

Esercitazione Produttore e Consumatore

1 - Introduzione Produttore e Consumatore

Fare un programma che crea un **Produttore** ed un **Consumatore** con il seguente ruolo:



- *Produttore*: genera all'infinito un numero casuale tra 0 e 1023 ogni Xms, dove X è un numero casuale tra 100 e 1000.
- *Consumatore*: deve consumare i numeri generati dal Produttore e ogni volta che li consuma stampa il numero letto e stampa una statistica dei numeri pari e dei numeri dispari letti

Il programma deve garantire che i dati prodotti vengano consumati in ordine FIFO e che non vi devono essere *perdite* dei dati prodotti dal Consumatore.

Suggerimenti

Utilizzare **quattro** classi una per Produttore, una per Consumatore, una classe Buffer ed il **Main** per avviare tutto il sistema

1.1 - Variante LIFO

Modifica il codice in modo tale che i dati nel buffer non siano più FIFO ma LIFO

1.2 - Variante priorità

Modifica il codice in modo tale che i numeri PRIMI sono i primi ad essere estratti dal buffer, stampare per ogni valore estratto dal buffer il suo tempo di permanenza all'interno del buffer

1.3 – Tempo elaborazione

Modificare il codice in modo tale che il produttore produca un valore ogni 100, 250, 500, o 100 ms, mentre il Consumatore estrae i valori ogni 200, 400, 600, 800 o 1000 ms. Il Produttore e Consumatore terminano dopo aver prodotto/consumato 5000 valori.

Il programma deve visualizzare il tempo di esecuzione del Produttore, del Consumatore e di tutto il programma. Effettuare le prove con buffer di 5, 20, o 50 elementi

2 – Un Produttore e tanti Consumatori

Fare un programma che crea un **Produttore** ed un valore C di **Consumatore** con il seguente ruolo:

- *Produttore*: genera all'infinito un numero casuale R tra -1024 e 1023 e attende $R/100 \times 50\text{ms}$ se positivo, altrimenti 200ms
- *Consumatore*: ogni consumatore deve contare i numeri negativi e calcolare la media degli ultimi 5 valori positivi. Ogni volta che consuma un valore stampa la statistica aggiornata e poi aspetta 100ms

Il programma deve garantire che i dati prodotti vengo consumati in ordine FIFO e che non vi devono essere *perdite* dei dati prodotti dal Consumatore e ogni dato può essere consumato da un solo Consumatore.

Progettare il sistema in modo da MASSIMIZZARE la scalabilità, cioè fare in modo che più Consumatori possono “accedere contemporaneamente” al Buffer

2.1 – Variante multicast

Come andrebbe modificato il Buffer in modo tale da garantire che ogni valore prodotto arrivi a TUTTI i Consumatori?

2.2 – Best Effort

Modificare il programma in modo tale che il produttore non sia MAI messo in attesa, accettando la possibilità di perderei dei dati (il programma deve stampare periodicamente il numero di dati “persi”)

3 – In pizzeria

Fare un programma che simula una pizzeria dove un forno produce da 1 ad 8 pizze alla volta e vi sono tre camerieri per prendere l'ordine e servire ai tavoli, i camerieri possono prendere ordini da un tavolo oppure portare le pizze ad un tavolo. Non è detto che il cameriere che prende l'ordine sia lo stesso che serve al tavolo.

La pizzeria ha a disposizione 20 tavoli ed una capienza di massimo 100 persone. La simulazione deve visualizzare quante persone sono ai tavoli e termina quando tutte le pizze sono state consegnate.