DISTRIBUTED SINGLE FILE IMAGE SYSTEM

NAME: MUHAMMAD TARIQ AIJAZ

ERP: 09827

SUBJECT: DISTRIBUTED OPERATING SYSTEM

INSTRUCTOR: SIR SHABBIR MUKHI

PROGRAMMING LANGUAGE USED:

CLIENT COMMAND LIST:

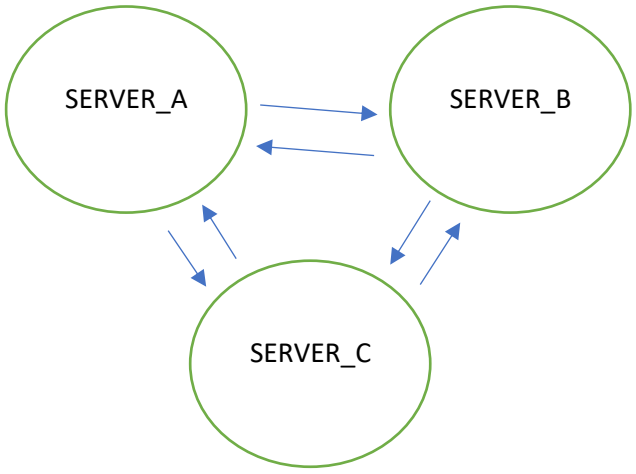
ls	Lists all contents of the current directory.
mkdir <directory name>	Creates a new directory.
mkfile <filename>	Creates a new empty file in the current directory.
cd <path>	Changes current directory to the specified path.
upload <filename>	Uploads a file to the current directory in the file system with the specified file name.
open <filename>	Opens the file in a text editor and then pushes the updates to the filesystem

NOTE: Both server and client take command line argument. 'A' to tell server and client that it is server_A and client_A. Similarly it goes for 'B' & 'C'.

ARCHITECTURE:

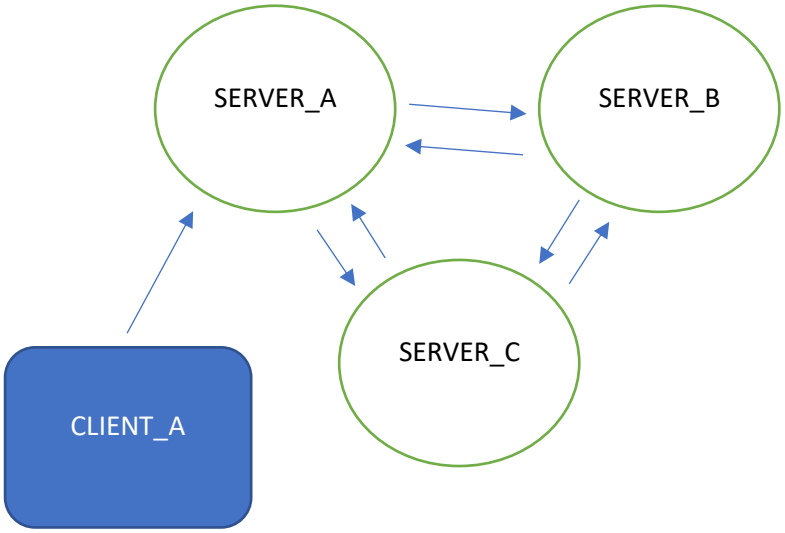
Server_to_Server Connections:

In this architecture, each node, i.e. is a TCP server, is connected to every other node in the system. Each node has its own maintained directory structure and all the files are distributed over the network on different nodes. A list of servers is provided to every other server so that they can connect with each other. When one server is up, it constantly sending request to the other servers whether they are up or down. Once another server is up it is connected automatically and the directory structure of that server that has been recently connected gets updated. Similarly, this happens with all the other servers as well.



Connection of a client:

Client initiates a connection with a server by giving the IP and Port number of that respective server. Fail to provide right IP and Port will result in socket connection error. Once the connection is made, the client can execute all the above-mentioned commands without any blocking mode because for every connection a client thread is made on server so that the server doesn't gets blocked whenever a new connection is established.



Configuration File:

The configuration file is placed with every server. It contains three main things, IP and port of itself, and connections of the server that it must connect with when they are up.

```
{
    "HOST": "localhost",
    "PORT":30000,
    "connections":{
        "Server_B": {
                    "ip": "localhost",
                    "port": 30001
                    },
        "Server_C": {
                    "ip": "localhost",
                    "port": 30002
                    }
    }
}
```

Directory Structure:

Each server has a directory structure that describes the current directory structure of the that node. When a server is up, it loads up its respective directory structure. Whenever a client makes a new file or makes a new directory, directory structure gets updated and send its own structure to other servers so that they can update their directory structure to maintain a single file system image for the client. At the same time replication of file is also done by the servers and update their directory structure.
In the following image, it explains the attributes of the directory structure.

Limitations:

- Some of the error handling on the server side may not be done.
- While doing replication, directory structure not updating properly but physically creating file on the servers.