

Server.c

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <stdio.h>
#include <unistd.h> /* close */

#define SERVER_PORT 1500

int main (int argc, char *argv[]) {
    int sd, newSd, cliLen;
    struct sockaddr_in cliAddr, servAddr;
    char line[MAX_MSG];
    int len;

    sd = socket(AF_INET, SOCK_STREAM, 0);
    if(sd<0) {
        perror("cannot open socket ");
        return ERROR;
    }

    /* bind server port */
    servAddr.sin_family = AF_INET;
    servAddr.sin_addr.s_addr =
htonl(INADDR_ANY);
    servAddr.sin_port = htons(SERVER_PORT);

    if (bind(sd, (struct sockaddr *) &servAddr,
sizeof(servAddr))<0) {
        perror("cannot bind port ");
        return ERROR;
    }

    listen(sd,5);
    while(1) {

        printf("%s: waiting for data on port TCP
%u\n",argv[0],SERVER_PORT);

        cliLen = sizeof(cliAddr);
        newSd = accept(sd, (struct sockaddr *) &cliAddr,
&cliLen);
        if(newSd<0) {
            perror("cannot accept connection ");
            return ERROR;
        }

        /* init line */
        memset(line,0x0,MAX_MSG);

        /* receive segments */
        if ( (len=read(newSd,line,MAX_MSG))> 0) {
```

```
            printf("%s: received from %s:TCP%d : %s\n",
argv[0],
                inet_ntoa(cliAddr.sin_addr),
                ntohs(cliAddr.sin_port), line);

            /* reverse the line */
            write(newSd,line,len);
        }
        else
            printf("Error receiving data\n");
        close(newSd);
    }
}
```

CLIENT.C

```
include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <stdio.h>
#include <unistd.h> /* close */

#define SERVER_PORT 1500
#define MAX_MSG 100

int main (int argc, char *argv[]) {

    int sd, rc, i;
    struct sockaddr_in localAddr, servAddr;
    struct hostent *h;
    char msg[300];

    if(argc < 3) {
        printf("usage: %s <server> <text>\n",argv[0]);
        exit(1);
    }

    h = gethostbyname(argv[1]);
    if (h==NULL) {
        printf("%s: unknown host %s\n",argv[0],argv[1]);
        exit(1);
    }

    servAddr.sin_family = h->h_addrtype;
    memcpy((char *) &servAddr.sin_addr.s_addr, h->h_addr_list[0], h->h_length);
    servAddr.sin_port = htons(SERVER_PORT);

    /* create socket */
    sd = socket(AF_INET, SOCK_STREAM, 0);
    if(sd<0) {
        perror("cannot open socket ");
        exit(1);
    }

    /* connect to server */
    rc = connect(sd, (struct sockaddr *) &servAddr, sizeof(servAddr));
    if(rc<0) {
        perror("cannot connect ");
        exit(1);
    }

    write(rc, argv[1],strlen(argv[1])+1 );
    read(rc, msg, 300);
    printf("Received back: %s\n", msg);
    close(rc);

    return 0;

}
```