

# 리눅스시스템 및 응용

Week 2

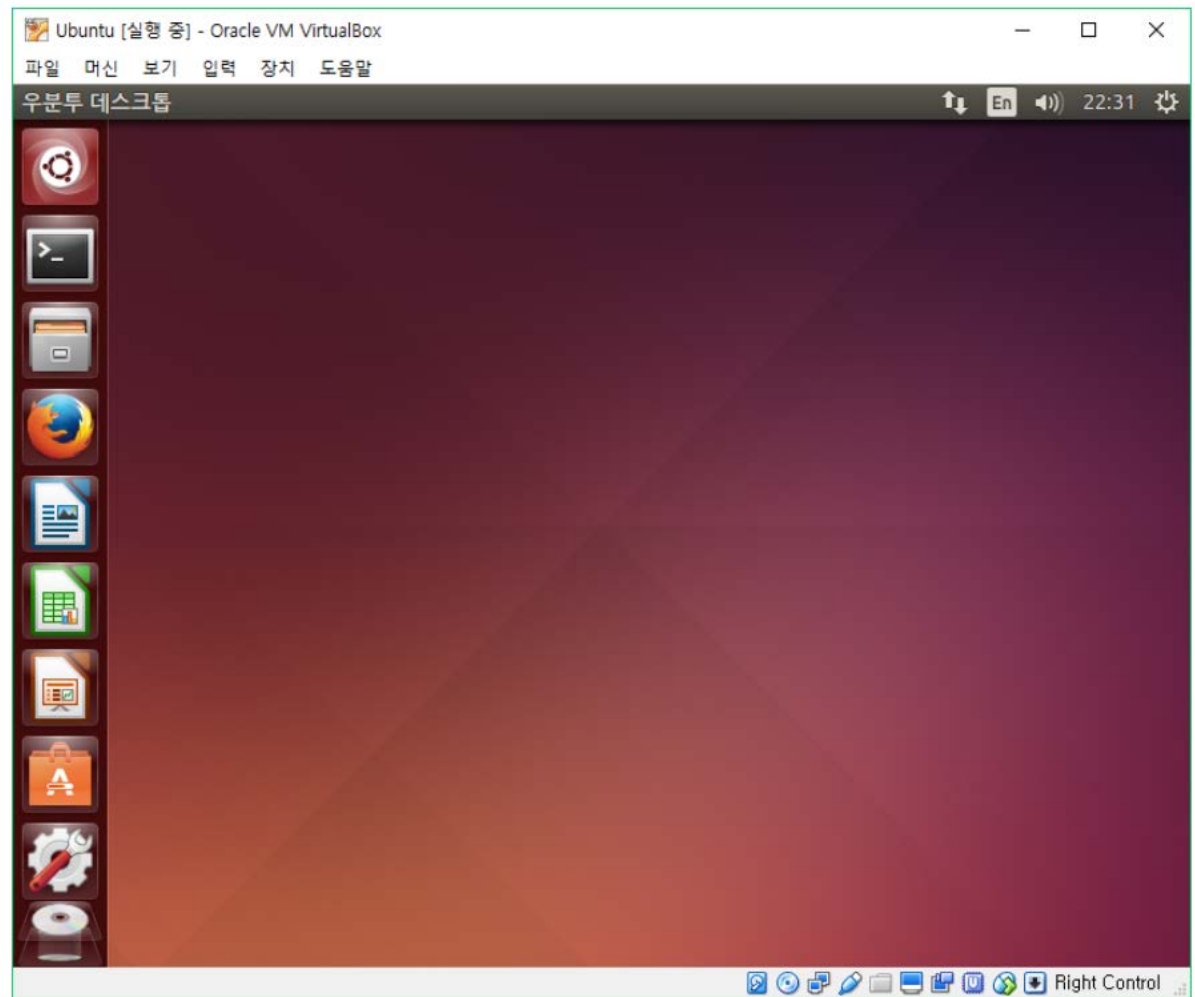
**Linux Desktop**  
**File Command**

# 학습목표

---

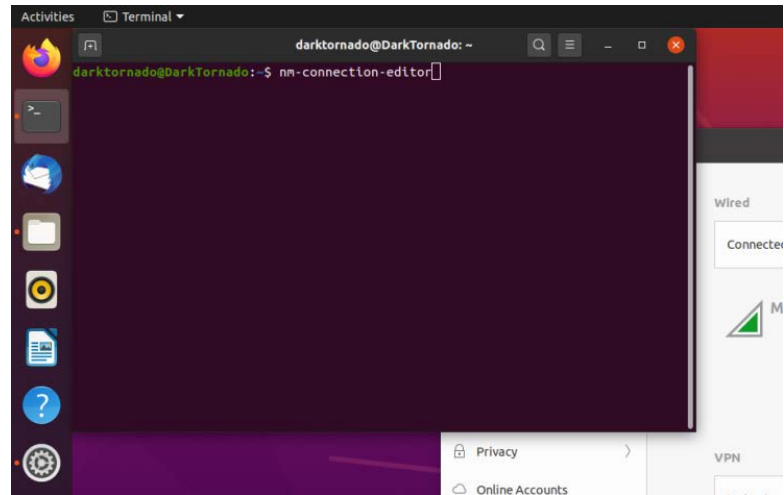
- Linux Desktop Tour
- Terminal Window
- Home Directory
- Shell 기능
- Command format
- File Commands

- 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(foreground)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

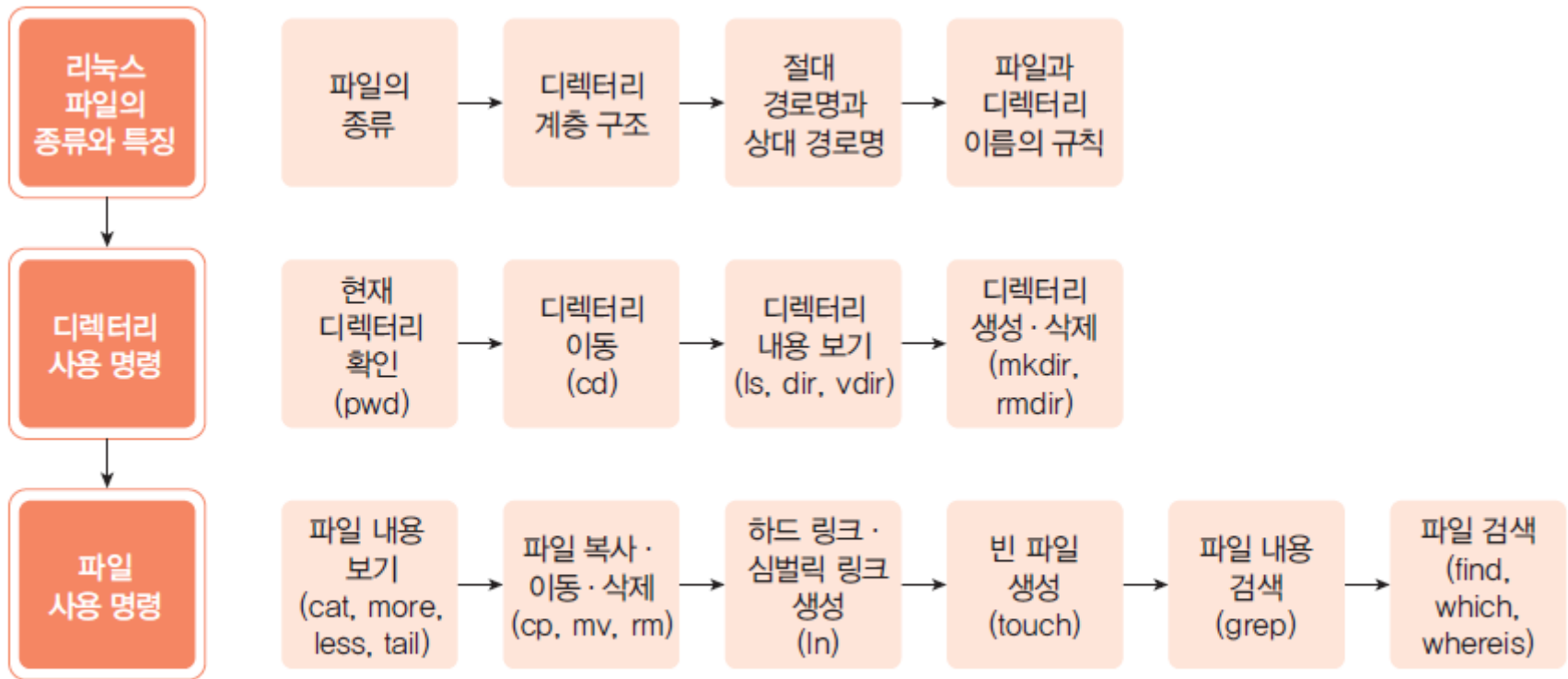


그림 2-1 2장의 내용 구성



---

- 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

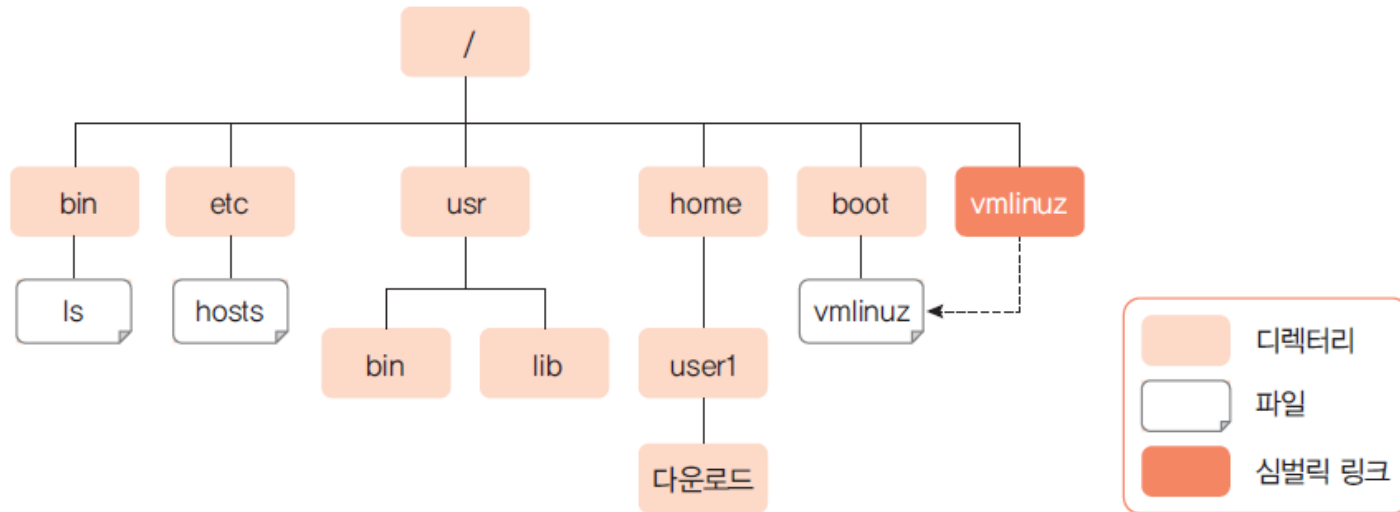


그림 2-2 디렉터리 계층 구조의 예

# 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	rwxr-xr-x	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?**