**Name: Sneha Tiwari**

**Univ. Roll No.: 11500119052**

**Assignment 10 – Concurrent Server**

**Client Code:**

#include <arpa/inet.h>

#include <stdio.h>

#include <string.h>

#include <sys/socket.h>

#include <unistd.h>

#define PORT 8080

int main(int argc, char const\* argv[])

{

int sock = 0, valread;

struct sockaddr\_in serv\_addr;

if ((sock = socket(AF\_INET, SOCK\_STREAM, 0)) < 0) {

printf("\n Socket creation error \n");

return -1;

}

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

serv\_addr.sin\_port = htons(PORT);

if (inet\_pton(AF\_INET, "127.0.0.1", &serv\_addr.sin\_addr)<= 0) {

printf("\nInvalid address/ Address not supported \n");

return -1;

}

if (connect(sock, (struct sockaddr\*)&serv\_addr,sizeof(serv\_addr))< 0) {

printf("\nConnection Failed \n");

return -1;

}

for(;;)

{

char hello[1024];

scanf("%[^\n]%\*c",hello);

char buffer[1024] = { 0 };

send(sock, hello, strlen(hello), 0);

valread = read(sock, buffer, 1024);

if (strncmp("exit", buffer, 4) == 0) {

printf("Server Exit...\n");

break;

}else

{

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

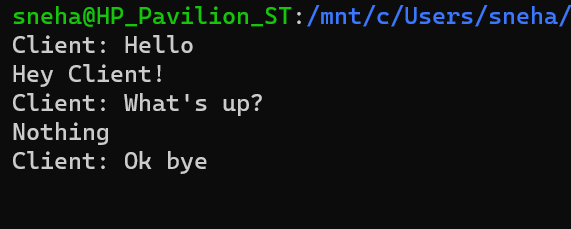
}

}

return 0;

}

**Output:**

****

**Server Code:**

#include <netinet/in.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <unistd.h>

#define PORT 8080

int main(int argc, char const\* argv[])

{

int server\_fd, new\_socket, valread;

struct sockaddr\_in address;

int opt = 1;

int addrlen = sizeof(address);

if ((server\_fd = socket(AF\_INET, SOCK\_STREAM, 0))== 0) {

perror("socket failed");

exit(EXIT\_FAILURE);

}

if (setsockopt(server\_fd, SOL\_SOCKET,SO\_REUSEADDR | SO\_REUSEPORT, &opt,sizeof(opt))) {

perror("setsockopt");

exit(EXIT\_FAILURE);

}

address.sin\_family = AF\_INET;

address.sin\_addr.s\_addr = INADDR\_ANY;

address.sin\_port = htons(PORT);

if (bind(server\_fd, (struct sockaddr\*)&address,sizeof(address))< 0) {

perror("bind failed");

exit(EXIT\_FAILURE);

}

if (listen(server\_fd, 3) < 0) {

perror("listen");

exit(EXIT\_FAILURE);

}

if ((new\_socket

= accept(server\_fd, (struct sockaddr\*)&address,

(socklen\_t\*)&addrlen))

< 0) {

perror("accept");

exit(EXIT\_FAILURE);

}

for(;;)

{

char buffer[1024] = { 0 };

valread = read(new\_socket, buffer, 1024);

if (strncmp("exit", buffer, 4) == 0) {

printf("Server Exit...\n");

break;

}else

{

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

}

char hello[1024];

scanf("%[^\n]%\*c",hello);

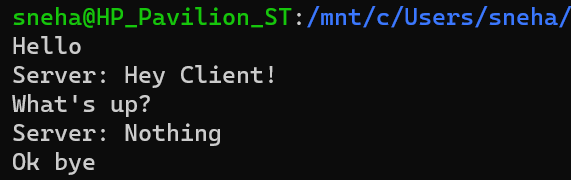
send(new\_socket, hello, strlen(hello), 0);

}

return 0;

}

**Output:**

****

**------------------------------------------------------------**