

Reti di Calcolatori T

Reti di Calcolatori L-A

Appello del 14/07/2011

Compito

Cognome:
Nome:
Matricola:

Tempo a disposizione: 2h

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 **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 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.

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.

Si richiede il progetto della gestione dei servizi **Provinci**, per impersari che vogliono condividere i provincini dei loro cantanti promossi. Provincini mette a disposizione un insieme di funzionalità per la gestione e la condivisione di spezzoni di audio (file dati binari) dei provincini canori. Tutti i file (per i singoli provincini) vengono mantenuti in uno stesso direttorio e ad ogni provincino sono associate alcune informazioni specifiche: il **nome del provincino** (unico all'interno del sistema e rappresentato come una stringa col seguente formato: ggmmaaaa_nomeTalentScout_nomeCantante); il **nome del file** (unico all'interno del direttorio); la **durata del provincino** (in secondi); un certo numero di **parole chiave** (fino ad un massimo di K) per l'indicizzazione dei provincini.

In particolare, si richiede di realizzare le seguenti funzionalità:

1. **inserimento di un nuovo provincino**: questa operazione richiede il nome del provincino, il nome del file, la durata e le parole chiave, e trasferisce il file (audio) dal cliente al servitore aggiornando opportunamente la struttura dati;
2. **scaricamento dei provincini che corrispondono ad una parola chiave**: scarica dal servitore al cliente i file di tutti i provincini con la parola chiave cercata;
3. **ricerca di provincini con durata compresa in un intervallo di tempo**: richiede gli istanti di inizio e fine dell'intervallo, e restituisce l'elenco di tutti le informazioni dei provincini (nome del provincino, durata e parole chiave) che ricadono all'interno dell'intervallo di durata cercata;
4. **aggiunta di una parola chiave**: questa operazione richiede il titolo del provincino e la parola chiave da aggiungere; quindi aggiorna la struttura dati.

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

Nome Provincino	File	Durata	Parola1	...	ParolaK
14061978_Lele_Jane	1.avi	180	basso	...	soffice
21091988_Jappi_Maria	2.avi	345	andante	...	L
...
L	L	L	L	...	L
01111983_Cecco_Elvon	124.mov	830	acuto	...	Dolce

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

Parte C

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

- ricercare i provincini di una certa durata in tempo;
- aggiungere una parola chiave.

Il progetto RPC si basa su un'interfaccia (contenuta nel file *RPC_xFile.x*) in cui vengono definite le procedure invocabili in remoto dal client:

- La procedura **ricerca_provincini** accetta come parametro d'ingresso una struttura dati con gli istanti di inizio e fine dell'intervallo di durata di interesse, e restituisce una lista delle informazioni di al massimo 8 provincini (da un minimo di 0 ad un massimo di 8), tutti con durata compresa fra i due istanti indicati; si usi una opportuna struttura dati contenente nome, durata e parole chiave di ogni provincino.
- La procedura **aggiungi_parola** accetta come parametri d'ingresso il nome del provincino e la parola chiave, quindi aggiorna la struttura dati del server e restituisce l'esito dell'operazione: esito positivo se tutto è andato a buon fine; esito negativo altrimenti, ad esempio se il provincino non esiste, se la parola chiave aggiunta eccede le K parole chiave consentite, o se è già presente.

Si progettino inoltre:

- **RPC_Server** (contenuta nel file *RPC_Server.c*), che implementa le procedure del server invocabili in remoto;
- **RPC_Client** (contenuta nel file *RPC_Client.c*), il processo che realizza l'interazione con l'utente proponendo ciclicamente i servizi che utilizzano le due procedure remote, e stampa a video i risultati, fino alla fine del file di input da tastiera.

Parte Java

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

- inserire un nuovo provincino;
- scaricare i file dei provincini che corrispondono ad una parola chiave.

Si noti il vincolo di usare un'unica connessione per tutte le interazioni fra client e server.

Più in dettaglio:

- Il **client** è organizzato come un **processo ciclico fino alla fine del file di input**. 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 ogni sessione cliente**. Alla ricezione del 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.
Nel caso di **inserimento di un nuovo provincino**, il client richiede all'utente e invia al server il nome del provincino (verificandone il formato), il nome del file, la durata e le parole chiave (fino ad un massimo di K); quindi, riceve una conferma di inserimento nella struttura dati, invia il file corrispondente e stampa l'esito dell'operazione a video.
Nel caso di **scaricamento dei file dei provincini che corrispondono ad una parola chiave**, il client richiede all'utente e invia al server la parola chiave; quindi riceve i file che corrispondono alla parola chiave cercata salvandoli sul file system locale.
- Il **server** è organizzato come un **processo che gestisce in modo parallelo** l'interazione col client generando un figlio all'arrivo di ogni richiesta da un nuovo client. Il figlio gestisce la intera sessione di interazione con il cliente: con una prima lettura discrimina quale delle due funzionalità si è richiesta, poi gestisce opportunamente invii e ricezioni per l'operazione stessa, infine si pone in attesa di nuove richieste dallo stesso client sulla stessa connessione; alla lettura dell'EOF, il figlio termina e chiude la sessione di servizio.

Per ogni richiesta di **inserimento di un nuovo provincino**, il figlio riceve il nome del provincino, il nome del file, la durata e le parole chiave, effettua tutti i controlli sui dati di input ricevuti; quindi, invia una conferma di inserimento con successo nella struttura dati e riceve il contenuto del file audio, oppure invia una notifica di errore, ad esempio, se esiste già un provincino con lo stesso titolo o se il formato del nome del provincino è sbagliato. **Si considerino e si gestiscano possibili interferenze durante operazioni di inserimento parallele.**

Per ogni richiesta di **scaricamento dei file dei provincini che corrispondono ad una parola chiave**, il figlio riceve la parola chiave; quindi invia nome e file audio dei provincini che corrispondono alla parola chiave cercata.