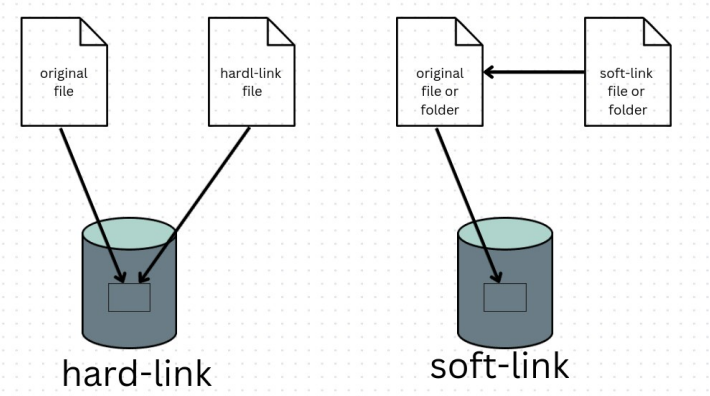
Soft Link



A soft link (or symbolic link) in Linux is a special type of file that acts as a pointer or shortcut to another file or directory. It allows you to access the target file without affecting its actual content or location. Unlike a hard link, a soft link can point to files on different file systems or partitions.

Creating Soft-Link

```
$ ln -s target link_name
```

example:

Target can be both file path or directory path. Here let's try with file.

```
 amit@archlinux  ~/Test/example2  ln -s file1.txt soft-link.txt
 amit@archlinux  ~/Test/example2  ls -l
total 4
drwxr-xr-x 2 amit users 4096 Dec  5 18:11 apple
-rw-r--r-- 1 amit users    0 Dec  5 18:11 file1.txt
lrwxrwxrwx 1 amit users    9 Dec  5 18:12 soft-link.txt -> file1.txt
 amit@archlinux  ~/Test/example2
```

Advantages of Soft Links

1. **Flexibility:**
    - Can link to files or directories on different file systems or partitions.

Example:
In my system /mnt and /home are different filesystem. Creating softlink of file1.txt present in /home/amit/Test/example2 in /mnt filesystem.

```
 amit@archlinux  ~/Test/example2  ls
apple  file1.txt
 amit@archlinux  ~/Test/example2
```

```
✘ amit@archlinux  /mnt  sudo ln -s /home/amit/Test/example2/file1.txt softlink.txt
[sudo] password for amit:
 amit@archlinux  /mnt  ls
softlink.txt  usbdrive
 amit@archlinux  /mnt
```

2. **Easily identifiable:**
    - Soft links are visible and can be distinguished from regular files (shown with an arrow in the `ls -l` output).

Example:

```
 amit@archlinux  ~/Test/example2  ls -l
total 0
-rw-r--r-- 1 amit users 0 Dec  5 18:11 file1.txt
lrwxrwxrwx 1 amit users 9 Dec  5 18:25 soft-link.txt -> file1.txt
 amit@archlinux  ~/Test/example2
```

3. **Independence from file changes:**
    - Changes to the target file are immediately reflected when accessed via the soft link.

```
 amit@archlinux  ~/Test/example2  echo "This is the example of softlink" >> file1.txt
 amit@archlinux  ~/Test/example2  cat file1.txt
This is the example of softlink
 amit@archlinux  ~/Test/example2  cat soft-link.txt
This is the example of softlink
 amit@archlinux  ~/Test/example2
```

4. **Reduced duplication:**
    - No need to create multiple copies of the same file.

Limitations of Soft Links

1. Broken links

- If the target file is deleted or moved, the soft link becomes a "dangling" or "broken" link.

```
 amit@archlinux  ~/Test/example2  rm file1.txt
 amit@archlinux  ~/Test/example2  ls -l
total 0
lrwxrwxrwx 1 amit users 9 Dec  5 18:25 soft-link.txt -> file1.txt
 amit@archlinux  ~/Test/example2  cat soft-link.txt
cat: soft-link.txt: No such file or directory
 ✘ amit@archlinux  ~/Test/example2
```

2. Performance overhead

- Slightly slower than hard links since the system must resolve the link's path.

3. Less secure

- Mismanaged symbolic links can create security vulnerabilities (e.g., pointing to system files).

4. Storage Usage

- Soft links consume a small amount of disk space for storing the link's metadata.

```
amit@archlinux ~/Test/example2 ls -li
total 4
10620888 drwxr-xr-x 2 amit users 4096 Dec  5 18:38 original
10621105 lrwxrwxrwx 1 amit users    8 Dec  5 18:38 soft-link -> original
```

Here, 4096 bytes is size of original directory and 8 bytes is size of soft-link.

Usecases:

1. Shortcuts to Files or Directories:

- Quickly access commonly used files without navigating deep directories.

  `$ ln -s /var/log/syslog /syslog_link`

2. Shared Configuration Files:

- Manage a single configuration file linked to different services.

  `$ ln -s /etc/global_config.conf /etc/service1/config.conf`

3. Version Management:

- Point a symbolic link to the current version of software or libraries.

  $ **`ln -s /opt/software/v2.0/current  /opt/software/latest`**