



# Linux On Azure

Dokyun Kim  
MCT, MVP (Developer Technologies)



## 오늘 배울 내용

1. 리눅스 기초
2. 리눅스 관리 핵심 명령
3. 리눅스에서 Azure 다루기

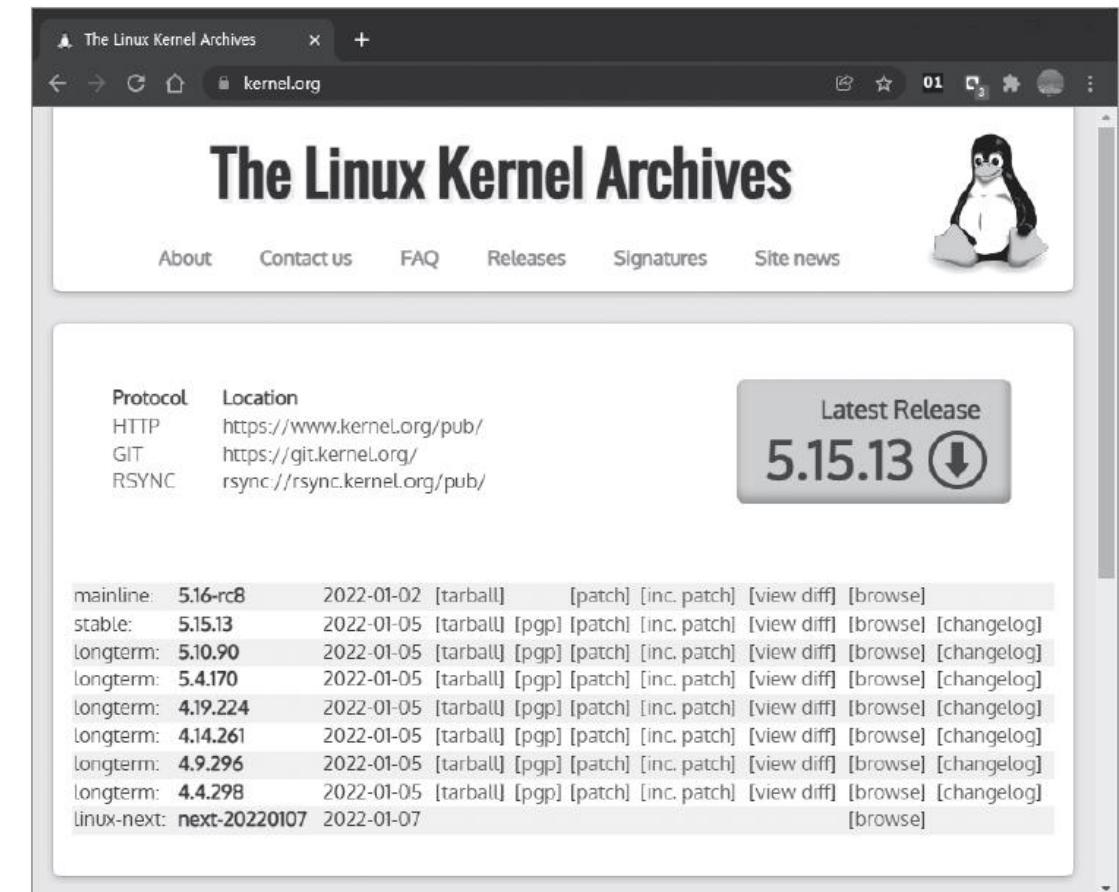
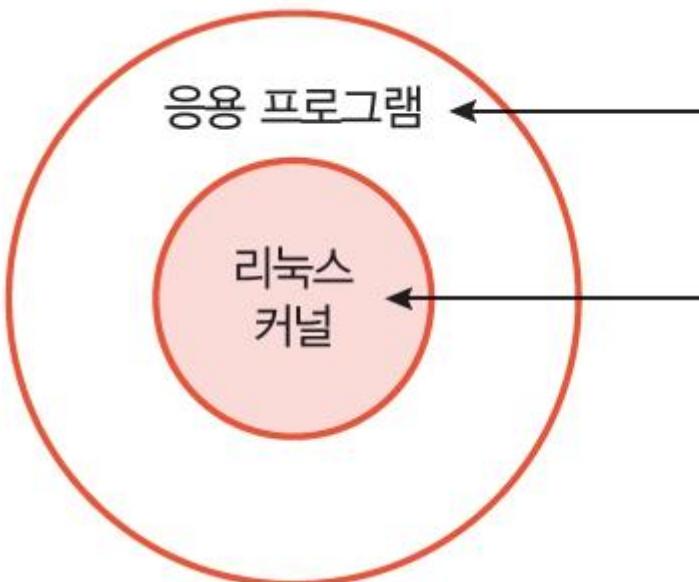
# 리눅스 VM 배포 및 연결

작업 순서	설정 항목	리눅스 VM
① Azure Portal 로그인	리소스 그룹	RG##
② 리눅스 VM 배포	가상 머신 이름	vmlinux##
③ SSH를 사용한 연결	지역	한국 중부
	이미지	Ubuntu Server 22.04 LTS
	크기	Standard_B2s
	사용자 이름	Tony
	인증 방식	암호: Pa55w.rd1234
	공용 인바운드 포트	SSH (22), 80
	OS 디스크 유형	프리미엄 SSD
	가상 네트워크	vnet-msaischool
	서브넷	snet-linux
	부트 진단	사용 안 함
	확장	N/A

# 리눅스 기본 사항

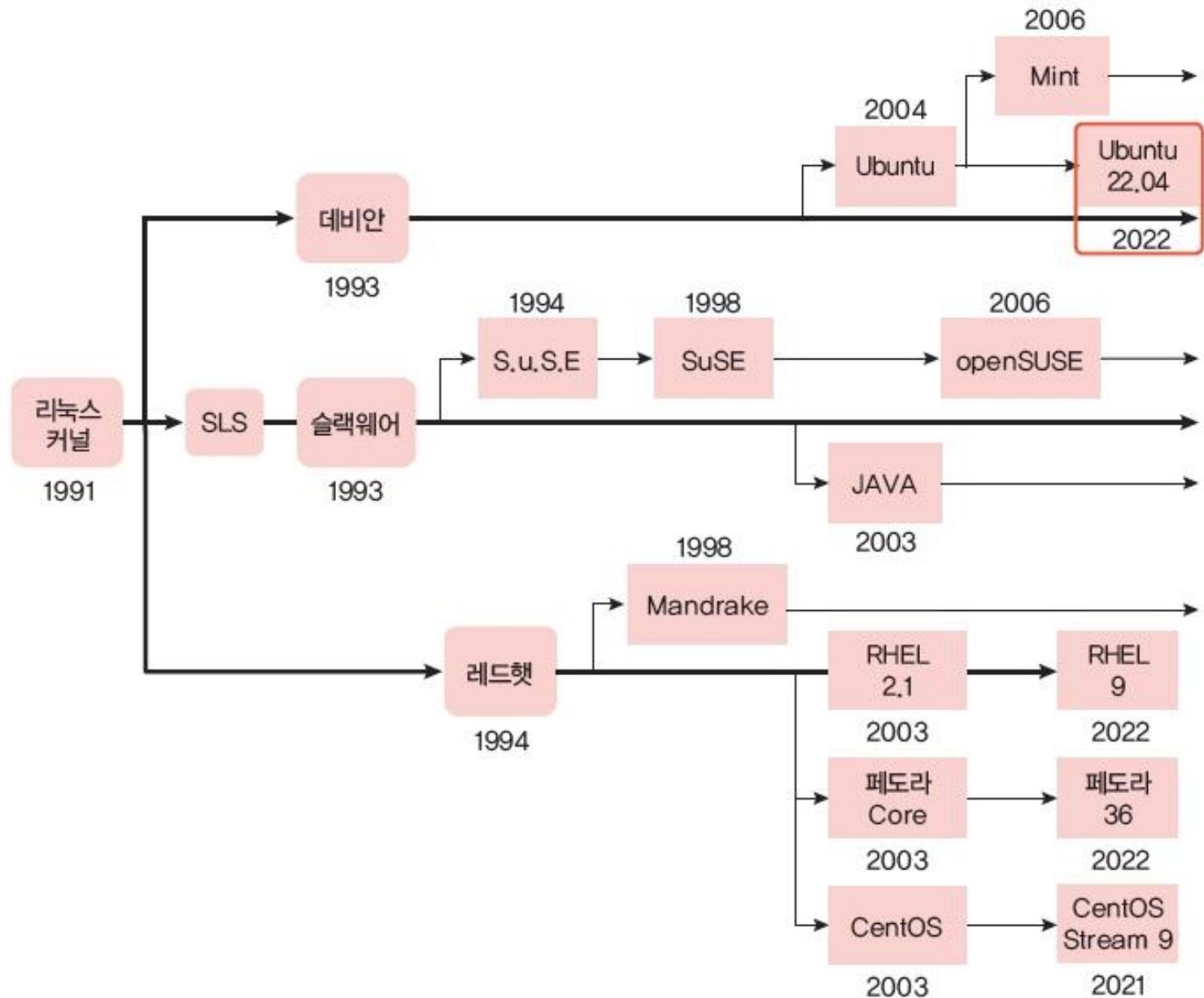
- 리눅스는 유닉스와의 호환성을 제공하는 오픈 소스로 시작
- 임베디드 시스템, 모바일 기기 등 다양한 환경에서 사용
- 리눅스의 시작과 발전
  - ✓ 핀란드의 리누스 베데딕트 토팔스
  - ✓ 미닉스라는 교육용 운영체제 참조
  - ✓ 1991년 8월 26일 리눅스 탄생일, 2022년은 리눅스 탄생 31주년

# 리눅스 커널과 응용 프로그램



# 리눅스 배포판

- 리눅스 커널과 응용 프로그램을 적절하게 구성한 버전
- 누구나 본인이 원하는 대로 구성하여 배포 가능



# Ubuntu 리눅스의 버전

- 배포판이 나온 연도와 월로 구성(예: 22.04 – 2022년 4월 버전)
- 버전에 따라 코드명 부여(예: 21.10 – Impish Indri, 장난꾸러기 여우원숭이)
- 일반 버전과 장기 지원 버전
  - ✓ 장기 지원 버전은 4번에 한 번씩 발표하고 5년간 지원
  - ✓ 일반 버전은 4월, 10월에 발표하고 9개월간 지원

20.04 (LTS)	Focal Fossa (중심의 포사)	5.4.0	2020년 04월 23일	~ 2025년 04월
20.10	Groovy Gorilla (멋진 고릴라)	5.8.0	2020년 10월 22일	~ 2021년 07월
21.04	Hirsute Hippo (털이 많은 하마)	5.11.0	2021년 04월 22일	~ 2022년 01월
21.10	Impish Indri (장난꾸러기 여우원숭이)	5.13.0	2021년 10월 14일	~ 2022년 07월

# 기초 명령 사용법 - 명령행 편집

- 문자 지우기: 나 를 사용
- 단어 지우기: **ctrl+w**

```
user1@myubuntu:~$ linux Ubuntu → Ubuntu를 지우려면 +w를 누른다.  
user1@myubuntu:~$ linux 
```

- 행 지우기: **ctrl+u**

```
user1@myubuntu:~$ linux Ubuntu → 모두 지우려면 +u를 누른다.  
user1@myubuntu:~$ 
```

# 기초 명령 사용법 – 명령의 구조와 사용

## 명령의 구조

- **형식** 명령 [옵션] [인자]
- **명령** 리눅스를 사용하기 위해 사용자가 입력하는 명령에는 date, man, ls, cp, mv 등 수백 가지가 있다.
- **옵션** 옵션을 사용하여 명령의 세부 기능을 선택할 수 있다. 리눅스의 기능을 풍부하게 하는 중요한 특징으로, 명령에 따라 옵션이 있는 경우도 있고 없는 경우도 있다. 옵션은 - 나 -- 기호로 시작하며 영문 소문자나 대문자로 구성된다. 명령에 따라 어떤 옵션이 있고 그 기능이 무엇인지는 해당 명령의 사용법을 참조해야 한다.
- **인자** 인자는 명령으로 전달되는 값이며 주로 파일명이나 디렉터리명이 사용된다. 명령에 따라 인자가 없을 수도 있고, 각 명령에 필요한 인자도 각기 다르므로 사용법을 참조해야 한다.

# 기초 명령 사용법 - 명령과 옵션

- 명령만 사용한 경우

```
user1@myubuntu:~$ ls  
공개  다운로드  문서  바탕 화면  비디오  사진  음악  템플릿
```

- 명령과 옵션을 사용한 경우

```
user1@myubuntu:~$ ls -a  
.  .bash_logout  .cache  .local  공개  문서  비디오  음악  
..  .bashrc       .config  .profile  다운로드  바탕화면  사진  템플릿
```

# 기초 명령 사용법 - 명령과 옵션, 인자

- 명령과 인자를 사용한 경우

```
user1@myubuntu:~$ ls /tmp
VMwareDnD
dbus-CIeUbrtCOW
snap.snap-store
(생략)
```

- 명령, 옵션, 인자를 사용한 경우

```
user1@myubuntu:~$ ls -a /tmp
.
..
.ICE-unix
.Test-unix
.X0-lock
.X1-lock
.X1024-lock
.X1025-lock
.X11-unix
(중략)
```

# 기초 명령 1

- date 명령

## date

- **기능** 날짜와 시간을 출력한다.
- **형식** date

```
user1@myubuntu:~$ date  
2022. 01. 10. (월) 16:44:39 KST
```

- clear 명령

## clear

- **기능** 화면을 지운다.
- **형식** clear

```
user1@myubuntu:~$ clear
```

# 기초 명령 2

## ■ man 명령

man

- **기능** 명령 사용 방법을 화면에 출력한다.
- **형식** man 명령

```
user1@myubuntu:~$ man clear
```

clear(1)

General Commands Manual

clear(1)

NAME

→ 명령에 의한 간략한 설명

clear - clear the terminal screen

SYNOPSIS

→ 명령의 사용 방법 요약

clear [-Ttype] [-V] [-x]

DESCRIPTION

→ 명령에 대한 상세한 기술

clear clears your screen if this is possible, including its scrollback

# 기초 명령 3

## ■ passwd 명령

### passwd

- **기능** 사용자 계정의 비밀번호를 변경한다.
- **형식** passwd [인자]

```
user1@myubuntu:~$ passwd
user1에 대한 암호 변경 중
현재 비밀번호:
새 암호:
새 암호 재입력:
passwd: 암호를 성공적으로 업데이트했습니다
```

# 셸 기본 사용법 - 셸의 종류

- 본셸(sh): 최초의 셸, 유닉스 v7에서 처음 등장, 현재는 배시셸 등 다른 셸로 대체
- C셸(csh): 캘리포니아대학교에서 빌 조이가 개발, 2BSD 유닉스에서 발표
  - ✓ 앤리어스, 히스토리 기능 포함, 셸 스크립트 구문 형식이 C언어와 동일
- 콘셸(ksh): 1980년대 중반 AT&T 벨연구소에서 개발, SVR4 유닉스에서 발표
  - ✓ 본셸과 호환성 유지, 앤리어스, 히스토리 기능 제공
- 배시셸(bash): 1988년 브레인 폭스가 개발
  - ✓ 본셸과 호환성 유지, C셸/콘셸의 편리한 기능 모두 포함
  - ✓ 리눅스의 기본 셸로 제공
- 대시셸(dash): 1997년 허버스 슈가 리눅스에 이식
  - ✓ 본셸을 기반으로 개발, POSIX 표준 준수하며 작은 크기로 개발
  - ✓ 우분투는 6.10버전부터 본셸 대신 대시셸을 사용

# 셸 기본 사용법 - 셸 지정 및 변경

- 사용자의 기본 셸은 /etc/passwd 파일에 저장

```
user1@myubuntu:~$ grep user1 /etc/passwd  
user1:x:1000:1000:user1,,,:/home/user1:/bin/bash
```

- 기본 셸 변경하기

## chsh

- 기능** 사용자 로그인 셸을 바꾼다.
- 형식** chsh [옵션] [사용자명]
- 옵션** -s shell: 지정하는 셸(절대 경로)로 로그인 셸을 바꾼다.  
-l: /etc/shells 파일에 지정된 셸을 출력한다.
- 사용 예** chsh -l  
chsh -s /bin/sh user1  
chsh

```
user1@myubuntu:~$ chsh -s /bin/sh user1
```

암호:

```
user1@myubuntu:~$
```



```
user1@myubuntu:~$ grep user1 /etc/passwd  
user1:x:1000:1000:user1:/home/user1:/bin/sh
```

# 셸 기본 사용법 - 출력 명령

## echo

- **기능** 화면에 한 줄의 문자열을 출력한다.
- **형식** echo [-n] [문자열]
- **옵션** -n: 마지막에 줄 바꿈을 하지 않는다.
- **사용 예**

```
echo text
echo -n text
```

## printf

- **기능** 자료를 형식화하여 화면에 출력한다.
- **형식** printf [형식] [인수]
- **옵션** 형식: %d, \n 등 C 언어의 printf 함수의 형식을 지정한다.
- **사용 예**

```
printf text
printf "text\n"
printf "%d\n" 100
```

```
user1@myubuntu:~$ echo linux
linux
user1@myubuntu:~$ echo "linux ubuntu"
linux ubuntu
```

```
user1@myubuntu:~$ printf linux
linux
user1@myubuntu:~$ printf "linux ubuntu \n"
linux ubuntu
user1@myubuntu:~$ printf "%d + %d = %d\n" 10 10 20
10 + 10 = 20
```

# 셸 기본 사용법 - 특수 문자 \*

사용 예	의미
ls *	현재 디렉터리의 모든 파일과 서브 디렉터리를 나열한다. 서브 디렉터리의 내용도 출력한다.
cp * /tmp	현재 디렉터리의 모든 파일을 /tmp 디렉터리 아래로 복사한다.
ls -F t*	t, tmp, temp와 같이 파일명이 t로 시작하는 모든 파일의 이름과 파일 종류를 출력한다. t도 해당 한다는 데 주의한다.
cp *.txt .. /ch3	확장자가 txt인 모든 파일을 상위 디렉터리 아래의 ch3 디렉터리로 복사한다.
ls -l h*d	파일명이 h로 시작하고 d로 끝나는 모든 파일의 상세 정보를 출력한다. hd, had, hard, h12345d 등이 조건에 맞는 모든 파일의 정보를 볼 수 있다.

# 셸 기본 사용법 - 특수 문자 ?와 []

사용 예	의미
ls t?.txt	t 다음에 임의의 한 문자가 오고 파일의 확장자가 txt인 모든 파일의 이름을 출력한다. t1.txt, t2.txt, ta.txt 등이 해당된다. 단, t.txt는 제외한다.
ls -l tmp[135].txt	tmp 다음에 1, 3, 5 중 하나가 오고 파일의 확장자가 txt인 모든 파일의 이름을 출력한다. tmp1.txt, tmp3.txt, tmp5.txt 파일이 있으면 해당 파일의 상세 정보를 출력한다. 단, tmp.txt는 제외 한다.
ls -l tmp[1-3].txt	[1-3]은 1부터 3까지의 범위를 의미한다. 따라서 ls -l tmp[123].txt와 결과가 같다. 즉 tmp1.txt, tmp2.txt, tmp3.txt 파일이 있으면 해당 파일의 상세 정보를 출력한다.
ls [0-9]*	파일명이 숫자로 시작하는 모든 파일의 목록을 출력한다.
ls [A-Za-z]*[0-9]	파일명이 영문자로 시작하고 숫자로 끝나는 모든 파일의 목록을 출력한다.

# 셸 기본 사용법 - 특수 문자 ~와 -, ;과 |

사용 예	의미
cp *.txt ~/ch3	확장자가 txt인 모든 파일을 현재 작업 중인 사용자의 홈 디렉터리 아래 tmp 디렉터리로 복사한다.
cp ~user2/linux.txt .	user2라는 사용자의 홈 디렉터리 아래에서 linux.txt 파일을 찾아 현재 디렉터리로 복사한다.
cd -	이전 작업 디렉터리로 이동한다.

사용 예	의미
date ; ls ; pwd	왼쪽부터 차례대로 명령을 실행한다. 즉 날짜를 출력한 후 현재 디렉터리의 파일 목록을 출력하고, 마지막으로 현재 작업 디렉터리의 절대 경로를 보여준다.
ls -al   more	루트 디렉터리에 있는 모든 파일의 상세 정보를 한 화면씩 출력한다. ls -al / 명령의 결과가 more 명령의 입력으로 전달되어 페이지 단위로 출력되는 것이다.

# 셸 기본 사용법 - 특수 문자 ''와 "", ``

사용 예	의미
echo '\$SHELL'	\$SHELL 문자열이 화면에 출력된다.
echo "\$SHELL"	셸 환경 변수인 SHELL에 저장된 값인 현재 셸의 종류가 화면에 출력된다. /bin/bash를 예로 들 수 있다.

사용 예	의미
echo "Today is `date`"	'date'가 명령으로 해석되어 date 명령의 실행 결과로 바뀐다. 결과적으로 다음과 같이 출력된다. Today is 2021. 12. 12. (일) 15:42:40 KST
ls /usr/bin/`uname -m`	uname -m 명령의 실행 결과를 문자열로 바꾸어 파일 이름으로 사용한다.

# 셸 기본 사용법 - 특수 문자 \ 와 >, <, >>

사용 예	의미
ls -l t\*	t*라는 이름을 가진 파일의 상세 정보를 출력한다. \ 없이 t*를 사용하면 t로 시작하는 모든 파일의 상세 정보를 출력한다.
echo \\$SHELL	\$SHELL을 화면에 출력한다. echo '\$SHELL'과 결과가 같다.

사용 예	의미
ls -l > res	ls -l 명령의 실행 결과를 화면이 아닌 res 파일에 저장한다.
ls -l >> res	ls -l의 명령의 결과를 res파일의 끝부분에 추가한다.
cat < text	cat 명령의 입력을 text 파일에서 받는다.

# 셸 기본 사용법 - 변수 정의

## 셸 변수 정의

- **형식** 변수명=문자열
- **사용 예** SOME=test

```
user1@myubuntu:~/linux_ex/ch4$ SOME=test  
user1@myubuntu:~/linux_ex/ch4$ echo $SOME  
test
```

# 셸 기본 사용법 - 별칭

## alias

- **기능** 앤리어스를 생성한다.
- **형식** alias 이름= '명령'
- **사용 예** alias: 현재 설정된 별칭 목록을 출력한다.  
alias 이름= '명령': 명령을 수정해 사용하는 경우다.  
alias 이름= '명령;명령2;...': 여러 명령을 하나의 이름으로 사용하는 경우다.

```
alias l='ls -CF'  
alias la='ls -A'  
alias ll='ls -alF'  
alias ls='ls --color=auto'
```

## unalias

- **기능** 앤리어스를 삭제한다.
- **형식** unalias 앤리어스

# 셸 기본 사용법 - 히스토리

## history

- **기능** 히스토리(명령 입력 기록)를 출력한다.
- **형식** history

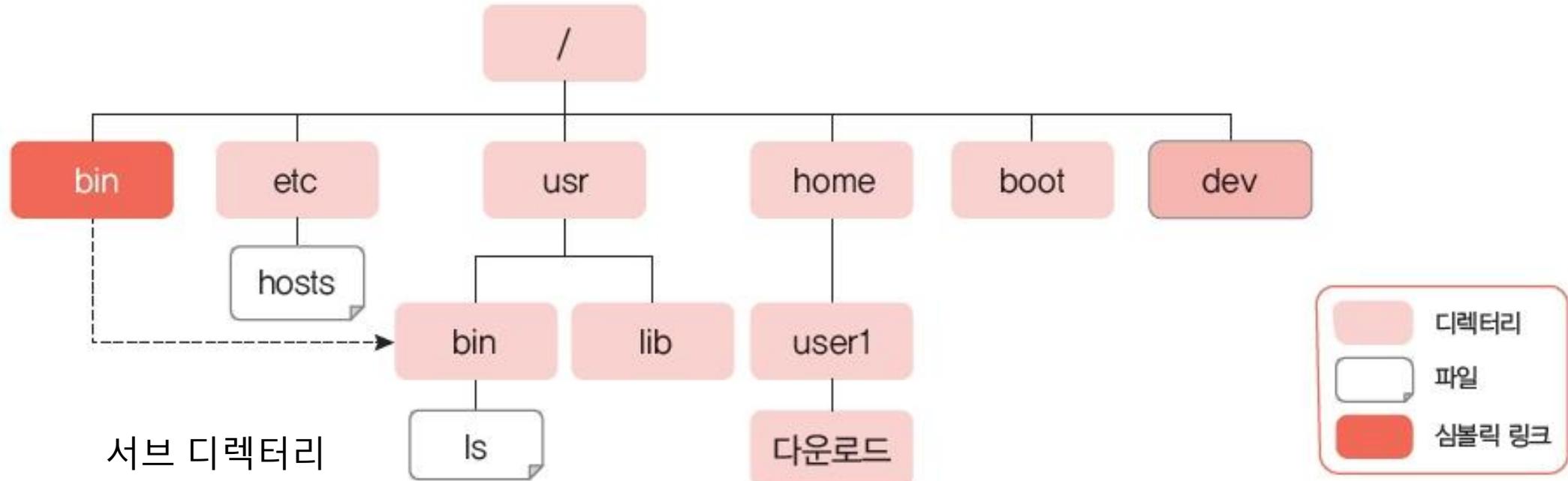
사용법	기능
!!	바로 직전에 실행한 명령을 재실행한다.
!번호	히스토리에서 해당 번호의 명령을 재실행한다.
!문자열	히스토리에서 해당 문자열로 시작하는 마지막 명령을 재실행한다.

```
user1@myubuntu:/tmp$ history
```

(생략)

```
340 unalias cd
341 function cdpwd { cd $1;pwd; }
342 cdpwd /tmp
343 cdpwd
344 cdpwd /tmp
345 history
```

# 파일과 디렉터리 - 계층 구조



자식디렉터리(하위 디렉터리)

부모디렉터리(상위 디렉터리)

작업디렉터리,  
홈 디렉터리

# 파일과 디렉터리 - file

- 파일의 종류를 알아보는 명령

```
user1@myubuntu:~$ file .profile  
.profile: ASCII text  
user1@myubuntu:~$ file 다운로드  
다운로드: directory
```



텍스트 파일, 디렉터리

```
user1@myubuntu:~$ file /bin/bash  
/bin/bash: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV),  
(생략)
```



실행 파일

# 파일과 디렉터리 - 주요 기능

디렉터리	기능
dev	장치 파일이 담긴 디렉터리다.
home	사용자 홈 디렉터리가 생성되는 디렉터리다.
media	DVD/CD나 USB 같은 외부 장치를 연결(마운트라고 함)하는 디렉터리다.
opt	추가 패키지가 설치되는 디렉터리다.
root	root 계정의 홈 디렉터리다. 루트(/) 디렉터리와 다른 것이므로 혼동하지 않도록 한다.
sys	리눅스 커널과 관련된 파일이 있는 디렉터리다.
usr	기본 실행 파일과 라이브러리 파일, 헤더 파일 등 많은 파일이 있다. 참고로 usr는 'Unix System Resource'의 약자다.
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 디렉터리다.
proc	프로세스 정보 등 커널 관련 정보가 저장되는 디렉터리다.
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.

# 파일과 디렉터리 - 현재 디렉터리 확인

## pwd

- **기능** 현재 디렉터리의 위치를 확인한다. 즉 현재 디렉터리의 절대 경로를 출력한다.
- **형식** pwd

```
user1@myubuntu:~$ pwd  
/home/user1
```

# 파일과 디렉터리 - 디렉터리 내용 확인

## ls

- **기능** 디렉터리의 내용을 출력한다.
- **형식** ls [옵션] [디렉터리(파일)]
- **옵션**
  - a: 숨김 파일을 포함하여 모든 파일의 목록을 출력한다.
  - d: 디렉터리 자체의 정보를 출력한다.
  - i: 첫 번째 행에 inode 번호를 출력한다.
  - l: 파일의 상세 정보를 출력한다.
  - A: .(마침표)와 ..(마침표 두 개)를 제외한 모든 파일 목록을 출력한다.
  - F: 파일의 종류를 표시한다(\*: 실행 파일, /: 디렉터리, @: 심볼릭 링크).
  - L: 심볼릭 링크 파일의 경우 원본 파일의 정보를 출력한다.
  - R: 하위 디렉터리의 목록까지 출력한다.
- **사용 예** ls      ls -F      ls -al /tmp

# 파일과 디렉터리 - 파일 속성

```
user1@myubuntu:~$ ls -l /etc/hosts
-rw-r--r-- 1 root root 223 12월  7 21:37 /etc/hosts
```

번호	속성 값	의미
①	-	파일의 종류(-: 일반 파일, d: 디렉터리)
②	rw-r--r--	파일을 읽고 쓰고 실행할 수 있는 접근 권한 표시
③	1	하드 링크의 개수
④	root	파일 소유자의 로그인 ID
⑤	root	파일이 속한 그룹 이름
⑥	223	파일의 크기(바이트 단위)
⑦	12월 7 21:37	파일이 마지막으로 수정된 날짜
⑧	/etc/hosts	파일명

# 파일과 디렉터리 - 접근 권한 종류

권한	파일	디렉터리
읽기	파일을 읽거나 복사할 수 있다.	ls 명령으로 디렉터리 목록을 볼 수 있다(ls 명령의 옵션은 실행 권한이 있어야 사용할 수 있다).
쓰기	파일을 수정 · 이동 · 삭제할 수 있다(디렉터리에 쓰기 권한이 있어야 한다).	파일을 생성하거나 삭제할 수 있다.
실행	파일을 실행할 수 있다(셸 스크립트나 실행 파일의 경우).	cd 명령을 사용할 수 있다. 파일을 디렉터리로 이동하거나 복사할 수 있다.

# 파일과 디렉터리 - 접근 권한 표기 방법

```
user1@myubuntu:~$ ls -l /etc/hosts
```

```
-rw-r--r-- 1 root root 223 12월  7 21:37 /etc/hosts
```



rw-

소유자

r--

그룹

r--

기타 사용자

# 파일과 디렉터리 - 접근 권한 변경

## chmod

- **기능** 파일이나 디렉터리의 접근 권한을 변경한다.
  - **형식** chmod [옵션] 권한 파일(디렉터리)
  - **옵션** -R: 하위 디렉터리까지 모두 변경할 수 있다.
- 
- 기호 모드: 접근 권한을 변경하기 위해 문자와 기호를 사용하여 권한을 표시
  - 숫자 모드: 접근 권한을 변경하기 위해 숫자를 사용

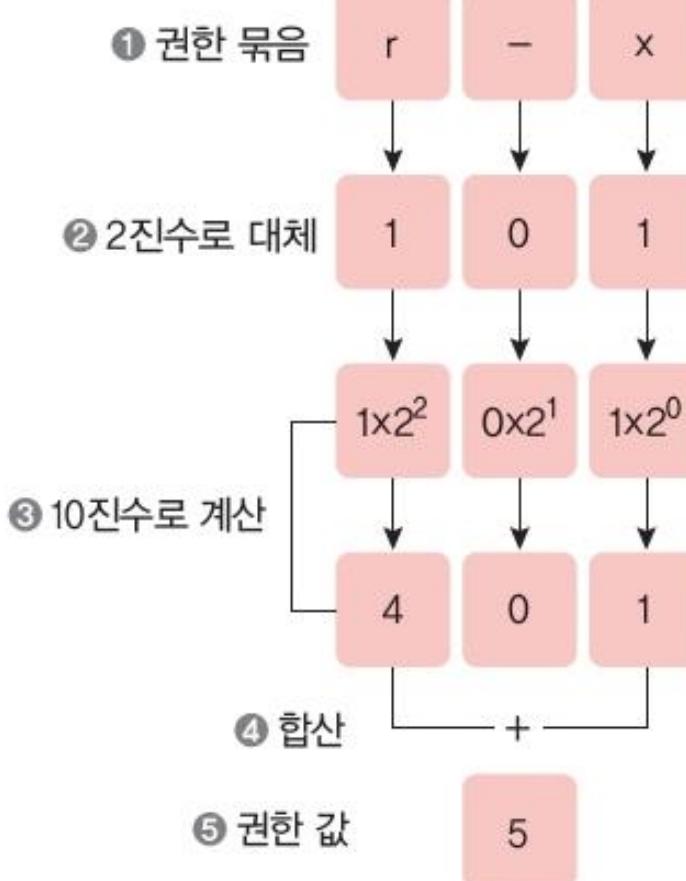
# 파일과 디렉터리 - 기호 모드



구분	문자/기호	의미
사용자 카테고리 문자	u	파일 소유자
	g	파일 소유 그룹
	o	소유자와 그룹 이외의 기타 사용자
	a	전체 사용자
연산자 기호	+	권한 부여
	-	권한 제거
	=	접근 권한 설정
접근 권한 문자	r	읽기 권한
	w	쓰기 권한
	x	실행 권한

권한 표기	의미
u+w	소유자(u)에게 쓰기(w) 권한 부여(+)
u-x	소유자(u)의 실행(x) 권한 제거(-)
g+w	그룹(g)에 쓰기(w) 권한 부여(+)
o-r	기타 사용자(o)의 읽기(r) 권한 제거(-)
g+wx	그룹(g)에 쓰기(w)와 실행(x) 권한 부여(+)
+wx	모든 사용자에게 쓰기(w)와 실행(x) 권한 부여(+)
a+rwx	모든 사용자에게 읽기(r), 쓰기(w), 실행(x) 권한 부여(+)
u=rwx	소유자(u)에게 읽기(r), 쓰기(w), 실행(x) 권한 부여(=)
go+w	그룹(g)과 기타 사용자(o)에게 쓰기(w) 권한 부여(+)
u+x,go+w	소유자(u)에게 실행(x) 권한을 부여하고(+) 그룹(g)과 기타 사용자(o)에게 쓰기(w) 권한 부여(+)

# 파일과 디렉터리 - 숫자 모드



# 파일과 디렉터리 - 디렉터리 생성

## mkdir

- **기능** 디렉터리를 생성한다.
- **형식** `mkdir [옵션] [디렉터리]`
- **옵션** `-p`: 하위 디렉터리를 계층적으로 생성할 때 중간 단계의 디렉터리가 없으면 자동으로 중간 단계 디렉터리를 생성하고 지정한 디렉터리를 생성한다.
- **사용 예** `mkdir temp`

```
user1@myubuntu:~$ mkdir temp
```

```
user1@myubuntu:~$ ls temp
```

```
user1@myubuntu:~$ ls
```

```
temp 공개 다운로드 문서 바탕화면 비디오
```

```
user1@myubuntu:~$ mkdir -p temp/mid/han
```

```
user1@myubuntu:~$ mkdir tmp1 tmp2 tmp3
```

```
user1@myubuntu:~$ ls
```

```
temp tmp2 공개 문서 비디오 음악
```

```
tmp1 tmp3 다운로드 바탕화면 사진 템플릿
```

# 파일과 디렉터리 - 디렉터리 삭제

## rmdir

- **기능** 디렉터리를 삭제한다.
- **형식** `rmdir [옵션] [디렉터리]`
- **옵션** `-p`: 지정한 디렉터리를 삭제하고, 그 디렉터리의 부모 디렉터리가 빈 디렉터리일 경우 부모 디렉터리도 자동으로 삭제한다.
- **사용 예** `rmdir temp`

```
user1@myubuntu:~$ rmdir tmp3
```

```
user1@myubuntu:~$ ls
```

```
temp tmp1 tmp2 공개
```

빈 디렉터리가 아니면 삭제 불가

```
user1@myubuntu:~$ rmdir temp
```

```
rmdir: failed to remove 'temp': 디렉터리가 비어있지 않음
```

# 파일과 디렉터리 - 파일 내용 연속 출력

## cat

- **기능** 파일 내용을 출력한다.
- **형식** cat [옵션] [파일]
- **옵션** -n: 행 번호를 붙여서 출력한다.
- **사용 예** cat file1      cat -n file1

```
user1@myubuntu:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 myubuntu

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
```

```
user1@myubuntu:~$ cat -n /etc/hosts
1 127.0.0.1 localhost
2 127.0.1.1 myubuntu
3
4 # The following lines are desirable for
5 ::1 ip6-localhost ip6-loopback
6 fe00::0 ip6-localnet
7 ff00::0 ip6-mcastprefix
8 ff02::1 ip6-allnodes
9 ff02::2 ip6-allrouters
```

# 파일과 디렉터리 - 파일 화면 단위 출력

## more

- **기능** 파일 내용을 화면 단위로 출력한다.
- **형식** more [옵션] [파일]
- **옵션** + 행 번호: 출력을 시작할 행 번호를 지정한다.
- **사용 예** more file1

- ✓ 스페이스바: 다음 화면 출력
- ✓ 엔터키: 한 행씩 스크롤

```
user1@myubuntu:~$ more /etc/services
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, officially ports have two entries
# even if the protocol doesn't support UDP operations.
#
(생략)
qotd      17/tcp      quote
chargen   19/tcp      ttyst source
chargen   19/udp     ttyst source
ftp-data  20/tcp
ftp       21/tcp
—More—(5%)
```

# 파일과 디렉터리 - 파일 복사

## cp

- **기능** 파일이나 디렉터리를 복사한다.
- **형식** cp [옵션] [파일1(디렉터리1)] [파일2(디렉터리2)]
- **옵션** -i: 파일2가 이미 존재하면 덮어쓸 것인지 물어본다.  
-r: 디렉터리를 복사할 때 지정한다.
- **사용 예** cp file1 file2  
cp f1 f2 f3 dir1  
cp -r dir1 dir2

## 인자를 여러 개 지정할 경우

```
user1@myubuntu:~/linux_ex/ch2$ cp /etc/hosts /etc/services temp
```

```
user1@myubuntu:~/linux_ex/ch2$ ls temp
```

```
hosts services text1 text2
```

# 파일과 디렉터리 - 파일 이동

## mv

- **기능** 파일 또는 디렉터리를 이동하거나 이름을 바꾼다.
- **형식** mv [옵션] [파일1(디렉터리1)] [파일2(디렉터리2)]
- **옵션** -i: 파일2(디렉터리2)가 존재하면 덮어쓸 것인지 물어본다.
- **사용 예** mv file1 file2

파일을 파일로 이동하기(파일명 바꾸기)

```
user1@myubuntu:~/linux_ex/ch2$ mv text1 data1
user1@myubuntu:~/linux_ex/ch2$ ls
data1  one  temp  temp2
```

# 파일과 디렉터리 - 파일 삭제

## rm

- **기능** 파일을 삭제한다.
- **형식** rm [옵션] [파일 또는 디렉터리]
- **옵션** -i: 파일을 정말 삭제할 것인지 확인한다.  
-r: 디렉터리를 삭제할 때 지정한다.
- **사용 예** rm file      rm -r dir

```
user1@myubuntu:~/linux_ex/ch2$ rm data2
user1@myubuntu:~/linux_ex/ch2$ ls
data1  one  temp
```

# 파일과 디렉터리 - 빈 파일 만들기

## touch

- **기능** 빈 파일을 생성한다.
- **형식** touch [-acm] [-r ref\_file | -t time] [파일]
- **옵션**
  - a: 접근 시간만 변경한다.
  - m: 수정 시간만 변경한다.
  - t [[CC]YY]MMDDhhmm[.ss]: 시간을 직접 입력한다.
- **사용 예** touch test

```
user1@myubuntu:~/linux_ex/ch2$ touch test
user1@myubuntu:~/linux_ex/ch2$ ls -l test
-rw-rw-r-- 1 user1 user1 0 12월  7 21:47 test
```

# 파일과 디렉터리 - 파일 내용 검색

## grep

- **기능** 지정한 패턴이 포함된 행을 찾는다.
- **형식** grep [옵션] [패턴] [파일]
- **옵션**
  - i: 대문자 · 소문자를 모두 검색한다.
  - l: 지정한 패턴이 포함된 파일명을 출력한다.
  - n: 행 번호를 출력한다.
- **사용 예**

```
grep root /etc/passwd
grep -n unix ~/*.txt
grep -l hello *.c
```

```
user1@myubuntu:~/linux_ex/ch2$ cp /etc/services data
user1@myubuntu:~/linux_ex/ch2$ grep NNTP data
nntpss          563/tcp        snntp      # NNTP over SSL
```

```
user1@myubuntu:~/linux_ex/ch2$ grep -n NNTP data
121:nntpss 563/tcp snntp # NNTP over SSL
```

# 파일과 디렉터리 - 파일 찾기

## find

- **기능** 지정한 위치에서 검색 조건에 맞는 파일을 찾는다.
- **형식** find [경로] [검색 조건] [동작]
- **검색 조건** -name filename: 파일명으로 검색한다.  
-type 파일 종류: 파일 종류로 검색한다.  
-user loginID: 지정한 사용자가 소유한 모든 파일을 검색한다.  
-perm 접근 권한: 지정한 사용 권한과 일치하는 파일을 검색한다.
- **동작** -exec 명령 {} \;; 검색된 파일에 명령을 실행한다.  
-ok 명령 {} \;; 사용자의 확인을 받아서 명령을 실행한다.  
-print: 검색된 파일의 절대 경로명을 화면에 출력한다(기본 동작).  
-ls: 검색 결과를 긴 목록 형식으로 출력한다.
- **사용 예** find ~ -name hello.c  
find /tmp -user user10 -exec rm {} \;

```
user1@myubuntu:~/linux_ex/ch2$ find /usr/bin -name ls
/usr/bin/ls
```

# 입출력 - 표준 입출력



파일 디스크립터	파일 디스크립터 대신 사용하는 이름	정의
0	stdin	명령의 표준 입력
1	stdout	명령의 표준 출력
2	stderr	명령의 표준 오류

# 입출력 - 출력 리디렉션

- 명령의 결과를 화면 출력이 아닌 파일에 저장
- 파일 덮어쓰기: >
  - ✓ 명령의 실행결과를 기존 파일의 내용에 덮어쓴다.
- 파일 내용에 추가하기: >>
- 오류를 파일에 저장하기

2>

- **기능** 표준 오류 메시지를 파일에 저장한다.
- **형식** 명령 2> 파일명

>

- **기능** 파일 리다이렉션(덮어쓰기)을 한다.
- **형식** 명령 1> 파일명  
명령 > 파일명

>>

- **기능** 파일에 내용을 추가한다.
- **형식** 명령 >> 파일명

```
user1@myubuntu:~/linux_ex/ch4$ ls /abc 2> ls.err
```

→ 표준 오류를 리다이렉션한다.

```
user1@myubuntu:~/linux_ex/ch4$ cat ls.err
```

ls: '/abc'에 접근할 수 없습니다: 그런 파일이나 디렉터리가 없습니다

→ 파일에 저장된 메시지이다.

# 프로세스 관리 - 목록 확인

## ps

- **기능** 현재 실행 중인 프로세스에 대한 정보를 출력한다.
  - **형식** ps [옵션]
  - **옵션** <유닉스 옵션>
    - e: 시스템에서 실행 중인 모든 프로세스의 정보를 출력한다.
    - f: 프로세스에 대한 자세한 정보를 출력한다.
    - u uid: 특정 사용자에 대한 모든 프로세스의 정보를 출력한다.
    - p pid: pid로 지정한 특정 프로세스의 정보를 출력한다.
  - <BSD 옵션>
    - a: 터미널에서 실행시킨 프로세스의 정보를 출력한다.
    - u: 프로세스 소유자 이름, CPU 사용량, 메모리 사용량 등 상세 정보를 출력한다
    - x: 시스템에서 실행 중인 모든 프로세스의 정보를 출력한다.
  - <GNU 옵션>
  - pid PID 목록: 목록으로 지정한 특정 PID 정보를 출력한다.
- **사용 예** ps  
ps -ef  
ps aux

```
user1@myubuntu:~$ ps
```

PID	TTY	TIME	CMD
1584	pts/0	00:00:00	bash
1653	pts/0	00:00:00	ps

```
user1@myubuntu:~$ ps -f
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
user1	1584	1583	0	00:00	pts/0	00:00:00	-bash
user1	1660	1584	0	00:16	pts/0	00:00:00	ps -f

항목	의미	항목	의미
UID	프로세스를 실행한 사용자 ID	STIME	프로세스의 시작 날짜나 시간
PID	프로세스 번호	TTY	프로세스가 실행된 터미널의 종류와 번호
PPID	부모 프로세스 번호	TIME	프로세스 실행 시간
C	CPU 사용량(% 값)	CMD	실행되고 있는 프로그램 이름(명령)

# 프로세스 관리 - 고급 목록 확인

- 전체 프로세스 목록 출력하기: -e 옵션과 -f 옵션

```
user1@myubuntu:~$ ps -ef | more
UID      PID  PPID  C STIME TTY          TIME CMD
root        1      0  0 12월18 ?        00:00:01 /sbin/init splash
root        2      0  0 12월18 ?        00:00:00 [kthreadd]
root        3      2  0 12월18 ?        00:00:00 [rcu_gp]
root        4      2  0 12월18 ?        00:00:00 [rcu_par_gp]
root        6      2  0 12월18 ?        00:00:00 [kworker/0:0H-events_highpri]
(생략)
```

- 특정 사용자의 프로세스 목록

```
user1@myubuntu:~$ ps -fu user1
UID      PID  PPID  C STIME TTY          TIME CMD
user1    1446      1  0 00:00 ?        00:00:00 /lib/systemd/systemd --user
user1    1448    1446  0 00:00 ?        00:00:00 (sd-pam)
user1    1456    1446  0 00:00 ?        00:00:00 /usr/bin/pipewire
user1    1457    1446  0 00:00 ?        00:00:00 /usr/bin/pipewire-media-sess
user1    1458    1446  0 00:00 ?        00:00:00 /usr/bin/pulseaudio --daemon
```

# 프로세스 관리 - 프로세스 종료

## ■ PID 사용한 종료

### kill

- **기능** 지정한 시그널을 프로세스에게 보낸다.
- **형식** kill [-시그널] PID...
- **시그널** 2: 인터럽트 시그널을 보낸다(**Ctrl**+C).  
9: 프로세스를 강제로 종료한다.  
15: 프로세스와 관련된 파일들을 정리하고 종료한다. 종료되지 않는 프로세스가 있을 수 있다.
- **사용 예** kill 1001  
kill -15 1001  
kill -9 1001

## ■ 프로세스 명령 이름으로 종료: pkill

```
user1@myubuntu:~$ pkill man
```

# 프로세스 관리 - 포그라운드 및 백그라운드 작업

- 사용자가 입력한 명령이 실행되어 결과가 출력될 때까지 기다리는 방식으로 처리되는 프로세스를 포그라운드 프로세스

```
user1@myubuntu:~$ sleep 100
```

→ 포그라운드 작업

→ sleep 명령이 끝날 때까지 기다려야 한다.

- 백그라운드 방식으로 명령을 실행하면 명령의 처리가 끝나는 것과 관계없이 곧바로 프롬프트가 출력되어 사용자가 다른 작업을 계속할 수 있음

```
user1@myubuntu:~$ sleep 100 &
```

→ 백그라운드 작업

```
user1@myubuntu:~$
```

→ 프롬프트가 바로 나와 다른 명령을 실행시킬 수 있다.

# 프로세스 관리 - 작업 전환

## ■ 작업 목록 확인: jobs

- +: 가장 최근에 접근한 작업
- : + 작업 바로 전에 접근한 작업
- 공백: 그 외의 작업

### jobs

- 기능** 백그라운드 작업을 모두 보여준다. 특정 작업 번호를 지정하면 해당 작업의 정보만 보여준다.
- 형식** jobs [%작업 번호]
- %작업번호** %번호: 해당 번호의 작업 정보를 출력한다.  
%+ 또는 %: 작업 순서가 +인 작업 정보를 출력한다.  
% -: 작업 순서가 -인 작업 정보를 출력한다.
- 사용 예** jobs %1

jobs

user1@myubuntu:~\$ jobs

[1]- 실행중

sleep 100 &

[2]+ 실행중

find / -name passwd > pw.dat 2>&1 &

## ■ 작업 전환

명령	기능
<code>Ctrl+z</code> 또는 stop %작업 번호	포그라운드 작업을 중지한다(종료하는 것이 아니라 잠시 중단하는 것이다).
bg %작업 번호	작업 번호가 지시하는 작업을 백그라운드 작업으로 전환한다.
fg %작업 번호	작업 번호가 지시하는 작업을 포그라운드 작업으로 전환한다.

# 프로세스 관리 - 특정 프로세스 정보 검색

- 'ps -ef | grep' 명령의 형태로 연결해서 사용

```
user1@myubuntu:~$ ps -ef | grep bash
user1      2096     2095  0 13:18 pts/0    00:00:00 -bash
user1      2111     2096  0 13:18 pts/0    00:00:00 grep --color=auto bash
```

## pgrep

- **기능** 지정한 패턴과 일치하는 프로세스의 정보를 출력한다.
- **형식** pgrep [옵션] [패턴]
- **옵션**
  - x: 패턴과 정확히 일치하는 프로세스의 정보를 출력한다.
  - n: 패턴을 포함하고 있는 가장 최근 프로세스의 정보를 출력한다.
  - u 사용자 이름: 특정 사용자에 대한 모든 프로세스를 출력한다.
  - l: PID와 프로세스 이름을 출력한다.
  - t term: 특정 단말기와 관련된 프로세스의 정보를 출력한다.
- **사용 예** pgrep bash

```
user1@myubuntu:~$ pgrep -x bash
2096
```

bash 패턴을 지정하여 검색

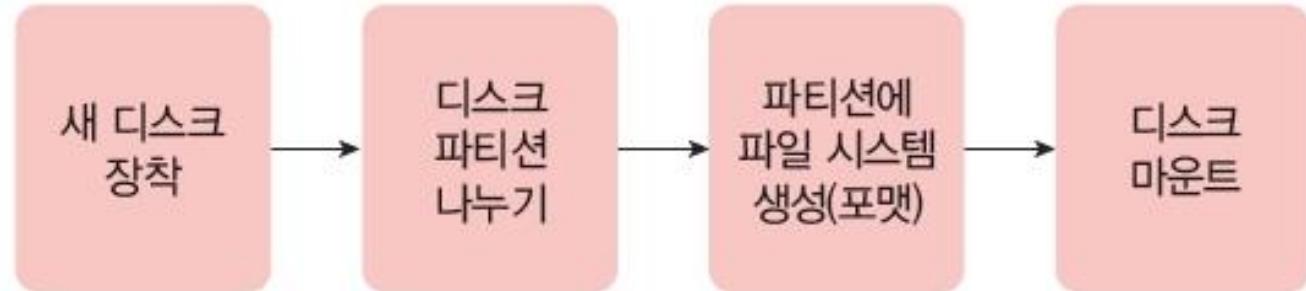
```
user1@myubuntu:~$ pgrep -l bash
2096 bash
```

-l 옵션은 PID와 명령 이름을 출력

# 디스크 관리 - 디스크 추가 과정

- 디스크 장치의 이름과 파티션 표시

- ✓ /dev/sd로 시작하는 이름 사용
  - /dev/sda: 첫 번째 디스크
  - /dev/sdb: 두 번째 디스크
  - /dev/sdc: 세 번째 디스크



- ✓ 장치명이 /dev/sda인 경우 파티션 이름 표시

- /dev/sda: 첫 번째 디스크 전체를 의미하는 장치명
- /dev/sda0: 디스크의 첫 번째 파티션
- /dev/sda1: 디스크의 두 번째 파티션

# 디스크 관리 - 파티션 관리

## fdisk

- **기능** 디스크의 파티션 생성, 삭제, 보기 등 파티션을 관리한다.
- **형식** fdisk [옵션] [장치명]
- **옵션** -b 크기: 섹터 크기를 지정한다(512, 1024, 2048, 4096).  
-l: 파티션 테이블을 출력한다.
- **사용 예** fdisk /dev/sdb  
fdisk -l

명령	기능	명령	기능
a	부팅 파티션을 설정한다.	p	파티션 테이블을 출력한다.
b	BSD 디스크 라벨을 편집한다.	q	작업 내용을 저장하지 않고 종료한다.
c	도스 호환성을 설정한다.	s	새로운 빈 Sun 디스크 라벨을 생성한다.
d	파티션을 삭제한다.	t	파티션의 시스템 ID를 변경한다(파일 시스템 종류 변경).
l	사용 가능한 파티션의 종류를 출력한다.	u	항목 정보를 변경 · 출력한다.
m	도움말을 출력한다.	v	파티션 테이블을 검사한다.
n	새로운 파티션을 추가한다.	w	파티션 정보를 디스크에 저장하고 종료한다.
o	새로운 빈 DOS 파티션을 생성한다.	x	실린더 개수 변경 등 전문가를 위한 부가적 기능이다.

# 디스크 관리 - 파일 시스템 생성

## mkfs

- **기능** 리눅스 파일 시스템을 만든다.
- **형식** mkfs [옵션] [장치명]
- **옵션** -t 종류: 파일 시스템의 종류를 지정한다(기본값은 ext2).
- **사용 예** mkfs /dev/sdb1  
mkfs -t ext4 /dev/sdb1

```
user1@myubuntu:~$ sudo mkfs /dev/sdb1
mke2fs 1.46.3 (27-Jul-2021)

Creating filesystem with 128000 4k blocks and 128000 inodes
Filesystem UUID: b77be8f5-1299-4223-a831-58280caf302e
Superblock backups stored on blocks:
            32768, 98304

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

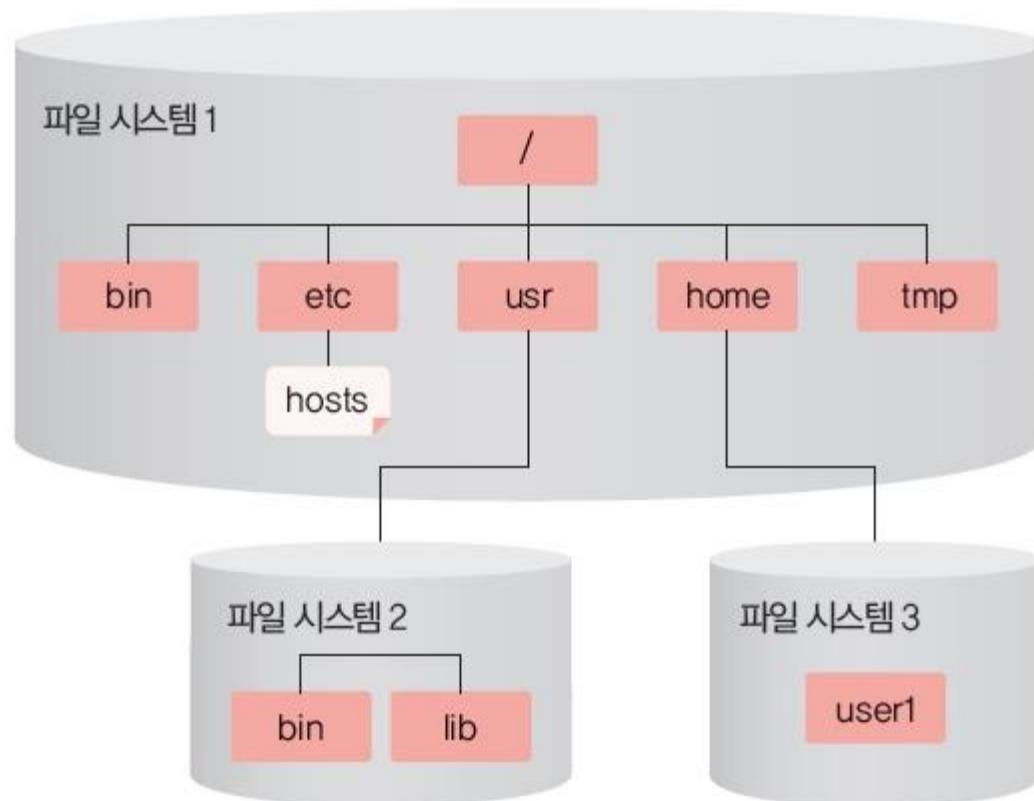
# 디스크 관리 - 파일 시스템 마운트

## ■ 마운트(mount)

- 파일 시스템을 디렉터리 계층 구조의 특정 디렉터리와 연결하는 것
- 파일 시스템이 디렉터리 계층 구조와 연결되지 않으면 사용자가 해당 파일 시스템에 접근할 수 없음

## ■ 마운트 포인트

- 디렉터리 계층 구조에서 파일 시스템이 연결되는 디렉터리



# 디스크 관리 - 파일 시스템 마운트 설정 파일

## ■ /etc/fstab

- 파일 시스템의 마운트 설정 정보 저장
- 부팅 시 파일 시스템 마운트를 위해 로드

- 구조



```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>      <dump> <pass>
# / was on /dev/sda3 during installation
UUID=265c8913-ee1e-4034-885f-207969b0a23b /          ext4    errors=remount-
ro 0      1
# /boot/efi was on /dev/sda2 during installation
UUID=A7A5-E2E2  /boot/efi      vfat    umask=0077      0      1
/swapfile           none        swap     sw      0      0
```

# 디스크 관리 - 마운트 및 마운트 해제 명령

## mount

- **기능** 파일 시스템을 마운트한다.
- **형식** mount [옵션] [장치명 마운트 포인트]
- **옵션**
  - t 파일 시스템 종류: 파일 시스템 종류를 지정한다.
  - o 마운트 옵션: 마운트 옵션을 지정한다.
  - f: 마운트를 할 수 있는지 점검만 한다.
  - r: 읽기만 가능하게 마운트한다(-o ro와 동일).

- **사용 예** mount

```
mount /dev/sdb1 /
```

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

mount 명령만 사용하는 경우: 현재 마운트 정보 출력

## umount

- **기능** 파일 시스템을 언마운트한다.
- **형식** umount [옵션] 장치명 또는 마운트 포인트
- **옵션** -t 파일 시스템 종류: 파일 시스템 종류를 지정한다.
- **사용 예** umount /dev/sdb1  
umount /mnt

장치	mount 명령 형식의 예
ext2 파일 시스템	mount -t ext2 /dev/sdb1 /mnt
ext3 파일 시스템	mount -t ext3 /dev/sdb1 /mnt
ext4 파일 시스템	mount -t ext4 /dev/sdb1 /mnt mount /dev/sdb1 /mnt
CD-ROM	mount -t iso9660 /dev/cdrom /mnt/cdrom
윈도 디스크	mount -t vfat /dev/hdc /mnt
USB 메모리	mount /dev/sdc1 /mnt → 리눅스용 USB 메모리의 경우 mount -t vfat /dev/sdc1 /mnt → 윈도용 USB 메모리의 경우
읽기 전용 마운트	mount -r /dev/sdb1 /mnt
읽기/쓰기 마운트	mount -w /dev/sdb1 /mnt
원격 디스크 마운트	mount -t nfs 서버 주소:/NFS 서버 측 디렉터리 /mnt

# 디스크 관리 - 파일 시스템별 디스크 사용량 확인

## df

- **기능** 디스크의 남은 공간에 대한 정보를 출력한다.
- **형식** df [옵션] [파일 시스템]
- **옵션**
  - a: 모든 파일 시스템을 대상으로 디스크 사용량을 확인한다.
  - k: 디스크 사용량을 KB 단위로 출력한다.
  - m: 디스크 사용량을 MB 단위로 출력한다.
  - h: 디스크 사용량을 알기 쉬운 단위(GB, MB, KB 등)로 출력한다.
  - t 파일 시스템 종류: 지정한 파일 시스템 종류에 해당하는 디스크의 사용량을 출력한다.
  - T: 파일 시스템 종류를 출력한다.
- **사용 예** df  
df -h

# 리눅스VM에 데이터 디스크 추가하기

## 작업 순서

- ① 데이터 디스크 추가
- ② 디스크 포맷 및 검증
- ③ 디스크 마운트
- ④ fstab 파일 업데이트

### 1. VM에 연결된 디스크 확인

```
lsblk -o NAME,HCTL,SIZE,MOUNTPOINT | grep -i "sd"
```

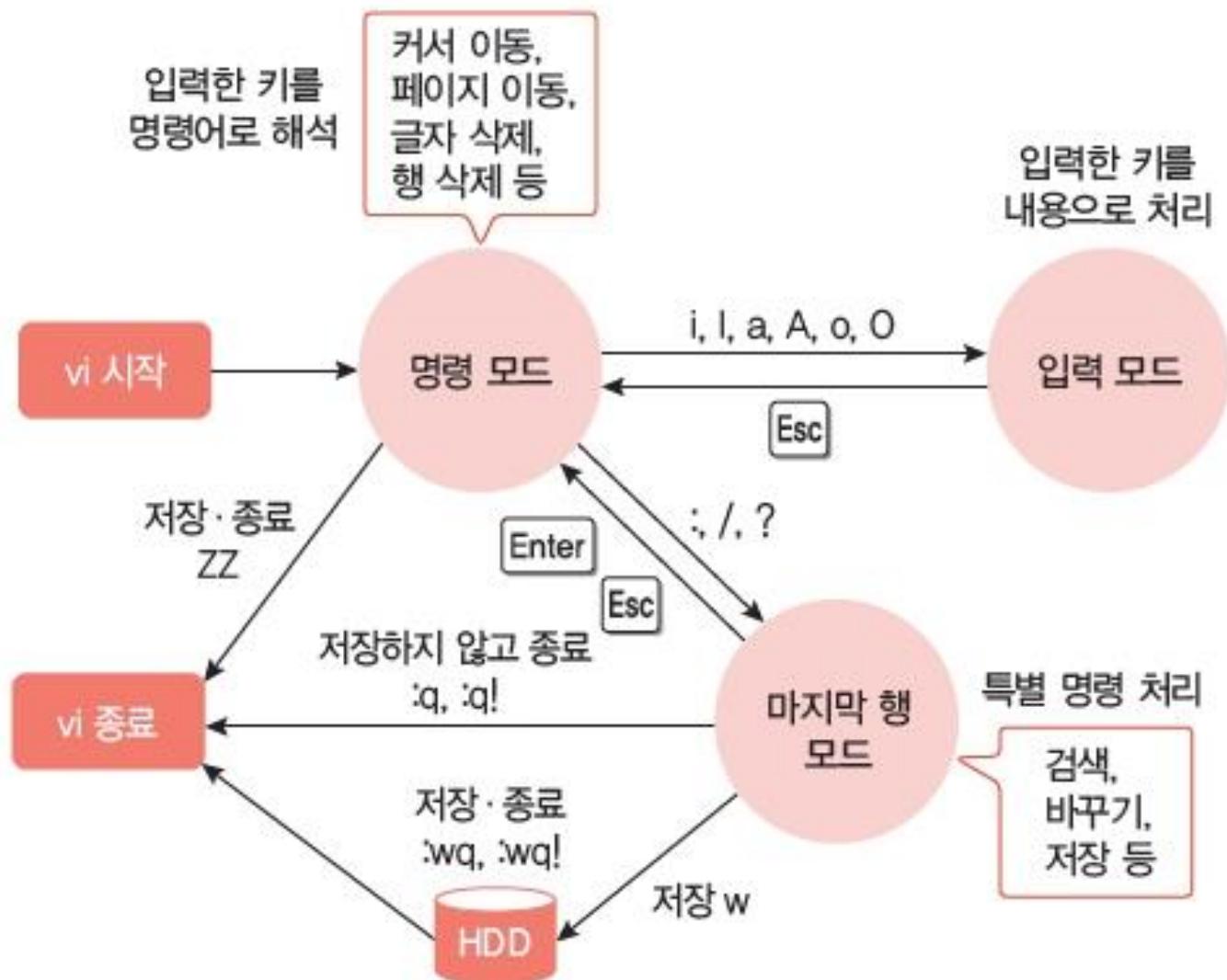
### 2. 디스크 포맷 및 검증

```
sudo parted /dev/sdc --script mklabel gpt mkpart xfspart xfs 0% 100%
sudo mkfs.xfs /dev/sdc1
sudo partprobe /dev/sdc1
```

### 3. 디스크 마운트

```
sudo mkdir /datadrive
sudo mount /dev/sdc1 /datadrive
```

# vi 편집기 - 동작 모드



# vi 편집기 – 시작과 종료

```
[user1@localhost ~]$ vi test.txt
```

→ test.txt라는 파일이 열린다. 이 파일이 없으면  
빈 파일이 열린다.

```
[user1@localhost ~]$ vi
```

→ 빈 파일이 열린다. 나중에 파일명을 정할 수 있다.

모드	명령키	기능
마지막 행 모드	:q	vi에서 작업한 것이 없을 때 그냥 종료한다.
	:q!	작업한 내용을 저장하지 않고 종료한다.
	:w 파일명	작업한 내용을 저장만 한다. 파일명을 지정하면 다른 파일로 저장한다.
	:wq, :wq!	작업한 내용을 저장하고 vi를 종료한다.
명령 모드	ZZ( <b>Shift</b> +zz)	작업한 내용을 저장하고 vi를 종료한다.

# vi 편집기 - 모드 전환

- 명령 모드 -> 입력 모드 전환

- 입력 모드 -> 명령 모드 전환

✓ Esc 키

명령키	기능
i	현재 커서 자리에 입력한다.
a	현재 커서 다음 자리에 입력한다.
o	커서가 위치한 행의 다음 행에 입력한다.
I(대문자 i)	커서가 위치한 행의 첫 칼럼으로 이동하여 입력한다.
A	커서가 위치한 행의 마지막 칼럼으로 이동하여 입력한다.
O	커서가 위치한 행의 이전 행에 입력한다.

# vi 편집기 - 특정 행 이동

명령키	기능
G( <b>Shift</b> +g)	파일의 마지막 행으로 커서가 이동한다.
행 번호G( <b>Shift</b> +g)	지정한 행 번호로 커서가 이동한다.
:행 번호	지정한 행 번호로 커서가 이동한다(마지막 행 모드).
:\$	파일의 마지막 행으로 커서가 이동한다(마지막 행 모드).

# vi 편집기 - 내용 삭제

명령키	기능
x, #x	커서 위치의 글자를 삭제한다. #에는 삭제할 글자 수를 지정한다.
dw, #dw	커서 위치의 단어를 삭제한다. #에는 삭제할 단어 수를 지정한다.
dd, #dd	커서 위치의 행을 삭제한다. #에는 삭제할 행의 수를 지정한다.
D( <b>Shift</b> +d)	커서 위치부터 행의 끝까지 삭제한다.

ubuntu editor vi study  
I like **linu** ubuntu linux

x 명령키로  
| 삭제

ubuntu editor vi study  
I like **inu** ubuntu linux

dw로  
inu 삭제

ubuntu editor vi study  
I like **ubuntu** linux

커서 위치부터  
행의 끝까지 삭제: D

ubuntu editor vi study  
I like **like**

# vi 편집기 - 복사 및 잘라내기, 붙이기

명령키	기능
yy, #yy	커서가 위치한 행을 복사한다. #에는 복사할 행의 수를 지정한다.
p	커서가 위치한 행의 아래쪽에 붙인다.
P	커서가 위치한 행의 위쪽에 붙인다.
dd, #dd	커서가 위치한 행을 잘라둔다. 삭제와 같은 기능이다. #에는 잘라둘 행의 수를 지정한다.

```
ubuntu editor vi study  
I like
```

- ① 2yy 명령키로 두 행을 모두 복사
- ② 아래 행으로 이동하여 p 명령키로 붙이기

```
ubuntu editor vi study  
I like  
ubuntu editor vi study  
I like
```

# vi 편집기 - 검색

명령키	기능
/문자열	문자열을 아래 방향으로 검색한다.
?문자열	문자열을 위 방향으로 검색한다.
n	원래 찾던 방향으로 다음 문자열을 검색한다.
N	역방향으로 다음 문자열을 검색한다.

```
I like  
I like  
ubuntu editor vi study  
ubuntu editor vi study  
I like  
I like  
~  
(생략)  
/ubuntu
```

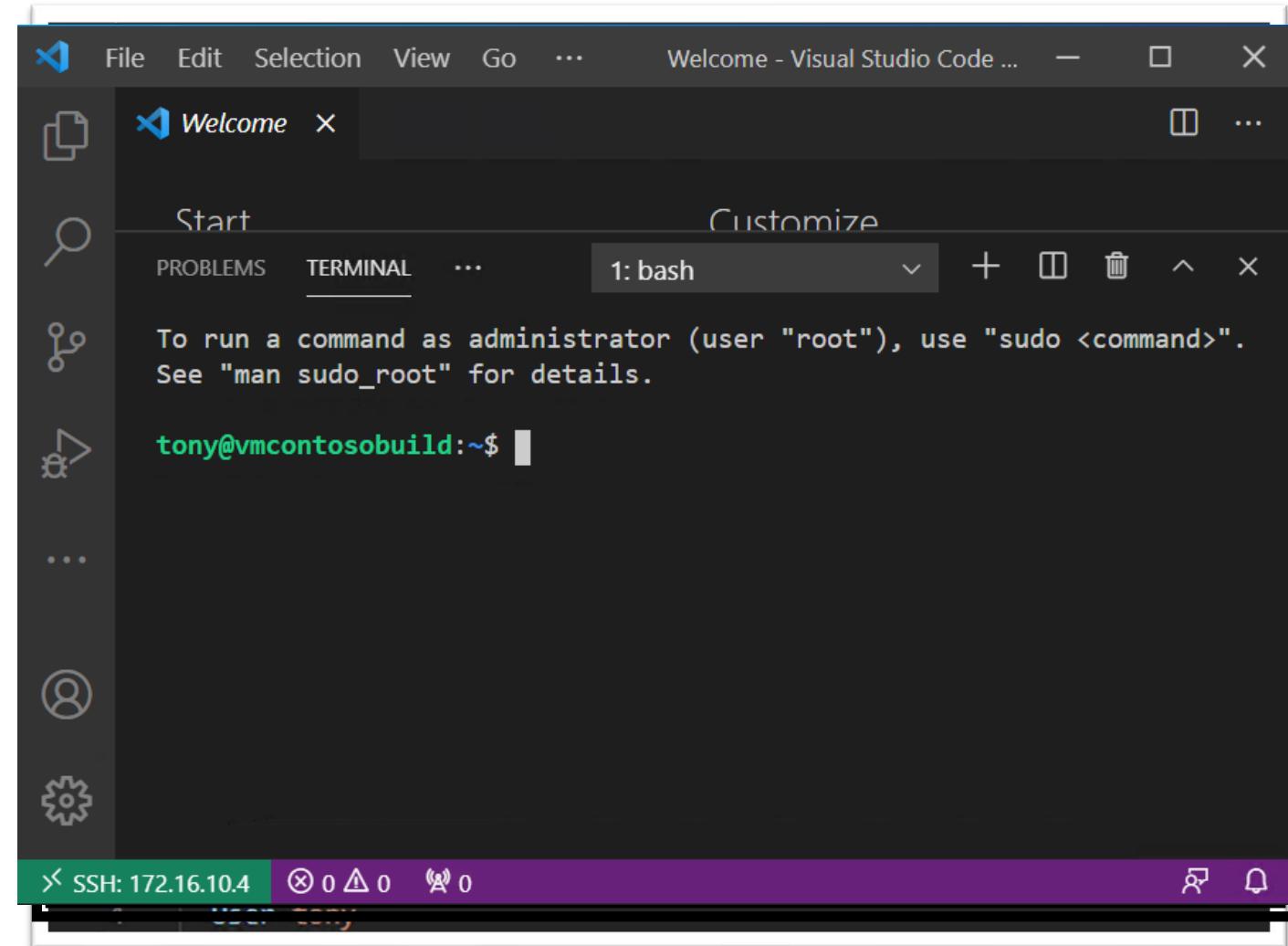
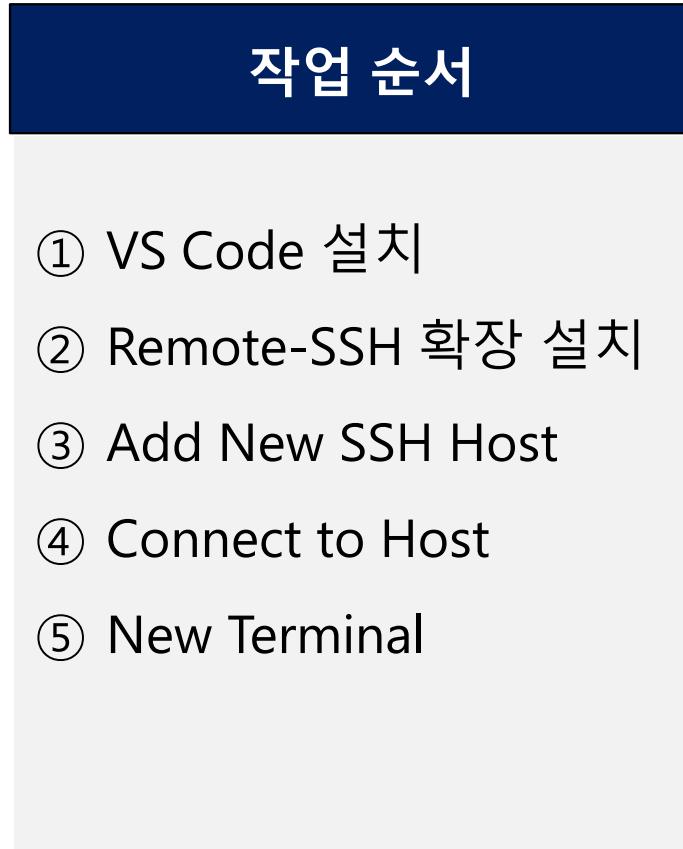
계속 ubuntu를  
검색하려고 n을  
입력하면 4행의  
ubuntu로 커서 이동

- ① 검색하려고 /을 입력하면 커서가 마지막 행으로 이동
- ② 검색할 문자열인 'ubuntu' 입력
- ③ Enter를 누르면 6행이 파일의 마지막 행이므로 'search hit BOTTOM, continuing at TOP'이라는 메시지 출력
- ④ 3행의 ubuntu로 커서 이동



```
I like  
I like  
ubuntu editor vi study  
ubuntu editor vi study  
I like  
I like  
~  
(생략)  
/search hit BOTTOM, continuing at TOP
```

# VS Code로 원격 리눅스 VM에 SSH 연결 구성



# Azure CLI / PowerShell on Linux

## 작업 순서

- ① Azure CLI 설치
- ② Azure CLI 확인
- ③ PowerShell 설치
- ④ Azure PowerShell 설치
- ⑤ Azure PowerShell 확인

Directory: /home/tony/.local/share/powershell/Modules					
ModuleType	Version	PreRelease	Name	PSEdition	ExportedCommands
Script	9.7.0		Az	Core,Desk	{Disable-AzDataCollecti...
Script	2.12.2		Az.Accounts	Core,Desk	{Disable-AzAdvisorRecom...
Script	2.0.0		Az.Advisor	Core,Desk	{Get-AzAksCluster, New...
Script	5.4.0		Az.Aks	Core,Desk	{Resume-AzAnalysisServi...
Script	1.1.4		Az.AnalysisServices	Core,Desk	{Add-AzApiManagementApi...
Script	4.0.2		Az.ApiManagement	Core,Desk	{Get-AzAppConfiguration...
Script	1.2.0		Az.AppConfiguration	Core,Desk	{Get-AzApplicationInsig...
Script	2.2.2		Az.ApplicationInsights	Core,Desk	{Get-AzAttestationPolic...
Script	2.0.0		Az.Attestation	Core,Desk	{Get-AzAutomanageBestPr...
Script	1.0.0		Az.Automanage	Core,Desk	{Set-AzAutomationPython...
Script	1.9.1		Az.Automation	Core,Desk	{Remove-AzBatchAccount, ...
Script	3.4.0		Az.Batch	Core,Desk	{Get-AzBillingInvoice, ...
Script	2.0.1		Az.Billing	Core,Desk	{Clear-AzCdnEndpointCon...
Script	2.1.0		Az.Cdn	Core,Desk	{Get-AzCloudService, Ge...
Script	1.2.0		Az.CloudService	Core,Desk	{Get-AzCognitiveService...
Script	1.13.1		Az.CognitiveServices	Core,Desk	{Remove-AzAvailabilityS...
Script	5.7.1		Az.Compute	Core,Desk	{Get-AzConfidentialLedg...
Script	1.0.0		Az.ConfidentialLedger	Core,Desk	{Add-AzContainerInstanc...
Script	3.2.0		Az.ContainerInstance	Core,Desk	

