

Ubuntu Booting Process

Bok, JongSoon
javaexpert@nate.com
<https://github.com/swacademy/Ubuntu>

Booting Process

1. Pre-BIOS
2. BIOS
3. MBR
4. Boot loader
5. Kernel
6. Init Process
7. Runlevel



PC Booting Process

The diagram consists of two red square brackets. The first bracket is positioned to the right of the first three items of the list (Pre-BIOS, BIOS, MBR) and is labeled 'PC Booting Process'. The second bracket is positioned to the right of the remaining four items (Boot loader, Kernel, Init Process, Runlevel) and is labeled 'Linux Booting Process'.

Linux Booting Process

For details :

- <http://www.slashroot.in/linux-booting-process-step-step-tutorial-understanding-linux-boot-sequence>
- <http://www.golinuxhub.com/2014/03/step-by-step-linux-boot-process.html>
- <http://www.thegeekstuff.com/2011/02/linux-boot-process/>
- <http://www.slashroot.in/what-mbr-and-gpt-partition-table>

Pre-BIOS Phase

■ SMPS

- Switching Mode Power Supply
- Provide the perfect required voltage level to the motherboard and other computer components.
- Converts AC to DC.
- Checks the voltage level's its providing to the motherboard.
- If the power signal level is perfect, will send a ***POWER GOOD*** signal to the motherboard timer.
- Then, the motherboard timer will stop sending reset signal to the CPU.
- Then, the power level is good and the computer can boot.

Pre-BIOS Phase (Cont.)

■ Bootstrapping

- The CPU knows where to search for instructions.
- This is how the computer will come to know where the *BIOS* program is located.
- The address location
 - Is located in the ROM.
 - Is almost always constant in X86 based computers.
 - Is FFFF:0000h.
 - Is the last region of the ROM.
 - Contains one instruction to jump to another memory address location.
 - This *JUMP* command, will tell the location of the BIOS program in the ROM.

BIOS

- Stands for Basic Input Output System.
- Performs some system integrity checks - *POST*
- POST
 - Stands for Power on Self Test.
 - Timer IC's
 - DMA controllers
 - CPU
 - Video ROM
 - Motherboard
 - Keyboard
 - Printer port
 - Hard Drive etc.

BIOS (Cont.)

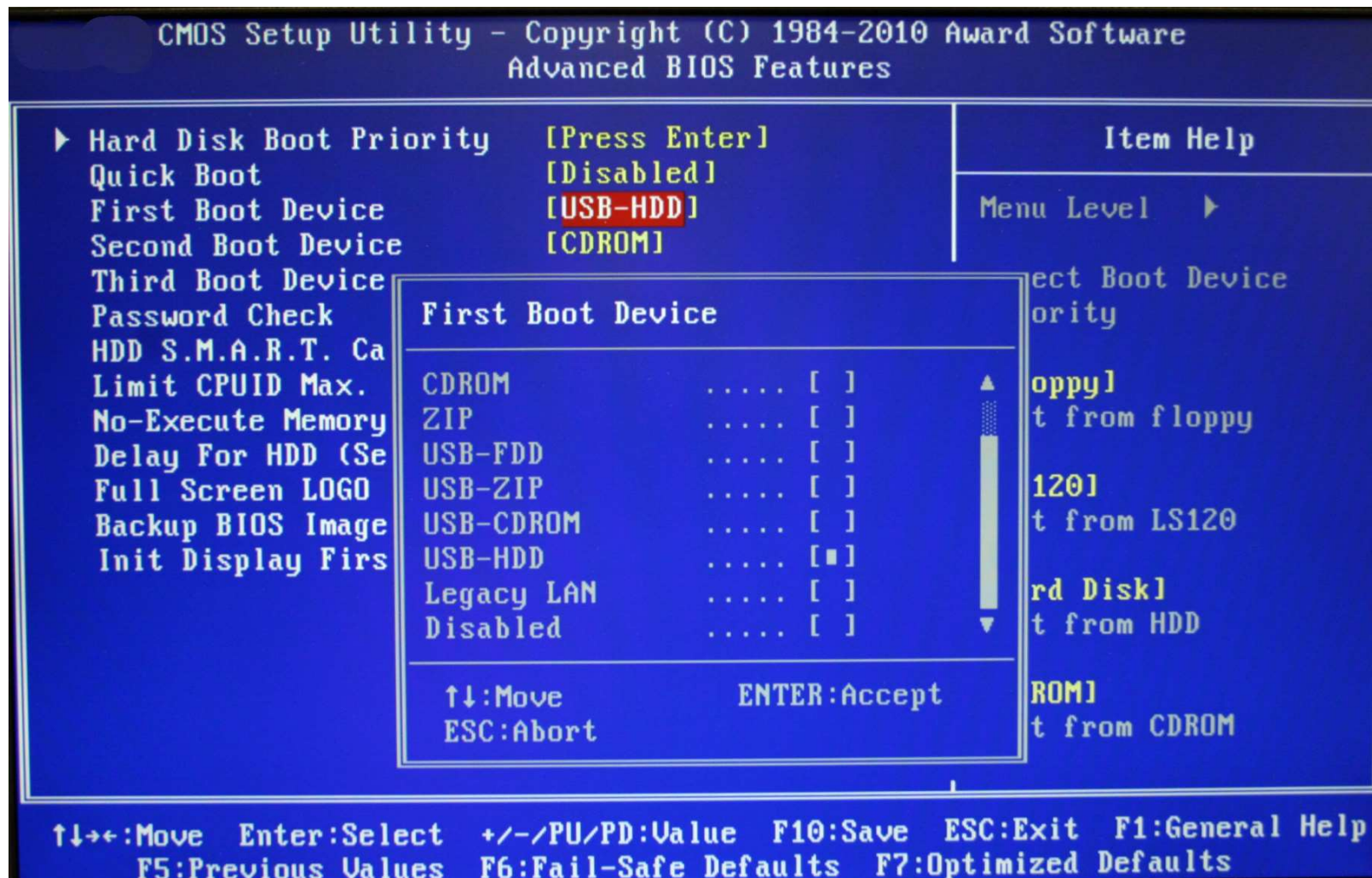
- Once the POST check is completed successfully, will look *CMOS* settings to know what is the boot order.
- Boot order tells where to look for the operating system.
 - CD ROM
 - HARD DISK
 - USB
 - Floppy DISK



BIOS (Cont.)

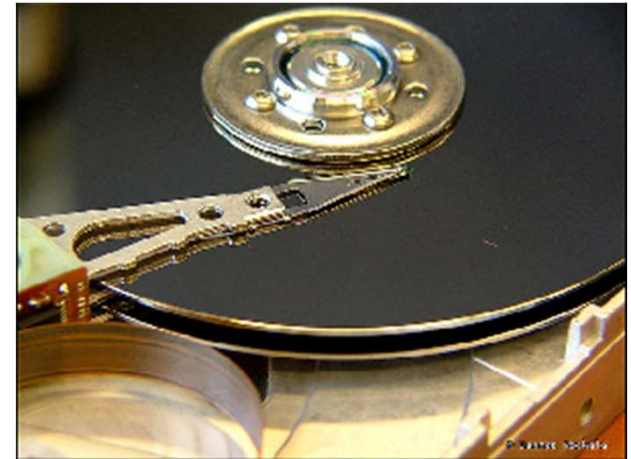
- Searches, loads, and executes the boot loader program.
- You can press a key (typically *F12* or *F2*, but it depends on your system) during the BIOS startup to change the boot sequence.
- Once the boot loader program is detected and loaded into the memory, gives the control to it.
- So, in simple terms BIOS loads and executes the *MBR* boot loader.

BIOS (Cont.)



MBR

- Stands for Master Boot Record.
- It is located in the 1st sector of the bootable disk. Typically `/dev/hda`, or `/dev/sda`
- Is less than 512 bytes in size.
- Has three components :
 - Primary boot loader info in 1st 446 bytes
 - Partition table info in next 64 bytes
 - MBR validation check in last 2 bytes.
- Contains information about *GRUB* (or *LILO* in old systems).
- So, in simple terms MBR loads and executes the GRUB boot loader.



MBR (Cont.)

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000000000	FA	33	C0	8E	D0	BC	00	7C	8B	F4	50	07	50	1F	FE	FC	ú3Ä!D% !ôP P ûü
0000000010	BF	00	06	B9	00	01	F2	A5	EA	1D	06	00	00	BE	BE	07	¿ ¹ òwê %%
0000000020	B3	04	80	3C	80	74	0E	80	3C	00	75	1C	83	C6	10	FE	³ [< t ! < u !Æ þ
0000000030	CB	75	EF	CD	18	8B	14	8B	4C	02	8B	EE	83	C6	10	FE	Euif ! !L !i!Æ þ
0000000040	CB	74	1A	80	3C	00	74	F4	BE	8B	06	AC	3C	00	74	0B	Ët ! < tó% ! ~ < t
0000000050	56	BB	07	00	B4	0E	CD	10	5E	EB	F0	EB	FE	BF	05	00	V» ' í ^ëšëþí
0000000060	BB	00	7C	B8	01	02	57	CD	13	5F	73	0C	33	C0	CD	13	>> , Wí _s 3Äí
0000000070	4F	75	ED	BE	A3	06	EB	D3	BE	C2	06	BF	FE	7D	81	3D	Oui%ê ëO%Ä ¿þ}! =
0000000080	55	AA	75	C7	8B	F5	EA	00	7C	00	00	49	6E	76	61	6C	U³uÇ!Šë Inval
0000000090	69	64	20	70	61	72	74	69	74	69	6F	6E	20	74	61	62	id partition tab
00000000A0	6C	65	00	45	72	72	6F	72	20	6C	6F	61	64	69	6E	67	le E=co loading
00000000B0	20	6F	70	65	72	61	74	69	6E	67	20	73	79	73	74	65	operating syste
00000000C0	6D	00	4D	69	73	73	69	6E	67	20	6F	70	65	72	61	74	m Missing operat
00000000D0	69	6E	67	20	73	79	73	74	65	6D	00	00	81	22	52	13	ing system !"R
00000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000001B0	00	00	00	00	00	00	00	00	B0	6B	97	42	00	00	80	01	°k!B !
00000001C0	01	00	07	FE	FF	FE	3F	00	00	00	9A	E5	3F	01	00	00	þýþ? !ä?
00000001D0	C1	FE	07	FE	FF	FE	D9	E5	3F	01	9A	24	40	01	00	00	Áþ þýþÜä? !s@
00000001E0	C1	FE	07	FE	FF	FE	73	0A	80	02	4E	80	21	10	00	00	Áþ þýþs ! N!!
00000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA	U³

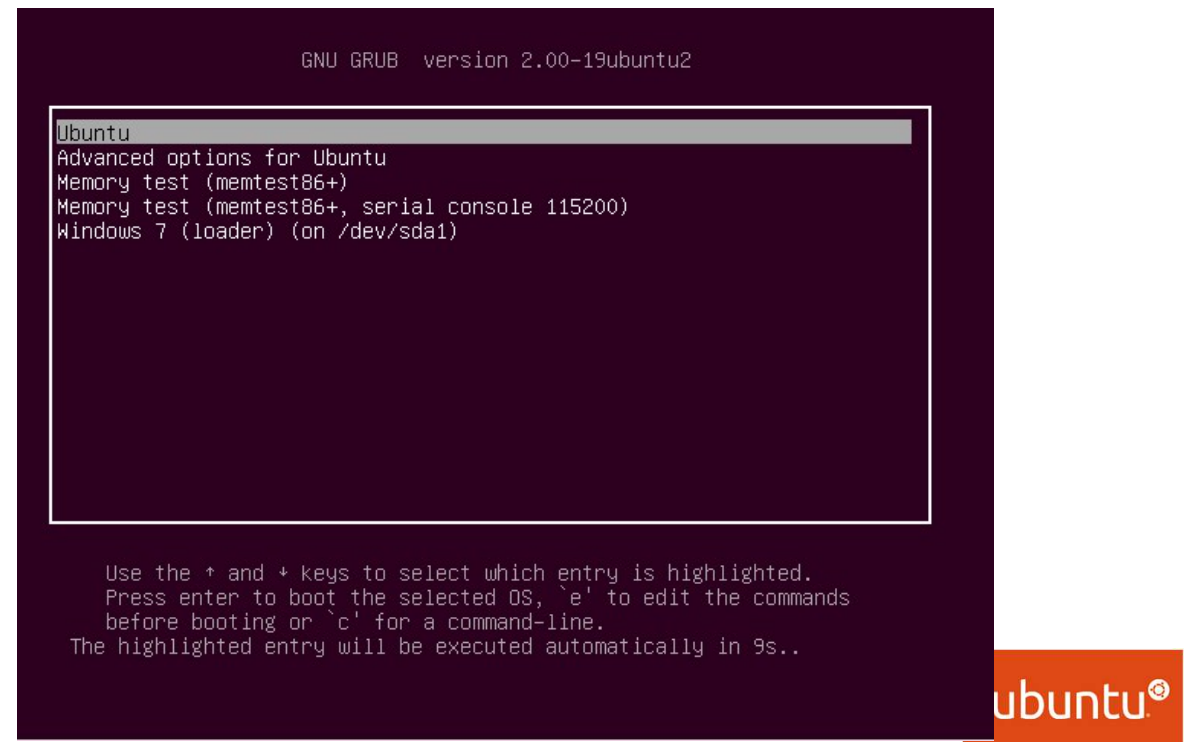
Partition-IDs (Dateisystem)	Partition-Startsektor: 00 00 00 3F	Partition-Größe: 01 3F E5 9A	Active Partition (00 - nicht aktiv)
--------------------------------	---------------------------------------	---------------------------------	--

Partitionstabelle	<div style="display: flex; align-items: center;"> <div style="width: 20px; height: 20px; background-color: yellow; border: 1px solid black; margin-right: 5px;"></div> <div style="width: 20px; height: 20px; background-color: green; border: 1px solid black; margin-right: 5px;"></div> <div style="width: 20px; height: 20px; background-color: red; border: 1px solid black; margin-right: 5px;"></div> <div style="width: 20px; height: 20px; background-color: blue; border: 1px solid black; margin-right: 5px;"></div> <div style="width: 20px; height: 20px; background-color: gray; border: 1px solid black;"></div> </div>	Bootprogramm 446 Byte Partition 1 16 Byte Partition 2 16 Byte Partition 3 16 Byte Partition 4 16 Byte MBR-Signatur 2 Byte 512 Byte
-------------------	--	--

Image from : <http://datenrettung-hannover.com/2015/10/20/mbr-was-ist-das/>
 For details : <http://cappleblog.co.kr/590>

GRUB

- Stands for Grand Unified Bootloader.
- If you have multiple kernel images installed on your system, you can choose which one to be executed.



GRUB (Cont.)

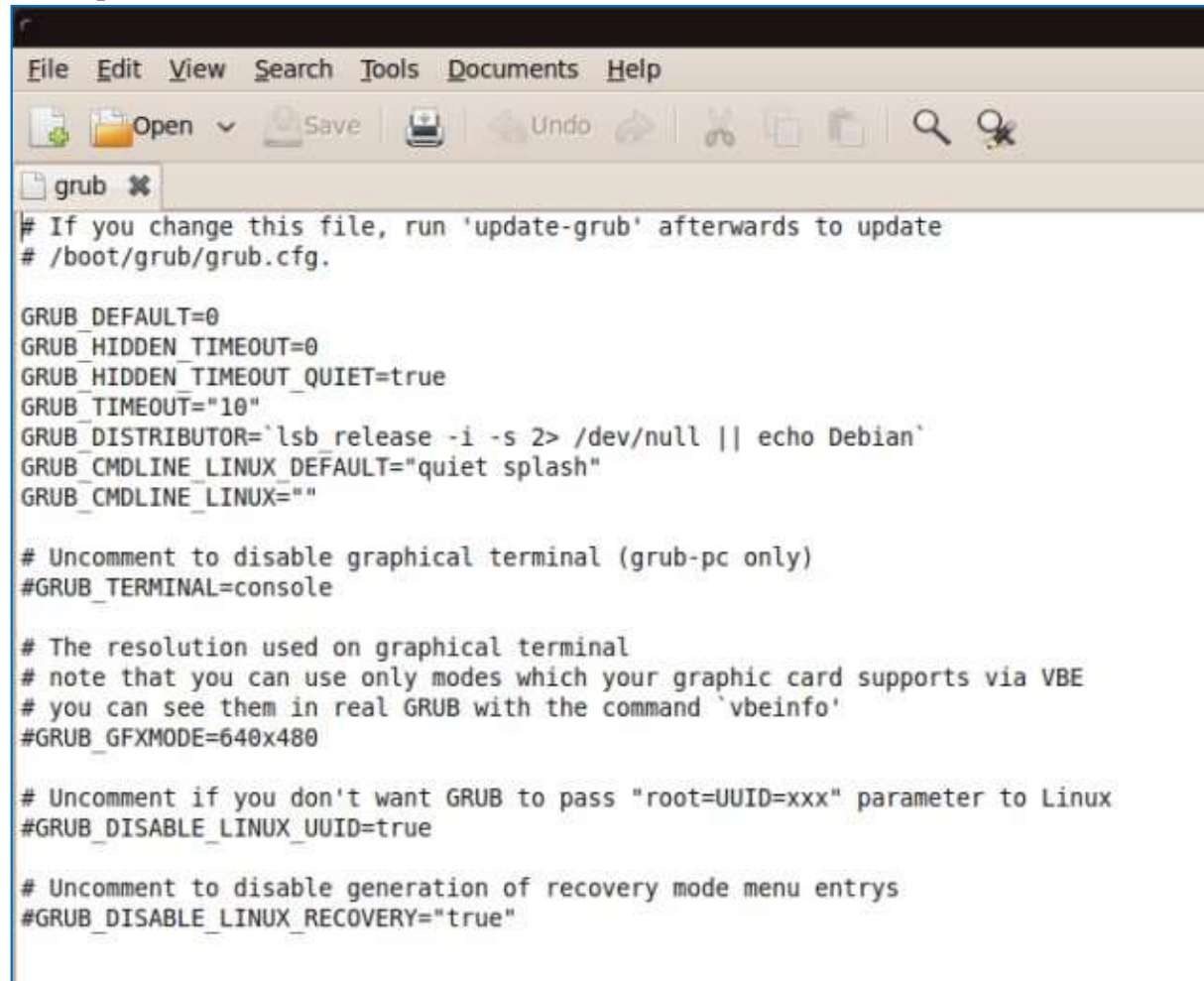
- Displays a splash screen, waits for few seconds.
 - If don't enter anything, loads the default kernel image as specified in the grub configuration file.
 - Has the knowledge of the filesystem.
- Cf.* The older Linux loader *LILO* didn't understand filesystem.
- Grub configuration file is **/boot/grub/grub.conf** in CentOS or **/etc/default/grub** in Ubuntu.

For more info :

- <http://www.golinuxhub.com/2014/03/what-is-grub-boot-loader.html>
- <http://www.howtogeek.com/196655/how-to-configure-the-grub2-boot-loaders-settings/>
- <http://www.dedoimedo.com/computers/grub-2.html>



GRUB (Cont.)

A screenshot of a text editor window with a menu bar (File, Edit, View, Search, Tools, Documents, Help) and a toolbar with icons for Open, Save, Undo, and others. The file name 'grub' is shown in the title bar. The content of the file is a GRUB configuration script for Debian, starting with a comment about updating the file and listing various configuration options like GRUB_DEFAULT, GRUB_HIDDEN_TIMEOUT, and GRUB_CMDLINE_LINUX.

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.

GRUB_DEFAULT=0
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT="10"
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

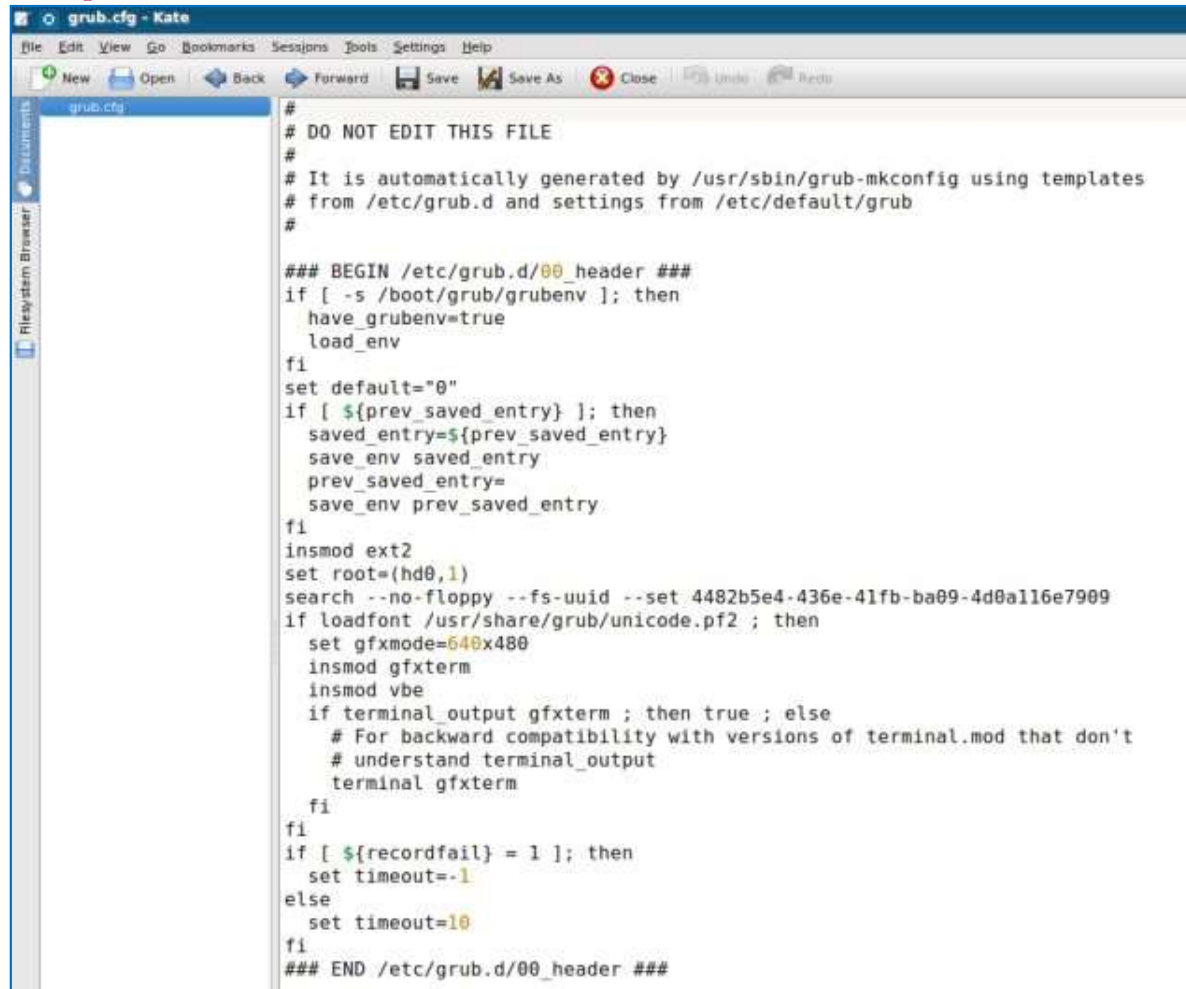
# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_LINUX_RECOVERY="true"
```

Source from : <http://www.dedoimedo.com/computers/grub-2.html>

GRUB (Cont.)



```
# DO NOT EDIT THIS FILE
#
# It is automatically generated by /usr/sbin/grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#
### BEGIN /etc/grub.d/00_header ###
if [ -s /boot/grub/grubenv ]; then
    have_grubenv=true
    load_env
fi
set default="0"
if [ ${prev_saved_entry} ]; then
    saved_entry=${prev_saved_entry}
    save_env saved_entry
    prev_saved_entry=
    save_env prev_saved_entry
fi
insmod ext2
set root=(hd0,1)
search --no-floppy --fs-uuid --set 4482b5e4-436e-41fb-ba09-4d0a116e7909
if loadfont /usr/share/grub/unicode.pf2 ; then
    set gfxmode=640x480
    insmod gfxterm
    insmod vbe
    if terminal_output gfxterm ; then true ; else
        # For backward compatibility with versions of terminal.mod that don't
        # understand terminal_output
        terminal gfxterm
    fi
fi
if [ ${recordfail} = 1 ]; then
    set timeout=-1
else
    set timeout=10
fi
### END /etc/grub.d/00_header ###
```

Source from : <http://www.dedoimedo.com/computers/grub-2.html>

GRUB (Cont.)

- Has 3 stages
 - Stage 1 in the MBR
 - Stage 1.5 in the remainder of the first cylinder of the disk
 - Stage 2 within in file on the disk
- Contains kernel and *initrd* image.
- So, in simple terms GRUB just loads and executes *Kernel* and *initrd* images.

```
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-194.el5PAE)
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.18-194.el5PAE ro root=LABEL=/
    initrd /boot/initrd-2.6.18-194.el5PAE.img
```

```
instructor@Ubuntu1404:/boot$
instructor@Ubuntu1404:/boot$ ls vm*
vmlinuz-4.2.0-27-generic  vmlinuz-4.2.0-36-generic
vmlinuz-4.2.0-35-generic
instructor@Ubuntu1404:/boot$
```

Lab 1 : Modifying GRUB Config.

1. GRUB_TIMEOUT
2. GRUB_HIDDEN_TIMEOUT

Kernel

- The bootloader starts the kernel running.
- Configures hardware and memory allocated to the system.
- *initrd* stands for Initial RAM Disk.
- *initrd* is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted.
- Contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

Kernel (Cont.)

```
grub.cfg x
142     if [ x${feature_platform_search_hint} = xy ]; then
143         search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos2
        --hint-efi=hd0,msdos2 --hint-baremetal=ahci0,msdos2 df8eaa26-97a2-4272-
        be63-5dcb90e782e0
144     else
145         search --no-floppy --fs-uuid --set=root df8eaa26-97a2-4272-
        be63-5dcb90e782e0
146     fi
147     linux /vmlinuz-4.2.0-36-generic root=UUID=7e888517-d3be-476a-
        a8b4-2955c7b5119f ro quiet splash $vt_handoff
148     initrd /initrd.img-4.2.0-36-generic
149
150 submenu 'Advanced options for Ubuntu' $menuentry_id_option 'gnulinux-
        advanced-7e888517-d3be-476a-a8b4-2955c7b5119f' {
151     menuentry 'Ubuntu, with Linux 4.2.0-36-generic' --class ubuntu --
        class gnu-linux --class gnu --class os $menuentry_id_option
        'gnulinux-4.2.0-36-generic-advanced-7e888517-d3be-476a-
        a8b4-2955c7b5119f' {
152         recordfail
153         load_video
154         gfxmode $linux_gfx_mode
```

Kernel (Cont.)

- Mounts the root file system as specified in the “**root=**” in **grub.conf**(CentOS, Fedora...) or in **grub.cfg**(Ubuntu)
- Uncompresses the *initrd* image.
- Next, mounts it and loads all the necessary drivers.
- Looks out for hard disk types be it a *LVM* or *RAID*.
- Unmounts *initrd* image and frees up all the memory occupied by the disk image.
- Then, mounts the *root partition* as specified in **grub.conf**(CentOS, Fedora...) or in **grub.cfg**(Ubuntu) as *read-only*.
- Next, runs the *init* process

Kernel (Cont.)

- Since **init** was the 1st program to be executed by *Linux Kernel*, it has the process id (**PID**) of 1.
- Do a **ps -ef | grep init** and check the **pid**.

```
instructor@Ubuntu1404:~$ ps -ef | grep init
root          1      0  0 06:08 ?        00:00:01 /sbin/init
instruc+    2217    1624  0 06:08 ?        00:00:00 init --user
instruc+    4264    2771  0 07:22 pts/9    00:00:00 grep --color=auto init
instructor@Ubuntu1404:~$
```

Init

- Executes the system to boot into the run level as specified in `/etc/inittab` traditionally.
- But, Ubuntu uses different *runlevels* than other Linux distros.
- In Ubuntu, uses `rc*.d` instead of `inittab`.
- Open `/etc/init/rc-sysinit.conf`, `/etc/init.d/README` and `/usr/share/doc/upstart/README.Debian.gz/README.Debian` to know more.

Init (Cont.)

- Except Ubuntu, other Linux distros use following default boot *runlevels* inside **/etc/inittab**.

```
# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
```

Runlevels

- When the Linux system is booting up, might see various services getting started.
- For example, it might say “**starting sendmail OK**”.
- Those are the *runlevel* programs, executed from the run level directory as defined by your run level.

Runlevels (Cont.)

- Depending on your default *init* level setting, the system will execute the programs from one of the following directories.
 - Run level 0 – `/etc/rc.d/rc0.d/`
 - Run level 1 – `/etc/rc.d/rc1.d/`
 - Run level 2 – `/etc/rc.d/rc2.d/`
 - Run level 3 – `/etc/rc.d/rc3.d/`
 - Run level 4 – `/etc/rc.d/rc4.d/`
 - Run level 5 – `/etc/rc.d/rc5.d/`
 - Run level 6 – `/etc/rc.d/rc6.d/`

Runlevel in Ubuntu

- Ubuntu uses different *runlevels* than other Linux distros.
- It also uses **rc*.d** instead of **inittab**.
- Open **/etc/rcS.d/README** and **/etc/init.d/README** to know more.
- Ubuntu has 2 different mechanisms for starting system services:
 - The traditional mechanism based on run levels, and **.d** and **/etc/rcn.d** directories
 - A new mechanism known as **upstart**.

See <http://www.pathbreak.com/blog/ubuntu-startup-init-scripts-runlevels-upstart-jobs-explained>

* Going from 0 to 6.



Runlevel in Ubuntu (Cont.)

Unix's Runlevel

Run level	Name	Description
0	Halt	Shuts down the system
1	Single-user mode	Mode for administrative tasks.
2	Multi-User mode	Does not configure network interfaces and does not export networks services
3	Multi-User Mode with Networking	Starts the system normally
4	Not used / user definable	For special purpose
5	Start the system normally with GUI display manager	Run level 3 + display manager
6	Reboot	Reboots the system
s or S	Single-user mode	Does not configure network interfaces, or start daemons.

Runlevel in Ubuntu (Cont.)

Ubuntu's Runlevel

Run level	Name	Script Directory
0	Halt	/etc/rc0.d
1, S, s	Single-user mode	/etc/rc1.d, /etc/rcS.d
2	Graphical Multi-User mode + Networking(default)	/etc/rc2.d
3	Same to Runlevel 2	/etc/rc3.d
4		/etc/rc4.d
5		/etc/rc5.d
6		/etc/rc6.d
	Reboot	

Get Current Runlevel

- Use the **runlevel** command to get current run level.
- **runlevel** is available in Ubuntu as well as redhat based distros like CentOS.

```
$ runlevel
```

```
N 2
```

```
$ who -r
```

```
instructor@Ubuntu1404:~$  
instructor@Ubuntu1404:~$ who -r  
run-level 2 2016-06-12 06:08  
instructor@Ubuntu1404:~$
```

Get Current Runlevel (Cont.)

- In Ubuntu, /etc/init/rc-sysinit.conf

```
rc-sysinit.conf x
1# rc-sysinit - System V initialisation compatibility
2#
3# This task runs the old System V-style system initialisation scripts,
4# and enters the default runlevel when finished.
5
6description      "System V initialisation compatibility"
7author           "Scott James Remnant <scott@netsplit.com>"
8
9start on (filesystem and static-network-up) or failsafe-boot
10stop on runlevel
11
12# Default runlevel, this may be overridden on the kernel command-line
13# or by faking an old /etc/inittab entry
14env DEFAULT_RUNLEVEL=2
15
16emits runlevel
17
18# There can be no previous runlevel here, but there might be old
19# information in /var/run/utmp that we pick up, and we don't want
20# that.
```

System Halt using Runlevels

- System halt

```
$ sudo init 0
```

- Restart

```
$ sudo init 6
```

Upstart

- Is an event-based replacement for the `/sbin/init` daemon.
- Since the introduction of *Upstart* some time in 2006 (since Ubuntu *6.10 Edgy Eft*), or more relevantly *9.10 Karmic* where most of the system services were converted, the boot process changed somewhat.
- Handles starting of tasks and services during boot.
- Provides the same facilities as the traditional `init` system.



Refer to

<http://upstart.ubuntu.com/cookbook/>

<https://www.digitalocean.com/community/tutorials/the-upstart-event-system-what-it-is-and-how-to-use-it>

<https://help.ubuntu.com/community/UbuntuBootupHowto>



Upstart (Cont.)

■ Directories and Configs

- `/etc/init` is where the *upstart init* configs live.
- Essentially execute whatever is required to replace *sysvinit* scripts.
- `/etc/init.d` is where all the traditional *sysvinit* scripts and the backward compatible scripts for *upstart* live.
- The backward compatible scripts basically run *service* myservice start instead of doing anything themselves.
- Some just show a notice to use the *service* command.

Upstart (Cont.)

■ Directories and Configs

- `/etc/init/rc-sysinit.conf` controls execution of traditional scripts added manually or with `update-rc.d` to traditional *runlevels* in `/etc/rc*`
- `/etc/default` has configuration files allowing to control the behavior of both traditional *sysvinit* scripts and new *upstart* configs.

Upstart (Cont.)

- If same service config file is located both `/etc/init` and `/etc/init.d`, config file in `/etc/init` is applied *primarily*.

```
instructor@Ubuntu1404:~$  
instructor@Ubuntu1404:~$ ls /etc/init/ssh.conf  
/etc/init/ssh.conf  
instructor@Ubuntu1404:~$  
instructor@Ubuntu1404:~$ ls /etc/init.d/ssh  
/etc/init.d/ssh  
instructor@Ubuntu1404:~$ █
```

Upstart (Cont.)

- Upstart scripts is located *servicename.conf* files in **/etc/init** in Ubuntu.

```
instructor@Ubuntu1404:~$  
instructor@Ubuntu1404:~$ ls /etc/init  
acpid.conf                      network-interface.conf  
alsa-restore.conf              network-interface-container.conf  
alsa-state.conf               network-interface-security.conf  
alsa-store.conf               network-manager.conf  
anacron.conf                  nmbd.conf  
apport.conf                   passwd.conf  
avahi-cups-reload.conf         plymouth.conf  
avahi-daemon.conf             plymouth-log.conf  
binfmt-support.conf           plymouth-ready.conf  
bluetooth.conf                plymouth-shutdown.conf  
bootmisc.sh.conf              plymouth-splash.conf  
checkfs.sh.conf               plymouth-stop.conf  
checkroot-bootclean.sh.conf   plymouth-upstart-bridge.conf  
checkroot.sh.conf             procs.conf  
console.conf                  pulseaudio.conf  
console-font.conf             rc.conf  
console-setup.conf            rcS.conf  
container-detect.conf         rc-sysinit.conf
```

Upstart (Cont.)

- The basic command syntax is:

\$ **sudo** [*start* | *stop* | *restart* | *status*] **servicename**

- Or

\$ **sudo initctl** [*start* | *stop* | *restart* | *reload* | *status*] **servicename**

```
instructor@Ubuntu1404:~$  
instructor@Ubuntu1404:~$ ls -l /sbin | grep initctl  
-rwxr-xr-x 1 root root 193512 7 18 2014 initctl  
lrwxrwxrwx 1 root root 7 5 16:33 initctl -> initctl  
lrwxrwxrwx 1 root root 7 5 16:33 initctl -> initctl  
lrwxrwxrwx 1 root root 7 5 16:33 initctl -> initctl  
lrwxrwxrwx 1 root root 7 5 16:33 initctl -> initctl  
lrwxrwxrwx 1 root root 7 5 16:33 initctl -> initctl
```

Upstart (Cont.)

■ initctl

- init daemon control tool

- Syntax

```
$ sudo initctl COMMANDS JOB
```

- *start* : JOB starts
- *stop* : JOB stops
- *restart* : JOB restarts
- *reload* : Sends the SIGHUP signal to running process of the names JOB instance.
- *status* : Outputs to standard output.
- *list* : Outputs the status of each to standard output.

Upstart (Cont.)

```
instructor@Ubuntu1404:~$  
instructor@Ubuntu1404:~$ ls /etc/init/ssh.conf  
/etc/init/ssh.conf  
instructor@Ubuntu1404:~$ sudo status ssh  
ssh start/running, process 4463  
instructor@Ubuntu1404:~$ sudo stop ssh  
ssh stop/waiting  
instructor@Ubuntu1404:~$ sudo start ssh  
ssh start/running, process 4488  
instructor@Ubuntu1404:~$ sudo restart ssh  
ssh start/running, process 4494  
instructor@Ubuntu1404:~$ sudo reload ssh  
instructor@Ubuntu1404:~$ sudo initctl status ssh  
ssh start/running, process 4494  
instructor@Ubuntu1404:~$ sudo initctl restart ssh  
ssh start/running, process 4507  
instructor@Ubuntu1404:~$
```


Upstart (Cont.)

- On booting, how to automatically service start

```
instructor@Ubuntu1404:/etc/init$  
instructor@Ubuntu1404:/etc/init$ pwd  
/etc/init  
instructor@Ubuntu1404:/etc/init$ cat ssh.conf  
# ssh - OpenBSD Secure Shell server  
#  
# The OpenSSH server provides secure shell access to the system.  
  
description      "OpenSSH server"  
  
start on runlevel [2345]  
stop on runlevel [!2345]  
  
respawn  
respawn limit 10 5  
umask 022  
  
env SSH_SIGSTOP=1  
expect stop  
  
# 'sshd -D' leaks stderr and confuses things in conjunction with 'console log'  
console none  
  
pre-start script  
    test -x /usr/sbin/sshd || { stop; exit 0; }  
    test -e /etc/ssh/sshd_not_to_be_run && { stop; exit 0; }  
  
    mkdir -p -m0755 /var/run/sshd
```

Traditional *init* Script

■ service

- Run a System V init script
- Syntax

\$ sudo service SCRIPT COMMAND

- *start* : This will start a service, if it's not running
- *stop* : This will stop a service, if it's running
- *restart* : This will stop, then start a service
- *status* : This will display the status of a service
- **/etc/init.d**
 - The directory containing System V init scripts.

Traditional *init* Script (Cont.)

```
instructor@Ubuntu1404:~$  
instructor@Ubuntu1404:~$ ls /etc/init.d  
acpid          friendly-recovery  pulseaudio      skeleton  
anacron        grub-common       rc              smbd  
apparmor       halt              rc.local        speech-dispatcher  
apport         irqbalance        rcS             ssh  
avahi-daemon   kerneloops        README          sudo  
binfmt-support killprocs          reboot          thermald  
bluetooth     kmod              resolvconf      udev  
brltty        lightdm           rsync           umountfs  
console-setup networking         rsyslog         umountnfs.sh  
cron          nmbd              samba           umountroot  
cups          ondemand          samba-ad-dc     unattended-upgrades  
cups-browsed  postfix           saned           urandom  
dbus          pppd-dns          sendsigs        winbind  
dns-clean    procpd            single          x11-common  
instructor@Ubuntu1404:~$ sudo service ssh status  
ssh start/running, process 4507  
instructor@Ubuntu1404:~$ sudo service ssh restart  
ssh stop/waiting  
ssh start/running, process 4538  
instructor@Ubuntu1404:~$ █
```

Traditional *init* Script (Cont.)

- The `/etc/init.d` directory contains scripts, which can *start* | *stop* | *restart* services.
- These are invoked with a *start|stop* argument at startup and shutdown

Traditional *init* Script (Cont.)

- Specify which scripts in `/etc/init.d` are enabled for run level *n*.
- For example, `/etc/rc2.d` specifies which scripts in `/etc/init.d` are enabled for run level *2*.
- At startup and shutdown, only these enabled scripts are invoked.
- Entries in `/etc/rcn.d` directories are symlinks to scripts in `/etc/init.d`, but with a special prefix of the format `[S|K]nn`

Refer to <http://www.pathbreak.com/blog/ubuntu-startup-init-scripts-runlevels-upstart-jobs-explained>

Traditional *init* Script (Cont.)

```
instructor@Ubuntu1404:/etc/rc2.d$ pwd
/etc/rc2.d
instructor@Ubuntu1404:/etc/rc2.d$ ls -l
total 4
-rw-r--r-- 1 root root 677 2017-06-17 13:59 README
lrwxrwxrwx 1 root root 20 2017-06-17 16:32 S20kerneloops -> ../init.d/kerneloops
lrwxrwxrwx 1 root root 17 2017-06-17 22:28 S20postfix -> ../init.d/postfix
lrwxrwxrwx 1 root root 15 2017-06-17 16:32 S20rsync -> ../init.d/rsync
lrwxrwxrwx 1 root root 27 2017-06-17 16:32 S20speech-dispatcher -> ../init.d/speech-dispatcher
lrwxrwxrwx 1 root root 15 2017-06-17 16:32 S50saned -> ../init.d/saned
lrwxrwxrwx 1 root root 19 2017-06-17 16:32 S70dns-clean -> ../init.d/dns-clean
lrwxrwxrwx 1 root root 18 2017-06-17 16:32 S70pppd-dns -> ../init.d/pppd-dns
lrwxrwxrwx 1 root root 21 2017-06-17 16:32 S99grub-common -> ../init.d/grub-common
lrwxrwxrwx 1 root root 18 2017-06-17 16:32 S99ondemand -> ../init.d/ondemand
lrwxrwxrwx 1 root root 18 2017-06-17 16:32 S99rc.local -> ../init.d/rc.local
instructor@Ubuntu1404:/etc/rc2.d$
```

Traditional *init* Script (Cont.)

- **S** means the script is *enabled* for this run level.
- **K** means the script is *disabled* for this run level.
- **nn** is a sequence number that can be used to control the sequence of starting services, so that services which depend on other services are started only after those other services are started.

Traditional *init* Script (Cont.)

- Below is a listing of `/etc/rc2.d`.
- It shows that tomcat6, dovecot and postfix are not *automatically* started in run level 2.
- However, they can be started manually.

```
1 K08tomcat6
2 K76dovecot
3 K80postfix
4 S20gpm
5 S20winbind
6 S50rsync
7 S70dns-clean
8 S70pppd-dns
9 S91apache2
10 S99grub-common
11 S99ondemand
12 S99rc.local
```


Traditional *init* Script (Cont.)

- The command **chkconfig** is no longer available in Ubuntu.
- The equivalent command to **chkconfig** is **update-rc.d**.
- This command nearly supports all the new versions of Ubuntu.

```
instructor@Ubuntu1404:~$  
instructor@Ubuntu1404:~$  
instructor@Ubuntu1404:~$ sudo apt-get install chkconfig  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Package chkconfig is not available, but is referred to by another package.  
This may mean that the package is missing, has been obsoleted, or  
is only available from another source  
  
E: Package 'chkconfig' has no installation candidate  
instructor@Ubuntu1404:~$
```

Traditional *init* Script (Cont.)

- Use the **update-rc.d** command to *enable* or *disable* a service at a run level.

- Syntax:

```
$ sudo update-rc.d name enable|disable runlevel
```

- Example:

```
$ sudo update-rc.d dovecot disable 2
```

- Or

```
$ sudo update-rc.d dovecot defaults
```


Traditional *init* Script (Cont.)

- **sysv-rc-conf** is an alternate option for Ubuntu.

\$ sudo apt-get install sysv-rc-conf

```
instructor@Ubuntu1404: ~  
instructor@Ubuntu1404:~$  
instructor@Ubuntu1404:~$  
instructor@Ubuntu1404:~$ sudo apt-get install sysv-rc-conf  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  libappindicator1 libindicator7 linux-headers-4.2.0-27  
  linux-headers-4.2.0-27-generic linux-image-4.2.0-27-generic  
  linux-image-extra-4.2.0-27-generic  
Use 'apt-get autoremove' to remove them.  
The following extra packages will be installed:  
  libcurses-perl libcurses-ui-perl libterm-readkey-perl  
The following NEW packages will be installed:  
  libcurses-perl libcurses-ui-perl libterm-readkey-perl sysv-rc-conf  
0 upgraded, 4 newly installed, 0 to remove and 4 not upgraded.  
Need to get 387 kB of archives.  
After this operation, 1,276 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y
```

Traditional *init* Script (Cont.)

- Instead of **chkconfig --list**

\$ **sysv-rc-conf --list**

```
instructor@Ubuntu1404: ~  
instructor@Ubuntu1404:~$ sudo sysv-rc-conf --list  
acpid  
anacron  
apparmor      S:on  
apport  
avahi-daemon  
binfmt-suppo  
bluetooth  
brltty        S:on  
console-setu  
cron  
cups  
cups-browsed  
dbus  
dns-clean     1:on      2:on      3:on      4:on      5:on  
friendly-rec  
grub-common   2:on      3:on      4:on      5:on  
halt          0:on  
irqbalance
```

Traditional *init* Script (Cont.)

- Instead of **chkconfig --list xxxx**
\$ **sudo sysv-rc-conf --list xxxx**

```
instructor@Ubuntu1404:~$  
instructor@Ubuntu1404:~$ sudo sysv-rc-conf --list ssh  
ssh  
instructor@Ubuntu1404:~$ sudo sysv-rc-conf --list postfix  
postfix      0:off      1:off      2:on       3:on       4:on       5:on       6:off  
instructor@Ubuntu1404:~$
```

Traditional *init* Script (Cont.)

- **sysv-rc-conf** command line usage :

```
$ sysv-rc-conf --list [service name]
```

```
$ sysv-rc-conf [--level <runlevels>]  
    <service name> <on | off>
```

```
$ sudo sysv-rc-conf -level 35 ssh off
```

```
$ sudo sysv-rc-conf atd on
```

Traditional *init* Script (Cont.)

- Since Ubuntu 15.05, `/etc/init.d` has been replaced by `/usr/lib/systemd`.
- Scripts can still be started and stoped by *service*.
- But, the primary command is now *systemctl*.
- The *chkconfig* command was left behind, and now you do this with *systemctl*.
- So instead of : `chkconfig enable apache2`
- Should look for the service name, and then enable it
`systemctl status apache2`
`systemctl enable apache2.service`

Refer to <http://stackoverflow.com/questions/20680050/how-do-i-install-chkconfig-on-ubuntu>
http://www.zdnet.co.kr/news/news_view.asp?article_id=20140217092543

Daemon

- Is a background process.
- Most are created by starting a process, forking it and exiting the parent.
- Init daemon
 - Is the direct or indirect ancestor of all other processes.
 - Automatically adopts all orphaned processes.

```
instructor@Ubuntu1404:~$ pstree
init- ModemManager—2*[{ModemManager}]
      NetworkManager—dhclient
                       dnsmasq
                       3*[{NetworkManager}]
      accounts-daemon—2*[{accounts-daemon}]
      acpid
      avahi-daemon—avahi-daemon
```

Daemon (Cont.)

■ Kernel threads

- Are not children of init.
- Can be started before all the userspace processes.
- Are typically used to manage hardware that's why they are directly handled by the kernel and have high priority.

```
instructor@Ubuntu1404:~$ ps -ef | more
UID          PID    PPID  C STIME TTY          TIME CMD
root           1         0  0  22:40 ?           00:00:01 /sbin/init
root           2         0  0  22:40 ?           00:00:00 [kthreadd]
root           3         2  0  22:40 ?           00:00:00 [ksoftirqd/0]
root           4         2  0  22:40 ?           00:00:00 [kworker/0:0]
root           5         2  0  22:40 ?           00:00:00 [kworker/0:0H]
root           7         2  0  22:40 ?           00:00:01 [rcu_sched]
root           8         2  0  22:40 ?           00:00:00 [rcu_bh]
root           9         2  0  22:40 ?           00:00:01 [rcuos/0]
root          10         2  0  22:40 ?           00:00:00 [rcuob/0]
```