

### 3° Esercitazione (da svolgere): Socket Java con connessione

Sviluppare un'applicazione C/S che effettui il **trasferimento di tutti i file di un direttorio dal server al client** (multiple get).

In particolare:

- il client chiede all'utente il *nome del direttorio*, relativo al direttorio corrente dove viene lanciato il server, si connette al server (con **java.net.Socket**), crea uno stream di output attraverso cui inviare la richiesta, e uno stream di input da cui *ricevere i file selezionati* o un'eventuale risposta negativa, nel caso in cui il direttorio non esista lato server. I file richiesti vengono salvati nel direttorio corrente sul client sovra-scrivendo file esistenti che abbiano lo stesso nome.
- il server attende una richiesta di connessione da parte del client (su **java.net.ServerSocket**), usa la socket (**java.net.Socket**) prodotta dalla connessione per creare uno stream di input da cui ricevere la richiesta, e uno stream di output su cui *inviare*, se disponibili, *i file richiesti* o un'eventuale risposta negativa, dopodichè chiude la connessione. Si noti che si **deve** utilizzare **la stessa socket** per il trasferimento di tutti i file.

### Note per la realizzazione

- Per il trasferimento dei file bisogna studiare un protocollo per la gestione del trasferimento multiplo. Ad esempio il server, **prima dell'invio del singolo file**, può inviare il **nome del file** e il **numero di byte** da cui il file è composto, quindi il server **invia lo stream di byte**. Il server segue questo protocollo per ogni file e utilizza **la medesima socket** per effettuare tutti gli invii. Il ciclo si ripete fino a quando tutti i file nel direttorio richiesto sono stati inviati, quindi il server notifica la fine delle trasmissioni chiudendo la comunicazione.
- Il server deve essere realizzato *come server concorrente e parallelo*. Per ogni nuova richiesta ricevuta il processo padre, dopo aver accettato la richiesta, attiva un processo figlio a cui affida il completamento del servizio richiesto.

## Proposta di estensione

### Trasferimento di più direttori

Si estenda il programma sviluppato in modo che gestisca il trasferimento di **più direttori** dal server al client.

**Protocollo:** si estenda il protocollo attuale in modo da abilitare il trasferimento di **diversi direttori** utilizzando la **stessa unica connessione dal cliente al server**.

Per ogni richiesta, il client dovrà creare in locale un direttorio con lo stesso nome di quello richiesto, dove verranno salvati i file inviati dal server, quindi dovrà salvare in tale direttorio i file in arrivo dal server, e mettersi in attesa di una nuova richiesta dell'utente.

Il server, invece, per ogni richiesta, deve inviare tutti i file del direttorio, e notificare la fine della trasmissione del direttorio.

Si gestisca inoltre la **terminazione dell'interazione fra client e server**: il client deve poter indicare al server la propria intenzione di chiusura dell'interazione. Una volta terminata la sessione client e server (*processo figlio del server principale*) terminano la propria esecuzione.

### Get e put coordinate

Si estenda ulteriormente il programma sviluppato in modo da abilitare il funzionamento del server in modalità sia **get** (trasferimento dal server al client) che **put** (trasferimento dal client al server).

**Protocollo:** si estendano i protocolli proposti, in modo da gestire la **sessione di lavoro fra client e server** utilizzando sempre una **stessa connessione**.

Per ogni richiesta, il client richiede all'utente il tipo della richiesta che viene inoltrata al server (put o get); poi, client e server si coordinano per portare a termine l'operazione richiesta. Al termine, il client si pone in attesa di una nuova richiesta dell'utente fino alla terminazione dell'interazione.

### Consegna

*Chi vuole può inviare via email lo svolgimento dell'estensione ai docenti, in modo da discutere la soluzione ed eventualmente pubblicarla sul sito del corso*