**Title:** Linux networking commands Emulator

**Aim:** Study and Perform Linux networking commands emulation using python or C++.

**Objective:**

1. To study various Linux networking commands.
2. To study and understand Linux Emulator.

**Theory:**

A network consists of several computers connected together. The network can be as simple as a few computers connected in your home or office, or as complicated as a large university network or even the entire Internet. When your computer is part of a network, you have access to those systems either directly or through services like mail and the web. There are a variety of networking programs that you can use. Some are handy for performing diagnostics to see if everything is working properly

**Emulator:**

In computing, an emulator is hardware or software that enables one computer system (called the host) to behave like a computer system (called the guest). An emulator typically enables the host system to run software or use peripheral devices designed for the guest system.

**Emulation:**

Emulation is a strategy in digital preservation to combat obsolescence. Emulation focuses on recreating an original computer environment, which can be time-consuming and difficult to achieve, but valuable because of its ability to maintain a closer connection to the authenticity of the digital object.

Emulation addresses the original hardware and software environment of the digital object, and recreates it on a current machine. The emulator allows the user to have access to any kind of application or operating system on a current platform, while the software runs as it did in its original environment. Jeffery Rothenberg, an early proponent of emulation as a digital preservation strategy states, "the ideal approach would provide a single extensible, long-term solution that can be designed once and for all and applied uniformly, automatically, and in synchrony (for example, at every refresh cycle) to all types of documents and media". He further states that this should not only apply to out of date systems, but also be upwardly mobile to future unknown systems. Practically speaking, when a certain application is released in a new version, rather than address compatibility issues and migration for every digital object created in the previous version of that application, one could create an emulator for the application, allowing access to all of said digital objects.

Most emulators just emulate hardware architecture i.e. if operating system firmware or software is required for the desired software, it must be provided as well (and may itself be emulated). Both the OS and the software will then be interpreted by the emulator, rather than being run by native hardware. Apart from this interpreter for the emulated binary machine's language, some other hardware (such as input or output devices) must be provided in virtual form as well; for example, if writing to a specific memory location should influence what is displayed on the screen, then this would need to be emulated.

**Basic Networking Commands:**

**1] hostname:**

Tells the user the host name of the computer they are logged into. Hostname command is used for finding host/domain name and IP address.

***hostname :*** with no options displays the machines host name

***hostname–d*** : displays the domain name the machine belongs to

***hostname –f*** : displays the fully qualified host and domain name

***hostname –I*** : displays the IP address for the current machine

**example:**

# hostname mitpune.com

**2] ping:**

The *ping* command (named after the sound of an active sonar system) sends echo requests to the host you specify on the command line, and lists the responses received their round trip time.

You simply use ping as:

***ping*** ip\_or\_host\_name

**3] netstat:**

Displays contents of /proc/net files. It works with the Linux Network Subsystem, it will tell you what the status of ports are ie. Open, closed, waiting, masquerade connections. It will also display various other things. It has many different options.

Syntax:

***netstat*** [address\_family\_options] [--tcp|-t] [--udp|-u] [--raw|-w] [--listening|-l] [--all|-a] [--numeric|-n] [--numeric-hosts][--numeric-ports][--numeric-ports] [--symbolic|-N] [--extend|-e[--extend|-e]] [--timers|-o] [--program|-p] [--verbose|-v] [--continuous|-c] [delay]

**4] ifconfig:**

View network configuration, it displays the current network adapter configuration. It is handy to determine if you are getting transmit (TX) or receive (RX) errors.

**5] dig:**

*dig* (domain information groper) is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried. Most DNS administrators use dig to troubleshoot DNS problems because of its flexibility, ease of use and clarity of output. Other lookup tools tend to have less functionality than dig.

Syntax:

***dig*** [@server] [-b address] [-c class] [-f filename] [-k filename] [-m] [-p port#] [-q name] [-t type] [-x addr] [-y [hmac:]name:key] [-4] [-6] [name] [type] [class] [queryopt...]

dig [-h] dig [global-queryopt...] [query...]

**6] traceroute:**

*traceroute* will show the route of a packet. It attempts to list the series of hosts through which your packets travel on their way to a given destination. Also have a look at *xtraceroute* (one of severalgraphical equivalents of this program).

Syntax:

***traceroute*** machine\_name\_or\_ip

**7] host:**

Performs a simple lookup of an internet address (using the Domain Name System, DNS).

Syntax:

***host*** ip\_address

or

***host*** domain\_name

**8] Tcpdumb:**

This is a sniffer, a program that captures packets off a network interface and interprets them for you. It understands all basic internet protocols, and can be used to save entire packets for later inspection.

Syntax:

using -c option, you can capture specified number of packets. The below example will only capture 6 packets.

# ***tcpdump*** -c 5 -i eth0

The below tcpdump command with option -A displays the package in ASCII format. It is a character-encoding scheme format.

***# tcpdump*** -A -i eth0

To list number of available interfaces on the system, run the following command with -D option.

***# tcpdump*** -D

The following command with option -XX capture the data of each packet, including its link level header in HEX and ASCII format.

# ***tcpdump*** -XX -i eth0

**9] Tracepath:**

Tracepath performs a very similar function to traceroute the main difference is that tracepath doesn't take complicated options. It traces path to destination discovering MTU along this path. It uses UDP port port or some random port. It is similar to traceroute, only does not require superuser privileges and has no fancy options.

Syntax:

***tracepath [-nc]*** <destination>[/<port>]

**10] Iwconfig:**

iwconfig command in Linux is use to configure a wireless network interface. You can see and set the basic Wi-Fi details like SSID channel and encryption. You can refer man page of iwconfig to know more.

Syntax:

# iwconfig [interface]

**11] SSH:**

SSH is a secure protocol that encrypts all data sent between the client computer and the computer it is connecting to. SSH applications usually allow both interactive terminal sessions on the remote machine and the ability to transfer files securely. ssh (SSH client) is a program for logging into a remote machine and for executing commands on a remote machine. It is intended to replace rlogin and rsh, and provide secure encrypted communications between two untrusted hosts over an insecure network. X11 connections and arbitrary TCP/IP ports can also be forwarded over the secure channel.

ssh connects and logs into the specified hostname The user must prove his/her identity to the remote machine using one of several methods depending on the protocol version used.

The command line SSH command to open a terminal on a remote machine is just called ssh.

Syntax:

***ssh*** username@hostname

where username is the name of your account on the remote system and hostname is the hostname of the remote system that is being connected to.

**12] SCP:**

To copy one or more local files up to a remote server.

Syntax:

***scp*** local\_file(s) user@hostname:destination\_directory

**Conclusion:**  we studied the basic linux networking commands emulation using python/C++

**FAQs:**

1. What is the use of traceroute command?
2. What is the significance of TTL field in traceroute Command.