CS5250 ASSIGNEMNT 1

YANG MO A0091836X

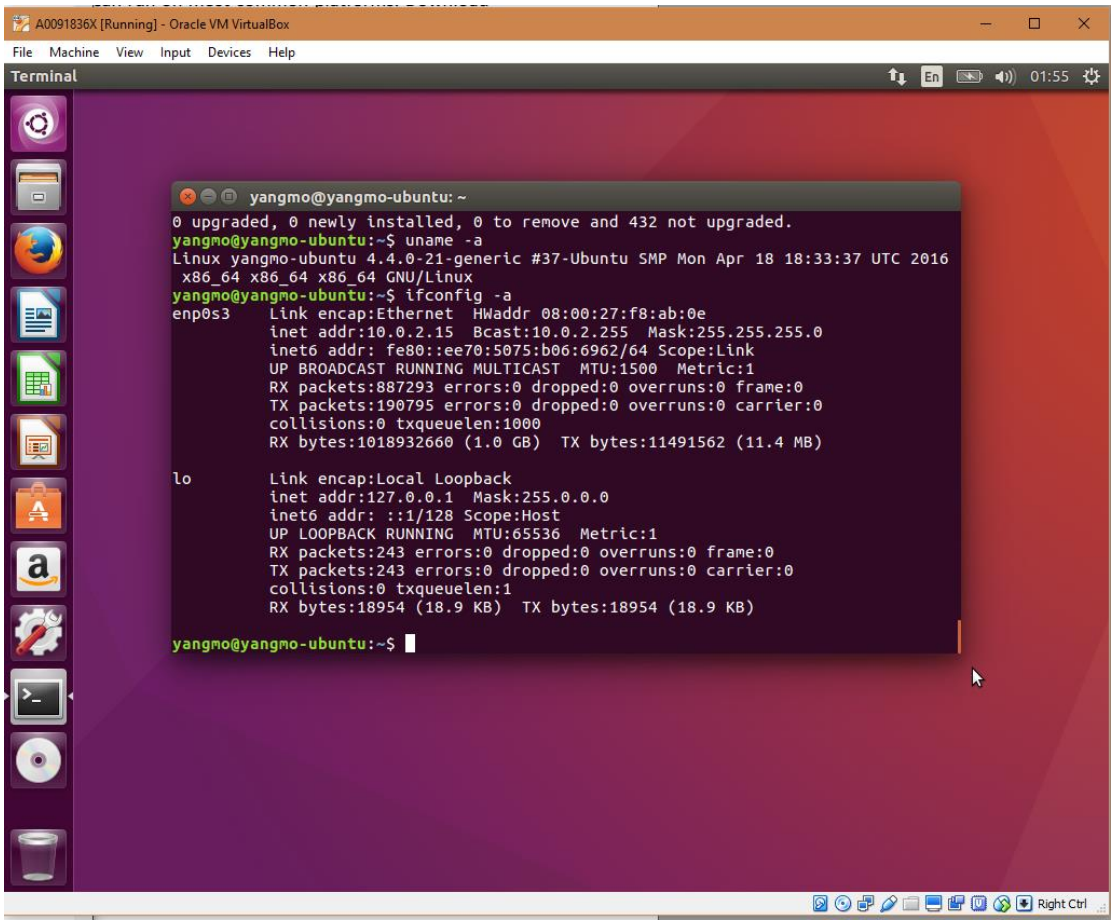
Question 1

1.1 Install a Linux guest machine

Both virtual box and Ubuntu 16.04 are installed successfully.

To output OS and kernel version, use command : **uname -a**

To output MAC address, use command : **ifconfig -a**



As can be seen in the above screenshot,

OS is **GNU/Linux**

Kernel Version is **#37-Ubuntu SMP Mon Apr 18 18:33:37 UTC 2016**

Kernel Release is **4.4.0-21-generic**

MAC Address is **08:00:27:F8:AB:0E**

1.2 Build a specific kernel

Important: After google searching for several online tutorials, I found that in order to use run "`make menuconfig`", `libncurses5 libncurses5-dev` libraries are needed. According to the online tutorial: http://mitchtech.net/compile-linux-kernel-on-ubuntu-12-04-lts-detailed/

Below are the extra dependencies needed to be installed:

```
git-core libncurses5 libncurses5-dev libelf-dev asciidoc binutils-dev linux-source fakeroot build-essential crash kexec-tools makedumpfile kernel-wedge kernel-package
```

Besides, when trying to run `make -j5` (since I am giving VM 4 processors, using `make -j5` is faster than `make -j`), some SSL error showed up. It turned out to be due to the missing package: `libssl-dev`

So in conclusion, it is necessary to run the following command before config and complie kernel:

```
sudo apt-get update
```

```
sudo apt-get install git-core libncurses5 libncurses5-dev libelf-dev asciidoc binutils-dev linux-source fakeroot build-essential crash kexec-tools makedumpfile kernel-wedge kernel-package libssl-dev
```

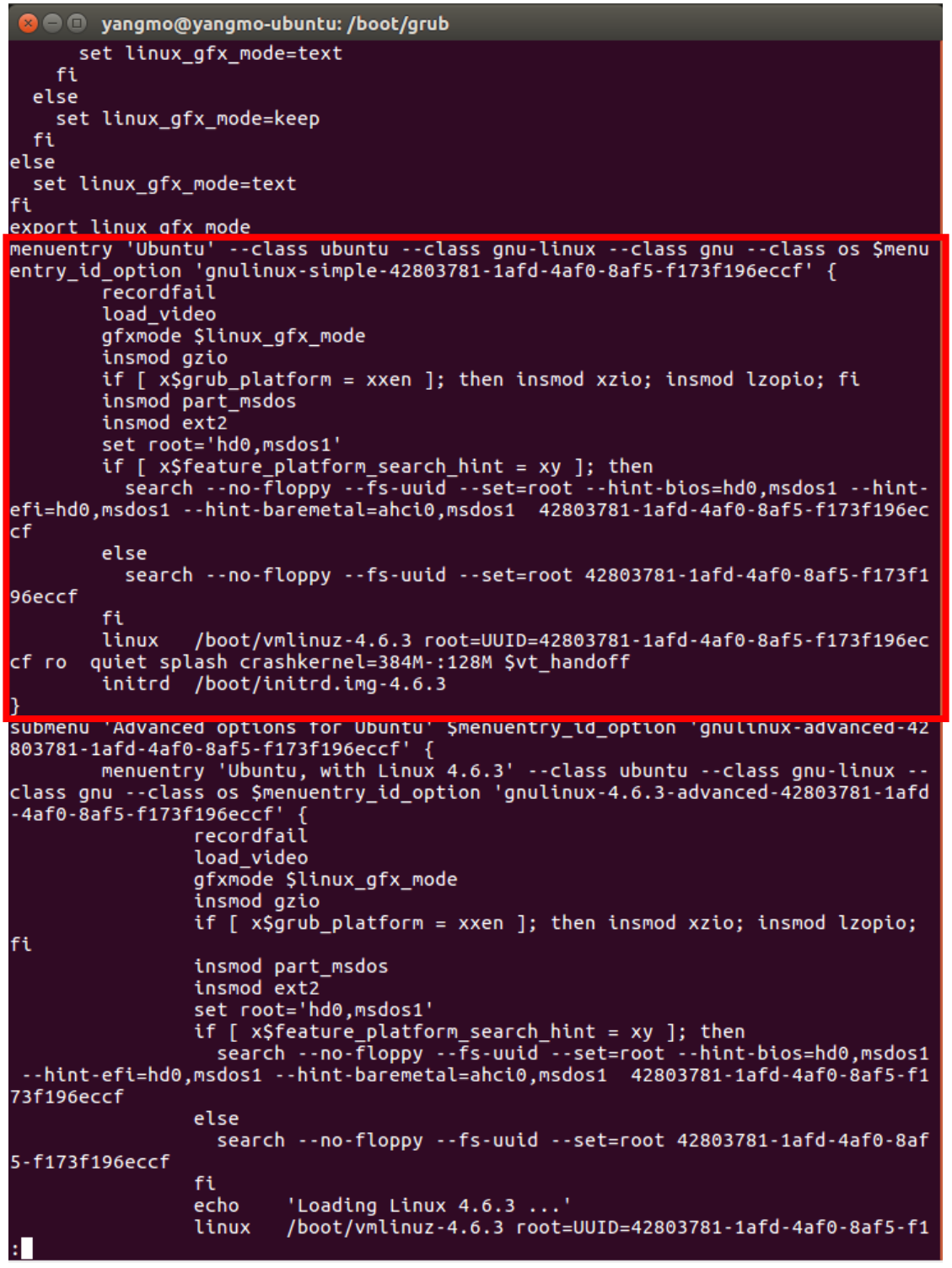
After the final command "`make install`".

The console print out:

```
Generating grub configuration file ...
Warning: Setting GRUB_TIMEOUT to a non-zero value when GRUB_HIDDEN_TIMEOUT is se
t is no longer supported.
Found linux image: /boot/vmlinuz-4.6.3
Found initrd image: /boot/initrd.img-4.6.3
Found linux image: /boot/vmlinuz-4.4.0-62-generic
Found initrd image: /boot/initrd.img-4.4.0-62-generic
Found linux image: /boot/vmlinuz-4.4.0-21-generic
Found initrd image: /boot/initrd.img-4.4.0-21-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
yangmo@yangmo-ubuntu:~/linux-4.6.3$
```

New linux and initrd image are already found in the correct place. When checking /boot/grub/grub.cfg file, new entries are already inserted correctly from the beginning of the menu:

```
yangmo@yangmo-ubuntu: /boot/grub
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#
```



According to the online tutorial: http://www.howtogeek.com/196655/how-to-configure-the-grub2-boot-loaders-settings/ to change the booting order of kernel, instead of directly changing grub.cfg file, we need to change `/etc/default/grub`, in which GRUB_DEFAULT value is the index of kernel in the entry menu to boot up. The default value in this file is already 0 which means the first entry:

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

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 enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# 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_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
/etc/default/grub (END)
```

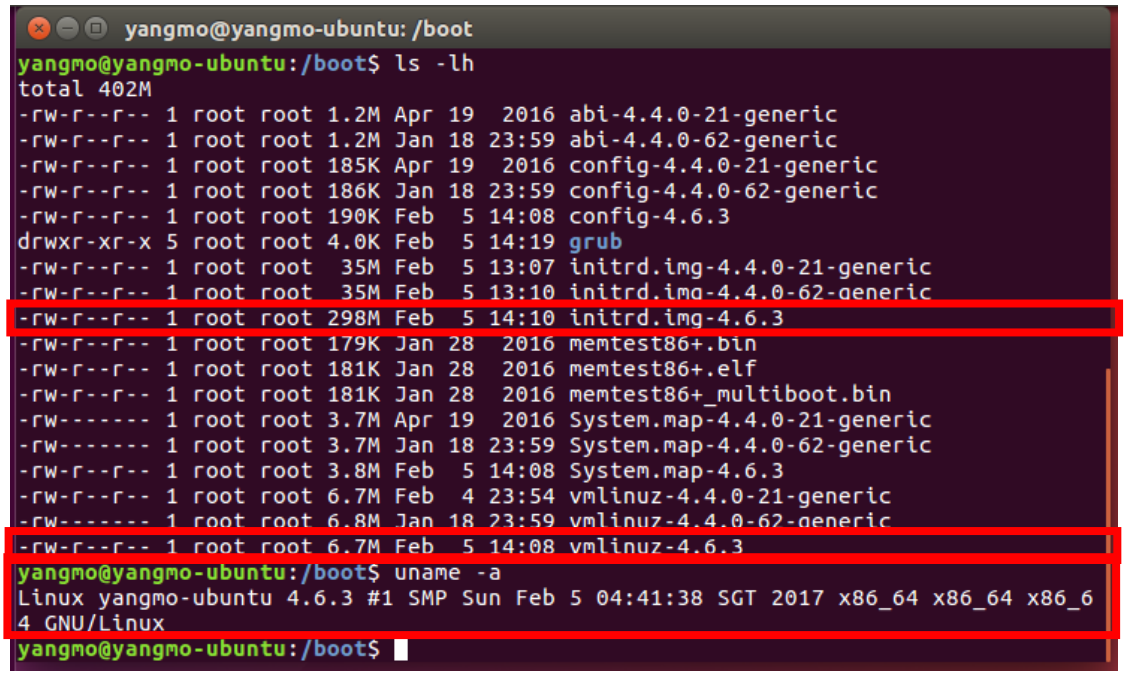
To make sure newly installed Kernel is to be booted, just run command `sudo update-grub`:

```
yangmo@yangmo-ubuntu:/boot/grub$ sudo update-grub
Generating grub configuration file ...
Warning: Setting GRUB_TIMEOUT to a non-zero value when GRUB_HIDDEN_TIMEOUT is se
t is no longer supported.
Found linux image: /boot/vmlinuz-4.6.3
Found initrd image: /boot/initrd.img-4.6.3
Found linux image: /boot/vmlinuz-4.4.0-62-generic
Found initrd image: /boot/initrd.img-4.4.0-62-generic
Found linux image: /boot/vmlinuz-4.4.0-21-generic
Found initrd image: /boot/initrd.img-4.4.0-21-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
```

The bootloader should now be confirmed to load new installed kernel 4.6.3

1.3 Reboot using new kernel

Next simply shutdown and restart the VM to verify the new kernel running:



It can also be seen that new initrd image is too large, we will reduce it and vmlinuz in question 3.

Question 2

***Q**: During the "make menuconfig" stage, there are three different choices, built-in, excluded and module. (i) What do they mean? (ii) Which are the ones that will appear in the kernel image?*

A(i):

built-in: Compile and include this feature inside of the kernel.

excluded: Do not compile the feature

module: Compile this feature as a module, separate from the kernel

A(ii):

Only built-in features will appear in kernel image. Module feature will be in a filesystem hierarchy rooted at /lib/modules/[kernel-release]

Question 3

Upon some research and trials and the help of online resources:

1. http://unix.stackexchange.com/questions/270390/how-to-reduce-the-size-of-the-initrd-when-compiling-your-kernel/270394
2. http://mitchtech.net/compile-linux-kernel-on-ubuntu-12-04-lts-detailed/
3. http://www.jinbuguo.com/kernel/longterm-linux-kernel-options.html

I found the following command and config could effectively reduce both initrd(tremendously) and vmlinuz(around 30% smaller) comparing to the default config and commands used in question 1 while in the **meantime still make the kernel able to run succesfully**.

1. Run "`make localmodconfig`" to update the configuration to only compile modules that are actually used in the system(according to reference 2 above)

```
yangmo@yangmo-ubuntu:~/linux-4.6.3$ make localmodconfig
using config: '.config'
vboxguest config not found!!
vboxsf config not found!!
vboxvideo config not found!!
*
* Restart config...
*
*
* PCI GPIO expanders
*
AMD 8111 GPIO driver (GPIO_AMD8111) [N/m/y/?] n
BT8XX GPIO abuser (GPIO_BT8XX) [N/m/y/?] (NEW)
```

2. When "make menuconfig", I made the following changes to the .config as below:

   General setup

   --Kernel Compression Mode -> XZ

   --Optimize for size -> 'Y'

   --Configure standard kernel features(expert users)

   --Include all symbols in kallsyms -> 'N'

   --Enable full-sized data structures for core -> 'N'

   --Enable SLUB debugging suppot -> 'N'

   Enable loadable module support

   --Compress modules on installation

   --Compression algorithm -> 'XZ'

   Device Drivers

   -- Serial ATA and Parallel ATA drivers (libata)

   --Verbose ATA error reporting -> 'N'

Kernel Hacking

-- Compile-time checks and compiler options

-- Compile the kernel with debug info -> 'N'

-- Strip assembler-generated symbols during link -> 'Y'

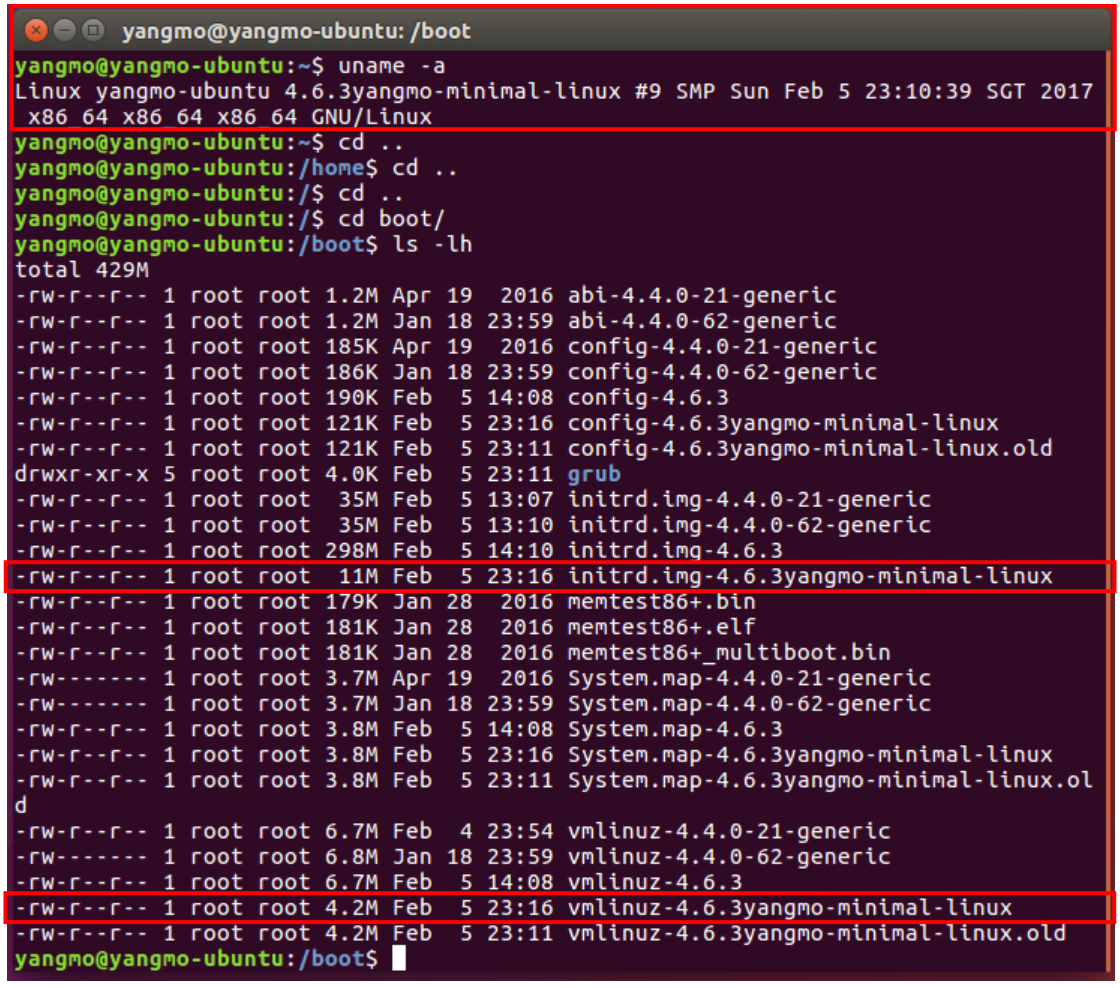
-- Enable unused/obsolete exported symbols -> 'N'

-- Collect scheduler debugging info -> 'N'

-- Collect scheduler statistics -> 'N'

-- Verbose BUG() reporting (adds 70K) -> 'N'

3. For initrd, use `sudo make INSTALL_MOD_STRIP=1 modules_install` instead of `sudo make modules_install`
4. After above steps, reboot Ubuntu, I finally got an much smaller kernel running successfully:



a. As seen from above, kernel release is now 4.6.3yangmo-minimal-linux
b. Initrd.img-4.6.3yangmo-minimal-linux is just 11M compared to 298M in question 1
c. Vmlinuz-4.6.3yangmo-minimal-linux is just 4.2M compared to 6.7M in question 1