# Kalinga Institute of Industrial Technology

# CN LAB-18.8.21

NAME: Ankit Raj
ROLL NO: 1906534

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

Server.c

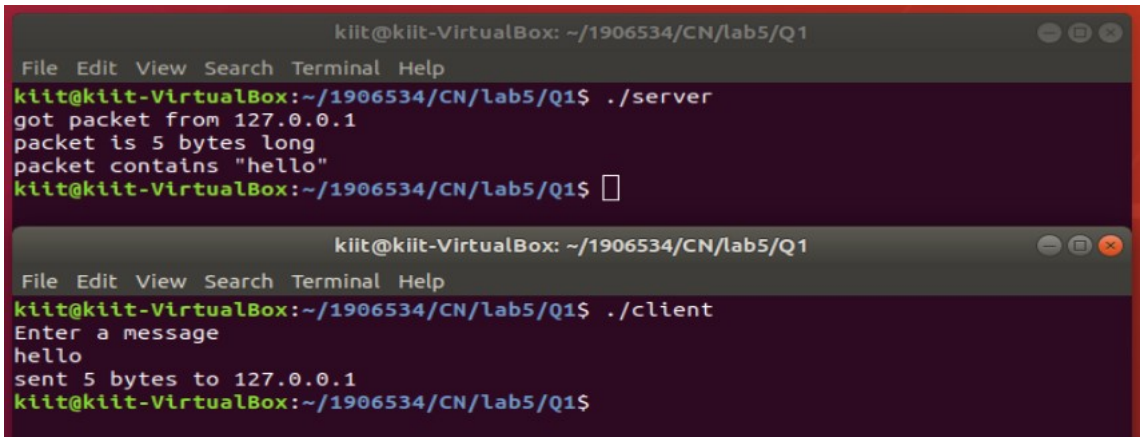
```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#define MYPORT 4952
#define MAXBUFLEN 200
int main()
{
int sockfd;
struct sockaddr_in my_addr;
struct sockaddr_in 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

```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
```

```c
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:**




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

Server.c

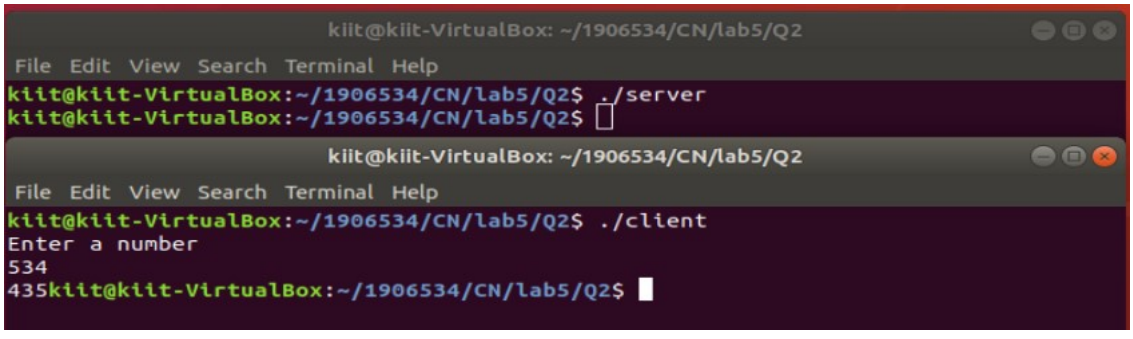
```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#define MYPORT 4952
#define MAXBUFLEN 200
int main()
{
int sockfd;
struct sockaddr_in my_addr;
struct sockaddr_in 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



## 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

```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;
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

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

```c
#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