Access Linux files systems

Linux #redhat #filesystem #files

- Filesystem is a structured way where all files and directories are stored
  - What are the different types of Linux filesystems?
    - There are ext2, ext3, ext4, XFS...
- To access those files, we need navigation tools
- Following are the basic tools or commands to access Linux file system
  - `ls`
  - `cd`
  - `pwd`
  - `df` - List the filesystem on disk
  - `du` - Disk utilization by files
  - `fdisk` - Command used to create a file system, or create partitions.
  - Absolute and relative path (absolute path always begins with `/`)
  - Tilde `~`
  - `.` and `..`

The `ls`, `cd`, `pwd` commands

- Exemplified in Manage Files from the command line
- Remember:
  - `ls -ltr`

The `df` and `du` commands

- Exemplified in Monitor and manage Linux processes
- Remember:
  - `df -h` - list filesystems in human readable units
  - `du` - It will give you the size of each file in your system

The `fdisk` command

- You cannot run `fdisk` by itself (you have to specify an option)

Example using `fdisk` command:

```
[root@localhost ~]# fdisk -l
```

- The `-l` option stands for "list" and specifies to list the partition tables for the specified devices and then exit.

Output:

```
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc8a29178

Device     Boot   Start      End  Sectors Size Id Type
/dev/sda1  *       2048  2099199  2097152   1G 83 Linux
/dev/sda2       2099200 41943039 39843840  19G 8e Linux LVM


Disk /dev/mapper/rhel-root: 17 GiB, 18249416704 bytes, 35643392 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/mapper/rhel-swap: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

- It will list all your filesystem and where is it mounted.
- Why is there LVM on the 19G (`/dev/sda2`) drive? and why not on the 1G one (`/dev/sda1`)?
  - It's because sda1 has been mounted on `/boot`
    - You can verify this by running `df -h` and see where the `/dev/sda1` disk is mounted on.
    - For `/boot` or sometimes the swap you do not need an LVM filesystem.
- LVM allows you to create redundancy within your filesystem.

To create a new disk:

- Let's say you attach a new disk to your Linux OS, then you have to come here and run `fdisk` and specify the the name of the disk.
  Example using `fdisk` command:

```
[root@localhost ~]# fdisk /dev/sdb
```

- As the name `/dev/sda` is already taken (that is our principal disk), we can use the name `/dev/sdb`.
- When running this command it will give you more options on how you could partition that disk.

Absolute and relative path

- Absolute path is something that you could `cd` into it by specifying a complete path.
- For example look at the locations:

Example using `cd` command:

```
[root@localhost mmarin]# pwd
/home/mmarin
[root@localhost mmarin]# cd
```

```
[root@localhost ~]# pwd
/root
```

- Look at the first location
- if you do `cd` at any point, it will take you to the current user home directory, in this case it was root.

In this example we want to get into the `/home/mmarin/Desktop/` directory using relative path
Example using relative path:

```
[root@localhost ~]# cd /home
[root@localhost home]# pwd
/home
[root@localhost home]# ls
babubutt  ironman  mmarin  spiderman
[root@localhost home]# cd mmarin
[root@localhost mmarin]# cd Desktop
[root@localhost Desktop]# pwd
/home/mmarin/Desktop
```

- Note we use multiple `cd` commands to change between directories and subdirectories.

Same example using absolute path
Example using absolute path:

```
[root@localhost ~]# cd /home/mmarin/Desktop
[root@localhost Desktop]# pwd
/home/mmarin/Desktop
```

The `~` sign

- Go to the current user home directory.

Example using `~` sign:

```
[root@localhost Desktop]# cd ~
[root@localhost ~]# pwd
/root
```

- Note we are in the previous example location (`/home/mmarin/Desktop`)
- In this example we want to go directly to our user home directory (root user).
  - Remember the home directory of the root user is just `/root`
- If you do not type `~` and you just type `cd` it will still bring you to your home directory.
- This sign is more useful when you want to move files to your user's home directory.

The `.` and `..` signs

- `.` refers to the current working directory
- `..` refers to the previous directory from which the current working directory is a subdirectory of.

Example using `.` sign:

```
[root@localhost ~]# cd .
[root@localhost ~]# pwd
/root
```

- If you do `cd .` it will keep you in the same directory.

Example using `..` sing:

```
[root@linuxtest ~]# cd /home/mmarin/Desktop/
[root@linuxtest Desktop]# cd ..
[root@linuxtest mmarin]# pwd
/home/mmarin
[root@linuxtest mmarin]# cd ..
[root@linuxtest home]# pwd
/home
[root@linuxtest home]# cd ..
[root@linuxtest /]# pwd
/
[root@linuxtest /]# cd ..
[root@linuxtest /]#
```

- Coming back in a relative pathway.