



운영체제  
Assignment 1 과제

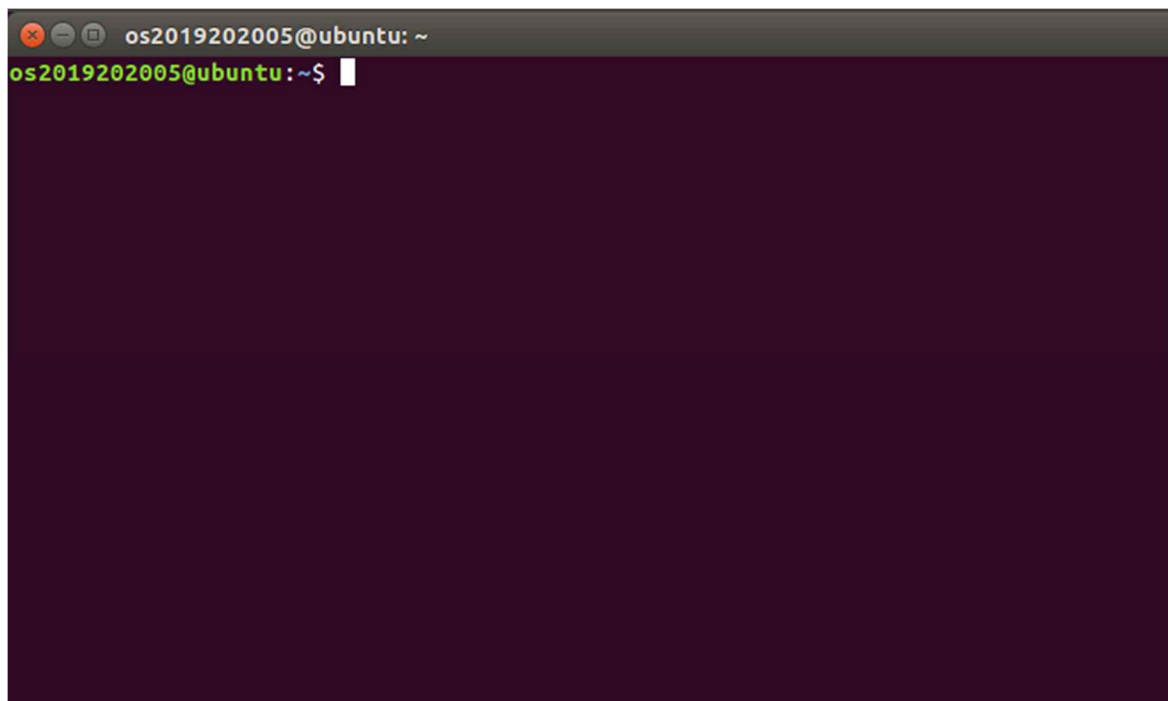
수업      명 : 운영체제  
과제      이름 : assignment1  
담당 교수님 : 최상호 교수님  
학          번 : 2019202005  
이          름 : 남종식

## Assignment 1-1

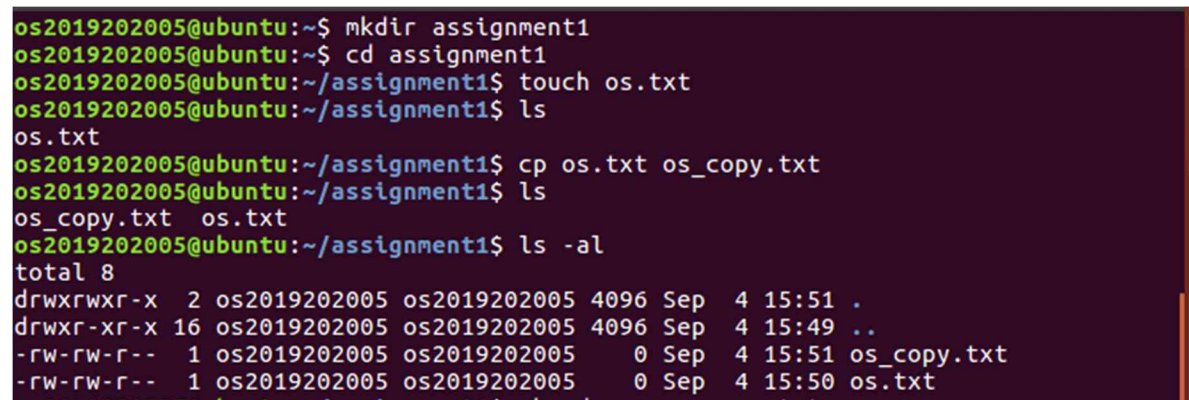
### Introduction

이번 과제는 계정 ID를 os학번으로 설정한 후 잘 설정되었는 지 터미널에 출력해보는 과제와 몇가지의 리눅스 명령어를 과제의 요구 사항에 따라서 사용해 보는 과제입니다. 과제 요구 사항에 따라 약 5가지의 리눅스 명령어를 사용해 과 pdf에 예시 화면과 동일하게 출력되어야 합니다. 이를 통해 이전에 배운 리눅스 명령어들을 학습할 수 있습니다.

### Result



계정 ID를 os학번으로 설정한 후 터미널을 캡처한 화면입니다.



1. mkdir명령어를 통해 assignment1 디렉토리를 생성했습니다.
2. cd명령어를 통해 assignment1 디렉토리로 이동 후 touch명령어를 통해 os.txt 빈 파일을 생성했습니다.
3. cp명령어를 통해 os.txt를 os\_copy.txt로 복사했습니다.

```
os2019202005@ubuntu:~/assignment1$ chmod 444 os_copy.txt
os2019202005@ubuntu:~/assignment1$ ls -al
total 12
drwxrwxr-x  2 os2019202005 os2019202005 4096 Sep  4 15:57 .
drwxr-xr-x 16 os2019202005 os2019202005 4096 Sep  5 01:00 ..
-r--r--r--  1 os2019202005 os2019202005   0 Sep  4 15:51 os_copy.txt
-rw-rw-r--  1 os2019202005 os2019202005  14 Sep  4 15:57 os.txt
os2019202005@ubuntu:~/assignment1$ echo os_2019202005 > os.txt
os2019202005@ubuntu:~/assignment1$ cat os.txt
os_2019202005
```

4. chmod 444 명령어를 통해 os\_copy.txt의 권한을 모든 대상에게 읽기만 부여했습니다.
5. echo명령어를 통해 os.txt에 os\_학번을 작성하고 cat명령어를 통해 os.txt에 작성된 os\_학번을 터미널에 출력했습니다.

## 고찰

과제를 진행하면서 저번 학기 시스템프로그래밍 수업에서 배웠던 리눅스 명령어를 사용하여 디렉토리 및 파일을 생성, 복사하고, 권한을 변경하며, 파일에 내용을 작성하고 출력하는 방법을 상기시킬 수 있었습니다. 리눅스 명령어를 오랜만에 사용해봤는데 대부분 명령어들이 아직 기억이 나서 과제를 진행함에 있어서 큰 문제는 없었습니다.

## Reference

2023 3-1학기 시스템프로그래밍 강의자료

## Assignment 1-2

### Introduction

이번 과제는 우분투에서 kernel을 다운로드 받은 후에 컴파일을 진행하는 것입니다. 먼저 커널 소스를 다운로드 받고 이를 압축해제를 합니다. 그 후 커널 환경 설정을 진행하고 컴파일을 분할하여 수행할 thread 수를 전해 커널 컴파일을 합니다. 컴파일 된 모듈을 이동시키고 커널을 부트로더에 등록하면 모든 과정이 완료됩니다.

### Result

```
os2019202005@ubuntu:~/Downloads$ ls
code_1.82.1-1694163687_amd64.deb  linux-4.19.67  linux-4.19.67.tar.xz
os2019202005@ubuntu:~/Downloads$ cd linux-4.19.67/
os2019202005@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  mm          samples  sound    virt
os2019202005@ubuntu:~/Downloads/linux-4.19.67$ vi Makefile
```

sudo wget <https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.67.tar.xz>

명령어를 통해 커널 소스를 다운로드 합니다. 이때 sudo를 통해 root권한을 가질 수 있습니다. tar -Jxvf linux-4.19.67.tar.xz 명령어를 통해 압축 해제를 합니다. cd Linux-4.19.67 명령어를 통해 디렉토리를 이동합니다. vi Makefile을 통해 Makefile을 수정합니다.

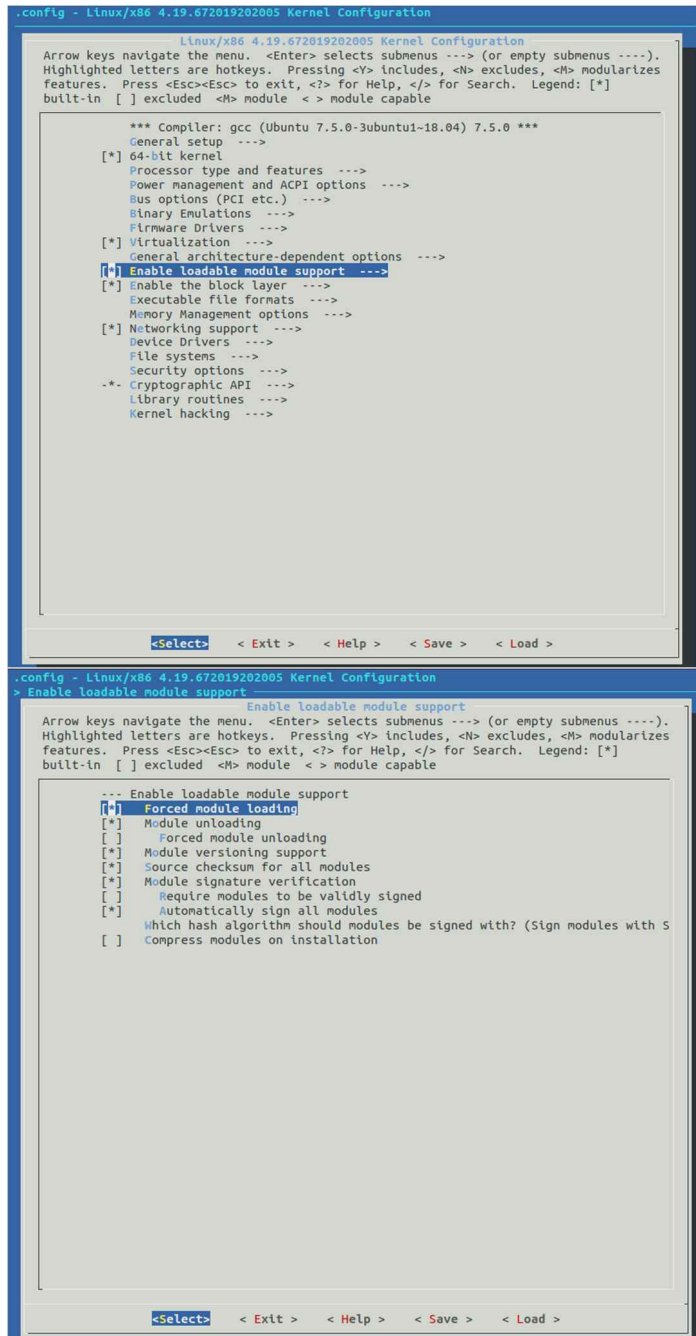
```
# SPDX-License-Identifier: GPL-2.0
VERSION = 4
PATCHLEVEL = 19
SUBLEVEL = 67
EXTRAVERSION = -2019202005
NAME = "People's Front"
```

EXTRAVERSION에 -2019202005로 수정했습니다.

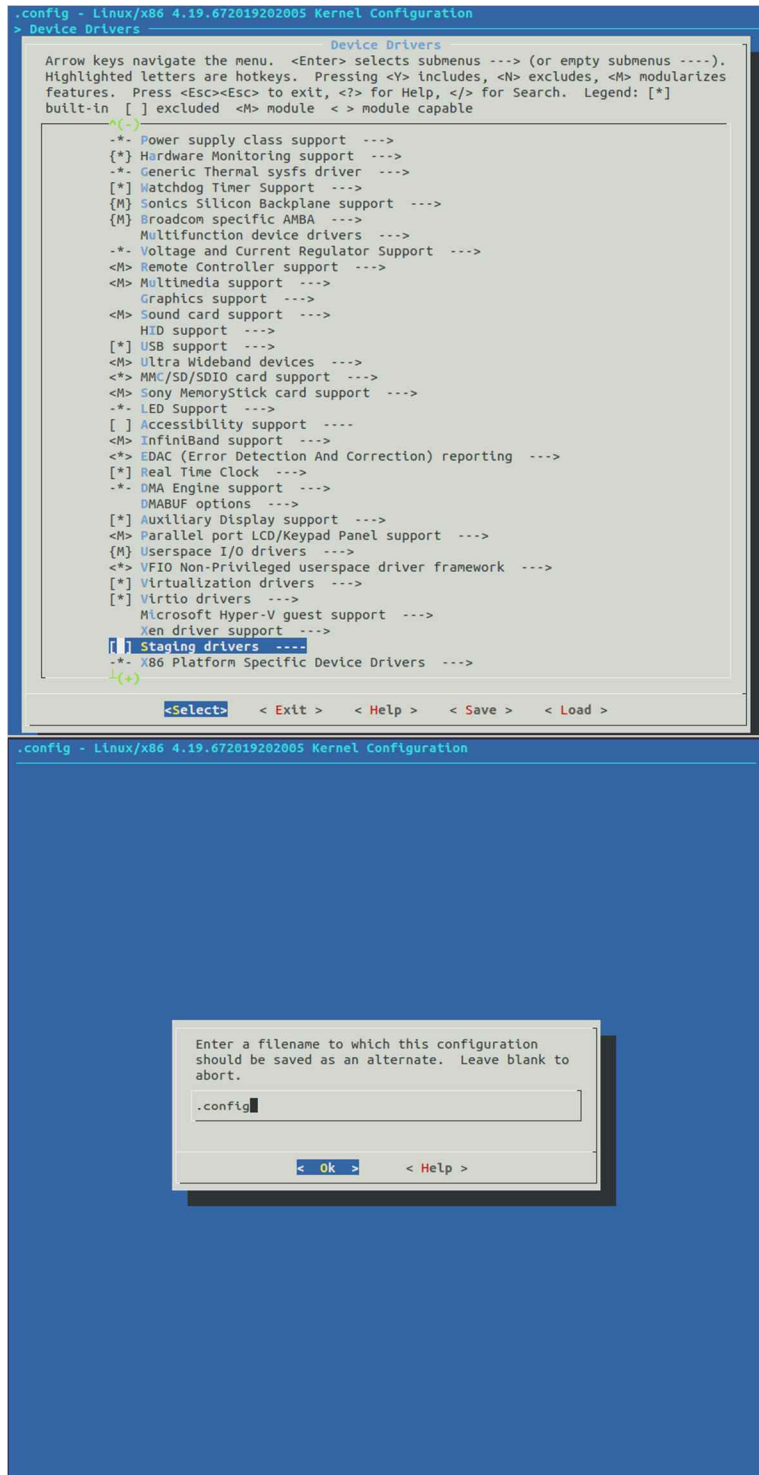
```
os2019202005@ubuntu:~/Downloads/linux-4.19.67$ sudo apt install build-essential libncurses5-dev bison flex libssl-dev libelf-dev
```

```
os2019202005@ubuntu:~/Downloads/linux-4.19.67$ sudo make menuconfig
```

커널 환경설정을 진행했습니다.



커널 모듈 적재 시 발생할 수 있는 문제 해결하기 위해 "Forced module loading" 체크했습니다.



컴파일 시 문제가 될 수 있는 모듈을 제거하고 설정을 파일(.config)에 저장했습니다.

```
os2019202005@ubuntu:~/Downloads/linux-4.19.67$ make -j6
```

컴파일을 나누어 수행할 thread수를 6으로 설정한 후 커널 컴파일을 진행했습니다.

```
os2019202005@ubuntu:~/Downloads/linux-4.19.67$ sudo make modules_install
```

```
os2019202005@ubuntu:~/Downloads/linux-4.19.67$ sudo make install
```

Module install 진행과 컴파일 된 커널을 boot loader에 등록했습니다.

```
os2019202005@ubuntu:~/Downloads/linux-4.19.67$ sudo vi /etc/default/grub
```

다음으로 grub설정 파일 수정을 진행했습니다.

```
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"
```

GRUB\_HIDDEN\_TIMEOUT\_QUIET를 false로 바꿔 줌으로써 grub메뉴를 확인할 수 있습니다.

이 과정들이 끝난 후 reboot를 진행하고 grub메뉴에서 컴파일한 커널을 선택을 합니다.

```
os2019202005@ubuntu:~$ uname -r
4.19.67-2019202005
```

uname -r을 커널 버전이 바뀐 것을 확인할 수 있습니다.

## **고찰**

이번 과제에서 커널 컴파일을 진행하면서 어려운 점은 없었지만 시간이 너무 오래 걸려서 조금 힘들었습니다. 그리고 중간에 오류가 있었는데 해결 방법을 찾는 시간 또한 오래 걸렸습니다. Grub을 설정하는 과정에서 네트워크 관련해서 문제가 있었는데 구글링을 통해 이를 해결할 수 있었습니다. 시간 문제를 제외하고는 수월하게 진행할 수 있었습니다.

## **Reference**

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

2023-2\_OSLab\_Assignment\_1.pdf



## Assignment 1-3

### Introduction

이번 과제는 linux kernel message를 출력하는 과제입니다. printk함수를 통해서 커널에서 메시지를 출력할 수 있고 printk()로 출력된 메시지는 dmesg 명령어를 통해 확인할 수 있습니다. 과제에서는 특정 문자가 포함된 메시지 열이 몇 번째 열인지 보고 싶기 때문에 cscope -R을 통해서 DB구축 및 수행을 한 후 파일 수정 후 dmesg | grep XXX -n명령어를 이용합니다.

### Result

```
os2019202005@ubuntu:~/Downloads/linux-4.19.67$ sudo apt install exuberant-ctags
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
```

sudo apt install exuberant-ctags명령어를 통하여 exuberant-ctags패키지를 설치합니다. ctags 명령어를 사용하여 소스 코드에서 태그를 생성할 수 있으며, 이 태그를 활용하여 코드 내에서 빠르게 이동하거나 정의를 검색할 수 있습니다.

```
os2019202005@ubuntu:~/Downloads/linux-4.19.67$ sudo apt install cscope
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
```

sudo apt install cscope명령어를 통하여 cscope패키지를 설치하고 이를 통해 코드베이스를 색인화하고 코드를 쉽게 검색할 수 있습니다.

```
static int __init agp_init(void)
{
    if (!agp_off)
        printk(KERN_INFO "os2019202005_Linux agpgart interface v%d.%d\n",
                AGPGART_VERSION_MAJOR, AGPGART_VERSION_MINOR);
        printk(KERN_INFO "os2019202005_arg in __init agp_init(void)");
    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
"Downloads/linux-4.19.67/drivers/char/agp/backend.c" 367L, 9160C 340,3-17 96%
```

cscope -R을 통해 데이터 베이스 구축 및 실행한 후 Linux agp...이 실행되는 지점을 찾아 printk함수를 통해 커널 메시지를 출력했습니다.

```
os2019202005@ubuntu:~$ dmesg | grep "os2019202005" -n
2:[ 0.000000] Linux version 4.19.67-2019202005 (os2019202005@ubuntu) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.10)) #3 SMP Mon Sep 18 06:10:08 PDT 2023
1410:[ 7.422735] os2019202005_Linux agpgart interface v0.103
1411:[ 7.422735] os2019202005_arg in __init agp_init(void)
```

dmesg | grep "os2019202005" -n을 통해 os2019202005가 포함된 문자열이 몇 번째 열인지 알 수 있습니다.

## 고찰

이번 과제에서는 ctags와 cscope를 처음 배우고 사용해봤는데 데이터 베이스를 구축하고 실행한다는 점이 신기했습니다. 데이터 베이스 구축 후 원하는 문자열을 찾는 것도 재미있었으며 커널에 대해서 잘 몰랐는데 조금씩 배우니까 흥미로웠습니다. Assignment1는 첫번째 과제였던 만큼 큰 어려움 없이 수월하게 진행할 수 있었습니다.

## Reference

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

2023-2\_OSLab\_Assignment\_1.pdf