

Lecture: Network configuration

Practice

- Manual Configuration
 - o Set the Hostname
 - edit the following file into a text editor:

```
#vi /etc/sysconfig/network
```

- Change the **hostname** value to the fully qualified name of your computer.

```
HOSTNAME=primarydns.centos.local
```

- Save and close; the new name will be applied after your next reboot.
- o Set a DHCP Address
 - Edit the first network interface, **eth0**:

```
#vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

- Modify the file to look similar to the example below:

```
DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
BOOTPROTO=dhcp
IPV4_FAILURE_FATAL=yes
NAME="System eth0"
```

- Save and close
- Restart the network service.

```
#service network restart
```

- o **Set a Static IP Address**
 - Edit the first network interface, **eth0**:
 - Modify the file to look similar to the example below:

```
DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
BOOTPROTO=none
IPADDR=192.168.225.132
PREFIX=24
GATEWAY=192.168.225.2
IPV4_FAILURE_FATAL=yes
NAME="System eth0"
```

- Save and close
- Restart the network service.

- Manipulate the routing table
 - o Display current routing table

```
# netstat -nr
```

```
or
# ip route list
```

- add a default gateway

```
# ip route add 192.168.225.0/24 dev eth0
```

- make routing changes persistent across server reboots
 - To set default gateway edit `/etc/sysconfig/network` as follows:

```
NETWORKING=yes
## server name ##
HOSTNAME=primarydns.centos.local
## Default route ##
GATEWAY=192.168.225.2
NETWORKING_IPV6=yes
IPV6_AUTOCONF=no
```

- Save and close
- Restart the networking service

```
# service network restart
# ip route list
```

- To verify new settings ping to the default gateway and external network:

```
# ping 192.168.225.2
# ping google.com
```

- check network connections (state, source/destination, addresses, bandwidth usage)
 - **ss** - display network connection statistics such as all TCP/UDP connections
 - established connection per protocol (e.g., display all established ssh connections)
 - display all the tcp sockets in various state such as ESTABLISHED or FIN-WAIT-1.
 - **netstat** - display network connections, routing tables, interfaces.
 - **tcptrack and iftop** - displays information about TCP connections it sees on a network interface and display bandwidth usage on an interface
- display currently Established, Closed, Orphaned and Waiting TCP sockets

```
# ss -s
```

```
Total: 529 (kernel 726)
TCP: 1403 (estab 286, closed 1099, orphaned 0, synrecv 0, timewait
1098/0), ports 774
Transport Total      IP      IPv6
*      726          -      -
RAW      0           0       0
UDP      27          13      14
TCP      304         298       6
INET     331         311      20
FRAG      0           0       0
```

```

or
# netstat -s
    - display all open network ports
# ss -l

or

# netstat -tulpn

    - display all TCP sockets

# ss -t -a

or

# netstat -nat

    - display all UDP sockets
# ss -u -a

or

# netstat -nau

    - view established connections only

# netstat -natu | grep 'ESTABLISHED'

```

Lecture: Change runlevels and shutdown or reboot system Practice

a) Shutting Down and Restarting the System

- the shutdown command allows you to schedule the power-down or reboot, to give users warning to save their work

```

    o Halt at 6pm:
# shutdown -h 18:00

```

```

    o Reboot thirty minutes from now:
# shutdown -r +30

```

- Setting the Default Runlevel: the system's default runlevel on bootup is configured in */etc/inittab*
 - o To configure a default runlevel of 3, */etc/inittab* should contain the line:

```
id:3:initdefault
```

- o There should be only one initdefault line in */etc/inittab*

- Determining the Current Runlevel

- o The runlevel command prints the system's previous and current runlevels:

```
$ /sbin/runlevel
```

```
3
```

- Switching Runlevel

- o init is responsible for controlling runlevels, so switching runlevels involves telling init to do something:

```
# telinit 1
```

- o to switch into a given runlevel you can alternatively use init itself, with the same syntax:

```
# init 5
```

- Starting or Stopping Individual Services

- o This will stop your network service, so you'll be out of your ssh terminal, if connected.

```
# /etc/init.d/network stop
```

- o Start the network service

```
# /etc/init.d/network start
```

- Restart the network service

```
# /etc/init.d/network restart
```

```
or
```

```
# /etc/init.d/network reload
```

Exercises

1.

- Look in */etc/init.d* or */etc/rc.d/init.d* to see what services can be started by init.
- Try running the script for *crond*, and use it to stop the cron service, and then start it up again.
- Take a quick look at the program in a text editor (it's a small shell script) to get a rough idea of what it does.
- Look in the *rc3.d* directory to see what services are killed and started when switching to runlevel 3.
- Use *telinit* to change to single-user mode.
- Once in single-user mode, use *top* to see what processes are left running.

2.

- Reboot the machine by changing to runlevel 6.
- When the GRUB prompt appears, type Tab to see a list of operating systems to boot. Type the name of the one you want followed by a space and the number 1, to indicate that you want to boot straight into single-user mode.
- Change back to runlevel 3.

Lecture: Jobs

Exercises:

1.

- Start a process by running **man bash** and suspend it with **Ctrl+Z**.

- b. Run **gedit** in the background, using **&**.
- c. Use **jobs** to list the backgrounded and stopped processes.
- d. Use the **fg** command to bring **man** into the foreground, and quit from it as normal.
- e. Use **fg** to foreground **gedit**, and terminate it with **Ctrl+C**.
- f. Run **gedit** again, but this time without **&**. It should be running in the foreground (so you can't use the shell). Try suspending it with **Ctrl+Z** and see what happens. To properly put it into the background, use **bg**.

Lecture: Processes

Practice:

Shortcuts for **TOP** command:

- a) Show Processes sorted by any Top output column – press **O**(upper-case)
By default top command displays the processes in the order of CPU usage. When the top command is running, press **M** (**upper-case**) to display processes sorted by memory usage.
Select sort field via field letter: ex: for CPU, **shift+p**, memory, **shift+m**, users, **shift+u**, **shift+r** (show reverse).
- b) Kill a task without exiting from Top – Press **k**
- c) Display all CPUs / Cores in the Top output – Press **l**
Top output by default shows CPU line for all the CPUs combined together
- d) Refresh Unix Top command output on demand (or) change refresh interval
By default, linux top command updates the output every 3.0 seconds. When you want to update the output on-demand, press **space bar**. To change the output update frequency, press **d** in interactive mode, and enter the time in seconds.
- e) Highlight running processes in the Linux Top command output – Press **z** or **b**
- f) Quit Top command after a specified number of iterations using **top -n**

\$ top -n 2

- g) Executing Unix Top command in Batch Mode

\$ top -b -n 1

This option is very helpful when you want to capture the unix top command output to a readable text file.

- h) Decrease number of Processes displayed in Top output – Press **n**
Press **n** in the Interactive mode, which prompts for a number (try 3).
- i) Save Top configuration settings – Press **W**
If you've made any interactive top command configurations suggested in the above examples, you might want to save those for all future top command output. Once you've saved the top configuration, next time when you invoke the top command all your saved top configuration options will be used automatically.

To save the top configuration, press **W**, which will write the configuration files to `~/toprc`.

Once you've saved the top configuration, next time when you invoke the top command all your saved top configuration options will be used automatically.

- Modify Process Execution Priorities
 - view the current priority of a processes

#ps -l

- look at the PRI column

- Niceness ranges from +19 (very nice) to -20 (not very nice)
- run a command at increased niceness (lower priority):

#nice -10 long-running-command &

- run a command at decreased niceness (higher priority):

#nice -n -15 important-command &

- renice changes the niceness of existing processes
- example: set the process with pid 2984 to the maximum niceness (lower priority):

\$ renice 20 2984

- set the process with pid 3598 to a lower niceness (higher priority):

\$ renice -15 3598

- You can also change the niceness of all user's processes:

\$ renice 15 -u user1

Exercises:

1.

- a. Use **top** to show the processes running on your machine.
- b. Make top sort by memory usage, so that the most memory-hungry processes appear at the top.
- c. Restrict the display to show only processes owned by you.
- d. Try killing one of your processes – make sure it's nothing important :).
- e. Display a list of all the processes running on the machine using **ps** (displaying the full command line for them).
- f. Get the same listing as a tree, using both **ps** and **pstree**.
- g. Have **ps** sort the output by system time used.

2.

a. Create the following shell script, called **forever**, in your home directory:

```
#!/bin/sh
while [ 1 ]; do
echo hello... >/dev/null;
done
```

Make it executable and run it in the background as follows:

```
$ chmod a+rx forever
$ ./forever &
```

b. Use **ps -l** to check the script's nice level

c. Run the script with **nice** and give it a niceness of 15. Try running it alongside a less nice version, and see what the difference is in top

d. Try using **nice** or **renice** to make a process niceness less than 0