1. Simple Hello

Client.c

#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(7891);

/\* 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;

}

Server.c

#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(7891);

/\* 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);

/\*—- Send message to the socket of the incoming connection —-\*/

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

send(newSocket,buffer,13,0);

return 0;

}