제4장 파일 사용

4.1 파일 복사

파일 복사: cp(copy)

- 사용법

```
$ cp [-i] 파일1 파일2
```

파일1을 파일2에 복사한다. -i는 대화형 옵션이다.

- 예

```
$ cp cs1.txt cs2.txt
$ ls -l cs1.txt cs2.txt
-rw-r--r-- 1 chang cs 2088 4월 16일 13:37 cs1.txt
-rw-r--r-- 1 chang cs 2088 4월 16일 13:45 cs2.txt
$ cp /etc/hosts hostnames
```

파일 복사: cp(copy)

- 대화형 옵션: cp -i
  - 복사 대상 파일과 이름이 같은 파일이 이미 존재하면 덮어쓰기(overwrite) !
  - 보다 안전한 사용법: 대화형 –i(interactive) 옵션을 사용

- 예

```
$ cp –i cs1.txt cs2.txt
cp: overwrite ‘cs2.txt’? n
```

파일 복사: cp(copy)

- 파일을 디렉터리로 복사

```
$ cp 파일 디렉터리
```

파일을 지정된 디렉터리에 복사한다.

```
$ cp 파일1 ... 파일n 디렉터리
```

여러 개의 파일들을 지정된 디렉터리에 모두 복사한다.

- 예

```
$ cp cs1.txt /tmp
$ ls -l /tmp/cs1.txt
-rw-r--r-- 1 chang cs 2088 4월 16일 14:31 /tmp/cs1.txt
$ cp cs1.txt cs2.txt /tmp
```

파일 복사: cp(copy)

- 디렉터리 전체 복사 : cp –r

```
$ cp [-r] 디렉터리1 디렉터리2
```

r은 리커전 옵션으로 디렉터리1 전체를 디렉터리2에 복사한다.

  - 하위 디렉터리를 포함한 디렉터리 전체를 복사

- 예

```
$ cp -r test temp
```

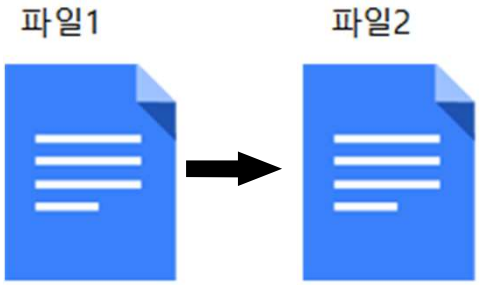
4.2 파일 이동

파일 이동: mv(move)

- 사용법

$ mv [-i] 파일1 파일2

파일1의 이름을 파일2로 변경한다. -i는 대화형 옵션이다.



- 예

```
$ mv cs2.txt cs3.txt
$ ls -l
-rw-r--r-- 1 chang faculty 2088 4월 16일 13:37 cs1.txt
-rw-r--r-- 1 chang faculty 2088 4월 16일 13:56 cs3.txt
```

파일 이동: mv(move)

- 대화형 옵션: cp -i
  - 이동 대상 파일과 이름이 같은 파일이 이미 존재하면 덮어쓰기(overwrite)
  - 보다 안전한 사용법: 대화형 –i(interactive) 옵션을 사용

- 예

```
$ mv –i cs1.txt cs3.txt
mv: overwrite ‘cs3.txt’? n
```

파일 이동: mv(move)

- 파일을 디렉터리로 이동

```
$ mv 파일 디렉터리
```

파일을 지정된 디렉터리로 이동한다.

```
$ mv 파일1 ... 파일n 디렉터리
```

여러 개의 파일들을 지정된 디렉터리로 모두 이동한다.

- 예

```
$ mv cs3.txt /tmp
$ ls -l /tmp/cs3.txt
-rw-r--r-- 1 chang cs 2088 4월 16일 13:56 /tmp/cs3.txt
$ mv cs1.txt cs3.txt /tmp
```

파일 이동: mv(move)

- 디렉터리 이름 변경

```
$ mv 디렉터리1 디렉터리2
```

디렉터리1을 지정된 디렉터리2로 이름을 변경한다.

- 예

```
$ mkdir temp
$ mv temp tmp
```

4.3 파일 삭제

파일 삭제: rm(remove)

- 사용법

```
$ rm [-i] 파일+
```

파일(들)을 삭제한다. -i는 대화형 옵션이다.

- 예

```
$ rm cs1.txt
$ rm cs1.txt cs3.txt
```

- 대화형 옵션 : rm -i

```
$ rm -i cs1.txt
rm: remove 'cs1.txt'? n
```

디렉터리 전체 삭제

- 디렉터리 전체 삭제: rm –r

```
$ rm [-ri] 디렉터리
```

–r은 리커전 옵션으로 디렉터리 아래의 모든 것을 삭제한다. –i는 대화형 옵션

- 예

```
$ rm test
rm: cannot remove 'test': 디렉터리입니다
$ rmdir test
rmdir: failed to remove 'test': 디렉터리가 비어있지 않음
$ rm –ri test
rm: descend into directory 'test'? y
rm: remove regular file 'test/cs3.txt'? y
Rm: remove regular file ‘test/cs1.txt’? y
rm: remove directory 'test'? y
```

4.4 링크

링크

- 링크
  - 기존 파일에 대한 또 하나의 새로운 이름

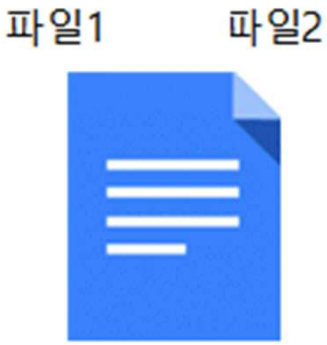
- 사용법

```
$ ln [-s] 파일1 파일2
```

파일1에 대한 새로운 이름(링크)로 파일2를 만들어 준다. -s 옵션은 심볼릭 링크

```
$ ln [-s] 파일1 디렉터리
```

파일1에 대한 링크를 지정된 디렉터리에 같은 이름으로 만들어 준다.



하드 링크(hard link)

- 하드 링크
  - 기존 파일에 대한 새로운 이름이라고 생각할 수 있다.
  - 실제로 기존 파일을 대표하는 i-노드를 가리켜 구현한다.

- 예

```
$ ln hello.txt hi.txt
$ ls -l
-rw------- 2 chang cs 15 11월 7일 15:31 hello.txt
-rw------- 2 chang cs 15 11월 7일 15:31 hi.txt
```

- 질문
  - 이 중에 한 파일의 내용을 수정하면 어떻게 될까?
  - 이 둘 중에 한 파일을 삭제하면 어떻게 될까?

심볼릭 링크(symbolic link)

- 심볼릭 링크
  - 다른 파일을 가리키고 있는 별도의 파일이다.
  - 실제 파일의 경로명을 저장하고 있는 일종의 특수 파일이다.
  - 이 경로명이 다른 파일에 대한 간접적인 포인터 역할을 한다.

- 예

```
$ ln -s hello.txt hi.txt
$ ls -l
-rw------- 1 chang cs 15 11월 7일 15:31 hello.txt
lrwxrwxrwx 1 chang cs 9 1월 24일 12:56 hi.txt -> hello.txt

$ ln -s /usr/bin/gcc cc
$ ls -l cc
lrwxrwxrwx. 1 chang chang 12 7월 21 20:09 cc -> /usr/bin/gcc
```

4.5 파일 속성

파일 속성(file attribute)

- 파일 크기, 종류, 접근권한, 링크 수, 소유자 및 그룹, 수정 시간

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

파일 속성	의미
파일 크기	파일의 크기(K 바이트 단위)
파일 종류	일반 파일(-), 디렉터리(d), 링크(l), 파이프(p), 소켓(s), 디바이스(b 혹은 c) 등의 파일 종류를 나타낸다.
접근권한	파일에 대한 소유자, 그룹, 기타 사용자의 읽기(r)/쓰기(w)/실행(x) 권한
하드 링크 수	파일에 대한 하드 링크 개수
소유자 및 그룹	파일의 소유자 ID 및 소유자가 속한 그룹
파일 크기	파일의 크기(바이트 단위)
최종 수정 시간	파일을 생성 혹은 최후로 수정한 시간

파일 종류

- 리눅스에서 지원하는 파일 종류

파일 종류	표시	설명
일반 파일	-	데이터를 갖고 있는 텍스트 파일 또는 이진 파일
디렉터리 파일	d	디렉터리 내의 파일들의 이름들과 파일 정보를 관리하는 파일
문자 장치 파일	c	문자 단위로 데이터를 전송하는 장치를 나타내는 파일
블록 장치 파일	b	블록 단위로 데이터를 전송하는 장치를 나타내는 파일
FIFO 파일	p	프로세스 간 통신에 사용되는 이름 있는 파이프
소켓	s	네트워크를 통한 프로세스 간 통신에 사용되는 파일
심볼릭 링크	l	다른 파일을 가리키는 포인터와 같은 역할을 하는 파일

파일 종류

- 사용법

```
$ file 파일
```

파일의 종류에 대한 자세한 정보를 출력한다.

- 예

```
$ file cs1.txt
cs1.txt: ASCII text
$ file a.out
a.out: ELF 32-bit LSB executable, ...
```

4.6 접근권한

접근권한(permission mode)

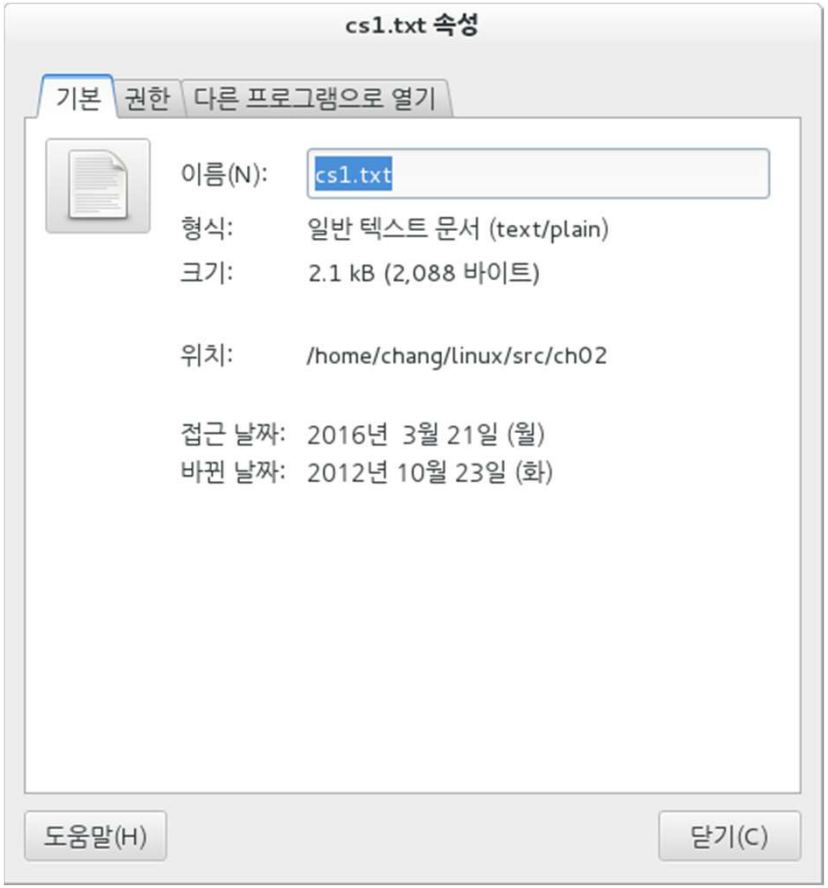
- 파일에 대한 읽기(r), 쓰기(w), 실행(x) 권한

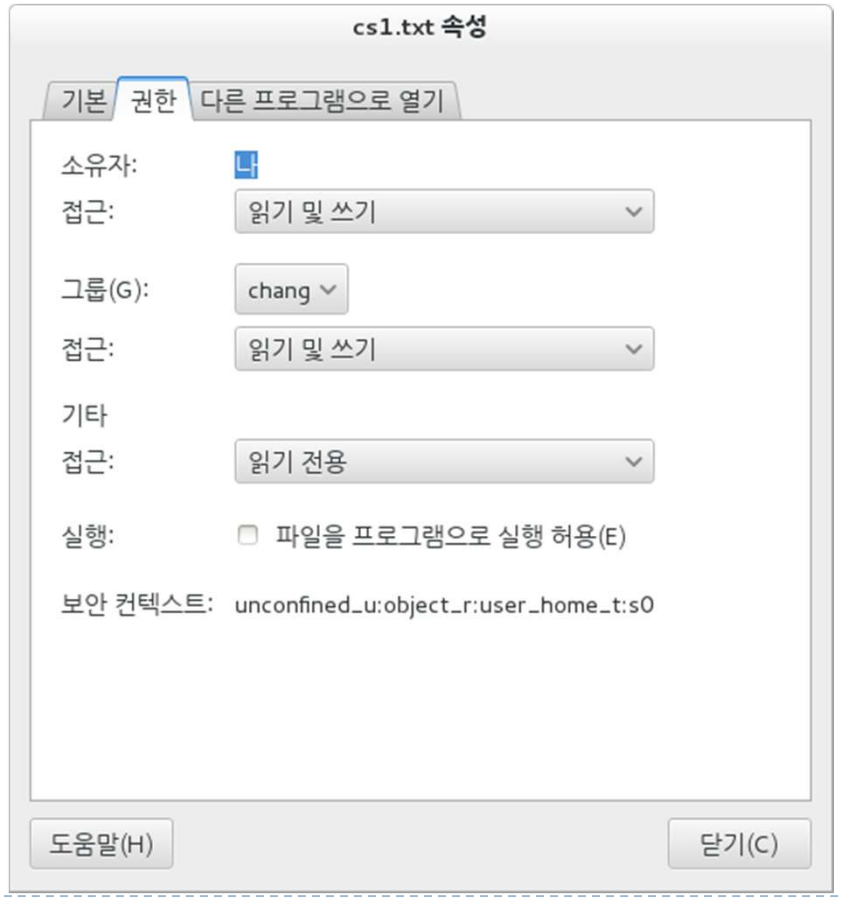
권한	파일	디렉터리
r	파일에 대한 읽기 권한	디렉터리 내에 있는 파일명을 읽을 수 있는 권한
w	파일에 대한 쓰기 권한	디렉터리 내에 파일을 생성하거나 삭제할 수 있는 권한
x	파일에 대한 실행 권한	디렉터리 내로 탐색을 위해 이동할 수 있는 권한

- 소유자(owner)/그룹(group)/기타(others)로 구분하여 관리
- 예: rwx r-x r-x

X 윈도우의 GNOME 데스크톱에서 속성 확인

```
$ ls -sl cs1.txt
4 -rw-rw-r-- 1 chang cs 2088 10월 23일 13:37 cs1.txt
```





접근권한의 예

접근권한	의미
rwxrwxrwx	소유자, 그룹, 기타 사용자 모두 읽기,쓰기,실행 가능
rwxr-xr-x	소유자만 읽기,쓰기,실행 가능, 그룹, 기타 사용자는 읽기,실행 가능
rw-rw-r--	소유자와 그룹만 읽기,쓰기 가능, 기타 사용자는 읽기만 가능
rw-r--r--	소유자만 읽기,쓰기 가능, 그룹과 기타 사용자는 읽기만 가능
rw-r-----	소유자만 읽기,쓰기 가능 그룹은 읽기만 가능
rwx------	소유자만 읽기,쓰기,실행 가능

4.7 접근권한 변경

접근권한 변경: chmod(change mode)

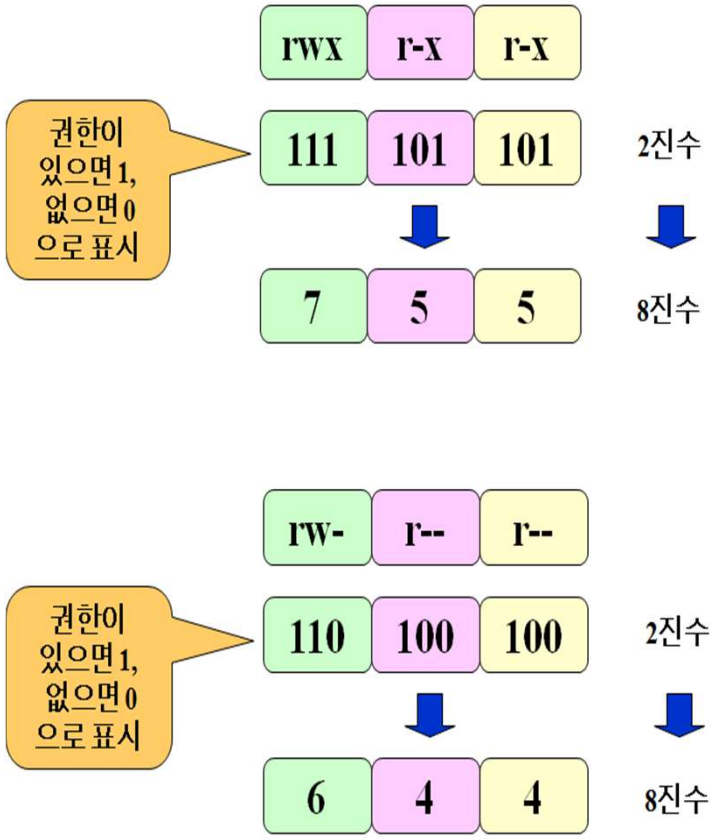
- 사용법

```
$ chmod [-R] 접근권한 파일 혹은 디렉터리
```

파일 혹은 디렉터리의 접근권한을 변경한다. -R 옵션을 사용하면 지정된 디렉터리 아래의 모든 파일과 하위 디렉터리에 대해서도 접근권한을 변경한다.

접근권한 표현: 8진수

- 접근권한 8진수 변환



- 사용 예

```
$ chmod 644 cs1.txt
$ ls -l cs1.txt
-rw-r--r-- 1 chang … cs1.txt
```

접근권한	8진수
rwxrwxrwx	777
rwxr-xr-x	755
rw-rw-r--	664
rw-r--r--	644
rw-r-----	640
rwx------	700

접근권한 표현: 기호

- 기호를 이용한 접근권한 변경

사용자범위	연산자	권한
[u|g|o|a]+	[+|-|=]	[r|w|x]+

구분	기호와 의미
사용자 범위	u(user:소유자), g(group:그룹), o(others:기타 사용자), a(all:모든 사용자)
연산자	+(권한 추가), -(권한 제거), =(권한 설정)
권한	r(읽기 권한), w(쓰기 권한), x(실행 권한)

기호를 이용한 접근권한 변경

- 예

```
$ chmod g-w cs1.txt
$ ls -l cs1.txt
-rw-r--r-- 1 chang cs 2088 4월 16일 13:37 cs1.txt

$ chmod o-r cs1.txt
$ ls -l cs1.txt
-rw-r----- 1 chang cs 2088 4월 16일 13:37 cs1.txt

$ chmod g+w,o+rw cs1.txt
$ ls -l cs1.txt
-rw-rw-rw- 1 chang cs 2088 4월 16일 13:37 cs1.txt
```

4.8 기타 파일 속성 변경

소유자 변경: chown(change owner)

- 사용법

```
$ chown 사용자 파일
$ chown [-R] 사용자 디렉터리
```

파일 혹은 디렉터리의 소유자를 지정된 사용자로 변경한다.

-R 옵션: 디렉터리 아래의 모든 파일과 하위 디렉터리에 대해서도 소유자를 변경한다.

- 예

```
$ chown hong cs1.txt
chown: changing ownership of 'cs1.txt': 명령을 허용하지 않음
$ su
암호:
$ chown hong cs1.txt
$ ls -l cs1.txt
-rw-r--r--. 1 hong cs 2088 10월 21 16:25 cs1.txt
```

그룹 변경: chgrp(change group)

- 사용법

```
$ chgrp 그룹 파일

$ chgrp [-R] 그룹 디렉터리
```

파일 혹은 디렉터리의 그룹을 지정된 그룹으로 변경한다. -R 옵션을 사용하면 지정된 디렉터리 아래의 모든 파일과 하위 디렉터리에 대해서도 그룹을 변경한다.

최종 수정 시간 변경: touch

- 사용법

```
$ touch 파일
```

파일의 최종 사용 시간과 최종 수정 시간을 현재 시간으로 변경한다.

- 예

```
$ touch cs1.txt
$ ls -l cs1.txt
-rw-r--r--. 1 chang cs 905 7월 13 19:06 cs1.txt
```

핵심 개념

- 링크는 기존의 파일에 대한 또 하나의 새로운 이름으로 하드 링크와 심볼릭(소프트) 링크가 있다.
- 파일은 이름뿐만 아니라 타입, 크기, 소유자, 접근권한, 수정 시간 등의 파일 속성을 갖는다.
- 파일의 접근권한은 소유자, 그룹, 기타로 구분하여 관리한다.