

AN INTRODUCTION TO LINUX NETWORKING

In this article, we'll explore networking under GNU/Linux. You'll find it interesting to manage the entire network through certain valid keystrokes known as commands. Imagine that you have to access the contents of several other machines from a mount point in your machine. Then imagine shutting down, rebooting, and installing applications on those remote machines, all at one time? Could you configure the WLAN and LAN interfaces from the CLI? This tutorial gives you some insights to the exciting bytes on controlling your network under GNU/Linux.

First we will learn the 'Hello World' of a networked machine.

Let us ping!

ping is a universal command that is available on every operating system to test the reachability of a network. When you shoot your terminal with the ping command and an IP address as its argument, the machine will try to send some bits of raw data towards the machine owning that IP address. If some machine exists with that IP address, it will send back certain bits. Thus the machine receives the bits and it confirms that a path is available from the current machine to the other through a network. We can check the existence of certain machines on the network by just pinging.

To see which machines are up in the current network, let's write a bit of shell script.

Open vim as the root:

```
# vim /usr/bin/netup.sh
```

...and key in the following lines in it:

```
#!/bin/bash

for i in 192.168.1.{1..255}; // checks 192.168.1.x class of IPs.

do

ping -c2 $i > /dev/null;

[ $? -eq 0 ] && echo $i is up.

trap "exit" SIGINT // To force exit when Ctrl+C keystroke is applied.

done
```

Save the file, and make it executable by running the following command:

```
chmod a+x /usr/bin/netup.sh
```

Now, run the script as:

```
[slynux@gnuxbox ~]$ netup.sh

192.168.1.1 is up.

192.168.1.3 is up.

192.168.1.4 is up.
```

Configuring your network

Now, let us look at how to configure your machine on the network. You can configure it using two methods. It can be configured manually by the `ifconfig` command for static IP addressing or it can be done via the DHCP (Dynamic Host Control Protocol).

Static IP addressing is the one that you explicitly instruct the system to use by giving an IP address for a given Ethernet or wireless interface. In case you're using the DHCP, simply issuing the `dhclient` command will fetch the system an available IP address from the DHCP server in your network. Note that it may not be the same IP address that your machine fetches each time you issue `dhclient`.

Interface cards

Machines are networked either via network cables or using wireless protocols. LAN cards used for networking are known as Ethernet and wireless LAN (WLAN) cards. We interface the network via this outlet. In *nix platforms, Ethernet cards or WLAN cards are denoted as `eth0`, `eth1`, etc, or `wlan0`, `wlan1`, etc, respectively.

ifconfig

We have `ifconfig`, a.k.a the interface config, for setting up a network on the machine. To get information about the availability of interface devices available on the current machine, open a terminal and execute the following as the root:

```
[slynux@gnuxbox ~]# ifconfig -a

eth0    Link encap:Ethernet HWaddr 00:1C:23:FB:37:22

        inet6 addr: fe80::21c:23ff:febf:3722/64 Scope:Link

        UP BROADCAST MULTICAST MTU:1500 Metric:1
```

RX packets:9724 errors:0 dropped:0 overruns:0 frame:0

TX packets:2720 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:2400589 (2.2 MiB) TX bytes:645396 (630.2 KiB)

Interrupt:17

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:76320 errors:0 dropped:0 overruns:0 frame:0

TX packets:76320 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:87151068 (83.1 MiB) TX bytes:87151068 (83.1 MiB)

wlan0 Link encap:Ethernet HWaddr 00:1C:BF:87:25:D2

inet addr:192.168.1.143 Bcast:192.168.1.255 Mask:255.255.255.0

inet6 addr: fe80::21c:bfff:fe87:25d2/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:45302 errors:0 dropped:0 overruns:0 frame:0

```
TX packets:37510 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:31091293 (29.6 MiB) TX bytes:9734025 (9.2 MiB)
```

Here I have three interfaces — eth0, lo and wlan0, where:

- ♣ eth0 corresponds to the Ethernet card
- ♣ lo corresponds to a loopback device that points to the localhost network
- ♣ wlan0 corresponds to the wireless LAN card

Static IP addressing

For static IP addressing, issue the following command as the root:

```
ifconfig <device name> <ip address>
```

For example:

```
ifconfig eth0 192.168.0.2
```

ifconfig -a gives you details of all interface devices and configurations. In order to receive details of only one Ethernet device, execute ifconfig eth0. The following is an example output:

```
[root@gnubox slynux]# ifconfig eth0  
eth0  Link encap:Ethernet HWaddr 00:1C:23:FB:37:22
```

```
inet addr:192.168.0.2 Bcast:192.168.0.255 Mask:255.255.255.0

inet6 addr: fe80::21c:23ff:febf:3722/64 Scope:Link

UP BROADCAST MULTICAST MTU:1500 Metric:1

RX packets:9724 errors:0 dropped:0 overruns:0 frame:0

TX packets:2720 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:2400589 (2.2 MiB) TX bytes:645396 (630.2 KiB)

Interrupt:17
```

Now, let us go about setting the subnet mask. This is done easily with `ifconfig`, as follows:

```
ifconfig eth0 192.168.0.2 netmask 255.255.255.0
```

Now that the IP address and subnet mask is configured (or reconfigured), how do you get your Ethernet up (i.e., available) and down? The `ifup` and `ifdown` commands help you with that as follows:

```
ifup eth0

ifdown eth0
```

Source : <http://www.opensourceforu.com/2009/02/recipes-for-networking/>