

ADVERTISEMENT



Auth0®

# Understanding UNIX / Linux File Systems

Author: Vivek Gite

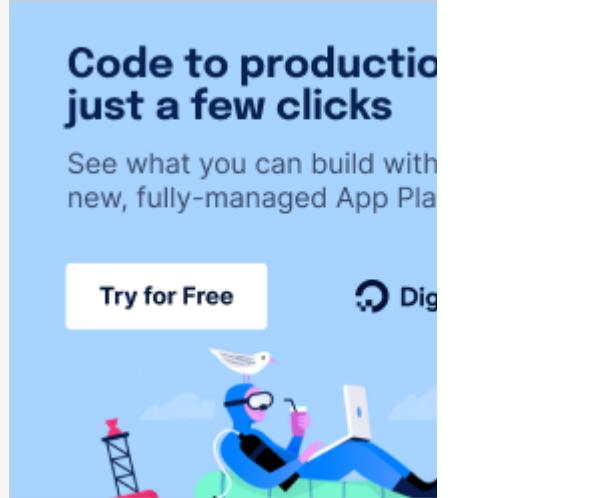
Last updated: December 23, 2018

[50 comments](#)

A conceptual understanding of the file system, especially data structure and related terms will help you become a successful system administrator. I have seen many new Linux system administrator without any clue about the file system. The conceptual knowledge can be applied to restore the file system in an emergency situation.



ADVERTISEMENT



## What is a file in Linux or Unix?

A file is a collection of data items stored on disk. Or, it is a device which can store the information, data, music (mp3 files), picture, movie, sound, PDF book and more. All data must be stored on your computer in the form of a file. Files are always associated with devices like hard disk, floppy disk, USB pen drive and more. A file is the last object in your file system tree. See [Linux/UNIX – rules for naming file and directory names](#).

## How to list directory contents

Use the ls command:

```
ls  
ls -l  
ls -Fl  
ls -l /etc/
```

```
vivek@nixcraft-asus:~/bin$ ls -l /etc/
total 1564
drwxr-xr-x  3 root root      4096 Apr  4  2018 acpi
-rw-r--r--  1 root root      3028 Apr  4  2018 adduser.conf
-rw-r--r--  1 root root       51 Jul  5 23:13 aliases
drwxr-xr-x  2 root root    12288 Dec 14 22:18 alternatives
-rw-r--r--  1 root root      401 May 29  2017 anacrontab
drwxr-xr-x  3 root root      4096 Dec 15 14:56 ansible
-rw-r--r--  1 root root      234 Jun 29 16:23 antidote.conf
drwxr-xr-x  3 root root      4096 Aug  6  00:23 apache2
-rw-r--r--  1 root root      433 Oct  2  2017 apg.conf
drwxr-xr-x  6 root root      4096 Apr  4  2018 apt
drwxr-xr-x  3 root root      4096 Oct  6 10:51 apparmor
drwxr-xr-x  9 root root      4096 Dec 15 14:56 apparmor.d
drwxr-xr-x  5 root root      4096 Nov 14 18:27 apport
-rw-r--r--  1 root root      769 Jan 23  2018 appstream.conf
drwxr-xr-x  6 root root      4096 Aug 12 15:23 apt
-rw-r-----  1 root daemon   144 Feb 20  2018 at.deny
drwxr-xr-x  3 root root      4096 Sep  6 14:44 avahi
-rw-r--r--  1 root root     2319 Feb  6  2018 bash.bashrc
-rw-r--r--  1 root root      45 Apr  2  2018 bash_completion
drwxr-xr-x  2 root root      4096 Nov 28 13:54 bash_completion.d
-rw-r--r--  1 root root      367 Jan 27  2016 bindresvport.blacklist
drwxr-xr-x  2 root root      4096 Mar 27  2018 binfmt.d
drwxr-xr-x  2 root root      4096 Jul 12 21:05 bluetooth
-rw-r--r--  1 root root     2039 Dec 13  2016 bmon.conf
drwxr-xr-x  2 root root      4096 Jun 27 23:21 bonobo-activation
-rw-r-----  1 root root      33 Apr  4  2018 brlapi.key
drwxr-xr-x  7 root root      4096 Apr  4  2018 brltty
-rw-r--r--  1 root root    25341 Jan 19  2018 brltty.conf
drwxr-xr-x  3 root root      4096 Apr  4  2018 ca-certificates
-rw-r--r--  1 root root     6864 Apr 15  2018 ca-certificates.conf
-rw-r--r--  1 root root     6488 Apr  4  2018 ca-certificates.conf.dpkg-old
drwxr-xr-x  2 root root      4096 Apr  4  2018 calendar
-rw-r--r--  1 root root      119 Sep 14  2017 catdocrc
drwxr-s---  2 root dip      4096 Nov  9 18:11 chatscripts
drwxr-xr-x  4 root root      4096 Dec 12 16:16 chromium-browser
-rw-r--r--  1 root root      963 Jun  3  2017 colordiffrc
-rw-r--r--  1 root root    55019 Feb  9  2017 complete.tcsh
drwxr-xr-x  2 root root      4096 Sep  6 14:43 console-setup
-rw-r--r--  1 root root     6543 Oct  1  2017 cowpoke.conf
drwxr-xr-x  2 root root      4096 Apr  4  2018 cracklib
drwxr-xr-x  2 root root      4096 Oct 25 12:19 cron.d
drwxr-xr-x  2 root root      4096 Dec 15 14:56 cron.daily
drwxr-xr-x  2 root root      4096 Apr  4  2018 cron.hourly
drwxr-xr-x  2 root root      4096 Apr  4  2018 cron.monthly
-rw-r--r--  1 root root      722 Nov 16  2017 crontab
drwxr-xr-x  2 root root      4096 Nov 28 13:54 cron.weekly
```

© www.cyberciti.biz

Using ls command to list information about the files on Linux and Unix-like systems

The ls -l command gives full information and indicates the type of filesystem object stored on disk. For example:

drwxr-xr-x	3	root	root	4096	Apr	4	2018	acpi
-rw-r--r--	1	root	root	3028	Apr	4	2018	adduser.conf

The following information is displayed for each file from above outputs:

Field	Description
drwxr-xr-	File mode
xr-x	
3	Number of links to file
root	File owner name
root	File group name
4096	number of bytes in the file
Apr 4 2018	Abbreviated month, day-of-month file was last modified, hour file last modified, and minute file last modified
acpi	The pathname/filename

The acpi is a directory indicated by first character **d** in **drwxr-xr-x** and the adduser.conf is a file indicated by first character **-** in **-rw-r--r--**. Let us try to understand meaning of **drwxr-xr-x** in ls command output.

## More on the file mode

To understand the **drwxr-xr-x** file mode let us divide into three groups:

1. group 1 : d
2. group 2 : **rwx**
3. group 3 : r-x
4. group 4 : r-x

The file mode (group 1) printed consists of the file type and the permissions. The entry type character (group 1) describes the type of file, as follows:

Character	File type
-	Regular file.
b	Block special file.
c	Character special file.
d	Directory.

l	Symbolic link.
p	FIFO.
s	Socket.
w	Whiteout.

The next three groups are for owner permissions (group 2), group permissions (group 3), and other permissions (group 4). So each field has three character positions:

## Patreon supporters only guides 😎

- No ads and tracking
- In-depth guides for developers and sysadmins at [Opensourceflare](#) ✨
- Join my Patreon to support independent content creators and start reading latest guides:
  - ↪ [How to set up Redis sentinel cluster on Ubuntu or Debian Linux](#)
  - ↪ [How To Set Up SSH Keys With YubiKey as two-factor authentication \(U2F/FIDO2\)](#)
  - ↪ [How to set up Mariadb Galera cluster on Ubuntu or Debian Linux](#)

[Join Patreon →](#)

1. r : Read only file permission
2. w : Write only file permission
3. x : Execute only file permission
4. – : No permission

So group 2 has **rwx** permission it means you have read, write and executable permission on the file.

## What is a directory?

A directory is a group of files. A directory is divided into two types:

- Root directory – Strictly speaking, there is only one root directory in your Linux and Unix-like system, which is denoted by / (forward slash). It is root of your entire file system and can not be renamed or deleted.
- Sub directory – Directory under root (/) directory is subdirectory which can be created, renamed by the user.

Directories are used to organize your data files, programs more efficiently.

## How to create a directory

Use the [mkdir command](#):

```
mkdir dir1
```

Next list newly created directory with the help of ls command:

```
ls -ld dir1
```

To change the working directory use the cd command:

```
cd dir1
```

To print the current working directory, run the [pwd command](#):

```
pwd
```

Let us create two sub-directories and a file, run:

```
mkdir foo  
mkdir -v bar
```

Next [create a file named demo.txt in Linux/Unix from a bash shell prompt](#), run:

```
echo "This is a test" > demo.txt
```

List everything:

```
ls -l
```

```
vivek@nixcraft-asus:/tmp$ pwd  
/tmp  
vivek@nixcraft-asus:/tmp$ mkdir dir1  
vivek@nixcraft-asus:/tmp$  
vivek@nixcraft-asus:/tmp$ ls -ld dir1  
drwxr-xr-x 2 vivek vivek 4096 Dec 23 16:57 dir1  
vivek@nixcraft-asus:/tmp$ cd dir1  
vivek@nixcraft-asus:/tmp/dir1$  
vivek@nixcraft-asus:/tmp/dir1$ pwd  
/tmp/dir1  
vivek@nixcraft-asus:/tmp/dir1$ mkdir foo  
vivek@nixcraft-asus:/tmp/dir1$ mkdir -v bar  
mkdir: created directory 'bar'  
vivek@nixcraft-asus:/tmp/dir1$ echo "This is a test" > demo.txt  
vivek@nixcraft-asus:/tmp/dir1$ ls -l  
total 12  
drwxr-xr-x 2 vivek vivek 4096 Dec 23 16:57 bar  
-rw-r--r-- 1 vivek vivek 15 Dec 23 16:57 demo.txt  
drwxr-xr-x 2 vivek vivek 4096 Dec 23 16:57 foo  
vivek@nixcraft-asus:/tmp/dir1$ pwd  
/tmp/dir1  
vivek@nixcraft-asus:/tmp/dir1$ █
```

© www.cyberciti.biz

Creating a new directory, sub-directories, and file on Linux

# Linux supports numerous file system types

- Ext2: This is like UNIX file system. It has the concepts of blocks, inodes and directories.
- Ext3: It is ext2 filesystem enhanced with journalling capabilities. Journalling allows fast file system recovery. Supports POSIX ACL (Access Control Lists).
- Isofs (iso9660): Used by CDROM file system.
- Sysfs: It is a ram-based filesystem initially based on ramfs. It is used to exporting kernel objects so that end user can use it easily.
- Procfs: The proc file system acts as an interface to internal data structures in the kernel. It can be used to obtain information about the system and to change certain kernel parameters at runtime using the sysctl command. For example, [you can find out CPU information on Linux with following cat command:](#)

```
$ cat /proc/cpuinfo
```

- Or you can enable or disable routing/forwarding of IP packets between interfaces with following command:

```
# cat /proc/sys/net/ipv4/ip_forward
# echo "1" > /proc/sys/net/ipv4/ip_forward
# echo "0" > /proc/sys/net/ipv4/ip_forward
```

- NFS: Network file system allows many users or systems to share the same files by using a client/server methodology. NFS allows sharing all of the above file system.
- Linux also supports Microsoft NTFS, vfat, and many other file systems. See Linux kernel source tree Documentation/filesystem directory for list of all supported filesystem.

You can find out what type of file systems currently mounted with mount command.

```
$ mount
```

OR

```
$ cat /proc/mounts
```

```
vivek@nixcraft-asus:~$ ssh vivek@192.168.2.201 "mount"
/dev/sd0a on / type ffs (local)
/dev/sd0k on /home type ffs (local, nodev, nosuid)
/dev/sd0d on /tmp type ffs (local, nodev, nosuid)
/dev/sd0f on /usr type ffs (local, nodev)
/dev/sd0g on /usr/X11R6 type ffs (local, nodev)
/dev/sd0h on /usr/local type ffs (local, nodev, wxallowed)
/dev/sd0j on /usr/obj type ffs (local, nodev, nosuid)
/dev/sd0i on /usr/src type ffs (local, nodev, nosuid)
/dev/sd0e on /var type ffs (local, nodev, nosuid)
vivek@nixcraft-asus:~$ ssh vivek@192.168.2.204 "cat /proc/mounts"
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=2005232k,nr_inodes=501308,mode=755 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,seclabel,nosuid,nodev,size=2019080k,nr_inodes=504770 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=2019080k,nr_inodes=504770,mode=755 0 0
tmpfs /sys/fs/cgroup tmpfs ro,seclabel,nosuid,nodev,noexec,size=2019080k,nr_inodes=504770,mode=755 0 0
cgroup2 /sys/fs/cgroup/unified cgroup2 rw,seclabel,nosuid,nodev,noexec,relatime,nsdelegate 0 0
cgroup /sys/fs/cgroup/systemd cgroup rw,seclabel,nosuid,nodev,noexec,relatime,xattr,name=systemd
pstree /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/memory cgroup rw,seclabel,nosuid,nodev,noexec,relatime,memory 0 0
cgroup /sys/fs/cgroup/freezer cgroup rw,seclabel,nosuid,nodev,noexec,relatime,freezer 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/cpu,cpuacct cgroup rw,seclabel,nosuid,nodev,noexec,relatime,cpu,cpuacct 0 0
cgroup /sys/fs/cgroup/pids cgroup rw,seclabel,nosuid,nodev,noexec,relatime,pids 0 0
cgroup /sys/fs/cgroup/blkio cgroup rw,seclabel,nosuid,nodev,noexec,relatime,blkio 0 0
cgroup /sys/fs/cgroup/perf_event cgroup rw,seclabel,nosuid,nodev,noexec,relatime,perf_event 0 0
cgroup /sys/fs/cgroup/hugetlb cgroup rw,seclabel,nosuid,nodev,noexec,relatime,hugetlb 0 0
cgroup /sys/fs/cgroup/devices cgroup rw,seclabel,nosuid,nodev,noexec,relatime,devices 0 0
cgroup /sys/fs/cgroup/cpuset cgroup rw,seclabel,nosuid,nodev,noexec,relatime,cpuset 0 0
configfs /sys/kernel/config configfs rw,relatime 0 0
/dev/mapper/fedora_fedora28--nixcraft-root / xfs rw,seclabel,relatime,attr2,inode64,noquota 0 0
selinuxfs /sys/fs/selinux selinuxfs rw,relatime 0 0
systemd-1 /proc/sys/fs/binfmt_misc autofs rw,relatime,fd=38,pgrp=1,timeout=0,minproto=5,maxproto=2,realtime=0
hugetlbfs /dev/hugepages hugetlbfs rw,seclabel,relatime,pagesize=2M 0 0
debugfs /sys/kernel/debug debugfs rw,seclabel,relatime 0 0
mqqueue /dev/mqueue mqqueue rw,seclabel,relatime 0 0
tmpfs /tmp tmpfs rw,seclabel,nosuid,nodev,size=1827008k,nr_inodes=456752 0 0
/dev/vda1 /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=194100k,mode=700,uid=1000,gid=1000 0 0
vivek@nixcraft-asus:~$
```

© www.cyberciti.biz

mount running on OpenBSD Unix box and display /proc/mounts on Linux

## What is a UNIX/Linux File system?

A UNIX file system is a collection of files and directories stored on disk. Each file system is stored in a separate whole disk partition. The following are a few of the file system:

- `/` – Special file system that incorporates the files under several directories including `/dev`, `/sbin`, `/tmp` and more
- `/usr` – Stores application programs
- `/var` – Stores log files, mails and other data
- `/tmp` – Stores temporary files

See [The importance of Linux partitions](#) for more information.

## But what is in a File system?

Again file system divided into two categories:

- User data – stores actual data contained in files
- Metadata – stores file system structural information such as superblock, inodes, directories

Next time I will write more about Metadata objects – superblock, inodes, directories with actual linux commands so that you can understand and master the concepts 😊

Continue reading rest of the Understanding Linux file system series:

This entry is 1 of 9 in the **Conceptual overview of the Linux or Unix file system Tutorial** series. Keep reading the rest of the series:

1. Understanding UNIX / Linux File System
2. [Understanding UNIX / Linux filesystem directories](#)
3. [Understanding the Linux file system directories / hierarchy](#)

4. [Understanding UNIX / Linux filesystem Superblock](#)
5. [Understanding filesystem Inodes](#)
6. [What is a hard and symbolic \(soft\) link in Linux or Unix?](#)
7. [Why isn't it possible to create hard links across file system boundaries?](#)
8. [Surviving a Linux Filesystem Failures](#)
9. [Linux / UNIX Delete or Remove Files With Inode Number](#)

💡 Get the latest tutorials on Linux, Open Source & DevOps via

RSS feed →

Weekly email newsletter →

ADVERTISEMENT



## The #1 Authentication Platform

Secure Authentication-as-a-Service Built for Developers, Trusted by Enterprises

🔍 To search, type & hit enter...

### Related Posts



[Exporting Networked File Systems Via Another...](#)