

Name : Sanchit Nilkanth Rajmane
MIS : 112103113
Branch : Computer DIV 2
Batch : T2

Assignment 2

linux commands

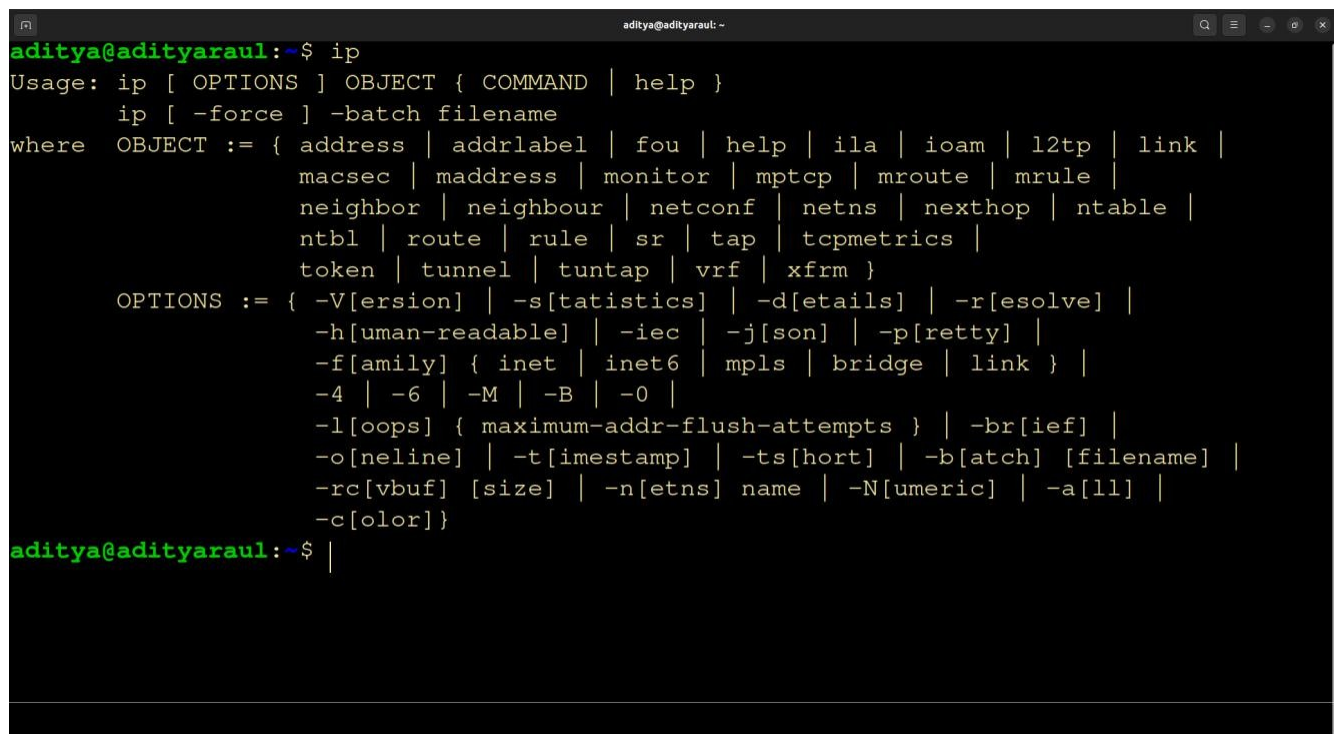
1. ifconfig

```
aditya@adityaraul: ~  
aditya@adityaraul:~$ ifconfig  
enp8s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.200.56.59 netmask 255.255.254.0 broadcast 10.200.57.255  
    inet6 fe80::d95c:15ca:551b:6812 prefixlen 64 scopeid 0x20<link>  
    ether 8c:8c:aa:a5:d7:4e txqueuelen 1000 (Ethernet)  
    RX packets 90830 bytes 128505656 (128.5 MB)  
    RX errors 0 dropped 26 overruns 0 frame 0  
    TX packets 44919 bytes 3112749 (3.1 MB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 1213 bytes 109200 (109.2 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 1213 bytes 109200 (109.2 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
wlp7s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.48.248 netmask 255.255.255.0 broadcast 192.168.48.255  
    inet6 2401:4900:5038:8f57:b9be:1bae:6608:50f9 prefixlen 64 scopeid 0x0<global>  
    inet6 2401:4900:5038:8f57:1578:a89d:adb9:b896 prefixlen 64 scopeid 0x0<global>  
    inet6 fe80::be51:d8ab:6817:10c prefixlen 64 scopeid 0x20<link>  
    ether 8c:c8:4b:a7:87:2b txqueuelen 1000 (Ethernet)  
    RX packets 17978 bytes 20199436 (20.1 MB)  
    RX errors 0 dropped 2 overruns 0 frame 0  
    TX packets 8711 bytes 1797574 (1.7 MB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
aditya@adityaraul:~$
```

The command **ifconfig** stands for **interface configurator**. This command enables us to initialize an interface, assign IP address, enable or disable an interface. It display route and network interface. We can view **IP address**, **MAC address** and **MTU** (Maximum

Transmission Unit) with ifconfig command. The names of the active network interfaces. This system includes *w/o1* and *lo* (the loopback interface).

2. ip

A terminal window with a dark background and light green text. The prompt is 'aditya@adityaraul:~\$'. The user has entered 'ip', and the terminal displays the usage and options for the 'ip' command. The output is as follows:

```
aditya@adityaraul:~$ ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
       ip [ -force ] -batch filename
where  OBJECT := { address | addrlabel | fou | help | ila | ioam | l2tp | link |
                  macsec | maddress | monitor | mptcp | mroute | mrule |
                  neighbor | neighbour | netconf | netns | nexthop | ntable |
                  ntbl | route | rule | sr | tap | tcpmetrics |
                  token | tunnel | tuntap | vrf | xfrm }
      OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
                  -h[uman-readable] | -iec | -j[son] | -p[retty] |
                  -f[amily] { inet | inet6 | mpls | bridge | link } |
                  -4 | -6 | -M | -B | -O |
                  -l[oops] { maximum-addr-flush-attempts } | -br[ief] |
                  -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
                  -rc[vbuf] [size] | -n[etns] name | -N[umeric] | -a[ll] |
                  -c[olor]}
```

The **ip command** is a Linux net-tool for system and network administrators. IP stands for Internet Protocol and as the name suggests, the tool is used for configuring network interfaces. This command is **used to show or manipulate routing, devices, and tunnels** .

1. **link (l)** – used to display and modify network interfaces.
2. **address (addr/a)** – used to display and modify protocol addresses (IP, IPv6).
3. **route (r)** – used to display and alter the routing table.

3. traceroute

```
aditya@adityaraul:~$ traceroute
Usage:
  traceroute [ -46dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ] [ -m max_ttl ] [ -N squeries ] [ -p port ] [ -t tos ] [ -l flow_label ] [ -w MAX,HERE,NEAR ] [ -q nqueries ] [ -s src_addr ] [ -z sendwait ] [ --fwmark=num ] host [ packetlen ]
Options:
  -4                               Use IPv4
  -6                               Use IPv6
  -d --debug                       Enable socket level debugging
  -F --dont-fragment              Do not fragment packets
  -f first_ttl --first=first_ttl   Start from the first_ttl hop (instead from 1)
  -g gate,... --gateway=gate,...  Route packets through the specified gateway
                                   (maximum 8 for IPv4 and 127 for IPv6)
  -I --icmp                       Use ICMP ECHO for tracerouting
  -T --tcp                        Use TCP SYN for tracerouting (default port is 80)
  -i device --interface=device    Specify a network interface to operate with
  -m max_ttl --max-hops=max_ttl   Set the max number of hops (max TTL to be
                                   reached). Default is 30
  -N squeries --sim-queries=squeries Set the number of probes to be tried
                                   simultaneously (default is 16)
  -n                               Do not resolve IP addresses to their domain names
  -p port --port=port            Set the destination port to use. It is either
                                   80 for TCP, 8 for ICMP, or 1 for UDP
```

```
aditya@adityaraul:~$ traceroute
-t tos --tos=tos                Set the TOS (IPv4 type of service) or TC (IPv6
                                traffic class) value for outgoing packets
-l flow_label --flowlabel=flow_label Use specified flow_label for IPv6 packets
-w MAX,HERE,NEAR --wait=MAX,HERE,NEAR Wait for a probe no more than HERE (default 3)
                                times longer than a response from the same hop,
                                or no more than NEAR (default 10) times than some
                                next hop, or MAX (default 5.0) seconds (float
                                point values allowed too)
-q nqueries --queries=nqueries   Set the number of probes per each hop. Default is
                                3
-r                               Bypass the normal routing and send directly to a
                                host on an attached network
-s src_addr --source=src_addr    Use source src_addr for outgoing packets
-z sendwait --sendwait=sendwait  Minimal time interval between probes (default 0).
                                If the value is more than 10, then it specifies a
                                number in milliseconds, else it is a number of
                                seconds (float point values allowed too)
-e --extensions                 Show ICMP extensions (if present), including MPLS
-A --as-path-lookups            Perform AS path lookups in routing registries and
                                print results directly after the corresponding
                                addresses
-M name --module=name           Use specified module (either builtin or external)
```

The **traceroute** command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes. The above image shows how the traceroute command is used to reach the Google(172.217.26.206) host from the local machine and it also prints detail about all the hops that it visits in between.

4. tracepath

```
aditya@adityaraul: ~$ tracepath www.google.com
1?: [LOCALHOST] pmtu 1500
1:  ??? 4.266ms
1:  _gateway 3.817ms
2:  no reply
2:  10.100.100.2 4831.561ms
2:  10.100.100.2 5289.953ms
2:  10.100.100.2 4704.611ms
3:  10.1.99.2 3714.138ms
3:  10.1.99.2 2907.895ms
4:  210.212.183.61 2019.669ms asymm 5
4:  210.212.183.61 6471.116ms asymm 5
4:  210.212.183.61 5492.634ms asymm 5
5:  no reply
6:  no reply
7:  no reply
5:  ??? 10570.469ms asymm 6
5:  ??? 19579.933ms asymm 6
9:  no reply
10: no reply
11: no reply
12: no reply
13: no reply
14: no reply
15: no reply
16: no reply
17: no reply
```

The **tracepath** command in Linux is used to traces path to destination discovering MTU along this path. It uses UDP port or some random port. It is similar to **traceroute**, but it does not require superuser privileges and has no fancy options. The situation with IPv4 is worse because commercial IP routers do not return enough information in ICMP error messages. Probably, it will change, when they will be updated.

5.ping

```
aditya@adityaraul: ~$ ping www.wikipedia.com
PING www.wikipedia.com(ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3)) 56 data bytes
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=1 ttl=55 time=247 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=2 ttl=55 time=116 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=3 ttl=55 time=126 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=4 ttl=55 time=127 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=5 ttl=55 time=233 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=6 ttl=55 time=189 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=7 ttl=55 time=277 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=8 ttl=55 time=138 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=9 ttl=55 time=216 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=10 ttl=55 time=242 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=11 ttl=55 time=264 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=12 ttl=55 time=287 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=13 ttl=55 time=309 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=14 ttl=55 time=332 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=15 ttl=55 time=150 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=16 ttl=55 time=173 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=17 ttl=55 time=195 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=18 ttl=55 time=263 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=19 ttl=55 time=130 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=20 ttl=55 time=242 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=21 ttl=55 time=330 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=22 ttl=55 time=229 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=23 ttl=55 time=139 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=24 ttl=55 time=155 ms
64 bytes from ncredir-lb.egsin.wikimedia.org (2001:df2:e500:ed1a::3): icmp_seq=25 ttl=55 time=101 ms
```

Ping is short for **Packet Internet Groper**. This command is mainly used for checking the network connectivity among host/server and host. The ping command takes the URL or IP address as input and transfers the data packet to a specified address along with a "PING" message. Then, it will get a reply from the host/server. This time is known as "latency". The **ping** command is a general utility which is used for checking whether any network is present and if a host is attainable. We can test if the server is up and executing using this command.

6. nslookup

```
aditya@adityaraul:~$ nslookup amazon.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   amazon.com
Address: 52.94.236.248
Name:   amazon.com
Address: 54.239.28.85
Name:   amazon.com
Address: 205.251.242.103

aditya@adityaraul:~$
```

nslookup, which stands for "name server lookup", finds information about a named domain. By default, **nslookup** translates a domain name to an IP address (or vice versa). The nslookup command is a tool used to query Domain Name System (DNS) servers and retrieve information about a specific domain or IP address.

7. netstat

```
aditya@adityaraul:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp6      0      0 adityaraul:51764        2606:4700:90:0:79:https ESTABLISHED
tcp6      0      0 adityaraul:37192        2a04:4e42:200::64:https ESTABLISHED
tcp6      0      0 adityaraul:51068        2a04:4e42:400::64:https ESTABLISHED
udp       0      0 adityaraul:bootpc       _gateway:bootps        ESTABLISHED
udp       0      0 adityaraul:bootpc       10.1.101.29:bootps     ESTABLISHED

Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State      I-Node  Path
unix   3      [ ]        SEQPACKET  CONNECTED  36366   @7a210
unix   2      [ ]        DGRAM      CONNECTED  29083   /run/user/1000/systemd/notify
unix   3      [ ]        SEQPACKET  CONNECTED  36365   @0de9f
unix   3      [ ]        SEQPACKET  CONNECTED  36370   @abe45
unix   3      [ ]        SEQPACKET  CONNECTED  36369   @e31cc
unix   2      [ ]        DGRAM      CONNECTED  52519   /run/wpa_supplicant/wlp7s0
unix   3      [ ]        SEQPACKET  CONNECTED  42807   @a8750
unix   3      [ ]        SEQPACKET  CONNECTED  42805   @84816
unix   4      [ ]        DGRAM      CONNECTED  22653   /run/systemd/notify
unix   2      [ ]        DGRAM      CONNECTED  22667   /run/systemd/journal/syslog
unix  20      [ ]        DGRAM      CONNECTED  22676   /run/systemd/journal/dev-log
unix   9      [ ]        DGRAM      CONNECTED  22678   /run/systemd/journal/socket
unix   3      [ ]        STREAM     CONNECTED  36654   /run/user/1000/gvfsd/socket-9ZFHgVhY
unix   3      [ ]        STREAM     CONNECTED  28305
unix   3      [ ]        STREAM     CONNECTED  34015
unix   3      [ ]        STREAM     CONNECTED  20252   /run/systemd/journal/stdout
unix   3      [ ]        STREAM     CONNECTED  51627   /run/user/1000/pulse/native
unix   3      [ ]        STREAM     CONNECTED  31346   /run/user/1000/pulse/native
```



```
aditya@adityaraul: ~$ netstat -tlnp
tcp        0 0 0.0.0.0:22 0.0.0.0:22 LISTENING  sshd
tcp        0 0 0.0.0.0:80 0.0.0.0:80 LISTENING  nginx
tcp        0 0 0.0.0.0:443 0.0.0.0:443 LISTENING  nginx
tcp        0 0 0.0.0.0:8080 0.0.0.0:8080 LISTENING  nginx
tcp        0 0 0.0.0.0:9090 0.0.0.0:9090 LISTENING  nginx
tcp6       0 0 :::22  :::22  LISTENING  sshd
tcp6       0 0 :::80  :::80  LISTENING  nginx
tcp6       0 0 :::443 :::443  LISTENING  nginx
tcp6       0 0 :::8080 :::8080 LISTENING  nginx
tcp6       0 0 :::9090 :::9090 LISTENING  nginx
udp        0 0 0.0.0.0:53 0.0.0.0:53 *      *
udp6       0 0 :::53  :::53  *      *
unix 3 [ ]  STREAM  CONNECTED  26565  /run/systemd/journal/stdout
unix 3 [ ]  STREAM  CONNECTED  33933
unix 3 [ ]  STREAM  CONNECTED  30031  /run/dbus/system_bus_socket
unix 3 [ ]  STREAM  CONNECTED  28953
unix 3 [ ]  STREAM  CONNECTED  56953  /run/user/1000/pulse/native
unix 3 [ ]  STREAM  CONNECTED  33392  /run/user/1000/wayland-0
unix 3 [ ]  STREAM  CONNECTED  31232  /run/dbus/system_bus_socket
unix 3 [ ]  STREAM  CONNECTED  42915  /run/user/1000/bus
unix 2 [ ]  DGRAM   CONNECTED  33898
unix 3 [ ]  STREAM  CONNECTED  31778  /run/dbus/system_bus_socket
unix 3 [ ]  STREAM  CONNECTED  30256  /run/user/1000/at-spi/bus
unix 3 [ ]  STREAM  CONNECTED  82639
unix 3 [ ]  STREAM  CONNECTED  30257  /run/user/1000/at-spi/bus
unix 3 [ ]  STREAM  CONNECTED  31031
unix 3 [ ]  STREAM  CONNECTED  64965
unix 3 [ ]  STREAM  CONNECTED  29267  /run/user/1000/at-spi/bus
unix 3 [ ]  STREAM  CONNECTED  62918
unix 3 [ ]  STREAM  CONNECTED  50766
unix 3 [ ]  STREAM  CONNECTED  28379
unix 3 [ ]  STREAM  CONNECTED  29151  /run/user/1000/bus
unix 3 [ ]  STREAM  CONNECTED  24305  /run/user/1000/bus
unix 3 [ ]  STREAM  CONNECTED  33955
unix 3 [ ]  STREAM  CONNECTED  30253  /run/user/1000/at-spi/bus
unix 3 [ ]  STREAM  CONNECTED  31030
unix 3 [ ]  DGRAM   CONNECTED  33823
unix 3 [ ]  STREAM  CONNECTED  28828
```

netstat command without any argument displays information about the Linux networking subsystem. By default, netstat displays a list of open sockets. Netstat is a command line utility to display all the network connections on a system. It displays all the tcp, udp and unix socket connections. Apart from connected sockets it also displays listening sockets that are waiting for incoming connections.

8.dig

```
aditya@adityaraul:~$ dig google.com

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41850
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;google.com.                IN      A

;; ANSWER SECTION:
google.com.                 142     IN      A      142.250.67.142

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sun Sep 10 23:24:26 IST 2023
;; MSG SIZE rcvd: 55

aditya@adityaraul:~$
```

dig command stands for **Domain Information Groper**. It is used for retrieving information about DNS name servers. It is basically used by network administrators. It is used for verifying & troubleshooting DNS problems and to perform DNS lookups and returns the queried answers from the name server.

9. host

```
aditya@adityaraul: ~$ host
Usage: host [-aCdilrTvVw] [-c class] [-N ndots] [-t type] [-W time]
          [-R number] [-m flag] [-p port] hostname [server]
-a is equivalent to -v -t ANY
-A is like -a but omits RRSIG, NSEC, NSEC3
-c specifies query class for non-IN data
-C compares SOA records on authoritative nameservers
-d is equivalent to -v
-l lists all hosts in a domain, using AXFR
-m set memory debugging flag (trace|record|usage)
-N changes the number of dots allowed before root lookup is done
-p specifies the port on the server to query
-r disables recursive processing
-R specifies number of retries for UDP packets
-s a SERVFAIL response should stop query
-t specifies the query type
-T enables TCP/IP mode
-U enables UDP mode
-v enables verbose output
-V print version number and exit
-w specifies to wait forever for a reply
-W specifies how long to wait for a reply
-4 use IPv4 query transport only
-6 use IPv6 query transport only
aditya@adityaraul: ~$
```

To display the Domain's [IP](#) address, execute the host command followed by the domain name, as follows:

```
aditya@adityaraul: ~$ host google.com
google.com has address 142.250.67.142
google.com has IPv6 address 2404:6800:4009:811::200e
google.com mail is handled by 10 smtp.google.com.
aditya@adityaraul: ~$
```

Linux **host** command displays domain name for given IP address or vice-versa. It also performs DNS lookups related to the DNS query. The host command's default behavior displays a summary of its command-line arguments and supported options.

10. route

```
aditya@adityaraul: ~$ route
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
default          _gateway       0.0.0.0         UG    600    0      0 wlp7s0
default          _gateway       0.0.0.0         UG   20100    0      0 enp8s0
10.200.56.0      0.0.0.0        255.255.254.0   U     100    0      0 enp8s0
link-local       0.0.0.0        255.255.0.0     U     1000    0      0 enp8s0
192.168.48.0     0.0.0.0        255.255.255.0   U      600    0      0 wlp7s0
aditya@adityaraul:~$ |
```

The **route** command displays and manipulate IP routing table for your system. A router is a device which is basically used to determine the best way to route packets to a destination. It displays all existing routing table entries on our system. It shows that if the destination is within the network range of 10.100.96.0 to 255.255.240.255, then the gateway is *, which is 0.0.0.0. This is a special address which represents an invalid or non-existent destination.

11. iwconfig

```
aditya@adityaraul:~$ iwconfig
lo          no wireless extensions.

enp8s0      no wireless extensions.

wlp7s0      IEEE 802.11  ESSID:"Aditya "
            Mode:Managed  Frequency:5.18 GHz  Access Point: 7E:0E:9F:35:C4:8A
            Bit Rate=292.5 Mb/s   Tx-Power=20 dBm
            Retry short limit:7   RTS thr:off   Fragment thr:off
            Power Management:on
            Link Quality=70/70   Signal level=-37 dBm
            Rx invalid nwid:0   Rx invalid crypt:0   Rx invalid frag:0
            Tx excessive retries:36   Invalid misc:1391   Missed beacon:0

aditya@adityaraul:~$
```

The command **iwconfig** configures a wireless network interface. You can view and set basic wi-fi details like SSID and encryption.

12. wget

```
aditya@adityaraul: ~$ wget www.amazon.com
--2023-09-10 23:32:20-- http://www.amazon.com/
Resolving www.amazon.com (www.amazon.com)... 2600:9000:237b:8800:7:49a5:5fd2:8621, 2600:9000:237b:b400:7:49a5:5fd2:8621, 2600:9000:237b:c200:7:49a5:5fd2:8621, ...
Connecting to www.amazon.com (www.amazon.com)|2600:9000:237b:8800:7:49a5:5fd2:8621|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.amazon.com/ [following]
--2023-09-10 23:32:20-- https://www.amazon.com/
Connecting to www.amazon.com (www.amazon.com)|2600:9000:237b:8800:7:49a5:5fd2:8621|:443... connected.
HTTP request sent, awaiting response... 503 Service Unavailable
2023-09-10 23:32:21 ERROR 503: Service Unavailable.

aditya@adityaraul: ~$
```

Command **wget** stands for **web get**. The above output clearly shows that the wget command connects to the `www.google.com` server. Wget is the non-interactive network downloader which is used to download files from the server even when the user has not logged on to the system and it can work in the background without hindering the current process.

13. arp

```
aditya@adityaraul: ~$ arp
Address          HWtype  HWaddress      Flags Mask    Iface
_gateway         ether    7e:0e:9f:35:c4:8a  C             wlp7s0
10.200.57.5       ether    60:2e:20:48:a9:69  C             enp8s0
10.200.57.12      ether    60:2e:20:48:ab:a9  C             enp8s0
10.200.56.160     ether    60:18:95:24:7f:c9  C             enp8s0
10.200.57.13      ether    60:2e:20:48:ad:29  C             enp8s0
_gateway         ether    b8:38:61:70:4e:00  C             enp8s0
10.200.56.28      ether    84:69:93:52:33:5c  C             enp8s0
aditya@adityaraul: ~$
```

The **arp** command displays and modifies the Internet-to-adaptor address translation tables used by the Address in Networks and communication management. The **arp** command displays the current ARP entry for the host specified by the `HostName` variable.


```
aditya@adityaraul: ~  
aditya@adityaraul:~$ telnet  
telnet> h  
Commands may be abbreviated.  Commands are:  
  
close                close current connection  
logout              forcibly logout remote user and close the connection  
display             display operating parameters  
mode                try to enter line or character mode ('mode ?' for more)  
open                connect to a site  
quit                exit telnet  
send                transmit special characters ('send ?' for more)  
set                 set operating parameters ('set ?' for more)  
unset               unset operating parameters ('unset ?' for more)  
status              print status information  
toggle              toggle operating parameters ('toggle ?' for more)  
slc                 set treatment of special characters  
  
z                   suspend telnet  
environ             change environment variables ('environ ?' for more)  
telnet> |
```

The **telnet** command in Linux stands for a '*terminal over network*'. It helps you connect to a telnet server. Using the telnet command, you can set up a client-server connection with a remote server using the TCP protocol through a remote port.

16.whois

```
aditya@adityaraul: ~  
aditya@adityaraul:~$ whois google.com  
Domain Name: GOOGLE.COM  
Registry Domain ID: 2138514_DOMAIN_COM-VRSN  
Registrar WHOIS Server: whois.markmonitor.com  
Registrar URL: http://www.markmonitor.com  
Updated Date: 2019-09-09T15:39:04Z  
Creation Date: 1997-09-15T04:00:00Z  
Registry Expiry Date: 2028-09-14T04:00:00Z  
Registrar: MarkMonitor Inc.  
Registrar IANA ID: 292  
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com  
Registrar Abuse Contact Phone: +1.2086851750  
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited  
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited  
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited  
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited  
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited  
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited  
Name Server: NS1.GOOGLE.COM  
Name Server: NS2.GOOGLE.COM  
Name Server: NS3.GOOGLE.COM  
Name Server: NS4.GOOGLE.COM  
DNSSEC: unsigned  
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/  
>>> Last update of whois database: 2023-09-10T18:07:05Z <<<  
  
For more information on Whois status codes, please visit https://icann.org/epp
```

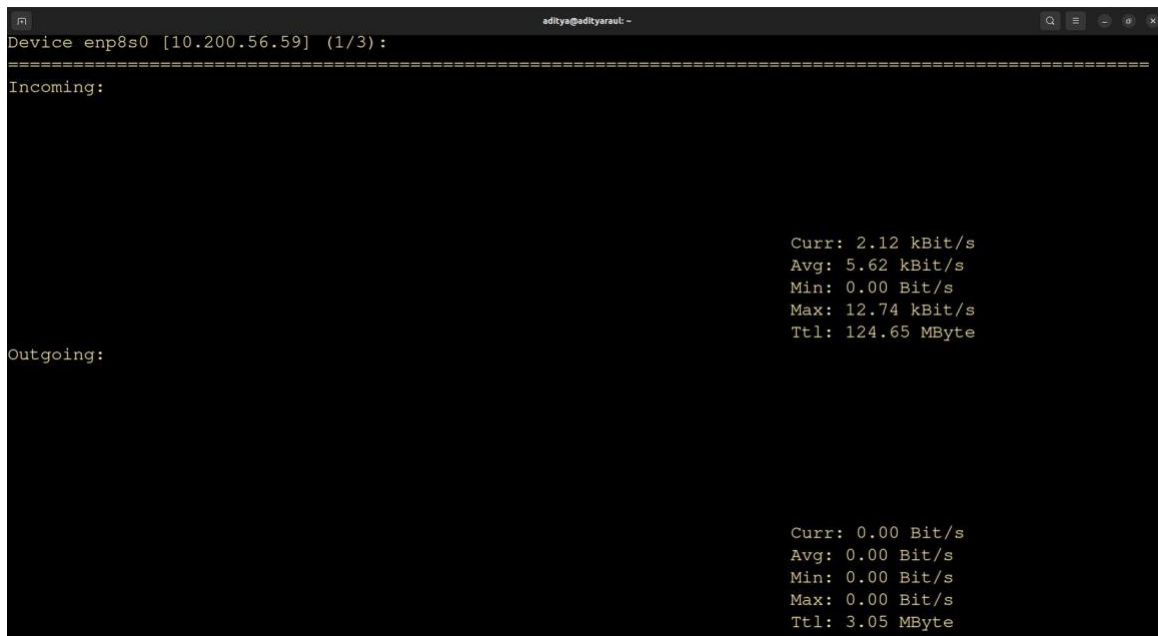
whois command searches a user name directory and displays information about the user ID or nickname specified in the Name parameter. The whois command tries to reach ARPANET host internic.net where it examines a user-name database to obtain information.

17. ifplugst

```
aditya@adityaraul: ~  
aditya@adityaraul:~$ ifplugstatus  
lo: link beat detected  
enp8s0: link beat detected  
wlp7s0: link beat detected
```

ifpulgst command tells us whether a cable is plugged into our network interface or not. It is commonly used to identify the connectivity status of a network cable and determine if a network link is active or inactive.

18. nload



nload is a command-line tool used for monitoring network traffic and bandwidth usage in real-time. It will display the incoming and outgoing traffic using two graphs. This console-based application also displays info like the total amount of transferred data and min/max network.

19. mail

```
aditya@adityaraul:~$ mail -s "Hello CN Assignment" rauladitya22@gmail.com <<< 'How Are you'
aditya@adityaraul:~$ mail
"/var/mail/aditya": 1 message 1 new
>N 1 Mail Delivery Syst Sun Sep 10 23:48 75/242
? |
```

Mail is the quickest way of communicating messages. There are different email client servers that allow users to send emails, besides this, Linux by default provides the feature of sending an email using the “**mail**” command through its terminal. We can write the subject, message along with the email address of the recipient and send it by just executing a single command.