운영체제 실습

[Assignment#1]

Class : 목34

Professor : 최상호 교수님

Student ID : 2021202003

Name : 강준우

Introduction

이번 과제에서는 1-1부터 1-3까지 각각 순서대로 과제하는 모습이 나열되어 있다.

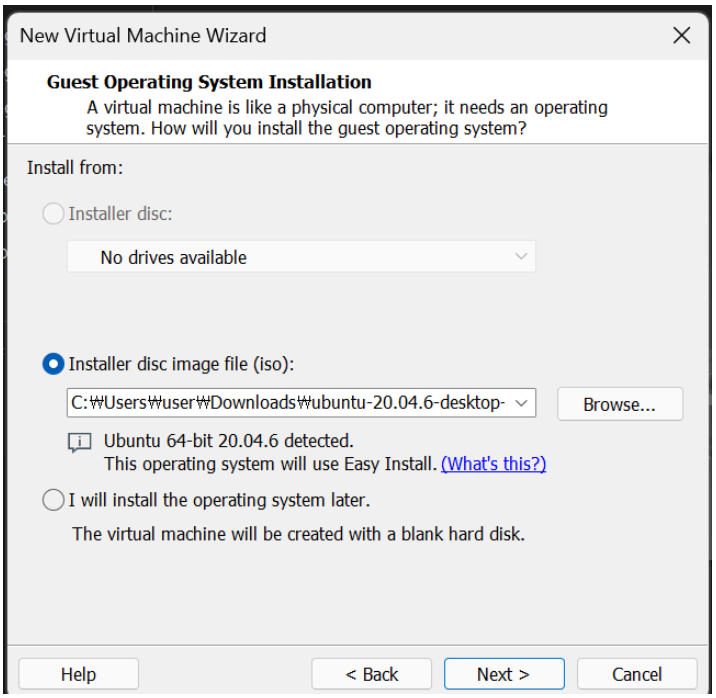
1-1에서는 vmware를 통한 우분투 설치를 목표로 한다.

1-2에서는 해당 Kernel 5.4.282 Compile을 설치 및 실행하도록 한다.

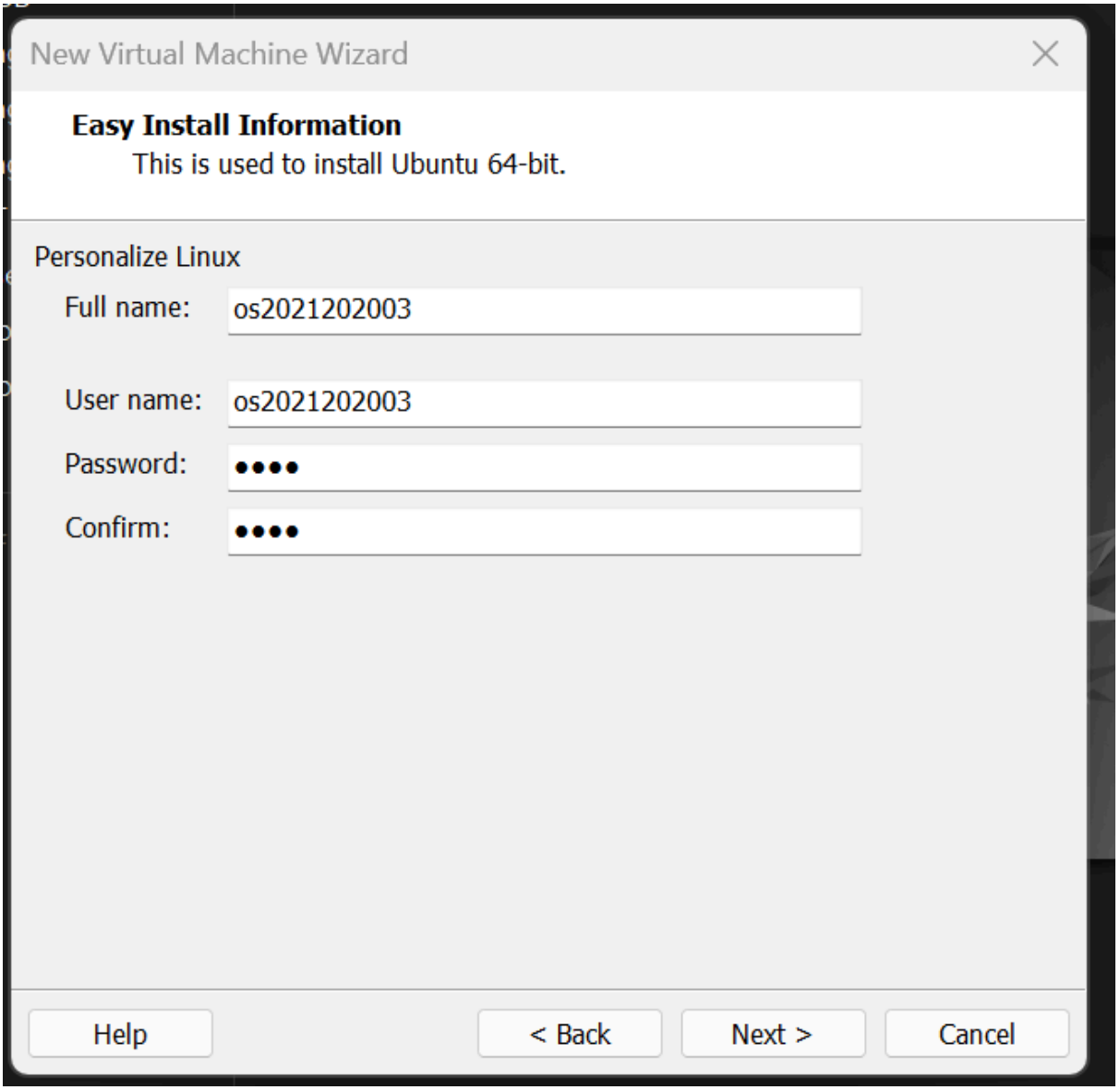
1-3에서는 운영체제 내부로 들어가 코드를 수정해본다.

해당 과정들을 수행하면서 우분투 내용을 수정하고 접근하며 조작하는 방법들을 자연스럽게 익힌다.

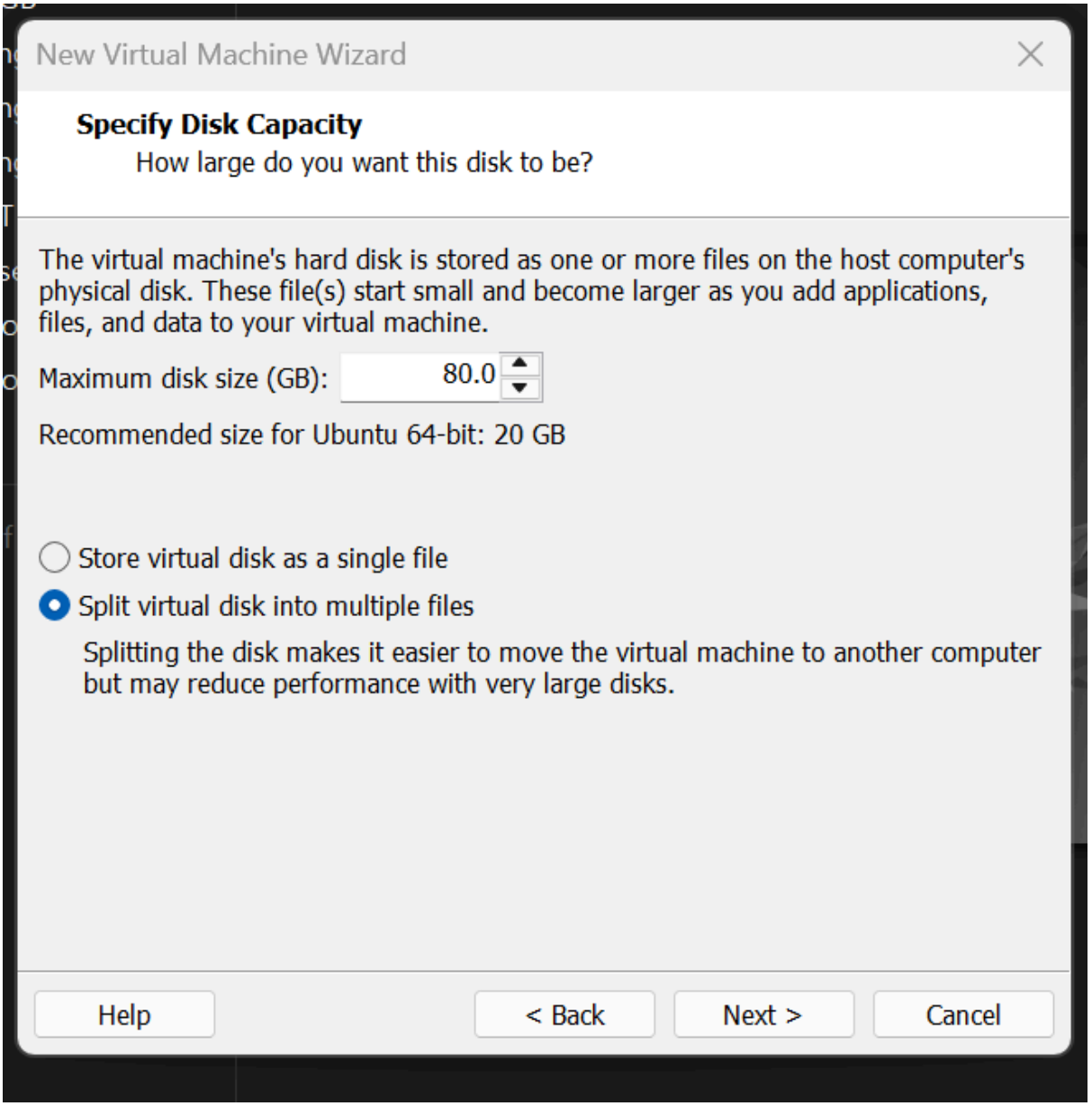
결과화면



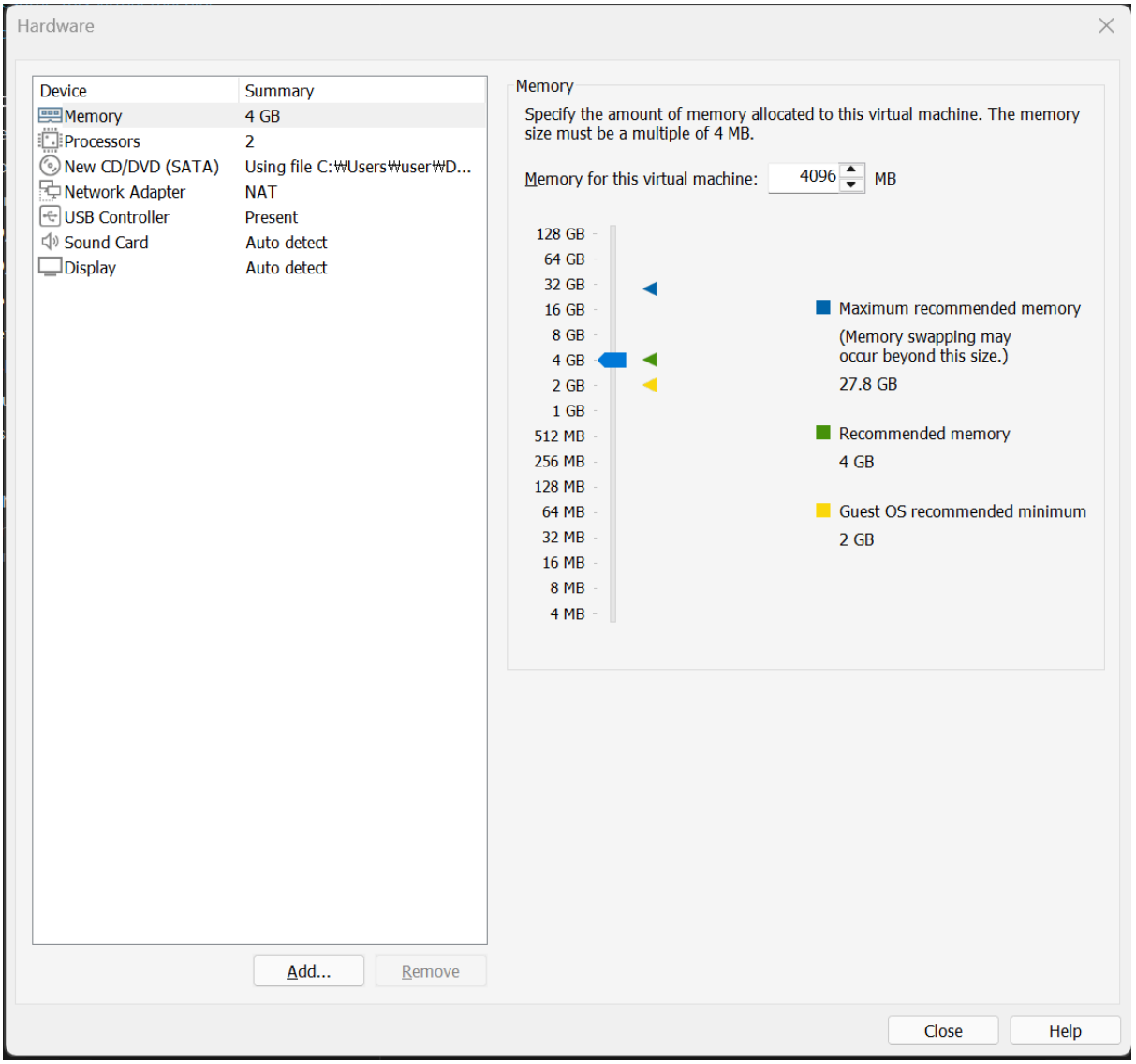
20.04.6버전 우분투 선택

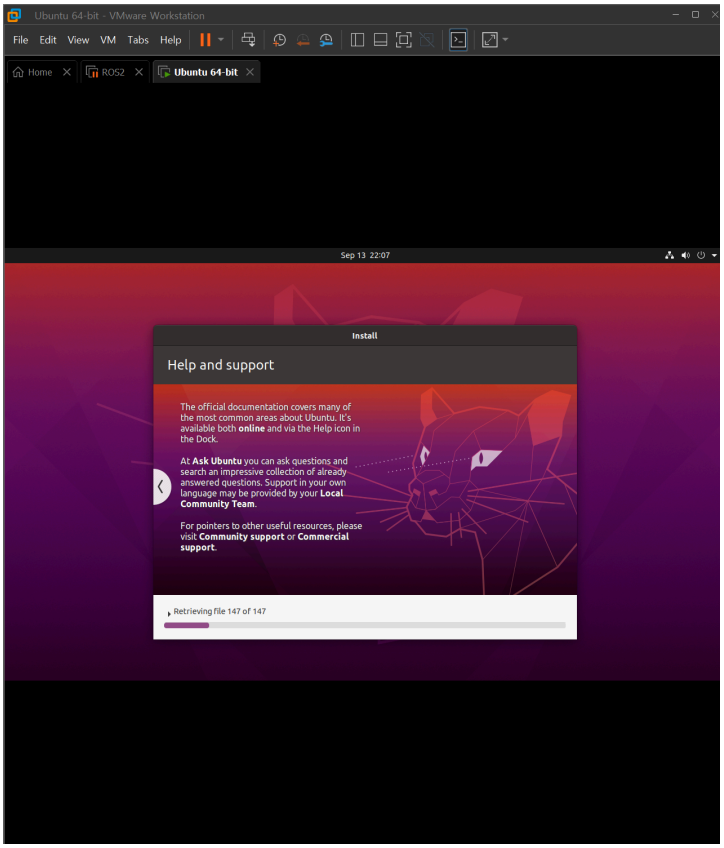

New Virtual Machine Wizard
Easy Install Information
This is used to install Ubuntu 64-bit.
Personalize Linux
Full name:
os2021202003
User name:
os2021202003
Password:
Confirm:
Help
< Back
Next >
Cancel

이름 설정 모습

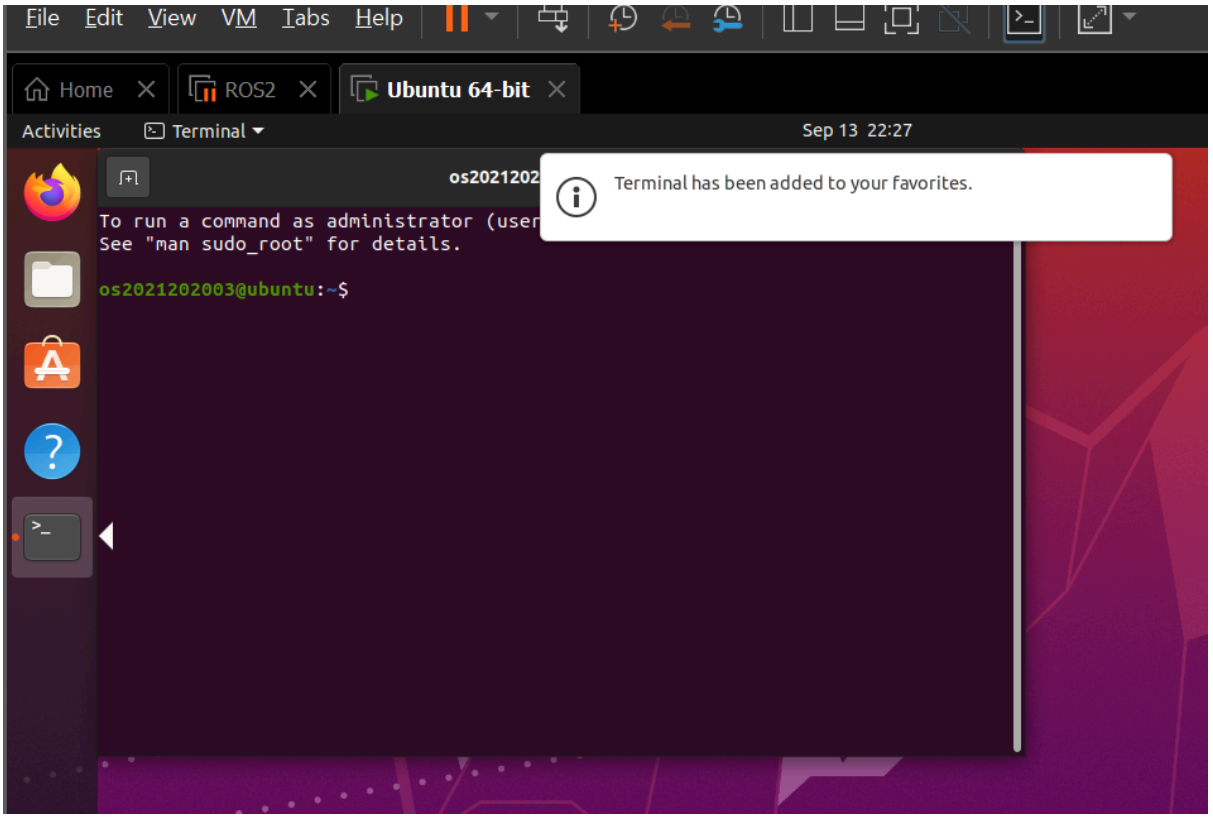

New Virtual Machine Wizard
Specify Disk Capacity
How large do you want this disk to be?
The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.
Maximum disk size (GB):
80.0
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

디스크 용량 설정 모습

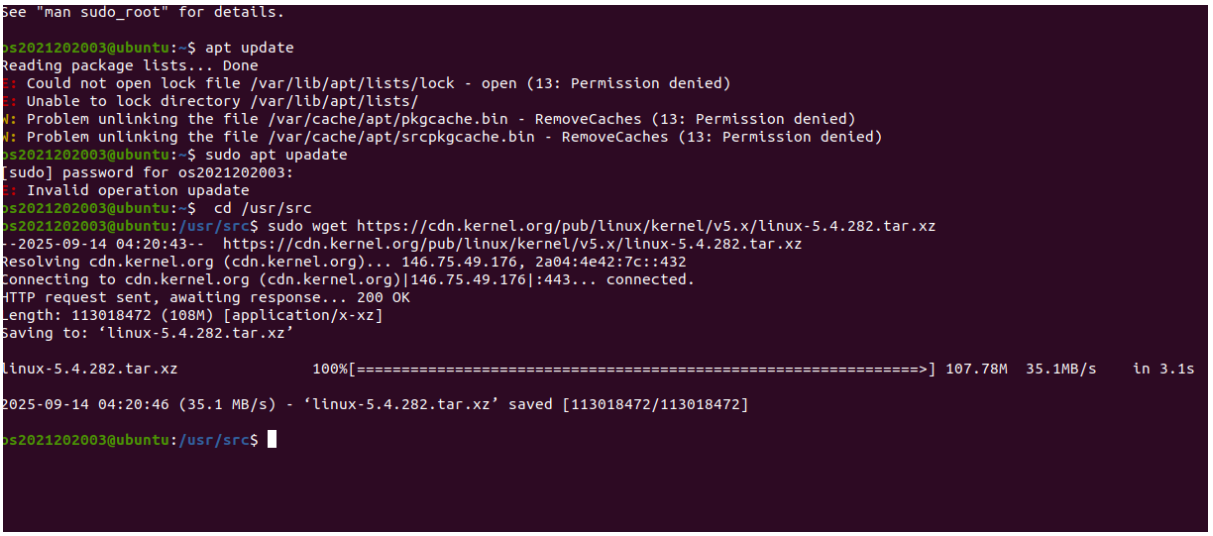

Hardware
Device
Summary
Memory
4 GB
Processors
2
New CD/DVD (SATA)
Using file C:₩Users₩user₩D...
Network Adapter
NAT
USB Controller
Present
Sound Card
Auto detect
Display
Auto detect
Add...
Remove
Memory
Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.
Memory for this virtual machine:
4096
MB
128 GB
64 GB
32 GB
16 GB
8 GB
4 GB
2 GB
1 GB
512 MB
256 MB
128 MB
64 MB
32 MB
16 MB
8 MB
4 MB
Maximum recommended memory
(Memory swapping may occur beyond this size.)
27.8 GB
Recommended memory
4 GB
Guest OS recommended minimum
2 GB
Close
Help



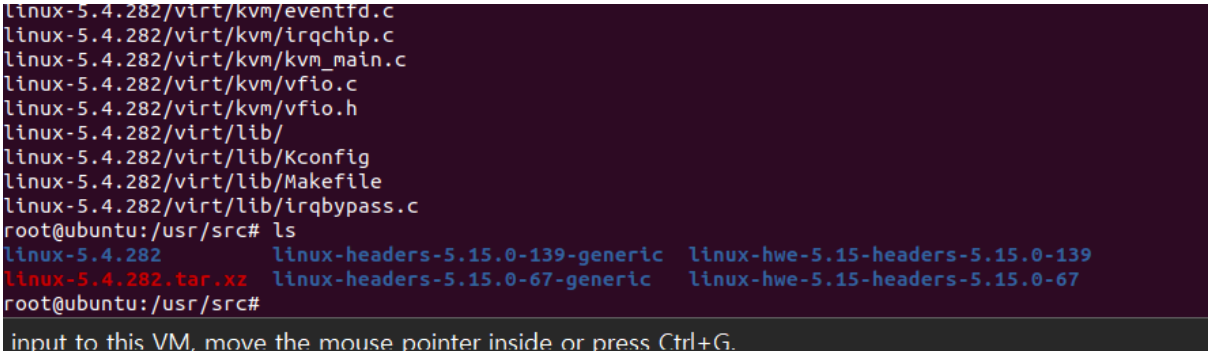
설치가 자동으로 되고 있는 모습



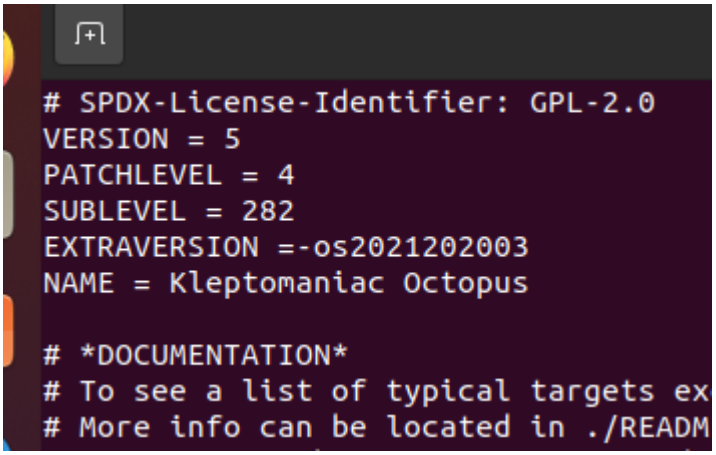
잘 설치되어 터미널을 연 모습



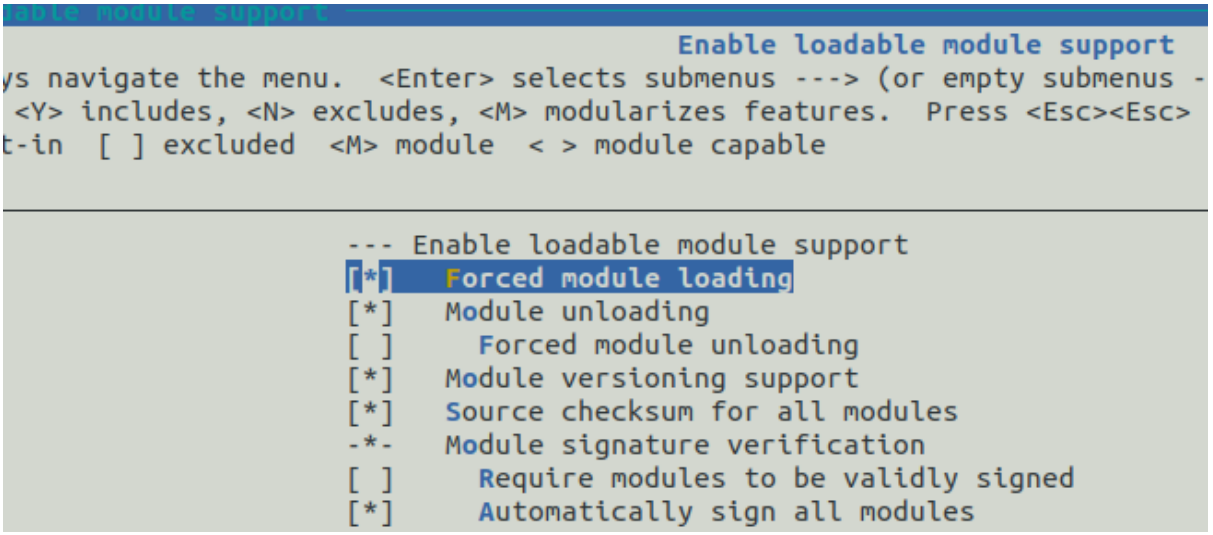
sudo wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.4.282.tar.xz한 모습

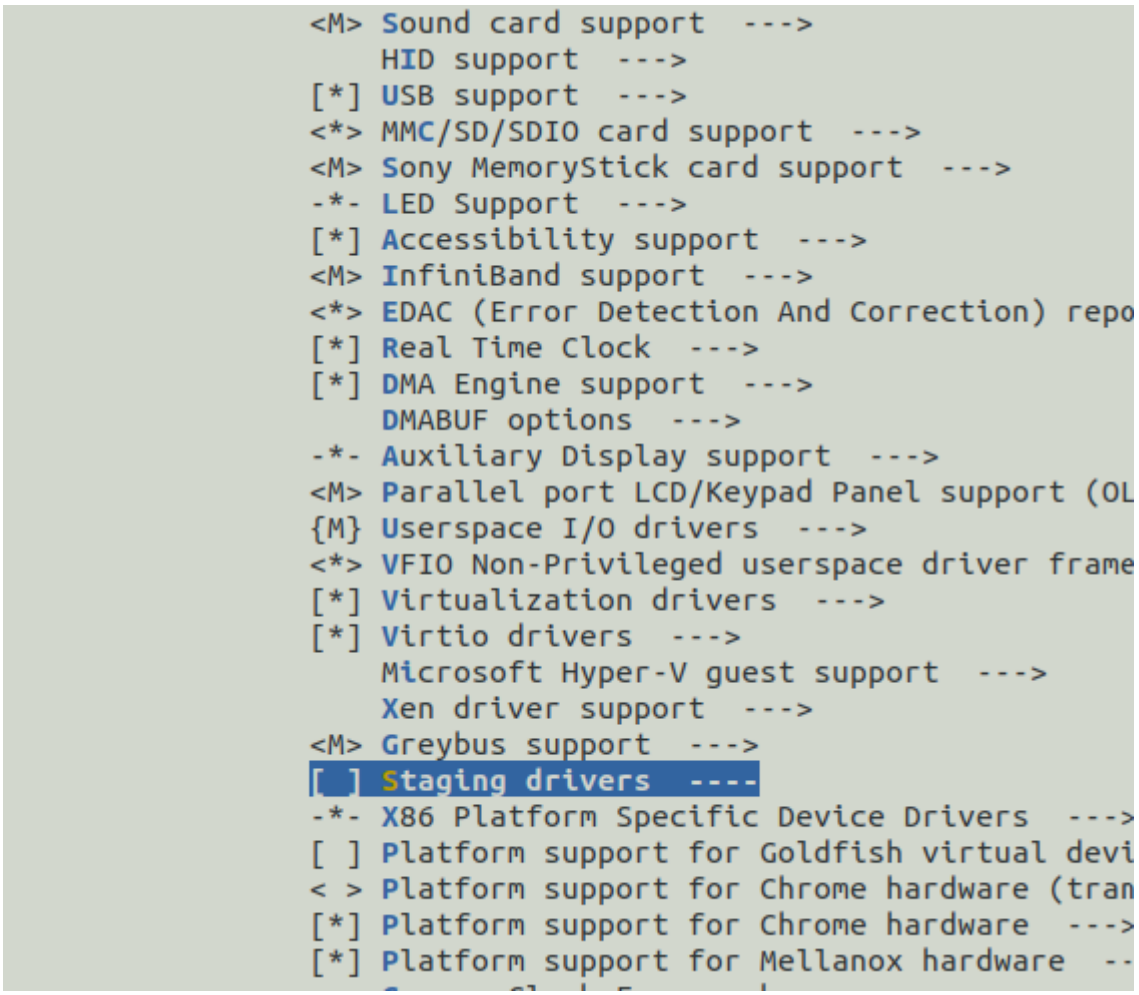


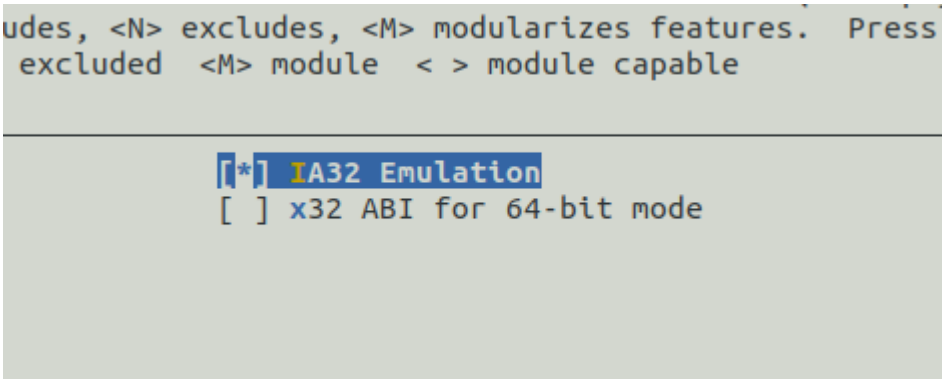
집파일 오픈한 모습



Makefile 수정한 모습


Enable loadable module support
ys navigate the menu. <Enter> selects submenus ---> (or empty submenus -
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc>
t-in [ ] excluded <M> module < > module capable
--- Enable loadable module support
[*] Forced module loading
[*] Module unloading
[ ] Forced module unloading
[*] Module versioning support
[*] Source checksum for all modules
-*- Module signature verification
[ ] Require modules to be validly signed
[*] Automatically sign all modules

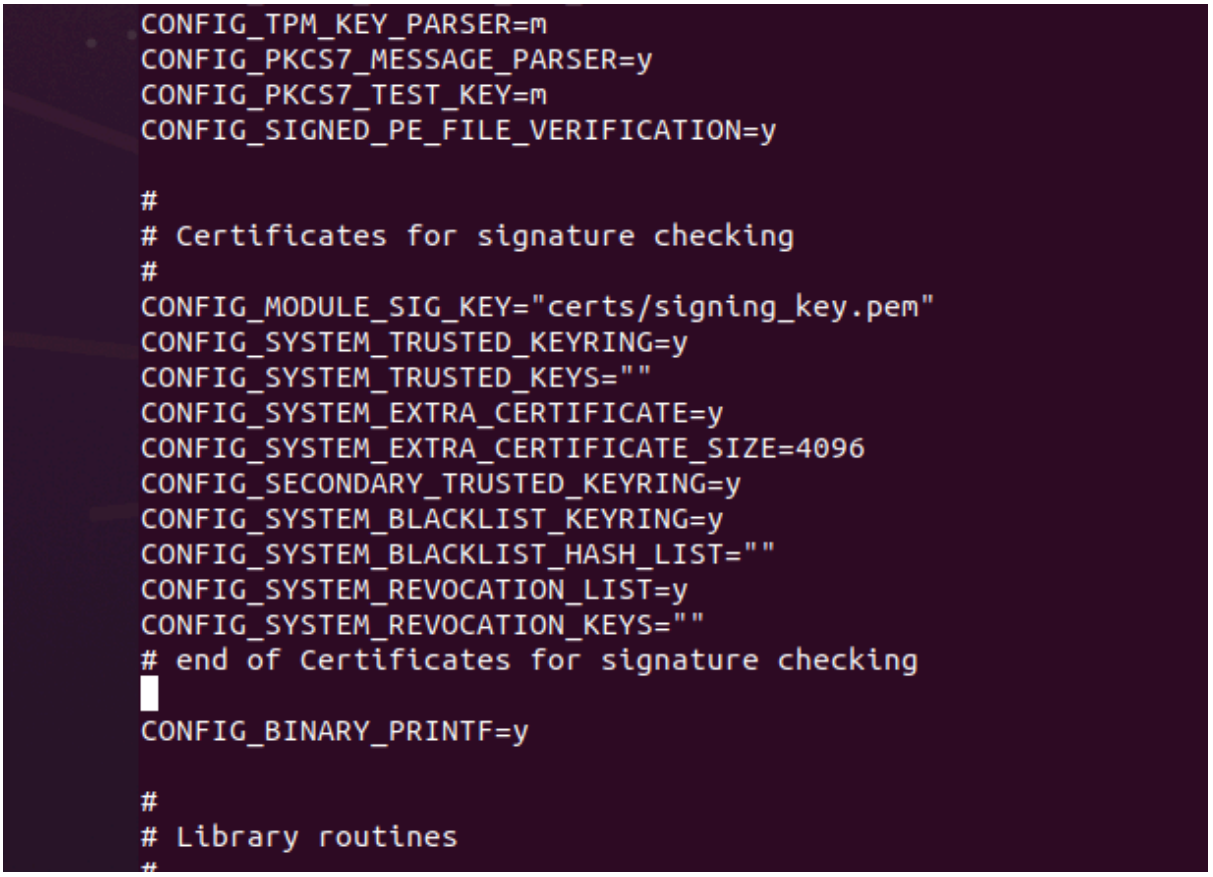

<M> Sound card support --->
HID support --->
[*] USB support --->
<*> MMC/SD/SDIO card support --->
<M> Sony MemoryStick card support --->
-*- LED Support --->
[*] Accessibility support --->
<M> InfiniBand support --->
<*> EDAC (Error Detection And Correction) repo
[*] Real Time Clock --->
[*] DMA Engine support --->
DMABUF options --->
-*- Auxiliary Display support --->
<M> Parallel port LCD/Keypad Panel support (OL
{M} Userspace I/O drivers --->
<*> VFIO Non-Privileged userspace driver frame
[*] Virtualization drivers --->
[*] Virtio drivers --->
Microsoft Hyper-V guest support --->
Xen driver support --->
<M> Greybus support --->
[ ] Staging drivers ----
-*- X86 Platform Specific Device Drivers --->
[ ] Platform support for Goldfish virtual devi
< > Platform support for Chrome hardware (tran
[*] Platform support for Chrome hardware --->
[*] Platform support for Mellanox hardware --



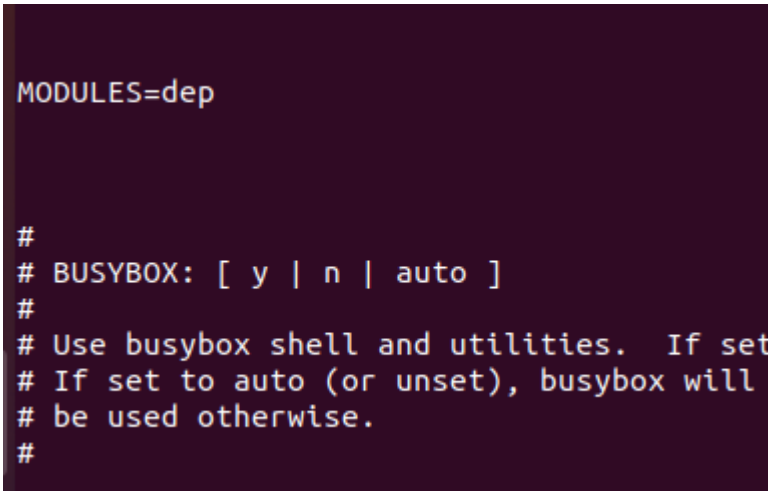
커널 환경설정 모습입니다.

```
root@ubuntu:/usr/src/linux-5.4.282# sudo make menuconfig
scripts/kconfig/mconf  Kconfig
#
# using defaults found in /boot/config-5.15.0-139-generic
#
/boot/config-5.15.0-139-generic:3290:warning: symbol value 'm'
/boot/config-5.15.0-139-generic:3968:warning: symbol value 'm'
/boot/config-5.15.0-139-generic:3969:warning: symbol value 'm'
/boot/config-5.15.0-139-generic:4470:warning: symbol value 'm'
/boot/config-5.15.0-139-generic:5325:warning: symbol value 'm'
/boot/config-5.15.0-139-generic:5557:warning: symbol value 'm'
/boot/config-5.15.0-139-generic:6812:warning: symbol value 'm'
/boot/config-5.15.0-139-generic:8873:warning: symbol value 'm'
/boot/config-5.15.0-139-generic:9958:warning: symbol value 'm'
/boot/config-5.15.0-139-generic:9959:warning: symbol value 'm'


*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
```

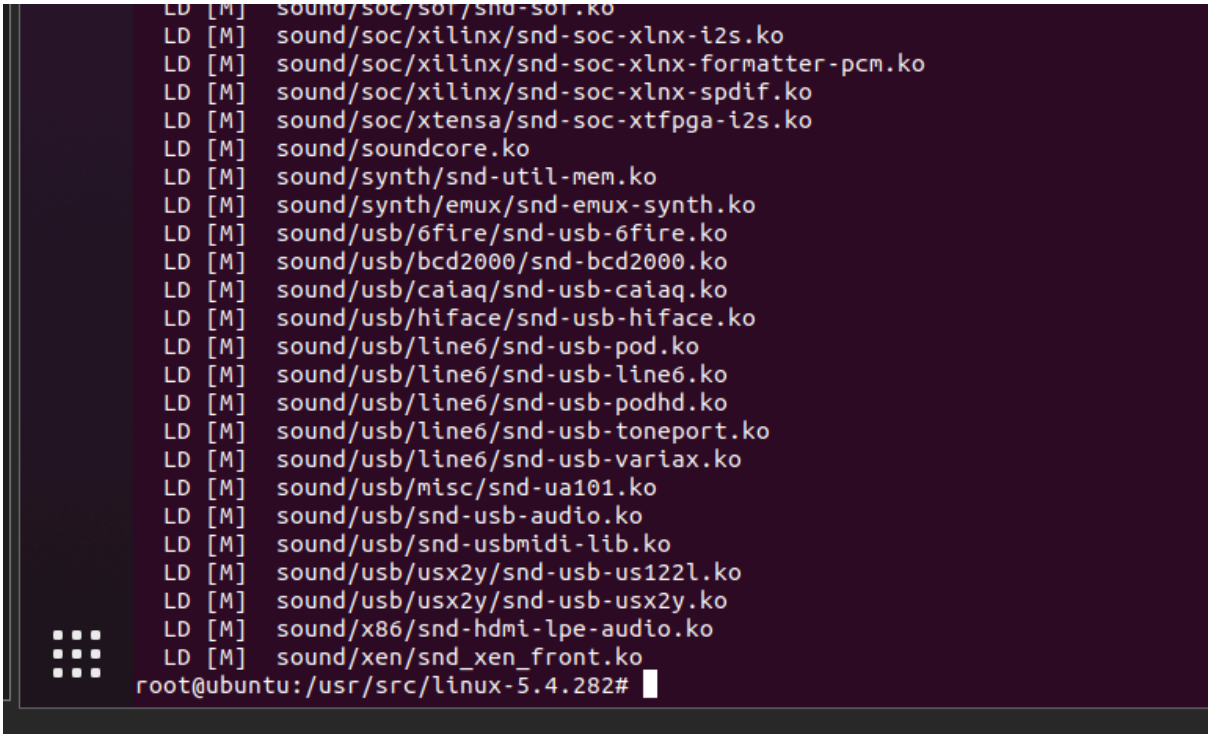
저장하고 나간모습



vi .config 수정모습

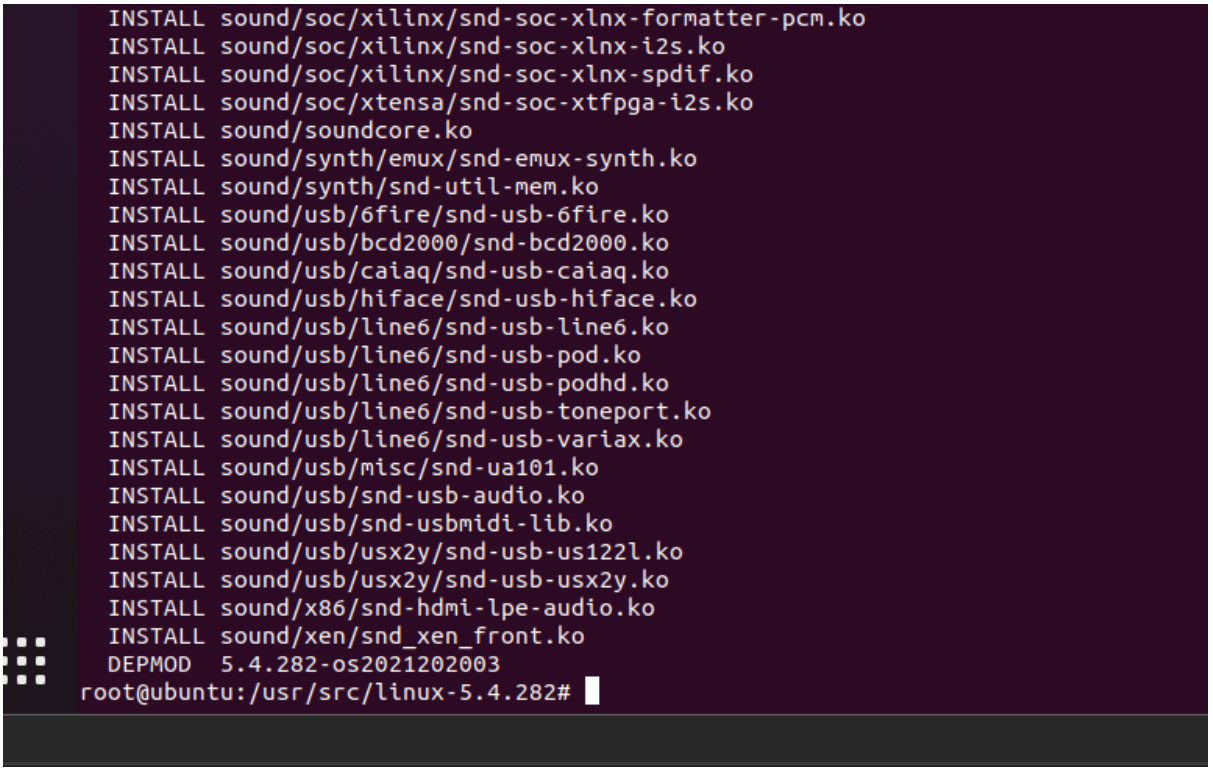


Initramfs.conf 수정모습



sudo make -jN을 수행한 모습

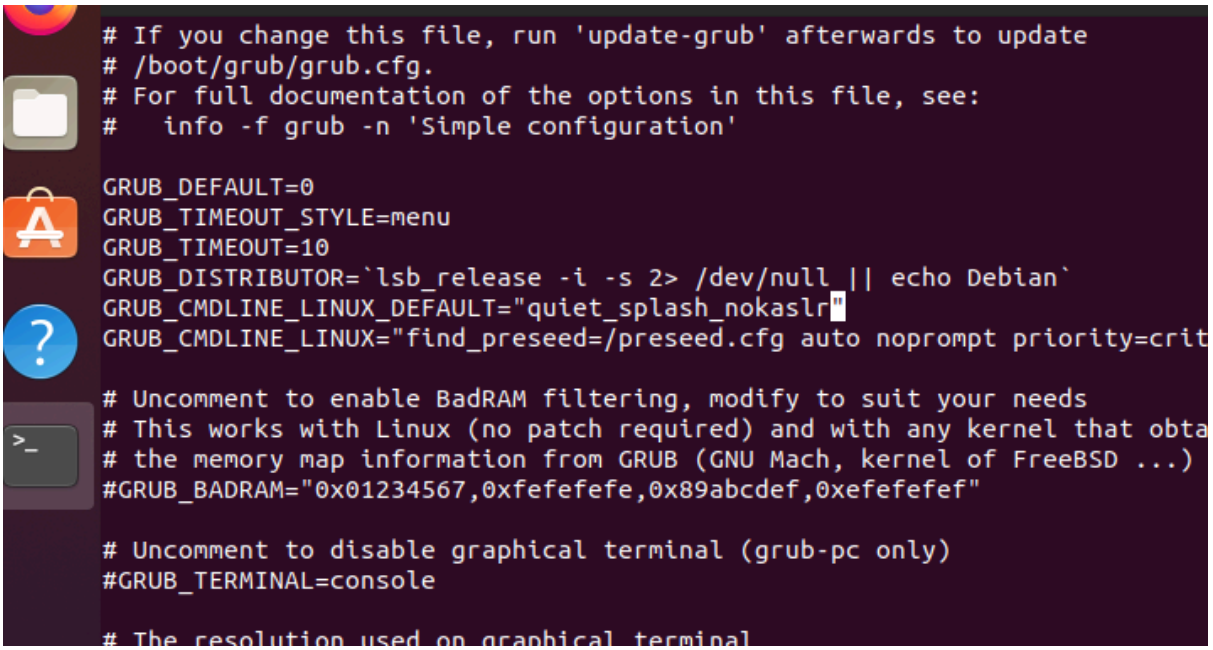
현재 환경설정에서 cpu를 4로 부여해 8로 잡았습니다.



모듈 설치한 모습

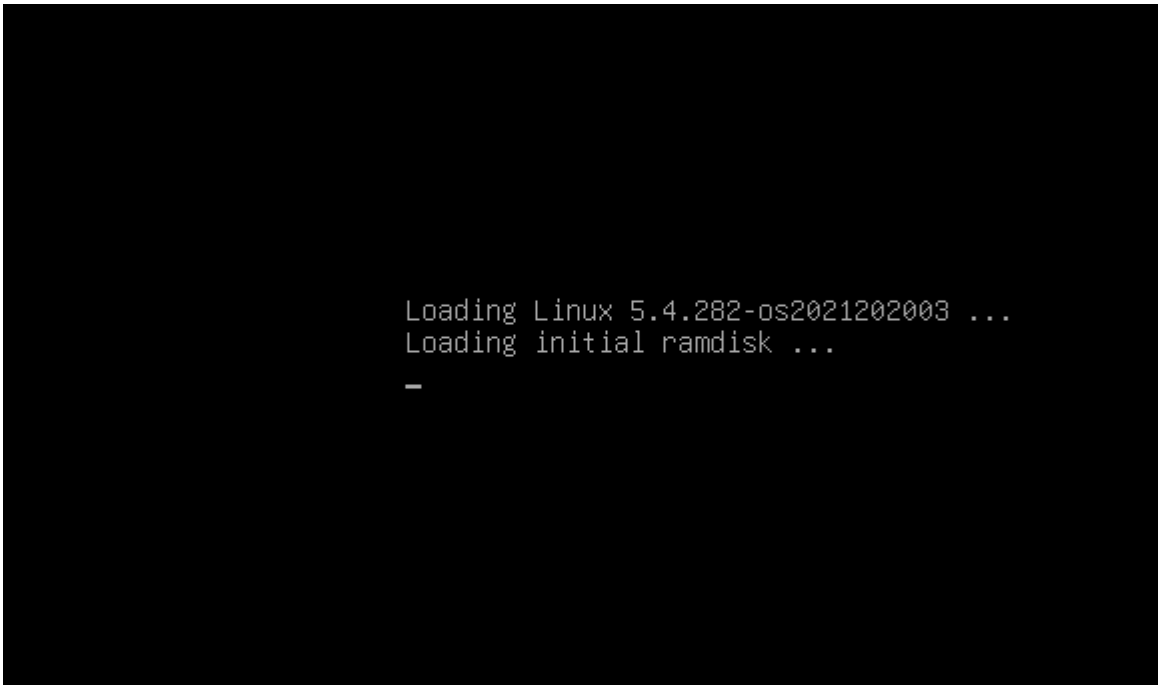
```
root@ubuntu:/usr/src/linux-5.4.282# sudo make install
sh ./arch/x86/boot/install.sh 5.4.282-os2021202003 arch/x86/boot/bzImage \
        System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.4.282-os2021202003 /boot/vmlinuz-5.4.282-os2021202003
update-initramfs: Generating /boot/initrd.img-5.4.282-os2021202003
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 5.4.282-os2021202003 /boot/vmlinuz-5.4.282-os2021202003
run-parts: executing /etc/kernel/postinst.d/update-notifier 5.4.282-os2021202003 /boot/vmlinuz-5.4.282-os2021202003
run-parts: executing /etc/kernel/postinst.d/xx-update-initrd-links 5.4.282-os2021202003 /boot/vmlinuz-5.4.282-os2021202003
I: /boot/initrd.img.old is now a symlink to initrd.img-5.15.0-139-generic
I: /boot/initrd.img is now a symlink to initrd.img-5.4.282-os2021202003
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.4.282-os2021202003 /boot/vmlinuz-5.4.282-os2021202003
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.15.0-139-generic
Found initrd image: /boot/initrd.img-5.15.0-139-generic
Found linux image: /boot/vmlinuz-5.15.0-67-generic
Found initrd image: /boot/initrd.img-5.15.0-67-generic
Found linux image: /boot/vmlinuz-5.4.282-os2021202003
Found initrd image: /boot/initrd.img-5.4.282-os2021202003
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
```

Sudo make install 한 모습



```
root@ubuntu:/home/os2021202003# sudo vi /etc/default/grub
root@ubuntu:/home/os2021202003# sudo update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.15.0-139-generic
Found initrd image: /boot/initrd.img-5.15.0-139-generic
Found linux image: /boot/vmlinuz-5.15.0-67-generic
Found initrd image: /boot/initrd.img-5.15.0-67-generic
Found linux image: /boot/vmlinuz-5.4.282-os2021202003
Found initrd image: /boot/initrd.img-5.4.282-os2021202003
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
root@ubuntu:/home/os2021202003#
```

Grub파일을수정한모습



Reboot후 해당 5.4.282로 접속하는 모습

```
os2021202003@ubuntu:~$ uname -r
5.4.282-os2021202003
os2021202003@ubuntu:~$
```

이후 5.4.282가 불러져오는것을 확인 가능합니다.

```
root@ubuntu:/usr/src/linux-5.4.282# sudo apt install cscope
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  cscope-el
The following NEW packages will be installed:
  cscope
0 upgraded, 1 newly installed, 0 to remove and 66 not upgraded.
Need to get 210 kB of archives.
After this operation, 1,255 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 cscope amd64 15.9-1 [210 kB]
Fetched 210 kB in 2s (104 kB/s)
Selecting previously unselected package cscope.
(Reading database ... 163232 files and directories currently installed.)
Preparing to unpack .../cscope_15.9-1_amd64.deb ...
Unpacking cscope (15.9-1) ...
Setting up cscope (15.9-1) ...
Processing triggers for man-db (2.9.1-1) ...
```

Cscope 설치 후 검색

```
EXPORT_SYMBOL_GPL(agp_remove_bridge);

int agp_off;
int agp_try_unsupported_boot;
EXPORT_SYMBOL(agp_off);
EXPORT_SYMBOL(agp_try_unsupported_boot);

static int __init agp_init(void)
{
        if (!agp_off)
                printk(KERN_INFO "os2021202003_Linux agpgart interface v%d.%d\n",
                                AGPGART_VERSION_MAJOR, AGPGART_VERSION_MINOR);

        return 0;

}

static void __exit agp_exit(void)
{
}

#ifndef MODULE
```

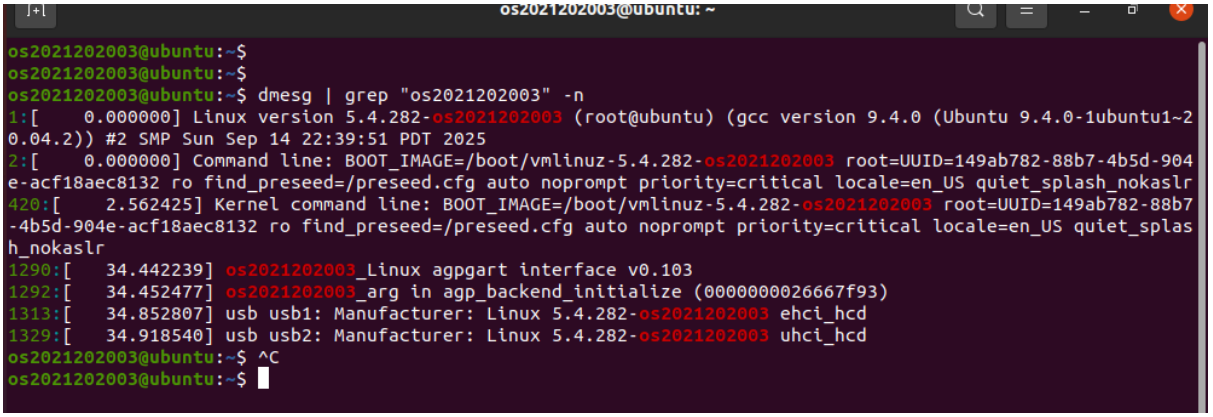
Linux agpgart interface를 검색해 경로가

/usr/src/linux-5.4.282/drivers/char/agp/backend.c임을 확인, 수정합니다.

```
}


static int agp_backend_initialize(struct agp_bridge_data *bridge)
{
        int size_value, rc, got_gatt=0, got_keylist=0;
        printk(KERN_INFO "os2021202003_arg in %s (%p)\n", __func__, bridge);
        bridge->max_memory_agp = agp_find_max();
        bridge->version = &agp_current_version;

        if (bridge->driver->needs_scratch_page) {
                struct page *page = bridge->driver->agp_alloc_page(bridge);

                if (!page) {
                        dev_err(&bridge->dev->dev,
                                "can't get memory for scratch page\n");
                        return -ENOMEM;
                }

                bridge->scratch_page_page = page;
                bridge->scratch_page_dma = page_to_phys(page);
```

이후 펑선 기반 함수를 찾아 그중 하나인 agp_backend_initialize함수의 내부에 출력문을 추가 구현해두었습니다.

출력 내용 수정한 모습



Grep을 이용해 1290 1292번에서 변경되어 있는 모습을 확인 가능하다. 0000000026667f93는 해당 function agr이다

고찰

개인적으로 make하는 과정에서 처음에 cpu를 기본 2로 잡고 돌렸더니 켜두고 자고 일어났는데도 터미널이 끝나지 않아서 매우 당황했다. Cpu를 4로 잡은 후 N을8로 했을때도 2시간 30분정도가 지나는 모습인데 자료에서 이정도로 시간이 많이 걸린다고 언급을 해주지 않아서 문제가 생긴줄알고 몇차례 강제 중지를하고 가상환경을 재시작한

기억이 난다. 또한 grep를 사용해서 확인하려할때 바꾼 2줄말고도 나머지 줄들이 출력이 되어서 당황했는데 이후에 추가로 검색하여 1, 2, 420라인은 커스텀 커널 문자열과 커맨드라인이 수정한 커널로 부팅한것을 확인되는 라인이었고, 나머지 밑에 라인 2개는 루트 허브가 초기화되면서 찍힌 로그임을 알게 되었다. 해당 요소들이 뭔지 추가적으로 알아보게되는 계기가 되었다.

Reference

실습자료 이용했습니다. 감사합니다.