|  |
| --- |
| **External Project Report on**  **Computer Networking (CSE3034)** |

## **Develop a client server model to declare the winner of a game.**



**Submitted by**

|  |  |  |
| --- | --- | --- |
| **Name 01 – Aparna Mohanta** | **Reg. No.:** | **1841017008** |
| **Name 02 – Guljabin Quadri** | **Reg. No.:** | **1841017111** |
| **Name 03 – Ankit Kumar** | **Reg. No.:** | **1841017135** |
| **Name 04 – Paul joseph** | **Reg. No.:** | **1841017115** |

**B. Tech. CSIT 5th Semester (Section - B)**

**Submitted to**



# Table of Contents

[1. Introduction 1](#_Toc60738435)

[1.1. Problem Definition 1](#_Toc60738436)

[1.2. Report Organization 1](#_Toc60738437)

[2. Theoretical Background 2](#_Toc60738438)

[2.1. Theory 2](#_Toc60738439)

[2.2. Pseudocodes 2](#_Toc60738440)

[3. Implementation, Results and Interpretation of Results 3](#_Toc60738441)

[3.1. Software used 3](#_Toc60738442)

[3.2. Program Listing 3](#_Toc60738443)

[3.3. Results 3](#_Toc60738444)

[3.4. Interpretation of Results 3](#_Toc60738445)

[4. Conclusion 4](#_Toc60738446)

[References 5](#_Toc60738447)

# Introduction

## Problem Definition

## Develop a client server model to declare the winner of a game.

## Report Organization

There is a client and a server. Two players(A and B) are sitting on them. They are

playing a game. Initially a player(A) writes(send) a number for the opponent(B). After that game

starts and the winner is declared.

# Theoretical Background

## Theory

There is a client and a server. Two players(A and B) are sitting on them. They are

playing a game. Initially a player(A) writes(send) a number for the opponent(B). After that game

starts. B will generate a random number from 1-10. Player(B) is supposed to write a number, which

is 1 or 2 or 3......or 10 more than the received number from A. Simillarly, A will generate a random

number from 1-10. Player(A) is supposed to write a number, which is 1 or 2 or 3......or 10 more than

the received number from B. The player who writes 100 or bigger is declared winner. Let the game

begin with 21.

## Pseudocodes

SERVER:

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<string.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<time.h>

int main()

{

int sockfd, newSocket, len, n;

char buffer[200];

struct sockaddr\_in serveraddr, cliaddr;

len = sizeof(serveraddr);

serveraddr.sin\_family=AF\_INET;

serveraddr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

serveraddr.sin\_port=htons(0);

CLIENT:  
int main()

{

int clientSocket,len,n;

long port;

struct sockaddr\_in cli\_addr;

char buffer[1024];

len = sizeof(cli\_addr);

# Implementation, Results and Interpretation of Results

## Software used

1. Ubuntu Vbox

2. Terminal-bash

## Program Listing

**//server**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<string.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<time.h>

int main()

{

int sockfd, newSocket, len, n;

char buffer[200];

struct sockaddr\_in serveraddr, cliaddr;

len = sizeof(serveraddr);

serveraddr.sin\_family=AF\_INET;

serveraddr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

serveraddr.sin\_port=htons(0);

sockfd=socket(AF\_INET,SOCK\_STREAM,0);

bind(sockfd,(struct sockaddr\*)&serveraddr,len);

getsockname(sockfd,(struct sockaddr\*)&serveraddr,&len);

printf("Port for client=%ld\n",(long)ntohs(serveraddr.sin\_port));

listen(sockfd,5);

newSocket=accept(sockfd,(struct sockaddr\*)&cliaddr,&len);

while(1)

{

recv(newSocket, buffer, 1024, 0);

//buffer[strlen(buffer)-1]='\0';

int y;

sscanf(buffer,"%d",&y);

if(y>=100 || y<=1)

{

printf("\n\n\*\*\*\*\*CLIENT(B) IS THE WINNER!!\*\*\*\*\*\n\n\*\*\*CONNECTION TERMINATED\*\*\*\n\n\n");

sleep(3);

send(newSocket, "terminated", 1024, 0);

break;

}

/\*

else if(strcmp(buffer,"yes")==0)

{

continue;

}

\*/

else{

int x;

printf("\n\nCLIENT :%s",buffer);

sscanf(buffer,"%d",&x);

// Generate a random number between 1-10

srand(time(NULL));

int randnum;

randnum = (rand() % 10)+1;

int currvalue = x+randnum;

snprintf(buffer,sizeof(buffer),"%d",currvalue);

printf("\nSERVER :%s",buffer);

send(newSocket, buffer, 1024, 0);

if(currvalue>=100)

{

printf("\n\n\*\*\*\*\*SERVER(A) IS THE WINNER!!\*\*\*\*\*\n\n\*\*\*CONNECTION TERMINATED\*\*\*\n\n\n");

break;

}

}

}

exit(0);

close(newSocket);

}

**//client**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<unistd.h>

#include<time.h>

int main()

{

int clientSocket,len,n;

long port;

struct sockaddr\_in cli\_addr;

char buffer[1024];

len = sizeof(cli\_addr);

printf("Enter the port number, you got from server side:");

scanf("%ld",&port);

while((getchar())!='\n');

cli\_addr.sin\_family=AF\_INET;

cli\_addr.sin\_addr.s\_addr=inet\_addr("127.0.0.1");

cli\_addr.sin\_port=htons(port);

clientSocket = socket(AF\_INET, SOCK\_STREAM, 0);

connect(clientSocket,(struct sockaddr \*)&cli\_addr,len);

printf("\n\*\*\*Type \"exit\" to terminate\*\n");

printf("\n\nCLIENT(Enter a number) :");

fgets(buffer, 1024, stdin);

send(clientSocket, buffer, 1024, 0);

int x,y;

while(1)

{

recv(clientSocket, buffer, 1024, 0);

// receive number from server

printf("SERVER :%s \n",buffer);

sscanf(buffer,"%d",&x);

if(x>=100)

{

printf("\n\*\*\*SERVER(A) is the winner.\*\*\*\n\nConnection terminated!!\n\n");

break;

}

// Generate a random number between 1-10

srand(time(NULL));

int randnum;

randnum = (rand() % 10)+1;

int currvalue = x+randnum;

snprintf(buffer,sizeof(buffer),"%d",currvalue);

printf("\nCLIENT :%s\n",buffer);

// printf("Enter yes to continue or no to exit(yes/no):\n");

// fgets(buffer, 1024, stdin);

sleep(2);

send(clientSocket, buffer, 1024, 0); // send number to server

//buffer[strlen(buffer)-1]='\0';

if(currvalue>=100)

{

printf("\n\n\*\*\*\*\*CLIENT(B) IS THE WINNER!!\*\*\*\*\*\n\n\*\*\*CONNECTION TERMINATED\*\*\*\n\n\n");

break;

}

sscanf(buffer,"%d",&y);

if(y>=100 || y<=1)

{

printf("Connection terminated!!\n\n");

break;

}

}

printf("\n\n");

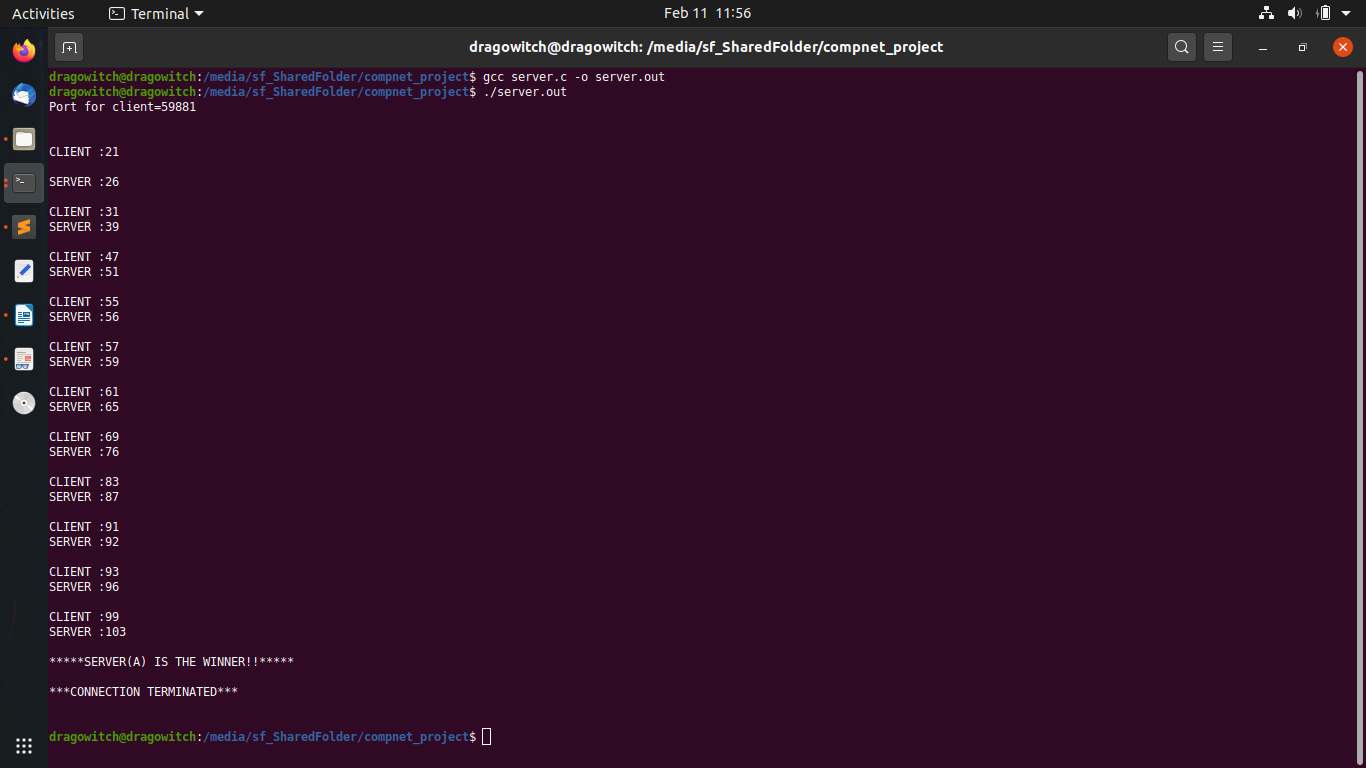
close(clientSocket);

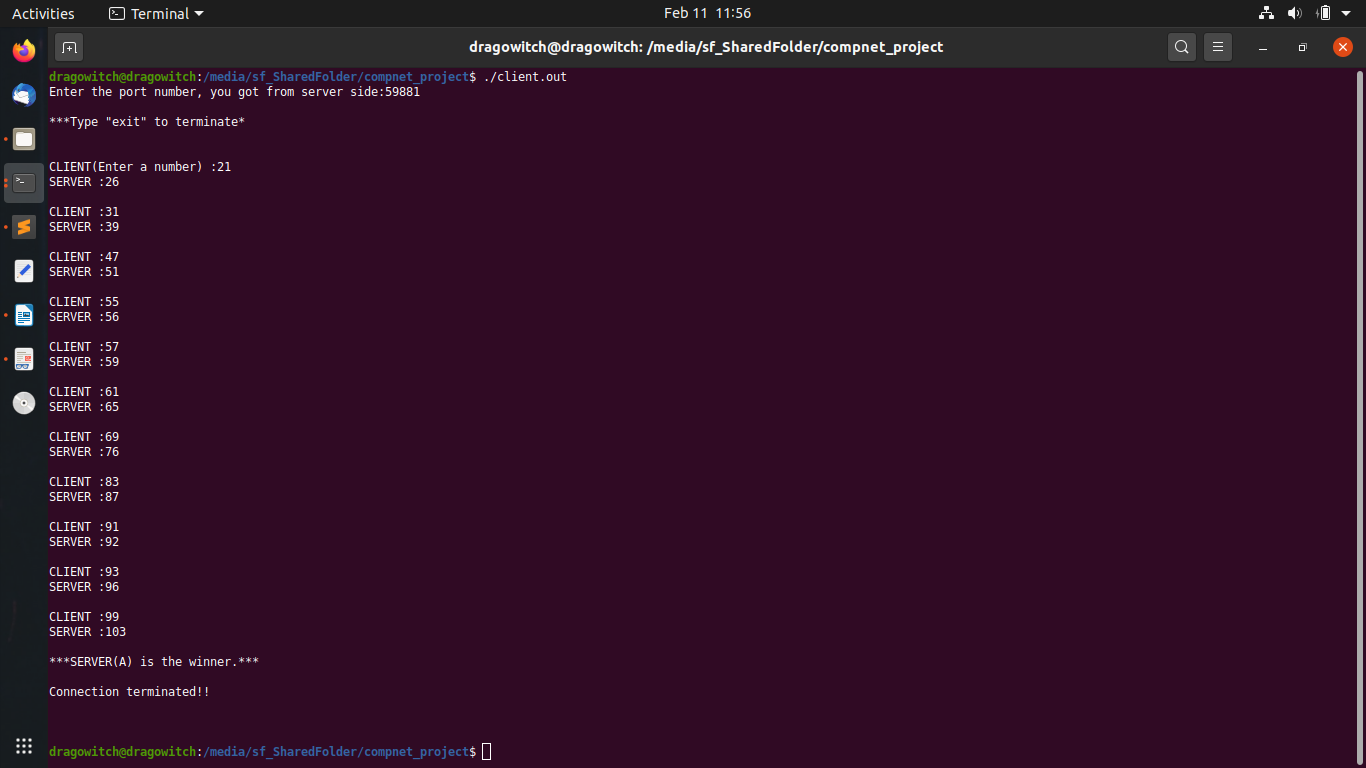
exit(0);

//return 0;

}

## Results





## Interpretation of Results

The Game Has been Completed and the SERVER(A) is the winner.

# Conclusion

**Server assigns a Specific random port no to the Client and then the client joins the game Server is assigned as player no A and Client as Player no B,**

**Then the game starts and the each player takes their turn and enter a number the first player to enter the number 100 or bigger number will be declared the winner of this game.**

**The game will start from 21 and end on 100 or more**

# References

**Web ARTICLES LIKE – Geeksforgeeks.com**

**Sanfondry.com**

**StackOverflow.com**

**Video Guidance & reference – Youtube.com**