

Identifying File types in Linux

Introduction

When navigating the Linux file system you are sure to encounter different file types. The most used and obvious file types are regular files and directories. However, the Linux operating system has more to offer in terms of file types as it also includes another 5 file types. Let's look at them one by one

Identifying Linux File types

There is only 1 command you need to know, which will help you to identify and categorize all the seven different file types found on the Linux system.

```
$ ls -ld <file name>
```

Here is an example output of the above command.

```
$ ls -ld /etc/services
-rw-r--r-- 1 root root 19281 Feb 14 2012 /etc/services
```

ls command will show the file type as an encoded symbol found as the first character of the file permission part. In this case it is "-", which means "regular file". It is important to point out that Linux file types are not to be mistaken with file extensions. Let us have a look at a short summary of all the seven different types of Linux file types and **ls** command identifiers:

1. **-** : regular file
2. **d** : directory
3. **c** : character device file
4. **b** : block device file
5. **s** : local socket file
6. **p** : named pipe
7. **l** : symbolic link

- **Regular file**

The regular file is a most common file type found on the Linux system. It governs all different files such as text files, images, binary files, shared libraries, etc. You can create a regular file with the **touch** command:

```
$ touch linuxcareer.com
$ ls -ld linuxcareer.com
-rw-rw-r-- 1 root root 0 Jan 10 12:52 linuxcareer.com
```

The first character of the **ls** command, in this case "-", denotes the identification code for the regular file. To remove a regular file you can use the **rm** command:

```
$ rm linuxcareer.com
```

- **Directory**

Directory is second most common file type found in Linux. Directory can be created with the **mkdir** command:

```
$ mkdir FileTypes
$ ls -ld FileTypes/
drwxrwxr-x 2 root root 4096 Jan 10 13:14 FileTypes/
```

As explained earlier, directory can be identified by "d" symbol from the **ls** command output. To remove empty directory use the **rmdir** command.

```
$ rmdir FileTypes
```

When trying to remove directory with the **rmdir** command, which contains additional files you will get an error message:

```
rmdir: failed to remove `FileTypes/': Directory not empty
```

In this case you need to use a command:

```
$ rm -r FileTypes/
```

- **Character device**

Character and block device files allow users and programs to communicate with hardware peripheral devices. For example:

```
$ ls -ld /dev/vmmon
crw----- 1 root root 10, 165 Jan  4 10:13 /dev/vmmon
```

In this case the character device is the vmware module device.

- **Block Device**

Block devices are similar to character devices. They mostly govern hardware as hard drives, memory, etc.

```
$ ls -ld /dev/sda
brw-rw---- 1 root disk 8, 0 Jan  4 10:12 /dev/sda
```

- **Socket Files**

Sockets files are used for communication between processes. Generally, they are used by services such as X windows, syslog and etc.

```
$ ls -ld /dev/log
srw-rw-rw- 1 root root 0 Jan  4 10:13 /dev/log
```

Sockets can be created by socket system call and removed by the **unlink** or **rm** commands.

- **Named Pipes**

Similarly as Local sockets, named pipes allow communication between two local processes. They can be created by the **mknod** command and removed with the **rm** command.

- **Symbolic Links**

With symbolic links an administrator can assign a file or directory multiple identities. Symbolic link can be thought of as a pointer to an original file. There are two types of symbolic links:

- hard links
- soft links

The difference between hard and soft links is that soft links use file name as reference and hard links use direct reference to the original file. Furthermore, hard links cannot cross file systems and partitions. To create symbolic soft link we can use **ln -s** command:

```
$ echo file1 > file1
$ ln -s file1 file2
$ cat file2
file1
$ ls -ld file2
lrwxrwxrwx 1 root root 5 Jan 10 14:42 file2 -> file1
```