

Basic file system in Unix / Linux

A file system is a logical collection of files on a partition or a disk. A partition is an information store and can be combined into a hard disk if desired.

A file system is a logical collection of files on a partition or a disk. A partition is an information store and can be combined into a hard disk if desired.

Your hard disk may have a variety of partitions that usually contain only one file system, such as a / home file system.

A file system of a partitioning area allows to maintain and manage various file systems logically.

Everything in Unix is considered as a file, including physical devices like DVD-ROMs, USB, floppy disks, .

Directory structure in Unix / Linux

Unix uses a hierarchical file system structure, which resembles a top-down tree structure, with root (/) at the base of the file system and all other directories spread from there.

A file system is a collection of files and directories that have the following properties:

It has a root directory (/) that contains other files and directories.

Each file and directory is uniquely identified by its name, the directory in which it resides, and a unique identifier, which is called inode specifically.

By convention, the root directory has an inode number of 2 and the lost + found directory has an inode number of 3. Inode numbers 0 and 1 are not used. Inode numbers can be sent by the specified in -i function of the ls command.

It has another characteristic that is self-contained. There is no dependency between one file system and another.

Folders have their own purposes and often keep the same types of information for easy file placement. Here are the directories that exist on large versions of Unix.

Folder Description / It is the main directory that contains only the highest-level directories needed in the file structure. / **bin** This location puts executable files. They are available to all users. / **dev** These are control devices. / **etc** Supervisors directory commands, configuration files, disk configuration files, valid user lists, ethernet, hosts, where to send serious messages. / **lib** Contains shared library files and sometimes kernel-related files. / **boot** Contains files to boot the system. / **home** Contains the main directory for users and other accounts. / **mnt** Used to mount temporary filesystems, such as cdrooms and floppy disks. / **proc** Contains all processes marked as a file by the number of processes or other information that are the driving force of the system. / **tmp**

Keep temporary files used between system boot. / **usr** Used for mixed purposes, or can be used by many users. Includes administrative commands, shared files, library files, and others. / **var** A specific directory, containing long variable files such as log and print files and any other type of file that may contain a variable number of data. / **sbin** Contains binary (executable) files, usually for system management. Examples of utilities include *fdisk* and *ifconfig*. / **kernel** Contains kernel files.

Navigate the file system in Unix / Linux

Now that you understand the basics of the file system, you can start navigating the files you need. The following table is the commands you will use to navigate the system.

Command Description **cat filename** Displays a file name. **cd dirname** Move you to the specified directory. **cp file1 file2** Copy a file / folder to the specified location. **file filename** Identify file type (binary, text, .). **find filename dir** Find a file / directory. **head filename** Indicates the start of a file. **less filename** Browse through a file from the end or from the beginning. **ls dirname** Only the contents of the specified directory. **mkdir dirname** Create a predefined directory. **more filename** Browse through a file from start to finish. **mv file1 file2** Move the location or rename a file / folder. **pwd** Only the current directory that the user is currently in. **rm filename** Remove a file. **rmdir dirname** Remove a directory. **tail filename** Only the end of a file. **touch filename** Create an empty file or modify the properties of an existing file.s. **whereis filename** Indicates the location of a file. **which filename** Indicates the location of a file if it is in your PATH.

Df command in Unix / Linux

The first way to manage the partition space is with the **df** command (disk free abbreviation). The **df -k** command displays the disk space used in kilobytes, as shown below:

```
$ df -k Filesystem 1K - blocks Used Available Use % Mounted on / dev / vzfs 1
```

Some directories, like / **devices** , only 0 kilobytes are used. These files are special file systems, and although they reside on disks under slashes, they themselves do not occupy disk space.

The output of **df -k** is usually the same on all Unix systems. The following lists what it includes:

Column Description **Filesystem** File system name. **kbytes** Total number of kilobytes available on storage **used** Total the total number of kilobytes used (by file). **avail** The total number of kilobytes available for use (remaining unused). **capacity** Percent of the total amount of space used by the file. **Mounted on** What file system is mounted on.

You can use the **-h** function (human readable) to display the output in a format that is only the size of the comment in the easiest way to understand.

Du command in Unix / Linux

Command **du** (short for disk usage) gives you the ability to identify folders to indicate the use of disk space on a specific directory.

This command is useful if you want to specify the space used for a specific directory. The following command displays the number of blocks that each folder occupies. A single block can be 512 bytes or 1 kilo byte depending on your system.

```
$ du / etc 10 / etc / cron . d 126 / etc / default 6 / etc / dfs . $
```

The **-h** function makes it easier for you to understand and explain.

```
$ du - h / etc 5k / etc / cron . d 63k / etc / default 3k / etc / dfs . $
```

Mounting (Mounting) file system in Unix / Linux

A file system must be mounted in order to be used by the system. To observe what is currently mounted (available for use) on your system, use the following command:

```
$ mount / dev / vzfs on / type reiserfs ( rw , usrquota , grpquota ) proc on /
```

The **/mnt** directory, by convention, is where temporary bindings (such as CD-ROMs, floppy disks, etc.) are placed there. If you need to mount a file system, you can use the **mount** command with the following syntax:

```
mount - t file_system_type device_to_mount directory_to_mount_to
```

For example, if you want to mount a CD-ROM to the **/mnt/cdrom** directory, you can enter from the keyboard as follows:

```
$ mount - t iso9660 / dev / cdrom / mnt / cdrom
```

This assumes that your CD-ROM device is called **/dev/cdrom** and you want to connect it to **/mnt/cdrom**. You refer to the **Man Page** for more specific information or type **mount -h** from the command line for help information.

After mounting, you can use the **cd** command to navigate the latest file system through the connection point you just created.

Unmount (unmounting) the file system in Unix / Linux

To unmount the file system from your system, use the **umount** command by specifying an access point or device.

For example, to remove **cdrom**, use the following command:

```
$ umount / dev / cdrom
```

The **mount** command gives you the ability to access the file system, but on the most modern Unix systems, the auto-connect **function performs** this **process implicitly** for the user and does not require any intervention.

User and group quotas (quotas) in Unix / Linux

User and group quotas provide techniques by which the amount of space used by a single user or all users in a particular group may be limited to be defined by the administrator.

Quotas operate through two restrictions that allow users to perform certain actions in the amount of space or the number of blocks of the disk that the manager has defined the limit for.

Soft Limit : If the user has exceeded the limit, requires more space to use, a few more blocks will be provided.

Hard Limit : When a user uses a limit, requiring additional space to be used, no blocks will be added.

Here are the commands that are used to manage quotas:

Command Description quota Displays the disk usage and limits for a user in a group. edquota This is a quota editor. User or group quota can be edited using this command. quotacheck Scan file system for disk usage, create, check and repair quota files. setquota It is also a command in the quota editor. quotaon This statement to the system that disk quotas should be enabled on one or more file systems. quotaoff This declares to the system that disk quotas should be disabled on one or more file systems. repquota Print the disk usage and quota for the specified file system.

According to Tutorialspoint

Previous lesson: Regular Expression in Unix / Linux

Next lesson: User management in Unix / Linux

You finished reading the article "**Basic file system in Unix / Linux**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.