

Produttore e consumatore at work: crivello concorrente

L'esercizio consiste nell'implementare in Java Concorrente l'algoritmo del Crivello di Eratostene per calcolare i numeri primi nell'intervallo 1..N utilizzando una pipeline di schemi produttore e consumatore.

L'algoritmo del crivello di Eratostene è basato sulla seguente idea.

Per selezionare i numeri primi compresi tra 1 ed N prima si scrivono in sequenza tutti i numeri. Poi si iniziano a togliere i numeri divisibili per 2. Poi i numeri divisibili per il primo numero che rimane subito dopo il 2. Poi per quello seguente ecc. Alla fine rimarranno soli i numeri primi.

Simulazione algoritmo:

<https://jochen-ziegenbalg.github.io/materialien/Demos+Simulationen/Sieve-of-Eratosthenes/Sieve-of-Eratosthenes-Simulation.html>

In questo algoritmo possiamo sfruttare la concorrenza costruendo una pipeline di thread T1 T2 T3... associati ai numeri primi P1 P2 P3 ...

A partire da $P_1=2$, ogni thread T_i è associato al numero primo P_i e ha un canale di comunicazione (coda FIFO) con il thread precedente e successivo.

Il thread T_i riceve una sequenza di numeri dal thread T_{i-1} e passa a T_{i+1} solo i numeri non divisibili per P_i (cioè filtra i numeri non primi rispetto a $P_1 \dots P_i$).

Se T_i non ha ancora un thread successore, al primo numero P_i+1 non divisibile per P_i che riceve crea il thread T_{i+1} associandolo a P_{i+1} e poi continua a filtrare i numeri seguenti come descritto precedentemente.

Implementare un programma Java concorrente che implementi questo algoritmo.

Possibile architettura:

