

# 과제 #0

## - 리눅스 VM 생성 및 커널 컴파일

---

제출 기한: 4/5 (일) 23:59

제출 방법: IEILMS “과제 #0”

# 과제 학습 목표

---

- 차후 과제의 진행을 위한 환경 설정 및 커널 빌드 방법 학습

~~1. 리눅스 운영체제를 설치해본다.~~

2. 나만의 리눅스 커널을 빌드하여, 운영체제에 설치한다.

- Build: compile and install



# 과제 학습 목표

---

- 차후 과제의 진행을 위한 환경 설정 및 커널 빌드 방법 학습
1. 클라우드에서 리눅스 가상머신을 생성한다.
    - J-Cloud 이용: <http://jcloud.jbnu.ac.kr/>
  2. 나만의 리눅스 커널을 빌드하여, 운영체제에 설치한다.
    - Build: compile and install

# 과제 제출 내용 (IEILMS 과제 게시판)

---

- 보고서 (**꼭 PDF로만!! 제출해주세요**)
  - 제목, 학번, 이름, 캡처 화면 3개, 어려웠던 점 및 해결 방안
  - 캡처 화면 외의 내용은 A4 한 장을 절대 넘기지 말 것
- 커널 컴파일 완료 후 시간 캡처 화면 2개
  - 1개: 전체 컴파일 후, 1개: 모듈 및 커널 설치 후
    - 슬라이드 후반부에 예시 있음
  - time 명령을 반드시 앞에 붙여 사용할 것!
- 커널 설치 후 hostnamectl 결과 화면 캡처 1개
  - 슬라이드 후반부에 예시 있음
- (**중요**) IEILMS 제출 시, 글은 확인하지 않음
  - 그냥 점 dot 만 찍을 것. 하고 싶은 말은 보고서에 작성

# 순서

---

1. J-Cloud 에서 우분투 가상머신 (인스턴스) 생성 및 접속
2. 개발 편의를 위한 설정
3. 리눅스 커널 다운로드 및 컴파일



---

# 1. J-Cloud 에서 우분투 가상머신 (인스턴스) 생성 및 접속



# J-Cloud

---



JCloud는 전북대학교 컴퓨터공학부에서 제공하는  
교육 및 연구용 클라우드 서비스입니다.

전북대 학생들로 구성된 J-DevOps 팀이 OpenStack을 기반으로 구축하였고,  
교육 및 연구 용도에 맞춰 필요한 사항을 수정하며 운영합니다.

2018년 2월 현재, 시험 운영 중이며 아래 두 가지 서비스를 제공 중입니다.

*(로고를 클릭하면 해당 서비스로 이동합니다.)*



# J-Cloud services

---



## JCloud 컴퓨팅 서비스

클라우드 상의 가상 머신, 가상 네트워크, 가상 스토리지 등의 서비스를 제공합니다.

학생들은 수업 및 개인 학습 용도로 활용할 수 있습니다.



## Web IDE 서비스

웹 상에서 프로그램 개발 및 테스트를 수행할 수 있는 Web IDE 서비스를 제공합니다.

C,C++,Java,Python,Net,PHP,Node.js 등 다양한 언어를 지원합니다.  
본 서비스는 JCloud 를 기반으로 동작하고 있습니다.





# J-Cloud 사용 시 장점

---

- In short, 학교 서버에 있고 어디서든 접속 가능한 가상머신
- 빠르다: 커널 컴파일 약 20분 내외
- 항상 켜져 있다
- 내 컴퓨터/ 내 노트북이 아니다
- 어디서나 접속 가능하다 (집/학교 어디서든 SSH 이용)
- 단점
  - 내 컴퓨터에 설치된 VM이 아니기 때문에 마음대로 다루기가 어렵다
    - 사실 조금만 익숙해지면 해결됨. VM 이기 때문에 불편한 점은 똑같음

# 사용 방법

---

- J-Cloud 시스템: <http://jcloud.jbnu.ac.kr>
- 소개 및 사용 방법: <https://jcloud-devops.github.io/>
  - Document 에서 아래 참조
    - 인스턴스 생성 방법
    - SSH 접속은 윈도우 터미널, Xshell, putty 등 사용 (#13 슬라이드 내용 꼭 보고 진행)
- OS 수강생들의 계정은 모두 생성해놓은 상태
  - ID: os학번, PW: 학번
  - 예) os41983//41983
- 문의: [jcloud@jbnu.ac.kr](mailto:jcloud@jbnu.ac.kr)
- 참고: 학기 종료 직후, 인스턴스 및 계정 삭제됨
  - 컴퓨터공학과 학생들은 학번으로 계정이 생성되어 있음
  - 타 학과 학생들은 별도 신청하여 사용 가능

# OS 수업용 인스턴스(가상머신) 만들기

---

- Details
  - Instance Name: ID 와 동일하게 os학번
    - Count: 1 (1인당 1대만 만들어서 사용할 것)
  - Availability Zone: “Any Availability Zone”
- Source: Image, “Ubuntu 18.04.2 ssh7777” 선택
  - Create New Volume: “No”
- Flavor: os2020.flavor
- Network: os2020.internal
- Key Pair: 미리 생성해서 꼭 다운로드 받고, 파일 잘 관리할 것
  - J-Cloud 상의 가상머신에 접속할 때 사용하는 인증파일
- 그 외 기본 설정 사용

# J-Cloud 에서 인스턴스 생성

- 인스턴스 생성 후 화면
  - 본인의 인스턴스에 대해 IP 확인하고, SSH 접속하여 다음 내용들을 진행할 것

The screenshot shows the JCloud web interface for the 'os2019' project. The left sidebar contains navigation links: Project, API Access, Compute, Overview, Instances (selected), Images, Key Pairs, Server Groups, Volumes, Network, and Identity. The main content area is titled 'Instances' and shows a table of two instances.

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/> hcpark.test 2	Ubuntu 18.04.2 ssh7777	192.168.0.3	os2019.flavor	hcpark-jcloud-rocky	Active	nova	None	Running	35 minutes	Create Snapshot
<input type="checkbox"/> hcpark.test	Ubuntu 18.04.2 ssh7777	192.168.0.9	Not available	hcpark-test	Active	nova	None	Running	1 day	Create Snapshot

# SSH 클라이언트를 이용한 접근

매우 중요!!!

- <https://jcloud-devops.github.io/user-guide.html>
  - SSH 연결 방법을 확인하고, 본 수업에서는 아래 공용 IP 사용함
  - 공용 IP: 203.254.143.134
  - SSH 포트 번호: 19xxx
    - xxx는 본인 인스턴스 IP 의 끝자리. 예) 192.168.0.3 → 19003, 192.168.0.199 → 19199
  - User ID: ubuntu (인스턴스를 만들때 사용한 이미지마다 다름. 본 수업에 사용한 이미지에만 해당)
  - Password는 없음. 키페어가 패스워드 역할을 함
- SSH 클라이언트 (자유롭게 본인에 맞는 프로그램 사용할 것)
  - SSH 프로토콜을 이용해 원격 서버에 접속하는 프로그램
  - PuTTY 등이 유명함: 단순하고 기본 기능 충실. 그러나 너무 단순함. 키페어 사용 방법이 복잡함
  - 윈도우 10에서 제공: 구글 “윈도우 10 Open SSH” 검색
- Xshell (추천)
  - 다양한 기능. 개인용, 교육용 무료 버전 배포
  - [http://www.netsarang.co.kr/download/free\\_license.html](http://www.netsarang.co.kr/download/free_license.html)



# J-Cloud 관련 팁

---

- 크롬 브라우저를 사용할 것
  - 특히 인터넷 익스플로러는...좀...이제 그만 좀...
    - Known issue: key pair 파일 자동 다운로드가 안됨
- 도움말 및 비밀번호 변경 방법
  - 오른쪽 위 본인 계정 버튼을 클릭하면,
    - 도움말: Help 클릭 -> JCloud Documentation page 로 이동
    - 비밀번호 변경: Settings 클릭 -> (화면 전환 후 왼쪽 메뉴에서 ) change password
- JCloud 에서 인스턴스 재부팅 방법
  - 본인 인스턴스에 대해, 오른쪽 끝 메뉴에서 soft reboot, hard reboot, shutdown 등을 진행할 수 있음
  - 그리고 전원이 꺼진 상태에서는 start instance 를 이용해 시작 가능
  - 해당 메뉴에서 인스턴스를 삭제할 수도 있음 (복구 불가능)

---

## 2. 개발 편의를 위한 설정



# 부트메뉴 보기

- `$ sudo -s` (시스템 관리를 위해 root 유저로 변경)
- `# vi /etc/default/grub.d/50-cloudimg-settings.cfg`
  - 부팅 관련 설정 변경. 첫 번째 파일
  - `GRUB_TIMEOUT=3`
    - 3초간 메뉴를 표시하도록 변경. 원래 값은 0
- `# vi /etc/default/grub`
  - 부팅 관련 설정 변경. 두 번째 파일
  - 메뉴가 보이도록 하고, 마지막 부팅 때 사용한 커널이 default 가 되도록 변경
    - `GRUB_DEFAULT=saved`
    - `GRUB_SAVEDEFAULT=true` (<- 새로 추가해야 함)
    - `GRUB_TIMEOUT_STYLE=menu`
    - `GRUB_TIMEOUT=3`
- `# update-grub`
  - 변경 내용을 적용하여 부트로더 재설정



# (수정할 2개 파일 내용)

```
# Cloud Image specific Grub settings for Generic Cloud Images
# CLOUD_IMG: This file was created/modified by the Cloud Image build process

# Set the recordfail timeout
GRUB_RECORDFAIL_TIMEOUT=0

# Do not wait on grub prompt
GRUB_TIMEOUT=3

# Set the default commandline
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0"

# Set the grub console type
GRUB_TERMINAL=console
```

```
# 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=saved
GRUB_SAVEDefault=true
#GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=menu
#GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=3
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0"
GRUB_CMDLINE_LINUX=""
```



# 부트메뉴 보기

- J-Cloud 홈페이지에서 Console 화면을 켜놓고, SSH 클라이언트에서 reboot 수행
  - Instances 메뉴에서 본인 Instance 이름 선택
  - Console 메뉴 선택
  - “Click here to show only console” 선택하여 전체 화면 수행
    - 콘솔 화면 내를 클릭하여 조작 가능
    - 본인이 원하는 커널로 부팅
    - 3초 만에 선택하기 어려우면 숫자를 늘리면 됨
- J-Cloud console에서 바로 로그인하려면? (이것도 꼭 해둘 것)
  - SSH 접속한 후, `$ sudo adduser <사용할id>` 입력
  - 패스워드만 설정하고 나머지는 enter (분실하지 않도록 간단한 암호: 1234)
  - J-Cloud console 화면에서 “Click here to show only console” 선택하고 로그인
  - (키페어 분실 등으로 로그인 못하는 상황에서 긴급 복구 용으로 활용)

# (J-Cloud Console 화면)

9cb0314-55e5-4eee-8325-4953fe063735&title=hcpark.test2(917d6fba-01e0-4e26-a3b8-29073fe991ed)

Connected (unencrypted) to: QEMU (instance-0000004b)

GNU GRUB version 2.02

Ubuntu

\*Advanced options for Ubuntu

Use the ↑ and ↓ keys to select which entry is highlighted.  
Press enter to boot the selected OS, 'e' to edit the commands  
before booting or 'c' for a command-line.  
The highlighted entry will be executed automatically in 1s.



# (adduser)

---

```
ubuntu@hcpark:~$ sudo adduser hcpark
Adding user `hcpark' ...
Adding new group `hcpark' (1001) ...
Adding new user `hcpark' (1001) with group `hcpark' ...
Creating home directory `/home/hcpark' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hcpark
Enter the new value, or press ENTER for the default
  Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
ubuntu@hcpark:~$
```



# 파일을 주고 받으려면?

---

- Xshell 의 경우, zmodem 기능을 이용해 파일 전송 가능
  - # apt-get install lrzsz
  - 파일 업로드: Drag & drop
  - 파일 다운로드
    - \$ sz sample.txt
    - 이후 다운로드 받을 폴더 선택
- 인터넷에 있는 파일을 다운로드 받고 싶을 때
  - wget 명령을 이용. 인터넷에서 얻은 파일 링크를 입력
  - 예제
    - \$ wget https://jcloud-devops.github.io/files/JCloud%20v3.pdf



---

# 3. 리눅스 커널

## 다운로드 및 컴파일



# 새로운 커널의 설치

---

- Why?

- 새로운 커널에 추가된 새로운 기능을 이용하고 싶을 때
- 기존 커널을 원하는 대로 수정하고 싶을 때

- How?

- Package 이용: 배포본에서 제공되는 패키지를 이용하여 설치
- Manual: 리눅스 소스 코드를 컴파일하여 설치
  1. 배포본에서 제공되는 소스 코드를 컴파일하여 설치
    - 배포본에 맞게 소스 코드에 수정을 가한 경우가 많음
  2. 공식 배포된 리눅스 커널을 기반으로 설치
    - Vanilla kernel에 배포본에 맞는 패치를 가한 후, 컴파일 및 설치하여야 함
    - Vanilla kernel: 수정되지 않은 공식 커널 소스 코드

# 참고. Package 로 설치하는 방법 (하면 안됨)

- 현재 user: ubuntu (기본 계정)
- \$ sudo -s
  - 시스템 관리를 위해 root 유저로 변경
- # apt-get update
  - 패키지 정보 업데이트
- # apt-cache search linux-image | grep generic
  - 패키지 리스트에서 linux-image 검색하고,  
그 중에서 generic 문구가 포함된 항목만 출력
- # apt-get install linux-image-4.18.0-16-generic
  - 리스트 중 가장 최신의 signed 버전으로 설치
- # reboot
  - 재시작하여 새로운 커널로 부팅 (확인: hostnamectl)

```
linux-image-4.15.0-45-generic - Signed kernel image generic
linux-image-4.15.0-46-generic - Signed kernel image generic
linux-image-4.18.0-13-generic - Signed kernel image generic
linux-image-4.18.0-14-generic - Signed kernel image generic
linux-image-4.18.0-15-generic - Signed kernel image generic
linux-image-4.18.0-16-generic - Signed kernel image generic
linux-image-generic-hwe-18.04 - Generic Linux kernel image
linux-image-generic-hwe-18.04-edge - Generic Linux kernel image
```

이 슬라이드는 참고 자료임.  
수행하지 마세요



# 리눅스 커널 공식 페이지: kernel.org

## The Linux Kernel Archives

[About](#)[Contact us](#)[FAQ](#)[Releases](#)[Signatures](#)[Site news](#)**Protocol**[HTTP](#)[GIT](#)[RSYNC](#)**Location**<https://www.kernel.org/pub/><https://git.kernel.org/><rsync://rsync.kernel.org/pub/>**Latest Stable Kernel:****5.0.2**

mainline:	<b>5.1-rc1</b>	2019-03-17	<a href="#">[tarball]</a>	<a href="#">[patch]</a>	<a href="#">[view diff]</a>	<a href="#">[browse]</a>	
stable:	<b>5.0.2</b>	2019-03-13	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
stable:	<b>4.20.16</b>	2019-03-13	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>4.19.29</b>	2019-03-13	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>4.14.106</b>	2019-03-13	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>4.9.163</b>	2019-03-13	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>4.4.176</b>	2019-02-23	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>3.18.136 [EOL]</b>	2019-02-23	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	<b>3.16.63</b>	2019-02-11	<a href="#">[tarball]</a>	<a href="#">[pgp]</a>	<a href="#">[patch]</a>	<a href="#">[inc. patch]</a>	<a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
linux-next:	<b>next-20190319</b>	2019-03-19					<a href="#">[browse]</a>



# Manual compiling and installation

여기서부터  
순서대로 진행

- 현재 상태 확인
  - 리눅스 커널 버전 확인
  - 작업 경로 확인
- 소스 코드 다운로드
  - Source 패키지 경로 enable
  - 패키지 정보 업데이트
  - 소스 코드 패키지 다운로드 및 확인
  - 소스 압축 해제
- 커널 컴파일
  - 환경 설정
  - 컴파일 및 설치
  - 재부팅 후 확인하기



# 현재 상태 확인

- `$ sudo -s`
  - 시스템 관리를 위해 root 유저로 변경
  - 앞으로도 커널 소스 관련은 모두 root 권한으로 진행함
- `# dpkg-reconfigure tzdata`
  - 시간대 변경: Asia, Seoul
  - `# date` 로 시간 확인
- `# pwd`
  - 현재 작업 경로 확인
- `# ls -al`
  - 현재 경로의 파일 리스팅
- `# hostnamectl`
  - 현재 커널 버전 확인

```
Last login: Tue Mar 12 07:18:04 2019 from 114.70.
ubuntu@hccpark:~$ sudo -s
root@hccpark:~# dpkg-reconfigure tzdata

Current default time zone: 'Asia/Seoul'
Local time is now:      Wed Mar 20 12:31:17 KST 2019.
Universal Time is now:  Wed Mar 20 03:31:17 UTC 2019.

root@hccpark:~# date
Wed Mar 20 12:31:21 KST 2019
root@hccpark:~# pwd
/home/ubuntu
root@hccpark:~# ls -al
total 44
drwxr-xr-x 5 ubuntu ubuntu 4096 Mar 20 12:30 .
drwxr-xr-x 3 root  root  4096 Mar 12 16:17 ..
-rw-r--r-- 1 ubuntu ubuntu  104 Mar 20 12:30 .Xauthority
-rw-r--r-- 1 root  root    58 Mar 12 16:17 .bash_history
-rw-r--r-- 1 ubuntu ubuntu 220 Apr  5 2018 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Apr  5 2018 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Mar 12 16:17 .cache
drwx----- 3 ubuntu ubuntu 4096 Mar 12 16:17 .gnupg
-rw-r--r-- 1 ubuntu ubuntu  807 Apr  5 2018 .profile
drwx----- 2 ubuntu ubuntu 4096 Mar 12 16:17 .ssh
-rw-r--r-- 1 ubuntu ubuntu   0 Mar 12 16:17 .sudo_as_admin_successful
-rw-r--r-- 1 root  root   902 Mar 12 16:17 .viminfo
root@hccpark:~# hostnamectl
  Static hostname: hccpark
            Icon name: computer-vm
            Chassis: vm
            Machine ID: 32cd22818fee491ab370e3a29dbdb4f7
            Boot ID: 5f2ab619ec2443dbb731acb7fc9e3f66
    Virtualization: kvm
  Operating System: Ubuntu 18.04.2 LTS
            Kernel: Linux 4.15.0-45-generic
    Architecture: x86-64
root@hccpark:~#
```

# 소스 코드 다운로드

- # vi /etc/apt/sources.list
  - Source 패키지 경로 enable (page down 으로 맨 아랫줄 확인 후, 맨 앞 # 제거)

```
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://nova.clouds.archive.ubuntu.com/ubuntu/ bionic-backports main restricted universe multiverse
deb-src http://nova.clouds.archive.ubuntu.com/ubuntu/ bionic-backports main restricted universe multiverse

deb http://security.ubuntu.com/ubuntu bionic-security main restricted
deb-src http://security.ubuntu.com/ubuntu bionic-security main restricted
deb http://security.ubuntu.com/ubuntu bionic-security universe
deb-src http://security.ubuntu.com/ubuntu bionic-security universe
deb http://security.ubuntu.com/ubuntu bionic-security multiverse
deb-src http://security.ubuntu.com/ubuntu bionic-security multiverse

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
deb http://archive.canonical.com/ubuntu bionic partner
deb-src http://archive.canonical.com/ubuntu bionic partner
~
~
~
~
~
```



# 소스 코드 다운로드

- # apt-get update
  - 패키지 정보 업데이트

```
root@hcpark:~# vi /etc/apt/sources.list
root@hcpark:~# apt-get update
Get:1 http://archive.canonical.com/ubuntu bionic InRelease [10.2 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:3 http://archive.canonical.com/ubuntu bionic/partner Sources [1904 B]
Get:4 http://archive.canonical.com/ubuntu bionic/partner amd64 Packages [2308 B]
Get:5 http://archive.canonical.com/ubuntu bionic/partner Translation-en [1272 B]
Hit:6 http://nova.clouds.archive.ubuntu.com/ubuntu bionic InRelease
Get:7 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/universe Sources [35.6 kB]
Get:9 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/main Sources [76.8 kB]
Get:11 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates/universe Sources [137 kB]
Get:12 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [282 kB]
Get:13 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates/main Sources [254 kB]
Get:14 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [556 kB]
Get:15 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [127 kB]
Get:16 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [207 kB]
Get:17 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [744 kB]
Get:18 http://nova.clouds.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [193 kB]
Fetched 2880 kB in 5s (553 kB/s)
Reading package lists... Done
root@hcpark:~#
```



# 소스 코드 다운로드

- # apt install linux-source-4.18.0

```
root@hcpark:~# apt install linux-source-4.18.0
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'apt autoremove' to remove it.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-7 gcc gcc-7 gcc-7-base libasan4 libatomic1 libbinutils libc-dev-bin libc
  linux-libc-dev make manpages-dev
Suggested packages:
  binutils-doc cpp-doc gcc-7-locales gcc-multilib autoconf automake libtool flex bison gdb gcc-doc gcc-7-multilib gcc-7-doc libgcc1-d
  libquadmath0-dbg glibc-doc libncurses-dev | ncurses-dev kernel-package libqt3-dev make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-7 gcc gcc-7 gcc-7-base libasan4 libatomic1 libbinutils libc-dev-bin libc
  linux-libc-dev linux-source-4.18.0 make manpages-dev
0 upgraded, 29 newly installed, 0 to remove and 32 not upgraded.
Need to get 157 MB of archives.
After this operation, 262 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```



# 소스 코드 다운로드

- 리눅스 소스 패키지가 설치된 경로로 이동

- # cd /usr/src/linux-source-4.18.0/
- # ls -alh
  - 파일 및 용량 확인 (약 120MB)

```
root@hcpark:~# cd /usr/src/linux-source-4.18.0/
root@hcpark:/usr/src/linux-source-4.18.0# pwd
/usr/src/linux-source-4.18.0
root@hcpark:/usr/src/linux-source-4.18.0# ls -alh
total 126408
drwxr-xr-x  4 root root    4096 Mar 19 02:48 .
drwxr-xr-x  7 root root    4096 Mar 19 02:48 ..
drwxr-xr-x 13 root root    4096 Mar 19 02:48 debian
drwxr-xr-x  9 root root    4096 Mar 19 02:48 debian.hwe
-rw-r--r--  1 root root 129421773 Feb 12 11:06 linux-source-4.18.0.tar.bz2
root@hcpark:/usr/src/linux-source-4.18.0#
```

- 압축 해제

- # bzip2 -d linux-source-4.18.0.tar.bz2
- # tar xf linux-source-4.18.0.tar

- 리눅스 작업 경로로 이동 및 확인

- # cd linux-source-4.18.0/
- # pwd
  - /usr/src/linux-source-4.18.0/linux-source-4.18.0
- 이후 커널 컴파일 등의 작업은 이 위치에서 수행함
  - a.k.a. 커널 소스 디렉토리 or kernel source path





# 관련 도구 설치

- # time apt-get install kernel-package bison flex libncurses5-dev libssl-dev libelf-dev

- 약 5분 이상 걸림
  - J-Cloud에서 여러 사람이 동시에 작업하면 더 오래 걸림
  - 과제 마감 직전에 급하게 하지 말고 미리 미리 해두자
- 중간에 다음 슬라이드처럼 물어보면
  - Install the package maintainer's version

```
root@hpcpark:/usr/src/linux-source-4.18.0/linux-source-4.18.0# time apt-get install kernel-package
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'apt autoremove' to remove it.
The following additional packages will be installed:
  build-essential dblatex dblatex-doc docbook-dsssl docbook-utils docbook-xml docbook-xsl dpkg-dev fakeroot fontconfig-config fonts-d
  gettext ghostscript gsfonts intltool-debian javascript-common kernel-common libalgorithm-diff-perl libalgorithm-diff-xs-perl libalg
  libcupsfilters1 libcupsimage2 libdata-dump-perl libdatatriel libdpkg-perl libdrm-amdgpu libdrm-intel libdrm-nouveau libdrm-radeon1
  libfile-listing-perl libfile-mimeinfo-perl libfile-which-perl libfont-afm-perl libfontconfig1 libfontenc1 libgl1 libgl1-mesa-dri li
  libhtml-form-perl libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libhttp-cookies-perl libhttp-daemon
  libipc-system-simple-perl libjpeg8 libjpeg2dec0 libjpeg-turbo8 libjpeg8 libjs-jquery libkpathsea6 liblcms2-2 libllvm7 liblwp-mediat
  libnet-smtp-ssl-perl libnet-ssleay-perl libnspr4 libnss3 libosp5 libostylelc2 libpaper-utils libpaper1 libpciaccess0 libpixman-1-0
  libsm6 libsombok3 libstdc++-7-dev libsyntaxperl libsys-hostname-long-perl libtcl8.6 libtexlua52 libtexluajit2 libthai-data libthai0 l
  libwww-robotrules-perl libxml1-protocol-perl libxml1-xcb1 libxaw7 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-rend
  libxml-parser-perl libxml-twig-perl libxml-xpathengine-perl libxml2-utils libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libx
  perl-openssl-defaults po-debconf poppler-data preview-latex-style python python-apt python-minimal python2.7 python2.7-minimal rake
  sgmlspl t1utils tcl tcl8.6 tex-common tex-gyre texlive texlive-base texlive-bibtex-extra texlive-binaries texlive-extra-utils texli
  texlive-pictures texlive-plain-generic texlive-science texlive-xetex tipa tk tk8.6 unzip x11-common x11-utils x11-xserver-utils xbi
Suggested packages:
  docbook graphicsmagick-imagemagick-compat | imagemagick inkscape latex-cjk-all pdf-viewer texlive-lang-all texlive-lang-cyrillic tr
  | docbook-xsl-doc docbook-xsl-saxon fop libsaxon-java libxalan2-java libxslthl-java xalan debian-keyring fonts-noto g++-multilib g+
  libncurses-dev libdigest-hmac-perl libgssapi-perl cups-common bzip liblcms2-utils libcrypt-ssleay-perl libencode-hanextra-perl libpo
  doc-base libmail-box-perl poppler-utils fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic f
  ruby-dev bundler sgml-base-doc perlsgml w3-recs tcl-tclreadline debhelper perl-tk xpdf-reader | pdf-viewer chkxet dvipng fra
  libspreadsheet-parseexcel-perl texlive-latex-extra-doc texlive-latex-recommended-doc texlive-pstricks dot2tex prerex ruby-tcltk | l
  texlive-htmlxml xfonts-cyrillic
Recommended packages:
  uboot-mkimage
The following NEW packages will be installed:
  build-essential dblatex dblatex-doc docbook-dsssl docbook-utils docbook-xml docbook-xsl dpkg-dev fakeroot fontconfig-config fonts-d
  gettext ghostscript gsfonts intltool-debian javascript-common kernel-common kernel-package libalgorithm-diff-perl libalgorithm-diff
  libcups2 libcupsfilters1 libcupsimage2 libdata-dump-perl libdatatriel libdpkg-perl libdrm-amdgpu libdrm-intel libdrm-nouveau2 libdr
  libfile-listing-perl libfile-mimeinfo-perl libfile-which-perl libfont-afm-perl libfontconfig1 libfontenc1 libgl1 libgl1-mesa-dri li
  libhtml-form-perl libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libhttp-cookies-perl libhttp-daemon
  libipc-system-simple-perl libjpeg8 libjpeg2dec0 libjpeg-turbo8 libjpeg8 libjs-jquery libkpathsea6 liblcms2-2 libllvm7 liblwp-mediat
  libnet-smtp-ssl-perl libnet-ssleay-perl libnspr4 libnss3 libosp5 libostylelc2 libpaper-utils libpaper1 libpciaccess0 libpixman-1-0
  libsm6 libsombok3 libstdc++-7-dev libsyntaxperl libsys-hostname-long-perl libtcl8.6 libtexlua52 libtexluajit2 libthai-data libthai0 l
  libwww-robotrules-perl libxml1-protocol-perl libxml1-xcb1 libxaw7 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-rend
  libxml-parser-perl libxml-twig-perl libxml-xpathengine-perl libxml2-utils libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libx
  perl-openssl-defaults po-debconf poppler-data preview-latex-style python python-apt python-minimal python2.7 python2.7-minimal rake
  sgmlspl t1utils tcl tcl8.6 tex-common tex-gyre texlive texlive-base texlive-bibtex-extra texlive-binaries texlive-extra-utils texli
  texlive-pictures texlive-plain-generic texlive-science texlive-xetex tipa tk tk8.6 unzip x11-common x11-utils x11-xserver-utils xbi
0 upgraded, 233 newly installed, 0 to remove and 32 not upgraded.
Need to get 367 MB of archives.
After this operation, 1076 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```



# 관련 도구 설치 (중간에 이렇게 물어봄)

```
Modified configuration file
kernel-img.conf: A new version (/etc/kernel-img.conf.ucf-new) of configuration file /etc/kernel-img.conf is available, but the version installed currently has been locally modified.
What do you want to do about modified configuration file kernel-img.conf?

install the package maintainer's version
keep the local version currently installed
show the differences between the versions
show a side-by-side difference between the versions
start a new shell to examine the situation

<Ok>
```



# Configuration 설정 (커널 소스 디렉토리에서 작업)

- # make oldconfig
  - 현재 부팅된 커널의 설정을 복사하여 새로운 커널 컴파일에 이용
  - 만약 새롭게 추가된 설정들이 있다면 사용 여부를 묻는 화면이 나옴
    - Enter 를 입력하면 기본값이 선택됨. Enter 쪽 누르고 있으면 됨.

```
Compiler optimization level
> 1. Optimize for performance (CC_OPTIMIZE_FOR_PERFORMANCE)
   2. Optimize for size (CC_OPTIMIZE_FOR_SIZE)
choice[1-2?]: 1
Enable bpf() system call (BPF_SYSCALL) [Y/?] y
  Permanently enable BPF JIT and remove BPF interpreter (BPF_JIT_ALWAYS_ON) [Y/n/?] y
Enable userfaultfd() system call (USERFAULTFD) [Y/n/?] y
Enable rseq() system call (RSEQ) [Y/n/?] (NEW) █
```

- # make menuconfig
  - 추가로 수정할 필요가 있을 경우, GUI를 제공함
  - 현재는 추가 수정할 필요 없음
  - 혹시 들어갔다면, tab 키를 이용해 아랫쪽에서 exit 선택 (enter)
    - 아니면 ctrl + c



# Configuration 설정

.config - Linux/x86 4.18.20hpcpark Kernel Configuration

Linux/x86 4.18.20hpcpark 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 7.3.0-27ubuntu1~18.04) 7.3.0 \*\*\*

[\*] 64-bit kernel

General setup --->

[\*] Enable loadable module support --->

[\*] Enable the block layer --->

Processor type and features --->

Power management and ACPI options --->

Bus options (PCI etc.) --->

Executable file formats / Emulations --->

[\*] Networking support --->

Device Drivers --->

Ubuntu Supplied Third-Party Device Drivers --->

Firmware Drivers --->

File systems --->

Kernel hacking --->

Security options --->

-\*- Cryptographic API --->

[\*] Virtualization --->

Library routines --->

<Select>

< Exit >

< Help >

< Save >

< Load >



# 커널 이름 수정

- # vi Makefile
  - Extraversion 뒤에 dash + 본인 학번 입력
  - E.g.) EXTRAVERSION = -2017111222
- 문자열 뒤에 빈 공간이 있으면 안됨
  - Vi 의 경우, “:set list” 명령을 입력해서 공백 문자 확인 가능 (아래 오른쪽 그림 처럼 EOL 이 표시됨)

주의! 잘 읽어보고 진행할 것.  
여기서 오류 많이 남

```
# SPDX-License-Identifier: GPL-2.0
VERSION = 4
PATCHLEVEL = 18
SUBLEVEL = 20
EXTRAVERSION = -hcpark
NAME = Merciless Moray
# *DOCUMENTATION*
```

```
# SPDX-License-Identifier: GPL-2.0$
VERSION = 4$
PATCHLEVEL = 18$
SUBLEVEL = 20$
EXTRAVERSION = -hcpark$
NAME = Merciless Moray$
$
# *DOCUMENTATION*$
```



# 컴파일 하기 전에 확인

- Makefile에 이름이 정확히 수정되었나?
- 용량이 충분한가?
  - 약 20GB 이상 필요 (configuration 에 따라 달라질 수 있음)
  - df- h 로 확인

```
root@hcpark:/usr/src/linux-source-4.18.0/linux-source-4.18.0# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            2.0G   0    2.0G   0% /dev
tmpfs           395M  752K  394M   1% /run
/dev/vda1       39G   4.6G   34G  12% /
tmpfs           2.0G   0    2.0G   0% /dev/shm
tmpfs           5.0M   0    5.0M   0% /run/lock
tmpfs           2.0G   0    2.0G   0% /sys/fs/cgroup
/dev/vda15      105M  3.6M  101M   4% /boot/efi
tmpfs           395M   0    395M   0% /run/user/1000
root@hcpark:/usr/src/linux-source-4.18.0/linux-source-4.18.0#
```

```
root@hcpark:/usr/src/linux-source-4.18.0# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            2.0G   0    2.0G   0% /dev
tmpfs           395M  728K  394M   1% /run
/dev/vda1       39G   27G   13G  68% /
tmpfs           2.0G   0    2.0G   0% /dev/shm
tmpfs           5.0M   0    5.0M   0% /run/lock
tmpfs           2.0G   0    2.0G   0% /sys/fs/cgroup
/dev/vda15      105M  3.6M  101M   4% /boot/efi
tmpfs           395M   0    395M   0% /run/user/1000
root@hcpark:/usr/src/linux-source-4.18.0#
```

# Compile!

---

- `# time make -j 48`
  - 커널 소스 컴파일
  - `-j` option: To compile with 48 compiling threads
  - Recommend number of threads = 4x number of your CPU cores
  - `# lscpu`
    - to know the number of CPU cores
  - J-Cloud 에서 약 10-20분. 컴파일 하는 동안 심심하면?
    - SSH 세션을 추가로 열고, `sudo top` 실행
      - 현재 CPU, Memory 등 usage 확인 (윈도우의 작업관리자와 유사)
      - 구글에서 사용 방법 검색해볼 것
      - 모든 코어가 100% 열심히 일을 하고 있는가? Thread 개수와 어떤 관계일까?
- `# time make -j 48 modules_install install`
  - to install the compiled kernel in your system and update the boot menu
  - 중간에 또 패키지 설치를 물어보면, 설치할 것
    - Install the package maintainer's version

# 오류 화면

- 아래와 같은 오류가 나타날 수 있음
  - 메모리 부족으로 인한 오류
- 무시하고 다시 컴파일 재개
  - `time make -j 48`

```
CC [M] drivers/gpu/drm/nouveau/nv50_fence.o
CC [M] drivers/gpu/drm/nouveau/nv84_fence.o
CC [M] drivers/gpu/drm/nouveau/nvc0_fence.o
LD [M] drivers/gpu/drm/nouveau/nouveau.o
AR drivers/gpu/drm/built-in.a
AR drivers/gpu/built-in.a
Makefile:1061: recipe for target 'drivers' failed
make: *** [drivers] Error 2

real    11m37.800s
user    117m5.451s
sys     20m10.856s
root@hcpark:/usr/src/linux-source-4.18.0/linux-source-4.18.0#
```



# 컴파일 결과 화면 (제출할 화면 #1)

- 작업 경로 아래의 arch/x86\_64/boot/ 에 bzImage 파일이 잘 생성되었으면 컴파일이 잘 완료된 것
  - 생성된 시간으로 확인해 봄
  - 에러가 난 경우, 중간 과정에서 에러가 나고 종료되어 있을 것

```
LD [M] sound/usb/snd-usb-audio.ko
LD [M] sound/usb/snd-usbmidi-lib.ko
LD [M] sound/usb/usx2y/snd-usb-us122l.ko
LD [M] sound/usb/usx2y/snd-usb-usx2y.ko
LD [M] sound/x86/snd-hdmi-lpe-audio.ko
LD [M] ubuntu/hio/hio.ko
LD [M] ubuntu/vbox/vboxguest/vboxguest.ko
LD [M] ubuntu/vbox/vboxsf/vboxsf.ko
LD [M] ubuntu/xr-usb-serial/xr_usb_serial_common.ko
LD [M] virt/lib/irqbypass.ko

real    9m30.483s
user    147m35.628s
sys     29m3.209s
root@hpcpark:/usr/src/linux-source-4.18.0/linux-source-4.18.0# ls -al arch/
.gitignore alpha/      arm/      c6x/      hexagon/  m68k/     mips/     nios2/
Kconfig     arc/      arm64/    h8300/    ia64/     microblaze/ nds32/    openrisc/
root@hpcpark:/usr/src/linux-source-4.18.0/linux-source-4.18.0# ls -alh arch/x86_64/boot/bzImage
lrwxrwxrwx 1 root root 22 Mar 20 03:22 arch/x86_64/boot/bzImage -> ../../x86/boot/bzImage
root@hpcpark:/usr/src/linux-source-4.18.0/linux-source-4.18.0# date
Wed Mar 20 03:27:01 UTC 2019
```



# 모듈 인스톨 결과 화면 (제출할 화면 #2)

- 아래 결과에는 warning 이 있으나, 슬라이드를 따라 정확히 수행하면 문제 없음

```
Generating grub configuration file ...
dpkg: warning: version '4.18.20_hcpark' has bad syntax: invalid character in version number
dpkg: warning: version '4.18.20_hcpark~old' has bad syntax: invalid character in version number
dpkg: warning: version '4.18.20_hcpark' has bad syntax: invalid character in version number
Found linux image: /boot/vmlinuz-4.18.20_hcpark
Found initrd image: /boot/initrd.img-4.18.20_hcpark
dpkg: warning: version '4.18.20_hcpark~old' has bad syntax: invalid character in version number
Found linux image: /boot/vmlinuz-4.18.20_hcpark.old
Found initrd image: /boot/initrd.img-4.18.20_hcpark
Found linux image: /boot/vmlinuz-4.15.0-46-generic
Found initrd image: /boot/initrd.img-4.15.0-46-generic
Found linux image: /boot/vmlinuz-4.15.0-45-generic
Found initrd image: /boot/initrd.img-4.15.0-45-generic
done

real    5m6.822s
user    3m6.765s
sys     1m2.783s
root@hcpark:/usr/src/linux-source-4.18.0/linux-source-4.18.0#
```

# 재부팅 및 확인 (제출할 화면 #3)

- # reboot
  - 재부팅이 완료된 이후,
- \$ hostnamectl
  - 본인이 수정한 커널 이름대로 나오는지 확인

```
ubuntu@hcpark:~$ hostnamectl
  Static hostname: hcpark
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 32cd22818fee491ab370e3a29dbdb4f7
        Boot ID: 761c59caba7a45b0bd7f91aeec4fef62
  Virtualization: kvm
  Operating System: Ubuntu 18.04.2 LTS
        Kernel: Linux 4.18.20-hcpark
  Architecture: x86_64
ubuntu@hcpark:~$
```



---


부록:

# Virtual Box 를 이용한 리눅스 가상머신 설치



# Virtual Box 설치

- <https://www.virtualbox.org/wiki/Downloads>



## VirtualBox

### Download VirtualBox

Here, you will find links to VirtualBox binaries and its source code.

#### VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

- **VirtualBox 5.1.14 platform packages.** The binaries are released under the terms of the GPL version 2.
  - [Windows hosts](#)
  - [OS X hosts](#)
  - [Linux distributions](#)
  - [Solaris hosts](#)
- **VirtualBox 5.1.14 Oracle VM VirtualBox Extension Pack** [All supported platforms](#)  
Support for USB 2.0 and USB 3.0 devices, VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel VMs.  
The Extension Pack binaries are released under the [VirtualBox Personal Use and Evaluation License \(PUEL\)](#).  
*Please install the extension pack with the same version as your installed version of VirtualBox:  
If you are using **VirtualBox 5.0.32**, please download the extension pack [here](#).*
- **VirtualBox 5.1.14 Software Developer Kit (SDK)** [All platforms](#)

See the [changelog](#) for what has changed.

About

Screenshots

Downloads

Documentation

End-user docs

Technical docs

Contribute

Community



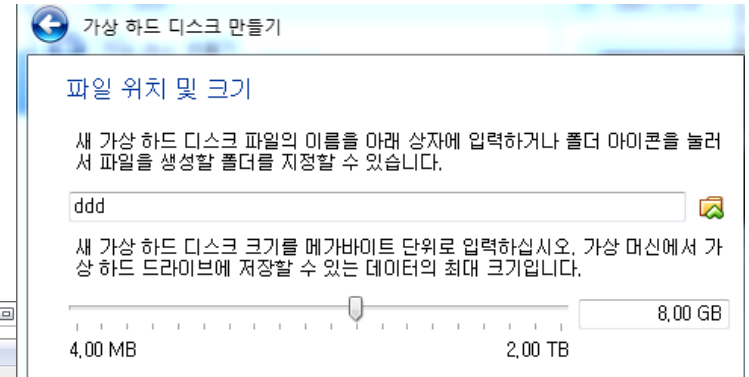
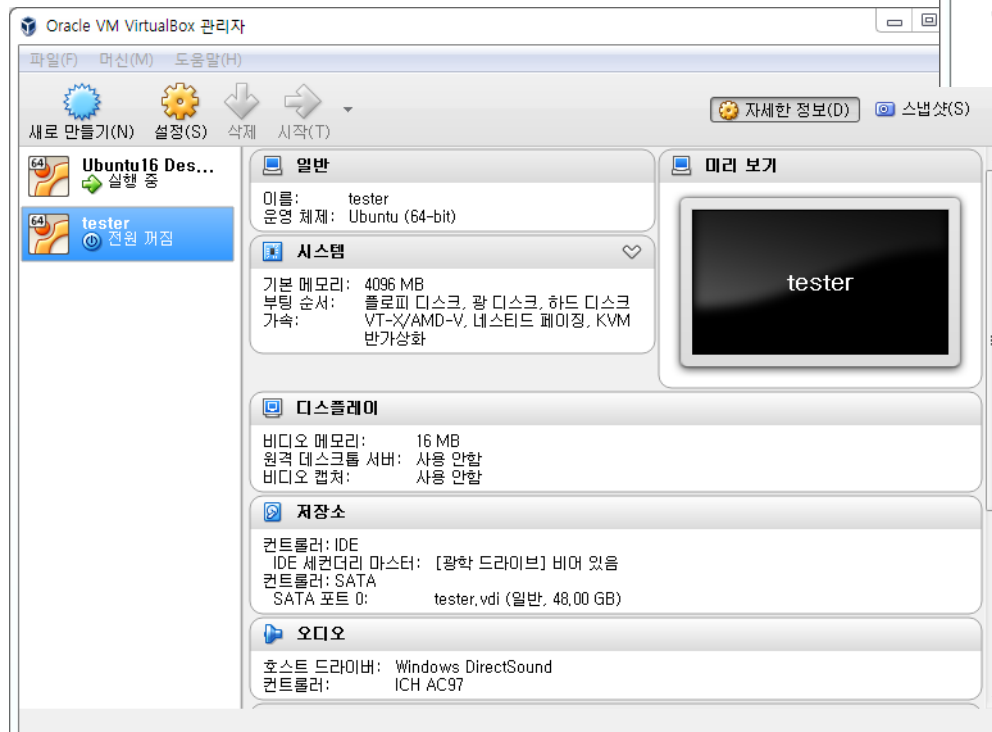
# Virtual Box 사용 유의사항

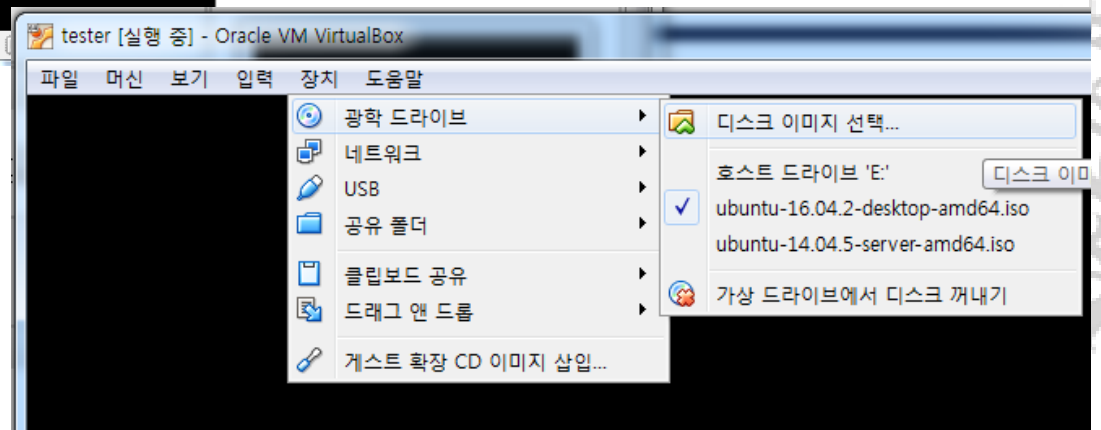
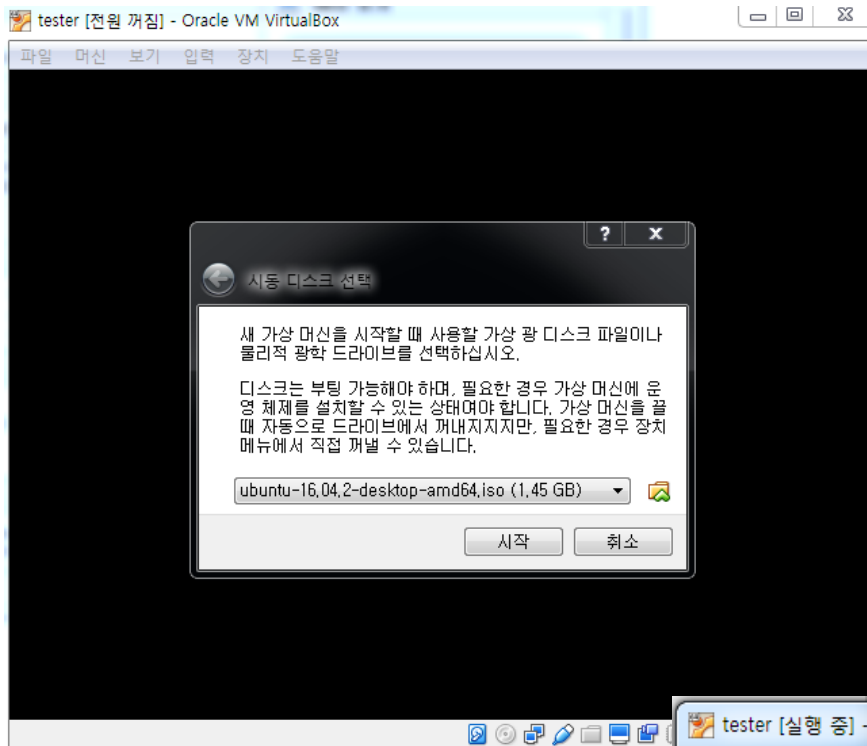
---

- 1번 설치 후, 학기가 끝날 때까지 업데이트 하지 말 것
  - 별 문제도 없는데 업데이트 했다가 괜히 문제가 생길 수 있음
  - 심각한 결함에 대한 업데이트가 있을 때만 수행
- VM 종료 시에는 항상 VM 내에서, 리눅스에서 정상 종료 시킬 것
  - Virtual Box 에서 강제 종료 하는 경우 VM 이미지가 손상될 수 있음
- 수정한 파일들을 상시 백업할 것
  - 하드 용량이 넉넉하다면 이미지 파일 자체를 백업해도 좋음
  - Samba 로 연결되어 있으므로, 윈도우의 저장 매체에 수동, 자동으로 백업 수행할 것

# Virtual Box VM 이미지 생성

- 메모리: 1 GB 이상, 2 GB 추천
- 스토리지: 40 GB 이상





# 리눅스 설치

---

- 우분투 리눅스 다운로드
  - <https://www.ubuntu.com/download/desktop>
  - Ubuntu 16.04.2 LTS
    - Old version: <http://old-releases.ubuntu.com/releases/16.04.2/>
      - ubuntu-16.04.2-desktop-amd64.iso 선택
- 설치 옵션: 대체로 default 옵션을 사용하면 됨
- 중요한 선택들
  - (가상) 디스크를 지우고 우분투 설치 (erase disk and install Ubuntu)
    - 홈디렉토리 암호화 사용하지 말 것
    - 디스크 파티션 LVM 사용하지 말 것
  - 자동로그인
  - 암호 잘 기억할 것



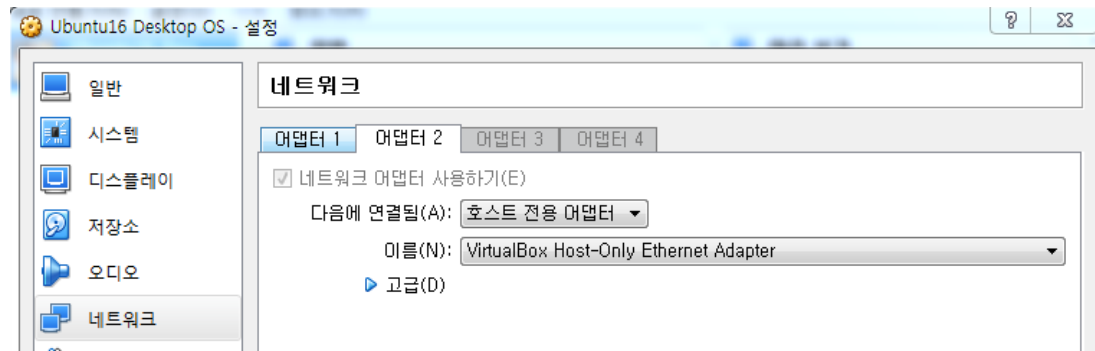
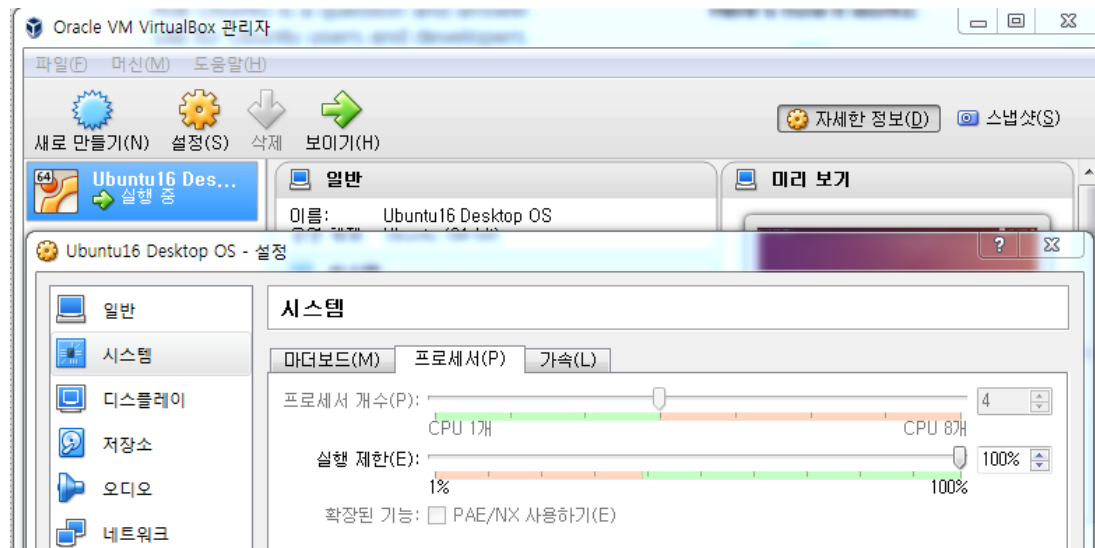
# 추가 설치

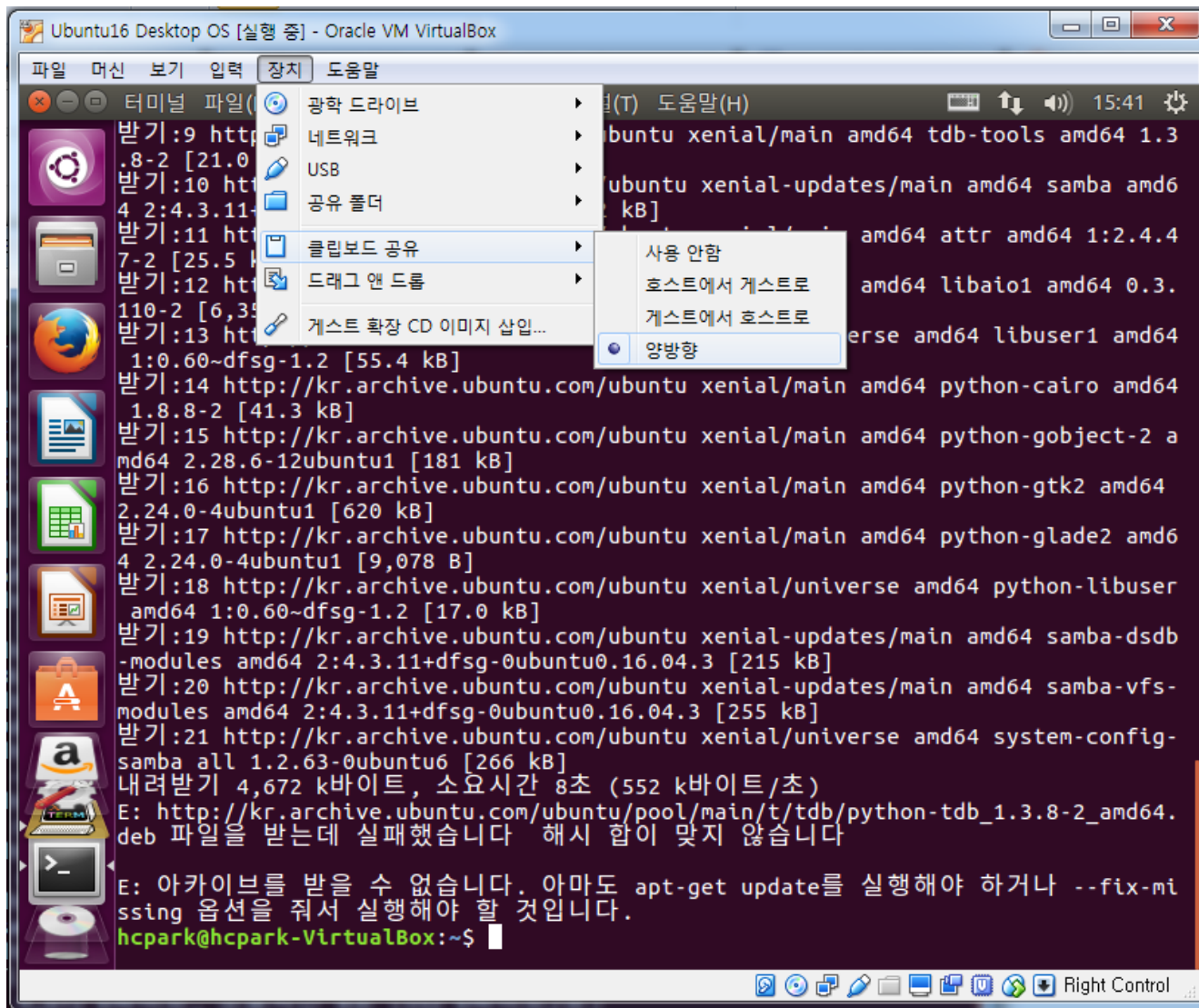
---

- 게스트 확장 CD 이미지 (설치 후 재부팅)
  - Run 클릭 후, 설정한 암호 입력
  - 클립보드 공유 - 양방향 설정
- (Terminal 실행 후 터미널 창에서 명령 실행)
- OpenSSH
  - `sudo apt-get install -y openssh-server`
- Samba server
  - `sudo apt-get install -y samba samba-common python-glade2 system-config-samba`

# 가상 머신 설정 변경

- 설치 후 시스템 종료
  - shutdown -h now
- 가상머신 설정
  - 여러 CPU를 이용하게 프로세서 개수 상향
    - 빠른 컴파일을 위해
    - 전체 프로세서 개수의 절반
  - 네트워크 추가
    - 어댑터2
    - 호스트 전용 어댑터
    - 호스트 윈도우와 통신을 위해





Ubuntu16 Desktop OS [실행 중] - Oracle VM VirtualBox

파일 마신 보기 입력 장치 도움말

hcpark@hcpark-VirtualBox: ~

```
hcpark@hcpark-VirtualBox:~$ ping 192.168.56.2
PING 192.168.56.2 (192.168.56.2) 56(84) bytes of data.
^C
--- 192.168.56.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

hcpark@hcpark-VirtualBox:~$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=128 time=0.320 ms
^C
--- 192.168.56.102 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.320/0.320/0.320/0.000 ms

hcpark@hcpark-VirtualBox:~$ ifconfig
enp0s3      Link encap:Ethernet  HWaddr 08:00:27:d0:5b:8b
            inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
            inet6 addr: fe80::f3e9:724c:e2ed:4a68/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:65 errors:0 dropped:0 overruns:0 frame:0
            TX packets:137 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:15199 (15.1 KB)  TX bytes:14196 (14.1 KB)

enp0s8      Link encap:Ethernet  HWaddr 08:00:27:54:c7:ef
            inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
            inet6 addr: fe80::505f:13fd:58e2:464d/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:301 errors:0 dropped:0 overruns:0 frame:0
            TX packets:134 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:38893 (38.8 KB)  TX bytes:21647 (21.6 KB)

lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:104 errors:0 dropped:0 overruns:0 frame:0
            TX packets:104 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1
            RX bytes:10785 (10.7 KB)  TX bytes:10785 (10.7 KB)

hcpark@hcpark-VirtualBox:~$
```

Right Control



VirtualBox Host-Only Network 상태

일반

연결

IPv4 연결:

인터넷에 연결되어 있지 않음

IPv6 연결:

인터넷에 연결되어 있지 않음

미디어 상태:

사용함

시간:

03:09:40

속도:

1.0 Gbps

자세히(E)...

작업

보냄

0

받은

0

속성(P)

사용 안 함(D)

진단(G)

닫기(C)

네트워크 연결 세부 정보

네트워크 연결 세부 정보(D):

속성	값
연결별 DNS 접미사	
설명	VirtualBox Host-Only Ethernet Adapter
물리적 주소	0A-00-27-00-00-1B
DHCP 사용	예
IPv4 주소	192.168.56.102
IPv4 서브넷 마스크	255.255.255.0
임대 시작 날짜	2017년 2월 23일 목요일 오후 3:51:01
임대 만료 날짜	2017년 2월 23일 목요일 오후 4:11:01
IPv4 기본 게이트웨이	
IPv4 DHCP 서버	192.168.56.100
IPv4 DNS 서버	
IPv4 WINS 서버	
TCP/IP를 통해 NetBIOS...	예
링크-로컬 IPv6 주소	fe80::c504:6fe6:565d:3414%27
IPv6 기본 게이트웨이	
IPv6 DNS 서버	fec0:0:0:ffff::1%1

닫기(C)

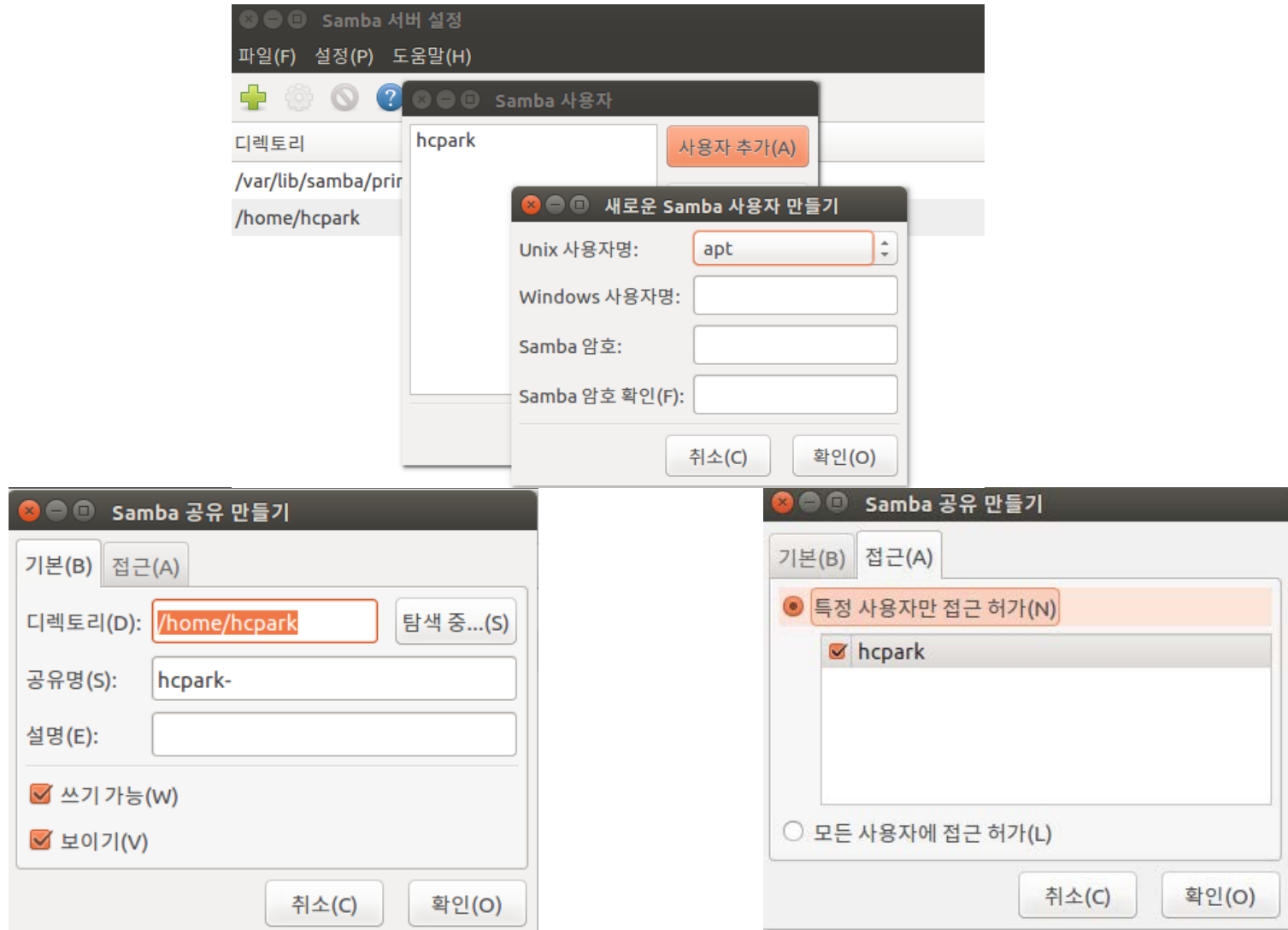
# Samba 설정

---

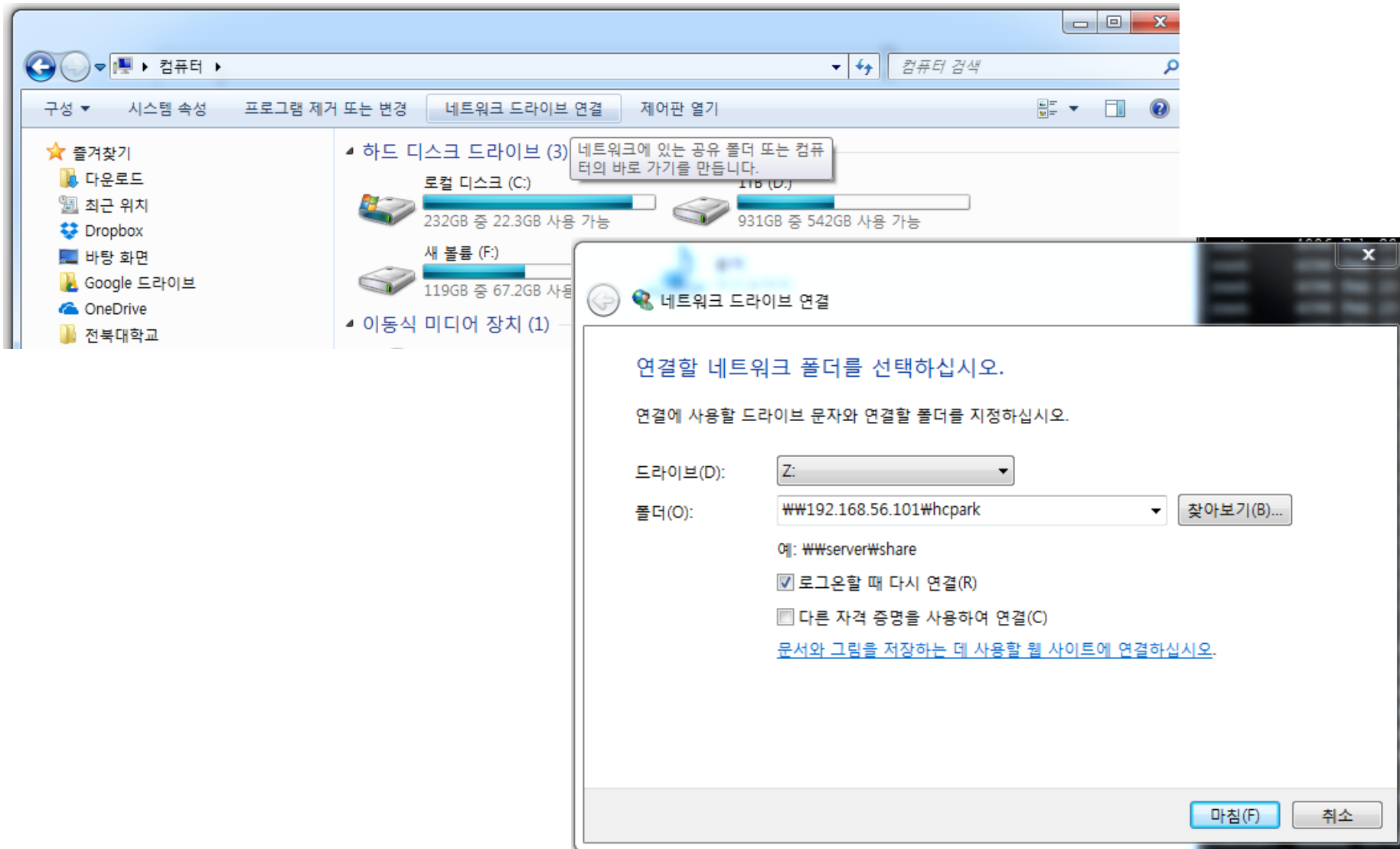
- system-config-samba
  - 첫 실행 시 에러
  - 해결: touch /etc/libuser.conf
- 접근 사용자 추가
  - 본인 계정 (예. hcpark)
- 접근 디렉토리 추가
  - 본인 홈 디렉토리 (예. /home/hcpark)



# Samba 설정



# Samba 네트워크 드라이브 연결





# SSH 클라이언트를 이용한 접근

---

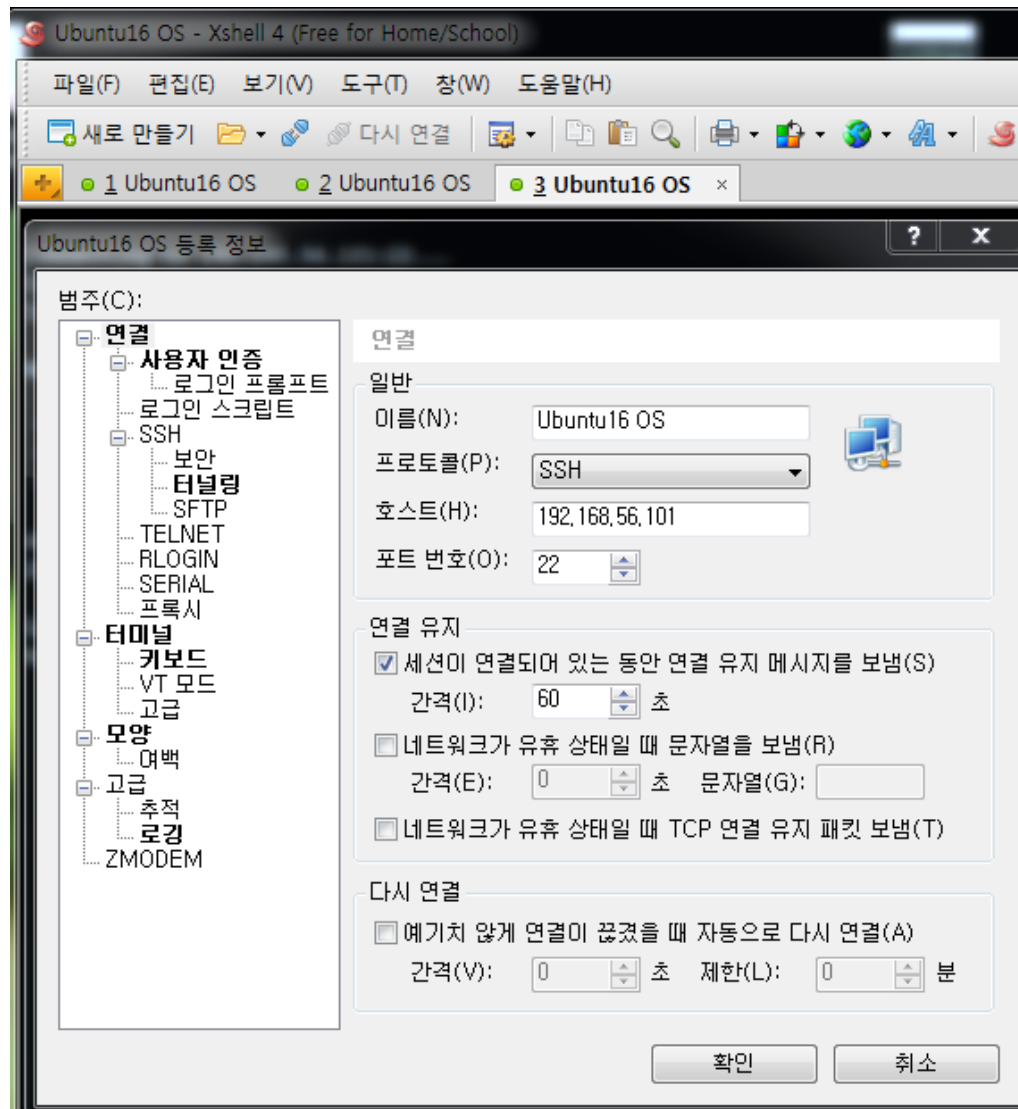
- 시큐어 셸(Secure Shell, SSH)
  - 네트워크 상의 다른 컴퓨터에 로그인하거나,
  - 원격 시스템에서 명령을 실행하고,
  - 다른 시스템으로 파일을 복사할 수 있도록 해 주는 프로토콜
- 기존의 rsh, rlogin, 텔넷 등을 대체
  - 강력한 인증 방법
  - 안전하지 못한 네트워크에서 안전하게 통신을 할 수 있는 기능
- 기본적으로는 22번 포트 사용
- 가장 일반적인 원격 세션 접속 프로토콜

# SSH 클라이언트를 이용한 접근

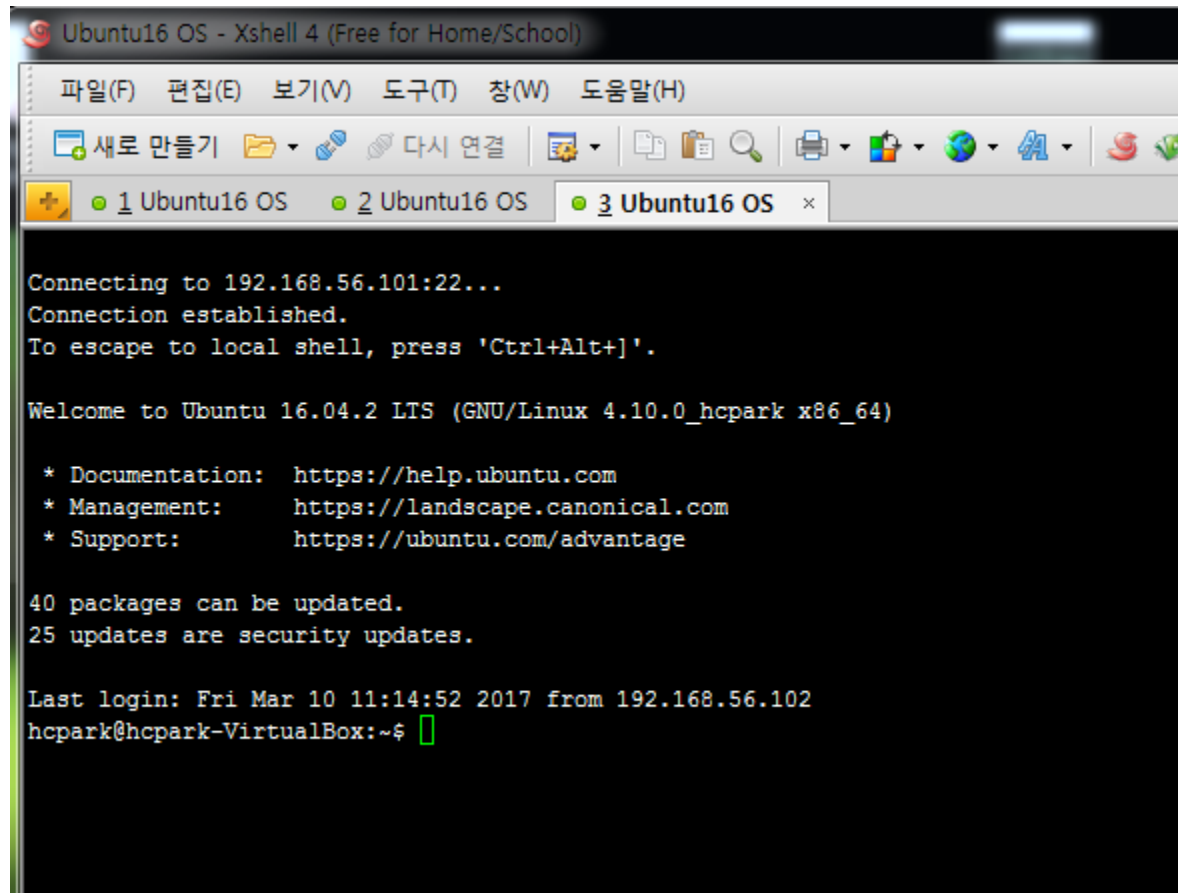
---

- SSH 클라이언트
  - SSH 프로토콜을 이용해 원격 서버에 접속하는 프로그램
  - PuTTY 등이 유명함
    - 단순하고 기본 기능 충실. 그러나 너무 단순함
- Xshell 5
  - 개인용, 교육용 무료 버전 배포
  - [http://www.netsarang.co.kr/download/free\\_license.html](http://www.netsarang.co.kr/download/free_license.html)
- 자유롭게 본인에 맞는 프로그램 사용할 것

# SSH 클라이언트를 이용한 접근



# SSH 클라이언트를 이용한 접근



The screenshot shows an Xshell 4 terminal window titled "Ubuntu16 OS - Xshell 4 (Free for Home/School)". The window has a menu bar with "파일(F)", "편집(E)", "보기(V)", "도구(T)", "창(W)", and "도움말(H)". Below the menu is a toolbar with icons for "새로 만들기", "다시 연결", and other functions. The terminal displays the following text:

```
Connecting to 192.168.56.101:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.

Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.10.0_hcpark x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

40 packages can be updated.
25 updates are security updates.

Last login: Fri Mar 10 11:14:52 2017 from 192.168.56.102
hcpark@hcpark-VirtualBox:~$
```

