

CPSC 3780 - Course Project Tutorial

Group Members: Everett Blakley, Mitchell Sulz-Martin, Dallan McCarthy

This document outlines the steps for running the program outlined by the CPSC3780 Course Project. To begin, download or clone the source code from the project repository, found at the following link:

Project Repository: https://github.com/everettblakley/CPSC3780_project

This project was designed to run under two scenarios:

- **Scenario 1:** One server and one client
- **Scenario 2:** One server and many (≤ 5) clients

The following steps outline how to use the program under each scenario to transfer files:

Scenario 1

1. In the terminal, run `cd path/to/project/S1`
2. To start the server, run `cd server`, and then `make server`. The server terminal should display

```
g++ -std=c++11 -g -o server -I ../../include ../../src/*.cpp server.cpp
./server
Please enter the port you would like to start the server on:
```
3. Enter the port number (i.e. 30000). The server will attempt to open a socket on that port and the port right after (i.e. 30001). These ports will be used transmit ACK/NAK frames and DAT/NAT frames, respectively. If the sockets are successfully opened, the terminal will display:

```
Running server of ports 30000 and 30001...
```

If not, the terminal will display

```
Exception was caught:Error binding to port!
Exiting.
```
4. In another terminal (on any machine on the network), run `cd path/to/project/S1/client`
5. Run `make clean`, followed by `make client`. The client terminal should display

```
g++ -std=c++11 -g -o client -I ../../include ../../src/*.cpp client.cpp
./client
Enter the IP address of the server (or localhost):
```
5. Enter the IP address of the server (142.66.140.XX), or type `localhost` if the server is running locally. The client terminal should display

```
Please enter the server port number:
```
6. Enter the port number the server is running on (i.e. 30000). If successfully binding to that port, the terminal should display

```
Attempting to connect to [IP Address]

The available files for transfer are:
0. First_file.txt
1. Second_file.txt
etc...
Please select the file number to transfer:
```
7. Enter the index of the filename you wish to transfer. If there are no errors, the file should begin to transmit into the terminal over the network. The program will also output the file transferred to the `client` directory with the same name as the input file.
8. Repeat steps 4 - 7 to transfer another files

Scenario 2

1. In the terminal, run `cd path/to/project/S2`
2. To start the server, run `cd server`, and then `make server`. The server terminal should display

```
g++ -std=c++11 -g -lpthread -pthread -o server -I ../../include
../../src/*.cpp server.cpp
./server
Running server on ports 30000 and 30001...
```

Note that the port numbers have been hard-coded here (as compared to the input statement with S1). If there is an error binding to port, open `server.cpp` in and edit the `serverPort` variable at the top of the program

3. In another terminal (on any machine on the network), run `cd path/to/project/S2/client`
4. Run `make clean`, followed by `make client`. The client terminal should display

```
g++ -std=c++11 -g -o client -I ../../include ../../src/*.cpp client.cpp
./client
Enter the IP address of the server (or localhost):
```

5. Enter the IP address of the server (142.66.140.XX), or type `localhost` if the server is running locally. The client terminal should display

```
Please enter the server port number:
```

6. Enter the port number the server is running on (i.e. 30000). If successfully binding to that port, the terminal should display

```
Attempting to connect to [IP Address]
```

```
The available files for transfer are:
```

```
0. First_file.txt
```

```
1. Second_file.txt
```

```
etc...
```

```
Please select the file number to transfer:
```

7. Enter the index of the filename you wish to transfer. If there are no errors, the file should begin to transmit into the terminal over the network. The program will also output the file transferred to the `client` directory with the same name as the input file.
8. Repeat steps 4 - 7 to transfer another file