

Socket Programming

Our Example Programs

Application No.	Program Description	Related Source Files
1	Setting up a connection to the Web server (any server) using its IP address	1_client_google_using_ip.c (No server code needed from our side)
2	Data Communication to the Web server (any server) server using its IP address	2_client_data_send_to_web_server.c (No server code needed from our side)
3	Bi-directional data communication between the Web server and client	3_client_receive_response_from_web_server.c (No server code needed from our side)
4	Fetching IP address based on hostname (No need to remember IP address anymore)	4_client_get_web_server_ip_address.c (No server code needed from our side)
5	Client Server communication using TCP (Server listening on fixed port 8080)	TCP_Server.c TCP_Client.c
6	Client Server communication using UDP (Server listening on fixed port 8080)	UDP_Server.c UDP_Client.c

Our Example Programs

Application No.	Program Description	Related Source Files
7	Client Server communication using TCP (Server listening on a specified port)	<code>tcp_server_v1.c</code> <code>tcp_simple_client.c</code>
8	Client Server communication using TCP (Server listening on a specified port and runs forever, forking off a separate process for each connection)	<code>tcp_server_v2.c</code> <code>tcp_simple_client.c</code>
9	Client Server communication using TCP (Server listening on a specified port and runs forever using multi-threading)	<code>tcp_server_v3.c</code> <code>tcp_simple_client.c</code>



1_client_google_using_ip.c (~/Desktop/Sockets Lab) - gedit

```
1 #include<stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <sys/types.h>
5 #include<sys/socket.h>
6 #include<arpa/inet.h> //inet_addr
7 #include <netinet/in.h>
8
9 int main(int argc , char *argv[])
10 {
11     int socket_desc;
12     struct sockaddr_in server;
13
14     //Create socket
15     socket_desc = socket(AF_INET , SOCK_STREAM , 0);
16
17     if (socket_desc == -1)
18     {
19         printf("Could not create socket");
20     }
21
22     // check IP address passed to inet_addr() using other program client_get_server_ip_address.c as it keeps changing
23
24     server.sin_addr.s_addr = inet_addr("172.217.166.68");
25     server.sin_family = AF_INET;
26     server.sin_port = htons( 80 );
27
28     //Connect to remote server
29     if (connect(socket_desc , (struct sockaddr *)&server , sizeof(server)) < 0)
30     {
31         puts("connect error");
32         return 1;
33     }
34
35     puts("Connected");
36     return 0;
37 }
```

jayprakash@jayprakash-System-AsusPG500: ~/Desktop/Sockets Lab

```
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ gcc 1_client_google_using_ip.c -o 1.o
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ ./1.o
Connected
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$
```

The screenshot shows a typical Linux desktop environment with several windows open:

- Code Editor:** A window titled "2_client_data_send_to_web_server.c (~/Desktop/Sockets Lab) - gedit" containing C code for a socket client. The code uses `socket()`, `connect()`, and `send()` to interact with a web server at 172.217.166.68:80.
- Terminal:** A terminal window titled "jayprakash@jayprakash-System-AsusPG500: ~/Desktop/Sockets Lab". It shows the command `gcc` being run to compile the source code into an executable named "2.o", followed by the command `./2.o` which outputs "Connected".
- File Manager:** A window titled "2_client_data_send_to_web_server.c" showing the file's contents.



3_client_receive_response_from_web_server.c (~/Desktop/Sockets Lab) - gedit

```
12
13 int main(int argc , char *argv[])
14 {
15     int socket_desc;
16     struct sockaddr_in server;
17     char *message , server_reply[2000];
18
19     //Create socket
20     socket_desc = socket(AF_INET , SOCK_STREAM , 0);
21     if (socket_desc == -1)
22     {
23         printf("Could not create socket");
24     }
25
26     // check IP address passed to inet_addr() using other program client_get_server_ip_address.c as it keeps changing
27
28     server.sin_addr.s_addr = inet_addr("172.217.166.68");
```

jayprakash@jayprakash-System-AsusPG500: ~/Desktop/Sockets Lab

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$ gcc 3_client_receive_response_from_web_server.c -o 3.o

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$./3.o

Connected

Data Send

Reply received

HTTP/1.1 200 OK

Date: Fri, 19 Jun 2020 18:43:26 GMT

Expires: -1

Cache-Control: private, max-age=0

Content-Type: text/html; charset=ISO-8859-1

P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."

Server: gws

X-XSS-Protection: 0

X-Frame-Options: SAMEORIGIN

Set-Cookie: 1P_JAR=2020-06-19-18; expires=Sun, 19-Jul-2020 18:43:26 GMT; path=/; domain=.google.com; Secure

Set-Cookie: NID=204=eQpz5HPmiz0iL0caYqHiU1d4RqMQ5EcP6lezdSUtm8X23eugMiivbhb9C7j6YCXRQfz6Kc4oVKZFq4wpDU_YR2Ier_EzZ7mTqy1Aht-_reGSBGu3oomBARggzFu5M5P7Pe_OrHYyDr5tNCioxw7CFbCeSHv54CekkbJtVoitNg; expires=Sat, 19-Dec-2020 18:43:26 GMT; path=/; domain=.google.com; HttpOnly

Accept-Ranges: none

Vary: Accept-Encoding

Transfer-Encoding: chunked

56f2

<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleleg/1x/googleleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="78D8s8oGcNiniUM3jlp+bw==">(function(){window.google={kEI:'TgftXp70N9PE4-EP4qeY2As',kEXPI:'0,202123,3,4,32,1151584,5663,731,223,5104,207,3204,10,1226,364,1499,576,241,383,246,5,971,66,317,196,211,754,533,1850,334,137,97,186,76,40,3,888,162,1122299,1197722,427,89,3

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$

```
4_client_get_web_server_ip_address.c (~/Desktop/Sockets Lab) - gedit
13 int main(int argc , char *argv[])
14 {
15     char *hostname, *input_hostname;
16 //    char *hostname = "www.daiict.ac.in";
17     char ip[100];
18     struct hostent *he;
19     struct in_addr **addr_list;
20     int i;
21
22     while(1){
23         printf("Enter hostname for which you want to fetch the IP address:");
24         scanf("%s",hostname);
25         input_hostname = hostname; // Just to store it for use in the program using input_hostname
26
27         if ( (he = gethostbyname( hostname ) ) == NULL)
28         {
29             //gethostbyname failed
30             perror("gethostbyname");
31             return 1;
32         }
33
34         //Cast the h_addr_list to in_addr , since h_addr_list also has the ip address in long format only
35         addr_list = (struct in_addr **) he->h_addr_list;
36
37         for(i = 0; addr_list[i] != NULL; i++)
38         {
39             //Return the first one;
40             strcpy(ip , inet_ntoa(*addr_list[i]) );
41         }
42
43         printf(" %s Host name resolved to IP address: %s\n\n" , input_hostname , ip);
44         //return 0;
45     }
}
```

```
jayprakash@jayprakash-System-AsusPG500: ~/Desktop/Sockets Lab
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ gcc 4_client_get_web_server_ip_address.c -o 4.o
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ ./4.o
Enter hostname for which you want to fetch the IP address:google.com
google.com Host name resolved to IP address: 172.217.166.238

Enter hostname for which you want to fetch the IP address:google.co.in
google.co.in Host name resolved to IP address: 216.58.196.99

Enter hostname for which you want to fetch the IP address:flipkart.com
flipkart.com Host name resolved to IP address: 163.53.78.128

Enter hostname for which you want to fetch the IP address:daiict.ac.in
daiict.ac.in Host name resolved to IP address: 104.238.110.159

Enter hostname for which you want to fetch the IP address:www.iitm.ac.in
www.iitm.ac.in Host name resolved to IP address: 14.139.160.172
```

```
jayprakash@jayprakash-System-AsusPG500:~
```

```
jayprakash@jayprakash-System-AsusPG500:~$ sudo netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0 127.0.1.1:53            0.0.0.0:*              LISTEN    2046/dnsmasq
tcp      0      0 127.0.0.1:631           0.0.0.0:*              LISTEN    1728/cupsd
tcp6     0      0 ::1:631                ::*:*                  LISTEN    1728/cupsd
udp      0      0 0.0.0.0:48578          0.0.0.0:*              LISTEN    716/avahi-daemon: r
udp      0      0 0.0.0.0:631           0.0.0.0:*              LISTEN    1029/cups-browsed
udp      0      0 127.0.1.1:53            0.0.0.0:*              LISTEN    2046/dnsmasq
udp      0      0 0.0.0.0:68             0.0.0.0:*              LISTEN    2042/dhclient
udp      0      0 0.0.0.0:39053          0.0.0.0:*              LISTEN    2042/dhclient
udp      0      0 224.0.0.251:5353         0.0.0.0:*              LISTEN    2342/chrome
udp      0      0 224.0.0.251:5353         0.0.0.0:*              LISTEN    2374/chrome --type=
udp      0      0 0.0.0.0:5353           0.0.0.0:*              LISTEN    716/avahi-daemon: r
udp6     0      0 ::::42668              ::*:*                  LISTEN    716/avahi-daemon: r
udp6     0      0 ::::17365              ::*:*                  LISTEN    2042/dhclient
udp6     0      0 ::::5353               ::*:*                  LISTEN    716/avahi-daemon: r
```

```
jayprakash@jayprakash-System-AsusPG500:~$ sudo ss -tunlp
```

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
tcp	UNCONN	0	0	*:48578	*:*
				users:(("avahi-daemon",716,15))	
tcp	UNCONN	0	0	*:631	*:*
				users:(("cups-browsed",1029,8))	
tcp	UNCONN	0	0	127.0.1.1:53	*:*
				users:(("dnsmasq",2046,4))	
tcp	UNCONN	0	0	*:68	*:*
				users:(("dhclient",2042,6))	
tcp	UNCONN	0	0	*:39053	*:*
				users:(("dhclient",2042,20))	
tcp	UNCONN	0	0	224.0.0.251:5353	*:*
				users:(("chrome",2342,169))	
tcp	UNCONN	0	0	224.0.0.251:5353	*:*
				users:(("chrome",2374,46))	
tcp	UNCONN	0	0	*:5353	*:*
				users:(("avahi-daemon",716,13))	
tcp	UNCONN	0	0	*:42668	*:*
				users:(("avahi-daemon",716,16))	
tcp	UNCONN	0	0	:::17365	*:*
				users:(("dhclient",2042,21))	
tcp	UNCONN	0	0	:::5353	*:*
				users:(("avahi-daemon",716,14))	
tcp	LISTEN	0	5	127.0.1.1:53	*:*
				users:(("dnsmasq",2046,5))	
tcp	LISTEN	0	128	127.0.0.1:631	*:*
				users:(("cupsd",1728,11))	
tcp	LISTEN	0	128	:::1:631	*:*
				users:(("cupsd",1728,10))	

```
jayprakash@jayprakash-System-AsusPG500:~$
```



TCP_Client.c (~/Desktop/Sockets Lab) - gedit

```
1 /* To compile this program use the following command */
2 /* $ gcc TCP_Client.c -o TCP_Client.o */
3
4 /* To run this program use the following command */
5 /* $ ./TCP_Client.o localhost 8080 */
6
7 // Client side C/C++ program to demonstrate Socket programming
8 #include <stdio.h>
9 #include <sys/socket.h>
10 #include <arpa/inet.h>
11 #include <unistd.h>
12 #include <string.h>
13 #define PORT 8080
14
15 int main(int argc, char const *argv)
16 {
17     int sock = 0, valread;
18     struct sockaddr_in serv_addr;
19     char *hello = "Hello from TCP Client";
20     char buffer[1024] = {0};
21     if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0)
22     {
23         printf("\n Socket creation error \n");
24         return -1;
25     }
26
27     serv_addr.sin_family = AF_INET;
28     serv_addr.sin_port = htons(PORT);
29
30     // Convert IPv4 and IPv6 addresses from text to binary form
31     if(inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr) <= 0)
32     {
33         printf("\nInvalid address/ Address not supported \n");
34         return -1;
35     }
36 }
```

jayprakash@jayprakash-System-AsusPG500:~\$ sudo netstat -tunlp

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:8080	0.0.0.0:*	LISTEN	4902/TCP_Server.o
tcp	0	0	127.0.0.1:53	0.0.0.0:*	LISTEN	2046/dnsmasq
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN	1728/cupsd
tcp6	0	0	::1:631	::*	LISTEN	1728/cupsd
udp	0	0	0.0.0.0:48578	0.0.0.0:*	LISTEN	716/avahi-daemon: r
udp	0	0	0.0.0.0:631	0.0.0.0:*	LISTEN	1029/cups-browsed
udp	0	0	127.0.0.1:53	0.0.0.0:*	LISTEN	2046/dnsmasq
udp	0	0	0.0.0.0:68	0.0.0.0:*	LISTEN	2042/dhclient

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$./TCP_Server.o &

[1] 4902

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$ t

Hello message sent to TCP client

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$./TCP_Client.o localhost 8080

Hello message sent to TCP Server

Hello from TCP server

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$

```
tcp_simple_client.c (~/Desktop/Sockets Lab) - gedit
tcp_server_v1.c x tcp_simple_client.c x

30
31     if (sockfd < 0)
32         perror("ERROR opening socket");
33
34     server = gethostbyname(argv[1]);
35
36     if (server == NULL) {
37         fprintf(stderr,"ERROR, no such host\n");
38         exit(0);
39     }
40
41     bzero((char *) &serv_addr, sizeof(serv_addr));
42     serv_addr.sin_family= AF_INET;
43     bcopy((char *)server->h_addr,
44           (char *)&serv_addr.sin_addr.s_addr,
45           server->h_length);
46     serv_addr.sin_port = htons(portno);
47
48     if (connect(sockfd,(struct sockaddr *)&serv_addr,sizeof(serv_addr)) < 0)
49         perror("ERROR connecting");
50
51     printf("Please enter the message: ");
52     bzero(buffer,256);
53     fgets(buffer,255,stdin);
54
55     n = write(sockfd,buffer,strlen(buffer));
56
57     if (n < 0)
58         perror("ERROR writing");
59
60     bzero(buffer,256);
61     n = read(sockfd,buffer,255);
62
63     if (n < 0)
64         perror("ERROR reading");
65
```

```
jayprakash@jayprakash-System-AsusPG500:~$ sudo netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp        0      0 0.0.0.0:2500           0.0.0.0:*          LISTEN     5205/tcp_server_v1.
tcp        0      0 127.0.1.1:53           0.0.0.0:*          LISTEN     2046/dnsmasq
tcp        0      0 127.0.0.1:631          0.0.0.0:*          LISTEN     1728/cupsd
tcp6       0      0 ::1:631              ::*:*               LISTEN     1728/cupsd
udp        0      0 0.0.0.0:48578          0.0.0.0:*          LISTEN     716/avahi-daemon: r
udp        0      0 0.0.0.0:631           0.0.0.0:*          LISTEN     1029/cups-browsed
udp        0      0 127.0.1.1:53           0.0.0.0:*          LISTEN     2046/dnsmasq
udp        0      0 0.0.0.0:68            0.0.0.0:*          LISTEN     2042/dhclient
```

```
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ ./tcp_server_v1.o 2500 &
[1] 5205
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$
```

Here is the message: I am a simple TCP client sending a test message.

```
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ ./tcp_simple_client.o localhost 2500
Please enter the message: I am a simple TCP client sending a test message.
I got your message
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$
```



tcp_server_v2.c (~/Desktop/Sockets Lab) - gedit

```
1 /* To compile this program use the following command */
2 /* $ gcc tcp_server_v2.c -o tcp_server_v2.o */
3
4 /* To run this program use the following command */
5 /* $ ./tcp_server_v2.o <port number> & */
6
7 /* For e.g., $ ./tcp_server_v2.o 2500 & */
8
9 /* A simple server in the internet domain using TCP
10   The port number is passed as an argument to the process
11   This version runs forever, forking a new
12   process for each connection
13 */
14
15 #include <stdio.h>
16 #include <stdlib.h>
17 #include <string.h>
18 #include <sys/types.h>
19 #include <sys/socket.h>
20 #include <netinet/in.h>
21
22 void dostuff(int); /* function prototype */
23
24 /*
25 void error(char *msg)
26 {
27     perror(msg);
28     exit(1);
29 }
30 */
31
32 int main(int argc, char *argv[])
33 {
34     int sockfd, newsockfd, portno,
35         struct sockaddr_in serv_addr, cli_addr;
36
37     if (argc < 2) {
```

jayprakash@jayprakash-System-AsusPG500:~\$ ps -a

PID	TTY	TIME	CMD
5582	pts/0	00:00:00	tcp_server_v2.o
5586	pts/0	00:00:00	tcp_server_v2.o <defunct>
5589	pts/0	00:00:00	tcp_server_v2.o <defunct>
5595	pts/22	00:00:00	ps

jayprakash@jayprakash-System-AsusPG500:~\$

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$./tcp_simple_client.o localhost 2500

Please enter the message: This is a test message from the second client

I got your message

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$./tcp_simple_client.o localhost 2500

Please enter the message: This is a test message from the first client

I got your message

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$./tcp_server_v2.o 2500 &

[1] 5582

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab\$

Here is the message: This is a test message from the first client

Here is the message: This is a test message from the second client

Terminal

En 5:57 PM JayPrakash

A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window and a dock with various application icons.

The terminal window displays the source code for a TCP client program named `tcp_simple_client.c`. The code initializes a socket, binds it to port 2500, and performs a connection to a server at localhost port 2500. It handles incoming connections and prints messages to the client.

```
1b {
17     int sockfd, portno, n;
18
19     struct sockaddr_in serv_addr;
20     struct hostent *server;
21
22     char buffer[256];
23     if (argc < 3) {
24         fprintf(stderr,"usage %s hostname port\n", argv[0]);
25         exit(0);
26     }
27
28     portno = atoi(argv[2]);
29     sockfd = socket(AF_INET, SOCK_STREAM, 0);
30
31     if (sockfd < 0)
32         perror("ERROR opening socket");
33
34     server = gethostbyname(argv[1]);
35
36     if (server == NULL) {
37         fprintf(stderr,"ERROR, no such host\n");
38         exit(0);
39     }
40
41     bzero((char *) &serv_addr, sizeof(serv_addr));
42     serv_addr.sin_family = AF_INET;
43     bcopy((char *)server->h_addr,
44           (char *)&serv_addr.sin_addr.s_addr,
45           server->h_length);
46     serv_addr.sin_port = htons(portno);
```

jayprakash@jayprakash-System-AsusPG500:~\$ ps -a
PID TTY TIME CMD
4975 pts/21 00:00:00 tcp_server_v3.o
5011 pts/23 00:00:00 tcp_simple_clie
5041 pts/25 00:00:00 tcp_simple_clie
5067 pts/26 00:00:00 ps
jayprakash@jayprakash-System-AsusPG500:~\$ ps -a
PID TTY TIME CMD
4975 pts/21 00:00:00 tcp_server_v3.o
5041 pts/25 00:00:00 tcp_simple_clie
5083 pts/26 00:00:00 ps
jayprakash@jayprakash-System-AsusPG500:~\$ ps -a
PID TTY TIME CMD
4975 pts/21 00:00:00 tcp_server_v3.o
5088 pts/26 00:00:00 ps
jayprakash@jayprakash-System-AsusPG500:~\$

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets\ Lab/
Please enter the message: This test messag is from Client 2
Hello Client , I have received your connection. And now I will assign a handler for you
Greetings! I am your connection handler
Its my duty to communicate with you
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets\ Lab\$

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets\ Lab

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets\ Lab\$ clear
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets\ Lab\$./tcp_server_v3.o 2500 &

[1] 4975

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets\ Lab\$ Connection accepted
Handler assigned
Connection accepted
Handler assigned

jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets\ Lab/
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets\ Lab\$./tcp_simple_client.o localhost 2500
Please enter the message: This test message is from Client 1
Hello Client , I have received your connection. And now I will assign a handler for you
Greetings! I am your connection handler
Its my duty to communicate with you
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets\ Lab\$

Terminal

En 6:04 PM JayPrakash



UDP_Server.c (~/Desktop/Sockets Lab) - gedit

```
8 #include <arpa/inet.h>
9 #include <netinet/in.h>
10
11 #define PORT      8080
12 #define MAXLINE 1024
13
14 // Driver code
15 int main() {
16     int sockfd;
17     char buffer[MAXLINE];
18     char *hello = "Hello from server";
19     struct sockaddr_in servaddr, cliaddr;
20
21     // Creating socket file descriptor - AF_INET for IPv4, AF_INET6 for IPv6 + SOCK_STREAM for TCP / SOCK_DGRAM for UDP
22     if ( (sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) {
23         perror("Socket creation failed");
24         exit(EXIT_FAILURE);
25     }
26
27     memset(&servaddr, 0, sizeof(servaddr));
28     memset(&cliaddr, 0, sizeof(cliaddr));
29
30     // Filling server information
31     servaddr.sin_family = AF_INET; // IPv4 - AF_INET for IPv4/ AF_INET6 for IPv6
32     servaddr.sin_addr.s_addr = INADDR_ANY;
33     servaddr.sin_port = htons(PORT);
34
35     // Bind the socket with the server address
36     if ( bind(sockfd, (const struct sockaddr *)&servaddr,
37               sizeof(servaddr)) < 0 )
38     {
39         perror("Bind failed");
40         exit(EXIT_FAILURE);
41     }
```

jayprakash@jayprakash-System-AsusPG500:~\$ ps -a

PID	TTY	TIME	CMD
5292	pts/21	00:00:00	UDP_Server.o
5297	pts/26	00:00:00	ps

jayprakash@jayprakash-System-AsusPG500:~\$ ps -a

PID	TTY	TIME	CMD
5329	pts/26	00:00:00	ps

jayprakash@jayprakash-System-AsusPG500:~\$

jayprakash@jayprakash-System-AsusPG500: ~/Desktop/Sockets Lab

```
[1]+  Terminated                  ./tcp_server_v3.o 2500
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ gcc UDP_Server.c -o UDP_Server.o
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ ./UDP_Server.o 9000 &
[1] 5292
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ Client : Hello from client
Hello message sent.
```

jayprakash@jayprakash-System-AsusPG500: ~/Desktop/Sockets Lab

```
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ gcc UDP_Client.c -o UDP_Client.o
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ ./UDP_Client.o localhost 9000
Hello message sent.
Server : Hello from server
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ 
```



```
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ gcc TCP_Server_without_threads.c -o TCP_Server_without_threads.o
jayprakash@jayprakash-System-AsusPG500:~/Desktop/Sockets Lab$ ./TCP_Server_without_threads.o
Listener on port 8888
Waiting for connections ...
New connection , socket fd is 4 , ip is : 127.0.0.1 , port : 55502
Welcome message sent successfully
Adding to list of sockets as 0
Host disconnected , ip 127.0.0.1 , port 55502
New connection , socket fd is 4 , ip is : 127.0.0.1 , port : 55504
Welcome message sent successfully
Adding to list of sockets as 0
New connection , socket fd is 5 , ip is : 127.0.0.1 ,
Welcome message sent successfully
Adding to list of sockets as 1
New connection , socket fd is 6 , ip is : 127.0.0.1 ,
Welcome message sent successfully
Adding to list of sockets as 2
New connection , socket fd is 7 , ip is : 127.0.0.1 ,
Welcome message sent successfully
Adding to list of sockets as 3
```

```
jayprakash@jayprakash-System-AsusPG500:~
```

```
jayprakash@jayprakash-System-AsusPG500:~ telnet localhost 8888
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
ECHO Daemon v1.0
```



```
jayprakash@jayprakash-System-AsusPG500:~$ ps -a
 PID TTY      TIME CMD
 4090 pts/1    00:01:11 gedit
 7946 pts/9    00:00:00 TCP_Server_with
 8009 pts/10   00:00:00 telnet
 8089 pts/25   00:00:00 ps
jayprakash@jayprakash-System-AsusPG500:~$
```

```
jayprakash@jayprakash-System-AsusPG500:~
```

```
jayprakash@jayprakash-System-AsusPG500:~ telnet localhost 8888
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
ECHO Daemon v1.0
```



```
jayprakash@jayprakash-System-AsusPG500:~
```

```
jayprakash@jayprakash-System-AsusPG500:~$ telnet localhost 8888
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
ECHO Daemon v1.0
```

Summary

- Sockets help application process to communicate with each other using standard Unix file descriptors
- Two types of Internet sockets: `SOCK_STREAM` and `SOCK_DGRAM`
- Many routines exist to help ease the process of communication