Università degli Studi dell'Insubria
Dipartimento di Scienze Teoriche e Applicate
Programmazione concorrente e distribuita
Prof. Luigi Lavazza

Esame dell' 8 giugno 2021

Descrizione del sistema da realizzare

Si desidera realizzare un sistema distribuito che gestisce la distribuzione di notizie.

Ogni notizia ha un "tipo notizia" (politica, attualità, scienza, sport) e un contenuto, che per semplicità consiste in una stringa.

Il sistema è costituito dai seguenti elementi.

- Un insieme di programmi di tipo FruitoreNotizie
- Un programma Pubblicatore.

Il programma Pubblicatore è provvisto di un insieme di thread ProduttoreNotizie in grado di produrre notizie. Le notizie sono memorizzate nel Pubblicatore, che si occupa della loro distribuzione.

Ogni programma di tipo FruitoreNotizie è in grado di ricevere notizie dal Pubblicatore. A questo scopo ogni FruitoreNotizie informa il Pubblicatore del tipo di notizie che gli devono essere comunicate. I programmi FruitoreNotizie non prendono mai l'iniziativa di leggere le notizie: aspettano che vengano loro comunicate dal Pubblicatore.

Il Pubblicatore riceve dai programmi FruitoreNotizie richieste del tipo:

- un certo FruitoreNotizie è interessato a ricevere notizie di un dato tipo T (se un FruitoreNotizie è interessato a più di un tipo di notizie, dovrà comunicare ciascun tipo separatamente al Pubblicatore).
- un certo FruitoreNotizie non è più interessato a ricevere notizie di tipo T (se un FruitoreNotizie non è più interessato a diversi tipi di notizie, dovrà comunicare ciascun tipo separatamente al Pubblicatore).

Il Pubblicatore e i vari programmi FruitoreNotizie sono processi che girano su macchine potenzialmente diverse.

In ogni momento esiste una unica notizia per ogni tipo.

Ciascun ProduttoreNotizie esegue iterativamente la seguente sequenza di operazioni:

- decide casualmente il tipo T della notizia a cui contribuire
- produce in modo causale una stringa X
- concatena X al contenuto della notizia di tipo T.

Il Pubblicatore esegue ogni P secondi le seguenti operazioni, per ogni tipo T di notizia:

- comunica la notizia di tipo T a tutti i FruitoreNotizie interessati a quel tipo di notizia
- cancella il contenuto della notizia di tipo T, in modo che il prossimo contributo da un ProduttoreNotizie verrà aggiunto alla stringa vuota.

Per il testing del sistema si ponga P=5 secondi. Di conseguenza bisogna che i ProduttoreNotizie aggiornino la notizia con un periodo ben inferiore a 5 secondi.

Si realizzi il sistema in modo da gestire le eccezioni più comuni. Ad es. se un FruitoreNotizie diventa irraggiungibile (si disconnette), il Pubblicatore deve continuare a funzionare regolarmente, anche se non riesce a mandargli più notizie.

Per la realizzazione del sistema si può scegliere se utilizzare RMI o socket.

NB: il Pubblicatore e i FruitoreNotizie devono essere processi distinti, in grado di girare su macchine distinte, quindi non possono essere thread di un unico processo. Viceversa, Pubblicatore e ProduttoreNotizie sono thread di uno stesso processo. Realizzare il sistema con almeno 5 thread ProduttoreNotizie contemporaneamente attivi.

Implementare i programmi in modo che scrivano a terminale cosa stanno facendo.

Consigli e suggerimenti

Leggete attentamente le specifiche del sistema. Programmi che fanno cose diverse da quanto richiesto saranno valutati insufficienti.

Nel caso ci sia qualche elemento di importanza marginale non perfettamente specificato (ad es., come ProduttoreNotizie deve generare le stringhe X) qualunque implementazione ragionevole va bene. Se si riscontra una lacuna essenziale nelle specifiche, chiedere delucidazioni al docente.

Si consiglia di realizzare il sistema in modo incrementale, non “tutto insieme”. Ad es., si può prima realizzare il Pubblicatore con i suoi ProduttoreNotizie come programma locale e poi trasformarlo in un programma distribuito con i FruitoreNotizie che fanno da client.

Cosa occorre consegnare

Bisogna obbligatoriamente consegnare il codice sorgente (file .java) e le indicazioni necessarie per compilare ed eseguire il programma.

Regole:

1) Non utilizzare librerie diverse da quelle standard di Java.
2) Non usare versioni non standard di Java. Considerate che per la correzione si usa openjdk 11, quindi fare in modo che il programma consegnato funzioni con tale versione di Java.
3) Nel codice sorgente non utilizzare caratteri diversi da quelli ANSI. Ad es., non usare le lettere accentate. Pertanto, nelle stringhe e anche nei commenti, evitare accuratamente l’uso di lettere accentate.
4) Allegare istruzioni di compilazione ed esecuzione chiare e sintetiche. Si veda come esempio il file "readme" nella soluzione proposta per gli appelli del 9/6/20 e seguenti.

Tutti i programmi che contravverranno a queste regole saranno valutati negativamente (cioè prenderanno un voto insufficiente).

Come consegnare

Bisogna caricare un unico file .zip sul sito dell’e-learning nello spazio previsto per la risposta alla domanda. Il file deve avere un nome che comprende i cognomi di tutti i membri del gruppo di progetto.

La consegna va effettuata entro la data e ora di chiusura, inderogabilmente. Non verranno presi in considerazione progetti non consegnati come prescritto. Si consiglia pertanto di provvedere con un certo anticipo sulla scadenza, per evitare possibili problemi.

Tutti i membri di un team di progetto devono effettuare la consegna del medesimo file.

Solo per gli studenti con disabilità o DSA partecipanti all'appello: la consegna deve avvenire entro il tempo indicato, incrementato del tempo aggiuntivo previsto dal Progetto Formativo Individualizzato. La consegna va fatta per email. Attenzione: come da regolamento, gli studenti che desiderano avvalersi del tempo aggiuntivo devono comunicarlo al docente con 48 ore di anticipo. Chi non lo avrà fatto non potrà avvalersi del tempo aggiuntivo.

Criteri di valutazione

I programmi che presentano errori di compilazione (o che risultano altrimenti impossibili o difficili da compilare per mancata o cattiva documentazione) avranno una valutazione insufficiente.

I programmi che vanno in deadlock, in crash o che siano affetti da gravi problemi di altro tipo avranno una valutazione insufficiente.

I programmi inutilmente complessi verranno penalizzati. Le implementazioni semplici e pulite saranno premiate.