

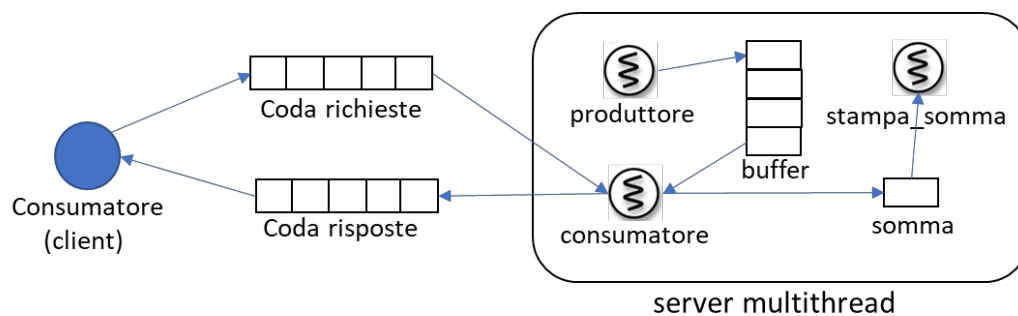
## Testo della prova

Si realizzi in linguaggio C/C++ una applicazione **multiprocesso** che realizzi uno schema **client-server** con server **multithread**. Client e server comunicano attraverso due **code di messaggi UNIX**, una coda per le richieste e una coda per le risposte.

Il processo client (file consumatore.c) invia 12 richieste di consumo al server sulla 'coda richieste', attendendo sulla 'coda risposte' un messaggio di risposta con il valore restituito dal server.

Il server detiene una **buffer condiviso di 4 elementi gestito come coda circolare** e genera 1 thread **produttore**, 1 thread **consumatore**, e 1 thread **stampa\_somma**. Il produttore genera 12 valori casuali tra 1 e 10 e li inserisce nel buffer, attendendo se il buffer è pieno. Il consumatore riceve le richieste dalla 'coda richieste', effettua il consumo attendendo qualora il buffer sia vuoto, e risponde al client sulla 'coda risposte' col valore letto. Dopo aver inviato la risposta, il consumatore aggiorna, *in mutua esclusione*, una **variabile condivisa somma**, sommando al suo valore corrente il valore appena consumato dal buffer. L'ulteriore thread **stampa\_somma** è in attesa che si verifichi la condizione che siano stati sommati 12 valori nella variabile **somma**. Una volta verificata la condizione, stampa a video il contenuto della variabile **somma** e termina.

I processi client e server vengono generati da un processo **main** attraverso la primitiva **exec**.



File da completare:

- server.c
- consumatore.c
- main.c