

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.

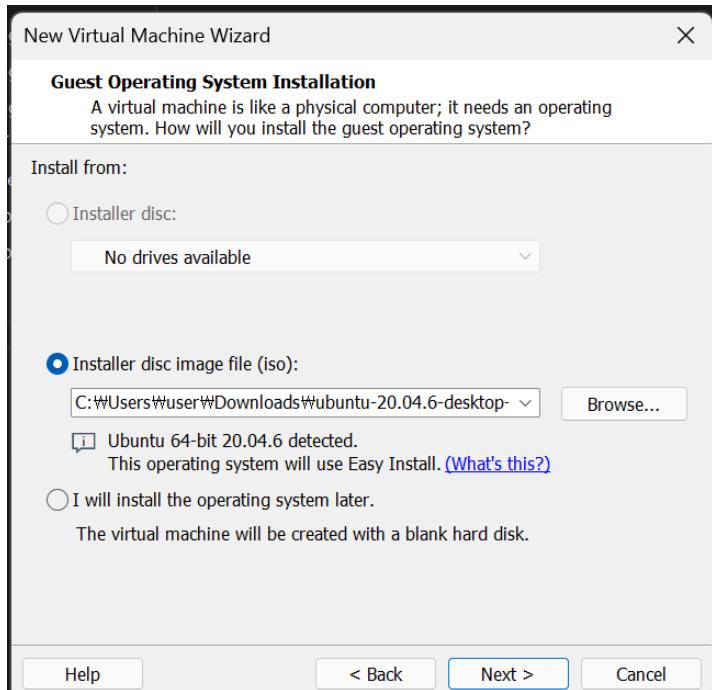
1-1에서는 vmware를 통한 우분투 설치를 목표로 한다.

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

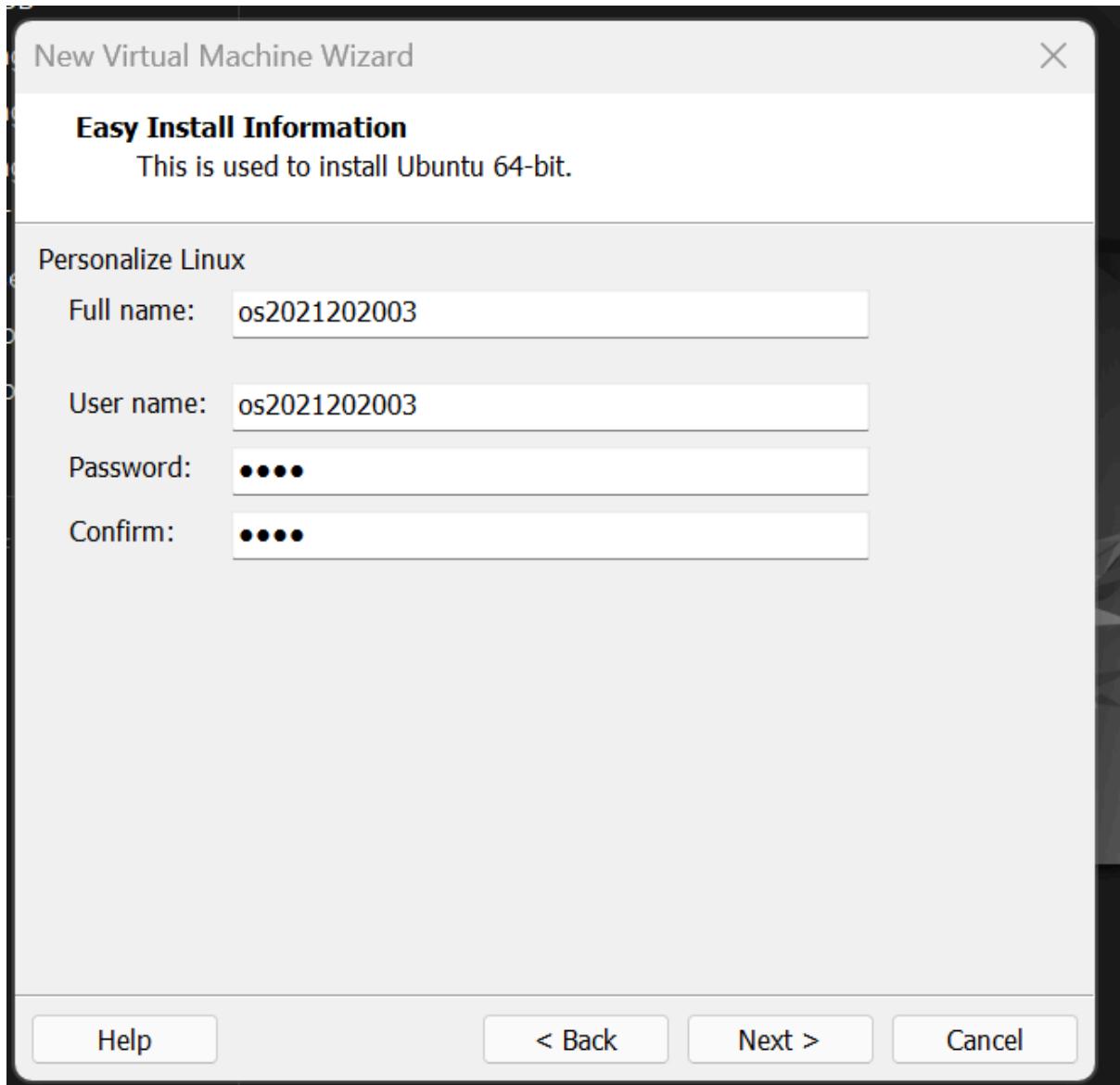
1-3에서는 운영체제 내부로 들어가 코드를 수정해본다.

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

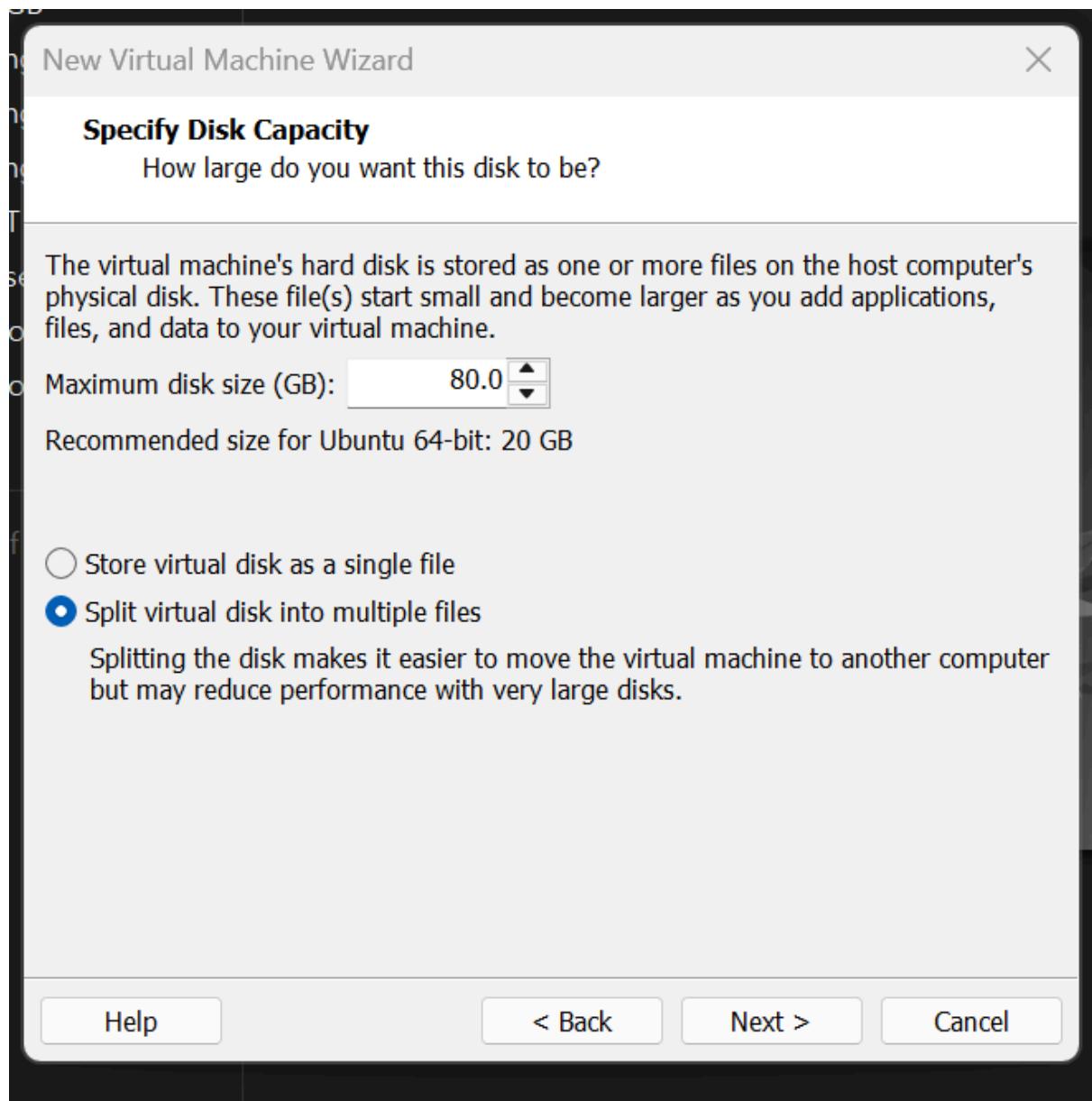
# 결과화면



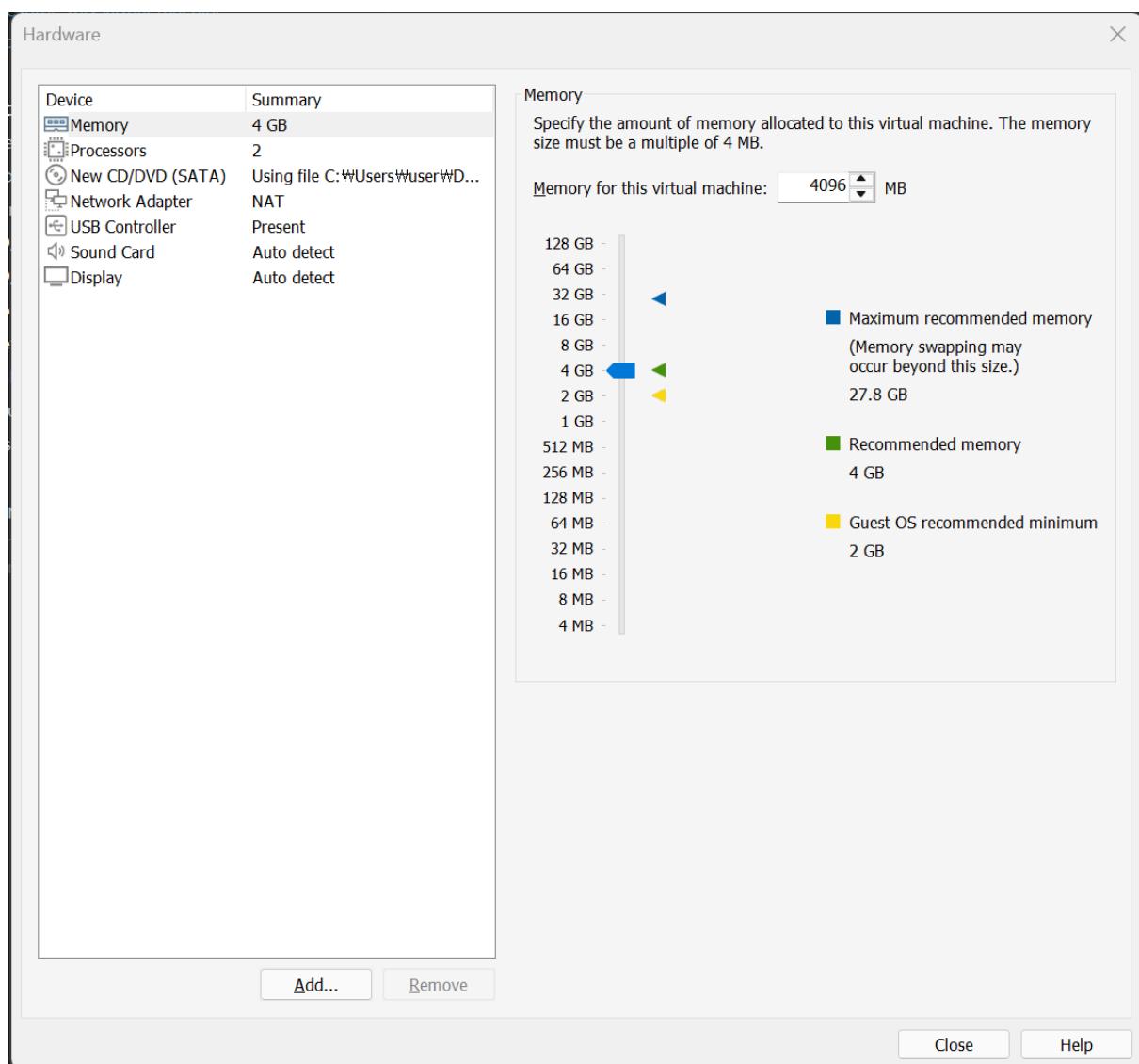
20.04.6버전 우분투 선택

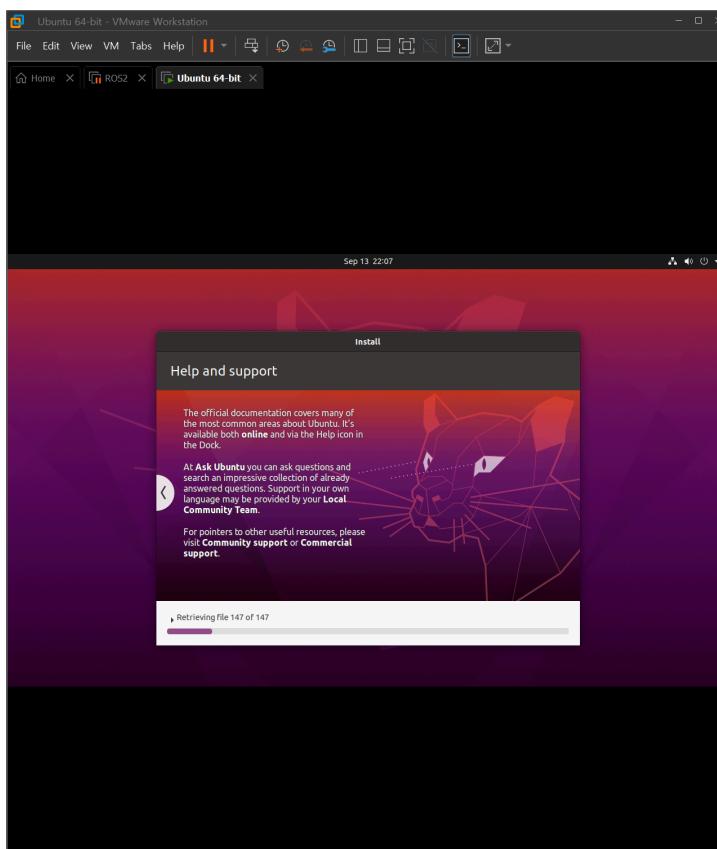


이름 설정 모습

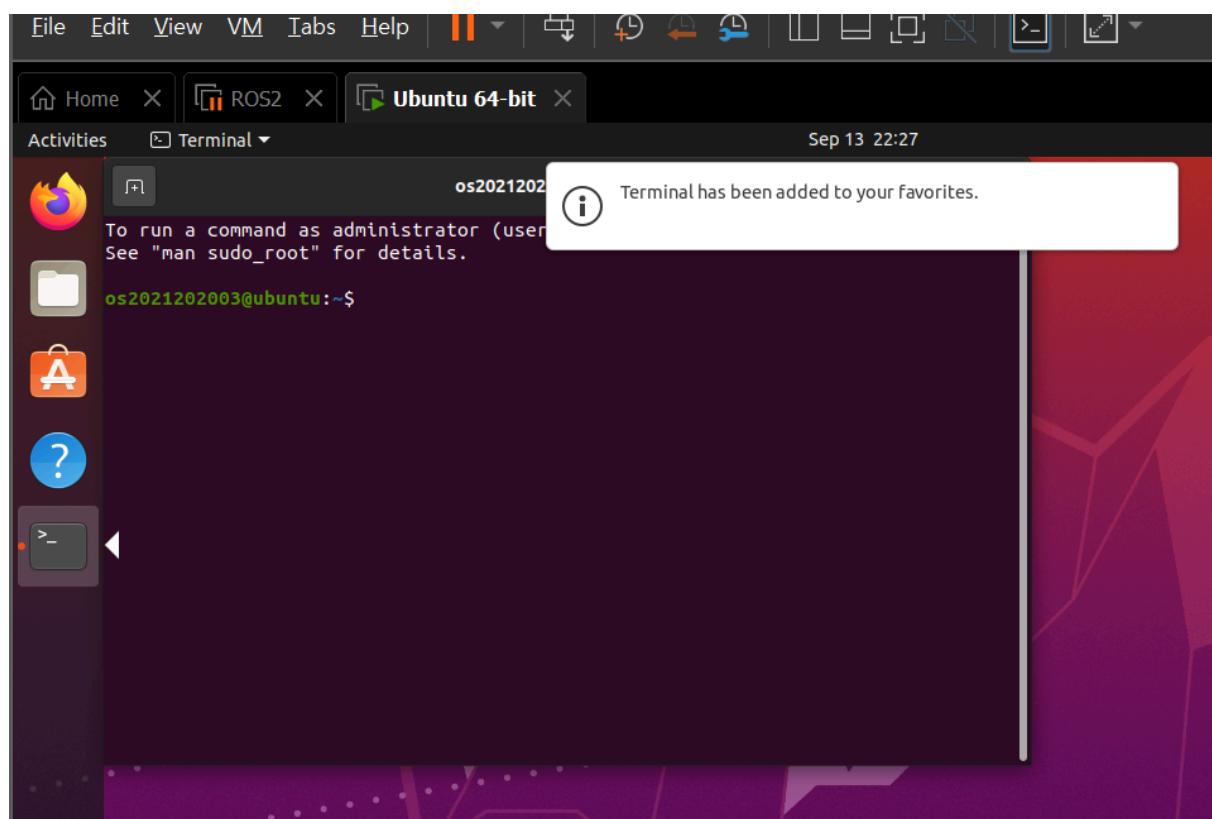


디스크 용량 설정 모습





설치가 자동으로 되고 있는 모습



## 잘 설치되어 터미널을 연 모습

```
See "man sudo_root" for details.

os2021202003@ubuntu:~$ apt update
Reading package lists... Done
E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)
E: Unable to lock directory /var/lib/apt/lists/
N: Problem unlinking the file /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permission denied)
N: Problem unlinking the file /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permission denied)
os2021202003@ubuntu:~$ sudo apt update
[sudo] password for os2021202003:
E: Invalid operation update
os2021202003@ubuntu:~$ cd /usr/src
os2021202003@ubuntu:/usr/src$ sudo wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.4.282.tar.xz
--2025-09-14 04:20:43-- https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.4.282.tar.xz
Resolving cdn.kernel.org (cdn.kernel.org)... 146.75.49.176, 2a04:4e42:7c::432
Connecting to cdn.kernel.org (cdn.kernel.org)|146.75.49.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 113018472 (108M) [application/x-xz]
Saving to: 'linux-5.4.282.tar.xz'

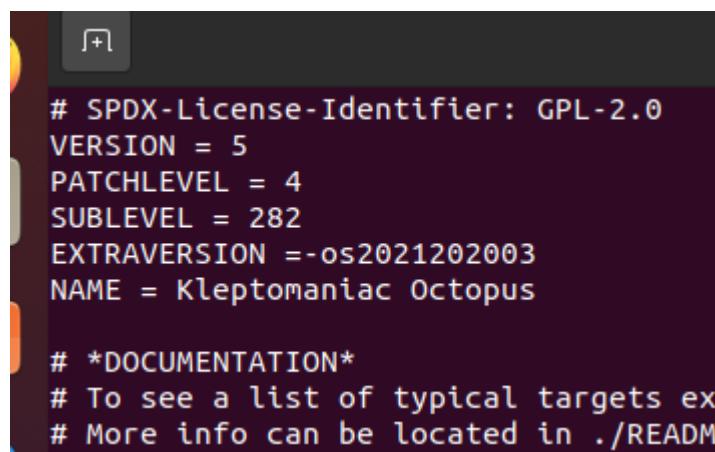
linux-5.4.282.tar.xz      100%[=====] 107.78M  35.1MB/s   in 3.1s
2025-09-14 04:20:46 (35.1 MB/s) - 'linux-5.4.282.tar.xz' saved [113018472/113018472]

os2021202003@ubuntu:/usr/src$
```

sudo wget <https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.4.282.tar.xz>한 모습

```
linux-5.4.282/virt/kvm/eventfd.c
linux-5.4.282/virt/kvm/irqchip.c
linux-5.4.282/virt/kvm/kvm_main.c
linux-5.4.282/virt/kvm/vfio.c
linux-5.4.282/virt/kvm/vfio.h
linux-5.4.282/virt/lib/
linux-5.4.282/virt/lib/Kconfig
linux-5.4.282/virt/lib/Makefile
linux-5.4.282/virt/lib/irqbypass.c
root@ubuntu:/usr/src# ls
linux-5.4.282          linux-headers-5.15.0-139-generic  linux-hwe-5.15-headers-5.15.0-139
linux-5.4.282.tar.xz  linux-headers-5.15.0-67-generic    linux-hwe-5.15-headers-5.15.0-67
root@ubuntu:/usr/src#
input to this VM, move the mouse pointer inside or press Ctrl+G.
```

집파일 오픈한 모습



```
# SPDX-License-Identifier: GPL-2.0
VERSION = 5
PATCHLEVEL = 4
SUBLEVEL = 282
EXTRAVERSION =-os2021202003
NAME = Kleptomaniac Octopus

# *DOCUMENTATION*
# To see a list of typical targets ex
# More info can be located in ./README
```

Makefile 수정한 모습

```
Loadable module support
                                Enable loadable module support
Use 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 --->
-*-
[*] Accessibility support --->
<M> InfiniBand support --->
<*> EDAC (Error Detection And Correction) repository
[*] Real Time Clock --->
[*] DMA Engine support --->
    DMABUF options --->
-*-
[*] Auxiliary Display support --->
<M> Parallel port LCD/Keypad Panel support (OLPC)
{M} Userspace I/O drivers --->
<*> VFIO Non-Privileged userspace driver framework
[*] 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 device
<-> Platform support for Chrome hardware (translators)
[*] Platform support for Chrome hardware --->
[*] Platform support for Mellanox hardware --->
```

```
udes, <N> excludes, <M> modularizes features. Press
excluded <M> module < > module capable
```

```
[*] IA32 Emulation
[ ] x32 ABI for 64-bit mode
```

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

```
CONFIG TPM KEY PARSER=m
CONFIG PKCS7 MESSAGE PARSER=y
CONFIG PKCS7 TEST KEY=m
CONFIG SIGNED PE FILE VERIFICATION=y

#
# Certificates for signature checking
#
CONFIG_MODULE_SIG_KEY="certs/signing_key.pem"
CONFIG_SYSTEM_TRUSTED_KEYRING=y
CONFIG_SYSTEM_TRUSTED_KEYS=""
CONFIG_SYSTEM_EXTRA_CERTIFICATE=y
CONFIG_SYSTEM_EXTRA_CERTIFICATE_SIZE=4096
CONFIG_SECONDARY_TRUSTED_KEYRING=y
CONFIG_SYSTEM_BLACKLIST_KEYRING=y
CONFIG_SYSTEM_BLACKLIST_HASH_LIST=""
CONFIG_SYSTEM_REVOCATION_LIST=y
CONFIG_SYSTEM_REVOCATION_KEYS=""
# end of Certificates for signature checking
#
CONFIG_BINARY_PRINTF=y

#
# Library routines
#
```

vi .config 수정모습

```
MODULES=dep

#
# BUSYBOX: [ y | n | auto ]
#
# Use busybox shell and utilities. If set
# If set to auto (or unset), busybox will
# be used otherwise.
#
```

Initramfs.conf 수정모습

```
LD [M] sound/soc/sor/snd-sor.ko
LD [M] sound/soc/xilinx/snd-soc-xlnx-i2s.ko
LD [M] sound/soc/xilinx/snd-soc-xlnx-formatter-pcm.ko
LD [M] sound/soc/xilinx/snd-soc-xlnx-spdif.ko
LD [M] sound/soc/xtensa/snd-soc-xtfpga-i2s.ko
LD [M] sound/soundcore.ko
LD [M] sound/synth/snd-util-mem.ko
LD [M] sound/synth/emux/snd-emux-synth.ko
LD [M] sound/usb/6fire/snd-usb-6fire.ko
LD [M] sound/usb/bcd2000/snd-bcd2000.ko
LD [M] sound/usb/caiaq/snd-usb-caiaq.ko
LD [M] sound/usb/hiface/snd-usb-hiface.ko
LD [M] sound/usb/line6/snd-usb-pod.ko
LD [M] sound/usb/line6/snd-usb-line6.ko
LD [M] sound/usb/line6/snd-usb-podhd.ko
LD [M] sound/usb/line6/snd-usb-toneport.ko
LD [M] sound/usb/line6/snd-usb-variax.ko
LD [M] sound/usb/misc/snd-ua101.ko
LD [M] sound/usb/snd-usb-audio.ko
LD [M] sound/usb/snd-usbmidi-lib.ko
LD [M] sound/usb/usx2y/snd-usb-us122l.ko
LD [M] sound/usb/usx2y/snd-usb-usx2y.ko
LD [M] sound/x86/snd-hdmi-lpe-audio.ko
LD [M] sound/xen/snd_xen_front.ko
root@ubuntu:/usr/src/linux-5.4.282#
```

sudo make -jN을 수행한 모습

현재 환경설정에서 cpu를 4로 부여해 8로 잡았습니다.

```
INSTALL sound/soc/xilinx/snd-soc-xlnx-formatter-pcm.ko
INSTALL sound/soc/xilinx/snd-soc-xlnx-i2s.ko
INSTALL sound/soc/xilinx/snd-soc-xlnx-spdif.ko
INSTALL sound/soc/xtensa/snd-soc-xtfpga-i2s.ko
INSTALL sound/soundcore.ko
INSTALL sound/synth/emux/snd-emux-synth.ko
INSTALL sound/synth/snd-util-mem.ko
INSTALL sound/usb/6fire/snd-usb-6fire.ko
INSTALL sound/usb/bcd2000/snd-bcd2000.ko
INSTALL sound/usb/caiaq/snd-usb-caiaq.ko
INSTALL sound/usb/hiface/snd-usb-hiface.ko
INSTALL sound/usb/line6/snd-usb-line6.ko
INSTALL sound/usb/line6/snd-usb-pod.ko
INSTALL sound/usb/line6/snd-usb-podhd.ko
INSTALL sound/usb/line6/snd-usb-toneport.ko
INSTALL sound/usb/line6/snd-usb-variax.ko
INSTALL sound/usb/misc/snd-ua101.ko
INSTALL sound/usb/snd-usb-audio.ko
INSTALL sound/usb/snd-usbmidi-lib.ko
INSTALL sound/usb/usx2y/snd-usb-us122l.ko
INSTALL sound/usb/usx2y/snd-usb-usx2y.ko
INSTALL sound/x86/snd-hdmi-lpe-audio.ko
INSTALL sound/xen/snd_xen_front.ko
DEPMOD 5.4.282-os2021202003
root@ubuntu:/usr/src/linux-5.4.282#
```

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

```
# 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_TIMEOUT_STYLE=menu
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet_splash_nokaslr"
GRUB_CMDLINE_LINUX="find_preseed=/preseed.cfg auto noprompt priority=critical"

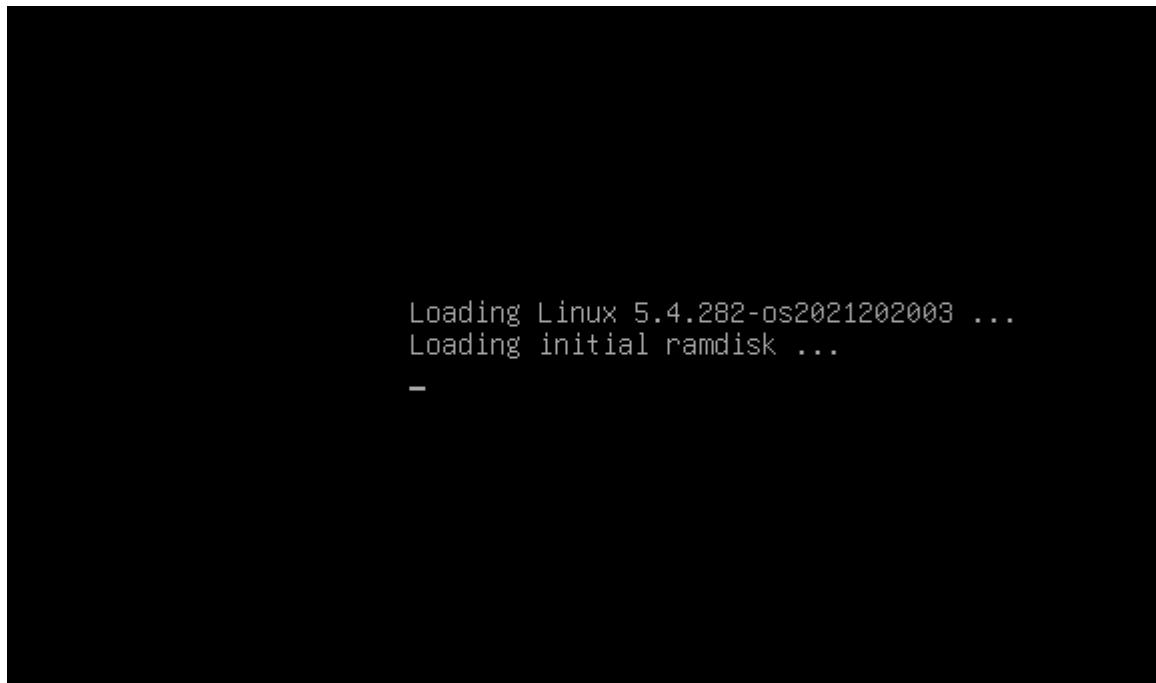
# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

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

# The resolution used on graphical terminal
```

```
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.

```

        return result;
    }

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);

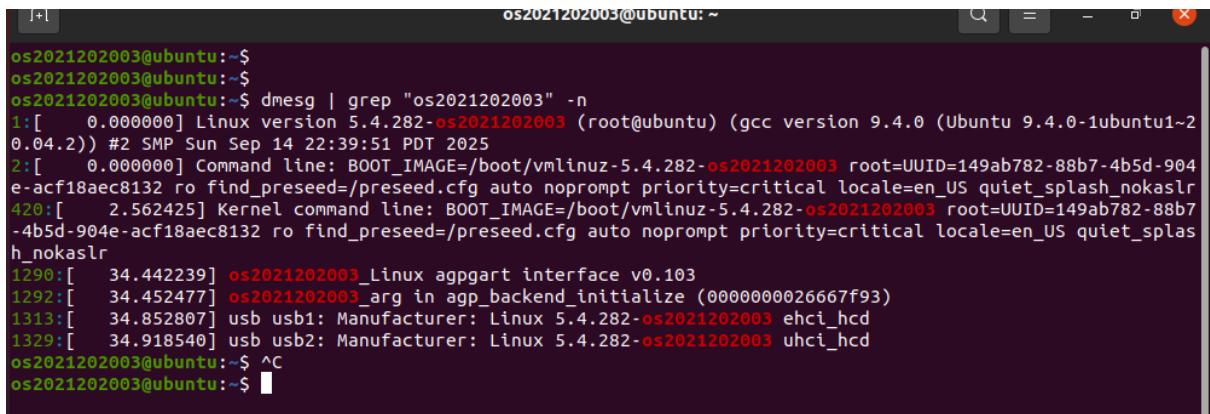
        if (!page) {
            dev_err(&bridge->dev->dev,
                    "can't get memory for scratch page\n");
            return -ENOMEM;
        }

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

```

이후 평선 기반 함수를 찾아 그중 하나인 `agp_backend_initialize` 함수의 내부에 출력문을 추가 구현해두었습니다.

출력 내용 수정한 모습



```

os2021202003@ubuntu:~$ 
os2021202003@ubuntu:~$ 
os2021202003@ubuntu:~$ dmesg | grep "os2021202003" -n
1:[ 0.000000] Linux version 5.4.282-os2021202003 (root@ubuntu) (gcc version 9.4.0 (Ubuntu 9.4.0-1ubuntu1~0.04.2)) #2 SMP Sun Sep 14 22:39:51 PDT 2025
2:[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.282-os2021202003 root=UUID=149ab782-88b7-4b5d-904e-acf18aec8132 ro find_preseed=/preseed.cfg auto noprompt priority=critical locale=en_US quiet_splash_nokaslr
420:[ 2.562425] Kernel command line: BOOT_IMAGE=/boot/vmlinuz-5.4.282-os2021202003 root=UUID=149ab782-88b7-4b5d-904e-acf18aec8132 ro find_preseed=/preseed.cfg auto noprompt priority=critical locale=en_US quiet_splash_nokaslr
1290:[ 34.442239] os2021202003_Linux agpgart interface v0.103
1292:[ 34.452477] os2021202003_arg in agp_backend_initialize (0000000026667f93)
1313:[ 34.852807] usb usb1: Manufacturer: Linux 5.4.282-os2021202003 ehci_hcd
1329:[ 34.918540] usb usb2: Manufacturer: Linux 5.4.282-os2021202003 uhci_hcd
os2021202003@ubuntu:~$ ^C
os2021202003@ubuntu:~$ 

```

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 N을 8로 했을때도 2시간 30분정도가 지나는 모습인데 자료에서 이정도로 시간이 많이 걸린다고 언급을 해주지 않아서 문제가 생긴줄 알고 몇차례 강제 중지를하고 가상환경을 재시작한

기억이 난다. 또한 `grep`를 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.