/\*--------------------------\*/

/\* Code for a simple client \*/

/\*--------------------------\*/

#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 <netdb.h>

void error(const char \*msg)

{

perror(msg);

exit(0);

}

int main()

{

int sockfd, portno, n;

struct sockaddr\_in serv\_addr;

struct hostent \*server;

char buffer[256],hostname[20],portnumber[10];

/\* Ask user to provide hostname and port number \*/

printf("\nClient requires the following to operate");

printf("\nPlease provide server hostname: ");

scanf("%s",hostname);

printf("\nPlease provide server port num: ");

scanf("%s",portnumber);

getchar();

/\* Open socket from client end \*/

portno = atoi(portnumber);

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

if (sockfd < 0)

error("ERROR opening socket");

/\* Connect to server using hostname \*/

server = gethostbyname(hostname);

if (server == NULL) {

fprintf(stderr,"ERROR, no such host\n");

exit(0);

}

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

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

bcopy((char \*)server->h\_addr, (char \*)&serv\_addr.sin\_addr.s\_addr, server->h\_length);

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

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

error("ERROR connecting");

/\* Client connected -- Wait for user message \*/

printf("\nConnected to server 'localhost' at port num '%s'",portnumber);

printf("\nReady for communication ...");

printf("\n");

/\* Take user input for message \*/

printf("\nPlease enter the message: ");

bzero(buffer,sizeof(buffer));

fgets(buffer,sizeof(buffer),stdin);

/\* Write message to socket for communication \*/

printf("\nWriting message to socket ...");

if (write(sockfd,buffer,sizeof(buffer)) < 0)

error("ERROR writing to socket");

printf("\nMessage sent/written to socket");

/\* Listen to the socket for client message \*/

printf("\nClient listening for server ACK ...");

printf("\n");

/\* Read ACK from server \*/

bzero(buffer,256);

if (read(sockfd,buffer,sizeof(buffer)) < 0)

error("ERROR reading from socket");

printf("\nRead successful for server ACK");

printf("\nServer sent this ACK : %s",buffer);

/\* Close the socket and end communication \*/

printf("\nClosing the socket ...\n\n");

close(sockfd);

return 0;

}

/\*--------------------------\*/

/\* Code for a simple server \*/

/\*--------------------------\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

void error(const char \*msg)

{

perror(msg);

exit(1);

}

int main()

{

int sockfd, newsockfd, portno;

char buffer[256],portnumber[10];

struct sockaddr\_in serv\_addr, cli\_addr;

socklen\_t clilen;

/\* Ask user to provide a port number \*/

printf("\nServer requires a port number to operate.");

printf("\nPlease provide a port number: ");

scanf("%s",portnumber);

/\* Open socket from server end \*/

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

if (sockfd < 0)

error("ERROR opening socket");

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

portno = atoi(portnumber);

/\* Bind socket to the server \*/

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

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

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

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

error("ERROR on binding");

/\* Listen to the socket for client connection \*/

printf("\nServer created\thostname : 'localhost'\n\t\tport num : '%s'",portnumber);

printf("\nCreate a client and send connection request");

printf("\nServer listening for client connection ...");

printf("\n");

listen(sockfd,5);

/\* Accept client connection on request \*/

clilen = sizeof(cli\_addr);

newsockfd = accept(sockfd, (struct sockaddr \*) &cli\_addr, &clilen);

if (newsockfd < 0)

error("ERROR on accept");

bzero(buffer,sizeof(buffer));

/\* Listen to the socket for client message \*/

printf("\nClient connected");

printf("\nServer listening for client message ...");

printf("\n");

/\* Read client message from socket \*/

if (read(newsockfd,buffer,sizeof(buffer)) < 0)

error("ERROR reading from socket");

printf("\nRead successful for client message");

printf("\nClient sent this message : %s",buffer);

/\* Send ACK for client message \*/

if (write(newsockfd,"Thanks, I got your message",sizeof(buffer)) < 0)

error("ERROR writing to socket");

printf("\nACK sent in response to client message");

/\* Close the socket and end communication \*/

printf("\nClosing the socket ...\n\n");

close(newsockfd);

close(sockfd);

return 0;

}