

# Reti di Calcolatori T

## Appello del 08/01/2024

### Compito 2

Cognome: .....  
Nome: .....  
Matricola: .....

Tempo a disposizione: 3h

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. In entrambi gli esercizi, sia in Java che in C, si effettuino gli opportuni controlli sugli argomenti della richiesta e si gestiscano le interazioni e 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 la progettazione e la realizzazione di servizi invocabili da parte di più clienti sul file system di una macchina server remota (direttorio remoto). I file binari possono avere qualsiasi estensione (non \*.txt) e sono composti da dati binari. I file di testo sono i file con nome con suffisso \*.txt e composti solo da linee di caratteri ASCII stampabili (di lunghezza massima prefissata). In particolare, si realizzino i seguenti servizi:

1. **Conteggio delle linee che contengono almeno un certo numero di occorrenze di uno specificato carattere all'interno di tutti i file di testo presenti su un direttorio remoto:** questa operazione richiede all'utente il nome del direttorio, un carattere, e il numero di occorrenze, quindi scandisce ciascun file di testo riga per riga contando il numero di righe che contengono (in qualsiasi posizione della riga) un numero di occorrenze del carattere uguale o superiore a quelle indicate dall'utente; infine, restituisce a console l'esito dell'operazione.
2. **Eliminazione di tutte le occorrenze di singoli caratteri numerici all'interno di un file di testo:** questa operazione richiede all'utente il nome di un file, elimina tutte le occorrenze di caratteri numerici (da '0' a '9'), e visualizza sul cliente a video il numero di eliminazioni effettuate dal server.
3. **Lista dei file di testo di un direttorio remoto i cui nomi contengono almeno uno specificato numero di occorrenze di un carattere indicato dall'utente:** questa operazione richiede all'utente un nome di direttorio, un carattere e il numero di occorrenze, quindi visualizza la lista dei file il cui nome contiene un numero di occorrenze del carattere uguale o maggiore di quelle indicate dall'utente (ad esempio, se l'utente indica come carattere '0' e come occorrenze 3, il file '0B0C0.txt' viene riportato mentre i file '0DEF.txt' o il file 'XYZ00.txt' non sono riportati).
4. **Trasferimento dal server al client di tutti i file binari di un direttorio il cui nome contiene almeno un carattere numerico:** questa operazione richiede all'utente il nome del direttorio e trasferisce tutti i file binari in esso contenuti (i cui nomi abbiano almeno un carattere numerico e non abbiano suffisso \*.txt), salvandone il contenuto sul direttorio del client.

Ogni comando considera come direttorio di partenza il direttorio corrente del client e del server. Se ad esempio il cliente richiede la cancellazione di un file (usando una notazione relativa) nel direttorio corrente, il server cerca di cancellare il file nel direttorio da dove è stato lanciato. Si richiede inoltre di non usare nella soluzione comandi Unix (es. il comando ls), ma solo primitive di sistema (es. opendir() in C) e funzioni di libreria (es. metodi della classe File in Java).

## Parte Java

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

- eliminare tutte le occorrenze di caratteri numerici all'interno di un file di testo,
- ricevere la lista dei file di testo di un direttorio i cui nomi contengono almeno uno specificato numero di occorrenze di un carattere indicato dall'utente.

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 elimina\_occorrenze accetta come parametro d'ingresso il nome del file, elimina tutte le occorrenze di caratteri numerici (da '0' a '9') all'interno del file, e restituisce un intero positivo con il numero di eliminazioni effettuate (>=0) in caso di successo, oppure -1 in caso di insuccesso, ad esempio errori o altro.
- Il metodo lista\_file\_carattere accetta come parametri d'ingresso nome del direttorio, carattere e numero di occorrenze minimo, e restituisce la lista dei nomi dei file di testo trovati, se ci sono, il cui nome contiene un numero di occorrenze del carattere maggiore o uguale a quello specificato. In caso di direttorio inesistente, prevedere una segnalazione di errore.

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, propone 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:

- contare il numero delle linee che contengono almeno uno specificato numero di occorrenze di un carattere indicato all'interno di tutti i file di testo presenti sul direttorio remoto,
- trasferire dal server al client tutti i file binari di un direttorio dato il cui nome contiene almeno un carattere numerico,

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 trasferimento dal server al client di tutti i file binari con nome che contiene un numerico per un direttorio specificato utilizzando un'unica socket stream. Per ogni richiesta, il client chiede all'utente il nome del direttorio, quindi invia al server il nome del direttorio e riceve i file salvandone il contenuto sul direttorio del client.
- Il client\_datagram è organizzato come un processo filtro ciclico fino alla fine del file di input e realizza la funzionalità conteggio del numero di linee con almeno numero specificato di occorrenze del carattere all'interno di tutti i file di testo presenti sul direttorio (corrente) remoto. Per ogni richiesta, il client chiede all'utente e invia al server il carattere e il numero di occorrenze, quindi riceve l'esito e lo stampa a video.
- Il server multiservizio discrimina le richieste utilizzando la primitiva select: il server gestisce in modo parallelo la funzionalità di trasferimento dal server al client di tutti i file binari di un direttorio; mentre la funzionalità di conteggio del numero di linee con occorrenze del carattere all'interno di tutti i file di testo presenti sul direttorio (corrente) remoto è realizzata in modo seriale. Per ogni richiesta di trasferimento dal server al client per un direttorio di tutti i file binari con nomi con un numerico, il figlio legge il nome del direttorio, quindi attua un protocollo per inviare tutti i file binari trovati al client. Per ogni richiesta di conteggio del numero di linee con occorrenze del carattere all'interno dei file di testo del direttorio remoto, il server legge direttorio, carattere e numero di occorrenze, quindi esegue il conteggio, scandendo ciascun file di testo contando il numero di righe con almeno un numero di occorrenze del carattere pari (maggiore o uguale) a quelle indicate dall'utente, e invia la risposta al client, come intero positivo con il numero totale di occorrenze contate (>=0) in caso di successo, -1 in caso di problemi.