

Università degli Studi di Napoli Federico II  
Corso di Laurea in Ingegneria Informatica  
Esame di Sistemi Operativi  
Proff. Cinque, Cotroneo, Natella

**Prova pratica del 30/11/2023**  
**Durata della prova: 75 minuti**

Lo studente completi 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 ....

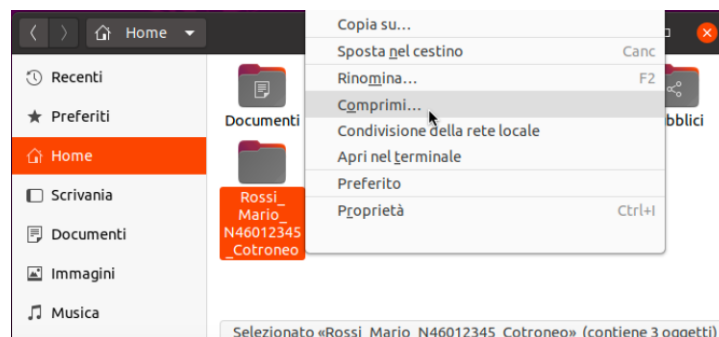
.... per la consegna, dalla shell (assumendo di essere ancora nella cartella di lavoro),

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

```
cd ..
```

```
tar -czvf ./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, collegarsi al Team dell'esame ed effettuare la consegna facendo l'upload del file compresso appena creato.

## Testo della prova

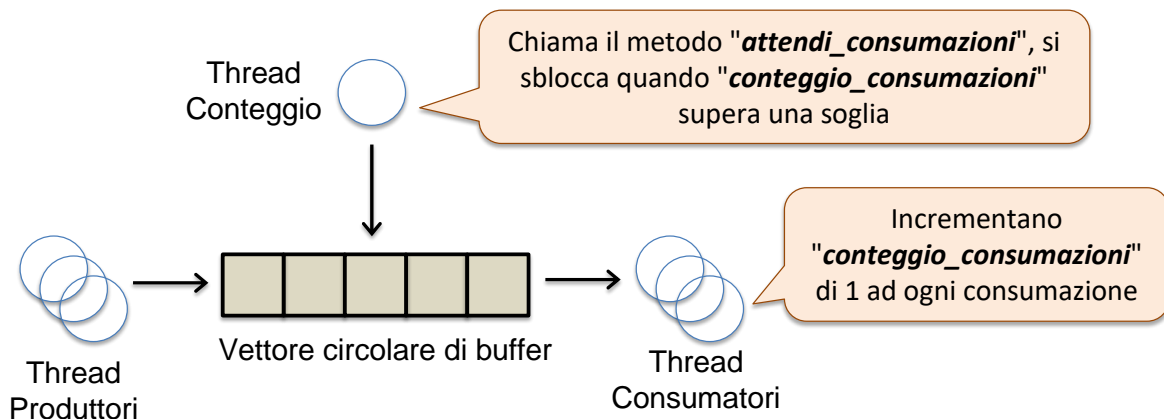
Si realizzi in linguaggio C/C++ una applicazione **multithread** che realizzi lo schema **produttore-consumatore con vettore di buffer circolare**.

Il programma dovrà creare un oggetto monitor, che implementi i tradizionali metodi di produzione e consumazione su vettore circolare.

In aggiunta, il monitor dovrà contenere una variabile "conteggio\_consumazioni", ed implementare un ulteriore metodo, definito come: `int attendi_consumazioni(MonitorProdCons * p, int soglia)`. Quando un thread chiama questo metodo, il thread viene sospeso su una variabile condition, in attesa che siano effettuate un numero di consumazioni maggiore o uguale al valore "soglia".

La variabile "conteggio\_consumazioni" viene incrementata di 1 dai consumatori ad ogni consumazione. Quando la variabile "conteggio\_consumazioni" supera il valore "soglia", il thread sospeso viene sbloccato, e la variabile "conteggio\_consumazioni" viene azzerata.

Il programma dovrà avviare 3 thread di tipo produttore, 3 thread di tipo consumatore, e 1 thread di tipo "conteggio". Ogni produttore dovrà effettuare 10 produzioni, e ogni consumatore dovrà effettuare 10 consumazioni. Il thread di conteggio dovrà chiamare più volte il metodo "attendi\_consumazioni" con soglia pari a 5, fino a quando non siano state effettuate 30 consumazioni in totale.



File da completare:

- main.c
- prodcons.h
- prodcons.c