Socket Programming Project

COMPE 560

Date: 11/10/2021

Omar Alshatti, 823420787

Dante Gonzalez Corbett, 822722869

Overview(Source Code at End and attached in submission):

In this project, we will be creating three programs, two client programs and one server program. The aim is to have firsthand experiencing with creating sockets and sending data through a TCP connection. We will be sending a C struct from our first client that has a character, integer, and floating number to the server. Then, it will modify these values and be send to the 2nd client which will print it then terminate the connection. We will be using the Jason server as our server, the Volta server as our first client, and a Ubuntu machine running in virtual machine as our 2nd client.

Body:

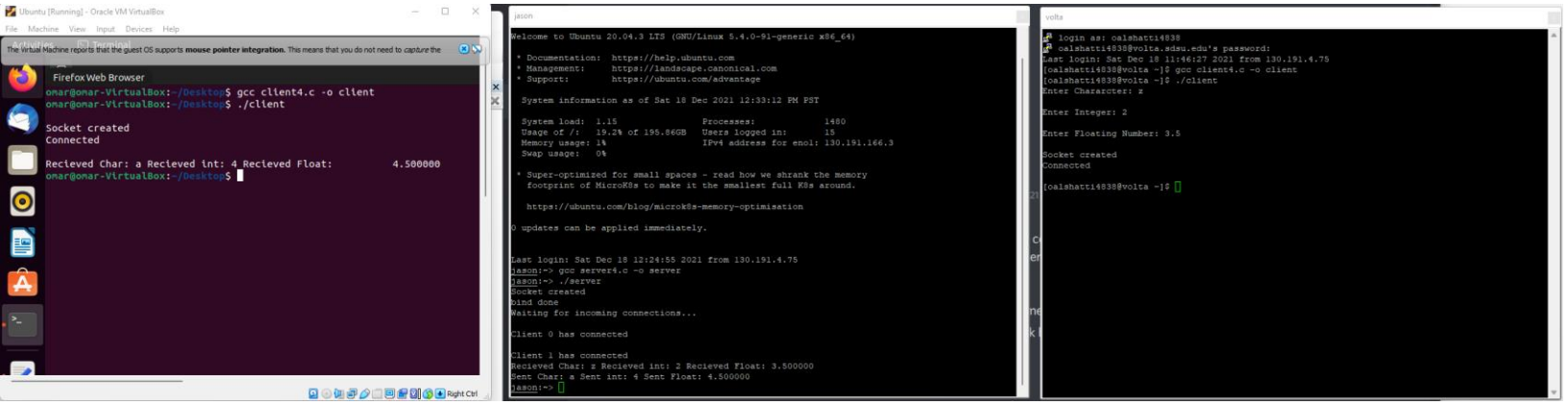

Figure 1(All Screens beside each other to show that they were run simultaneously)

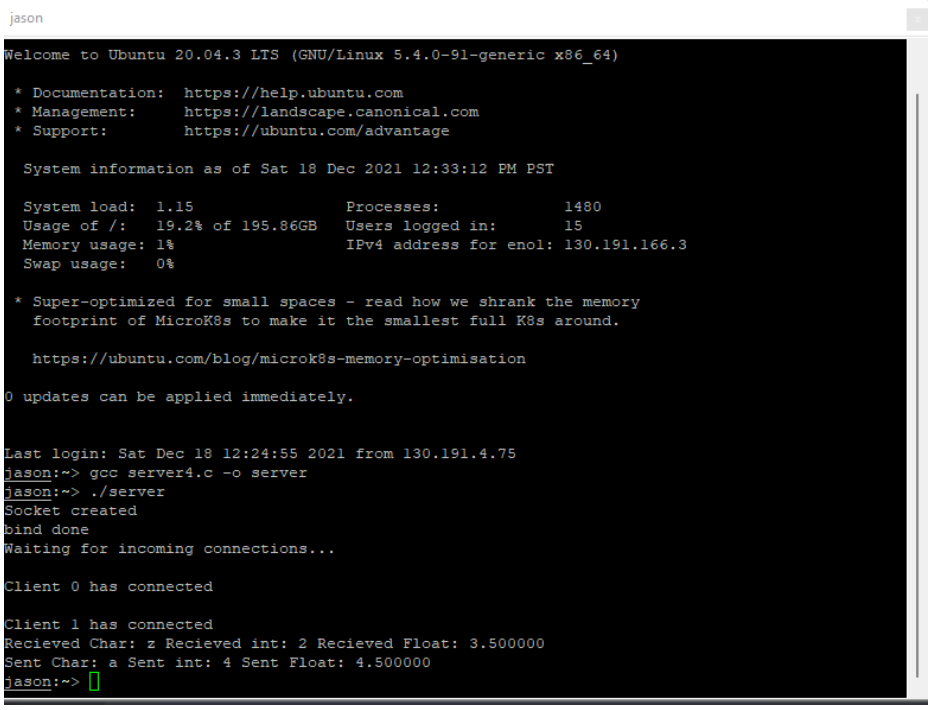

Figure 2(Server Screen)

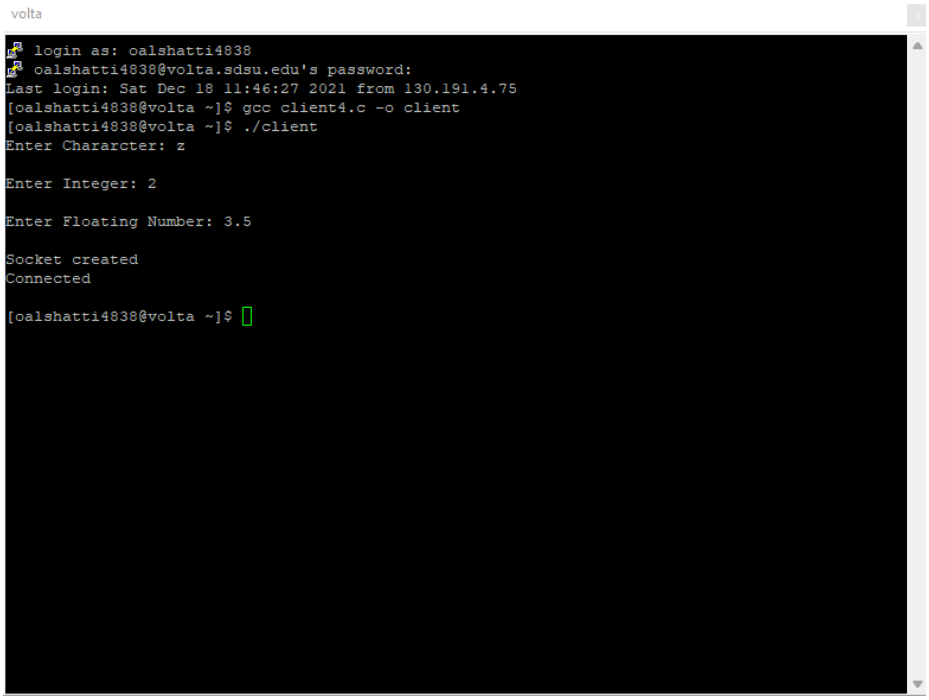

Figure 3(Client 0 screen which will send the C struct)

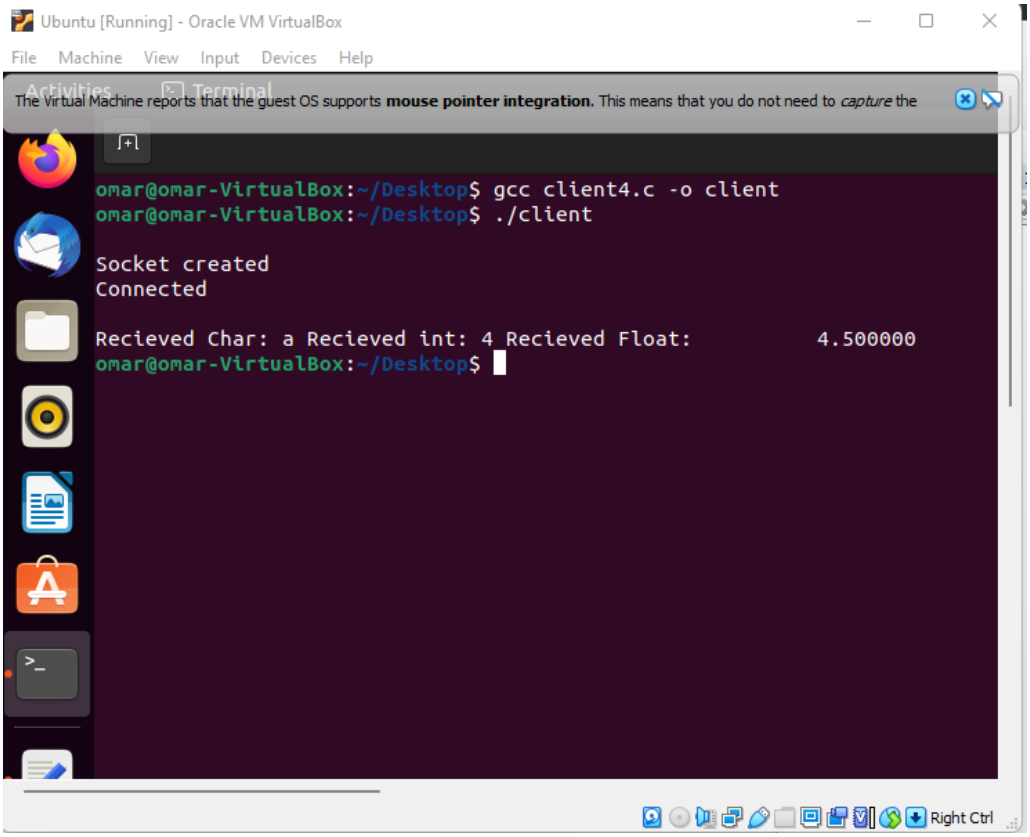

Figure 4(Client1 screen which receives C struct)

Conclusion:

We were able to successfully establish a connection between all clients and the server which allowed us to achieve our goal. While the process of sending C structs has been simple because of the similar compilers and architectures between all machines, we would have to serialize at first through client, deserialize it in our server, then do the same process from server to our next client.

Server Code:

```
#include<stdio.h>
#include<string.h>
#include<sys/socket.h>
#include<arpa/inet.h>
#include<unistd.h>

struct message
{
char a ;
int b;
float c;
};

int main()
{
	struct message Before;
	struct message After;
	int socket_desc  , c , read_size;
	struct sockaddr_in server , client;
	char client_message[2000];
int client_sock[2];

	//Create socket
	socket_desc = socket(AF_INET , SOCK_STREAM , 0);
	if (socket_desc == -1)
	{
		printf("Could not create socket");
	}
	puts("Socket created");

	//Prepare the sockaddr_in structure
	server.sin_family = AF_INET;
	server.sin_addr.s_addr = INADDR_ANY;
	server.sin_port = htons( 5023 );//specify port 5023 for connection

	//Bind
	if( bind(socket_desc,(struct sockaddr *)&server , sizeof(server)) < 0)
	{
		//print the error message
		perror("bind failed. Error");
```

```
        return 1;
    }
    puts("bind done");

    //Listen
    listen(socket_desc , 3);

    //Accept and incoming connection
    puts("Waiting for incoming connections...");
    c = sizeof(struct sockaddr_in);

    //accept connection from an incoming clients
   for(int i = 0; i<2; i++)
    {
     client_sock[i] = accept(socket_desc, (struct sockaddr *)&client, (socklen_t*)&c);
    printf("\nClient %d has connected\n", i);
    }
    if (client_sock[0] < 0)
    {
        perror("accept failed");
        return 1;
    }



    //Receive a message from client 0
     recv(client_sock[0] , &Before , sizeof(Before) , 0);
   //print message that was recieved from client 0
    printf("Recieved Char: %c Recieved int: %d Recieved Float: %f",Before.a,Before.b,Before.c);
    if(Before.a=='z'||Before.a=='Z')//change character with rollover support
    {
    After.a=Before.a - 25;//
    }
    else
    {
    After.a = Before.a +1;//add 1 if char is not rolled over
    }
    After.b =Before.b*2;//double integer
    After.c= Before.c + 1;//add 1 to float
    printf("\nSent Char: %c Sent int: %d Sent Float: %f\n",After.a,After.b,After.c);
    while( send(client_sock[1] , &After , sizeof(After) , 0)<0){}//send modified struct to client 1



    return 0;
}
```

Client 1 Code(That will input the Struct):

```
#include <stdio.h>      //printf
#include <string.h>     //strlen
#include <sys/socket.h> //socket
#include <arpa/inet.h>  //inet_addr
#include <unistd.h>
struct message
{
char a ;
int b;
float c;
};
struct message1
{
char a ;
int b;
float c;
};



int main()
{
        char d; int e;float f;
        printf("Enter Chararcter: ");
scanf("%c",&d);
printf("\nEnter Integer: ");
scanf("%d",&e);
printf("\nEnter Floating Number: ");
scanf("%f",&f);
        struct message m ={.a = d,.b =e,.c=f};
        struct message1 m1;


        char t[100];
        int sock;
        struct sockaddr_in server;
        char message[1000] , server_reply[2000];


        //Create socket
        sock = socket(AF_INET , SOCK_STREAM , 0);
        if (sock == -1)
        {
```

```
            printf("Could not create socket");
        }
        puts("\nSocket created");
    //specify port and IP address
        server.sin_addr.s_addr = inet_addr("130.191.166.3");
        server.sin_family = AF_INET;
        server.sin_port = htons( 5023 );

        //Connect to remote server
        if (connect(sock , (struct sockaddr *)&server , sizeof(server)) < 0)
        {
            perror("connect failed. Error");
            return 1;
        }

        puts("Connected\n");









                //Send some data
            if( send(sock , &m , sizeof(message) , 0) < 0)
            {

                return 1;
            }



            recv(sock , &m1 , sizeof(m1) , 0);
}
```

Client 2 Code(That will print the Struct):

```
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
struct message
{
char a ;
int b;
float c;
};
struct message1
{
char a ;
int b;
float c;
};



int main()
{
        struct message1 m1;

        char t[100];
        int sock;
        struct sockaddr_in server;
        char message[1000] , server_reply[2000];

        //Create socket
        sock = socket(AF_INET , SOCK_STREAM , 0);
        if (sock == -1)
        {
                printf("Could not create socket");
        }
        puts("\nSocket created");
       //Specify port and IP address
        server.sin_addr.s_addr = inet_addr("130.191.166.3");
        server.sin_family = AF_INET;
        server.sin_port = htons( 5023);

        //Connect to remote server
        if (connect(sock , (struct sockaddr *)&server , sizeof(server)) < 0)
        {
                perror("connect failed. Error");
                return 1;
        }
  puts("Connected\n");
```

```
            //recieve struct and store in local struct
              while(recv(sock , &m1 , sizeof(m1) , 0)<0){}

               printf("Recieved Char: %c Recieved int: %d Recieved Float:
%f\n",m1.a,m1.b,m1.c);// PRINT OUTPUT


        close(sock); //close sock and terminate procceses on server


}
```