

목차

- 00. 기본
- 01. 리눅스 파일의 종류와 특징
- 02. 디렉터리 사용 명령
- 03. 파일 사용 명령

그림 1-49 터미널 동작시키기

■ 터미널 동작시키기

- 터미널 창에서 명령을 입력하여 작업 가능

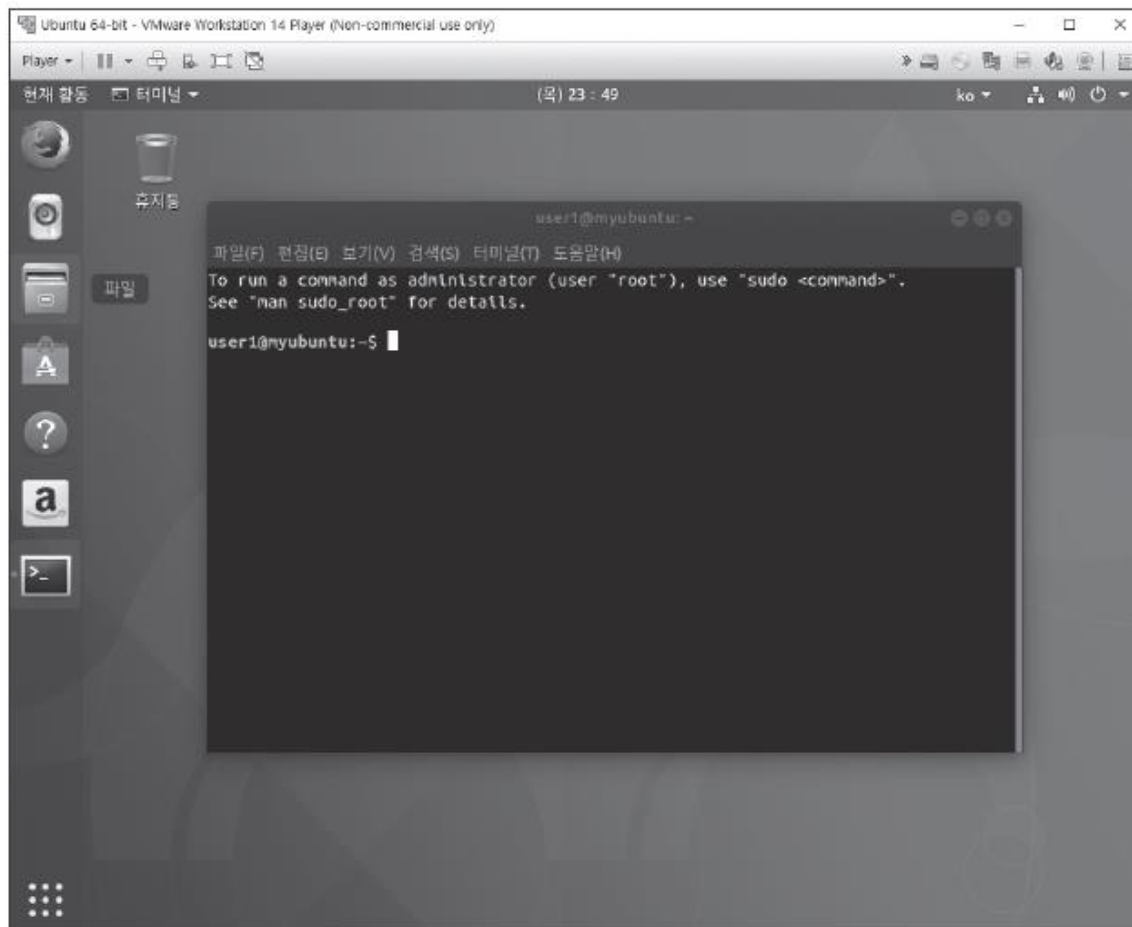


그림 1-50 터미널 창

■ 프롬프트 기호와 홈 디렉터리

- 프롬프트
 - 사용자의 명령 입력을 기다리는 표시
 - 셸에 따라 다르게 나타남: 배시 셸의 경우 \$로 표시, 시스템 관리자의 경우 #으로 표시
- 기본 디렉터리
 - 터미널 창의 기본 디렉터리는 사용자의 홈 디렉터리에 위치
 - 'user1@myubuntu:~\$' : user1은 사용자 계정, myubuntu는 호스트의 이름으로 우분투 시스템 자체를 의미
 - 사용자의 홈 디렉터리는 사용자 계정을 등록할 때 지정

■ 명령 행 편집 방법

- 프롬프트 다음에 명령을 입력하여 리눅스 사용
- 문자 지우기: 백스페이스키(<-)나 Delete키를 사용
- 단어 지우기: Ctrl+w, 단어는 공백 문자로 구분
- 문장 지우기: Ctrl+u

■ 명령의 구조

명령의 구조

- **형식** 명령 [옵션] [인자]
- **명령** 리눅스를 사용하기 위해 사용자가 입력하는 다양한 명령은 `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 있는 경우도 있고 없는 경우도 있다. 옵션은 `-` 또는 `--`로 시작하며 영문 소문자나 대문자로 구성된다. 명령에 따라 어떤 옵션이 있고 그 기능이 무엇인지는 해당 명령의 사용법을 참조한다.
- **인자** 인자는 명령으로 전달되는 값이며 주로 파일명이나 디렉터리명이 사용된다. 명령에 따라 없을 수도 있고 각 명령에 필요한 인자가 각기 다르므로 사용법을 참조해야 한다.

■ 명령 사용법

■ 명령만 사용한 경우

```
user1@myubuntu:~$ ls
examples.desktop  공개  다운로드  문서  바탕화면  비디오  사진  음악  템플릿
user1@myubuntu:~$
```

■ 명령과 옵션을 사용한 예

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

■ 명령과 인자를 사용한 예

```
user1@myubuntu:~$ ls /tmp
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-colord.service-2o0tEo
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-fwupd.service-0hDA3B
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-rtkit-daemon.service-4upSu3
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-systemd-resolved.service-R9KYbi
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-systemd-timesyncd.service-QIA4iH
user1@myubuntu:~$
```

■ 명령 사용법

- 명령과 옵션과 인자를 사용한 예

```
user1@myubuntu:~$ ls -a /tmp
.
..
.ICE-unix
.Test-unix
.X0-lock
.X1024-lock
.X11-unix
.XIM-unix
.font-unix
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-colord.service-2o0tEo
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-fwupd.service-0hDA3B
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-rtkit-daemon.service-4upSu3
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-systemd-resolved.service-R9KYbi
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-systemd-timesyncd.service-QIA4iH
user1@myubuntu:~$
```

■ 기초 명령 사용법

▪ date 명령

- 현재 날짜와 시간을 출력

date

- **기능** 날짜와 시간을 출력한다.
- **형식** date

```
user1@myubuntu:~$ date
2018. 01. 10. (수) 00:07:16 KST
user1@myubuntu:~$
```


■ 기초 명령 사용법

■ clear 명령

- 현재 보이는 화면을 지우고 커서를 화면 좌측 상단으로 이동

`clear`

- **기능** 화면을 지운다.
- **형식** `clear`

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

■ 기초 명령 사용법

▪ man 명령

- 리눅스가 제공하는 각종 명령의 사용법 보여줌

man

- **기능** 명령 사용법을 화면에 출력한다.
- **형식** man 명령

- man 다음에 자세히 알고 싶은 명령 지정

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

■ 기초 명령 사용법

- man 명령의 실행 결과

```
clear(1)
```

```
clear(1)
```

NAME

→ 명령에 대한 간략한 설명

clear - clear the terminal screen

SYNOPSIS

→ 명령의 사용법 요약

clear

DESCRIPTION

→ 명령에 대한 상세한 설명

clear clears your screen if this is possible, including its scrollback buffer (if the extended "E3" capability is defined). clear looks in the environment for the terminal type and then in the terminfo database to determine how to clear the screen.

clear ignores any command-line parameters that may be present.

SEE ALSO

→ 명령과 관련된 참고 사항

tput(1), terminfo(5)

This describes ncurses version 5.9 (patch 20130608).

clear(1)

■ 기초 명령 사용법

- passwd 명령
 - 사용자 계정의 비밀번호 변경

passwd

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


```
user1@myubuntu:~$ passwd
user1에 대한 암호 변경 중
(현재) UNIX 암호:
새 UNIX 암호 입력:
새 UNIX 암호 재입력:
passwd: 암호를 성공적으로 업데이트했습니다
user1@myubuntu:~$
```

■ 터미널 종료하기

- exit 명령을 사용하여 접속 해제

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

- Ctrl+d를 사용한 접속 해제

```
user1@myubuntu:~$ ^d → +d
```

01 리눅스 파일의 종류와 특징

■ 파일의 종류

- 파일: 관련 있는 정보들의 집합
- 리눅스는 파일을 효과적으로 관리하기 위해 디렉터리를 사용
- 리눅스에서 파일은 사용 목적에 따라 일반 파일, 디렉터리, 심벌릭 링크, 장치 파일로 구분

■ 일반 파일(Regular File)

- 데이터를 저장하는 데 주로 사용
- 각종 텍스트 파일, 실행 파일, 이미지 파일 등 리눅스에서 사용하는 대부분의 파일은 일반 파일에 해당
- 텍스트 파일은 문서 편집기를 사용하여 내용을 보거나 편집할 수 있으나, 실행 파일이나 이미지 파일의 경우 해당 파일의 내용을 확인할 수 있는 특정 응용 프로그램이 있어야 확인할 수 있음
- 실행 파일이나 이미지 파일의 경우 바이너리 형태로 데이터가 저장되어 바이너리 파일이라고 함

■ 디렉터리(Directory)

- 리눅스에서는 디렉터리도 파일로 취급
- 디렉터리 파일에는 해당 디렉터리에 저장된 파일이나 하위 디렉터리에 대한 정보가 저장

■ 심벌릭 링크

- 원본 파일을 대신하여 다른 이름으로 파일명을 지정한 것(윈도의 바로가기 파일과 비슷)

■ 장치파일

- 리눅스에서는 하드디스크나 키보드 같은 각종 장치도 파일로 취급

01 리눅스 파일의 종류와 특징

■ 파일의 종류 확인: file 명령

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

01 리눅스 파일의 종류와 특징

■ 디렉터리 계층 구조

- 리눅스에서는 파일을 효율적으로 관리하기 위해 디렉터리를 계층적으로 구성 -> 트리(tree) 구조
- 모든 디렉터리의 출발점은 루트(root, 뿌리) 디렉터리이며, /(빗금)으로 표시

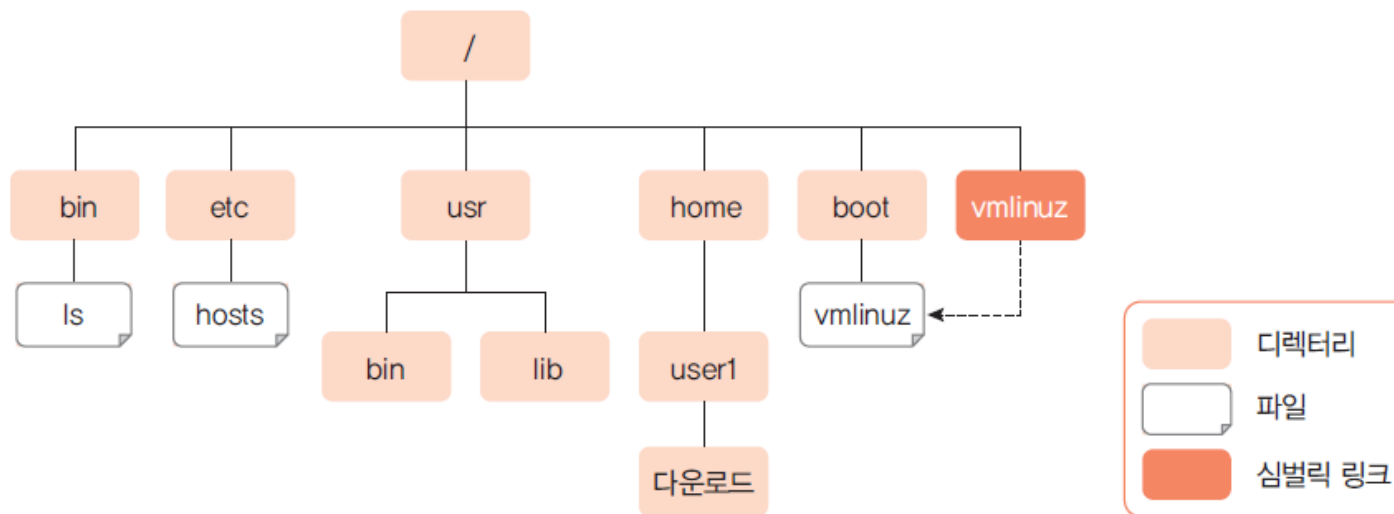


그림 2-2 디렉터리 계층 구조의 예

- 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

01 리눅스 파일의 종류와 특징

루트 디렉터리의 서브 디렉터리

```
user1@myubuntu:~$ ls -F /
bin/      dev/      initrd.img@  lost+found/  opt/      run/      srv/      tmp/      vmlinuz@
boot/     etc/      lib/         media/       proc/     sbin/     swapfile  usr/
cdrom/    home/     lib64/       mnt/         root/     snap/     sys/      var/
user1@myubuntu:~$
```

- /: 해당 파일이 디렉터리임을 표시
- @: 심벌릭 링크

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

01 리눅스 파일의 종류와 특징

표 2-1 디렉터리의 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.

01 리눅스 파일의 종류와 특징

■ 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

01 리눅스 파일의 종류와 특징

■ 현재 디렉터리가 user1일 때

- user1의 절대 경로명: /home/user1
- user1 아래 다운로드의 절대 경로명: /home/user1/다운로드
- '다운로드'의 상대 경로명: 다운로드 또는 ./다운로드
- hosts 파일의 상대 경로명: ../../etc/hosts

디렉터리/파일	절대 경로명	상대 경로명
/		
home		
boot		
lib		
ls		

■ 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

02 디렉터리 사용 명령

■ 현재 디렉터리 확인하기

pwd

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

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

02 디렉터리 사용 명령

■ 디렉터리 이동하기

cd

- **기능** 지정한 디렉터리로 이동한다.
- **형식** cd [디렉터리]
- **사용 예** cd cd /tmp cd 다운로드

- 절대 경로명으로 이동할 디렉터리 지정

```
user1@myubuntu:~$ cd /tmp
user1@myubuntu:/tmp$ pwd
/tmp
user1@myubuntu:/tmp$
```

- 상대 경로명으로 이동할 디렉터리 지정

```
user1@myubuntu:/tmp$ cd ../usr/lib
user1@myubuntu:/usr/lib$ pwd
/usr/lib
user1@myubuntu:/usr/lib$
```

02 디렉터리 사용 명령

■ 홈 디렉터리로 이동하는 방법

- `cd /home/user1`: 절대 경로명을 사용하여 홈 디렉터리로 이동
- `cd ../../home/user1`: 현재 `/usr/lib` 디렉터리에 있으므로 이를 기준으로 상대 경로명을 사용하여 홈 디렉터리로 이동
- `cd ~`: 홈 디렉터를 나타내는 기호인 `~`를 사용하여 홈 디렉터리로 이동
- `cd`: 목적지를 지정하지 않고 `cd` 명령만 사용하면 해당 계정의 홈 디렉터리로 이동
- 이 중 가장 간단한 방법은 `cd` 명령 사용

```
user1@myubuntu:/usr/lib$ cd
user1@myubuntu:~$ pwd
/home/user1
user1@myubuntu:~$
```


02 디렉터리 사용 명령

■ 디렉터리 내용보기

`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, @: 심벌릭 링크).
 - L: 심벌릭 링크 파일의 경우 원본 파일의 정보를 출력한다.
 - R: 하위 디렉터리의 목록까지 출력한다.
- **사용 예** `ls` `ls -F` `ls -al /tmp`

02 디렉터리 사용 명령

■ 현재 디렉터리 내용 확인: ls

- 옵션이나 디렉터를 지정하지 않고 ls 명령만 사용

```
user1@myubuntu:~$ ls
examples.desktop  공개  다운로드  문서  바탕화면  비디오  사진  음악  템플릿
user1@myubuntu:~$
```

■ 숨김 파일 확인: ls -a

- 리눅스에서 .으로 시작하면 숨김 파일이며 그냥 ls 명령으로는 볼 수 없음
- -a 옵션을 사용하면 숨김 파일 확인 가능

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

02 디렉터리 사용 명령

■ 파일의 종류 표시: ls -F

- -F 옵션 : 파일의 종류를 구분하여 표시
- /: 디렉터리, @: 심벌릭 링크, *: 실행파일, 표시 없음: 일반파일

```
user1@myubuntu:~$ ls -F
examples.desktop  다운로드/  바탕화면/  사진/  템플릿/
공개/             문서/      비디오/   음악/
user1@myubuntu:~$
```

- -a 옵션과 연결하여 사용

```
user1@myubuntu:~$ ls -aF
./          .bashrc    .mozilla/  공개/      비디오/
../         .cache/    .profile   다운로드/  사진/
.ICEauthority .config/   .sudo_as_admin_successful 문서/      음악/
.bash_logout .local/    examples.desktop 바탕화면/  템플릿/
user1@myubuntu:~$
```

02 디렉터리 사용 명령

■ 지정한 디렉터리 내용 출력하기

- 인자로 디렉터리 지정하면 해당 디렉터리 내용을 출력

```
user1@myubuntu:~$ ls /tmp
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-colord.service-2o0tEo
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-fwupd.service-0hDA3B
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-rtkit-daemon.service-4upSu3
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-systemd-resolved.service-R9KYbi
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-systemd-timesyncd.service-QIA4iH
user1@myubuntu:~$
```

- 옵션과 인자를 함께 사용: -F 옵션

```
user1@myubuntu:~$ ls -F /tmp
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-colord.service-2o0tEo/
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-fwupd.service-0hDA3B/
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-rtkit-daemon.service-4upSu3/
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-systemd-resolved.service-R9KYbi/
systemd-private-afe5e884236d4e96919e4a7f84cd37f0-systemd-timesyncd.service-QIA4iH/
user1@myubuntu:~$
```

02 디렉터리 사용 명령

■ 상세한 정보 출력하기: -l 옵션

- 디렉터리에 있는 파일들의 상세 정보 출력

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

합계 44

```
-rw-r--r--  1 user1 user1 8980 11월  8 23:15 examples.desktop
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 공개
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 다운로드
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 문서
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 바탕화면
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 비디오
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 사진
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 음악
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 템플릿
user1@myubuntu:~$
```

02 디렉터리 사용 명령

■ 상세한 정보 출력하기: -l 옵션

- 출력된 정보의 의미

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)

02 디렉터리 사용 명령

■ 디렉터리 자체 정보 확인: -d 옵션

- 디렉터리의 자체 정보 출력

```
user1@myubuntu:~$ ls -l /
합계 1214992
drwxr-xr-x  2 root root      4096 11월 10 00:05 bin
drwxr-xr-x  3 root root      4096 11월 10 00:04 boot
drwxrwxr-x  2 root root      4096 11월  8 23:13 cdrom
drwxr-xr-x 18 root root     3900 11월 10 00:02 dev
drwxr-xr-x 121 root root    12288 11월 11 10:28 etc
(생략)
user1@myubuntu:~$ ls -ld /
drwxr-xr-x 24 root root 4096 11월  8 23:16 /
user1@myubuntu:~$
```

■ 파일 존재 확인

- 인자로 지정한 파일이 없으면 없다는 메시지 출력

```
user1@myubuntu:~$ ls .bashrc
.bashrc
user1@myubuntu:~$ ls game
ls: 'game'에 접근할 수 없습니다: 그런 파일이나 디렉터리가 없습니다
user1@myubuntu:~$
```

02 디렉터리 사용 명령

■ ls 명령의 심벌릭 링크 : dir, vdir

- 윈도우의 cmd 창에서 사용하는 명령과 동일

```
user1@myubuntu:~$ dir
examples.desktop  공개  다운로드  문서  바탕화면  비디오  사진  음악  템플릿
user1@myubuntu:~$ vdir
합계 44
-rw-r--r--  1 user1 user1 8980 11월  8 23:15 examples.desktop
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 공개
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 다운로드
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 문서
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 바탕화면
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 비디오
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 사진
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 음악
drwxr-xr-x  2 user1 user1 4096 11월  8 23:24 템플릿
user1@myubuntu:~$
```


02 디렉터리 사용 명령

■ 디렉터리 만들기

`mkdir`

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

02 디렉터리 사용 명령

■ 디렉터리 한 개 만들기

- 디렉터리를 한 개만 만들려면 mkdir 명령에 인자로 생성하려는 디렉터리명을 지정
- 디렉터리명은 상대 경로명이나 절대 경로명으로 지정

```
user1@myubuntu:~$ mkdir temp
user1@myubuntu:~$ ls temp
user1@myubuntu:~$ ls
examples.desktop  공개      문서      비디오   음악
temp              다운로드  바탕화면  사진     템플릿
user1@myubuntu:~$
```

■ 동시에 디렉터리 여러 개 만들기

- 디렉터리 이름을 여러 개 지정하면 동시에 만들 수 있음
- 디렉터리 이름은 공백문자로 구분

```
user1@myubuntu:~$ mkdir tmp1 tmp2 tmp3
user1@myubuntu:~$ ls
examples.desktop  tmp1  tmp3  다운로드  바탕화면  사진  템플릿
temp              tmp2  공개  문서      비디오   음악
user1@myubuntu:~$
```

02 디렉터리 사용 명령

■ 중간 디렉터리 자동으로 만들기 : -p 옵션

- -p 옵션: 디렉터리명으로 지정한 경로 중 중간 단계의 디렉터리가 없을 경우 자동으로 중간 단계 디렉터리를 생성한 후 최종 디렉터리를 생성
- 예: 경로에서 중간 단계 디렉터리가 없으므로 디렉터리를 생성 못함

```
user1@myubuntu:~$ mkdir temp/mid/han
mkdir: 'temp/mid/han' 디렉터리를 만들 수 없습니다: 그런 파일이나 디렉터리가 없습니다
user1@myubuntu:~$
```

- 예: -p 옵션 사용

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

temp/mid:
han

temp/mid/han:
user1@myubuntu:~$
```

02 디렉터리 사용 명령

■ 디렉터리 삭제하기

rmmdir

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

- 예: tmp3 디렉터리 삭제

```
user1@myubuntu:~$ rmmdir tmp3
user1@myubuntu:~$ ls
examples.desktop  tmp1  공개      문서      비디오  음악
temp              tmp2  다운로드  바탕화면  사진    템플릿
user1@myubuntu:~$
```

- 디렉터리가 비어있지 않으면 삭제 불가

```
user1@myubuntu:~$ rmmdir temp
rmmdir: failed to remove 'temp': 디렉터리가 비어 있지 않음
user1@myubuntu:~$
```

■ 실습

- ① 현재 위치를 확인한다. 홈 디렉터리가 아니면 홈 디렉터리로 이동한다.
- ② 실습을 위한 기본 디렉터를 만든다. 먼저 홈 디렉터리에 `linux_ex` 디렉터를 만들고 그 디렉터리로 이동한다. 앞으로 모든 실습은 이 디렉터리 아래에서 한다.
- ③ 2장의 실습이니 `ch2` 디렉터를 만들고 그 디렉터리로 이동하여 현재 위치를 알아본다.
- ④ `one`, `two`, `three` 디렉터를 동시에 만든다.
- ⑤ `one` 디렉터리 아래에 `tmp/test` 디렉터를 만든다. 중간 경로인 `tmp` 디렉터리가 자동 생성되도록 한다.
- ⑥ `one` 디렉터를 `rmdir` 명령으로 삭제해본다. 삭제되는가?
- ⑦ `two`, `three` 디렉터를 동시에 삭제한다.
- ⑧ 실습을 마치고 홈 디렉터리로 이동한다.

03 파일 다루기

- 파일의 내용을 보는 명령
- 파일을 복사하는 명령
- 파일을 삭제하고 이동하는 명령
- 하드 링크와 심벌릭 링크를 생성하는 명령
- 빈 파일을 만드는 명령

03 파일 다루기

■ 파일 내용 연속 출력하기

- 텍스트 파일 내용 확인

cat

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

- 예: /etc/hosts 파일 내용 확인

```
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
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
user1@myubuntu:~$
```

■ 파일 내용 연속 출력하기

- 예: 행 번호 붙이기(-n 옵션)

```
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 IPv6 capable hosts
 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
user1@myubuntu:~$
```


03 파일 다루기

■ 화면 단위로 파일 내용 출력하기

more

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

- 아직 출력되지 않은 내용이 더 있으면 화면 하단에 '--More--(0%)'와 같이 표시
- 예: /etc/services 파일 내용 보기

```
user1@myubuntu:~$ more /etc/services
# Network services, Internet style
(생략)
msp                18/tcp                # message send protocol
--More--(4%)
```

- 스페이스바: 다음 화면 출력, 엔터키: 한 줄씩 스크롤, /문자열: 해당 문자열 검색, q: 종료

03 파일 다루기

■ 개선된 화면 단위 파일 내용 출력하기

- 스크롤 되어 지나간 내용도 확인 가능

less

- **기능** 파일 내용을 화면 단위로 출력한다.
- **형식** less 파일
- **사용 예** less file1

- 예: /etc/services

```
user1@myubuntu:~$ less /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)
daytime      13/tcp
daytime      13/udp
netstat      15/tcp
gotd         17/tcp      quote
msp          18/tcp                        # message send protocol
/etc/services
```

표 2-4 less 명령에서 사용하는 키와 동작

키	동작
j	한 줄씩 다음 행으로 스크롤한다.
k	한 줄씩 이전 행으로 스크롤한다.
Space Bar , Ctrl +f	다음 화면으로 이동한다.
Ctrl +b	이전 화면으로 이동한다.

■ 파일 뒷부분 출력하기

tail

- **기능** 파일 뒷부분의 몇 행을 출력한다.
- **형식** tail [옵션] 파일
- **옵션** +행 번호: 지정한 행부터 끝까지 출력한다.
 - 숫자: 화면에 출력할 행의 수를 지정한다(기본값은 10).
 - f: 파일 출력을 종료하지 않고 주기적으로 계속 출력한다.

- 예: /etc/services 파일의 마지막 10행 출력

```
user1@myubuntu:~$ tail /etc/services
vboxd          20012/udp
binkp          24554/tcp      # binkp fidonet protocol
asp            27374/tcp      # Address Search Protocol
asp            27374/udp
csync2         30865/tcp      # cluster synchronization tool
dirproxy       57000/tcp      # Detachable IRC Proxy
tfido          60177/tcp      # fidonet EMSI over telnet
fido           60179/tcp      # fidonet EMSI over TCP

# Local services
user1@myubuntu:~$
```

03 파일 다루기

■ 파일 뒷부분 출력하기

- 지정한 숫자만큼 출력하기 : -n 숫자 옵션
- 예: /etc/services 파일의 마지막 7

```
user1@myubuntu:~$ tail -7 /etc/services
asp                27374/udp
csync2             30865/tcp          # cluster synchronization tool
dircproxy          57000/tcp          # Detachable IRC Proxy
tfido              60177/tcp          # fidonet EMSI over telnet
fido               60179/tcp          # fidonet EMSI over TCP

# Local services
user1@myubuntu:~$
```

- 파일 내용을 주기적으로 반복 출력하기 : -f 옵션
 - -f 옵션을 사용하면 파일 출력이 종료되지 않고 대기 상태가 되며 파일 내용이 주기적으로 반복 출력

```
user1@myubuntu:~$ tail -f /etc/services
vboxd              20012/udp
(생략)
# Local services
^C
user1@myubuntu:~$
```

03 파일 다루기

■ 파일(디렉터리) 복사하기

cp

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

■ 두 인자가 모두 파일인 경우 : 파일을 다른 파일로 복사

- 예: /etc/hosts 파일을 현재 디렉터리에 text1 파일로 복사

```
user1@myubuntu:~/linux_ex/ch2$ ls
one
user1@myubuntu:~/linux_ex/ch2$ cp /etc/hosts text1
user1@myubuntu:~/linux_ex/ch2$ ls
one  text1
user1@myubuntu:~/linux_ex/ch2$
```

03 파일 다루기

■ 두 번째 인자가 디렉터리인 경우

- 파일을 해당 디렉터리 아래에 복사
- 예: temp 디렉터리에 text1 파일 복사

```
user1@myubuntu:~/linux_ex/ch2$ mkdir temp
user1@myubuntu:~/linux_ex/ch2$ cp text1 temp
user1@myubuntu:~/linux_ex/ch2$ ls temp
text1
user1@myubuntu:~/linux_ex/ch2$
```

- 예: 원본 파일과 다른 이름으로 복사

```
user1@myubuntu:~/linux_ex/ch2$ cp text1 temp/text2
user1@myubuntu:~/linux_ex/ch2$ ls temp
text1  text2
user1@myubuntu:~/linux_ex/ch2$
```

- 예: 쓰기 권한이 없는 디렉터리에 파일을 복사하려고 하면 다음과 같은 오류가 발생

```
user1@myubuntu:~/linux_ex/ch2$ cp text1 /etc
cp: 일반 파일 '/etc/text1'을(를) 생성할 수 없음: 허가 거부
user1@myubuntu:~/linux_ex/ch2$
```

■ 인자를 여러 개 지정할 경우

- cp 명령에서 첫 번째 인자의 자리에 파일명을 여러 개 지정할 수 있는데, 두 번째 인자는 반드시 디렉터리여야 한다.
- 이럴 경우 마지막에 지정한 디렉터리로 앞서 지정한 파일들이 모두 복사된다.
- 예: /etc/hosts와 /etc/services를 temp 디렉터리에 복사

```
user1@myubuntu:~/linux_ex/ch2$ cp /etc/hosts /etc/services temp
user1@myubuntu:~/linux_ex/ch2$ ls temp
hosts  services  text1  text2
user1@myubuntu:~/linux_ex/ch2$
```

■ -i 옵션 사용하기

- 두 번째 인자로 지정한 파일이 이미 있는 경우 덮어서 복사할 것인지 확인

```
user1@myubuntu:~/linux_ex/ch2$ cp -i /etc/hosts text1
cp: 'text1'를 덮어쓸까요? n
user1@myubuntu:~/linux_ex/ch2$
```

■ 디렉터리 복사하기

- 디렉터리를 복사하려면 -r 옵션 사용
- 예: -r 옵션을 지정하지 않을 경우

```
user1@myubuntu:~/linux_ex/ch2$ cp temp temp2
cp: -r not specified; omitting directory 'temp'
user1@myubuntu:~/linux_ex/ch2$
```

- 두 번째 인자로 지정한 목적지 디렉터리가 존재하지 않는 경우 새로 생성
- 디렉터리가 복사되면 원본 디렉터리 아래에 있던 모든 내용도 함께 복사
- 예: temp 디렉터를 temp2 디렉터리로 복사

```
user1@myubuntu:~/linux_ex/ch2$ cp -r temp temp2
user1@myubuntu:~/linux_ex/ch2$ ls -F temp2
hosts  services  temp/  text1  text2
user1@myubuntu:~/linux_ex/ch2$
```


■ 파일 이동하고 파일명 바꾸기

mv

- **기능** 파일을 이동한다.
- **형식** mv [옵션] 파일1(디렉터리1) 파일2(디렉터리2)
- **옵션** -i: 파일2(디렉터리2)가 존재하면 덮어쓸 것인지 물어본다.
- **사용 예** mv file1 file2

- 파일을 다른 디렉터리로 이동하거나 파일명을 바꿀 때는 mv(move) 명령을 사용
- 디렉터리를 이동하거나 디렉터리명을 바꿀 때도 mv 명령을 사용
- mv 명령의 첫 번째 인자는 원본 파일명이나 디렉터리명을 지정하며, 두 번째 인자는 목적지 파일명이나 디렉터리명을 지정

■ 파일을 파일로 이동하기(파일명 바꾸기)

- 파일을 다른 파일로 이동하는 것은 결국 원본 파일의 파일명을 다른 파일명으로 바꾸는 것
- 만약 두 번째 인자로 지정한 파일명이 이미 존재하는 파일이면 원본 파일의 내용으로 덮어쓰고 기존의 내용이 삭제
- 두 번째 인자로 지정한 파일명이 존재하지 않는 파일이라면 새 파일이 생성
- 예: text1 파일을 data1 파일로 이동(파일명 변경)

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

03 파일 다루기

■ 파일을 다른 디렉터리로 이동하기

- 두 번째 인자로 디렉터리를 지정할 경우 원본 파일을 지정한 디렉터리로 이동
- 예: data1 파일을 temp 디렉터리로 이동

```
user1@myubuntu:~/linux_ex/ch2$ mv data1 temp
user1@myubuntu:~/linux_ex/ch2$ ls
one temp temp2
user1@myubuntu:~/linux_ex/ch2$ ls temp
data1 hosts services text1 text2
user1@myubuntu:~/linux_ex/ch2$
```

- 두 번째 인자에 디렉터리와 파일명을 함께 지정할 경우, 파일이 지정한 디렉터리로 이동하면 파일명도 바뀌게 됨

```
user1@myubuntu:~/linux_ex/ch2$ cp temp/data1 text1
user1@myubuntu:~/linux_ex/ch2$ ls
one temp temp2 text1
user1@myubuntu:~/linux_ex/ch2$ mv text1 temp/data2
user1@myubuntu:~/linux_ex/ch2$ ls temp
data1 data2 hosts services text1 text2
user1@myubuntu:~/linux_ex/ch2$
```

- 쓰기 권한이 없는 디렉터리로 파일을 이동하려고 할 경우 오류 발생

```
user1@myubuntu:~/linux_ex/ch2$ mv temp/data2 /etc
mv: 'temp/data2'를 '/etc/data2'로 옮길 수 없음: 허가 거부
user1@myubuntu:~/linux_ex/ch2$
```

03 파일 다루기

■ 파일 여러 개를 디렉터리로 이동하기

- mv 명령으로 파일 여러 개를 지정한 디렉터리로 한 번에 이동 가능
- 두 번째 인자는 반드시 디렉터리여야 함

```
user1@myubuntu:~/linux_ex/ch2$ ls temp
data1 data2 hosts services text1 text2
user1@myubuntu:~/linux_ex/ch2$ mv temp/data1 temp/data2 .
user1@myubuntu:~/linux_ex/ch2$ ls
data1 data2 one temp temp2
user1@myubuntu:~/linux_ex/ch2$ ls temp
hosts services text1 text2
user1@myubuntu:~/linux_ex/ch2$
```

■ -i 옵션 사용하기

- 두 번째 인자에 지정한 파일명이 기존에 있는 파일일 경우 덮어서 이동할 것인지를 물어봄

```
user1@myubuntu:~/linux_ex/ch2$ mv -i data1 data2
mv: 'data2'를 덮어쓸까요? n
user1@myubuntu:~/linux_ex/ch2$ ls
data1 data2 one temp temp2
user1@myubuntu:~/linux_ex/ch2$
```

■ 디렉터리를 디렉터리로 이동하기(디렉터리명 바꾸기)

- 인자를 모두 디렉터리로 지정하면 디렉터리가 이동
- 두 번째 인자가 기존에 있던 디렉터리가 아닐 경우에는 디렉터리명이 변경
- 예: temp2 디렉터리가 temp3 디렉터리로 이름 변경

```
user1@myubuntu:~/linux_ex/ch2$ mv temp2 temp3
user1@myubuntu:~/linux_ex/ch2$ ls
data1 data2 one temp temp3
user1@myubuntu:~/linux_ex/ch2$
```

- 두 번째 인자가 기존에 있던 디렉터리일 경우, 원본 디렉터리가 두 번째 인자로 지정된 디렉터리 아래로 이동
- 예: temp3 디렉터리가 temp 디렉터리 아래로 이동

```
user1@myubuntu:~/linux_ex/ch2$ ls
data1 data2 one temp temp3
user1@myubuntu:~/linux_ex/ch2$ mv temp3 temp
user1@myubuntu:~/linux_ex/ch2$ ls
data1 data2 one temp
user1@myubuntu:~/linux_ex/ch2$ ls temp
hosts services temp3 text1 text2
user1@myubuntu:~/linux_ex/ch2$
```

■ 파일 삭제하기

rm

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

- 삭제할 파일을 인자로 지정하면 해당 파일이 삭제
- 바로 삭제되어 복구할 수도 없으므로 파일을 삭제할 때는 신중해야 함
- 예: data2 파일 삭제

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

■ -i 옵션 사용하기

- -i 옵션을 지정하고 rm 명령을 사용하면 정말 삭제할 것인지 물어봄

```
user1@myubuntu:~/linux_ex/ch2$ rm -i data1
rm: 일반 파일 'data1'를 제거할까요? n
user1@myubuntu:~/linux_ex/ch2$ ls
data1  one  temp
user1@myubuntu:~/linux_ex/ch2$
```

■ 디렉터리 삭제하기

- rm 명령으로 디렉터를 지울 때는 -r 옵션을 지정(삭제된 디렉터리는 복구 불가능)
- 예: -r 옵션을 지정하지 않을 경우 오류 메시지 출력

```
user1@myubuntu:~/linux_ex/ch2$ cd temp
user1@myubuntu:~/linux_ex/ch2/temp$ ls
hosts  services  temp3  text1  text2
user1@myubuntu:~/linux_ex/ch2/temp$ rm temp3
rm: 'temp3'를 지울 수 없음: 디렉터리입니다
user1@myubuntu:~/linux_ex/ch2/temp$
```

■ 디렉터리 삭제하기

- 예: rmdir 명령으로 temp3을 삭제하려고 하면 temp3 디렉터리가 비어 있지 않다고 오류 메시지 출력

```
user1@myubuntu:~/linux_ex/ch2/temp$ rmdir temp3
rmdir: failed to remove 'temp3': 디렉터리가 비어 있지 않음
user1@myubuntu:~/linux_ex/ch2/temp$ ls temp3
hosts  services  temp  text1  text2
user1@myubuntu:~/linux_ex/ch2/temp$
```

- 예: -r 옵션 지정

```
user1@myubuntu:~/linux_ex/ch2/temp$ ls
hosts  services  temp3  text1  text2
user1@myubuntu:~/linux_ex/ch2/temp$ rm -r temp3
user1@myubuntu:~/linux_ex/ch2/temp$ ls
hosts  services  text1  text2
user1@myubuntu:~/linux_ex/ch2/temp$
```


■ 디렉터리 삭제하기

- -i 옵션을 사용: 삭제하려는 디렉터리 아래에 있는 파일이나 서브 디렉터리를 삭제할 것인지 계속 물어봄

```
user1@myubuntu:~/linux_ex/ch2/temp$ cd ..
user1@myubuntu:~/linux_ex/ch2$ ls
data1  one  temp
user1@myubuntu:~/linux_ex/ch2$ rm -ri temp
rm: 'temp' 디렉터리로 내려가겠습니까? y
rm: 일반 파일 'temp/services'를 제거할까요? y
rm: 일반 파일 'temp/hosts'를 제거할까요? n
rm: 일반 파일 'temp/text1'를 제거할까요? n
rm: 일반 파일 'temp/text2'를 제거할까요? n
rm: 디렉토리 'temp'를 제거할까요? n
user1@myubuntu:~/linux_ex/ch2$ ls temp
hosts  text1  text2
user1@myubuntu:~/linux_ex/ch2$
```

03 파일 다루기

■ 파일 링크

- 파일 링크는 기존에 있는 파일에 새로운 파일명을 붙이는 것
- 하드링크: 기존 파일에 새로운 파일명을 추가로 생성
- 심벌릭 링크: 원본 파일을 가리키는 새로운 파일을 생성

■ 리눅스 파일의 구성

- 파일 = 파일명+inode+데이터 블록
 - 파일명: 사용자가 파일에 접근할 때 사용하는 파일의 이름
 - inode: 파일에 대한 정보를 가지고 있는 특별한 구조체로서 외부적으로는 번호로 표시되고, 내부적으로는 파일의 종류 및 크기, 소유자, 파일 변경 시간, 파일명 등 파일 상세 정보와 데이터 블록의 주소를 저장
- 파일의 inode 번호는 `ls -li` 명령으로 확인 가능
 - 파일명 앞에 출력된 숫자가 inode 번호
 - 파일 이름은 다르지만 inode 번호가 같다면 같은 파일

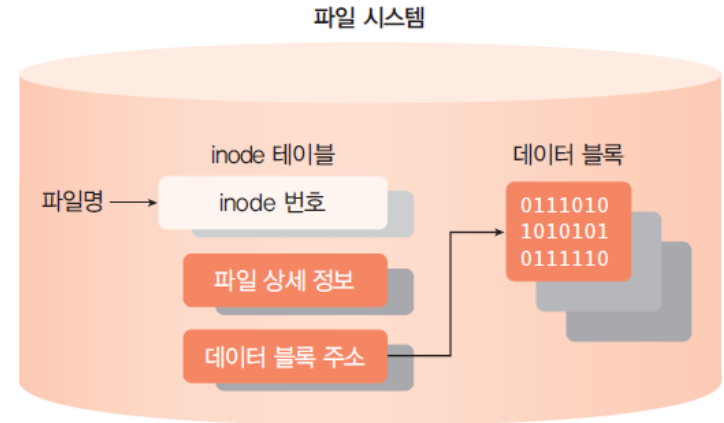


그림 2-3 리눅스 파일의 구성 요소

```
user1@myubuntu:~/linux_ex/ch2$ ls -li
655669 data1  655672 temp  655678 test.org
user1@myubuntu:~/linux_ex/ch2$
```

03 파일 다루기

■ 하드 링크 만들기 : ln

ln

- **기능** 파일의 링크를 생성한다.
- **형식** ln [옵션] 원본 파일 링크 파일
- **옵션** -s: 심벌릭 링크 파일을 생성한다.
- **사용 예** ln test lntest
ln -s test lntest

- 파일에 여러 개의 이름을 붙일 수 있는데, 이때 붙이는 파일명을 하드 링크라고 함
- 예: data1에 대한 하드 링크로 data1.ln 생성

```
user1@myubuntu:~/linux_ex/ch2$ ls -l
합계 12
-rw-r--r--  1 user1 user1  223 11월 11 22:29 data1
drwxrwxr-x  2 user1 user1 4096 11월 11 23:00 temp
-rw-r--r--  1 user1 user1  223 11월 11 23:37 test.org
user1@myubuntu:~/linux_ex/ch2$ ln data1 data1.ln
user1@myubuntu:~/linux_ex/ch2$ ls -l
합계 16
-rw-r--r--  2 user1 user1  223 11월 11 22:29 data1
-rw-r--r--  2 user1 user1  223 11월 11 22:29 data1.ln
drwxrwxr-x  2 user1 user1 4096 11월 11 23:00 temp
-rw-r--r--  1 user1 user1  223 11월 11 23:37 test.org
user1@myubuntu:~/linux_ex/ch2$
```

■ 하드 링크 만들기 : ln

- inode 비교: 다음 예에서 inode가 같은 파일은?

```
user1@myubuntu:~/linux_ex/ch2$ ls -li
655669 data1 655669 data1.ln 655672 temp 655678 test.org
user1@myubuntu:~/linux_ex/ch2$
```

03 파일 다루기

■ 하드 링크 만들기 : ln

- 파일의 내용도 동일

```
user1@myubuntu:~/linux_ex/ch2$ cat data1
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
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
user1@myubuntu:~/linux_ex/ch2$ cat data1.ln
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
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
user1@myubuntu:~/linux_ex/ch2$
```

■ 하드 링크 만들기 : ln

- 링크와 복사의 차이: 무엇이 다른가?

```
user1@myubuntu:~/linux_ex/ch2$ cp data1 data1.cp
user1@myubuntu:~/linux_ex/ch2$ ls -li
655669 data1 655472 data1.cp 655669 data1.ln 655672 temp 655678 test.org
user1@myubuntu:~/linux_ex/ch2$
```

- data1.cp 파일의 내용을 수정해도 data1 파일에는 반영 안 됨

■ 심벌릭 링크 만들기 : -s 옵션

- 예: data1 파일의 심벌릭 링크로 data1.sl을 생성

```
user1@myubuntu:~/linux_ex/ch2$ ln -s data1 data1.sl
user1@myubuntu:~/linux_ex/ch2$ ls -li
655669 data1          655669 data1.ln  655672 temp
655472 data1.cp      655671 data1.sl  655678 test.org
user1@myubuntu:~/linux_ex/ch2$
```

- 심벌릭 링크의 inode 번호를 보면 원본 파일과 다른 번호
- ls -li 명령으로 확인해보면 파일의 종류가 'l(소문자 L)'로 표시되고 파일명도 '->'를 사용하여 원본 파일이 무엇인지를 알려줌

```
user1@myubuntu:~/linux_ex/ch2$ ls -li data1.sl
lrwxrwxrwx 1 user1 user1 5 11월 12 11:26 data1.sl -> data1
user1@myubuntu:~/linux_ex/ch2$
```

■ 심벌릭 링크와 하드 링크의 차이

- 심벌릭 링크는 하드 링크와 비교하여 다음과 같은 몇 가지 특징이 있다.
 - 파일의 종류가 l(소문자 L)로 표시된다.
 - 하드 링크의 개수가 증가하지 않는다. 즉 원본 파일에 이름을 추가하는 것이 아니다.
 - 파일명 뒤에 원본 파일명이 표시된다(-> data1).
 - inode 번호가 원본 파일과 다르다. 즉 원본 파일과 심벌릭 링크 파일은 별개의 파일이다.
- 심벌릭 링크 파일의 내용은 원본 파일의 경로
- 심벌릭 링크에서는 원본 파일이 삭제되면 심벌릭 링크로 연결할 수 없다는 점을 주의해야 한다

```
user1@myubuntu:~/linux_ex/ch2$ rm data1
user1@myubuntu:~/linux_ex/ch2$ cat data1.sl
cat: data1.sl: 그런 파일이나 디렉터리가 없습니다
user1@myubuntu:~/linux_ex/ch2$
```


■ 빈 파일 생성하기, 수정 시간 변경하기 : touch

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

- 인자를 지정하지 않으면 빈 파일 생성

```
user1@myubuntu:~/linux_ex/ch2$ touch test
user1@myubuntu:~/linux_ex/ch2$ ls -l test
-rw-rw-r-- 1 user1 user1 0 11월 12 11:38 test
user1@myubuntu:~/linux_ex/ch2$
```

■ 빈 파일 생성하기, 수정 시간 변경하기 : touch

- 이미 있는 파일을 touch 명령으로 옵션 없이 사용하면 파일의 수정 시간이 현재 시간으로 변경
- 예: data1.cp의 수정 시간을 touch 명령을 사용하여 현재 시간으로 변경

```
user1@myubuntu:~/linux_ex/ch2$ ls -l data1.cp
-rw-r--r-- 1 user1 user1 223 11월 12 11:24 data1.cp
user1@myubuntu:~/linux_ex/ch2$ date
2017. 11. 12. (일) 11:39:13 KST
user1@myubuntu:~/linux_ex/ch2$ touch data1.cp
user1@myubuntu:~/linux_ex/ch2$ ls -l data1.cp
-rw-r--r-- 1 user1 user1 223 11월 12 11:39 data1.cp
user1@myubuntu:~/linux_ex/ch2$
```

■ 빈 파일 생성하기, 수정 시간 변경하기 : touch

- -t 옵션 사용하여 변경할 시간 지정 가능

시간 표시

• 형식 [[CC]YY]MMDDhhmm[.ss]

CC: 연도의 첫 두 자리

YY: 연도의 마지막 두 자리

MM: 달(01~12 범위에서 지정)

DD: 날짜(01~31 범위에서 지정)

hh: 시간(00~23 범위에서 지정)

mm: 분(00~59 범위에서 지정)

ss: 초(00~59 범위에서 지정)

표 2-5 연도 지정 방법

YY	69~99	00~68
CC	19	20

```
user1@myubuntu:~/linux_ex/ch2$ ls -l test
-rw-rw-r-- 1 user1 user1 0 11월 12 11:38 test
user1@myubuntu:~/linux_ex/ch2$ touch -t 12311200 test
user1@myubuntu:~/linux_ex/ch2$ ls -l test
-rw-rw-r-- 1 user1 user1 0 12월 31 2017 test
user1@myubuntu:~/linux_ex/ch2$
```

■ 파일 내용 검색하기 : grep

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

03 파일 다루기

■ grep

- 예: 인자로 지정한 문자열 검색

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

- 예: 검색된 행의 줄 번호 출력(-n)

```
user1@myubuntu:~/linux_ex/ch2$ grep -n NNTP data
198:nntp      563/tcp      sntp      # NNTP over SSL
user1@myubuntu:~/linux_ex/ch2$
```

■ 파일 찾기 : find

find

- **기능** 지정한 위치에서 조건에 맞는 파일을 찾는다.
- **형식** `find [경로 검색 조건] [동작]`
- **옵션**
 - `-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 {} \;`

■ 파일 찾기 : find

- 예: /usr 디렉터리에서 ls 파일의 위치를 검색
 - 접근 권한이 없는 디렉터리는 검색할 수 없어서 '허가 거부' 메시지가 출력

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

- 특정 사용자 계정이 소유자인 파일을 찾고 싶으면 다음 예와 같이 -user 옵션을 사용

```
user1@myubuntu:~/linux_ex/ch2$ find /home -user user1
/home/user1
/home/user1/다운로드
/home/user1/.cache
/home/user1/.cache/gstreamer-1.0
/home/user1/.cache/gstreamer-1.0/registry.x86_64.bin
/home/user1/.cache/update-manager-core
/home/user1/.cache/update-manager-core/meta-release
(생략)
```

■ 파일 찾기 : find

- find 명령으로 검색한 모든 파일을 대상으로 동일한 작업을 수행하려면 -exec나 -ok 옵션 사용
- 예: /tmp 디렉터리 아래에 있는 user1 계정 소유의 파일을 전부 찾아서 삭제할 경우
 - find 명령으로 찾은 파일의 절대 경로가 exec 다음의 {}가 있는 위치에 삽입되어 명령이 처리
 - rm 명령과 {} 사이, {}와 \ 사이에 공백이 있어야 하며, \ 과 ;은 공백 없이 붙어야 함

```
user1@myubuntu:~/linux_ex/ch2$ find /tmp -user user1 -exec rm {} \;  
find: '/tmp/systemd-private-afe5e884236d4e96919e4a7f84cd37f0-colord.service-  
2o0tEo': 허가 거부  
rm: 쓰기 보호된 일반 파일 '/tmp/.X0-lock'를(을) 지울까요? n  
find: '/tmp/systemd-private-afe5e884236d4e96919e4a7f84cd37f0-systemd-resolved.  
service-R9KYbi': 허가 거부  
find: '/tmp/systemd-private-afe5e884236d4e96919e4a7f84cd37f0-systemd-timesyncd.  
service-QIA4iH': 허가 거부  
find: '/tmp/systemd-private-afe5e884236d4e96919e4a7f84cd37f0-rtkit-daemon.  
service-4upSu3': 허가 거부  
find: '/tmp/systemd-private-afe5e884236d4e96919e4a7f84cd37f0-fwupd.service-  
0hDA3B': 허가 거부  
user1@myubuntu:~/linux_ex/ch2$
```


■ 파일 찾기 : find

- find 명령으로 검색한 파일을 삭제하기 전에 하나씩 확인하고 싶으면 -exec 대신 -ok를 사용

```
user1@myubuntu:~/linux_ex/ch2$ find /tmp -user user1 -ok rm {} \;  
find: '/tmp/systemd-private-afe5e884236d4e96919e4a7f84cd37f0-colord.service-  
2o0tEo': 허가 거부  
< rm ... /tmp/.X0-lock > ? n  
find: '/tmp/systemd-private-afe5e884236d4e96919e4a7f84cd37f0-systemd-resolved.  
service-R9KYbi': 허가 거부  
find: '/tmp/systemd-private-afe5e884236d4e96919e4a7f84cd37f0-systemd-timesyncd.  
service-QIA4iH': 허가 거부  
find: '/tmp/systemd-private-afe5e884236d4e96919e4a7f84cd37f0-rtkit-daemon.  
service-4upSu3': 허가 거부  
find: '/tmp/systemd-private-afe5e884236d4e96919e4a7f84cd37f0-fwupd.service-  
0hDA3B': 허가 거부  
user1@myubuntu:~/linux_ex/ch2$
```

■ 명령의 위치 찾기 : whereis

- /bin, /usr/bin, /etc, /usr/etc, /sbin, /usr/sbin, /usr/share/man 등 정해진 디렉토리를 검색하여 명령의 위치검색

whereis

- **기능** 지정한 경로에서 명령의 바이너리 파일이나 매뉴얼 파일의 위치를 찾는다.
- **형식** whereis [옵션] 파일
- **옵션** -b: 바이너리 파일만 검색한다.
-m: 매뉴얼 파일만 검색한다.
-s: 소스 파일만 검색한다.
- **사용 예** whereis ls

- 예: mv 명령의 위치 검색

```
user1@myubuntu:~/linux_ex/ch2$ whereis mv
mv: /bin/mv /usr/share/man/man1/mv.1.gz
user1@myubuntu:~/linux_ex/ch2$
```

■ 명령의 위치 찾기 : which

- 에일리어스나 PATH 환경 변수로 지정된 경로에서 파일을 검색

which

- **기능** 명령어 파일의 위치를 찾아서 그 경로나 에일리어스를 출력한다.
- **형식** which 명령
- **사용 예** which ls

- 예: mv 명령의 위치 검색

```
user1@myubuntu:~/linux_ex/ch2$ which mv
/bin/mv
user1@myubuntu:~/linux_ex/ch2$
```