



Università degli Studi di Bologna
Facoltà di Ingegneria

Corso di Reti di Calcolatori T

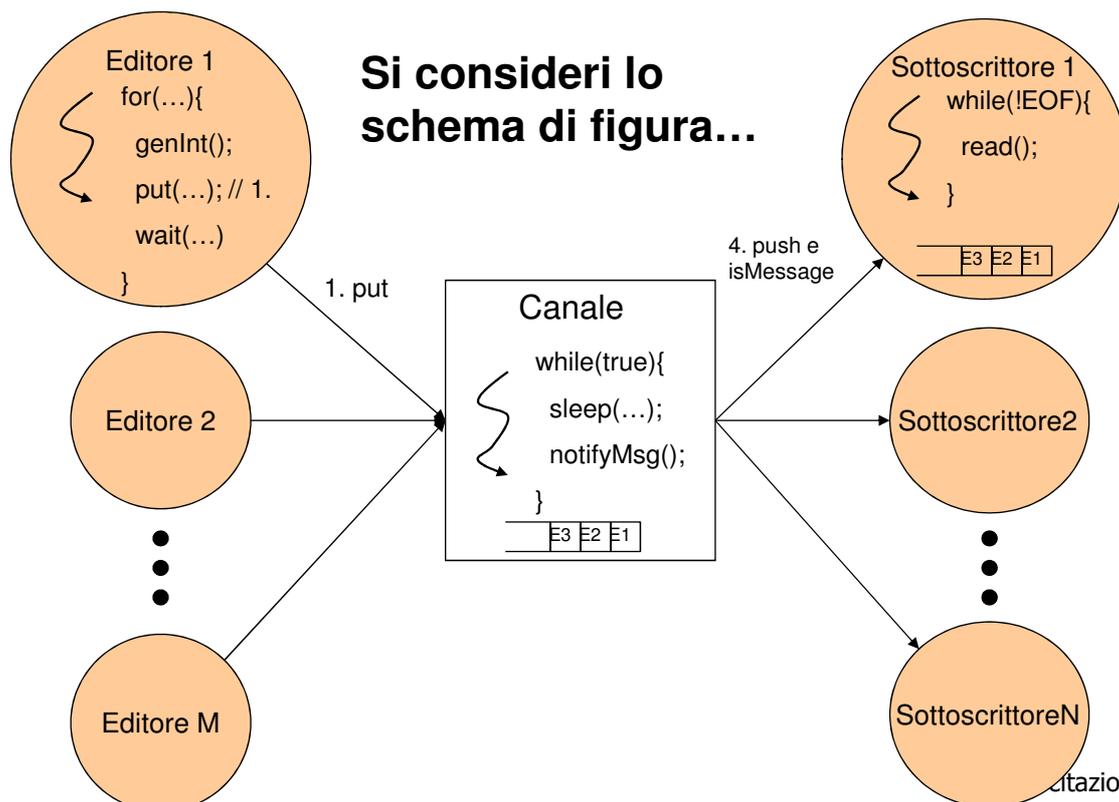
*Esercitazione 0 (proposta)
Multithreading in Java*

Luca Foschini

Anno accademico 2013/2014

Esercitazione 0 1

Architettura di riferimento



Esercitazione 0 2

Proposta (per passi): Passo 1

Modificare l'Editore per consentire editor multipli

si consiglia di aggiungere un campo intero da utilizzare come identificativo, come accade già per i Sottoscrittori.

- **Eliminare la generazione con input da Console** sostituendola con una generazione automatica di stringhe. Si eseguano cioè `N_PUBBLICAZIONI`, ad esempio 7: ciascuna pubblicazione (stringa pubblicata) contiene l'identificatore dell'editore e il numero dell'iterazione corrente, ad esempio: "E1, pubblicazione 1".
- **Il tempo** tra una generazione e l'altra viene **fissato in modo casuale** fra 2 e 7 secondi (`(Math.random() * 5000) + 2000`).
- **Modificare il programma di test** vedendo cosa accade nel caso di più Editori che utilizzino tutti lo stesso Canale.
Quali **problemi** possono verificarsi in questo caso? Il **funzionamento è ancora corretto** o sono necessari ulteriori accorgimenti per evitare deadlock e starvation?

Esercitazione 0 3

Proposta (per passi): Passo 2

Modificare l'architettura del Canale in modo da disaccoppiare il momento di generazione dell'evento dalla notifica.

Ridefinire il Canale come **entità autonoma ed attiva** per separare meglio le altre entità.

Il funzionamento finale che si vuole ottenere è il seguente: gli Editori pubblicano gli eventi generati in modo asincrono presso il Canale ed il Canale **attivo**, ogni 2 secondi, **notifica a tutti i Sottoscrittori** registrati gli eventi ricevuti dagli Editori dall'ultimo istante di notifica (cioè 2 sec. prima).

Esercitazione 0 4

Proposta (per passi): Passo 2 – Dettagli

Si noti che per ottenere il comportamento desiderato bisognerà modificare il Canale in modo da **renderlo un processo** e **aggiungere una coda degli eventi**.

In particolare:

- Il **Canale** è un thread che deve realizzare il comportamento descritto nel lucido precedente. Si realizzi la coda degli eventi come una struttura dati di tipo *Array* di MAX_EVENTS elementi. Contestualmente agli aggiornamenti di tale struttura dati sarà necessario aggiornare anche un *contatore* che ne indichi il grado di riempimento.
NB: *la struttura dati contenenti gli eventi ha dimensione finita*. Si vuole perciò realizzare il metodo put in modo che sia sospensivo in caso di coda degli eventi del Canale piena. In tal caso, perciò i thread degli Editori verranno sospesi fino a quando c'è spazio nella coda.
- L'**Editore** è lo stesso descritto al passo 1. Si noti che rispetto all'Editore progettato nell'esercitazione svolta questo Editore *non* pubblica l'EOF, essendo l'interazione da console sostituita con una generazione automatica.
- Il **Sottoscrittore** rimane invariato, stampa cioè semplicemente a video un certo numero di eventi, ad esempio MAX_EVENTS, poi termina.

Esercitazione 0 5



Proposta di estensione



Si estenda il programma sviluppato in modo che gestisca **la generazione e la distruzione di tutti i processi** messi in esecuzione (Canale compreso); il risultato che si vuole ottenere è il seguente.

Attivazione: gli Editori **non** devono **effettuare pubblicazioni prima che il Canale sia stato attivato**; in altre parole, gli Editori vengono sospesi fino a quando il Canale è stato attivato. In modo simile, il Canale non effettua notifiche prima che almeno un Sottoscrittore si sia registrato.

Terminazione: quando tutti i Sottoscrittori terminano l'esecuzione e rimangono attivi solo degli Editori, tali **Editori devono terminare senza effettuare ulteriori pubblicazioni** e, una volta terminati tutti gli Editori, anche **il Canale termina**; in modo duale si vuole gestire la terminazione nel caso in cui gli Editori terminino prima dei Sottoscrittori.

Esercitazione 0 6

Suggerimenti

Si suggerisce di istanziare, all'interno del main del programma, una **struttura dati** che mantenga **il numero di Editori e Sottoscrittori attivi**. Tale struttura dati, nota agli Editori, ai Sottoscrittori e al Canale, verrà utilizzata dagli stessi per gestire in modo sincronizzato la terminazione di tutti i processi.

Consegna

Chi vuole può inviare via email lo svolgimento dell'estensione ai docenti, in modo da discutere la soluzione ed eventualmente pubblicarla sul sito del corso