



## CN LAB-18.8.21

NAME: Ankit Raj

ROLL NO: 1906534

Q.1. UDP SOCKET CLIENT WILL SEND A MESSAGE, SERVER WILL ECHO BACK.

### Server.c

```
#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 <arpa/inet.h>
#define MYPORT 4952
#define MAXBUFLEN 200
int main()
{
    int sockfd;
    struct sockaddr_in my_addr;
    struct sockaddr_in their_addr;
    socklen_t addr_len;
    int numbytes;
    char buf[MAXBUFLEN];
    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
        perror("socket");
        exit(1);
    }
    my_addr.sin_family = AF_INET;
    my_addr.sin_port = htons(MYPORT);
    my_addr.sin_addr.s_addr = INADDR_ANY;
    if (bind(sockfd, (struct sockaddr *)&my_addr, sizeof my_addr) == -1) {
        perror("bind");
        exit(1);
    }
    addr_len = sizeof their_addr;
    if ((numbytes = recvfrom(sockfd, buf, MAXBUFLEN-1, 0,
        (struct sockaddr *)&their_addr, &addr_len)) == -1) {
        perror("recvfrom");
        exit(1);
    }
    printf("got packet from %s\n", inet_ntoa(their_addr.sin_addr));
    printf("packet is %d bytes long\n", numbytes);
    buf[numbytes] = '\0';
    printf("packet contains \"%s\"\n", buf);
    close(sockfd);
    return 0;
}
```

### Client.c

```
#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 <arpa/inet.h>
#include <netdb.h>
#define SERVERPORT 4952
```

```

int main()
{
int sockfd;
struct sockaddr_in their_addr;
int numbytes;
char arg[30];
if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
perror("socket");
exit(1);
}
their_addr.sin_family = AF_INET;
their_addr.sin_port = htons(SERVERPORT);
their_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
printf("Enter a message\n");
gets(arg);
if ((numbytes = sendto(sockfd, arg, strlen(arg), 0,
(struct sockaddr *)&their_addr, sizeof their_addr)) == -1) {
perror("sendto");
exit(1);
}
printf("sent %d bytes to %s\n", numbytes, inet_ntoa(their_addr.sin_addr));
close(sockfd);
return 0;
}

```

## OUTPUT:

The first terminal window shows the server running. It receives a packet from 127.0.0.1, which is 5 bytes long and contains the string "hello".

```

kiit@kiit-VirtualBox: ~/1906534/CN/lab5/Q1
File Edit View Search Terminal Help
kiit@kiit-VirtualBox:~/1906534/CN/lab5/Q1$ ./server
got packet from 127.0.0.1
packet is 5 bytes long
packet contains "hello"
kiit@kiit-VirtualBox:~/1906534/CN/lab5/Q1$ 

```

The second terminal window shows the client running. It prompts the user to enter a message, and the user enters "hello". The client then sends 5 bytes to 127.0.0.1.

```

kiit@kiit-VirtualBox: ~/1906534/CN/lab5/Q1
File Edit View Search Terminal Help
kiit@kiit-VirtualBox:~/1906534/CN/lab5/Q1$ ./client
Enter a message
hello
sent 5 bytes to 127.0.0.1
kiit@kiit-VirtualBox:~/1906534/CN/lab5/Q1$ 

```

## Q.2. UDP SOCKET CLIENT WILL SEND INTEGER, SERVER WILL RETURN REVERSE OF IT.

### Server.c

```

#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 <arpa/inet.h>
#define MYPORT 4952
#define MAXBUFLEN 200
int main()
{
int sockfd;
struct sockaddr_in my_addr;
struct sockaddr_in their_addr;
socklen_t addr_len;
int numbytes;
int buf;
if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
perror("socket");
exit(1);
}
my_addr.sin_family = AF_INET;
my_addr.sin_port = htons(MYPORT);
my_addr.sin_addr.s_addr = INADDR_ANY;
if (bind(sockfd, (struct sockaddr *)&my_addr, sizeof my_addr) == -1) {
perror("bind");
}
}

```

```

exit(1);
}
addr_len = sizeof their_addr;
if ((numbytes = recvfrom(sockfd, &buf, sizeof(buf) , 0,
(struct sockaddr *)&their_addr, &addr_len)) == -1) {
perror("recvfrom");
exit(1);
}
int x=buf;
int rev=0,j;
while(x>0)
{
j=x%10;
rev=rev*10+j;
x/=10;
}
sendto(sockfd, &rev, sizeof(rev), 0,
(struct sockaddr *)&their_addr, sizeof their_addr);
close(sockfd);
return 0;
}

```

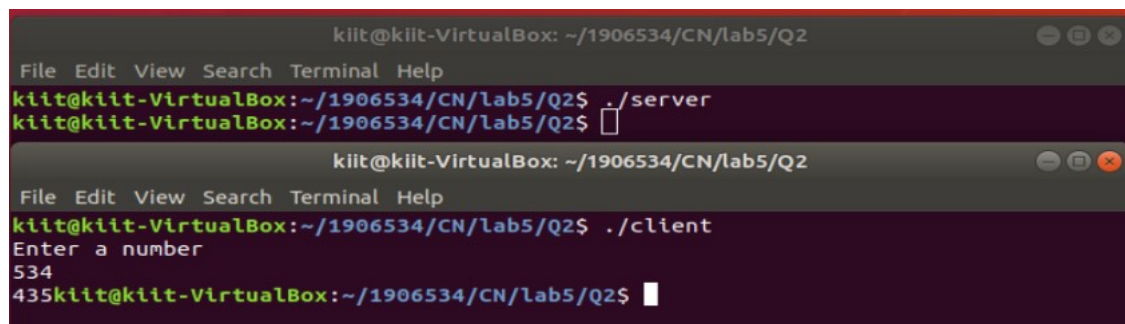
## Client.c

```

#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 <arpa/inet.h>
#include <netdb.h>
#define SERVERPORT 4952
int main()
{
int sockfd;
struct sockaddr_in their_addr;
int numbytes,arg,buf;
socklen_t addr_len;
if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
perror("socket");
exit(1);
}
their_addr.sin_family = AF_INET;
their_addr.sin_port = htons(SERVERPORT);
their_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
printf("Enter a message\n");
scanf("%d",&arg);
if ((numbytes = sendto(sockfd, &arg, sizeof(arg), 0,
(struct sockaddr *)&their_addr, sizeof their_addr)) == -1) {
perror("sendto");
exit(1);
}
recvfrom(sockfd, &buf, sizeof(buf) , 0,
(struct sockaddr *)&their_addr, &addr_len);
printf("%d",buf);
close(sockfd);
return 0;
}

```

## OUTPUT



The image shows two terminal windows from a VirtualBox environment. The top window shows the execution of the server program, where the user enters a number and the program runs successfully. The bottom window shows the execution of the client program, where the user enters a number and the program receives a response from the server.

```

kiit@kiit-VirtualBox: ~/1906534/CN/lab5/Q2
File Edit View Search Terminal Help
kiit@kiit-VirtualBox:~/1906534/CN/lab5/Q2$ ./server
kiit@kiit-VirtualBox:~/1906534/CN/lab5/Q2$

kiit@kiit-VirtualBox: ~/1906534/CN/lab5/Q2
File Edit View Search Terminal Help
kiit@kiit-VirtualBox:~/1906534/CN/lab5/Q2$ ./client
Enter a number
534
435kiit@kiit-VirtualBox:~/1906534/CN/lab5/Q2$

```

### Q.3. UDP SOCKET CLIENT WILL SEND INT ARRAY, SERVER WILL RETURN SORTED ARRAY ( SEND THE DIGITS OF YOUR ROLL NUMBER). BOTH SERVER AND CLIENT DISPLAY SORTED ARRAY.

#### Server.c

```
#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 <arpa/inet.h>
#define MYPOR 4952
#define MAXBUFL 200
int main()
{
    int sockfd;
    struct sockaddr_in my_addr;
    struct sockaddr_in their_addr;
    socklen_t addr_len;
    int numbytes;
    int buf;
    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
        perror("socket");
        exit(1);
    }
    my_addr.sin_family = AF_INET;
    my_addr.sin_port = htons(MYPOR);
    my_addr.sin_addr.s_addr = INADDR_ANY;
    if (bind(sockfd, (struct sockaddr *)&my_addr, sizeof my_addr) == -1) {
        perror("bind");
        exit(1);
    }
    addr_len = sizeof their_addr;
    int a[7];
    for(int i=0;i<7;i++)
    {
        recvfrom(sockfd, &buf, sizeof(buf), 0,
        (struct sockaddr *)&their_addr, &addr_len);
        a[i]=buf;
    }
    for(int i=0;i<7;i++)
    {
        int temp;
        for(int j=0;j<7-i-1;j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
    for(int i=0;i<7;i++)
    {
        sendto(sockfd, &a[i], sizeof(a[i]), 0,
        (struct sockaddr *)&their_addr, sizeof their_addr);
    }
    close(sockfd);
    return 0;
}
```

#### Client.c

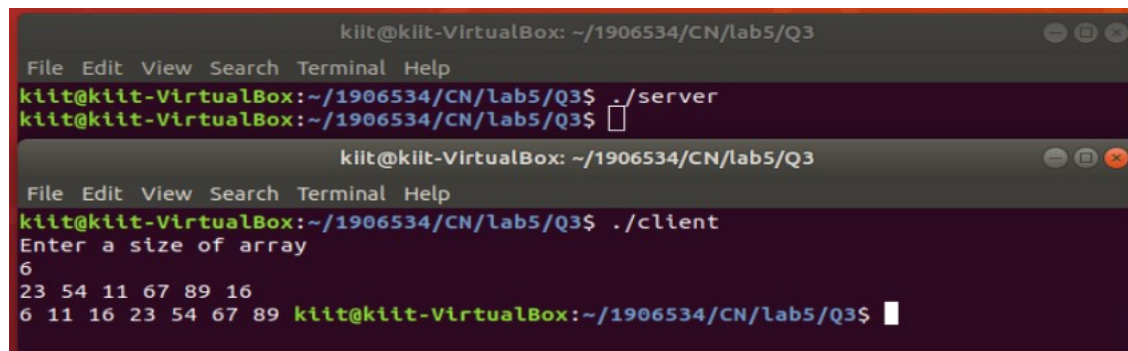
```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
```

```

#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#define SERVERPORT 4952
int main()
{
    int sockfd;
    struct sockaddr_in their_addr;
    int numbytes,arg,buf;
    socklen_t addr_len;
    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
        perror("socket");
        exit(1);
    }
    their_addr.sin_family = AF_INET;
    their_addr.sin_port = htons(SERVERPORT);
    their_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    printf("Enter a message\n");
    for(int i=0;i<7;i++)
    {scanf("%d",&arg);
    sendto(sockfd, &arg, sizeof(arg), 0,
    (struct sockaddr *)&their_addr, sizeof their_addr);
    }
    int a[7];
    for(int i=0;i<7;i++)
    {
    recvfrom(sockfd, &buf, sizeof(buf) , 0,
    (struct sockaddr *)&their_addr, &addr_len);
    printf("%d ",buf);
    }
    close(sockfd);
    return 0;
}

```

## OUTPUT



The screenshot shows two terminal windows from a VirtualBox environment. The top window shows the execution of the server program, which prompts for a message. The bottom window shows the execution of the client program, which prompts for the size of an array and then displays a sequence of numbers.

```

kiit@kiit-VirtualBox: ~/1906534/CN/lab5/Q3
File Edit View Search Terminal Help
kiit@kiit-VirtualBox:~/1906534/CN/lab5/Q3$ ./server
kiit@kiit-VirtualBox:~/1906534/CN/lab5/Q3$

kiit@kiit-VirtualBox: ~/1906534/CN/lab5/Q3
File Edit View Search Terminal Help
kiit@kiit-VirtualBox:~/1906534/CN/lab5/Q3$ ./client
Enter a size of array
6
23 54 11 67 89 16
6 11 16 23 54 67 89 kiit@kiit-VirtualBox:~/1906534/CN/lab5/Q3$

```