Assignment 1

Name: Abhishek Srivastava

Registration Number: 19BCE10071

## 2) Introduction to Socket Programming, Basic Linux Commands

Sockets are the "virtual" endpoints of any kind of network communications done between 2 hosts over in a network. For example when you type www.google.com in your web browser, it opens a socket and connects to google.com to fetch the page and show it to you. Same with any chat client like gtalk or skype. Any network communication goes through a socket.

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, the *client*, connects to the other process, the *server*, typically to make a request for information. A good analogy is a person who makes a phone call to another person.

Notice that the client needs to know of the existence of and the address of the server, but the server does not need to know the address of (or even the existence of) the client prior to the connection being established.

Notice also that once a connection is established, both sides can send and receive information.

The system calls for establishing a connection are somewhat different for the client and the server, but both involve the basic construct of a *socket*.

A socket is one end of an interprocess communication channel. The two processes

each establish their own socket.

**The steps involved in establishing a socket on the *client* side are as follows:**

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

**Some Basic Linux Commands:**

1. pwd — When you first open the terminal, you are in the home directory of your user.
2. ls — Use the "ls" command to know what files are in the directory you are in.
3. cd — Use the "cd" command to go to a directory.
4. mkdir & rmdir — Use the mkdir command when you need to create a folder or a directory. But rmdir can only be used to delete an empty directory. To delete a directory containing files, use rm.
5. rm - Use the rm command to delete files and directories.
6. touch — The touch command is used to create a file.
7. man — To know more about a command and how to use it, use the man command.
8. cp — Use the cp command to copy files through the command line.
9. mv — Use the mv command to move files through the command line.
10. locate — The locate command is used to locate a file in a Linux system, just like the search command in Windows.

## 3) To study various types of Connectors.

Registered Jack 45 (RJ45)

The cable connector that is found on almost all UTP and STP cables is a Registered Jack 45 which is mostly commonly referred to as RJ45. This type of connector resembles the older RJ11 connectors that most people are familiar with from wired telephones. Figure 5 below shows an example of a RJ45 connector:

Straight Tip (ST)

The Straight Tip (ST) connector is often seen on the end of a multi-mode cable; it has been commonly seen along with the SC connector for the last 20 years but is being slowly replaced by multi-fiber connectors (LC and MTP). Figure 6 below shows an example of a ST connector:

Subscriber Connector (SC)

The Subscriber Connector (SC) can be seen commonly on MMF or SMF; as with SC connectors, the ST connector is slowly being replaced by multi-fiber connectors. Figure 7 below shows an example of an SC connector:

Lucent Connector (LC)

The Lucent Connector (LC) was developed for high-density deployments where multiple fibers would be terminated within a confined space. Unlike the SC and ST connectors, the LC connector is always duplex connecting a pair of fibers at a time. Figure 8 below shows an example of a LC connector:

Multi-fiber Push On (MPO)

The Multi-fiber Push On (MPO) connector is another duplex connector that offers an easy options for connection. As the name suggests, it was designed to be able to be connected multiple times without the creation of any potential connector issues. It is often also referred to as Multi-fiber Termination Push-on (MTP); the MTP connector is a brand name (US Conec).