Last updated on: 2020/01/30 03:28

File Hierarchy System

The Linux filesystem hierarchy starts with the **root** represented by a **/** character. Under the root can be found other directories containing task specific files. The hierarchy conforms to a standard called the **Linux File Hierarchy System**.

Directory Contents

Directory	Contents
/bin	Contains user programs such as ls, cp e.t.c.. Note that under RHEL 7 / CentOS7, this is a soft link (shorcut) to **/usr/bin**.
/boot	Contains bootloader files, kernels and initrd (INItial Ram Disk) files.
/dev	Contains nodes for accessing all the peripherals and devices connected to the system. The *udev* binary takes care of dynamically creating and deleting the relevant nodes automatically.
/etc	Contains static configuration files.
/home	Contains a directory for each registered user of the system except for root.
/lib	Contains common 32 bit libraries for applications and modules. Note that under RHEL 7 / CentOS7, this is a soft link (shorcut) to **/usr/lib**.
/lib64	Contains common 64 bit libraries for applications and modules. Note that under RHEL 7 / CentOS7, this is a soft link (shorcut) to **/usr/lib64**.
/lost+found	Contains damaged file fragments found by the *fsck* command.
/media	Contains a folder for each of the mounted external file systems (CDRom DVDRom, USB Key e.t.c.).
/misc	**RHEL 5, 6 and CentOS 5, 6 only**. Contains mount points for local directories mounted via the automounter.
/mnt	Contains a folder for each external file system mounted temporarily by root.
/net	**RHEL 5, 6 and CentOS 5, 6 only**. Contains mount points for network directories mounted via the automounter.
/opt	Contains optional application packages.
/proc	Contains a virtual file system that documents kernel and process status information as text files.
/root	The home directory of the root user.
/run	Replaces the /var/run directory. Note that under RHEL 7 / CentOS7, /var/run is a soft link (shorcut) to **/run**.
/sbin	Contains essential system administration binaires. Note that under RHEL 7 / CentOS7, this is a soft link (shorcut) to **/usr/sbin**.

Directory	Contents
/selinux	Contains a virtual file system used by SELINUX.
/srv	Contains site specific data **s**e**rv**ed by the system (www,ftp,databases e.t.c.).
/sys	Contains a virtual file system that describes devices for *udev*.
/tmp	Contains the temporary files created by the system and by applications.
/usr	Contains user commands in */usr/bin*, HOWTOs in */usr/share/doc*, manuals in */usr/share/man* and is the *Secondary Hierarchy* for read-only user data.
/var	Contains variable files. i.e. files that continually change such as log files and spool files.

Directory Structure

```
[trainee@centos5 /]$ ls -l
total 138
drwxr-xr-x   2 root root  4096 août 25 13:27 bin
drwxr-xr-x   4 root root  1024 août 25 14:03 boot
drwxr-xr-x  11 root root  3800 août 25 14:06 dev
drwxr-xr-x 102 root root 12288 août 25 14:06 etc
drwxr-xr-x   3 root root  4096 août 25 13:36 home
drwxr-xr-x  14 root root  4096 août 25 13:42 lib
drwx------   2 root root 16384 août 25 13:16 lost+found
drwxr-xr-x   3 root root  4096 août 25 14:06 media
drwxr-xr-x   2 root root     0 août 25 14:06 misc
drwxr-xr-x   2 root root  4096 mai 11  2011 mnt
drwxr-xr-x   2 root root     0 août 25 14:06 net
drwxr-xr-x   4 root root  4096 août 25 13:44 opt
dr-xr-xr-x 143 root root     0 août 25 14:05 proc
drwxr-x---   4 root root  4096 août 25 14:01 root
drwxr-xr-x   2 root root 12288 août 25 14:03 sbin
drwxr-xr-x   4 root root     0 août 25 14:05 selinux
drwxr-xr-x   2 root root  4096 mai 11  2011 srv
drwxr-xr-x  11 root root     0 août 25 14:05 sys
drwxrwxrwt  14 root root  4096 août 25 14:09 tmp
```

```
drwxr-xr-x  14 root root  4096 août 25 13:23 usr
drwxr-xr-x  21 root root  4096 août 25 13:31 var
```

```
[trainee@centos6 /]$ ls -l
total 98
dr-xr-xr-x.   2 root root  4096  9 août  12:52 bin
dr-xr-xr-x.   5 root root  1024  7 déc.   2014 boot
drwxr-xr-x.  19 root root  3820 25 août  11:29 dev
drwxr-xr-x. 119 root root 12288 25 août  11:28 etc
drwxr-xr-x.   3 root root  4096  3 mai    2013 home
dr-xr-xr-x.  20 root root 12288  9 août  12:52 lib
drwx------.   2 root root 16384  3 mai    2013 lost+found
drwxr-xr-x.   2 root root  4096  7 déc.   2014 media
drwxr-xr-x.   2 root root     0 25 août  11:28 misc
drwxr-xr-x.   3 root root  4096  5 juil. 12:22 mnt
drwxr-xr-x.   2 root root     0 25 août  11:28 net
drwxr-xr-x.   6 root root  4096  7 déc.   2014 opt
dr-xr-xr-x. 154 root root     0 25 août  11:27 proc
dr-xr-x---.  10 root root  4096  9 août  12:58 root
dr-xr-xr-x.   2 root root 12288  9 août  12:52 sbin
drwxr-xr-x.   7 root root     0 25 août  11:27 selinux
drwxr-xr-x.   2 root root  4096 23 sept.  2011 srv
drwxr-xr-x.  13 root root     0 25 août  11:27 sys
drwxrwxrwt.  16 root root  4096 25 août  11:30 tmp
drwxr-xr-x.  13 root root  4096  3 mai    2013 usr
drwxr-xr-x.  22 root root  4096  9 août  12:50 var
```

```
[trainee@centos7 /]$ ls -l
total 32
lrwxrwxrwx.   1 root root    7 Mar  8 13:41 bin -> usr/bin
dr-xr-xr-x.   4 root root 4096 Jun  4 15:00 boot
drwxr-xr-x.  19 root root 3280 Jul  7 15:55 dev
drwxr-xr-x. 131 root root 8192 Jul 23 17:05 etc
drwxr-xr-x.   4 root root   47 Jul  5 14:11 home
```

```
lrwxrwxrwx.   1 root root    7 Mar  8 13:41 lib -> usr/lib
lrwxrwxrwx.   1 root root    9 Mar  8 13:41 lib64 -> usr/lib64
drwxr-xr-x.   2 root root    6 Jun 10  2014 media
drwxr-xr-x.   3 root root   18 Jul  5 13:57 mnt
drwxr-xr-x.   4 root root   47 Jun  4 09:36 opt
dr-xr-xr-x. 177 root root    0 Jul  7 15:53 proc
dr-xr-x---.   5 root root 4096 Aug 25 11:31 root
drwxr-xr-x.  35 root root 1100 Jul 23 15:40 run
lrwxrwxrwx.   1 root root    8 Mar  8 13:41 sbin -> usr/sbin
drwxr-xr-x.   2 root root    6 Jun 10  2014 srv
dr-xr-xr-x.  13 root root    0 Jul  7 15:53 sys
drwxrwxrwt.  25 root root 4096 Jul 23 15:40 tmp
drwxr-xr-x.  13 root root 4096 Mar  8 13:41 usr
drwxr-xr-x.  22 root root 4096 Jul  7 15:53 var
```

File Types

The three major file types under Linux are :

- Ordinary files,
- Directories,
- Special files or Devices.

Note that :

- Ordinary files can be anything from text files to binaries.
- The length of a file name is limited to 225 characters, including the file extension.
- Linux is case sensitive.
- If a file name starts with a dot (**.**), it is a hidden file.

The mount command

In order to be able to use external file systems, such as a CDRom or DVDRom, Linux needs to be informed of their availability. This is accomplished by using the **mount** command:

```
# mount /dev/<special_file> /mnt/<directory_name> [Enter]
```

where **/dev/<special_file>** is the file system to mount and **/mnt/<directory_name>** is the target directory where the mounted file system will be available to the system. The directory **/mnt/<directory_name>** must exist prior to using the **mount** command.

In the case where the **mount** command is used without options, the current mounted file systems are shown:

```
[root@centos5 ~]# mount
/dev/sda2 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
```

```
[root@centos6 /]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda1 on /boot type ext3 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
```

```
[root@centos7 ~]# mount
```

```
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=1449668k,nr_inodes=362417,mode=755)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,mode=755)
tmpfs on /sys/fs/cgroup type tmpfs (rw,nosuid,nodev,noexec,seclabel,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup
(rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/systemd/systemd-cgroups-agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpuacct,cpu)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/net_cls type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
configfs on /sys/kernel/config type configfs (rw,relatime)
/dev/sda2 on / type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=32,pgrp=1,timeout=300,minproto=5,maxproto=5,direct)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel)
mqueue on /dev/mqueue type mqueue (rw,relatime,seclabel)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
nfsd on /proc/fs/nfsd type nfsd (rw,relatime)
/dev/sda1 on /boot type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
```

Important : Note that with RHEL 5 / CentOS 5 the default filesystem is **ext3**, with RHEL 6 / CentOS 6 the default filesystem is **ext4**, whereas with RHEL 7 / CentOS the default filesystem is **xfs**. Please see the unit **Managing Disks, Swap Space and Filesystems** for further coursework concerning ext3, ext4, xfs filesystems.

Command Line Switches

The following switches can be used with the mount command:

```
[root@centos7 ~]# mount --help

Usage:
 mount [-lhV]
 mount -a [options]
 mount [options] [--source] <source> | [--target] <directory>
 mount [options] <source> <directory>
 mount <operation> <mountpoint> [<target>]

Options:
 -a, --all               mount all filesystems mentioned in fstab
 -c, --no-canonicalize   don't canonicalize paths
 -f, --fake              dry run; skip the mount(2) syscall
 -F, --fork              fork off for each device (use with -a)
 -T, --fstab <path>      alternative file to /etc/fstab
 -h, --help              display this help text and exit
 -i, --internal-only     don't call the mount.<type> helpers
 -l, --show-labels       lists all mounts with LABELs
 -n, --no-mtab           don't write to /etc/mtab
 -o, --options <list>    comma-separated list of mount options
 -O, --test-opts <list>  limit the set of filesystems (use with -a)
```

```
 -r, --read-only         mount the filesystem read-only (same as -o ro)
 -t, --types <list>      limit the set of filesystem types
     --source <src>      explicitly specifies source (path, label, uuid)
     --target <target>   explicitly specifies mountpoint
 -v, --verbose           say what is being done
 -V, --version           display version information and exit
 -w, --rw, --read-write  mount the filesystem read-write (default)

 -h, --help     display this help and exit
 -V, --version  output version information and exit

Source:
 -L, --label <label>     synonym for LABEL=<label>
 -U, --uuid <uuid>       synonym for UUID=<uuid>
 LABEL=<label>           specifies device by filesystem label
 UUID=<uuid>             specifies device by filesystem UUID
 PARTLABEL=<label>       specifies device by partition label
 PARTUUID=<uuid>         specifies device by partition UUID
 <device>                specifies device by path
 <directory>             mountpoint for bind mounts (see --bind/rbind)
 <file>                  regular file for loopdev setup

Operations:
 -B, --bind              mount a subtree somewhere else (same as -o bind)
 -M, --move              move a subtree to some other place
 -R, --rbind             mount a subtree and all submounts somewhere else
 --make-shared           mark a subtree as shared
 --make-slave            mark a subtree as slave
 --make-private          mark a subtree as private
 --make-unbindable       mark a subtree as unbindable
 --make-rshared          recursively mark a whole subtree as shared
 --make-rslave           recursively mark a whole subtree as slave
 --make-rprivate         recursively mark a whole subtree as private
 --make-runbindable      recursively mark a whole subtree as unbindable
```

```
For more details see mount(8).
```

The /etc/fstab file

In the case where the **mount** command is used with the **-a** option, all mount points specified in the **/etc/fstab** file are mounted:

```
[root@centos6 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Fri May  3 13:33:42 2013
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=b9f29672-c84e-4d3b-b132-189758a084eb /                       ext4    defaults        1 1
UUID=01baf03d-df0d-479b-b3e4-81ce63b8dec3 /boot                   ext4    defaults        1 2
UUID=2646a33a-65f3-4501-9ced-9459435fd774 swap                    swap    defaults        0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
```

```
[root@centos7 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Sun Mar  8 12:38:10 2015
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
```

```
UUID=b35de665-5ec8-4226-a533-58a1b567ac91 /                       xfs     defaults        1 1
UUID=e8d3bd48-1386-411c-9675-41c3f8f1a309 /boot                   xfs     defaults        1 2
UUID=11a4d11d-81e4-46a7-82e0-7796cd597dc9 swap                    swap    defaults        0 0
```

Understanding the /etc/fstab file

Each line in **/etc/fstab** has 6 fields :

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
Special file or UUID or Virtual File System	Mount Point	Filesystem Type	Comma separated list of options	Used by the dump command (1 = dump, 0 or empty = do not dump)	The order in which the *fsck* command checks the disks/partitions at boot time

The **UUID** (*Universally Unique Identifier*) is a randomly generated 128 bit string that is automatically generated by the system when a filesystem is created on the partition.

Mountpoint Options

The most important mount point options are as follows:

Option	Filesystem	Description	Default Value
defaults	All	Use default options: rw, suid, dev, exec, auto, nouser, and async.	N/A [1)]
auto/noauto	All	Do or do not mount when “mount -a” is given.	auto
rw/ro	All	Mount the filesystem read-write/read-only.	rw
suid/nosuid	All	Allow/disallow set-user-identifier or set-group-identifier bits to take effect.	suid
dev/nodev	All	Interpret/do not interpret character or block special devices on the filesystem.	dev
exec/noexec	All	Permit/do not permit execution of binaries.	exec
sync/async	All	All I/O to the filesystem should be done synchronously/asynchronously.	async
user/nouser	All	Allow/disallow a user to mount. The mount point is read from the /etc/fstab file. Only the user that mounted the filesystem can unmount it.	N/A
users	All	Allow every user to mount and unmount the filesystem.	N/A

Option	Filesystem	Description	Default Value
owner	All	Allow device owner to mount.	N/A
atime/noatime	POSIX	Do not use noatime feature, then the inode access time is controlled by kernel defaults/Do not update inode access times on this filesystem	atime
uid=value	Non-Linux filesystems	Set the owner of the root of the filesystem.	root
gid=value	Non-Linux filesystems	Set the group of the root of the filesystem.	N/A
umask=value	Non-Linux filesystems	Set the umask. The default is the umask of the current process. The value is given in octal.	N/A
dmask=value	Non-Linux filesystems	Set the umask applied to directories only. The value is given in octal.	Current processes' umask
fmask=value	Non-Linux filesystems	Set the umask applied to regular files only. The value is given in octal.	Current processes' umask

The umount command

To unmount a file system, you need to use the **umount** command. For example:

```
# umount /mnt/target_directory [Entrée]
```

Command Line Switches

The following switches can be used with the umount command:

```
[root@centos7 ~]# umount --help

Usage:
 umount [-hV]
 umount -a [options]
 umount [options] <source> | <directory>
```

```
Options:
 -a, --all               unmount all filesystems
 -A, --all-targets       unmount all mountpoins for the given device
                         in the current namespace
 -c, --no-canonicalize   don't canonicalize paths
 -d, --detach-loop       if mounted loop device, also free this loop device
     --fake              dry run; skip the umount(2) syscall
 -f, --force             force unmount (in case of an unreachable NFS system)
 -i, --internal-only     don't call the umount.<type> helpers
 -n, --no-mtab           don't write to /etc/mtab
 -l, --lazy              detach the filesystem now, and cleanup all later
 -O, --test-opts <list>  limit the set of filesystems (use with -a)
 -R, --recursive         recursively unmount a target with all its children
 -r, --read-only         In case unmounting fails, try to remount read-only
 -t, --types <list>      limit the set of filesystem types
 -v, --verbose           say what is being done

 -h, --help     display this help and exit
 -V, --version  output version information and exit

For more details see umount(8).
```

Unix File Systems

Each file system contains the following :

- superblock
- inodes
- data blocks

Superblock

The superblock contains :

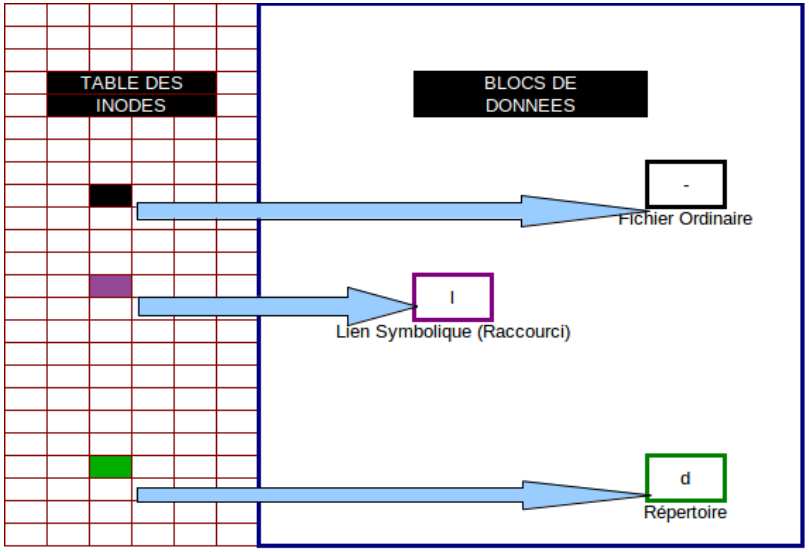
- the block size,
- the size of the file system,
- the number of mounts for the file system,
- a pointer to the root of the file system,
- pointers to the free inodes,
- pointers to free data blocks.

Inodes

Each file is represented by an **inode**. An inode contains the following information:

- the file type : **-**, **d**, **l**, **b**, **c**, **p**, **s**,
- file permissions, for example : **rwx rw- r–**,
- the number of hard links,
- the UID of the file creator or the current UID attributed by the **chown** command,
- the GID of the creating process or the current GID attributed by the **chgrp** command,
- the file size in bytes,
- the date of the last modification of the file's inode content : **ctime**,
- the date of the last modification of the file contents : **mtime**,
- the date of the last access : **atime**,
- allocation addresses that point to the data blocks used by the file.

For example:



Execute the following command:

```
[root@centos7 ~]# ls -ld /dev/console /dev/sda1 /etc /etc/passwd
crw-------.   1 root root 5, 1 Sep 28 10:48 /dev/console
brw-rw----.   1 root disk 8, 1 Sep 28 10:48 /dev/sda1
drwxr-xr-x. 136 root root 8192 Sep 28 10:57 /etc
-rw-r--r--.   1 root root 2267 Sep 22 14:29 /etc/passwd
```

The first character of each line indicates the file type:

- **-** - an ordinary file,
- **d** - a directory,
- **l** - a symbolic link,
- **b** - a bloc type peripheral,
- **c** - a character type peripheral,
- **p** - a named pipe for communication between processes,
- **s** - a network socket.

To see the inode numbers, execute the previous command with, in addition, the **-i** option:

```
[root@centos7 ~]# ls -ldi /dev/console /dev/sda1 /etc /etc/passwd
    4683 crw-------.   1 root root 5, 1 Sep 28 10:48 /dev/console
    8107 brw-rw----.   1 root disk 8, 1 Sep 28 10:48 /dev/sda1
33595521 drwxr-xr-x. 136 root root 8192 Sep 28 10:57 /etc
35670335 -rw-r--r--.   1 root root 2267 Sep 22 14:29 /etc/passwd
```

Data Blocks

File data is stored in data blocks. In the case of a directory, the data block contains a table referencing the inodes and the names of the contents of the directory.

The name of the file is stored in the parent directory's data block and not in the inode. This means that a file can be referenced by one or more different names. To add a name to a data block, you need to create what is called a **hard link**.

Hard (Physical) Links

A hard link is created by using the **ln** command.

```
[root@centos7 ~]# cd /tmp; mkdir inode; cd inode; touch file1; ls -ali
total 0
287056 drwxr-xr-x.  2 root root  60 Sep 28 12:16 .
 11071 drwxrwxrwt. 10 root root 240 Sep 28 12:16 ..
287058 -rw-r--r--.  1 root root   0 Sep 28 12:16 file1
```

file1 shows an inode number of **287058** and a single name, indicated by the number **1** in the third column:

```
287058 -rw-r--r--.  1 root root   0 Sep 28 12:16 file1
```

now create the hard link and check the result:

```
[root@centos7 inode]# ln file1 file2
```

```
[root@centos7 inode]# ls -ali
total 0
287056 drwxr-xr-x.  2 root root  80 Sep 28 12:18 .
 11071 drwxrwxrwt. 10 root root 240 Sep 28 12:16 ..
287058 -rw-r--r--.  2 root root   0 Sep 28 12:16 file1
287058 -rw-r--r--.  2 root root   0 Sep 28 12:16 file2
```

Now you can see two lines, one for file1 and a second for file2:

```
287058 -rw-r--r--.  2 root root   0 Sep 28 12:16 file1
287058 -rw-r--r--.  2 root root   0 Sep 28 12:16 file2
```

file1 and **file2** are referenced by the same inode. As a result the number of names has been increased to two in the thrid column.

> **Important** - Hard links cannot be created across file system boundaries. A hard link can only be created if the source file exists.

Soft (Symbolic) Links

A soft link is a shortcut to a file or directory. A soft link is created using the same **ln** command with the **-s** option.

```
[root@centos7 inode]# ln -s file1 file3
[root@centos7 inode]# ls -ali
total 0
287056 drwxr-xr-x.  2 root root 100 Sep 28 12:30 .
 11071 drwxrwxrwt. 10 root root 240 Sep 28 12:16 ..
287058 -rw-r--r--.  2 root root   0 Sep 28 12:16 file1
287058 -rw-r--r--.  2 root root   0 Sep 28 12:16 file2
333487 lrwxrwxrwx.  1 root root   5 Sep 28 12:30 file3 -> file1
```

Note here that the soft link is referenced by a separate inode.

> **Important** - A soft link can be created across file system boundaries and can be created even when the source file does not exist.

<html>

</html>

---

1)

Not Applicable