Linux

apropos -you can quickly figure out which commands lead to the actions you want to perform
eg- apropos ifconfig
output
ifconfig (8) - configure a network interface
ifdata (1) - get network interface info without parsing ifconfig ou...

apropos "description"
By typing the above, you'll get a list of all commands that match the "description" string with said command's help string. So if I were to type:

apropos "list directory"
output
dir (1) - list directory contents
ls (1) - list directory contents
ntfsls (8) - list directory contents on an NTFS filesystem
vdir (1) - list directory contents

History

The history command will list all of the commands you entered since the terminal launched along with an identifying number next to each command. You can repeat any of the listed commands by typing:
!#
where # is the number listed to the command you want to repeat. It's much more convenient than mashing the **Up** key a million times to find that one command that needs repeating.

Similarly, you can type **!!** to repeat the last entered command.

Run Commands At a Specific Time

Run set of command in future ,the below command open a prompt to enter the commands to be exceuted.

at 8:30 AM 03/21/15

cItrl+D to quit the prompt .

top

display Linux processes

The top program provides a dynamic real-time view of a running system. It can display system summary information as well as a list of processes or threads currently being managed by the Linux kernel.

htop

sudo apt-get install htop

get a full overview of all the processes running on your system along with details like process IDs, CPU and RAM usage, and how long they've been running.
htop is ease to use compared to top and cursor movement top,down,left,right and Other features, like sorting, make it easier to find what you need, and the color-coded text makes it all easier to read at a glance.

ranger

sudo apt-get install ranger

type ranger in the command line and your terminal will transform into an interface that makes it easy to navigate your entire filesystem using just a keyboard.

What is ppa?

On Ubuntu, the software on your system is managed by something called a package manager. The package manager maintains a list of repositories, which are source locations for package downloads. Every Linux distro comes with a core set of repositories.
But what if you want to install an application that doesn't exist in the core repositories? You have to find a repository that does have it, then manually add that repository to your package manager. That's where **personal package archives (PPAs)** come in handy.

sudo add-apt-repository <PPA repository>

Package Management

Redhat and fedora ---> RPM package management tool

Debian,ubuntu,Mint--->APT
Arch linux --->Pacman

Repositories

Repositories organise the tons of software in linux.Repositories are archive of software that run on your computer.

In Ubuntu

repositories are classified into 4 categories – (categorise based on different levels of support)
1.Main-->contains software that is officially supported
2. Restricted-->has software that is supported but is not available under a free (doesn't refer to cost, but the license) license
3. Universe -->contains software that is not officially supported but is maintained by the Linux Community
4.Multiverse -->houses software that is not free.

Reference site:http://www.makeuseof.com

whereis-->whereis - locate the binary, source, and manual page files for a command

systemd(A modern Service Manager for Linux)

is a software suite for central management and configuration of the **Linux** operating system. It consists of server applications (daemons), run time libraries, development tools, and command line utilities.

Systemd is a collection of system management daemons, utilities and libraries which serves as a replacement of **System V init** daemon. Systemd functions as central management and configuration platform for UNIX like system.

The init daemon is going to be replaced with daemon systemd on some of the Linux Distributions, while a lot of them have already implemented it.

What is init?

In Linux, init is a abbreviation for Initialization. The init is a daemon process which starts as soon as the computer starts and continue running till, it is shutdown. In-fact init is the first process that starts when a computer boots, making it the parent of all other running processes directly or indirectly and hence typically it is assigned "pid=1".

Kernal Panic -->If somehow **init** daemon could not start, no process will be started and the system will reach a stage called "**Kernel Panic**".

System V init -->init is most commonly referred to as System V init

System V -->is the first commercial UNIX Operating System designed and usages of init on most of the Linux Distribution of today is identical with System V OS with a few exception like Slackware using BSD-style and Gentoo using custom init.

Some of the init replacement

1. **Upstart –** A init replacement daemon implemented in Ubuntu GNU/Linux and designed to start process asynchronously.
2. **Epoch –** A init replacement daemon built around simplicity and service management, designed to start process single-threaded.
3. **Mudar –** A init replacement daemon written in Python, implemented on Pardus GNU/Linux and designed to start process asynchronously.
4. **systemd –** A init replacement daemon designed to start process in parallel, implemented in a number of standard distribution – Fedora, OpenSuSE, Arch, RHEL, CentOS, etc.

Systemd

A systemd is a System Management Daemon named with UNIX convention to add 'd' at the end of daemon.So that they can be easily recognized.Similar to init systemd is the parent of all other processes directly or indirectly and is the first process that starts at the boot hence typicallyt assigned a "pid=1" .

Why there was a need to replace init?

A init process starts serially ie one task starts only after the last startup was successful and it was loaded in the memory.This often resulted into delayed and long booting time.

Some of the features of systemd

Simpler boot process
Concurrent and parallel processing at boot
Better API
Simple unit syntax

Shutdown Computer at Specific Time

sudo shutdown 21:00

This will tell your computer to shut down at the specific time you have provided. You can also tell the system to shutdown after specific amount of minutes:

sudo shutdown +15

Search for Files

find /home/user -type f
The above command will search for all files in the directory /home/user.

find . -type f -size 10M
The above command will search from current directory for all files that are larger than 10MB.

find /home/user/files/ -type f -exec chmod 644 {} \;
The above command will will find all file in the specified directory and change the permission of all files.

find . -name tecmint.txt
The above command will find all the files whose name is tecmint.txt in current working dir.

find . -iname tecmint.txt
find all files whose name is tecmint.txt that contains capital and small letters.

find / -type d -name Techmint
the above command will find all directories whose name is Tecmint.

find . -type f -name "*.php"
the above command will find all files ending with php

find . -type f -perm 0777 -print
the above command will find all files with 777 permission.

How to Run or Repeat a Linux Command Every X Seconds Forever

The shortest period which you can run cron command is every 1 minute. Believe it or not, in many cases this is too slow.

1. Use watch Command

Watch is a Linux command that allows you to execute a command or program periodically and also shows you output on the screen. This means that you will be able to see the program output in time. By default watch re-runs the command/program every 2 seconds. The interval can be easily changed to meet your requirements.

watch free -m

The above command will check your system free memory and update the results of the free command every two seconds.

watch -n 10 script.sh
above command will execute script.sh every 10 seconds

watch -t free -m
above command will hide the top left heading of the watch command

uptime - Tell how long the system has been running.

uptime gives a one line display of the following information. The current time, how long the system has been running, how many users are currently logged on, and the system load averages for the past 1, 5,

and 15 minutes.

watch uptime

du - estimate file space usage
Summarize disk usage of each FILE, recursively for directories.

Monitor Progress of Copy Command
In Linux, while copying files from one location to other using **cp** command, the progress of data is not shown, to see the progress of data being copied, you can use the **watch** command along with **du -s** command to check the disk usage in real time.

```
# cp ubuntu-15.10-desktop-amd64.iso /home/tecmint/ &
# watch -n 0.1 du -s /home/tecmint/ubuntu-15.10-desktop-amd64.iso
```

sleep - delay for a specified amount of time.
Pause for NUMBER seconds. SUFFIX may be 's' for seconds (the default), 'm' for minutes, 'h' for hours or 'd' for days. Unlike most implementations that require NUMBER be an integer, here NUMBER may be an arbitrary floating point number. Given two or more arguments, pause for the amount of time specified by the sum of their values.

for loop Example

```
for i in {1..10}; do echo -n "This is a test in loop $i "; date ; sleep 5; done
```

while loop Example

```
while true; do echo -n "This is a test of while loop";date ; sleep 5; done
```

How to Find and Kill Running Processes in Linux

What is a Process in Linux?

A process on a Linux system can be a running occurrence of an application or program. You can also refer to processes as tasks executing in the operating system.

When a process is running, it keeps on shifting from one state to another and a process can in one of the following states:

1. Running: meaning the process is either executing or it is just set to be executed.
2. Waiting: meaning that the process is waiting for an event or for a system resource to carry out a task.

There are two types of waiting process under Linux namely interruptible and uninterruptible.

A waiting process that can be interrupted by signals is called Interruptible, while a waiting process that is directly waiting on hardware conditions and cannot be interrupted under any conditions is called uninterruptible.

1. Stopped: meaning that the process has been stopped, using a signal.

2. Zombie: meaning the process has been stopped abruptly and is dead.

When killing processes, the kill command is used to send a named signal to a named process or groups of processes. The default signal is the TERM signal.

Remember that the kill command can be a built-in function in many modern shells or external located at/bin/kill.

How to Find Process PID in Linux

In Linux every process on a system has a PID (Process Identification Number) which can be used to kill the process.

You can identify the PID of any process by using the **pidof** command as follows:

pidof firefox

How to Kill Processes in Linux

pidof firefox

kill 9378

You can also send a named signal to the process by using the signal name or numbers as follows:

kill -STGTERM 9541

Using the signal number to kill a process:

pidof firefox

kill -9 9647

In the above example, the number `9` is the signal number for the SIGKILL signal.

How to Kill Multiple Process PID's in Linux

kill -9 9756 9867 9876

A Guide to Kill, Pkill and Killall Commands to Terminate a Process in Linux

For a kill command a Signal Name could be:

Signal Name	Signal Value	Behaviour
SIGHUP	1	Hangup

SIGKILL	9	Kill Signal
SIGTERM	15	Terminate

Clearly from the behaviour above **SIGTERM** is the default and safest way to kill a process. **SIGHUP** is less secure way of killing a process as **SIGTERM**. **SIGKILL** is the most unsafe way among the above three, to kill a process which terminates a process without saving.

In order to kill a process, we need to know the **Process ID** of a process. A Process is an instance of a program. Every-time a program starts, automatically an unique **PID** is generated for that process. Every Process in Linux, have a pid. The first process that starts when Linux System is booted is – **init process**, hence it is assigned a value of '**1**' in most of the cases.

Init is the master process and can not be killed this way, which insures that the master process don't gets killed accidentally. **Init** decides and allows itself to be killed, where kill is merely a request for a shutdown.

To know all the processes and correspondingly their assigned pid, run.

ps -A

Different ways to find pid of a process

pidof firefox

ps aux |grep firefox

ps -ef|grep firefox

pgrep firefox

How about killing a process using process name

You must be aware of process name, before killing and entering a wrong process name may screw you.

pkill mysqld

What if a process have too many instances and a number of child processes, we have a command ‘killall‘. This is the only command of this family, which takes process name as argument in-place of process number.

To kill all mysql instances along with child processes, use the command as follow.
killall mysqld