Name: Md.Abdullah
ID : IT-17015

Lab report no : 07

Lab report name: Python for networking

Objectives:
Learn how to find

- ✔ find network interfaces
- ✔ interface ip address
- ✔ interface status

Theory:

Explain in your own words what is a network interface?

Ans:
Physical connection to network interface.
A network interface can refer to any kind of software interface to networking hardware. For instance, if you have two network cards in your computer, you can control and configure each network interface associated with them individually.

Explain why it is relevant to communicate using sockets?

Ans:
Provide an organized way of communicate.
Sockets allow communication between two different processes on the same or different machines. To be more precise, it's a way to talk to other computers using standard Unix file descriptors. In Unix, every I/O action is done by writing or reading a file descriptor.

Explain why you sniffing the network interface? Give examples?

Ans:
In order to capture the packets circulating in the network, for example for debugging a protocol.Packet sniffing is the practice of gathering, collecting, and logging some or all packets that pass through a computer network, regardless of how the packet is addressed. In this way, every packet, or a defined subset of packets, may be gathered for further analysis. You as a network administrators can use the collected data for a wide variety of purposes like monitoring bandwidth and traffic.

How many network interface usually you find in your pc?

Ans:
1. Wireless
2. Ethernet
3. Loopback

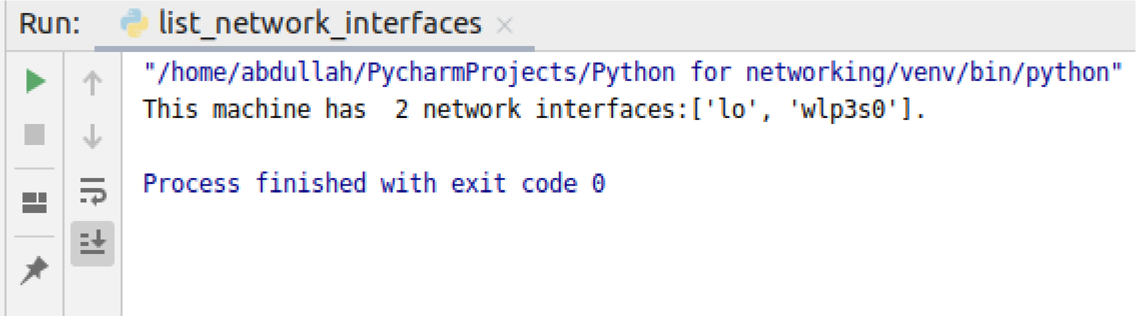
Program:

1.Enumerating interfaces on your machine

Ans:

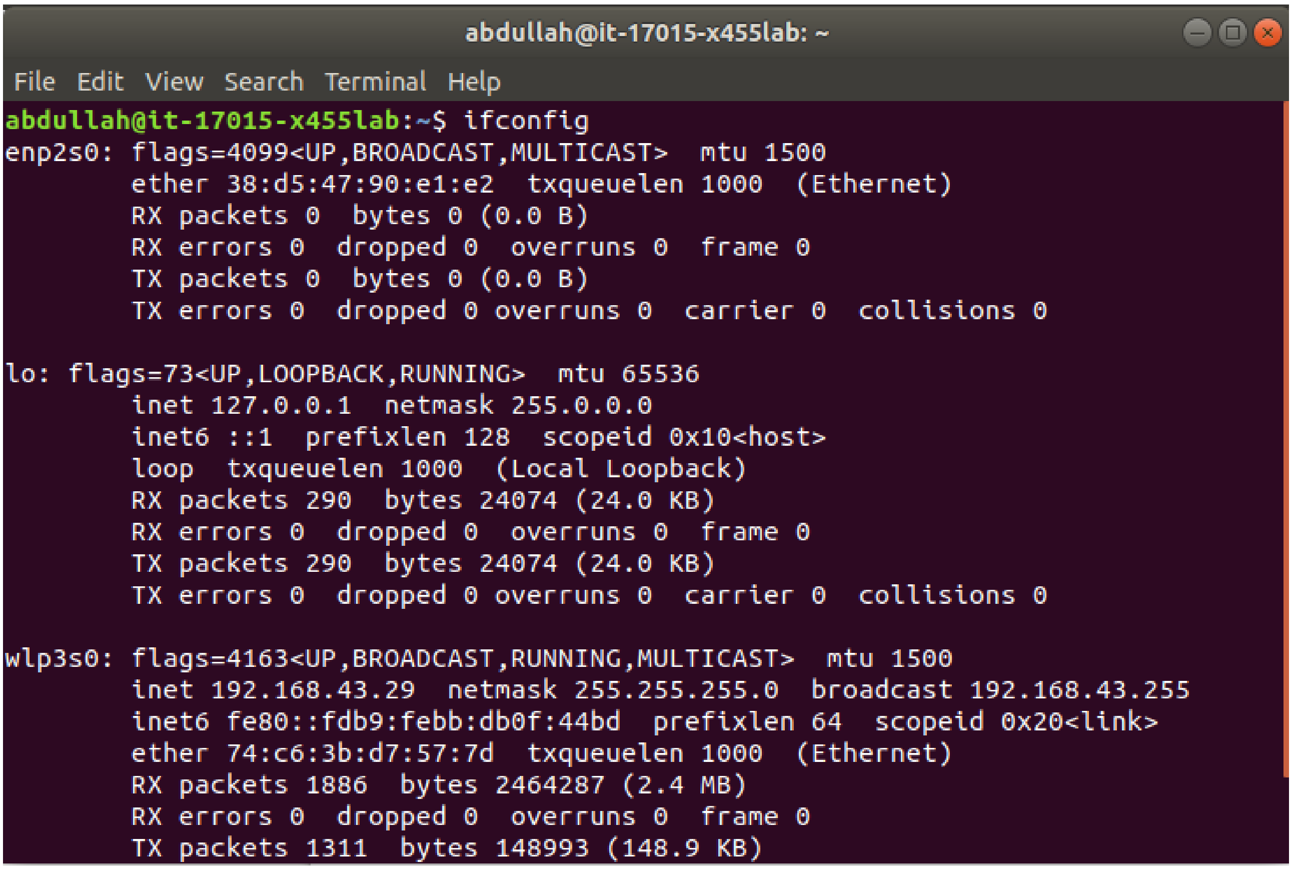
```
import sys
import socket
import fcntl
import struct
import array
SIOCGIFCONF = 0x8912
STUCT_SIZE_32 = 32
STUCT_SIZE_64 = 40
PLATFORM_32_MAX_NUMBER = 2**32
DEFAULT_INTERFACES = 8
def list_interfaces():
    interfaces = []
    max_interfaces = DEFAULT_INTERFACES
    is_64bits = sys.maxsize > PLATFORM_32_MAX_NUMBER
    struct_size = STUCT_SIZE_64 if is_64bits else STUCT_SIZE_32
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    while True:
       bytes = max_interfaces * struct_size
       interface_names = array.array('B', '\0' * bytes)
       sock_info = fcntl.ioctl(sock.fileno(),SIOCGIFCONF,struct.pack('iL',bytes,
interface_names.buffer_info()[0]) )
       outbytes = struct.unpack('iL', sock_info)[0]
       if outbytes == bytes:
            max_interfaces *= 2
       else:
         break
    namestr = interface_names.tostring()
    for i in range(0, outbytes, struct_size):
        interfaces.append((namestr[i:i+16].split('\0', 1)[0]))
    return interfaces
if __name__ == '__main__':
    interfaces = list_interfaces()
    print("This machine has  % s network interfaces:% s." %(len(interfaces), interfaces))
```

Output:



```
"/home/abdullah/PycharmProjects/Python for networking/venv/bin/python"
This machine has  2 network interfaces:['lo', 'wlp3s0'].

Process finished with exit code 0
```

Checking from the terminal:


abdullah@it-17015-x455lab: ~
File Edit View Search Terminal Help

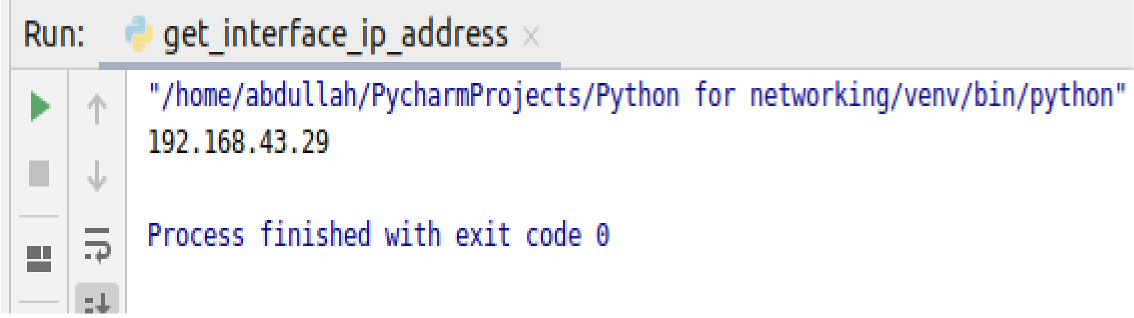
```
abdullah@it-17015-x455lab:~$ ifconfig
enp2s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 38:d5:47:90:e1:e2  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 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 290  bytes 24074 (24.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 290  bytes 24074 (24.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.29  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 fe80::fdb9:febb:db0f:44bd  prefixlen 64  scopeid 0x20<link>
        ether 74:c6:3b:d7:57:7d  txqueuelen 1000  (Ethernet)
        RX packets 1886  bytes 2464287 (2.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1311  bytes 148993 (148.9 KB)
```

2.Finding the IP address for a specific interface on your machine

Ans:

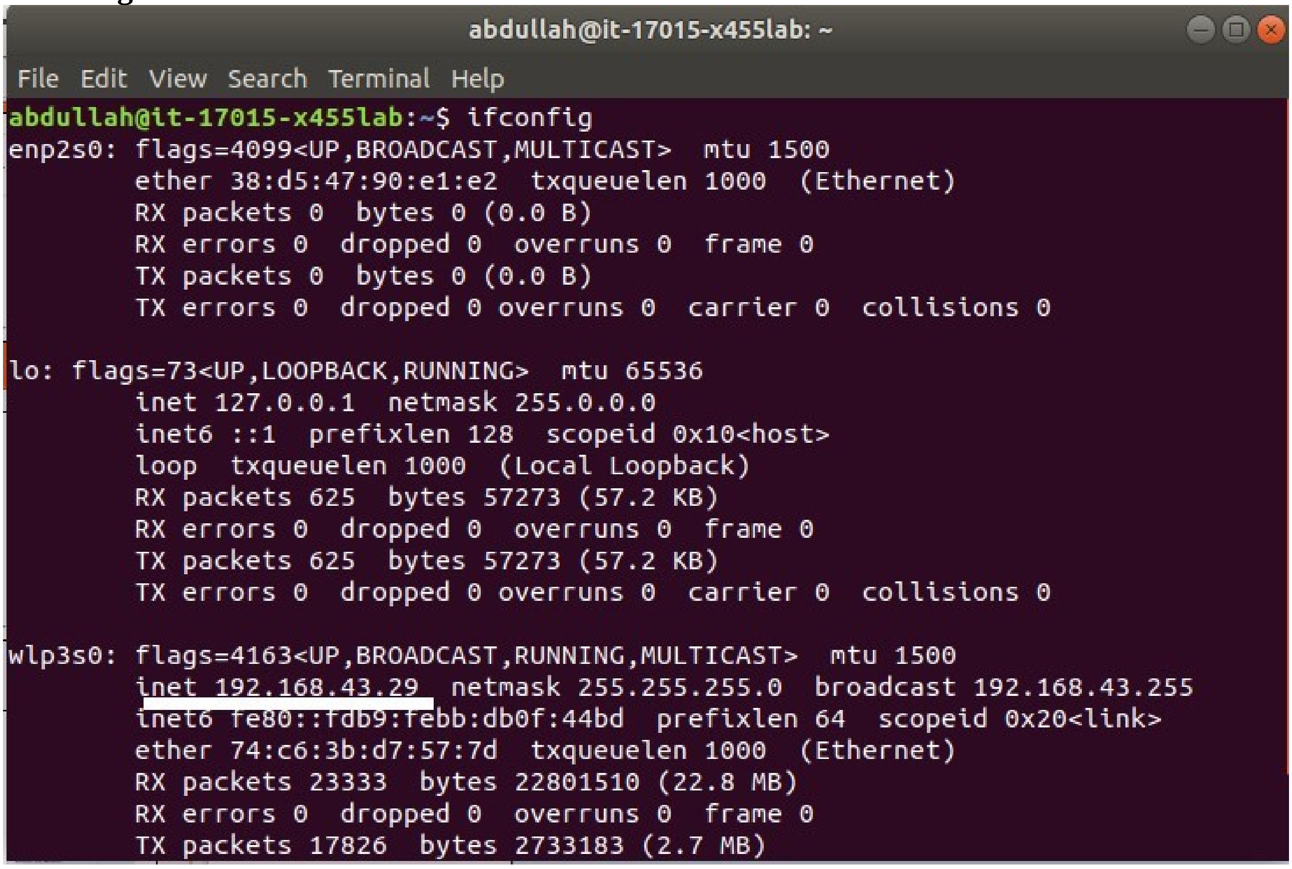
```python
import socket
import fcntl
import struct
def get_ip_address(ifname):
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    return socket.inet_ntoa(fcntl.ioctl(
        s.fileno(),
        0x8915,  # SIOCGIFADDR
        struct.pack('256s', ifname[:15])
    )[20:24])
print(get_ip_address('wlp3s0'))
```

Output:



```
"/home/abdullah/PycharmProjects/Python for networking/venv/bin/python"
192.168.43.29

Process finished with exit code 0
```

Checking from the terminal:


abdullah@it-17015-x455lab: ~
File Edit View Search Terminal Help

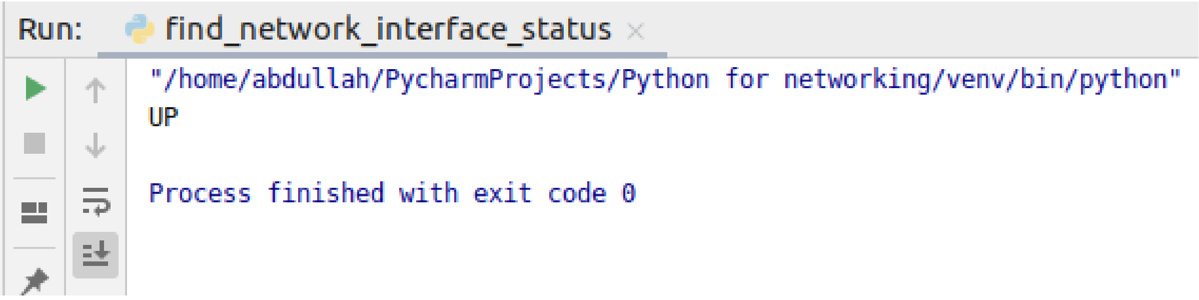
```
abdullah@it-17015-x455lab:~$ ifconfig
enp2s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 38:d5:47:90:e1:e2  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 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 625  bytes 57273 (57.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 625  bytes 57273 (57.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.29  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 fe80::fdb9:febb:db0f:44bd  prefixlen 64  scopeid 0x20<link>
        ether 74:c6:3b:d7:57:7d  txqueuelen 1000  (Ethernet)
        RX packets 23333  bytes 22801510 (22.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17826  bytes 2733183 (2.7 MB)
```

3.Finding whether an interface is up on your machine

Ans:
Here I give interface name ‘wlp3s0’

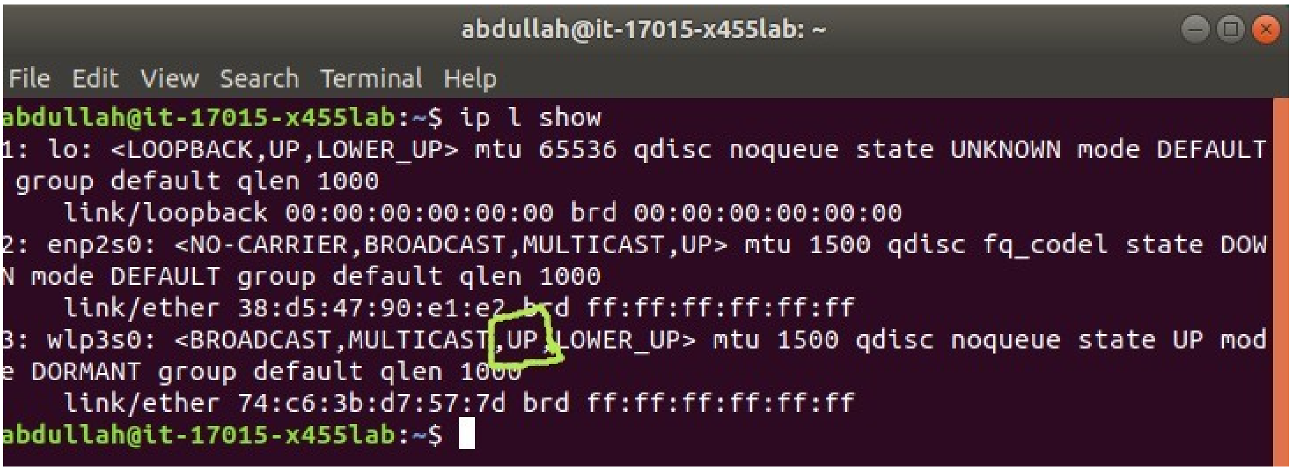
```
from pyroute2 import IPRoute
ip = IPRoute()
state = ip.get_links(ip.link_lookup(ifname='wlp3s0'))[0].get_attr('IFLA_OPERSTATE')
print(state)
ip.close()
```

Output:



```
"/home/abdullah/PycharmProjects/Python for networking/venv/bin/python"
UP

Process finished with exit code 0
```

Checking from the terminal:


abdullah@it-17015-x455lab: ~
File Edit View Search Terminal Help

```
abdullah@it-17015-x455lab:~$ ip l show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
 group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp2s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOW
N mode DEFAULT group default qlen 1000
    link/ether 38:d5:47:90:e1:e2 brd ff:ff:ff:ff:ff:ff
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mod
e DORMANT group default qlen 1000
    link/ether 74:c6:3b:d7:57:7d brd ff:ff:ff:ff:ff:ff
abdullah@it-17015-x455lab:~$ 
```

4.**Write a script that provides the interfaces, IP and status.**

Ans:

```
import sys
import socket
import fcntl
import struct
import array
import argparse
import nmap
SIOCGIFCONF = 0x8912 #from C library sockios.h
STUCT_SIZE_32 = 32
STUCT_SIZE_64 = 40
PLATFORM_32_MAX_NUMBER = 2**32
DEFAULT_INTERFACES = 8
SAMPLE_PORTS = '21-23'
def list_interfaces():
    interfaces = []
    max_interfaces = DEFAULT_INTERFACES
    is_64bits = sys.maxsize > PLATFORM_32_MAX_NUMBER
    struct_size = STUCT_SIZE_64 if is_64bits else STUCT_SIZE_32
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    while True:
        bytes = max_interfaces * struct_size
        interface_names = array.array('B', '\0' * bytes)
        sock_info = fcntl.ioctl(sock.fileno(), SIOCGIFCONF, struct.pack('iL',bytes,
interface_names.buffer_info()[0]))
        outbytes = struct.unpack('iL', sock_info)[0]
        if outbytes == bytes:
            max_interfaces *= 2
        else:
            break
    namestr = interface_names.tostring()
    for i in range(0, outbytes, struct_size):
        interfaces.append((namestr[i:i + 16].split('\0', 1)[0]))
    return interfaces
def get_ip_address(ifname):
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    return socket.inet_ntoa(fcntl.ioctl(s.fileno(), 0x8915,struct.pack('256s', ifname[:15]))[20:24])
def get_interface_status(ifname):
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    ip_address = socket.inet_ntoa(fcntl.ioctl(sock.fileno(), 0x8915,struct.pack('256s', ifname[:15]))
[20:24])
    nm = nmap.PortScanner()
    nm.scan(ip_address, SAMPLE_PORTS)
    return nm[ip_address].state()
if __name__ == '__main__':
    interfaces = list_interfaces()
    print("This machine has %s network interfaces: %s." % (len(interfaces),  interfaces))
    i = 0
    while i < len(interfaces):
        print("Interface [%s] --> IP: %s" % (interfaces[i],get_ip_address(interfaces[i])))
        print("Interface [%s] is: %s" % (interfaces[i], get_interface_status(interfaces[i])))
        i = i + 1
```

Discussion:

I can successfully run all the programs above and see my interfaces ,interfaces ip addresses,interfaces status.