

# Reti di Calcolatori T

## Reti di Calcolatori L-A

### Appello del 30/01/2015

#### Compito 2

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!!) e non un unico file compresso.

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.

\*\*\*\*\*

Si richiede la progettazione e la realizzazione di **servizi invocabili sul file system di una macchina server** (direttorio remoto), da parte di più clienti.

Ogni comando considera come direttorio di partenza il direttorio corrente, in cui siano stati lanciati rispettivamente il client e il server. Se per esempio il client richiede la cancellazione di un file (usando una notazione relativa) nel direttorio corrente, il server cerca di cancellare il file nel direttorio dove è stato lanciato.

I servizi considerano solo file di testo per realizzare le seguenti funzioni:

1. **lista dei file di un direttorio remoto e di tutti i sottodirettori (di primo livello) i cui nomi contengano un sequenza di 2 o più consonanti consecutive:** questa operazione richiede all'utente il nome del direttorio, quindi visualizza la lista dei file del direttorio e di ogni sottodirettorio il cui nome contiene una sequenza di due o più caratteri consecutivi alfabetici, consonanti maiuscole e minuscole (diversi da 'a', 'A', 'e', 'E', 'i', 'I', 'o', 'O', 'u', 'U');
2. **aggiunta di numerazione delle righe in un file di testo remoto:** questa operazione richiede all'utente il nome di un file, aggiunge all'inizio di ogni riga un intero progressivo, da 1 al numero di righe, e visualizza a video l'esito dell'operazione;
3. **eliminazione di tutte le occorrenze di caratteri vocali all'interno di un file di testo remoto:** questa operazione richiede all'utente il nome di un file remoto ed elimina dal file tutte le occorrenze di caratteri alfabetici vocali ('a', 'A', 'e', 'E', 'i', 'I', 'o', 'O', 'u', 'U', maiuscole o minuscole), visualizzando a video il numero di eliminazioni effettuate;
4. **trasferimento di tutti file di un direttorio remoto i cui nomi contengono almeno due caratteri minuscoli dal server al client:** questa operazione richiede all'utente il nome del direttorio, e trasferisce i file di quel direttorio il cui nome contiene due caratteri alfabetici minuscoli salvandoli sul direttorio del client.

Si richiede inoltre di **non** usare nella soluzione comandi Unix (es. il comando **ls**), ma solo primitive di sistema (es. **opendir()** in C) o funzioni di libreria (es. metodi della **classe File** in Java).

## Parte C

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

- ricevere la lista dei file di un direttorio remoto e di tutti i sottodirettori (di primo livello) i cui nomi contengano un sequenza di 2 o più consonanti consecutive;
- aggiungere la numerazione delle righe in un file di testo remoto.

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 **lista\_file** accetta come parametro d'ingresso una struttura contenente il nome di un direttorio, e restituisce una struttura contenente la lista dei nomi di file trovati con consonanti consecutive (fino a un massimo di 10). In caso di nessun file presente o di direttorio inesistente, prevedere una segnalazione di errore.
- La procedura **numerazione\_righe** accetta come parametro d'ingresso il nome del file e aggiunge all'inizio di ogni riga un intero progressivo, da 1 al numero di righe, per restituire un intero positivo con il numero di righe numerate ( $\geq 0$ ) in caso di successo, oppure -1 in caso di errori o file inesistente.

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 realizzi le operazioni remote per:

- eliminare tutte le occorrenze di caratteri vocali all'interno di un file di testo remoto,
- trasferire tutti file di un direttorio remoto i cui nomi contengono almeno due caratteri alfabetici minuscoli dal server al client.

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 filtro**, ossia ciclico che esegue 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 la intera sessione con il server; 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 **trasferimento di tutti file di un direttorio remoto i cui nomi contengono almeno due caratteri minuscoli dal server al client**, il client chiede all'utente e invia al server il nome del direttorio, quindi riceve nome e contenuto di tutti i file e li salva sul file system locale. Nel caso di **eliminazione di tutte le occorrenze di caratteri vocali all'interno di un file di testo remoto**, il client invia il nome del file e riceve l'esito del numero di eliminazioni stampandolo a video.
- Il **server** è organizzato come un **unico processo che gestisce in modo parallelo** l'interazione col client generando un figlio all'arrivo di ogni richiesta da un nuovo client. Il figlio che serve quella sessione con una prima lettura discrimina il tipo di funzionalità richiesto, poi gestisce opportunamente invii e ricezioni per 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 **trasferimento di tutti file di un direttorio remoto i cui nomi contengono almeno due caratteri minuscoli dal server al client**, il figlio riceve il nome del direttorio, quindi invia al client tutti i file i cui nomi contengono almeno due caratteri alfabetici minuscoli. Per ogni richiesta di **eliminazione di tutte le occorrenze di caratteri vocali all'interno di un file di testo remoto**, il figlio riceve il nome del file, quindi effettua l'operazione di eliminazione sul file locale e invia la risposta al client: un intero positivo che indica il numero di eliminazioni effettuate ( $\geq 0$ ) in caso di successo, -1 in caso di problemi.