

Client/Server FTP

F. Bernaudo

18 giugno 2024

Sommario

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

Il codice di riferimento del progetto è 304y2z3qvak.

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 Requisiti di progettazione

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

3 Tecnologie utilizzate

Per lo sviluppo è stato scelto Python3 come linguaggio di programmazione, data la sua semplicità e il vasto numero di librerie esistenti. Nello specific, sono state usate le seguenti librerie, comprese nella standard library:

- OS: libreria di riferimento per operazioni riguardanti io sistema operativo, è stata utilizzata prevalentemente per la gestione dei path.
- SOCKET: libreria di riferimento per la gestione dei socket, è stata utilizzata per generare e gestire i socket utilizzati per interfacce di comunicazione della soluzione.

4 FTP: definizione

FTP (File Transfer Protocol), è un protocollo di comunicazione utilizzato per il trasferimento di file tra due entità. Esso è uno dei primi protocolli creati per la rete internet, allo scopo di incoraggiare il trasferimento remoto di file e renderlo affidabile. La prima specifica dello standard risale al 1971, ad opera del MIT e ,allo stato dell'arte, la specifica utilizzata è l'RFC 959, che è anche quella di riferimento per questo progetto.

5 Standard RFC 959

L'RFC 959 è l'attuale standard di riferimento per FTP. Definito nel 1985, il documento è composto da 8 capitoli e 3 appendici. Il contenuto di queste suddivisioni sarà brevemente descritto nelle sottosezioni seguenti.

5.1 Introduzione

Nel capitolo introduttivo viene definito l'FTP sottolineandone gli scopi e viene tracciata una timeline sull'evoluzione del protocollo e sugli standard che si sono susseguiti negli anni e che ne hanno documentato l'evoluzione, oltre che elencare alcuni termini utili alla comprensione dei capitoli successivi. Di seguito viene mostrata l'architettura FTP, di tipo client/server, come mostrato in figura:

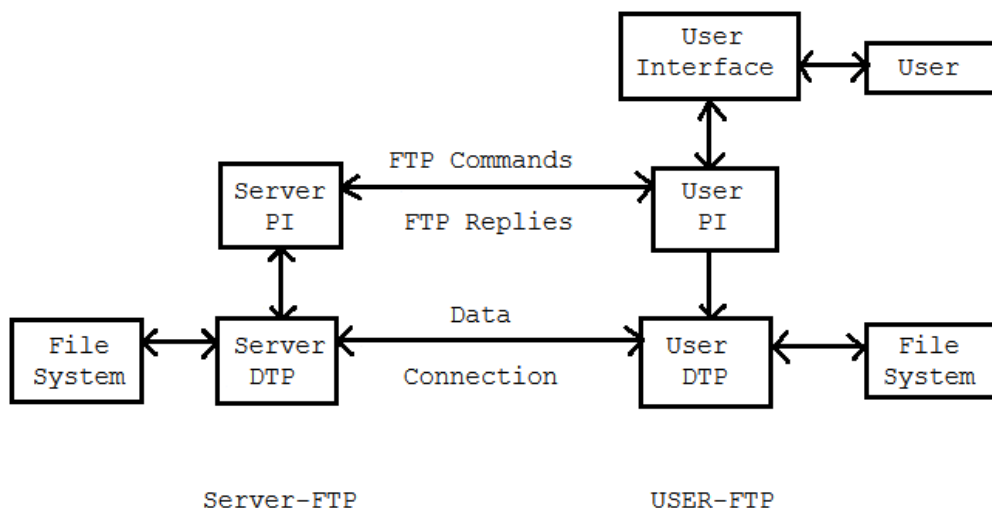


Figura 1: Architettura FTP.

Nel modello descritto in figura, il PI (Protocol Interpreter), che rappresenta l'interfaccia di comunicazione lato client, avvia la connessione lato client. La connessione di controllo segue il protocollo Telnet. Successivamente vengono trasmessi i comandi FTP dal PI - user al processo PI - server, I comandi FTP specificano la natura della connessione dati e il tipo di operazione da

eseguire. A seconda del tipo di connessione, viene implementata una procedura di creazione della connessione dati e di scambio di file. Il DTP (Data Transfer Process) gestisce la connessione dati che a differenza del PI, non è permanente ma viene creata e chiusa ad ogni trasferimento file. Altro punto fondamentale è che le connessioni PI devono restare aperte durante il trasferimento dati, ed è responsabilità del client richiederne la chiusura. Inoltre, se viene interrotta la connessione per i comandi durante un trasferimento dati, il server dovrà chiudere la connessione dati.

5.2 Data Transfer functions

Nel capitolo 3, oltre ad essere sottolineate ancora una volta le differenze tra PI e DTP, vengono descritte le modalità che possono essere utilizzate per il trasferimento dati. Vengono esplorate diverse tematiche quali :

- Modalità di trasferimento dati, fondamentali per capire come i dati vengono inviati durante una connessione FTP.
- Strutture dati, ovvero tutte le possibili strutture che i dati possono avere durante un trasferimento.
- Gestione delle connessioni dati, ovvero tutte le possibili modalità di creazione e gestione delle connessioni dati FTP.
- Trasmissione dati, ovvero l'effettiva trasmissione dei dati tra hosts.
- Gestione errori, ovvero la gestione di tutti gli errori che possono verificarsi durante il trasferimento dati.

5.3 File Transfer functions

Nel Capitolo 4 viene descritto tutto ciò che è relativo ai comandi che possono essere utilizzati durante il trasferimento file. Esso comprende:

- Comandi FTP: i comandi FTP effettivi che rappresentano le varie operazioni.
- Parametri: parametri che possono essere specificati per aggiungere dettagli a un certo comando.
- Codici di stato: codici di risposta che segnalano il successo o meno di una certa operazione.
- Gestione degli errori

5.4 Declarative Specifications

Il Capitolo 5 sono descritte ulteriori specifiche implementative atte a migliorare l'efficienza del trasferimento file.

5.5 State Diagrams

Nel capitolo 6 sono descritti i diagrammi di stato per una semplice implementazione del protocollo FTP. Ogni gruppo di comandi FTP o sequenze di comandi ha il proprio diagramma di stato, e i raggruppamenti dei comandi sono stati determinati creando un modello per ciascun comando e raggruppando insieme i comandi con modelli strutturalmente identici. Per ogni comando o sequenza di comandi, ci sono tre possibili esiti: successo (S), fallimento (F) ed errore (E), inoltre il simbolo "B" rappresenta lo stato "inizio" e il simbolo "W" rappresenta lo stato "in attesa di risposta".

5.6 Typical FTP Scenario

In questo capitolo viene descritto un tipico scenario di trasferimento file tramite protocollo FTP.

5.7 Connection Establishment

Nell'ultimo capitolo viene semplicemente specificato che la connessione di controllo viene stabilita tramite TCP tra la porta del processo utente e quella del processo server, e che il protocollo è assegnato alla porta 21.

5.8 Appendici

Le tre appendici che chiudono lo standard forniscono dettagli aggiuntivi su specifiche tematiche le quali sono:

- Informazioni sulla struttura delle pagine che possono essere trasferite mediante il protocollo.
- Informazioni sui comandi di gestione delle directories su un host remoto.
- Informazioni sulle varie RFC riguardanti il protocollo FTP.

6 Architettura e descrizione del progetto

L'architettura implementata prevede, come da standard, una connessione di controllo (PI), utilizzata per i comandi e permanente per il ciclo di vita della soluzione, e una connessione dati (DTP), creata all'occorrenza per ogni file da trasferire e chiusa alla fine del trasferimento.

Le due connessioni sono state create attraverso la library 'socket' di python. Le porte utilizzate non sono quelle standard per FTP. I comandi "RNFR", "RNTO", "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, è 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. Di seguito verranno illustrate le varie fasi e i comandi utilizzati, con allegato un esempio di esecuzione.

6.1 Autenticazione

Il sistema di autenticazione prevede due possibilità: l'accesso come utente registrato e l'accesso come utente anonimo. 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 di auth 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 2: Auth.

6.2 Visibilità

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

6.3 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.

6.4 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 e soltanto all'atto del trasferimento dati. Come specificato nelle sezioni precedenti, la connessione dati è Passiva.

6.5 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 3: LIST.

6.6 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 4: RNFR - RNTTO.

6.7 Operazione DELE

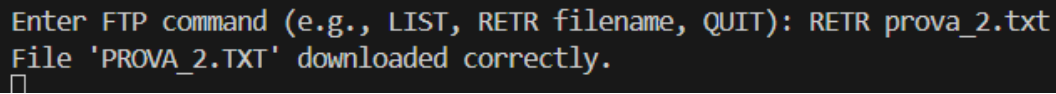
Eseguendo il comando "DELE nomefile" il server cancellerà il file richiesto

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

Figura 5: DELE.

7 Operazione RETR

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

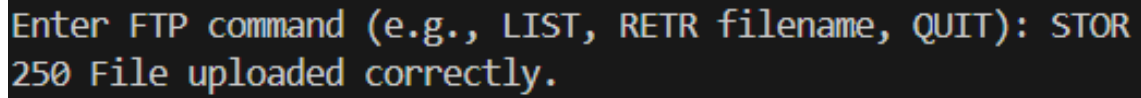


```
Enter FTP command (e.g., LIST, RETR filename, QUIT): RETR prova_2.txt
File 'PROVA_2.TXT' downloaded correctly.
□
```

Figura 6: RETR.

8 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 7: STOR.