

운영체제실습

assignment 1

담당교수 : 김태석 교수님

학 번 : 2021202058

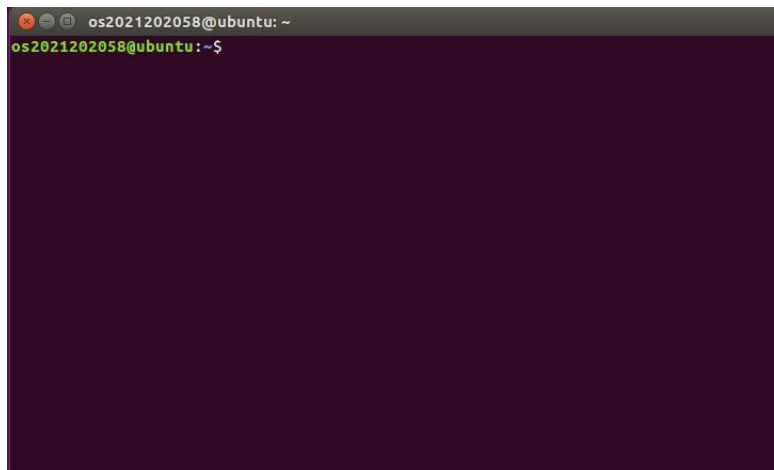
성 명 : 송채영

1. Introduction

VMWare 를 이용해 Ubuntu 를 설치해 Linux 상에서의 실습환경을 구축한다. Linux 상의 프로그래밍과 명령어(command)에 대해 이해하고 실습을 통해 각각의 기능에 대해 알아본다. 더 나아가 Linux (kernel 4.19)에 대해 알아보며 우분투 내에서 kernel 을 다운로드 하여 컴파일 해보는 과정을 실습해보며 compile kernel 에 대해 이해한다. 마지막으로 ctags, cscope 명령어와 printk() 함수를 활용해 linux kernel Message 를 출력해본다.

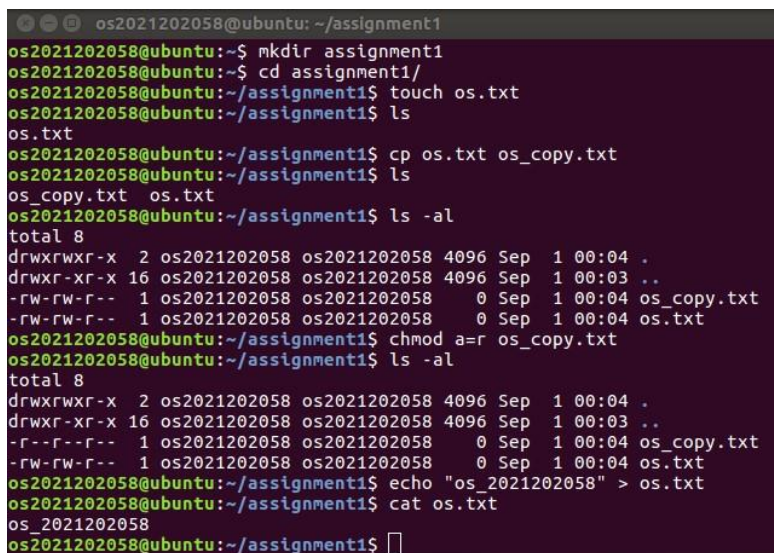
2. Result

- Assignment 1-1



```
os2021202058@ubuntu: ~  
os2021202058@ubuntu:~$
```

Linux installation 에 해당하는 부분으로 계정 ID 가 os 학번으로 되어있는 것을 위 터미널 화면을 통해 확인할 수 있다.



```
os2021202058@ubuntu: ~/assignment1  
os2021202058@ubuntu:~$ mkdir assignment1  
os2021202058@ubuntu:~$ cd assignment1/  
os2021202058@ubuntu:~/assignment1$ touch os.txt  
os2021202058@ubuntu:~/assignment1$ ls  
os.txt  
os2021202058@ubuntu:~/assignment1$ cp os.txt os_copy.txt  
os2021202058@ubuntu:~/assignment1$ ls  
os_copy.txt  os.txt  
os2021202058@ubuntu:~/assignment1$ ls -al  
total 8  
drwxrwxr-x  2 os2021202058 os2021202058 4096 Sep  1 00:04 .  
drwxr-xr-x 16 os2021202058 os2021202058 4096 Sep  1 00:03 ..  
-rw-rw-r--  1 os2021202058 os2021202058   0 Sep  1 00:04 os_copy.txt  
-rw-rw-r--  1 os2021202058 os2021202058   0 Sep  1 00:04 os.txt  
os2021202058@ubuntu:~/assignment1$ chmod a=r os_copy.txt  
os2021202058@ubuntu:~/assignment1$ ls -al  
total 8  
drwxrwxr-x  2 os2021202058 os2021202058 4096 Sep  1 00:04 .  
drwxr-xr-x 16 os2021202058 os2021202058 4096 Sep  1 00:03 ..  
-r--r--r--  1 os2021202058 os2021202058   0 Sep  1 00:04 os_copy.txt  
-rw-rw-r--  1 os2021202058 os2021202058   0 Sep  1 00:04 os.txt  
os2021202058@ubuntu:~/assignment1$ echo "os_2021202058" > os.txt  
os2021202058@ubuntu:~/assignment1$ cat os.txt  
os_2021202058  
os2021202058@ubuntu:~/assignment1$
```

Linux command 에 해당하는 부분이다.

1. mkdir 명령어를 사용하여 "assignment1" 명의 디렉토리를 생성하였다. mkdir 은 Make Directory 의 줄임말로 디렉토리를 생성할 때 사용하는 명령어이다.
2. cd 명령어를 사용하여 assignment1 디렉토리로 이동한 후 touch 명령어를 사용해 "os.txt"명의 빈 파일을 생성하였다. cd 는 change directory 의 줄임말로 현재 작업 중인 디렉토리의 위치를 이동하는 명령어이다. touch 명령어를 옵션 없이 사용할 경우 빈 파일을 생성한다.
3. os.txt 를 os_copy.txt 명으로 복사하기 위해 cp 를 사용하였다. cp 는 copy 의 줄임말로 파일이나 디렉토리를 복사할 때 사용하는 명령어이다.
4. os_copy.txt 의 권한을 모든 대상에게 읽기만 부여하기 위해 chmod 명령어를 사용하였고, a=r 을 통해 모든 대상에게 읽기권한만 주었다. chmod 는 change 와 mode 를 조합한 명령어로 파일의 권한을 변경할 수 있다. a 는 모든 사용자 all 을 뜻하며 r 은 read 를 의미한다.
5. os.txt 에 os_학번을 echo 명령어를 사용해 작성한 후 cat 명령어를 사용하여 터미널에 출력하였다. echo 는 출력 명령어로 사용되며 echo "os_2021202058" > os.txt 를 통해 파일명이 존재하면 파일에 해당 내용을 쓴다. cat 은 하나의 파일이나 여러 개의 파일을 출력할 때 사용한다.

- Assignment 1-2

우분투 내에 kernel 을 다운로드 하고 컴파일 하는 과정이다.

```
os2021202058@ubuntu:~$ sudo apt update
[sudo] password for os2021202058:
Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
499 packages can be upgraded. Run 'apt list --upgradable' to see them.
os2021202058@ubuntu:~$ whoami
os2021202058
os2021202058@ubuntu:~$
```

sudo 명령어를 사용해 root 권한으로 프로그램을 실행하였고, update 를 통해 시스템에서 사용 가능한 소프트웨어 패키지 업데이트 정보를 업데이트 하였다. whoami 명령어를 통해 현재 로그인한 사용자의 이름을 확인할 수 있다.

```

os2021202058@ubuntu:~$ cd /home/os2021202058/Downloads/
os2021202058@ubuntu:~/Downloads$ sudo wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.67.tar.xz
--2023-09-17 00:54:18-- https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.67.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: 103291756 (99M) [application/x-xz]
Saving to: 'linux-4.19.67.tar.xz'

linux-4.19.67.tar.xz      100%[=====] 98.51M  84.5MB/s   in 1.2s

2023-09-17 00:54:20 (84.5 MB/s) - 'linux-4.19.67.tar.xz' saved [103291756/103291756]

os2021202058@ubuntu:~/Downloads$

```

cd 명령어를 통해 현재 작업 중인 디렉토리의 위치를 Downloads 로 이동하였다. sudo 명령어를 사용해 root 권한으로 프로그램을 실행하였고 kernel 를 다운로드 받았다.

```

os2021202058@ubuntu:~/Downloads$ ls
linux-4.19.67.tar.xz
os2021202058@ubuntu:~/Downloads$ tar -Jxvf linux-4.19.67.tar.xz

```

ls 명령어를 통해 현재 디렉토리 내의 파일을 확인하였다. 이어서 tar -Jxvf 명령어를 사용해 kernel source 를 압축 해제 하였다.

```

os2021202058@ubuntu:~/Downloads$ ls
linux-4.19.67  linux-4.19.67.tar.xz
os2021202058@ubuntu:~/Downloads$ cd linux-4.19.67/
os2021202058@ubuntu:~/Downloads/linux-4.19.67$ ls
arch  COPYING  Documentation  fs      ipc      kernel  MAINTAINERS  net      scripts  tools
block CREDITS  drivers        include Kbuild  lib      Makefile     README  security  usr
certs crypto  firmware      init    Kconfig LICENSES  samples  sound     virt
os2021202058@ubuntu:~/Downloads/linux-4.19.67$ vi Makefile
os2021202058@ubuntu:~/Downloads/linux-4.19.67$

```

cd 명령어를 통해 linux-4.19.67 폴더로 이동한 후 ls 명령어를 통해 현재 디렉토리 내의 파일을 확인하였다. vi 는 텍스트 편집기 중 하나로 텍스트 파일을 만들거나 수정할 수 있다. vi 명령어를 사용해 Makefile 을 수정하였다.

```

# SPDX-License-Identifier: GPL-2.0
VERSION = 4
PATCHLEVEL = 19
SUBLEVEL = 67
EXTRAVERSION = -2021202058
NAME = "People's Front"

# *DOCUMENTATION*
# To see a list of typical targets execute "make help"
# More info can be located in ./README
# Comments in this file are targeted only to the developer, do not
# expect to learn how to build the kernel reading this file.

# That's our default target when none is given on the command line
PHONY := _all
_all:

# o Do not use make's built-in rules and variables
# (this increases performance and avoids hard-to-debug behaviour);
# o Look for make include files relative to root of kernel src
MAKEFLAGS += -rR --include-dir=$(CURDIR)

# Avoid funny character set dependencies
-- INSERT --

```

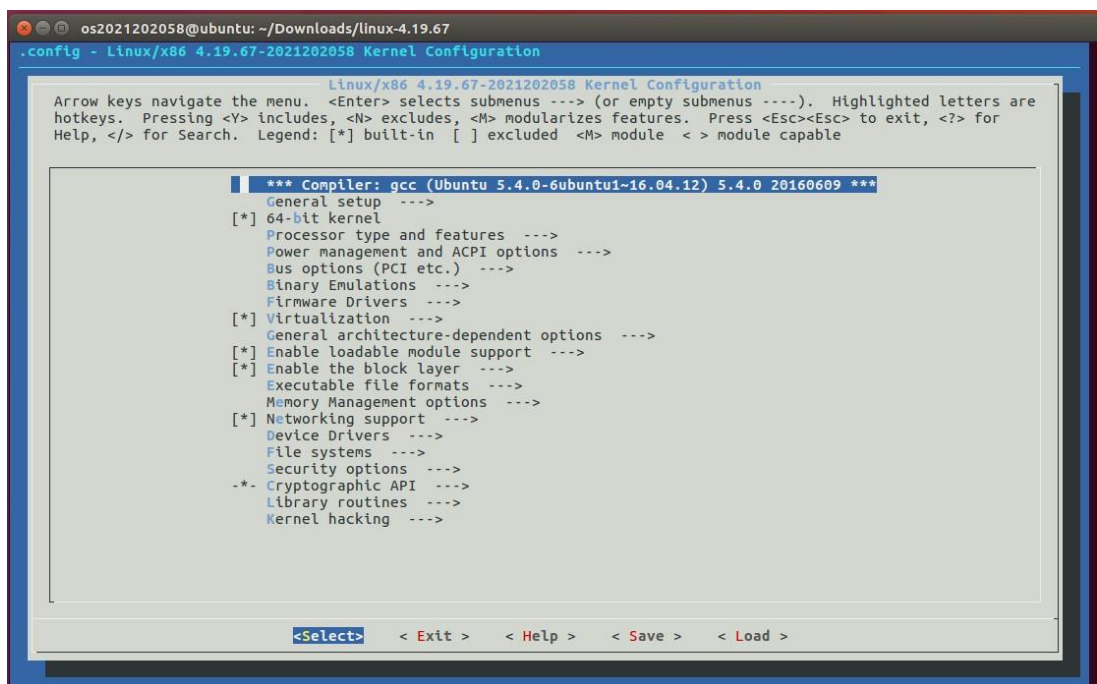
Makefile 이며 EXTRAVERSION 을 -학번으로 수정하였다. EXTRAVERSION 은 linux 에서 커널 버전 정보를 의미한다.

```

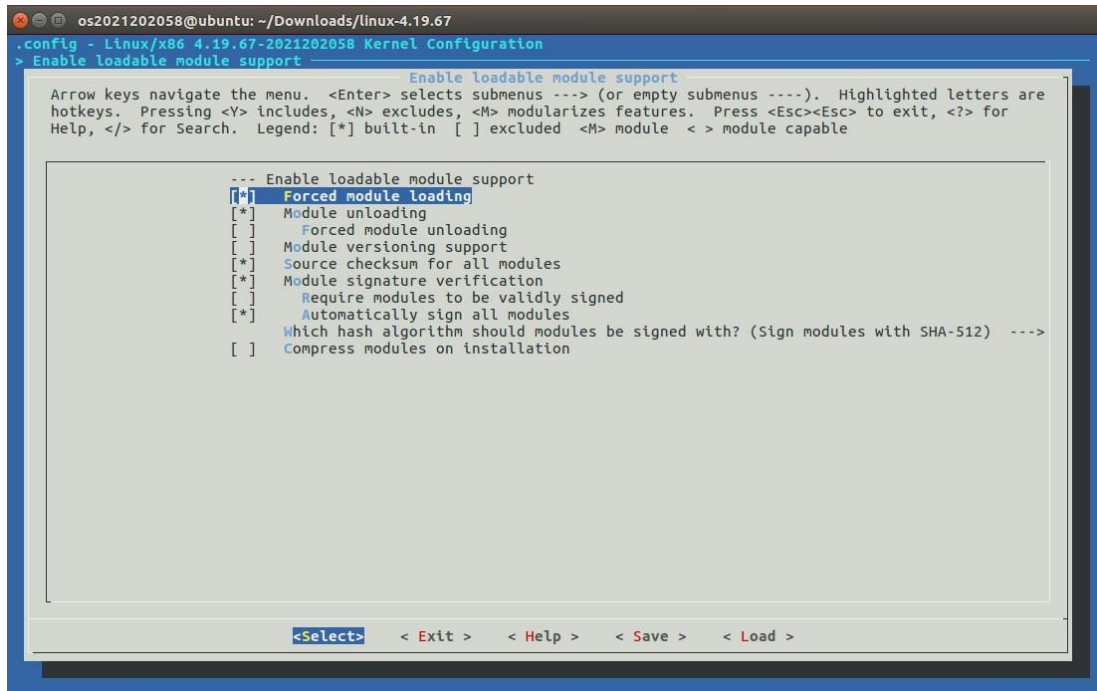
os2021202058@ubuntu:~/Downloads/linux-4.19.67$ sudo apt install build-essential libncurses5-dev bison flex libssl-dev libelf-dev
[sudo] password for os2021202058:
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.1ubuntu2).
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libbison-dev libelf-dev libsigsegv2 libssl-doc libtinfo-dev m4 zlib1g-dev
Suggested packages:
  bison-doc ncurses-doc
The following NEW packages will be installed:
  bison flex libbison-dev libelf-dev libelf-dev libncurses5-dev libsigsegv2 libssl-dev libssl-doc libtinfo-dev m4
  zlib1g-dev
0 upgraded, 12 newly installed, 0 to remove and 12 not upgraded.
Need to get 4,006 kB of archives.
After this operation, 15.5 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

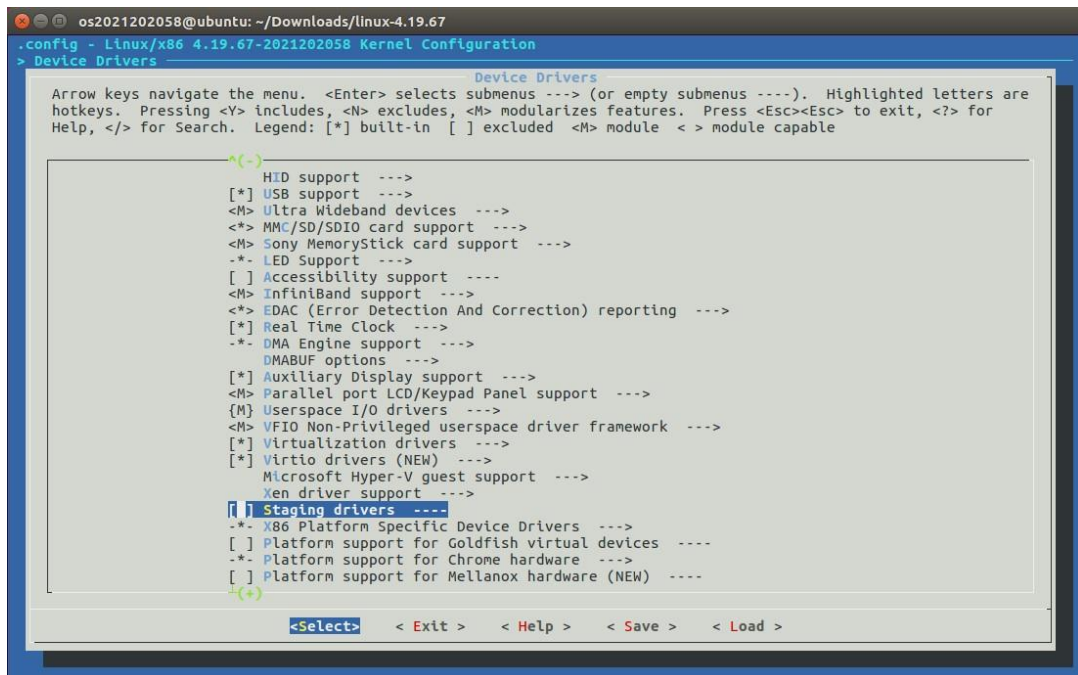
sudo 명령어를 사용해 root 권한으로 프로그램을 실행하였고, kernel 컴파일에 필요한 패키지를 설치하는 부분이다.



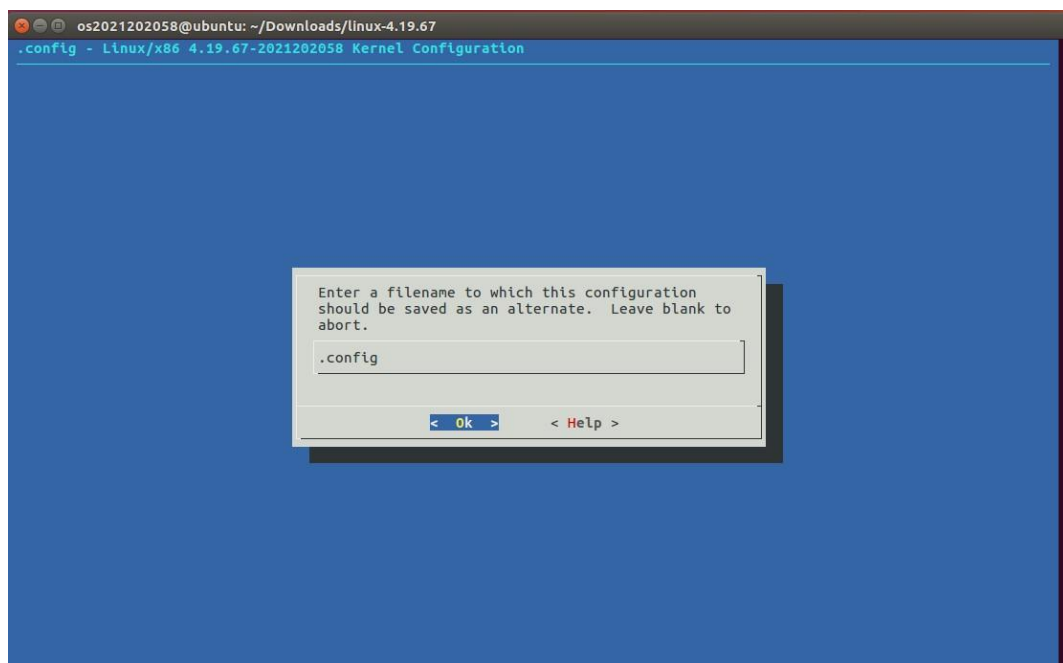
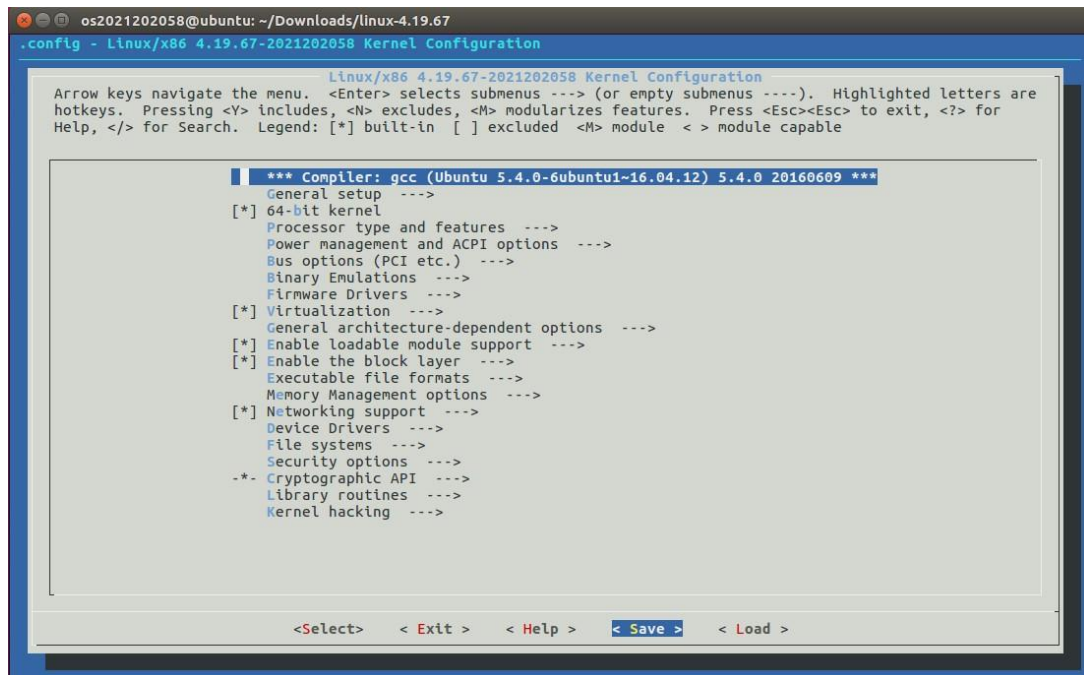
sudo 명령어를 사용해 root 권한으로 프로그램을 실행하였고, sudo make menuconfig 명령어를 통해 리눅스 커널을 컴파일 하기 위해 사용하였다. 인터페이스를 통해 커널 구성을 조정하였다.



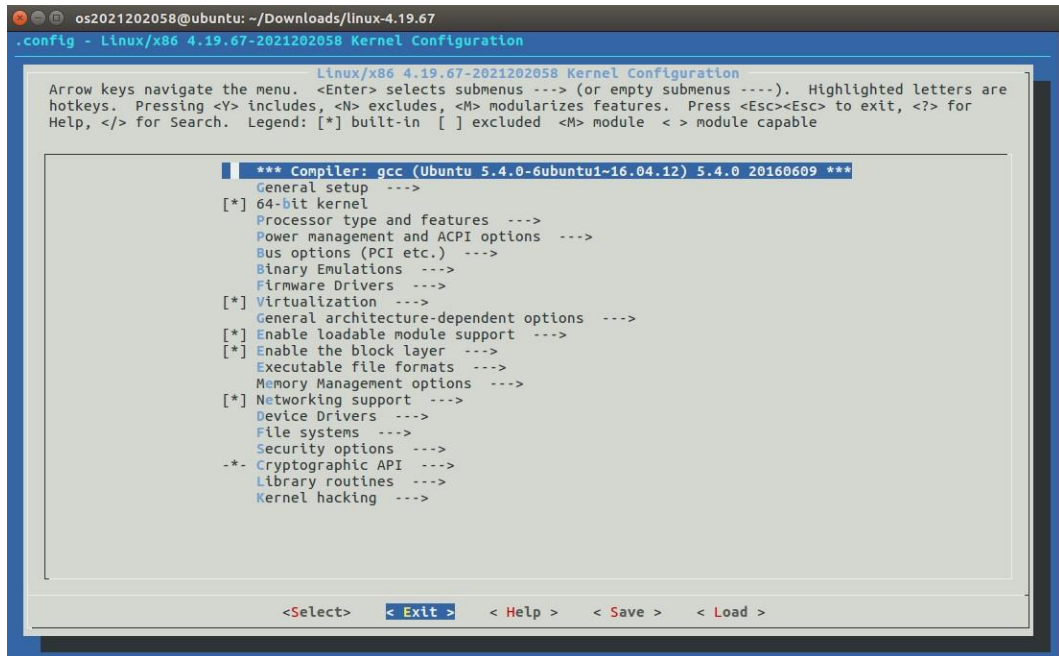
첫 번째로 커널 모듈 적재시 발생할 수 있는 문제를 해결하기 위해 Enable loadable module support 부분에서 Forced module loading 부분을 설정을 적용하였다. Enable loadable module support 기능은 동적인 모듈기능을 사용할 수 있도록 지원한다. Forced module loading 기능은 현재 커널에서 사용중인 모듈이어도 모듈제거를 강제로 할 수 있게 한다.



두 번째로 컴파일 시 문제가 될 수 있는 모듈을 제거하기 위해 Device Drivers 부분에서 Staging drivers 부분을 설정을 해제하였다. Device Drivers 기능은 시스템의 각 장치 드라이브에 대한 옵션에 관한 부분이다.



세 번째로 설정을 .config 파일에 저장하였다.



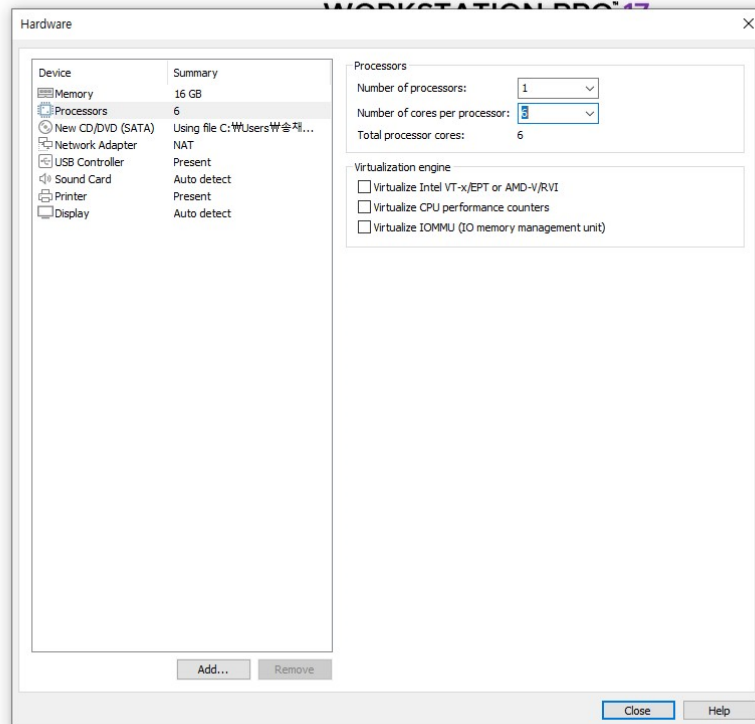
kernel 환경 설정을 끝낸 후 설정을 나갔다.

```
scripts/kconfig/mconf Kconfig
#
# using defaults found in /boot/config-4.15.0-29-generic
#
/boot/config-4.15.0-29-generic:890:warning: symbol value 'm' invalid for HOTPLUG_PCI_SHPC
/boot/config-4.15.0-29-generic:1144:warning: symbol value 'm' invalid for NF_NAT_REDIRECT
/boot/config-4.15.0-29-generic:1147:warning: symbol value 'm' invalid for NF_TABLES_INET
/boot/config-4.15.0-29-generic:1148:warning: symbol value 'm' invalid for NF_TABLES_NETDEV
/boot/config-4.15.0-29-generic:1331:warning: symbol value 'm' invalid for NF_TABLES_IPV4
/boot/config-4.15.0-29-generic:1336:warning: symbol value 'm' invalid for NF_TABLES_ARP
/boot/config-4.15.0-29-generic:1343:warning: symbol value 'm' invalid for NF_NAT_MASQUERADE_IPV4
/boot/config-4.15.0-29-generic:1378:warning: symbol value 'm' invalid for NF_TABLES_IPV6
/boot/config-4.15.0-29-generic:1388:warning: symbol value 'm' invalid for NF_NAT_MASQUERADE_IPV6
/boot/config-4.15.0-29-generic:1416:warning: symbol value 'm' invalid for NF_TABLES_BRIDGE
/boot/config-4.15.0-29-generic:3992:warning: symbol value 'm' invalid for HW_RANDOM_TPM
/boot/config-4.15.0-29-generic:4941:warning: symbol value 'm' invalid for LIRC
/boot/config-4.15.0-29-generic:6167:warning: symbol value 'm' invalid for SND_SOC_INTEL_SST_TOPLEVEL
/boot/config-4.15.0-29-generic:6172:warning: symbol value 'm' invalid for SND_SOC_INTEL_MACH
/boot/config-4.15.0-29-generic:7725:warning: symbol value 'm' invalid for DELL_SMBIOS_WMI
/boot/config-4.15.0-29-generic:7726:warning: symbol value 'm' invalid for DELL_SMBIOS_SMM

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

```
os2021202058@ubuntu:~/Downloads/linux-4.19.67$ make -j12
```

make 명령어는 makefile 을 통해 프로젝트를 자동으로 빌드하고 컴파일 한다. -j 옵션은 병렬로 make 를 실행하도록 하며 동시에 실행할 작업의 수를 의미한다.



Processors 부분을 6 으로 설정해주었기 때문에 위에서 12 개의 작업을 병렬로 실행하도록 하였다.

```
os2021202058@ubuntu:~/Downloads/linux-4.19.67$ sudo make modules_install
[sudo] password for os2021202058:
```

이어서 Module install 부분이다. sudo 명령어를 사용해 root 권한으로 프로그램을 실행하였다. 컴파일 된 모듈을 /lib/modules 아래 설치한다.

```
os2021202058@ubuntu:~/Downloads/linux-4.19.67$ sudo make install
sh ./arch/x86/boot/install.sh 4.19.67-2021202058 arch/x86/boot/bzImage \
System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.19.67-2021202058 /boot/vmlinuz-4.19.67-2021202058
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.19.67-2021202058 /boot/vmlinuz-4.19.67-2021202058
update-initramfs: Generating /boot/initrd.img-4.19.67-2021202058
run-parts: executing /etc/kernel/postinst.d/pm-utils 4.19.67-2021202058 /boot/vmlinuz-4.19.67-2021202058
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 4.19.67-2021202058 /boot/vmlinuz-4.19.67-2021202058
run-parts: executing /etc/kernel/postinst.d/update-notifier 4.19.67-2021202058 /boot/vmlinuz-4.19.67-2021202058
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 4.19.67-2021202058 /boot/vmlinuz-4.19.67-2021202058
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-4.19.67-2021202058
Found initrd image: /boot/initrd.img-4.19.67-2021202058
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
os2021202058@ubuntu:~/Downloads/linux-4.19.67$
```

마지막으로 compile 된 kernel 을 boot Loader 에 등록하는 부분이다. sudo 명령어를 사용해 root 권한으로 프로그램을 실행하였으며 컴파일 된 커널을 부트 로더에 등록한다.

```
os2021202058@ubuntu:~/Downloads/linux-4.19.67$ sudo vi /etc/default/grub
os2021202058@ubuntu:~/Downloads/linux-4.19.67$
```

```

os20211202058@ubuntu: ~/Downloads/linux-4.19.67
# 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=false
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
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"
~
~
~

```

이어서 grub 을 설정하였다. grub 은 대부분의 리눅스 배포판에서 사용하는 부트로더로 실행과 동시에 grub 설정 파일을 읽고 boot menu 를 화면에 출력한다. 또한 사용자가 선택한 kernel image 를 메모리에 적재하고 제어권을 커널에 전달한다. sudo 명령어를 사용해 root 권한으로 프로그램을 실행하였으며, vi /etc/default/grub 명령어는 grub 부트로더의 설정파일을 편집하기 위한 부분이다. grub 설정 파일에서 GRUB_HIDDEN_TIMEOUT = 0 , 즉 0 초 후에 자동으로 부팅하도록 구성된 것을 비활성화하였다. 또한 GRUB_HIDDEN_TIMEOUT_QUIET = false 로 설정하여 메뉴가 표시되지 않게 하였다.

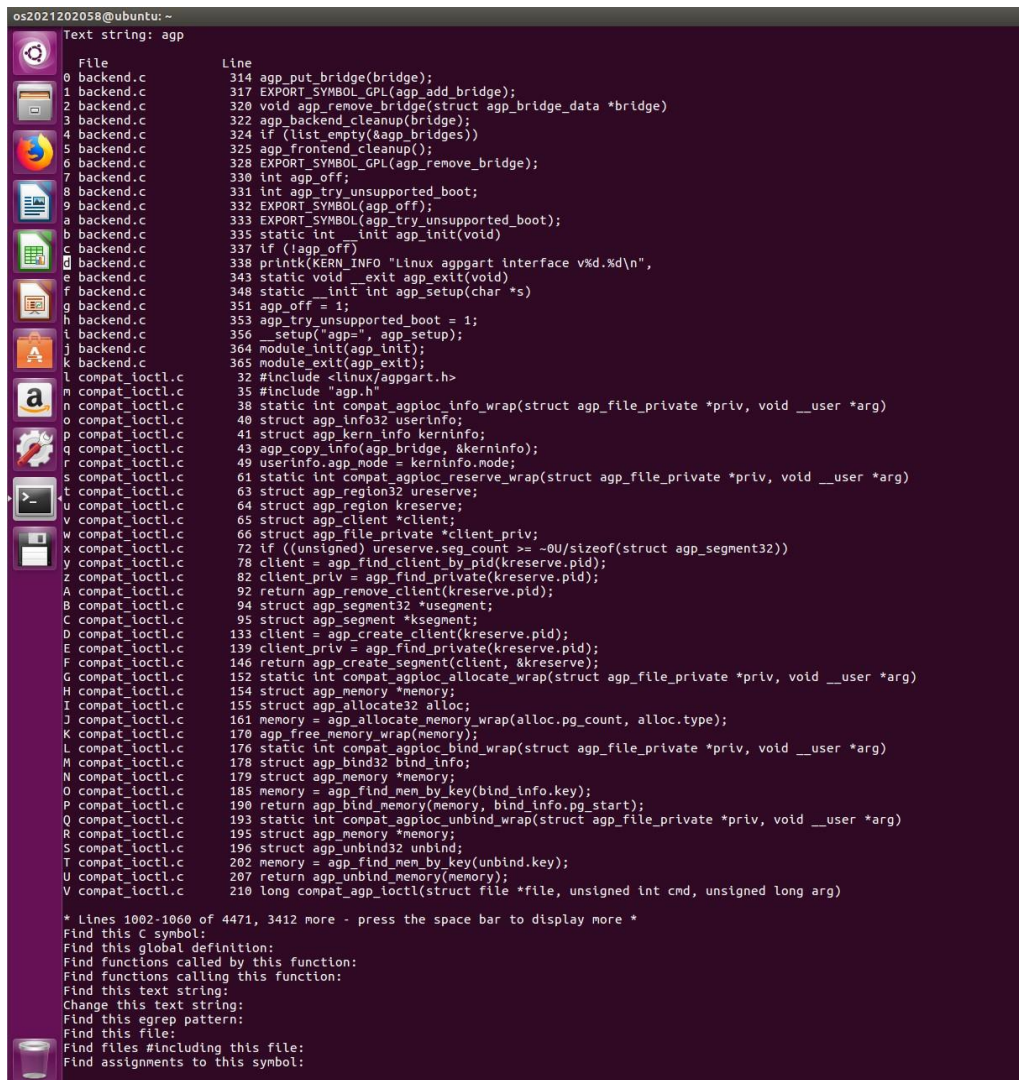
```
os2021202058@ubuntu: ~  
os2021202058@ubuntu:~$ uname -r  
4.19.67-2021202058  
os2021202058@ubuntu:~$
```

재부팅 후 `uname` 명령어를 통해 시스템 정보를 조회하고 표시하였다. `-r` 옵션을 사용하여 `-kernel-release`, 즉 커널 버전 및 릴리스 정보를 표시한다. 따라서 4.19.67 커널로 재부팅 한 결과 커널 버전과 릴리스가 4.19.67 과 학번으로 나온 것을 확인할 수 있다.

- Assignment 1-3

```
os2021202058@ubuntu: ~/Downloads/linux-4.19.67
os2021202058@ubuntu:~/Downloads/linux-4.19.67$ cscope -R
```

cscope 는 linux 에서 변수, 함수, 구조체 등을 검색하기 위해 사용한다. -R 옵션은 현재 디렉터리부터 모든 하위 디렉터리의 소스 코드 파일을 대상으로 함을 의미한다. 이를 통해 Database 를 구축하고 실행할 수 있다.



Find this text string 부분에 agp 를 검색한 결과이다.

```
static int __init agp_init(void)
{
    if (!agp_off)
        printk(KERN_INFO "os2021202058_Linux agpgart interface v%d.%d\n",
            AGPGART_VERSION_MAJOR, AGPGART_VERSION_MINOR);
    printk(KERN_INFO "os2021202058_arg in __init agp_init ( void )\n")
    return 0;
}
```

위 코드는 Linux agp...이 실행되는 지점으로 커널 메시지가 Linux Kernel Message 양식과 같이 출력되도록 커널 코드를 수정한 부분이다. 해당 소스코드의 path 는 drivers/char/agp/backend.c 이다. 또한 Linux agp...을 실행시키는 함수의 함수명인 _init agp_init 과 argument 의 값인 void 를 출력하도록 커널 코드를 수정하였다.

위의 코드에서 printk 함수를 사용하여 출력되도록 하였는데, 커널에서 메시지를 출력하는 함수이다. KERN_INFO 는 informational 을 의미한다.

module comlile 과 reboot 과정을 거친 후 dmesg | grep 으로 확인한 결과이다.

```
os2021202058@ubuntu:~$ dmesg | grep "os2021202058" -n
11:[ 0.000000] Linux version 4.19.67-2021202058 (os2021202058@ubuntu) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1-16.04.10)) #2 SMP Sun Sep 17 07:07:14 PDT 2023
1403:[ 0.747309] os2021202058_Linux agpgart interface v0.103
1404:[ 0.747309] os2021202058_arg tn _init agp_init ( void )
os2021202058@ubuntu:~$
```

dmesg 는 printk 로 출력한 내용을 확인할 수 있으며 grep 은 텍스트를 검색하고 필터링하는데 사용하는 명령어이다. 해당 명령어로 확인하였을 때 메시지가 중복되지 않고 한 set 로 연달아 출력이 되는 것을 확인할 수 있다.

3. 고찰

assignment 1-1 을 진행하면서 시스템프로그래밍에서 자주 사용하였던 명령어가 아닌 chmod, echo 등을 사용하여 처음에는 익숙하지 않았던 것 같다. 또한 assignment 1-2 에 해당하는 커널 컴파일 과정에서 apt update 를 하고 apt upgrade 를 해주어 우분투 버전이 16.04.5 가 아닌 16.04.12 로 upgrade 되었다. upgrade 를 해주는 과정에서 버전이 바뀌어 우분투를 다시 깔았다. 마지막으로 assignment 1-3 의 결과에서 맨 윗줄에 linux version 과 gcc version 등이 났는데, sudo 명령어를 통해 과제를 진행하였기 때문에 뜬 것 같다는 생각이 들었다.

4. Reference

리눅스 커널 컴파일 https://www.linux.co.kr/bbs/board.php?bo_table=lecture&wr_id=2360

운영체제실습 강의자료 참조