CN LAB

-1906555

Gourav Samantaray

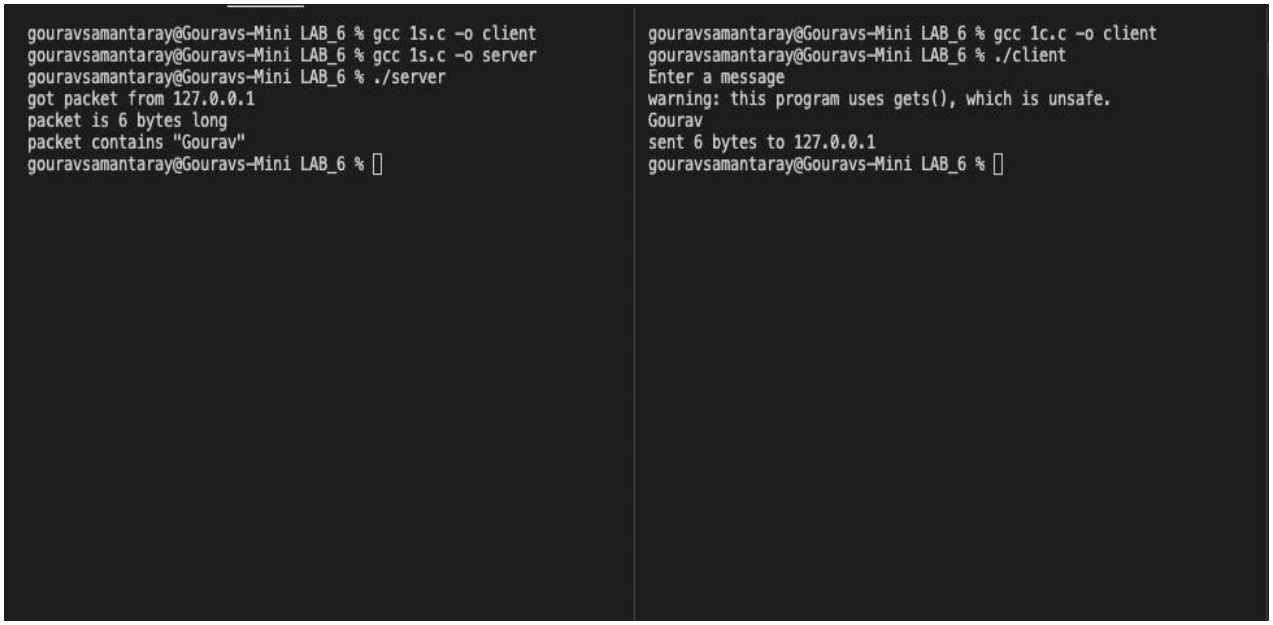
1. Client will send a message and the server will echo it back.

```c
/*
** A datagram sockets "server" demo
*/
#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 // the port users will be connecting to
#define MAXBUFLEN 200
int main()
{
int sockfd;
struct sockaddr_in my_addr; // my address information
struct sockaddr_in their_addr; // connector's address
information
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; // host byte order
my_addr.sin_port = htons(MYPORT); // short, network byte
order
my_addr.sin_addr.s_addr = INADDR_ANY; // automatically fill
with my IP
//memset(my_addr.sin_zero, '\0', sizeof my_addr.sin_zero);
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;
}
```

```
/*
** A datagram "client" demo
*/
#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 // the port users will be connecting
to
int main()
{
int sockfd;
struct sockaddr_in their_addr; // connector's address
information
//struct hostent *he;
int numbytes;
char arg[30];
if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
perror("socket");
exit(1);
}
their_addr.sin_family = AF_INET; // host byte order
their_addr.sin_port = htons(SERVERPORT); // short, network
byte order
their_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
//memset(their_addr.sin_zero, '\0', sizeof
their_addr.sin_zero);
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;
}
```



2. Program to reverse a number sent by client and returning the reversed number by the server to client

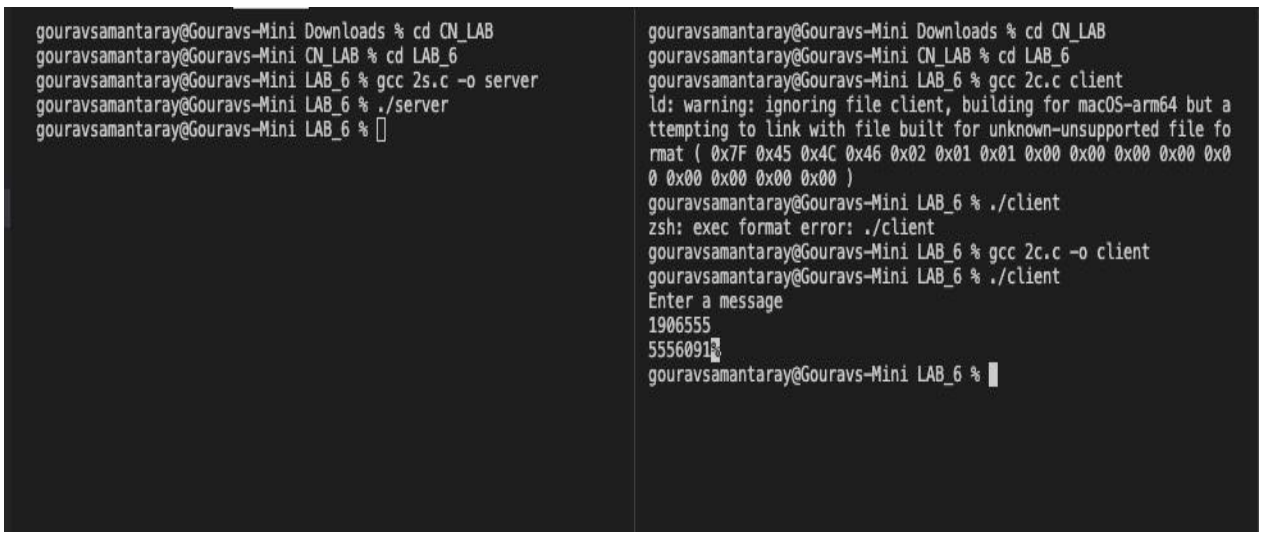
```c
/*
** A datagram sockets "server" demo
*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#define MYPORT 4952 // the port users will be connecting to
#define MAXBUFLEN 200
int main()
{
int sockfd;
```

```
struct sockaddr_in my_addr; // my address information
struct sockaddr_in their_addr; // connector's address
information
socklen_t addr_len;
int numbytes;
//char buf[MAXBUFLEN];
int buf;
if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
perror("socket");
exit(1);
}
my_addr.sin_family = AF_INET; // host byte order
my_addr.sin_port = htons(MYPORT); // short, network byte
order
my_addr.sin_addr.s_addr = INADDR_ANY; // automatically fill
with my IP
//memset(my_addr.sin_zero, '\0', sizeof my_addr.sin_zero);
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);
}
//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 %d ",buf);
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;
}
```

```
/*
** A datagram "client" demo
*/
#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 // the port users will be connecting
to
int main()
{
int sockfd;
struct sockaddr_in their_addr; // connector's address
information
//struct hostent *he;
int numbytes,arg,buf;
socklen_t addr_len;
//char arg[30];
if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
perror("socket");
exit(1);
}
```

```
their_addr.sin_family = AF_INET; // host byte order
their_addr.sin_port = htons(SERVERPORT); // short, network
byte order
their_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
//memset(their_addr.sin_zero, '\0', sizeof
their_addr.sin_zero);
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);
}
//printf("sent %d bytes to %s\n", numbytes,
inet_ntoa(their_addr.sin_addr));
recvfrom(sockfd, &buf, sizeof(buf) , 0,
(struct sockaddr *)&their_addr, &addr_len);
printf("%d",buf);
close(sockfd);
return 0;
}
```



3. Client will send an array of numbers server will sort and return it.

```
/*
** A datagram sockets "server" demo
*/
#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 // the port users will be connecting to
#define MAXBUFLEN 200
int main()
{
int sockfd;
struct sockaddr_in my_addr; // my address information
struct sockaddr_in their_addr; // connector's address
information
socklen_t addr_len;
int numbytes;
//char buf[MAXBUFLEN];
int buf;
if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
perror("socket");
exit(1);
}
my_addr.sin_family = AF_INET; // host byte order
my_addr.sin_port = htons(MYPORT); // short, network byte
order
my_addr.sin_addr.s_addr = INADDR_ANY; // automatically fill
with my IP
```

```
//memset(my_addr.sin_zero, '\0', sizeof my_addr.sin_zero);
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;
}
}
}
//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 %d ",buf);
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;
}
```

```
/*
** A datagram "client" demo
*/
#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 // the port users will be connecting
to
int main()
{
int sockfd;
struct sockaddr_in their_addr; // connector's address
information
//struct hostent *he;
int numbytes,arg,buf;
socklen_t addr_len;
//char arg[30];
if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
perror("socket");
exit(1);
}
their_addr.sin_family = AF_INET; // host byte order
their_addr.sin_port = htons(SERVERPORT); // short, network
byte order
their_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
//memset(their_addr.sin_zero, '\0', sizeof
their_addr.sin_zero);
```

```
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];
//printf("sent %d bytes to %s\n", numbytes,
inet_ntoa(their_addr.sin_addr));
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;
}
```

```
gouravsamantaray@Gouravs-Mini LAB_6 % gcc 3s.c -o server
gouravsamantaray@Gouravs-Mini LAB_6 % ./server
gouravsamantaray@Gouravs-Mini LAB_6 %
```

```
gouravsamantaray@Gouravs-Mini LAB_6 % gcc 3c.c -o client
gouravsamantaray@Gouravs-Mini LAB_6 % ./client
Enter a message
1
9
0
6
5
5
5
0 1 5 5 5 6 9
gouravsamantaray@Gouravs-Mini LAB_6 %
```