Tishk International University
Science Faculty
IT Department

Open Source OS (Linux)

Lecture07: Linux Networking

4th Grade - Fall Semester 2021-2022

Instructor: Alaa Ghazi

Lecture07: Linux Networking Topics

1 Basics

2 ifconfig for Determining Network Interfaces

3 ifup / ifdown Commands

4 Assigning Temporary IP Address

5 Address Resolution Protocol (ARP)

6 Displaying and Setting the Hostname

7 DNS hostnames

8 DNS Troubleshooting

9 The /etc/hosts file

10 Dynamic Host Configuration Protocol (DHCP)

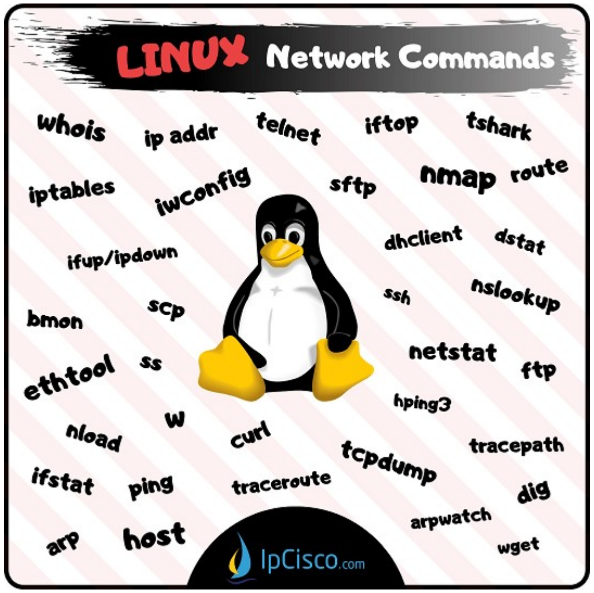
11 Starting and Stopping Services

12 Enabling and Disabling Services

13 Testing Connectivity with Ping

14 The netstat Command

15 Packet Sniffing with tcpdump



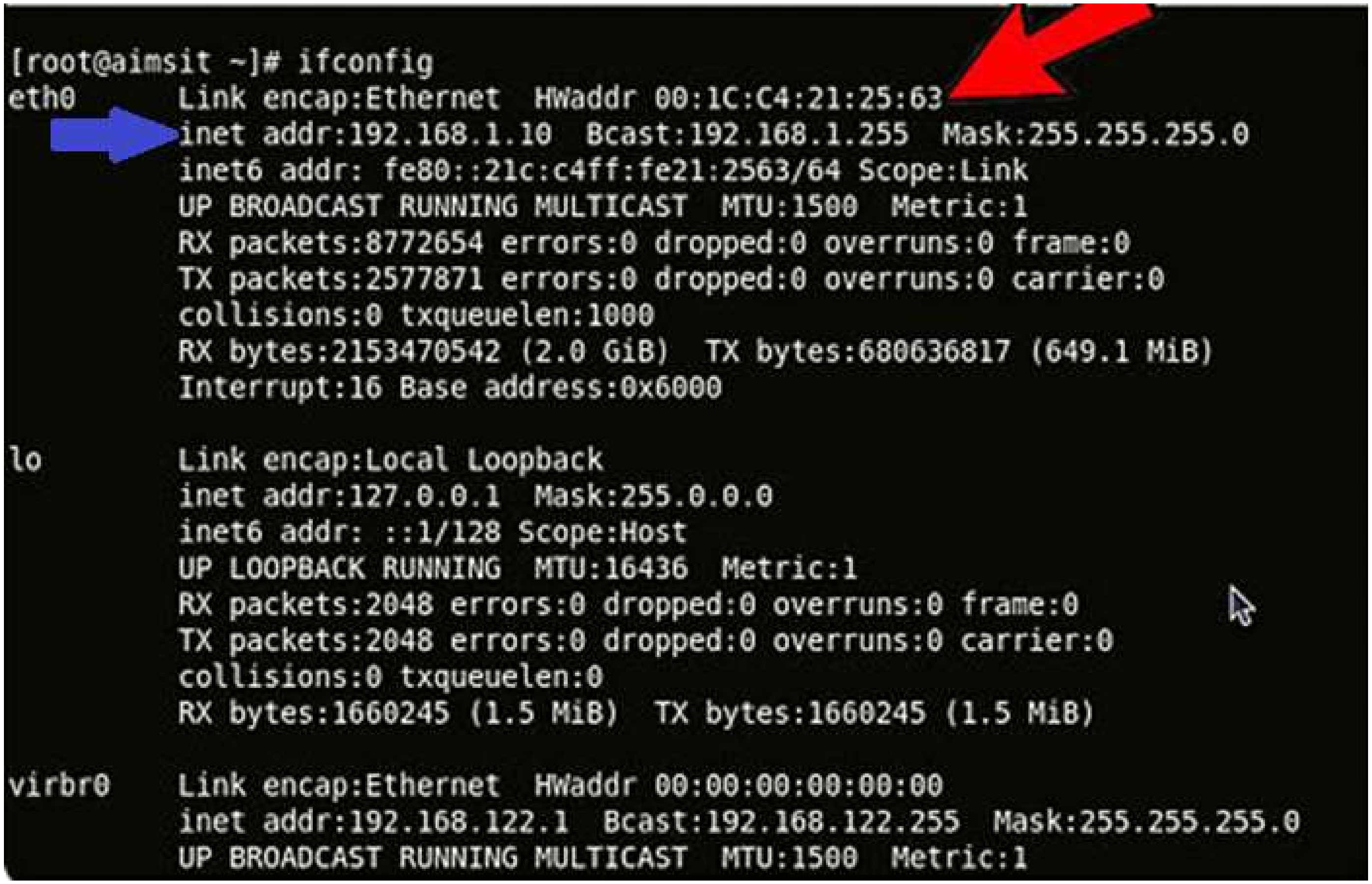
1 Basics

Host & Interface

- Hosts are computers/machines//individual systems.
- Each host can have one or more network interfaces
- Each interface represents a connection to a different physical network
- Each interface can have One Hardware address (MAC address).
- Each interface can have multiple IP addresses.
- Each host (machine) can have one routing table.
- **Software Loopback interface**: which does not have hardware associated with it, and does not require a physical connection to a network and is used for the communication of network applications hosted in the same system.
- **Loopback IP address**: is a special IP address, 127.0.0.1, reserved for use with the software loopback interface of the network card. For example it can be used to open the home page of web server installed on the same machine.

2 ifconfig for Determining Network Interfaces

- **ifconfig** is short for **Interface configuration**, it prints information about available interfaces and their configuration.
- The MAC address will be listed next to **Hwaddr**
- The IP address will be listed after **inet addr**



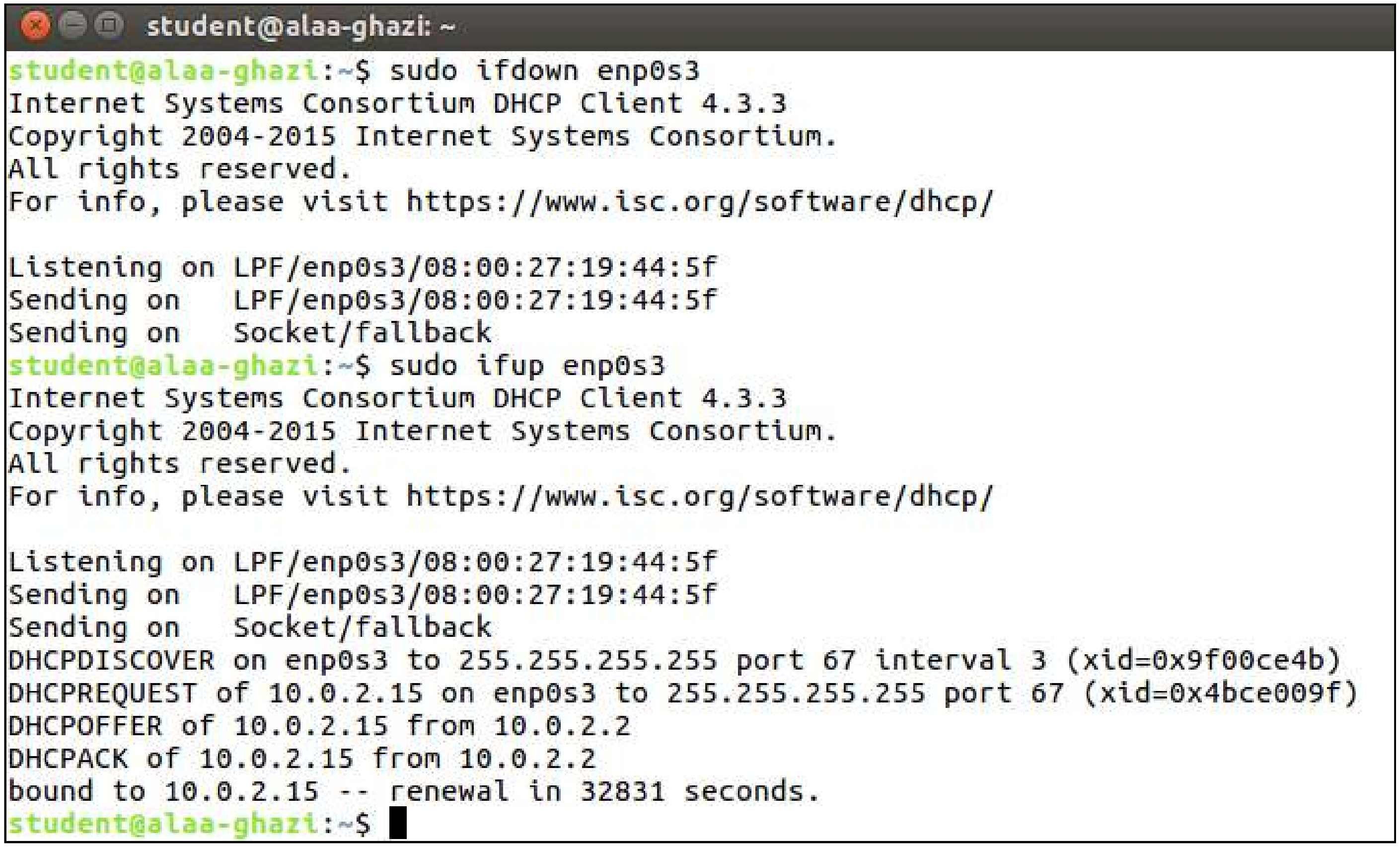
```
[root@aimsit ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1C:C4:21:25:63
          inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::21c:c4ff:fe21:2563/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8772654 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2577871 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2153470542 (2.0 GiB)  TX bytes:680636817 (649.1 MiB)
          Interrupt:16 Base address:0x6000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2048 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2048 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1660245 (1.5 MiB)  TX bytes:1660245 (1.5 MiB)

virbr0    Link encap:Ethernet  HWaddr 00:00:00:00:00:00
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

3 ifup / ifdown Commands

- Can be used to bring the interface up and down



```
student@alaa-ghazi:~$ sudo ifdown enp0s3
Internet Systems Consortium DHCP Client 4.3.3
Copyright 2004-2015 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/enp0s3/08:00:27:19:44:5f
Sending on   LPF/enp0s3/08:00:27:19:44:5f
Sending on   Socket/fallback
student@alaa-ghazi:~$ sudo ifup enp0s3
Internet Systems Consortium DHCP Client 4.3.3
Copyright 2004-2015 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/enp0s3/08:00:27:19:44:5f
Sending on   LPF/enp0s3/08:00:27:19:44:5f
Sending on   Socket/fallback
DHCPDISCOVER on enp0s3 to 255.255.255.255 port 67 interval 3 (xid=0x9f00ce4b)
DHCPREQUEST of 10.0.2.15 on enp0s3 to 255.255.255.255 port 67 (xid=0x4bce009f)
DHCPOFFER of 10.0.2.15 from 10.0.2.2
DHCPACK of 10.0.2.15 from 10.0.2.2
bound to 10.0.2.15 -- renewal in 32831 seconds.
student@alaa-ghazi:~$ █
```

4 Assigning Temporary IP Address

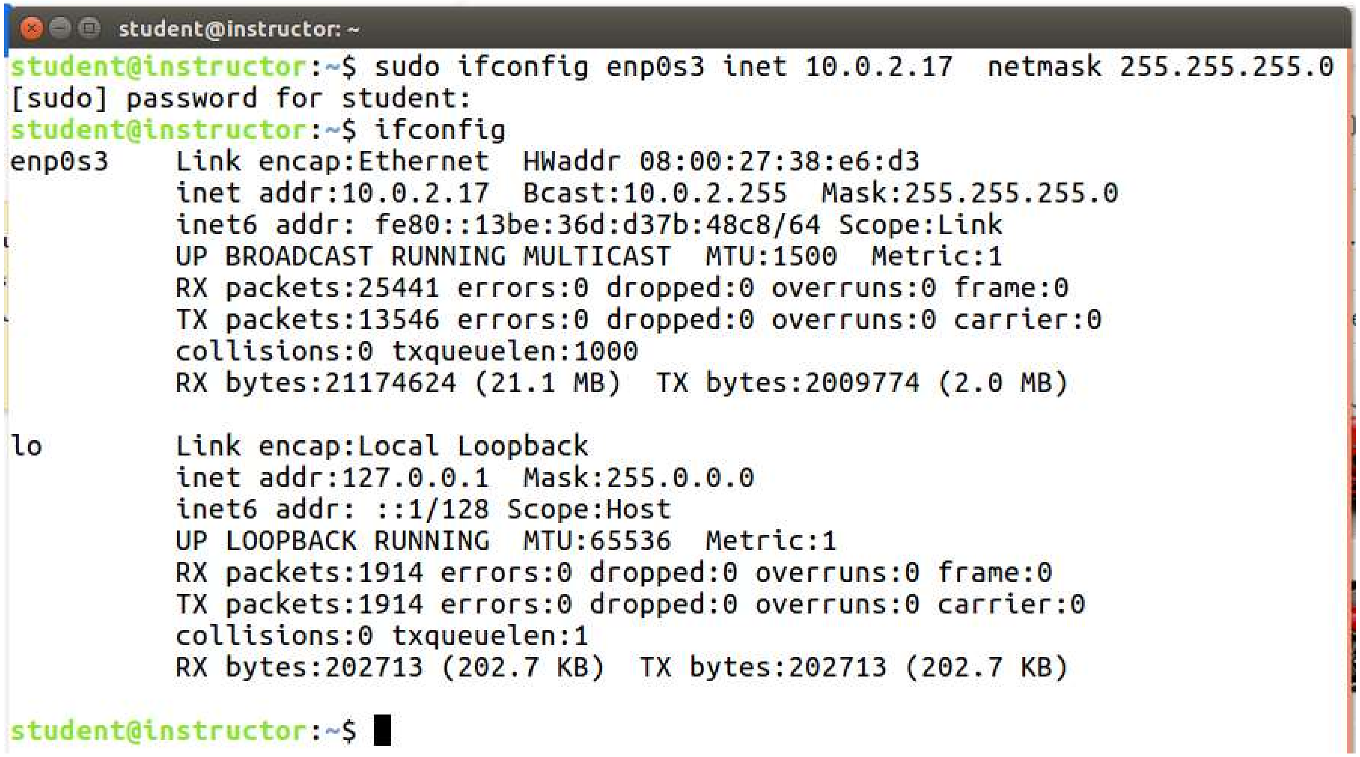
Changes made with ifconfig are not permanent (do not modify interfaces file).

Function: Configure network interface parameters

Syntax: ifconfig [-AaC] [interface] [address_family] [address [dest_address]] [parameters]

Examples

```
$ sudo ifconfig enp0s3 inet 10.0.2.17  netmask 255.255.255.0
$ ifconfig
```



```
student@instructor:~$ sudo ifconfig enp0s3 inet 10.0.2.17  netmask 255.255.255.0
[sudo] password for student:
student@instructor:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:38:e6:d3
          inet addr:10.0.2.17  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::13be:36d:d37b:48c8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25441 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13546 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:21174624 (21.1 MB)  TX bytes:2009774 (2.0 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1914 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1914 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:202713 (202.7 KB)  TX bytes:202713 (202.7 KB)

student@instructor:~$ 
```

5 Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP) discovers the hardware (MAC) address associated with a particular IP address on other computers in the network.

The ARP Table keeps a list of each IP address and its matching MAC address.

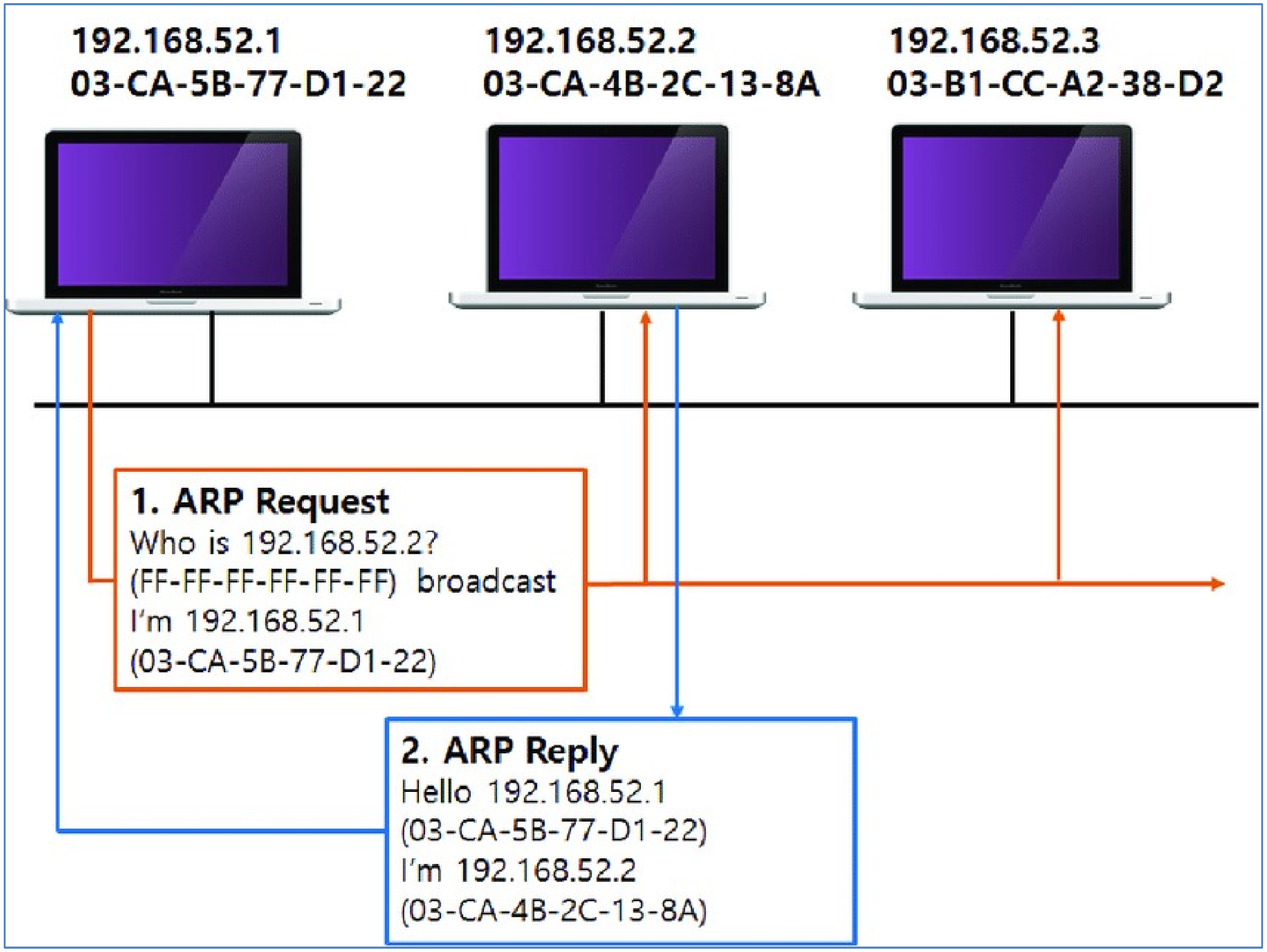
The ARP Table is dynamic, but users on a network can also configure a static ARP entries containing IP addresses and MAC addresses.

The **arp** command show ARP table, adds or deletes entries, and flushes the table.

arp command options:

Command	Action
arp -a	displays the arp table entries
arp -s hostname hwaddr	manually create an arp address mapping entry
arp -d hostname	remove any entry for the specified host

How ARP Protocol Works (not required in the exam)



192.168.52.1
03-CA-5B-77-D1-22

192.168.52.2
03-CA-4B-2C-13-8A

192.168.52.3
03-B1-CC-A2-38-D2

1. ARP Request
Who is 192.168.52.2?
(FF-FF-FF-FF-FF-FF) broadcast
I'm 192.168.52.1
(03-CA-5B-77-D1-22)

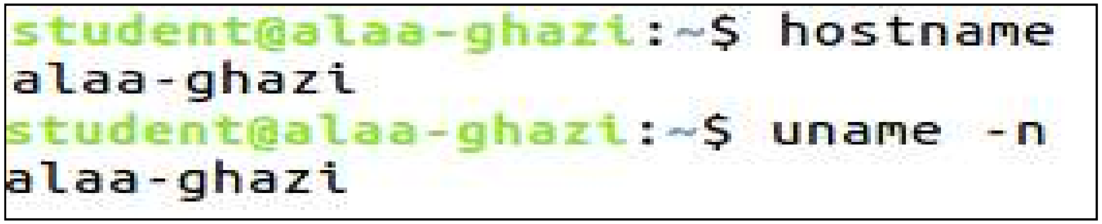
2. ARP Reply
Hello 192.168.52.1
(03-CA-5B-77-D1-22)
I'm 192.168.52.2
(03-CA-4B-2C-13-8A)

6 Displaying and Setting the Hostname

- To display the hostname in Linux use below commands

$ hostname

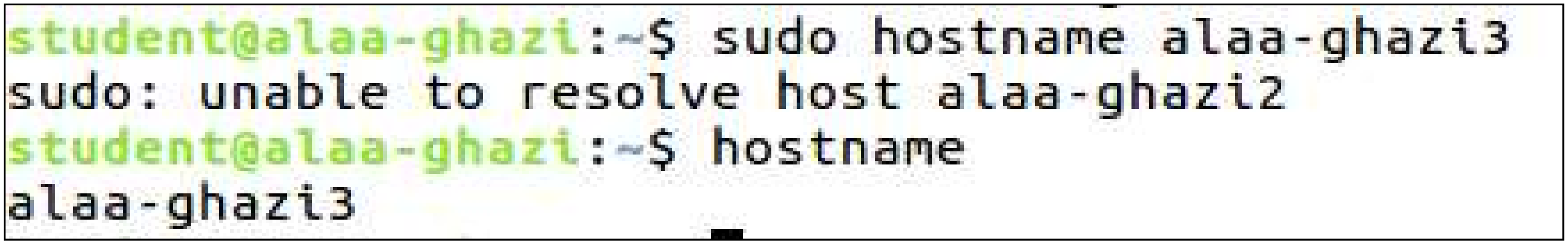
$ uname –n



```
student@alaa-ghazi:~$ hostname
alaa-ghazi
student@alaa-ghazi:~$ uname -n
alaa-ghazi
```

- To change hostname use below command

$ sudo hostname <newhostname>



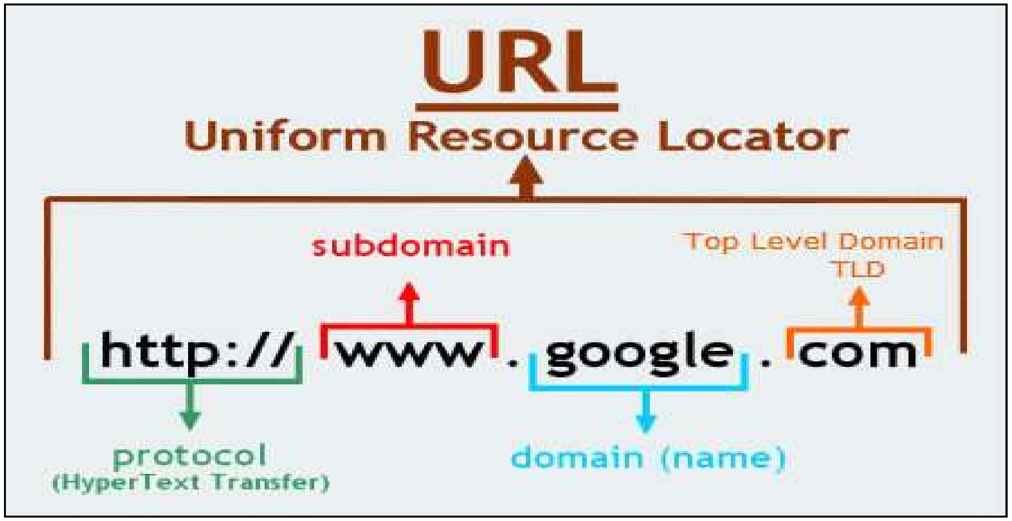
```
student@alaa-ghazi:~$ sudo hostname alaa-ghazi3
sudo: unable to resolve host alaa-ghazi2
student@alaa-ghazi:~$ hostname
alaa-ghazi3
```

7 DNS hostnames

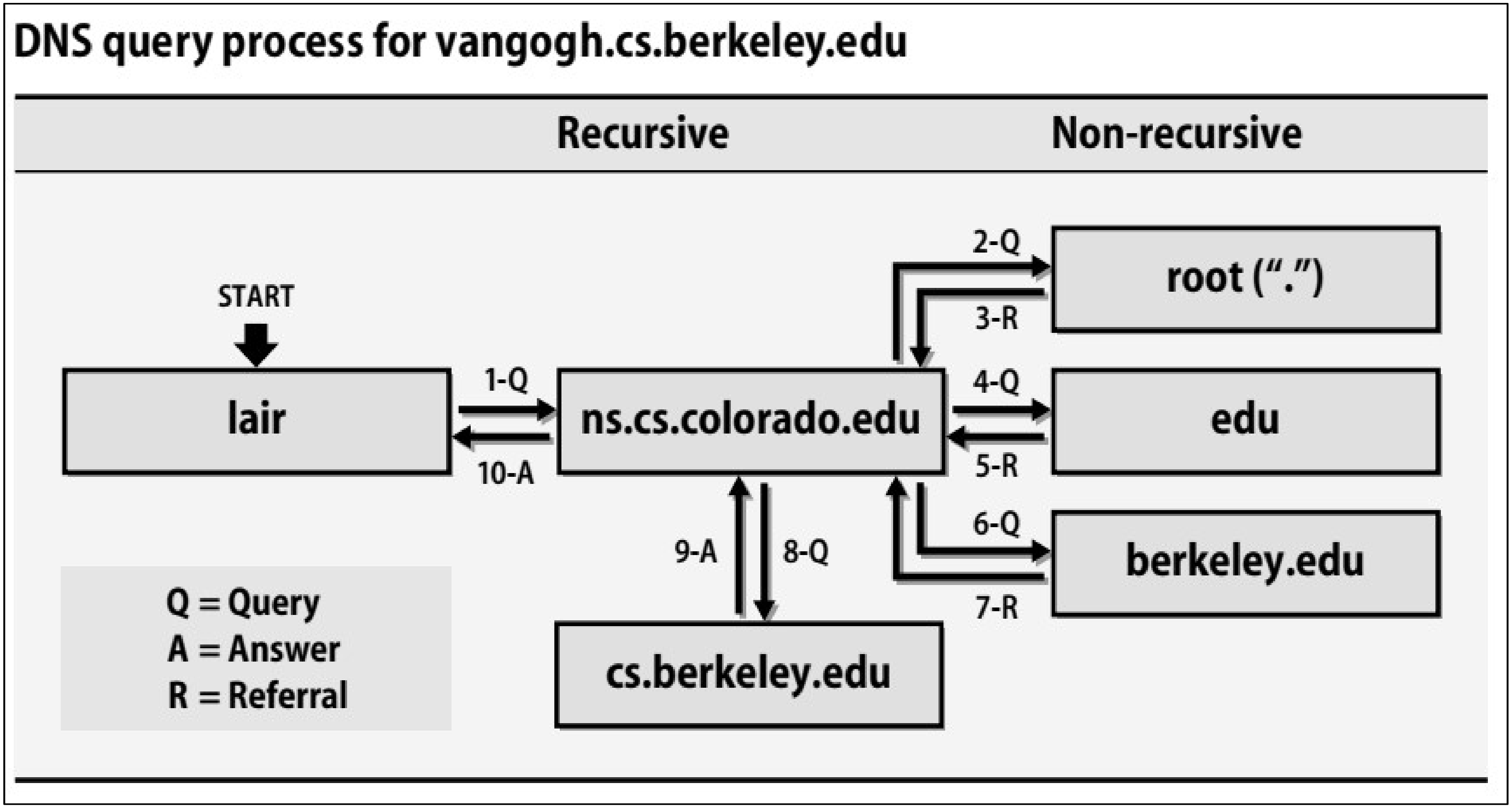
- **DNS hostnames**: human-readable name which can be resolved to an IP address
- **Uniform Resource Locator (URL)** is the address of a resource on the Internet and the protocol used to access it.
- Fully Qualified Domain Name (FQDN): is the complete name for a specific host on the internet. Example is: www.google.com

FQDN contains:

- TLD (Top Level Domain): examples are: .com, .net, .org,
- Domains: below (to the left of) TLD
- sub-domain: below (to the left of) the domain



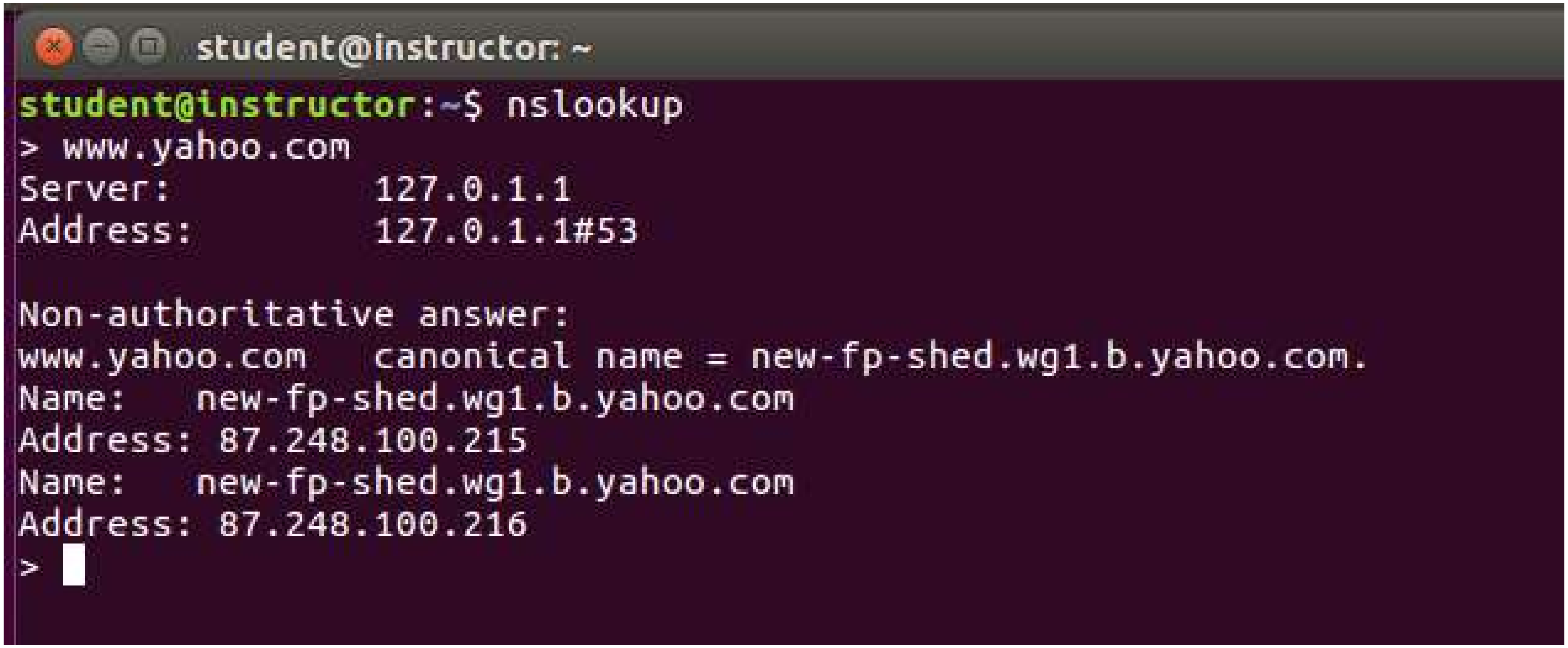
DNS Query Example (not required in the exam)



DNS query process for vangogh.cs.berkeley.edu

8 DNS Troubleshooting

nslookup is a command-line administrative tool for testing and troubleshooting **DNS** servers (**Domain Name Server**). It is used to query specific **DNS** resource records (**RR**)

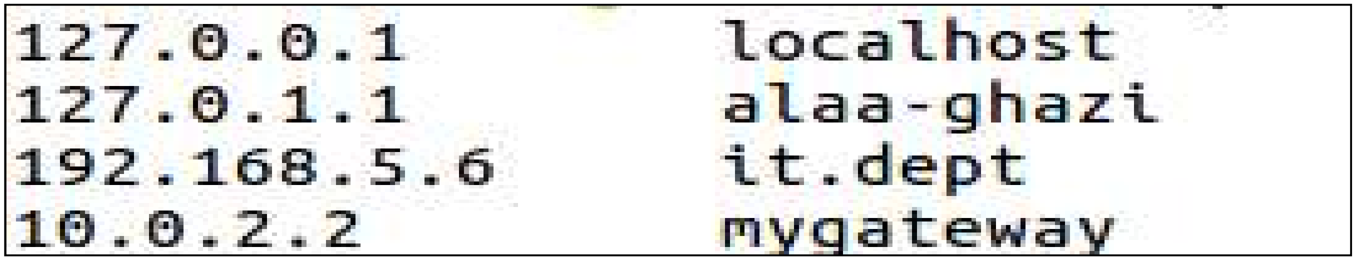


```
student@instructor:~$ nslookup
> www.yahoo.com
Server:         127.0.1.1
Address:        127.0.1.1#53

Non-authoritative answer:
www.yahoo.com   canonical name = new-fp-shed.wg1.b.yahoo.com.
Name:   new-fp-shed.wg1.b.yahoo.com
Address: 87.248.100.215
Name:   new-fp-shed.wg1.b.yahoo.com
Address: 87.248.100.216
> 
```

9 The /etc/hosts file

- /etc/hosts – is a file contains a list of IP addresses and associated hostnames.
- Format: Each line in hosts file will correspond to one IP address entry followed by associated hostname.



```
127.0.0.1          localhost
127.0.1.1          alaa-ghazi
192.168.5.6        it.dept
10.0.2.2           mygateway
```

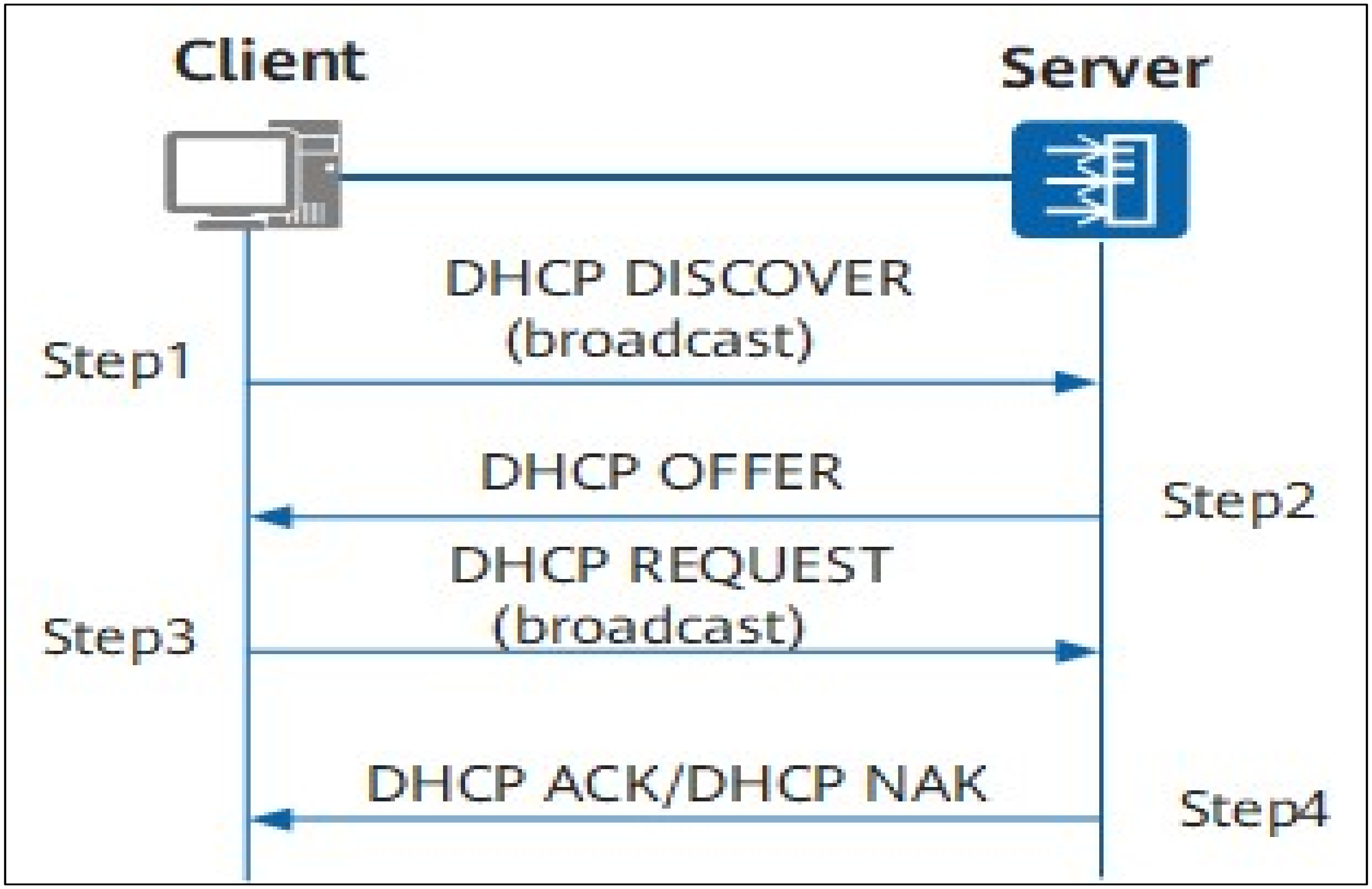
- So you can refer to the host 10.0.2.2 by name mygateway
- /etc/hosts is local to your Linux system. It does not propagate to the rest of the network.
- In Linux by default the system looks at hostnames in files then DNS

10 Dynamic Host Configuration Protocol (DHCP)

● **Dynamic Host Configuration Protocol (DHCP)** is a network management protocol used to automate the process of configuring devices on IP networks
● DHCP servers assign IP address configuration to DHCP Clients
(IP Address, netmask, gateway, DNS servers)

●Each IP is "leased" from the pool of IP addresses the DHCP server manages.
● The lease expiration time is configurable on the DHCP server.
(1hr, 1day, 1 week, etc.)
● The client must renew the lease if it wants to keep using the IP address. If no renewal is received, the IP is available to other DHCP clients.

How DHCP Works Diagram (not required in the exam)

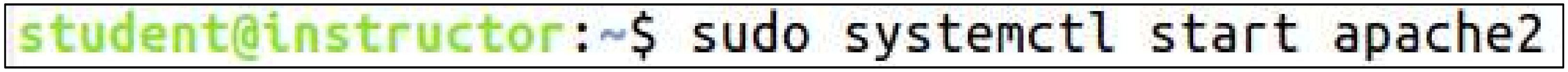


11 Starting and Stopping Services

- To start a Linux service, use the **start** command. If you are running as a non-root user, you will have to use sudo since this will affect the state of the operating system. The command is:

sudo systemctl start application

Example:

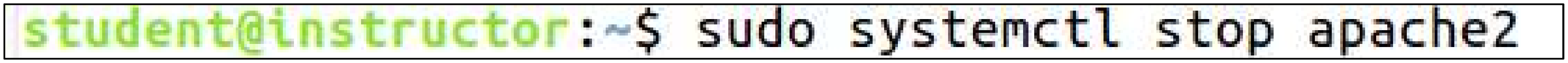


```
student@instructor:~$ sudo systemctl start apache2
```

To stop a service use below command. Stopping a service with this command is temporary, and the service will not start automatically in the next boot (if not enabled)

sudo systemctl stop application

Example:

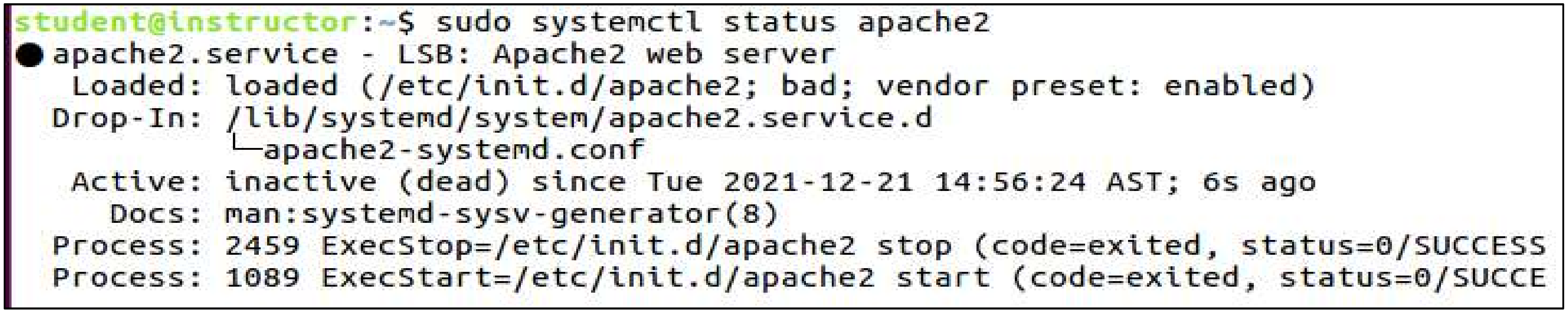


```
student@instructor:~$ sudo systemctl stop apache2
```

To check the status of the service use below command:

sudo systemctl status application

Example:



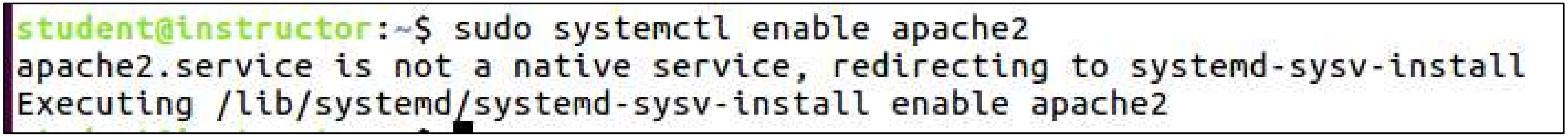
```
student@instructor:~$ sudo systemctl status apache2
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: inactive (dead) since Tue 2021-12-21 14:56:24 AST; 6s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 2459 ExecStop=/etc/init.d/apache2 stop (code=exited, status=0/SUCCESS
  Process: 1089 ExecStart=/etc/init.d/apache2 start (code=exited, status=0/SUCCE
```

12 Enabling and Disabling Services

- To tell Linux to start services automatically at boot, you must enable them. This will not start the service immediately, but the service will start automatically at next boot. To start a service at boot, use the command:

sudo systemctl enable application

Example:

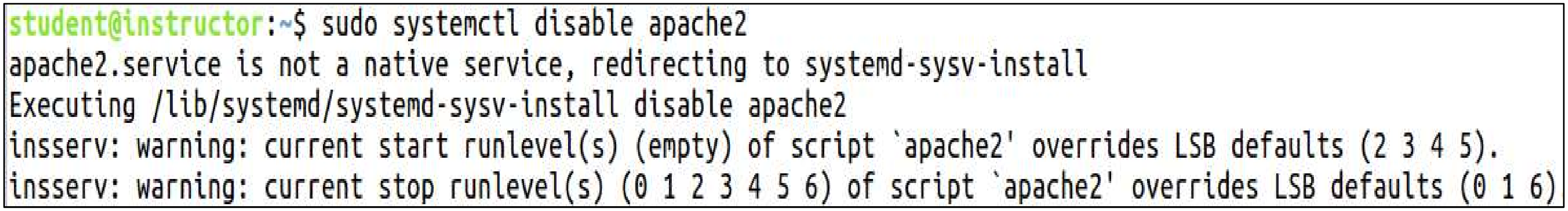


```
student@instructor:~$ sudo systemctl enable apache2
apache2.service is not a native service, redirecting to systemd-sysv-install
Executing /lib/systemd/systemd-sysv-install enable apache2
```

- To tell Linux to not to start services automatically at boot, you must disable them. This will not stop the service immediately, but the service will not start automatically at next boot. To disable a service at boot, use the command:

sudo systemctl disable application

Example:



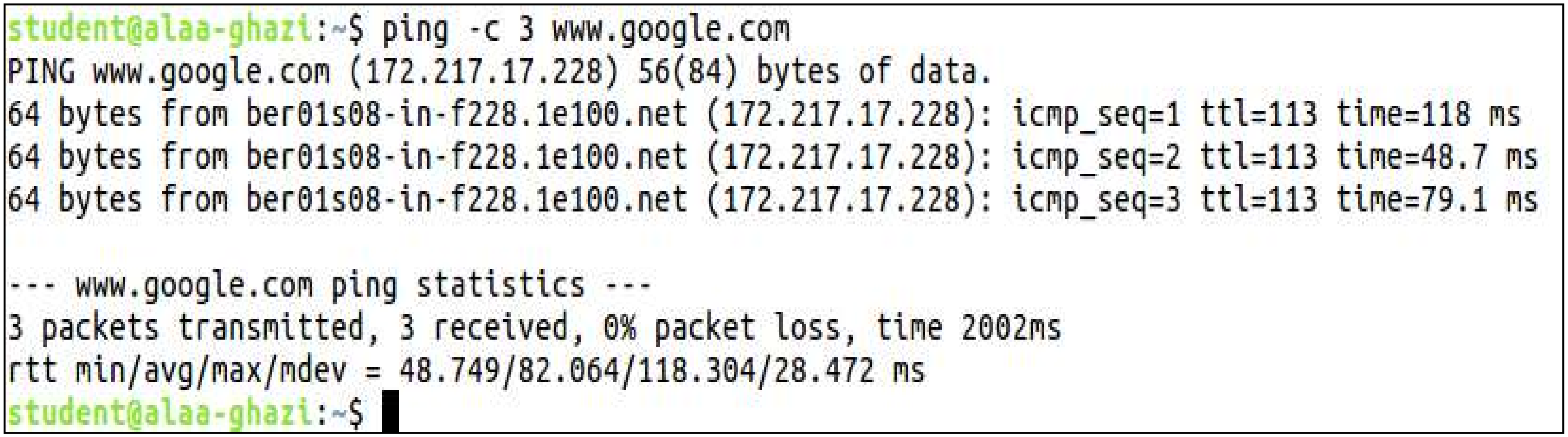
```
student@instructor:~$ sudo systemctl disable apache2
apache2.service is not a native service, redirecting to systemd-sysv-install
Executing /lib/systemd/systemd-sysv-install disable apache2
insserv: warning: current start runlevel(s) (empty) of script `apache2' overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script `apache2' overrides LSB defaults (0 1 6)
```

13 Testing Connectivity with Ping

ping sends ICMP ECHO_REQUEST packet to a host in order to check if destination host is reachable

Format:

```
ping HOST
ping -c COUNT HOST
```



```
student@alaa-ghazi:~$ ping -c 3 www.google.com
PING www.google.com (172.217.17.228) 56(84) bytes of data.
64 bytes from ber01s08-in-f228.1e100.net (172.217.17.228): icmp_seq=1 ttl=113 time=118 ms
64 bytes from ber01s08-in-f228.1e100.net (172.217.17.228): icmp_seq=2 ttl=113 time=48.7 ms
64 bytes from ber01s08-in-f228.1e100.net (172.217.17.228): icmp_seq=3 ttl=113 time=79.1 ms

--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 48.749/82.064/118.304/28.472 ms
student@alaa-ghazi:~$
```

14 The `netstat` Command

netstat command shows network status. It has below options:

-n Display numerical addresses and ports.

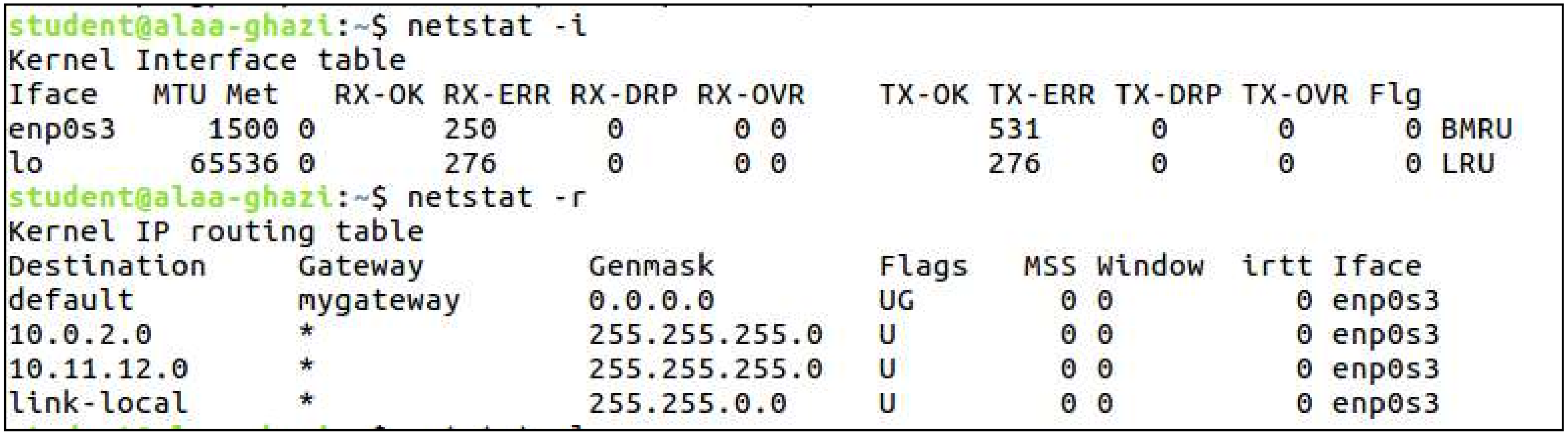
-i Displays a list of network interfaces.

-r Displays the route table.

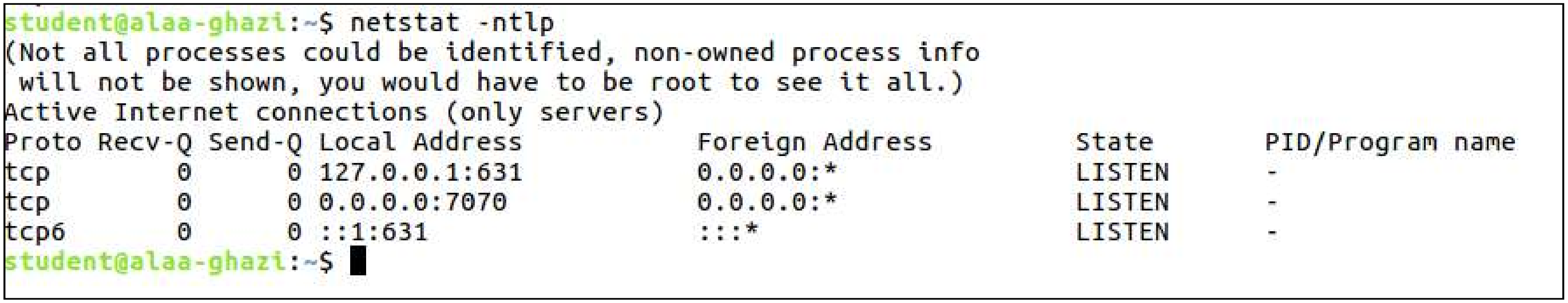
-p Display the PID and program used.

-l Display listening sockets.

Examples are:



```
student@alaa-ghazi:~$ netstat -i
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enp0s3    1500 0       250      0      0 0           531      0      0      0 BMRU
lo       65536 0       276      0      0 0           276      0      0      0 LRU
student@alaa-ghazi:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         mygateway       0.0.0.0         UG        0 0          0 enp0s3
10.0.2.0        *               255.255.255.0   U         0 0          0 enp0s3
10.11.12.0      *               255.255.255.0   U         0 0          0 enp0s3
link-local      *               255.255.0.0     U         0 0          0 enp0s3
```



```
student@alaa-ghazi:~$ netstat -ntlp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:7070            0.0.0.0:*               LISTEN      -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
student@alaa-ghazi:~$ 
```

15 Packet Sniffing with `tcpdump`

tcpdump monitoring sent/received data for each connection.

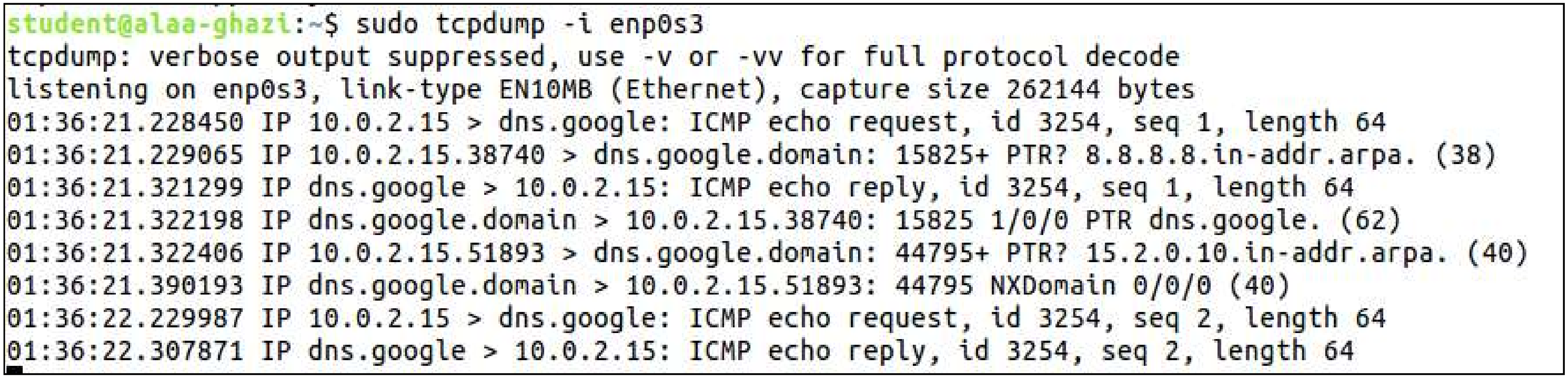
tcpdump options:
-n Display numerical addresses and ports.

-A Display ASCII (text) output.

-v Verbose mode. Produce more output.

-vvv Even more verbose output.

Example is:



```
student@alaa-ghazi:~$ sudo tcpdump -i enp0s3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
01:36:21.228450 IP 10.0.2.15 > dns.google: ICMP echo request, id 3254, seq 1, length 64
01:36:21.229065 IP 10.0.2.15.38740 > dns.google.domain: 15825+ PTR? 8.8.8.8.in-addr.arpa. (38)
01:36:21.321299 IP dns.google > 10.0.2.15: ICMP echo reply, id 3254, seq 1, length 64
01:36:21.322198 IP dns.google.domain > 10.0.2.15.38740: 15825 1/0/0 PTR dns.google. (62)
01:36:21.322406 IP 10.0.2.15.51893 > dns.google.domain: 44795+ PTR? 15.2.0.10.in-addr.arpa. (40)
01:36:21.390193 IP dns.google.domain > 10.0.2.15.51893: 44795 NXDomain 0/0/0 (40)
01:36:22.229987 IP 10.0.2.15 > dns.google: ICMP echo request, id 3254, seq 2, length 64
01:36:22.307871 IP dns.google > 10.0.2.15: ICMP echo reply, id 3254, seq 2, length 64
```

LAB 07
Linux Networking

LAB 07 : Linux Networking

Practice with each command below:

TEST 1) Test the command   ifconfig

TEST 2) Using nano tool edit the /etc/hosts and add the entry

```
8.8.8.8 mydns
```

TEST 4) Test the commands ifup, ifdown on interface enp0s3

TEST 5) Test the tool nslookup and find the DNS record of www.tiu.edu.iq

TEST 6) Test the arp command with options: -a

TEST 7) Install the web server application apache2

TEST 8) Using firefox as a web client, clear firefox history, then open home page of appache2 using loopback address 127.0.0.1

TEST 9) Stop apache2 service, and repeat **TEST 8**

TEST 10) Start apache2 service, and repeat **TEST 8**

TEST 11) Disable apache2 service, reboot, and repeat **TEST 8**

TEST 12) Enable apache2 service, reboot, and repeat **TEST 8**

TEST 13) Test the netstat command with options: -n , -r

TEST 14) Open two terminals in First one run the command

```
tcpdump  -i  enp0s3
```

In the second terminal run the command

```
ping –c 2 mydns
```