

## Demonstration

### File Categories in Linux

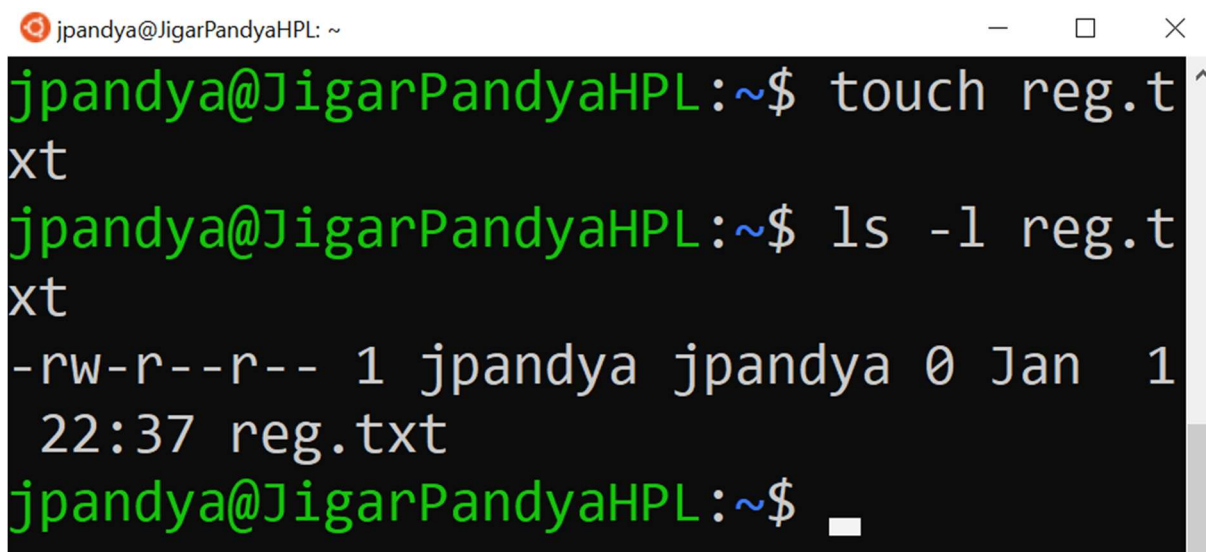
Know that `ls -l` command shows long listing of files. First character has meaning as below for appropriate file category.

Notice that this file categories are architecture-based categories on the usage of files and not just extensions.

-	Regular File
d	Directory File
p	Named pipe/FIFO for IPC
s	Socket for network communication
c	Character device file
l	Soft Link
b	Block file
...	

### Regular files

These are normal files i.e. text file.

A terminal window titled 'jpandya@JigarPandyaHPL: ~' with standard window controls. The terminal shows the following commands and output:

```
jpandya@JigarPandyaHPL:~$ touch reg.txt
jpandya@JigarPandyaHPL:~$ ls -l reg.txt
-rw-r--r-- 1 jpandya jpandya 0 Jan 1 22:37 reg.txt
jpandya@JigarPandyaHPL:~$
```

### Directory Files

Directory is a type of file which can be operated upon with commands. When we create a subdirectory, its very much like editing its parent directory and adding one more entry. The name of files and subdirectories are actually written as content of directory housing them.

```
jpandya@JigarPandyaHPL: ~  
jpandya@JigarPandyaHPL:~$ mkdir demo  
ir  
jpandya@JigarPandyaHPL:~$ ls -ld demo  
dir  
drwxr-xr-x 1 jpandya jpandya 4096 Jan  
1 22:33 demodir  
jpandya@JigarPandyaHPL:~$
```

### Character device file

Terminal is a type of file only. We can literally write to it and it will be read by hardware to display on the screen. Also is true for a printer like devices.

```
jpandya@JigarPandyaHPL: ~  
jpandya@JigarPandyaHPL:~$ tty  
/dev/tty2  
jpandya@JigarPandyaHPL:~$ ls -l /dev/tty2  
crw-rw---- 1 jpandya tty 4, 2 Jan 1 22:10 /dev/tty2  
jpandya@JigarPandyaHPL:~$
```

```
jpandya@JigarPandyaHPL: ~  
jpandya@JigarPandyaHPL:~$ tty  
/dev/tty2  
jpandya@JigarPandyaHPL:~$ Hi  
jpandya@JigarPandyaHPL: ~  
jpandya@JigarPandyaHPL:~$ echo "Hi">/dev/tty2  
jpandya@JigarPandyaHPL:~$
```

A pseudo file used to flush access logs. The file size will always be zero no matter how much you write to it. Its an example of a complete consumer.

```
jpandya@JigarPandyaHPL: ~  
jpandya@JigarPandyaHPL:~$ ls -l /dev/null  
crw-rw-rw- 1 root root 1, 3 Jan  1 22:10 /dev/null  
jpandya@JigarPandyaHPL:~$
```

## PIPE

A named pipe is a type of file in which process can write and another process can read from. Pipe is an example of FiFO (First In First Out). The image below shows that in left terminal a pipe is created using `mkfifo` command. Right hand side is actually waiting on data from the pipe. The user running command on left side to flow data which were displayed as output on right side automatically.

```
jpandya@JigarPandyaHPL: ~/demopipe  
jpandya@JigarPandyaHPL:~/demopipe$ mkfifo testpipefile  
jpandya@JigarPandyaHPL:~/demopipe$ ls -l testpipefile  
prw-r--r-- 1 jpandya jpandya 0 Jan  1 22:14 testpipefile  
jpandya@JigarPandyaHPL:~/demopipe$  
jpandya@JigarPandyaHPL:~/demopipe$  
jpandya@JigarPandyaHPL:~/demopipe$ echo "Hi please stream this data to pipe" > testpipefile  
jpandya@JigarPandyaHPL:~/demopipe$  
jpandya@JigarPandyaHPL:~/demopipe$  
jpandya@JigarPandyaHPL:~/demopipe$  
jpandya@JigarPandyaHPL:~/demopipe$  
jpandya@JigarPandyaHPL:~/demopipe$ cat testpipefile  
Hi please stream this data to pipe  
jpandya@JigarPandyaHPL:~/demopipe$
```

## Symbolic Link / Soft Link / Short-cut file

Know that short cut to another file/location is itself a type of file in linux. It holds only location of destination file. We can create using `ln -s source linkname` and verify that the inode numbers of shortcut differs than the destination.

```
jpandya@JigarPandyaHPL: ~$ ln -s /home/jpandya/demodir/datafile.txt df.txt
jpandya@JigarPandyaHPL: ~$ ls -li df.txt
lrwxrwxrwx 1 jpandya jpandya 34 Jan 1 22:52 df.txt -> /home/jpandya/demodir/datafile.txt
jpandya@JigarPandyaHPL: ~$ ls -li df.txt
9288674231551910 lrwxrwxrwx 1 jpandya jpandya 34 Jan 1 22:52 df.txt -> /home/jpandya/demodir/datafile.txt
jpandya@JigarPandyaHPL: ~$
```

```
jpandya@JigarPandyaHPL: ~/demodir$ ls -li datafile.txt
422124651103581 -rw-r--r-- 1 jpandya jpandya 0 Jan 1 22:50 datafile.txt
jpandya@JigarPandyaHPL: ~/demodir$
```

Moreover, we know digit 0, 1, 2 can be alternatively used for accessing standard input – 0 (keyboard), standard output – 1 (Screen) and standard error – 2 (Screen or log file).

```
lrwxrwxrwx 1 root root 15 Jan 1 22:10 /dev/stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root 15 Jan 1 22:10 /dev/stdin -> /proc/self/fd/0
lrwxrwxrwx 1 root root 15 Jan 1 22:10 /dev/stdout -> /proc/self/fd/1
jpandya@JigarPandyaHPL: ~$
```

See that in below deliberately I am generating error and redirecting it to a file.

```
jpandya@JigarPandyaHPL: ~$ mkdir demoparent
jpandya@JigarPandyaHPL: ~$ ls -ld demoparent
drwxr-xr-x 1 jpandya jpandya 4096 Jan 1 23:08 demoparent
jpandya@JigarPandyaHPL: ~$ chmod u-w demoparent
jpandya@JigarPandyaHPL: ~$ ls -ld demoparent
dr-xr-xr-x 1 jpandya jpandya 4096 Jan 1 23:08 demoparent
jpandya@JigarPandyaHPL: ~$ touch err.txt
jpandya@JigarPandyaHPL: ~$ cat err.txt
jpandya@JigarPandyaHPL: ~$ mkdir demoparent/demochilddir 2>err.txt
jpandya@JigarPandyaHPL: ~$ cat err.txt
mkdir: cannot create directory 'demoparent/demochilddir': Permission denied
jpandya@JigarPandyaHPL: ~$ chmod u+w demoparent
jpandya@JigarPandyaHPL: ~$ echo > err.txt
jpandya@JigarPandyaHPL: ~$ cat err.txt
jpandya@JigarPandyaHPL: ~$ mkdir demoparent/demochilddir 2>err.txt
jpandya@JigarPandyaHPL: ~$ cat err.txt
jpandya@JigarPandyaHPL: ~$ ls -ld demoparent
drwxr-xr-x 1 jpandya jpandya 4096 Jan 1 23:10 demoparent
jpandya@JigarPandyaHPL: ~$ ls -ld demoparent/demochilddir
drwxr-xr-x 1 jpandya jpandya 4096 Jan 1 23:10 demoparent/demochilddir
jpandya@JigarPandyaHPL: ~$
```

Socket

In the client-server architecture sockets help communicate and are endpoints. Using various socket programming library i.e. c sockets, win sockets, etc we can create and sockets. For example mysql daemon internally relies on a socket file.

### **Block File**

Mostly partitions are block files. May use fdisk -l to list out partitions and ls -l <device-path>. Know that having block I/O is faster than byte by byte processing.

Demonstrated By:

Jigar Pandya

<https://linkedin.com/in/jigar-pandya>

Document last updated on 1st Jan, 2021.