

File Link

(Hard Link & Soft Link)

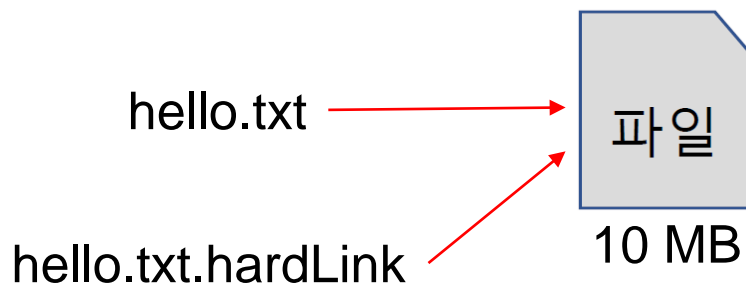
소프트웨어융합학부

노기섭 교수

(kafa46@cju.ac.kr)

link의 종류와 개념

하드링크 (hard link)

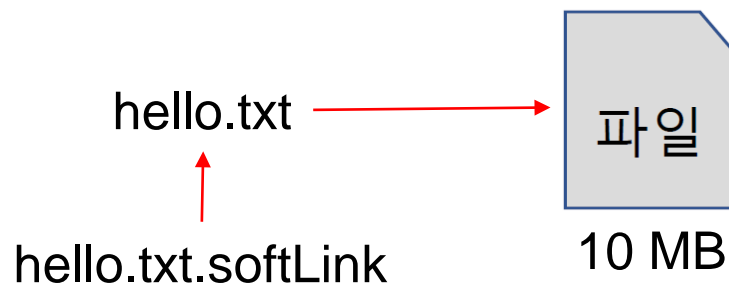


$10\text{ MB} + 10\text{ MB} \neq 20\text{ MB}$

$10\text{ MB} + 10\text{ MB} = 10\text{ MB}$

- 원본과 사본 구분이 없음
- 하나의 내용을 변경하면 동시 변경
- 하드 링크를 지워도 원본은 보존
- 원본을 삭제해도 하드링크는 보존

소프트 링크 (soft link, Symbolic link)

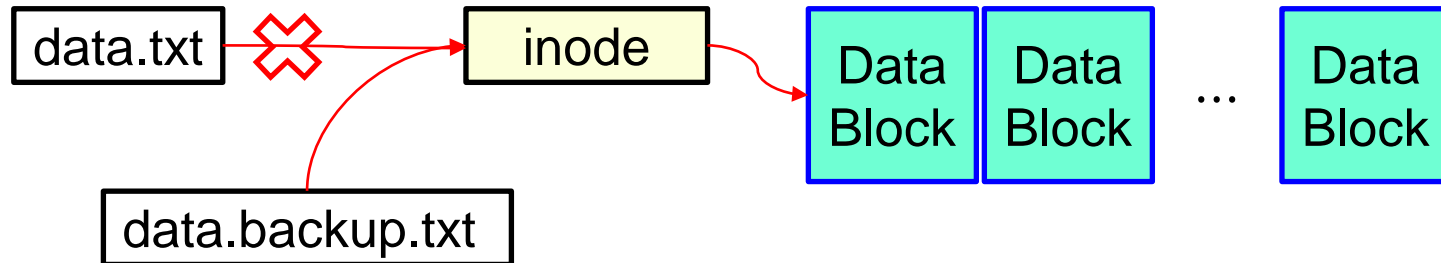


$10\text{ MB} + \alpha\text{ Bytes} = 10.x\text{ MB}$

- 소프트링크는 바로가기 역할(주로 사용)
- 하나의 내용을 변경하면 동시 변경
- 심볼릭 링크를 지워도 원본은 보존됨
- 파일을 삭제하면 소프트링크는 동작 안함

Hard Link, 조금만 더 구체적으로...

■ Hard link 구조



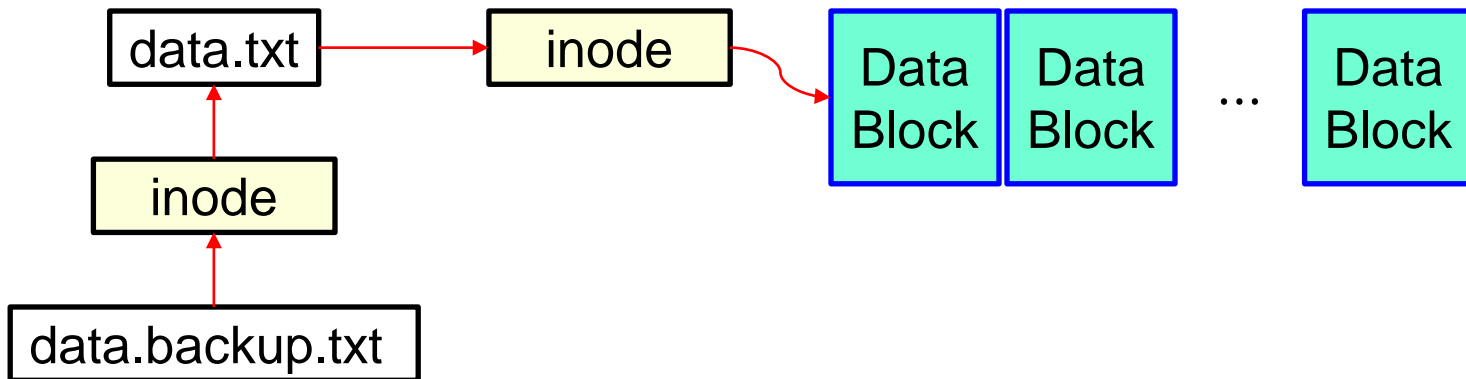
■ Hard link 활용

- “원본과 사본의 구분이 없다” 를 생각하면 쉽습니다.
 - 백업할 때 주로 사용 (동등한 파일이 생성됨)
 - 모든 파일을 통으로 복사하는 것보다 하나의 파일을 추가로 생성
- 백업 파일을 만들고 원본 파일은 삭제
- 원본과 사본 구분이 없기 때문에 실수 삭제도 가능
- 언제 데이터는 삭제되나요? → inode를 가리키는(참조하는, reference 하는) 포인터가 없을 때 입니다.

Soft Link , 조금만 더 구체적으로...

■ Soft link 구조

- Soft link는 아무리 삭제해도 원본에 영향 없음
 - Hard link는 원본/사본 구분이 없기 때문에 모두 지울 위험성도 존재
- 원본을 삭제하면 soft link는 더 이상 작동 안됨
 - ➔ 원본에 영향을 주지 않는 바로가기를 기능이 필요할 때 사용
(Hard link 보다 약간 느릴 수 있음)



link 생성 명령어

■ 도움말

```
$ ln --help
Usage: ln [OPTION]... [-T] TARGET LINK_NAME
      or: ln [OPTION]... TARGET
      or: ln [OPTION]... TARGET... DIRECTORY
      or: ln [OPTION]... -t DIRECTORY TARGET...
In the 1st form, create a link to TARGET with the name LINK_NAME.
In the 2nd form, create a link to TARGET in the current directory.
In the 3rd and 4th forms, create links to each TARGET in DIRECTORY.
Create hard links by default, symbolic links with --symbolic.
By default, each destination (name of new link) should not already exist.
When creating hard links, each TARGET must exist. Symbolic links
can hold arbitrary text; if later resolved, a relative link is
interpreted in relation to its parent directory.
```

```
[OPTION]
:
-s, --symbolic      make symbolic links instead of hard links
:
```

- Hard link
\$ ln 원본파일 링크이름
- Soft link
\$ln -s 원본파일 링크이름

간단한 Link 만들기 실습

```
$ echo "hello world" > hello.txt
```

사전준비(파일 생성)

```
$ ln -s hello.txt hello.softlink
```

소프트링크(심볼릭링크) 만들기

```
$ ln hello.txt hello.hardlink
```

하드링크

```
$ ls -ali
```

파일 링크 확인 (all + long + inode(주소))

원본 삭제 후 하드링크 작동 확인 해보기

원본 삭제 후 소프트링크 작동 확인 해보기

언제 Soft Link를 사용할까?

■ 언제 Soft (Symbolic) Link 사용할까?

/bin: 이것은 Linux의 기본 명령이 등록 되어 있는 directory

/usr/bin: /usr/bin은 /bin과 달리 일반 사용자가 사용하는 명령어가 등록된 디렉토리

- 시스템 운영을 보다 편리하게 하고 싶을 때
- 예시: Python 버전이 여러 개 설치된 경우 → Python2.7 & Python3.7

```
$ which python3
```

```
/usr/bin/python3
```

```
$ ll /usr/bin/python*
```

- “python” 명령어를 이용해 원하는 버전의 python을 적절하게 실행하고 싶다!
- 다른 여러 버전이 생기더라도 동일한 명령어를 설정하고 싶다!

```
lrwxrwxrwx 1 root root 18 Sep 28 07:02 /usr/bin/python -> /usr/bin/python2.7*
```

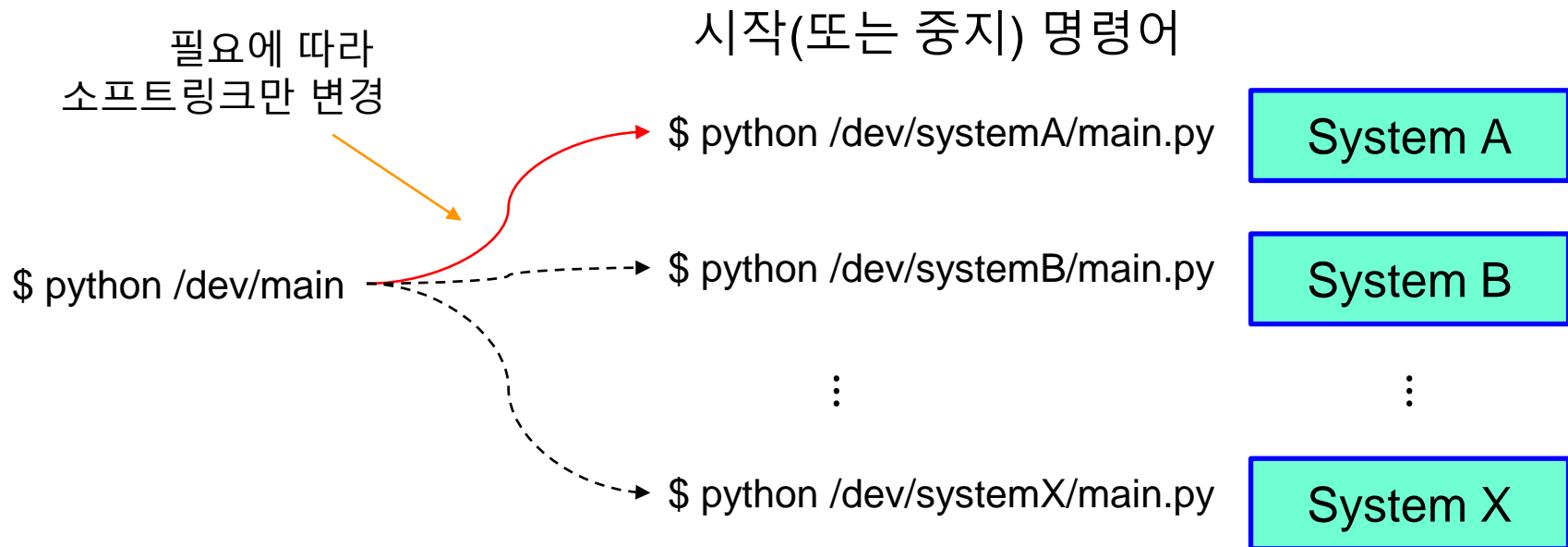
```
lrwxrwxrwx 1 root root 9 Jul 28 2021 /usr/bin/python2 -> python2.7*
```

```
-rwxr-xr-x 1 root root 3592504 Jul 1 10:30 /usr/bin/python2.7*
```

```
-rwxr-xr-x 1 root root 5905480 Jun 29 12:14 /usr/bin/python3.10*
```

일반적인 Soft Link 활용

■ 여러 개 시스템(또는 실행파일)을 만든 경우



소프트웨어
꼰대 강의!



수고하셨습니다 ..^..