Linux - for Self-Driving Vehicles

Introduction To Linux

What is Linux ?

- Just like Windows , and Mac OS,Linux is an Operating System.
- Developed in 1991 by linus Torvalds.
- Open Source!
- Extremely customizable.

Command Line Interface

What is CLI ?

The most frequent tasks that you perform on your PC is creating, moving or deleting Files.

File Management:

1. Terminal (Command Line Interface - CLI)
2. File manager (Graphical User Interface -GUI)

Using CLI, we could write commands to do the same as GUI.

Why learn Command Line Interface?

- Commands are flexible and offer more options.
- Work on multiple files at a time.
- CLI is Fast.

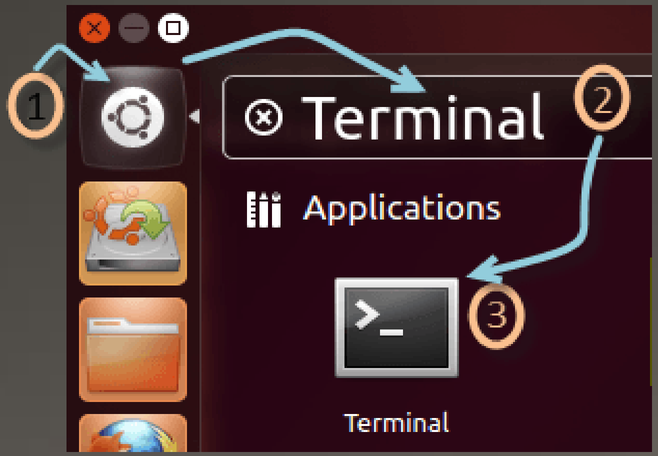
GUI is important too !!

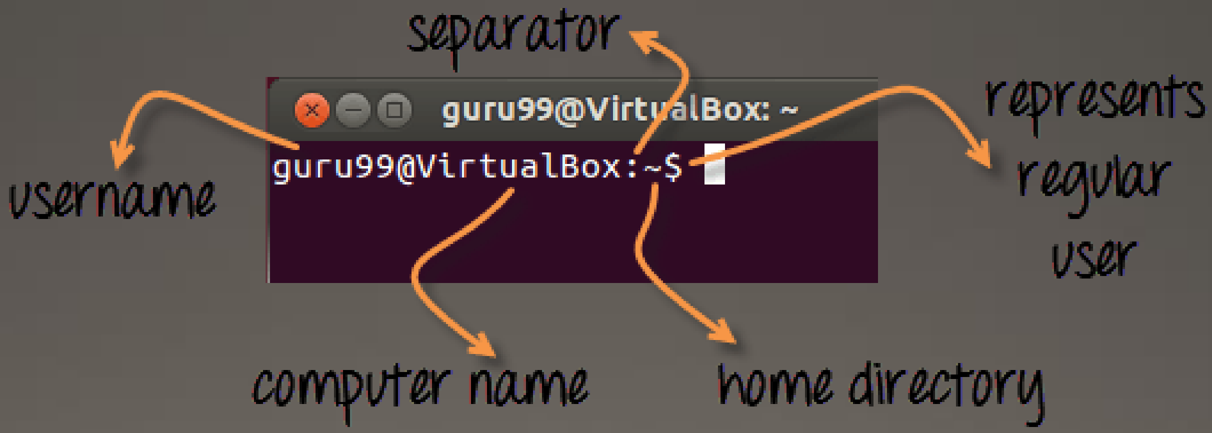
- Browsing
- Performance Graphs
- Editing Video and Images

Launching the CLI on Ubuntu

There are 2 ways to launch the terminal.

1) Go to the Dash and type terminal
2) Or you can press CTRL + Alt + T to launch the Terminal




```
guru99@VirtualBox: ~
guru99@VirtualBox:~$
```

Basic Commands

Present Working Directory

The directory that you are currently browsing is called the Present working directory.
If you want to determine the directory you are presently working on,
use the command -pwd

Relative & Absolute Paths

- ➢ Absolute path is the full path to reach file
  - Example: If you want to access Pictures directory:

    cd /home/user/Pictures
- ➢ Relative path allows you to browse another subdirectory
  - It saves you from effort to type complete paths all the time.
  - Example: If you are on home directory and want to access Pictures directory :

    cd Pictures or cd ./Pictures

Changing Directories

If you want to change your current directory use the 'cd' command.

Command	Description
cd or cd ~	Navigate to HOME directory
cd ..	Move one level up
cd	To change to a particular directory
cd /	Move to the root directory

Listing files (ls)

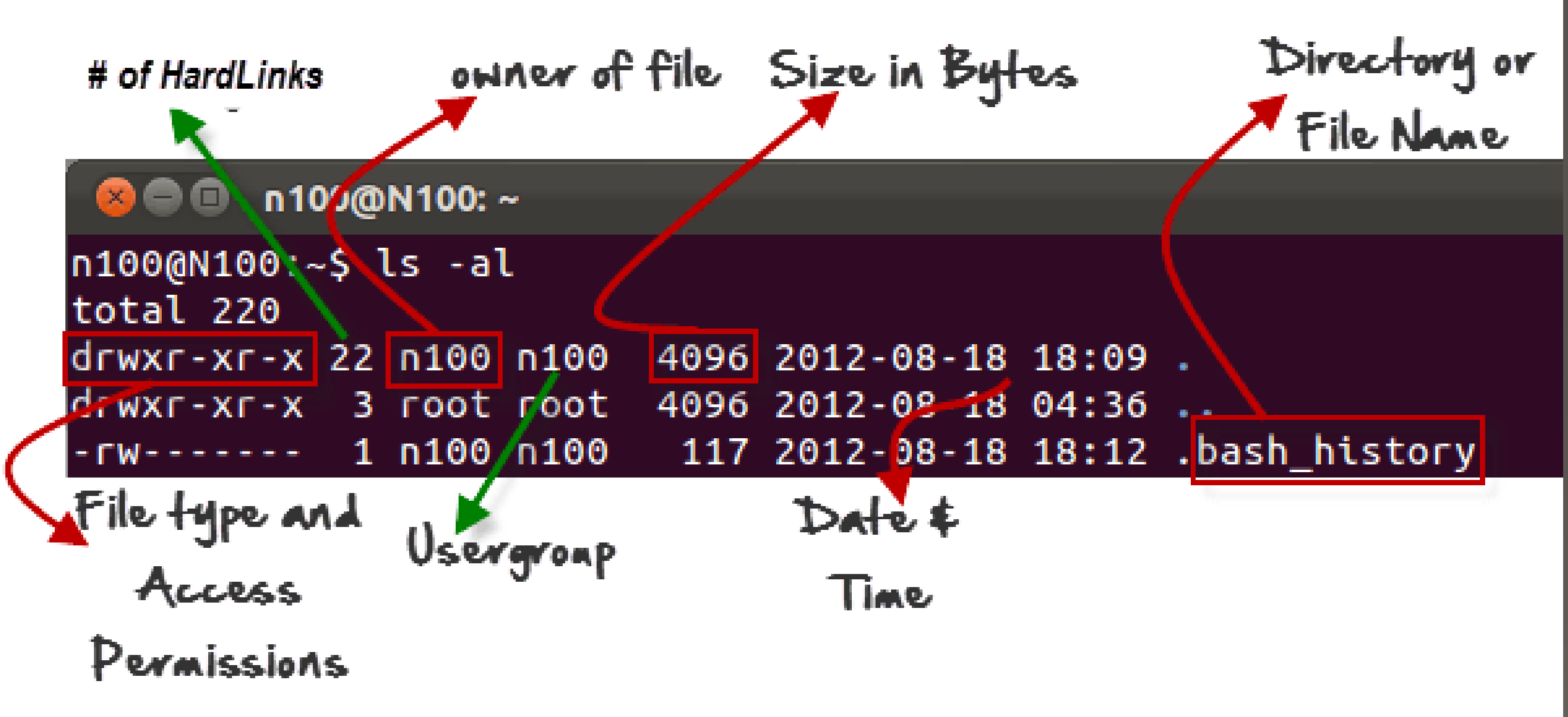
If you want to see the list of files on your UNIX or Linux system, use the 'ls' command.

It shows the files /directories in your current directory.

Command	Description
ls	Lists all files and directories in the present working directory
ls - R	Lists files in sub-directories as well
ls - a	Lists hidden files as well
ls - al	Lists files and directories with detailed information like permissions, size, owner, etc.

- ➢ Directories are denoted in blue color.
- ➢ Files are denoted in white.

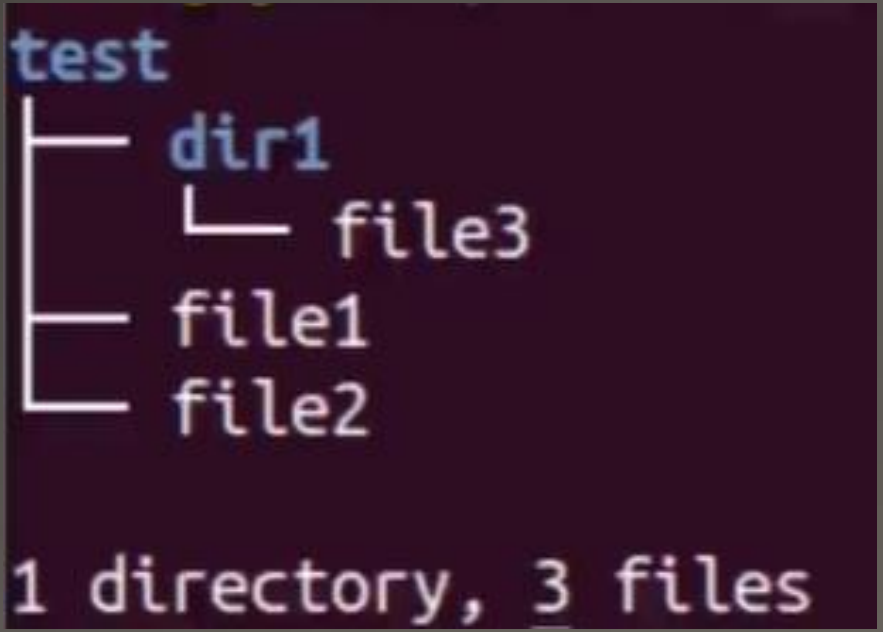
Files Detailed Information



```
n100@N100:~$ ls -al
total 220
drwxr-xr-x 22 n100 n100   4096 2012-08-18 18:09 .
drwxr-xr-x  3 root root   4096 2012-08-18 04:36 ..
-rw-------  1 n100 n100    117 2012-08-18 18:12 .bash_history
```

Directory Tree

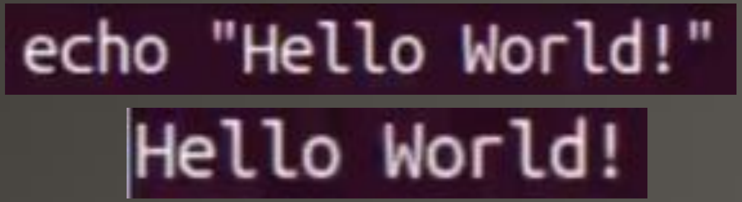
- (tree) List tree from current directory that includes all sub-directories and files.



```
test
├── dir1
│   └── file3
├── file1
└── file2

1 directory, 3 files
```

Echo text

- (echo) prints line of text “echoing”.

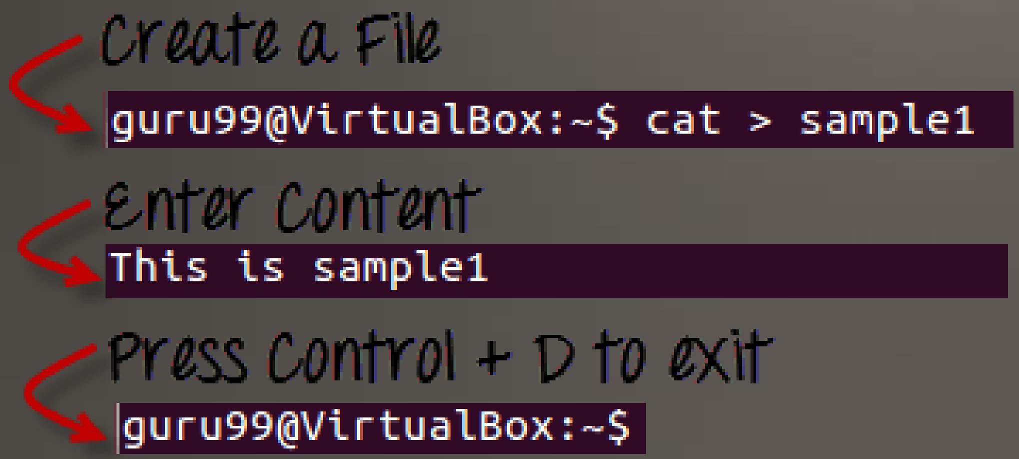


```
echo "Hello World!"
Hello World!
```

Creating & Viewing Files

- The 'cat' server command is used to display text files. It can also be used for copying, combining and creating new text files.

  1. cat > filename
  2. Add content
  3. Press 'ctrl + d' to return to command prompt.



Create a File

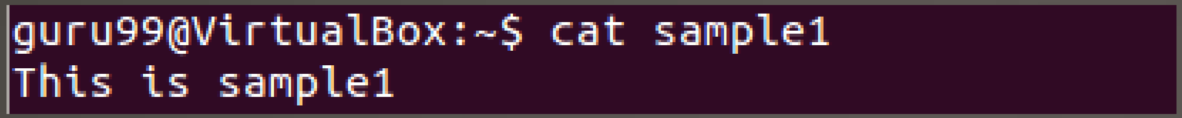
```
guru99@VirtualBox:~$ cat > sample1
```

Enter Content

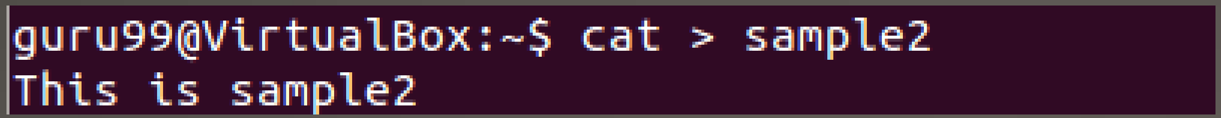
```
This is sample1
```

Press Control + D to exit

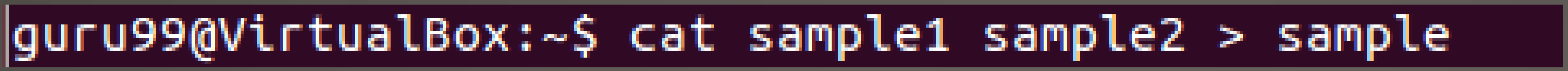
```
guru99@VirtualBox:~$
```



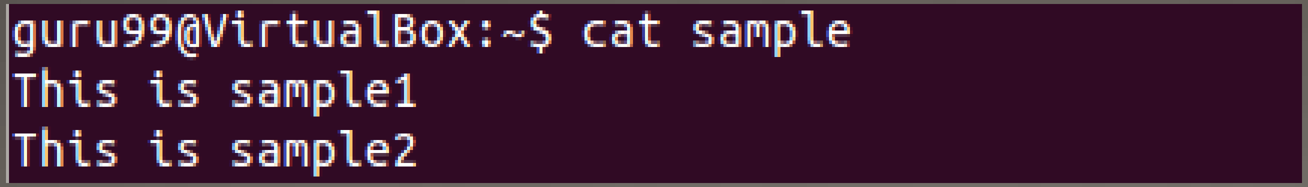
```
guru99@VirtualBox:~$ cat sample1
This is sample1
```



```
guru99@VirtualBox:~$ cat > sample2
This is sample2
```



```
guru99@VirtualBox:~$ cat sample1 sample2 > sample
```



```
guru99@VirtualBox:~$ cat sample
This is sample1
This is sample2
```

Copying Files

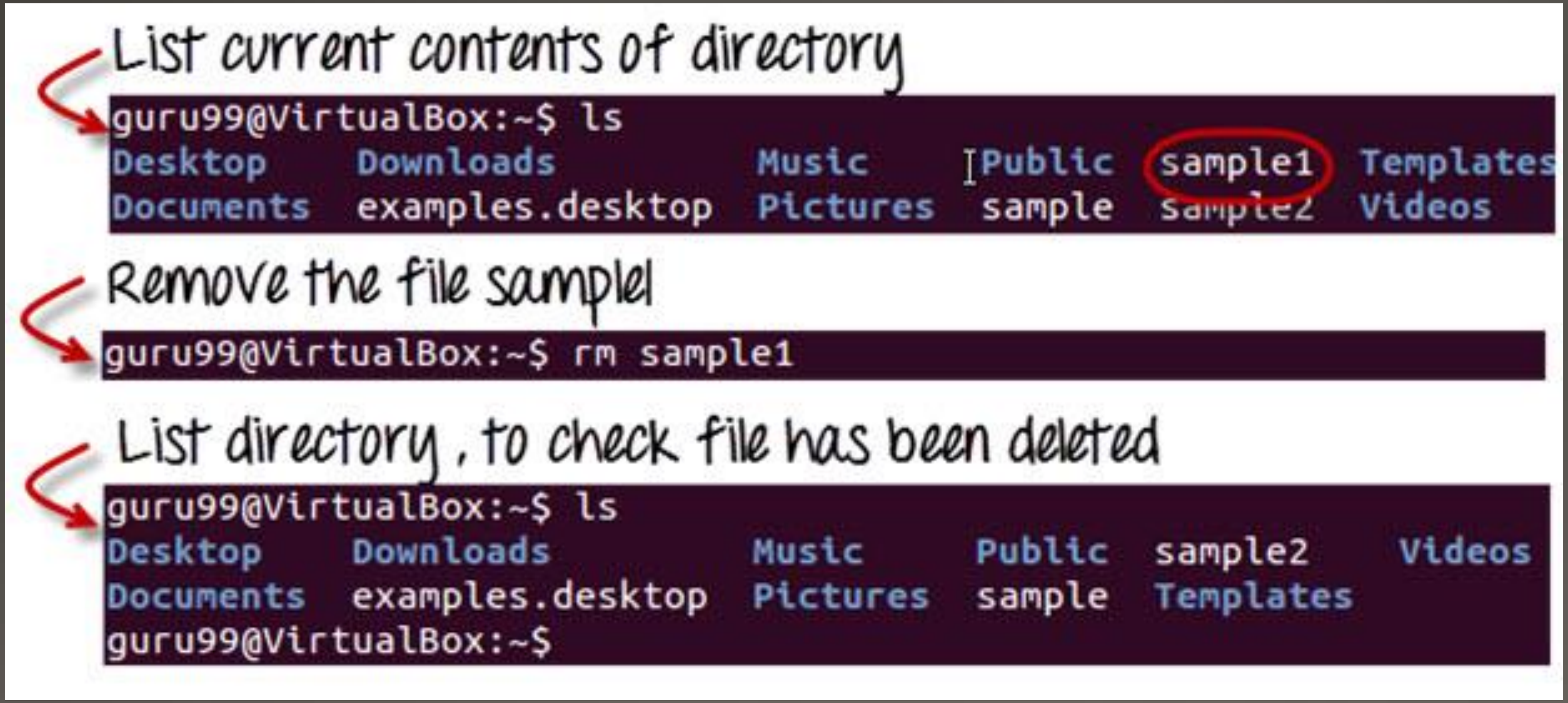
- (cp) copy files from source to destination.
- (cp <source> <destination>)
- (cp –r) is for copying directories recursively

Creating file

- (touch) Creates an empty file.

Deleting Files

- The 'rm' command removes files from the system without confirmation.



List current contents of directory

```
guru99@VirtualBox:~$ ls
Desktop      Downloads           Music      Public   sample1   Templates
Documents    examples.desktop    Pictures   sample   sample2   Videos
```

Remove the file sample1

```
guru99@VirtualBox:~$ rm sample1
```

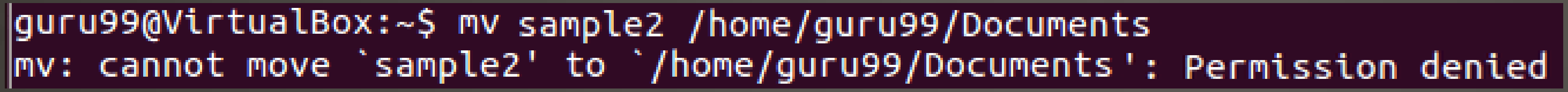
List directory , to check file has been deleted

```
guru99@VirtualBox:~$ ls
Desktop      Downloads           Music      Public   sample2     Videos
Documents    examples.desktop    Pictures   sample   Templates
guru99@VirtualBox:~$
```

Moving and Re-naming files

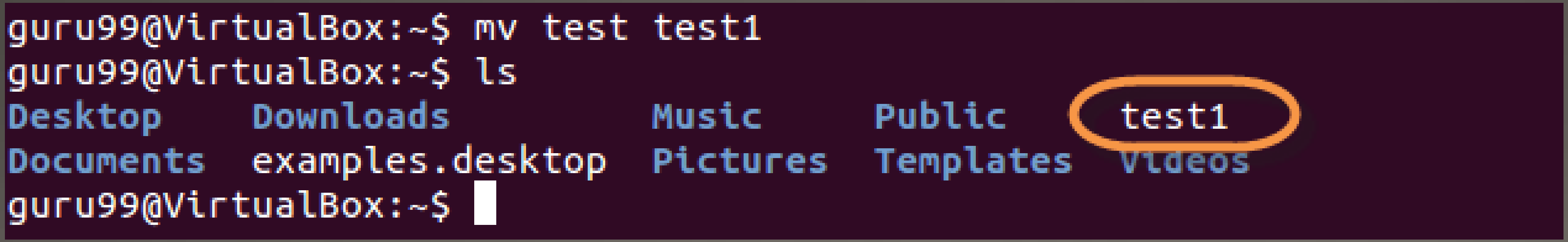
- To move a file, use the command "mv".

✗



```
guru99@VirtualBox:~$ mv sample2 /home/guru99/Documents
mv: cannot move `sample2' to `/home/guru99/Documents': Permission denied
```

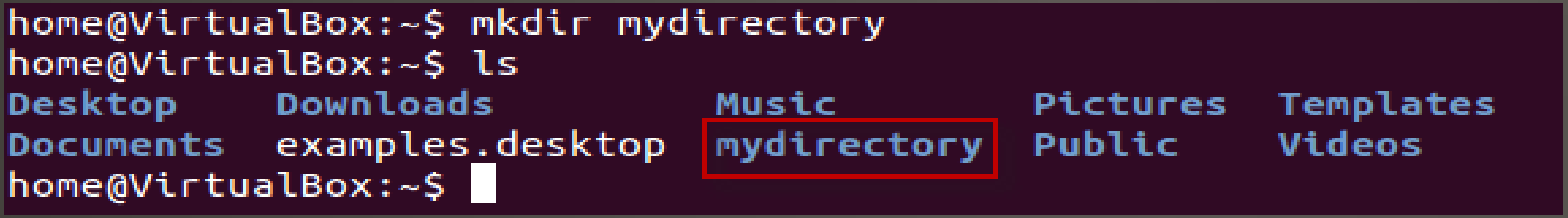
- sudo command_you_want_to_execute
- sudo mv sample2 /home/quru99/Documents
- For renaming file:



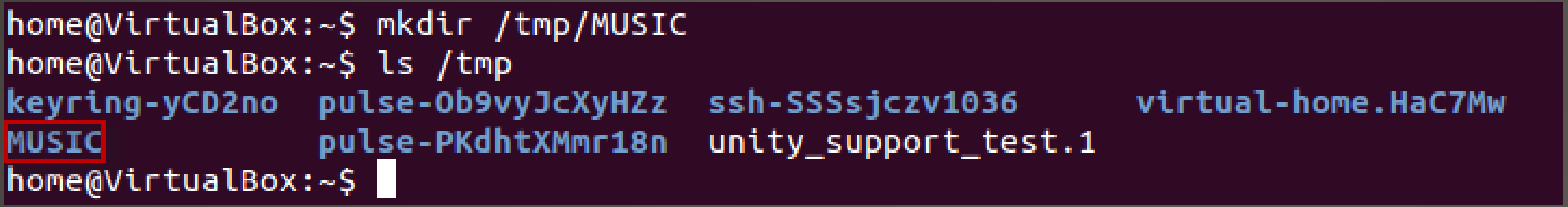
```
guru99@VirtualBox:~$ mv test test1
guru99@VirtualBox:~$ ls
Desktop     Downloads          Music     Public     test1
Documents   examples.desktop   Pictures  Templates  Videos
guru99@VirtualBox:~$ 
```

Directory Manipulations

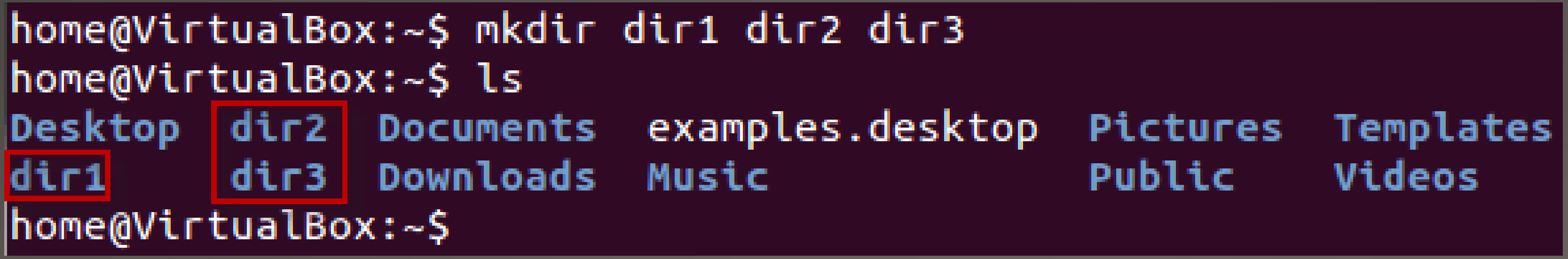
- Directories can be created on a Linux operating system using "mkdir" command.



```
home@VirtualBox:~$ mkdir mydirectory
home@VirtualBox:~$ ls
Desktop     Downloads          Music          Pictures   Templates
Documents   examples.desktop   mydirectory    Public     Videos
home@VirtualBox:~$ 
```



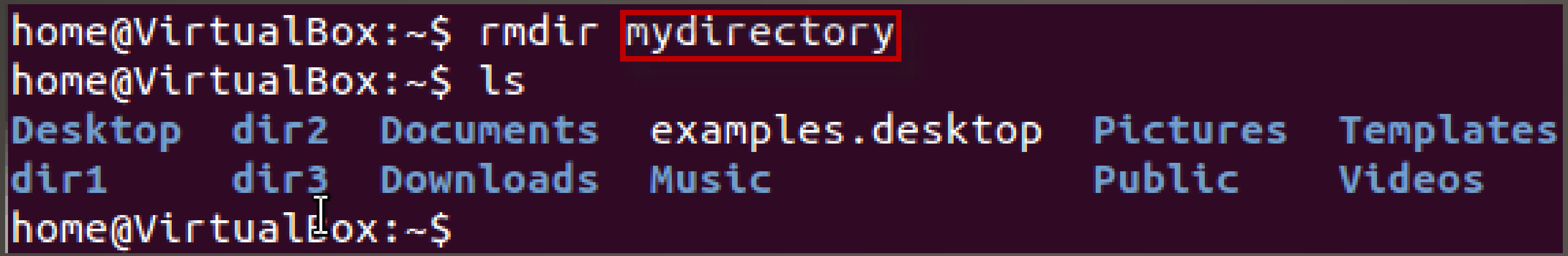
```
home@VirtualBox:~$ mkdir /tmp/MUSIC
home@VirtualBox:~$ ls /tmp
keyring-yCD2no   pulse-Ob9vyJcXyHZz   ssh-SSSsjczv1036       virtual-home.HaC7Mw
MUSIC            pulse-PKdhtXMmr18n   unity_support_test.1
home@VirtualBox:~$ 
```



```
home@VirtualBox:~$ mkdir dir1 dir2 dir3
home@VirtualBox:~$ ls
Desktop  dir2  Documents  examples.desktop  Pictures  Templates
dir1     dir3  Downloads  Music             Public    Videos
home@VirtualBox:~$
```

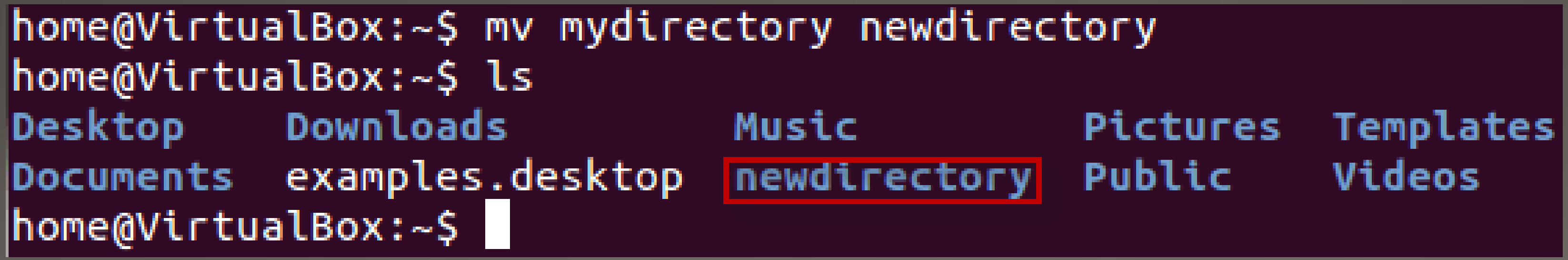
Removing Directories

- To remove a directory, use the command –rmdir | rm -r



```
home@VirtualBox:~$ rmdir mydirectory
home@VirtualBox:~$ ls
Desktop  dir2  Documents  examples.desktop  Pictures  Templates
dir1     dir3  Downloads  Music             Public    Videos
home@VirtualBox:~$
```

Renaming Directory



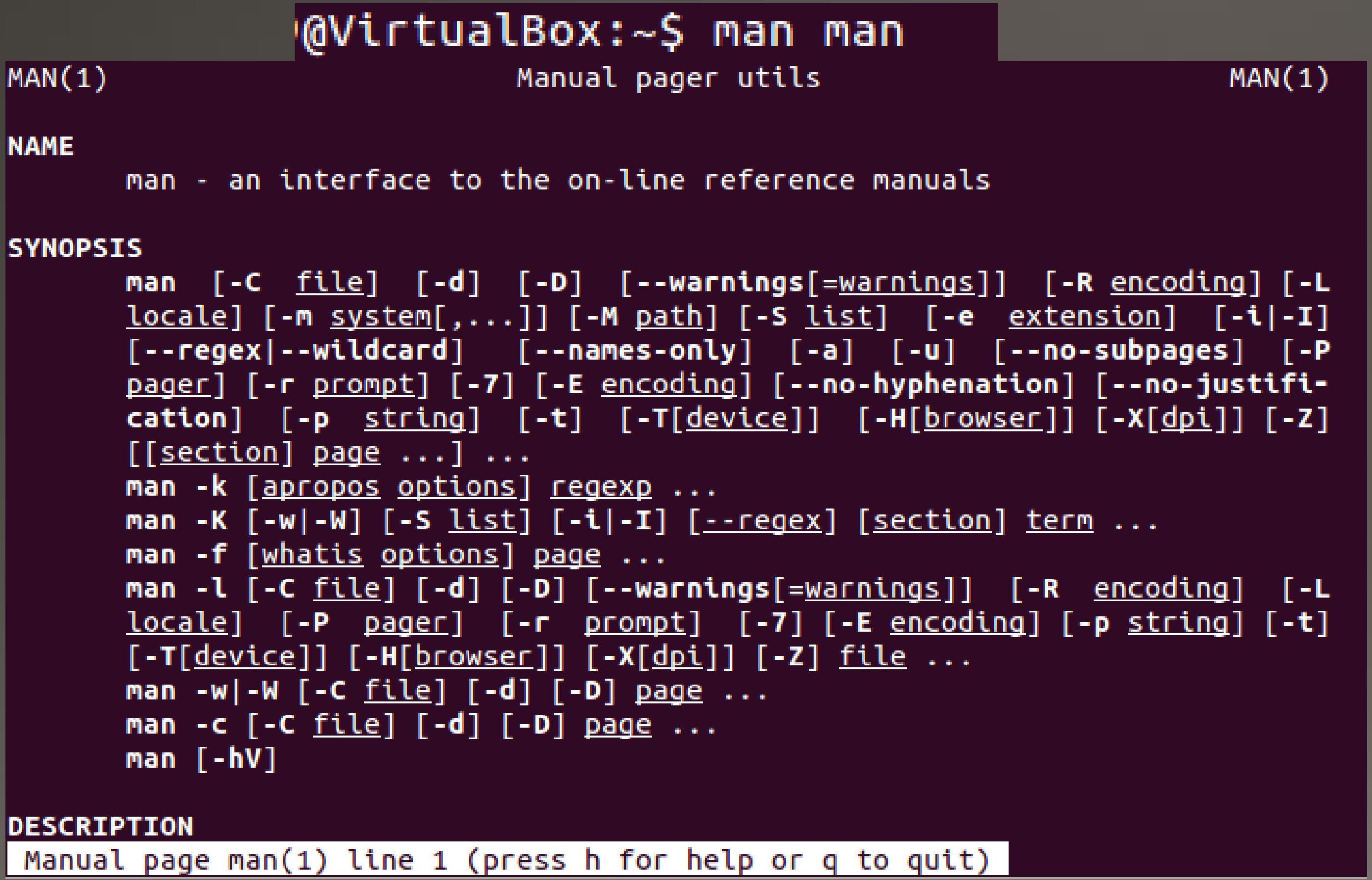
```
home@VirtualBox:~$ mv mydirectory newdirectory
home@VirtualBox:~$ ls
Desktop    Downloads         Music         Pictures  Templates
Documents  examples.desktop  newdirectory  Public    Videos
home@VirtualBox:~$
```

How to Make Parent Directories

- Building a structure with multiple subdirectories using mkdir requires adding the -p option. This makes sure that mkdir adds any missing parent directories in the process.
- For example:
  - if you want to create “dirtest2” in “dirtest1” inside the Linux directory (i.e., Linux/dirtest1/dirtest2), run the command:
  - mkdir –p Linux/dirtest1/dirtest2
  - Use ls -R to show the recursive directory tree.
  - Without the -p option ????????

The 'Man' command

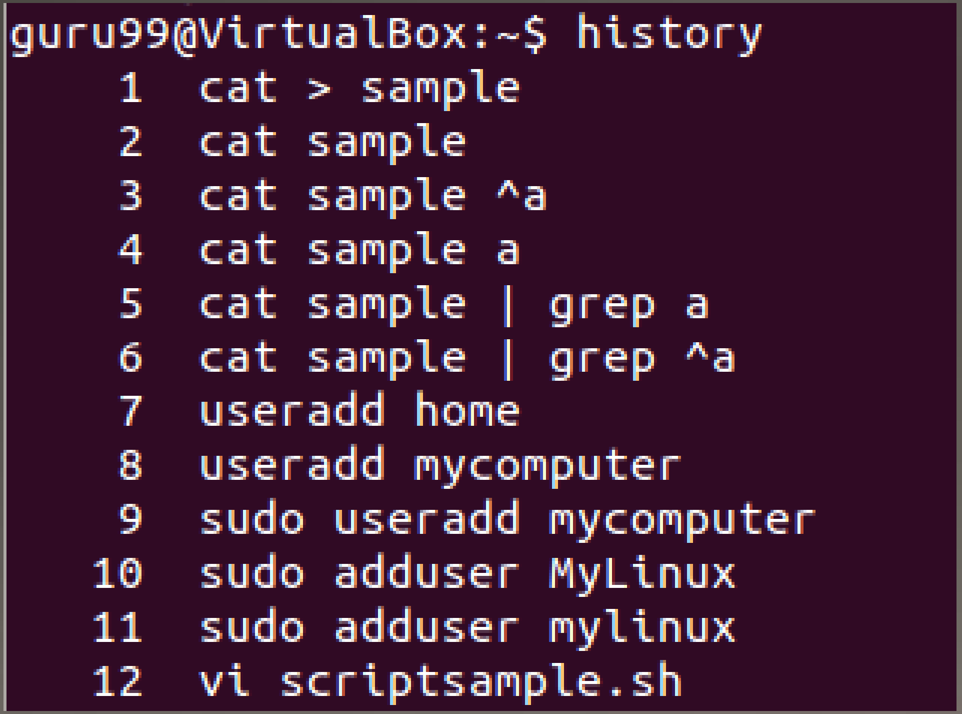
- Man stands for manual which is a reference book of a Linux operating system. It is similar to HELP file found in popular software.



```
@VirtualBox:~$ man man
MAN(1)                        Manual pager utils                        MAN(1)

NAME
       man - an interface to the on-line reference manuals

SYNOPSIS
       man  [-C  file]  [-d]  [-D]  [--warnings[=warnings]]  [-R encoding] [-L
       locale] [-m system[,...]] [-M path] [-S list]  [-e  extension]  [-i|-I]
       [--regex|--wildcard]  [--names-only]  [-a]  [-u]  [--no-subpages]  [-P
       pager] [-r prompt] [-7] [-E encoding] [--no-hyphenation] [--no-justifi-
       cation]  [-p  string]  [-t]  [-T[device]]  [-H[browser]] [-X[dpi]] [-Z]
       [[section] page ...] ...
       man -k [apropos options] regexp ...
       man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
       man -f [whatis options] page ...
       man -l [-C file] [-d] [-D] [--warnings[=warnings]]  [-R  encoding]  [-L
       locale]  [-P  pager]  [-r  prompt]  [-7] [-E encoding] [-p string] [-t]
       [-T[device]] [-H[browser]] [-X[dpi]] [-Z] file ...
       man -w|-W [-C file] [-d] [-D] page ...
       man -c [-C file] [-d] [-D] page ...
       man [-hV]

DESCRIPTION
 Manual page man(1) line 1 (press h for help or q to quit)
```

The History Command

- History command shows all the basic commands in Linux that you have used in the past for the current terminal session.
- This can help you refer to the old commands you have entered and re-used them in your operations again.



```
guru99@VirtualBox:~$ history
    1  cat > sample
    2  cat sample
    3  cat sample ^a
    4  cat sample a
    5  cat sample | grep a
    6  cat sample | grep ^a
    7  useradd home
    8  useradd mycomputer
    9  sudo useradd mycomputer
   10  sudo adduser MyLinux
   11  sudo adduser mylinux
   12  vi scriptsample.sh
```

The clear command

- This command clears all the clutter on the terminal and gives you a clean window to work on, just like when you launch the terminal.

Editing Files

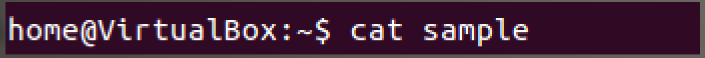
- The most common file editors are (**nano**) ,(**vim**)and (**gedit**).
- (**nano**) is simpler to use , but (vim) is more powerful .
- Example : **nano** file.txt
- To exit (**nano**) use (ctrl + x) then (y) to save file.

Pipe, Grep and Sort Command in Linux

What is a Pipe in Linux?

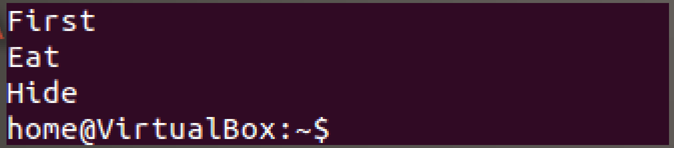
- The Pipe is a command in Linux that lets you use two or more commands such that output of one command serves as input to the next.
- When you use 'cat' command to view a file which spans multiple pages, the prompt quickly jumps to the last page of the file, and you do not see the content in the middle.
- To avoid this, you can pipe the output of the 'cat' command to 'less' which will show you only one scroll length of content at a time.

while using 'cat' command



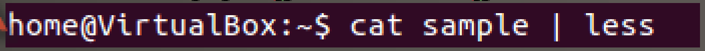
```
home@VirtualBox:~$ cat sample
```

The screen zooms to the end of the file



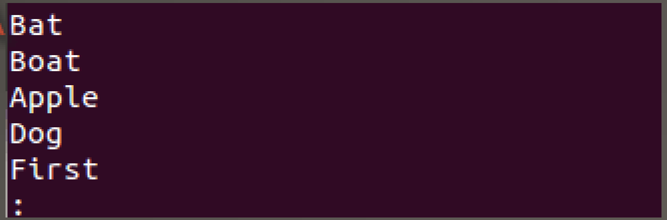
```
First
Eat
Hide
home@VirtualBox:~$
```

But with piping and using 'less' command



```
home@VirtualBox:~$ cat sample | less
```

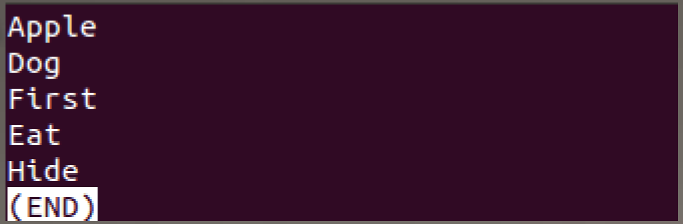
You can scroll file content using the arrow keys or PageUp/PageDown as you read



```
Bat
Boat
Apple
Dog
First
:
```

Once you reach end of file,

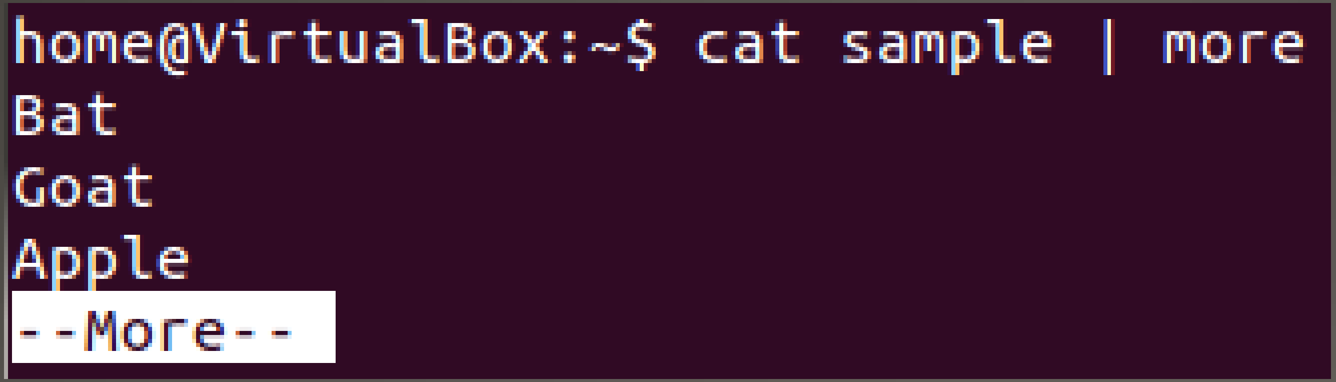
Press q to exit



```
Apple
Dog
First
Eat
Hide
(END)
```

'pg' and 'more' commands

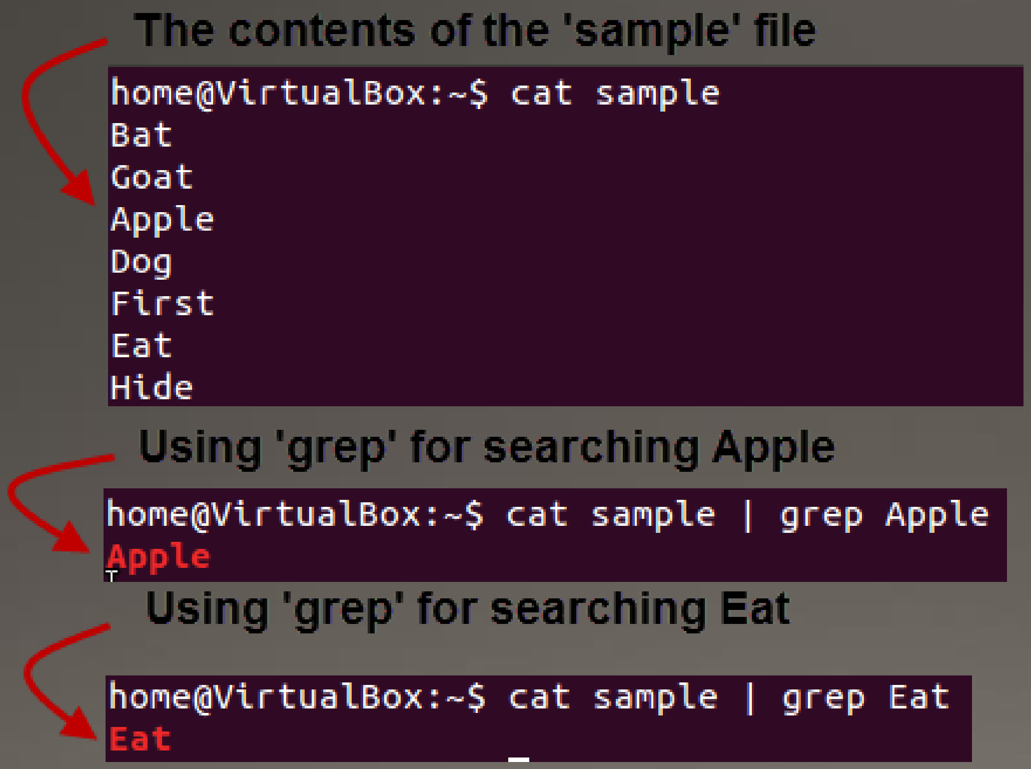
- Instead of 'less’,
- you can also use. cat Filename | pg or cat Filename | more



```
home@VirtualBox:~$ cat sample | more
Bat
Goat
Apple
--More--
```

The 'grep' command

- Suppose you want to search a particular information the postal code from a text file.



The contents of the 'sample' file

```
home@VirtualBox:~$ cat sample
Bat
Goat
Apple
Dog
First
Eat
Hide
```

Using 'grep' for searching Apple

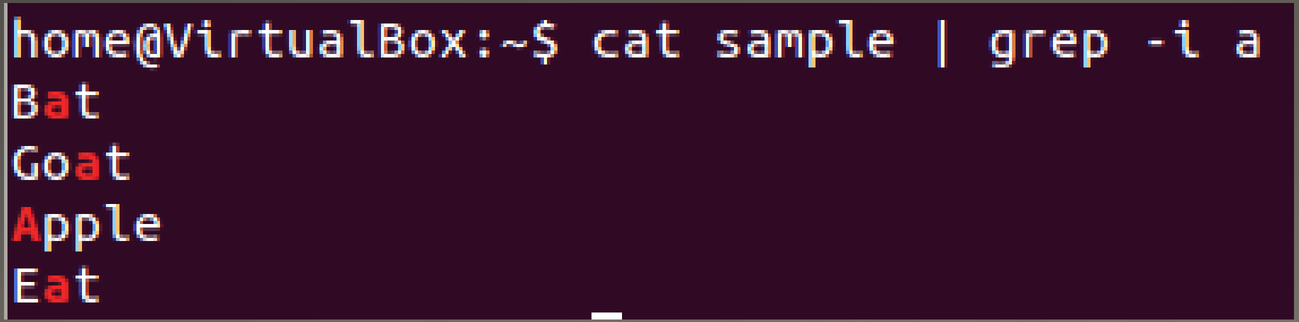
```
home@VirtualBox:~$ cat sample | grep Apple
Apple
```

Using 'grep' for searching Eat

```
home@VirtualBox:~$ cat sample | grep Eat
Eat
```

The 'grep' command

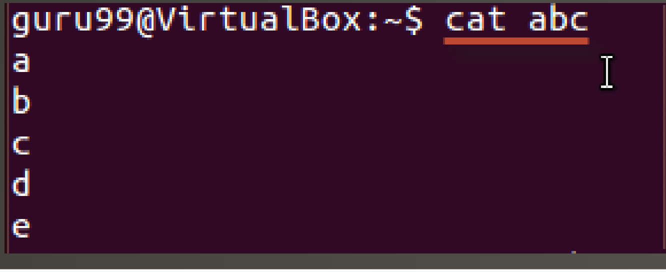
Option	Function
-v	Shows all the lines that do not match the searched string
-c	Displays only the count of matching lines
-n	Shows the matching line and its number
-i	Match both (upper and lower) case
-l	Shows just the name of the file with the string
-R	will follow all symbolic links



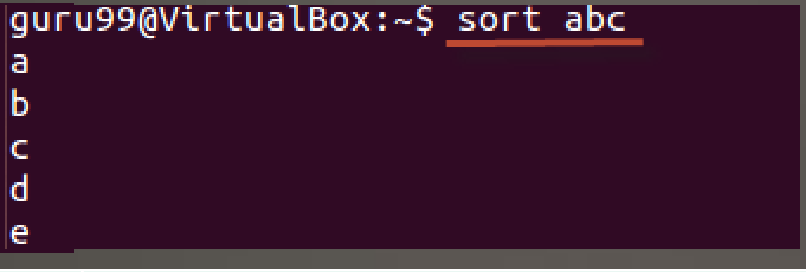
```
home@VirtualBox:~$ cat sample | grep -i a
Bat
Goat
Apple
Eat
```

The 'sort' command

This command helps in **sorting out the contents of a file alphabetically.**

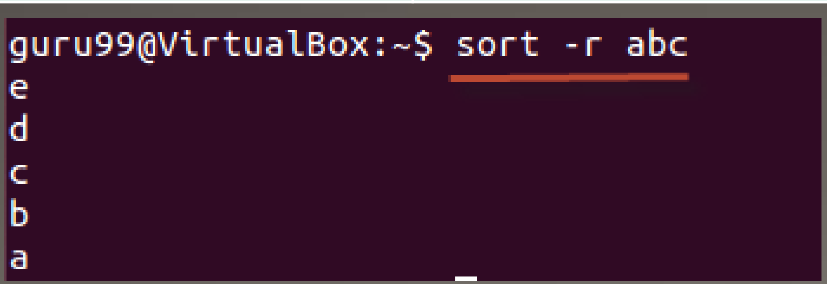


```
guru99@VirtualBox:~$ cat abc
a
b
c
d
e
```



```
guru99@VirtualBox:~$ sort abc
a
b
c
d
e
```

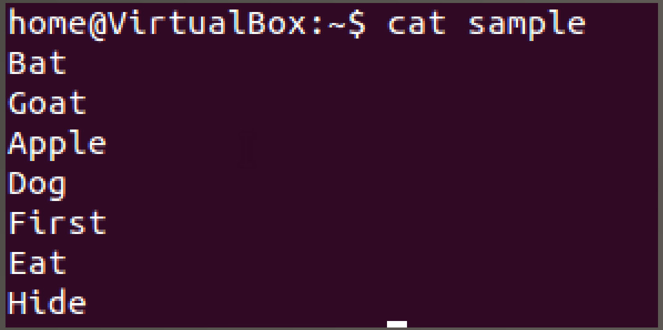
Option	Function
-r	Reverses sorting
-n	Sorts numerically
-f	Case insensitive sorting



```
guru99@VirtualBox:~$ sort -r abc
e
d
c
b
a
```

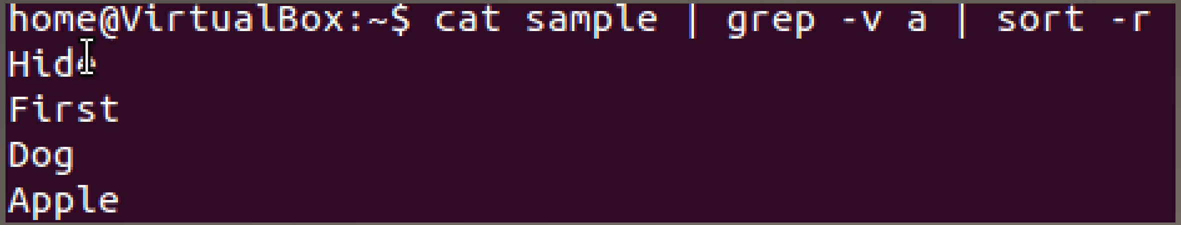
Filtering file

- When you pipe two commands, the "filtered " output of the first command is given to the next.



```
home@VirtualBox:~$ cat sample
Bat
Goat
Apple
Dog
First
Eat
Hide
```

- **We want to highlight** only the lines that do not contain the character 'a', but the result should be in reverse order.



```
home@VirtualBox:~$ cat sample | grep -v a | sort -r
Hide
First
Dog
Apple
```

Installing Software

- In debian-based distributions , software is packaged and get distributed in “deb” format.
- (apt) command is used to download these packages ,then it uses (apkg) command to install it on your system.
- First , we need to update the caches of software packages .
    - **sudo apt update**
- Then ,we cloud install the desired package .Example: to install (tree)
    - **sudo apt install tree**

Processes

- Any running program or command in Linux is called a process.
- Each running process has a process ID (PID)
- Processes could be started , stopped , run in background and in foreground.
- Background process runs independently of the user, which means no interaction with the user.
- We could Monitor resources of running processes like **cpu** and memory used using different commands like **(top),(htop),(ps)**

Processes

Command	Description
bg	To send a process to the background
fg	To run a stopped process in the foreground
top	Details on all Active Processes
ps	Give the status of processes running for a user
ps PID	Gives the status of a particular process
pidof	Gives the Process ID (PID) of a process
kill PID	Kills a process
nice	Starts a process with a given priority
renice	Changes priority of an already running process
df	Gives free hard disk space on your system
free	Gives free RAM on your system

Coding in Linux

GNU Toolchain

- Collection of programing tool produced by GNU project.
- GNU project is a mass collaboration project that give control to freely use , view , and modify software .
- These tools are used for developing applications and operating systems.
- GNU Toolchain include:
  - (**gcc**) compiler for C projects
  - (**g++**) compiler for C++ projects
  - (**make**) Automation tool for compilation and build
  - (**binutils**) a suite of tools including linker , assembler and other tools
  - (**gdb**) code debugging tool

Development Editing and IDEs

- Popular IDEs or code editors used on Linux for C and C++ are :
  - Eclipse
  - Vscode
  - Sublime
  - Atom
  - Geany
  - Vim
- Python , Java , Nodejs , and many more languages could be developed on Linux too

Networking

Networking Commands

- (**ifconfig**) command prints all interfacing along with it’s configurations.
- (**ping**) command used to analyze connection to other node and track performance.
  - Example : ping google.com
  - It uses ICMP protocol.
- (**telnet**) connects to remote host shell.
- (**ssh**) or secure shell command is used to connect to remote host securely.
  - Both telent and ssh are commonly used to connect to embedded Linux target.

Soft links

- A soft (Symbolic) link is a type of file in Linux that points to another file or folder on your computer.
- Soft links are similar to shortcuts in windows .
- To create a soft link use (**ln –s**) command.
  - **$ ln –s <path to the file/folder to be linked> <link path>**
- Soft link files are of file type (l) .

Important Commands

Shutdown and Reboot the System

### shutdown

- it can be used to shutdown a system or restart it. It is commonly used to shutdown or reboot both local and remote machines.

```
shutdown [OPTION] [TIME] [MESSAGE]
```

- # shutdown -h now

The h option is for halt which means to stop. The second parameter is the time parameter. "now" means that shutdown the system right away.

Important Commands

Shutdown and Reboot the System

reboot

- This will perform a graceful shutdown and restart of the machine. This is what happens when you click restart from your menu.

```
# reboot
```

Important Commands

hostname Command

hostname

- The Linux hostname command is used to view or change a system’s domain and hostname. It can also check a computer’s IP address.

```
# hostname [options] [new_hostname]
```

- Use the [options] parameter to add more specific instructions to the hostname command. Without it, the default output shows your computer’s hostname: Use the [new_hostname] parameter when you want to change your computer’s hostname.

Important Commands

hostname Command

hostname

- The Linux hostname command is used to view or change a system’s domain and hostname. It can also check a computer’s IP address.

```
# hostname [options] [new_hostname]
```

- Use the [options] parameter to add more specific instructions to the hostname command. Without it, the default output shows your computer’s hostname: Use the [new_hostname] parameter when you want to change your computer’s hostname.

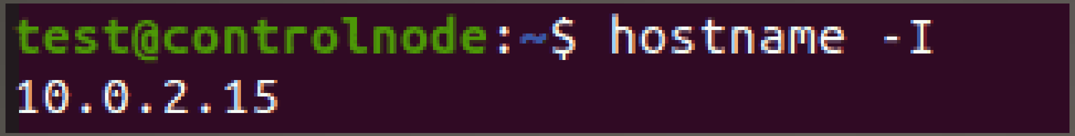
Important Commands

hostname Command

Display All Network Addresses

- Use the -I or --all-ip-addresses option to display all of the host’s network addresses. Unlike -i, this option doesn’t depend on hostname resolution:

#hostname -I
#hostname --all-ip-addresses



```
test@controlnode:~$ hostname -I
10.0.2.15
```

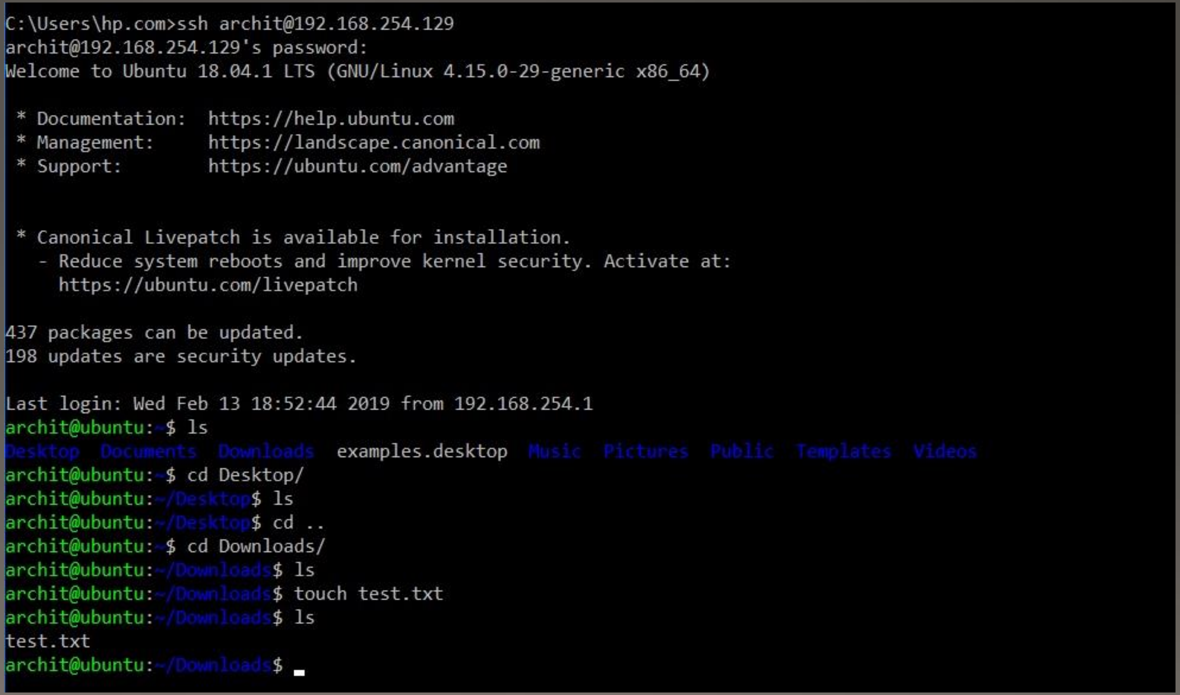
Important Commands

ssh command

### ssh

- ssh stands for "Secure Shell". It is a protocol used to securely connect to a remote server/system.

# ssh user_name@host(IP/Domain_name)



```
C:\Users\hp.com>ssh archit@192.168.254.129
archit@192.168.254.129's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

437 packages can be updated.
198 updates are security updates.

Last login: Wed Feb 13 18:52:44 2019 from 192.168.254.1
archit@ubuntu:~$ ls
Desktop  Documents  Downloads  examples.desktop  Music  Pictures  Public  Templates  Videos
archit@ubuntu:~$ cd Desktop/
archit@ubuntu:~/Desktop$ ls
archit@ubuntu:~/Desktop$ cd ..
archit@ubuntu:~$ cd Downloads/
archit@ubuntu:~/Downloads$ ls
archit@ubuntu:~/Downloads$ touch test.txt
archit@ubuntu:~/Downloads$ ls
test.txt
archit@ubuntu:~/Downloads$ _
```

Important Commands

chmod command

chmod

- In Unix-like operating systems, the chmod command is used to change the access mode of a file.
- The name is an abbreviation of change mode.

```
# chmod [reference][operator][mode] file...
```

Reference	Class	Description
u	owner	file's owner
g	group	users who are members of the file's group
o	others	users who are neither the file's owner nor members of the file's group
a	all	All three of the above, same as ugo

Important Commands

chmod command

chmod

- ➢The operator is used to specify how the modes of a file should be adjusted.

Operator	Description
+	Adds the specified modes to the specified classes
-	Removes the specified modes from the specified classes
=	The modes specified are to be made the exact modes for the specified classes

Important Commands

chmod command

chmod

- The modes indicate which permissions are to be granted or removed from the specified classes.

Mode	Description
r	Permission to read the file.
w	Permission to write (or delete) the file.
x	Permission to execute the file, or, in the case of a directory, search it.

Important Commands

chmod command

Absolute form

- The other way to use the chmod command is the absolute form, in which you specify a set of three numbers that together determine all the access classes and types.

Permission	Number
Read (r)	4
Write (w)	2
Execute (x)	1

Important Commands

chmod command

Absolute form

- Add the numbers of the permissions you want to give; for example:For file myfile, to grant read, write, and execute permissions to yourself (4+2+1=7), read and execute permissions to users in your group (4+0+1=5), and only execute permission to others (0+0+1=1), you would use:

```
#chmod 751 myfile
```

Important Commands

chmod command

Absolute form

Permission	Number
Read by owner	400
Write by owner	200
Execute by owner	100
Read by group	040
Write by group	020
Execute by group	010
Read by others	004
Write by others	002
Execute by others	001
anyone can do anything (read, write, or execute)	777

Important Commands

exit command

exit

- exit command in Linux is used to exit the shell where it is currently running. It takes one more parameter as [N] and exits the shell with a return of status N.

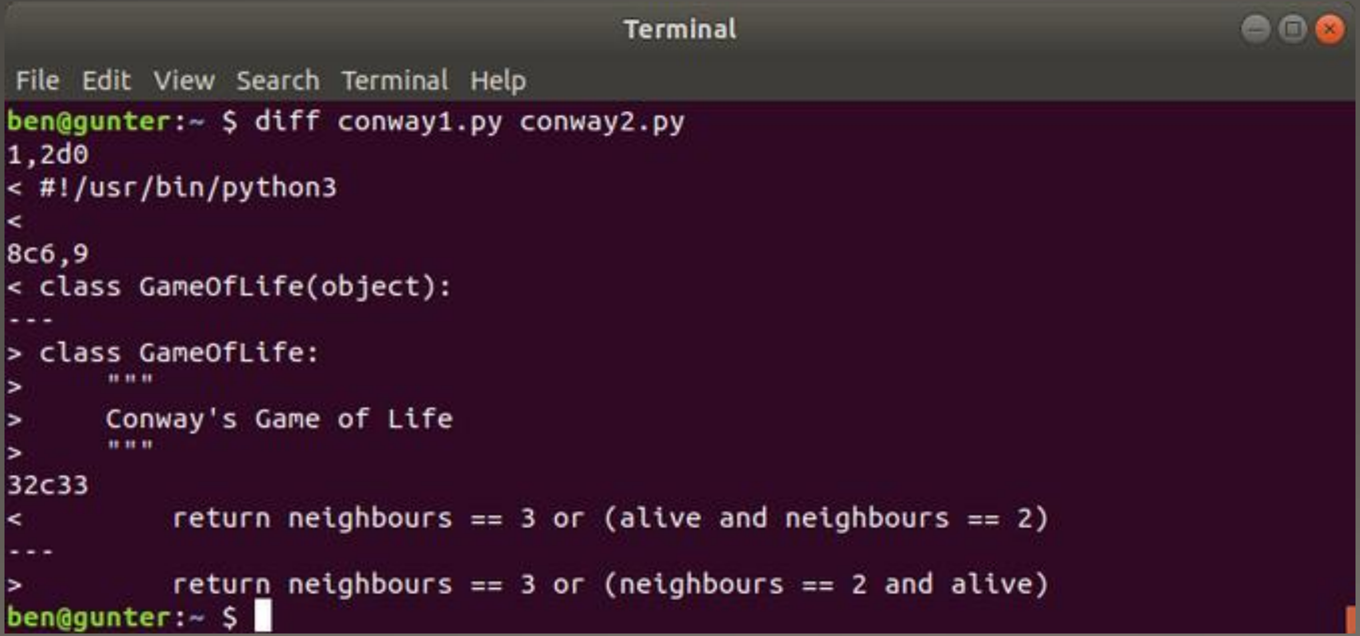
```
#exit [n]
```

Important Commands

Meld vs. the diff command

diff

- If you have two similar files (perhaps one is a modified version of the other) and want to see the changes between them, you could run the diff command to see their differences in the terminal:



```
ben@gunter:~ $ diff conway1.py conway2.py
1,2d0
< #!/usr/bin/python3
<
8c6,9
< class GameOfLife(object):
---
> class GameOfLife:
>     """
>     Conway's Game of Life
>     """
32c33
<         return neighbours == 3 or (alive and neighbours == 2)
---
>         return neighbours == 3 or (neighbours == 2 and alive)
ben@gunter:~ $ 
```

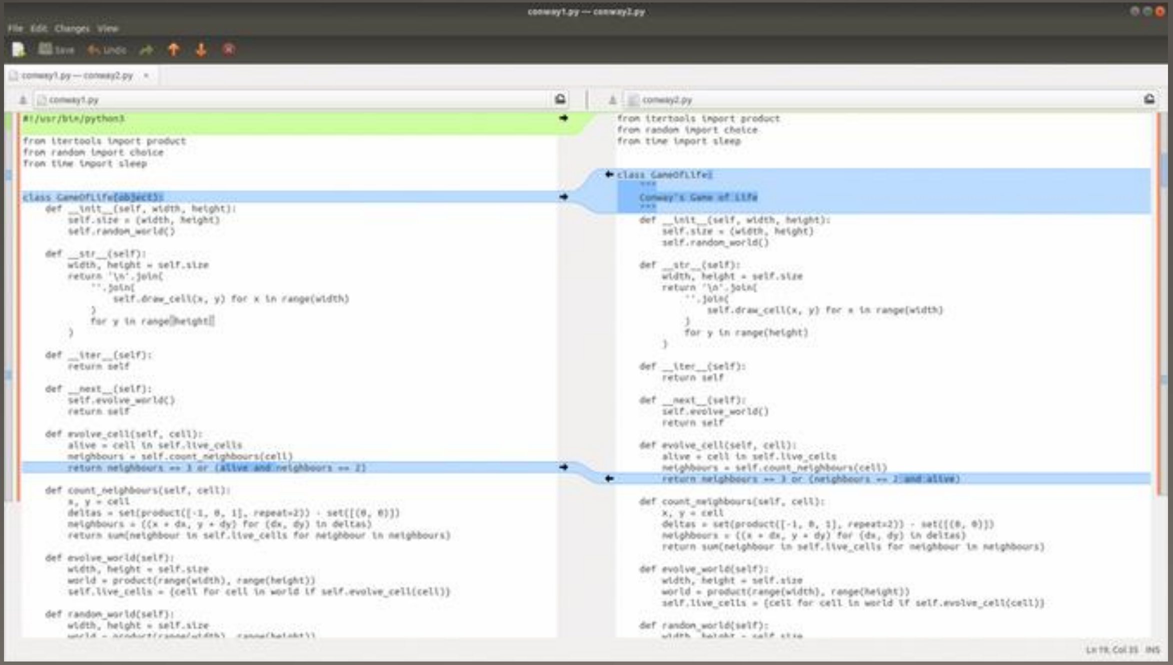
Important Commands

Meld vs. the diff command

meld

- Here's the same example using the meld command. You can run the same comparison from the command line with:

```
$ meld file1.py file2.py
```



conway1.py:

```
#!/usr/bin/python3

from itertools import product
from random import choice
from time import sleep


class GameOfLife(object):
    def __init__(self, width, height):
        self.size = (width, height)
        self.random_world()

    def __str__(self):
        width, height = self.size
        return '\n'.join(
            ''.join(
                self.draw_cell(x, y) for x in range(width)
            )
            for y in range(height)
        )

    def __iter__(self):
        return self

    def __next__(self):
        self.evolve_world()
        return self

    def evolve_cell(self, cell):
        alive = cell in self.live_cells
        neighbours = self.count_neighbours(cell)
        return neighbours == 3 or (alive and neighbours == 2)

    def count_neighbours(self, cell):
        x, y = cell
        deltas = set(product([-1, 0, 1], repeat=2)) - set([(0, 0)])
        neighbours = ((x + dx, y + dy) for (dx, dy) in deltas)
        return sum(neighbour in self.live_cells for neighbour in neighbours)

    def evolve_world(self):
        width, height = self.size
        world = product(range(width), range(height))
        self.live_cells = {cell for cell in world if self.evolve_cell(cell)}

    def random_world(self):
        width, height = self.size
        world = product(range(width), range(height))
```

conway2.py:

```
from itertools import product
from random import choice
from time import sleep


class GameOfLife:
    """
    Conway's Game of Life
    """
    def __init__(self, width, height):
        self.size = (width, height)
        self.random_world()

    def __str__(self):
        width, height = self.size
        return '\n'.join(
            ''.join(
                self.draw_cell(x, y) for x in range(width)
            )
            for y in range(height)
        )

    def __iter__(self):
        return self

    def __next__(self):
        self.evolve_world()
        return self

    def evolve_cell(self, cell):
        alive = cell in self.live_cells
        neighbours = self.count_neighbours(cell)
        return neighbours == 3 or (neighbours == 2 and alive)

    def count_neighbours(self, cell):
        x, y = cell
        deltas = set(product([-1, 0, 1], repeat=2)) - set([(0, 0)])
        neighbours = ((x + dx, y + dy) for (dx, dy) in deltas)
        return sum(neighbour in self.live_cells for neighbour in neighbours)

    def evolve_world(self):
        width, height = self.size
        world = product(range(width), range(height))
        self.live_cells = {cell for cell in world if self.evolve_cell(cell)}

    def random_world(self):
        width, height = self.size
```

Important Commands

Using Command-Line SFTP

SFTP

- SFTP is Secure File Transfer Protocol. It is similar to ssh, but its primary purpose is to enable file transfers between a local machine and a remote machine, whereas the ssh (Secure Shell) protocol is for opening up a general command shell on a remote machine where you have an account.
- The login format for command line sftp is exactly the same as with the ssh command, but with the "sftp" command:

```
#sftp username@host_name
```

Important Commands

Check hardware information on Linux with hwinfo command

Hwinfo

- The hwinfo command is a very handy command line tool that can be used to check details about hardware components. It reports information about most hardware units including cpu, hdd controllers, usb controllers, network card, graphics cards, multimedia, printers etc.

```
# sudo apt-get install hwinfo
```

Important Commands

Display all information

Hwinfo

- Running hwinfo without any options would display detailed information about all hardware units

```
# hwinfo
```

Display brief information

- The "--short" option will display brief information about the hardware and not the details.

```
# hwinfo --short

# sudo hwinfo --short --usb

#sudo hwinfo --short --usb --cpu --block
```

nmap

Find Devices Connected to Your Network with nmap

- Open the Ubuntu command line
- Install the network scanning tool nmap
  - sudo apt install nmap
  - nmap 10.42.0.*

nm-connection-editor

network management framework

udevadm

udevadm expects a command and command specific options. It controls the runtime behavior of udev, requests kernel events, manages the event queue, and provides simple debugging mechanisms.

Synopsis

```
udevadm [--debug] [--version] [--help]

udevadm info options
udevadm trigger [options]
udevadm settle [options]
udevadm control command
udevadm monitor [options]
udevadm test [options] devpath
```

Look at this site: https://man7.org/linux/man-pages/man8/udevadm.8.html

courses to be finished

https://app.pluralsight.com/library/courses/getting-started-linux/table-of-contents

https://app.pluralsight.com/library/courses/getting-started-linux-command-line/table-of-contents

https://www.udacity.com/course/linux-command-line-basics--ud595

Thank You