Day1

A) Familiarise the basics of network configuration files and networking commands in Linux.

i) Write functions of following network configuration files.

a) **/etc/init.d/network** := A high-level script that operates on all network interfaces, not just one. It runs ifup or ifdown for each interface as needed, and also handles other details: adding routes, creating a lock file to indicate that networking is enabled, and much more. It even toggles the loopback interface, which might be more than you intended, if you just want to block outside traffic.

b) **/etc/sysconfig/netwok** := network file specifies additional information that is valid to all network interfaces on the system. The following entries from /etc/sysconfig/network define that IPv4 networking is enabled, IPv6 networking is not enabled, the host name of the system, and the IP address of the default network gateway

c) **/etc/sysconfig/network-scripts** := Using Red Hat Linux, all network communications occur between configured interfaces and physical networking devices connected to the system. The different types of interfaces that exist are as varied as the physical devices they support.
The configuration files for network interfaces and the scripts to activate and deactivate them are located in the /etc/sysconfig/network-scripts/ directory. While the existence of interface files can differ from system to system, the three different types of files that exist in this directory, interface configuration files, interface control scripts, and network function files, work together to enable Red Hat Linux to use various network devices.

d) **/etc/sysconfig/network-scripts/ifcfg-eth0** := One of the most common interface files which controls the first Ethernet network interface card or NIC in the system. In a system with multiple NICs, there are multiple ifcfg-ethX files (where X is a unique number corresponding to a specific interface). Because each device has its own configuration file, an administrator can control how each interface functions individually.

e) **/etc/nsswitch.con**f := Used to configure which services are to be used to determine information such as hostnames, password files, and group files. The last two ones, password files, and group files in our case are not used, since we don't use NIS services on our server. Thus, we will focus on the hosts line in this file.

f) **/etc/hosts** := As your machine gets started, it will need to know the mapping of some hostnames to IP addresses before DNS can be referenced. This mapping is kept in the /etc/hosts file. In the absence of a name server, any network program on your system consults this file to determine the IP address that corresponds to a host name.

ii) Identify the suitable command from the above list for the following purposes:

Imagine you as a system administrator of a company A123.

a) You have multiple network interface cards in your system. You need to provide additional configuration for only the first NIC.

Ans: Suitable file: /etc/sysconfig/network-scripts/ifcfg-eth0

This file is specific to the configuration of the first network interface card (NIC) named eth0. Each NIC has a corresponding configuration file in /etc/sysconfig/network-scripts/, and ifcfg-eth0 is typically used for the first NIC.

b) You need to control searches for users, IP addresses and group information.

Ans: Suitable file: /etc/nsswitch.conf

This file defines the order and sources for resolving various system information such as users, groups, hosts, and services. For example, it can specify whether to look in /etc/hosts, DNS, LDAP, or NIS for hostnames and IP addresses.

c) You need to specify routing and host information for the network interfaces

Ans: Suitable file: /etc/hosts

This file is used for static hostname-to-IP address mapping. It allows you to specify hostnames and their corresponding IP addresses locally. This can act as a fallback or alternative to DNS for hostname resolution.

B) Familiarize the Linux Networking Commands:

i) Write the syntax & explain the need of following networking commands

1. Ping

Syntax: ping [options] <destination>

Explanation: Sends ICMP echo requests to test connectivity to a specific host or IP address. Useful for checking if a host is reachable and measuring latency.

Example: ping google.com

2. Ifconfig

Syntax: ifconfig [interface] [options]

Explanation: Used to configure or display network interface settings such as IP addresses, netmasks, and MAC addresses.
Replaced by ip command in newer systems.

Example: ifconfig eth0

3. Traceroute

Syntax: traceroute [options] <destination>

Explanation: Displays the route packets take to reach a destination, showing each hop and its latency.
Useful for diagnosing network routing issues.

Example: traceroute google.com

4. Netstat

Syntax: netstat [options]

Explanation: Displays network statistics, including active connections, routing tables, and interface statistics.
Deprecated and replaced by ss in newer systems.

Example: netstat -tuln

5. Nslookup

Syntax: nslookup [domain_name]

Explanation: Queries DNS servers to resolve domain names to IP addresses and vice versa.
Useful for troubleshooting DNS-related issues.

Example: nslookup example.com

6. Route

Syntax: route [options]

Explanation: Displays and manipulates the system’s IP routing table.
Replaced by ip route in modern systems.

Example: route -n

7. Host

Syntax: host [options] <domain_name>

Explanation: Performs DNS lookups to resolve domain names into IP addresses.
Similar to nslookup, but simpler in usage.

Example: host example.com

8. Iwconfig

Syntax: iwconfig [interface] [options]

Explanation: Configures and displays wireless network interface parameters, such as SSID and frequency.
Works specifically for wireless interfaces.

Example: iwconfig wlan0

9. Hostname

Syntax: hostname [options] [new_hostname]

Explanation: Displays or sets the system's hostname.
Useful for identifying the machine on a network.

Example: hostname
hostname newhostname

10. Nload

Syntax: nload [options] [interface]

Explanation: A real-time console-based tool to monitor network traffic bandwidth usage.
Displays incoming and outgoing traffic separately.

Example: nload eth0

C) Identify the suitable command from the above list for performing the following

i) Test whether “Google” is up and accessible

- **Command:**

  ```
  bash
  Copy code
  ping google.com
  ```

- **Explanation:**
  Sends ICMP echo requests to check if Google is reachable. If responses are received, Google is up and accessible.

---

ii) Send 5 messages of buffer size 1000 bytes to any URL

- **Command:**

  ```
  bash
  Copy code
  ping -c 5 -s 1000 <URL>
  ```

- **Example:**

  ```
  bash
  Copy code
  ping -c 5 -s 1000 google.com
  ```

- **Explanation:**
  - `-c 5`: Sends exactly 5 messages.
  - `-s 1000`: Sets the size of each message to 1000 bytes.

---

iii) List all UDP sockets

- **Command:**

  ```
  bash
  Copy code
  netstat -u
  ```

- **Explanation:**
  - `-u`: Displays UDP sockets.
  - Alternatively, on modern systems, use:

```
bash
Copy code
ss -u
```

---

iv) Print the IP address details of google.com

- **Command:**

  ```
  bash
  Copy code
  nslookup google.com
  ```

- **Explanation:**
  Resolves the domain `google.com` into its associated IP addresses. Alternatively, you can use:

  ```
  bash
  Copy code
  host google.com
  ```

---

v) Display the hostname of your computer

- **Command:**

  ```
  bash
  Copy code
  hostname
  ```

- **Explanation:**
  Prints the system's current hostname. If needed, this can also be used to set a new hostname.

Day2

Fork :- is a system call in the Operating System used to create a copy of the process, the copied process created is known as the child process. The main process or the process from which the child process is created is known as the parent process. The parent and child processes are in separate address spaces in fork(). A Child process will have its Process ID different from the parent's ID.

- Total no. of child process generated will be: 2^n – 1.
- Total no of processes executed: 2^n.
-

Here n is the number of times forks called.

Remember that child processes return 0, while parent processes return a positive value (probably +1) at the fork call.

Fork() returns 3 values 0, 1, -1. 0 means it is child process 1 or greater than 1 means it is parent process and -1 means some error occured.