Version : **2023.01**

Updated : 2023/05/10 16:15

LCE401 - File Hierarchy System

Contents

Linux 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**.

```
[trainee@centos8 ~]$ cd /
```

```
[trainee@centos8 /]$ ls -l
total 18
lrwxrwxrwx.   1 root root    7 May 10  2019 bin -> usr/bin
dr-xr-xr-x.   6 root root 1024 May  8 08:14 boot
drwxr-xr-x.  19 root root 3020 Sep  6 07:59 dev
drwxr-xr-x.  91 root root 8192 Sep  6 07:59 etc
drwxr-xr-x.   3 root root   21 May  8 07:42 home
lrwxrwxrwx.   1 root root    7 May 10  2019 lib -> usr/lib
lrwxrwxrwx.   1 root root    9 May 10  2019 lib64 -> usr/lib64
drwxr-xr-x.   2 root root    6 May 10  2019 media
drwxr-xr-x.   2 root root    6 May 10  2019 mnt
drwxr-xr-x.   2 root root    6 May 10  2019 opt
dr-xr-xr-x. 126 root root    0 Sep  6 07:59 proc
dr-xr-x---.   2 root root  135 Sep  1 11:10 root
drwxr-xr-x.  29 root root  860 Sep  6 08:00 run
lrwxrwxrwx.   1 root root    8 May 10  2019 sbin -> usr/sbin
drwxr-xr-x.   2 root root    6 May 10  2019 srv
dr-xr-xr-x.  13 root root    0 Sep  6 07:59 sys
drwxrwxrwt.   8 root root  172 Sep  6 08:00 tmp
drwxr-xr-x.  12 root root  144 May  8 07:38 usr
drwxr-xr-x.  21 root root 4096 May  8 08:13 var
```

Directory	Contents
/bin	Contains user programs such as ls, cp e.t.c.. Note that under RHEL 8 / CentOS8, 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 8 / CentOS8, this is a soft link (shorcut) to **/usr/lib**.
/lib64	Contains common 64 bit libraries for applications and modules. Note that under RHEL 8 / CentOS8, this is a soft link (shorcut) to **/usr/lib64**.
/media	Contains a folder for each of the mounted external file systems (CDRom DVDRom, USB Key e.t.c.).
/mnt	Contains a folder for each external file system mounted temporarily by root.

Directory	Contents
/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 8 / CentOS8, /var/run is a soft link (shorcut) to **/run**.
/sbin	Contains essential system administration binaires. Note that under RHEL 8 / CentOS8, this is a soft link (shorcut) to **/usr/sbin**.
/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.

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:

```
[trainee@centos8 /]$ su -
Password: fenestros
[root@centos8 ~]# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=1881944k,nr_inodes=470486,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 (ro,nosuid,nodev,noexec,seclabel,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup
(rw,nosuid,nodev,noexec,relatime,seclabel,xattr,release_agent=/usr/lib/systemd/systemd-cgroups-
agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,devices)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,net_cls,net_prio)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,memory)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,cpuset)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,pids)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,cpu,cpuacct)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,perf_event)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,blkio)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,hugetlb)
```

```
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,freezer)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,rdma)
none on /sys/kernel/tracing type tracefs (rw,relatime,seclabel)
configfs on /sys/kernel/config type configfs (rw,relatime)
/dev/mapper/cl_centos8-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=31,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=3826)
mqueue on /dev/mqueue type mqueue (rw,relatime,seclabel)
debugfs on /sys/kernel/debug type debugfs (rw,relatime,seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
/dev/sda1 on /boot type ext4 (rw,relatime,seclabel)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=382500k,mode=700,uid=1000,gid=1000)
tmpfs on /run/user/42 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=382500k,mode=700,uid=42,gid=42)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
```

This informations is stored in the **/etc/mtab** file:

```
[root@centos8 ~]# cat /etc/mtab
sysfs /sys sysfs rw,seclabel,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
devtmpfs /dev devtmpfs rw,seclabel,nosuid,size=1881944k,nr_inodes=470486,mode=755 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,seclabel,nosuid,nodev 0 0
devpts /dev/pts devpts rw,seclabel,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,seclabel,nosuid,nodev,mode=755 0 0
tmpfs /sys/fs/cgroup tmpfs ro,seclabel,nosuid,nodev,noexec,mode=755 0 0
cgroup /sys/fs/cgroup/systemd cgroup
rw,seclabel,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/systemd/systemd-cgroups-agent,name=systemd
0 0
pstore /sys/fs/pstore pstore rw,seclabel,nosuid,nodev,noexec,relatime 0 0
bpf /sys/fs/bpf bpf rw,nosuid,nodev,noexec,relatime,mode=700 0 0
```

```
cgroup /sys/fs/cgroup/devices cgroup rw,seclabel,nosuid,nodev,noexec,relatime,devices 0 0
cgroup /sys/fs/cgroup/net_cls,net_prio cgroup rw,seclabel,nosuid,nodev,noexec,relatime,net_cls,net_prio 0 0
cgroup /sys/fs/cgroup/memory cgroup rw,seclabel,nosuid,nodev,noexec,relatime,memory 0 0
cgroup /sys/fs/cgroup/cpuset cgroup rw,seclabel,nosuid,nodev,noexec,relatime,cpuset 0 0
cgroup /sys/fs/cgroup/pids cgroup rw,seclabel,nosuid,nodev,noexec,relatime,pids 0 0
cgroup /sys/fs/cgroup/cpu,cpuacct cgroup rw,seclabel,nosuid,nodev,noexec,relatime,cpu,cpuacct 0 0
cgroup /sys/fs/cgroup/perf_event cgroup rw,seclabel,nosuid,nodev,noexec,relatime,perf_event 0 0
cgroup /sys/fs/cgroup/blkio cgroup rw,seclabel,nosuid,nodev,noexec,relatime,blkio 0 0
cgroup /sys/fs/cgroup/hugetlb cgroup rw,seclabel,nosuid,nodev,noexec,relatime,hugetlb 0 0
cgroup /sys/fs/cgroup/freezer cgroup rw,seclabel,nosuid,nodev,noexec,relatime,freezer 0 0
cgroup /sys/fs/cgroup/rdma cgroup rw,seclabel,nosuid,nodev,noexec,relatime,rdma 0 0
none /sys/kernel/tracing tracefs rw,seclabel,relatime 0 0
configfs /sys/kernel/config configfs rw,relatime 0 0
/dev/mapper/cl_centos8-root / xfs rw,seclabel,relatime,attr2,inode64,logbufs=8,logbsize=32k,noquota 0 0
selinuxfs /sys/fs/selinux selinuxfs rw,relatime 0 0
systemd-1 /proc/sys/fs/binfmt_misc autofs
rw,relatime,fd=31,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=3826 0 0
mqueue /dev/mqueue mqueue rw,seclabel,relatime 0 0
debugfs /sys/kernel/debug debugfs rw,seclabel,relatime 0 0
hugetlbfs /dev/hugepages hugetlbfs rw,seclabel,relatime,pagesize=2M 0 0
fusectl /sys/fs/fuse/connections fusectl rw,relatime 0 0
/dev/sda1 /boot ext4 rw,seclabel,relatime 0 0
sunrpc /var/lib/nfs/rpc_pipefs rpc_pipefs rw,relatime 0 0
tmpfs /run/user/1000 tmpfs rw,seclabel,nosuid,nodev,relatime,size=382500k,mode=700,uid=1000,gid=1000 0 0
tmpfs /run/user/42 tmpfs rw,seclabel,nosuid,nodev,relatime,size=382500k,mode=700,uid=42,gid=42 0 0
gvfsd-fuse /run/user/1000/gvfs fuse.gvfsd-fuse rw,nosuid,nodev,relatime,user_id=1000,group_id=1000 0 0
```

Important : Note that the filesystem on **/dev/sda1** is **ext4** wheras the filesystem on **/dev/mapper/cl_centos8-root** est **xfs**. For further coursework concerning ext3, ext4, xfs and btrfs filesystems please see the module **LCE504 - Managing Disks, Swap Space and Filesystems** from the **LCE500 - CentOS 8 Linux (RHEL 8) - Technician** course.

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@centos8 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Wed Jun 16 06:21:32 2021
#
# 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.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/cl_centos8-root /                       xfs     defaults        0 0
UUID=1c04981e-5317-4b73-9695-3ce25246835d /boot                   ext4    defaults        1 2
/dev/mapper/cl_centos8-swap 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
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 (Obsolete)	Non-Linux filesystems	Set the umask applied to directories only. The value is given in octal.	Current processes' umask
dir_mode=value (Replaces dmask)	Non-Linux filesystems	Set the umask applied to directories only. The value is given in octal.	Current processes' umask
fmask=value (Obsolete)	Non-Linux filesystems	Set the umask applied to regular files only. The value is given in octal.	Current processes' umask
file_mode=value (Replaces fmask)	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]
```

Unix File Systems

Each file system contains the following :

- Superblock
- Inodes
- Data blocks
- Indirect 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.

The Superblock is duplicated every 8 or 16MB when using ext3 and ext4 filesystems. To repair a broken superblock, use the following command :

```
# e2fsck -f -b 8193 /dev/sda1 [Enter]
```

For example, to view the primary and backup superblock locations on ext filesystems, use the following command:

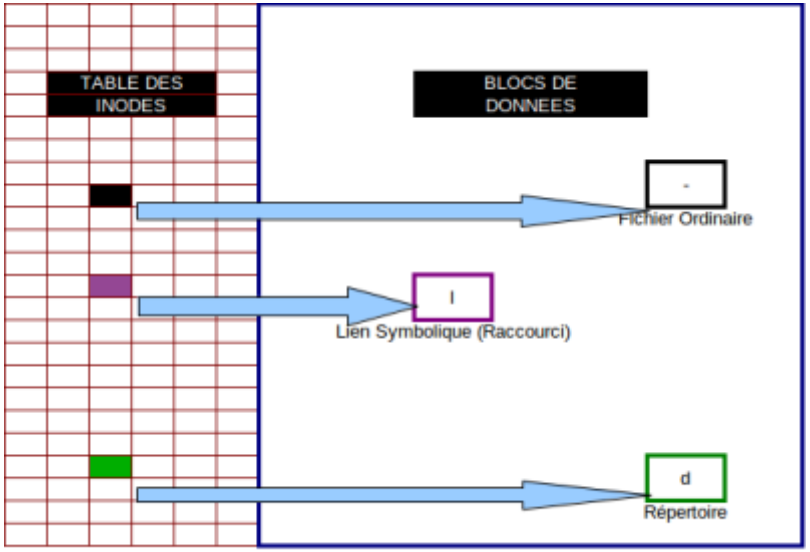
```
[root@centos8 ~]# dumpe2fs /dev/sda1 | grep -i superbloc
dumpe2fs 1.44.6 (5-Mar-2019)
  Primary superblock at 1, Group descriptors at 2-4
  Backup superblock at 8193, Group descriptors at 8194-8196
  Backup superblock at 24577, Group descriptors at 24578-24580
  Backup superblock at 40961, Group descriptors at 40962-40964
  Backup superblock at 57345, Group descriptors at 57346-57348
  Backup superblock at 73729, Group descriptors at 73730-73732
  Backup superblock at 204801, Group descriptors at 204802-204804
  Backup superblock at 221185, Group descriptors at 221186-221188
```

Inodes

Chaque fichier est représenté par un **inode**. L'inode contient : 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@centos8 ~]# ls -ld /dev/console /dev/sda1 /etc /etc/passwd
crw-------.  1 root root 5, 1 Sep  6 07:59 /dev/console
brw-rw----.  1 root disk 8, 1 Sep  6 07:59 /dev/sda1
drwxr-xr-x. 91 root root 8192 Sep  6 07:59 /etc
-rw-r--r--.  1 root root 1383 May  8 07:42 /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@centos8 ~]# ls -ldi /dev/console /dev/sda1 /etc /etc/passwd
   8816 crw-------.  1 root root 5, 1 Sep  6 07:59 /dev/console
  12701 brw-rw----.  1 root disk 8, 1 Sep  6 07:59 /dev/sda1
8388737 drwxr-xr-x. 91 root root 8192 Sep  6 07:59 /etc
8893162 -rw-r--r--.  1 root root 1383 May  8 07:42 /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@centos8 ~]# cd /tmp; mkdir inode; cd inode; touch file1; ls -ali
total 0
 9199611 drwxr-xr-x. 2 root root  22 Sep  6 08:31 .
16800396 drwxrwxrwt. 9 root root 185 Sep  6 08:31 ..
 8398272 -rw-r--r--. 1 root root   0 Sep  6 08:31 file1
```

file1 shows an inode number of **8398272** and a single name, indicated by the number **1** in the third column:

8398272 -rw-r–r–. 1 root root 0 Sep 6 08:31 file1

Now create the hard link and check the result:

```
[root@centos8 inode]# ln file1 file2
```

```
[root@centos8 inode]# ls -ali
total 0
 9199611 drwxr-xr-x. 2 root root  38 Sep  6 08:32 .
16800396 drwxrwxrwt. 9 root root 185 Sep  6 08:31 ..
 8398272 -rw-r--r--. 2 root root   0 Sep  6 08:31 file1
 8398272 -rw-r--r--. 2 root root   0 Sep  6 08:31 file2
```

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

8398272 -rw-r–r–. 2 root root 0 Sep 6 08:31 file1
8398272 -rw-r–r–. 2 root root 0 Sep 6 08:31 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 : A hard link can only be created when the two files are on the same filesystem and when 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@centos8 inode]# ln -s file1 file3
[root@centos8 inode]# ls -ali
total 0
 9199611 drwxr-xr-x. 2 root root  54 Sep  6 08:34 .
16800396 drwxrwxrwt. 9 root root 185 Sep  6 08:31 ..
 8398272 -rw-r--r--. 2 root root   0 Sep  6 08:31 file1
 8398272 -rw-r--r--. 2 root root   0 Sep  6 08:31 file2
 8398273 lrwxrwxrwx. 1 root root   8 Sep  6 08:34 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.

---

1)
Not Applicable