

Reti di Calcolatori T

Esercitazione 10

19/12/2016

Tempo a disposizione: 2h

Cognome:
Nome:
Matricola:

È obbligatorio inserire Cognome, Nome, e numero di Matricola 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 tutti i file sorgente e tutti gli eseguibili prodotti singolarmente (per favore, solo quelli relativi ai file sorgente consegnati!!!).

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.

In entrambi gli esercizi, sia in Java che in C, si effettuino gli opportuni controlli sugli argomenti della richiesta e si gestiscano le eccezioni verso l'utente, tenendo presente i criteri secondo cui si possa ripristinare il funzionamento del programma oppure si debba forzarne la terminazione.

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.

Si richiede il progetto dei servizi SanRomolo, per la gestione delle votazioni di cantanti di una nota gara canora.

I servizi di SanRomolo mantengono per ogni cantante le seguenti informazioni: il nome del cantante, unico all'interno del sistema; la categoria di appartenenza del cantante ('Campioni' o 'NuoveProposte'); il numero di voti, un intero inteso come numero di voti ottenuti finora dal cantante, e nome del file audio (anch'esso unico all'interno del sistema) contenente la registrazione della performance del cantante.

Si vogliono realizzare le funzionalità di gestione:

1. **visualizza cantanti per categoria**: questa operazione richiede il nome della categoria e restituisce la lista di tutti i cantanti presenti riportando anche i voti ottenuti;
2. **visualizzazione del cantante più votato per ciascuna categoria**: questa operazione restituisce nome e numero di voti del cantante più votato per ciascuna delle due categorie;
3. **votazione di un cantante**: questa operazione richiede il nome del cantante, aggiunge un voto e visualizza il numero di voti ottenuti.
4. **scaricamento degli audio dei cantanti di una categoria con più di M voti**: questa operazione richiede il nome della categoria e il numero di voti M e scarica dal server al cliente tutti i file audio relativi.

Si progetti con particolare attenzione la struttura dati Votazioni che mantiene, per ogni cantante, lo stato, fino ad un massimo di N cantanti (L, per libero a default). La struttura dati è da implementare opportunamente nei diversi ambienti richiesti, Java e C.

Nome	Categoria	Voto	File audio
Mandarino	Campioni	1500	vicino.avi
AmaraBianca	Campioni	2000	immobilismo.avi
L	L	-1	L
...
Zucchero	NuoveProposte	550	ascolto.avi
Amari	NuoveProposte	800	cosaE.avi

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 cantanti per categoria;
- visualizzare il cantante più votato per ciascuna categoria.

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_cantanti** accetta come parametro d'ingresso il nome della categoria; quindi restituisce la lista dei cantanti della categoria cercata, riportando per ciascuno di essi il numero di voti ottenuti. La struttura dati restituita deve implementare la lista come array.
- Il metodo **visualizzazione_piu_votato** non ha parametri di ingresso; all'invocazione, il server percorre la struttura dati alla ricerca del cantante più votato per ciascuna categoria e restituisce la struttura dati contenente nome e numero di voti per ciascuno dei due cantanti più votati se è andata a buon fine, un'indicazione di errore altrimenti, ad esempio, se non ci sono cantanti in una certa categoria.

Si progettino inoltre 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:

- scaricare gli audio dei cantanti di una categoria con più di M voti;
- votare un cantante;

utilizzando un'unica connessione per ogni sessione cliente.

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 **scaricamento degli audio dei cantanti di una categoria con più di M voti** utilizzando **socket stream** e un'unica **socket**. Per ogni richiesta, il client richiede all'utente e invia al server il nome della categoria e il numero di voti; quindi riceve i file audio con il loro nome e li salva 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 **votazione di un cantante** utilizzando **socket datagram**. Per ogni richiesta, il client richiede all'utente e invia al server il nome del cantante, quindi riceve l'esito della votazione (numero di voti ottenuti o errore, ad esempio se il cantante non è presente nella struttura dati) stampandolo a video.
- Il **server** principale unico discrimina le richieste utilizzando la primitiva select: il **server gestisce in modo parallelo** la funzionalità di scaricamento degli audio dei cantanti di una categoria con più di M voti; mentre la funzionalità di votazione di un cantante è **realizzata in modo seriale**. Per ogni richiesta di **scaricamento degli audio dei cantanti di una categoria con più di M voti**, il server riceve il nome della categoria e la soglia del numero di voti e restituisce al client tutti i file audio dei cantanti con un numero di voti maggiore della soglia degli M voti, passando il nome del file e il contenuto per ciascuno. Per ogni richiesta di **votazione di un cantante**, il figlio riceve il nome del cantante e controlla l'esistenza del cantante. Se il cantante esiste, incrementa il numero di voti del cantante e spedisce al client il numero di voti ottenuti, altrimenti restituisce un intero negativo per indicare un errore.