Government Engineering College, Thrissur

CS334 – Network Programming Lab

Documentation -

Exp 6 – Concurrent File Server

Date of Submission

09 May 2021

Submitted By

Kowsik Nandagopan D

Roll No 31

TCR18CS031

GECT CSE S6

Experiment 6

Develop concurrent file server which will provide the file requested by client if it exists. If not, the server sends an appropriate message to the client. Server should also send its process ID (PID) to clients for display along with file or the message.

Executing program

- Code is provided in **Client.java** and **Server.java** along with this documentation. You can open the terminal in Linux (Ubuntu 20.04 tested). Then run the command

```
javac Server.java
java Server
```

- Open a **new terminal** and then run. We can *run as many clients in parallel*.

```
javac Client.java
java Client
```

- Server and Client will start their respective sockets and work on the same machine (IP 127.0.0.1, Port 5000).
- **Please save the files should accessible from client in *send* folder. To store the received files create a *receive* folder**
- **Note: Server is fully automated. Client has to send file name along with MIME type to the server.**
- We can see output side-by-side in the console

P. T. O

Output / Screenshots

Server

```
┌─hp@hp ~/Documents/S6/Network Lab/Exp6 ‹master*›
└─$ javac Server.java
┌─hp@hp ~/Documents/S6/Network Lab/Exp6 ‹master*›
└─$ tree .
.
├── Client.class
├── ClientHandler.class
├── Client.java
├── receive
├── send
│   ├── test.png
│   └── test.txt
├── Server.class
└── Server.java

2 directories, 7 files
┌─hp@hp ~/Documents/S6/Network Lab/Exp6 ‹master*›
└─$ java Server
Server socket created
Client1 Connected
Requested file test.png of size 464222 bytes
```

Client

```
┌─hp@hp ~/Documents/S6/Network Lab/Exp6 ‹master*›
└─$ javac Client.java
┌─hp@hp ~/Documents/S6/Network Lab/Exp6 ‹master*›
└─$ java Client
Client socket created
Enter file name (with MIME type): test.png
PID: 30335
0% 1% 2% 3% 4% 5% 6% 7% 7% 8% 9% 10% 11% 12% 13% 14% 14% 15% 16% 17% 18% 19% 2
0% 21% 22% 22% 23% 24% 25% 26% 27% 28% 29% 29% 30% 31% 32% 33% 34% 35% 36% 37%
 37% 38% 39% 40% 41% 42% 43% 44% 44% 45% 46% 47% 48% 49% 50% 51% 52% 52% 53% 5
4% 55% 56% 57% 58% 59% 59% 60% 61% 62% 63% 64% 65% 66% 67% 67% 68% 69% 70% 71%
 72% 73% 74% 74% 75% 76% 77% 78% 79% 80% 81% 82% 82% 83% 84% 85% 86% 87% 88% 8
9% 89% 90% 91% 92% 93% 94% 95% 96% 97% 97% 98% 99% 100%
Client Exit ...
┌─hp@hp ~/Documents/S6/Network Lab/Exp6 ‹master*›
└─$ tree .
.
├── Client.class
├── ClientHandler.class
├── Client.java
├── receive
│   └── test.png
├── send
│   ├── test.png
│   └── test.txt
├── Server.class
└── Server.java

2 directories, 8 files
┌─hp@hp ~/Documents/S6/Network Lab/Exp6 ‹master*›
└─$
```

Explanation:

1. Compile both source code
2. In server tree command we can see that **receive folder** is empty
3. Run Server
4. Run Client
5. Type the file name available in **send folder** in server. Else message in client will be display that such file doesn't exists in server. Here it is *test.png* is the file name.
6. After sending the file name we can see that server returns its PID. Then the actual file transfer occur. It is buffered file transfer so, we have to know how much is received. In order to show that we are printing percentage.
7. Tree command in client console shows that *test.png* is received