



ENROLMENT NO.: 186407279

NAME: Aman Tiwari

ADDRESS: G-284 gazipur village Delhi 110096

PROGRAMME TITLE: BCA 5

COURSE CODE: BCSL-056

COURSE TITLE: Network Programming and Administration Lab

ASSIGNMENT CODE: .....

STUDY CENTRE CODE: 0769 (Shyamla College sahadra )

PHONE NO.: 9891062743 / 8448179216

EMAIL ID: amantiwari8861@gmail.com



Date

SIGNATURE

Ans 1: (a) /\* \* \* \* \* CLIENT CODE \* \* \* \* \*/

```

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
int main() {
    int clientSocket, Port Num, nBytes;
    char Buffer [1024];
    struct sockaddr_in ServerAddr;
    socklen_t addrlen = sizeof(ServerAddr);
    ClientSocket = socket (PF_INET, SOCK_STREAM, 0);
    Port Num = 7891;
    Server Addr.sin_family = AF_INET;
    Server Addr.sin_port = htons (Port Num);
    Server Addr.sin_addr.s_addr = inet_addr ("127.0.0.1");
    memset (&Server Addr.sin_zero, '\0', sizeof(Server Addr.sin_zero));
    addrlen = sizeof(Server Addr);
    Connect (Client Socket, (struct sockaddr*)&Server Addr, addrlen);
    while (1) {
        printf ("Type a Sentence to Send to Server:\n");
        fgets (buffer, 1024, stdin);
    }
}

```

```

printf("You typed! %s", buffer);
nBytes = strlen(buffer) + 1;
Send(clientSocket, buffer, nBytes, 0);
recv(clientSocket, buffer, 1024, 0);
printf("Received from Server: %s\n\n", buffer);
}
return 0;
}

```

Output:

Type a Sentence to Send to Server:  
hello world

You typed: hello world

Received from Server: helloworld

Type a Sentence to Send to Server:

```

/* * * * * UDP SERVER CODE * * * * */
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
#include <stdlib.h>
int main()
{
    int udpSocket, nBytes;
    char buffer[1024];

```

Struct sockaddr\_in Server Addr., Client Addr.  
 Struct sockaddr - storage server storage;  
 int i;  
 /\* Create UDP Socket \*/  
 Udp Socket = Socket (PF\_INET, SOCK\_DGRAM, 0);  
 /\* Configure setting in address struct \*/  
 Server Addr.sin\_family = AF\_INET;  
 Server Addr.sin\_port = htons (7891);  
 Server Addr.sin\_addr.s\_addr = inet\_addr  
 ("127.0.0.1");  
 memset (Server Addr.sin\_zero, '0', sizeof  
 Server Addr.sin\_zero);  
 /\* Bind socket with address struct \*/  
 bind (Udp Socket, (Struct sockaddr \*) & Server  
 Addr, sizeof (Server Addr));  
 /\* Initialize size variable to be used later on \*/  
 adder\_size = sizeof Server Storage;  
 while (1) {  
 /\* Try to receive any incoming UDP datagram.  
 Address and Port of  
 requesting client will be stored on Server  
 Storage Variable \*/  
 nBytes = recvfrom (Udp Socket, buffer, 1024, 0,  
 (Struct sockaddr \*) & Server  
 Storage, & adder\_size);
 }

```

/* Convert message received to uppercase */
for (i = 0; i < nBytes - 1; i++)
    buffer[i] = toupper(buffer[i]);
/* Send uppercase message back to client, using
   Server storage as the address */
Sendto(cudsSocket, buffer, nBytes, 0, (struct
    sockaddr *) &ServerStorage, addy_size);
}

return 0;
}

```

(b) Socket is created ; Most interprocess communication uses the client server model. These terms refer to the two processes which will be communicating with each other. One of the two processes which will be communication with each other . one of the two processes the Client , connects to the other process , the Server typically to make a request for information . A good analogy ~~who~~ is a person who makes call to another person.

The steps involved in establishing a socket on the client side are as following

1. Create a Socket with the `socket()` System Call
2. Connect the Socket to the address of the server using the `connect()` System Call.
3. Send and receive data. There are a number of ways to do this, but the simplest is to use the `read()` and `write()` System Calls.

The steps involved in establishing a socket on the Server Side are as follows:

1. Create a Socket with the `socket()` System Call
2. Bind the Socket to an address using the `bind()` System Call. For a Server socket on the Internet, an address consists of a Port number on the host machine.
3. Listen for Connections with the `listen()` System Call
4. Accept a Connection with the `accept()` System Call. This Call typically blocks until a Client connects with the server.
5. Send and receive data.

Arguments used for sending and receiving message:-

Sending Messages

To send a message, use the `tsend(sc)` function with the following signature:

int  
 tpsend (int cd, char \* data, long len, long flags,  
 long \*revent)

The following table describes the arguments to the tpsend () function.

### tpsend () Function Argument

Argument	Description
cd	Specifies the Connection description return by the tpcconnect () function identifying the Connection over which the data is sent.
data	Pointer to a data buffer. When establishing the Connection, you can send data simultaneously by setting the data argument to point to a buffer previously allocated by tpalloc().
len	length of the data buffer .if the buffer is self - defining (for example, an FIDL buffer), you can set len to 0. If you do not specify a value for data , this argument is ignored.
revent	Pointer to event value set when an error is encountered (that is, when tpeerrno(s) is set to TPEEVENT).

**Flag**

Specifies the flag settings. For a list of valid flag settings, refer to `tpsend(3c)` in the BEA Tuxedo ATMIC function reference.

**Receiving Message****int**

`tprec(int cd, char** data, long* len, long flags,  
long* revent)`

The following table describes the argument to the `tprecv()` function.

Argument	Description
cd	Specifies the connection descriptor. If a subordinate program issue the call, the cd argument should be set to the value specified in the TPSVCINFO structure for the program.
data	Pointer to a data buffer. The data argument must point to a buffer previously allocated by <code>tpalloc()</code> .
len	Length of the data buffer. If the buffer is self-defining (for example, an FML buffer), you can set len to 0.
revent	Pointer to event value set when an error is encountered (that is, when tprec is set to TPEEVENT).

flag / specifies the flag settings. Refer to `tprecv(3c)` in the BEA Tuxedo ATM C function reference for a list of valid flags.

Ans 2 = (a) Cat:- Cat (concatenate) Command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, Concatenate files. So let us see some frequently used cat Commands.

Command:-  
\$ Cat filename

Output

It will show Content of Given filename.

Sort:- SORT Command is used to Sort a file, arranging the records in a Particular order. By default, the Sort Command sorts file assuming the contents are ASCII. using options. In Sort Command, it can also be used to Sort numerically.

Command:-  
\$ sort file.txt

Output :-  
 abishhek  
 Chitransh

**Ping !** - PING (Packet internet Groper) Command is used to check the network connectivity between host and Server / host. This Command takes as input the IP address or the URL and sends a data packet to the specified address with the message "PING" and get a response from the Server / host this time a recorded which is called latency. fast ping low latency means faster connection.

Command :-

Sudo Ping -v

**kill** :- kill Command in Linux (located in /bin/kill), is a built-in command which is used to terminate processes manually. kill Command Sends a Signal to a Process which terminate the process.

Command :

\$ kill -1

Ps-ag:- Linux provides us a utility called Ps for viewing information related with the processes on a system which stands as abbreviation for "Process Status". Ps command is used to list the currently running running processes and their PIDs along with some other information depends on different option.

Command  
Ps [option]

Output

```
[root@rhel7 ~]# Ps
PID TTY TIME CMD
12330 pts/0 00:00:00 bash
21621 pts/0 00:00:00 Ps
```

(b) To check if a particular process is running or not

Command:

Ps [option]

Output

```
[root@rhel7 ~]# Ps
PID TTY TIME CMD
12330 pts/0 00:00:00 bash
```

21621 pts/0 00:00:00 ls

To see currently running processes and other information like memory and CPU usages

vivek@nixcraft:~\$ ps -aux

vivek@nixcraft:~\$ sudo ps -a

List all file names that have a letter h as 4th character in the name.

Is l grep ".\*\*xh.\*"

# list all files names that have 4th character as h

list all the file names that have the number 2 or 5 in the name

find .l(-name "??.cpp"- o- name "12345.cpp")

Ans 3 = Setup a network Connection in Linux:  
 Configuring a network connection from a Linux machine can be challenging.  
 Fortunately, many new Linux distributions.

Come with some type of network management tool that can help you automatically connect to a wireless network.

### Step 1:- Check network Connectivity

The ping command is a well-known utility that quickly checks for a connection to an address. Enter the following on the command line:-

`ping -c3 opensource.com`

Where the `-c3` parameter option indicates you will call this domain name only three times.

### Step 2: Check connection information

You can check network information using the `ip add` command.

Running this command shows device information like the IP address, and more. You'll need some of this information like the device inter and IP address, later to set up a static connection, so go ahead and grab it.

Step 3: Check network information.

Network information can be found inside the etc / sysconfig / network - scripts directory by entering ls / etc / sysconfig / network - scripts.

Step - 4: Show available connections.

The nmcli tool shows the available connections currently used to connect to the network. Enter the following command nmcli con show.

Step - 5 Checks that the network connection is on you used the ping command above to check that you can receive data packets, but now use the systemctl command for network to monitor, update and troubleshoot the network.

Step - 6 =

Now You're ~~not~~ ready to add a static connection. Using the device name you generated from if add in Step 2, modify and enter the following command to add a new connection.

Step 7: Verify the Connection is added to the network - scripts Path.

There are two ways to modify the new Connection info using the nmcli tool. One is by using the following command:

`nmcli con mod.`

Step-8 Confirm You can see the Connection  
Check that My favorite Cafe is visible as  
the available Connection. Use the following  
Command to bring the Connection up.

Please note that some \_Connection\_Name  
is the name of Your Connection.

Step-9 Modify the Connection to be Static

Open the file /etc/syconfig/network-scripts/  
`ifcfg-some-connection-name` (it is my  
favoriteCafe in this example), using a text  
editor , such as Vim , Emacs - or Nano :

1. Modify BOOTPROTO to be static.
2. Add IPADDR. This can be found from  
the ip add Command or your Connected  
network

3. Add NETMASK. This can be found from the ip add Command or your Connected network.
4. Add GATEWAY. This can be found the ip add Command network. You might also need to add DNS, PREFIX, or other information depending on how your network and the machine are set up.

Step 10: Confirm the new connection is active. That should do it! But make sure by checking if the new connection is working. Run the `nmcli con & show` command again to start the new connection.

`nmcli dev show DEVICE-NAME`

Where `DEVICE-NAME` is the name of your network device

Configure it for wireless networking - what you will need, in order to be able to establish this connection, are the following:

- `ifconfig`: Enable your wireless device.
- `iwlist`: List the available wireless access points.
- `iwconfig`: Configure your wireless connection.
- `dhclient`: Get your IP Address via DHCP.
- `wpa-supplicant`: for use with WPA authentication.

Make Sure You have all of the above tools on A-die your Computer before you continue To test for this tools You can, from within Your terminal window, issue the Commands.

- Which ifconfig
- Which iwlist
- Which iwconfig
- which dhclient
- Which wpa - supplicant

Where wlan 0 is the name of your wireless device. The abont Command will bring your wireless device up so it is ready to use. The next phase is to Scan for any wireless access Points with the command:

where NETWORK-NAME is the name is the name of the network you want to connect to the WIRELESS - KEY is the security key for that network. NOTE: The iwconfig command default to HEX Value for wireless keys.

If You need ascii you have to append the "S" Prefix to your key like so:

iwconfig wlan0 essid NETWORK-NAME key:  
WIRELESS - KEY

Ans Q2 The DHCP protocol lets a DHCP client, that is your network host to lease network configuration parameters such as an IP address. In fact, lease parameters are not limited to IP address only as they may also include the following configuration settings.

- IP addresses and network masks
- Domain Name Servers (DNS)
- Default Gateways
- WINS servers
- Syslog hosts
- proxy servers
- NTP servers
- X font servers
- Syslog hosts.

### DHCP Server Installation

Standard DHCP Server implementation available in various Linux distributions is an open source version maintained by ISC.

Use the following Linux commands to install DHCP on your Linux server:

Debian and Ubuntu:

```
# apt-get install isc-dhcp-server
```

Redhat & fedora:

# Yum install dhcP

### Basic DHCP Configuration

By Default, DHCP server configuration does not pre-lease any subnet on which DHCP Server should lease IP addresses.

DHCP default and max lease time

At this point we can add few additional setting to our DHCP configuration, namely the default and max lease time expiry.

default - lease - time is a value in seconds.

Subnet 10.11.0 network 255.255.255.0 {

range 10.11.310.1.1.254;

}

Subnet 192.168.0.0 network 255.255.0.0 {

}

### Define DNS Server

Another Configuration Parameter Possible to be

Set by DHCP Server to its Client is a definition of DNS server.

default - lease - time 600;

max - lease - time 7200;

Subnet 10.11.0 network 255.255.255.0 {

range 10.11.310.1.1.254;

option domain-name-servers 10.1.1.1, 8.8.8.8;

Subnet 10.1.1.0 netmask 255.255.0.0 {

Subnet 10.1.1.0 netmask 255.255.255.0 {

range 10.1.1.3 10.1.1.254;

option routers 10.1.1.1;

Set default gateway

DHCP also allows for client's gateway configuration. To set any client on the local network network to use default gateway 10.1.1.1, and change "option routers 10.1.1.1" into dhcph.conf file as demonstrated below.

default-lease-time 600;

max-lease-time 7200;

Subnet 10.1.1.0 netmask 255.255.255.0 {

range 10.1.1.3 10.1.1.254;

option domain-name-servers 10.1.1.1, 8.8.8.8;

option routers 10.1.1.1;

{

Subnet 192.168.0.0 netmask 255.255.0.0 {

~~200~~ ?

Subnet 10.1.1.0 netmask 255.255.255.0 {

range 10.1.1.3 10.1.1.254;

option routers 10.1.1.1;

{

DHCP will now set DHCP client with gateway 10.1.1.1.

### Host Specific Configuration

There maybe a need to set static IP address to a particular host on the network such as printer, web server and etc.

default-lease-time 600;

max-lease-time 7200;

Subnet 10.1.1.0 network 255.255.255.0 {

range 10.1.1.3 10.1.1.254;

option domain-name-server 10.1.1.1, 8.8.8.8;

option routers 10.1.1.1;

}

Subnet 192.168.0.0 network 255.255.0.0 {

}

host printer {

hardware ethernet 00:16:d3:b7:8f:86;

fixed-address 10.1.1.100;

}

host web - Server {

hardware ethernet 00:17:a4:c2:44:12;

fixed-address 10.1.1.200;

}

The above DHCP Configuration file will permanently assign the IP address 10.1.1.100 to a host "Printer" with a mac address 00:16:d3:b7:8f:86 and

IP address 10.11.200 to host "Web-server" with MAC address 00:17:1A:4C:44:22.

### Configuring DHCP relay agent

If your DHCP server does not have an access to some particular network it does not mean that it cannot provide its services there.  
# apt-get install isc-dhcp-relay.

Then Create a Configuration file /etc/default/isc-dhcp-relay with two lines:-

Servers = "192.168.5.5"

Interfaces = "eth0"

### BOOTP Support

ISC DHCP Server is back work compatible with BOOTP. The following is a BOOTP client declaration to be defined in DHCP's main Configuration dhcpcd.conf. file:

host boot {

hardware ethernet 00:00:2e:55:12:09;

fixed-address 128.123.1.3;

file name "/Path/to/tftpboot/bootp.Boot";

Sort each column of the table and show the result: SORT Command is used to sort a file, arranging the records in a particular order.

- SORT Command sorts the contents of a text file, line by line.
- Sort is a standard command line program that prints the lines of its input or concatenation of all files listed in its argument list in sorted order.
- The Sort Command is a command line utility for sorting lines of text file. It supports sorting alphabetically, in reverse order, by number, by Month and can also remove duplicates.
- The Sort Command can also sort by item not at the beginning of the line, ignore case sensitivity and return whether a file is sorted or not. Sorting is done based on one or more sort keys extracted from each line of input.
- By default the entire input is taken as sort key. Blank space is the default field separator.

### Examples

Suppose you create a data file with name file.txt  
 Command:

\$ Cat > file .txt

abhishek

Chitransh

Satish

rajan

neveen

divyam

haresh

Sorting a file : Now use the Sort command  
 Syntax:

\$ Sort filename.txt

Command:

\$ Sort file .txt

Output:

abhishek

Chitransh

divyam

haresh

neveen

rajan

Satish