

Laboratorio di Reti di Calcolatori

Prova Java Socket - appello di giugno 2020

Tema d'esame:

Si realizzi un sistema di tele-conferenza tra due utenti come segue: il sistema comprende un server, che coordina la collaborazione tra gli utenti, e una coppia di utenti, A e B. Il comportamento dei processi è definito dalle seguenti specifiche:

SERVER

1. è single-thread e comunica con i processi utente utilizzando i servizi di TCP
2. se riceve una registrazione da un processo utente:
 - nel messaggio di registrazione riceve il nome del processo utente
 - se il nome è unico (non esiste utente già registrato col medesimo nome), il server rende ACK all'utente e ne ricorda nome e indirizzo
 - altrimenti il server rende NACK all'utente
3. quando il server ha ricevuto la registrazione dei due utenti, entra in stato idle
4. quando il server in stato idle riceve dall'utente A registrato l'intenzione di iniziare una conversazione, opera come segue:
 - va in stato working
 - assegna il diritto di parola ad A
 - si sospende in attesa di ricevere l'intervento di A
5. quando il server in stato working riceve dall'utente X ($X \in \{A, B\}$) - avente correntemente diritto di parola - il testo dell'intervento, opera come segue:
 - invia uno ACK ad X
 - invia l'intervento di X all'altro processo (diciamo Y), notificandogli che è il suo turno di parola
 - si sospende in attesa di ricevere l'intervento di Y
6. quando il server in stato working riceve dall'utente X ($X \in \{A, B\}$) - avente correntemente diritto di parola - la notifica che la conversazione è chiusa, opera come segue:
 - invia uno ACK a X
 - notifica all'altro processo (diciamo Y) che la conversazione è terminata e ne attende ACK
 - torna in stato idle
7. il server non termina mai.

UTENTE:

1. è single-thread e comunica con il server utilizzando i servizi di TCP
2. alla partenza riceve da tastiera il nome da utilizzare per la registrazione con il server. Si connette al server e gli manda un messaggio di registrazione con il proprio nome, attendendone la risposta:
 - se la risposta è ACK, l'utente continua con il funzionamento descritto ai punti successivi
 - altrimenti l'utente notifica un messaggio di errore a video e torna a chiedere l'inserimento da tastiera di un nome

3. l'utente registrato come A attende l'indicazione da tastiera di voler iniziare una conversazione. L'utente registrato come B attende un messaggio dal server
4. se l'utente A riceve da tastiera il segnale di inizio conversazione, opera come segue:
 - invia al server la richiesta di iniziare una nuova conversazione, e attende il diritto di parola dal server
 - richiede da tastiera il testo dell'intervento e lo invia al server, attendendone uno ACK
 - si blocca in attesa di ricevere dal server la risposta dell'altro processo
5. se un utente X ($X \in \{A, B\}$) riceve dal server l'intervento dell'altro processo con la notifica che è il suo turno di parola, opera come segue:
 - richiede da tastiera il testo dell'intervento e lo invia al server, attendendone uno ACK
 - si blocca in attesa di ricevere dal server la risposta dell'altro processo
6. se un utente riceve dal server il segnale di fine conversazione, opera come segue:
 - invia al server uno ACK e torna ad eseguire il punto (3)
7. se un utente riceve da tastiera la notifica di fine conversazione
 - invia al server la notifica di chiusura conversazione
 - attende di ricevere dal server uno ACK e torna al punto (3)
8. nessun utente termina mai.

Tutti gli interventi sono stringhe di al massimo 20 B.

La definizione del formato dei messaggi, così come i dettagli implementativi non specificati sopra, sono a discrezione dello studente. Tutti i messaggi ricevuti da ogni processo devono essere mostrati a video.

Modalità di consegna:

Lo studente deve consegnare due file *server.java* e *utente.java*; qualunque altro file presente nella consegna non verrà considerato per la correzione.

1. Prima di consegnare, assicurarsi di salvare il contenuto di entrambi i file sorgente prodotti.
2. Collegarsi al sito <http://upload.di.unimi.it>
3. Autenticarsi con login name e password di Ateneo (cioè @studenti.unimi.it)
4. Fare upload dei due file sorgente
5. Fare logout dal sito.

NB: in caso si procedesse ad effettuare la consegna dell'elaborato più di una volta, concatenare un numero progressivo al nome file, così che sia possibile individuare l'ultima versione. Per ognuno dei due file verrà considerata solo l'ultima versione caricata.