Parth Shukla
190905104
Lab 1

Q) Write a client and server program to show data exchange using UDP protocol.

// UDP server

```
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/in.h>
#define PORT 5000
#define MAXLINE 1000

int main(){
    char buffer[100];
    char *message = "hello client";
    int listenfd,len,n;
    struct sockaddr_in servaddr,cliaddr;
    bzero(&servaddr,sizeof(servaddr));

    //create UDP socket
    listenfd = socket(AF_INET,SOCK_DGRAM,0);
    if(listenfd == -1){
        printf("Error in creating socket");
        exit(1);
    }
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    servaddr.sin_port = htons(PORT);
    servaddr.sin_family = AF_INET;

    //bind server address to socket descriptor
    bind(listenfd,(struct sockaddr*)&servaddr,sizeof(servaddr));

    //receive the datagram
    len = sizeof(cliaddr);

    n = recvfrom(listenfd,buffer,sizeof(buffer),0,(struct sockaddr*)&cliaddr,&len);
    buffer[n] = '\0';
    // puts(buffer);
    printf("message from client:%s \n",buffer);

    sendto(listenfd,buffer,n,0,(struct sockaddr*)&cliaddr,sizeof(cliaddr));
    getchar();
    close(listenfd);
}
```

// UDP client

```c
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include <stdlib.h>
#define PORT 5000
#define MAXLINE 1000

int main(){
    char buffer[100];
    char *message = "hello server";
    int sockfd,len,n;
    struct sockaddr_in servaddr,cliaddr;
    //clear servaddr
    bzero(&servaddr,sizeof(servaddr));

    //create UDP socket
    sockfd = socket(AF_INET,SOCK_DGRAM,0);
    if(sockfd == -1){
        printf("Error in socket\n");
        exit(1);
    }
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    servaddr.sin_port = htons(PORT);
    servaddr.sin_family = AF_INET;

    //bind client address to socket descriptor
    // bind(sockfd,(struct sockaddr*)&cliaddr,sizeof(cliaddr));

    sendto(sockfd,message,MAXLINE,0,(struct sockaddr*)&servaddr,sizeof(servaddr));
    len = sizeof(cliaddr);

    n = recvfrom(sockfd,buffer,sizeof(buffer),0,(struct sockaddr*)&cliaddr,&len);
    buffer[n] = '\0';
    printf("message to server:%s \n",buffer);
    getchar();
    close(sockfd);
}
```

```
    pclose
Student@project-lab:~/190905104_CN/lab1$ ./udpser
message from client:hello server
```

```
Student@project-lab:~/190905104_CN/lab1$ gcc udpcli.c -o udpcli
Student@project-lab:~/190905104_CN/lab1$ ./udpcli
message to server:hello server
```

Q) Write a client and server program to show data exchange using TCP protocol.

// TCP server

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/wait.h>
#include <signal.h>


int main(){
        /* Declare descriptors
        Socket descriptor: handles connection
        Network descriptor: handles data transmission
        */
        int sd,nd,n,len,reult;
        struct sockaddr_in seraddress, cliaddr;
        // Buffer of size 256 bytes
        char buf[256];
        // Creating socket - SOCK_STREAM to define that it is a connection based socket
        sd=socket(AF_INET, SOCK_STREAM,0);
        if(sd == -1){
                printf("Error in creating socket");
      exit(1);
        }
        seraddress.sin_family=AF_INET;
        seraddress.sin_addr.s_addr=INADDR_ANY;
        seraddress.sin_port=htons(10200);

        // Bind socket
        bind(sd,(struct sockaddr*)&seraddress,sizeof(seraddress));
        // Listening to queue
        listen(sd,5);
        len=sizeof(cliaddr);
        // Accepting connection from client
        nd=accept(sd,(struct sockaddr*)&cliaddr,&len);
```

```
        // Read data
        n=read(nd,buf,sizeof(buf));
    buf[n]='\0';
    printf("message from client: %s\n",buf);
    n=write(nd,buf,strlen(buf));
        getchar();
    close(nd);
    close(sd);
}


// TCP Client


#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/wait.h>
#include <signal.h>


int main(){
        int sd,nd,n,len,reult,n1;
        struct sockaddr_in seraddress, cliaddr;

        char buf[256], buf1[256];
        // Creating socket for the client
        sd=socket(AF_INET, SOCK_STREAM,0);
        if(sd == -1){
                printf("Error in creating socket");
      exit(1);
        }
        seraddress.sin_family=AF_INET;
        seraddress.sin_addr.s_addr=INADDR_ANY;
        seraddress.sin_port=htons(10200);
        len=sizeof(seraddress);
        // Client establishes connection
        connect(sd,(struct sockaddr*)&seraddress,len);
        printf("enter the message to send: \n");
        gets(buf);
        // Writing to the buffer
        n=write(sd,buf,strlen(buf));
        // reading from the buffer returned by server
    n1=read(sd,buf1,sizeof(buf1));
    buf1[n1]='\0';
```

```
        printf("message from server: %s\n",buf1);
        getchar();
    close(sd);
}
```

```
Student@project-lab:~/190905104_CN/lab1$ ./tcpser
message from client: hello from client
```

```
Student@project-lab:~/190905104_CN/lab1$ ./tcpcli
enter the message to send:
hello from client
message from server: hello from client
```

1) Write a UDP client-server program where client sends rows of a matrix to the server combines them together as a two dimensional matrix and display the same.

```
^C
Student@project-lab:~/190905104_CN/lab1$ ./udpser

1 2 3
4 5 6
7 8 9
```

```
4 5 6
Student@project-lab:~/190905104_CN/lab1$ gcc udpcli_matrix.c -o udpcli
Student@project-lab:~/190905104_CN/lab1$ ./udpcli
Enter dimension of matrix
3 3
Enter the matrix
1 2 3
4 5 6
7 8 9
Student@project-lab:~/190905104_CN/lab1$
```