

Final Report – CSE344

Mehmet Mahir Kayadelen
Student no. 210104004252

June 2023

1 Composition

For this project, I used two c file server.c and client.c one makefile for compile that for easy testability.

2 Program Logic

The server program acts as a file receiver, accepting connections from multiple clients and storing the received files in a server directory. The client program acts as a file uploader, connecting to the server and synchronizing a local directory with the server directory using the rsync command. Together, the server and client programs enable file transfer and synchronization between a central server and multiple clients.

Server side:

The server program is responsible for receiving files from multiple clients and storing them in a specified server directory.

It takes command-line arguments for the server directory, thread pool size, and port number. After it creates a server socket and binds it to the specified port. After listens for incoming client connections.

When a client connects, it accepts the connection and creates a child process to handle the file transfer.

Within the child process, it receives the file count from the client, receives each file individually, and stores them in the server directory. After successfully receiving all files, it closes the client connection and continues to accept new connections.

Client side:

The client program is responsible for synchronizing a local directory with the server directory using the rsync command.

It takes command-line arguments for the client directory and port number. It creates a client socket and connects to the server using the specified port. Once the connection is established, it calls the `upload_directory` function to synchronize the client directory with the server directory.

Within the `upload_directory` function, it forks a child process to execute the rsync command.

The child process executes the rsync command, which recursively synchronizes the directories, including deleting any files in the server directory that are not present in the client directory.

The parent process waits for the child process to complete and checks the exit status to determine the success or failure of the directory synchronization.

After the synchronization is complete, it closes the client socket.