

Prova pratica del 23/02/2022
Durata della prova: 75 minuti

Lo studente completerà il programma a corredo di questo documento, seguendo le seguenti indicazioni.

La prova sarà valutata come segue:

- **A:** Prova svolta correttamente.
- **B:** Il programma non esegue correttamente, con errori minori di programmazione o di concorrenza.
- **C:** Il programma non esegue correttamente, con errori significativi (voto max: 22).
- **INSUFFICIENTE:** Il programma non compila o non esegue, con errori gravi di sincronizzazione.

Istruzioni per la consegna dell'elaborato

L'elaborato dovrà essere svolto in una cartella dal nome: Cognome Nome Matricola Docente

Esempio:

```
mkdir Rossi_Mario_N46012345_Cotroneo
```

```
cd Rossi_Mario_N46012345_Cotroneo
```

.... copiare nella cartella i file forniti per l'esame

.... sviluppare il programma

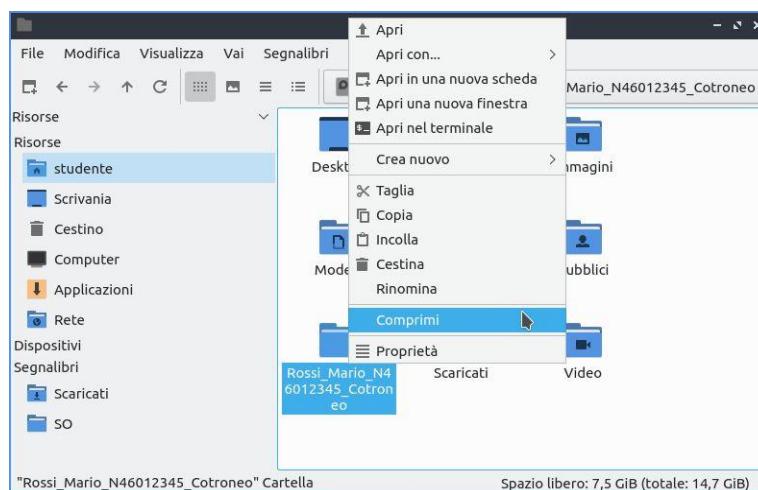
.... per la consegna, dalla shell (assumendo di essere ancora nella cartella di lavoro),

creare un file compresso ("tar") con i seguenti comandi:

```
cd ..
```

```
tar -cvzf ./Rossi_Mario_N46012345_Cotroneo.tar.gz ./Rossi_Mario_N46012345_Cotroneo
```

In alternativa, è consentito creare il file compresso "tar" tramite l'interfaccia grafica.

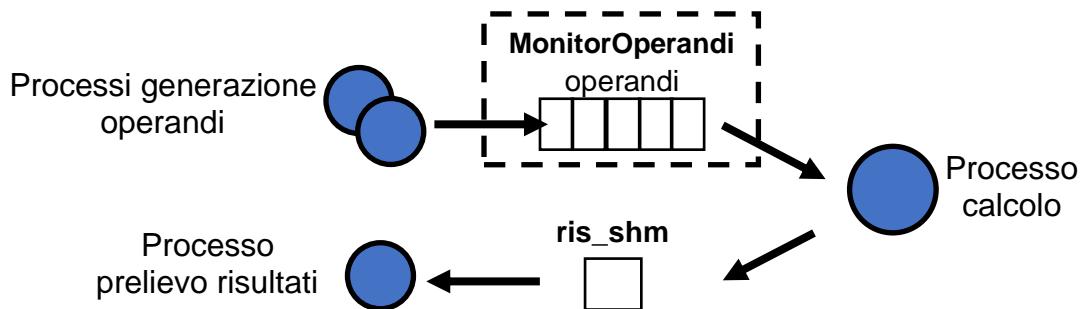


All'interno della macchina virtuale, aprire il browser all'indirizzo specificato dal docente e scaricare i file da completare.

Alla consegna, attendere una notifica del docente.

Testo della prova

Si realizzi in linguaggio C un'applicazione **multiprocesso** (vedi figura) per parallelizzare dei calcoli statistici, basata su costrutto **monitor signal and wait** e **semafori** e sullo **schema produttore-consumatore**.



In particolare:

- I **processi generazione operandi** (2 in totale), invocano ognuno 6 volte il metodo `inserisci_operando(...)` passando dei valori casuali tra 1 e 100. Il valore andrà inserito in un pool di buffer di **dimensione pari a 5**. Se tale pool risulta essere già pieno, il processo chiamante viene posto in attesa.
- Il **processo di calcolo** dovrà prelevare **3 operandi** dal pool di buffer invocando il metodo `preleva_operandi(...)` (4 chiamate, per un totale di 12 prelievi). Tale metodo ritorna **tre valori da estrarre** (un array di 3 elementi) dal pool di buffer `operandi[]`. Se tale pool non contiene almeno 3 elementi, il processo chiamante viene posto in attesa. La sincronizzazione per l'accesso al pool di buffer deve essere disciplinato attraverso una **coda circolare**.

Dopo aver prelevato i 3 operandi, il processo di *calcolo* deve **effettuare il calcolo** della media, minimo, e massimo dei valori prelevati, attendere 2 secondi, ed inserire il risultato in una memoria condivisa (**ris_shm**) attraverso il metodo `inserisci_risultato(...)` (4 chiamate, per un totale di 4 produzioni). L'area di memoria condivisa deve essere disciplinata attraverso l'uso di semafori.

- Il **processo di prelievo del risultato** dovrà prelevare il risultato delle statistiche dalla memoria condivisa **ris_shm**, chiamando il metodo `preleva_risultato(...)` (4 volte in totale). Per ogni prelievo il processo dovrà stampare le statistiche a video.