

il path -o avra' root nella directory del server specificata con ft\_root\_directory"

Questa affermazione l'ho intesa come se -o fosse un path relativo che, concatenato con ft\_root\_directory formerebbe un path assoluto. **ft\_root\_directory + -o path = absolute path**

### legalità indirizzo IP

Per verificare la **sintassi** di un indirizzo IPv4 controllo se esso è composto da quattro ottetti (numeri) separati da punti. Ogni ottetto deve essere un numero compreso tra 0 e 255.

Per verificare se un indirizzo IP è **raggiungibile** o **esistente**, puoi utilizzare vari metodi. Uno dei metodi più comuni è utilizzare il comando **ping**, che invia pacchetti ICMP Echo Request a un indirizzo IP e attende risposte. Se l'indirizzo IP risponde, è raggiungibile.

Controllo solo la raggiungibilità poiché se raggiungibile significa anche che la sintassi è corretta.

### legalità porta

La **legalità** di una porta in termini di validità è determinata dal suo valore numerico. Una porta è considerata valida se è un numero intero compreso tra 1 e 65535 (controllo nel parsing degli argomenti sia del client che del server).

Per verificare se una porta è **raggiungibile** o **aperta** su un determinato indirizzo IP, si deve tentare di stabilire una connessione TCP (connect sul client).

### il server deve accettare connessioni da uno o piu' client...

Crea un nuovo **thread** per gestire la comunicazione con il client, in questo modo il processo del server continuerà a esistere anche dopo la chiusura della connessione con il client (elimino il thread e non il processo).

### ...e gestirle concorrentemente

**Array** di puntatori ai client connessi, con **mutex** per evitare race condition. Uso i mutex anche per gestire le operazioni (w,r,l) come sezione critica.

### controllo di non eccedere lo spazio su disco

### **WRITE**

Il client deve inviare un file al server.

Prima di inviare il file, nella funzione **write\_mode** al momento dell'apertura, controlla se c'è abbastanza spazio sul path in cui il server lo dovrebbe salvare.

E' il server che grazie ad una send, manda il percorso completo al client (**receive\_fullpath**), in questo modo può controllare lo spazio prima di inviare i dati.

## READ

Il server deve inviare un file al client.

Prima di inviare il file, nella funzione **handle\_read**, al momento dell'apertura, controlla se c'è abbastanza spazio sul path in cui il client lo dovrebbe salvare.

E' il client, che grazie ad una send, manda il percorso in cui verrà scritto il file al server (**receive\_path**) che in questo modo può controllare lo spazio prima di inviare i dati.

I send e receive da ambo i lati inviano il percorso usando la tecnica dei byte /0 all'inizio per evitare errori di comunicazione.

Richiesta di accesso a file non esistente (per la lettura) → file open: no such file or directory

myFTclient -w -a server\_address -p port -f local\_path/filename\_local

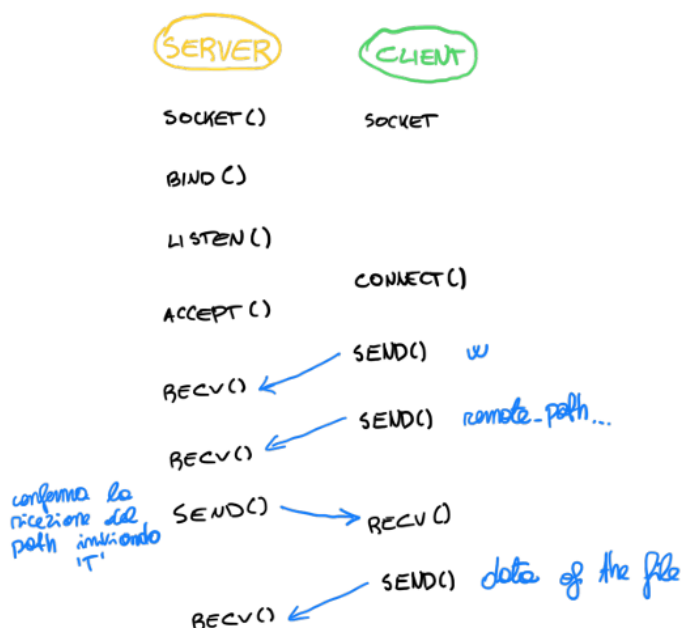
myFTclient -r -a server\_address -p port -f remote\_path/filename\_remote

L'ho inteso come se - o fosse:

- **ft\_root\_directory + local\_path/filename\_local.**
- **ft\_root\_directory + remote\_path/filename\_remote**

(Se questo percorso non esiste lo creo).

WRITE



READ

