



Linux

Terminal commands

Dan Richter

07 Sep 2020

Let's start taking a look at some of the terminal commands used in Linux. If you are familiar with some of the dos/Windows commands, I will reference their counterparts for each of them.

Navigating the file system

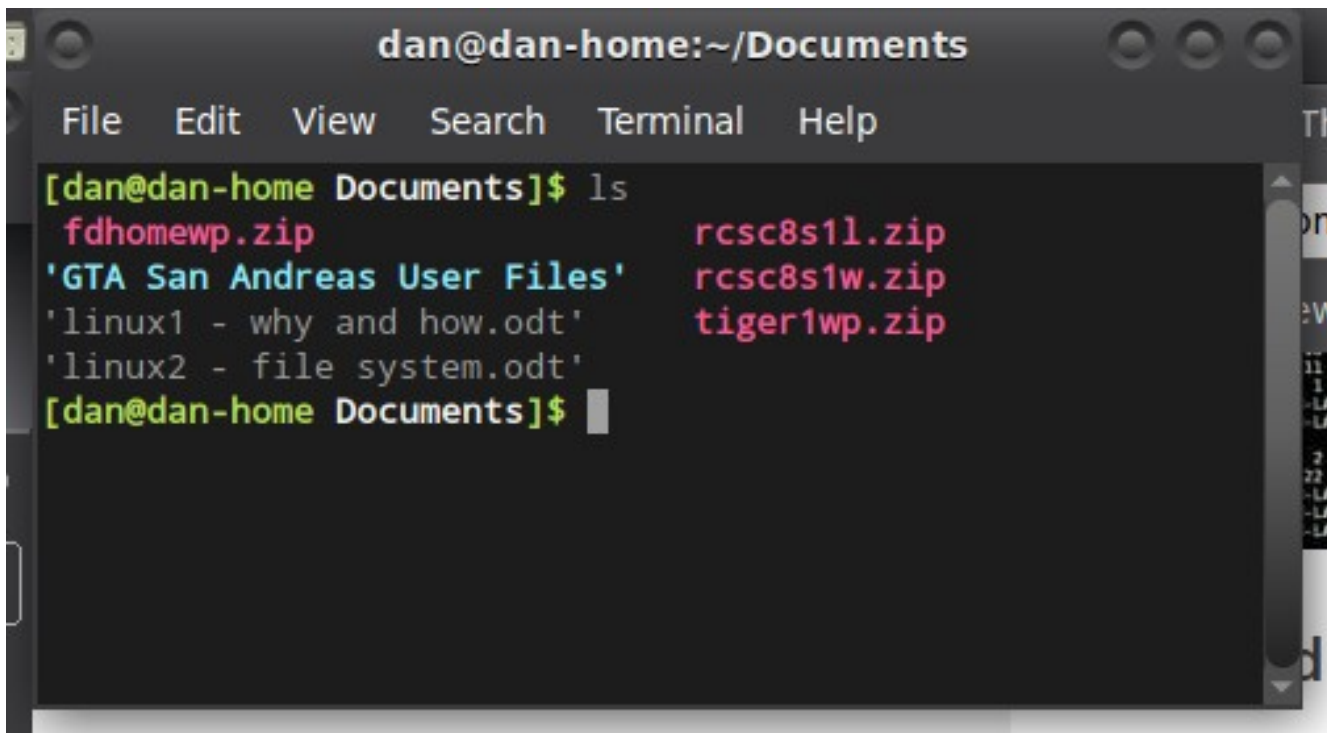
`pwd`

`pwd` stands for Print Work Directory. It does exactly what you think it would do, it tells you where you are in the file system.

```
dan@dan-home:~/Documents
File Edit View Search Terminal Help
[dan@dan-home Documents]$ pwd
/home/dan/Documents
[dan@dan-home Documents]$
```

ls

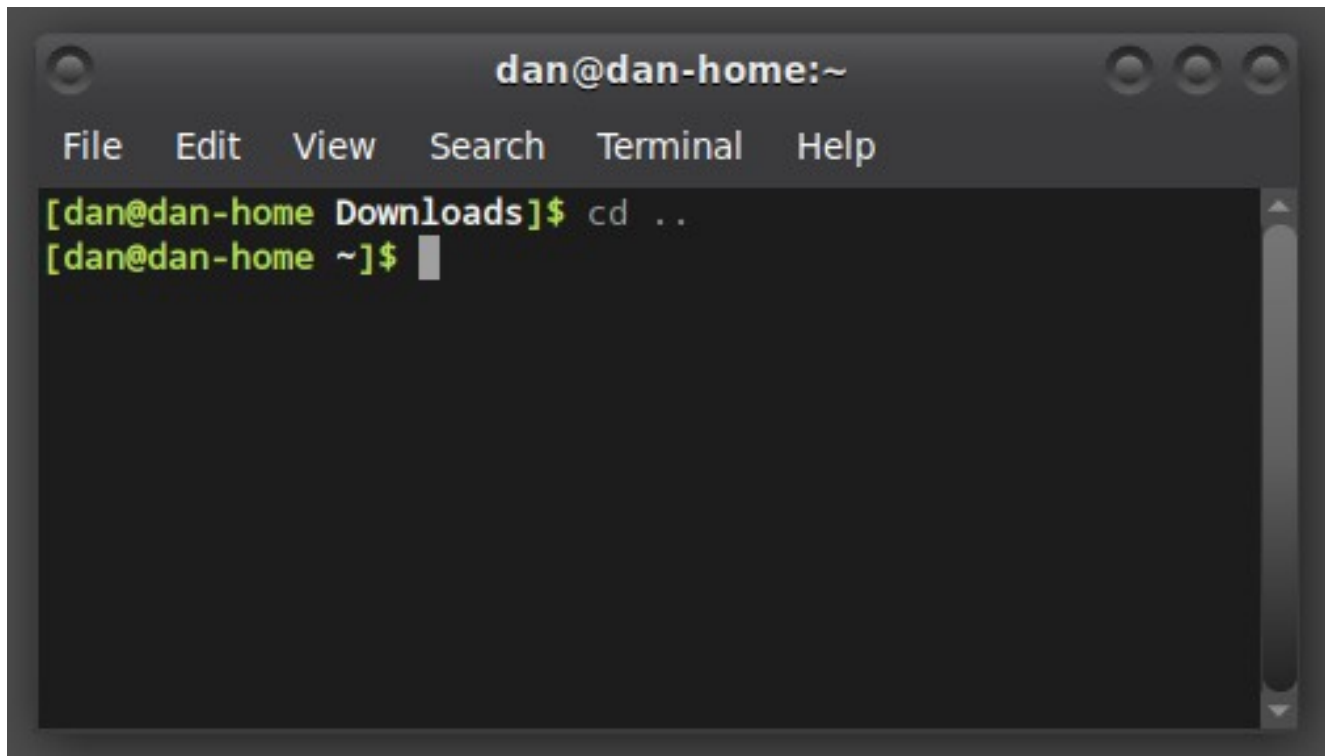
If you have ever used the 'dir' command in dos/Windows, this is the Linux equivalent. It will give you the contents of the directory that you are in.

A terminal window titled 'dan@dan-home:~/Documents' with a menu bar containing 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows the command '[dan@dan-home Documents]\$ ls' and its output. The output is a two-column list of files: 'fdhomewp.zip', 'rcsc8s1l.zip', 'GTA San Andreas User Files', 'rcsc8s1w.zip', 'linux1 - why and how.odt', 'tiger1wp.zip', and 'linux2 - file system.odt'. The prompt '[dan@dan-home Documents]\$' is shown again at the bottom with a cursor.

```
dan@dan-home:~/Documents
File Edit View Search Terminal Help
[dan@dan-home Documents]$ ls
fdhomewp.zip          rcsc8s1l.zip
'GTA San Andreas User Files' rcsc8s1w.zip
'linux1 - why and how.odt'  tiger1wp.zip
'linux2 - file system.odt'
[dan@dan-home Documents]$
```

cd

This works exactly the same as the 'cd' command in dos/Windows. It allows you to change from one directory to another. In the following example, the 'cd' command was used to go back (up) to the previous directory. In this case, from Downloads to Home.

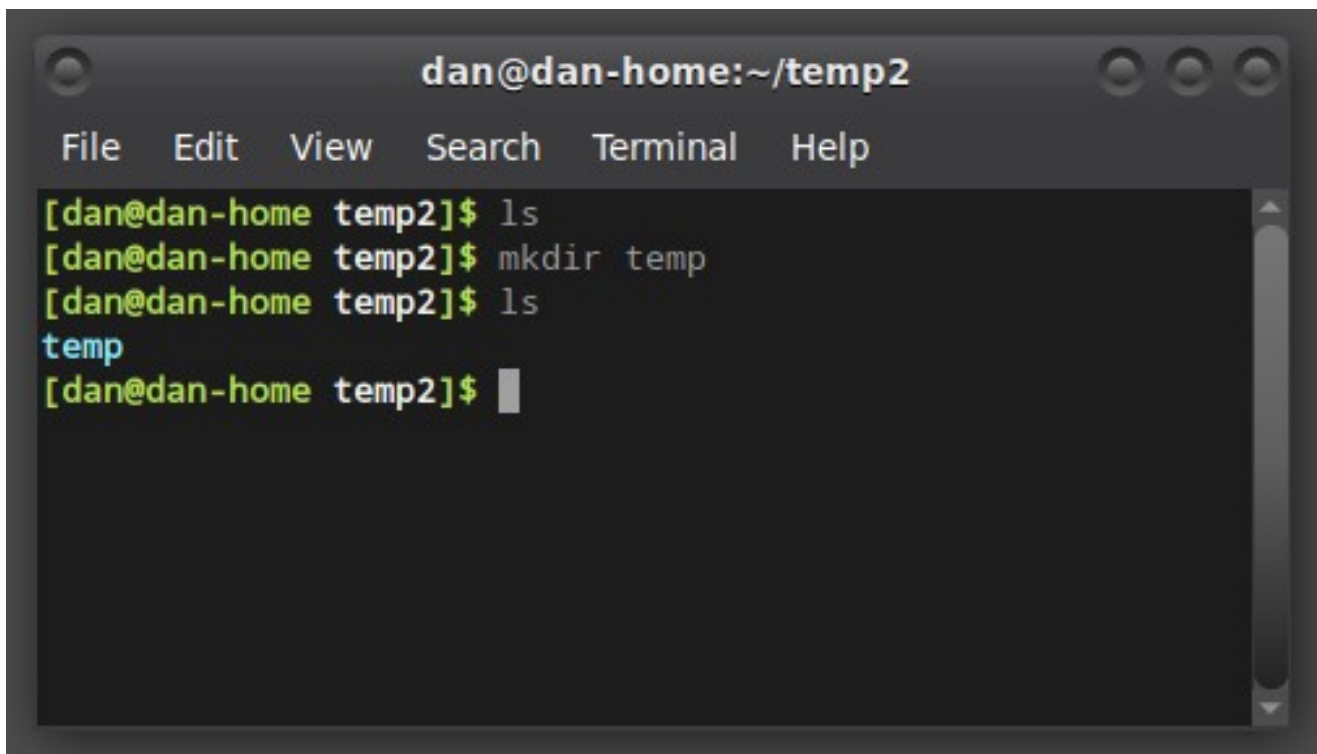
A screenshot of a terminal window titled 'dan@dan-home:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows a prompt '[dan@dan-home Downloads]\$' followed by the command 'cd ..'. The next line shows the prompt '[dan@dan-home ~]\$' with a cursor, indicating the command was successful and the user moved to the home directory.

```
dan@dan-home:~  
File Edit View Search Terminal Help  
[dan@dan-home Downloads]$ cd ..  
[dan@dan-home ~]$
```

As 'cd' is one of the most used commands, I wanted to give you a little more information about it. First of all, there are some shortcuts that can be used. One of which, I used in the previous example. Instead of typing 'cd /home/dan' to go back to the home directory, I typed in 'cd ..' which will take you back one directory, in this case from /home/dan/Documents to /home/dan. Another way this could be done is to use the '~' key. This represents your /home directory. So, I could have typed 'cd ~' and would have been taken to the home directory as well. Or, I could have typed in 'cd' by itself, and Linux would take me back to the home directory. All of these are very useful and quick ways to move around in the Linux file system.

mkdir

This command is used for creating a new directory. It is similar to the 'md' command in dos/Windows. As you can see in the following example, the temp2 directory did not contain anything. After running the 'mkdir temp', there was a sub-directory called temp inside of the directory.

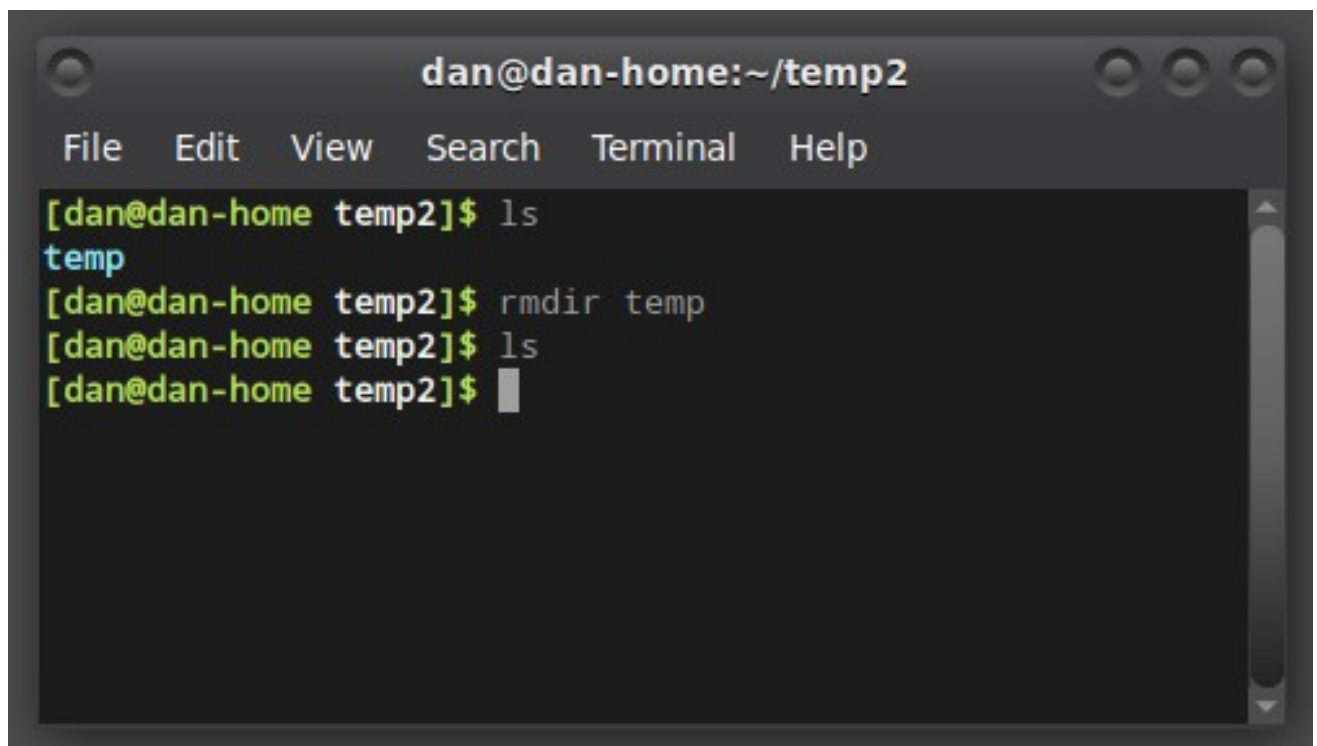
A terminal window titled 'dan@dan-home:~/temp2' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
[dan@dan-home temp2]$ ls  
[dan@dan-home temp2]$ mkdir temp  
[dan@dan-home temp2]$ ls  
temp  
[dan@dan-home temp2]$
```

The terminal window has a dark background. The prompt is green. The output 'temp' is blue. A vertical scrollbar is on the right.

`rmdir`

This is the opposite of the last command, as it will remove a directory from the command line. This time, in the example, you can see the 'temp' sub-directory was there, and after running the 'rmdir temp' command, it was removed.

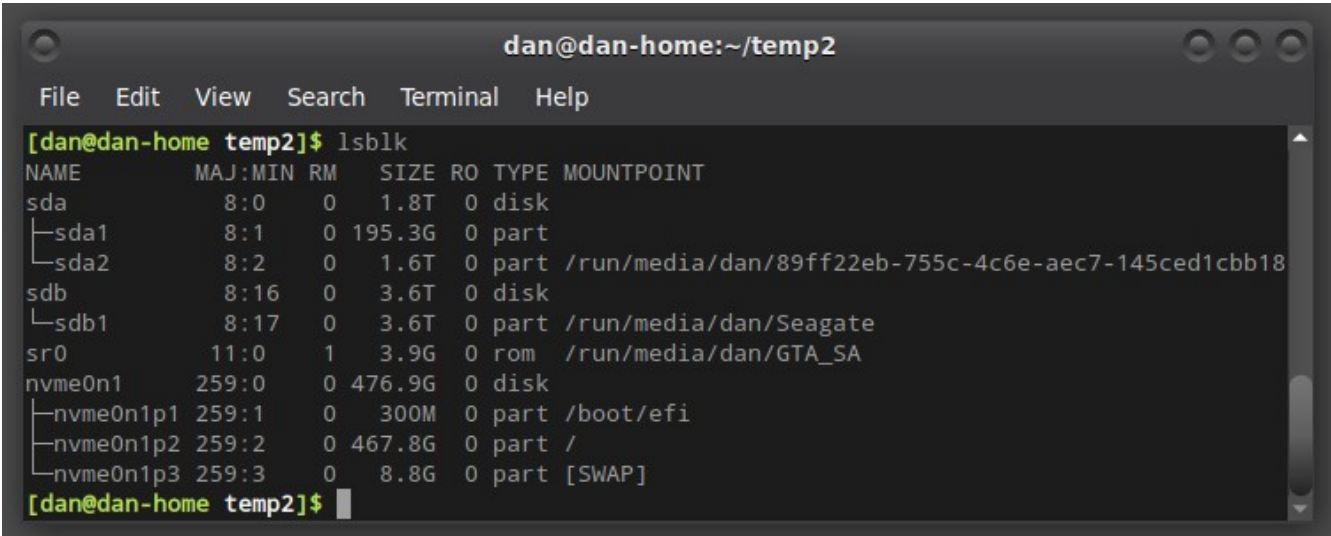
A terminal window titled 'dan@dan-home:~/temp2' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
[dan@dan-home temp2]$ ls  
temp  
[dan@dan-home temp2]$ rmdir temp  
[dan@dan-home temp2]$ ls  
[dan@dan-home temp2]$
```

The terminal window has a dark background. The prompt is green. The output 'temp' is blue. A vertical scrollbar is on the right.

lsblk

If you need to look and see a list of the block devices on your system, you can use this 'lsblk' command. In this example, it shows a tree structure of the block devices on my system.



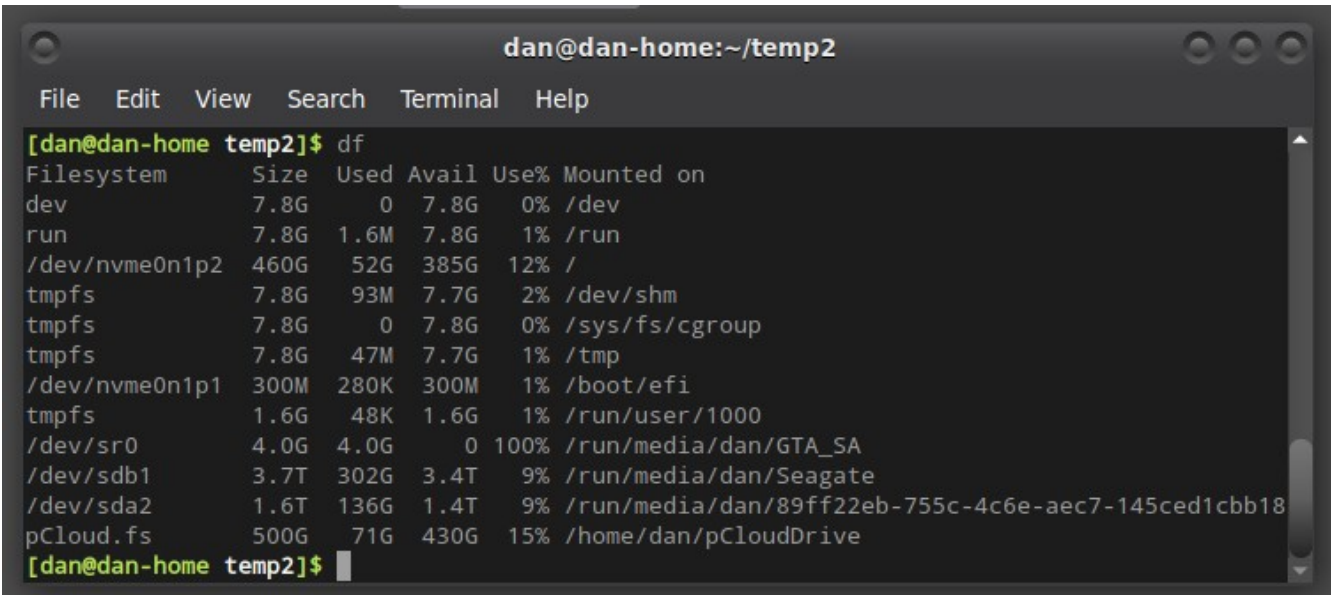
```
dan@dan-home:~/temp2
File Edit View Search Terminal Help
[dan@dan-home temp2]$ lsblk
NAME            MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda              8:0    0   1.8T  0 disk
├─sda1           8:1    0 195.3G  0 part
├─sda2           8:2    0   1.6T  0 part /run/media/dan/89ff22eb-755c-4c6e-aec7-145ced1cbb18
└─sdb            8:16    0   3.6T  0 disk
   └─sdb1         8:17    0   3.6T  0 part /run/media/dan/Seagate
sr0             11:0    1   3.9G  0 rom  /run/media/dan/GTA_SA
nvme0n1         259:0    0 476.9G  0 disk
├─nvme0n1p1      259:1    0   300M  0 part /boot/efi
├─nvme0n1p2      259:2    0 467.8G  0 part /
└─nvme0n1p3      259:3    0    8.8G  0 part [SWAP]
[dan@dan-home temp2]$
```

mount

If you need to manually mount a drive in Linux, the 'mount' command is used.

df

This is one of the most important commands that can be used in the terminal. This will show you the available space on each of the drives mounted on your system.



```
dan@dan-home:~/temp2
File Edit View Search Terminal Help
[dan@dan-home temp2]$ df
Filesystem      Size  Used Avail Use% Mounted on
dev             7.8G   0    7.8G   0% /dev
run             7.8G 1.6M   7.8G   1% /run
/dev/nvme0n1p2  460G  52G  385G  12% /
tmpfs           7.8G  93M   7.7G   2% /dev/shm
tmpfs           7.8G   0   7.8G   0% /sys/fs/cgroup
tmpfs           7.8G  47M   7.7G   1% /tmp
/dev/nvme0n1p1  300M 280K  300M   1% /boot/efi
tmpfs           1.6G  48K   1.6G   1% /run/user/1000
/dev/sr0        4.0G  4.0G   0 100% /run/media/dan/GTA_SA
/dev/sdb1       3.7T 302G  3.4T   9% /run/media/dan/Seagate
/dev/sda2       1.6T 136G  1.4T   9% /run/media/dan/89ff22eb-755c-4c6e-aec7-145ced1cbb18
pCloud.fs       500G  71G  430G  15% /home/dan/pCloudDrive
[dan@dan-home temp2]$
```

As this is already getting a bit lengthy, I'll stop here. In the next update, we'll take a look at some more commands that will give you a lot of power on a Linux system.