



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

Esame del 20 luglio 2021

Leggete attentamente tutto questo documento. Programmi che fanno cose diverse da quanto richiesto riceveranno una valutazione insufficiente.

Descrizione del sistema da realizzare

Si desidera realizzare il sistema distribuito costituito da un programma server e da molti client. Il server e ciascun client possono risiedere su macchine diverse, come in ogni sistema distribuito.

Il server fornisce un semplice servizio di “eco”, restituendo a ciascun client la stringa che quest'ultimo gli ha inviato.

Si desidera realizzare il sistema simulando il funzionamento di RMI attraverso l'uso di socket. In particolare:

- Un processo Registry offre le stesse funzionalità del registry RMI (operazioni di bind e lookup).
- il server pubblica il proprio riferimento remoto (costituito da indirizzo IP e port) sul registry (che può risiedere su una macchina diversa da quella su cui gira il server), mediante una operazione bind;
- il client reperisce il riferimento remoto al server cercandolo sul registry, con una operazione lookup;
- il client crea un oggetto locale al quale passa il riferimento remoto al server; questo oggetto si comporta da proxy del server;
- per l'esecuzione dei servizi del server, il client si rivolge al proxy locale, il quale provvede a comunicare col server; in questo modo il codice del client non contiene operazioni che riguardano esplicitamente l'uso dei socket, come avviene con RMI.

Il server offre un'interfaccia (l'equivalente di una interfaccia remota RMI) comprendente unicamente un metodo “echo” avente un argomento e un risultato di tipo String. Il metodo echo restituisce semplicemente la stringa ricevuta.

Per testare il sistema, si può implementare un client che

1. Si rivolge al registry per recuperare il riferimento remoto al server;
2. Genera casualmente diverse richieste di echo, che vengono inviate al server.

NB: le stringhe argomenti di echo possono essere generate come si preferisce. L'unico vincolo è che devono contenere un identificatore del client (in questo modo il client sa che gli viene restituita una stringa che lui stesso ha emesso in precedenza). Ad es., il client XYZ genera stringhe “cli_XYZ_pippo”, “cli_XYZ_pluto”, ecc.

Si realizzi il sistema in modo da gestire le eccezioni più comuni, con particolare riferimento a quelle relative alle comunicazioni remote.

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

Consigli e suggerimenti

Server, Registry e Client devono essere processi distinti, in grado di girare su macchine distinte, quindi non possono essere thread di un unico processo. NB: si possono usare client multi-thread per il testing, ma NON vanno consegnati.

Nel caso ci sia qualche elemento di importanza marginale non perfettamente specificato sopra, qualunque implementazione ragionevole va bene.

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. NB: i programmi verranno compilati con javac (versione 11.0) e attivati direttamente da terminale. Il fatto che funzionino correttamente in qualche IDE è irrilevante agli effetti dell'esame.
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.
5. È severamente vietato mettere diverse classi nello stesso file (i file sono gratis: non c'è nessun bisogno di fare economia).

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.

Pratiche di programmazione discutibili verranno penalizzate, anche se non strettamente