

[illegible]

## Testo della prova

Si realizzi in linguaggio C/C++ un'applicazione **multiprocesso** basata su **Monitor di Hoare**, per simulare lo streaming video mediante lo schema **produttore-consumatore**. Utilizzare un vettore circolare di buffer condiviso. Si ipotizzi che i frame siano troppo grandi per essere memorizzati nel vettore di buffer. Invece, ogni elemento del vettore dovrà includere le seguenti informazioni:

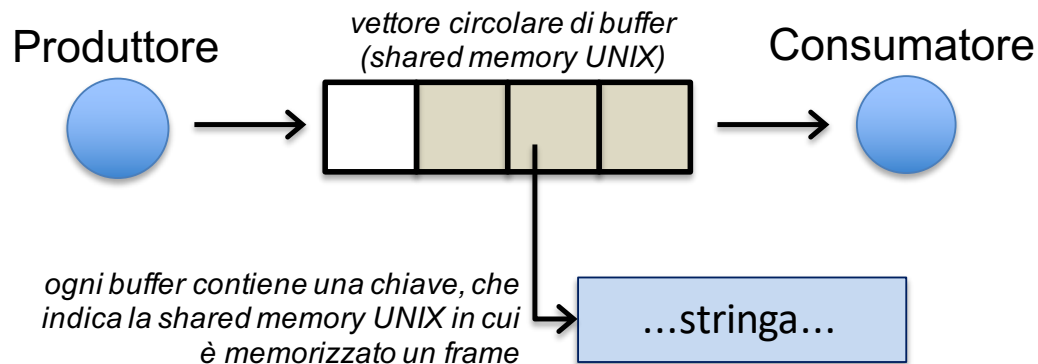
- Stato del buffer (LIBERO/OCCUPATO/IN USO, valore intero)
- PID del processo produttore (valore intero)
- Dimensioni del frame in byte (valore intero)
- Chiave di una shared memory UNIX su cui è memorizzato il frame (valore intero)

I frame dovranno essere contenuti in una shared memory UNIX separata per ogni frame, aggiuntiva alla shared memory del vettore circolare di buffer. Un frame è fatto da una stringa di caratteri, generata casualmente di lunghezza tra 2 e 20 caratteri alfabetici, incluso il carattere terminatore.

Per produrre un frame, il processo produttore dovrà creare una nuova shared memory, farne l'attach, copiarvi una stringa, e farne il detach<sup>1</sup>. Il produttore ne inserirà la chiave nel buffer. Il processo consumatore preleverà dal buffer la chiave, con cui dovrà fare la "attach" della shared memory per poter leggere il contenuto del frame, per poi farne il detach e la cancellazione.

Per generare le chiavi dei frame, utilizzare **ftok()**. È consentito cablare nel codice il primo parametro di **ftok** (percorso), e variare il secondo parametro (carattere) ad ogni produzione incrementandolo di 1.

Si simuli un produttore e un consumatore, con un vettore di 4 buffer. Effettueranno rispettivamente 10 produzioni e 10 consumazioni in un ciclo, attendendo 1 secondo tra le iterazioni. Simulare che ogni produzione duri 1 secondo, utilizzando la primitiva **sleep()**.



File da completare:

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

---

<sup>1</sup> Per il detach di una shared memory, usare la primitiva: `shmdt(const void *shmaddr);`