numero di file iniziale deve essere passato via costruttore. La classe `Sala` deve offrire il metodo pubblico

`self.testaEoccupa(i, j)`. Dato il posto di coordinate `(i, j)`, occupa il posto in tali coordinate se questo è libero. Restituisce `False` in caso di fallimento, e `True` altrimenti.

**Punto 2.** Aggiungi alla classe sviluppata nel punto precedente il metodo `self.trovaPosto()`. Cerca di occupare un posto qualsiasi tra quelli rimasti liberi. Restituisce la coppia `(i, j)` dove `(i, j)` sono le coordinate del posto trovato. Se la sala è piena, restituisce la coppia `(-1, -1)`. Da notare che “qualsiasi” non vuol dire necessariamente “casuale”. scegli tu la strategia più adeguata per trovare un posto libero.

**Punto 3.** Sposta il codice che stampa il contenuto della sala all’interno della classe `Sala`; rendi questo codice thread-safe; fai in modo che il thread `Display` si attivi stampando il contenuto della Sala solo quando c’è una effettiva modifica;

**Punto 4.** Fai in modo che il Thread `Display` termini quando tutti gli altri thread `PartecipanteSafe` hanno terminato.

*Per tutti i punti: I thread che prima gestivano direttamente l’occupazione dei posti e la visualizzazione devono adesso avvalersi dei nuovi metodi introdotti nella classe `Sala`.*

# SALVA SPESSO

## ESERCIZIO 2 - LINGUAGGI DI SCRIPTING

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

Scrivi uno script `folder_stats.pl` che si occupi di ricavare alcune proprietà di una data cartella. In particolare, lo script deve poter essere invocato nel seguente modo:

```
./folder_stat.pl PATH
```

dove `PATH` è obbligatorio ed è il path ad una **cartella**.

Lo script dovrà stampare su STDOUT:

1. il numero e il nome delle **sottocartelle** della cartella indicata da `PATH`
2. la dimensione ed il nome del **file** più grande

A questo punto lo script rimarrà in attesa di ricevere in input dall'utente il nome di una sottocartella o del file più voluminoso. Stampa una stringa di errore se il nome indicato non è nell'elenco stampato in precedenza. Se si riceve in input:

- il nome del file, lo script stamperà su STDOUT la data di ultima modifica
- il nome di una delle sottocartelle, lo script stamperà su STDOUT il contenuto della sottocartella stessa

### Esempio:

- Se lo script fosse invocato come

```
./folder_stat.pl /usr
```

e il contenuto di `/usr` fosse il seguente

```
drwxr-xr-x  2 root root 36864 Mar 13 11:51 bin
drwxr-xr-x  2 root root  4096 Apr 15  2020 games
drwxr-xr-x 36 root root  4096 May 10  2022 include
drwxr-xr-x 82 root root  4096 May 10  2022 lib
-rw-r--r-- 1 angilica angilica  3769 Jul 18  2022 file.txt
-rw-r--r-- 1 angilica angilica 1677957 May 10  2022 output
```

Lo script dovrebbe stampare

```
4
bin
games
include
lib
output: 1677957
```

- se si ricevesse a questo punto in input `bin`  
bisognerebbe stampare il nome delle cartelle e dei file contenuti in `/usr/bin`;  
se si ricevesse invece in input `output`

bisognerebbe stampare May 10 2022