

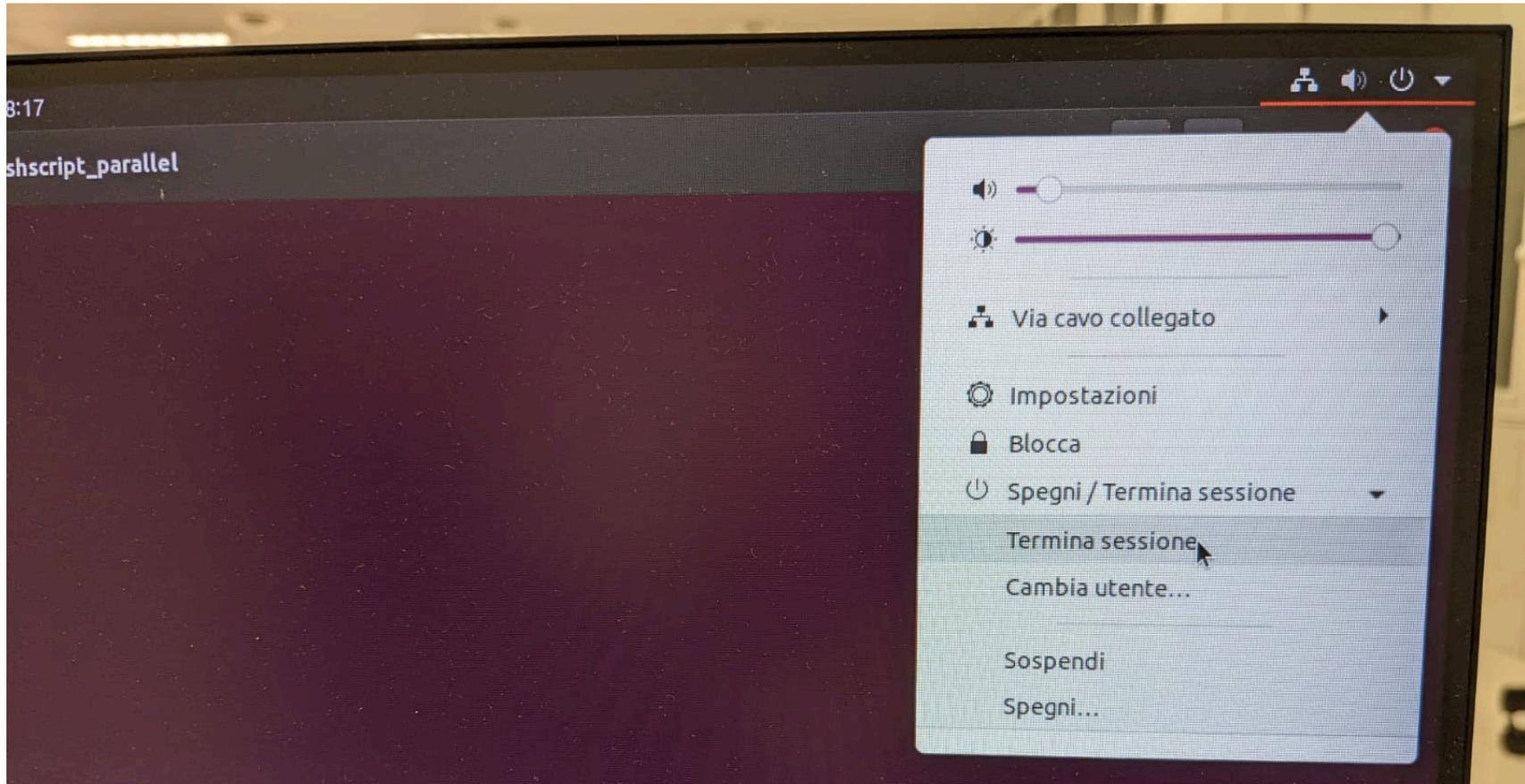
Corso di Sistemi Operativi e Reti

Corso di Sistemi Operativi

Prova scritta - Gennaio 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 fornito. 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)

1. Implementa un meccanismo di timeout per le operazioni in `Calcolatrice`. Se una certa operazione non viene estratta ed elaborata entro un certo tempo, questa viene annullata attraverso l'eliminazione dell'operazione in B. Il timeout di una operazione viene fissato in fase di inserimento.
2. Inventi un meccanismo per arrestare in maniera pulita una istanza di `Calcolatrice` in esecuzione, ad esempio invocando un apposito metodo `stop()`. Tale metodo deve arrestare in maniera pulita e thread-safe ogni thread che compone la calcolatrice.
3. Arricchisci la classe `InteriCombinati` con tutto ciò che serve per tenere traccia del numero di operazioni effettuate per ciascun tipo di operazione e per ciascun elemento dell'array V , a partire dal momento della creazione dell'istanza di classe. Prevedi un apposito metodo che stampi a video le statistiche attuali, e un metodo per azzerare le statistiche.

Questo è tutto. Come sempre, è tuo specifico compito decidere cosa fare per tutti i dettagli che non sono stati espressamente definiti.

SALVA SPESSO

ESERCIZIO 2 - LINGUAGGI DI SCRIPTING

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

Scrivi uno script in Perl chiamato `user_stat.pl`. Lo script deve fornire informazioni sull'utente che lo ha eseguito, producendo dunque in output::

- a) Il nome dell'utente che ha eseguito lo script.
- b) L'elenco dei PID dei processi in esecuzione associati all'utente corrente.
- c) Lo spazio totale occupato su disco dalla home dell'utente, includendo in modo ricorsivo le sottocartelle. Utilizza il comando `du` per ottenere questa informazione.

Esempio:

Se lo script viene eseguito dall'utente `pinco`, l'output potrebbe essere il seguente:

punto a:

```
pinco
```

punto b:

```
516
```

```
575
```

```
576
```

punto c:

```
10.4G /home/pinco
```