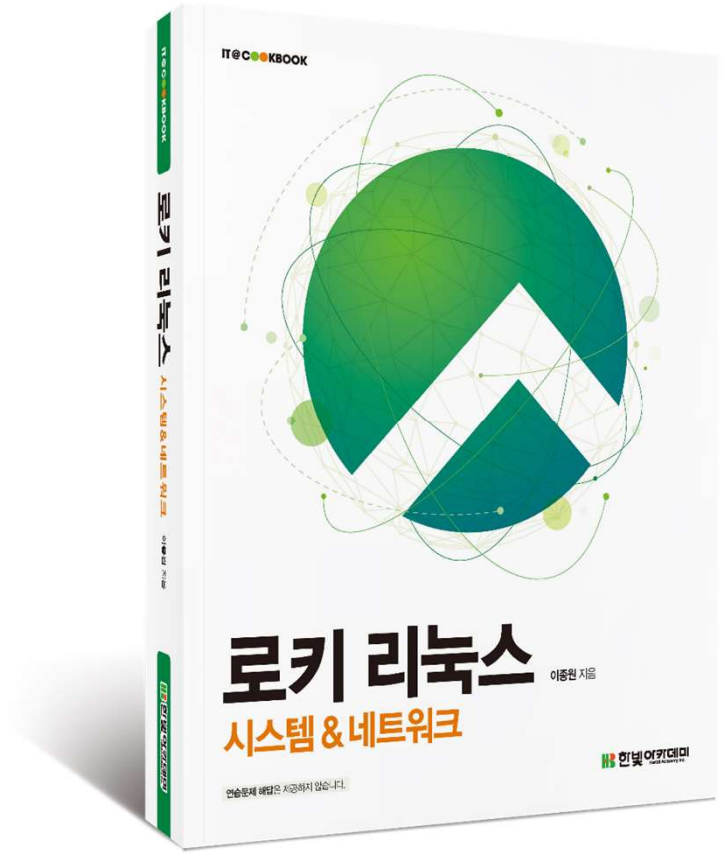

Chapter 02 디렉터리와 파일 사용법

목차

학습목표

- 리눅스 파일의 종류와 특징을 설명할 수 있다.
- 디렉터리 계층 구조를 보고 절대 경로명과 상대 경로명을 작성할 수 있다.
- 디렉터리를 이동하고, 디렉터리의 내용을 확인할 수 있다.
- 디렉터리를 만들고 삭제할 수 있다.
- 다양한 명령으로 파일 내용을 확인할 수 있다.

학습목표

- 파일을 복사·이동·삭제할 수 있다.
- 파일 링크의 특징을 설명하고, 하드 링크와 심볼릭 링크를 만들 수 있다.
- 파일의 내용과 위치를 검색할 수 있다.

00 Preview

00 Preview

2장의 내용 구성

- 리눅스 파일의 종류와 특징, 디렉터리 사용 명령, 파일을 다루는 명령 순으로 구성
- 디렉터리의 계층 구조 이해 후 이동·생성·삭제하는 명령 학습
- 파일의 내용을 보는 명령과 파일을 생성·복사·이동·삭제하는 명령 학습
- 하드 링크와 심볼릭 링크를 다룸

01 리눅스의 파일과 디렉터리

01 리눅스의 파일과 디렉터리

리눅스

- 리눅스도 유닉스 계열의 운영체제이므로 유닉스처럼 시스템 관련 정보와 장치를 관리하기 위해 파일을 사용

파일과 디렉터리

- 리눅스는 파일을 효과적으로 관리하기 위해 디렉터리를 사용
- 파일은 관련 있는 정보들의 집합
- 디렉터리는 폴더처럼 계층 구조를 가지고 있음
- 리눅스는 전체 파일을 용도에 따라 계층적인 디렉터리로 구분하여 관리

01 리눅스의 파일과 디렉터리

■ 파일의 종류

- 일반 파일 : 데이터 저장에 사용. 텍스트 파일, 실행 파일, 이미지 파일 등 리눅스에서 사용하는 대다수 파일은 일반 파일에 해당
- 디렉터리 : 해당 디렉터리에 저장된 파일 이나 서브 디렉터리에 대한 정보가 저장
- 심볼릭 링크 : 원본 파일을 대신하도록 원본 파일을 다른 파일명으로 지정한 것
- 장치 파일 : 리눅스 시스템에 부착된 장치들을 관리하기 위한 특수 파일

01 리눅스의 파일과 디렉터리

■ 파일의 종류 확인

- 파일의 종류를 확인하는 file 명령으로 현재 user1 사용자의 홈 디렉터리에 있는 .bash_profile과 다운로드의 종류를 알아보면 다음과 같은 결과가 나옴

```
[user1@localhost ~]$ file .bash_profile
.bash_profile: ASCII text
[user1@localhost ~]$ file 다운로드
다운로드: directory
```

- /usr/bin/more 파일의 종류도 확인

```
[user1@localhost ~]$ file /usr/bin/more
/usr/bin/more: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV),
(생략)
[user1@localhost ~]$ file /usr/bin/yum
/usr/bin/yum: symbolic link to dnf-3
```

01 리눅스의 파일과 디렉터리

■ 디렉터리의 계층구조

- 트리구조 : 파일을 효율적으로 관리하기 위해 디렉터리를 계층적으로 구성
- 하부 디렉터리로 나뉘고 각 디렉터리에 파일이 저장
- 리눅스에서도 모든 디렉터리의 출발점을 루트root 디렉터리라고 하며 /(슬래시)로 표시

01 리눅스의 파일과 디렉터리

■ 루트 디렉터리와 서브 디렉터리

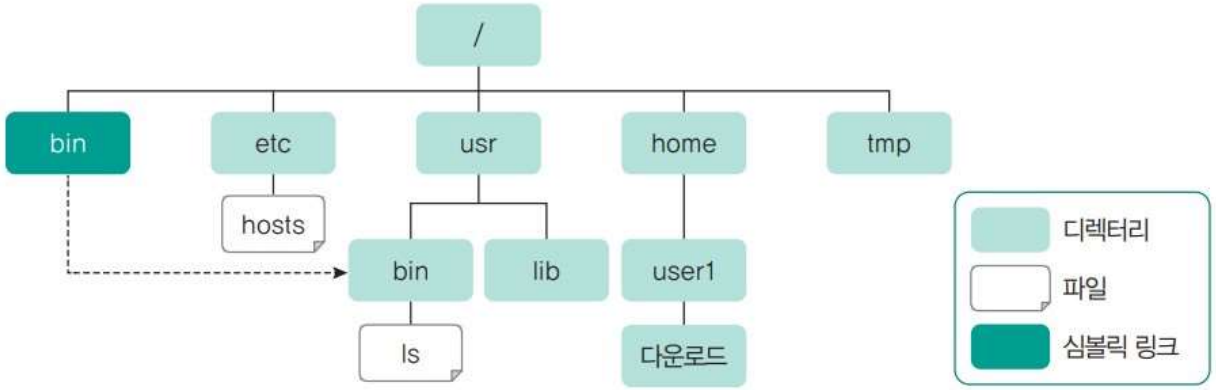

그림 2-1 디렉터리 계층 구조의 예

- 최상단에 루트 디렉터리 (/)가 있고, 그 아래에 etc, usr, home, tmp 같은 디렉터리가 있음
- 디렉터리 아래에 있는 디렉터리를 서브 디렉터리sub directory 또는 하위 디렉터리라고 함
- 서브 디렉터리의 입장에서 보면 위에 자신을 포함하고 있는 디렉터리가 있는데, 이를 부모 디렉터리 parent directory 또는 상위 디렉터리라고 함
- 상위 디렉터리는 ..(마침표 두 개)로 표시하며, .(마침표 한 개)는 현재 디렉터리를 말함

01 리눅스의 파일과 디렉터리

■ 루트 디렉터리

- 유일하게 부모 디렉터리가 없는 디렉터리. 아래에는 기본적으로 서브 디렉터리가 있음

```
[user1@localhost ~]$ ls -F /
afs/  boot/  etc/   lib@    media/  opt/   root/  sbin@  sys/  usr/
bin@  dev/   home/  lib64@  mnt/    proc/  run/   srv/   tmp/  var/
```

- 디렉터리 이름의 끝에 붙은 /는 해당 파일이 디렉터리임을, @(앳)은 심볼릭 링크임을 뜻함
- lib, sbin, bin, lib64는 디렉터리가 아니라 심볼릭 링크

01 리눅스의 파일과 디렉터리

■ 디렉터리의 주요기능

표 2-1 디렉터리의 주요 기능

디렉터리	기능
dev	장치 파일이 담긴 디렉터리
home	사용자 홈 디렉터리가 생성되는 디렉터리
media	DVD나 CD, USB 같은 외부 장치를 마운트(연결)하는 디렉터리
opt	추가 패키지가 설치되는 디렉터리
root	root 계정의 홈 디렉터리. 루트(/) 디렉터리와 다른 것이므로 혼동하지 않도록 한다.
sys	리눅스 커널과 관련된 파일이 있는 디렉터리
usr	기본 실행 파일과 라이브러리 파일, 헤더 파일 등 많은 파일을 가지고 있다. usr는 'Unix System Resource'의 약자

01 리눅스의 파일과 디렉터리

■ 디렉터리의 주요기능

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 디렉터리
proc	프로세스 정보 등 커널 관련 정보가 저장되는 디렉터리
run	실행 중인 서비스와 관련된 파일이 저장된다.
srv	FTP나 Web 등 시스템에서 제공하는 서비스의 데이터가 저장된다.
tmp	시스템 사용 중에 발생하는 임시 데이터가 저장된다. 시스템을 재시작하면 이 디렉터리에 있는 파일은 모두 삭제된다.
var	시스템 운영 중에 발생하는 데이터나 로그 등 내용이 자주 바뀌는 파일이 주로 저장된다.

01 리눅스의 파일과 디렉터리

작업 디렉터리

- 현재 사용 중인 디렉터리를 작업 디렉터리 또는 현재 디렉터리라고 함
- 작업 디렉터리의 위치는 pwd 명령으로 확인

홈 디렉터리

- 각 사용자에게 할당된 디렉터리. 처음 사용자 계정을 만들 때 지정
- 사용자는 자신의 홈 디렉터리 아래에 파일이나 서브 디렉터리를 생성하며 작업
- 홈 디렉터리는 ~(물결표)로 표시

01 리눅스의 파일과 디렉터리

■ 절대 경로명과 상대 경로명

- 경로명 : 디렉터리 계층 구조에 있는 특정 파일이나 디렉터리의 위치를 나타내는 것

표 2-2 절대 경로명과 상대 경로명의 특징

구분	특징
절대 경로명	• 반드시 /로 시작한다. • 루트 디렉터리부터 시작하여 특정 파일이나 디렉터리의 위치에 이르기까지 중간에 있는 모든 디렉터리의 이름을 표시한다. • 특정 위치를 가리키는 절대 경로명은 항상 동일하다.
상대 경로명	• / 이외의 문자로 시작한다. • 현재 디렉터리를 기준으로 서브 디렉터리로 내려가면 그냥 서브 디렉터리명을 사용한다. • 현재 디렉터리를 기준으로 상위 디렉터리로 가려면 ..(마침표 두 개)로 시작한다. • 상대 경로명은 현재 디렉터리가 어디냐에 따라 달라진다.

01 리눅스의 파일과 디렉터리

■ 절대 경로명과 상대 경로명

혼자해보기 절대 경로명과 상대 경로명 연습하기

[그림 2-1]의 디렉터리 계층 구조를 기준으로 다음 경로명을 알아보자. 현재 디렉터리는 user1이라고 가정한다.

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

01 리눅스의 파일과 디렉터리

■ 파일과 디렉터리 이름의 규칙

- 파일과 디렉터리 이름은 255자까지 가능
- 파일과 디렉터리 이름의 영문은 대문자와 소문자를 구별하여 다른 글자로 취급
- 파일과 디렉터리의 이름이 .(마침표)로 시작되면 숨김 파일로 간주
- 파일과 디렉터리 이름으로 모든 ASCII 문자를 사용할 수 있음
- 혼란을 줄 수 있으므로 /와 특수문자 사용은 삼가기

02 디렉터리 관련 명령

02 디렉터리 관련 명령

■ 현재 디렉터리 확인

- 현재 디렉터리를 확인하는 명령은 pwd

pwd

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

- user1 계정으로 로그인하면 현재 디렉터리는 user1 계정의 홈 디렉터리가 됨

```
[user1@localhost ~]$ pwd
/home/user1
```

02 디렉터리 관련 명령

■ 디렉터리 이동

cd

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

- 디렉터리에서 다른 디렉터리로 이동할 때는 cd 명령을 사용
- cd 명령과 함께 이동하고자 하는 목적지 디렉터리를 지정하면 해당 디렉터리로 이동
- 이동할 디렉터리의 경로명으로 절대 경로명과 상대 경로명 둘 다 사용할 수 있음

02 디렉터리 관련 명령

■ 디렉터리 이동

```
[user1@localhost ~]$ cd /tmp
[user1@localhost tmp]$ pwd
/tmp
```

- 이동한 뒤 pwd 명령을 사용하여 현재 디렉터리가 바뀌었는지 확인 (~가 tmp로 바뀜)
- 프롬프트에 현재 디렉터리의 이름을 표시하도록 설정되어 있는 것

```
[user1@localhost tmp]$ cd ../usr/lib
[user1@localhost lib]$ pwd
/usr/lib
```

- 상대 경로명을 이용하여 디렉터리를 이동할 경우 상위 디렉터리로 이동해야 하므로 ..(마침표 두 개)로 시작

02 디렉터리 관련 명령

■ 원래의 홈 디렉터리로 이동방법

- 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 디렉터리를 나타내는 기호인 ~를 사용하여 홈 디렉터리로 이동
- cd: 목적지를 지정하지 않고 cd 명령만 사용하면 해당 계정의 홈 디렉터리로 이동

```
[user1@localhost lib]$ cd
[user1@localhost ~]$ pwd
/home/user1
```

02 디렉터리 관련 명령

■ 디렉터리 내용 확인

- ls : 디렉터리에 있는 파일이나 서브 디렉터리 등 디렉터리의 내용을 보는 명령
- ls 명령 은 다양한 기능을 제공하는 옵션을 사용하고, 내용을 보고 싶은 목적지 디렉터리를 인자로 지정할 수 있다

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: 서브 디렉터리의 목록까지 출력한다.
- 사용 예 ls ls -F ls -al /tmp

02 디렉터리 관련 명령

현재 디렉터리의 내용 확인하기: ls

- 옵션이나 디렉터리를 지정하지 않고 ls 명령을 사용하면 현재 디렉터리의 내용을 출력

```
[user1@localhost ~]$ ls
공개  다운로드  문서  바탕화면  비디오  사진  서식  음악
```

숨김 파일 확인하기: -a 옵션

- 리눅스에서는 파일명이나 디렉터리명을 .(마침표)로 시작하면 숨김 파일이 됨
- ls 명령만 사용해서는 보이지 않고 -a 옵션을 지정해야 함

```
[user1@localhost ~]$ ls -a
.   .bash_history  .bash_profile  .cache   .local   .viminfo  다운로드  바탕화면  사진   음악
..  .bash_logout   .bashrc        .config  .mozilla           공개      문서      비디오  서식
```

- 현재 디렉터리를 나타내는 .(마침표)와 상위 디렉터리를 나타내는 ..(마침표 두 개)도 확인할 수 있음

02 디렉터리 관련 명령

■ 파일의 종류 표시하기: -F 옵션

- ls 명령 에서도 -F 옵션을 사용하면 파일의 종류를 구분하는 기호가 표시 됨
- 파일명 뒤에 /가 붙으면 디렉터리, @이 붙으면 심볼릭 링크, *가 붙으면 실행 파일을 의미하고, 아무 표시도 없으면 일반 파일

```
[user1@localhost ~]$ ls -F
공개/  다운로드/  문서/  바탕화면/  비디오/  사진/  서식/  음악/
```

02 디렉터리 관련 명령

■ 옵션 여러 개 사용하기

- 옵션을 연결할 때는 -(하이픈) 뒤에 옵션만 나열
- 숨김 파일을 보여주는 a 옵션과 파일의 종류를 보여주는 F 옵션을 연결하여 사용하면 숨김 파일의 종류도 알 수 있음
- .(마침표)와 ..(마침표 두 개)에도 /가 붙어 있음

```
[user1@localhost ~]$ ls -aF
./  .bash_history .bash_profile .cache/  .local/  .viminfo 다운로드/ 바탕화면/ 사진/   음악/
../ .bash_logout  .bashrc       .config/ .mozilla/        공개/     문서/     비디오/ 서식/
```

02 디렉터리 관련 명령

■ 지정한 디렉터리의 내용 출력하기

- 해당 디렉터리로 이동하지 않고도 디렉터리의 내용을 확인할 수 있음

```
[user1@localhost ~]$ ls /tmp
systemd-private-8ffa9a5505d343ebad65e618c92bfce2-ModemManager.service-XTd1Fa
systemd-private-8ffa9a5505d343ebad65e618c92bfce2-bluetooth.service-IeKFUN
systemd-private-8ffa9a5505d343ebad65e618c92bfce2-chronyd.service-Li1yDE
(생략)
```

- 옵션과 인자를 함께 사용할 수도 있음

```
[user1@localhost ~]$ ls -F /tmp
systemd-private-8ffa9a5505d343ebad65e618c92bfce2-ModemManager.service-XTd1Fa/
systemd-private-8ffa9a5505d343ebad65e618c92bfce2-bluetooth.service-IeKFUN/
systemd-private-8ffa9a5505d343ebad65e618c92bfce2-chronyd.service-Li1yDE/
(생략)
```

02 디렉터리 관련 명령

■ 상세 정보 출력하기: -l 옵션

- 디렉터리에 있는 파일들의 상세한 정보를 보려면 -l 옵션을 사용

```
[user1@localhost ~]$ ls -l
합계 0
drwxr-xr-x. 2 user1 user1 6  7월  9 10:48 공개
drwxr-xr-x. 2 user1 user1 6  7월  9 10:48 다운로드
drwxr-xr-x. 2 user1 user1 6  7월  9 10:48 문서
(생략)
```

02 디렉터리 관련 명령

■ -l 옵션을 지정하여 출력한 정보의 의미

표 2-3 파일의 상세 정보

필드 번호	필드 값	의미
1	d	다음과 같은 파일 종류를 나타낸다. -: 일반(정규) 파일 d: 디렉터리 파일 l: 심볼릭 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)
2	rwxr-xr-x	파일 접근 권한. 파일의 소유자, 그룹, 기타 사용자가 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	6	파일 크기(바이트 단위)
7	7월 9 10:48	파일이 마지막으로 수정된 시간
8	공개	파일명

02 디렉터리 관련 명령

■ 디렉터리의 자체 정보 확인하기: -d 옵션

- 디렉터리의 자체 정보를 확인할 때는 -d 옵션을 사용

```
[user1@localhost ~]$ ls -l /
합계 24
dr-xr-xr-x.   2 root root       6   5월   16   2022    afs
lrwxrwxrwx.   1 root root       7   5월   16   2022    bin        → usr/bin
dr-xr-xr-x.   5 root root 409   6   7월   9    10:47   boot
(생략)
[user1@localhost ~]$ ls -ld /
dr-xr-xr-x. 18 root root 235  7월  9 10:39 /
```

02 디렉터리 관련 명령

ls 명령과 비슷한 명령: dir, vdir

- 디렉터리의 내용을 보는 dir과 vdir 명령

```
[user1@localhost ~]$ dir
공개  다운로드  문서  바탕화면  비디오  사진  서식  음악
[user1@localhost ~]$ vdir
합계 0
drwxr-xr-x. 2 user1 user1 6  7월  9 10:48 공개
drwxr-xr-x. 2 user1 user1 6  7월  9 10:48 다운로드
drwxr-xr-x. 2 user1 user1 6  7월  9 10:48 문서
(생략)
```

02 디렉터리 관련 명령

■ 디렉터리 생성

- 홈 디렉터리와 쓰기 권한이 있는 디렉터리에 사용자의 필요에 따라 디렉터리를 생성할 수 있음

mkdir

- 기능 디렉터리를 생성한다.
- 형식 mkdir [옵션] [디렉터리]
- 옵션 -p: 서브 디렉터리를 계층적으로 생성할 때 중간 단계의 디렉터리가 없으면 자동으로 중간 단계 디렉터리를 생성한 뒤 지정한 디렉터리를 생성한다.
- 사용 예 mkdir temp

02 디렉터리 관련 명령

디렉터리 한 개 만들기

- 디렉터리를 한 개만 만들려면 mkdir 명령에 인자로 생성하려는 디렉터리를 지정하면 됨

```
[user1@localhost ~]$ mkdir temp
[user1@localhost ~]$ ls
temp  공개  다운로드  문서  바탕화면  비디오  사진  서식  음악
```

동시에 디렉터리 여러 개 만들기

```
[user1@localhost ~]$ mkdir tmp1 tmp2 tmp3
[user1@localhost ~]$ ls
temp  tmp1  tmp2  tmp3  공개  다운로드  문서  바탕화면  비디오  사진  서식  음악
```

02 디렉터리 관련 명령

■ 중간 디렉터리를 자동으로 만들기: -p 옵션

- mkdir 명령 다음에 -p 옵션을 사용하면, 생성할 디렉터리로 지정한 경로 중 중간 단계의 디렉터리가 없을 경우 자동으로 중간 단계 디렉터리를 생성한 후 최종 디렉터리를 만듦
- mkdir 명령에 -p 옵션을 사용하지 않은 경우

```
[user1@localhost ~]$ mkdir temp/mid/han
mkdir: ‘temp/mid/han’ 디렉터리를 만들 수 없습니다: 그런 파일이나 디렉터리가 없습니다
```

- mkdir 명령에 -p 옵션을 사용한 경우

```
[user1@localhost ~]$ mkdir -p temp/mid/han
[user1@localhost ~]$ ls -R temp
temp:
mid

temp/mid:
han

temp/mid/han:
```

02 디렉터리 관련 명령

■ 디렉터리 삭제

- 디렉터리가 더 이상 필요 없거나 잘못 만들어졌다면 삭제할 수 있음

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

02 디렉터리 관련 명령

■ 디렉터리 삭제

- mkdir 명령 예에서 만든 tmp3을 삭제하는 예

```
[user1@localhost ~]$ rmdir tmp3
[user1@localhost ~]$ ls
temp  tmp1  tmp2  공개  다운로드  문서  바탕화면  비디오  사진  서식  음악
```

- rmdir 명령으로 디렉터리를 삭제할 때는 해당 디렉터리가 비어 있어야 함
- 디렉터리에 파일이나 서브 디렉터리가 남아 있으면 rmdir로 디렉터리를 삭제할 수 없음
- 비어 있지 않은 디렉터리를 삭제하려 했을 때

```
[user1@localhost ~]$ rmdir temp
rmdir: failed to remove 'temp': 디렉터리가 비어있지 않음
```

02 디렉터리 관련 명령

■ 디렉터리 만들고 삭제하기

① 현재 위치를 확인

- 홈 디렉터리가 아니면 홈 디렉터리로 이동

```
[user1@localhost ~]$ cd
[user1@localhost ~]$ pwd
/home/user1
```

② 기본 디렉터리 만들기

- 홈 디렉터리에 Test 디렉터리를 만들고 그 디렉터리로 이동

```
[user1@localhost ~]$ mkdir Test
[user1@localhost ~]$ cd Test
[user1@localhost Test]$
```

02 디렉터리 관련 명령

■ 디렉터리 만들고 삭제하기

③ 디렉터리를 만들기

- 디렉터리로 이동하여 현재 위치를 알아

```
[user1@localhost Test]$ mkdir ch2
[user1@localhost Test]$ cd ch2
[user1@localhost ch2]$
```

④ 디렉터리 동시에 만들기

```
[user1@localhost ch2]$ mkdir one two three
[user1@localhost ch2]$ ls
one  three  two
```

02 디렉터리 관련 명령

■ 디렉터리 만들고 삭제하기

⑤ tmp/test 디렉터리 만들기

- 중간 경로인 tmp 디렉터리가 자동 생성되도록 함

```
[user1@localhost ch2]$ mkdir -p one/tmp/test
[user1@localhost ch2]$ ls -R one
one:
tmp

one/tmp:
test

one/tmp/test:
```

02 디렉터리 관련 명령

■ 디렉터리 만들고 삭제하기

⑥ one 디렉터리를 rmdir 명령으로 삭제

```
[user1@localhost ch2]$ rmdir one
rmdir: failed to remove 'one': 디렉터리가 비어있지 않음
```

⑦ two, three 디렉터리를 동시에 삭제

```
[user1@localhost ch2]$ rmdir two three
[user1@localhost ch2]$ ls
one
```

02 디렉터리 관련 명령

■ 디렉터리 만들고 삭제하기

⑧ 홈 디렉터리로 이동

```
[user1@localhost ch2]$ cd
[user1@localhost ~]$
```

혼자해보기 디렉터리 만들고 삭제하기

❶ 실습 디렉터리 아래의 ch2 디렉터리로 한 번에 이동한다.
❷ 현재 작업 디렉터리를 절대 경로로 출력한다.
❸ 현재 디렉터리의 숨김 파일을 포함하여 모든 파일을 출력한다.
❹ me 디렉터리를 생성한다.
❺ me 디렉터리 아래에 you, him 디렉터리를 동시에 생성한다.
❻ me 디렉터리를 삭제한다. 삭제되지 않는다면 그 이유는 무엇인가?
❼ 실습을 마치고 홈 디렉터리로 이동한다.

03 파일 관련 명령

03 파일 관련 명령

■ 파일 내용 출력

- 파일 내용을 연속으로 출력하기: cat

cat

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

03 파일 관련 명령

■ 파일 내용 출력

- /etc/hosts 파일의 내용을 cat 명령으로 확인

```
[user1@localhost ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

- cat 명령의 -n 옵션을 사용하면 행 번호가 붙어서 출력

```
[user1@localhost ~]$ cat -n /etc/hosts
     1  127.0.0.1   localhost localhost.localdomain localhost4 localhost4.
localdomain4
     2  ::1         localhost localhost.localdomain localhost6 localhost6.
localdomain
```

03 파일 관련 명령

■ 파일 내용을 화면 단위로 출력하기: more

- 파일 내용을 연속으로 출력하기 때문에 내용이 많으면 스크롤되어 내용을 확인하기가 어려움
- 이럴 때 화면 단위로 파일 내용을 출력해 주는 more 명령을 사용

more

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

03 파일 관련 명령

■ 파일 내용을 화면 단위로 출력하기: more

- more 명령은 파일 내용을 화면 단위로 출력하고, 출력할 내용이 더 있으면 화면 하단에 '--More--(0%)'와 같이 알려줌

```
[user1@localhost ~]$ more /etc/services
# /etc/services:
# $Id: services,v 1.49 2017/08/18 12:43:23 ovasik Exp $
#
# Network services, Internet style
# IANA services version: last updated 2016-07-08
(생략)
chargen         19/udp          ttytst source
ftp-data        20/tcp
--More--(0%)
```

03 파일 관련 명령

■ 파일 내용을 화면 단위로 출력하기: less

- more 명령은 이미 스크롤되어 지나간 내용을 다시 볼 수 없다는 것
- less 명령을 사용하면 파일 내용을 앞뒤로 스크롤하며 이동할 수 있음

less

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

표 2-4 less 명령에서 사용하는 키와 동작

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

03 파일 관련 명령

■ 파일 내용의 뒷부분 출력하기: tail

- tail은 파일 뒷부분의 몇 행을 출력, 기본값은 10으로 파일 뒷부분의 10행이 출력 됨

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.

03 파일 관련 명령

■ 파일 내용의 뒷부분 출력하기: tail

- /etc/services 파일의 마지막 10행이 출력 됨

```
[user1@localhost ~]$ tail /etc/services
aigairserver    21221/tcp       # Services for Air Server
ka-kdp          31016/udp       # Kollective Agent Kollective Delivery
ka-sddp         31016/tcp       # Kollective Agent Secure Distributed Delivery
edi_service     34567/udp       # dhanalakshmi.org EDI Service
axio-disc       35100/tcp       # Axiomatic discovery protocol
axio-disc       35100/udp       # Axiomatic discovery protocol
pmwebapi        44323/tcp       # Performance Co-Pilot client HTTP API
cloudcheck-ping 45514/udp       # ASSIA CloudCheck WiFi Management keepalive
cloudcheck      45514/tcp       # ASSIA CloudCheck WiFi Management System
spremotetablet  46998/tcp       # Capture handwritten signatures
```

03 파일 관련 명령

■ 파일 내용을 지정한 숫자만큼 출력하기: -숫자 옵션

- - 다음에 숫자를 지정하면 기본값인 10행이 아니라 지정한 숫자만큼 행을 출력

```
[user1@localhost ~]$ tail -3 /etc/services
cloudcheck-ping      45514/udp      # ASSIA CloudCheck WiFi Management keepalive
cloudcheck 45514/tcp                # ASSIA CloudCheck WiFi Management System
spremotetablet 46998/tcp            # Capture handwritten signatures
```

03 파일 관련 명령

■ 파일 내용을 주기적으로 반복 출력하기: -f 옵션

- -f 옵션을 사용하면 파일 출력이 종료되지 않고 대기 상태가 되며 파일 내용이 주기적으로 반복 출력
- 파일 내용의 변화를 확인할 때 편리

```
[user1@localhost ~]$ tail -f /etc/services
aigairserver   21221/tcp        # Services for Air Server
ka-kdp      31016/udp           # Kollective Agent Kollective Delivery
ka-sddp         31016/tcp       # Kollective Agent Secure Distributed Delivery
(생략)
cloudcheck 45514/tcp            # ASSIA CloudCheck WiFi Management System
spremotetablet 46998/tcp        # Capture handwritten signatures
```

03 파일 관련 명령

■ 파일 내용 출력하기

① less 명령으로 /etc/services 파일의 내용을 출력

```
[user1@localhost ~]$ less /etc/services
# /etc/services:
# $Id: services,v 1.49 2017/08/18 12:43:23 ovasik Exp $
#
# Network services, Internet style
# IANA services version: last updated 2016-07-08
(생략)
chargen         19/udp          ttytst source
ftp-data        20/tcp
/etc/services
```

03 파일 관련 명령

■ 파일 내용 출력하기

② Space Bar를 눌러 다음 페이지를 출력

```
(생략)
tftp            69/udp
gopher          70/tcp                  # Internet Gopher
gopher          70/udp
netrjs-1        71/tcp                  # Remote Job Service
netrjs-1        71/udp                  # Remote Job Service
:
```

03 파일 관련 명령

■ 파일 내용 출력하기

③ k 키를 네 번 눌러 위로 다시 이동

```
(생략)
bootps          67/udp
bootpc          68/tcp          dhcpc               # BOOTP client
bootpc          68/udp          dhcpc
tftp            69/tcp
tftp            69/udp
:
```

03 파일 관련 명령

■ 파일 내용 출력하기

④ '/IEEE'를 입력하여 파일 내용 중에서 문자열 'IEEE'가 있는 곳을 찾음

```
(생략)
bootps          67/udp
bootpc          68/tcp          dhcpc           # BOOTP client
bootpc          68/udp          dhcpc
tftp            69/tcp
tftp            69/udp
/IEEE
```

```
ieee-mms        651/tcp                         # IEEE MMS
ieee-mms        651/udp                         # IEEE MMS
hello-port      652/tcp                         # HELLO_PORT
hello-port      652/udp                         # HELLO_PORT
repscmd         653/tcp                         # RepCmd
(생략)
:
```

03 파일 관련 명령

■ 파일 내용 출력하기

⑤ n 키를 눌러 문자열 'IEEE'를 계속 찾음

⑥ q 키를 눌러 less 명령을 종료

혼자해보기 파일 내용 출력하기

❶ more 명령으로 /etc/services 파일을 출력한다.
❷ 파일 내용에서 Apple을 검색한다.
❸ 다른 곳에도 Apple이 있는지 추가로 확인한다.
❹ more 명령을 종료한다.

03 파일 관련 명령

■ 파일 복사

- 파일을 복사할 때는 cp명령을 사용

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

- cp 명령 첫 번째 인자로는 원본 파일이나 디렉터리를 지정, 두 번째 인자로는 목적지 파일이나 디렉터리를 지정

```
[user1@localhost ~]$ cd Test/ch2
[user1@localhost ch2]$
```

03 파일 관련 명령

■ 두 인자가 모두 파일인 경우

- 두 인자가 모두 파일이면 원본 파일을 다른 파일로 복사
- 두 번째 인자로 지정한 파일이 존재하지 않는다면 원본 파일을 복사하여 새로 만듦

```
[user1@localhost ch2]$ ls
one
[user1@localhost ch2]$ cp /etc/hosts text1
[user1@localhost ch2]$ ls
one  text1
```

03 파일 관련 명령

■ 두 번째 인자가 디렉터리인 경우

- 첫 번째 인자는 파일이고 두 번째 인자는 디렉터리인 경우, 파일을 해당 디렉터리 아래에 복사함

```
[user1@localhost ch2]$ mkdir temp
[user1@localhost ch2]$ cp text1 temp
[user1@localhost ch2]$ ls temp
text1
```

- 파일을 복사하면서 파일명을 원본 파일과 다르게 지정하면 새로운 파일명으로 복사됨

```
[user1@localhost ch2]$ cp text1 temp/text2
[user1@localhost ch2]$ ls temp
text1  text2
```

03 파일 관련 명령

■ 두 번째 인자가 디렉터리인 경우

- 파일을 다른 디렉터리에 복사할 때는 해당 디렉터리에 대해 쓰기 권한이 있어야 함
- 쓰기 권한이 없는 디렉터리에 파일을 복사하려고 하면 다음과 같은 오류가 발생함

```
[user1@localhost ch2]$ cp text1 /etc
cp: cannot create regular file '/etc/text1': 허가 거부
```

■ 인자를 여러 개 지정할 경우

- cp 명령에서 첫 번째 인자의 자리에 파일을 여러 개 지정할 수 있음
- 마지막 인자는 반드시 디렉터리여야 함

```
[user1@localhost ch2]$ cp /etc/hosts /etc/services temp
[user1@localhost ch2]$ ls temp
hosts  services  text1  text2
```

03 파일 관련 명령

-i 옵션 사용하기

```
[user1@localhost ch2]$ cp -i /etc/hosts text1
cp: overwrite 'text1'? n
```

디렉터리 복사하기

- -r 옵션을 지정하지 않고 temp 디렉터리를 temp2 디렉터리로 복사한 경우

```
[user1@localhost ch2]$ cp temp temp2
cp: -r not specified; omitting directory 'temp'
```

- -r 옵션을 지정하여 temp 디렉터리를 temp2 디렉터리로 복사한 경우

```
[user1@localhost ch2]$ cp -r temp temp2
[user1@localhost ch2]$ ls temp2
hosts  services  text1  text2
```

03 파일 관련 명령

■ 디렉터리 복사하기

- 두 번째 인자로 지정한 디렉터리가 이미 있는 디렉터리인 경우, 원본 디렉터리가 목적지 디렉터리 아래에 원본 디렉터리와 같은 이름으로 복사됨

```
[user1@localhost ch2]$ cp -r temp temp2
[user1@localhost ch2]$ ls temp2
hosts  services  temp  text1  text2
```

- 앞의 예에서 temp2 디렉터리가 생성되었으므로 이번에는 temp 디렉터리가 temp2 디렉터리 아래에 복사됨

03 파일 관련 명령

■ 파일 이동과 파일명 변경

- mv : 파일을 다른 디렉터리로 이동하거나 파일명을 바꿀 때 사용

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

- mv 명령의 첫 번째 인자로는 원본 파일이나 디렉터리를 지정하고, 두 번째 인자로는 목적지 파일이나 디렉터리를 지정함

03 파일 관련 명령

■ 파일을 파일로 이동하기(파일명 바꾸기)

- 파일을 다른 파일로 이동한다는 것은 결국 원본 파일의 이름을 다른 이름으로 바꾸는 작업임

```
[user1@localhost ch2]$ mv text1 data1
[user1@localhost ch2]$ ls
data1  one  temp  temp2
```

- 만약 두 번째 인자로 지정한 파일이 이미 존재하는 파일이라면 원본 파일의 내용으로 덮어쓰고 기존 내용이 삭제됨
- 두 번째 인자로 지정한 파일이 존재하지 않는 파일이라면 새 파일이 생성됨

03 파일 관련 명령

■ 파일을 다른 디렉터리로 이동하기

- 두 번째 인자로 디렉터리를 지정하는 경우 원본 파일을 지정한 디렉터리로 이동함
- 디렉터리만 지정하는지, 디렉터리와 파일을 함께 지정하는지에 따라 이동하는 파일의 이름이 달라질 수 있음

```
[user1@localhost ch2]$ mv data1 temp
[user1@localhost ch2]$ ls
one  temp  temp2
[user1@localhost ch2]$ ls temp
data1  hosts  services  text1  text2
```

03 파일 관련 명령

■ 파일을 다른 디렉터리로 이동하기

- 앞의 예에서 파일을 이동하여 ch2 디렉터리에 파일이 없으므로 먼저 파일을 복사한 후 이동을 실행

```
[user1@localhost ch2]$ cp temp/data1 text1
[user1@localhost ch2]$ ls
one  temp  temp2  text1
[user1@localhost ch2]$ mv text1 temp/data2
[user1@localhost ch2]$ ls temp
data1  data2  hosts  services  text1  text2
```

- 쓰기 권한이 없는 디렉터리로 파일을 이동하려고 한다면 다음과 같이 오류가 발생

```
[user1@localhost ch2]$ mv temp/data2 /etc
mv: cannot move 'temp/data2' to '/etc/data2': 허가 거부
```

03 파일 관련 명령

■ 여러 파일을 디렉터리로 이동하기

- mv 명령을 사용해 여러 개의 파일을 지정한 디렉터리로 한 번에 이동할 수 있음
- 첫 번째 인자에 파일을 여러 개 지정하면 됨
- 마지막 인자는 반드시 디렉터리여야 함

```
[user1@localhost ch2]$ mv temp/data1 temp/data2 .
[user1@localhost ch2]$ ls
data1  data2  one  temp  temp2
```

■ -i 옵션 사용하기

```
[user1@localhost ch2]$ mv -i data1 data2
mv: overwrite 'data2'? n
[user1@localhost ch2]$ ls
data1  data2  one  temp  temp2
```

03 파일 관련 명령

■ 디렉터리를 디렉터리로 이동하기(디렉터리명 바꾸기)

- mv 명령에서 인자를 모두 디렉터리로 지정하면 디렉터리가 이동함
- 두 번째 인자가 기존에 있던 디렉터리가 아닐 경우 디렉터리명이 변경됨

```
[user1@localhost ch2]$ mv temp2 temp3
[user1@localhost ch2]$ ls
data1  data2  one  temp  temp3
```

- 두 번째 인자가 존재하는 디렉터리인 경우, 원본 디렉터리가 두 번째 인자로 지정된 디렉터리 아래로 이동함

```
[user1@localhost ch2]$ mv temp3 temp
[user1@localhost ch2]$ ls
data1  data2  one  temp
[user1@localhost ch2]$ ls temp
hosts  services  temp3  text1  text2
```

03 파일 관련 명령

■ 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

- rm 명령을 사용할 때 파일을 인자로 지정하면 해당 파일이 삭제됨

```
[user1@localhost ch2]$ ls
data1  data2  one  temp
[user1@localhost ch2]$ rm data2
[user1@localhost ch2]$ ls
data1  one  temp
```

03 파일 관련 명령

-i 옵션 사용하기

- -i 옵션을 사용하여 data1 파일을 지우려고 시도한 예

```
[user1@localhost ch2]$ rm -i data1
rm: remove 일반 파일 'data1'? n
[user1@localhost ch2]$ ls
data1  one  temp
```

03 파일 관련 명령

■ 디렉터리 삭제하기

- rm 명령으로 디렉터리도 삭제 가능
- rmdir 명령으로도 디렉터리를 삭제 할 수 있는데, 이 경우에는 삭제하려는 디렉터리가 비어 있어야 함
- rm 명령으로 디렉터리를 삭제할 때는 -r 옵션을 지정해야 함

```
[user1@localhost ch2]$ cd temp
[user1@localhost temp]$ ls
hosts  services  temp3  text1  text2
[user1@localhost temp]$ rm temp3
rm: cannot remove 'temp3': 디렉터리입니다
```

- rm 명령으로 디렉터리를 삭제하려고 하면 디렉터리라서 삭제할 수 없다는 오류 메시지가 출력됨

03 파일 관련 명령

■ 디렉터리 삭제하기

- rmdir 명령으로 temp3 디렉터리를 삭제하려고 하면 temp3 디렉터리가 비어 있지 않다는 오류 메시지가 출력됨

```
[user1@localhost temp]$ rmdir temp3
rmdir: failed to remove 'temp3': 디렉터리가 비어있지 않음
```

- rm 명령에 -r 옵션을 지정하여 temp3 디렉터리를 삭제했을 경우

```
[user1@localhost temp]$ ls
hosts  services  temp3  text1  text2
[user1@localhost temp]$ rm -r temp3
[user1@localhost temp]$ ls
hosts  services  text1  text2
```

03 파일 관련 명령

■ 디렉터리 삭제하기

- 디렉터리를 삭제할 때도 -i 옵션을 사용할 수 있음

```
[user1@localhost temp]$ cd ..
[user1@localhost ch2]$ ls
data1  one  temp
[user1@localhost ch2]$ rm -ri temp
rm: descend into directory 'temp'? y
rm: remove 일반 파일 'temp/text1'? n
rm: remove 일반 파일 'temp/text2'? n
rm: remove 일반 파일 'temp/hosts'? n
rm: remove 일반 파일 'temp/services'? y
rm: remove 디렉토리 'temp'? n
[user1@localhost ch2]$ ls temp
hosts  text1  text2
```

03 파일 관련 명령

■ 디렉터리 삭제하기

- 디렉터리에 삭제되지 않은 파일이 있을 경우 디렉터리가 비어 있지 않다는 메시지를 출력하고 디렉터리를 삭제하지 않음

```
[user1@localhost ch2]$ rm -ri temp
rm: descend into directory 'temp'? y
rm: remove 일반 파일 'temp/text1'? n
rm: remove 일반 파일 'temp/text2'? n
rm: remove 일반 파일 'temp/hosts'? n
rm: remove 디렉터리 'temp'? y
rm: cannot remove 'temp': 디렉터리가 비어있지 않음
[user1@localhost ch2]$ ls
data1  one  temp
```

03 파일 관련 명령

■ 파일 복사·이동·삭제하기

① /etc/hosts 파일을 test.org로 복사

```
[user1@localhost ch2]$ cp /etc/hosts test.org
[user1@localhost ch2]$ ls
data1  one  temp  test.org
```

② test 디렉터리를 생성

```
[user1@localhost ch2]$ mkdir test
[user1@localhost ch2]$ ls
data1  one  temp  test  test.or
```

③ test.org 파일을 test 디렉터리로 복사

```
[user1@localhost ch2]$ cp test.org test
[user1@localhost ch2]$ ls test
test.org
```

03 파일 관련 명령

■ 파일 복사·이동·삭제하기

④ test 디렉터리에 있는 test.org의 파일명을 test.bak로 바꿈

```
[user1@localhost ch2]$ mv test/test.org test/test.bak
[user1@localhost ch2]$ ls test
test.bak
```

⑤ test.org 파일을 삭제

```
[user1@localhost ch2]$ rm test.org
[user1@localhost ch2]$ ls
data1  one  temp  test
```

⑥ test 디렉터리에 있는 test.bak를 현재 디렉터리에 test.org로 복사

```
[user1@localhost ch2]$ cp test/test.bak test.org
[user1@localhost ch2]$ ls
data1  one  temp  test  test.org
```

03 파일 관련 명령

■ 파일 복사·이동·삭제하기

⑦ test, one 디렉터리를 삭제

```
[user1@localhost ch2]$ rm -r test one
[user1@localhost ch2]$ ls
data1  temp  test.org
```

혼자해보기 파일 복사 · 이동 · 삭제하기

❶ test.org 파일을 test.txt로 복사한다.
❷ backup 디렉터리를 생성한다.
❸ text.txt 파일을 backup 디렉터리로 이동한다.
❹ backup 디렉터리의 이름을 work로 바꾼다.
❺ rmdir로 work 디렉터리를 삭제한다. 어떤 현상이 일어나는가? 왜 그런가?
❻ rm 명령으로 work 디렉터리를 삭제한다. 정말로 삭제할 것인지 물어보게 하려면 어떻게 해야 하는가?

03 파일 관련 명령

■ 파일 링크

- 파일 링크 : 기존 파일에 새로운 이름을 붙이는 것
- 디렉터리 계층 구조나 파일 경로명이 복잡할 경우, 짧게 줄인 다른 이름을 붙여서 간단하게 사용할 때 유용
- 하드 링크 : 기존 파일에 새로운 파일명을 추가로 생성하는 것
- 심볼릭 링크 : 원본 파일을 가리키는 새로운 파일을 만드는 것

03 파일 관련 명령

■ 리눅스 파일의 구성 알아보기

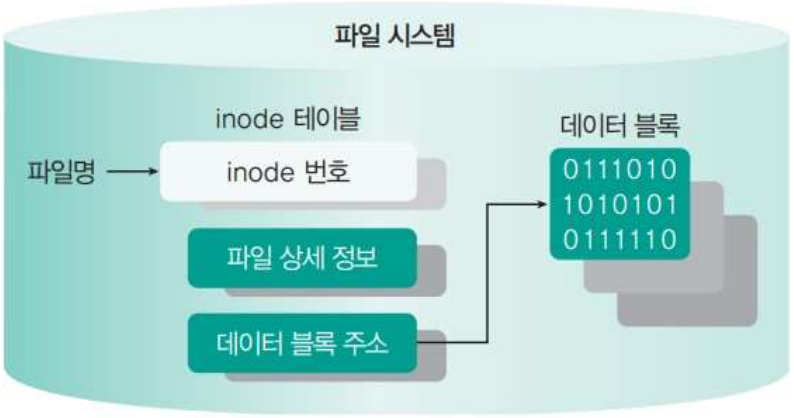

그림 2-2 리눅스 파일의 구성 요소

- 리눅스에서 파일은 '파일명+inode+데이터 블록'으로 구성
- 파일명 : 사용자가 파일에 접근할 때 사용하는 파일의 이름
- inode는 파일 정보가 담긴 특별한 구 조체로서, 외부적으로는 번호로 표시되고 내부적으로는 파일의 종류·크기·소유자, 파일 변 경 시간, 파일명 등 파일 상세 정보와 데이터 블록 주소가 저장되어 있음

03 파일 관련 명령

■ 리눅스 파일의 구성 알아보기

- 파일의 inode 번호는 ls -i 명령으로 알 수 있음

```
[user1@localhost ch2]$ ls -i
18660662 data1  18660663 temp  18660667 test.org
```

03 파일 관련 명령

■ 하드 링크 만들기: ln

- 하드링크 : 리눅스에서 한 파일에 여러 개의 이름을 붙일 때 붙이는 파일명
- 하드 링크는 ln 명령으로 만듦

ln

- **기능** 파일의 링크를 생성한다.
- **형식** ln [옵션] [원본 파일] [링크 파일]
- **옵션** -s: 심볼릭 링크 파일을 생성한다.
- **사용 예** `ln test hdtest` `ln -s test sbtest`

03 파일 관련 명령

■ 하드 링크 만들기: ln

- 기존 파일에 새로운 파일명을 붙여서 하드 링크를 만들기

```
[user1@localhost ch2]$ ls -l data1
-rw-r--r--. 1 user1 user1 158  8월  6 16:47 data1
[user1@localhost ch2]$ ln data1 data1.ln
[user1@localhost ch2]$ ls -l data1
-rw-r--r--. 2 user1 user1 158  8월  6 16:47 data1
```

- data1 파일에 하드 링크를 생성한 뒤 ls -i 명령으로 두 파일의 inode 기본값을 비교해 보면 같다는 것을 알 수 있음. 이름만 다르고 결국 같은 파일이기 때문

```
[user1@localhost ch2]$ ls -i
18660662 data1  18660662 data1.ln  18660663 temp  18660667 test.org
```

03 파일 관련 명령

■ 하드 링크 만들기: ln

- 하드 링크는 같은 파일에 이름만 다르게 붙이는 것이지만, 복사는 완전히 독립적인 파일을 만드는 것

```
[user1@localhost ch2]$ cp data1 data1.cp
[user1@localhost ch2]$ ls -i
18660662 data1  18660668 data1.cp  18660662 data1.ln  18660663 temp
18660667 test.org
```

- inode 번호가 다르다는 것은 서로 독립적인 파일이라는 뜻

03 파일 관련 명령

■ 심볼릭 링크 만들기: -s 옵션

- 심볼릭 링크 : 윈도의 바로가기처럼 원본 파일을 가리키는 파일
- 심볼릭 링크는 ln 명령 에 -s 옵션을 사용하여 만듦

```
[user1@localhost ch2]$ ln -s data1 data1.sl
[user1@localhost ch2]$ ls -i
18660662 data1   18660668 data1.cp   18660662 data1.ln   18660669 data1.sl   18660663
temp  18660667 test.org
```

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

```
[user1@localhost ch2]$ ls -l data1.sl
lrwxrwxrwx. 1 user1 user1 5  8월  6 20:22 data1.sl -> data1
```

03 파일 관련 명령

■ 심볼릭 링크 만들기: -s 옵션

- 심볼릭 링크와 하드링크의 차이
  - 파일의 종류가 l(소문자 L)로 표시된다
  - 하드 링크의 개수가 하나다. 즉, 원본 파일에 이름을 추가하는 것이 아니다
  - 파일명 뒤에 원본 파일의 이름이 표시된다(-> data1).
  - inode 번호가 원본 파일과 다르다. 즉, 원본 파일과 심볼릭 링크 파일은 별개의 파일이다

- 심볼릭 링크의 파일 내용에는 원본 파일의 경로가 담겨 있음
- ln -s로 심 볼릭 링크를 생성할 때 지정한 원본 파일의 경로가 저장됨
- 심볼릭 링크의 내용을 출력하면 원본 파일의 경로가 출력되는 것이 아니라 원본 파일의 내용이 출력

03 파일 관련 명령

■ 심볼릭 링크 만들기: -s 옵션

```
[user1@localhost ch2]$ cat data1.sl
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1  localhost localhost.localdomain localhost6 localhost6.localdomain6
```

- 심볼릭 링크를 열고 수정하면 원본 파일이 수정됨
- 하드 링크는 같은 파일 시스템에 있는 파일에만 생성할 수 있음
  디렉터리에는 생성할 수 없고, 다른 파일 시스템에 있는 파일이나 디렉터리에도 생성할 수 없음
- 심볼릭 링크가 하드 링크에 비해 탁월한 점은 디렉터리에도 지정할 수 있고 파일 시스템이 달라도 생성할 수 있다는 것

```
[user1@localhost ch2]$ rm data1
[user1@localhost ch2]$ cat data1.sl
cat: data1.sl: 그런 파일이나 디렉터리가 없습니다
```

- 원본 파일이 삭제되면 심볼릭 링크로 연결할 수 없음

03 파일 관련 명령

■ 하드 링크와 심볼릭 링크

① test.org 파일의 하드 링크로 test.ln을 생성

```
[user1@localhost ch2]$ ln test.org test.ln
[user1@localhost ch2]$ ls
data1.cp  data1.ln  data1.sl  temp  test.ln  test.org
```

② test.org 파일의 하드 링크로 test.ln2를 생성

```
[user1@localhost ch2]$ ln test.org test.ln2
[user1@localhost ch2]$ ls
data1.cp  data1.ln  data1.sl  temp  test.ln  test.ln2  test.org
```

③ test.org와 test.ln, test.ln2의 inode 번호가 같음을 확인

```
[user1@localhost ch2]$ ls -i
18660668 data1.cp  18660662 data1.ln  18660669 data1.sl  18660663 temp
18660667 test.ln   18660667 test.ln2  18660667 test.org
```

03 파일 관련 명령

■ 하드 링크와 심볼릭 링크

④ test.org와 test.ln, test.ln2의 하드 링크 개수를 확인

```
[user1@localhost ch2]$ ls -l
(생략)
-rw-r--r--. 3 user1 user1 158  8월  6 20:06 test.ln
-rw-r--r--. 3 user1 user1 158  8월  6 20:06 test.ln2
-rw-r--r--. 3 user1 user1 158  8월  6 20:06 test.org
```

⑤ temp 디렉터리에 대한 심볼릭 링크로 tmp를 만듦

```
[user1@localhost ch2]$ ln -s temp tmp
[user1@localhost ch2]$ ls -l tmp
lrwxrwxrwx. 1 user1 user1 4  8월  6 20:33 tmp -> temp
```

⑥ temp와 tmp 디렉터리의 내용이 같음을 확인

⑦ tmp 디렉터리와 test.ln, test.ln2, test.org 파일을 모두 삭제

03 파일 관련 명령

■ 하드 링크와 심볼릭 링크

혼자해보기 하드 링크와 심볼릭 링크

❶ /etc/hosts 파일을 복사하여 work 파일을 생성한다.
❷ work 파일의 하드 링크로 work.ln을 생성한다.
❸ work 파일의 심볼릭 링크로 work.sl을 생성한다.
❹ work 파일을 복사하여 work.cp 파일을 생성한다.
❺ work.sl 파일을 gedit로 열어서 마지막 행을 삭제한다.
❻ work, work.ln, work.sl, work.cp 파일의 내용을 확인한다. 내용이 같지 않은 파일은 어느 것인가?
❼ work, work.ln, work.sl, work.cp 파일을 삭제한다.

03 파일 관련 명령

■ 빈 파일 만들기, 접근/수정 시간 변경하기: touch

touch

- 기능 빈 파일을 생성한다.
- 형식 touch [옵션] [파일]
- 옵션 -a: 접근 시간만 변경한다.
  -m: 수정 시간만 변경한다.
  -t [[CC]YY]MMDDhhmm[.ss]: 시간을 직접 입력한다.
- 사용 예 touch test

- test 파일을 만드는 예

```
[user1@localhost ch2]$ touch test
[user1@localhost ch2]$ ls -l test
-rw-r--r--. 1 user1 user1 0  8월  6 20:37 test
```

03 파일 관련 명령

■ 빈 파일 만들기, 접근/수정 시간 변경하기: touch

- touch 명령을 사용하여 data1.cp의 수정 시간을 현재 시간으로 바꾼 예

```
[user1@localhost ch2]$ ls -l data1.cp
-rw-r--r--. 1 user1 user1 158  8월  6 20:20 data1.cp
[user1@localhost ch2]$ date
2023. 08. 06. (일) 20:39:13 KST
[user1@localhost ch2]$ touch data1.cp
[user1@localhost ch2]$ ls -l data1.cp
-rw-r--r--. 1 user1 user1 158  8월  6 20:39 data1.cp
```

03 파일 관련 명령

■ 빈 파일 만들기, 접근/수정 시간 변경하기: touch

- -t 옵션을 사용하면 변경할 시간을 지정할 수 있음

시간 표시

- 형식 [[CC]YY]MMDDhhmm[.ss]
- 설명 CC: 연도의 첫 두 자리
  YY: 연도의 마지막 두 자리
  MM: 달(01~12 범위 내 지정)
  DD: 날짜(01~31 범위 내 지정)
  hh: 시간(00~23 범위 내 지정)
  mm: 분(00~59 범위 내 지정)
  ss: 초(00~59 범위 내 지정)

- CC를 지정하지 않으면 YY 기본값에 따라 CC를 자동으로 인식

표 2-5 연도 지정 방법

YY	69~99	00~68
CC	19	20

03 파일 관련 명령

■ 빈 파일 만들기, 접근/수정 시간 변경하기: touch

- 연도를 지정하지 않고 월, 일, 시간만 지정하여 test 파일의 수정 시간을 변경하는 예

```
[user1@localhost ch2]$ ls -l test
-rw-r--r--. 1 user1 user1 0  8월  6 20:37 test
[user1@localhost ch2]$ touch -t 12311200 test
[user1@localhost ch2]$ ls -l test
-rw-r--r--. 1 user1 user1 0 12월 31  2023 test
```

03 파일 관련 명령

■ 파일 내용 검색하기: grep

- grep : 파일 내에서 특정 문자열을 검색할 때 사용

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

03 파일 관련 명령

■ 파일 내용 검색하기: grep

- grep의 가장 기본적인 사용법으로서 인자로 지정한 문자열을 검색하는 예

```
[user1@localhost ch2]$ grep DHCP /etc/services
dhcp-failover  647/tcp              # DHCP Failover
dhcp-failover  647/udp              # DHCP Failover
```

- -n 옵션을 사용하면 검색된 행 번호도 함께 출력됨

```
[user1@localhost ch2]$ grep -n DHCP /etc/services
1413:dhcp-failover   647/tcp                # DHCP Failover
1414:dhcp-failover   647/udp                # DHCP Failover
```

03 파일 관련 명령

■ 파일 찾기: find

- find : 리눅스의 디렉터리 계층 구조에서 특정 파일이 어느 디렉터리에 있는지 찾아 줌
  파일의 생성 일자와 이름, 소유자 등 다양한 조건에 맞는 파일을 찾음

find

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

03 파일 관련 명령

■ 파일 찾기: find

- find 명령으로 검색한 모든 파일을 대상으로 동일한 작업을 수행하려면 -exec나 -ok 옵션 을 지정함
- /tmp 디렉터리 아래에 있는 user1 계정 소유의 파일을 전부 찾아서 삭제하려면 find 명령을 사용
- rm 명령을 사용했으므로 디렉터리는 삭제할 수 없다는 메시지가 출력된 예

```
[user1@localhost ch2]$ find /tmp -user user1 -exec rm {} \;
find: '/tmp/systemd-private-8ffa9a5505d343ebad65e618c92bfce2-dbus-broker.service-
1OKbpx': 허가 거부
find: '/tmp/systemd-private-8ffa9a5505d343ebad65e618c92bfce2-bluetooth.service-
IeKFUN': 허가 거부
find: '/tmp/systemd-private-8ffa9a5505d343ebad65e618c92bfce2-chronyd.service-Li1yDE':
허가 거부
(생략)
```

03 파일 관련 명령

■ 파일 찾기: find

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

```
[user1@localhost ch2]$ find /home -user user1 -ok rm {} \;
< rm ... /home/user1 > ? n
< rm ... /home/user1/.mozilla > ? n
< rm ... /home/user1/.mozilla/extensions > ? ^C
```

03 파일 관련 명령

■ 명령의 위치 찾기: whereis, which

- whereis나 which 명령을 사용하면 특정 명령이 어느 위치에 있는지를 찾아서 절대 경로를 출력함

whereis

- 기능 지정된 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

- whereis 명령은 환경 변수 $PATH와 $MANPATH에 지정된 디렉터리를 검색하여 파일의 위치를 찾음

```
[user1@localhost ch2]$ echo $PATH
/home/user1/.local/bin:/home/user1/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin
[user1@localhost ch2]$ echo $MANPATH
```

03 파일 관련 명령

■ 명령의 위치 찾기: whereis, which

- mv 명령의 위치를 whereis 명령으로 검색한 예

```
[user1@localhost ch2]$ whereis mv
mv: /usr/bin/mv /usr/share/man/man1/mv.1.gz /usr/share/man/man1p/mv.1p.gz
```

- which 명령은 앨리어스나 PATH 환경 변수로 지정된 경로에서 파일을 찾

which

- 기능 명령 파일의 위치를 찾아서 그 경로나 앨리어스를 출력한다.
- 형식 which [명령]
- 사용 예 which ls

03 파일 관련 명령

■ 명령의 위치 찾기: whereis, which

- which 명령은 사용자 초기화 파일에 지정된 앨리어스를 먼저 찾아보기 때문에 앨리어스가 지정된 명령이라면 해당 앨리어스가 출력됨
- 앨리어스가 없으면 PATH 환경 변수에 지정 된 경로를 순서대로 찾아가며 파일이 있는지 검색
- which 명령은 파일을 찾으면 절대 경 로를 출력하고 바로 종료
- which 명령은 최대 하나의 경로만 출력하며 이 경로는 우리가 명령을 입력할 때 실행되는 파일

```
[user1@localhost ch2]$ which mv
/usr/bin/mv
```

03 파일 관련 명령

grep, find 명령 사용하기

① /etc/services 파일에서 문자열 'MacOS'가 있는 행을 찾아 행 번호와 함께 출력

```
[user1@localhost ch2]$ grep -n MacOS /etc/services
1437:mac-srvr-admin  660/tcp       # MacOS Server Admin
1438:mac-srvr-admin  660/udp       # MacOS Server Admin
```

② 홈 디렉터리에서 파일명이 data1.cp인 파일이 있는지 검색

```
[user1@localhost ch2]$ find ~ -name data1.cp
/home/user1/Test/ch2/data1.cp
```

③ 홈 디렉터리에서 파일명이 data1.cp인 파일을 찾아 temp 디렉터리로 이동

```
[user1@localhost ch2]$ find ~ -name data1.cp -exec mv {} temp \;
[user1@localhost ch2]$ ls temp
data1.cp  hosts  text1  text2
```

03 파일 관련 명령

grep, find 명령 사용하기

혼자해보기 grep, find

❶ /etc/passwd 파일에서 'user1'이 들어간 행을 검색한다.
❷ find 명령으로 홈 디렉터리에서 data1.ln 파일을 검색한다.
❸ find 명령으로 홈 디렉터리에서 data1.ln 파일을 찾아 data1로 복사한다.

Thank you!

한빛아카데미 Hanbit Academy, Inc.