<epam>

Files and Links

Linux Essentials

TRAINING

CENTER

<epam>

File types in Linux

Type	Prefix	Description
Ordinary files	-	Regular file, such as an ASCII text file, binary, image files
Directories	d	Files that store both ordinary and other special files and they are organized on the Linux file system in a hierarchy starting from the root (/) directory
Block files	b	Device files that provide a method of communication with system hardware components through the filesystem
Character special files	c	
Symbolic link files	l	A symbolic link files are files that point to other files, and they can either be directories or regular files.
Socket files	s	Files that provide a means of inter-process communication, but they can transfer data and information between process running on different environments.
Pipes	p	Files that allow inter-process communication by connecting the output of one process to the input of another.

File types in Linux

Ordinary file

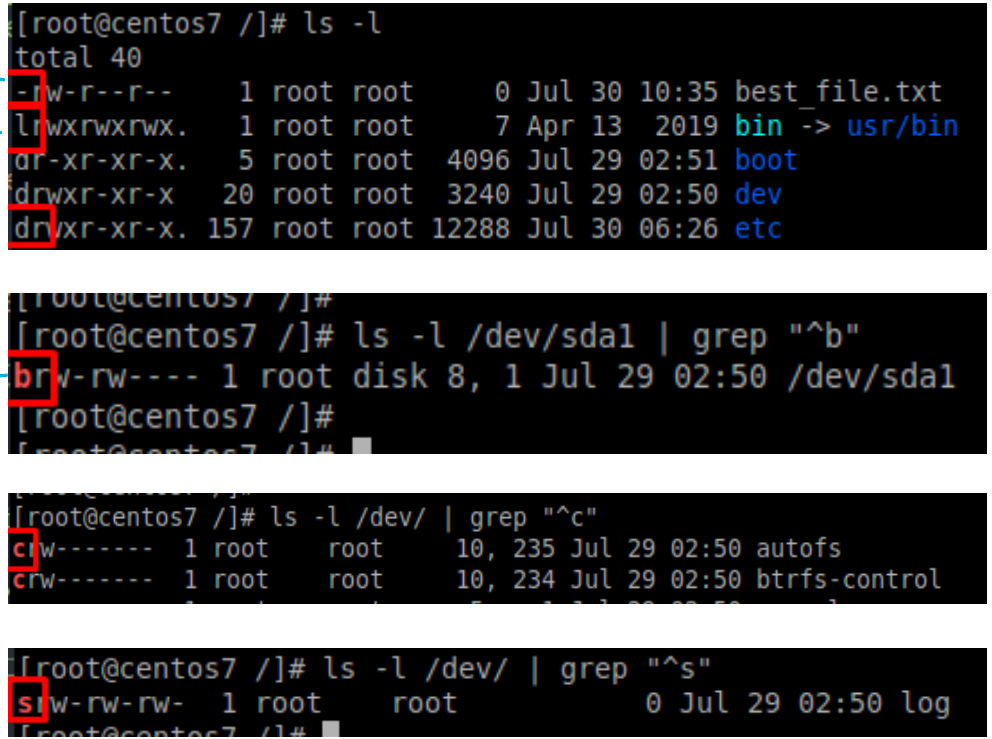
Symbolic link

Directory

Block files

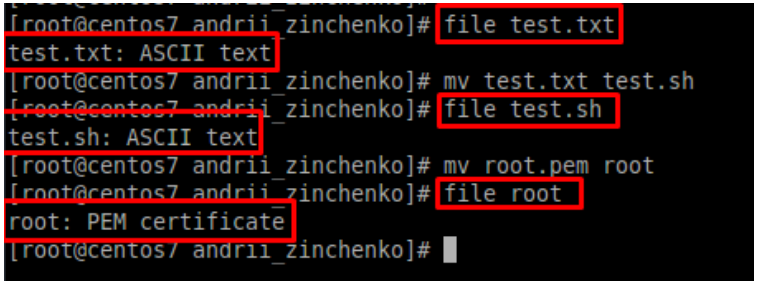
Character files

Socket files



File extension

- A file extension is a suffix that is added to the end of the base name of a file after period.
- OS like Windows require the use of extensions in order to identify what apps are associated with file type
- Unlike Windows, **Linux** doesn't rely on file extension. Instead it determines the type of a file via a code in the file header.
- Linux has a useful utility called **file** is used to identify file type



Commands with files

- **ls** – list information about the files (the current directory by default)

Key	Description
-a	List of all files including hidden
-d	Display data about directory, but not its content
-h	Using with –l displays files size not in bytes
-l	Output in long format
-r	Displays result of listing in reverse order
-t	Sort results of listing by changed time

```
[root@centos7 opt]# ls
containerd  rh  v10.txt  VBoxGuestAdditions-6.0.10
[root@centos7 opt]#
```

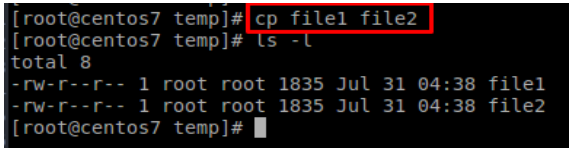
```
[root@centos7 opt]# ls -lh
total 8.0K
drwx--x--x. 4 root root   28 Jun  4  2019 containerd
drwxr-xr-x. 2 root root    6 Oct 30  2018 rh
-rwxrwxrwx. 1 root root 4.8K Jul 24  2019 v10.txt
drwxr-xr-x. 9 root root  147 Jul 22  2019 VBoxGuestAdditions-6.0.10
```

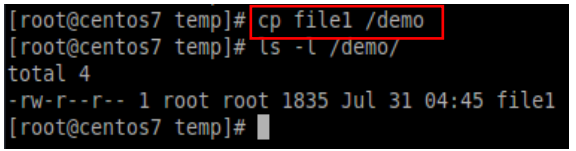
```
[root@centos7 etc]# ls -ltr | tail -n 10
-rw-r--r--   1 root root     1182 Jun 22 05:53 group
----------   1 root root      964 Jun 22 05:53 gshadow
-rw-r--r--   1 root root     2644 Jun 22 05:53 passwd
----------   1 root root     1517 Jun 22 05:53 shadow
drwxr-xr-x   4 root root       37 Jun 22 08:29 nginx
drwxr-xr-x.  2 root root      257 Jun 22 08:29 logrotate.d
drwxr-xr-x.  2 root root     4096 Jul 22 02:45 bash_completion.d
-rw-r--r--   1 root root      158 Jul 27 11:37 hosts
-rw-r--r--   1 root root       99 Jul 30 06:26 resolv.conf
drwxr-xr-x.  5 root lp       4096 Jul 30 10:41 cups
[root@centos7 etc]#
```

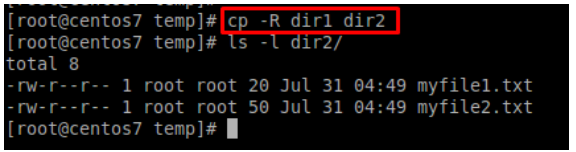
Commands with files

- **cp** – copy files and directories

Command	Result
cp file1 file2	Copy existing **file1** to a new **file2**
cp file1 /demo	Copy **file1** to directory **/demo**, located in root directory
cp -R dir1 dir2	Copy directory **dir1** to **dir2** recursively.



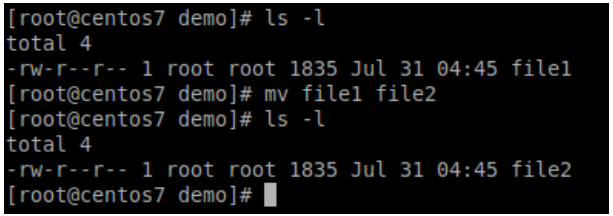


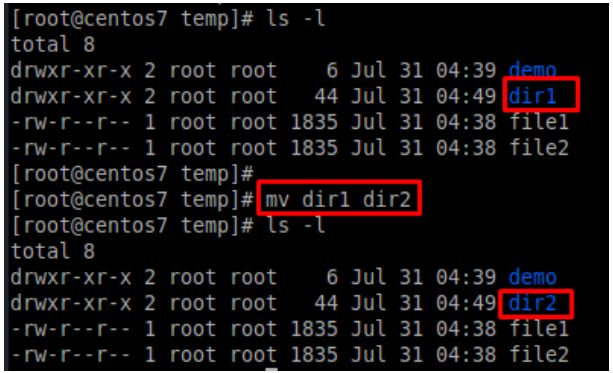


Commands with files

- **mv** – move and rename files

Command	Result
mv file1 file2	Rename **file1** to a **file2**
mv /dir1 /dir2	Rename directory **dir1** to **dir2**. If directory **dir2** exists and not empty, error message arise.
mv file1 /dir1	Move **file1** to directory **dir1** located in root directory





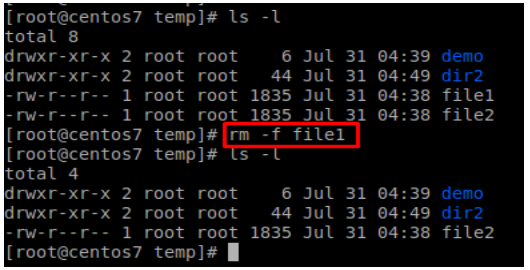
Commands with files

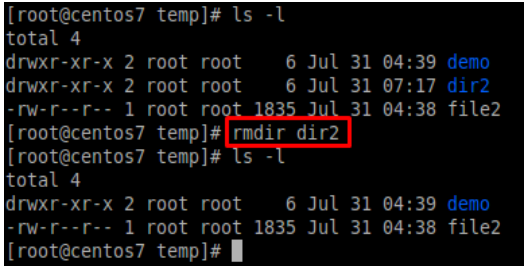
- **rm** – delete files and directories
- **rmdir** – delete empty directory

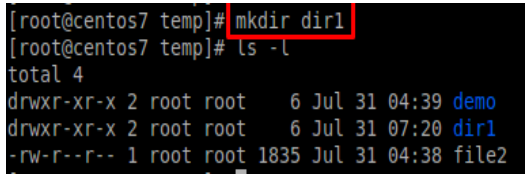
Command	Result
rm -f file1	Delete **file1**, -f option doesn't ask about confirmation
rm -R dir1	Delete **dir1** and its content.
rmdir dir1	Delete empty directory **dir1**

- **mkdir** – create directory

Command	Result
mkdir dir1	Create directory **dir1**

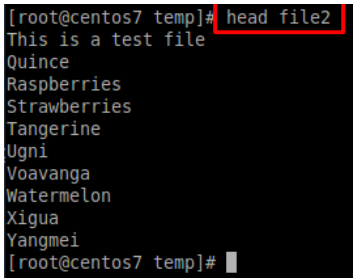


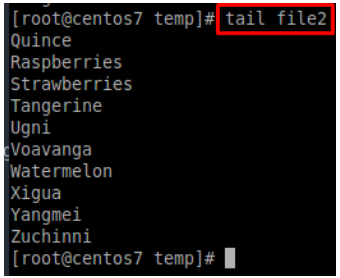


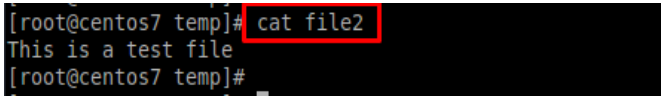


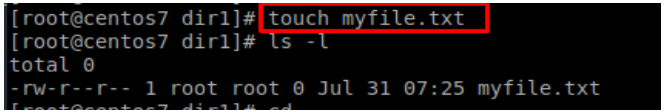
Commands with files

- **cat** – display content of file
- **touch** – create an ordinary file
- **less –** displays the contents of a file or a command output, one page at a time
- **tail –** print last 10 lines of a file.
- **head –** print first 10 lines of a file.









Link types in Linux

Type	Command	Description
Hard Link	ln SOURCE LINK	Hard link is a file that points to the same underlying inode, as another file. With a hard links you can link only to files (and not directories); you cannot reference a file on a different disk or volume, and they reference the same inode as the original source. A hard link will continue to remain usable, even if the original file is removed
Soft Link	ln -s SOURCE LINK	Symbolic links can link to directories, reference a file/folder on a different disk or volume, will exist as a broken (unusable) link if the original location is deleted, reference abstract filenames and directories (as opposed to physical locations), and are given their own, unique inode.

Hard & Soft Links

```
[root@centos7 temp]# ln file2 f2
[root@centos7 temp]# ls -l
total 8
drwxr-xr-x 2 root root   6 Jul 31 04:39 demo
drwxr-xr-x 2 root root  24 Jul 31 07:25 dir1
-rw-r--r-- 2 root root 110 Jul 31 07:35 f2
-rw-r--r-- 2 root root 110 Jul 31 07:35 file2
[root@centos7 temp]#
```

Hard link – a new file pointed to original

Original file "file2"

```
[root@centos7 temp]# ln -s dir1/myfile.txt file
[root@centos7 temp]# ls -l
total 4
drwxr-xr-x 2 root root   6 Jul 31 04:39 demo
drwxr-xr-x 2 root root  24 Jul 31 07:25 dir1
lrwxrwxrwx 1 root root  15 Jul 31 09:13 file -> dir1/myfile.txt
-rw-r--r-- 1 root root 110 Jul 31 07:35 file2
[root@centos7 temp]#
```

Symbolic link – reference to another file

```
[root@centos7 temp]# cat file
Example of symbolic link
[root@centos7 temp]# cat dir1/myfile.txt
Example of symbolic link
```