What is the difference between soft and hard links

s905060.gitbooks.io/site-reliability-engineer-handbook/content/what_is_the_difference_between_soft_and_hard_links.html

  - LSOF
  - SORT
  - Xargs
  - Iptables
  - xargs vs. exec {}
  - Hdparm
  - UNIQ
  - STAT
  - Execute Commands in the Background
  - TAIL
  - WGET
  - Date
  - FDISK
  - Mount
  - Make SWAP File
  - Create a New User
  - Create a New Group
  - Setup SSH Passwordless Login in OpenSSH
  - Parted
  - RSYNC
  - YUM
  - RPM
  - APT
  - Install from Source
  - Log Rotate
  - FREE
  - DF
  - DU
  - Sysctl
  - NICE
  - Renice
  - PS
  - DD
  - BC
  - LDD
  - getcap, setcap and file capabilities
  - Linux_Basename
  - PMAP
  - Alternative
  - Readlink
  - logrotate

  - PIDOF
  - Dmidecode
  - lshw
  - printenv
  - SS
  - w
  - Strace
  - pstree
  - USERMOD
  - ltrace
  - ethtool
  - IP
  - Sar
  - nethogs
  - zip
  - FPM
  - getent
  - ipmitool
  - Building RPMs
  - Megacli
    - Megacli package version
  - RKhunter
  - fping
  - blkid
  - FSCK
  - Package Manager
  - mktemp
  - ls
  - Comm
  - taskset
  - fio
  - tree
  - ARP
  - lsblk
- How-To
  - CentOS: nf_conntrack: table full, dropping packet
  - How To Fix “Error: database disk image is malformed” On CentOS \/ Fedora
  - Finding the PID of the process using a specific port?
  - How-To create hashed SSH password
  - How to display and kill zombie processes
  - Shell command to bulk change file extensions in a directory (Linux)

- 8 Powerful Awk Built-in Variables – FS, OFS, RS, ORS, NR, NF, FILENAME, FNR
- Changing the Time Zone
- HOW DO I DISABLE SSH LOGIN FOR THE ROOT USER?
- How-To rename the extension for a batch of files?
- How-To disable IPv6 on RHEL6 \/ CentOS 6 \/ etc
- How to clear the ARP cache on Linux?
- How-To crontab running as a specific user
- Ansible – exclude host from playbook execution
- HOWTO: Use Wireshark over SSH
- How-To Change Network Interface Name
- How-To Creating a Partition Size Larger Than 2TB
- Hot-To Linux Hard Disk Format Command
- Hadoop Troubleshooting
- Hive Troubleshooting
- HowTo Set up hostbased authentication for passphraseless SSH communication.
- Difference between a cold and warm reboot
- ls -l explained
- df falsely showing 100 per cent disk usage
- FSCK explained
- Manually generate password for \/etc\/shadow
- How To Change Timezone on a CentOS 6 and 7
- Setting ssh private key forwarding
- Persist keys in ssh-agent on OS X
- SSH Essentials: Working with SSH Servers, Clients, and Keys
- How to Change JVM Heap Setting (-Xms -Xmx) of Tomcat – Configure setenv.sh file – Run catalina.sh
- SSH ProxyCommand example: Going through one host to reach another server
- How to get Linux's TCP state statistics
- Linux TCP retransmission rate calculation
- How to determine OOM
- How-to check Java process heapsize
- Troubleshooting network issues
- How to check what sudo acces a user has?
- How to copy your key to a remote server?
- Linux date and Unix timstamp conversion
- SSH client personalized configuration
- How to Error Detection and Correction
- How To Kerberos
- How to identify defective DIMM from EDAC error on Linux

    - Howto Install and Configure Cobbler on Centos 6
    - How To Use GPG to Encrypt and Sign Messages on an Ubuntu 12.04 VPS
    - HowTo: Debug Crashed Linux Application Core Files Like A Pro
    - Create init script in CentOS 6
    - Linux Change Disk Label Name on EXT2 \/ EXT3 \/ EXT4 File Systems
    - How to retrieve and change partition's UUID Universally Unique Identifier on linux
    - Using Text-Mode Serial Console Redirection
    - How to Write Linux Init Scripts Based on LSB Init Standard
    - How to create a Debian package
    - How to create a RPM Package
    - How to solve EDAC DIMM CE Error
    - How to solve fsck.ext4: Unable to resolve UUID\/LABEL
    - How to expand an existing LSI raid array using MegaCli
    - How to change user GID and UID in Ubuntu
    - How to read a segfault kernel log message
    - How to add cron job via command line
    - How to restrict process CPU usage using nice, cpulimit, and cgroups
- Data Structure
- Service
    - Cloud-Init
    - ETCD
    - RESTful API HTTP methods
    - Web cache
    - Mesos
    - ELK
    - Cassandra
    - Hive
        - Hive notes
    - Elasticsearch
    - Scylla
    - Zookeeper
- Automation Tool
    - Ansible
    - Salt
        - Salt use notes
- Networking Devices
    - Cisco
    - Juniper
- Version Control
    - SVN

Soft Links =>

1. Soft link files will have different inode numbers then source file
2. If original file deleted then soft link file be of no use
3. Soft links are not updated
4. Can create links between directories
5. Can cross file system boundaries

Hard Links =>

1. Hard links will have the same inode number as source file
2. Hard links can not link directories
3. Can not cross file system boundaries
4. Hard links always refers to the source, even if moved or removed

---

The different semantics between hard and soft links make them suitable for different things.

Hard links:

indistinguishable from other directory entries, because every directory entry is hard link "original" can be moved or deleted without breaking other hard links to the same inode only possible within the same filesystem permissions must be the same as those on the "original" (permissions are stored in the inode, not the directory entry) can only be made to files, not directories

Symbolic links (soft links)

simply records that point to another file path. (ls -l will show what path a symlink points to) will break if original is moved or deleted. (In some cases it is actually desirable for a link to point to whatever file currently occupies a particular location) can point to a file in a different filesystem can point to a directory on some file system formats, it is possible for the symlink to have different permissions than the file it points to (this is uncommon)

Hard Link vs Soft Link

Once again, one of the very basic question asked in the interviews, What is the difference between Hard links and Soft links. You can explain few basic things, but emphasizing on internals about the linux filesystem, is something which can impress the interviewer.

What is a Soft Link or Symbolic Link or Symlink ? Symbolic links or Symlinks are the easiest to understand, because for sure you have used them, at least when you were using Windows. Soft links are very similar to what we say “Shortcut” in windows, is a way to link to a file or directory. Symlinks doesn’t contain any information about the destination file or contents of the file, instead of that, it simply contains the pointer to the location of the destination file. In more technical words, in soft link, a new file is created with a new inode, which have the pointer to the inode location of the original file. This can be better explained with a diagram:

Soft Links on Linux

Symbolic links are created with the “ln” command in linux. The syntax of the command is:

```
$ ln -s
```

-s = This flag tells to create a symlink (if you don’t use this it will create a hard link, which we will talk about soon).

For Example, if you want to create a soft link of one fo your favorite application, like gedit, on your desktop, use the command like this:

```
$ ln -s /usr/bin/gedit ~/Desktop/gedit
```

I hope now the concept of Soft Links should be clear.

What is a Hard Link ? Hard link is a bit different object when compared to a symlink. In softlink a new file and a new Inode is created, but in hard link, only an entry into directory structure is created for the file, but it points to the inode location of the original file. Which means there is no new inode creation in the hard link. This can be explained like this:

Hard Links in Linux

So, in hard link, you are referencing the inode directly on the disk, which means that there should be a way to know how many hard links exist to a file. For the same, in the inode information, you have an option for “links”, which will tell how many links exists to a file. You can find the same information by using this command:

```
$ stat <file name>
```

```
$ stat 01
Size: 923383 Blocks: 1816 IO Block: 4096 regular file
Device: 803h/2051d Inode: 12684895 Links: 3
Access: (0644/-rw-r–r–) Uid: ( 0/ root) Gid: ( 0/ root)
Access: 2012-09-07 01:46:54.000000000 -0500
Modify: 2012-04-27 06:22:02.000000000 -0500
Change: 2012-04-27 06:22:02.000000000 -0500
```

In this example, it means that the specific file have 2 hard links, which makes the count to 3.

You can create a hard link with the same command “ln” like this

```
# ln
```

So, to create a hard link of gedit program on your desktop, you will use the command like this:

```
# ln /usr/bin/gedit ~/Desktop/gedit
```

Now, the bigger question is, who will decide what is better and when to use soft link or hard link

When to use Soft Link: Link across filesystems: If you want to link files across the filesystems, you can only use symlinks/soft links. Links to directory: If you want to link directories, then you must be using Soft links, as you can’t create a hard link to a directory. When to use Hard Link: Storage Space: Hard links takes very negligible amount of space, as there are no new inodes created while creating hard links. In soft links we create a file which consumes space (usually 4KB, depending upon the filesystem) Performance: Performance will be slightly better while accessing a hard link, as you are directly accessing the disk pointer instead of going through another file.

Moving file location: If you move the source file to some other location on the same filesystem, the hard link will still work, but soft link will fail.

Redundancy: If you want to make sure safety of your data, you should be using hard link, as in hard link, the data is safe, until all the links to the files are deleted, instead of that in soft link, you will lose the data if the master instance of the file is deleted.

The above points gives you a small idea where to use what, but doesn't tell you that those are the only options. Everything depends on your setup.

How file is deleted having hard links: So, as it's pretty clear from the above article that hard links are just the reference to the main file location, and even if you delete one link, the data will still be intact. So, to remove a hard link, you need to remove all the links, which are referring to the file. Once the "link count" goes to "0", then the inode is removed by the filesystem, and file is deleted.

FAQs:

Q. What is the one line answer to the question "What is the main difference between hard links & soft links" ?

A. A softlink will have a different Inode number than the source file, which will be having a pointer to the source file but hardlink will be using the same Inode number as the source file.

Q. How can I find all the Soft Links in my system ?

A. Use this command for the same "find /etc -type l -exec ls -li {} \;"

Q. How can I find all the files having Hard Links in my system ?

A. Use this command for the same "find / -links +2 -type f -exec ls -li {} \;"

Q. How to find whether a file is a softlink ?

A. Simply using this command "ls -l" will tell you whether a file is pointing to some other file or not.

Q. How to check whether a file have any softlink pointing to it ?

A. Till now, I am not aware of any way to do that. If I will find any, I will surely update my post.

Q. How can I find out the source file of a hard link ?

A. No, you can't find out the source file of a hard link. Once hard link is created, there is no way to tell which was the first file created.

Q. Can I make a Soft link to a Hard link and Vice Versa ?

A. Yes, both soft links and hard links acts as normal files of the file system, so you can do both.