

제3장 기본 명령어

VEDA

I

3.1 기본 명령어

기본 명령어 사용

- 날짜 및 시간 확인

```
$ date
```

```
2016년 12월 26일 월요일 오후 01시 52분 02초
```

- 시스템 정보 확인

```
$ hostname
```

```
linux.sookmyung.ac.kr
```

```
$ uname
```

```
Linux
```

```
$ uname -a
```

```
Linux linux.sookmyung.ac.kr 3.10.0-123.el7.x86_64 #1 SMP
```

```
Mon Jun 30 12:09:22 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
```

▶ 3

기본 명령어 사용

- 사용자 정보 확인

```
$ whoami
```

```
chang
```

```
$ who
```

```
chang pts/1 2017-07-12 11:05 (:10.0)
```

```
brain pts/5 2017-07-12 13:46 (203.153.155.35)
```

```
...
```

- 디렉터리 내용 확인

```
$ ls
```

```
Desktop Music Templates Documents Pictures Videos ...
```

▶ 4

기본 명령어 사용

- 패스워드 변경

```
$ passwd
```

```
passwd: chang용 암호를 변경하는 중
```

```
기존 로그인 암호를 입력하십시오:
```

```
새 암호:
```

```
새 암호를 다시 입력하십시오:
```

```
passwd: 암호(chang용)가 성공적으로 변경되었습니다.
```

- 화면 정리

```
$ clear
```

▶ 5

온라인 매뉴얼: man

```
$ man ls
```

```
LS(1) User Commands LS(1)
```

```
NAME
```

```
ls - list directory contents
```

```
SYNOPSIS
```

```
ls [OPTION]... [FILE]...
```

```
DESCRIPTION
```

```
List information about the FILES (the current directory by default).
```

```
Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-  
fied.
```

```
Mandatory arguments to long options are mandatory for short options  
too.
```

```
-a, --all
```

```
do not ignore entries starting with .
```

```
-A, --almost-all
```

```
do not list implied . and ..
```

▶ 6

```
Manual page ls(1) line 1 (press h for help or q to quit)
```

명령어에 대한 간단한 설명: whatis

```
$ whatis ls
```

```
ls (1) - 경로의 내용을 나열한다.
```

```
ls (1p) - list directory contents
```

▶ 7

3.2 파일과 디렉터리

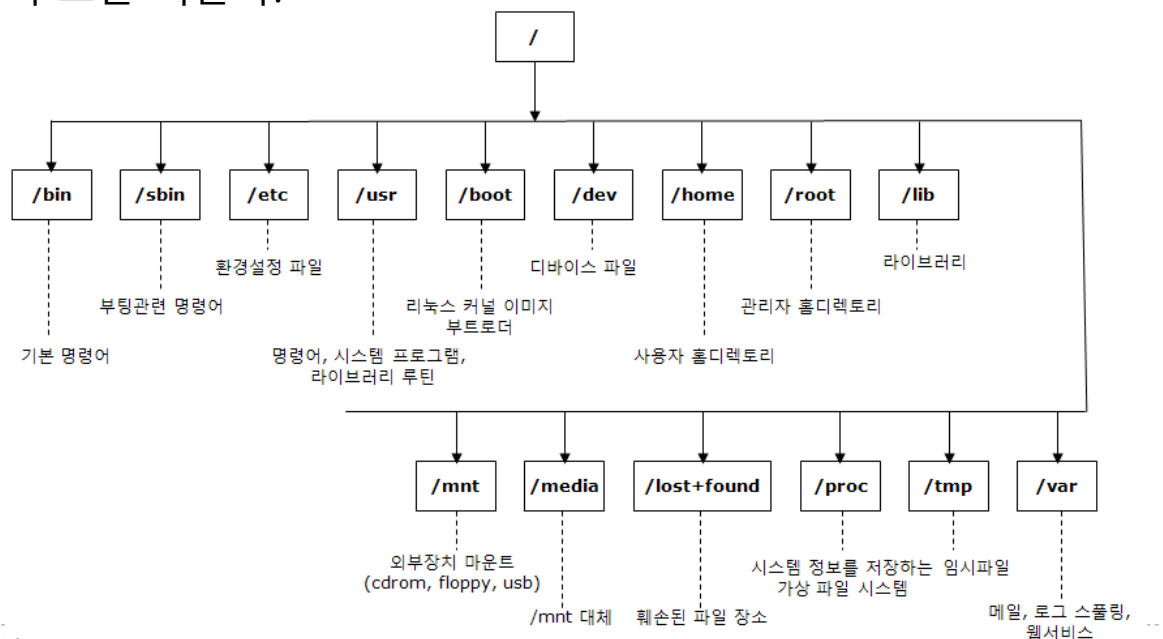
파일의 종류

- 일반 파일(ordinary file)
 - 데이터를 가지고 있으면서 디스크에 저장된다.
 - 텍스트 파일, 이진 파일
- 디렉터리(directory) 또는 폴더(folder)
 - 파일들을 계층적으로 조직화하는 데 사용되는 일종의 특수 파일
 - 디렉터리 내에 파일이나 서브디렉토리들이 존재한다.
- 장치 파일(device special file)
 - 물리적인 장치에 대한 내부적인 표현
 - 키보드(stdin), 모니터(stdout), 프린터 등도 파일처럼 사용
- 심볼릭 링크 파일
 - 어떤 파일을 가리키는 또 하나의 경로명을 저장하는 파일

▶ 9

디렉터리 계층구조

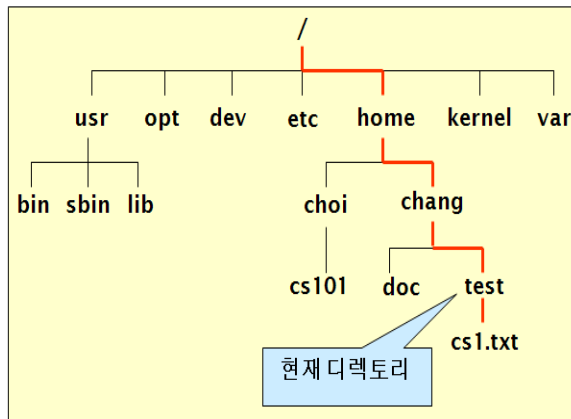
- 리눅스의 디렉터리는 루트로부터 시작하여 트리 형태의 계층 구조를 이룬다.



▶ 10

홈 디렉터리

- 홈 디렉터리(home directory)
 - 각 사용자마다 별도의 홈 디렉터리가 있음
 - 사용자가 로그인하면 홈 디렉터리에서 작업을 시작함



~ : 홈 디렉터리
.: 현재 디렉터리
.. : 부모 디렉터리

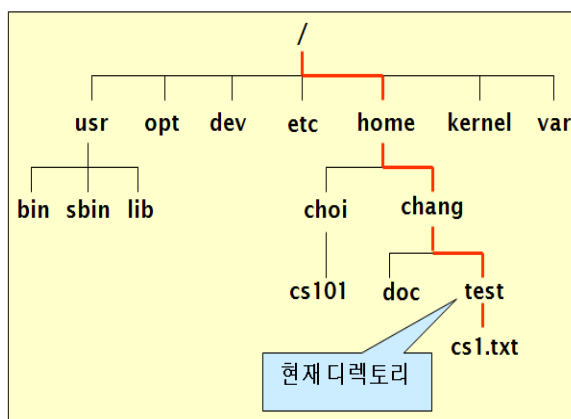
cs1.txt의 절대 경로명
/home/chang/test/cs1.txt

cs1.txt의 상대 경로명
cs1.txt

▶ 11

경로명

- 파일이나 디렉터리에 대한 정확한 이름
- 절대 경로명(absolute pathname)
 - 루트 디렉터리로부터 시작하여 경로 이름을 정확하게 적는 것
- 상대 경로명(relative path name)
 - 현재 작업 디렉터리부터 시작해서 경로 이름을 적는 것



cs1.txt의 절대 경로명
/home/chang/test/cs1.txt

cs1.txt의 상대 경로명
cs1.txt

▶ 12

3.3 디렉터리 명령어 기초

13

현재 작업 디렉터리 출력: `pwd` (print working directory)

- 사용법

```
$ pwd
```

현재 작업 디렉터리의 절대 경로명을 출력한다.

- 현재 작업 디렉터리(current working directory)
 - 현재 작업 중인 디렉터리
 - 로그인 하면 홈 디렉터리에서부터 작업이 시작된다.

- 예

```
$ pwd
/home/chang/Desktop
$ cd ~
$ pwd
/home/chang
```

디렉터리 이동: cd(change directory)

- 사용법

```
$ cd [디렉터리]
```

현재 작업 디렉터리를 지정된 디렉터리로 이동한다??

디렉터를 지정하지 않으면??홈 디렉터리로 이동한다??

- 예

```
$ cd
```

```
$ cd ~
```

```
$ cd Desktop
```

```
$ pwd
```

```
/home/chang/Desktop
```

```
$ cd ..
```

▶ 15

명령어의 경로 확인: which

- 사용법

```
$ which 명령어
```

명령어의 절대경로를 보여준다??

- 예

```
$ which ls
```

```
/bin/ls
```

```
$ which pwd
```

```
/usr/pwd
```

```
$ which passwd
```

```
/usr/passwd
```

▶ 16

3.4 디렉터리 리스트

17

디렉터리 리스트: ls(list)

- 사용법

```
$ ls(혹은 dir) [-aslFR] 디렉터리* 파일*
```

지정된 디렉터리의 내용을 리스트 한다. 디렉터리를 지정하지 않으면 현재 디렉터리 내용을 리스트 한다. 또한 파일을 지정하면 해당 파일만을 리스트 한다.

- 예

```
$ ls /  
bin dev home lib64 mnt proc run srv tmp var  
boot etc lib media opt root sbin sys usr  
$ ls ~  
Desktop Downloads Pictures Templates pl 다운로드  
Documents Music Public Videos linux tmp 사진  
$ cd Desktop  
$ ls  
cs1.txt
```

ls 명령어 옵션

- 주요 옵션

옵션	기능
-a	숨겨진 파일을 포함하여 모든 파일을 리스팅한다
-s	파일의 크기를 바이트 단위로 출력한다
-l	파일의 상세 정보를 출력한다
-F	파일의 종류를 표시하여 출력한다
-R	모든 하위 디렉터리들을 리스팅한다

▶ 19

ls 명령어 옵션

- **ls -s**

- -s(size) 옵션
- 디렉터리 내에 있는 모든 파일의 크기를 K 바이트 단위로 출력

```
$ ls -s
총 4
4 cs1.txt
```

- **ls -a**

- -a(all) 옵션
- 숨겨진 파일들을 포함하여 모든 파일과 디렉터리를 리스팅
- "."은 현재 디렉터리, ".."은 부모 디렉터리

```
$ ls -a
. .. cs1.txt
```

▶ 20

ls 명령어 옵션

● ls -l

- -l(long) 옵션
- 파일 속성(file attribute) 출력
 - 파일 이름, 파일 종류, 접근 권한, 소유자, 크기, 수정 시간 등

```
$ ls -sl cs1.txt
```

```
4 -rw-r--r-- 1 chang cs 2088 4월 16일 13:37 cs1.txt
```

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨

① 파일크기 ② 파일종류 ③ 접근권한 ④ 링크수 ⑤ 사용자 ID ⑥ 그룹 ID ⑦ 파일 크기
⑧ 최종 수정 시간 ⑨ 파일이름

▶ 21

ls 명령어 옵션

● ls -asl

```
$ ls -asl
```

```
총 8
```

```
0 drwxr-xr-x 2 chang cs 20 4월 16일 13:37 .
```

```
4 drwx----- 3 chang cs 4096 4월 16일 13:37 ..
```

```
4 -rw-r--r-- 1 chang cs 2088 4월 16일 13:37 cs1.txt
```

▶ 22

ls 명령어 옵션

- **ls -F**

- 기호로 파일의 종류를 표시
*: 실행파일, /: 디렉터리, @:심볼릭 링크

- 예

```
$ ls -F /  
bin@ dev/ home/ lib64@ mnt/ proc/ run/ srv/ tmp/ var/  
boot/ etc/ lib@ media/ opt/ root/ sbin@ sys/ usr/
```

ls 명령어 옵션

- **ls -R**

- -R(Recursive) 옵션
- 모든 하위 디렉터리 내용을 리스트 한다.

- 예

```
$ ls -R  
$ ls -R /
```

3.5 디렉터리 생성과 삭제

25

디렉터리 생성: mkdir(make directory)

- 사용법

```
$ mkdir [-p] 디렉터리+  
디렉터리들을 새로 만든다
```

- 예

```
$ cd ~ // 홈 디렉터리로 이동  
$ mkdir test  
$ mkdir test temp  
$ ls -l  
drwxrwxr-x. 2 chang chang 6 5월 12 10:12 temp  
drwxrwxr-x. 2 chang chang 6 5월 12 10:12 test
```

디렉터리 생성: mkdir

- 중간 디렉터리 자동 생성 옵션 -p
 - 필요한 경우에 중간 디렉터리를 자동으로 만들어 준다.
- 예 : ~/dest 디렉터리가 없는 경우

```
$ mkdir ~/dest/dir1
```

```
mkdir: '/home/chang/dest/dir1' 디렉터리를 만들 수 없습니다: 그런 파일이나 디렉터리가 없습니다
```

```
$ mkdir -p ~/dest/dir1
```

▶ 27

디렉터리 삭제 : rmdir(remove directory)

- 사용법

```
??rmdir 디렉터리+
```

```
디렉터리들을 삭제한다??
```

- 주의: 빈 디렉토리만 삭제할 수 있다.
- 예

```
$ rmdir test
rmdir: failed to remove 'test': 디렉터리가 비어있지 않음
```

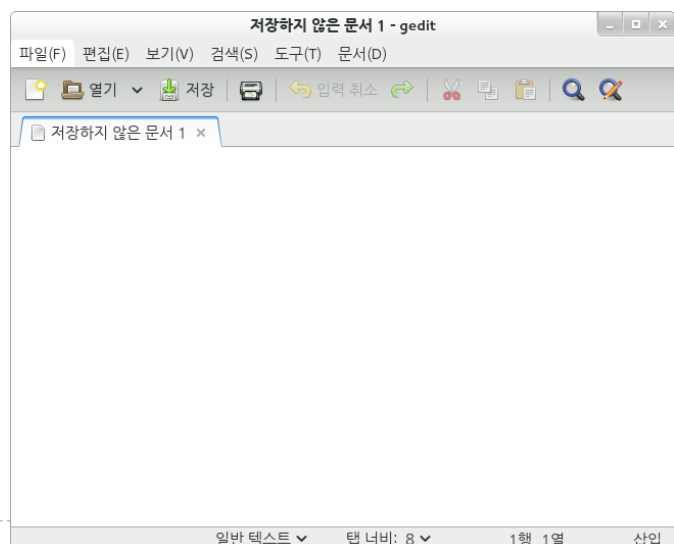
▶ 28

3.6 파일 내용 출력

29

간단한 파일 만들기: gedit

- GNOME이 제공하는 GUI 기반 문서편집기
- 사용방법
 - [프로그램] -> [보조 프로그램] -> [gedit 텍스트 편집기]
 - `$ gedit [파일이름] &`
- 기능
 - 파일: 새로 만들기, 열기, 저장 닫기
 - 편집: 입력취소, 잘라내기, 복사 붙여넣기
 - 보기: 도구모음, 상태표시줄, 전체화면
 - 검색: 찾기, 바꾸기
 - 검사: 맞춤법 검사
 - 문서: 모두 저장, 모두 닫기
 - 도움말



간단한 파일 만들기: cat

- cat 명령어 사용

```
$ cat > 파일
```

표준입력 내용을 모두 파일에 저장한다²⁾파일이 없으면 새로 만든다²⁾

- 예

```
$ cat > cs1.txt
```

```
...
```

```
^D
```

▶ 31

간단한 파일 만들기: touch

- touch 명령어 사용

```
$ touch 파일
```

파일 크기가 0인 이름만 있는 빈 파일을 만들어 준다²⁾

- 예

```
$ touch cs1.txt
```

```
$ ls -asl cs1.txt
```

```
0 -rw-rw-r--. 1 chang chang 0 5월 9 15:10 cs1.txt
```

▶ 32

파일 내용 출력

- 파일 내용 출력과 관련된 다음 명령어들
 - cat, more, head, tail, wc, 등

\$ 명령어 파일

\$ 명령어 파일*

\$ more 파일+

▶ 33

파일 내용 보기: cat

- 사용법

cat [-n] 파일*

파일들¹의 내용을 그대로 화면에 출력한다²파일을 지정하지 않으면 표준입력 내용을 그대로 화면에 출력한다³

- 예

\$ cat cs1.txt

Unix is a multitasking, multi-user computer operating system originally developed in 1969 by a group of AT&T employees at Bell Labs, including Ken Thompson, Dennis Ritchie, Brian Kernighan, Douglas McIlroy, and Joe Ossanna.

...

▶ 34

파일 내용 보기: cat

- 예

```
$ cat -n cs1.txt
```

```
1 Unix is a multitasking, multi-user computer operating system originally
2 developed in 1969 by a group of AT&T employees at Bell Labs, including
3 Ken Thompson, Dennis Ritchie, Brian Kernighan, Douglas McIlroy,
4 and Joe Ossanna.
...
```

```
$ cat          // 지정 파일 없음
```

```
Hello World !
```

```
Hello World !
```

```
Bye!
```

```
Bye!
```

```
^D
```

▶ 35

페이지 단위로 파일 내용 보기: more

- 사용법

```
$ more 파일+
```

파일들의 내용을 페이지 단위로 화면에 출력한다

- 예

```
$ more cs1.txt
```

```
Unix is a multitasking, multi-user computer operating system originally
developed in 1969 by a group of AT&T employees at Bell Labs, including
Ken Thompson, Dennis Ritchie, Brian Kernighan, Douglas McIlroy,
and Joe Ossanna.
```

```
...
```

```
During the late 1970s and early 1980s, the influence of Unix in academic
circles led to large-scale adoption of Unix(particularly of the BSD variant,
--계속--(59%)
```

▶ 36

파일 앞부분보기: head

- 사용법

```
$ head [-n] 파일*
```

파일들의 앞부분을 화면에 출력한다. 파일을 지정하지 않으면 표준입력 내용을 대상으로 한다.

- 예

```
$ head -5 cs1.txt
```

Unix is a multitasking, multi-user computer operating system originally developed in 1969 by a group of AT&T employees at Bell Labs, including Ken Thompson, Dennis Ritchie, Brian Kernighan, Douglas McIlroy, and Joe Ossanna.

▶ 37

파일 뒷부분보기: tail

- 사용법

```
$ tail [-n] 파일*
```

파일들의 뒷부분을 화면에 출력한다. 파일을 지정하지 않으면 표준입력 내용을 대상으로 한다.

- 예

```
$ tail cs1.txt
```

Linux, which is used to power data centers, desktops, mobile phones, and embedded devices such as routers, set-top boxes or e-book readers. Today, in addition to certified Unix systems such as those already mentioned, Unix-like operating systems such as MINIX, Linux, Android, and BSD descendants (FreeBSD, NetBSD, OpenBSD, and DragonFly BSD) are commonly encountered.

The term traditional Unix may be used to describe a Unix or an operating system that has the characteristics of either Version 7 Unix or UNIX System V.

▶ 38

단어 세기: wc(word count)

- 사용법

```
$ wc [-lwc] 파일*
```

파일에 저장된 줄(l) 단어(w) 문자(c)의 개수를 세서 출력한다

파일을 지정하지 않으면 표준입력 내용을 대상으로 한다

- 예

```
$ wc cs1.txt
38 318 2088 cs1.txt
$ wc -l cs1.txt
38 cs1.txt
$ wc -w cs1.txt
318 cs1.txt
$ wc -c cs1.txt
2088 cs1.txt
```

▶ 39

핵심 개념

- 리눅스의 디렉터리는 루트로부터 시작하여 계층구조를 이룬다.
- 절대 경로명은 루트 디렉터리부터 시작하고 상대 경로명은 현재 디렉터리부터 시작한다.

▶ 40