

Operating Systems

Assignment #1

담당교수 : 김태석

강의 시간 : 수2

학부 : 컴퓨터정보공학부

학번 : 2017202088

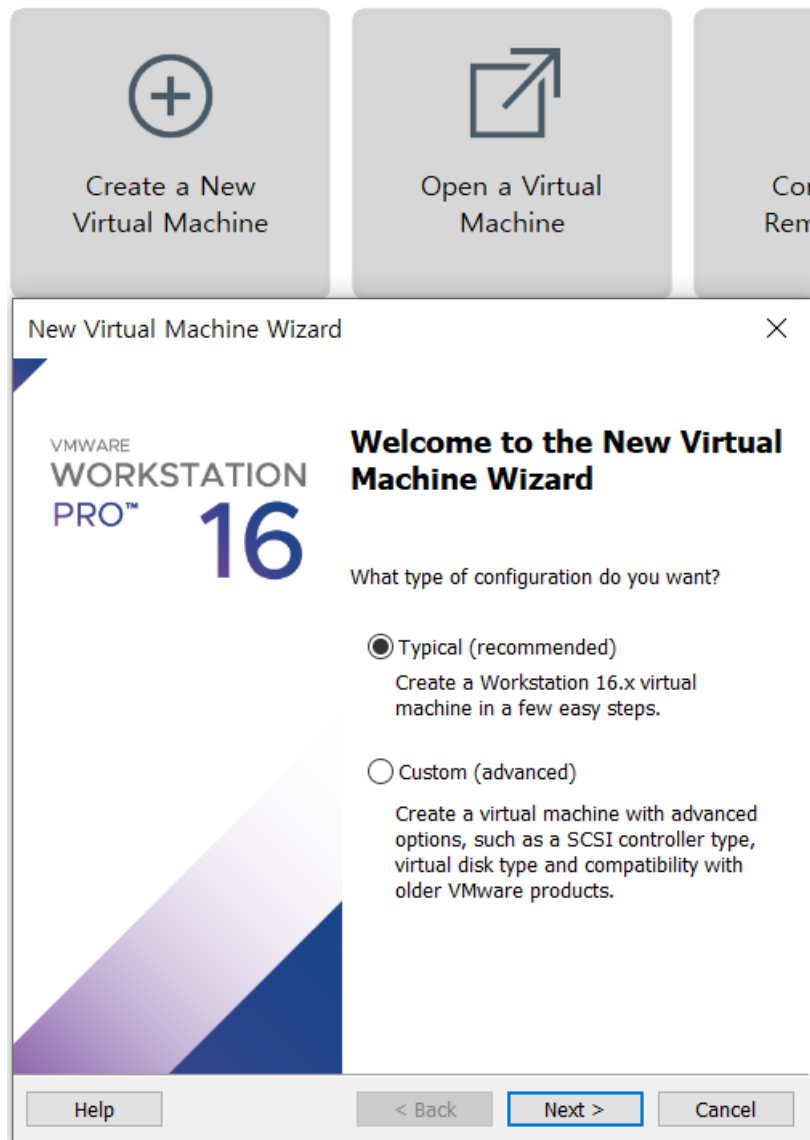
이름 : 신해담

1. Introduction

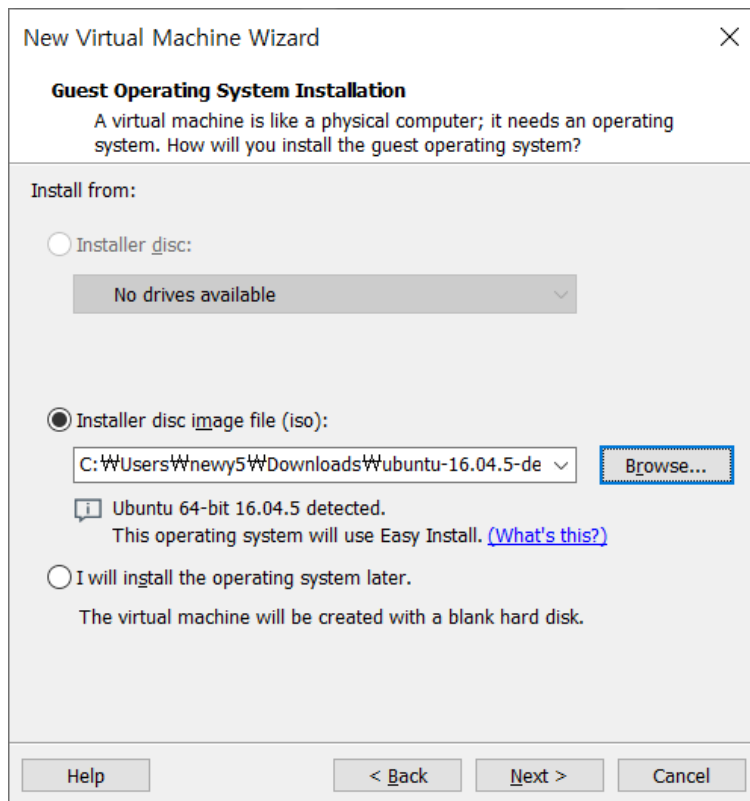
- Ubuntu linux를 설치하고 간단한 세팅을 한다. Ubuntu에 kernel을 다운로드하고 컴파일한다. 이 때, `uname -r`의 결과로 학번이 출력될 수 있도록 설정한다. Linux app...이 실행되는 지점의 커널 메시지가 학번, 함수명, argument를 출력하도록 커널 코드를 수정한다.

2. Assignment 1-1

- VMware에서 Create a New Virtual Machine을 누른다.



- ios파일을 설정한다.



New Virtual Machine Wizard [X]

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:

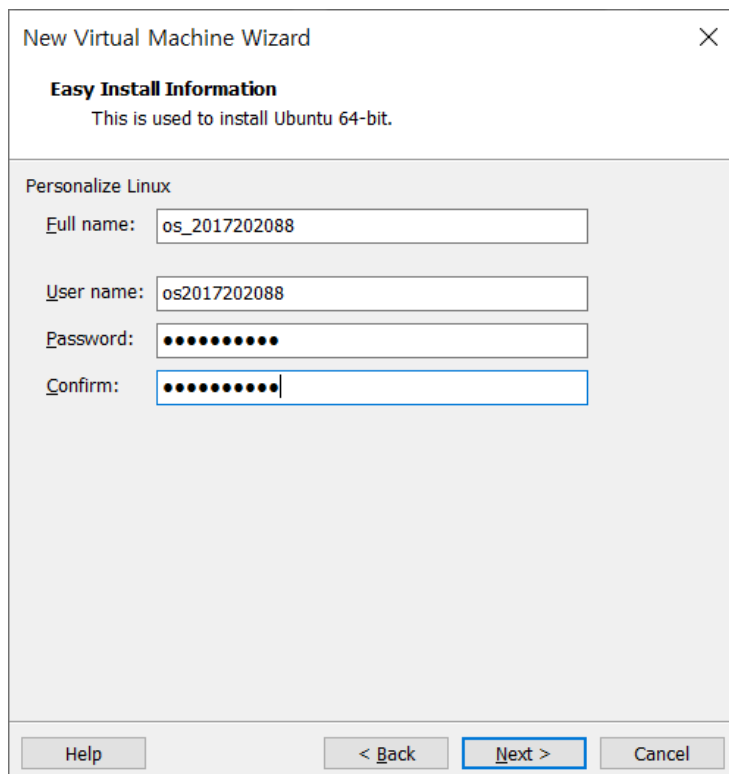
☐ Installer disc:
No drives available

☒ Installer disc image file (iso):
C:\Users\newy5\Downloads\ubuntu-16.04.5-de [Browse...]
 Ubuntu 64-bit 16.04.5 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

- 사용자 계정과 비밀번호를 설정한다.



New Virtual Machine Wizard [X]

Easy Install Information
This is used to install Ubuntu 64-bit.

Personalize Linux

Full name: os_2017202088

User name: os2017202088

Password: ●●●●●●●●

Confirm: ●●●●●●●●

Help < Back Next > Cancel

- 가상머신 이름을 설정한다.

New Virtual Machine Wizard [Close]

Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name:

Location:

The default location can be changed at Edit > Preferences.

< Back **Next >** Cancel

- 가상머신이 사용할 disk 크기를 설정한다.

New Virtual Machine Wizard [Close]

Specify Disk Capacity
How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (GB): [Up] [Down]

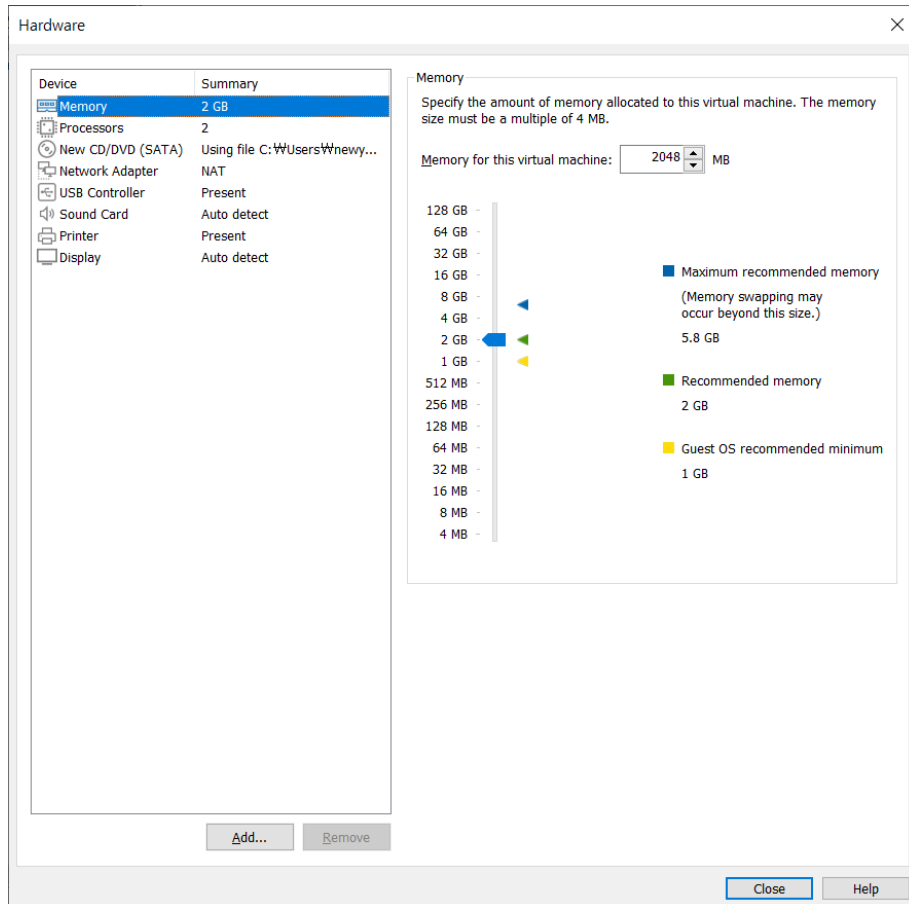
Recommended size for Ubuntu 64-bit: 20 GB

☐ Store virtual disk as a single file
☒ Split virtual disk into multiple files

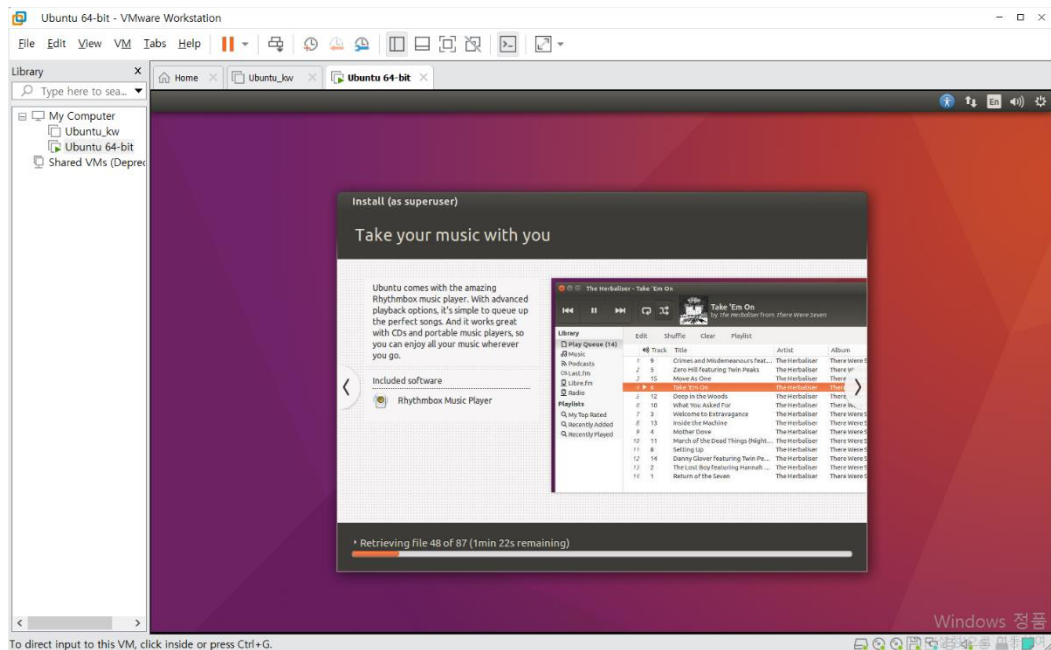
Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

Help < Back **Next >** Cancel

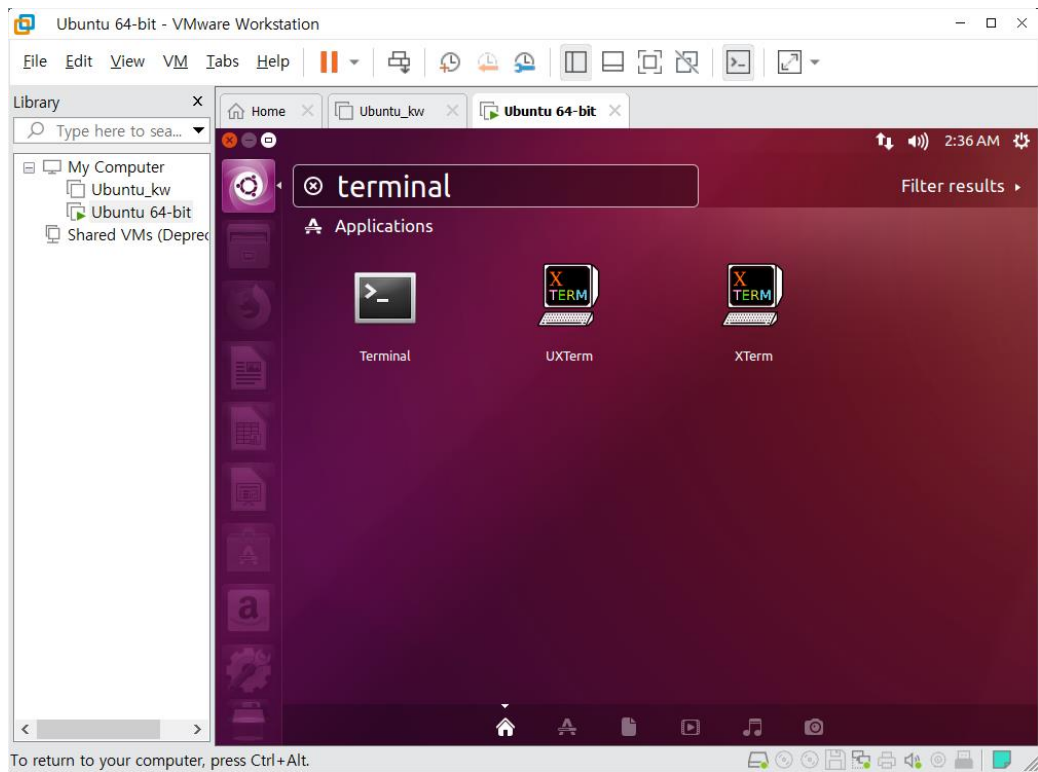
- 가상머신이 사용할 메모리, CPU를 설정한다.



- Ubuntu 리눅스가 설치된다.



- Terminal을 작업 표시줄에 고정한다.



3. Assignment 1-2

- wget 명령어로 Kernel을 다운로드한다.

```
os2017202088@ubuntu:~$ cd /home/os2017202088/Downloads/
os2017202088@ubuntu:~/Downloads$ sudo wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.67.tar.xz
--2021-09-18 23:01:34-- https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.67.tar.xz
Resolving cdn.kernel.org (cdn.kernel.org)... 151.101.1.176, 151.101.65.176, 151.101.129.176, ...
Connecting to cdn.kernel.org (cdn.kernel.org)|151.101.1.176|: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.x 100%[=====] 98.51M 46.0MB/s in 2.1s

2021-09-18 23:01:36 (46.0 MB/s) - 'linux-4.19.67.tar.xz' saved [103291756/103291756]
```

- tar -Jxvf 명령어로 다운받은 kernel의 압축을 해제한다.

```
os2017202088@ubuntu:~/Downloads$ tar -Jxvf linux-4.19.67.tar.xz
```

- 압축 해제한 파일로 이동하고, vim으로 Makefile을 연다.

```
os2017202088@ubuntu:~$ cd Downloads/linux-4.19.67/
os2017202088@ubuntu:~/Downloads/linux-4.19.67$ vim Makefile
```

- Makefile의 EXTRAVERSION에 -2017202088을 넣어서 uname -r 결과에 학번이 나오도록 한다.

```
# SPDX-License-Identifier: GPL-2.0
VERSION = 4
PATCHLEVEL = 19
SUBLEVEL = 67
EXTRAVERSION = -2017202088
NAME = "People's Front"
```

- apt install 명령어로 kernel 환경설정 프로그램을 받고, make menuconfig로 환경설정을 실행한다.

```
os2017202088@ubuntu:~/Downloads/linux-4.19.67$ sudo apt install build-essential libncurses5-dev bison flex libssl-dev libelf-dev
os2017202088@ubuntu:~/Downloads/linux-4.19.67$ sudo make menuconfig
```

- Kernel 환경설정을 해준다.

```
-- 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:
+ ( + )
```

```

.config - Linux/x86 4.19.67"2017202088" Kernel Configuration

Linux/x86 4.19.67"2017202088" 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 5.4.0-6ubuntu1~16.04.10) 5.4.0 2016
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 --->

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

{M} Userspace I/O drivers --->
<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 --->

```

- make -j4 명령어로 kernel을 컴파일한다.

```
os2017202088@ubuntu:~/Downloads/linux-4.19.67$ make -j4
```

- make modules_install 명령어로 module을 install한다.

```
os2017202088@ubuntu:~/Downloads/linux-4.19.67$ sudo make modules_install
```

- make install 명령어로 kernel을 boot loader에 등록한다.

```
os2017202088@ubuntu:~/Downloads/linux-4.19.67$ sudo make install
```

- vim /etc/default/grub로 Grub 설정파일을 열고 다음과 같이 편집한다.

```
os2017202088@ubuntu:~/Downloads/linux-4.19.67$ vim /etc/default/grub
```

```

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 priori

```

- Ubuntu를 reboot한 다음 uname -r 명령어로 커널 버전을 확인하면 학번이 함께 출력된다.

```
os2017202088@ubuntu:~/Downloads/linux-4.19.67$ reboot
```

```

os2017202088@ubuntu:~$ uname -r
4.19.67-2017202088

```


4. Assignment 1-3

- `dmesg | grep "Linux agp" -n` 명령어로 Linux agp... 커널 메시지를 확인한다. cscop를 사용해서 linux agpgart... text string의 위치를 검색해서 코드의 위치를 찾는다.

```
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:

[ 9.822162] IPv6: ADDRCONF(NETDEV_UP): ens33: link is not ready
[ 9.826834] IPv6: ADDRCONF(NETDEV_UP): ens33: link is not ready
[ 9.828953] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Cont
None
[ 9.829735] IPv6: ADDRCONF(NETDEV_CHANGE): ens33: link becomes ready
os2017202088@ubuntu:~/Downloads/linux-4.19.67$ dmesg | grep "Linux agp" -n
1248:[ 6.575790] Linux agpgart interface v0.103
```

- 해당 코드 위치로 들어가보면, `/linux-4.19.67/drivers/char/agp/backend.c`의 `agp_init` 함수에서 linux agp...를 출력하고 있다.

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

- 학번과 함수명, 파라미터를 출력하도록 코드를 수정한다.

```
static int __init agp_init(void)
{
    if (!agp_off) {
        printk(KERN_INFO "os2017202088_Linux agpgart interface v%d.%d\n",
                AGPGART_VERSION_MAJOR, AGPGART_VERSION_MINOR);
        printk(KERN_INFO "os2017202088_arg in agp_init(void)\n");
    }
    return 0;
}

static void __exit agp_exit(void)
{
}

#ifdef MODULE
static __init int agp_setup(char *s)
{
    "drivers/char/agp/backend.c" 368L, 9160C 333,24 95%
```

- make, make_modules_install, make install, reboot를 순서대로 수행한 다음, dmesg | grep "os2017202088" -n 명령어를 입력해서 수정한 코드가 제대로 반영되었는지 확인한다. 학번(os2017202088)과 함수명(agp_init), 파라미터(void)가 잘 출력되었다.

```
os2017202088@ubuntu:~/Downloads/linux-4.19.67$ dmesg | grep "os2017202088" -n
2:[    0.000000] Linux version 4.19.67-2017202088 (os2017202088@ubuntu) (gcc ver
sion 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.10)) #1 SMP Sun Sep 19 05:44:55
PDT 2021
1247:[    8.349462] os2017202088_Linux agpgart interface v0.103
1248:[    8.349462] os2017202088_arg in agp_init(void)
os2017202088@ubuntu:~/Downloads/linux-4.19.67$
```

5. 결론 및 고찰

- Kernel의 코드들이 분량이 크므로, cscop와 ctags를 잘 사용할 필요가 있다.