

Programmazione di Sistemi in Rete (A.A. 2017/18)

Esercitazione n. 5 del 23 ottobre 2017

Prof. Eugenio Zimeo

Esercizio 5.1

Scrivere una classe di thread (`Printer`) in Java in cui il metodo `run()` realizzi la scrittura a video ripetuta per 100 volte di una stringa composta dall'indice di iterazione (del ciclo) e da un messaggio memorizzato in una variabile di istanza, inizializzata dal costruttore. Scrivere un programma che testi il funzionamento della classe istanziando tre thread di tipo `Printer` caratterizzato ciascuno da un messaggio diverso. Utilizzare i due approcci visti a lezione per la definizione del corpo dei thread: (a) estensione della classe `Thread`; (b) implementazione dell'interfaccia `Runnable`.

Esercizio 5.2

Scrivere un'applicazione client/server in Java con socket orientate ai flussi (protocollo di trasporto: TCP) che consenta di effettuare in remoto il fattoriale di un numero intero. Il client dovrà leggere da `stdin` il numero di cui calcolare il fattoriale e inviarlo al server. Questo, una volta ricevuto il numero, ne calcola il fattoriale restituendo il valore ottenuto al client per la visualizzazione.

Modificare l'applicazione in modo che il **server abbia un comportamento concorrente**: ogni client sarà servito da uno specifico thread nel server.

Esercizio 5.3

Usando i thread in Java, realizzare la versione full-duplex dell'esercizio 4.2 (Chat Half Duplex). Si tenga presente che il codice che riguarda la lettura da `stdin` e la scrittura sul canale di rete e quello che riguarda la lettura dal canale di rete e la scrittura su `stdout` deve essere eseguito in modo concorrente al fine di consentire di inviare e ricevere messaggi in qualsiasi momento. Si implementino allo scopo due classi di thread: **Writer** e **Reader**.

Esercizio 5.4

Modificare l'applicazione dell'esercizio 5.2 in modo da realizzare il calcolo del fattoriale di un numero intero in parallelo (se sono disponibili risorse computazionali). Allo scopo, si suggerisce di suddividere il calcolo del fattoriale di n in $\prod_{i=0}^k (n-i) \cdot \prod_{i=k+1}^{n-1} (n-i)$.

Il server individua il numero k (in modo da realizzare un numero simile di operazioni in due attività diverse) e affida il calcolo di ciascuna produttoria a due thread (che chiamiamo worker). I risultati parziali vengono moltiplicati per produrre il risultato finale da inviare al client e visualizzare sullo standard output. Opzionalmente, si realizzi una versione distribuita del server in cui l'elaborazione delle due produttorie è affidata a dei processi worker.