

REPORT



운영체제

Operating System

Assignment 1

김태석 교수님

컴퓨터정보공학부

2017202037 오민혁

2020/09/14 (월)

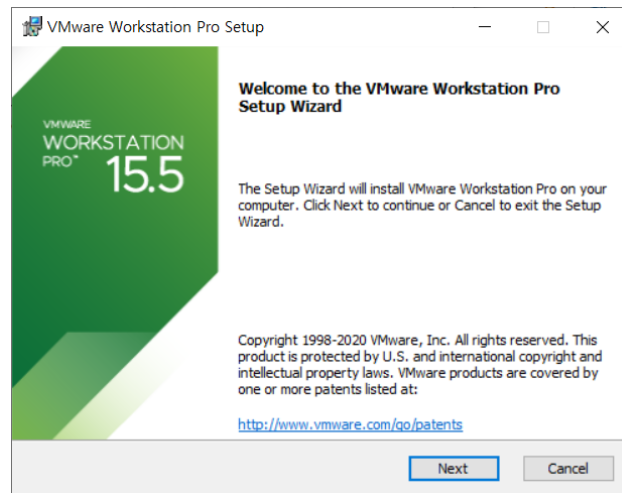


< Introduction >

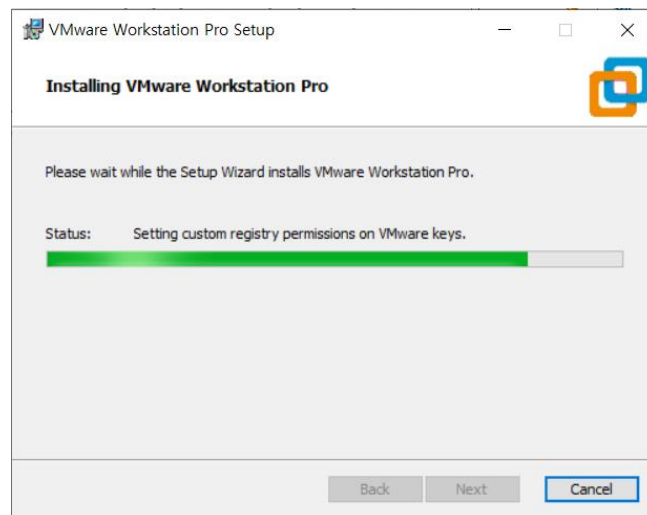
Assignment 1-1은 가상머신을 설치하고, 이 가상머신에 리눅스를 설치 하는 과정을 사진과 함께 설명하는 과제이다. 계정 ID는 os_학번 또는 os학번으로 설정해야 한다. 그 다음 ubuntu에서 kernel을 컴파일 하는 과정을 모두 캡처하고 설명한다.

Assignment 1-2는 dmesg로 kernel message를 확인 하는 과제이다. 이 과정에서 특정 위치에 본인의 학번이 찍히도록 커널 코드를 수정해야 한다.

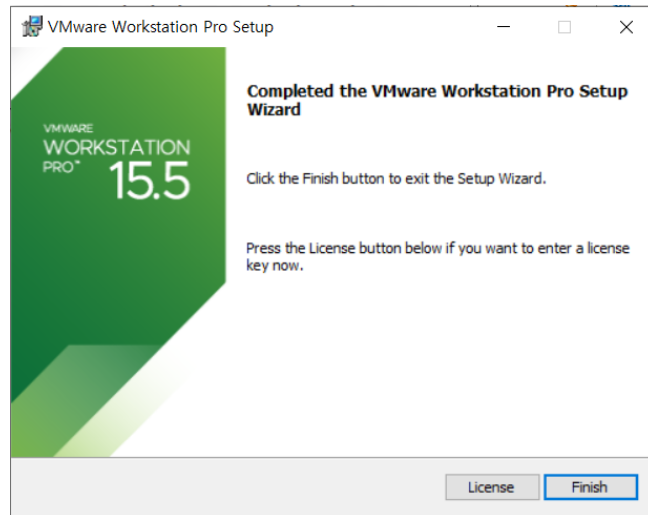
< Assignment1-1 >



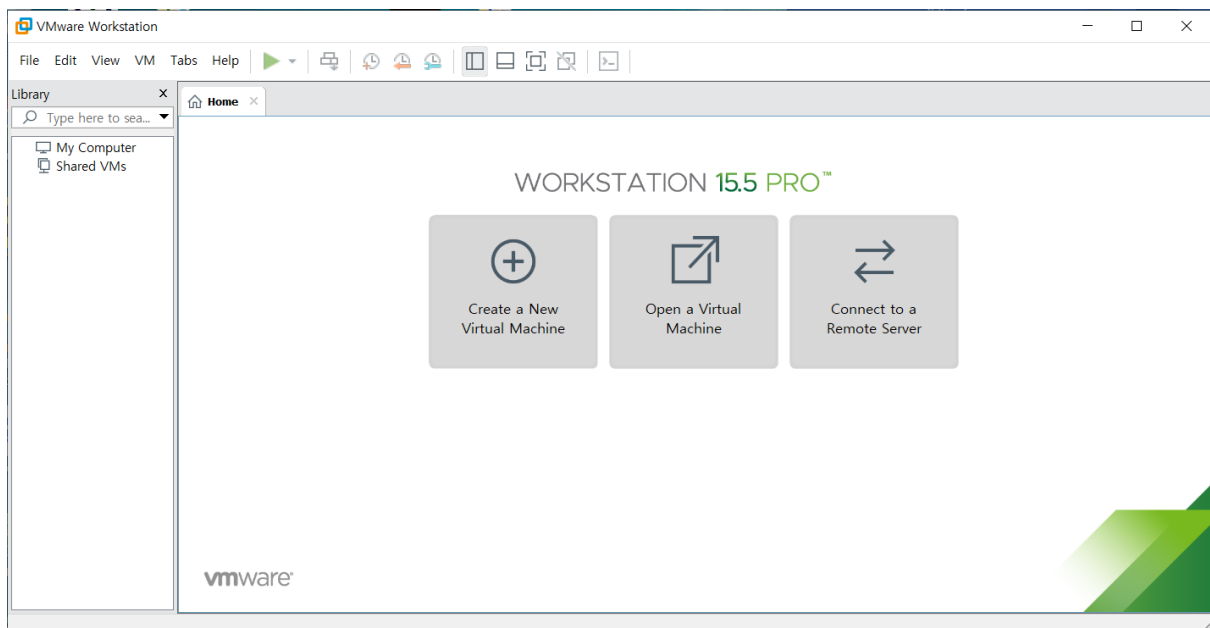
가상머신을 만들기 위한 도구인 VMware workstation을 설치하는 초기 화면이다.



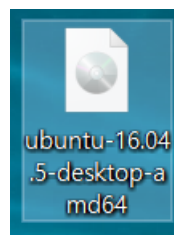
별 다른 설정 없이 Next를 계속 해서 클릭하면 자연스레 설치가 진행 된다.



마지막에 설치가 완료 된 후 Finish를 클릭하게 되면 설치가 완료된다.



VMware workstation 초기 화면이다. 여기서 Create a New Virtual Machine을 클릭하여 리눅스를 설치 하기 위한 가상머신을 설정해야 한다.



그 전에 위의 사진과 같은 리눅스 Ubuntu 버전 ISO 파일이 필요하다.

New Virtual Machine Wizard

Guest Operating System Installation

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

☐ Installer disc:

No drives available

☒ Installer disc image file (iso):

C:\Users\WOMH\Desktop\ubuntu-16.04.5-desktop

☐ Ubuntu 64-bit 16.04.5 detected.
This operating system will use Easy Install. ([What's this?](#))

☐ I will install the operating system later.
The virtual machine will be created with a blank hard disk.

iso 파일을 설정하는 과정이다.

New Virtual Machine Wizard

Easy Install Information

This is used to install Ubuntu 64-bit.

Personalize Linux

Full name: os2017202037

User name: os2017202037

Password: ●●●●

Confirm: ●●●●

리눅스에서 사용 할 계정과 password를 설정하는 과정이다.

New Virtual Machine Wizard ✕

Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name:

Location:
 Browse...

The default location can be changed at Edit > Preferences.

< Back Next > Cancel

가상머신의 이름과 경로를 설정하는 부분이다.

New Virtual Machine Wizard ✕

Specify Disk Capacity
How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (GB): ▲ ▼

Recommended size for Ubuntu 64-bit: 20 GB

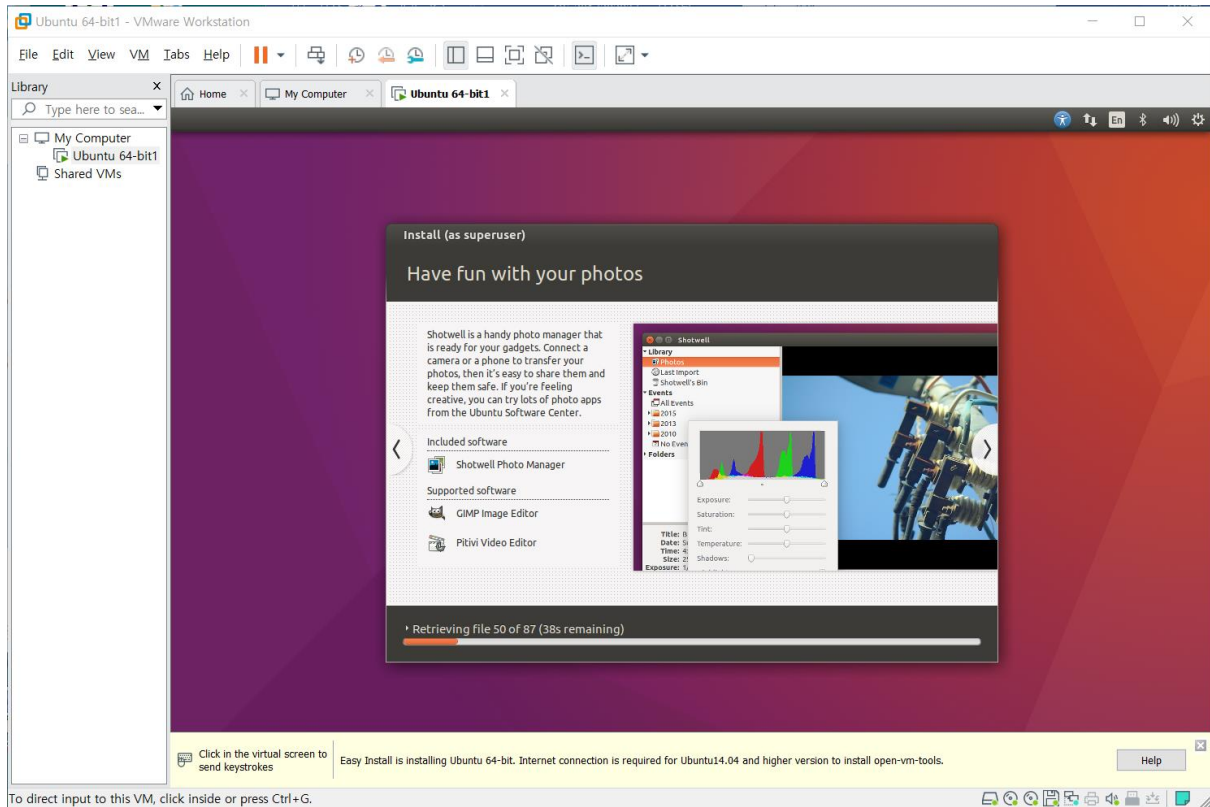
☐ Store virtual disk as a single file
☒ Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

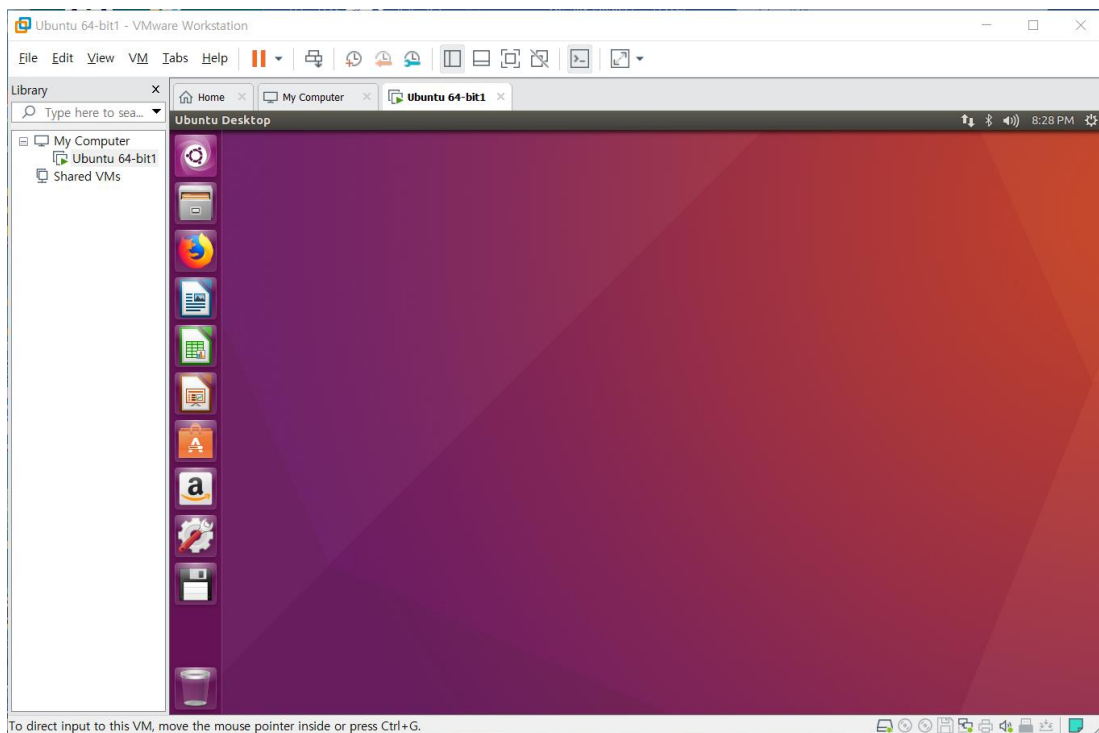
Help < Back Next > Cancel

할당할 용량까지 설정해주어야 한다.

이 때 본 PC의 하드웨어를 고려하여 설정해주어야 한다.

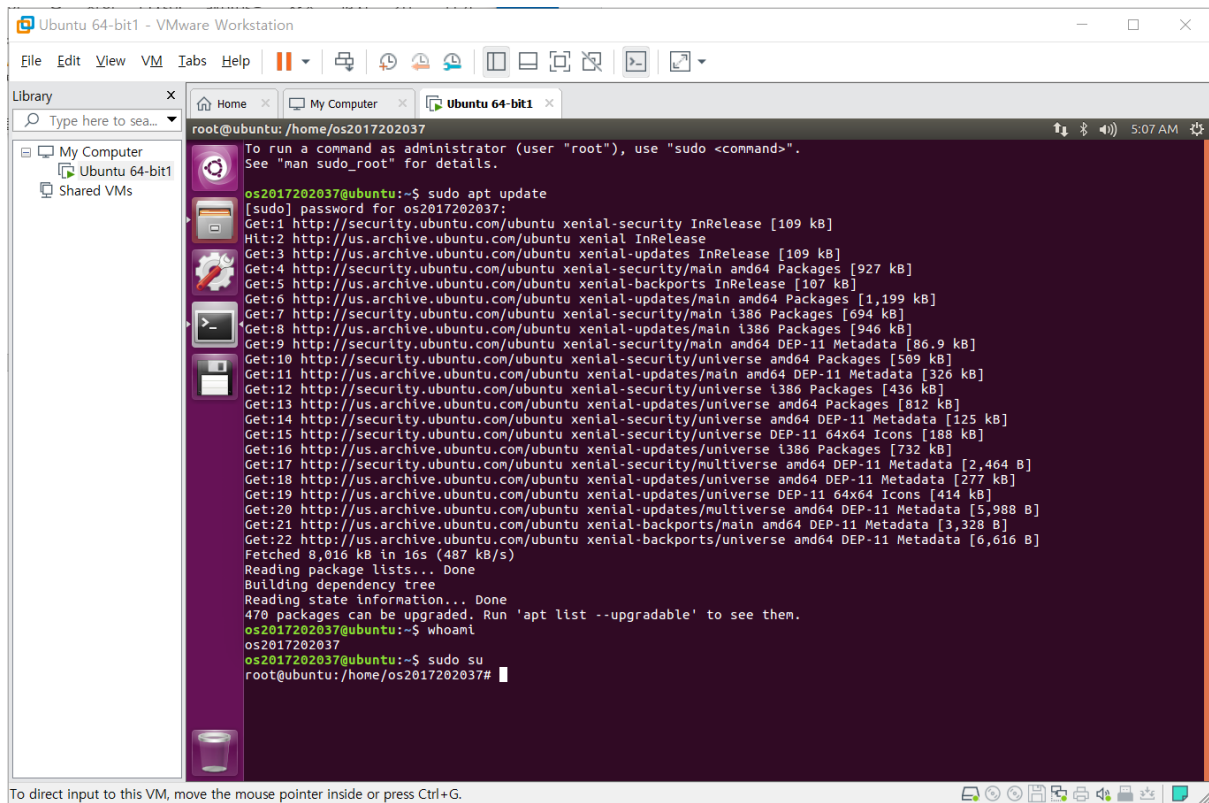


모든 설정을 완료 한 후, ubuntu가 가상머신에 설치되는 모습이다.



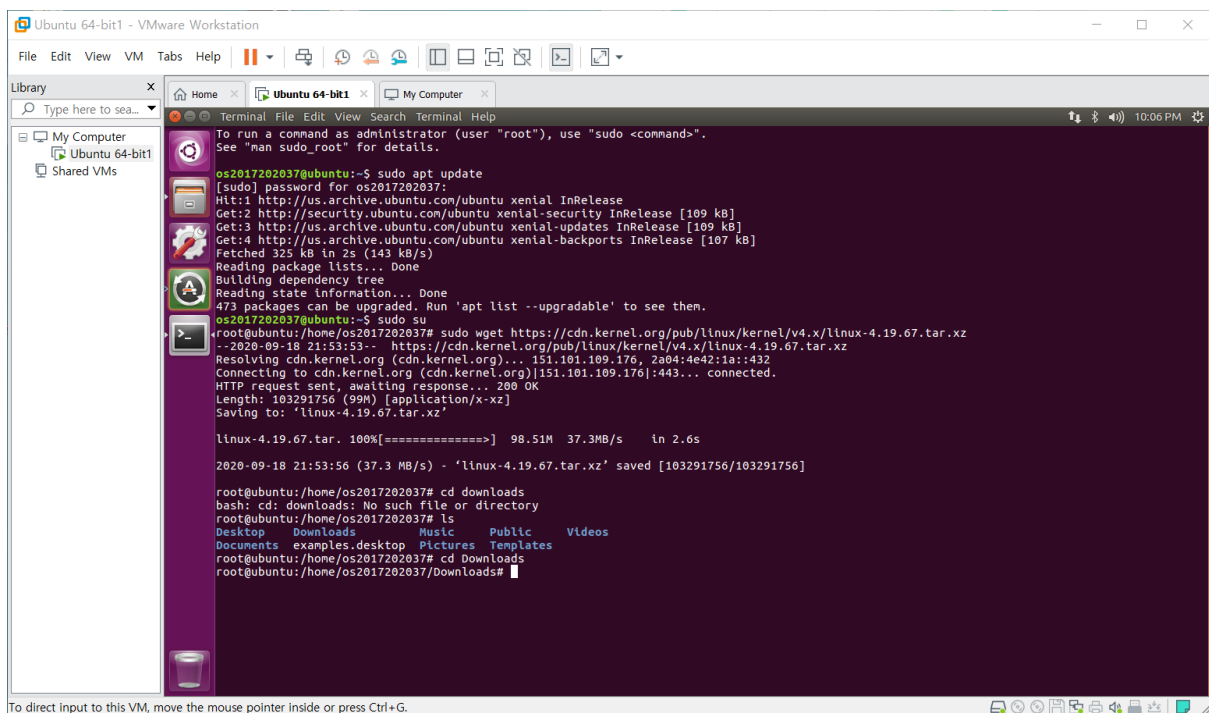
우분투를 설치 할 때, 설정한 계정과 패스워드를 이용하여 로그인 한 화면이다.

윈도우에도 바탕화면이 있듯이, 리눅스 우분투에도 이렇게 바탕화면이 있다.

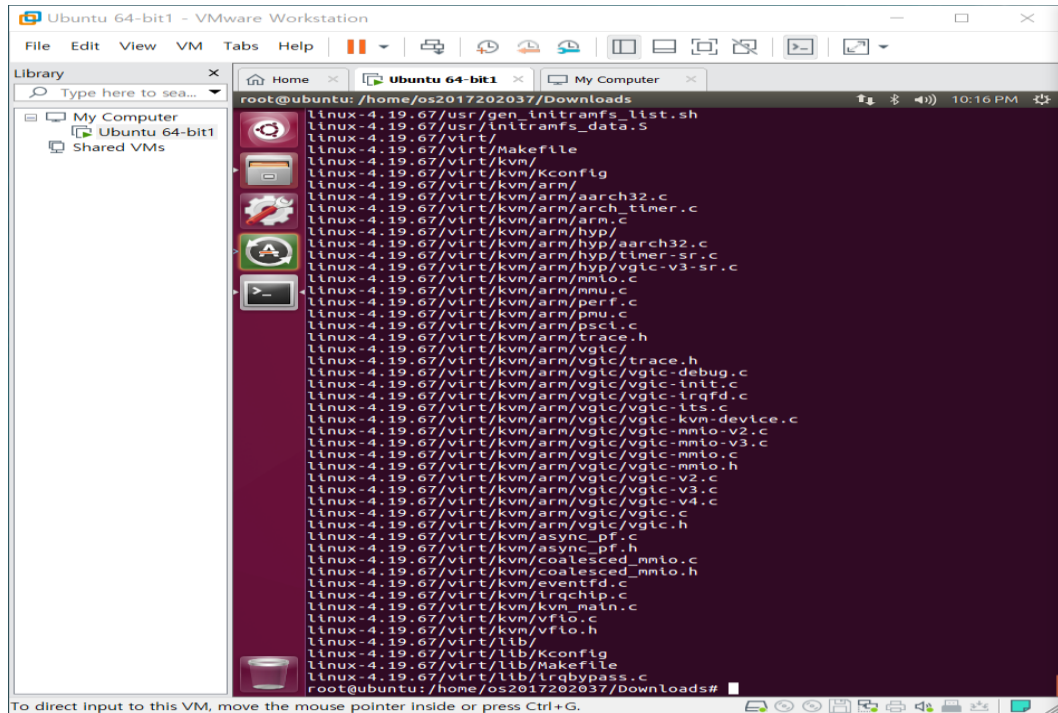


Kernel을 컴파일 하기 위해

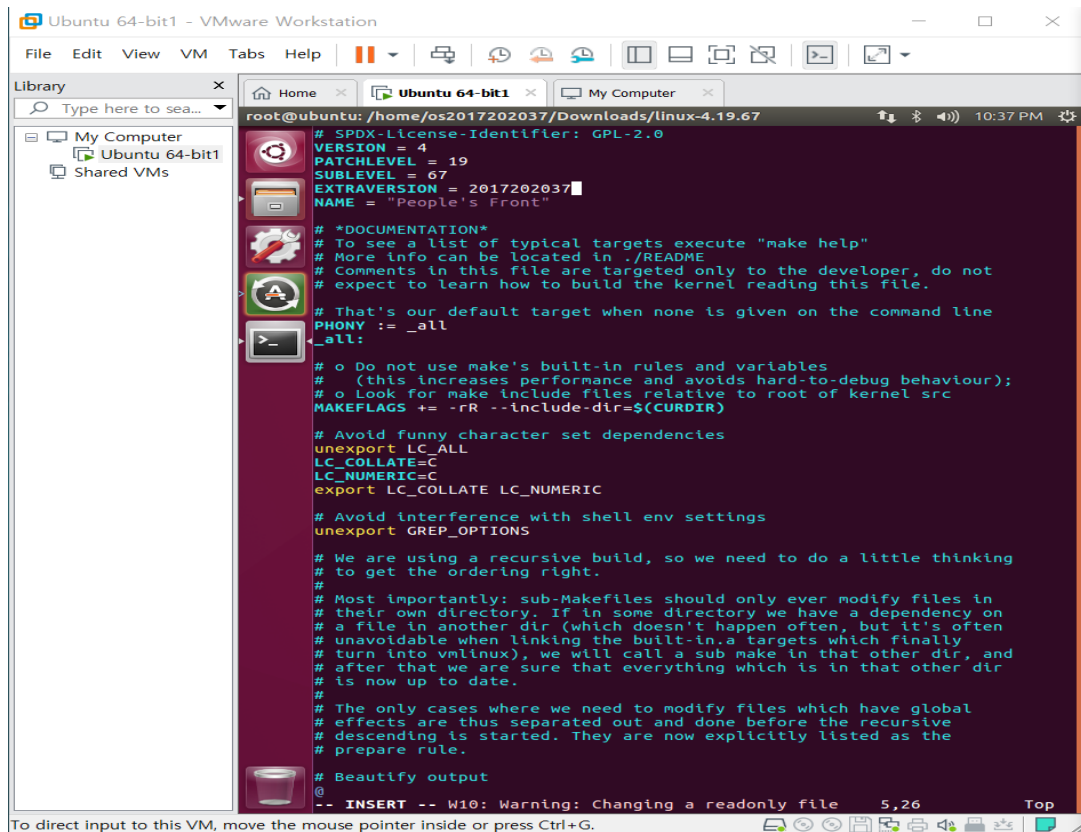
sudo 명령어를 이용하여 최고 관리자 (root) 권한으로 프로그램을 실행하였다.



그 다음 Kernel을 다운로드 받은 모습이다.

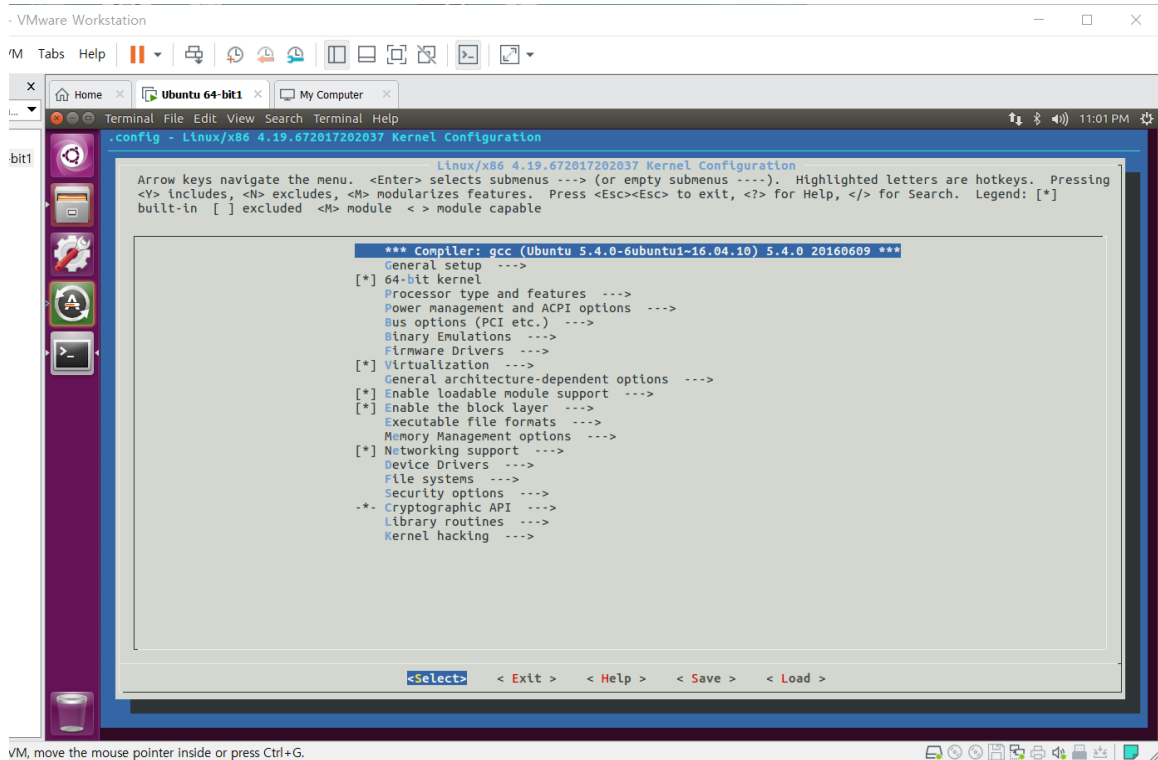
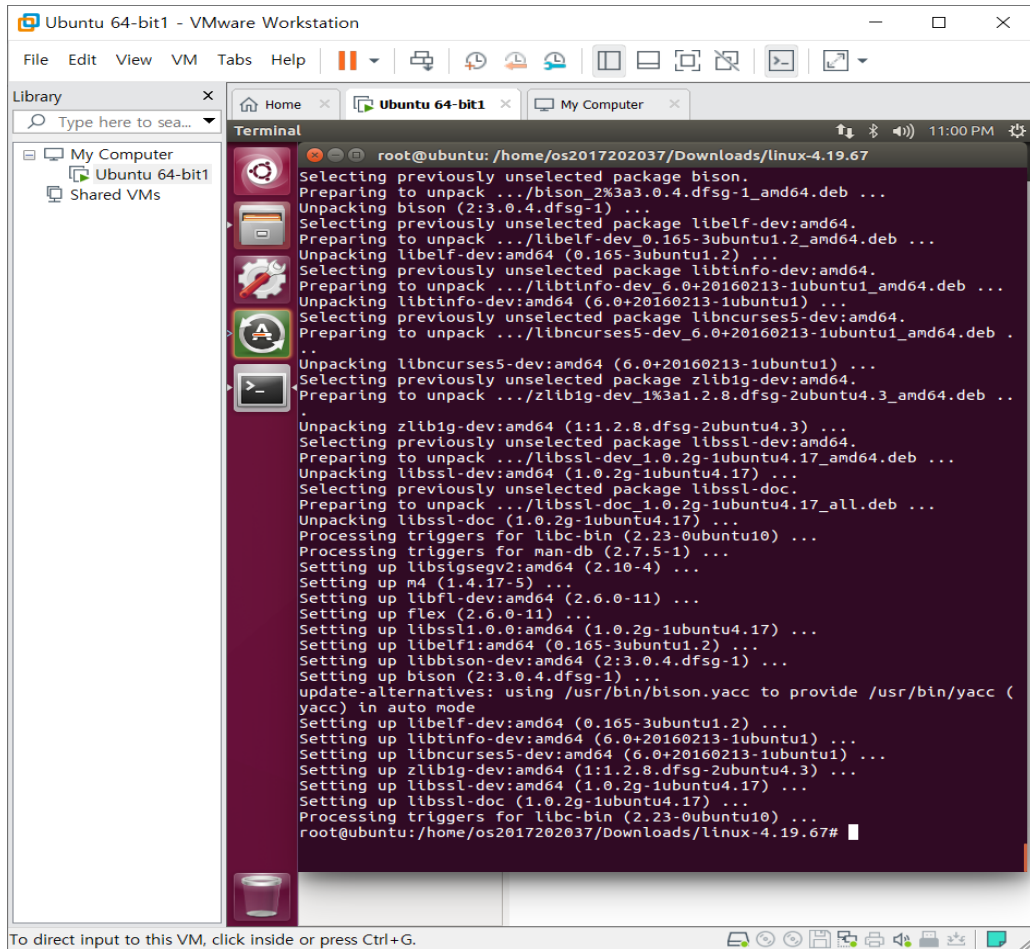


tar Jxvf linux-4.19.67.tar.xz 명령어를 통해 압축이 풀린 모습이다.

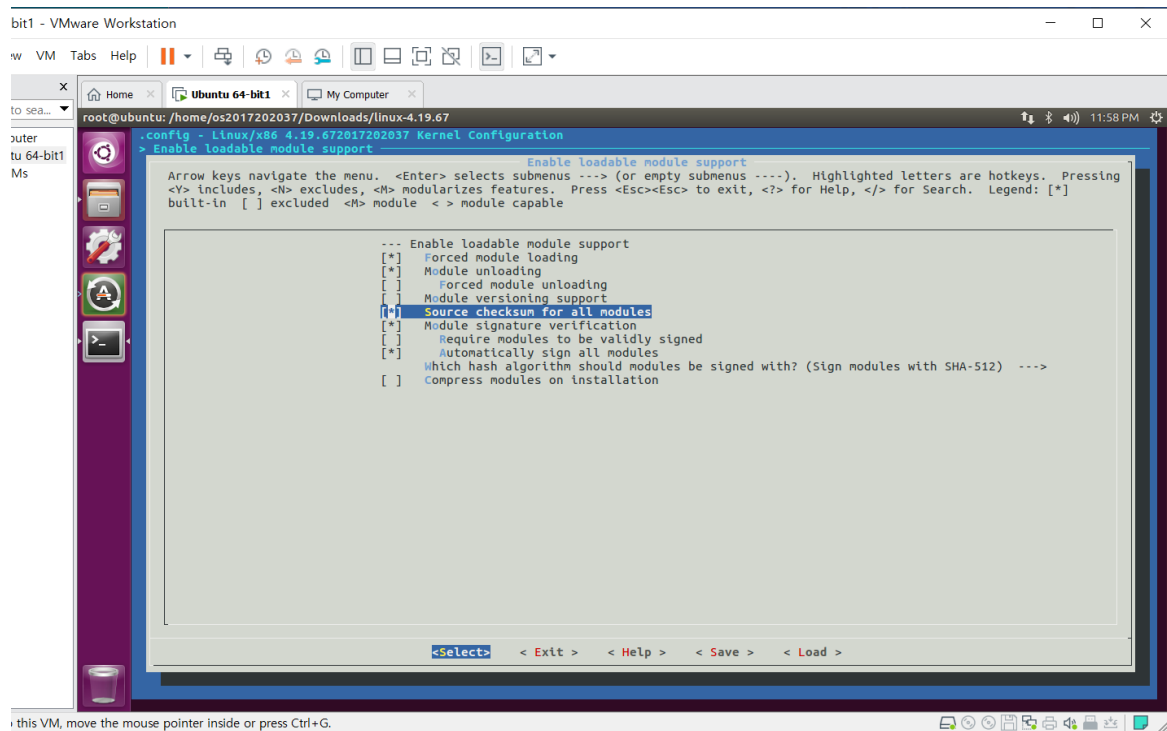


vi 편집기를 이용하여 Makefile을 열어

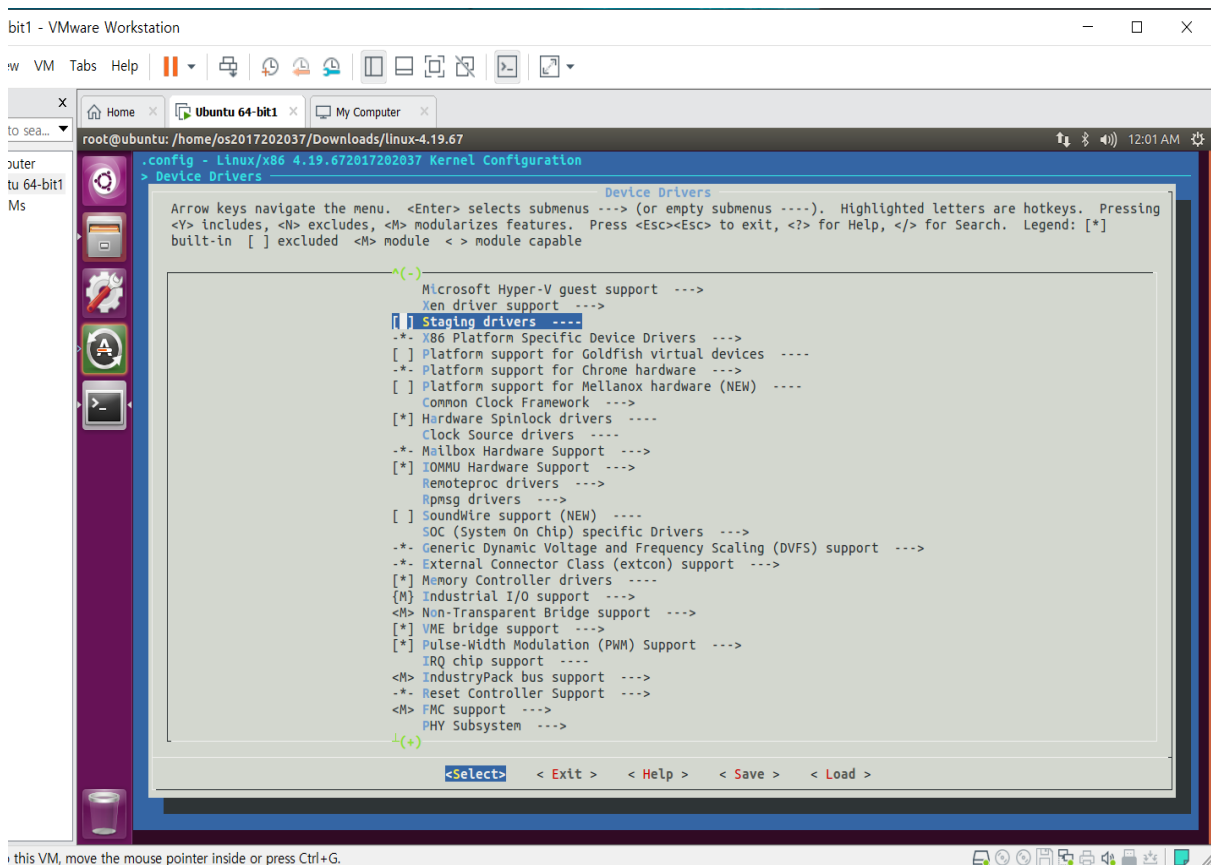
Kernel Extra version을 나의 학번으로 수정하였다.



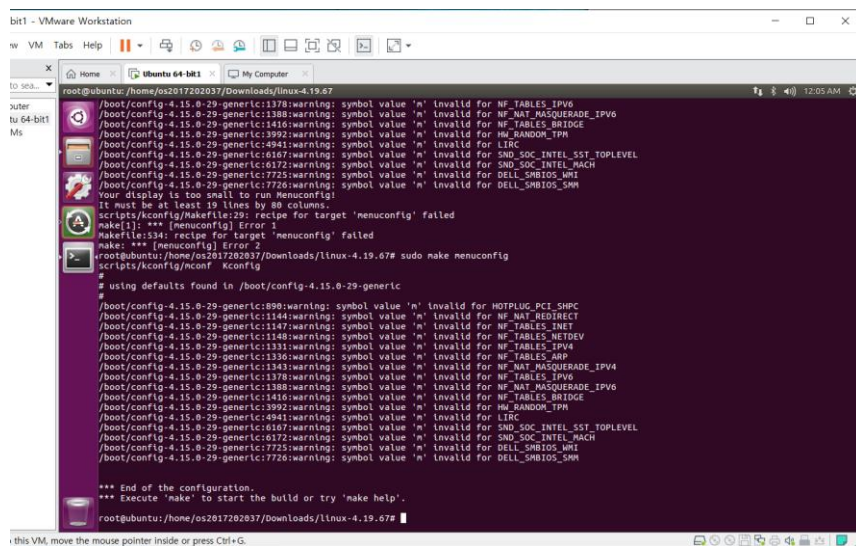
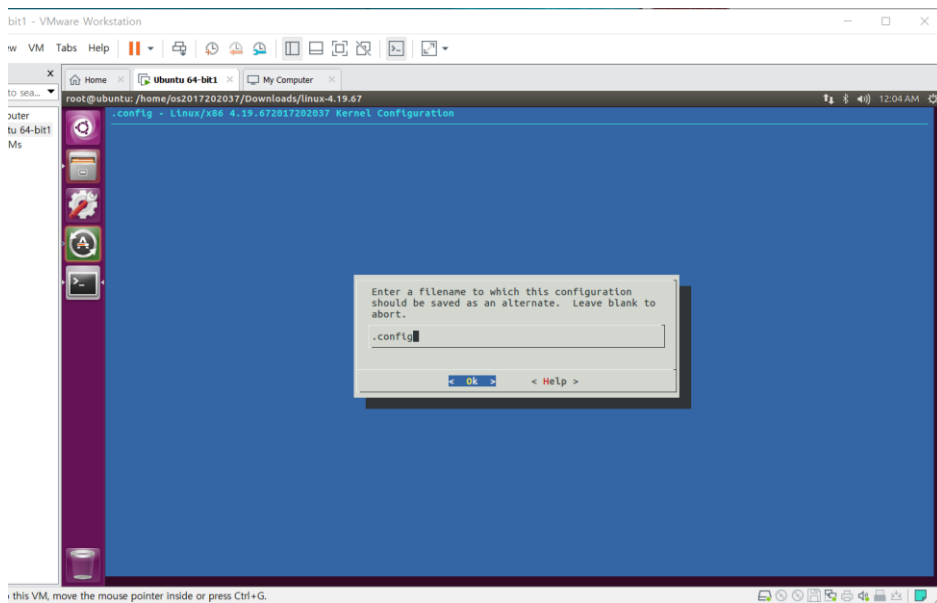
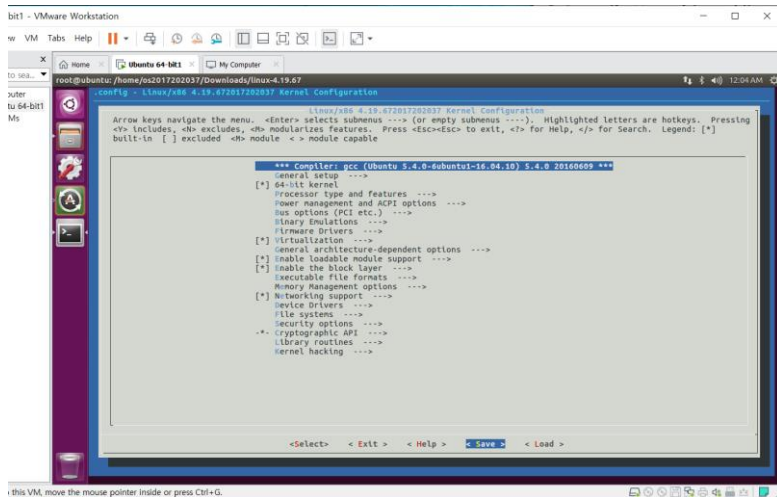
Kernel 환경 설정을 하기 위한 화면이다.



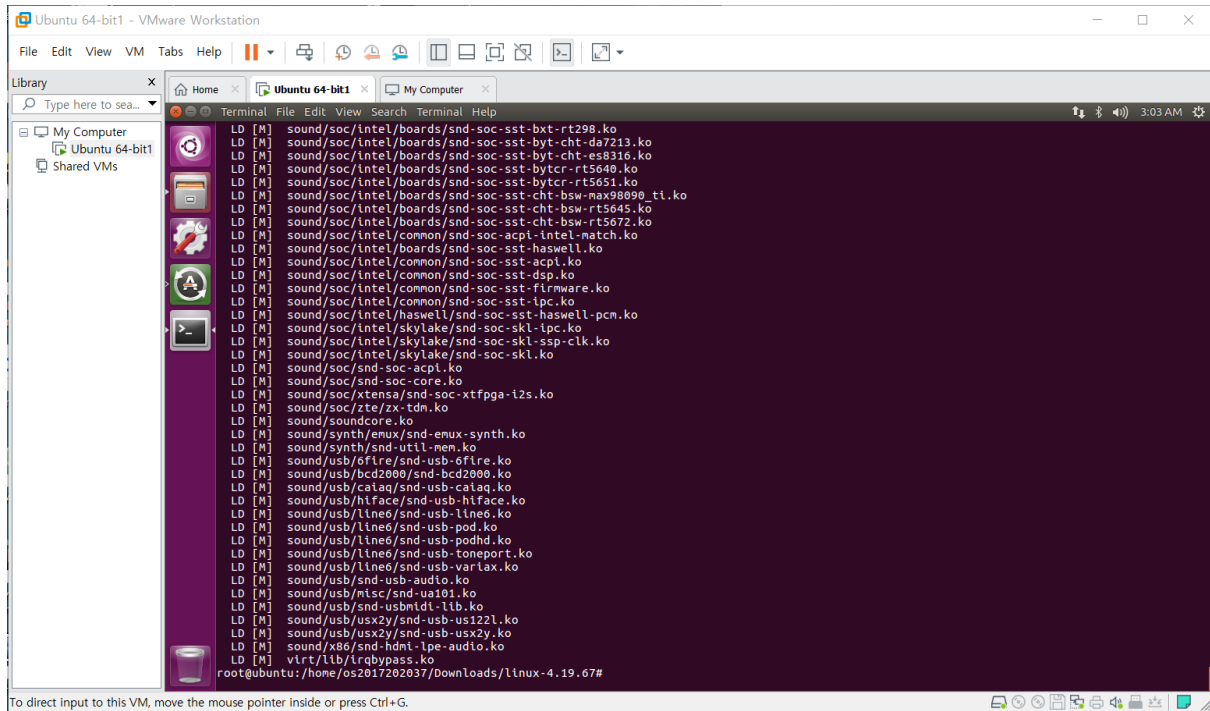
커널 모듈 적재 시 발생할 수 있는 문제를 해결하는 과정이다.



컴파일 시 문제가 될 수 있는 모듈을 제거하는 과정이다.

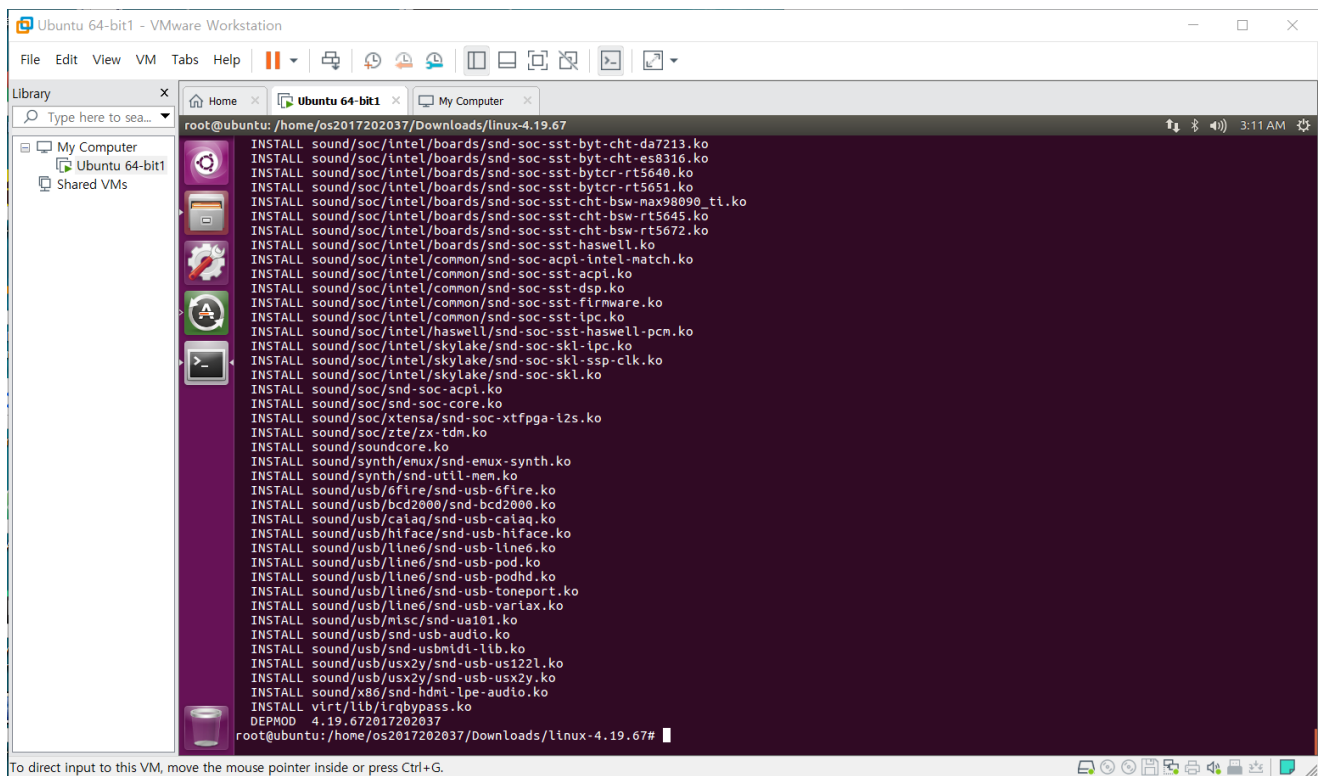


지금까지 설정 한 내용들을 파일(.config)에 저장하는 과정이다.



make -jn 명령어를 이용하여 Kernel 컴파일을 완료한 모습이다.

가상 머신에 할당된 CPU core 수는 4개임으로 thread의 수는 이에 2배인 8로 하였다.



make modules_install 명령어를 이용하여 Module install을 끝 마친 화면이다.

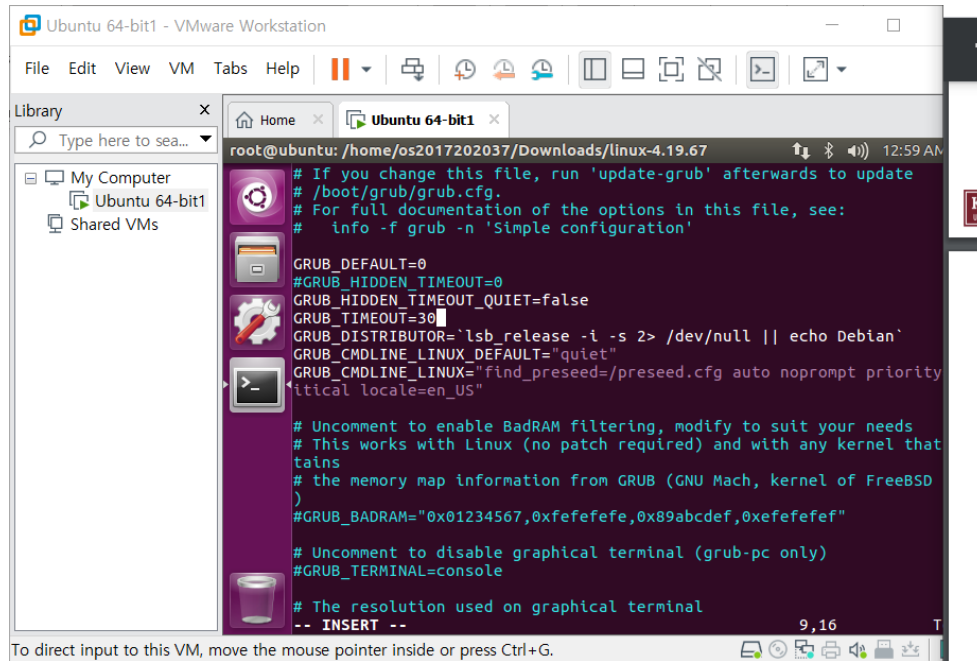
```
root@ubuntu: /home/os2017202037/Downloads/linux-4.19.67
os2017202037@ubuntu: ~$ sudo su
[sudo] password for os2017202037:
root@ubuntu: /home/os2017202037# make install\
> 1
make: *** No rule to make target 'install'. Stop.
root@ubuntu: /home/os2017202037# make install
make: *** No rule to make target 'install'. Stop.
root@ubuntu: /home/os2017202037# ls
Desktop Downloads Music Public Videos
Documents examples.desktop Pictures Templates
root@ubuntu: /home/os2017202037# cd Downloads
root@ubuntu: /home/os2017202037/Downloads# ls
linux-4.19.67 linux-4.19.67.tar.xz
root@ubuntu: /home/os2017202037/Downloads# cd linux-4.19.67
root@ubuntu: /home/os2017202037/Downloads/linux-4.19.67# make install
sh ./arch/x86/boot/install.sh 4.19.672017202037 arch/x86/boot/bzImage \
System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.19.672017202
037 /boot/vmlinuz-4.19.672017202037
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.19.6720172020
37 /boot/vmlinuz-4.19.672017202037
update-initramfs: Generating /boot/initrd.img-4.19.672017202037

run-parts: executing /etc/kernel/postinst.d/pm-utils 4.19.672017202037 /boo
t/vmlinuz-4.19.672017202037
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 4.19.672017
202037 /boot/vmlinuz-4.19.672017202037
run-parts: executing /etc/kernel/postinst.d/update-notifier 4.19.6720172020
37 /boot/vmlinuz-4.19.672017202037
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 4.19.67201720203
7 /boot/vmlinuz-4.19.672017202037
Generating grub configuration file ...
Warning: Setting GRUB_TIMEOUT to a non-zero value when GRUB_HIDDEN_TIMEOUT
is set is no longer supported.
Found linux image: /boot/vmlinuz-4.19.672017202037
Found initrd image: /boot/initrd.img-4.19.672017202037
Found linux image: /boot/vmlinuz-4.15.0-29-generic
Found initrd image: /boot/initrd.img-4.15.0-29-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
root@ubuntu: /home/os2017202037/Downloads/linux-4.19.67#
root@ubuntu: /home/os2017202037/Downloads/linux-4.19.67#
```

make install 명령어를 이용하여

Compile된 Kernel을 Boot Loader에 등록하는 과정이다.

kernel image, System Map을 /boot로 복사한다.



vi /etc/default/grub 명령어를 이용하여 Grub 설정 파일을 수정하였다.

GRUB_HIDDEN_TIMEOUT_QUIET를 false로 하였는데 이는 Grub 메뉴를 숨기지 않는다는 뜻이고 이에 따라 GRUB_HIDDEN_TIMEOUT 설정도 주석 처리 해주었다.

```
root@ubuntu:/home/os2017202037# uname -r
4.19.67-2017202037
```

reboot 명령어로 우분투를 재부팅 한 다음 uname -r 명령어를 이용하여
커널 버전을 확인하는 모습이다.

< Reference >

<https://whackurtistory.com/112> 압축 풀기 tar.xz

<https://norux.me/13> vi 오류 해결

<https://m.blog.naver.com/PostView.nhn?blogId=glryd2&logNo=110189400123&proxyReferer=https:%2F%2Fwww.google.com%2F> Read only 파일 강제 저장

< Assignment1-2 >

```
root@ubuntu:/home/os2017202037/Downloads/linux-4.19.67# apt install exuberant-ctags
```

ctags 명령어를 사용하기 위해 ctags를 설치하는 과정이다.

```
root@ubuntu:/home/os2017202037/Downloads/linux-4.19.67# ctags -R

root@ubuntu:/home/os2017202037/Downloads/linux-4.19.67#
root@ubuntu:/home/os2017202037/Downloads/linux-4.19.67# ls
arch          drivers      kernel       Module.symvers  tags
block         firmware    lib          net             tools
built-in.a    fs          LICENSES     README         usr
certs         include     MAINTAINERS  samples        virt
COPYING       init        Makefile     scripts        vmlinux
CREDITS       ipc         mm           security       vmlinux-gdb.py
crypto        Kbuild     modules.builtin  sound         vmlinux.o
Documentation Kconfig    modules.order  System.map
```

명령어 ctags는 source code를 위한 tag file을 생성해준다.

```
root@ubuntu:/home/os2017202037/Downloads/linux-4.19.67# apt install cscope
```

cscope를 설치해주는 모습이다.

```
root@ubuntu:/home/os2017202037/Downloads/linux-4.19.67
C symbol: start_kernel

  File      Function      Line
0 process.c  <global>          16 extern void start_kernel(void );
1 start_kernel.h <global>         11 extern asmlinkage void __init
    start_kernel(void );
2 bootp.c    start_kernel      135 start_kernel(void )
3 bootpz.c   start_kernel      263 start_kernel(void )
4 main.c     start_kernel      152 void start_kernel(void )
5 relocate.c relocate_kernel    305 void *kernel_entry =
    start_kernel;

* Lines 1-7 of 14, 8 more - press the space bar to display more *
Find this C symbol:
Find this global definition:
Find functions called by this function:
Find functions calling this function:
Find this text string:
Change this text string:
Find this egrep pattern:
Find this file:
Find files #including this file:
Find assignments to this symbol:
```

Cscope를 이용하여 start_kernel이 있는 위치를 찾는 모습이다.

```

C symbol: start_kernel

  File      Function      Line
0 head32.c  i386_start_kernel      56 start_kernel();
1 head64.c  x86_64_start_reservatio 472 start_kernel();
2 main.c    start_kernel            531 asmlinkage __visible void
                                   __init start_kernel(void )

* Press the space bar to display the first lines again *

Find this C symbol:
Find this global definition:
Find functions called by this function:
Find functions calling this function:
Find this text string:
Change this text string:
Find this egrep pattern:
Find this file:
Find files #including this file:
Find assignments to this symbol:

```

main.c 를 찾아 코드로 진입해 보았다.

```

pr_notice("%s", linux_banner);
printk(KERN_INFO"2017202037_OSLAB\n");
Files setup_arch(&command_line);
/*
 * Set up the the initial canary and entropy after arch
 * and after adding latent and command line entropy.
 */
add_latent_entropy();
add_device_randomness(command_line, strlen(command_line));
boot_init_stack_canary();
mm_init_cpumask(&init_mm);
setup_command_line(command_line);
setup_nr_cpu_ids();
setup_per_cpu_areas();
smp_prepare_boot_cpu(); /* arch-specific boot-cpu hooks */
boot_cpu_hotplug_init();

build_all_zonelists(NULL);
page_alloc_init();

pr_notice("Kernel command line: %s\n", boot_command_line);
/* parameters may set static keys */
553,2-9 47%

```

이렇게 start_kernel 함수 안에 코드를 추가하였는데

수정한 부분은 linux/init/main.c 파일에서 asmlinkage __visible void __init
start_kernel(void) 함수이다.

이 부분에 코드를 추가한 이유는 linux의 init 폴더에서 kernel을 초기화 하는 과
정이 이루어지기 때문이다.

또한 사진상에 보이는 pr_notice("%s", linux_banner) 이 부분이
리눅스의 버전 정보를 출력하는 부분이기 때문에 그 밑에 바로 학번이 나오도록
하기 위해 저 위치에 코드를 작성하였다.


```
root@ubuntu:/home/os2017202037/Downloads/linux-4.19.67# make
```

```
root@ubuntu:/home/os2017202037/Downloads/linux-4.19.67# make modules_install
```

```
DEPMOD 4.19.67-2017202037
```

```
root@ubuntu:/home/os2017202037/Downloads/linux-4.19.67# make install
```

```
done
```

```
root@ubuntu:/home/os2017202037/Downloads/linux-4.19.67# reboot
```

Module compile 후 reboot하는 과정이다.

```
os2017202037@ubuntu:~$ dmesg
[ 0.000000] Linux version 4.19.67-2017202037 (root@ubuntu) (gcc version 5.4.0
20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.12)) #3 SMP Tue Sep 22 00:16:33 PDT 2020
[ 0.000000] 2017202037_OSLAB
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.19.67-2017202037 root=UU
ID=84b873c5-e6a6-401e-af9a-7c7ff689ce91 ro find_preseed=/preseed.cfg auto noprom
pt priority=critical locale=en_US quiet
[ 0.000000] KERNEL supported cpus:
[ 0.000000] Intel GenuineIntel
[ 0.000000] AMD AuthenticAMD
[ 0.000000] Centaur CentaurHauls
[ 0.000000] Disabled fast string operations
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point regi
```

dmesg 명령어를 통해 printk()로 출력한 내용을 확인할 수 있다.

< Reference >

<https://extrememanager.net/26777> 우분투 APT Could not get lock /var/lib/dpkg/lock
오류 해결 방법

<https://wiki.kldp.org/KoreanDoc/html/EmbeddedKernel-KLDP/x1942.html> 리눅스
커널 부팅

< Conclusion >

<analysis>

리눅스 커널 컴파일 하는데 굉장히 오랜 시간이 걸리기 때문에 커널을 작업 하는 데 있어서 신중함이 필수적이다. 또한 리눅스가 GUI가 윈도우에 비해 잘 되어 있지 않기 때문에 이번 과제에 사용한 cscope, ctags와 같은 function들을 사용하는 것에 익숙해지는 것이 중요하다.

<고찰>

이번 과제를 하면서 운영체제의 속을 살짝 들여다 볼 수 있었는데, 굉장히 복잡했다. 평소 아무렇지 않게 사용하던 운영체제의 코드를 직접 수정 하려 하니 생각해야 될 부분이 한 두가지가 아니었다. 이렇게 간단한 수정 작업을 하는데도 너무 어려웠기 때문에 공부가 더 많이 필요하다고 느꼈다.