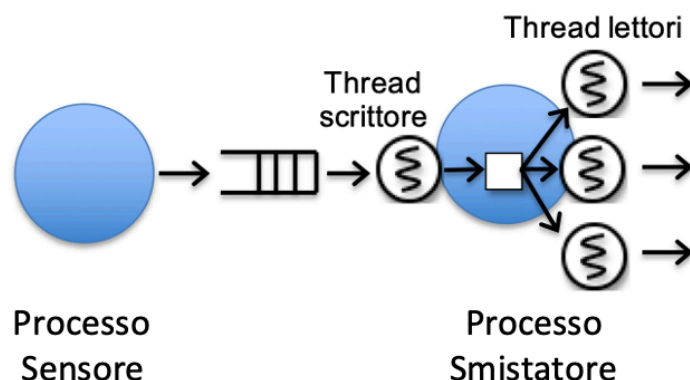


## *Testo della prova*

Si realizzi in linguaggio C/C++ un'applicazione **multithread** basata su **code di messaggi UNIX** che simuli la lettura di dati da parte di un sensore, e la distribuzione dei dati a più destinazioni. Un processo **sensore** deve periodicamente (ogni secondo) generare un valore numerico casuale\*, tra 0 e 10, e inviarlo tramite una coda di messaggi. Un processo **smistatore** deve ricevere i dati e stamparli a video.

Il processo smistatore deve includere 4 thread: 1 scrittore e 3 lettori, che comunicano attraverso una variabile condivisa secondo un modello di sincronizzazione lettori-scrittori. Il thread **scrittore** deve prelevare i messaggi dalla coda di messaggi, mettendosi in stato di attesa in caso non ce ne siano, e deve poi scrivere il valore ricevuto sulla variabile condivisa. I thread **lettori** devono leggere periodicamente (ogni secondo) il contenuto della variabile condivisa, moltiplicarlo per un intero casuale compreso tra 1 e 100 e stamparlo a video. Come nello schema generale dei lettori-scrittori, i thread lettori devono poter **leggere in concorrenza** la variabile condivisa.

Un processo padre dovrà creare gli altri processi attraverso **fork()** e poi avviarli come **eseguibili** distinti. Il processo sensore dovrà generare 10 messaggi in un ciclo, attendendo 1 secondo tra le iterazioni. Il thread scrittore effettuerà una scrittura ad ogni messaggio. I thread lettori dovranno leggere la variabile condivisa per 10 volte in un ciclo, attendendo un secondo tra le iterazioni, e stampando altrettanti messaggi.



\* Gli interi casuali possono essere generati attraverso la funzione:

$$\text{rand()} \% (\text{max\_number} + 1 - \text{minimum\_number}) + \text{minimum\_number}$$