

# Booting Linux

Chonnam National University  
School of Electronics and Computer  
Engineering

Kyungbaek Kim

# Booting



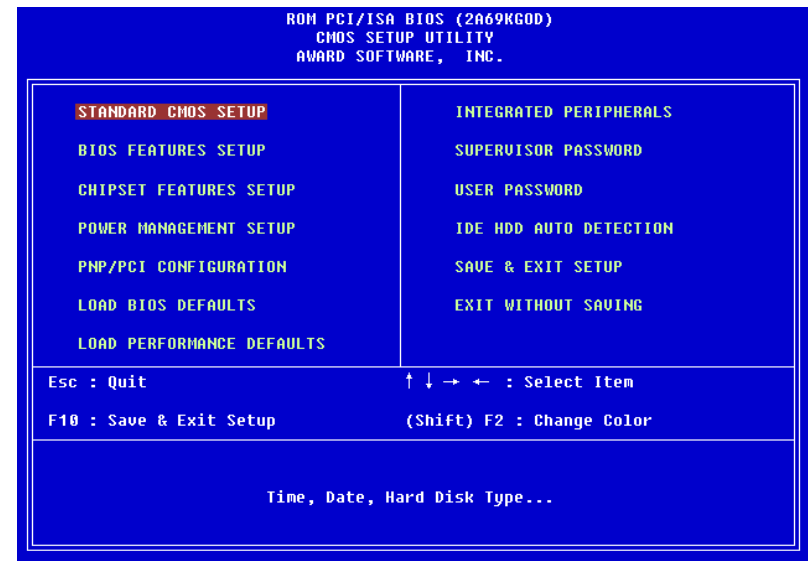
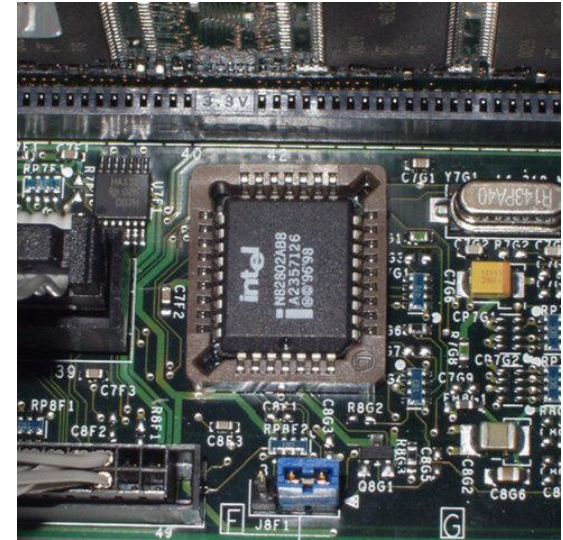
- Booting
  - A bootstrapping process that starts an operating system when user turns on a computer system
- Booting Sequence
  - The set of operations the computer performs when it is switched on that load an operating system

# Sequence of Linux Booting

- Power-up/Reset
  - 1) System startup : BIOS/Boot Monitor
  - 2) Stage1 boot loader : MBR (Master Boot Record)
  - 3) Stage2 boot loader : LILO, GRUB2, etc
  - 4) Kernel : Linux
  - 5) Init : User-space services

# BIOS

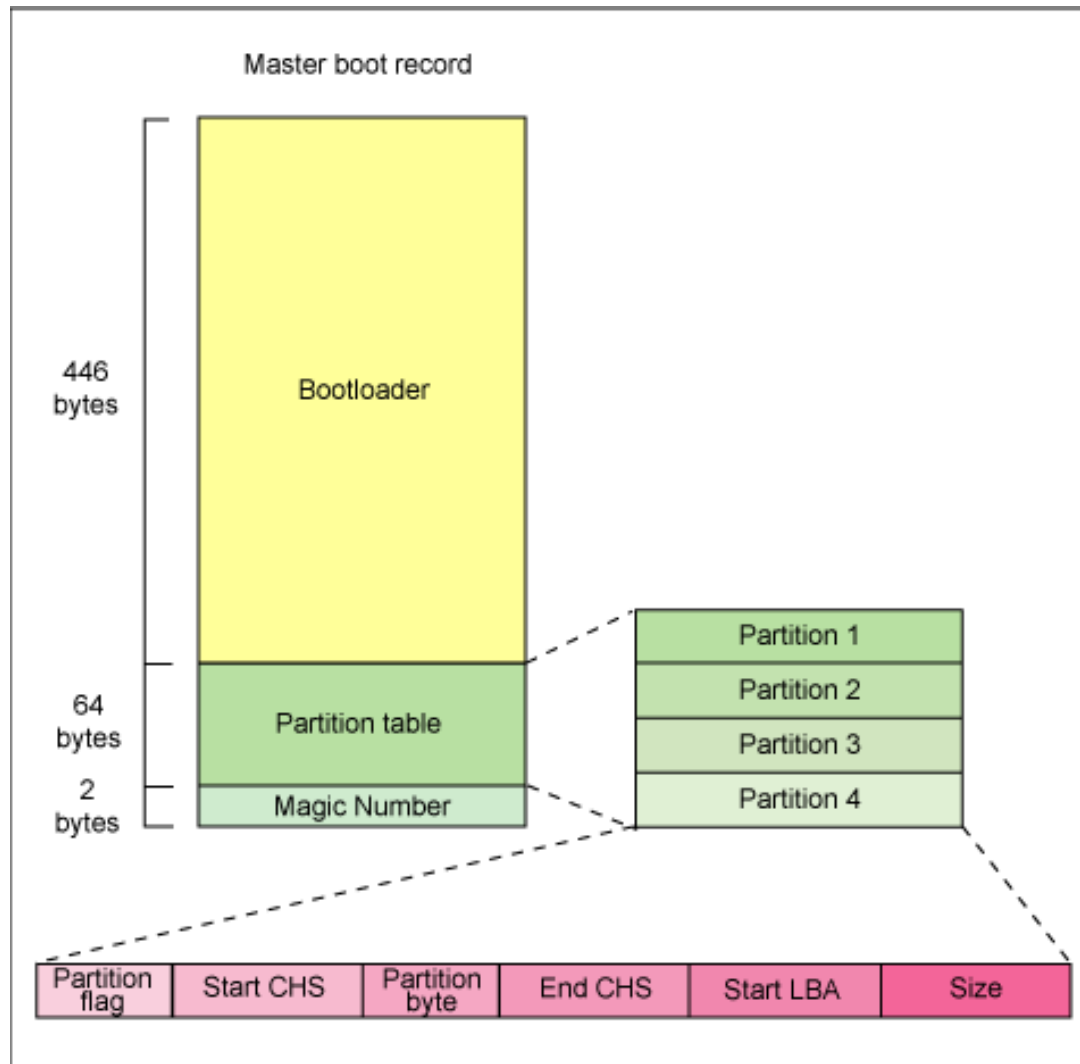
- Basic Input Output System
  - The software code runs by a computer when first powered on
- Primary function of BIOS
  - Recognize and control various device
  - e.g. *Find bootable devices* etc.
- Entering BIOS setup
  - Usually “F2” is used
  - c.f. “F12” is usually used forboot device menu



# MBR (Master Boot Record)

- Generally, Linux is booted from a hard disk, where the *Master Boot Record* contains the primary boot loader
- 512-byte sector, located in the first sector on the disk
  - Sector 1 of cylinder 0, head 0
- After the MBR is loaded into RAM, the BIOS yields control to it.

# Anatomy of MBR



# Details of MBR

- First 446 Bytes
  - Primary boot loader
  - Executable code and error message text
- Next 64 bytes
  - Partition table
  - A record for each of four partitions
- Final two bytes
  - Validation check of the MBR
  - Usually “0xAA55” is used

# Example

```
root@ubuntu: /tmp
File Edit View Terminal Help

root@ubuntu:/tmp# dd if=/dev/sda of=mbr.bin bs=512 count=1
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.000385055 s, 1.3 MB/s
root@ubuntu:/tmp# cat mbr.bin
0c000000|000000!00800
000000u000000|000t00L000|000000000000t|100 00d|<0t
00R000'0t00}000|00TH0A00U00Zr=00U0u7000t2100D00D00D000f01A|f0f000|f0\
0D00
000000}00f0cd0@f0D0000000000000f0f0`|f0uNf0\|f10f040010f0t00}7000000000
000Z02p1A0000f0f00`000100000000a0&Z|0000}0400}0.000GRUB GeomHard DiskRead Error
000000<u0000
0!0000c000000000c000000root@ubuntu:/tmp# od -xa mbr.bin
00000000 63eb 1090 d08e 00bc b8b0 0000 d88e c08e
          k c dle dle so P < nul 0 8 nul nul so X so @
00000020 befb 7c00 00bf b906 0200 a4f3 21ea 0006
          { > nul | ? nul ack 9 nul stx s $ j ! ack nul
00000040 be00 07be 0438 0b75 c683 8110 fefe 7507
          nul > > bel 8 eot u vt etx F dle soh ~ ~ bel u
00000060 ebf3 b416 b002 bb01 7c00 80b2 748a 8b01
          s k syn 4 stx 0 soh ; nul | 2 nul nl t soh vt
00000100 024c 13cd 00ea 007c eb00 00fe 0000 0000
          L stx M dc3 j nul | nul nul k ~ nul nul nul nul
```



# Boot Loader

- Kernel Loader
  - Kernel is loaded into memory by a boot loader
- Allows one of several operating systems to be loaded
  - Multiple versions of the Linux kernel
  - Dual-booting with Windows and other OSes
- Popular bootloader : LILO and GRUB

# LILO (Linux LOader)

- Not depend on a specific file system
- Can boot from harddisk and floppy
- Up to 16 different images
- Must change LILO when kernel image file or config file is changed
  - Config file : `/etc/lilo.conf`
    - Representation of “name=value” mapping
    - Linux kernels : “image=value”
    - Other OSes : “other=value”

# /etc/lilo.conf

```
boot = /dev/hda # put loader on the MBR
root = /dev/hda1 # device to mount as /

delay = 40 # 4 second delay
compact # may make booting faster
prompt # show interactive prompt
read-only # needed to allow root to be fscked

image = /vmlinuz-2.2.20 # stable kernel (default because it's 1st)
        label = linux-2.2.20
        alias = linux # shorter label
        vga = ask # let us choose the console size

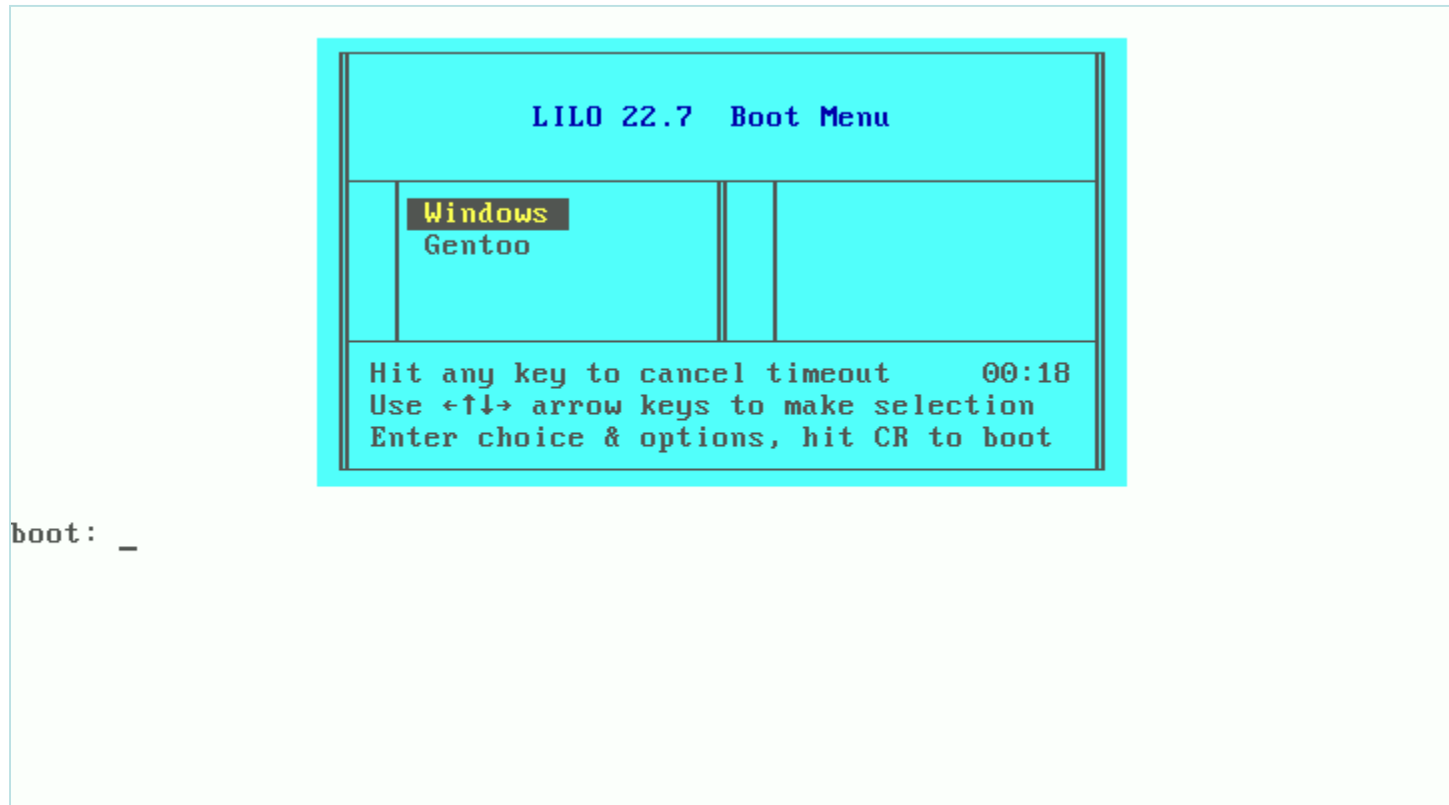
image = /vmlinuz-2.5.1 # cutting edge kernel
        label = linux-2.5.1

other = /dev/hda3 # Windows is installed on a different partition
        label = windows
        table = /dev/hda
```

# Useful Kernel Parameters in “lilo.conf”

- Kernel parameters can be specified in lilo.conf
- Common parameters have lilo.conf option names
  - “root=device”
    - Set the filesystem to mount as root
  - “ro” and “rw”
    - Mount the root filesystem read-only or read-write, respectively
  - “nfsroot=server”
    - Use a network filesystem as root
    - Netbook or diskless workstation
  - “init=program”
    - The name of the first program the kernel will run
    - Usually /sbin/init

# Example of LILO



<http://en.wikipedia.org/wiki/File:Lilo.png>

# Error of LILO

- L : First stage boot loader has been loaded and started.
  - Media failure or mismatch (bad disk parameters)
- LI : First stage boot loader was able to load the second stage boot loader.
  - Mismatch of boot program
- LIL : The second boot loader has been started, but descriptor table from the map file can not be loaded
  - Media failure
- **LILO** : All parts of LILO have been successfully loaded

# GRUB(**GR**and **U**nified **B**ootloader)

- Operating system independent boot loader
- A multiboot software packet from GNU
- Flexible command line interface
- File system access
- Support multiple executable format
- Support diskless system
- Etc.

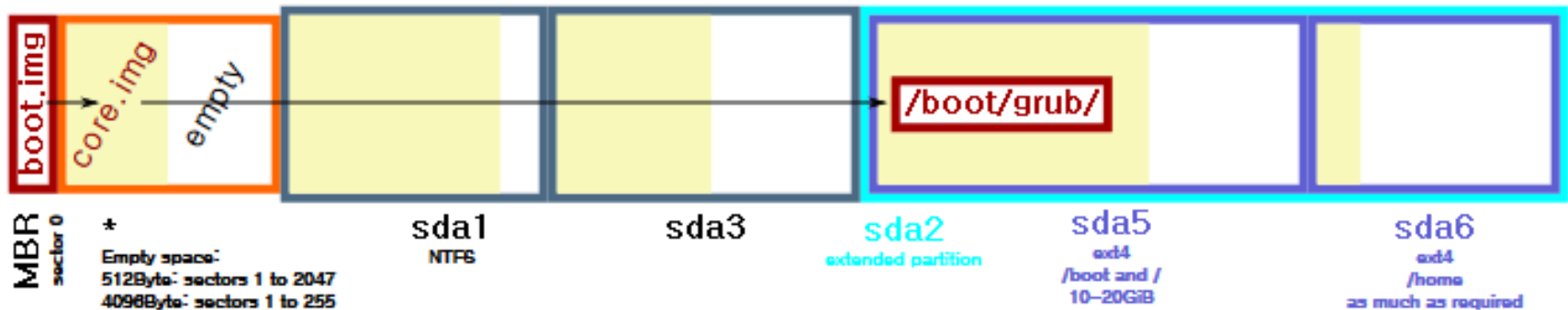
# How does GRUB work?

- 1) BIOS find a bootable device and transfer control to the master boot record (MBR)
- 2) The MBR contains GRUB stage 1 (**boot.img**). Given the small size of the MBR, stage 1 just load the next stage of GRUB
- 3) GRUB stage 1.5 located in the first 30 KB of hard disk immediately following the MBR (**core.img**). Stage1.5 loads Stage 2.
- 4) GRUB stage 2 receives control and displays to the user the GRUB boot menu (**/boot/grub**)
- 5) GRUB loads the user-selected (or default) kernel into memory and passes control on to the kernel



# GRUB2 boot up sequence

Example 1: an MBR-partitioned harddisc with sector size of 512 or 4096Bytes

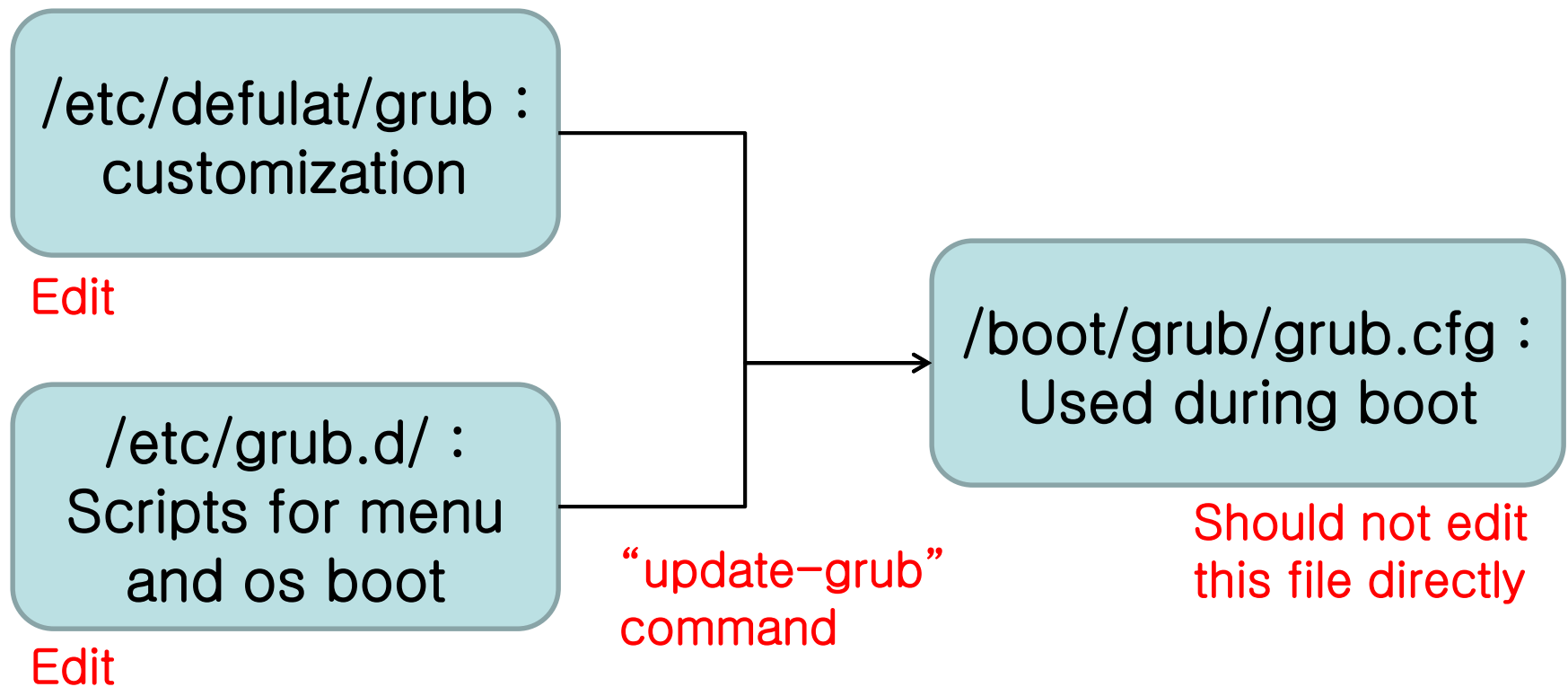


[https://en.wikipedia.org/wiki/GNU\\_GRUB](https://en.wikipedia.org/wiki/GNU_GRUB)

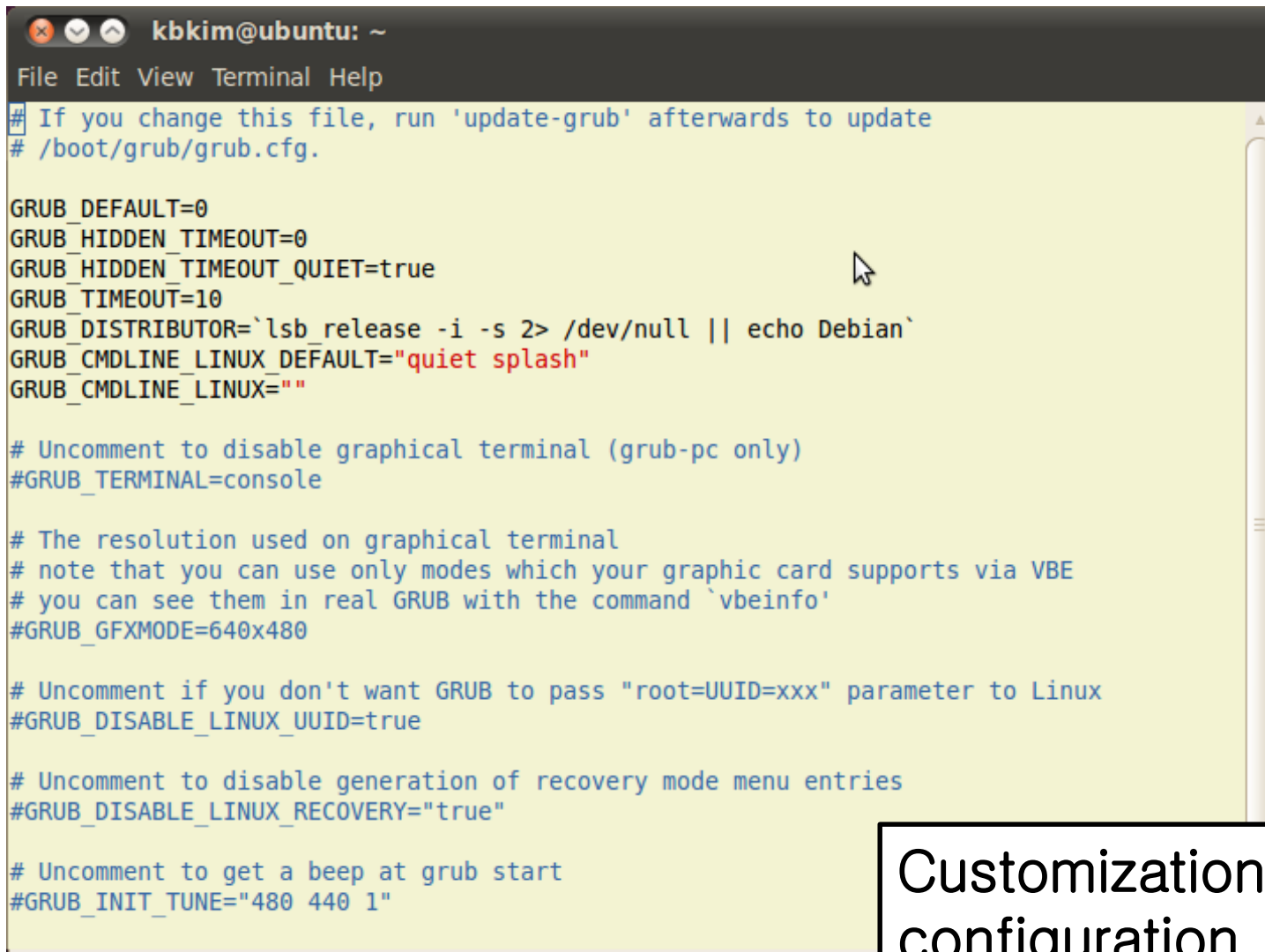
# Files related to GRUB2

```
kbkim@ubuntu: ~  
File Edit View Terminal Help  
kbkim@ubuntu:~$ ls -l /boot  
total 14376  
-rw-r--r-- 1 root root 652611 2012-01-04 06:22 abi-2.6.32-38-generic  
-rw-r--r-- 1 root root 116014 2012-01-04 06:22 config-2.6.32-38-generic  
drwxr-xr-x 3 root root 4096 2012-03-21 07:59 grub  
-rw-r--r-- 1 root root 8027732 2012-03-21 19:33 initrd.img-2.6.32-38-generic  
-rw-r--r-- 1 root root 160280 2010-03-23 02:37 memtest86+.bin  
-rw-r--r-- 1 root root 1692624 2012-01-04 06:22 System.map-2.6.32-38-generic  
-rw-r--r-- 1 root root 1196 2012-01-04 06:23 vmcoreinfo-2.6.32-38-generic  
-rw-r--r-- 1 root root 4048512 2012-01-04 06:22 vmlinuz-2.6.32-38-generic  
kbkim@ubuntu:~$ ls -l /boot/grub | grep grub  
-r--r--r-- 1 root root 3436 2012-03-21 07:59 grub.cfg  
-rw-r--r-- 1 root root 1024 2012-05-09 01:41 grubenv  
kbkim@ubuntu:~$ ls -l /etc/grub.d  
total 40  
-rwxr-xr-x 1 root root 4444 2012-01-20 06:41 00_header  
-rwxr-xr-x 1 root root 1416 2012-01-20 06:24 05_debian_theme  
-rwxr-xr-x 1 root root 4843 2012-01-20 06:41 10_linux  
-rwxr-xr-x 1 root root 918 2010-03-23 02:37 20_memtest86+  
-rwxr-xr-x 1 root root 6605 2012-01-20 06:41 30_os-prober  
-rwxr-xr-x 1 root root 214 2012-01-20 06:41 40_custom  
-rw-r--r-- 1 root root 483 2012-01-20 06:41 README  
kbkim@ubuntu:~$ ls -l /etc/default/grub  
-rw-r--r-- 1 root root 865 2012-02-14 02:47 /etc/default/grub  
kbkim@ubuntu:~$
```

# How modify the configuration?



# “/etc/default/grub”

A terminal window titled 'kdkim@ubuntu: ~' with a menu bar (File, Edit, View, Terminal, Help). The terminal displays the contents of the /etc/default/grub file. The text is as follows:

```
# 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"

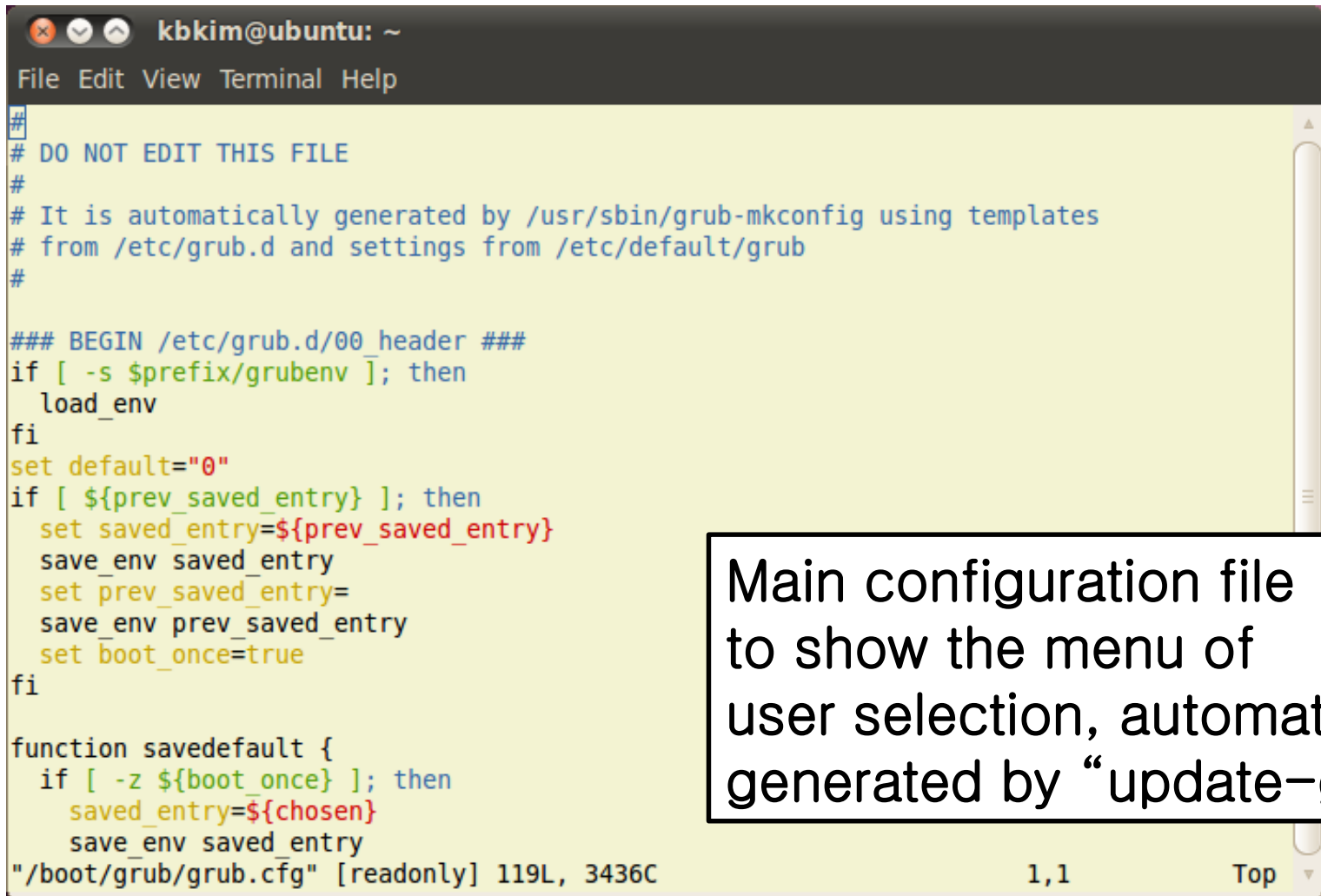
# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```

Customization configuration

# “/etc/grub.d/” directory

- Contains scripts for GRUB menu information and operating system boot scripts.
  - 00\_header → script load GRUB setting from /etc/default/grub (timeout, default, etc)
  - 05\_debian\_theme → defines background, colors, and themes
  - 10\_linux → loads menu entries for the installed distribution
  - 20\_memtest86+ → loads the memtest utility
  - 30\_os-prober → scan the hard disks for other operating systems and add them to the boot menu
  - 40\_custom → a template you can use to create additional entries

# /boot/grub/grub.cfg file



```
kbkim@ubuntu: ~
File Edit View Terminal Help
#
# 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 $prefix/grubenv ]; then
  load_env
fi
set default="0"
if [ ${prev_saved_entry} ]; then
  set saved_entry=${prev_saved_entry}
  save_env saved_entry
  set prev_saved_entry=
  save_env prev_saved_entry
  set boot_once=true
fi

function savedefault {
  if [ -z ${boot_once} ]; then
    saved_entry=${chosen}
    save_env saved_entry
  fi
}

"/boot/grub/grub.cfg" [readonly] 119L, 3436C 1,1 Top
```

Main configuration file to show the menu of user selection, automatically generated by “update-grub”

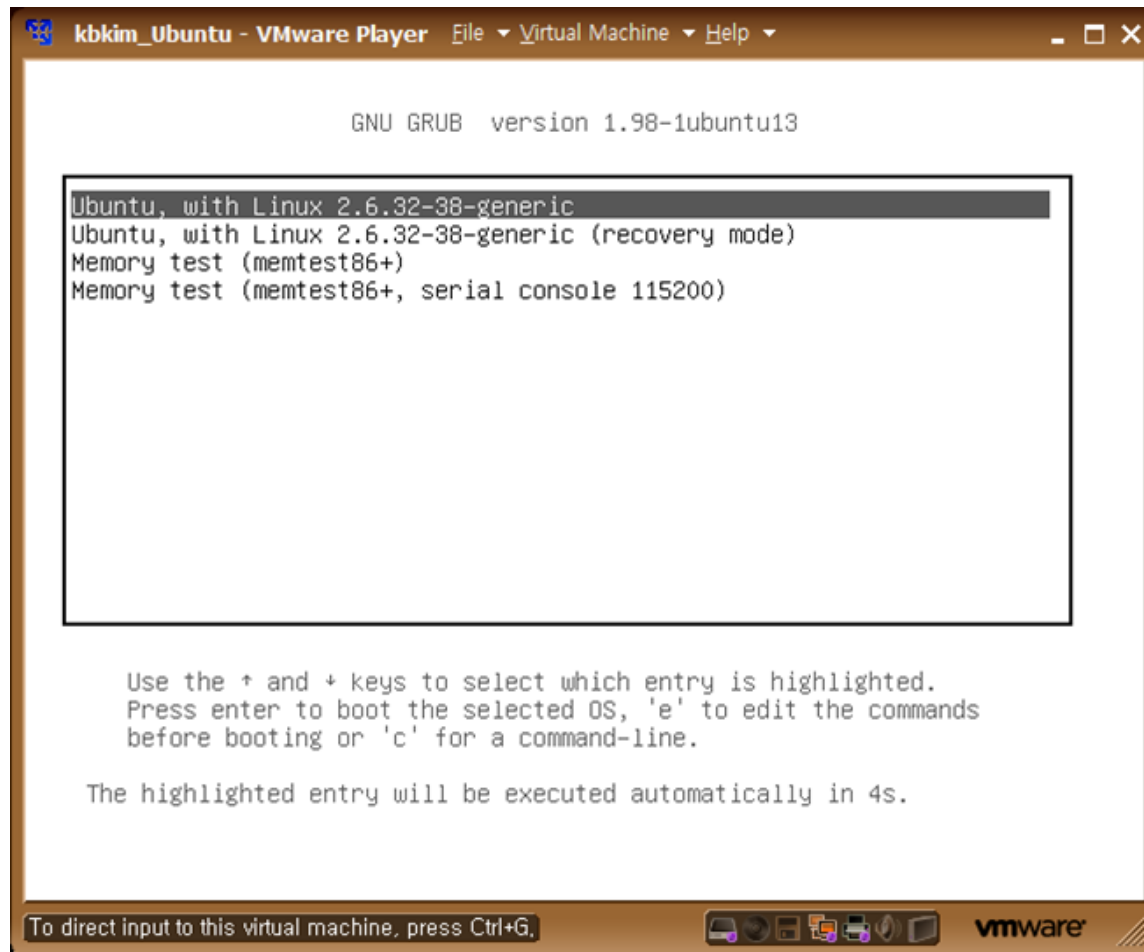
# Simple example :

## Show GRUB menu Always

```
kbkim@ubuntu: ~  
File Edit View Terminal Help  
# 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  
"/etc/default/grub" [readonly] 27 lines --3%-- 1,1 Top
```

```
kbkim@ubuntu: ~  
File Edit View Terminal Help  
kbkim@ubuntu:~$ sudo update-grub  
[sudo] password for kbkim:  
Generating grub.cfg ...  
Found linux image: /boot/vmlinuz-2.6.32-38-generic  
Found initrd image: /boot/initrd.img-2.6.32-38-generic  
Found memtest86+ image: /boot/memtest86+.bin  
done  
kbkim@ubuntu:~$  
kbkim@ubuntu:~$
```

# GRUB menu





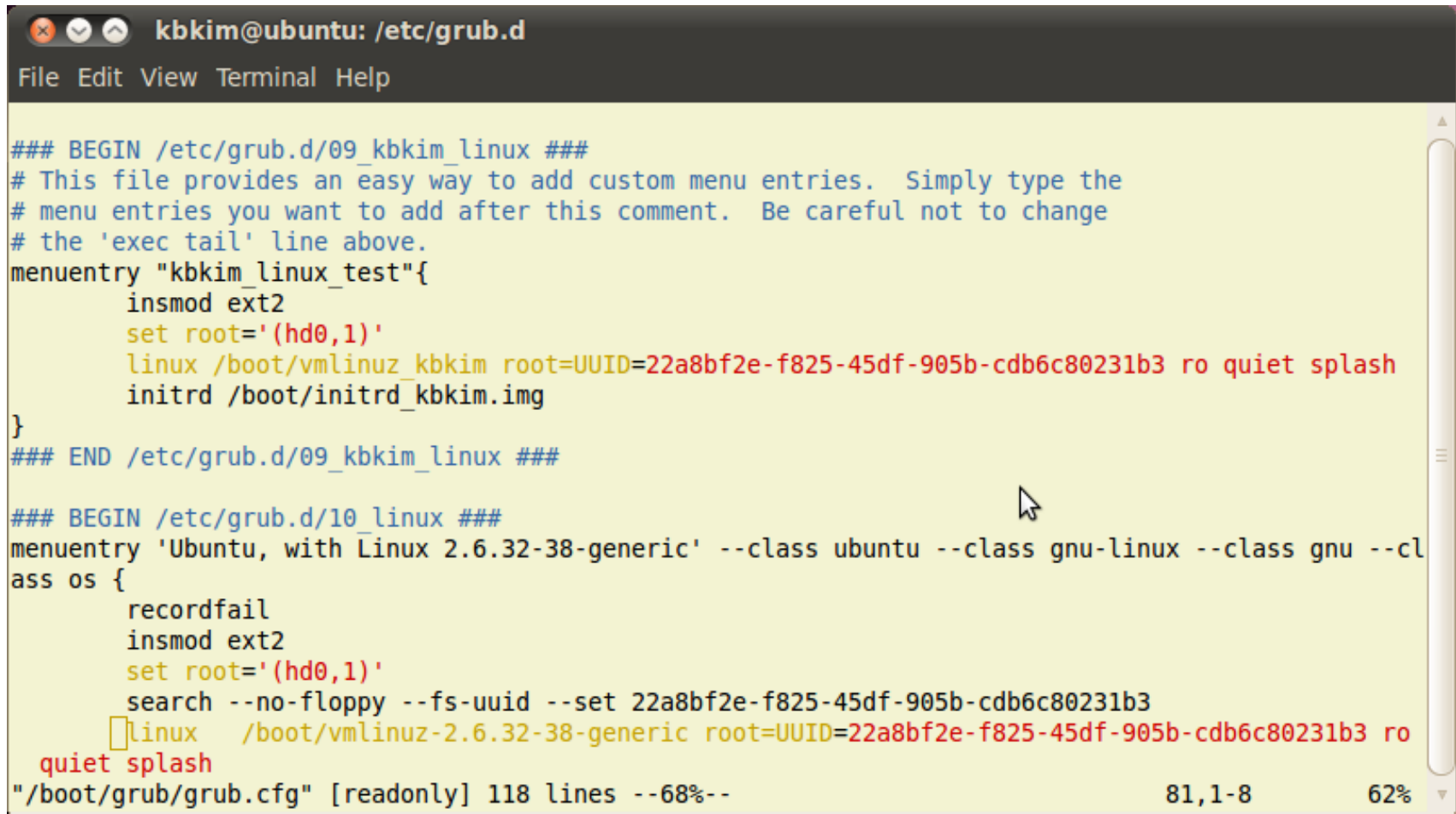
# Add a new selection

- By modifying 40\_custom script file
- XX\_osname
  - XX : values for ordering
  - osname : describing the os
- Adding a menuentry in the scriptfile
  - Linux : linux, initrd
  - Windows : chainloader
- Then, run “update-grub”

# Example

```
kbkim@ubuntu: /etc/grub.d
File Edit View Terminal Help
kbkim@ubuntu:/etc/grub.d$ ls -l
total 44
-rwxr-xr-x 1 root root 4444 2012-01-20 06:41 00_header
-rwxr-xr-x 1 root root 1416 2012-01-20 06:24 05_debian_theme
-rwxr-xr-x 1 root root 401 2012-05-10 07:51 09_kbkim_linux
-rwxr-xr-x 1 root root 4843 2012-01-20 06:41 10_linux
-rwxr-xr-x 1 root root 918 2010-03-23 02:37 20_memtest86+
-rwxr-xr-x 1 root root 6605 2012-01-20 06:41 30_os-prober
-rwxr-xr-x 1 root root 214 2012-01-20 06:41 40_custom
-rw-r--r-- 1 root root 483 2012-01-20 06:41 README
kbkim@ubuntu:/etc/grub.d$ cat 09_kbkim_linux
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
menuentry "kbkim_linux_test"{
    insmod ext2
    set root='(hd0,1)'
    linux /boot/vmlinuz_kbkim root=UUID=22a8bf2e-f825-45df-905b-cdb6c80231b3 ro quiet splash
    initrd /boot/initrd_kbkim.img
}
kbkim@ubuntu:/etc/grub.d$
```

# Modified grub.cfg after conducting “update-grub”

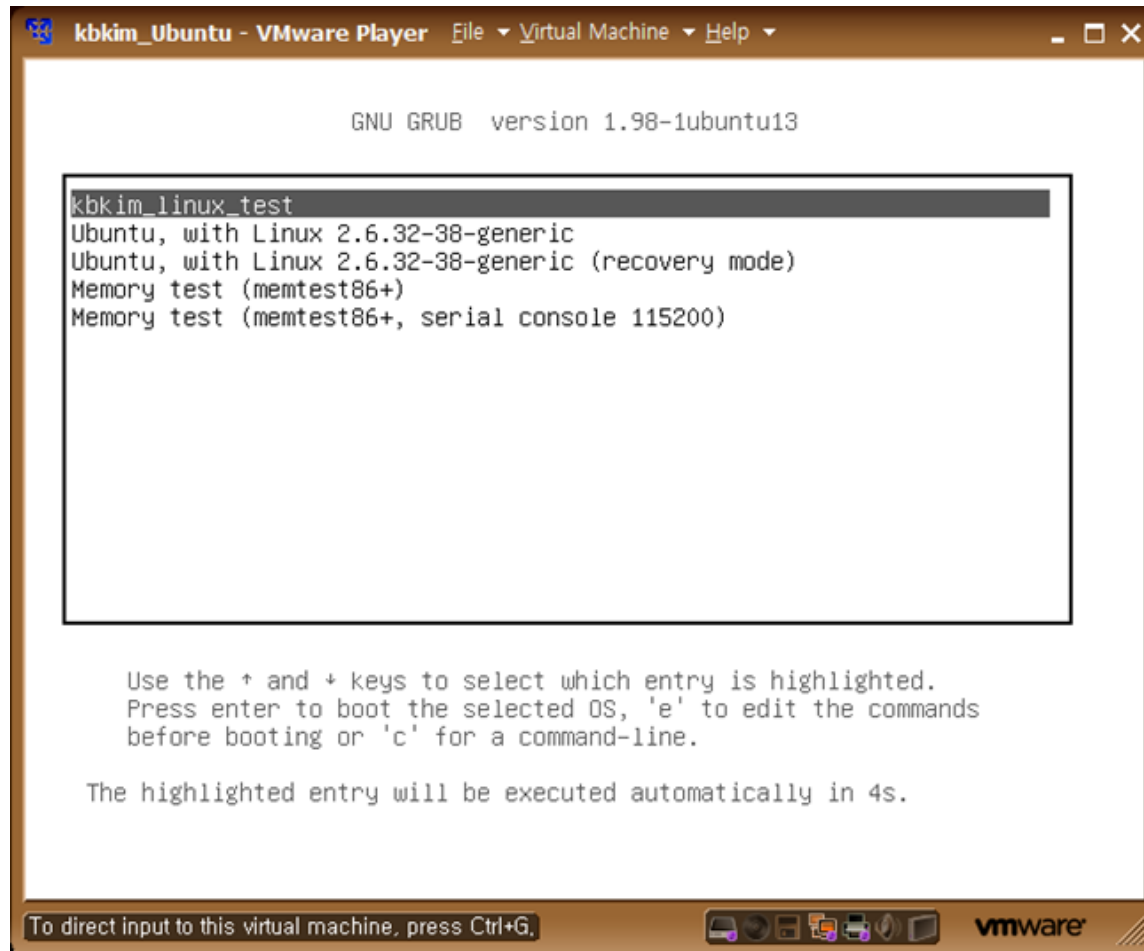


```
kbkim@ubuntu: /etc/grub.d
File Edit View Terminal Help

### BEGIN /etc/grub.d/09_kbkim_linux ###
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
menuentry "kbkim_linux_test"{
    insmod ext2
    set root='(hd0,1)'
    linux /boot/vmlinuz_kbkim root=UUID=22a8bf2e-f825-45df-905b-cdb6c80231b3 ro quiet splash
    initrd /boot/initrd_kbkim.img
}
### END /etc/grub.d/09_kbkim_linux ###

### BEGIN /etc/grub.d/10_linux ###
menuentry 'Ubuntu, with Linux 2.6.32-38-generic' --class ubuntu --class gnu-linux --class gnu --cl
ass os {
    recordfail
    insmod ext2
    set root='(hd0,1)'
    search --no-floppy --fs-uuid --set 22a8bf2e-f825-45df-905b-cdb6c80231b3
    linux /boot/vmlinuz-2.6.32-38-generic root=UUID=22a8bf2e-f825-45df-905b-cdb6c80231b3 ro
    quiet splash
}
"/boot/grub/grub.cfg" [readonly] 118 lines --68%-- 81,1-8 62%
```

# New Grub Menu



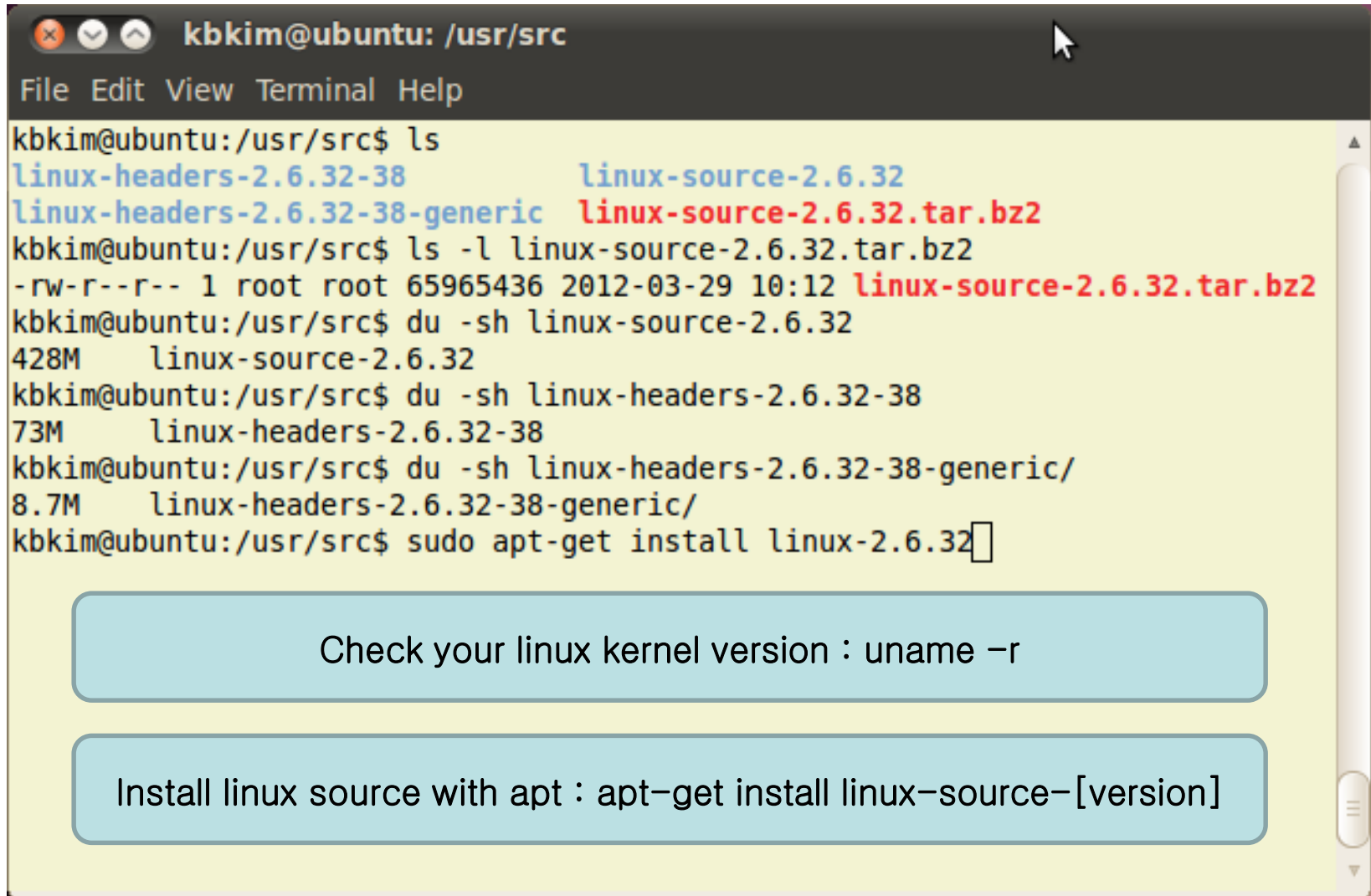
# Kernel Image

- The kernel is the central part of OS
  - Management of system's resources and the communication between hardware and software
- Kernel is always **stored in memory** during computer is on
- Kernel image is **not executable code, but a compressed image**

# You can compile kernel code

- Download linux source tar file and uncompress
  - ftp or wget or apt-get install
  - “tar -xjvf linux-source-x.x.xx.tar.bz2 -C /usr/src”
- Compile
  - “make menuconfig”
    - “make gconfig” or “make xconfig” for graphical configuration
  - “make” and “make module”
  - Becomes root, then “make modules\_install” and “make install”
    - Then you will get newly compiled linux images under “/boot/”

# Linux source directory



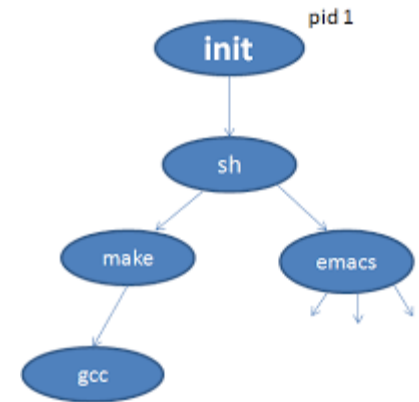
A terminal window titled 'kbbkim@ubuntu: /usr/src' with a menu bar (File, Edit, View, Terminal, Help). The terminal shows the following commands and output:

```
kbbkim@ubuntu:/usr/src$ ls
linux-headers-2.6.32-38          linux-source-2.6.32
linux-headers-2.6.32-38-generic linux-source-2.6.32.tar.bz2
kbbkim@ubuntu:/usr/src$ ls -l linux-source-2.6.32.tar.bz2
-rw-r--r-- 1 root root 65965436 2012-03-29 10:12 linux-source-2.6.32.tar.bz2
kbbkim@ubuntu:/usr/src$ du -sh linux-source-2.6.32
428M    linux-source-2.6.32
kbbkim@ubuntu:/usr/src$ du -sh linux-headers-2.6.32-38
73M     linux-headers-2.6.32-38
kbbkim@ubuntu:/usr/src$ du -sh linux-headers-2.6.32-38-generic/
8.7M    linux-headers-2.6.32-38-generic/
kbbkim@ubuntu:/usr/src$ sudo apt-get install linux-2.6.32
```

Below the terminal output, there are two light blue rounded rectangular boxes containing instructions:

- Check your linux kernel version : `uname -r`
- Install linux source with apt : `apt-get install linux-source-[version]`

# “init” Process



- The first thing the kernel does is to execute “init” program
- The ancestor of all processes in Linux
- Init process boots a Linux system to a specific system state based on the **runlevel**
  - Different runlevel provides different features and levels of functionality



# Runlevel

- Linux systems normally have seven runlevels, numbered from 0 to 6
  - Three levels are mandatory
    - 0 = halt, 6 = reboot, 1 = single-user
  - Four levels are user-defined
    - 2 ~ 5

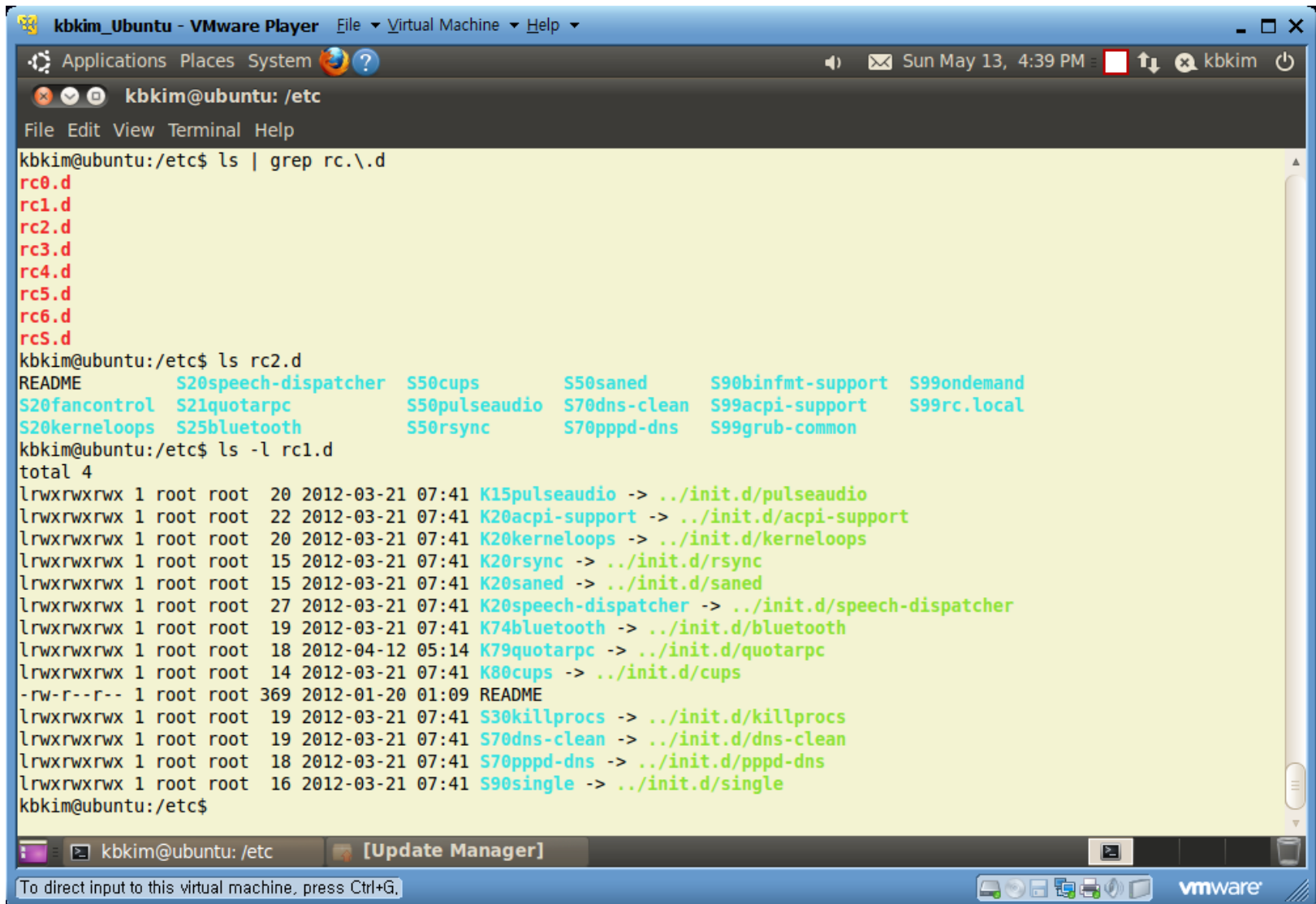
# Runlevel of Ubuntu

- 0 → known as “halt”, used to shutdown
- 1 → “single user”, which boots Ubuntu to a root access shell prompt where only root user may log in. (rescue mode)
- 2 → default runlevel for Ubuntu
- 3–5 → not used in Ubuntu
- 6 → used to reboot the system

# Runlevel 2 in Ubuntu

- It starts the system as normal
- It leaves you inside the X window system
  - For the case of a desktop linux distribution.
- Login prompt with Gnome
  - For the case of a desktop linux distribution.
- All the initial services are related to level 2 is stored in `/etc/rc2.d`
  - c.f.) For level 1  $\rightarrow$  `/etc/rc1.d`
  - c.f.) For level  $n \rightarrow$  `/etc/rcn.d`
  - Exception : `/etc/rcS.d`  $\rightarrow$  for single user

# Example



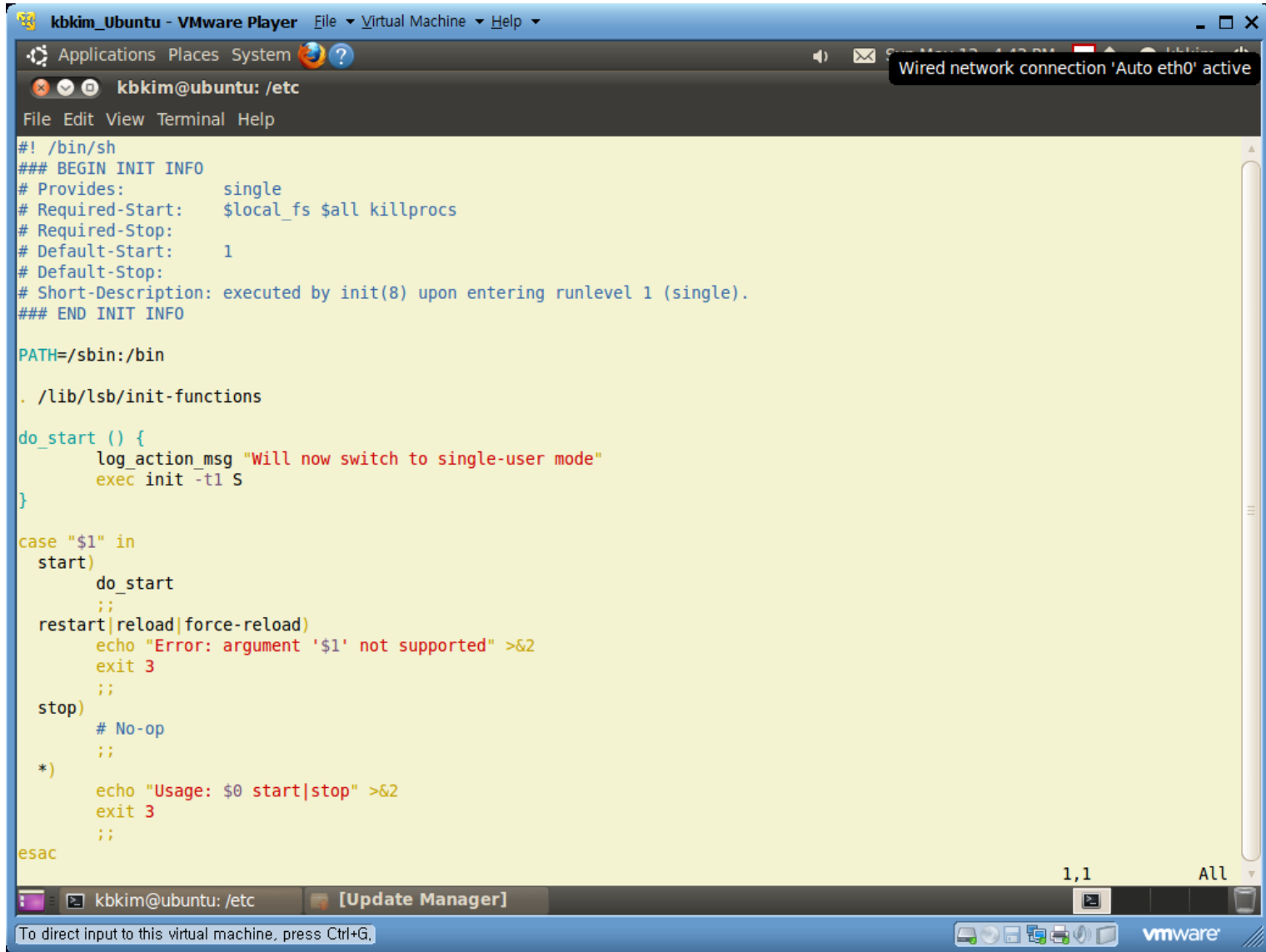
```
kbkim_Ubuntu - VMware Player  File Virtual Machine Help
Applications Places System ?
kbkim@ubuntu: /etc
File Edit View Terminal Help
kbkim@ubuntu:/etc$ ls | grep rc.\.d
rc0.d
rc1.d
rc2.d
rc3.d
rc4.d
rc5.d
rc6.d
rcS.d
kbkim@ubuntu:/etc$ ls rc2.d
README          S20speech-dispatcher  S50cups          S50saned          S90binfmt-support  S99ondemand
S20fancontrol  S21quotarpc          S50pulseaudio    S70dns-clean      S99acpi-support    S99rc.local
S20kerneloops  S25bluetooth         S50rsync         S70pppd-dns       S99grub-common
kbkim@ubuntu:/etc$ ls -l rc1.d
total 4
lrwxrwxrwx 1 root root 20 2012-03-21 07:41 K15pulseaudio -> ../init.d/pulseaudio
lrwxrwxrwx 1 root root 22 2012-03-21 07:41 K20acpi-support -> ../init.d/acpi-support
lrwxrwxrwx 1 root root 20 2012-03-21 07:41 K20kerneloops -> ../init.d/kerneloops
lrwxrwxrwx 1 root root 15 2012-03-21 07:41 K20rsync -> ../init.d/rsync
lrwxrwxrwx 1 root root 15 2012-03-21 07:41 K20saned -> ../init.d/saned
lrwxrwxrwx 1 root root 27 2012-03-21 07:41 K20speech-dispatcher -> ../init.d/speech-dispatcher
lrwxrwxrwx 1 root root 19 2012-03-21 07:41 K74bluetooth -> ../init.d/bluetooth
lrwxrwxrwx 1 root root 18 2012-04-12 05:14 K79quotarpc -> ../init.d/quotarpc
lrwxrwxrwx 1 root root 14 2012-03-21 07:41 K80cups -> ../init.d/cups
-rw-r--r-- 1 root root 369 2012-01-20 01:09 README
lrwxrwxrwx 1 root root 19 2012-03-21 07:41 S30killprocs -> ../init.d/killprocs
lrwxrwxrwx 1 root root 19 2012-03-21 07:41 S70dns-clean -> ../init.d/dns-clean
lrwxrwxrwx 1 root root 18 2012-03-21 07:41 S70pppd-dns -> ../init.d/pppd-dns
lrwxrwxrwx 1 root root 16 2012-03-21 07:41 S90single -> ../init.d/single
kbkim@ubuntu:/etc$
```

To direct input to this virtual machine, press Ctrl+G.

# /etc/rc*n*.d directory

- Contains *symbolic links* to system service scripts
  - Actual scripts reside in the /etc/init.d
- Control the sequence and the run
  - K15 or S10
  - K → the service should be killed (stop)
  - S → the service should be started (start)
  - Number (15 or 10) → order of run

# Example of Init Script



The screenshot shows a VMware Player window titled "kbkim\_Ubuntu - VMware Player". The window contains a terminal window titled "kbkim@ubuntu: /etc". The terminal displays the contents of an init script, which is a shell script for managing a service. The script includes metadata comments, environment settings, and logic for starting, stopping, and reloading the service. A notification bubble in the top right corner of the terminal window states "Wired network connection 'Auto eth0' active".

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          single
# Required-Start:    $local_fs $all killprocs
# Required-Stop:
# Default-Start:     1
# Default-Stop:
# Short-Description: executed by init(8) upon entering runlevel 1 (single).
### END INIT INFO

PATH=/sbin:/bin

. /lib/lsb/init-functions

do_start () {
    log_action_msg "Will now switch to single-user mode"
    exec init -t1 S
}

case "$1" in
    start)
        do_start
        ;;
    restart|reload|force-reload)
        echo "Error: argument '$1' not supported" >&2
        exit 3
        ;;
    stop)
        # No-op
        ;;
    *)
        echo "Usage: $0 start|stop" >&2
        exit 3
        ;;
esac
```

1,1 All

kbkim@ubuntu: /etc [Update Manager]

To direct input to this virtual machine, press Ctrl+G.

vmware

# Handling the system Runlevel

- Check the current Runlevel
  - “runlevel”
- Change the system Runlevel
  - “telinit”
  - e.g.) telinit 0 → shutdown your system
  - e.g.) telinit 3 → reboot with Runlevel 3

# Manually start an individual service

- An init script always takes an argument of “start” or “stop” to start or stop the relevant service
- For example, if the MySQL database server has an init script “/etc/init.d/mysql”
  - `$/etc/init.d/mysql start` → start MySQL
  - `$/etc/init.d/mysql stop` → stop MySQL
- Alternative method
  - `$service mysqld start/stop/restart`
  - `$systemctl start/stop/restart mysqld`
- Check the status of service
  - `$systemctl status mysqld`
  - `$ps -ef | grep mysqld`



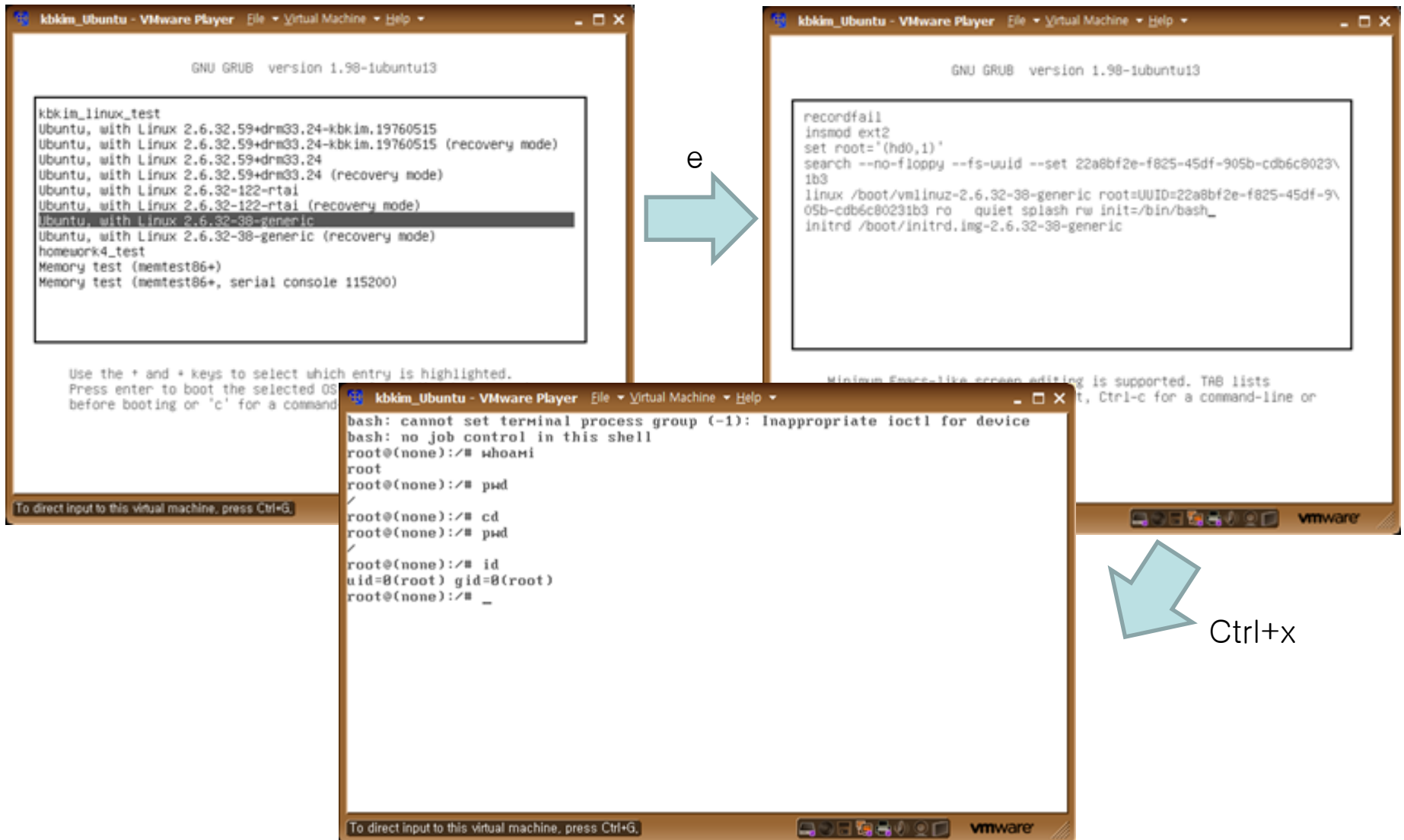
# Single-user mode

- A mode in which a multiuser computer OS boots into a single superuser.
- It is mainly used for maintenance of multi-user environments such as network servers.
- This mode can also be used for security purpose
- On some systems a lost superuser password can be changed by switching to single user mode, but not asking for the password in such circumstances is viewed as a security vulnerability

# Security hall or useful tool?

- When you lost your (or root) password, how do reset it?
  - Boot the system with single-user and start a bash with root privilege
  - Using GRUB2 and edit booting option of single user mode
    - Select a option and press ‘e’
    - Adding option of “*rw init=/bin/bash*” to boot image description (it is temporaly used)
    - Press ‘ctrl+x’ for booting with the modified option
  - After booting, the root prompt will be shown

# Example



# Shutting down and restarting

- To safely shut down the system
  - “halt” command as root
  - This is a safe shutdown
    - Stops all services
    - Disables all network interfaces
    - Unmount all file systems
- To safely reboot the system
  - “reboot” command as root
  - “Ctrl+Alt+Del” on the console

# shutdown command

- Alternative way of reboot/shutdown the Linux system
  - Define afterworks with options
    - “-r” : reboot after shutdown
    - “-h” : halt or poweroff after shutdown
    - “now” : right now do the action
  - Schedule the shutdown
    - e.g.) shutdown -h 18:00  
→ halt at 6:00 pm
    - e.g.) shutdown -r +30  
→ shutdown after 30 min, then reboot