

Descrizione progetto Luciano Furore

Il progetto in questione si occupa della creazione di un Client ed un Server, il client si connette al server mediante una porta che gli viene aperta dal server, in seguito alla connessione il client puo' effettuare l'accesso come Anonimo o come utente registrato, nel caso in cui il client affettui l'accesso come Anonimo, il server dara' la possibilita' di : visualizzare la lista di file di tutti gli utenti anonimi e nel caso di scaricarlo; nel caso in cui il client decida di effettuare l'accesso come utente registrato, il server dara' una lista piu ampia di comandi tra cui: la possibilita' di rinominare il nome di un file, cancellare un file e di mandare un file presente sul disco del client al server, ogni utente registrato ha una propria cartella sul server.

Server:

La classe server ha 3 metodi principali:

1. Il metodo `attendi` che si occupa di inizializzare il servizio e mettersi in ascolto su una porta fino a quando un client non si colleghi.
2. Il metodo `accedi`, l'inizio dell'interazione tra l'utente ed il server che si occupa di lanciare i metodi per permettere l'accesso come utente o come anonimo.
3. Il metodo `comunica` che e' il cuore del server, in base al tipo di accesso effettuato restituisce al client la lista di comandi che puo' richiedere, attende una richiesta e dopo averla ricevuta mediante uno switch-case lancia il metodo per gestirla. Per evitare errori ogni volta che riceve una stringa contenente la richiesta, il testo viene convertito in minuscolo (`.toLowerCase`) in modo da evitare errori banali.

Client:

La classe client ha 3 metodi principali:

1. Il metodo `connetti` che serve per far connettere il client al socket server sulla una specifica porta.
2. il metodo `comunicazioneIniziale` che il primo metodo che mette in comunicazione l'utente con il server, chiede in che modo vuole accedere (Come utente registrato o come anonimo), dopo che l'utente gidita la scelta viene lanciato il metodo che poi manda la richiesta la server
3. il metodo `comunica` e' il cuore del client, attende che l'utente inserisca una richiesta, lancia il metodo corrispondente a quella richiesta mediante uno switch-case, il metodo corrispondete manda la richiesta al server.

Ogni volta che il client lancia il metodo per inviare la richiesta al server, il metodo mette come parametro una stringa contenente la parola della richiesta, in questo modo ogni metodo gestisce autonomamente l'intero processo.

Gestione degli anonimi:

Gli utenti Anonimi visualizzano solo la cartella generica per tutti gli utenti che effettuano l'accesso come anonimo e possono solo ed unicamente effettuare il download dei file oltre a poter fare il log-out.

Gestione degli utenti registrati:

Gli utenti registrati hanno tutti una propria cartella presente nel database del server, ogni utente che effettua il login ha la possibilità di visualizzare la lista di file presenti nella propria cartella, cancellare e rinominare i file ed inviare un nuovo file dal proprio disco al server.

Controllo dei dati degli utenti registrati:

Quando il client utilizza il comando login per effettuare l'accesso come utente registrato il server richiede un nome ed una password, dopo averle ricevute il server va nella cartella database, apre la cartella che corrisponde al nome che l'utente ha inserito, se non trova la cartella sa che il nome inserito non esiste e restituisce un errore, nel caso in cui la trovi, apre il file presente nella cartella e ne legge il contenuto, se il contenuto del file è uguale alla password inserita dall'utente imposterà i permessi root a 2 e dirà all'utente che ha effettuato l'accesso con successo.

Gestione dei Download:

Ogni volta che viene effettuato il download, il file scaricato viene salvato nella cartella Download del Client. Quando l'utente chiede il download il client ed il server stabiliranno una nuova connessione per gestire il trasferimento del flusso dati.

Gestione dell'upload:

Ogni volta che il client prova ad effettuare un upload di un file sulla propria cartella del server, prende il file dalla cartella DiscoDelClient. Quando l'utente chiede l'upload il client ed il server stabiliranno una nuova connessione per gestire il trasferimento del flusso dati.

Funzionamento Generico:

Ogni volta che il client ed il server completano una richiesta viene richiamato nuovamente il metodo comunica in modo da permettere al client di poter effettuare una nuova richiesta, solo nel caso in cui l'utente chieda il log-out sia il client che il server chiameranno il metodo per gestire il log-in.