

## LAB SESSION- BOOTING UP LINUX!

The booting process was explained in the previous lab sessions. You already know that the bootloader is a small program or a set of instructions which loads the OS. Bootloading is a two stage process. The first stage is also called the **IPL, or the Initial Program Loader**. The sole purpose of the IPL is to load the second stage. This second stage is where most of the functionality of a bootloader resides, and is of prime importance.

There are two bootloaders which are available for Linux:

1. GRUB (Grand Unified Bootloader)
2. LiLo (Linux Loader)

While installing Linux, depending on the flavour, you might get a choice as to which of the above two bootloaders to install. Though both will do the job equally well. However, a comparison is provided:

1. LiLo has no interactive command interface, whereas GRUB does.
2. LiLo does not support booting from a network, whereas GRUB does.
3. LiLo stores information regarding the location of the operating systems it can to load physically on the MBR. If you change your LILO config file, you have to rewrite the LILO stage one boot loader to the MBR. Compared with GRUB, this is a much more risky option since a misconfigured MBR could leave the system unbootable. With GRUB, if the configuration file is configured incorrectly, it will simply default to the GRUB command-line interface.

### Configuring GRUB

The GRUB can be modified/customised/configured by editing the grub.cfg file.

The path for grub.cfg is:

`/boot/grub/grub.cfg`

Following is a screen capture of the grub file of Red Hat Linux.

```
default=0
timeout=10
splashimage=(hd1,2)/grub/splash.xpm.gz
password --md5 $1$opeVt0$Y.br.18LyAasRsGdSKLY1p1
title Red Hat Linux
    password --md5 $1$opeVt0$Y.br.18LyAasRsGdSKLY1p1
    root (hd1,2)
    kernel /vmlinuz-2.4.18-14 ro root=LABEL=/
    initrd /initrd-2.4.18-14.img
title Windows XP
    password --md5 $1$opeVt0$Y.br.18LyAasRsGdSKLY1p1
    rootnoverify (hd0,0)
    chainloader +1
```

- The `default=` option signals to GRUB which image to boot from by default after the timeout period. This relates to one of the images in the grub.conf file. 0 is the first specified, 1 is the second specified, etc. If you don't specify this option in the configuration file, it will boot the first image specified in the file.

- `timeout=` is the number of seconds the boot prompt will wait before automatically loading the default OS, in this case, Red Hat Linux.
- `splashimage=` is the location of the image to be used as the background for the GRUB GUI.
- The `password` option specifies the MD5-encrypted password used to gain access to GRUB's interactive boot options. Note this does not stop users loading your defined OS choices; this needs to be set on a per-title basis. To generate an md5 password, run the tool `grub-md5-crypt` (as root), which comes with GRUB. It will prompt you for the password you want to encrypt. It then will output the MD5-encrypted password. Copy this into your `grub.conf` after `password -md5` but on the same line. Usually this password can be set to the root password, since it is only root who can read the `grub.conf` file anyway.
- `title` identifies the specific OS that will be booted from at the user interface at runtime. Unlike with LILO, you can include spaces in this name.
- The `root` option tells GRUB where the OS file system actually lives. Grub references this disk as `(hd1,2)`, again the third partition of the second disk (disk 0 being the first disk, partition 0 being the first partition).
- `kernel: vmlinuz-X.X.XX-XX` is the name of the default boot kernel image within your root directory.
- `initrd: initrd-X.X.XX-XX.img` is the name of the default initrd file within your root directory.
- `chainloader +1` tells GRUB to use a chain loader to load this OS, which is required for loading Windows.

*Point of Interest:* setting the `timeout=-1` will make the GRUB wait for input from user, i.e. It will not boot the default OS.

## More about the booting process: Runlevels

Linux provided you with different runlevels. Each runlevel provides you with a different environment, and it's set of privileges. The list of runlevels, directory and explanation available with Ubuntu is as below:

0	<code>rc0.d</code>	Halts (shuts down) the system
1	<code>rc1.d</code>	Single-user mode (limited capabilities)
2	<code>rc2.d</code>	Multiuser mode with graphical login (full operation mode, X server initialised)
3	<code>rc3.d</code>	User-defined
4	<code>rc4.d</code>	User-defined
5	<code>rc5.d</code>	User-defined
6	<code>rc6.d</code>	Reboots system
S	<code>rcS.d</code>	Single-user mode

The default runlevel is 2.

The **single user mode** is used to drop to linux shell prompt, and perform root operations.

The runlevels 3, 4 and 5 can be custom defined by the user, and can be used to provide varied user privileges!

The runlevels can be changed by using the **'telinit #'** command, where # signifies the runlevel you want to change to.

*Point of interest:* You can boot into single user mode or the runlevel s, and change root password if you forgot it! So, it's always advisable to setup a GRUB password also, so that no one can do the step just described above. Of course, if you forgot your GRUB password, then you're a very forgetful person!

## Changing the root password!

1. When the boot screen comes up and asks you for the input, select the kernel which you want and press 'e'. 'e' stands for editing the input parameters for the kernel.
2. Append single at the end of the line and then press 'b'. 'b' stand for booting the parameters just passed to the GRUB. The changes are one time only and not saved to the grub.cfg file

Alternatively, for UBUNTU LINUX

Select the recovery module kernel and then select root shell.

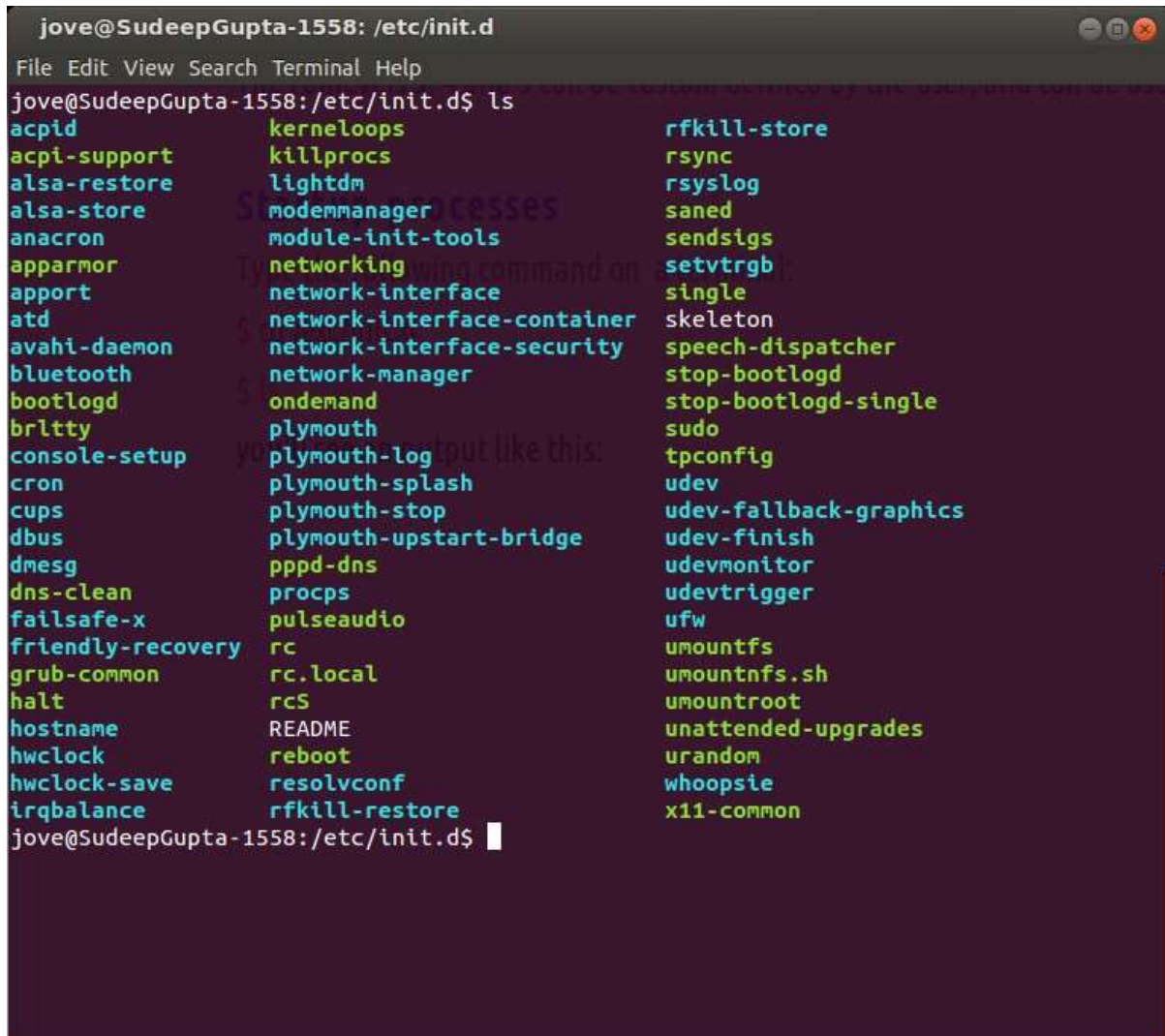
3. Type the commands.  
`$ passwd` //for changing root password  
`$ reboot` //for rebooting the system

## Startup processes

Type the following command on a terminal:

```
$ cd /etc/init.d  
$ ls
```

you'll see an output like this:



```
jove@SudeepGupta-1558: /etc/init.d  
File Edit View Search Terminal Help  
jove@SudeepGupta-1558:/etc/init.d$ ls  
acpid kerneloops rfkill-store  
acpi-support killprocs rsync  
alsa-restore lightdm rsyslog  
alsa-store modemmanager saned  
anacron module-init-tools sendsigs  
apparmor networking setvtrgb  
apport network-interface single  
atd network-interface-container skeleton  
avahi-daemon network-interface-security speech-dispatcher  
bluetooth network-manager stop-bootlogd  
bootlogd ondemand stop-bootlogd-single  
brltty plymouth sudo  
console-setup plymouth-log tpconfig  
cron plymouth-splash udev  
cups plymouth-stop udev-fallback-graphics  
dbus plymouth-upstart-bridge udev-finish  
dmesg pppd-dns udevmonitor  
dns-clean procps udevtrigger  
failsafe-x pulseaudio ufw  
friendly-recovery rc umountfs  
grub-common rc.local umountnfs.sh  
halt rcS umountroot  
hostname README unattended-upgrades  
hwclock reboot urandom  
hwclock-save resolvconf whoopsie  
irqbalance rfkill-restore x11-common  
jove@SudeepGupta-1558:/etc/init.d$
```

Service such as httpd or ssh can be started by typing the following commands:

```
$ sudo /etc/init.d/httpd start Or $ sudo /etc/init.d/ssh start
```

## Shutting down the System

The system can be shutdown from the terminal by giving the command:

```
$ sudo shutdown -h now
```

restarted by command:

```
$ sudo shutdown -r now
```

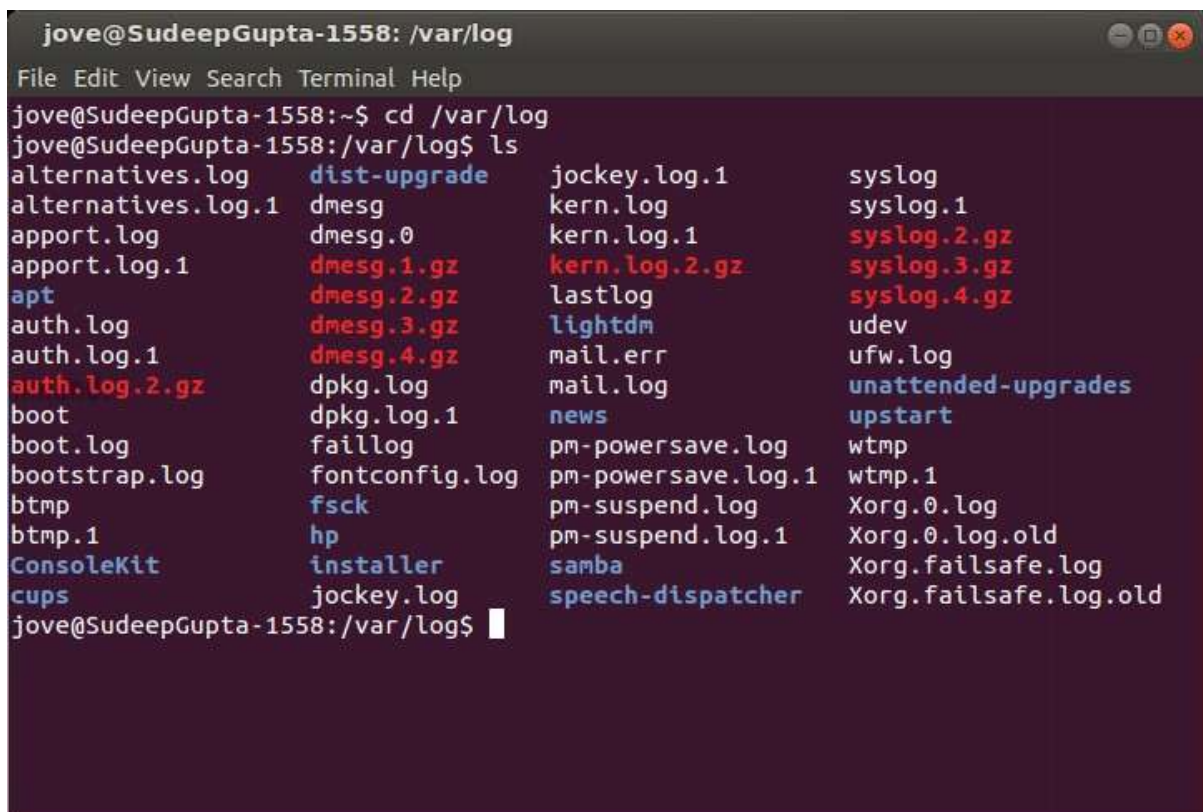
## Kernel

To know the version of the kernel that you're running, type the command:

```
$ uname -r
```

## Troubleshooting Kernel and System Changes

All logs are kept in the /var/log directory.



```
jove@SudeepGupta-1558: /var/log
File Edit View Search Terminal Help
jove@SudeepGupta-1558:~$ cd /var/log
jove@SudeepGupta-1558:/var/log$ ls
alternatives.log      dist-upgrade          jockey.log.1         syslog
alternatives.log.1   dmesg                 kern.log             syslog.1
appport.log          dmesg.0              kern.log.1           syslog.2.gz
appport.log.1        dmesg.1.gz           kern.log.2.gz       syslog.3.gz
apt                  dmesg.2.gz          lastlog              syslog.4.gz
auth.log             dmesg.3.gz          lightdm              udev
auth.log.1          dmesg.4.gz          mail.err             ufw.log
auth.log.2.gz       dpkg.log             mail.log             unattended-upgrades
boot                 dpkg.log.1          news                 upstart
boot.log            faillog              pm-powersave.log    wtmp
bootstrap.log       fontconfig.log       pm-powersave.log.1  wtmp.1
btm                  fsck                  pm-suspend.log      Xorg.0.log
btm.1               hp                    pm-suspend.log.1   Xorg.0.log.old
ConsoleKit          installer            samba                 Xorg.failsafe.log
cups                jockey.log           speech-dispatcher    Xorg.failsafe.log.old
jove@SudeepGupta-1558:/var/log$
```

A kernel ring buffer contains debug messages found during bootup. It stores the logs in a ring buffer which is maintained like the data structure 'circular queue'.

The kernel ring buffer information is stored in the file **dmesg**. Type the command to see the kernel log:

```
$ cat dmesg
```

the kernel log is displayed to you.

To search for

'usb' reports            `$ dmesg | grep -i usb`

'memory' reports        `$ dmesg | grep -i memory`

## Creating bootable USB sticks

There are two softwares which are specially available for creating bootable USB sticks, i.e. You no longer need a CD/DVD to install the LINUX OS.

1. Unetbootin
2. Universal USB installer

Depending on your preference, you can use any of the above tools to create a bootable USB stick. Both will do the job equally well. As such the official tool recommended for Ubuntu is Unetbootin.

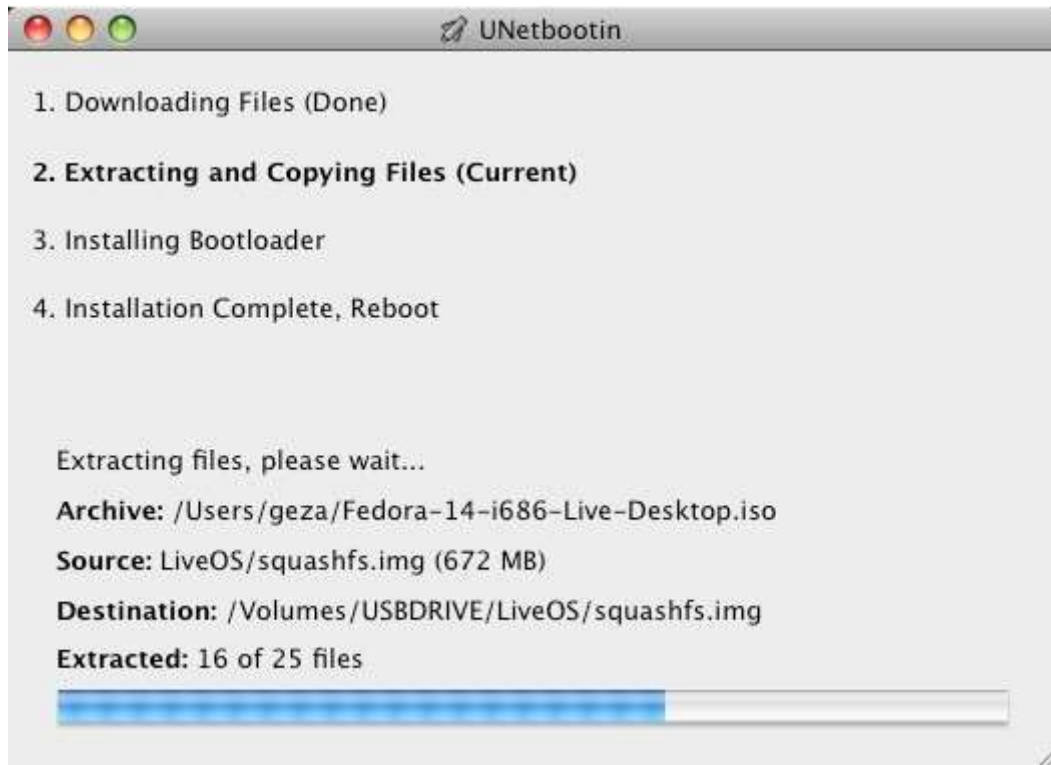
Steps:

1. download the iso image from the Ubuntu website. <http://www.ubuntu.com/>
2. download Unetbootin from <http://unetbootin.sourceforge.net/>
3. run unetbootin
4. you'll be greeted with the unetbootin GUI. Select the option of 'Diskimage', and give the path to the ISO file of Ubuntu that you downloaded.



5. Select the type as USB install, and give the drive letter for the USB stick.

6. Select 'OK' when you're ready to create the bootable stick.



7. Restart the system, and select the USB storage device as your booting device.
  8. The system will now boot using the USB stick!
-