

**EXPERIMENT-7 (a)**

---

**OBJECTIVE**

Write a programs in C: hello\_client (The server listens for, and accepts, a single TCP connection; it reads all the data it can from that connection, and prints it to the screen; then it closes the connection)

**PROGRAM**

```
/* CLIENT PROGRAM FOR TCP CONNECTION */
```

```
#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 argc, char *argv[])

{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server;
    char buffer[256];

    if (argc < 3)
    {
        fprintf(stderr, "usage %s hostname port\n", argv[0]);
```

```
exit(0);

    }
    portno = atoi(argv[2]);
sockfd = socket(AF_INET, SOCK_STREAM, 0);

    if (sockfd < 0)
        error("ERROR opening socket");

    server = gethostbyname(argv[1]);

    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, (structsockaddr *) &serv_addr, sizeof(serv_addr)) < 0)

        error("ERROR connecting");

    printf("Please enter the message: ");
    bzero(buffer, 256);
    fgets(buffer, 255, stdin);
    n = write(sockfd, buffer, strlen(buffer));

    if (n < 0)

        error("ERROR writing to socket");

    bzero(buffer, 256);
    n = read(sockfd, buffer, 255);
    if (n < 0)
        error("ERROR reading from socket");
```

```
printf("%s\n",buffer);  
close(sockfd);  
return 0;  
  
}
```

### **TO RUN-**

```
$ gcc -o client helloclient.c  
$ ./client localhostportno
```

### **OUTPUT**

```
Please enter the message: I m client  
I got your message
```

**EXPERIMENT-7 (b)**

---

**OBJECTIVE**

Write a program in C: hello\_server for TCP (The client connects to the server, sends the string "Hello, world!", then closes the connection)

**PROGRAM**

```
/* SERVER PROGRAM FOR TCP CONNECTION */
#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 argc, char *argv[])
{
    int sockfd, newsockfd, portno;
    socklen_t clilen;
    char buffer[256];
    struct sockaddr_in serv_addr, cli_addr;
    int n;
    if (argc < 2) {
        fprintf(stderr, "ERROR, no port provided\n");
        exit(1);
    }
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
```

```
bzero((char *) &serv_addr, sizeof(serv_addr));
portno = atoi(argv[1]);
serv_addr.sin_family = AF_INET;
serv_addr.sin_addr.s_addr = INADDR_ANY;
serv_addr.sin_port = htons(portno);
if (bind(sockfd, (structsockaddr *) &serv_addr,
sizeof(serv_addr)) < 0)
error("ERROR on binding");
listen(sockfd,5);
clilen = sizeof(cli_addr);
newsockfd = accept(sockfd,
(structsockaddr *) &cli_addr,
&clilen);
if (newsockfd< 0)
error("ERROR on accept");
bzero(buffer,256);
n = read(newsockfd,buffer,255);
if (n < 0) error("ERROR reading from socket");
printf("Here is the message: %s\n",buffer);
n = write(newsockfd,"I got your message",18);
if (n < 0) error("ERROR writing to socket");
close(newsockfd);
close(sockfd);
return 0;
}
```

**To RUN-**

```
$ gcc -o server helloserver.c
$ ./server portno
```

**OUTPUT**

Here is the message: I'm client