Version : **2024.01**

Last update : 2024/11/22 17:46

RH12401 - File Hierarchy System

Contents

Set Up your Keyboard

The current keyboard layout of the Red Hat 9.4 VM is **fr** :

```
[root@redhat9 ~]# localectl status
System Locale: LANG=en_US.UTF-8
```

```
    VC Keymap: fr
   X11 Layout: fr,us
  X11 Variant: ,
```

Find the appropriate keymap for your host PC:

```
[root@redhat9 ~]# localectl list-keymaps
ANSI-dvorak
adnw
al
al-plisi
amiga-de
amiga-us
apple-a1048-sv
apple-a1243-sv
apple-a1243-sv-fn-reverse
apple-internal-0x0253-sv
apple-internal-0x0253-sv-fn-reverse
applkey
at
at-mac
at-nodeadkeys
atari-de
atari-se
atari-uk-falcon
atari-us
az
azerty
ba
ba-alternatequotes
ba-unicode
ba-unicodeus
ba-us
backspace
```

```
bashkir
be
be-iso-alternate
be-latin1
be-nodeadkeys
be-oss
be-oss_latin9
be-wang
bg-cp1251
bg-cp855
bg_bds-cp1251
bg_bds-utf8
bg_pho-cp1251
bg_pho-utf8
bone
br
br-abnt
br-abnt2
br-dvorak
br-latin1-abnt2
br-latin1-us
br-nativo
br-nativo-epo
br-nativo-us
br-nodeadkeys
br-thinkpad
by
by-cp1251
by-latin
bywin-cp1251
ca
lines 1-58
```

Before you start, configure your keyboard accordingly. For example in the case of a **US** layout:

```
[root@redhat9 ~]# localectl set-keymap us

[root@redhat9 ~]# localectl status
System Locale: LANG=en_US.UTF-8
    VC Keymap: us
   X11 Layout: us
    X11 Model: pc105+inet
  X11 Options: terminate:ctrl_alt_bksp
```

LAB #1 - Linux File Hierarchy System

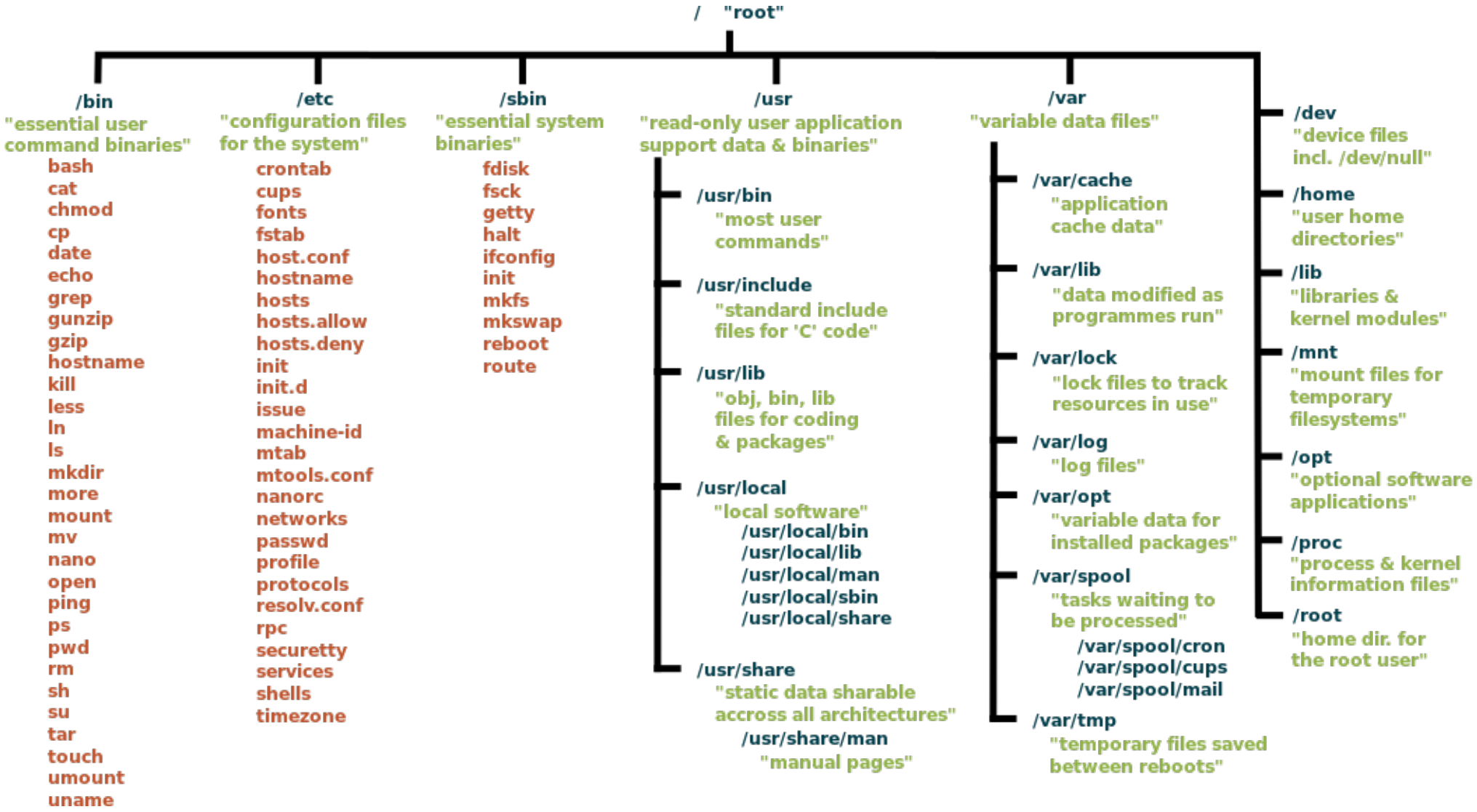
The Linux file system is organised around a single tree with a starting point called the **root**, represented by the character **/**. Below this root are directories containing files and sub-directories. The organisation of directories conforms to a standard called the **Linux File Hierarchy System**.

```
[trainee@redhat9 ~]$ cd /
[trainee@redhat9 /]$ ls -l
total 28
dr-xr-xr-x. 2 root root 6 Aug 10 2021 afs
lrwxrwxrwx. 1 root root 7 Aug 10 2021 bin -> usr/bin
dr-xr-xr-x. 5 root root 4096 Sep 25 12:30 boot
drwxr-xr-x. 20 root root 3320 Sep 25 12:44 dev
drwxr-xr-x. 133 root root 8192 Sep 25 12:44 etc
drwxr-xr-x. 3 root root 21 Oct 19 2023 home
lrwxrwxrwx. 1 root root 7 Aug 10 2021 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 Aug 10 2021 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 Aug 10 2021 media
drwxr-xr-x. 2 root root 6 Aug 10 2021 mnt
drwxr-xr-x. 2 root root 6 Aug 10 2021 opt
dr-xr-xr-x. 242 root root 0 Sep 25 12:44 proc
dr-xr-x---. 4 root root 4096 Oct 19 2023 root
drwxr-xr-x. 44 root root 1160 Sep 25 12:44 run
lrwxrwxrwx. 1 root root 8 Aug 10 2021 sbin -> usr/sbin
```

```
drwxr-xr-x. 2 root root 6 Aug 10 2021 srv
dr-xr-xr-x. 13 root root 0 Sep 25 12:44 sys
drwxrwxrwt. 15 root root 4096 Sep 25 12:48 tmp
drwxr-xr-x. 12 root root 144 Oct 19 2023 usr
drwxr-xr-x. 20 root root 4096 Oct 19 2023 var
```

/afs	Andrew File System (AFS) is a distributed file system that uses a set of trusted servers to present a consistent, location-transparent file namespace.
/bin	Contains user programs such as ls, cp e.t.c.. Note that under RHEL 9, this is a soft link (shortcut) to **/usr/bin**.
/boot	Contains bootloader files, kernels and initrd (INItial Ram Disk) files.
/dev	Contains device nodes for accessing all the peripherals and devices connected to the system. The *udev* binary takes care of dynamically creating and deleting the relevant device nodes automatically.
/etc	Contains static configuration files.
/home	Contains a directory for each registered user of the system except for root.
/lib	Contains shared 32 bit libraries for applications and modules. Note that under RHEL 9, this is a soft link (shortcut) to **/usr/lib**.
/lib64	Contains shared 64 bit libraries for applications and modules. Note that under RHEL 9, this is a soft link (shortcut) 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.
/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 9, /var/run is a soft link (shorcut) to **/run**.
/sbin	Contains essential system administration binaries. Note that under 9, this is a soft link (shortcut) 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.

Graphically, this can be represented as follows (“Standard-unix-filesystem-hierarchy.svg” by Ppgardne, Wikimedia Commons is licensed under CC BY-SA 4.0):



1.1 - 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.

1.2 - 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@redhat9 /]$ su -
Password: fenestros
[root@redhat9 ~]# 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=4096k,nr_inodes=976019,mode=755,inode64)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
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,size=1573912k,nr_inodes=819200,mode=755,inode64)
```

```
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)
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)
/dev/mapper/rhel-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=12983)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
none on /run/credentials/systemd-tmpfiles-setup-dev.service type ramfs
(ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
none on /run/credentials/systemd-sysctl.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
/dev/sda1 on /boot type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
none on /run/credentials/systemd-tmpfiles-setup.service type ramfs
(ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
tmpfs on /run/user/42 type tmpfs
(rw,nosuid,nodev,relatime,seclabel,size=786956k,nr_inodes=196739,mode=700,uid=42,gid=42,inode64)
tmpfs on /run/user/1000 type tmpfs
(rw,nosuid,nodev,relatime,seclabel,size=786956k,nr_inodes=196739,mode=700,uid=1000,gid=1000,inode64)
```

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

```
[root@redhat9 ~]# cat /etc/mtab
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
sysfs /sys sysfs rw,seclabel,nosuid,nodev,noexec,relatime 0 0
devtmpfs /dev devtmpfs rw,seclabel,nosuid,size=4096k,nr_inodes=976019,mode=755,inode64 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,seclabel,nosuid,nodev,inode64 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,size=1573912k,nr_inodes=819200,mode=755,inode64 0 0
```

```
cgroup2 /sys/fs/cgroup cgroup2 rw,seclabel,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot 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
/dev/mapper/rhel-root / xfs rw,seclabel,relatime,attr2,inode64,logbufs=8,logbsize=32k,noquota 0 0
selinuxfs /sys/fs/selinux selinuxfs rw,nosuid,noexec,relatime 0 0
systemd-1 /proc/sys/fs/binfmt_misc autofs
rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=12983 0 0
tracefs /sys/kernel/tracing tracefs rw,seclabel,nosuid,nodev,noexec,relatime 0 0
debugfs /sys/kernel/debug debugfs rw,seclabel,nosuid,nodev,noexec,relatime 0 0
mqueue /dev/mqueue mqueue rw,seclabel,nosuid,nodev,noexec,relatime 0 0
hugetlbfs /dev/hugepages hugetlbfs rw,seclabel,relatime,pagesize=2M 0 0
none /run/credentials/systemd-tmpfiles-setup-dev.service ramfs ro,seclabel,nosuid,nodev,noexec,relatime,mode=700
0 0
fusectl /sys/fs/fuse/connections fusectl rw,nosuid,nodev,noexec,relatime 0 0
configfs /sys/kernel/config configfs rw,nosuid,nodev,noexec,relatime 0 0
none /run/credentials/systemd-sysctl.service ramfs ro,seclabel,nosuid,nodev,noexec,relatime,mode=700 0 0
/dev/sda1 /boot xfs rw,seclabel,relatime,attr2,inode64,logbufs=8,logbsize=32k,noquota 0 0
none /run/credentials/systemd-tmpfiles-setup.service ramfs ro,seclabel,nosuid,nodev,noexec,relatime,mode=700 0 0
tmpfs /run/user/42 tmpfs
rw,seclabel,nosuid,nodev,relatime,size=786956k,nr_inodes=196739,mode=700,uid=42,gid=42,inode64 0 0
tmpfs /run/user/1000 tmpfs
rw,seclabel,nosuid,nodev,relatime,size=786956k,nr_inodes=196739,mode=700,uid=1000,gid=1000,inode64 0 0
```

Important: Note that the file system for /dev/sda1 and /dev/mapper/rhel-root is **xfs**.

1.3 - The umount command

To unmount a file system, use the umount command:

```
# umount /mnt/<target_directory>
```

or

```
# umount /dev/cdrom
```

1.4 - The /etc/fstab file

If the **mount** command is used with the **-a** option, all the file systems listed in the file called **/etc/fstab** will be mounted at the same time:

```
[root@redhat9 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu Oct 19 16:05:58 2023
#
# 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/rhel-root / xfs defaults 0 0
UUID=6f6c5bb9-30be-4734-bc23-03fed8541616 /boot xfs defaults 0 0
/dev/mapper/rhel-swap none swap defaults 0 0
```

Each line in this file contains 6 fields:

Field 1	**Field 2**	**Field 3**	**Field 4**	**Field 5**	**Field 6**
Special block file or UUID or virtual file system or a label	Mount point	File system type	Comma separated options	Used by *dump* (1 = to dump, 0 or empty = to ignore)	The order in which *fsck* checks file systems at boot time

The **UUID** (*Universally Unique Identifier*) is a 128-bit string. UUIDs are created automatically and at random when the filesystem is created on the partition. They can be modified by the administrator.

Mount Options

The most important mount options are :

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

Option	Filesystem	Description	Default Value
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

LAB #2 - Unix File System

Each file system contains the following :

- Superblock
- Inodes
- Data blocks
- Indirect blocks

2.1 - Superblock

The superblock contains:

- the size of the blocks
- the size of the file system
- the number of mounts performed for this file system
- a pointer to the root of the file system
- pointers to the list of free inodes
- pointers to the list of free data blocks

2.2 - 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.

To understand this better, type the following command:

```
[root@redhat9 ~]# ls -ld /dev/console /dev/sda1 /etc /etc/passwd
crw--w----. 1 root tty 5, 1 Sep 25 12:44 /dev/console
brw-rw----. 1 root disk 8, 1 Sep 25 12:44 /dev/sda1
drwxr-xr-x. 133 root root 8192 Sep 25 12:44 /etc
-rw-r--r--. 1 root root 2109 Oct 19 2023 /etc/passwd
```

The first character of each line can be one of the following:

- - 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 view the inode number, use the **-i** option:

```
[root@redhat9 ~]# ls -ldi /dev/console /dev/sda1 /etc /etc/passwd
12 crw--w----. 1 root tty 5, 1 Sep 25 12:44 /dev/console
273 brw-rw----. 1 root disk 8, 1 Sep 25 12:44 /dev/sda1
67154049 drwxr-xr-x. 133 root root 8192 Sep 25 12:44 /etc
```

```
68914044 -rw-r--r--. 1 root root 2109 Oct 19 2023 /etc/passwd
```

2.3 - Data blocks

Data is stored in data blocks. In the case of a directory, the data block contains a table which references the inodes and file names in the directory. This table is called a **catalog table**.

The name of a file is not stored in the inode but in a **catalog table**. This feature allows us to give two different names to the same file. To add a new name to a file, a **physical link** must be created.

2.4 - Physical (hard) links

A physical or hard link is created using the following command:

- ln file_name additional_name

To illustrate this point, type the following command line:

```
[root@redhat9 ~]# cd /tmp; mkdir inode; cd inode; touch file1; ls -ali
total 4
33625083 drwxr-xr-x. 2 root root 22 Sep 25 13:08 .
33554561 drwxrwxrwt. 16 root root 4096 Sep 25 13:08 .
33625154 -rw-r--r--. 1 root root 0 Sep 25 13:08 file1
```

Note the inode number of the file **file1**. Also note that the number in the third field of the file1 line has the value **1** :

```
33625154 -rw-r--r--. 1 root root 0 Sep 25 13:08 file1
```

Now create the hard link and check the result:

```
[root@redhat9 inode]# ln file1 file2
[root@redhat9 inode]# ls -ali
```

```
total 4
33625083 drwxr-xr-x. 2 root root 38 Sep 25 13:09 .
33554561 drwxrwxrwt. 16 root root 4096 Sep 25 13:08 .
33625154 -rw-r--r--. 2 root root 0 Sep 25 13:08 file1
33625154 -rw-r--r--. 2 root root 0 Sep 25 13:08 file2
```

Note the following two lines:

33625154 -rw-r–r–. 2 root root 0 Sep 25 13:08 file1
33625154 -rw-r–r–. 2 root root 0 Sep 25 13:08 file2

Both files, file1 and file2, are referenced by the same inode. The number of links is therefore increased by 1 (the number in the third field).

Important: A physical link can only be created if the two files are in the same filesystem and the source file exists.

2.5 - Symbolic Links

A symbolic link is a **shortcut** to another file or directory. A symbolic link is created using the following command:

- ln -s filename shortcut_name

To illustrate this point, type the following command:

```
[root@redhat9 inode]# ln -s file1 file3
[root@redhat9 inode]# ls -ali
total 4
33625083 drwxr-xr-x. 2 root root 54 Sep 25 13:10 .
33554561 drwxrwxrwt. 16 root root 4096 Sep 25 13:08 .
33625154 -rw-r--r--. 2 root root 0 Sep 25 13:08 file1
```

```
33625154 -rw-r--r--. 2 root root 0 Sep 25 13:08 file2
33625156 lrwxrwxrwx. 1 root root 8 Sep 25 13:10 file3 -> file1
```

Note that the symbolic link is referenced by another inode. The symbolic link points to file1.

Important: A symbolic link can be created even if the two files are in two different file systems and even if the source file does not exist.

---

1)

Not Applicable