EXERCISE:1 SOCKET PROGRAMMINNG

SERVER:

#include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

int main(){

int welcomeSocket, newSocket;

char buffer[1024];

struct sockaddr\_in serverAddr;

struct sockaddr\_storage serverStorage;

socklen\_t addr\_size;

/\*---- Create the socket. The three arguments are: ----\*/

/\* 1) Internet domain 2) Stream socket 3) Default protocol (TCP in this case) \*/

welcomeSocket = socket(PF\_INET, SOCK\_STREAM, 0);

/\*---- Configure settings of the server address struct ----\*/

/\* Address family = Internet \*/

serverAddr.sin\_family = AF\_INET;

/\* Set port number, using htons function to use proper byte order \*/

serverAddr.sin\_port = htons(8060);

/\* Set IP address to localhost \*/

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

/\* Set all bits of the padding field to 0 \*/

memset(serverAddr.sin\_zero, '\0', sizeof serverAddr.sin\_zero);

/\*---- Bind the address struct to the socket ----\*/

bind(welcomeSocket, (struct sockaddr \*) &serverAddr, sizeof(serverAddr));

/\*---- Listen on the socket, with 5 max connection requests queued ----\*/

if(listen(welcomeSocket,5)==0)

printf("Listening\n");

else

printf("Error\n");

/\*---- Accept call creates a new socket for the incoming connection ----\*/

addr\_size = sizeof serverStorage;

newSocket = accept(welcomeSocket, (struct sockaddr \*) &serverStorage, &addr\_size);

strcpy(buffer,"Hello World\n");

send(newSocket,buffer,13,0);

return 0;

OUTPUT:

[211716205052@Putty ~]$ vi server.c

[211716205052@Putty ~]$ gcc server.c

[211716205052@Putty ~]$ ./a.out

Listening

CLIENT:

#include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

int main(){

int clientSocket;

char buffer[1024];

struct sockaddr\_in serverAddr;

socklen\_t addr\_size;

/\*---- Create the socket. The three arguments are: ----\*/

/\* 1) Internet domain 2) Stream socket 3) Default protocol (TCP in this case) \*/

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

/\*---- Configure settings of the server address struct ----\*/

/\* Address family = Internet \*/

serverAddr.sin\_family = AF\_INET;

/\* Set port number, using htons function to use proper byte order \*/

serverAddr.sin\_port = htons(8060);

/\* Set IP address to localhost \*/

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

/\* Set all bits of the padding field to 0 \*/

memset(serverAddr.sin\_zero, '\0', sizeof serverAddr.sin\_zero);

/\*---- Connect the socket to the server using the address struct ----\*/

addr\_size = sizeof serverAddr;

connect(clientSocket, (struct sockaddr \*) &serverAddr, addr\_size);

/\*---- Read the message from the server into the buffer ----\*/

recv(clientSocket, buffer, 1024, 0);

/\*---- Print the received message ----\*/

printf("Data received: %s",buffer);

return 0;

}

OUTPUT:

[211716205052@Putty ~]$ vi client.c

[211716205052@Putty ~]$ vi client.c

[211716205052@Putty ~]$ gcc client.c

[211716205052@Putty ~]$ ./a.out

Data received: Hello World

EXERCISE 2: DATE AND TIME

SERVER:

#include"netinet/in.h"

#include"sys/socket.h"

#include"stdio.h"

#include"string.h"

#include"time.h"

main( )

{

struct sockaddr\_in sa;

struct sockaddr\_in cli;int sockfd,conntfd;int len,ch;char str[100];

time\_t tick;

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

if(sockfd<0)

{

printf("error in socket\n");

exit(0);

}

else printf("Socket opened");

bzero(&sa,sizeof(sa));

sa.sin\_port=htons(2108);

sa.sin\_addr.s\_addr=htonl(0);

if(bind(sockfd,(struct sockaddr\*)&sa,sizeof(sa))<0)

{

printf("Error in binding\n");

}

else

printf("Binded Successfully");

listen(sockfd,50);

for(;;)

{

len=sizeof(ch);

conntfd=accept(sockfd,(struct sockaddr\*)&cli,&len);

printf("Accepted");

tick=time(NULL);

snprintf(str,sizeof(str),"%s",ctime(&tick));

printf("%s",str);write(conntfd,str,100);

}

}

OUTPUT:

[211716205052@Putty ~]$ vi serverdatetime.c

[211716205052@Putty ~]$ gcc serverdatetime.c

[211716205052@Putty ~]$ ./a.out

Socket openedBinded SuccessfullyAccepted wed Jul 4 14:23:41 2018

SERVER:

#include"netinet/in.h"

#include"sys/socket.h"

#include"stdio.h"

main()

{

struct sockaddr\_in sa,cli;

int n,sockfd;

int len;char buff[100];

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

if(sockfd<0){ printf("\nError in Socket");

exit(0);

}

else printf("\nSocket is Opened");

bzero(&sa,sizeof(sa));

sa.sin\_family=AF\_INET;

sa.sin\_port=htons(2108);

if(connect(sockfd,(struct sockaddr\*)&sa,sizeof(sa))<0)

{

printf("\nError in connection failed");

exit(0);

}

else

printf("\nconnected successfully");

if(n=read(sockfd,buff,sizeof(buff))<0)

{

printf("\nError in Reading");

exit(0);

}

else

{printf("\nMessage Read %s",buff);

}}

OUTPUT:

[211716205052@Putty ~]$ vi clientdatetime.c

[211716205052@Putty ~]$ gcc clientdatetime.c

[211716205052@Putty ~]$ ./a.out

Socket is Opened

connected successfully

Message Read wed Jul 4 14:44:41 2018