Matricola:2048130

Nome: Matt Ramos

Introduzione

L'applicazione sviluppata consente il trasferimento di file tra un programma client (**myFTClient**) e un programma server (**myFTServer**) attraverso una connessione di rete. Il server è in ascolto continuo su un socket di rete e accetta richieste da più client, gestendo le operazioni di accesso ai file e garantendo la corretta esecuzione dei trasferimenti.

L'implementazione prevede due programmi scritti in linguaggio C e una libreria (**ftp_lib**) contenente funzioni comuni utilizzate da entrambi.

Struttura del programma

1. Il Client (myFTClient.c)

Il programma client (**myFTClient**) si occupa di connettersi al server e di inviare richieste di trasferimento file. Le principali operazioni gestite sono:

- **Interpretazione e gestione dei parametri in input**: utilizza getopt() per ottenere i parametri passati tramite riga di comando.
- **Connessione alla rete**: stabilisce la comunicazione con il server attraverso socket TCP/IP.
- **Scambio di messaggi con il server**: imposta e avvia l'operazione richiesta.
    - WRITE: richiesta di scrittura di un file dal client al server.
    - READ: richiesta di lettura di un file dal server al client.
    - LIST: richiesta di elencare i file presenti in una directory.
- **Gestione del trasferimento dati**:
    - Il client invia il nome del file al server.
    - In base all'operazione, avvia la trasmissione o la ricezione dei dati.
    - Implementa controlli sulla connessione e sulla disponibilità di spazio locale.

2. Il Server (myFTServer.c)

Il programma server (**myFTServer**) è progettato per gestire più connessioni in parallelo. È composto da due funzioni principali:

- **main()**:
    - Gestisce i parametri in input (indirizzo IP, porta, directory di lavoro).
    - Inizializza il socket di rete e si pone in ascolto di nuove connessioni.
    - Accetta le connessioni e crea un nuovo processo (con fork()) per ogni client.
- **handle_client()**:
    - Riceve richieste dai client e gestisce le operazioni richieste.
    - Si occupa di ricevere o inviare file, controllando lo stato della connessione e la presenza del file richiesto.
    - Implementa la gestione della directory per l'operazione LIST.

Libreria FTP (ftp_lib.c e ftp_lib.h)

La libreria ftp_lib include diverse funzioni di supporto utilizzate sia dal client che dal server:

- **waitingForFile()**: Verifica se un file è in uso da un altro processo e attende finché non diventa disponibile.
- **sendingData()**: Invia messaggi di stato tra client e server tramite socket.
- **creatingDirectories()**: Crea le directory lungo un percorso, se non esistono già.
- **socketClosed()**: Controlla se un socket è stato chiuso dal lato remoto utilizzando select().
- **checkSpaceAvailable()**: Verifica se c'è sufficiente spazio disponibile su disco prima di scrivere un file.
- **ends_with()**: Controlla se una stringa termina con un determinato suffisso.

Compilazione ed esecuzione

La compilazione dell'applicazione avviene separatamente per il server, il client e la libreria:

Compilazione della libreria:

```
gcc -c -fPIC ftp_lib.c -o ftp_lib.o
 gcc -shared -o libftp_lib.so ftp_lib.o
```

Compilazione del server e del client:

```
gcc -o myFTServer myFTServer.c -L. -lftp_lib
gcc -o myFTClient myFTClient.c -L. -lftp_lib
```

Conclusioni

Il progetto sviluppato fornisce un'implementazione robusta di un sistema FTP base, con supporto per operazioni di lettura, scrittura e visualizzazione dei file. La gestione della concorrenza nel server permette di servire più client simultaneamente, garantendo un trasferimento dati sicuro ed efficiente. La modularità introdotta dalla libreria ftp_lib consente di riutilizzare codice comune, migliorando la manutenzione e l'espandibilità del progetto.