CS4150: Computer Networks Lab

Lab8

111901030
Mayank Singla

Q1. You are given a virtual network with three hosts h1(192.168.1.2), h2(192.168.1.3), and h3(192.168.1.4).

- **Create four TCP servers on h2. Their servers are ADD server (port number 12500), SUB server (port number 12501), MUL server (port number 12502), and IDIV server (port number 12503). If a client supplies two integers to these servers, they should perform the operation specified in their name. Note that IDIV stands for *integer division*.**
- **Upon successful computation, these servers should send the computed value to a UDP server on host h3. This UDP server on h3 is expected to square the value received from h2 and send the result to h1 over a TCP connection.**
- **Create a program on host h1 which takes three arguments. The first two are integers and the third is the operation that needs to be performed on these integers (ADD, SUB, MUL, or IDIV). Then, depending on the operation specified, your program should send the computation request to one of the servers on host h2 and display the result it gets from h3.**

Note: In your implementation, information about h1 such as IP address and server port numbers should not be hard coded in the programs running on hosts h2 and h3

Classes Summary:

TCPServer:

TCPServer()	// Constructor, will create and bind the socket
getSockFD()	// Returns the socket file descriptor of the server
setSockFD()	// Set the socket file descriptor of the server
getPort()	// Get the port on which the server is listening
getIP()	// Get the IP address of the server
getServerAddrInfo()	// Get the address info of the server
startListening()	// Start listening on the server
acceptConnection()	// Accept Connection from the client
sendData()	// Send Data to the client given its socket FD
receiveData()	// Retrieve data from the client given its socket FD
static get_in_addr()	// Returns the `sockaddr_in` struct
static getIP()	// Returns the IP address from an address information
static getPort()	// Function to return the Port from an address information

TCPClient:

```
TCPClient()             // Constructor, will create socket and connect to the server
getSockFD()             // Returns the socket file descriptor of the server
setSockFD()             // Set the socket file descriptor of the server
getPort()               // Get the port on which the server is listening
getServerIP()           // Get the IP address of the server
getServerAddrInfo()     // Get the address info of the server
sendData()              // Send Data to the client given its socket FD
receiveData()           // Retrieve data from the client given its socket FD
static get_in_addr()    // Returns the `sockaddr_in` struct
static getIP()          // Returns the IP address from an address information
static getPort()        // Function to return the Port from an address information
```

UDPServer:

```
UDPServer()             // Constructor, will create and bind the socket
getSockFD()             // Returns the socket file descriptor of the server
setSockFD()             // Set the socket file descriptor of the server
getPort()               // Get the port on which the server is listening
getIP()                 // Get the IP address of the server
getServerAddrInfo()     // Get the address info of the server
sendData()              // Send Data to the client given its socket FD
receiveData()           // Retrieve data from the client given its socket FD
static get_in_addr()    // Returns the `sockaddr_in` struct
static getIP()          // Returns the IP address from an address information
static getPort()        // Function to return the Port from an address information
```

UDPClient:

```
UDPClient()             // Constructor, will create socket and connect to the server
getSockFD()             // Returns the socket file descriptor of the server
setSockFD()             // Set the socket file descriptor of the server
getPort()               // Get the port on which the server is listening
getServerIP()           // Get the IP address of the server
getServerAddrInfo()     // Get the address info of the server
sendData()              // Send Data to the client given its socket FD
receiveData()           // Retrieve data from the client given its socket FD
static get_in_addr()    // Returns the `sockaddr_in` struct
static getIP()          // Returns the IP address from an address information
static getPort()        // Function to return the Port from an address information
```

Functions

stringToLong() // To convert a numeric string to long long integer
endProg() // To end the program execution on receiving invalid arguments
sigchld_handler() // Handler for SIGCHLD exception which restores **errno** for **perror**

compute.cpp

I parse all the arguments received from the command line. Then, I created objects of the class **TCPServer** and **TCPClient**. I started listening on the server. Then, using the client object, H1 sends the required data to the corresponding server on H2, and using the server object, it waits for the connection from H3 and accepts it, and retrieves the data from H3 to show the final result. H1 is sending the message to H2 in the form: **`PORT|NUM1|NUM2`**

add.cpp / sub.cpp / mul.cpp / idiv.cpp

I created the function **processData()** which will process the data received from H1, perform the corresponding operation as per the file, and construct the data to be sent to H3. H2 will find the IP address of H1 and will send the data to H3 in the form: **`IP|PORT|RESULT.`** Then, I am creating objects of the class **TCPServer** and **UDPClient** and started listening on the server. Then, in an infinite loop, H2 will wait for connections to be accepted using the server object, and then it will read the data from the incoming connection. Then, after processing the data received, it will send the data in the required format to H3 using the client object.

square.cpp

I created the function **extractData()** which will extract the required fields from the data received from H2 and also square the result obtained. Then, I created an object of the class **UDPServer** and in an infinite loop, H3 waits for receiving the data using the server object. On receiving the data, it parses it and extracts the IP address and port of H1 using which, it creates an object of the class **TCPClient** and sends the result back to H1 using this object.

Here is an example of testing for all the services.

Starting all the services on H2

```
tc@h2:~$ g++ add.cpp -o add
tc@h2:~$ ./add
server: successfully created a TCP socket at 0.0.0.0:12500
client: successfully created a UDP socket for 192.168.1.4:4950
server: waiting for connections at 0.0.0.0:12500
```

```
tc@h2:~$ g++ sub.cpp -o sub
tc@h2:~$ ./sub
server: successfully created a TCP socket at 0.0.0.0:12501
client: successfully created a UDP socket for 192.168.1.4:4950
server: waiting for connections at 0.0.0.0:12501
```

```
tc@h2:~$ g++ mul.cpp -o mul
tc@h2:~$ ./mul
server: successfully created a TCP socket at 0.0.0.0:12502
client: successfully created a UDP socket for 192.168.1.4:4950
server: waiting for connections at 0.0.0.0:12502
```

```
tc@h2:~$ g++ idiv.cpp -o idiv
tc@h2:~$ ./idiv
server: successfully created a TCP socket at 0.0.0.0:12503
client: successfully created a UDP socket for 192.168.1.4:4950
server: waiting for connections at 0.0.0.0:12503
```

Starting the **square** service on H3

```
tc@h3:~$ g++ square.cpp -o square
tc@h3:~$ ./square
server: successfully created a UDP socket at 0.0.0.0:4950
```

Test ADD

```
tc@h1:~$ g++ compute.cpp -o compute
tc@h1:~$ ./compute 2 4 ADD
server: successfully created a TCP socket at 0.0.0.0:3490
server: waiting for connections at 0.0.0.0:3490
client: successfully connected to 192.168.1.3:12500
server: got connection from 192.168.1.4:1025
Message from client: 36
The result obtained: 36
tc@h1:~$
```

```
tc@h2:~$ g++ add.cpp -o add
tc@h2:~$ ./add
server: successfully created a TCP socket at 0.0.0.0:12500
client: successfully created a UDP socket for 192.168.1.4:4950
server: waiting for connections at 0.0.0.0:12500
server: got connection from 192.168.1.2:513
Message from client: 3490|2|4
```

```
tc@h3:~$ g++ square.cpp -o square
tc@h3:~$ ./square
server: successfully created a UDP socket at 0.0.0.0:4950
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|6"
client: successfully connected to 192.168.1.2:3490
```

Test SUB

```
tc@h1:~$ ./compute 3 6 SUB
server: successfully created a TCP socket at 0.0.0.0:3490
server: waiting for connections at 0.0.0.0:3490
client: successfully connected to 192.168.1.3:12501
server: got connection from 192.168.1.4:1025
Message from client: 9
The result obtained: 9
tc@h1:~$
```

```
tc@h2:~$ g++ sub.cpp -o sub
tc@h2:~$ ./sub
server: successfully created a TCP socket at 0.0.0.0:12501
client: successfully created a UDP socket for 192.168.1.4:4950
server: waiting for connections at 0.0.0.0:12501
server: got connection from 192.168.1.2:513
Message from client: 3490|3|6
```

```
tc@h3:~$ g++ square.cpp -o square
tc@h3:~$ ./square
server: successfully created a UDP socket at 0.0.0.0:4950
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|6"
client: successfully connected to 192.168.1.2:3490
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|-3"
client: successfully connected to 192.168.1.2:3490
```

Test MUL

```
tc@h1:~$ ./compute 5 5 MUL
server: successfully created a TCP socket at 0.0.0.0:3490
server: waiting for connections at 0.0.0.0:3490
client: successfully connected to 192.168.1.3:12502
server: got connection from 192.168.1.4:1025
Message from client: 625
The result obtained: 625
tc@h1:~$
```

```
tc@h2:~$ g++ mul.cpp -o mul
tc@h2:~$ ./mul
server: successfully created a TCP socket at 0.0.0.0:12502
client: successfully created a UDP socket for 192.168.1.4:4950
server: waiting for connections at 0.0.0.0:12502
server: got connection from 192.168.1.2:513
Message from client: 3490|5|5
```

```
tc@h3:~$ g++ square.cpp -o square
tc@h3:~$ ./square
server: successfully created a UDP socket at 0.0.0.0:4950
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|6"
client: successfully connected to 192.168.1.2:3490
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|-3"
client: successfully connected to 192.168.1.2:3490
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|25"
client: successfully connected to 192.168.1.2:3490
```

Test IDIV

```
tc@h1:~$ ./compute 7 3 IDIV
server: successfully created a TCP socket at 0.0.0.0:3490
server: waiting for connections at 0.0.0.0:3490
client: successfully connected to 192.168.1.3:12503
server: got connection from 192.168.1.4:1025
Message from client: 4
The result obtained: 4
tc@h1:~$
```

```
tc@h2:~$ g++ idiv.cpp -o idiv
tc@h2:~$ ./idiv
server: successfully created a TCP socket at 0.0.0.0:12503
client: successfully created a UDP socket for 192.168.1.4:4950
server: waiting for connections at 0.0.0.0:12503
server: got connection from 192.168.1.2:513
Message from client: 3490|7|3
```

```
tc@h3:~$ g++ square.cpp -o square
tc@h3:~$ ./square
server: successfully created a UDP socket at 0.0.0.0:4950
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|6"
client: successfully connected to 192.168.1.2:3490
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|-3"
client: successfully connected to 192.168.1.2:3490
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|25"
client: successfully connected to 192.168.1.2:3490
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|2"
client: successfully connected to 192.168.1.2:3490
```

Q2. In the above exercise, all four servers are active all the time. Create a simple TCP *inetd daemon* on h2 which will create instances of these servers based on demand. Note that you may need to modify the servers you have already created on h2 to work with the *inetd daemon*.

New Class:

TCPService:

string name	// name of the service (ADD|SUB|MUL|IDIV)
long long port	// port on which the service is running
string pathToExec	// path to the executable for the service
TCPServer *server	// TCPServer object for the service
TCPService()	// Constructor

inetd.cpp

I created the function **createServices()** which will return the vector of **TCPService** objects for all the services (ADD|SUB|MUL|IDIV). I created objects of **fd_set** to keep track of the socket file descriptors for the **select()** system call. I created all the services and created a **TCPServer** object for all the services and started listening to all of them. Then, I added the socket file descriptor of all the listeners to the **fd_set**. Then, in an infinite loop, I call the **select()** system call to wait for any file descriptor to become ready to read. **A TCP socket file descriptor will become ready to read when it is ready to accept a connection from the client.** Then, I checked the socket file descriptor of which service is ready to read and accepted the connection on that service. Then, I computed the IP address of the client using the static **getIP()** function of the class and created a child process using the **fork()** system call. In the child process, I copied the socket file descriptor of the connection to STDIN. I am not copying to STDOUT and STDERR as I want my sub-services to print the messages on the terminal. Then, I executed the executable for the sub-service using the **execv()** system call.

inetd_add.cpp / inetd_sub.cpp / inetd_mul.cpp / inetd_idiv.cpp

I added a new constructor to the **TCPServer** class which takes the socket file descriptor as an argument as this time, the server was already created in the **inetd.cpp**, we need to create a dummy server object with that socket file descriptor. In the main function, I am reading the client IP address from the command line which was passed as an argument by the **inetd** daemon. Then, I created the server object with socket file descriptor **STDIN_FILENO** as it was copied there for the child process by the **inetd** daemon. Then, I created an object of the class **UDPClient**. Then using the server object, I read the data received from H1 and sent the processed data to H3 using the client object.

Here is an example of testing for all the services.

Starting the **inetd** daemon on H2.

```
tc@h2:~$ g++ inetd.cpp -o inetd
tc@h2:~$ g++ inetd_add.cpp -o inetd_add
tc@h2:~$ g++ inetd_sub.cpp -o inetd_sub
tc@h2:~$ g++ inetd_mul.cpp -o inetd_mul
tc@h2:~$ g++ inetd_idiv.cpp -o inetd_idiv
tc@h2:~$ ./inetd
server: successfully created a TCP socket at 0.0.0.0:12500
server: successfully created a TCP socket at 0.0.0.0:12501
server: successfully created a TCP socket at 0.0.0.0:12502
server: successfully created a TCP socket at 0.0.0.0:12503
server: waiting for connections at 0.0.0.0:12500
server: waiting for connections at 0.0.0.0:12501
server: waiting for connections at 0.0.0.0:12502
server: waiting for connections at 0.0.0.0:12503
```

Starting the **square** service on H3

```
tc@h3:~$ g++ square.cpp -o square
tc@h3:~$ ./square
server: successfully created a UDP socket at 0.0.0.0:4950
```

Test ADD

```
tc@h1:~$ g++ compute.cpp -o compute
tc@h1:~$ ./compute 2 3 ADD
server: successfully created a TCP socket at 0.0.0.0:3490
server: waiting for connections at 0.0.0.0:3490
client: successfully connected to 192.168.1.3:12500
server: got connection from 192.168.1.4:1025
Message from client: 25
The result obtained: 25
tc@h1:~$
```

```
tc@h2:~$ ./inetd
server: successfully created a TCP socket at 0.0.0.0:12500
server: successfully created a TCP socket at 0.0.0.0:12501
server: successfully created a TCP socket at 0.0.0.0:12502
server: successfully created a TCP socket at 0.0.0.0:12503
server: waiting for connections at 0.0.0.0:12500
server: waiting for connections at 0.0.0.0:12501
server: waiting for connections at 0.0.0.0:12502
server: waiting for connections at 0.0.0.0:12503
server: got connection from 192.168.1.2:513
client: successfully created a UDP socket for 192.168.1.4:4950
Message from client: 3490|2|3
```

```
tc@h3:~$ ./square
server: successfully created a UDP socket at 0.0.0.0:4950
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|5"
client: successfully connected to 192.168.1.2:3490
```

Test SUB

```
tc@h1:~$ ./compute 6 8 SUB
server: successfully created a TCP socket at 0.0.0.0:3490
server: waiting for connections at 0.0.0.0:3490
client: successfully connected to 192.168.1.3:12501
server: got connection from 192.168.1.4:1025
Message from client: 4
The result obtained: 4
tc@h1:~$
```

```
tc@h2:~$ ./inetd
server: successfully created a TCP socket at 0.0.0.0:12500
server: successfully created a TCP socket at 0.0.0.0:12501
server: successfully created a TCP socket at 0.0.0.0:12502
server: successfully created a TCP socket at 0.0.0.0:12503
server: waiting for connections at 0.0.0.0:12500
server: waiting for connections at 0.0.0.0:12501
server: waiting for connections at 0.0.0.0:12502
server: waiting for connections at 0.0.0.0:12503
server: got connection from 192.168.1.2:513
client: successfully created a UDP socket for 192.168.1.4:4950
Message from client: 3490|2|3
server: got connection from 192.168.1.2:513
client: successfully created a UDP socket for 192.168.1.4:4950
Message from client: 3490|6|8
```

```
tc@h3:~$ ./square
server: successfully created a UDP socket at 0.0.0.0:4950
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|5"
client: successfully connected to 192.168.1.2:3490
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|-2"
client: successfully connected to 192.168.1.2:3490
```

Test MUL

```
tc@h1:~$ ./compute 5 6 MUL
server: successfully created a TCP socket at 0.0.0.0:3490
server: waiting for connections at 0.0.0.0:3490
client: successfully connected to 192.168.1.3:12502
server: got connection from 192.168.1.4:1025
Message from client: 900
The result obtained: 900
tc@h1:~$
```

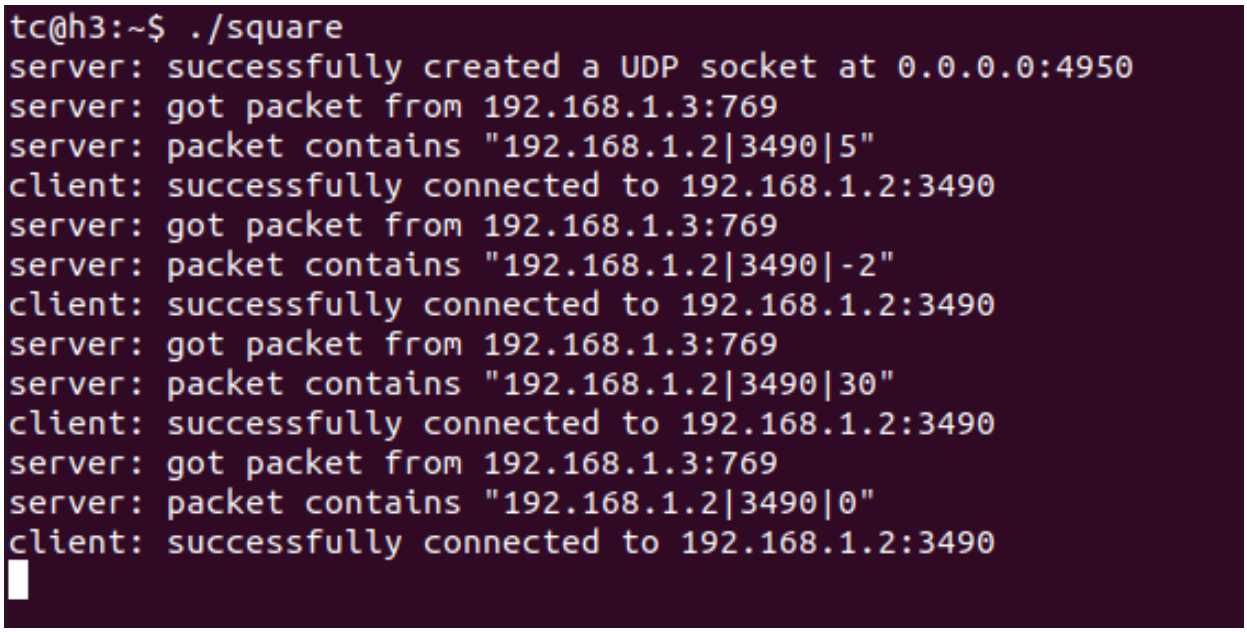
```
tc@h2:~$ ./inetd
server: successfully created a TCP socket at 0.0.0.0:12500
server: successfully created a TCP socket at 0.0.0.0:12501
server: successfully created a TCP socket at 0.0.0.0:12502
server: successfully created a TCP socket at 0.0.0.0:12503
server: waiting for connections at 0.0.0.0:12500
server: waiting for connections at 0.0.0.0:12501
server: waiting for connections at 0.0.0.0:12502
server: waiting for connections at 0.0.0.0:12503
server: got connection from 192.168.1.2:513
client: successfully created a UDP socket for 192.168.1.4:4950
Message from client: 3490|2|3
server: got connection from 192.168.1.2:513
client: successfully created a UDP socket for 192.168.1.4:4950
Message from client: 3490|6|8
server: got connection from 192.168.1.2:513
client: successfully created a UDP socket for 192.168.1.4:4950
Message from client: 3490|5|6
```

```
tc@h3:~$ ./square
server: successfully created a UDP socket at 0.0.0.0:4950
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|5"
client: successfully connected to 192.168.1.2:3490
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|-2"
client: successfully connected to 192.168.1.2:3490
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|30"
client: successfully connected to 192.168.1.2:3490
```

Test IDIV

```
tc@h1:~$ ./compute 6 10 IDIV
server: successfully created a TCP socket at 0.0.0.0:3490
server: waiting for connections at 0.0.0.0:3490
client: successfully connected to 192.168.1.3:12503
server: got connection from 192.168.1.4:1025
Message from client: 0
The result obtained: 0
tc@h1:~$
```

```
tc@h2:~$ ./inetd
server: successfully created a TCP socket at 0.0.0.0:12500
server: successfully created a TCP socket at 0.0.0.0:12501
server: successfully created a TCP socket at 0.0.0.0:12502
server: successfully created a TCP socket at 0.0.0.0:12503
server: waiting for connections at 0.0.0.0:12500
server: waiting for connections at 0.0.0.0:12501
server: waiting for connections at 0.0.0.0:12502
server: waiting for connections at 0.0.0.0:12503
server: got connection from 192.168.1.2:513
client: successfully created a UDP socket for 192.168.1.4:4950
Message from client: 3490|2|3
server: got connection from 192.168.1.2:513
client: successfully created a UDP socket for 192.168.1.4:4950
Message from client: 3490|6|8
server: got connection from 192.168.1.2:513
client: successfully created a UDP socket for 192.168.1.4:4950
Message from client: 3490|5|6
server: got connection from 192.168.1.2:513
client: successfully created a UDP socket for 192.168.1.4:4950
Message from client: 3490|6|10
```


tc@h3:~$ ./square
server: successfully created a UDP socket at 0.0.0.0:4950
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|5"
client: successfully connected to 192.168.1.2:3490
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|-2"
client: successfully connected to 192.168.1.2:3490
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|30"
client: successfully connected to 192.168.1.2:3490
server: got packet from 192.168.1.3:769
server: packet contains "192.168.1.2|3490|0"
client: successfully connected to 192.168.1.2:3490