

Reti di Calcolatori T

Proposta Esercitazione 5

14/11/2019

Tempo a disposizione: 2h 30'

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 **singolarmente tutti i file sorgente e tutti gli eseguibili prodotti** (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 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 nome e numero di voti di tutti i cantanti appartenenti alla categoria richiesta;
2. **visualizzazione del podio attuale**: questa operazione restituisce nome e numero di voti dei primi tre cantanti attualmente in testa nella competizione;
3. **votazione di un cantante**: questa operazione richiede il nome del cantante, aggiunge un voto e visualizza il numero di voti ottenuti dal cantante;
4. **download file audio delle canzoni con più di M voti**: questa operazione richiede il numero di voti M e visualizza scarica dal server al client i file audio relativi ai cantanti che hanno ottenuto un numero di voti voto maggiore di M.

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	Voti	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

Sviluppare un'applicazione C/S basata su **socket stream** che realizzi le operazioni remote per:

- **visualizzare cantanti per categoria,**
- **visualizzare il podio attuale,**

utilizzando un'**unica connessione per ogni sessione cliente**: questo vincolo è da **intendersi come prioritario e fondamentale**.

Più in dettaglio:

- Il **cliente** è organizzato come un **processo filtro ciclico che consuma l'input fino a fine file** e, per ogni iterazione del ciclo, chiede all'utente quale tipo di operazione vuole effettuare e realizza le interazioni col server utilizzando **una sola connessione per la intera sessione**; alla ricezione di fine file, libera opportunamente le risorse e termina.
Per ogni richiesta ricevuta dall'utente, il client prima invia il tipo di servizio al server, poi gestisce gli invii e le ricezioni necessarie alla realizzazione dello specifico servizio richiesto.
Nel caso della funzionalità di **visualizzazione dei cantanti per categoria**, il client chiede la categoria all'utente e la invia al server, quindi riceve i nomi dei cantanti e i relativi voti, stampandoli a video.
Nel caso della funzionalità di **visualizzazione del podio attuale**, il client riceve i dati (nome, categoria e numero voti) relativi ai tre cantanti più votati stampandoli a video.
- Il **server** è organizzato come un **unico processo che gestisce in modo parallelo** l'interazione coi clienti, generando un figlio **per tutta la sessione di richieste da quel client**. Per ogni richiesta, il processo figlio che serve la sessione con una prima lettura discrimina il tipo di funzionalità richiesto, poi gestisce opportunamente l'operazione e si pone in attesa di nuove richieste dallo stesso client; alla lettura della fine sessione, il figlio termina.
Per ogni richiesta di **visualizzazione dei cantanti per categoria**, il figlio riceve la categoria e restituisce la lista di cantanti (**nome e numero voti**) appartenenti a tale categoria in caso di successo, -1 in caso di problemi, ad esempio se la categoria richiesta non richiesta.
Per ogni richiesta di **visualizzazione del podio attuale**, il figlio percorre la struttura dati alla ricerca dei primi tre cantanti più votati e li restituisce al cliente, indicando per ognuno il voto attuale.

Parte C

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

- download file audio delle canzoni con più di M voti;
- votare un cantante;

utilizzando un'**unica connessione per ogni sessione cliente streaming**.

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 **download dei file audio delle canzoni con più di M voti**. Per ogni richiesta, il cliente acquisisce dall'utente il **numero di voti soglia**, quindi riceve i file audio relativi ai candidati che hanno ottenuto più di M voti.
- Il **client_datagram** è organizzato come un **processo filtro ciclico fino alla fine del file di input** e realizza la funzionalità di **voto per 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 visualizzazione dei cantanti con più di M voti; mentre la funzionalità di votazione di un cantante **è realizzata in modo seriale**.
Per ogni richiesta di **download dei file audio delle canzoni con più di M voti**, il server riceve il numero di voti e invia al client i file audio relativi ai cantanti che hanno ottenuto un numero di voti maggiore della soglia M.
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.