Warren Quattrocchi
CSC 138
Section 3

**Project 4: C implementation of client-server communications**

**Server.c**

```
/*Warren Quattrocchi
 * CSC 138
 * C socket
 * server.c
 */
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdlib.h>

#define SERVER_PORT 6512
#define MAX_PENDING 5
#define MAX_LINE 256

int main()
{
  struct sockaddr_in sin;
  char buf[MAX_LINE];
  int len;
  int s, new_s;

  bzero((char*)&sin,sizeof(sin));
  sin.sin_family = AF_INET;
  sin.sin_addr.s_addr = INADDR_ANY;
  sin.sin_port = htons(SERVER_PORT);

  //create a new socket
  if((s = socket(AF_INET,SOCK_STREAM,0)) < 0)
  {
    printf("Error creating socket\n");
    exit(0);
  }
  //bind the socket
  if((bind(s,(struct sockaddr *)&sin, sizeof(sin))) < 0)
  {
    printf("Error binding socket\n");
    exit(0);
  }
  //listen for incoming connections
  if((listen(s,MAX_PENDING)) < 0)
```

```
  {
    printf("Error listening\n");
    exit(0);
  }
  //loop forever
  while(1){
    //accept incoming connection
    if((new_s = accept(s,(struct sockaddr *)NULL,NULL)) < 0)
    {
      printf("Error on accept\n");
      exit(0);
    }
    bzero(buf, MAX_LINE);
    len = sizeof(sin);
    //loop while client is still sending
    while(recv(new_s, buf, MAX_LINE,0) > 0)
       printf("%s", buf);
    close(new_s);
  }
 close(s);
}
```

**Client.c**

```
/*Warren Quattrocchi
 * CSC 138
 * C socket
 * client.c
 */

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <string.h>
#include <stdlib.h>

#define SERVER_PORT 6512
#define MAX_LINE 256

int main(int argc, char * argv[])
{
   FILE *fp;
   struct hostent *hp;
   struct sockaddr_in sin;
   char *host;
   char buf[MAX_LINE];
   int s;
   int len;
   if(argc <2)
   {
      printf("usage: client [host ip]\n");
```

```
      exit(0);
   }
   host = argv[1];

   hp = gethostbyname(host);
   bzero((char*)&sin, sizeof(sin));
   sin.sin_family = AF_INET;
   bcopy(hp->h_addr,(char*)&sin.sin_addr, hp->h_length);
   sin.sin_port = htons(SERVER_PORT);

    //create new socket
    if((s = socket(PF_INET, SOCK_STREAM, 0)) < 0)
    {
      printf("Error creating socket\n");
      exit(0);
    }
    //connect socket to server
    if(connect(s, (struct sockaddr*)&sin, sizeof(sin)) < 0)
    {
      printf("Error connecting\n");
      exit(0);
    }
   //loop while input is given
   while(fgets(buf,MAX_LINE,stdin) > 0){
      buf[MAX_LINE-1] = '\0';
      len = strlen(buf) + 1;
      //send message to server
      if(send(s,buf,len,0) < 0)
      {
         printf("Error sending");
         exit(0);
      }
      //initialize buffer
      bzero(buf,MAX_LINE);
   }
}
```

**Output**