Esame Reti di Calcolatori

Progetto 1. Implementazione di un Server FTP

Luca Cefaloni L-31 LUMSA [MAT. 27505/410]

# Prefazione

Il progetto che viene spiegato è l’implementazione di un server e di un client TCP, con la possibilità di eseguire dei comandi specifici per la lettura e scrittura dei file, in oltre anche la possibilità di utilizzare due processi simultaneamente.

# Server TCP

La creazione del server è diviso in tre parti:

1. **Gestione e creazione degli utenti virtuali**

Questa parte di codice determina gli user che possono accedere al server, insieme ai rispettivi permessi di scrittura e lettura.

Con il comando -> add\_user(Nome, password, directory, permessi)

1. **Interprete del protocollo TCP, Gestione comandi ricevuti dal client**

Definiamo l’interprete, la gestione dei comandi, e la risposta del server quando si collega un user

1. **Server accettatore**

Importiamo il modulo socket definendo indirizzo IPv4 e porta. Con il comando :

server.gethostbyname("127.0.0.1").

definiamo il server:

FTPServer(indirizzoIP e porta, interprete)

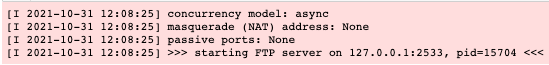
Diamo dei limiti relativi alle connessioni simultanee:

server.max\_cons = 256

il numero di connessioni accettate per lo stesso IP:

server.max\_cons\_per\_ip = 5

Eseguo il server con:

server.serve\_forever()

L’implementazione del server segue la rfc 959 che definisce il protocollo TCP

# Client TCP

L’implementazione del client invece segue un ciclo while con la possibilità di scegliere i comandi dati in consegna da eseguire sul server.

* **Connect**

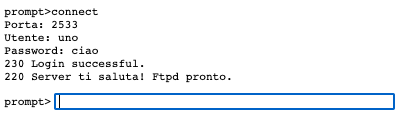
Esegue la connessione al server richiedendo porta utente e password, se non coincide con quelli definiti sul server ritorna alla scelta della lista dei comandi. Per eseguire la connessione al server il comando è server.connect(indirizzo IP, porta); invece per accedere è server.login(user, password).



Figura 1 Client

Figura 2 Server

* **Help**

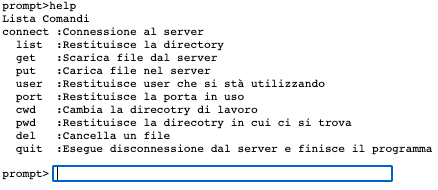
Stampa a schermo i possibili comandi

Figura 1Client

* **List**

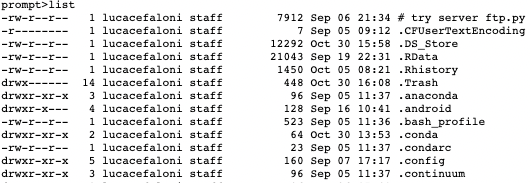
****Stampa la directory in cui si trova l’utente comando server.dir() {LIST}.

Figura 2 Client

* **Get**

Questa funzione scarica file dal server e per farlo il comando è: server.retrbinary(RETR con il nome del file, open(nome file, ‘wb’).write) {RETR}.

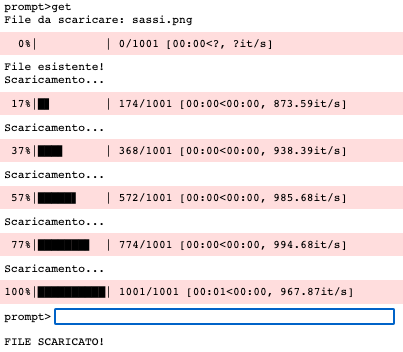


Figura 3 Client

Figura 4Server



* **Put**

Questo comando Carica file nel server, attraverso server.storbinary() dando il nome del file da caricare. {STOR}.

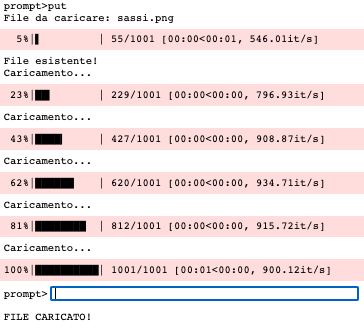




Figura 5Client

Figura 6Server

* **User**

Stampa a schermo l’user che ha fatto l’accesso al server.



Figura 7Client

* **Port**

Stampa a schermo la porta con cui l’user ha fatto l’accesso

Figura 8 Client

* **Cwd**

Cambia la directory di lavoro, il comando è: server.cwd(nome della directory in cui andare).

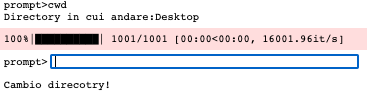


Figura 9Client



Figura 10Server

* **Pwd**

Restituisce la directory di lavoro, il comando per esguirlo è server.pwd()



Figura 11 Client

* **Del**

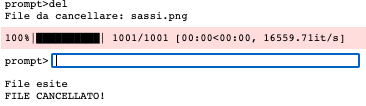
La unzione del elimina un file dal server, la funzione per eseguiro è server.delete



Figura 12Server

Figura 13Cilent

* Quit

La funzione quit esegue la disconnessione dal server:

server.quit()

Figura Server

Figura Client