Experiment – 5

client.c

#include <stdio.h>

#include <string.h>

#include <sys/socket.h>

#include <netinet/ip.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <unistd.h>

int main(){

int n, cid, sid, l, N, i, res;

struct sockaddr\_in s, c; char ch;

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

s.sin\_family = AF\_INET;

s.sin\_port = htons(1234);

inet\_aton("127.0.0.1",&s.sin\_addr);

puts("Ready to send"); l = sizeof(c);

connect(sid, (struct sockaddr\*)&s, sizeof(s));

char bitseq[100];

puts("Enter the message");

fgets(bitseq,100,stdin);

puts("Sequence obtained.Sending Sequence to Server...");

write(sid,bitseq,strlen(bitseq));

char new\_bitseq[101];

n = read(sid,&new\_bitseq,sizeof(new\_bitseq));

printf("\nThe modified bit sequence is: %s\n", new\_bitseq);close(sid); return 0;}

server.c

#include <stdio.h>

#include <string.h>

#include <pthread.h>

#include <sys/socket.h>

#include <netinet/ip.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <unistd.h>

#include <stdlib.h>

int sid;

struct sockaddr\_in s, c;

void\* multi\_client(void\* arg)

{

int cid, l, n;

char bits[100];

l = sizeof(c);

cid = accept(sid, (struct sockaddr\*)&s, &l);

if(cid<0){

printf("error");

exit(0);

}

n = read(cid,bits,sizeof(bits));

bits[n] = '\0';

int size\_l = strlen(bits);

int count = 0;

for(int i=0;i<size\_l;i++){

if(bits[i]=='1')

count++;}

//printf("%d",count);

if(count%2==1){

printf("%s",bits);

printf("\n1\n");

bits[size\_l]='1';}

else{

printf("%s",bits); printf("\n0\n");bits[size\_l]='0';}

write(cid,&bits,sizeof(bits));

close(cid);}

int main(){

int MAX\_CONNECTIONS = 3,temp;

char bits[100];

pthread\_t th[MAX\_CONNECTIONS];

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

s.sin\_family = AF\_INET;

s.sin\_port = htons(1234);

inet\_aton("127.0.0.1",&s.sin\_addr);

puts("Server started!");

bind(sid, (struct sockaddr\*)&s, sizeof(s));

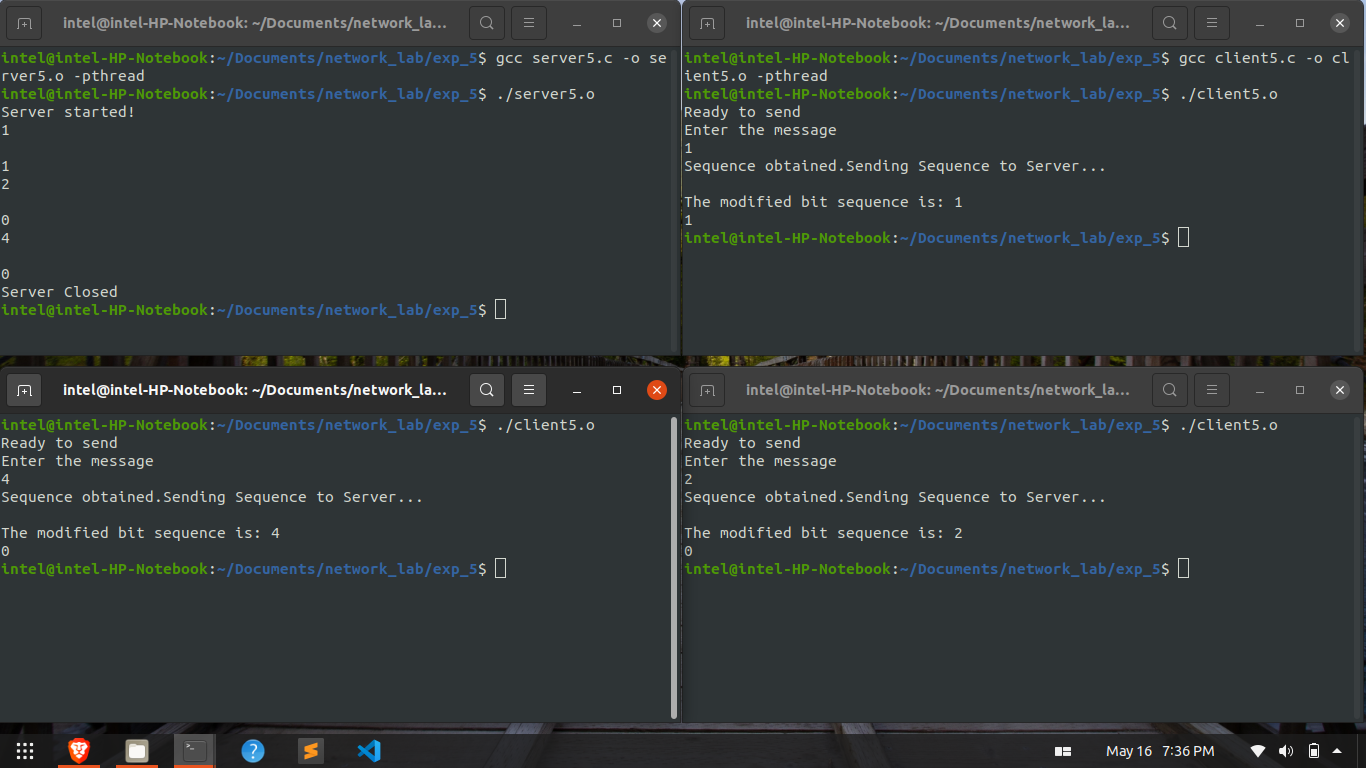
listen(sid,MAX\_CONNECTIONS); temp=MAX\_CONNECTIONS;

while(temp--)

pthread\_create(&th[temp],NULL,multi\_client,NULL);

temp=MAX\_CONNECTIONS;

while(temp--)

 pthread\_join(th[temp],NULL);

close(sid); puts("Server Closed"); return 0;}