

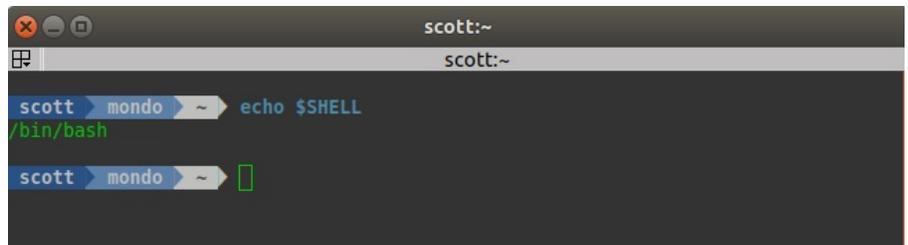
Scotti-BYTE Enterprise Consulting Services

OS Tutorial: Leveraging the Command Line

Modern operating systems all come with a graphical user interface (GUI). There is a tendency as a computer user to perform all operations using the GUI. Yet, in Microsoft Windows, Apple Mac OSX, and Linux operating systems there are a wealth of functions that can be performed from a simple command line. The command line, sometimes referred to as a console or terminal is an interface that allows entering textual commands to execute programs.

Many times, the command line is required to execute some operations because there may be no GUI equivalent command to achieve your goal. In this tutorial, we are going to focus on the Ubuntu Linux command line by contrasting similar commands in Windows. In Microsoft Windows, the command line is accessed by a program called cmd.exe. In Linux, you use a terminal application to access a command shell. There are many different command shells in Linux.

We will focus on Ubuntu which uses the GNU BASH command shell. To open a terminal use the CTRL-ALT-T command or search for terminal in the app drawer.

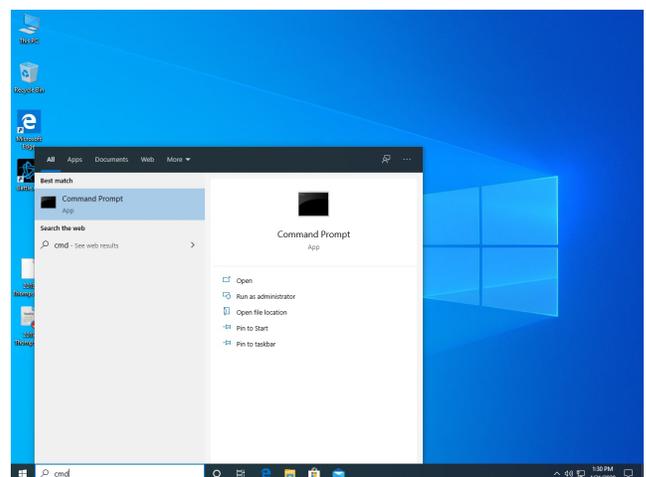


```
scott:~  
scott:~  
scott ~ondo ~ echo $SHELL  
/bin/bash  
scott ~ondo ~
```

To see which shell you are running, you can enter the "echo \$SHELL" command. In Linux, unlike Windows, upper and lowercase is extremely important. Commands must be entered exactly as described.

In Windows, the command shell usually is non-privileged. Windows 10 also uses the "Windows Powershell" shell which was invented in 2012 and provides many enhanced commands to access embedded functions in Windows instrumentation. In addition, it runs all of the standard DOS commands.

The legacy command shell can be accessed by clicking the Windows Start and then typing in "cmd".

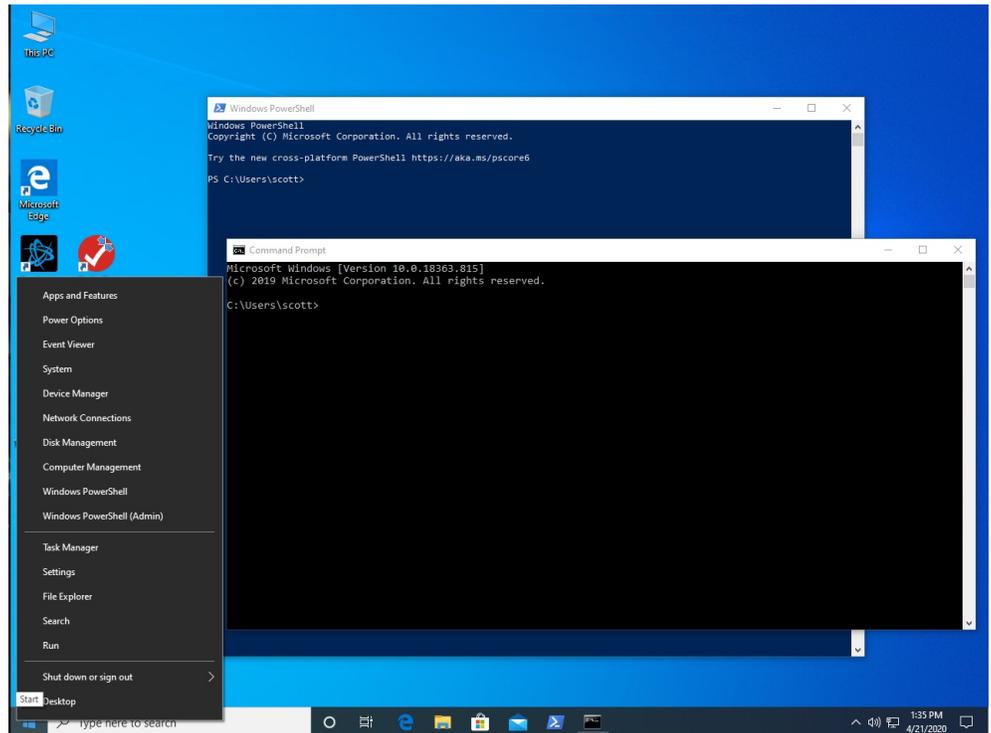


To access the Powershell, right click on Start and select either the PowerShell or the PowerShell with Admin privilege. You can also create a desktop shortcut for "cmd.exe" and then right click it to run with administrator privilege.

Windows marks certain functions as requiring admin privilege and those functions will not be executed unless you are granted admin privilege.

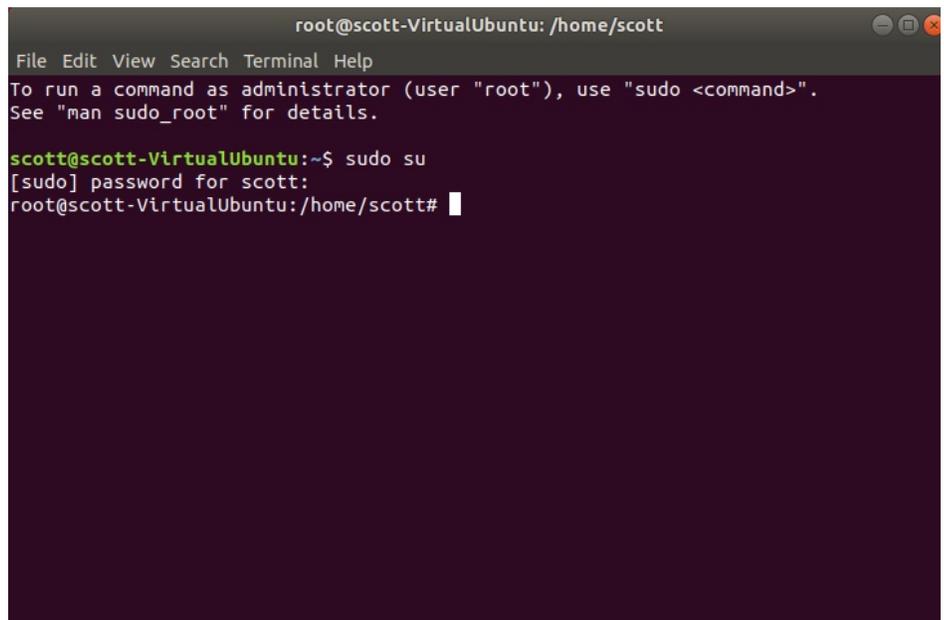
Ubuntu and Linux operating systems don't work the same way.

One thing that is the same is that only a user in Windows granted admin privilege on their user account can run admin functions. In Ubuntu, you must be a member of the "SuperUser" group in order to execute commands that require privilege.



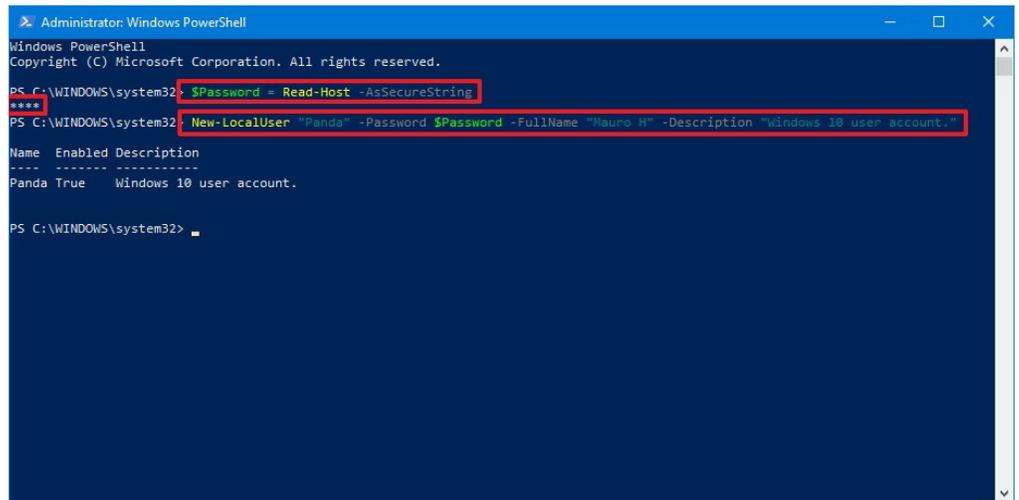
In Linux, the command "sudo" means, "do the operation with the superuser privilege". You can place "sudo" at the beginning of any command, or you can type "sudo su" to change over to the superuser. In Linux, the only type of privilege is "root" privilege. Root privilege means that you are the owner of the entire file system. When you are running as "superuser" you are running under the "root" username.

See the screen image at right and note the command prompt before privilege and the command prompt after obtaining root privilege. The prefix "root" reminds you that the terminal is running with privilege. You can type "exit" to return to the unprivileged shell terminal and exit again would exit the terminal application.



In Windows PowerShell, to create a new user account, execute the password command at right which will prompt you for a password and then execute the second command to create the user account.

To make the user "Panda" an administrator account, you would run an additional PowerShell command:



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> $Password = Read-Host -AsSecureString
****
PS C:\WINDOWS\system32> New-LocalUser "Panda" -Password $Password -FullName "Mauro H" -Description "Windows 10 user account."

Name Enabled Description
----
Panda True Windows 10 user account.

PS C:\WINDOWS\system32>
```

`Add-LocalGroupMember -Group Administrators -Member Panda`

In Ubuntu, usernames must be all lowercase. If you create a username with an uppercase character, it will not perform well in all situations. To add a user "test", the command is:

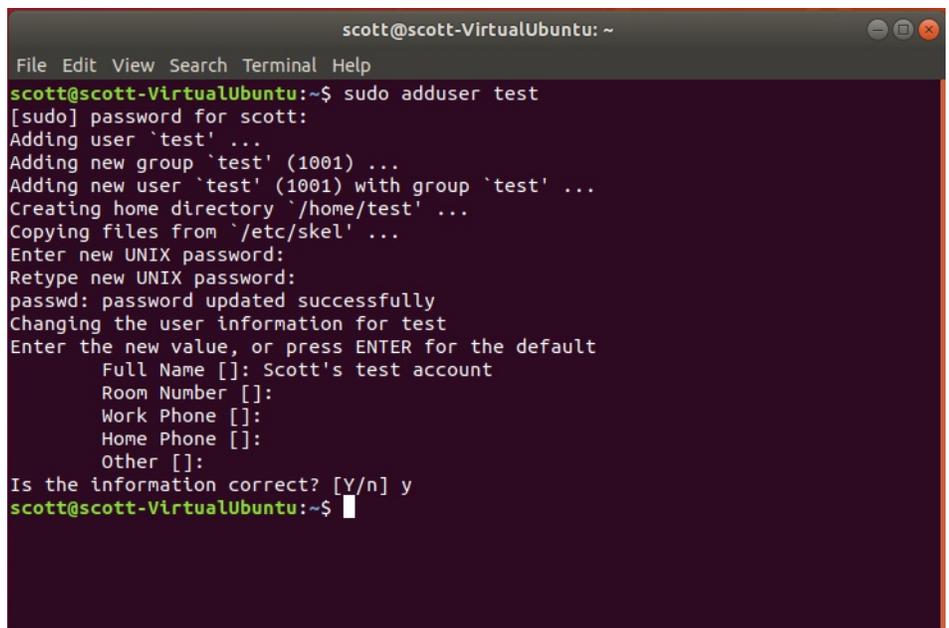
`sudo adduser test`

Follow the prompts in the screen image at right. You will be prompted for your password when the "sudo" command is executed.

In order to run "sudo", you must be a member of sudo group.

To make the "test" user a member of the sudo group enter the following command.

`usermod -aG sudo test`



```
scott@scott-VirtualUbuntu: ~
File Edit View Search Terminal Help
scott@scott-VirtualUbuntu:~$ sudo adduser test
[sudo] password for scott:
Adding user `test' ...
Adding new group `test' (1001) ...
Adding new user `test' (1001) with group `test' ...
Creating home directory `/home/test' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for test
Enter the new value, or press ENTER for the default
Full Name []: Scott's test account
Room Number []:
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n] y
scott@scott-VirtualUbuntu:~$
```

When you install software in Windows, you are generally running an MSI package which places files in appropriate folders, makes registry entries and creates shortcuts.

Installation of software in Ubuntu is much the same except that there is no registry. Every Linux distro uses a package manager to handle installation and removal of software. The Ubuntu package manager is "aptitude" or "apt" for short. Packages can be installed from the command line, from the Ubuntu Software center, or by double clicking them in the file manager.

In Linux, adding or removing software and having all necessary dependencies satisfied and managed is extremely intuitive and easy. To update the package database to see available updates:

```
sudo apt update
```

After the update command, initiate upgrades of software that have newer versions:

```
sudo apt upgrade
```

To install a new package in Ubuntu, an example of an installation of a package called "net-tools" follows:

```
sudo apt install net-tools
```

To remove a package from Ubuntu:

```
sudo apt remove --purge net-tools
```

The "--purge" is optional and removes the program settings as well as the package.

To remove dependencies which may no longer be needed in Ubuntu:

```
sudo apt autoremove
```

If the package manager ever indicates that it is corrupt or has broken packages:

```
sudo apt update --fix-missing  
sudo dpkg --configure -a  
sudo apt install -f
```

To clean up leftover entries in the package repository:

```
sudo apt clean  
sudo apt update
```

In Ubuntu, the apt package manager searches repositories on the system in order to find information to install programs.

To see these sources, open the app drawer and type "Software". Choose the "Software & Updates" app and the first tab lists "Ubuntu Software". The "Other Software" tab lists any add-on repositories that you may have defined.



Sometimes, a program is not a part of the Ubuntu repositories, or there may be a newer version of a program in a repository provided by the author. For example, the GNU Image Manipulation Program (GIMP) is available in the Ubuntu Software Center and can also be installed from the command line.

However, a much newer version of GIMP can be installed from the repository offered by the GIMP authors:

```
sudo add-apt-repository ppa:otto-kesselgulasch/gimp
sudo apt update
sudo apt install gimp
```

the "add-apt-repository" adds a new repository to search when performing installations. The "sudo apt update" updates the search indexes. The install command will now install the newer version of "gimp" found in that repository as opposed to the older version in the standard ubuntu repositories.

Once, you have defined a repository, anytime there is a new version of an installed program, all you need to do is:

```
sudo apt update
sudo apt upgrade
```

Then all new available versions would be installed.

When you run "sudo apt upgrade", it only upgrades that which has a new release available to the platform, as defined in /etc/apt/sources.list.

When you run "sudo apt dist-upgrade", it will intelligently install or remove packages as needed, in order to complete the upgrade. "sudo apt dist-upgrade" has a smart conflict resolution system, so it will attempt to upgrade the most important packages, at the expense of those deemed less important.

"sudo apt upgrade" does not remove packages, it only upgrades. Can you use "sudo apt dist-upgrade" as your regular upgrade tool? Certainly. Just remember, it will delete software--if it is required to complete the process.

There is another command line that can actually upgrade the version of Ubuntu installed if one is available:

```
sudo do-release-upgrade
```

The do-release-upgrade command is responsible for upgrading from one Ubuntu release to another. So if you want to upgrade from Ubuntu 18.10 to 19.04, you use do-release-upgrade.

However, in order to use this command, the system must first be fully upgraded. To do that, you should first run sudo apt upgrade, followed by sudo apt dist-upgrade. When those two complete, you can then run sudo do-release-upgrade and wait for the magic to complete.

There is a "Software Update" tool in the app drawer. It does the same as:

```
sudo apt update  
sudo apt upgrade
```



After the "software updater" runs:



Most Linux users get used to the command line interface and tend to use it more often than the GUI commands because of the perceived

level of control over what you are doing. Distros like Ubuntu have a wide variety of GUI commands making it a distro more comfortable to Windows users.

There is a program called neofetch that you can install that will give you your overall configuration of your system.



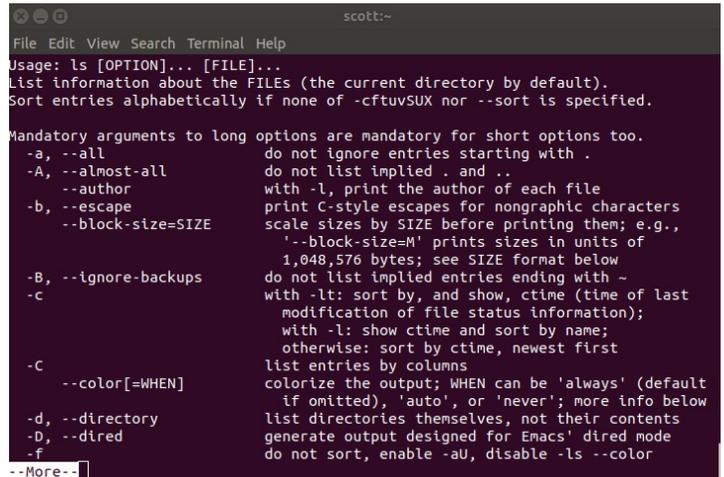
The equivalent of the "DIR" command in Windows is "ls" in Linux. All commands in linux are generally the name of the command followed by a dash (-) and then options which are typically a string of letters.

You can get help on a command "ls --help". One very powerful Linux feature is the pipe (|) denoted by the vertical bar. A pipe causes the output of the program listed first to become the input of the program listed after it. A "more" command causes a pause after it sees about 24 lines of output. So, the command "ls --help | more" will list the first page of help on the "ls" command and when you press the space bar it lists the next page of help.

The "enter" key moves the output down by a line instead of a page.

CTRL-C exits "more" and returns to the command line.

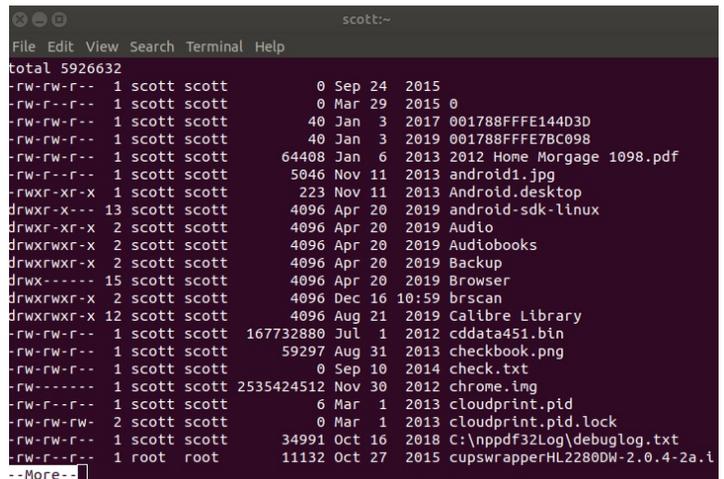
The "more" command is valuable in listing any data that extends for more than one page on the terminal.



```
scott:~  
File Edit View Search Terminal Help  
Usage: ls [OPTION]... [FILE]...  
List information about the FILES (the current directory by default).  
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.  
  
Mandatory arguments to long options are mandatory for short options too.  
-a, --all do not ignore entries starting with .  
-A, --almost-all do not list implied . and ..  
--author with -l, print the author of each file  
-b, --escape print C-style escapes for nongraphic characters  
--block-size=SIZE scale sizes by SIZE before printing them; e.g.,  
'--block-size=M' prints sizes in units of  
1,048,576 bytes; see SIZE format below  
-B, --ignore-backups do not list implied entries ending with ~  
-c with -lt: sort by, and show, ctime (time of last  
modification of file status information);  
with -l: show ctime and sort by name;  
otherwise: sort by ctime, newest first  
-C list entries by columns  
--color[=WHEN] colorize the output; WHEN can be 'always' (default  
if omitted), 'auto', or 'never'; more info below  
-d, --directory list directories themselves, not their contents  
-D, --dired generate output designed for Emacs' dired mode  
-f do not sort, enable -aU, disable -ls --color  
--More--
```

The command at the right is 'ls -l | more'. This is a "long" directory listing with the more command. A long listing shows columns with permissions, owner name, group name, file size, creation date, and filename.

If you had entered "ls -al" you would have gotten a long listing that also lists hidden files. Any file starting with a period in the filename is a hidden file in Linux.



```
scott:~  
File Edit View Search Terminal Help  
total 5926632  
-rw-rw-r-- 1 scott scott 0 Sep 24 2015  
-rw-r--r-- 1 scott scott 0 Mar 29 2015 0  
-rw-rw-r-- 1 scott scott 40 Jan 3 2017 001788FFFE144D3D  
-rw-rw-r-- 1 scott scott 40 Jan 3 2019 001788FFFE7BC098  
-rw-rw-r-- 1 scott scott 64408 Jan 6 2013 2012 Home Mortgage 1098.pdf  
-rw-r--r-- 1 scott scott 5046 Nov 11 2013 android1.jpg  
-rwxr-xr-x 1 scott scott 223 Nov 11 2013 Android.desktop  
drwxr-x--- 13 scott scott 4096 Apr 20 2019 android-sdk-linux  
drwxr-xr-x 2 scott scott 4096 Apr 20 2019 Audio  
drwxrwxr-x 2 scott scott 4096 Apr 20 2019 Audiobooks  
drwxrwxr-x 2 scott scott 4096 Apr 20 2019 Backup  
drwx----- 15 scott scott 4096 Apr 20 2019 Browser  
drwxrwxr-x 2 scott scott 4096 Dec 16 10:59 brscan  
drwxrwxr-x 12 scott scott 4096 Aug 21 2019 Calibre Library  
-rw-rw-r-- 1 scott scott 167732880 Jul 1 2012 cddata451.bin  
-rw-rw-r-- 1 scott scott 59297 Aug 31 2013 checkbook.png  
-rw-rw-r-- 1 scott scott 0 Sep 10 2014 check.txt  
-rw----- 1 scott scott 2535424512 Nov 30 2012 chrome.img  
-rw-r--r-- 1 scott scott 6 Mar 1 2013 cloudprint.pid  
-rw-rw-rw- 2 scott scott 0 Mar 1 2013 cloudprint.pid.lock  
-rw-rw-r-- 1 scott scott 34991 Oct 16 2018 C:\nppdf32Log\debuglog.txt  
-rw-r--r-- 1 root root 11132 Oct 27 2015 cupswrapperHL2280DW-2.0.4-2a.l  
--More--
```

Files in Linux have ownership by a username and a user group. The permissions fields above starts with a dash or a "d". A "d" means that the file is a directory. The next three letters are permissions for the owner. The next three letters are permissions for the group and the last three letters are permissions for "other". Other is anyone else on the system who is not the owner or group. The Three types of permission are "r" for readonly, "w" for read/write and "x" for execute. If a dash appears instead of the letter, it means that permission type is turned off.

To change the owner of a file:

chown testuser filename

To change the owner of a directory:

```
chown -R testuser dir-name
```

The uppercase "-R" means to make the ownership change recursive down into the directory structure.

There is also a change group:

```
chgrp testgroup filename
```

In Windows, disk drives are considered devices and a device has partitions that are mounted with a drive letter. In linux, the top level of the file system is root denoted by the forward slash (/). Additional disks and partitions are mounted to the one root partition making the file system appear either wider or deeper.

To mount a device, you create a folder:

```
sudo mkdir /media/mystuff
```

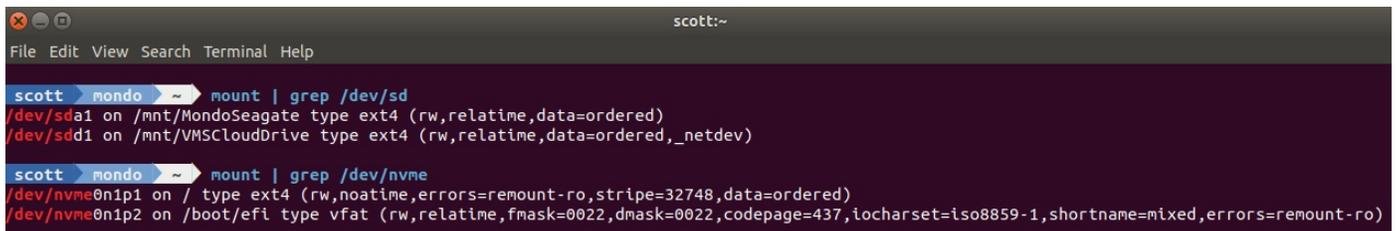
To mount a new partition to make it part of the file system, you use the "mount" command:

```
sudo mount /dev/sdb1 /media/mystuff
```

Since everything in Linux is a file, you will notice the devices look like file in the "/dev" folder. Typically, disk drives have names like sda, sdb, sdc and so on. Partitions have numbers. So, the first partition on the second hard drive would have a name like "/dev/sdb1". Newer devices like nvme drives have names like "/dev/nvme0n1". The first partition might be "/dev/nvme0n1p1".

Typically, the "sd" nomenclature means SCSI drive, although most every drive is some variation of SCSI today like SATA or SAS. There is also a "/dev/hd" naming convention. You don't see that often anymore because "hd" drives were the old IDE interface. Optical drives like CDs and DVDs have names like "/dev/sr0".

The mount command can also give status about which drives are mounted. A lot of specialty mounts exist on every system. There is a search command called "grep" (Globally search a Regular Expression and Print) that you can pipe to list particular drive mounts:



```
scott@mondo:~$ mount | grep /dev/sd
/dev/sda1 on /mnt/MondoSeagate type ext4 (rw,relatime,data=ordered)
/dev/sdd1 on /mnt/VMScloudDrive type ext4 (rw,relatime,data=ordered,_netdev)

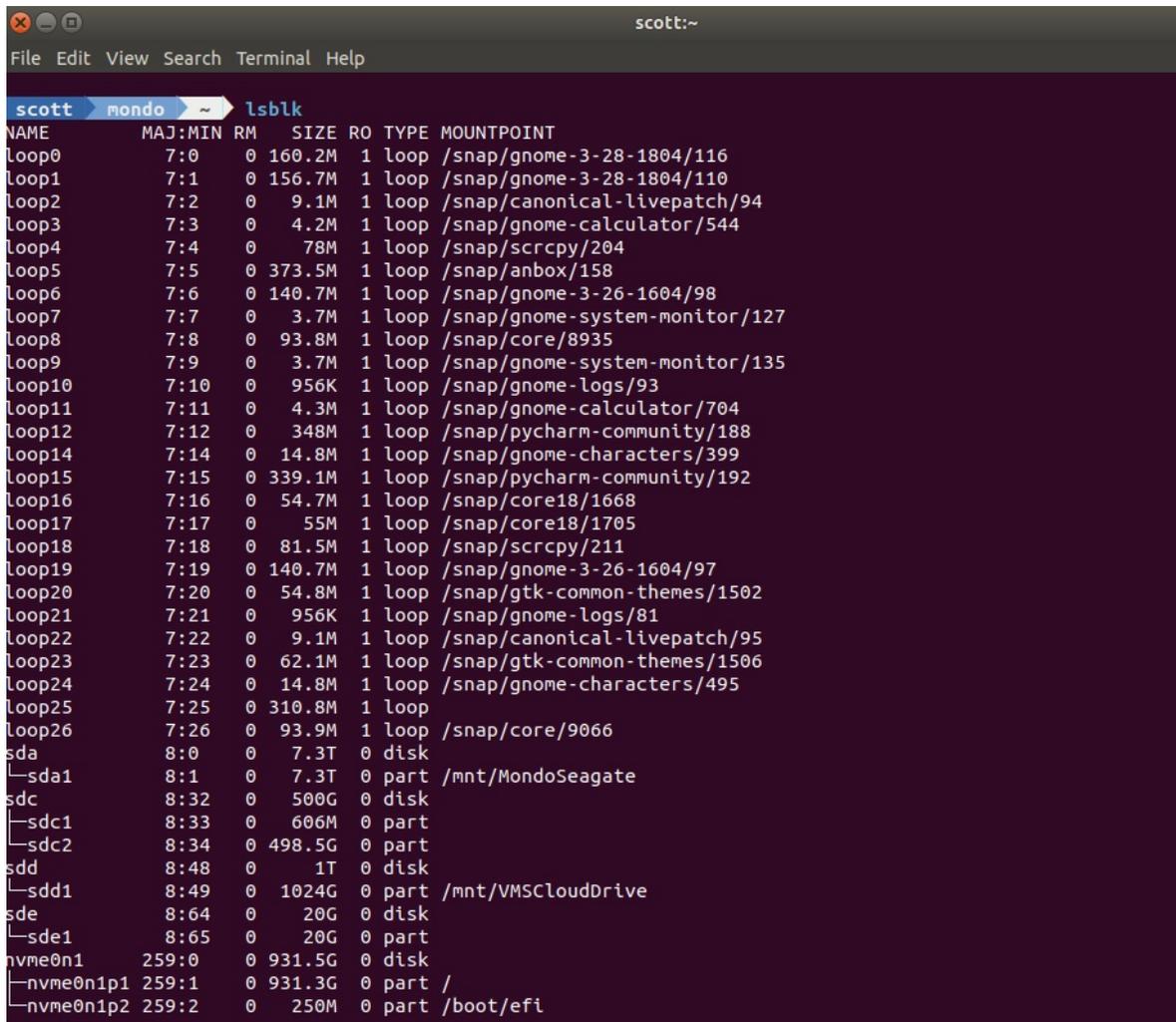
scott@mondo:~$ mount | grep /dev/nvme
/dev/nvme0n1p1 on / type ext4 (rw,noatime,errors=remount-ro,stripe=32748,data=ordered)
/dev/nvme0n1p2 on /boot/efi type vfat (rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro)
```

It is easy to see the power of pipes with "grep" used in the above "mount" command.

As mentioned above, any device that allows reading or writing data blocks is a "Block Device". In Ubuntu hard drives are represented by their drivers either as /hdX for an IDE drive or /sdX for a SCSI (or SATA in a modern system) drive.

You can list all block drives with the "lsblk" command whether they are mounted or not. This can be confusing because this command also lists loop devices which are block devices that map data blocks not to a physical device such as a hard disk or optical disk drive, but to the blocks of a regular file in a filesystem or to another block device.

For example, a loop device could be used to mount an ISO image as though it were a disk drive.



```
scott@mondo:~$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0       7:0      0 160.2M 1 loop /snap/gnome-3-28-1804/116
loop1       7:1      0 156.7M 1 loop /snap/gnome-3-28-1804/110
loop2       7:2      0   9.1M 1 loop /snap/canonical-livepatch/94
loop3       7:3      0   4.2M 1 loop /snap/gnome-calculator/544
loop4       7:4      0   78M 1 loop /snap/scrcpy/204
loop5       7:5      0 373.5M 1 loop /snap/anbox/158
loop6       7:6      0 140.7M 1 loop /snap/gnome-3-26-1604/98
loop7       7:7      0   3.7M 1 loop /snap/gnome-system-monitor/127
loop8       7:8      0  93.8M 1 loop /snap/core/8935
loop9       7:9      0   3.7M 1 loop /snap/gnome-system-monitor/135
loop10      7:10     0  956K 1 loop /snap/gnome-logs/93
loop11      7:11     0   4.3M 1 loop /snap/gnome-calculator/704
loop12      7:12     0  348M 1 loop /snap/pycharm-community/188
loop14      7:14     0  14.8M 1 loop /snap/gnome-characters/399
loop15      7:15     0 339.1M 1 loop /snap/pycharm-community/192
loop16      7:16     0  54.7M 1 loop /snap/core18/1668
loop17      7:17     0   55M 1 loop /snap/core18/1705
loop18      7:18     0  81.5M 1 loop /snap/scrcpy/211
loop19      7:19     0 140.7M 1 loop /snap/gnome-3-26-1604/97
loop20      7:20     0  54.8M 1 loop /snap/gtk-common-themes/1502
loop21      7:21     0  956K 1 loop /snap/gnome-logs/81
loop22      7:22     0   9.1M 1 loop /snap/canonical-livepatch/95
loop23      7:23     0  62.1M 1 loop /snap/gtk-common-themes/1506
loop24      7:24     0  14.8M 1 loop /snap/gnome-characters/495
loop25      7:25     0 310.8M 1 loop
loop26      7:26     0  93.9M 1 loop /snap/core/9066
sda         8:0      0  7.3T 0 disk
├─sda1      8:1      0  7.3T 0 part /mnt/MondoSeagate
sdc         8:32     0  500G 0 disk
├─sdc1      8:33     0  606M 0 part
└─sdc2      8:34     0 498.5G 0 part
sdd         8:48     0    1T 0 disk
└─sdd1      8:49     0 1024G 0 part /mnt/VMSCloudDrive
sde         8:64     0   20G 0 disk
└─sde1      8:65     0   20G 0 part
nvme0n1    259:0    0  931.5G 0 disk
├─nvme0n1p1 259:1    0  931.3G 0 part /
└─nvme0n1p2 259:2    0   250M 0 part /boot/efi
```

At the end of the "lsblk" listing above, you can see that this system has five drives named "sda", "sdc", "sdd", "sde", and "nvme0n1". The "sdb" is missing because on this system "sdb" is an SD card reader without any media in it at the time the command was issued.

Note that some drives above have more than one partition and that partitions are listed as type "part".

Partitions can be mounted to any folder in the file system, but this can get confusing. For that reason, it is somewhat of a convention to mount to a folder in "/mnt" or "/media".

Since the entire file system falls under root (/), a single Linux file system can mount local drives and network drives and become quite extensive. In the above listing /dev/sdd1 is mounted in /mnt/VMSCloudDrive which is an iSCSI target on my QNAP NAS.

There is also an "lspci" command to list all devices on the PCI bus of your computer.

```
scott@mondo:~$ lspci
00:00.0 Host bridge: Intel Corporation 8th Gen Core Processor Host Bridge/DRAM Registers (rev 07)
00:01.0 PCI bridge: Intel Corporation Xeon E3-1200 v5/E3-1500 v5/6th Gen Core Processor PCIe Controller (x16) (rev 07)
00:02.0 Display controller: Intel Corporation Device 3e92
00:08.0 System peripheral: Intel Corporation Xeon E3-1200 v5/v6 / E3-1500 v5 / 6th/7th Gen Core Processor Gaussian Mixture Model
00:14.0 USB controller: Intel Corporation 200 Series/Z370 Chipset Family USB 3.0 xHCI Controller
00:16.0 Communication controller: Intel Corporation 200 Series PCH CSME HECI #1
00:17.0 SATA controller: Intel Corporation 200 Series PCH SATA controller [AHCI mode]
00:1c.0 PCI bridge: Intel Corporation 200 Series PCH PCI Express Root Port #2 (rev f0)
00:1c.2 PCI bridge: Intel Corporation 200 Series PCH PCI Express Root Port #3 (rev f0)
00:1c.3 PCI bridge: Intel Corporation 200 Series PCH PCI Express Root Port #4 (rev f0)
00:1c.4 PCI bridge: Intel Corporation 200 Series PCH PCI Express Root Port #5 (rev f0)
00:1d.0 PCI bridge: Intel Corporation 200 Series PCH PCI Express Root Port #9 (rev f0)
00:1e.0 Signal processing controller: Intel Corporation 200 Series/Z370 Chipset Family Serial IO UART Controller #0
00:1f.0 ISA bridge: Intel Corporation Z370 Chipset LPC/eSPI Controller
00:1f.2 Memory controller: Intel Corporation 200 Series/Z370 Chipset Family Power Management Controller
00:1f.3 Audio device: Intel Corporation 200 Series PCH HD Audio
00:1f.4 SMBus: Intel Corporation 200 Series/Z370 Chipset Family SMBus Controller
01:00.0 VGA compatible controller: NVIDIA Corporation GP107 [GeForce GTX 1050 Ti] (rev a1)
01:00.1 Audio device: NVIDIA Corporation GP107GL High Definition Audio Controller (rev a1)
02:00.0 USB controller: ASMedia Technology Inc. ASM1142 USB 3.1 Host Controller
03:00.0 Network controller: Qualcomm Atheros QCA9377 802.11ac Wireless Network Adapter (rev 31)
04:00.0 Ethernet controller: Qualcomm Atheros Killer E2400 Gigabit Ethernet Controller (rev 10)
70:00.0 Non-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller SM981/PM981
```

This listing lists devices, including VGA controllers, USB controllers, sound, Bluetooth, wireless, and Ethernet controllers. This is considered a basic listing. You could also "lspci -v" for a verbose listing or even "lspci -vv" or "lspci -vvv" depending on how detailed you want the hardware information to be. Try that in windows!

You can also list the devices on the USB bus with "lsusb".

```
scott@mondo:~$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 002: ID 05e3:0616 Genesys Logic, Inc. hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 006: ID 0cf3:e009 Atheros Communications, Inc.
Bus 001 Device 011: ID 0557:2212 ATEN International Co., Ltd
Bus 001 Device 007: ID 046d:c016 Logitech, Inc. Optical Wheel Mouse
Bus 001 Device 004: ID 0409:005a NEC Corp. HighSpeed Hub
Bus 001 Device 107: ID 0b0e:1017 GN Netcom
Bus 001 Device 102: ID 046d:08b2 Logitech, Inc. QuickCam Pro 4000
Bus 001 Device 100: ID 046d:0823 Logitech, Inc.
Bus 001 Device 098: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 001 Device 005: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 001 Device 003: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 001 Device 014: ID 0764:0501 Cyber Power System, Inc. CP1500 AVR UPS
Bus 001 Device 038: ID 046d:0a59 Logitech, Inc.
Bus 001 Device 015: ID 0c76:161f JMTEK, LLC.
Bus 001 Device 010: ID 05e3:0610 Genesys Logic, Inc. 4-port hub
Bus 001 Device 002: ID 058f:6366 Alcor Micro Corp. Multi Flash Reader
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Up to this point, we have learned about installing packages, updating the OS, working with the file system, and understanding the configuration.

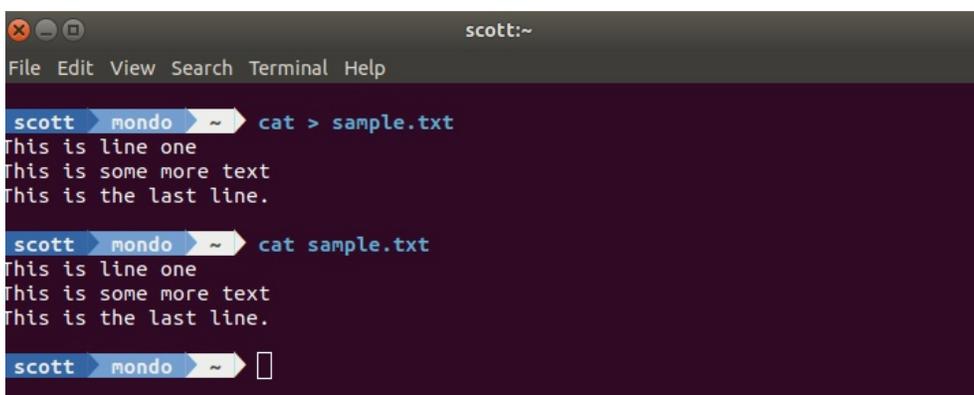
We have already seen the "ls" command to list the contents of a directory. The command "cat" is used to type a file:

```
cat test.txt | more
```

The "cd" command changes the default directory just like in windows. However, in Linux, folder names are separated by slashes (/) and not backslashes (\). That's because Linux is forward thinking and Windows is backwards.

"cd /" takes you to the root directory. "cd .." takes you up one level. "cd -" takes you to the previous directory. "pwd" shows your current directory location.

To create an empty file:

A terminal window titled "scott:~" with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is "scott@mondo ~". The user enters "cat > sample.txt" and types three lines of text: "This is line one", "This is some more text", and "This is the last line.". Then the user enters "cat sample.txt" and the same three lines of text are displayed. Finally, the user enters a blank command, and the prompt returns to "scott@mondo ~".

```
scott@mondo ~$ cat > sample.txt
This is line one
This is some more text
This is the last line.
scott@mondo ~$ cat sample.txt
This is line one
This is some more text
This is the last line.
scott@mondo ~$
```

```
touch empty.txt
```

To create a file with some text:

The first command above uses ">" to redirect keyboard input into the file. To end input, CTRL-D. A simple "cat" command lists the content of the file.

The "cp" command allows you to copy a file.

The "mv" command allows you to move a file from one place to another.

The "mv" command is also used to rename a file.

The "rm" command is used to remove (delete) a file.

The "mkdir" command is used to create a directory.

The "rmdir" command is used to remove a directory.

The "top" command is a system monitor.

The "man" command is a help command which is a little more verbose than the --help switch on a command:

```
man ls | more
```

For even more verbose help:

```
info ls
```

The "passwd" command lets you change your password.

There are also some command line shortcuts:

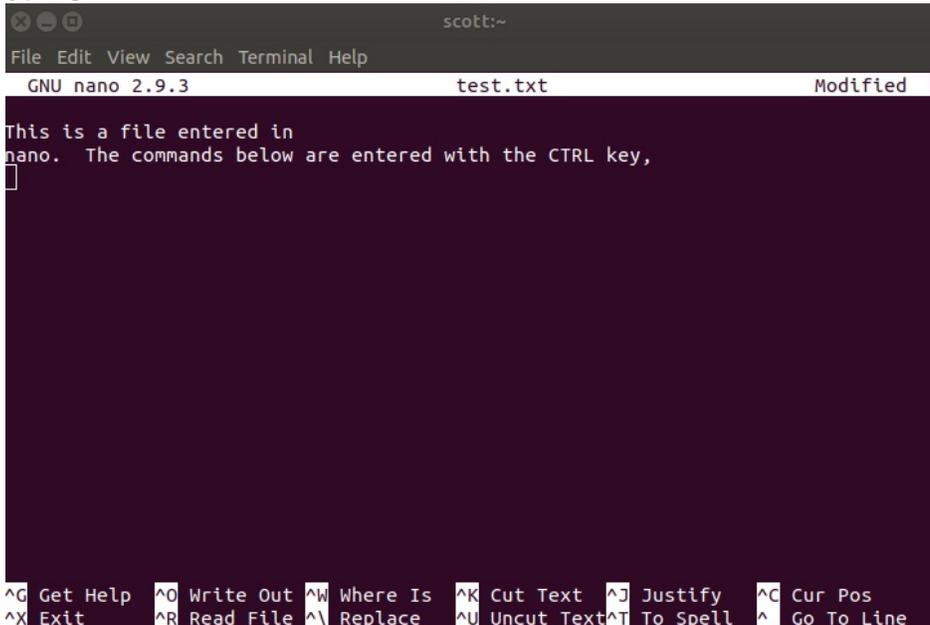
Ubuntu Terminal Shortcuts	Function
Ctrl + Shift + T	Open new tab on current terminal
Ctrl + Shift + W	Close the current tab
Ctrl + A	Move cursor to beginning of line
Ctrl + E	Move cursor to end of line
Ctrl + U	Clears the entire current line
Ctrl + K	Clears the command from the cursor right
Ctrl + W	Delete the word before the cursor
Ctrl + R	Allows you to search your history for commands matching what you have typed
Ctrl + C	Kill the current process
Ctrl + Z	Suspend the current process by sending the signal SIGSTOP
Ctrl + L	Clears the terminal output
Alt + F	Move forward one word
Alt + B	Move backward one word
Ctrl + Shift + C	Copy the highlighted command to the clipboard
Ctrl + Shift + V or Shift + Insert	Paste the contents of the clipboard
Up/Down Arrow keys	To scroll through your command history, allowing you to quickly execute the same command multiple times
TAB	Used to complete the command you are typing. If more than one command is possible, you can press it multiple times to scroll through the possible completions. If a very wide number of commands are possible, it can output a list of all possible completions.

Another valuable command to master is the "nano" text editor.

To edit a file:

```
nano test.txt
```

When you decide to exit (Ctrl+X), it will ask whether to save or discard the file. There is a lot of help on nano shortcuts, but it is a character oriented editor that understands cursor movement and typing.

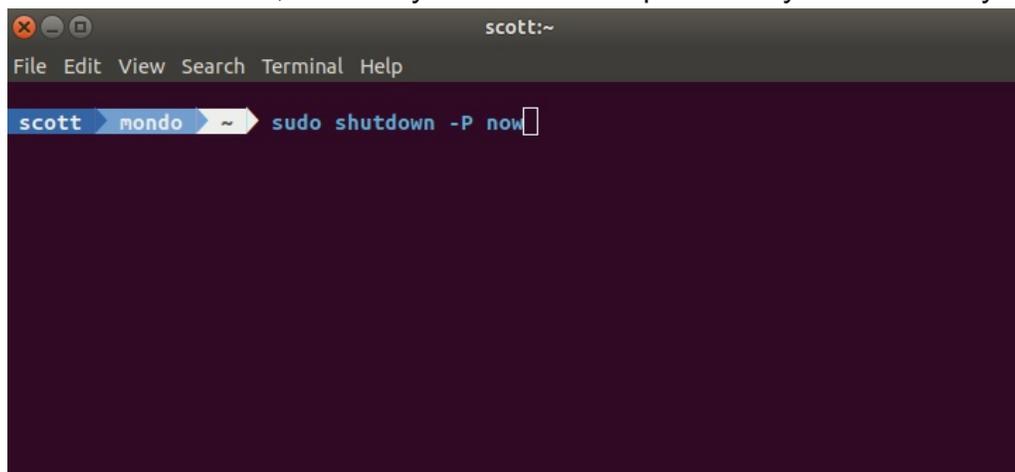


```
scott:~
File Edit View Search Terminal Help
GNU nano 2.9.3 test.txt Modified
This is a file entered in
nano. The commands below are entered with the CTRL key,
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

There are other more basic editors like "vi", but nano is easier to use. This document is about command line tools. There is a GUI text editor called "gedit". It is oriented towards GUI operations, but having a tool like "nano" is needed when you have only the command line available.

This document just covers the basics of being comfortable in the command line. Learning Linux means having control over all of the functions on your system rather than just functions that Microsoft wants you to have. That means that you will learn more and also make more mistakes.

One final command, to orderly shutdown and power off your Ubuntu system:



```
scott:~
File Edit View Search Terminal Help
scott@mondo: ~$ sudo shutdown -P now
```