

Name- Shatabdi Chakraborty

Regd No. 1841017152

Experiment 1

OBJECTIVE 1

To implement an echo client server using TCP/IP.

Client-

```
shatzz@LAPTOP-A2PUH10V: ~/CNLAB
#include<stdio.h>
#include<netinet/in.h>
#include<netdb.h>
#include<fcntl.h>
#include<unistd.h>
#define SERV_TCP_PORT 5035
int main(int argc, char**argv)
{
    int sockfd, newsockfd, clength;
    struct sockaddr_in serv_addr, cli_addr;
    char buffer[4096];
    sockfd=socket(AF_INET, SOCK_STREAM, 0);
    serv_addr.sin_family=AF_INET;
    serv_addr.sin_addr.s_addr=INADDR_ANY;
    serv_addr.sin_port=htons(SERV_TCP_PORT);
    bind(sockfd, (struct sockaddr*)&serv_addr, sizeof(serv_addr));
    printf("\nServer is Listening..");
    printf("\n"); listen(sockfd, 5);
    clength=sizeof(cli_addr);
    newsockfd=accept(sockfd, (struct sockaddr*)&cli_addr, &clength);
    printf("\nClient accepted");
    printf("\n"); read(newsockfd, buffer, 4096);
    printf("\nClient message:%s", buffer);
    write(newsockfd, buffer, 4096);
    printf("\n");
    close(sockfd);
    return 0;
}
```

Server-

```
shatzz@LAPTOP-A2PUH1OV: ~  
#include<netdb.h>  
#include<fcntl.h>  
#include<unistd.h>  
#include<arpa/inet.h>  
#define SERV_TCP_PORT 5035  
int main(int argc,char*argv[])  
{  
    int sockfd;  
    struct sockaddr_in serv_addr;  
    struct hostent *server;  
    char buffer[4096];  
    sockfd=socket(AF_INET,SOCK_STREAM,0);  
    serv_addr.sin_family=AF_INET;  
    serv_addr.sin_addr.s_addr=inet_addr("127.0.0.1");  
    serv_addr.sin_port=htons(SERV_TCP_PORT);  
    printf("\nConnected");  
    connect(sockfd,(struct sockaddr*)&serv_addr,sizeof(serv_addr));  
    printf("\nEnter the message\n");  
    printf("\nClient: ");  
    fgets(buffer,4096,stdin);  
    write(sockfd,buffer,4096);  
    printf("echo message by server: %s",buffer);  
    printf("\n");  
    close(sockfd);  
    return 0;  
}
```

Output-

```
shatzz@LAPTOP-A2PUH1OV:~$ vi eloiclient.c  
shatzz@LAPTOP-A2PUH1OV:~$ gcc eloiclient.c  
shatzz@LAPTOP-A2PUH1OV:~$ ./a.out  
  
Connected  
Enter the message  
  
Client: Hey This is Shatabdi  
echo message by server: Hey This is Shatabdi  
shatzz@LAPTOP-A2PUH1OV:~$  
  
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ vi eloiserver.c  
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ gcc eloiserver.c  
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ ./a.out  
  
Server is Listening..  
  
Client accepted  
Client message:Hey This is Shatabdi  
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$
```

OBJECTIVE 2

To Implement a chat of client server communication using TCP/IP

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 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(1);
    }
    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");
    }
    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("Connection Failed ");
    while(1)
    {
        bzero(buffer,255);
        fgets(buffer , 255,stdin);
```

```

n = write(sockfd,buffer,strlen(buffer));
if(n<0)
error("Error on writing");
bzero(buffer,255);
n = read(sockfd,buffer,255);
if(n<0)
error("error on reading");
printf("Server : %s",buffer);
int i = strncmp("Bye",buffer,3);
if(i==0)
break;
}
close(sockfd);
return 0;
}

```

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 argc , char *argv[])
{
if(argc < 2){
fprintf(stderr,"Port no not provided , program terminated");
exit(1);
}
int sockfd , newsockfd , portno , n;
char buffer[255];
struct sockaddr_in serv_addr , cli_addr;
socklen_t clilen;
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,(struct sockaddr *) &serv_addr , sizeof(serv_addr)) < 0)
error("Binding failed");

```

```

listen(sockfd,5);
clilen = sizeof(cli_addr);
newsockfd = accept(sockfd,(struct sockaddr *) &cli_addr,&clilen);
if(newsockfd < 0)
error("Error on accept.");
while(1)
{
bzero(buffer , 250);
n = read(newsockfd , buffer , 255);
if(n < 0)
error("Error on reading");
printf("Client : %s\n", buffer);
bzero(buffer , 255);
fgets(buffer , 255 , stdin);
n = write(newsockfd , buffer , strlen(buffer));
if(n<0)
error("error on writing");
int i = strncmp("Bye",buffer,3);
if(i==0)
break;
}
close(newsockfd);
close(sockfd);
return 0;
}

```

Output-

The image displays two terminal windows side-by-side, illustrating the output of a client-server program. The left window shows the client's execution, and the right window shows the server's execution.

Left Terminal (Client):

```

shatzz@LAPTOP-A2PUH10V:~/CNLAB$ gcc elo2client.c
shatzz@LAPTOP-A2PUH10V:~/CNLAB$ ./a.out 127.0.0.1 53662

Server : Hi What are you doing
I was reading a Novel.
Server : How are you?
I am fine.Thankyou
Server : What is your thinking about CN?
Its an interesting subject.

```

Right Terminal (Server):

```

shatzz@LAPTOP-A2PUH10V:~/CNLAB$ gcc elo2server.c
shatzz@LAPTOP-A2PUH10V:~/CNLAB$ ./a.out 53662
Client :

Hi What are you doing
Client : I was reading a Novel.

How are you?
Client : I am fine.Thankyou

What is your thinking about CN?
Client : Its an interesting subject.

```

OBJECTIVE 3

To implement date and time display from client to server using TCP sockets.

Client-

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include <time.h>
int main(int argc, char **argv)
{
    if(argc != 2)
    {
        printf("Enter Port Address");
        exit(0);
    }
    int port = atoi(argv[1]);
    printf("Port: %d\n", port);
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    char response[30];
    struct sockaddr_in serverAddress;
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_addr.s_addr = INADDR_ANY;
    serverAddress.sin_port = htons(port);
    connect(sockfd, (struct sockaddr*)&serverAddress, sizeof(serverAddress));
    printf("Connected to the server\n");
    recv(sockfd, response, 29, 0);
    printf("Time from server: %s", response);
    return 0;
}
```

Server-

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include <time.h>
#define BACKLOG 10
```

```

int main(int argc, char **argv)
{
    if(argc != 2)
    {
        printf("Enter the Port No \n");
        exit(0);
    }
    int port = atoi(argv[1]);
    int n_client = 0;
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in serverAddress;
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_addr.s_addr = INADDR_ANY;
    serverAddress.sin_port = htons(port);
    bind(sockfd, (struct sockaddr*)&serverAddress, sizeof(serverAddress));
    listen(sockfd, BACKLOG);
    printf("Listening on port %d\n", port);
    int i = 1;
    while(i)
    {
        int client_socket = accept(sockfd, NULL, NULL);
        n_client++;
        time_t currentTime;
        time(&currentTime);
        printf("Client %d requested at %s", n_client, ctime(&currentTime));
        send(client_socket, ctime(&currentTime), 30, 0);
    }
    return 0;
}

```

Output-

The image shows two terminal windows side-by-side. The left window shows the output of a client program (eio3client.c) being run multiple times. Each time, it connects to a server on port 53662 and prints the time from the server. The right window shows the output of a server program (eio3server.c) being run. It listens on port 53662 and prints the time from the server for each client connection, showing that the server is receiving requests from multiple clients at different times.

```

shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ ./a.out 53662
Port: 53662
Connected to the server
Time from server: Tue Nov  3 23:49:55 2020
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ gcc eio3client.c
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ ./a.out 53662
Port: 53662
Connected to the server
Time from server: Tue Nov  3 23:50:23 2020
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ gcc eio3client.c
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ ./a.out 53662
Port: 53662
Connected to the server
Time from server: Tue Nov  3 23:50:35 2020
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ gcc eio3client.c
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ ./a.out 53662
Port: 53662
Connected to the server
Time from server: Tue Nov  3 23:51:01 2020
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ gcc eio3client.c
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ ./a.out 53662
Port: 53662
Connected to the server
Time from server: Tue Nov  3 23:51:08 2020
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ ./a.out 53662

```

```

Time from server: Tue Nov  3 23:49:55 2020
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ gcc eio3server.c
shatzz@LAPTOP-A2PUH1OV:~/CNLAB$ ./a.out 53662
Listening on port 53662
Client 1 requested at Tue Nov  3 23:49:55 2020
Client 2 requested at Tue Nov  3 23:50:23 2020
Client 3 requested at Tue Nov  3 23:50:35 2020
Client 4 requested at Tue Nov  3 23:51:01 2020
Client 5 requested at Tue Nov  3 23:51:08 2020

```