

운영체제 실습

Assignment #1

Class : A
Professor : 김태석 교수님
Student ID : 2020202090
Name : 최민석

Introduction

이번 Assignment#1 에서는 앞으로의 수업과 실습을 위해 필요한 VMWare Pro 17 과 Ubuntu, Linux-5.4.282 커널을 설치하고 프로그래밍 툴을 연습한다.

VMWare Pro 버전은 일반 버전과 달리 운영체제의 백업을 위한 스냅샷을 지원한다. 운영체제가 정상적으로 동작하는 스냅샷을 저장하면, 커널 패닉 등 운영체제에 이상이 생겼을 때 스냅샷을 통해 해당 시점으로 운영체제를 복구할 수 있다. 이번 과제에선 VMWare Workstation Pro 17.5.2 윈도우 버전을 설치한다.

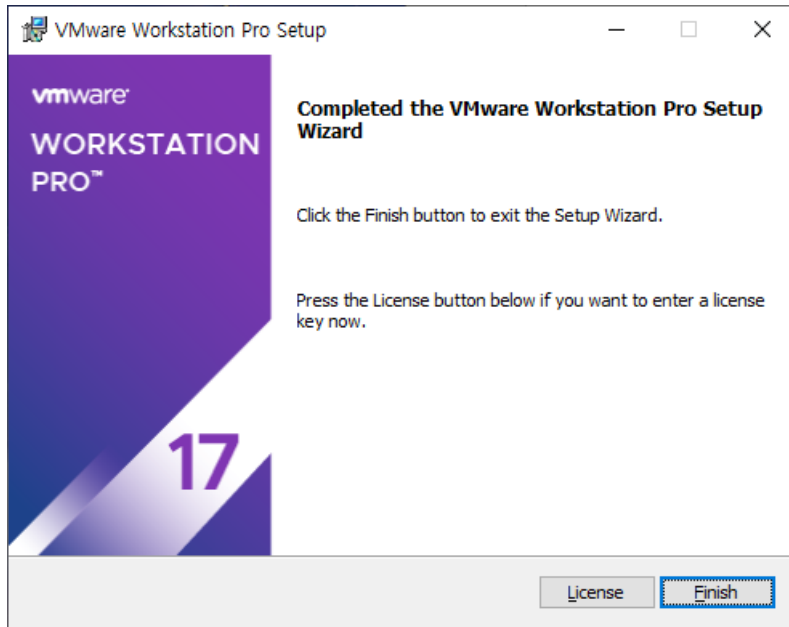
VMWare 에는 Ubuntu 20.04.6 64 비트 버전을 설치한다. VMWare 에서 가상 머신을 생성해 리소스와 스토리지를 할당하고 설치해서 윈도우 환경에서 가상 머신으로 Ubuntu 를 구동한다.

실습에 사용할 커널은 Linux-5.4.282 버전을 사용한다. Kernel.org 커널 아카이브에서 해당 버전을 다운로드 한 후 /usr/src 에 압축을 해제한 뒤 config 파일을 설정하고 커널 컴파일을 진행한 뒤 커널을 설치한다.

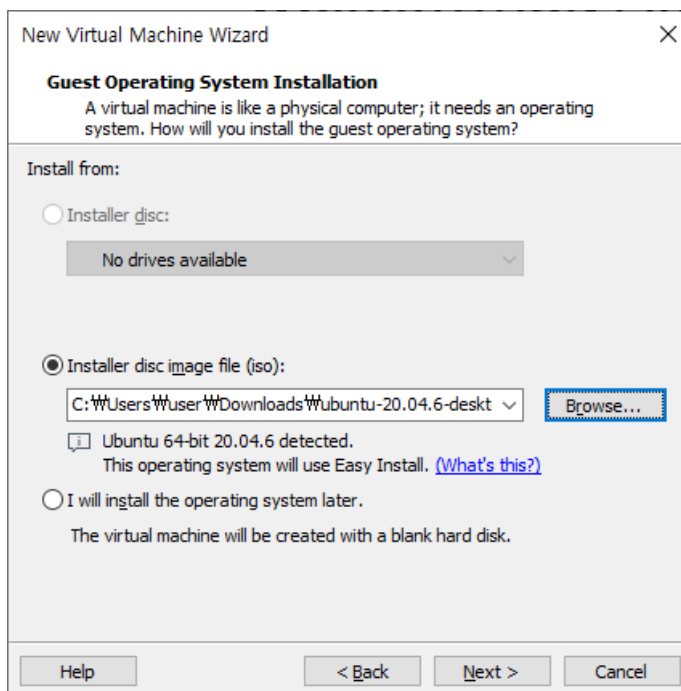
이후 vi 와 vim, ctag, cscope 를 이용해 커널의 코드를 수정해볼 것이다.

결과화면

1. Linux Installation



VMWare Workstation Pro 17 버전을 설치한다.



리눅스를 구동할 새로운 가상 머신을 생성한다.

New Virtual Machine Wizard

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

Personalize Linux

Full name:

User name:

Password:

Confirm:

Help < Back Next > Cancel

가상 머신에 로그인 할 계정과 비밀번호를 설정한다.

Virtual Machine Settings

Hardware Options

Device	Summary
Memory	8 GB
Processors	4
Hard Disk (SCSI)	40 GB
CD/DVD 2 (SATA)	Auto detect
CD/DVD (SATA)	Auto detect
Floppy	Auto detect
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Add... Remove

Memory

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: MB

128 GB -
64 GB -
32 GB -
16 GB -
8 GB -
4 GB -
2 GB -
1 GB -
512 MB -
256 MB -
128 MB -
64 MB -
32 MB -
16 MB -
8 MB -
4 MB -

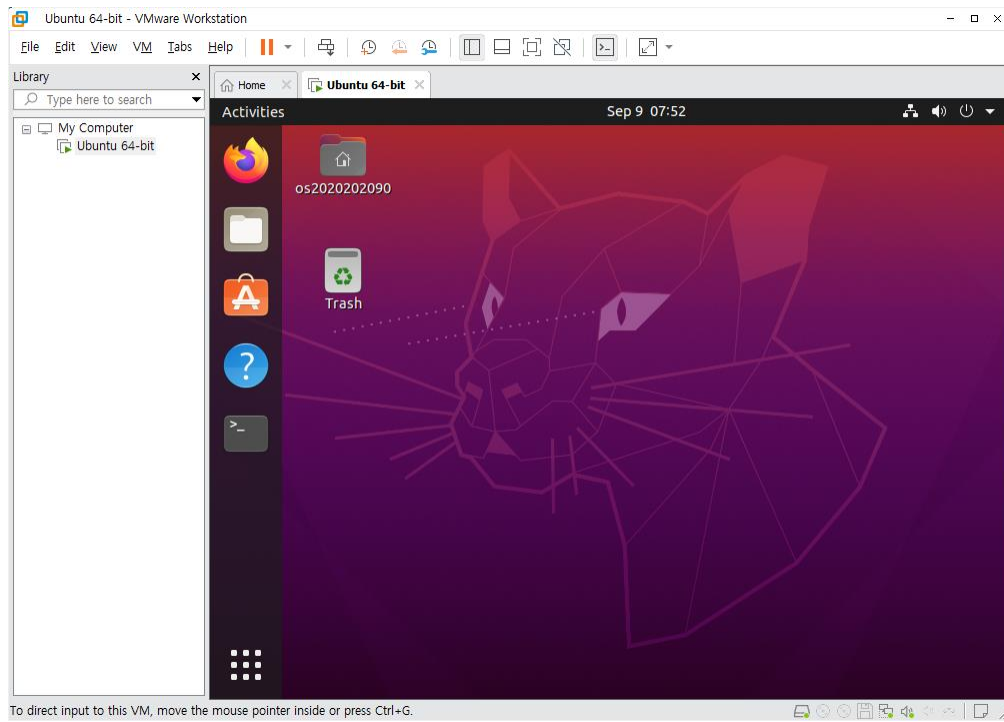
Maximum recommended memory
(Memory swapping may occur beyond this size.)
27.9 GB

Recommended memory
4 GB

Guest OS recommended minimum
2 GB

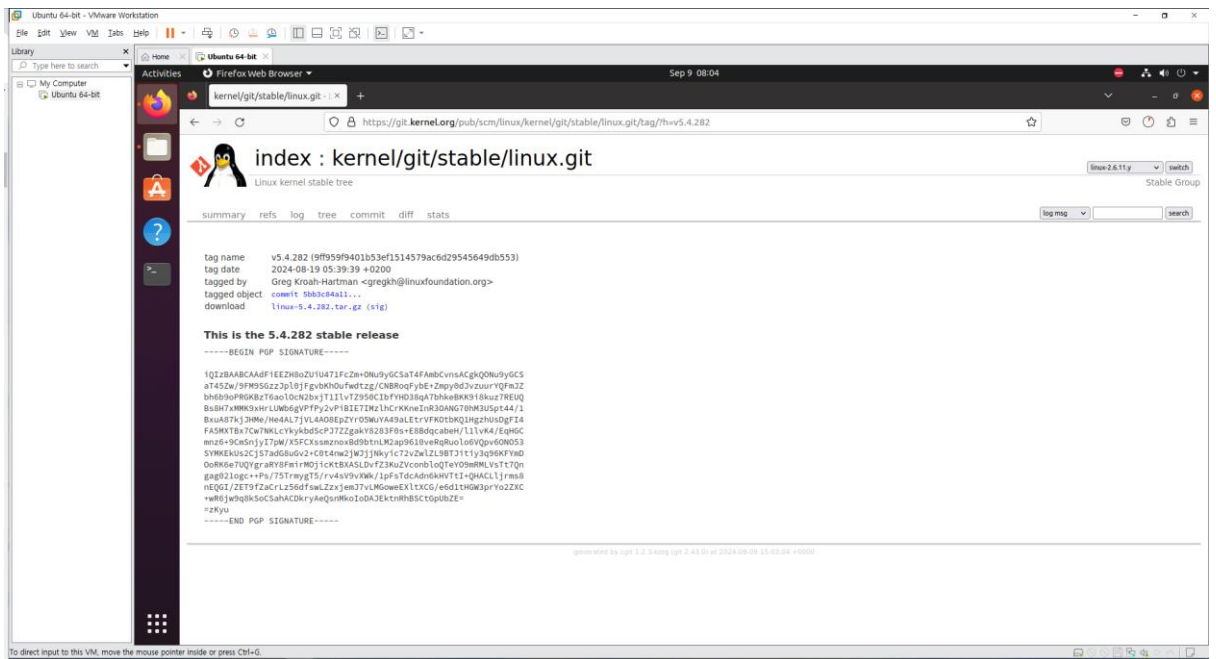
OK Cancel Help

가상 머신에 할당할 스토리지, 메모리, 프로세서를 설정한다.



가상 머신이 생성되면 Ubuntu 가 자동적으로 설치된다.

2. Linux Kernel Compile



kernel.org 에서 커널 5.4.282 를 다운로드 받는다.

```
os2020202090@ubuntu:~/Downloads$ sudo apt-get update
[sudo] password for os2020202090:
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
os2020202090@ubuntu:~/Downloads$ sudo apt-get install build-essential libncurses5
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
```

커널 컴파일 시 필요한 주요 패키지를 설치한다.

```
os2020202090@ubuntu:~/Downloads$ cd /usr/src
os2020202090@ubuntu:/usr/src$ ls
linux-5.4.282                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
linux-headers-5.15.0-67-generic
```

커널 파일을 /usr/src 에 압축 해제하고 확인한다.

```
# SPDX-License-Identifier: GPL-2.0
VERSION = 5
PATCHLEVEL = 4
SUBLEVEL = 282
EXTRAVERSION = -os2020202090
NAME = Kleptomaniac Octopus
```

커널을 컴파일 하기 전 Makefile 에서 엑스트라 버전을 학번으로 설정한다.
그리고 컨픽 환경을 설정한 후 시스템 폐기키 및 보안키를 비활성화하고

```
os2020202090@ubuntu:/usr/src/linux-5.4.282$ sudo make -j4
[sudo] password for os2020202090:
HOSTCC scripts/kconfig/conf.o
HOSTLD scripts/kconfig/conf
scripts/kconfig/conf --syncconfig Kconfig
SYSTBL arch/x86/include/generated/asm/syscalls_32.h
SYSHDR arch/x86/include/generated/asm/unistd_32_ia32.h
WRAP arch/x86/include/generated/uapi/asm/bpf_perf_event.h
```

커널 컴파일 명령어를 실행한다.

```
os2020202090@ubuntu: ~
os2020202090@ubuntu:~$ uname -r
5.4.282-os2020202090
os2020202090@ubuntu:~$
```

grub 설정을 업데이트 하고 재부팅한 다음 linux-5.4.282-os2020202090 으로 부팅한 후 커널 버전을 확인한다.

3. Practice programming tools

```
os2020202090@ubuntu:~$ sudo apt install exuberant-ctags
[sudo] password for os2020202090:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  vim | nvi | vile | emacs
The following NEW packages will be installed:
  exuberant-ctags
```

```
os2020202090@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
```

먼저 ctags 와 cscope 를 설치한다

```
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:
```

cscope 를 실행하고 Linux agpgart 을 검색한다


```

static int __init agp_init(void)
{
    if (!agp_off)
        printk(KERN_INFO "os2020202090_Linux agpgart interface v%d.%d\n",
s2020202090_arg in __initagp_int(void)",
                AGPGART_VERSION_MAJOR, AGPGART_VERSION_MINOR);
    return 0;
}

static void __exit agp_exit(void)
{
}

#ifdef MODULE
static __init int agp_setup(char *s)
"drivers/char/agp/backend.c" 366 lines, 9138 characters written

```

Linux agpgart 가 시작되는 drivers/char/agp/backend.c 의 printk()를 다음과 같이 수정하였다.

고찰

- Linux Installation

VMWare 17 Pro 를 설치한 후 Ubuntu 를 위한 가상 머신을 생성하고 리소스를 할당한 뒤 운영체제를 설치하였다. 이전 학기의 시스템프로그래밍에서 이미 해봤던 절차라서 간단했다.

VMWare 17 Pro 버전은 일반 버전과 다르게 스냅샷 기능을 지원하여 운영체제를 이전 시점으로 복구할 수 있다. 이번 운영체제 강의에서는 커널 코드를 수정하는 부분이 있어 운영체제가 손상될 수도 있는데, 스냅샷 기능을 활용하여 운영체제를 복구하는데 필요한 시간과 노력을 줄일 수 있을 것이라고 생각한다.

- Kernel 5.4.282 Compile

설치한 Ubuntu 에서 Linux-5.4.282 버전의 커널을 다운로드 받은 뒤 압축을 해제하고 /usr/src 디렉토리에 이동시킨다. 이후 필요한 패키지를 다운로드 받고 커널 컴파일을 수행하였다. 커널 코드는 매우 복잡하고 양이 많기 때문에 스레드 4 개를 할당해도 1 시간 이상 소요되었다.

스냅샷 기능을 활용하여 이 시점을 저장해두고 앞으로의 과제를 수행하는데 사용할 것이다.

- Practice programming tools

설치한 커널에서 Linux agp...가 시작되는 부분의 printk()함수를 수정하고 적용한 뒤 재부팅해서 정상적으로 메시지가 출력되는지 확인해보았다.

cscope 를 사용하여 해당 경로의 파일을 찾아서 vi 에디터를 통해 수정을 진행하였다.

Reference

- 24-2_OSLab_Assignment-1_v3.pdf
- 2024-2_OSLab_01_Kernel_compile.pdf
- 2024-2_OSLab_02_Programming_tools.pdf