

# 운영체제실습 보고서

## Assignment 1

학 과 : 컴퓨터정보공학부

담당교수 : 김태석 교수님

학 번 : 2018202018

이 름 : 유승재

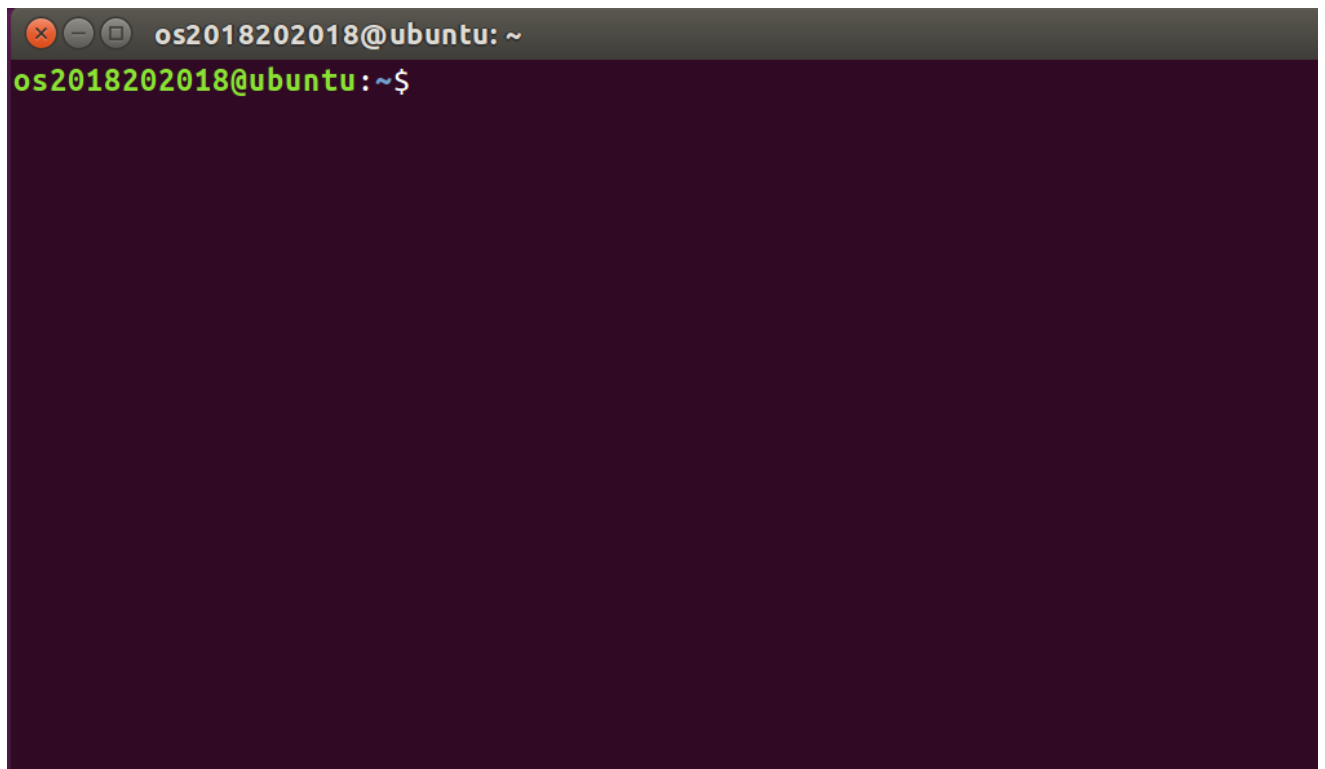
## 1. Introduction

이번 과제는 이전 시스템 프로그래밍 수업에서 진행했던 거의 복습 개념으로 linux 환경을 구축하기 위해 vmware 설치와 linux command의 사용 방법을 알아 보았습니다. 또한 우분투 내에서 커널을 다운로드 하여 커널을 직접 컴파일을 해보고, 또한 커널 내의 파일을 cscope, ctags 등의 명령어들을 이용하여 출력해봄으로써 커널을 직접 다루기 위해 기본적인 함수나 명령어를 사용해보는 과제입니다.

## 2. Result

### 1) Assignment 1-1

Linux Installation

A terminal window with a dark purple background. The title bar shows window control icons and the text 'os2018202018@ubuntu: ~'. The terminal displays the prompt 'os2018202018@ubuntu:~\$' in green text. The rest of the terminal area is empty, indicating the installation process is either just starting or has completed without further output shown.

```
os2018202018@ubuntu: ~$
```

## Linux command

```
os2018202018@ubuntu: ~/assignment1
os2018202018@ubuntu:~$ mkdir assignment1
os2018202018@ubuntu:~$ cd assignment1
os2018202018@ubuntu:~/assignment1$ touch os.txt
os2018202018@ubuntu:~/assignment1$ ls
os.txt
os2018202018@ubuntu:~/assignment1$ cp os.txt os_copy.txt
os2018202018@ubuntu:~/assignment1$ ls
os_copy.txt  os.txt
os2018202018@ubuntu:~/assignment1$ ls -al
total 8
drwxrwxr-x  2 os2018202018 os2018202018 4096 Sep 17 01:05 .
drwxr-xr-x 18 os2018202018 os2018202018 4096 Sep 17 01:04 ..
-rw-rw-r--  1 os2018202018 os2018202018   0 Sep 17 01:05 os_copy.txt
-rw-rw-r--  1 os2018202018 os2018202018   0 Sep 17 01:05 os.txt
os2018202018@ubuntu:~/assignment1$ chmod 444 os_copy.txt
os2018202018@ubuntu:~/assignment1$ ls -al
total 8
drwxrwxr-x  2 os2018202018 os2018202018 4096 Sep 17 01:05 .
drwxr-xr-x 18 os2018202018 os2018202018 4096 Sep 17 01:04 ..
-r--r--r--  1 os2018202018 os2018202018   0 Sep 17 01:05 os_copy.txt
-rw-rw-r--  1 os2018202018 os2018202018   0 Sep 17 01:05 os.txt
os2018202018@ubuntu:~/assignment1$ echo "os_2018202018" >> os.txt
os2018202018@ubuntu:~/assignment1$ cat os.txt
os_2018202018
os2018202018@ubuntu:~/assignment1$
```

## 2) Assignment 1-2

### Kernel 4.19.67 Compie

```
os2018202018@ubuntu:~/Downloads$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu xenial-security 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.
os2018202018@ubuntu:~/Downloads$
```

### Linux 4.19.67 압축 파일 다운로드

```
os2018202018@ubuntu:~/Downloads
os2018202018@ubuntu:~/Downloads$ sudo wget http://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.67.tar.xz
URL transformed to HTTPS due to an HSTS policy
--2023-09-12 00:48:37-- 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  4.99MB/s   in 27s

2023-09-12 00:49:05 (3.67 MB/s) - 'linux-4.19.67.tar.xz' saved [103291756/103291756]

os2018202018@ubuntu:~/Downloads$
```

### Linux 4.19.67 압축 해제

```

linux-4.19.67/virt/lib/irqbypass.c
os2018202018@ubuntu:~/Downloads$ ls
linux-4.19.67  linux-4.19.67.tar.xz
os2018202018@ubuntu:~/Downloads$

```

Vi Makefile을 통해 Makefile 수정

```

# SPDX-License-Identifier: GPL-2.0
VERSION = 4
PATCHLEVEL = 19
SUBLEVEL = 67
EXTRAVERSION = -2018202018
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
unexport LC_ALL
LC_COLLATE=C
LC_NUMERIC=C
export LC_COLLATE LC_NUMERIC

# Avoid interference with shell env settings
unexport GREP_OPTIONS

```

Kernel 환경 설정

```

os2018202018@ubuntu:~/Downloads/linux-4.19.67$ sudo apt install build-essential libncurses5-dev bison flex libssl-dev libelf-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.1ubuntu2).
The following additional packages will be installed:
  libbison-dev libelf1 libfl-dev libsigsegv2 libssl-doc libssl1.0.0 libtinfo-dev m4 zlib1g zlib1g-dev
Suggested packages:
  bison-doc ncurses-doc
The following NEW packages will be installed:
  bison flex libbison-dev libelf-dev libfl-dev libncurses5-dev libsigsegv2 libssl-dev libssl-doc libtinfo-dev m4 zlib1g-dev
The following packages will be upgraded:
  libelf1 libssl1.0.0 zlib1g
3 upgraded, 12 newly installed, 0 to remove and 496 not upgraded.
Need to get 5,183 kB of archives.
After this operation, 15.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

```
os2018202018@ubuntu: ~/Downloads/linux-4.19.67
.config - Linux/x86 4.19.67-2018202018 Kernel Configuration

Linux/x86 4.19.67-2018202018 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 [ ] excluded <M> module < > module capable

*** Compiler: gcc (Ubuntu 5.4.0-6ubuntu1~16.04.10) 5.4.0 20160609 ***
General setup --->
[*] 64-bit kernel
Processor type and features --->
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 --->
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 >
```

```
os2018202018@ubuntu: ~/Downloads/linux-4.19.67
.config - Linux/x86 4.19.67-2018202018 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 [ ] 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
Which hash algorithm should modules be signed with? (Sign modules with SHA-512)
[ ] Compress modules on installation
```

```
^(-)
<M> VFIO Non-Privileged userspace driver framework --->
[*] Virtualization drivers --->
[*] Virtio drivers (NEW) --->
Microsoft Hyper-V guest support --->
Xen driver support --->
[ ] Staging drivers ----
-* X86 Platform Specific Device Drivers --->
[ ] Platform support for Goldfish virtual devices ----
-* Platform support for Chrome hardware --->
[ ] Platform support for Mellanox hardware (NEW) ----
Common Clock Framework --->
[*] Hardware Spinlock drivers ----
Clock Source drivers ----
-* Mailbox Hardware Support --->
[*] IOMMU Hardware Support --->
Remoteproc drivers --->
Pmsg drivers --->
```

Enter a filename to which this configuration should be saved as an alternate. Leave blank to abort.

.config

< Ok >

< Help >

```
os2018202018@ubuntu:~/Downloads/linux-4.19.67$ sudo make menuconfig
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'.
os2018202018@ubuntu:~/Downloads/linux-4.19.67$
```

Module install

```
os2018202018@ubuntu:~/Downloads/linux-4.19.67
os2018202018@ubuntu:~/Downloads/linux-4.19.67$ sudo make modules_install
```

## Kernel Boot Loader에 등록

```
os2018202018@ubuntu: ~/Downloads/linux-4.19.67
os2018202018@ubuntu:~/Downloads/linux-4.19.67$ sudo make install
sh ./arch/x86/boot/install.sh 4.19.67-2018202018 arch/x86/boot/bzImage \
    System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.19.67-2018202018 /boot/vmlinuz-4.19.67-2018202018
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.19.67-2018202018 /boot/vmlinuz-4.19.67-2018202018
update-initramfs: Generating /boot/initrd.img-4.19.67-2018202018

run-parts: executing /etc/kernel/postinst.d/pm-utils 4.19.67-2018202018 /boot/vmlinuz-4.19.67-2018202018
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 4.19.67-2018202018 /boot/vmlinuz-4.19.67-2018202018
run-parts: executing /etc/kernel/postinst.d/update-notifier 4.19.67-2018202018 /boot/vmlinuz-4.19.67-2018202018
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 4.19.67-2018202018 /boot/vmlinuz-4.19.67-2018202018
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-4.19.67-2018202018
Found initrd image: /boot/initrd.img-4.19.67-2018202018
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
os2018202018@ubuntu:~/Downloads/linux-4.19.67$
os2018202018@ubuntu:~/Downloads/linux-4.19.67$
```

## Grub 설정 파일 수정

```
os2018202018@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

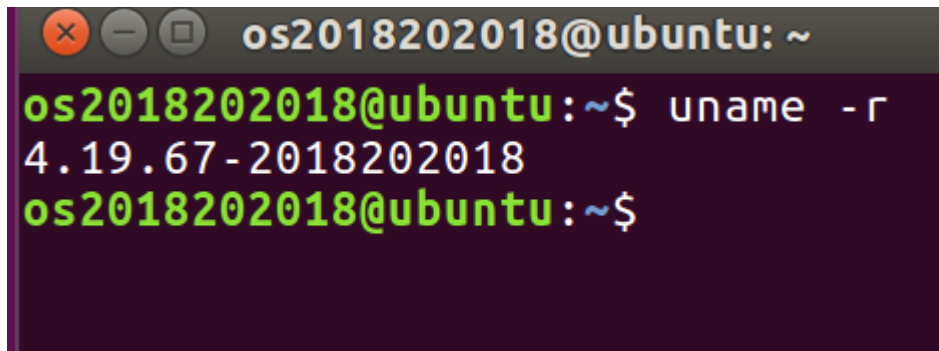
# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# 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"

~
~
```

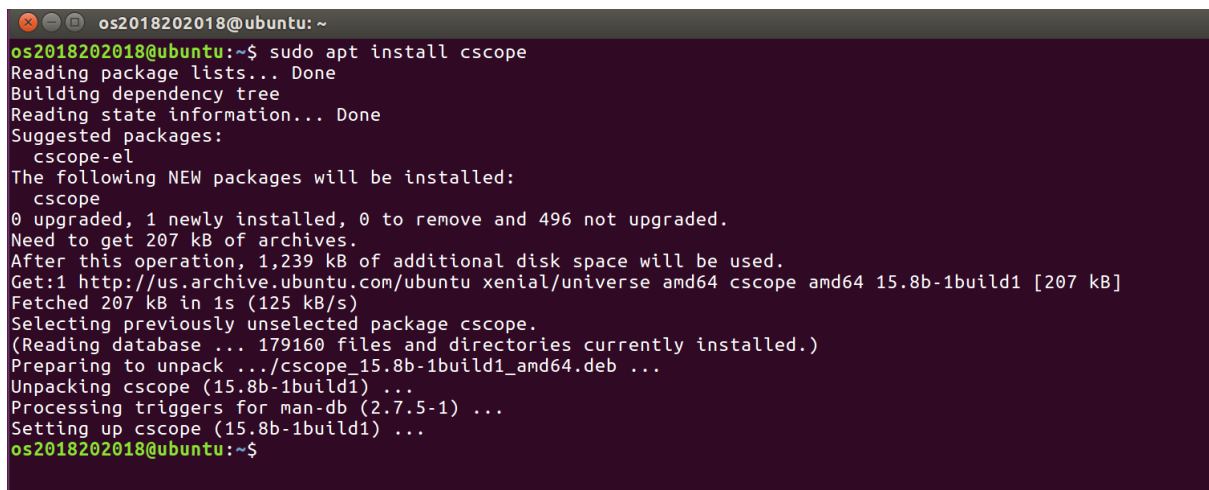
Uname -r 명령어를 실행할 경우 이렇게 나오는 것을 확인할 수 있습니다.

A terminal window with a dark purple background. The title bar shows a window icon, a close button, and the text 'os2018202018@ubuntu: ~'. The prompt is 'os2018202018@ubuntu:~\$'. The command 'uname -r' has been entered, and the output '4.19.67-2018202018' is displayed. The prompt is now 'os2018202018@ubuntu:~\$' again.

```
os2018202018@ubuntu:~$ uname -r
4.19.67-2018202018
os2018202018@ubuntu:~$
```

### 3) Assignment 1-3

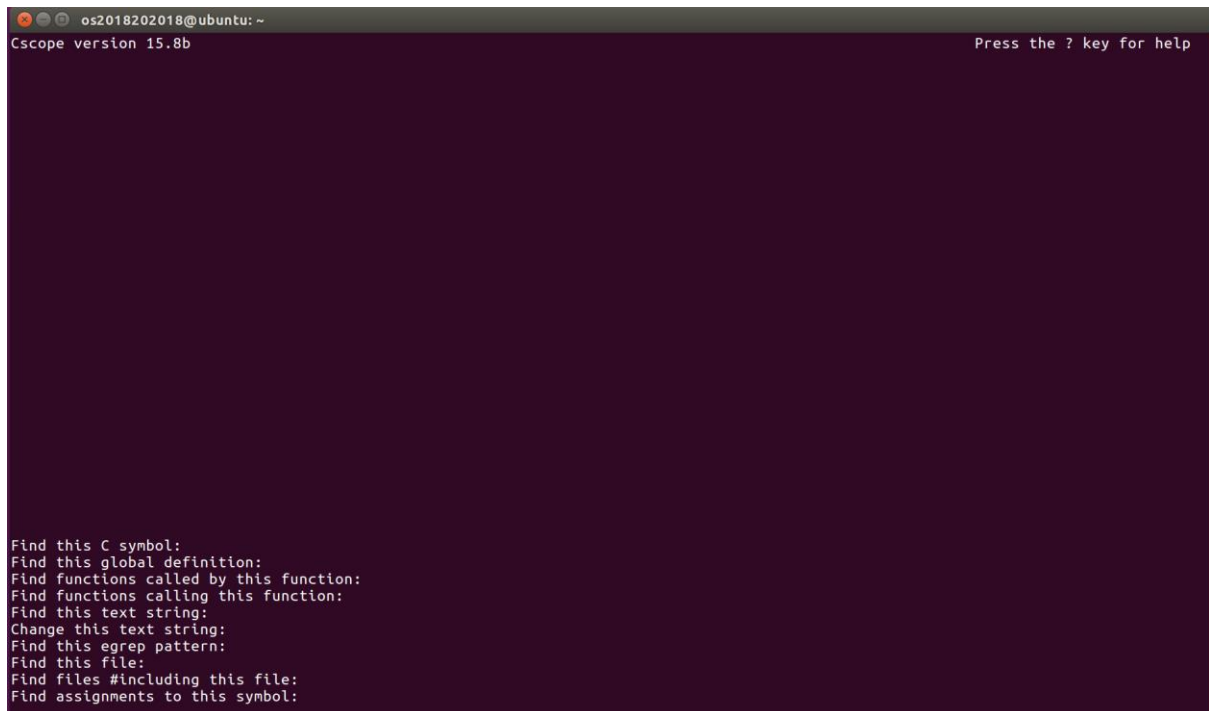
Cscope install

A terminal window with a dark purple background. The title bar shows a window icon, a close button, and the text 'os2018202018@ubuntu: ~'. The prompt is 'os2018202018@ubuntu:~\$'. The command 'sudo apt install cscope' has been entered. The output shows the package lists being read, the dependency tree being built, and the state information being read. It then suggests the 'cscope-el' package. The following NEW packages will be installed: 'cscope'. It shows that 0 packages are upgraded, 1 is newly installed, and 0 are to be removed. It also shows the disk space requirements and the download of the package from the archive. Finally, it shows the package being unpacked and set up. The prompt is now 'os2018202018@ubuntu:~\$' again.

```
os2018202018@ubuntu:~$ 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 496 not upgraded.
Need to get 207 kB of archives.
After this operation, 1,239 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 cscope amd64 15.8b-1build1 [207 kB]
Fetched 207 kB in 1s (125 kB/s)
Selecting previously unselected package cscope.
(Reading database ... 179160 files and directories currently installed.)
Preparing to unpack .../cscope_15.8b-1build1_amd64.deb ...
Unpacking cscope (15.8b-1build1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up cscope (15.8b-1build1) ...
os2018202018@ubuntu:~$
```



Cscope -R로 cscope 실행



```
os2018202018@ubuntu: ~  
Cscope version 15.8b  
Press the ? key for help  
  
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:
```

cscope에서 Linux agp 검색

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

결과 출력된 곳으로 이동

```
os2018202018@ubuntu: ~
Text string: Linux agp

File      Line
0 backend.c 338 printk(KERN_INFO "os2018202018_Linux agpgart interface v%d.%d\n",
1 backend.c 338 printk(KERN_INFO "os2018202018_Linux agpgart interface v%d.%d\n",
```

Backend.c 파일의 agp\_init(void) 함수 내의 내용 수정

```
static int __init agp_init(void)
{
    if (!agp_off){
        printk(KERN_INFO "os2018202018_Linux agpgart interface v%d.%d\n",
                    AGPGART_VERSION_MAJOR, AGPGART_VERSION_MINOR);
        printk(KERN_INFO "os2018202018_arg in agp_init ( void )\n");
    }
    return 0;
}
```

Dmesg | grep 을 통해 메시지 출력 확인

```
os2018202018@ubuntu: ~
os2018202018@ubuntu:~$ dmesg | grep "os2018202018" -n
1:[    0.000000] Linux version 4.19.67-2018202018 (os2018202018@ubuntu) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.10)
) #3 SMP Fri Sep 15 00:17:19 PDT 2023
1263:[    7.048317] os2018202018_Linux agpgart interface v0.103
1264:[    7.048317] os2018202018_arg in agp_init(void)
os2018202018@ubuntu:~$
```

Backend.c 파일의 위치를 찾기 위해 find 명령어를 이용

```
os2018202018@ubuntu: ~
os2018202018@ubuntu:~$ sudo find / -name backend.c
/home/os2018202018/Downloads/linux-4.19.67/drivers/char/agp/backend.c
/home/os2018202018/.local/share/Trash/files/linux-4.19.67/drivers/char/agp/backend.c
find: '/run/user/1000/gvfs': Permission denied
os2018202018@ubuntu:~$
```

Find 명령어를 통해

/home/os2018202018/Downloads/linux-4.19.67/drivers/char/agp/backend.c의 파일임  
을 확인할 수 있었습니다.

```
os2018202018@ubuntu: ~/Downloads/linux-4.19.67/drivers/char/agp
os2018202018@ubuntu:~$ sudo find / -name backend.c
/home/os2018202018/Downloads/linux-4.19.67/drivers/char/agp/backend.c
/home/os2018202018/.local/share/Trash/files/linux-4.19.67/drivers/char/agp/backend.c
find: '/run/user/1000/gvfs': Permission denied
os2018202018@ubuntu:~$ cd /home/os2018202018/Downloads/linux-4.19.67/drivers/char/agp
os2018202018@ubuntu:~/Downloads/linux-4.19.67/drivers/char/agp$ ls
agp.h      amd-k7-agp.c  compat_ioctl.c  frontend.o  intel-agp.c  isoch.c      modules.order  sis-agp.ko  uninorth-agp.c
ali-agp.c  ati-agp.c    compat_ioctl.h  generic.c   intel-agp.h  isoch.o      nvidia-agp.c  sis-agp.mod.c  via-agp.c
alpha-agp.c  backend.c    compat_ioctl.o  generic.o   intel-agp.o  Kconfig      parisc-agp.c  sis-agp.mod.o  via-agp.o
amd64-agp.c  backend.o    efficeon-agp.c  hp-agp.c    intel-gtt.c  Makefile     sgi-agp.c     sis-agp.o     sworks-agp.c
amd64-agp.o  builtin.a    frontend.c      i460-agp.c  intel-gtt.o  modules.builtin  sis-agp.c     sworks-agp.c
os2018202018@ubuntu:~/Downloads/linux-4.19.67/drivers/char/agp$
```

### 3. 고찰

이번 과제를 통해 시스템 프로그래밍 과목에서 익혔던 linux command를 다시 한번 상기시킬 수 있게 되는 과제가 되었습니다. 또한 이전까지는 직접 컨트롤해보지 못했던 kernel을 컴파일하고 컴파일한 kernel 내의 파일을 수정하는 방식으로 출력 메시지에 대해 변화를 주는 방식으로 kernel에 대해서 간접적으로 알게 되는 과제가 되었습니다.

### 4. Reference

2023-2\_OSLab\_03\_Linux\_Kernel.pdf

2023-2\_OSLab\_02\_Linux\_Command.pdf