

Reti di Calcolatori T

Prova d'esame

16/12/2011 - Compito 1

Tempo a disposizione: 2h

Cognome:
Nome:
Matricola:

È obbligatorio inserire Cognome Nome Matricola e Numero Compito all'inizio di ogni file sorgente, pena la non valutazione del compito, che viene stampato in modo automatico solo in caso siano presenti gli elementi detti sopra.

Si devono consegnare singolarmente tutti i file sorgente e gli eseguibili (per favore, solo quelli relativi ai file sorgente consegnatili!).

Leggete con attenzione le specifiche del problema prima di impegnarvi "a testa bassa" nello sviluppo delle singole parti. Naturalmente, ci aspettiamo che i componenti da consegnare siano stati provati e siano funzionanti.

Per favore, pulite le versioni consegnate: non lasciate parti commentate inutili, togliete linee non usate e variabili non usate, definite tutte le parti necessarie e consegnate solo le cose significative.

La prova intende valutare le capacità progettuali e di programmazione sia in ambiente Java che in ambiente C; pertanto è consigliato sviluppare entrambe le soluzioni richieste al meglio.

Per entrambi le parti, sia in Java che in C, si effettuino gli opportuni controlli sui parametri della richiesta e si gestiscano le eccezioni, tenendo presente i criteri secondo cui si possa ripristinare il funzionamento del programma oppure si debba forzarne la terminazione.

Si richiede il progetto dei servizi di **VotaSanremo**, per la gestione della votazione della nota manifestazione canora: i servizi mantengono, per ogni canzone, le seguenti informazioni: **cantante** e **titolo** intesi ciascuno come **identificatori** unici della canzone (ossia ogni titolo di canzone e ogni cantante formano ciascuno un dominio di nomi unici), il **nome del file** con il testo la canzone, e anche assunto come **nome** unico all'interno del direttorio, e un intero che riporta il **numero totale di voti ricevuti** (come da tabella sotto); si consideri il direttorio corrente come direttorio di partenza da cui si lanciano sia il client sia il server.

In particolare, si vogliono realizzare le seguenti funzionalità:

1. **visualizzazione delle canzoni con più di V voti**: questa operazione richiede il numero V di voti, quindi restituisce i titoli delle canzoni con un numero di voti maggiore di V;
2. **reperimento del testo di una canzone**: questa operazione richiede il titolo della canzone, quindi trasferisce dal server al client il file di testo cercato;
3. **espressione di un voto per una canzone**: questa operazione richiede all'utente il titolo o il cantante (si accetta uno qualsiasi dei due attributi come parametro di input) e aggiunge un voto alla canzone identificata;
4. **visualizzazione dei nomi dei cantanti delle due canzoni meno votate**: questa operazione restituisce i nomi dei cantanti corrispondenti alle due canzoni che hanno ricevuto il numero minore di voti fra quelle presenti nella struttura dati.

Si progetti con particolare attenzione la **struttura dati** che mantiene lo stato, fino ad un massimo di N canzoni (L, per libero come valore di default), da implementare nei diversi ambienti richiesti.

Cantante	Titolo	Voti	Nome file
L	L	L	L
Brasco	Io odio il giovedì	100	odioGiov.txt
Viga	Bar da Alfredo	50	alfredo.txt
...
L	L	L	L

Si considerino e si segnalino le possibilità di interferenze fra le operazioni, evitandole dove necessario.

Parte Java

Utilizzando **java RMI** sviluppare un'applicazione C/S che consenta di effettuare le operazioni remote per:

- visualizzare le canzoni con più di V voti;
- esprimere un voto per una canzone.

Il progetto RMI si basa su un'interfaccia (contenuta nel file *RMI_interfaceFile.java*) in cui vengono definiti i metodi invocabili in remoto dal client:

- Il metodo **visualizza_canzoni_conpiudi_V_voti** accetta come parametro d'ingresso il numero V di voti e restituisce la lista delle canzoni con un numero di voti maggiore di V.
- Il metodo **esprimi_voto** accetta come parametro d'ingresso una stringa che rappresenta il titolo o il cantante (cioè accetta uno qualsiasi dei due come parametro unico di input) e aggiorna la struttura dati aggiungendo un voto alla canzone scelta.

Si progettino le classi:

- **RMI_Server** (contenuta nel file *RMI_Server.java*), che implementa i metodi del server invocabili in remoto;
- **RMI_Client** (contenuta nel file *RMI_Client.java*), il processo che realizza l'interazione con l'utente **proponendo ciclicamente i servizi** che utilizzano i due metodi remoti, e stampa a video i risultati, fino alla fine del file di input da tastiera.

Parte C

Progettare un **servitore multiservizio (usando obbligatoriamente una select)** che consenta di effettuare le operazioni remote per:

- reperire il testo di una canzone;
- visualizzare i nomi dei cantanti delle due canzoni meno votate.

Più in dettaglio:

- Il **client_stream** è organizzato come un **processo filtro ciclico fino alla fine del file di input** e realizza la funzionalità di **reperimento del testo di una canzone** utilizzando **socket stream** e **un'unica socket**.

Per ogni richiesta, il client richiede all'utente e invia al server il titolo della canzone, quindi riceve il file corrispondente stampandolo a video e salvandolo anche sul file system locale.

- Il **client_datagram** è organizzato come un **processo filtro ciclico fino alla fine del file di input** e realizza la funzionalità di **visualizzazione dei nomi dei cantanti delle due canzoni meno votate** utilizzando **socket datagram**.

Per ogni richiesta, il client invia al server la richiesta; quindi riceve i nomi dei cantanti corrispondenti e li stampa a video.

- Il **server** principale unico discrimina le richieste utilizzando la primitiva select: il **server gestisce in modo parallelo** la funzionalità di reperimento del testo di una canzone; mentre la funzionalità di visualizzazione delle due canzoni meno votate **può essere realizzata in modo o seriale o parallelo**: si motivi la scelta fatta.

Per ogni richiesta di **reperimento del testo di una canzone**, il figlio riceve il titolo della canzone, controlla che il file corrispondente esista sul server, e ne trasmette nome e contenuto al client, oppure invia una notifica di errore, ad esempio, se il titolo o il file non esistono.

Per ogni richiesta di **visualizzazione dei nomi dei cantanti delle due canzoni meno votate**, il server riceve la richiesta, scorre la lista individuando le due canzoni che hanno ricevuto il numero minore di voti fra quelle presenti e restituisce al client i nomi dei cantanti corrispondenti.