Università degli Studi di Bologna
Scuola di Ingegneria

Corso di
Reti di Calcolatori T

Esercitazione 3 (proposta)
Socket C senza e con connessione

Luca Foschini

Michele Solimando

Anno accademico 2016/2017

Socket senza connessione: Client

Sviluppare un'applicazione C/S che realizzi un servizio di **conteggio** dei **caratteri**, delle **parole** (si usi la definizione di parola data nell'esercitazione 1) e delle **linee** di un **file di testo** presente sul server remoto.

Il **Client** chiede ciclicamente all'utente **il nome del file remoto**, quindi invia al server un unico datagramma contenente il nome del file e attende dal server la risposta: **tre interi** che indicano il numero di caratteri, parole e linee, se il file esiste sul server, altrimenti **una notifica di errore**, nel caso il file remoto non esista. In ogni caso la risposta viene stampata a video.

Socket senza connessione: Server

Il **Server** **attende datagrammi** dal client, li riceve, estrae il nome del file; quindi, se il file esiste, effettua il conteggio e **invia al client triple** che indicano le occorrenze di caratteri, parole e linee; altrimenti invia al client **un messaggio di errore** (ad esempio -1).

Il server può essere realizzato come **server sequenziale** o **concorrente e parallelo**.

Socket con connessione: Raddoppio caratteri

Sviluppare un'applicazione C/S per il **raddoppio di un carattere** all'interno di un file di testo: il nuovo contenuto del file viene mandato al cliente che ottiene un nuovo stream.

L'operazione prevede **l'invio del contenuto di un file e di un carattere** dal client al server, **il raddoppio** (lato server) di tutte le occorrenze del carattere, e la **spedizione del nuovo contenuto** (con tutte le occorrenze raddoppiate) dal server al client.

Socket con connessione: C / S

Il **Client** chiede all'utente il **carattere** e il **nome del file**, invia i dati al server, e riceve il **file trasformato** stampandolo a video.

Il **Server** gestisce in modo parallelo la funzionalità di raddoppio del carattere nelle linee selezionate. Per ogni richiesta il figlio riceve **il carattere**, quindi riceve **il file**, effettua il filtraggio richiesto e **spedisce le linee indietro** al client.

NOTA BENE: il server **NON** deve salvare il file in locale, ma agisce direttamente sul flusso di input ri-direzionandolo in output modificato.

Proposta di estensione: Socket connesse per mget e mput

Si vuole abilitare il trasferimento di **un direttorio dal server al client** (multiple get - mget) e **dal client al server** (multiple put - mput) utilizzando la **stessa unica connessione**.

Intendiamo in particolare che si possano trasferire, usando una unica connessione, **direttori interi (i file contenuti nel direttorio) nelle due direzioni**.

Si vuole realizzare la funzionalità spostando i file da un cliente ad un servitore **usando i direttori correnti** in cui i due processi sono stati invocati.

Proposta di estensione: mget e mput

Il **Client**, dopo essersi connesso al server, chiede ripetutamente all'utente il **nome del direttorio**, e se la richiesta è di mget o mput.

Nel primo caso di mget, il cliente **riceve i file selezionati** (sia nome sia contenuto) o un'eventuale risposta negativa, se il direttorio non esista lato server. I file richiesti vengono salvati nel direttorio corrente sul processo ricevente (client) sovrascrivendo file esistenti che abbiano lo stesso nome.

Nel secondo caso di mput, il cliente **invia i file** (sia nome sia contenuto) al server che li deve memorizzare.

Si ricordi che si deve **utilizzare la stessa socket** per il trasferimento di tutti i file. Il cliente chiude la connessione solo al termine specificato dalla fine del file di input (filtro).

Proposta di estensione: mget e mput

Il **Server** attende una richiesta di connessione da parte del client, poi deve processare le richieste di mget o mput comandate dal cliente. Il server deve essere realizzato come **server concorrente** e **parallelo**. Si ricordi che si deve **utilizzare la stessa socket** per il trasferimento di tutti i file.

Il server riceve la richiesta della operazione.

Se è una mget, allora riceve il nome del direttorio richiesto, e invia, se disponibili, **i file richiesti** o un'eventuale **risposta negativa**,

Se è una mput, allora riceve i file (nomi e contenuto) e li scrive sul suo file system. I file richiesti vengono salvati nel direttorio corrente sul server sovra-scrivendo file esistenti che abbiano lo stesso nome.

Consegna

Chi vuole può inviare lo svolgimento ai docenti, con lo strumento specificato sul sito del corso.