

운영체제 실습

[Assignment #1]

Class : 목 3
Professor : 최상호
Student ID : 2020202031
Name : 김재현

Introduction

리눅스를 설치하고, 우분투 내에서 5.4.282 Kernel 을 다운로드 하고 `uname -r` 명령어를 통해 버전을 확인합니다.

agpgart interface 가 실행되는 지점에서 Linux Kernel Message 출력하도록, Ctags, Cscope 등 programming tools 를 활용하여 Linux 해당 함수 및 파일을 찾아 수정합니다.

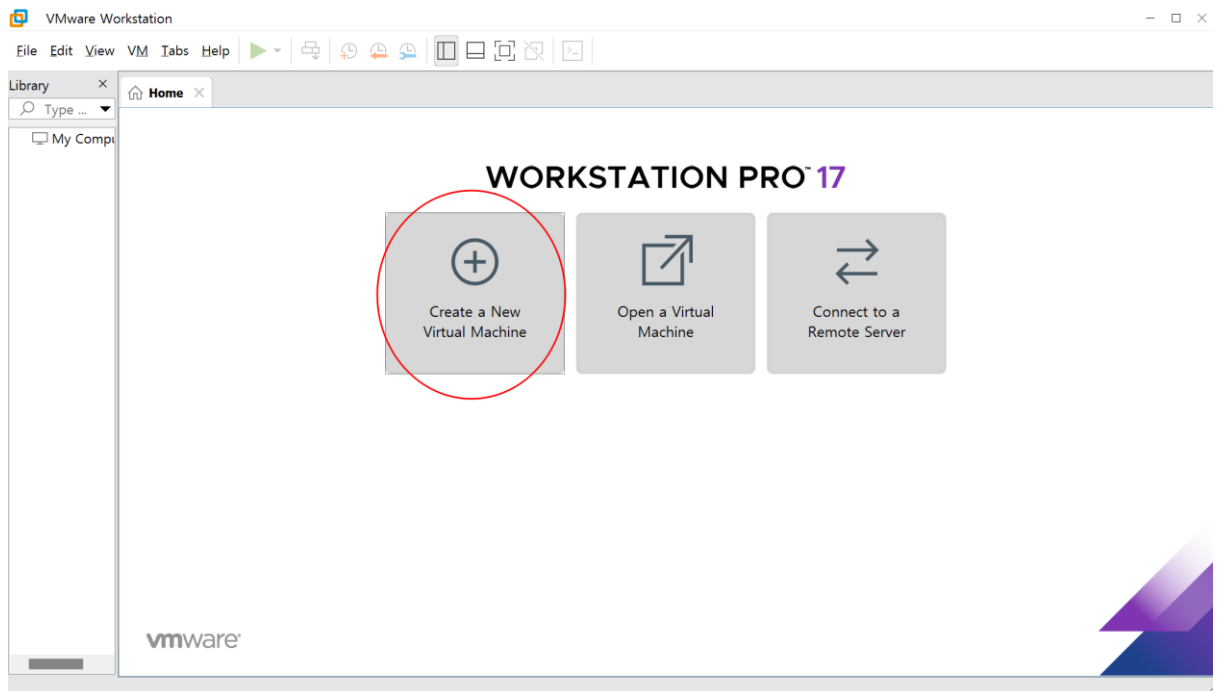
결과화면

Assignment 1-1

1. VMware 를 설치합니다.



2. Ubuntu 를 설치합니다.



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):

\\kk200\\Downloads\\ubuntu-20.04.6-desktop-amd64.iso

Browse...

Ubuntu 64-bit 20.04.6 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.

Help

< Back

Next >

Cancel

New Virtual Machine Wizard

Easy Install Information

This is used to install Ubuntu 64-bit.

Personalize Linux

Full name:

os2020202031

User name:

os2020202031

Password:

••••••••

Confirm:

••••••••

Help

< Back

Next >

Cancel

×

How large do you want this disk to be?

Maximum disk size (GB):

☐ Store virtual disk as a single file

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

[< Back](#)

Cancel

×

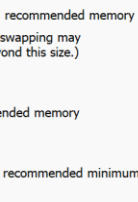
Add...

Remove

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

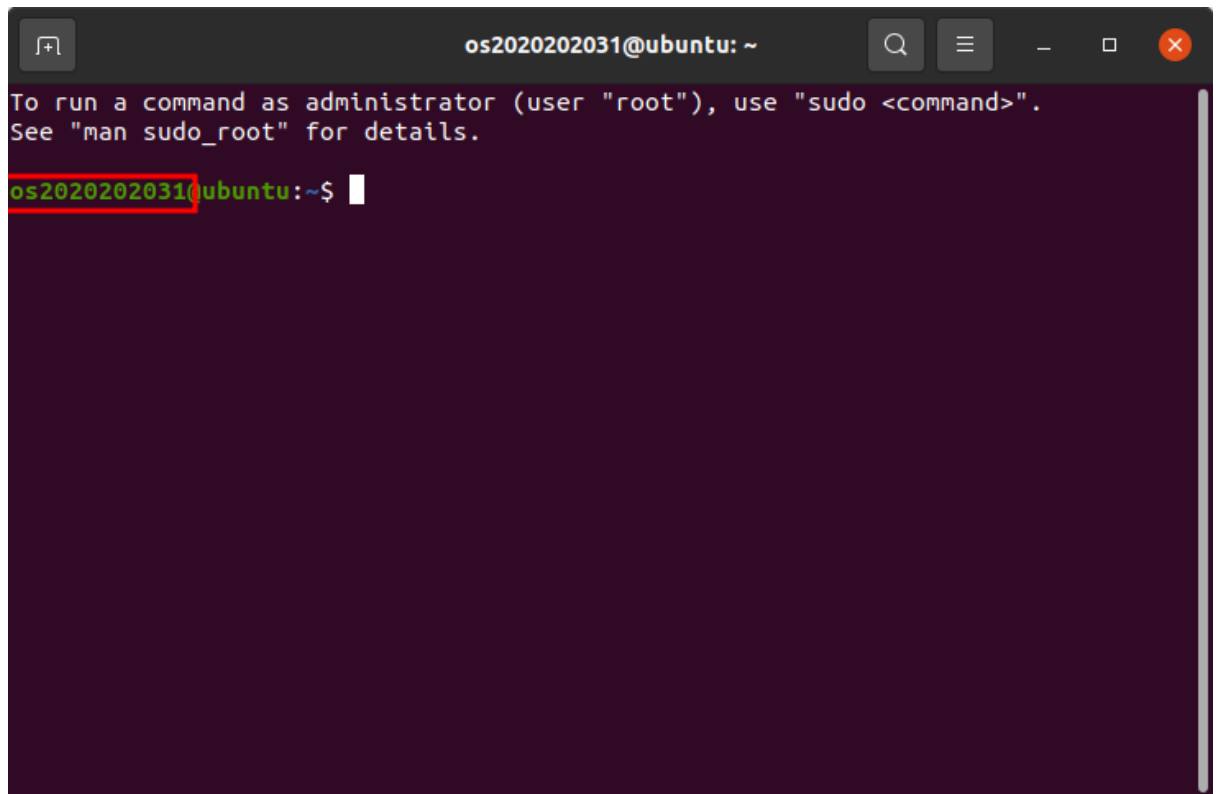
Memory for this virtual machine: 4096 MB

Memory for this virtual machine: 4096 MB



Close

Help



A terminal window with a dark purple background. The title bar at the top reads "os2020202031@ubuntu: ~". Below the title bar, there is a message: "To run a command as administrator (user "root"), use "sudo <command>". See "man sudo_root" for details." Below this message, the command prompt "os2020202031@ubuntu:~\$" is displayed, with the username "os2020202031" highlighted by a red rectangular box. A white cursor is positioned at the end of the prompt.

```
os2020202031@ubuntu: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
os2020202031@ubuntu:~$
```

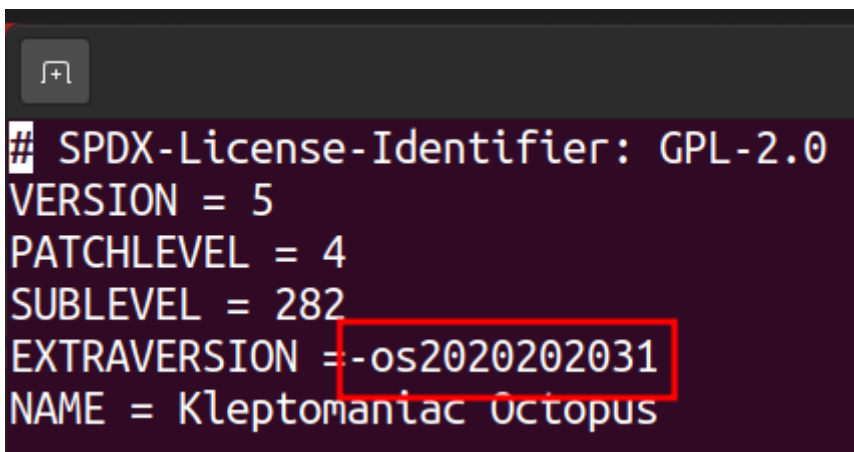
User ID 를 os 학번으로 설정했습니다.

Assignment 1-2

1. sudo apt update 입력

```
os2020202031@ubuntu:~$ sudo apt update
[sudo] password for os2020202031:
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
288 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

2. vi Makefile 입력 후 EXTRAVERSION 에 os 학번 입력



```
## SPDX-License-Identifier: GPL-2.0
VERSION = 5
PATCHLEVEL = 4
SUBLEVEL = 282
EXTRAVERSION = -os2020202031
NAME = Kleptomaniac Octopus
```

3. sudo apt install build-essential libncurses5-dev bison flex libssl-dev libelf-dev dwarves zstd 입력

```

root@ubuntu: /usr/src/linux-5.4.282
Setting up binutils-common:amd64 (2.34-6ubuntu1.9) ...
Setting up linux-libc-dev:amd64 (5.4.0-195.215) ...
Setting up libctf-nobfd0:amd64 (2.34-6ubuntu1.9) ...
Setting up libfakeroot:amd64 (1.24-1) ...
Setting up libc6-dbg:amd64 (2.31-0ubuntu9.16) ...
Setting up fakeroot (1.24-1) ...
update-alternatives: using /usr/bin/fakeroot-sysv to provide /usr/bin/fakeroot (fakeroot) in auto mode
Setting up libasan5:amd64 (9.4.0-1ubuntu1~20.04.2) ...
Setting up libsigsegv2:amd64 (2.12-2) ...
Setting up libquadmath0:amd64 (10.5.0-1ubuntu1~20.04) ...
Setting up libssl-dev:amd64 (1.1.1f-1ubuntu2.23) ...
Setting up libfl2:amd64 (2.6.4-6.2) ...
Setting up dwarves (1.21-0ubuntu1~20.04.1) ...
Setting up libubsan1:amd64 (10.5.0-1ubuntu1~20.04) ...
Setting up libcrypt-dev:amd64 (1:4.4.10-10ubuntu4) ...
Setting up libbinutils:amd64 (2.34-6ubuntu1.9) ...
Setting up libc-dev-bin (2.31-0ubuntu9.16) ...
Setting up libalgorithm-diff-xs-perl (0.04-6) ...
Setting up zstd (1.4.4+dfsg-3ubuntu0.1) ...
Setting up liblsan0:amd64 (10.5.0-1ubuntu1~20.04) ...
Setting up libitm1:amd64 (10.5.0-1ubuntu1~20.04) ...
Setting up libalgorithm-merge-perl (0.08-3) ...
Setting up libtsan0:amd64 (10.5.0-1ubuntu1~20.04) ...
Setting up libctf0:amd64 (2.34-6ubuntu1.9) ...
Setting up m4 (1.4.18-4) ...
Setting up libgcc-9-dev:amd64 (9.4.0-1ubuntu1~20.04.2) ...
Setting up bison (2:3.5.1+dfsg-1) ...
update-alternatives: using /usr/bin/bison.yacc to provide /usr/bin/yacc (yacc) in auto mode
Setting up libc6-dev:amd64 (2.31-0ubuntu9.16) ...
Setting up binutils-x86-64-linux-gnu (2.34-6ubuntu1.9) ...
Setting up flex (2.6.4-6.2) ...
Setting up libncurses-dev:amd64 (6.2-0ubuntu2.1) ...

```

4. sudo make menuconfig 입력

```

root@ubuntu: /usr/src/linux-5.4.282
Setting up libelf-dev:amd64 (0.176-1.1ubuntu0.1) ...
Setting up g++-9 (9.4.0-1ubuntu1~20.04.2) ...
Setting up g++ (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.8ubuntu1.1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
root@ubuntu: /usr/src/linux-5.4.282# sudo make menuconfig
HOSTCC scripts/basic/fixdep
UPD scripts/kconfig/mconf-cfg
HOSTCC scripts/kconfig/mconf.o
HOSTCC scripts/kconfig/Lxdialog/checklist.o
HOSTCC scripts/kconfig/Lxdialog/inputbox.o
HOSTCC scripts/kconfig/Lxdialog/menubox.o
HOSTCC scripts/kconfig/Lxdialog/textbox.o
HOSTCC scripts/kconfig/Lxdialog/util.o
HOSTCC scripts/kconfig/Lxdialog/yesno.o
HOSTCC scripts/kconfig/confdata.o
HOSTCC scripts/kconfig/expr.o
LEX scripts/kconfig/lexer.lex.c
YACC scripts/kconfig/parser.tab.[ch]
HOSTCC scripts/kconfig/lexer.lex.o
HOSTCC scripts/kconfig/parser.tab.o
HOSTCC scripts/kconfig/preprocess.o
HOSTCC scripts/kconfig/symbol.o
HOSTLD scripts/kconfig/mconf
scripts/kconfig/mconf Kconfig
#
# using defaults found in /boot/config-5.15.0-119-generic
#
/boot/config-5.15.0-119-generic:3290:warning: symbol value 'm' invalid for CHELSIO_IPSEC_INLINE
/boot/config-5.15.0-119-generic:3968:warning: symbol value 'm' invalid for RTW88_8822BE
/boot/config-5.15.0-119-generic:3969:warning: symbol value 'm' invalid for RTW88_8822CE
/boot/config-5.15.0-119-generic:4470:warning: symbol value 'm' invalid for SERIAL_LANTIQ
/boot/config-5.15.0-119-generic:5325:warning: symbol value 'm' invalid for PROC_THERMAL_MMIO_RAPL
/boot/config-5.15.0-119-generic:5557:warning: symbol value 'm' invalid for MFD_ARIZONA
/boot/config-5.15.0-119-generic:6812:warning: symbol value 'm' invalid for FB_SIMPLE
/boot/config-5.15.0-119-generic:8873:warning: symbol value 'm' invalid for ASHMEM
/boot/config-5.15.0-119-generic:9958:warning: symbol value 'm' invalid for ANDROID_BINDER_IPC
/boot/config-5.15.0-119-generic:9959:warning: symbol value 'm' invalid for ANDROID_BINDERFS

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

```



```
root@ubuntu: /usr/src/linux-5.4.282

.config - linux/x86_64 5.4.282-os20200202031 Kernel Configuration

Linux/x86_64 5.4.282-os20200202031 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

*** Compiler: gcc (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0 ***
General setup --->
[*] 64-bit kernel
Processor type and features --->
[ ] Force GDS Mitigation
Power management and ACPI options --->
Bus options (PCI etc.) --->
Binary Emulations --->
Firmware Drivers --->
[*] Virtualization --->
General architecture-dependent options --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
IO Schedulers --->
Executable file formats --->
Memory Management options --->
[*] Networking support --->
Device Drivers --->
File systems --->
Security options --->
-* Cryptographic API --->
Library routines --->
Kernel hacking --->

<select> < Exit > < Help > < Save > < Load >
```

```
root@ubuntu: /usr/src/linux-5.4.282

.config - linux/x86_64 5.4.282-os20200202031 Kernel Configuration
> Enable loadable module support

Enable loadable module support
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

--- 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
Which hash algorithm should modules be signed with? (Sign m
[ ] Compress modules on installation
[ ] Allow loading of modules with missing namespace imports
[*] Enable unused/obsolete exported symbols

<select> < Exit > < Help > < Save > < Load >
```

```
root@ubuntu: /usr/src/linux-5.4.282
.config - Linux/x86 5.4.282-os2020202031 Kernel Configuration

Linux/x86 5.4.282-os2020202031 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus --->). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

*** Compiler: gcc (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0 ***
General setup --->
[*] 64-bit kernel
Processor type and features --->
[ ] Force GDS Mitigation
Power management and ACPI options --->
Bus options (PCI etc.) --->
Binary Emulations --->
Firmware Drivers --->
[*] Virtualization --->
General architecture-dependent options --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
IO Schedulers --->
Executable file formats --->
Memory Management options --->
[*] Networking support --->
Device Drivers --->
File systems --->
Security options --->
-* Cryptographic API --->
Library routines --->
Kernel hacking --->

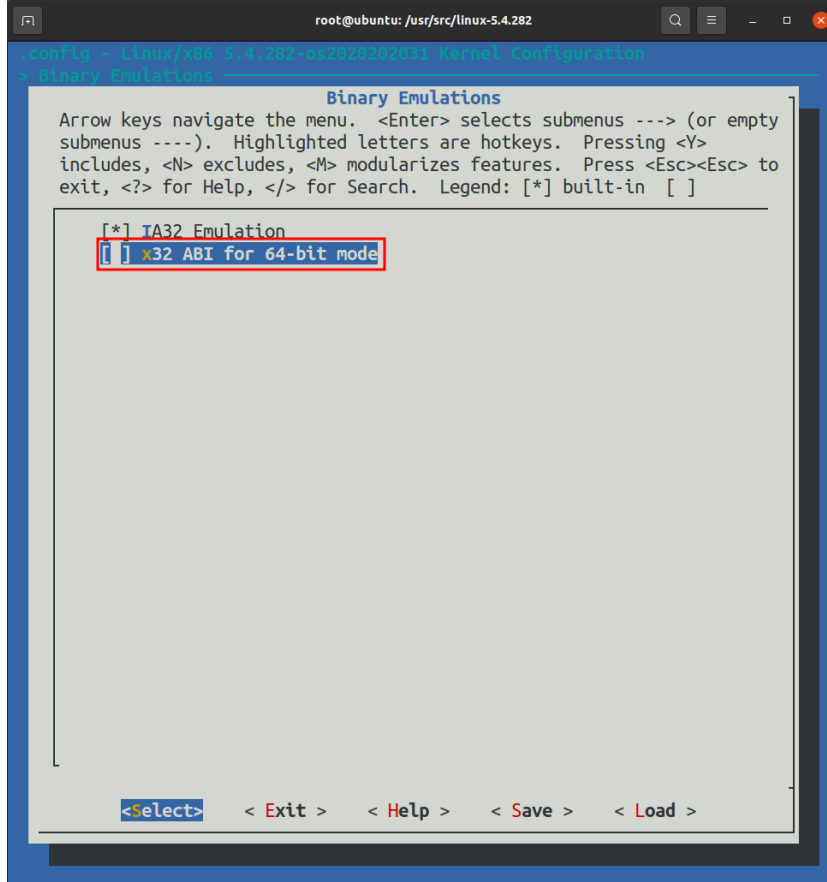
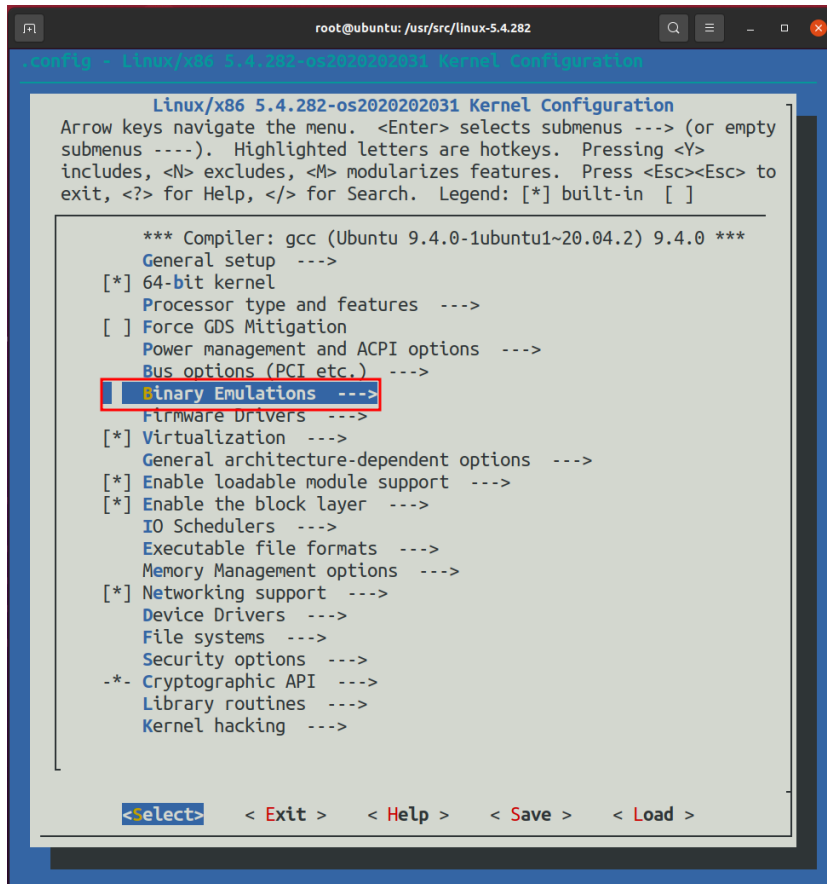
<Select> < Exit > < Help > < Save > < Load >
```

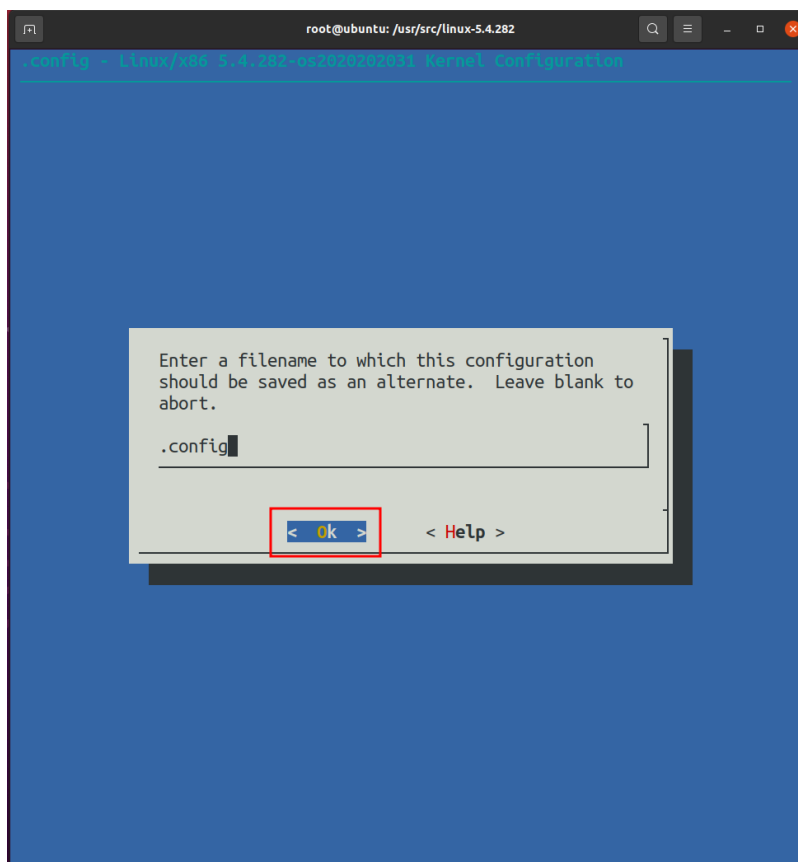
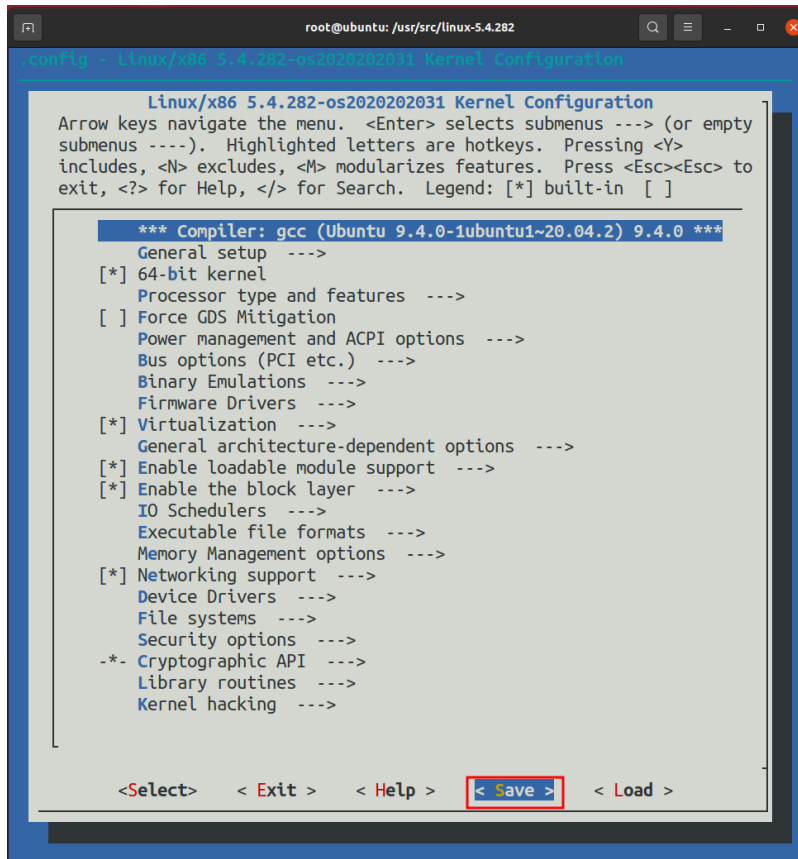
```
root@ubuntu: /usr/src/linux-5.4.282
.config - Linux/x86 5.4.282-os2020202031 Kernel Configuration
> Device Drivers

Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus --->). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

^(-)
[*] DMA Engine support --->
DMABUF options --->
-* Auxiliary Display support --->
<M> Parallel port LCD/Keypad Panel support (OLD OPTION)
{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 devices --->
< > Platform support for Chrome hardware (transitional)
[*] Platform support for Chrome hardware --->
[*] Platform support for Mellanox hardware --->
Common Clock Framework --->
[*] Hardware Spinlock drivers --->
Clock Source drivers --->
-* Mailbox Hardware Support --->
[*] IOMMU Hardware Support --->
Remoteproc drivers --->
Rpmmsg drivers --->
v(+)

<Select> < Exit > < Help > < Save > < Load >
```





```
root@ubuntu: /usr/src/linux-5.4.282
os2020202031@ubuntu:~$ sudo su
[sudo] password for os2020202031:
root@ubuntu:/home/os2020202031# cd /usr/src
root@ubuntu:/usr/src# sudo make menuconfig
make: *** No rule to make target 'menuconfig'. Stop.
root@ubuntu:/usr/src# ls
linux-5.4.282          linux-headers-5.15.0-67-generic
linux-5.4.282.tar.xz   linux-hwe-5.15-headers-5.15.0-119
linux-headers-5.15.0-119-generic  linux-hwe-5.15-headers-5.15.0-67
root@ubuntu:/usr/src# cd linux-5.4.282
root@ubuntu:/usr/src/linux-5.4.282# sudo make menuconfig
scripts/kconfig/mconf Kconfig

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

root@ubuntu:/usr/src/linux-5.4.282# sudo make menuconfig
scripts/kconfig/mconf Kconfig

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

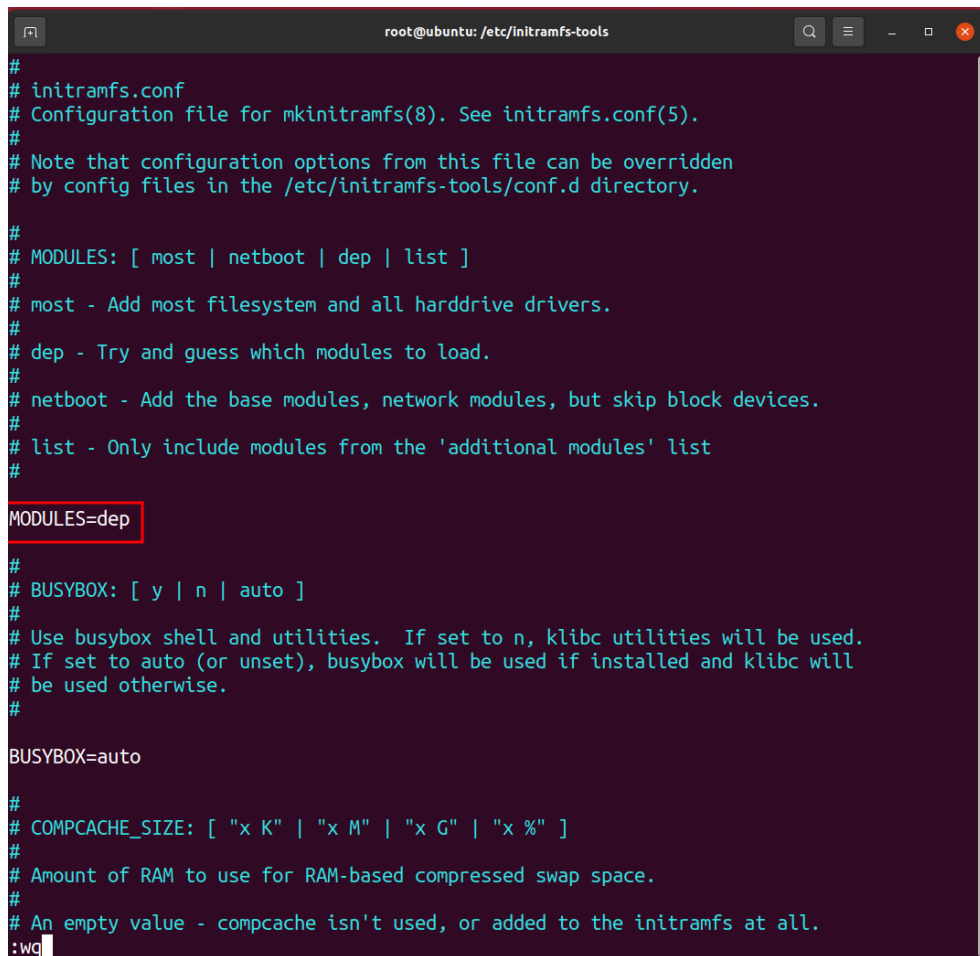
5. scripts/config --disable SYSTEM_TRUSTED_KEYS 입력

scripts/config --disable SYSTEM_REVOCATION_KEYS 입력

```
root@ubuntu: /usr/src/linux-5.4.282
root@ubuntu:/usr/src/linux-5.4.282# sudo vi .config
root@ubuntu:/usr/src/linux-5.4.282# scripts/config --disable SYSTEM_TRUSTED_KEYS
root@ubuntu:/usr/src/linux-5.4.282# scripts/config --disable SYSTEM_REVOCATION_KEYS
root@ubuntu:/usr/src/linux-5.4.282#
```

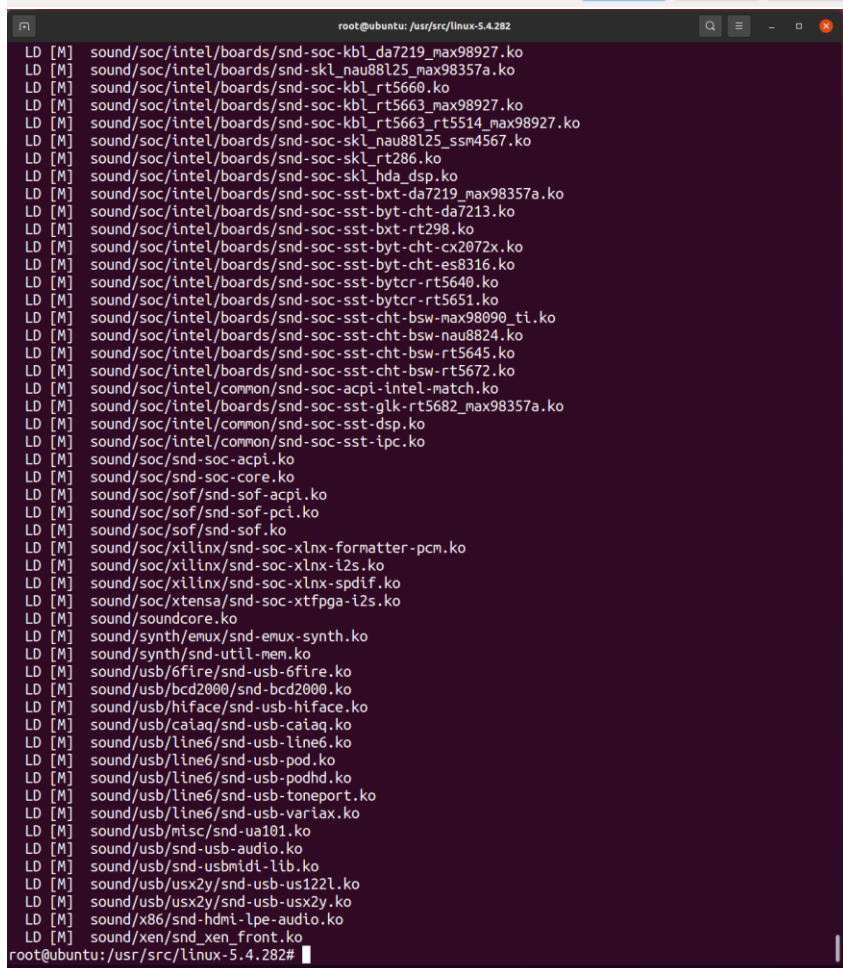
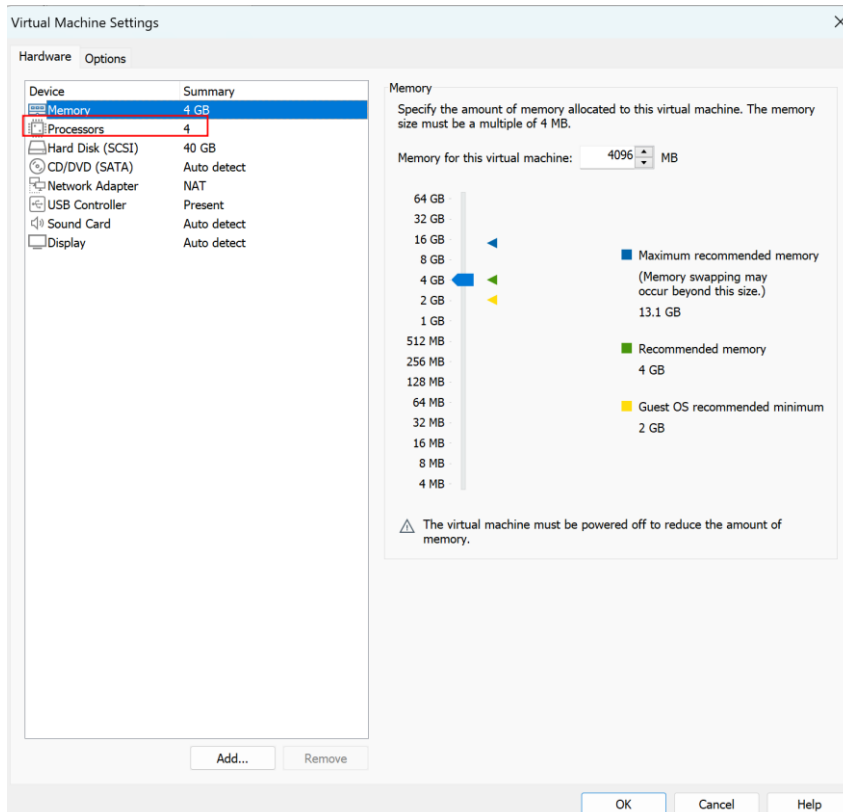
6. `cd /etc/initramfs-tools` 입력

`vi initramfs.conf` 입력



```
root@ubuntu: /etc/initramfs-tools
#
# initramfs.conf
# Configuration file for mkinitramfs(8). See initramfs.conf(5).
#
# Note that configuration options from this file can be overridden
# by config files in the /etc/initramfs-tools/conf.d directory.
#
# MODULES: [ most | netboot | dep | list ]
#
# most - Add most filesystem and all harddrive drivers.
#
# dep - Try and guess which modules to load.
#
# netboot - Add the base modules, network modules, but skip block devices.
#
# list - Only include modules from the 'additional modules' list
#
MODULES=dep
#
# BUSYBOX: [ y | n | auto ]
#
# Use busybox shell and utilities. If set to n, klibc utilities will be used.
# If set to auto (or unset), busybox will be used if installed and klibc will
# be used otherwise.
#
BUSYBOX=auto
#
# COMPCACHE_SIZE: [ "x K" | "x M" | "x G" | "x %" ]
#
# Amount of RAM to use for RAM-based compressed swap space.
#
# An empty value - compcache isn't used, or added to the initramfs at all.
:wq
```

7. `make -j8` 입력



8. make modules_install 입력

```
root@ubuntu: /usr/src/linux-5.4.282
INSTALL sound/soc/intel/boards/snd-soc-kbl_da7219_max98927.ko
INSTALL sound/soc/intel/boards/snd-soc-kbl_rt5660.ko
INSTALL sound/soc/intel/boards/snd-soc-kbl_rt5663_max98927.ko
INSTALL sound/soc/intel/boards/snd-soc-kbl_rt5663_rt5514_max98927.ko
INSTALL sound/soc/intel/boards/snd-soc-skl_hda_dsp.ko
INSTALL sound/soc/intel/boards/snd-soc-skl_nau88125_ssm4567.ko
INSTALL sound/soc/intel/boards/snd-soc-skl_rt286.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-bxt-da7219_max98357a.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-bxt-rt298.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-byt-cht-cx2072x.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-byt-cht-da7213.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-byt-cht-es8316.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-bytcr-rt5640.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-bytcr-rt5651.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-max98090_ti.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-nau8824.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-rt5645.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-rt5672.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-glk-rt5682_max98357a.ko
INSTALL sound/soc/intel/common/snd-soc-acpi-intel-match.ko
INSTALL sound/soc/intel/common/snd-soc-sst-dsp.ko
INSTALL sound/soc/intel/common/snd-soc-sst-ipc.ko
INSTALL sound/soc/snd-soc-acpi.ko
INSTALL sound/soc/snd-soc-core.ko
INSTALL sound/soc/sof/snd-sof-acpi.ko
INSTALL sound/soc/sof/snd-sof-pci.ko
INSTALL sound/soc/sof/snd-sof.ko
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-variix.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-os2020202031
root@ubuntu: /usr/src/linux-5.4.282#
```

9. make install 입력


```

root@ubuntu:/usr/src/linux-5.4.282# make install
sh ./arch/x86/boot/install.sh 5.4.282-os2020202031 arch/x86/boot/bzImage \
    System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.4.282-os2020202031 /boot/vmlinuz-5.4.282-os2020202031
update-initramfs: Generating /boot/initrd.img-5.4.282-os2020202031
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 5.4.282-os2020202031 /boot/vmlinuz-5.4.282-os2020202031
run-parts: executing /etc/kernel/postinst.d/update-notifier 5.4.282-os2020202031 /boot/vmlinuz-5.4.282-os2020202031
run-parts: executing /etc/kernel/postinst.d/xx-update-initrd-links 5.4.282-os2020202031 /boot/vmlinuz-5.4.282-os2020202031
I: /boot/initrd.img.old is now a symlink to initrd.img-5.15.0-119-generic
I: /boot/initrd.img is now a symlink to initrd.img-5.4.282-os2020202031
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.4.282-os2020202031 /boot/vmlinuz-5.4.282-os2020202031
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.15.0-119-generic
Found initrd image: /boot/initrd.img-5.15.0-119-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-os2020202031
Found initrd image: /boot/initrd.img-5.4.282-os2020202031
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
root@ubuntu:/usr/src/linux-5.4.282# █

```

10. vi /etc/default/grub 입력 및 수정

```
root@ubuntu: /usr/src/linux-5.4.282
# 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 locale=en_US"

# 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"
```

11. sudo update-grub 입력

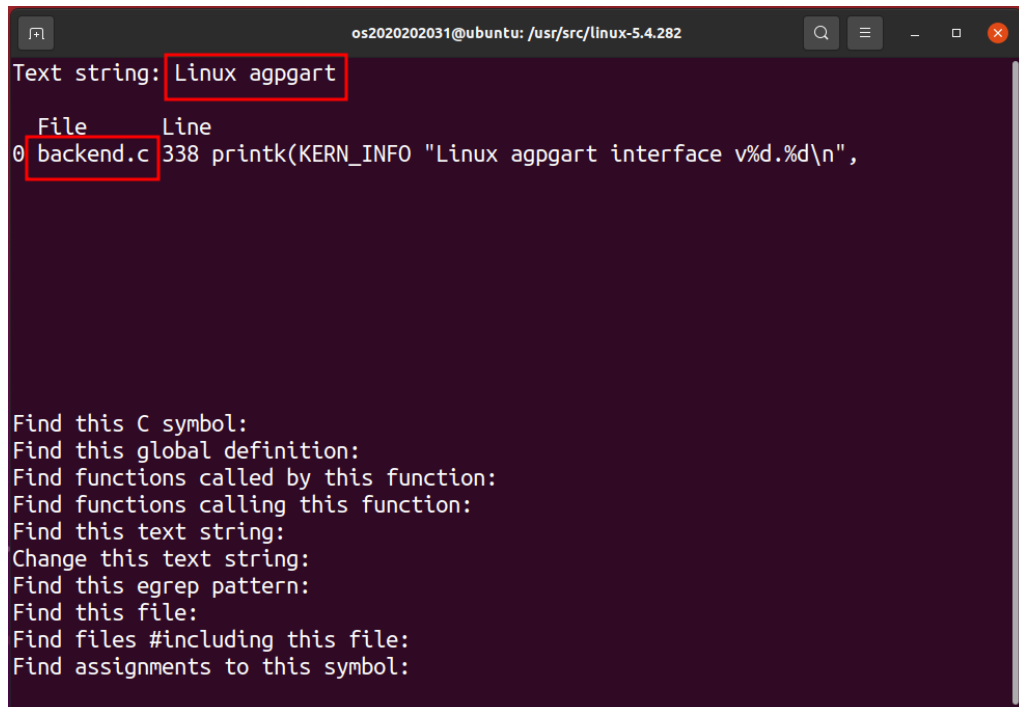
```
root@ubuntu:/usr/src/linux-5.4.282# sudo update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.15.0-119-generic
Found initrd image: /boot/initrd.img-5.15.0-119-generic
Found linux 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-os2020202031
Found initrd image: /boot/initrd.img-5.4.282-os2020202031
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
root@ubuntu:/usr/src/linux-5.4.282#
```

12. 커널 버전 확인

```
os2020202031@ubuntu:~$ uname -r
5.4.282-os2020202031
```

Assignment 1-3

1. cscope 에서 Find this text string:에 "Linux agpgart" text 를 검색



The screenshot shows the cscope interface in a terminal window. The title bar indicates the user is 'os20202031' on 'ubuntu' in the directory '/usr/src/linux-5.4.282'. The search results are displayed in a table with columns 'File' and 'Line'. The first result is 'backend.c' at line 338, which contains the text 'printk(KERN_INFO "Linux agpgart interface v%d.%d\n",'. Below the table, a list of search actions is provided: '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:', and 'Find assignments to this symbol:'.

```
os20202031@ubuntu: /usr/src/linux-5.4.282
Text string: Linux agpgart

File      Line
0 backend.c 338 printk(KERN_INFO "Linux agpgart interface v%d.%d\n",

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:
```

2. Linux agpgart interface 가 실행되는 지점에서

2020202031_Linux agpgart interface v0.103

2020202031_arg in 이 출력되도록 수정



The screenshot shows the source code for the Linux agpgart interface in a terminal window. The title bar indicates the user is 'root' on 'ubuntu' in the directory '/usr/src/linux-5.4.282'. The code is for the 'agp_init' function. A red box highlights the 'if (!agp_off)' block, which contains three 'printk' statements. The first two statements are for the version and the third is for the argument 'arg'. The code is as follows:

```
root@ubuntu: /usr/src/linux-5.4.282

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 "os20202031_Linux agpgart interface v%d.%d\n",
                    AGPGART_VERSION_MAJOR, AGPGART_VERSION_MINOR);
        printk(KERN_INFO "os20202031_arg in %s(void)", __func__);
    }
    return 0;
}

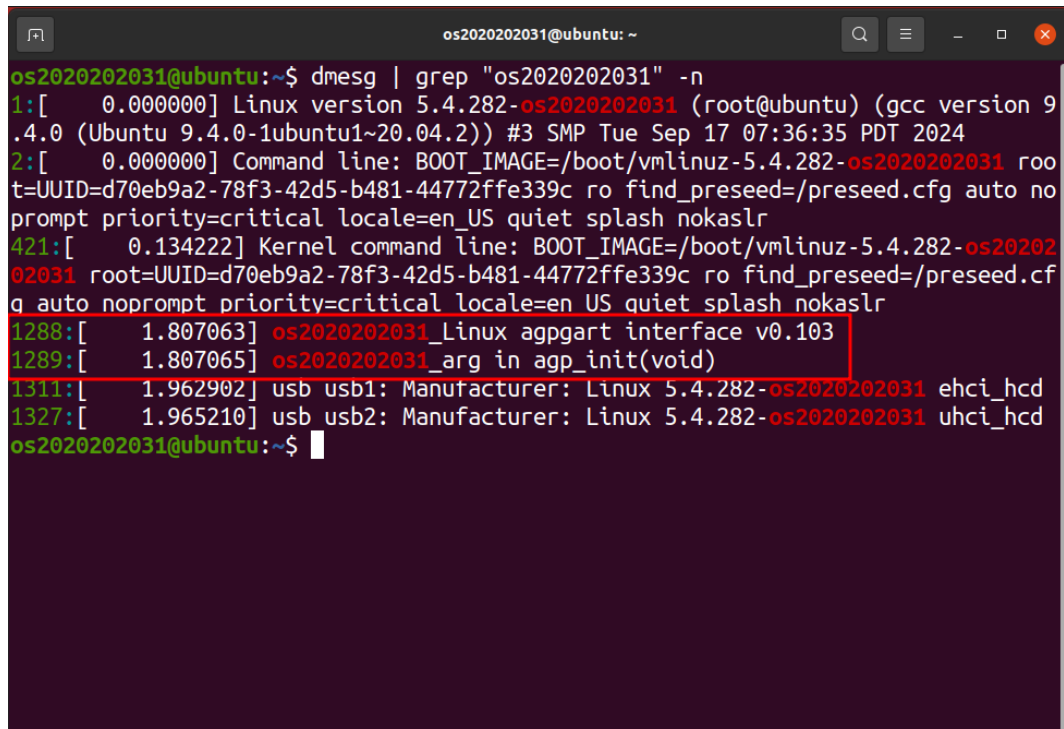
static void __exit agp_exit(void)
{
}

#ifdef MODULE
static __init int agp_setup(char *s)
{
    if (!strcmp(s, "off"))
        agp_off = 1;
    if (!strcmp(s, "try_unsupported"))
        agp_try_unsupported_boot = 1;
}
#endif
```

drivers/char/agp/backend.c 파일을 수정했습니다.

!agp_off 는 곧 agp on 을 의미하며, 이는 init 에서 한 번만 실행되므로, Linux agpgart interface 가 실행되는 지점에서 printk 가 한 번만 출력됩니다.

3. dmesg | grep "os2020202031" -n 실행결과 printk 가 잘 수행됐음을 알 수 있습니다.



```
os2020202031@ubuntu: ~  
os2020202031@ubuntu:~$ dmesg | grep "os2020202031" -n  
1:[    0.000000] Linux version 5.4.282-os2020202031 (root@ubuntu) (gcc version 9  
.4.0 (Ubuntu 9.4.0-1ubuntu1~20.04.2)) #3 SMP Tue Sep 17 07:36:35 PDT 2024  
2:[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.282-os2020202031 roo  
t=UUID=d70eb9a2-78f3-42d5-b481-44772ffe339c ro find_preseed=/preseed.cfg auto no  
prompt priority=critical locale=en_US quiet splash nokaslr  
421:[    0.134222] Kernel command line: BOOT_IMAGE=/boot/vmlinuz-5.4.282-os20202  
02031 root=UUID=d70eb9a2-78f3-42d5-b481-44772ffe339c ro find_preseed=/preseed.cf  
q auto noprompt priority=critical locale=en_US quiet splash nokaslr  
1288:[    1.807063] os2020202031_Linux agpgart interface v0.103  
1289:[    1.807065] os2020202031_arg in agp_init(void)  
1311:[    1.962902] usb usb1: Manufacturer: Linux 5.4.282-os2020202031 ehci_hcd  
1327:[    1.965210] usb usb2: Manufacturer: Linux 5.4.282-os2020202031 uhci_hcd  
os2020202031@ubuntu:~$
```

고찰

cscope -R 을 통해 Linux agpgart text 를 포함하는 파일을 찾고, 수정하는 과정에서 readonly 형식으로 open 됐다는 오류문구가 계속해서 등장했습니다. 이에 권한문제라고 생각하고 sudo cscope -R 로 실행했더니 잘 수정됐습니다.

Reference

수업자료 참고