

AASHNA KUNKOLIENKAR
LAB 1
CN LAB
190905304

Question 1)

UDP server-client implementation

Code for server side:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#define PORT 5004
#define MAXLINE 1000

int main()
{
    char buffer[100];
    char *message="Hello server";
    int listenfd, len,n;
    struct sockaddr_in servaddr,cliaddr;
    servaddr.sin_port=htons(PORT);
    servaddr.sin_family=AF_INET;
    servaddr.sin_addr.s_addr=inet_addr("127.0.0.1");

    bind(listenfd,(struct sockaddr*)&servaddr,sizeof(servaddr));
    len=sizeof(cliaddr);
    n=recvfrom(listenfd,buffer,sizeof(buffer),0,(struct sockaddr*)&cliaddr,&len);
    buffer[n]='\0';
    puts(buffer);

    sendto(listenfd,buffer,n,0,(struct sockaddr*)&cliaddr, len);
    printf("The hello message is sent to client");

    return 0;
}
```

```
$ gcc udpserver.c -o udpserver
$ ./udpserver
```

```
The hello message is sent to client$ □
```

Code for client side:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#define PORT 5004
#define MAXLINE 1000

int main()
{
    char buffer[100];
    char *message="Hello from client";
    int listenfd, len,n;
    struct sockaddr_in servaddr,cliaddr;
    servaddr.sin_port=htons(PORT);
    servaddr.sin_family=AF_INET;
    servaddr.sin_addr.s_addr=inet_addr("127.0.0.1");

    sendto(listenfd,message,sizeof(message),0,(struct sockaddr*)&servaddr, len);

    len=sizeof(cliaddr);
    n=recvfrom(listenfd,buffer,sizeof(buffer),0,(struct sockaddr*)&servaddr,&len);
    buffer[n]='\0';
    puts(buffer);

    printf("The hello message is sent to server");

    close(listenfd);
    return 0;
}
```

```
$ gcc udpclient.c -o udpclient
$ ./udpclient

The hello message is sent to server$
```

QUESTION 2)

TCP client-server implementation:

Code for the server side:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#define PORT 5004
#define MAXLINE 1000

int main()
{
    int sd,nd,n,len,result;
    struct sockaddr_in seraddress, cliaddress;
    char buf[256],buf1[256];
    sd=socket(AF_INET,SOCK_STREAM,0);
    bzero(&seraddress, sizeof(seraddress));
    seraddress.sin_port=htons(PORT);
    seraddress.sin_family=AF_INET;
    seraddress.sin_addr.s_addr=inet_addr("127.0.0.1");

    bind(sd,(struct sockaddr*)&seraddress,sizeof(seraddress));

    listen(sd,5);
    len=sizeof(cliaddress);

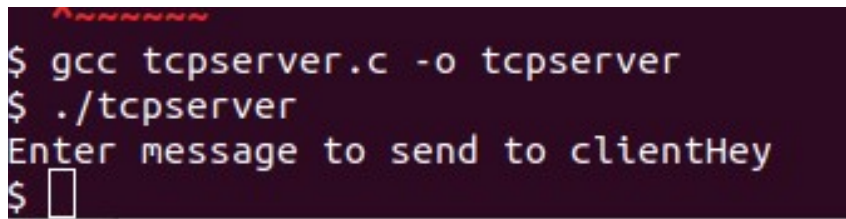
    accept(sd, (struct sockaddr*)&cliaddress,&len);
    printf("Enter message to send to client");
    scanf("%s",buf);

    n=write(sd,buf,strlen(buf));

    getchar();

    close(sd);
}
```

```
}
```



```
$ gcc tcpserver.c -o tcpserver
$ ./tcpserver
Enter message to send to clientHey
$
```

Code for the client side :

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#define PORT 5004
#define MAXLINE 1000

int main()
{
    int sd,nd,n,len,result;
    struct sockaddr_in seraddress, cliaddress;
    char buf[256],buf1[256];
    sd=socket(AF_INET,SOCK_STREAM,0);
    seraddress.sin_port=htons(PORT);
    seraddress.sin_family=AF_INET;
    seraddress.sin_addr.s_addr=inet_addr("127.0.0.1");

    len=sizeof(seraddress);
    connect(sd,(struct sockaddr*)&seraddress,len);
    printf("Connected to server...");
    printf("Enter message to send to server");
    scanf("%s",buf);
    printf(" \n Message sent to server");
    n=write(sd,buf,strlen(buf));
    getchar();
    close(nd);
}
```

```
    close(sd);  
}
```

```
$ gcc tcpcli.c -o tcpcli  
$ ./tcpcli  
Connected to server...Enter message to send to serverHello  
Message sent to server$
```