

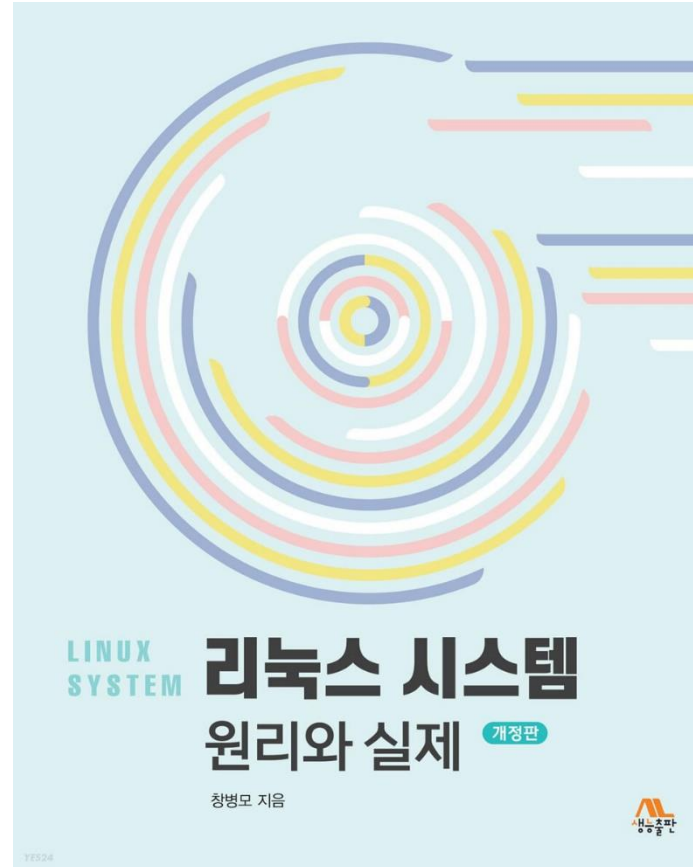
UNIX Basic Commands

Autumn 2024



AI융합학과
Seongbok Baik
sbbaik@dju.kr

00 Textbook & Lecture Note



숙명여대 창병모 교수

01 기본 명령어 사용

- 날짜 및 시간 확인

```
$ date
```

```
2022. 01. 01. (토) 12:26:10 KST
```

- 시스템 정보 확인

```
$ hostname
```

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

```
$ uname
```

```
Linux
```

```
$ uname -a
```

```
Linux Ubuntu 5.11.0-31-generic #33-Ubuntu SMP Wed Aug 11  
13:19:04 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

02 기본 명령어 사용

- 사용자 정보 확인

```
$ whoami  
chang
```

```
$ who  
chang      tty2    2022-01-09 12:19 (tty2)
```

- 디렉터리 내용 확인

```
$ ls  
공개   다운로드   문서   바탕화면   비디오   사진   음악   템플릿
```

03 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

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

05 명령어에 대한 간단한 설명: whatis

\$ whatis ls

ls (1) - 경로의 내용을 나열한다.

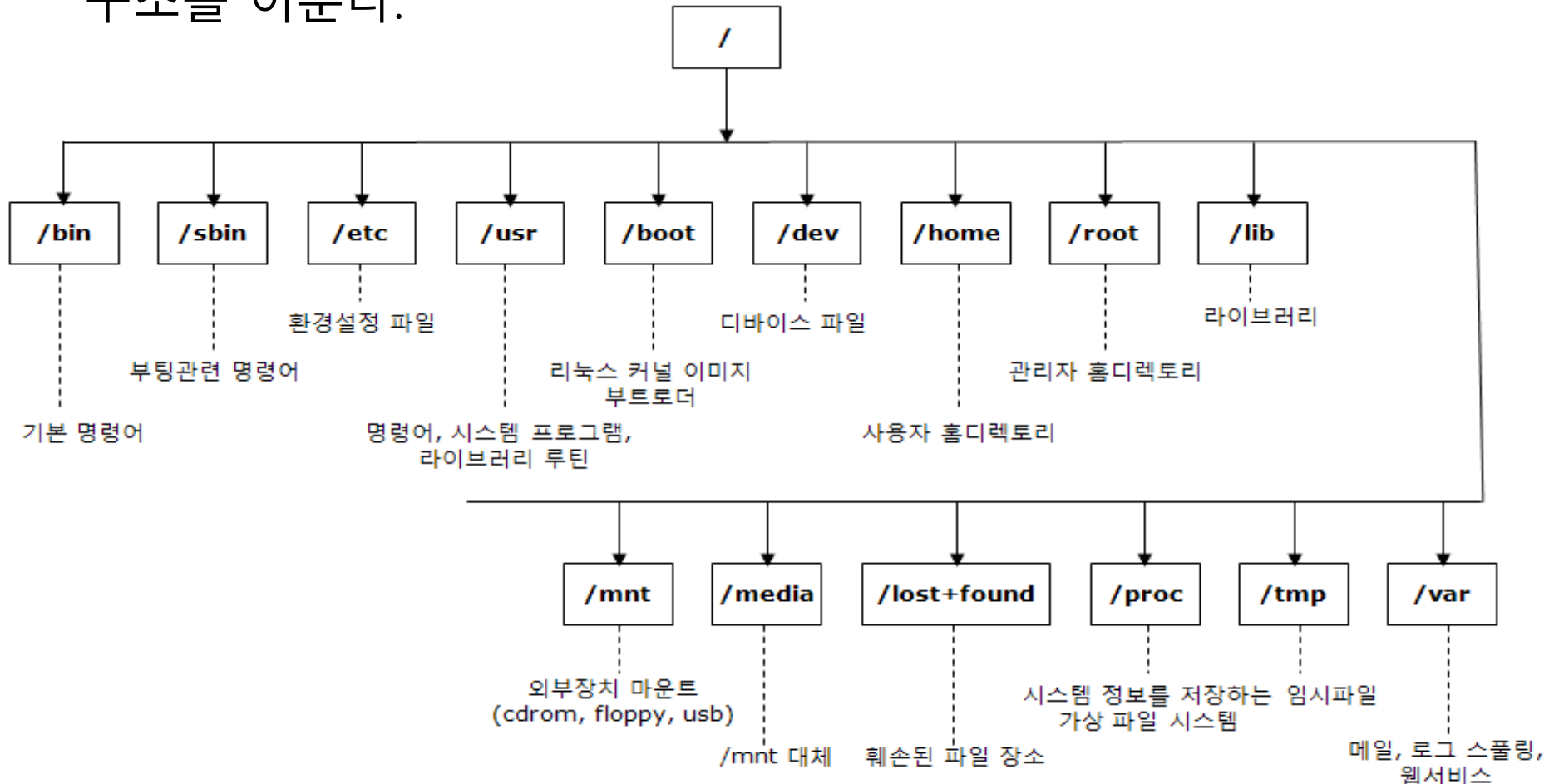
ls (1p) - list directory contents

06 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

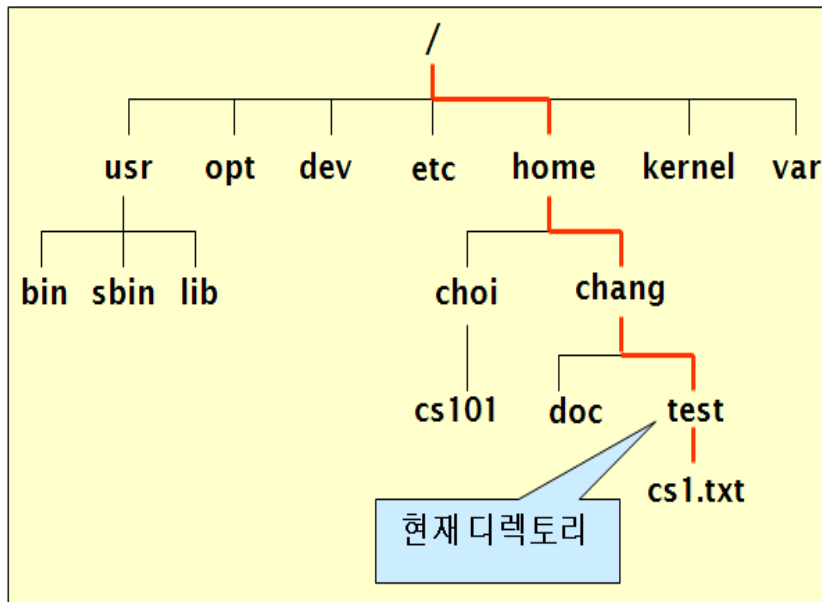
07 디렉터리 계층구조

- 리눅스의 디렉터리는 루트로부터 시작하여 트리 형태의 계층 구조를 이룬다.



08 홈 디렉터리

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



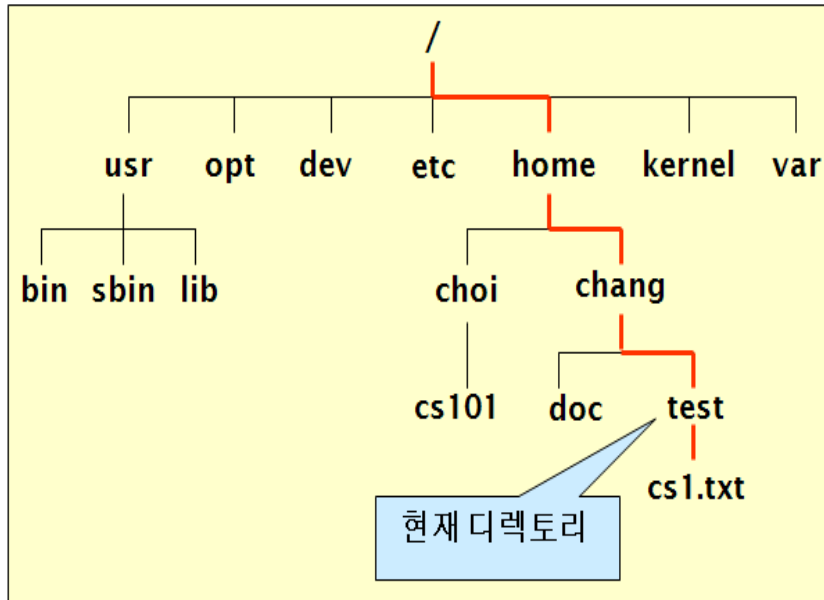
~ : 홈 디렉터리
 . : 현재 디렉터리
 .. : 부모 디렉터리

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

cs1.txt의 상대 경로명
 cs1.txt

09 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

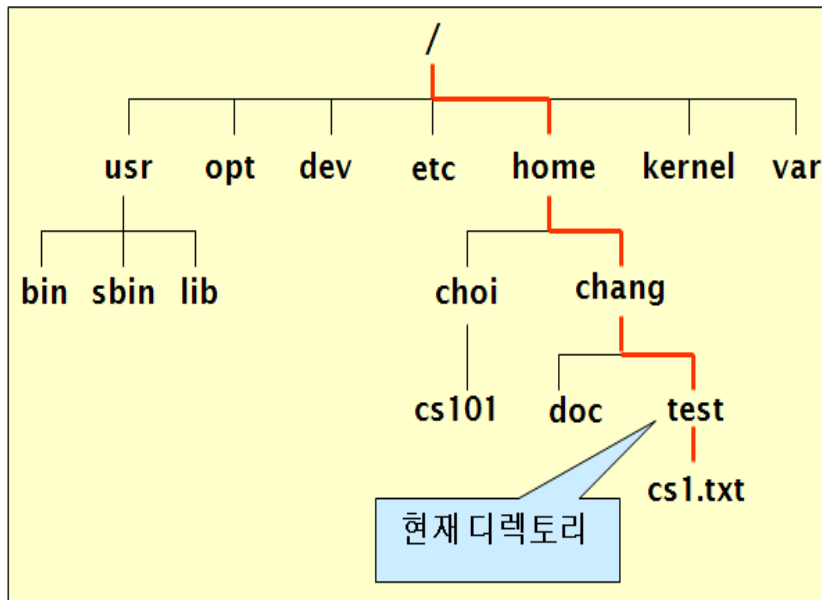


cs1.txt의 절대 경로명
/home/chang/test/cs1.txt

cs1.txt의 상대 경로명
cs1.txt

10 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

11 명령어의 경로 확인: which

- 사용법

```
$ which 명령어
```

명령어의 절대경로를 보여준다.

- 예

```
$ which ls
```

```
/bin/ls
```

```
$ which pwd
```

```
/usr/pwd
```

```
$ which passwd
```

```
/usr/passwd
```

02 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/바탕화면
```

```
$ cd ~
```

```
$ pwd
```

```
/home/chang
```

13 디렉터리 이동: `cd`(change directory)

- 사용법

```
$ cd [디렉터리]
```

현재 작업 디렉터를 지정된 디렉터리로 이동한다.

디렉터를 지정하지 않으면 홈 디렉터리로 이동한다.

- 예

```
$ cd
```

```
$ cd ~
```

```
$ cd 바탕화면
```

```
$ pwd
```

```
/home/chang/바탕화면
```

```
$ cd ..    // 부모 디렉터리로 이동
```

14 디렉터리 생성: `mkdir`(make directory)

- 사용법

```
$ mkdir [-p] 디렉터리+
```

디렉터리(들)을 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
```


15 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' 디렉터를 만들 수 없습니다: 그런 파일이나 디렉터리가 없습니다

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

16 디렉터리 삭제 : rmdir(remove directory)

- 사용법

```
$ rmdir 디렉터리+
```

디렉터리(들)을 삭제한다.

- 주의: 빈 디렉토리만 삭제할 수 있다.

- 예

```
$ rmdir test
```

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

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 ~
```

```
test 공개 다운로드 문서 바탕화면 비디오 사진 음악 템플릿
```

```
$ cd test
```

```
$ ls
```

```
cs1.txt
```

18 ls 명령어 옵션

- 주요 옵션

옵션	기능
-a	숨겨진 파일을 포함하여 모든 파일을 리스팅한다.
-s	파일의 크기를 K 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
```

```
합계 4
```

```
4 cs1.txt
```

● ls -a

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

```
$ ls -a
```

```
. .. cs1.txt
```

20 ls 명령어 옵션

● ls -l(long)

- 파일 속성(file attribute) 출력
- 블록 수, 파일 종류, 접근권한, 링크 수, 소유자명, 크기, 수정 시간, 파일 이름 등

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

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

①② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨

① 블록 수 ② 파일 종류 ③ 접근권한 ④ 링크 수 ⑤ 소유자명 ⑥ 그룹명
⑦ 파일 크기 ⑧ 최종 수정 시간 ⑨ 파일이름

21 ls 명령어 옵션

- **ls -asl**

```
$ ls -asl
```

```
합계 12
```

```
4 drwxr-xr-x 2 chang chang 4906 4월 16일 13:37 .  
4 drwx----- 3 chang chang 4096 4월 16일 13:37 ..  
4 -rw-r--r-- 1 chang chang 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/
```


23 ls 명령어 옵션

- **ls -R**

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

- 예

```
$ ls -R
```

```
$ ls -R /
```

24 간단한 파일 만들기: cat

- cat 명령어 사용

```
$ cat > 파일
```

표준입력 내용을 모두 파일에 저장한다. 파일이 없으면 새로 만든다.

- 예

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

```
...
```

```
^D
```

25 간단한 파일 만들기: touch

- touch 명령어 사용

```
$ touch 파일
```

파일 크기가 0인 이름만 있는 빈 파일을 만들어 준다.

- 예

```
$ touch cs1.txt
```

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

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

26 파일 내용 출력

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

\$ 명령어 파일

\$ 명령어 파일*

27 파일 내용 보기: cat

- 사용법

```
$ cat [-n] 파일*
```

파일(들)의 내용을 그대로 화면에 출력한다. 파일을 지정하지 않으면 표준입력 내용을 그대로 화면에 출력한다.

- 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.

...

28 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
```

29 페이지 단위로 파일 내용 보기: more

- 사용법

```
$ more 파일+
```

파일(들)의 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%)

30 파일 앞부분보기: head

- 사용법

```
$ head [-n] 파일*
```

파일(들)의 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.

31 파일 뒷부분보기: tail

- 사용법

```
$ tail [-n] 파일*
```

파일(들)의 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.

32 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
```

33 주요 내용

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

서로에게, 자신에게 친절합니다 - 허준이

