

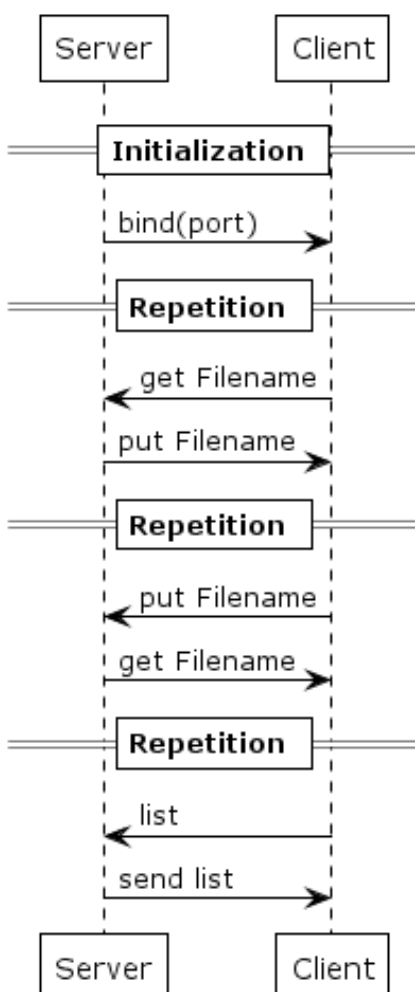
# Relazione programmazione di reti

Per questo progetto ho scelto di seguire la seconda traccia, realizzando un client e un server che comunicano seguendo il protocollo UDP, eseguendo varie transazioni come lo scambio dei file.

Tutti i file da inviare / ricevere devono stare nella stessa cartella degli script corrispondenti.

Il client può eseguire cinque comandi:

- **list** : il server manderà una lista dei file che sono disponibili
- **help** : stamperà a video la lista dei comandi che può eseguire
- **close** : chiuderà la connessione al socket ed il programma
- **get *filename*** : il server manderà il file richiesto, nel caso non disponibile manderà un errore
- **put *filename*** : manderà al server il file richiesto, nel caso non disponibile manderà un errore



Il Server rimane in attesa ciclando all'infinito fino a che non gli arriva una richiesta da parte di un client. Per ogni richiesta che arriva parte un thread, che la gestisce, permettendo perciò che vengano fatte più richieste in parallelo, evitando così di bloccarlo.

Per uscire della lettura di un file ho optato per inserire un timeout al socket pari a quattro secondi, per permettere ai file di essere trasferiti con più margine. Il buffer è stato settato al valore 1024 per ricevere i dati in fretta. Il server e il client hanno dei metodi in comune, come la `receive_file` e la `send_file`, ma ho optato di non creare una libreria in comune siccome nel mondo reale, il server e il client non staranno mai nello stesso posto.