

# Client/Server FTP

F. Bernaudo

14 maggio 2024

## Sommario

In questo progetto viene descritta la progettazione e l'implementazione di un'architettura client/server FTP.

## 1 Introduzione

In questo documento viene descritta la progettazione e l'implementazione di un'architettura client/server FTP. Si precisa che il progetto, seppur basato su un contesto reale, è solo una versione didattica che applica quelle che sono le metodologie di sviluppo di un progetto di questo tipo, e pertanto non saranno implementate tutte le funzionalità necessarie ad un funzionamento ottimale del prodotto.

Per informazioni aggiuntive riguardo l'implementazione si faccia riferimento ai commenti presenti nel codice sorgente.

## 2 Analisi dei Requisiti

Di seguito sono elencati i requisiti di progettazione:

- Utilizzare i comandi FTP definiti in RFC 959 per il canale PI.
- Stabilire una connessione Passiva.
- Il sistema deve permettere il login ed il logout di due tipologie di utenti: Utente registrato e Utente anonimo
- L'Utente registrato dovrà poter vedere solo la directory a lui assegnata (di cui potrà visualizzare il contenuto), potrà effettuare download e upload di files e potrà rinominare e cancellare files.
- L'utente anonimo potrà vedere solo la directory pubblica, la lista dei file contenuti e potrà effettuare il download.
- Utilizzare un file o una struttura dati hardcoded per gli utenti registrati.
- Non usare le porte riservate utilizzate solitamente da FTP.

Il sistema non dovrà contenere:

- Creazione di directory.
- Navigazione nel filesystem.
- Ridenominazione o rimozione delle directory.
- Servizio di registrazione
- Protocolli di sicurezza

### 2.1 Tecnologie utilizzate

Per lo sviluppo è stato scelto Python3 come linguaggio di programmazione, e nello specifico le librerie "os" per la gestione dei path, e "socket" per la gestione dei socket

## 2.2 Architettura FTP

L'architettura implementata prevede un canale di comunicazione utilizzato per i comandi(PI) e un canale di comunicazione usato per il trasferimento dei dati(DTP).

I comandi "RNFR", "RNT0", "QUIT", "DELE" fanno uso solo del PI.

I comandi "RETR", "LIST", "STOR" fanno uso anche del DTP.

La connessione per i dati è passiva, dato che, quando necessario, il server a creare il socket utilizzando una determinata porta per i dati, e poi invia l'informazione sul numero di porta al client per permettere la connessione. Al termine dell'operazione di trasferimento la connessione dati viene chiusa.

## 2.3 Autenticazione

Le credenziali per l'utente registrato sono hardcoded, e consistono in utente e password, che sono "admin" e "pass". Per quanto riguarda l'utenza anonima, basterà inserire come nome utente "anonymous" per potervi accedere. L'intero sistema si basa su una funzione di convalida credenziali e non sono stati implementati ulteriori protocolli di sicurezza.

Di seguito un esempio di funzionamento:

```
220 Welcome to FTP server

331 Please specify the username.

Username: admin
331 Please specify the password.

Password: pass
230 User logged in, proceed. your root directory is "documents"

Enter FTP command (e.g., LIST, RETR filename, QUIT): █
```

Figura 1: Auth.

## 2.4 Visibilità

Ogni utente può vedere esclusivamente la directory assegnata, che per l'utente registrato è "documents" e per l'utente anonimo è "public".

## 2.5 PI

Client e Server FTP comunicano attraverso due socket, una lato server e una lato client. L'host e le porte sono hardcoded, definite all'interno degli script e non sono state utilizzate porte di uso standard per FTP.

## 2.6 DTP

Client e Server FTP si scambiano dati attraverso due socket, una lato server e una lato client. le connessioni vengono aperte e chiuse solo per operazioni che richiedono il trasferimento di dati.

## 2.7 Operazione LIST

Eseguendo il comando "LIST nomedirectory" si potrà listare il contenuto della directory assegnata.

```
230 User logged in, proceed. your root directory is "documents"

Enter FTP command (e.g., LIST, RETR filename, QUIT): LIST documents
prova_2.txt
prova_35.txt
```

Figura 2: LIST.

## 2.8 Operazione RNFR - RNTTO

Eseguendo il comando "RNFR vecchionomefile" verrà inviata al server una richiesta, il server stesso poi resterà in attesa di ricevere un secondo comando. Eseguendo poi "RNTTO nuovonomefile" il server completerà l'operazione di ridenominazione.

```
Enter FTP command (e.g., LIST, RETR filename, QUIT): RNFR PROVA_1.txt
350 Requested file action pending further information.
Enter FTP command (e.g., LIST, RETR filename, QUIT): RNTTO prova_49.txt
250 File has been renamed.
```

Figura 3: RNFR - RNTTO.

## 2.9 Operazione DELE

Eseguendo il comando "DELE nomefile" il server cancellerà il file richiesto

```
Enter FTP command (e.g., LIST, RETR filename, QUIT): DELE prova_49.txt
250 File has been removed.
```

Figura 4: DELE.

## 3 Operazione RETR

Eseguendo il comando "RETR nomefile" verrà effettuato il download di nome file.

```
Enter FTP command (e.g., LIST, RETR filename, QUIT): RETR prova_2.txt
File 'PROVA_2.TXT' downloaded correctly.
□
```

Figura 5: RETR.

## 4 Operazione STOR

Eseguendo il comando "STOR filepath" si effettuerà l'upload di un certo file.

```
Enter FTP command (e.g., LIST, RETR filename, QUIT): STOR
250 File uploaded correctly.
```

Figura 6: STOR.