

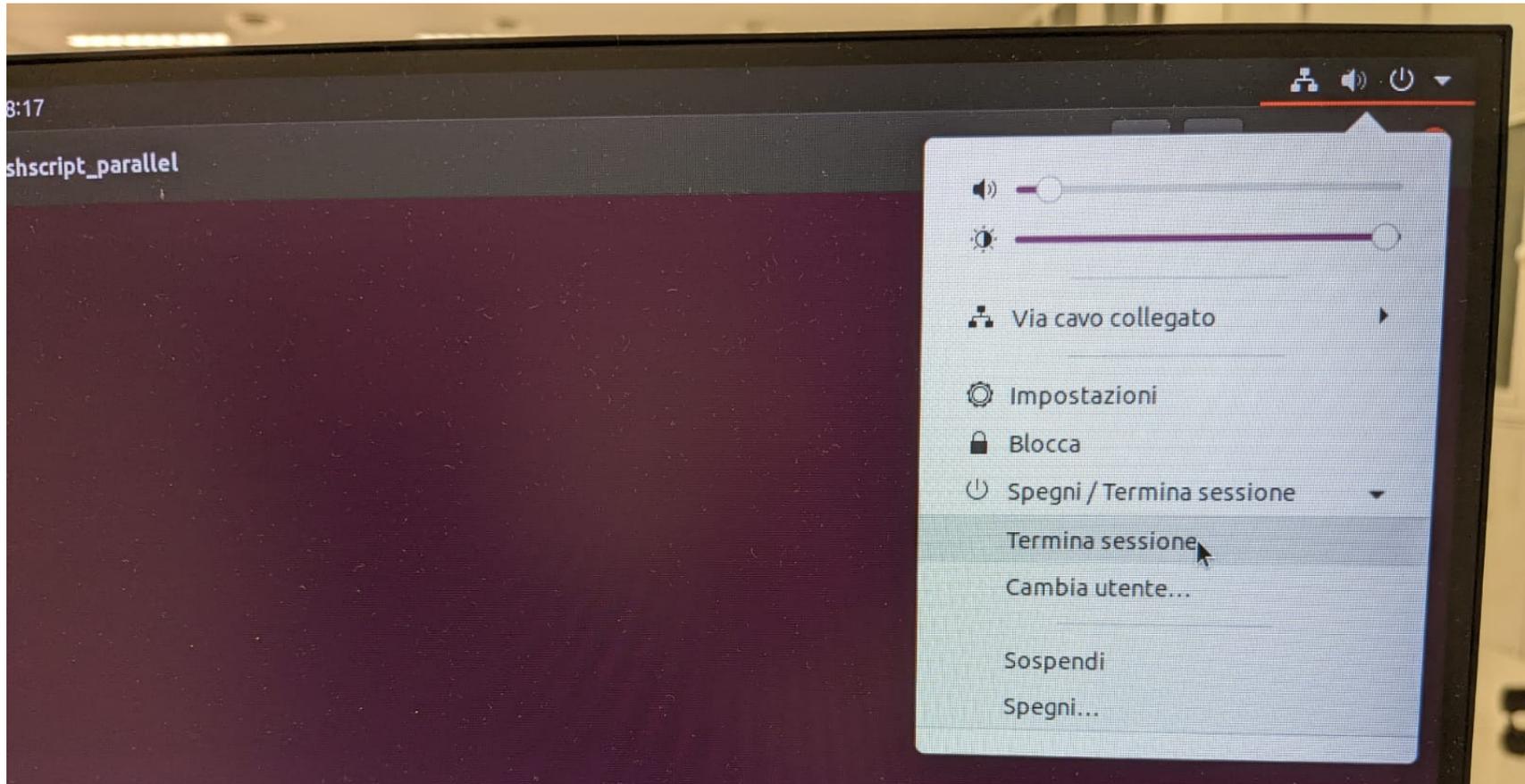
# Corso di Sistemi Operativi e Reti

## Corso di Sistemi OPERativi

Prova scritta - 18 Luglio 2023

### 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”**, lascia la postazione
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 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.** Implementa il metodo `imposta_colonna_prioritaria(i)`. Questo metodo specifica una colonna `i`. Dal momento in cui tale metodo viene invocato, l'ordine di estrazione dal bancone bar presente nell'implementazione di `get` deve privilegiare la colonna `i` nell'ordine di estrazione degli elementi. Quando `i` è completamente vuota va applicata la strategia di estrazione pre-esistente. Se il valore di `i` è fuori dall'intervallo dei valori ammissibili, la strategia di estrazione deve tornare a essere quella pre-esistente.

**Punto 2.** Implementa la funzione `attendi_invisibile(self)`. Il thread che invoca questo metodo va in attesa bloccante fintantochè non viene estratto un elemento invisibile.

**Punto 3.** Progetta la classe `bancone_combinato`. Tale classe incapsula due istanze di `banconebar` B1 e B2 e un thread S. Il costruttore della classe inizializza B1, B2 e crea ed avvia S. I metodi pubblici di tale classe devono essere gli stessi di `bancone bar`, ma così ridefiniti:

-`get`: estrae un elemento da B1;

-`put`: inserisce un elemento in B2;

-`attendiElemento`: attende che un elemento sia complessivamente estratto dal `bancone_combinato`;

Il thread S deve prelevare periodicamente elementi da B2 depositarli in B1.

Non è necessario implementare `printBancone` e `miglioraPosizione`.

**Punto 4.** Scrivi del codice multi-threaded che testi tutti i metodi di cui sopra. Assicurati che il tuo codice non si inceppi per via di errori di sintassi o altri problemi tecnici.

# SALVA SPESSE

## ESERCIZIO 2 - LINGUAGGI DI SCRIPTING

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

Scrivi uno script perl dal nome `cpu_balance.pl`. Lo script dovrà:

- individuare la `cpu` con minor carico di lavoro attuale (a parità di carico, preferisci la `cpu` con numero identificativo più alto: `cpu12` è da preferire a `cpu3`).
- individuare il pid del processo `yes`
- impostare l'affinità (affinity) del suddetto `pid` alla suddetta `cpu`