

# Command-line 101



[https://commons.wikimedia.org/wiki/File:Ken\\_Tompson\\_sitting\\_and\\_Dennis\\_Ritchie\\_at\\_PDP-11\\_\(2076612463\).jpg](https://commons.wikimedia.org/wiki/File:Ken_Tompson_sitting_and_Dennis_Ritchie_at_PDP-11_(2076612463).jpg)

GETTING TO  
KNOW THE  
UNIX/LINUX CLI



# What is the Command Line?

A terminal window, is a text-only window in a graphical user interface (GUI) that emulates a console.

A GUI is a type of human-computer interface (i.e., a system for people to interact with a computer) that uses windows, icons, pull-down menus and a pointer and that can be manipulated by a mouse (and usually to some extent by a keyboard as well).

A command is an instruction telling a computer to do something, such as launch a program.

A command line interface (CLI) is an all-text display mode that has a command interpreter running in it and that shows commands, including their arguments, and other data as they are typed in at the keyboard as well as the results of most such commands.

A command interpreter, also referred to as a shell, is a program whose primary function is to read commands that are typed into a console or terminal window and then execute (i.e., run) them.

# Navigation

(Where am I?)

```
> pwd
```

Shows the path to the current directory.

`"print working directory"`

# Navigation

(What's in here?)

```
> ls
```

List the contents of the current directory

Options:

- `-l` structured (includes permissions, owners, access dates) long list format
- `-R` recursive (show subdirectories too)
- `-a` all (show hidden system files too)

**"list contents"**

# SHALL WE PLAY A GAME?

**In your terminal, try:**

```
> ls --help
```

This will list all the available options with brief explanations for each. Some might not make sense yet - don't worry. Try a few out! Can you figure out what the following do?

```
> ls -r
```

```
> ls -A
```

```
> ls -s -h -S
```

# Useful CLI tips and tricks

If you want to...	Try:
See basic help	<code>&lt;command&gt; --help</code>
See the manual for a command	<code>man &lt;command&gt;</code>
Exit from manual or page view	<code>q</code>
Show last command you entered	up arrow
Autocomplete filename or path	tab
Kill a process	CTRL + c
Jump to beginning of a command	CTRL + a

# Navigation

(How do I get over there?)

```
> cd <directory name>
```

Changes to the specified directory

<pre>&gt; cd ..</pre>	Up one level
<pre>&gt; cd ../..</pre>	Up two levels (etc.)
<pre>&gt; cd ~</pre>	Home directory
<pre>&gt; cd</pre>	Home directory
<pre>&gt; cd -</pre>	Change to last directory

**"change directory"**



# Managing files and directories

READING, CREATING, COPYING, REMOVING, EDITING

# Creating directories

(How can I make a new folder?)

```
> mkdir <name>
```

Makes a new directory called <name>

```
> mkdir -v
```

Verbose - output info after

```
> mkdir -p
```

Create parents as needed

```
vagrant$ mkdir test/test1/test2/test3
mkdir: cannot create directory 'test/test1/test2/test3': No such file or directory
[~]
vagrant$ mkdir -p -v test/test1/test2/test3
mkdir: created directory 'test'
mkdir: created directory 'test/test1'
mkdir: created directory 'test/test1/test2'
mkdir: created directory 'test/test1/test2/test3'
```

“make directory”

# Deleting directories

(How can I remove an existing folder?)

```
> rmdir <name>
```

Deletes directory called <name>. Directory **MUST** be empty for the command to succeed.

```
> rmdir -v
```

Verbose - output info after

```
> rmdir -p
```

Delete parents as well  
(if also empty - if there are files in there or other directories it will fail)

**"remove directory"**

# Deleting files

(How can I remove an existing file?)

```
> rm <filename>
```

Deletes file called <filename>.

```
> rm -v
```

Verbose - output info after

```
> rm -i
```

Prompt before every deletion

```
> rm -r
```

Recursive - remove subdirs and contents below as well

```
> rm -f
```

Force - ignore warnings

**"remove"**



# DANGER!

`rm` is a powerful command, and you can easily accidentally delete a lot with the `-r` and `-f` options..

# Copying files

(How can I copy a file to another place?)

```
> cp <file1> <file2>
```

Makes a copy of <file1> in location and with name of <file2>. Example:

```
> cp atom/LICENSE /vagrant/atom-license.txt
```

NOTE: if /vagrant/atom-license.txt already exists, then it will be overwritten!

See the manual for other options

“copy”

# Moving files

(How can I move a file to another place?)

```
> mv <file1> <file2>
```

Moves <file1> to new location <file2>. Can rename file as part of the operation.

NOTE: if <file2> already exists, then it will be overwritten by the contents of <file1>

See the manual for other options

**"move"**

# Reading files

(How can I view the contents of a file?)

```
> less <filename>
```

Show 1 page at a time. Space bar will page through, q to quit

```
> cat <filename>
```

Print whole file in terminal

```
> head <filename>
```

Display first 10 lines of <filename>

```
> tail <filename>
```

Display last 10 lines

```
> tail -f <filename>
```

Display last 10 lines and contents as the file grows. Useful for logs, etc.





# BONUS ROUND

<https://www.therefinedgeek.com.au/index.php/2012/10/02/mcpixel-oh-the-lols-you-will-have/>

Show current date and time

```
> date
```

See a calendar of the current month

```
> cal
```

See a calendar of the current year

```
> cal -y
```

List users logged on to this server

```
> who
```

Show your username

```
> whoami
```

List users and session info

```
> w
```



# BONUS ROUND

<https://www.therefinedgeek.com.au/index.php/2012/10/02/mcpixel-oh-the-lols-you-will-have/>

Clear current commands out of view

> clear

Clear current commands out of view

> CTRL + L

Show full history of commands in this session

> history

copy to clipboard in the terminal

> CTRL + INSERT

paste from clipboard in terminal

> SHIFT + INSERT

Repeat the last command

> !!

Exit current session, close terminal

> exit

# Resources – Cheat Sheets

- [FossWire Unix/Linux Command Reference](#)

- [GettingGeneticsDone blog cheat sheet maintained at Tufts University](#)

- [CheatSheetWorld Unix/Linux cheat sheet](#)

- [LinOxide Linux cheat sheet](#)

There are [plenty more](#) out there!

## Unix/Linux Command Reference

FOSSwire.com

File Commands	System Info
<b>ls</b> - directory listing <b>ls -al</b> - formatted listing with hidden files <b>cd dir</b> - change directory to <i>dir</i> <b>cd</b> - change to home <b>pwd</b> - show current directory <b>mkdir dir</b> - create a directory <i>dir</i> <b>rm file</b> - delete <i>file</i> <b>rm -r dir</b> - delete directory <i>dir</i> <b>rm -f file</b> - force remove <i>file</i> <b>rm -rf dir</b> - force remove directory <i>dir</i> * <b>cp file1 file2</b> - copy <i>file1</i> to <i>file2</i> <b>cp -r dir1 dir2</b> - copy <i>dir1</i> to <i>dir2</i> ; create <i>dir2</i> if it doesn't exist <b>mv file1 file2</b> - rename or move <i>file1</i> to <i>file2</i> if <i>file2</i> is an existing directory, moves <i>file1</i> into directory <i>file2</i> <b>ln -s file link</b> - create symbolic link <i>link</i> to <i>file</i> <b>touch file</b> - create or update <i>file</i> <b>cat &gt; file</b> - places standard input into <i>file</i> <b>more file</b> - output the contents of <i>file</i> <b>head file</b> - output the first 10 lines of <i>file</i> <b>tail file</b> - output the last 10 lines of <i>file</i> <b>tail -f file</b> - output the contents of <i>file</i> as it grows, starting with the last 10 lines	<b>date</b> - show the current date and time <b>cal</b> - show this month's calendar <b>uptime</b> - show current uptime <b>w</b> - display who is online <b>whoami</b> - who you are logged in as <b>finger user</b> - display information about <i>user</i> <b>uname -a</b> - show kernel information <b>cat /proc/cpuinfo</b> - cpu information <b>cat /proc/meminfo</b> - memory information <b>man command</b> - show the manual for <i>command</i> <b>df</b> - show disk usage <b>du</b> - show directory space usage <b>free</b> - show memory and swap usage <b>whereis app</b> - show possible locations of <i>app</i> <b>which app</b> - show which <i>app</i> will be run by default
Process Management	Compression
<b>ps</b> - display your currently active processes <b>top</b> - display all running processes <b>kill pid</b> - kill process <i>pid</i> <b>killall proc</b> - kill all processes named <i>proc</i> * <b>bg</b> - lists stopped or background jobs; resume a stopped job in the background <b>fg</b> - brings the most recent job to foreground <b>fg n</b> - brings job <i>n</i> to the foreground	<b>tar cf file.tar files</b> - create a tar named <i>file.tar</i> containing <i>files</i> <b>tar xf file.tar</b> - extract the files from <i>file.tar</i> <b>tar czf file.tar.gz files</b> - create a tar with Gzip compression <b>tar xzf file.tar.gz</b> - extract a tar using Gzip <b>tar cjf file.tar.bz2</b> - create a tar with Bzip2 compression <b>tar xjf file.tar.bz2</b> - extract a tar using Bzip2 <b>gzip file</b> - compresses <i>file</i> and renames it to <i>file.gz</i> <b>gzip -d file.gz</b> - decompresses <i>file.gz</i> back to <i>file</i>
File Permissions	Network
<b>chmod octal file</b> - change the permissions of <i>file</i> to <i>octal</i> , which can be found separately for user, group, and world by adding: <ul style="list-style-type: none"><li>• 4 - read (r)</li><li>• 2 - write (w)</li><li>• 1 - execute (x)</li></ul> Examples: <b>chmod 777</b> - read, write, execute for all <b>chmod 755</b> - rwx for owner, rx for group and world For more options, see <b>man chmod</b> .	<b>ping host</b> - ping <i>host</i> and output results <b>whois domain</b> - get whois information for <i>domain</i> <b>dig domain</b> - get DNS information for <i>domain</i> <b>dig -x host</b> - reverse lookup <i>host</i> <b>wget file</b> - download <i>file</i> <b>wget -c file</b> - continue a stopped download
SSH	Installation
<b>ssh user@host</b> - connect to <i>host</i> as <i>user</i> <b>ssh -p port user@host</b> - connect to <i>host</i> on port <i>port</i> as <i>user</i> <b>ssh-copy-id user@host</b> - add your key to <i>host</i> for <i>user</i> to enable a keyed or passwordless login	Install from source: <b>./configure</b> <b>make</b> <b>make install</b> <b>dpkg -i pkg.deb</b> - install a package (Debian) <b>rpm -Uvh pkg.rpm</b> - install a package (RPM)
Searching	Shortcuts
<b>grep pattern files</b> - search for <i>pattern</i> in <i>files</i> <b>grep -r pattern dir</b> - search recursively for <i>pattern</i> in <i>dir</i> <b>command   grep pattern</b> - search for <i>pattern</i> in the output of <i>command</i> <b>locate file</b> - find all instances of <i>file</i>	<b>Ctrl+C</b> - halts the current command <b>Ctrl+Z</b> - stops the current command, resume with <b>fg</b> in the foreground or <b>bg</b> in the background <b>Ctrl+D</b> - log out of current session, similar to <b>exit</b> <b>Ctrl+W</b> - erases one word in the current line <b>Ctrl+U</b> - erases the whole line <b>!!</b> - repeats the last command <b>exit</b> - log out of current session

\* use with extreme caution.

FOSSWIRE.COM

# Resources – Tutorials

- [LinuxCommand – Learning the Shell tutorial](#)
- [LearnPythonTheHardWay – Command Line Crash Course](#)
- [Ryan's Tutorials – Linux Tutorial](#)
- [Learn Enough Society – Learn Enough of the Command Line to be Dangerous](#)

There are [plenty more](#) out there!

*COMMAND LN*

101

**MISSION ACCOMPLISHED**

