

Linux Command Line

Ubuntu

Richard Condit

Freegeek, Chicago
6 November 2016

① Background

History

Benefits

② Basics

Exploring directories and files

③ Advanced use of commands

Some search power

Redirection

The system

Batch commands

Installation

Once the only way to work with computer



The old way and the modern way

- Terminal
- GUI

The old way and the modern way

- Terminal
- GUI = Graphical User Interface

- More efficient for repetitive tasks
- Software used only via commands
- Emergency back-door after crash

But...

Fast and powerful ... but...

It is more difficult than graphical interfaces, since you have to know the commands. So use help:

<http://freegeekchicago.org/CommandLineClass>

<http://richardcondit.org/workshops/Linux/CheatSheet.pdf>

<http://richardcondit.org/workshops/Linux/LinePresentation.pdf>

and always Google

pwd Show current directory (cwd is the same)

ls List of files and directories (wildcard *, or with folder name)

mkdir *newdir* Create a new folder; also **rmdir** to erase

cd Change directory

nano *file* edits a file (with helpful listing of necessary commands)

rm *file* Erase (ie remove) a file

cp Copying: name existing file and a new file

mv Move (ie rename): name existing file and a new file

cat Show contents of a file

- Up arrow repeats last command
- Ctrl<r> to reverse-search commands (great tool)

Help pages: *man* or *--help*

- Best to save a cheatsheet; you don't have to remember:
<http://richardcondit.org/workshops/CommandLine/cheatsheet.pdf>
- Remember Google to find examples or tutorials!

- 1 Type a file listing a few basic commands you want to remember
- 2 Type another file listing basic nano commands
- 3 Create two new directories and copy one file into each
- 4 Test ls, cat, head commands
- 5 Test the reverse search

`grep` *item* Powerful search: finds any text matching *item*

| The vertical bar pipes output of one command into a second

Exercise:

List the root directory (`ls -l /`)

List one of the directories within it (`ls -l /usr/lib`)

Pipe the list to search (`ls -l /usr/lib | grep python`)

Use `grep` and pipe with one of your own files
(`cat filename | grep searchphrase`)

> The greater-than sign copies output on the screen into a file

- 1 List the files in your directory (`ls -l`)
- 2 Now list them into a file
`ls -l > myfiles.txt`
- 3 Explore the file (`nano myfiles.txt`)

- 1 Try saving output of `du` command for different folders
- 2 Redirect output of home directory into a file
`du /home/user > filename`
- 3 Use the `wc` command to see the size of the output

Exercise with grep and redirection

- 1 Redirect listing of home directory, but grep to find only pdf files
`ls -Rlt | grep pdf > filename`
- 2 Use the `wc` command to see the size of the output
- 3 Explore the file with `nano`

Many commands with details about system. Far more information than usually needed.

Usually used with Google: after googling tells you a detail you need.

- 1 df [disk space]
- 2 fdisk [disks]
- 3 iwconfig [network]
- 4 lspci [peripherals]
- 5 free -m [memory]
- 6 sudo lshw [hardware complete]

Collect several commands into one file: your own command

- 1 Type the following into a file named *mycmd*
- 2 `cat %1 %2 > newfile`
- 3 Save it and exit nano
- 4 At command-line type (*file1* and *file2* must already exist)
`chmod u+x mycmd`
`./mycmd file1 file2`

Check for the file named *newfile*

File list and loop

- 1 Type these 6 lines into a file *myfile2* and save

```
ext=$1
echo "These files have extension $ext:"
for file in `ls`
do
  echo " -" $file | grep $ext
done
```

(The single quote marks are the back-tick: top-left on keyboard)

- 2 `chmod myfile2 u+x`
- 3 `myfile pdf`

Adding the pdf after the command is called an argument. Carrying out a command within the do loop would make this a powerful tool.

```
sudo apt-get install lame
```

lame converts audio files from one format to another
for example, to convert .wav files on a CD to .mp3 files

```
sudo apt-get install pdftk
```

pdftk is a useful command-line utility for working with pdfs

Try

```
> cupsfilter myfile1 > myfile1.pdf
```

```
> cupsfilter myfile2 > myfile2.pdf
```

```
> pdftk myfile1.pdf myfile2.pdf cat output myfile3.pdf
```

- Efficient and powerful
- Frequently used by programmers
- Add to your tool-kit

A good tip: Don't expect to remember commands, else you will be very frustrated.

Count on Google for answers