

리눅스시스템 및 응용

Week 2

리눅스 소개 및 설치
(계속)

-
- 2주차 1번째 시간은
 - 지난주 리눅스 소개 부분 중 빠진 부분 계속 이어서 강의

Linux Kernel

• 커널의 구성

- 안정 버전(stable version): 이미 검증된 개발 완료 코드로 구성
- 메인라인 버전(mainline version): 토르발스가 개발 중인 버전, 안정 버전이 나오기 전에 추가된 기능을 미리 접하고 싶을 때 사용
- 프리패치 버전(prepatch version): 안정 버전이 나오기 전에 추가된 기능을 미리 접하고 싶을 때 사용

• 커널 버전 업그레이드

- 2019년 5월을 기준으로 안정화된 최신 커널은 5.1.15 버전
- 파일 이름에 붙은 숫자 **5.1.15의 5는 주버전**(major version), **1은 부버전**(minor version), **15는 패치 버전**(patch version)을 의미
- 배포판에 포함된 기본 커널을 사용자가 직접 최신 커널로 업그레이드할 수 있으며, '커널 업그레이드' 또는 '커널 컴파일'이라고 함

커널 버전	0.01	1.0	2.0	2.2	2.4	2.6	3.0	3.8	4.0	4.4	4.9	4.19	5.0
발표 연도	1991	1994	1996	1999	2001	2003	2011	2013	2015	2016	2017	2018	2019

Debian Linux / Ubuntu Linux

- 데비안 리눅스

- 유명한 리눅스 배포판 중 하나이며 1993년에 이언 머독(Ian Murdock)이 창시한 데비안 프로젝트(Debian Project)에서 제작
- 데비안 리눅스의 정식 버전은 1996년 1.1 버전(코드명 Buzz)으로 시작, 2019년 10 버전 발표
- 패키지의 설치와 업그레이드가 상당히 단순하고 apt 프로그램을 이용하여 소프트웨어 설치나 업데이트가 자동으로 진행됨

- 우분투 리눅스

- 데비안 리눅스를 기초로 유니티(unity) 데스크톱 환경을 사용하는 리눅스 배포판
- 첫 버전을 출시한 이후 계속 업그레이드됨, 인기 리눅스 배포판 중 하나
- 기본적으로 우분투 데스크톱과 우분투 서버를 배포
- 쿠분투(Kubuntu), 우분투 킬린(Ubuntu Kylin), 루분투(Lubuntu), 주분투(Xubuntu) 등 다양한 배포판 개발, 이를 통틀어 '우분투 플레이버스(Ubuntu flavours)'라고 부름

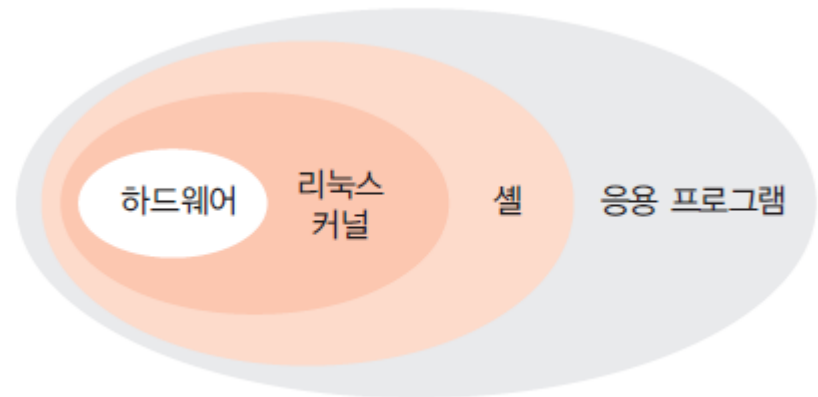
우분투 리눅스

- 우분투 리눅스 특징

- 데비안 기반의 리눅스 중 가장 성공한 데스크톱 배포판으로 많이 사용함
- 리눅스는 공개 소프트웨어이며 무료로 사용할 수 있다.
- 유닉스와의 완벽한 호환성을 유지한다.
- 서버용 운영체제로 많이 사용된다.
- 편리한 GUI 환경을 제공한다.
- 우분투는 데비안 GNU/리눅스에 바탕
- 그놈(GNOME)을 기본 데스크톱 환경으로 사용
- 우분투 개발의 지휘자로 알려진 마크 셔틀워스(Mark Shuttleworth)가 세운 영국 캐노니컬의 지원
- 우분투의 기본 철학: 전 세계의 누구나 어렵지 않게 리눅스를 사용하자는 것
- 버전 11.04부터 유니티를 기본 데스크톱으로 사용했으나 17.10 버전부터 그놈으로 바꿈
- 기본 시스템 도구 외에도 리브르오피스LibreOffice, 파이어폭스 웹 브라우저, 에볼루션(메일/일정 편집기), 옴퍼시(메신저) 등 다양한 응용 소프트웨어도 함께 제공

리눅스 배포판의 구조

- 커널: 리눅스의 핵심
 - 프로세스/메모리/파일 시스템/장치 관리
 - 컴퓨터의 모든 자원 초기화 및 제어 기능
- 셸: 사용자 인터페이스
 - 명령 해석
 - 프로그래밍 기능
 - 리눅스 기본 셸: 배시 셸(리눅스 셸)
- 응용 프로그램
 - 각종 프로그래밍 개발 도구
 - 문서 편집 도구
 - 네트워크 관련 도구 등



리눅스 배포판

버전	코드명	커널 버전	배포일	지원 기간
10.04(LTS)	Lucid Lynx(빛나는 스라소니)	2.6.32	2010년 4월 29일	~ 2013년 5월 9일
10.10	Maverick Meerkat(사나운 미어캣)	2.6.35	2010년 10월 10일	~ 2012년 4월 10일
11.04	Natty Narwhal(산뜻한 일각돌고래)	2.6.38	2011년 4월 28일	~ 2012년 10월 28일
11.10	Oneiric Ocelot(꿈꾸는 오실롯)	3.0.0	2011년 10월 13일	~ 2013년 5월 9일
12.04(LTS)	Precise Pangolin(꼼꼼한 천산갑)	3.2.0	2012년 4월 26일	~ 2017년 4월 28일
12.10	Quantal Quetzal(양질의 케찰)	3.5.0	2012년 10월 18일	~ 2014년 5월 16일
13.04	Raring Ringtail(근질근질한 링테일)	3.8.0	2013년 4월 25일	~ 2014년 1월 27일
13.10	Saucy Salamander(건방진 도롱뇽)	3.11.0	2013년 10월 17일	~ 2014년 7월 17일
14.04(LTS)	Trusty Tahr(믿음직한 산양)	3.13.0	2014년 4월 17일	~ 2019년 4월
14.10	Utopic Unicom(유토피아 유니콘)	3.16.0	2014년 10월 23일	~ 2015년 7월 23일
15.04	Vivid Vervet(활활한 버빗원숭이)	3.19.3	2015년 4월 23일	~ 2016년 2월 4일
15.10	Wily Werewolf(교활한 늑대 인간)	4.2.0	2015년 10월 22일	~ 2016년 7월 28일
16.04(LTS)	Xenial Xerus(다정한 땅다람쥐)	4.4.0	2016년 4월 21일	~ 2021년 4월
16.10	Yakkety Yak(수다스러운 야크)	4.8.0	2016년 10월 13일	~ 2017년 7월 20일
17.04	Zesty Zapus(자극하는 자무스)	4.10.0	2017년 4월 13일	~ 2018년 2월
17.10	Artful Aardvark(교활한 땅돼지)	4.13.0	2017년 10월 19일	~ 2018년 7월

- 최근 우분투 배포판 : 출처 나무위키

17.10	Artful Aardvark	2017년 10월 20일	4.13	2018년 07월 19일	-
<u>18.04 LTS</u>	Bionic Beaver	<u>2018년 04월 27일</u>	4.15	<u>2023년 04월</u>	-
18.10	Cosmic Cuttlefish	2018년 10월 18일	4.18	2019년 07월	-
19.04	Disco Dingo	2019년 4월 18일	5.0	2020년 1월	-
19.10	Eoan Ermine	2019년 10월 17일	5.3	2020년 7월	-
<u>20.04 LTS</u>	Focal Fossa	<u>2020년 04월 23일</u>	5.4	<u>2025년 4월</u>	-
20.10	Groovy Gorilla	2020년 10월 22일	5.8	2021년 7월	-
21.04	Hirsute Hippo	2021년 4월 22일	5.11	2022년 1월	-
21.10	Impish Indri	2021년 10월 14일	TBC	2022년 6월	-
<u>22.04 LTS</u>	미정	2022 미정	TBC	미정	-

- 우분투 리눅스 버전 주기

- 일반 버전은 약 6개월마다, LTS(장기, 5년) 버전은 2년마다 새로운 버전 발표
- Ex) 우분투 15.10 - 2015년 10월에 발표한 일반 버전,
우분투 18.04 LTS - 2018년 4월에 발표한 장기 지원 버전

- 최소 혹은 권장 하드웨어 사양

- CPU: 2GB 이상의 듀얼코어 프로세서
- 하드디스크의 여유 공간: 25GB 이상(추가 설치에 따라서 달라질 수 있음)
- 메모리: 2GB 이상
- 그래픽 카드: 1024×768 이상의 해상도 지원

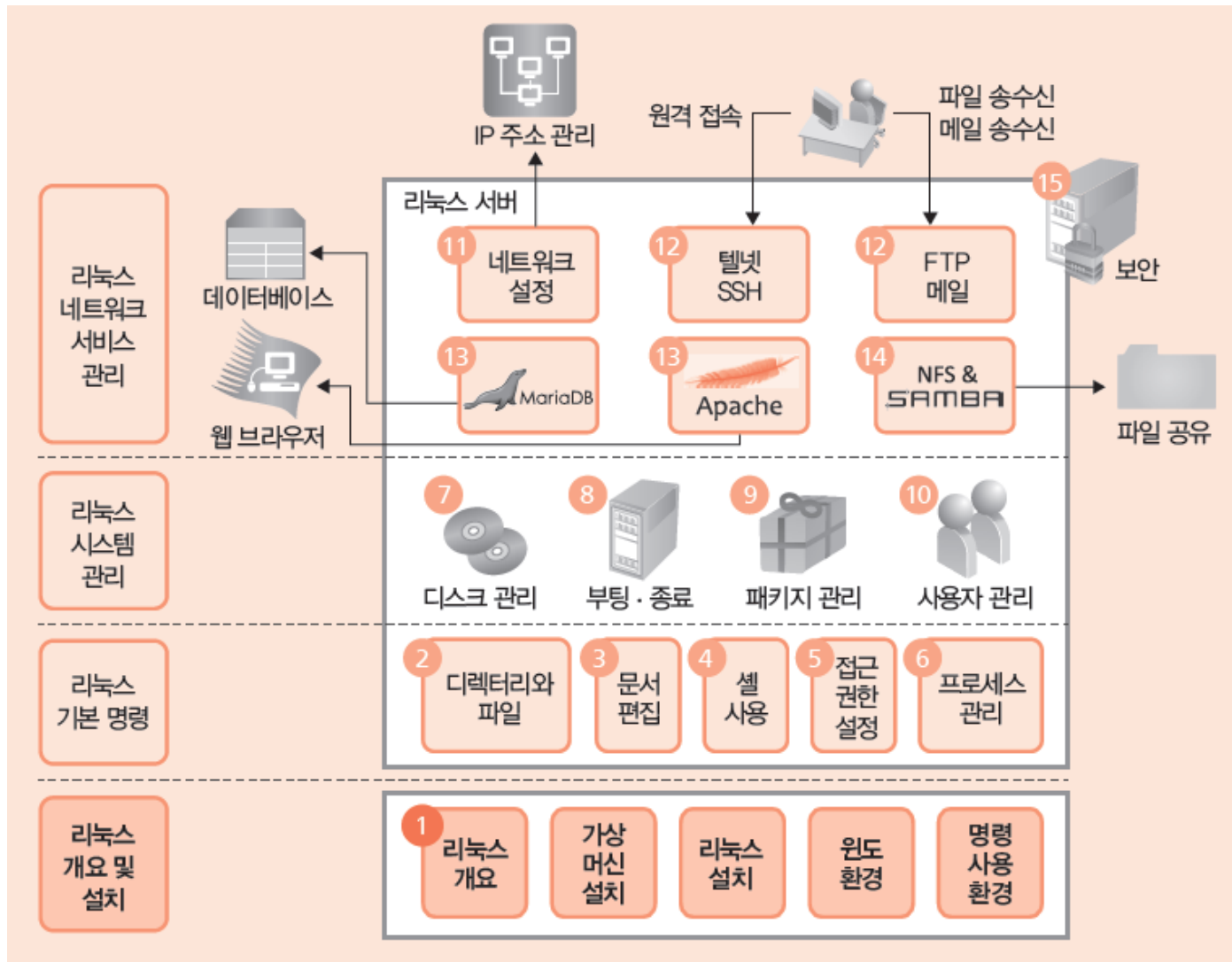
- 패키지에 대한 세부 내용

- 우분투 프로젝트 위키 <https://wiki.ubuntu.com/BionicBeaver/ReleaseNotes>

- 우분투의 의미와 로고

- 'Ubuntu'는 남아프리카 어느 부족의 말로, '너가 있으니 나도 있다'라는 뜻

• 리눅스 설치 후 환경



Virtual Machine 가상머신

- 가상머신 혹은 가상기계
- 하드웨어 위에 설치되는 모든 소프트웨어는 가상기계로 생각할 수 있음
- 가상머신
 - PC 1대만으로 여러 대의 PC를 운영하는 것처럼 만드는 방법
 - 가상머신 소프트웨어 또는 가상머신 프로그램이라 부름
- 대표적 가상머신 종류

가상 머신	호스트 OS	게스트 OS
VMware	윈도 계열 운영체제, 대부분의 리눅스, 애플 맥	윈도 계열 운영체제, 대부분의 리눅스 배포판, 솔라리스, Mac OS
버추얼 PC	윈도 계열 운영체제	윈도 계열 운영체제, 일부 리눅스, 솔라리스
버추얼 박스	윈도 계열 운영체제, 대부분의 리눅스, 애플 맥, 솔라리스	윈도 계열 운영체제, 대부분의 리눅스 배포판, 솔라리스, Mac OS, OpenBSD

- 가상머신 소프트웨어

- 컴퓨터에 설치된 운영체제(호스트 OS) 안에 가상의 컴퓨터를 만들고 이 가상의 컴퓨터 안에 또 다른 운영체제(게스트 OS)를 설치·운영할 수 있도록 제작된 소프트웨어
- PC에 설치되어 있는 윈도우를 **호스트 운영체제**(host operating system) 또는 **호스트 OS**, 가상머신에 설치한 운영체제를 **게스트 운영체제**(guest operating system) 또는 **게스트 OS**라고 함

- 일반적인 환경과의 비교

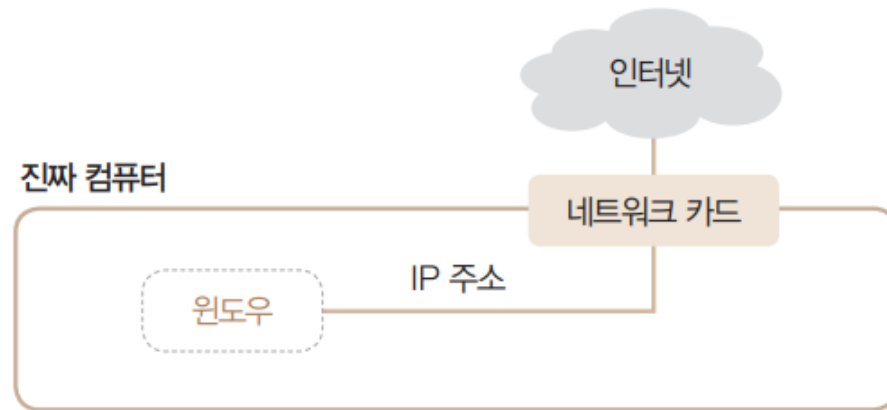
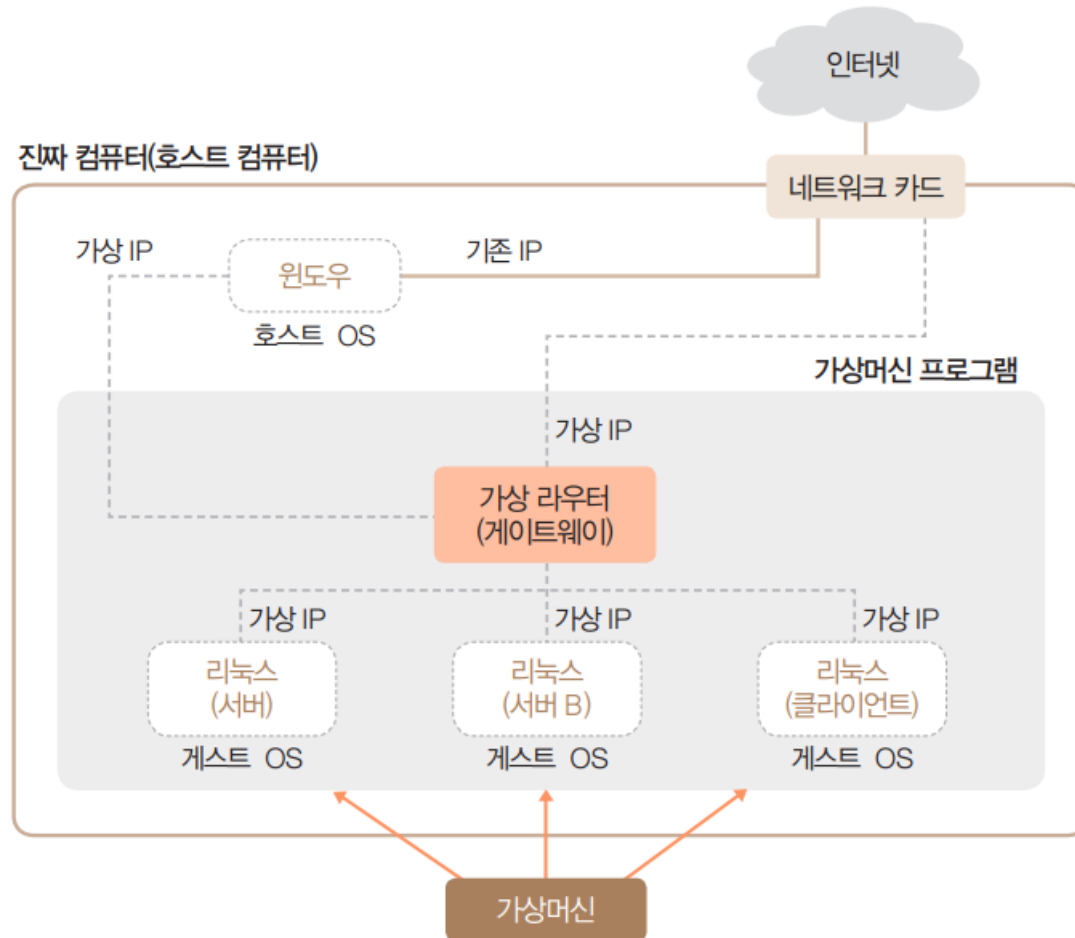


그림 1-7 컴퓨터 1대, 운영체제 1개로 구성된 환경

- 호스트 컴퓨터에는 윈도우가 설치되어 있고, 기존에 설정된 IP 주소를 이용하여 네트워크 카드를 통해 인터넷에 접속 가능

Virtual Machine



- VMware

구분	VMware Workstation Pro	VMware Workstation Player
호스트 OS	64비트 윈도우 7 이후	64비트 윈도우 7 이후
게스트 OS	모든 16비트, 32비트, 64비트 윈도우와 대부분의 리눅스 운영체제	모든 16비트, 32비트, 64비트 윈도우와 대부분의 리눅스 운영체제
라이선스	유료	유료(회사, 학교, 공공기관 등에서 사용하는 경우) 또는 무료(개인이 가정에서 사용하는 경우)
라이선스 키	유료(VMware 사이트에서 30일 평가판 다운로드)	무료인 경우 필요 없음
가상머신 생성 기능	○	○
스냅샷 기능	○	×
가상 네트워크 사용자 설정 기능	○	×(가상 네트워크 사용자 설정을 공식적으로 지원하지 않음)
비고	여러 가지 부가 기능이 있음	부가 기능이 별로 없음

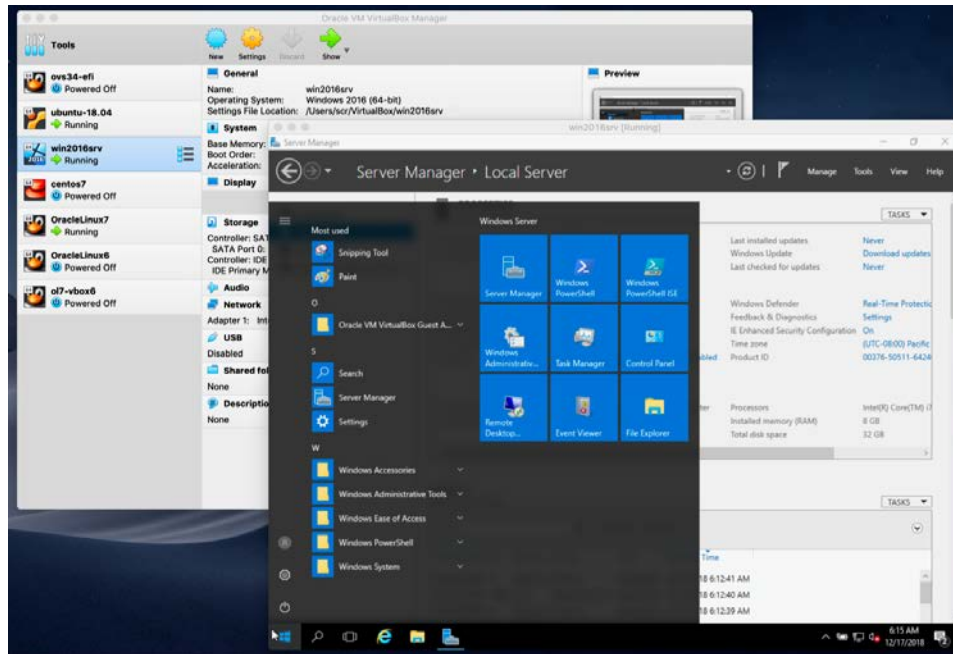
VirtualBox

- **User Manual** Version 6.1.26, 2021
- Oracle VM VirtualBox is a cross-platform virtualization application. What does that mean?
- For one thing, it installs on your existing Intel or AMD-based computers, whether they are running Windows, Mac OS X, Linux, or Oracle Solaris operating systems (OSes).
- Secondly, it extends the capabilities of your existing computer so that it can run multiple OSes, inside multiple virtual machines, at the same time.



**Oracle® VM
VirtualBox®**

- The following screenshot shows how Oracle VM VirtualBox, installed on an Apple Mac OS X computer, is running Windows Server 2016 in a virtual machine window.



- 장점

- Running multiple operating systems simultaneously
- Easier software installations
- Testing and disaster recovery
- Virtualization can significantly reduce hardware and electricity costs

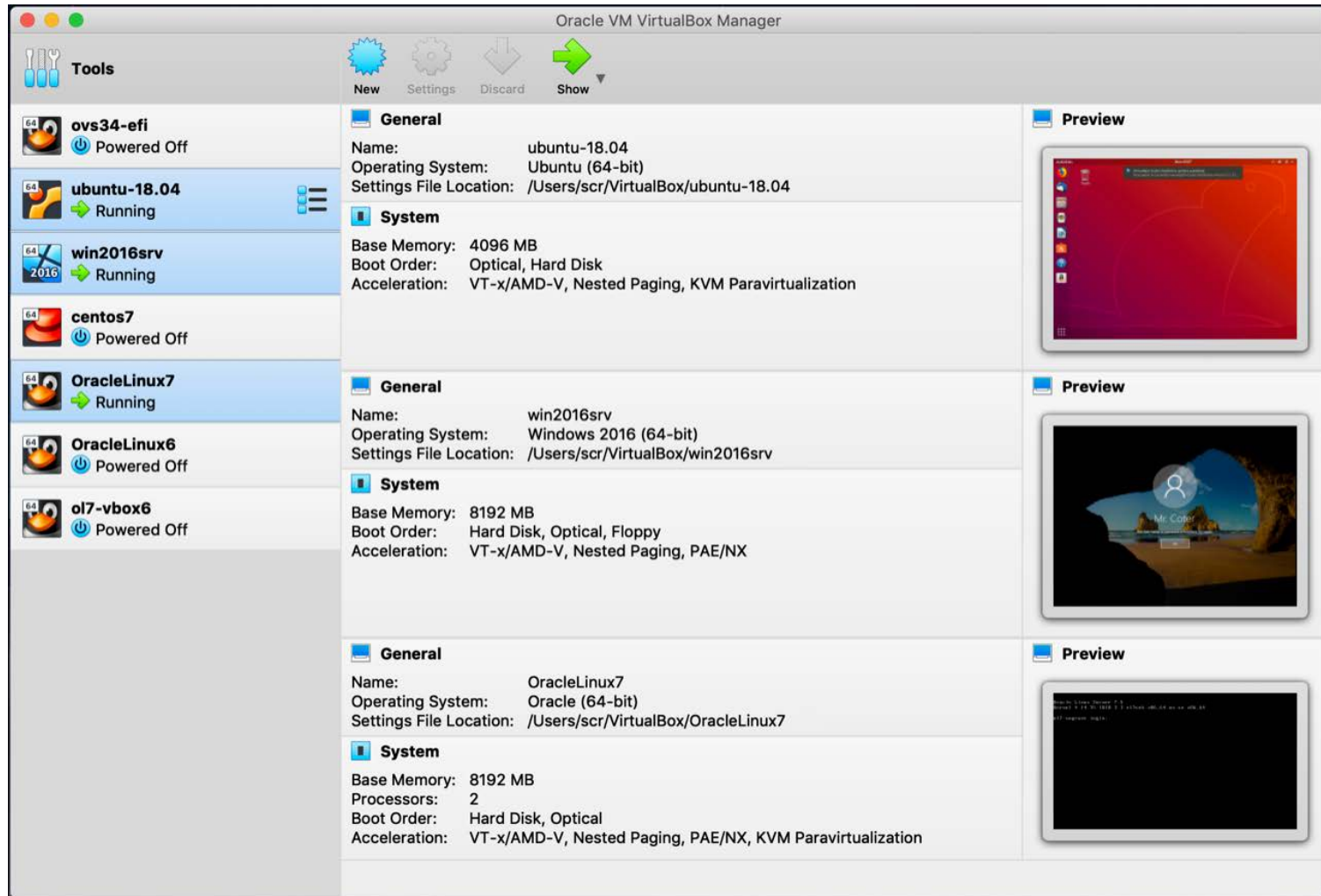
- terminology

- Host operating system (host OS)
 - This is the OS of the physical computer on which Oracle VM VirtualBox was installed
- Guest operating system (guest OS)
 - This is the OS that is running inside the virtual machine
 - run any x86 OS such as DOS, Windows, OS/2, FreeBSD, and OpenBSD.

- 설치 가능 OS

- Windows hosts (64-bit)
- Mac OS X hosts (64-bit)
- Linux hosts (64-bit) Includes the following:
 - Ubuntu 18.04 LTS, 19.03 and 19.10
 - Debian GNU/Linux 9 ("Stretch") and 10 ("Buster")
 - Oracle Linux 6, 7 and 8
 - CentOS/Red Hat Enterprise Linux 6, 7 and 8
 - Fedora 30 and 31
 - Gentoo Linux
 - SUSE Linux Enterprise server 12 and 15
 - openSUSE Leap 15.1
- Oracle Solaris hosts (64-bit only)

• 설치 사례



• 설치 요구사항

구분	매뉴얼상 최소 요구 사양	권장 사항 및 이 책의 실습 환경
CPU	64비트 x86 CPU, 1.3GHZ 이상	64비트를 지원하는 인텔 펜티엄 이상 또는 AMD 동급
RAM	2GB(4GB 이상 권장)	4GB 이상(8GB 권장)
HDD 여유 공간	프로그램 설치를 위해 150MB, 게스트 OS당 별도 공간 필요	게스트 OS당 1~10GB(이 책의 실습에서는 게스트 OS를 세 개 사용하므로 최소한 20~30GB의 여유 공간이 필요하며, HDD보다는 SSD를 권장)
화면 해상도	1024×768	1280×1024 이상 권장(게스트 OS는 1024×768 권장)
권장 호스트 OS	64비트 윈도우 7 이후의 모든 운영체제	64비트 윈도우 7 이후의 모든 운영체제

리눅스의 설치 방법

- 1. PC에 리눅스 단독 설치
 - 리눅스 전용으로 피시 혹은 노트북 사용
 - 저사양 기계를 리눅스 단독 설치하여 사용가능
 - 인터넷 연결시 연결 사이트에 따라 몇가지 문제점 있음.
- 2. Multi-Booting으로 설치
 - 먼저 Windows를 설치
 - 리눅스 설치 공간을 파티션하고 format은 하지 않음
 - 리눅스 설치 버전을 부팅 가능 USB로 만들거나 CD-ROM으로 미리 준비
 - 리눅스 설치로 부팅해서 리눅스 공간에 Swap, / , /home으로 파티션하고 설치
 - 장기적으로 사용시 추천
- 3. Windows 환경에서 가상머신으로 설치
 - 먼저 리눅스를 설치
 - Virtual Box 혹은 Vmware 설치
 - 가상머신 속에 리눅스 설치
 - 단기적으로 사용시 추천

리눅스 설치 (설치 방법 중 3번)

- 이번 학기 실습 환경은 설치 방법중 3번으로 실행
- 이미 윈도우가 설치된 상태에서 설치 가능
- 가능하면 VirtualBox를 설치하여 가상머신을 형성한 후
- 우분투 리눅스 데스크탑 최신 버전을 다운로드 받아서 설치
- VirtualBox 대신 VMware를 설치해도 됨
- 설치전 VirtualBox(혹은 Vmware)와 리눅스 데스크탑 버전 두가지를 먼저 다운로드 !!
- 가상머신 부터 설치
- 이후 가상머신에 우분투 리눅스 설치

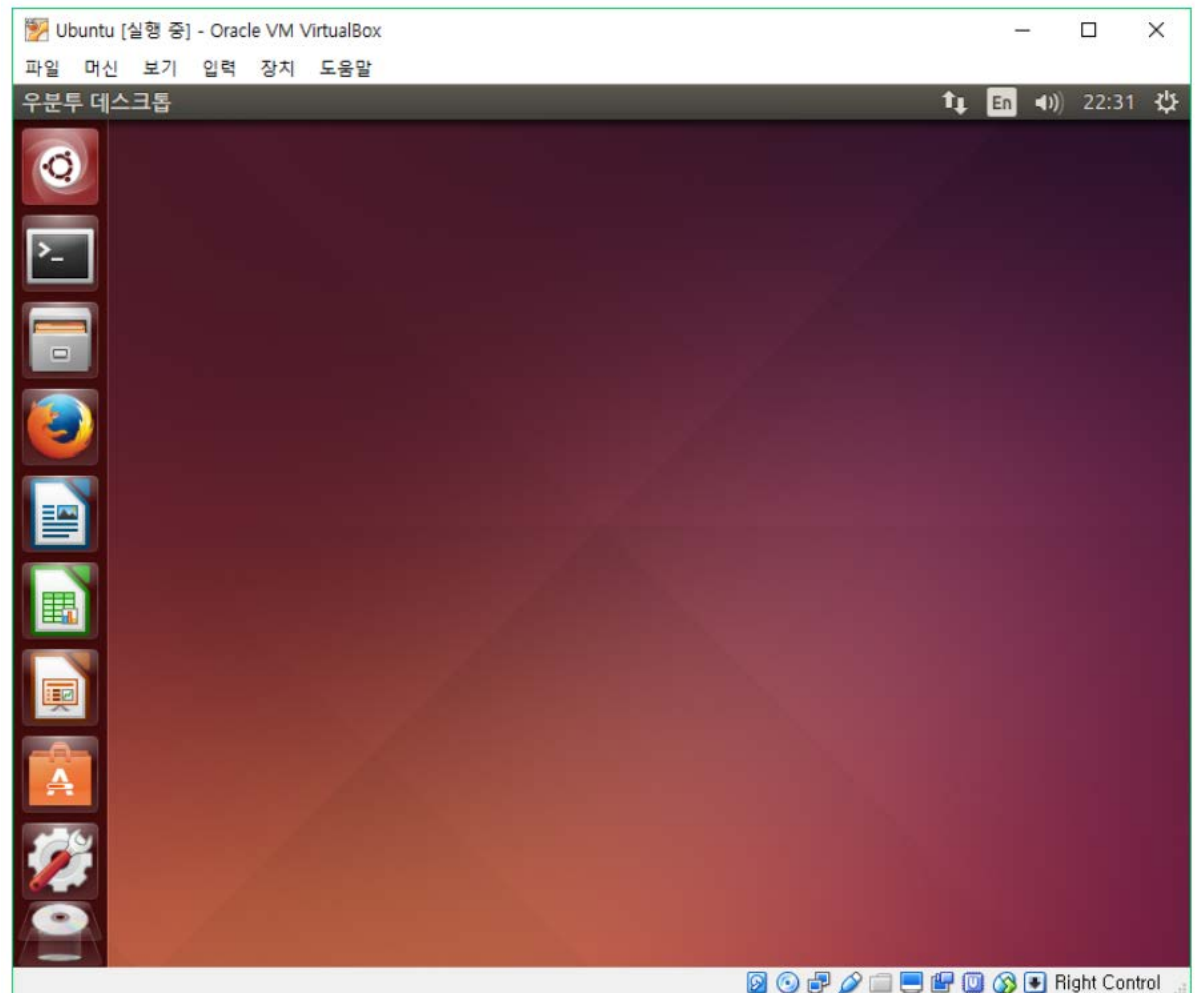
라떼 한잔하고 하는 잔소리...

- 이번 학기 지난 후 리눅스 명령어 사용법을 모른다면 이 과목을 수강하지 않은 것과 같습니다.
- 따라서 반드시 본인의 노트북/피시에 리눅스를 설치하고 동영상과 강의노트를 참고해서 반복적으로 리눅스 명령어를 사용해야 합니다.
- 충분히 연습한다면 평생 잊지 않고 언제든지 리눅스 혹은 유닉스 OS를 접한다면 명령어를 사용할 수 있습니다. 성공한 미래에 꼭 확인 ~~
- 이 강좌를 핸드폰 혹은 피시방에서 수강한다면 아무것도 남는 것 없이 한 학기를 보내게 됩니다.
- 핸드폰으로 강의 동영상 시청하고 출석만 하는 것은 절대 하지 말아야 합니다.
 - 그런 생각이면 이 강좌를 취소하고 다른 과목 수강하는 것을 **강추 !**
- 만약 이번학기 대면강의가 이루어 진다면 사용중인 노트북 혹은 집안 어딘가 사용하지 않고 버려둔 노트북에 리눅스를 설치하기 바랍니다
 - 혹은 라즈베리파이를 구하기 바랍니다.

학습목표

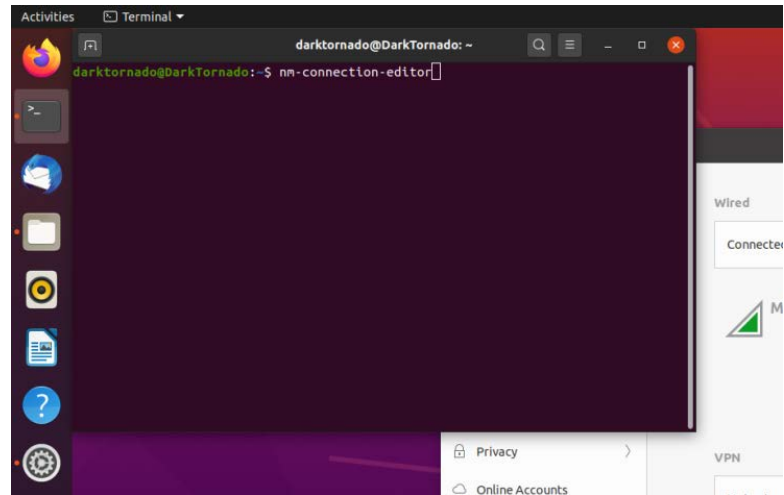
- Linux Desktop Tour
- Terminal Window
- Home Directory
- Shell 기능
- Command format
- File Commands

- 우분투 데스크탑 화면
- 자세한 내용은 동영상 강의 참조



Terminal Window

- 기능
 - 사용자-리눅스 사이 셸을 통하여 명령어 입력
 - 셸에서 명령어 결과 출력
- 각 터미널 창에는 셸(Shell)이 동작
- Enter 키를 입력할 때 마다 한 라인씩 보이는 것은 프롬프트(prompt)라고 한다
- 프롬프트는 셸이 정상 동작 하고 있고, 명령어 입력을 대기하고 있다는 표시
- 프롬프트의 형태는 사용자가 만들수 있고, 셸마다 약간 다르다



Home Directory

- 홈디렉토리는 사용자마다 login ID로 만들어 진다
- 위치는 /home 아래에 디렉토리로 만들어 진다
- 로그인 할 때 자동으로 홈디렉토리에 위치하게 된다
- 사용자는 홈디렉토리에 또 다른 디렉토리 혹은 파일을 둔다
- 자동으로 만들어지는 디렉토리/파일이 존재한다
- 숨겨진 디렉토리/파일도 존재하며, 대부분 사용자 환경 설정파일이다

Shell

- 셸은 사용자와 리눅스 사이에 존재하고, 사용자 인터페이스 역할
- 셸은 명령어를 입력 받고, 명령어를 실행 시키고, 명령어의 결과를 보여준다
- 셸은 사용자가 선택해서 사용할 수 있다
- 셸의 종류
 - sh(Bourne Shell), csh, ksh, bash(Born/Bourne Again Shell), ...
- 셸에서 다른 셸을 동작 시킬 수 있다
- 셸에서는 오직 한 명령어의 프로세스 만 전면(background)에 존재
- 나머지 프로세스는 후면(background)에 존재
- 셸 명령어 중에는 다른 창으로 열리는 명령어도 있다

Command format

- 명령어 형식

command p1 p2 p3 ...

- command(줄여서 cmd)

- 명령어 이름

- 반드시 실행 파일이어야 한다

- 실행파일 종류: 컴파일된 실행파일, 쉘스크립트 실행파일, 링크된 다른 실행파일

- p1, p2, p3,...

- 파라미터 라고 한다

- 명령어 실행에 따른 옵션 혹은 argument

- -p : 옵션 파라미터, 입력하면 옵션에 따라 명령어 실행

- - 기호가 붙지 않은 것은 argument, 예를 들면 디렉토리/파일 이름

- -- 기호는 argument를 동반하지 않는 옵션 파라미터, 예를 들면 --help

File 관련 명령어

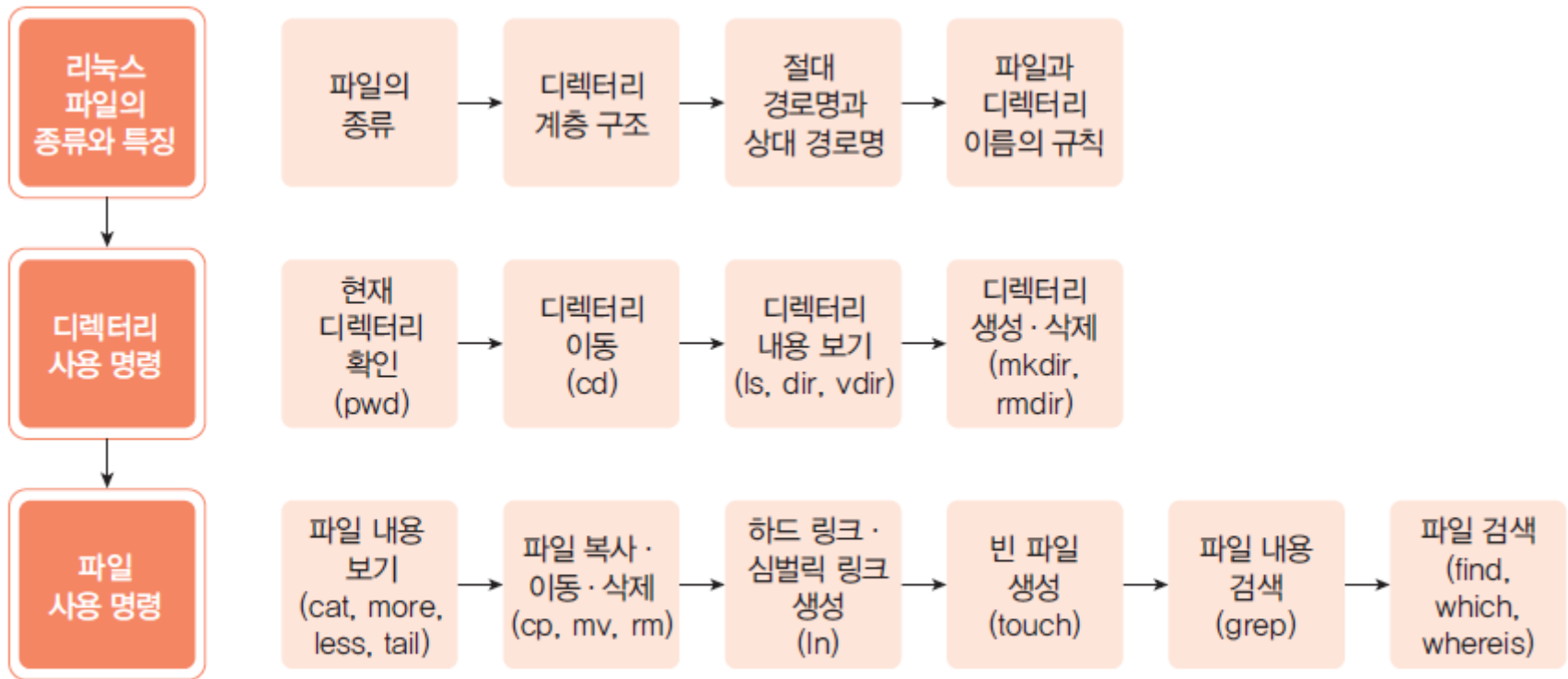


그림 2-1 2장의 내용 구성

- 디렉터리 계층 구조

- 리눅스에서는 파일을 효율적으로 관리하기 위해 디렉터리를 계층적으로 구성 -> 트리(tree) 구조
- 모든 디렉터리의 출발점은 루트(root, 뿌리) 디렉터리이며, /(빗금)으로 표시
- 하위 디렉터리(서브 디렉터리): 디렉터리 아래에 있는 디렉터리 (bin, etc, usr, home, boot)
- 상위 디렉터리(부모 디렉터리): '..'으로 표시
- 루트 디렉터리를 제외하고 모든 디렉터리에는 부모 디렉터리가 있음

- 파일 시스템 계층 구조

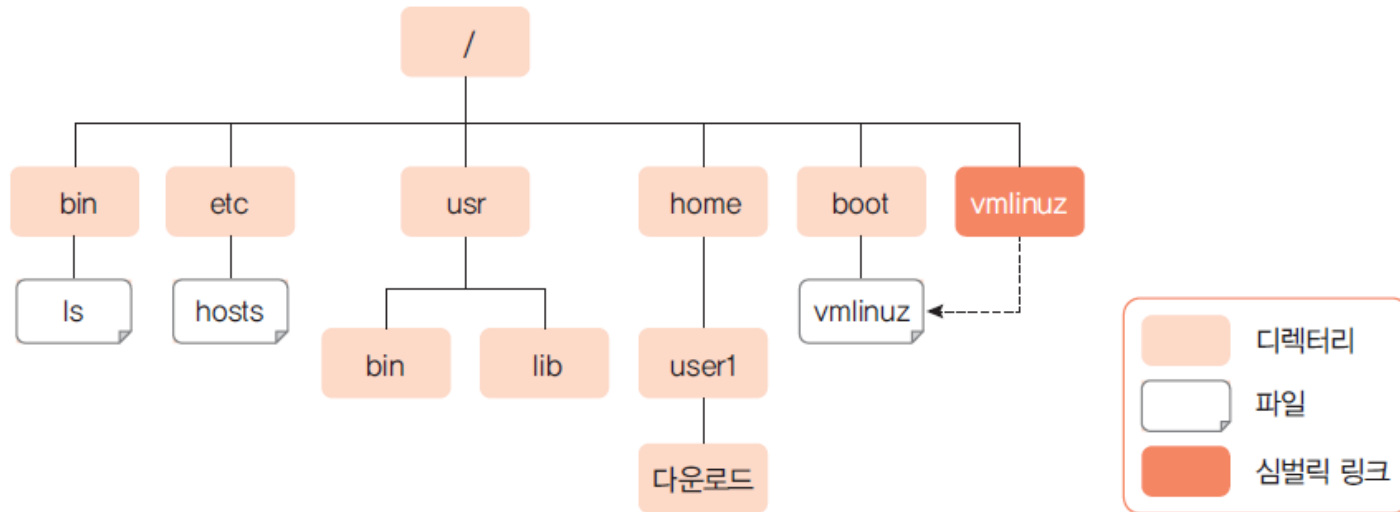


그림 2-2 디렉터리 계층 구조의 예

시스템 디렉토리의 기능

표 2-1 디렉토리의 주요 기능

디렉터리	기능
dev	장치 파일이 담긴 디렉터리이다.
home	사용자 홈 디렉터리가 생성되는 디렉터리이다.
media	CD-ROM이나 USB 같은 외부 장치를 연결(마운트라고 함)하는 디렉터리이다.
opt	추가 패키지가 설치되는 디렉터리이다.
root	root 계정의 홈 디렉터리이다. 루트(/) 디렉터리와 다른 것이므로 혼동하지 않도록 한다.
sys	리눅스 커널과 관련된 파일이 있는 디렉터리이다.
usr	기본 실행 파일과 라이브러리 파일, 헤더 파일 등 많은 파일이 있다. 참고로 usr는 'Unix System Resource'의 약자이다.
boot	부팅에 필요한 커널 파일을 가지고 있다.
etc	리눅스 설정을 위한 각종 파일을 가지고 있다.
lost+found	파일 시스템에 문제가 발생하여 복구할 경우, 문제가 되는 파일이 저장되는 디렉터리로 보통은 비어 있다.
mnt	파일 시스템을 임시로 마운트하는 디렉터리이다.
proc	프로세스 정보 등 커널 관련 정보가 저장되는 디렉터리이다.
run	실행 중인 서비스와 관련된 파일이 저장된다.
srv	FTP나 Web 등 시스템에서 제공하는 서비스의 데이터가 저장된다.
tmp	시스템 사용 중에 발생하는 임시 데이터가 저장된다. 이 디렉터리에 있는 파일은 재시작하면 모두 삭제된다.
var	시스템 운영 중에 발생하는 데이터나 로그 등 내용이 자주 바뀌는 파일이 주로 저장된다.

- 작업 디렉터리

- 현재 사용 중인 디렉터를 작업 디렉터리(working directory) 또는 현재 디렉터리(current directory)라고 함
- 현재 디렉터리는 '.' 기호로 표시
- 현재 디렉터리의 위치는 pwd 명령으로 확인

- 홈 디렉터리

- 각 사용자에게 할당된 디렉터리로 처음 사용자 계정을 만들 때 지정
- 사용자는 자신의 홈 디렉터리 아래에 파일이나 서브 디렉터를 생성하며 작업 가능
- 홈 디렉터리는 '~' 기호로 표시 : ~user1
- 홈디렉토리로 바로 가기
 - cd /home/user, cd ~home, cd \$HOME, cd

• 경로명

- 파일 시스템에서 디렉터리 계층 구조에 있는 특정 파일이나 디렉터리의 위치 표시
- 경로명에서 각 경로를 구분하는 구분자로 /을 사용
- 경로명에서 가장 앞에 있는 /은 루트 디렉터를 뜻하지만 경로명 중간에 있는 /은 구분자
- 예: bin/ls에서 맨 앞의 /은 루트 디렉터를 의미하고, 중간에 있는 /은 디렉터리 이름과 파일명을 구분하는 구분자

• 절대 경로명

- 항상 루트 디렉터리부터 시작
- 반드시 /으로 시작한다.
- 루트 디렉터리부터 시작하여 특정 파일이나 디렉터리의 위치까지 이동하면서 거치게 되는 모든 중간 디렉터리의 이름을 표시
- 특정 위치를 가리키는 절대 경로명은 항상 동일

• 상대 경로명

- 현재 디렉터를 기준으로 시작
- / 이외의 문자로 시작
- 현재 디렉터를 기준으로 서브 디렉터리로 내려가면 그냥 서브 디렉터리명을 추가
- 현재 디렉터를 기준으로 상위 디렉터리로 가려면 ..(마침표 두 개)를 추가
- 상대 경로명은 현재 디렉터리가 어디냐에 따라 달라짐

- 파일과 디렉터리 이름 규칙

- 파일과 디렉터리 이름에는 /을 사용할 수 없다. /은 경로명에서 구분자로 사용하기 때문이다.
- 파일과 디렉터리의 이름에는 알파벳, 숫자, 붙임표(-), 밑줄(_), 마침표(.)만 사용한다.
- 파일과 디렉터리의 이름에는 공백문자, *, |, ", ', @, #, \$, %, ^, & 등을 사용하면 안 된다.
- 파일과 디렉터리 이름의 영문은 대문자와 소문자를 구별하여 다른 글자로 취급한다.
- 파일과 디렉터리의 이름이 .(마침표)로 시작하면 숨김 파일로 간주한다.

- 파일 이름 예

- 좋은 이름: game.txt, hello.c, test, sample11
- 나쁜 이름: &game, *dir, my home, game₩
- 사용할 수 없는 이름: myhome/, /test, bad/name

File Commands

- cd
- ls, dir, vdir
- mv
- rm
- touch
- mkdir
- rmdir
- pwd
- file

표 2-2 파일의 상세 정보

필드 번호	필드 값	의미
1	d	파일 종류
2	rw-r--r--	파일 접근 권한. 파일 소유자, 그룹, 기타 사용자가 파일을 읽고 수정하고 실행할 수 있는 권한이 어떻게 부여되어 있는지를 보여준다.
3	2	하드 링크의 개수
4	user1	파일 소유자
5	user1	파일이 속한 그룹
6	4096	파일 크기(바이트 단위)
7	11월 8 23:24	파일이 마지막으로 수정된 시간
8	공개	파일명

표 2-3 파일의 종류

문자	파일 유형
-	일반(정규) 파일
d	디렉터리 파일
l	심벌릭 링크 파일
b	블록 단위로 읽고 쓰는 블록 장치 파일
c	섹터 단위로 읽고 쓰는 문자 장치 파일
p	파이프 파일(프로세스 간 통신에 사용되는 특수 파일)
s	소켓(네트워크 통신에 사용되는 특수 파일)

Question?