

NAME – AKASH KUMAR SINGH

REGD NO. 1841017025

CSIT A

CN LAB 1

OBJECTIVE 1>

TO IMPLEMENT AN ECHO CLIENT SERVER USING TCP/IP.

CLIENT

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<netdb.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;
}

```

SERVER

```

#include<stdio.h>
#include<netinet/in.h>
#include<netdb.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;
}

```

Output–

```

client.c  server.c
LAB_1 > echo_server > client.c > main(int, char *[])
10 struct sockaddr_in serv_addr;
11 struct hostent *server;
12 char buffer[4096];
13 sockfd=socket(AF_INET,SOCK_STREAM,0);
14 serv_addr.sin_family=AF_INET;
15 serv_addr.sin_addr.s_addr=inet_addr("127.0.0.1");
16 serv_addr.sin_port=htons(SERV_TCP_PORT);
17 printf("\nConnected");
You, 7 days ago • echo server program done

OUTPUT TERMINAL DEBUG CONSOLE SQL CONSOLE
root@DESKTOP-D020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sew5/CN/CN-lab/LAB_1/echo_server# gcc -w ser
ver.c
root@DESKTOP-D020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sew5/CN/CN-lab/LAB_1/echo_server# ./a.out
Server is listening..
Client accepted
Client message:hello
root@DESKTOP-D020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sew5/CN/CN-lab/LAB_1/echo_server#

4: wsl, wsl
root@DESKTOP-D020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sew5/CN/CN-lab/LAB_1/echo_server# gcc -w cli
ent.c
root@DESKTOP-D020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sew5/CN/CN-lab/LAB_1/echo_server# ./a.out
Connected
Enter the message
Client: hello
echo message by server: hello
root@DESKTOP-D020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sew5/CN/CN-lab/LAB_1/echo_server#

```

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, s
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");

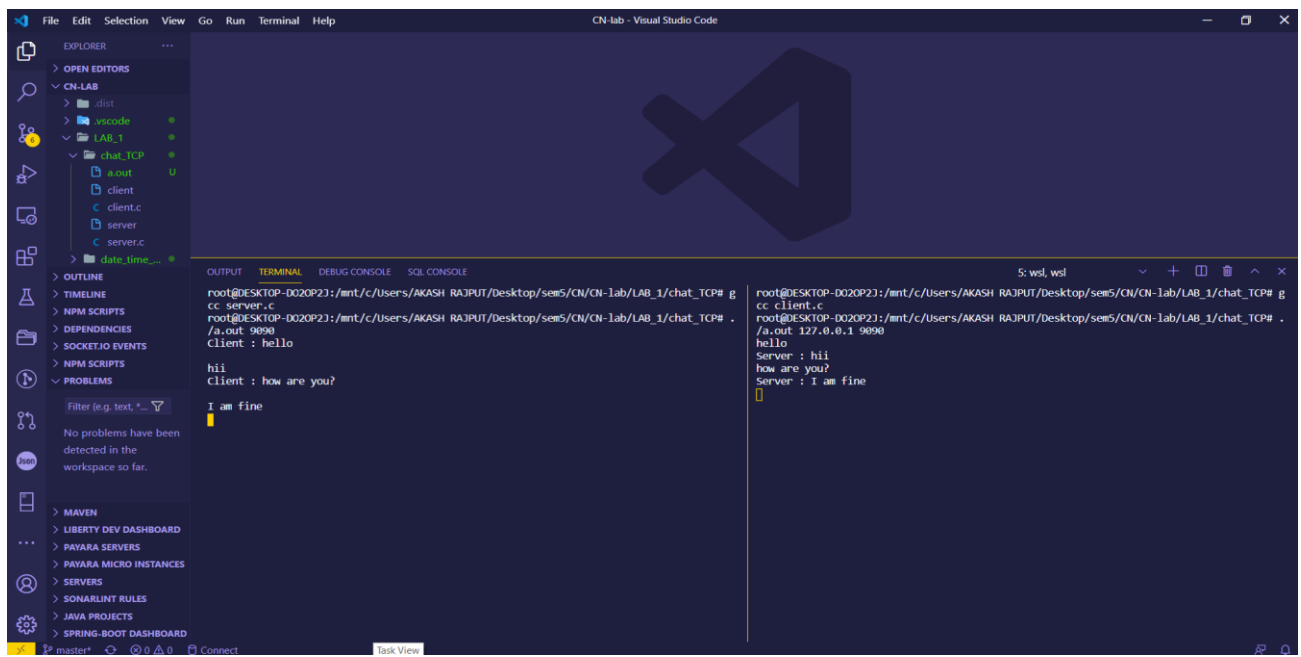
listen(sockfd,5);
clilen = sizeof(cli_addr);
newsockfd = accept(sockfd,(struct sockaddr *) &cli_addr,&clilen);
if(newsockfd < 0)
    error("Error on accept.");

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–



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);

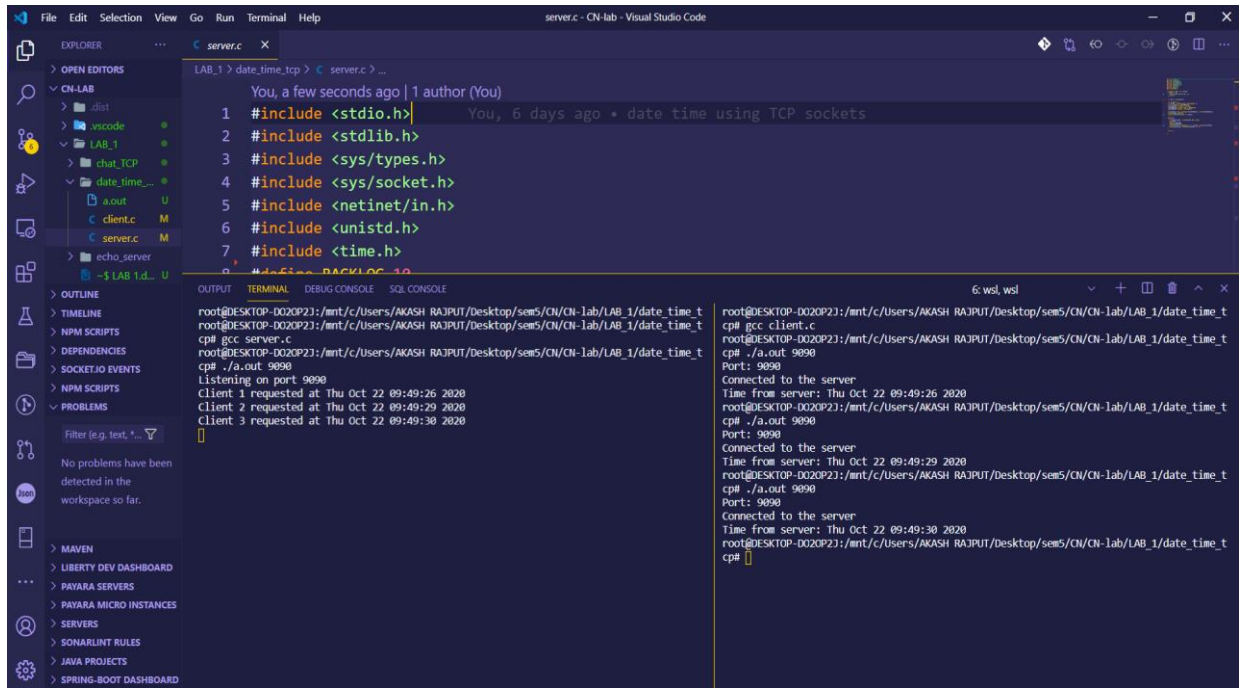
    int i = 1;
    while(i){
        int client_socket = accept(sockfd, NULL, NULL);
        n_client++;
        time_t currentTime;
        time(&currentTime);
        printf("Client %d requested at %s", n_client, ctime(&currentTime));
        send(client_socket, ctime(&currentTime), 30, 0);
    }

    return 0;
}

```

```
}
```

Output-



```
server.c - CN-lab - Visual Studio Code

EXPLORER
  > OPEN EDITORS
  > CN-LAB
    > .dist
    > .vscode
    > LAB_1
    > chat_TCP
    > date_time_tcp
      a.out
      client.c
      server.c
    > echo_server
    > -LAB 1.d... U

  > OUTLINE
  > TIMELINE
  > NPM SCRIPTS
  > DEPENDENCIES
  > SOCKET.IO EVENTS
  > NPM SCRIPTS
  > PROBLEMS
    Filter (e.g. text, *...)
    No problems have been detected in the workspace so far.

  > MAVEN
  > LIBERTY DEV DASHBOARD
  > PAIARA SERVERS
  > PAIARA MICRO INSTANCES
  > SERVERS
  > SONARLINT RULES
  > JAVA PROJECTS
  > SPRING-BOOT DASHBOARD

server.c
  1 You, a few seconds ago | 1 author (You)
  2 You, 6 days ago * date time using TCP sockets
  3
  4 #include <stdio.h>
  5 #include <stdlib.h>
  6 #include <sys/types.h>
  7 #include <sys/socket.h>
  8 #include <netinet/in.h>
  9 #include <unistd.h>
 10 #include <time.h>
 11
 12 int main()
 13 {
 14     int sockfd, new_fd;
 15     struct sockaddr_in serv_addr, client_addr;
 16     socklen_t sin_size;
 17     char s[1024];
 18     time_t now;
 19     struct tm *p_tm;
 20     char *date_time;
 21     int n;
 22
 23     sockfd = socket(AF_INET, SOCK_STREAM, 0);
 24     if (sockfd < 0)
 25     {
 26         perror("socket failed");
 27         exit(1);
 28     }
 29     serv_addr.sin_family = AF_INET;
 30     serv_addr.sin_addr.s_addr = INADDR_ANY;
 31     serv_addr.sin_port = htons(9090);
 32
 33     if (bind(sockfd, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0)
 34     {
 35         perror("bind failed");
 36         exit(1);
 37     }
 38     listen(sockfd, 5);
 39
 40     while (1)
 41     {
 42         new_fd = accept(sockfd, (struct sockaddr *)&client_addr, &sin_size);
 43         if (new_fd < 0)
 44             continue;
 45
 46         now = time(NULL);
 47         p_tm = localtime(&now);
 48         date_time = asctime(p_tm);
 49
 50         n = sprintf(s, "You, %s\n", date_time);
 51         write(new_fd, s, n);
 52         close(new_fd);
 53     }
 54
 55     return 0;
 56 }
```

```
root@DESKTOP-0020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sem5/CN/CN-lab/LAB_1/date_time_tcp# gcc client.c
root@DESKTOP-0020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sem5/CN/CN-lab/LAB_1/date_time_tcp# ./a.out 9090
Port: 9090
Connected to the server
Time from server: Thu Oct 22 09:49:26 2020
root@DESKTOP-0020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sem5/CN/CN-lab/LAB_1/date_time_tcp# ./a.out 9090
Port: 9090
Connected to the server
Time from server: Thu Oct 22 09:49:29 2020
root@DESKTOP-0020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sem5/CN/CN-lab/LAB_1/date_time_tcp# ./a.out 9090
Port: 9090
Connected to the server
Time from server: Thu Oct 22 09:49:30 2020
root@DESKTOP-0020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sem5/CN/CN-lab/LAB_1/date_time_tcp#
```

```
root@DESKTOP-0020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sem5/CN/CN-lab/LAB_1/date_time_tcp# gcc server.c
root@DESKTOP-0020P23:/mnt/c/Users/AKASH RAJPUT/Desktop/sem5/CN/CN-lab/LAB_1/date_time_tcp# ./a.out 9090
Listening on port 9090
Client 1 requested at Thu Oct 22 09:49:26 2020
Client 2 requested at Thu Oct 22 09:49:29 2020
Client 3 requested at Thu Oct 22 09:49:30 2020
```