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.

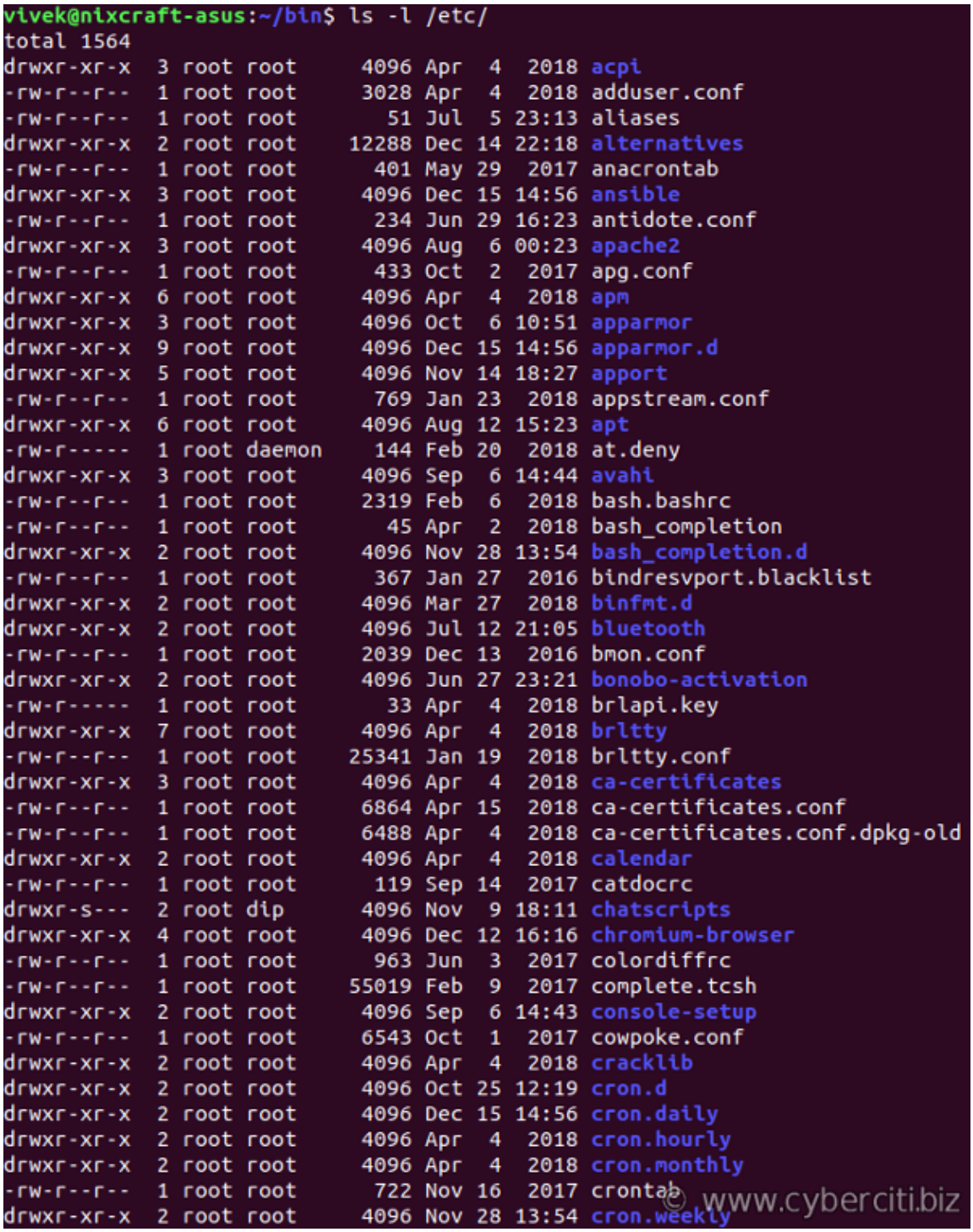
What is a file in Linux or Unix?

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/
```



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-x`	File mode
`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:

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:

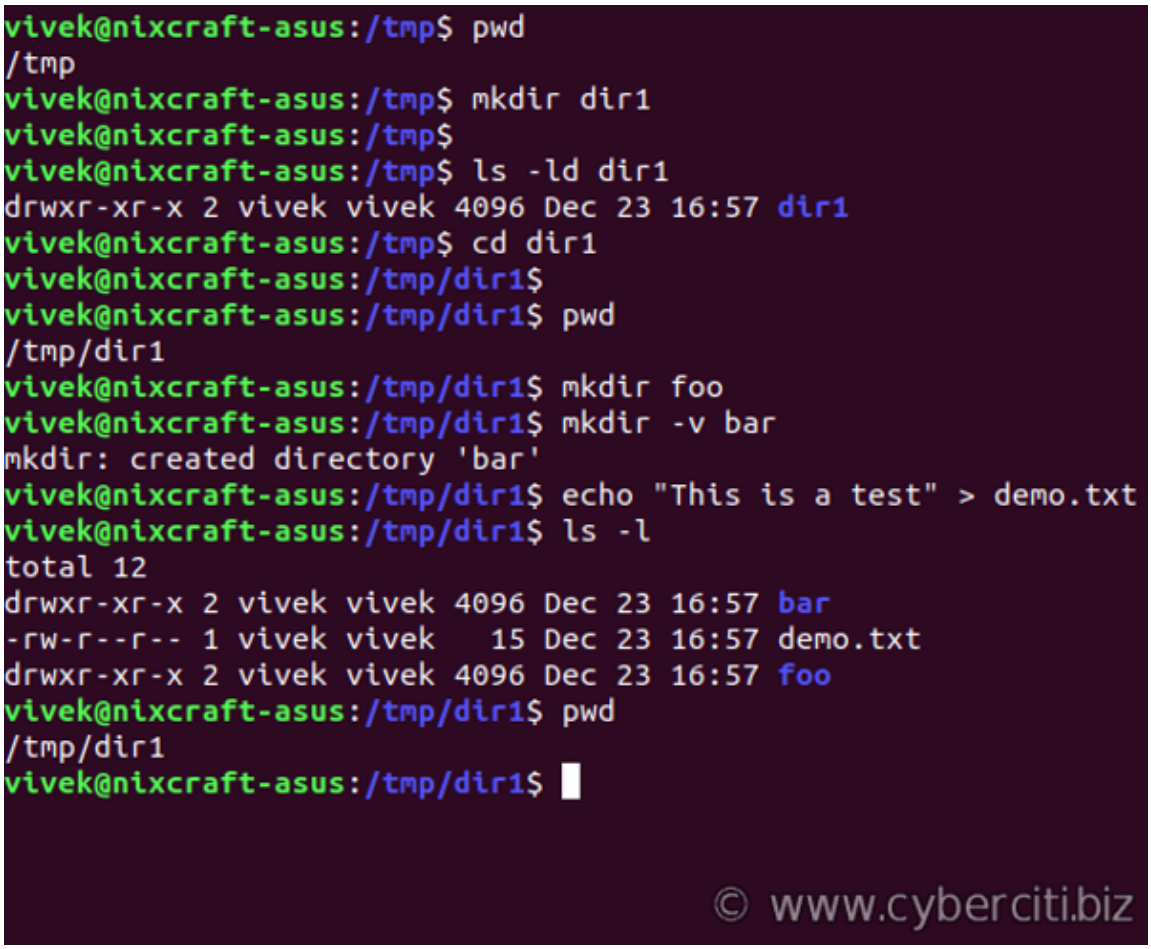
```
mkdir foo
mkdir –v bar
```

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
```



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 use 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
```

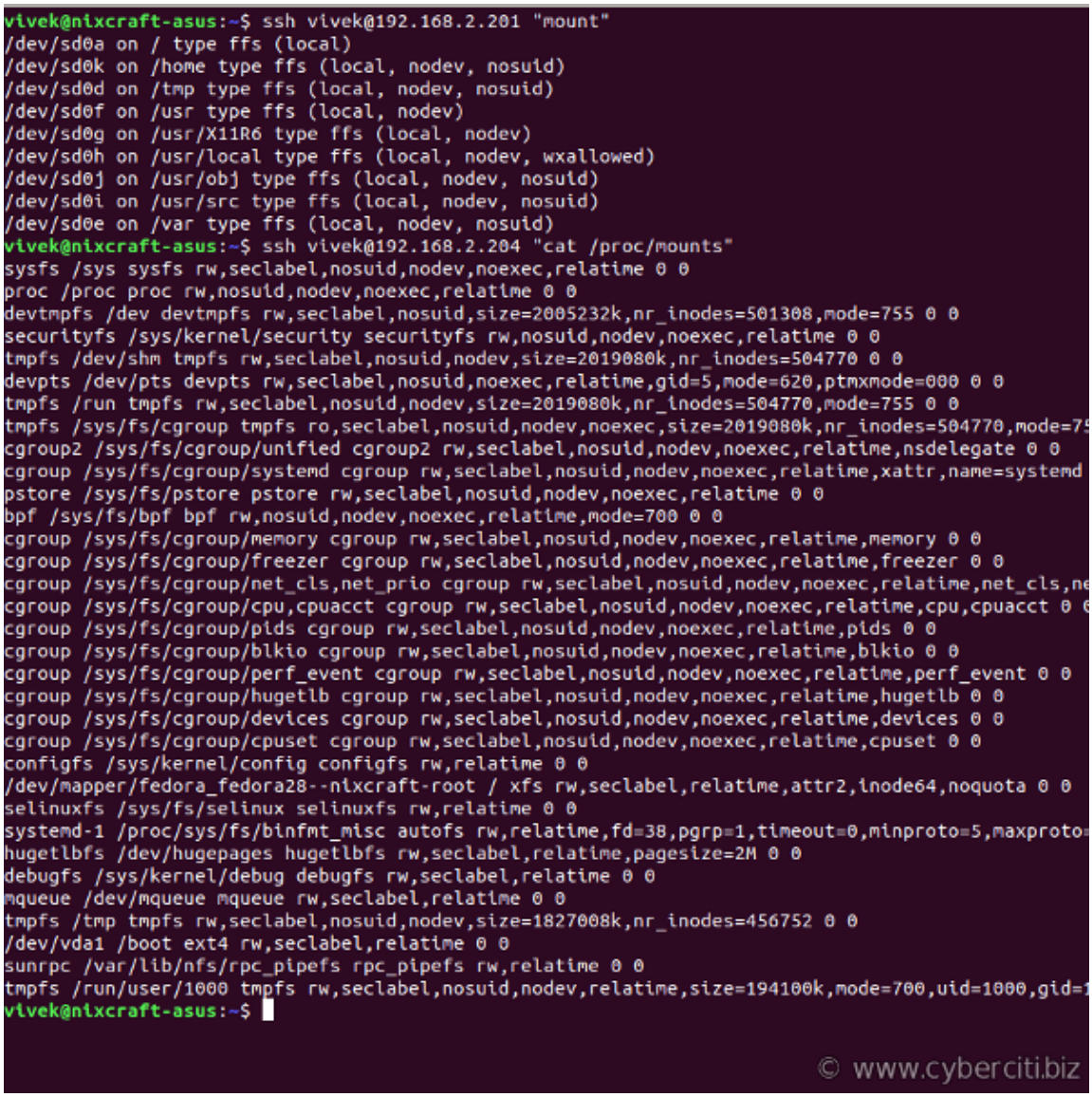
- NFS: Network file system allows many users or systems to share the same files by using a client/server methodology. NFS allows sharing all of the above file system.
- Linux also supports Microsoft NTFS, vfat, and many other file systems. See Linux kernel source tree Documentation/filesystem directory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



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

Related Posts

Exporting Networked File Systems Via Another…