Cheatography

Linux Basic Commands Cheat Sheet

by [deleted] via cheatography.com/124743/cs/23886/

Basic comands

`cat file1 file2 ...`	Print the contents of *file1*, *file2*, ...
`ls`	List the contents of a directory.
`ls -l`	Use a long listing format
`ls -a`	Do not ignore entries starting with `.`
`cp file1 file2`	Copy *file1* to *file2*
`cp file1 ... fileN dir`	Copy a number of files to a directory
`mv file1 file2`	Rename *file1* to *file2*
`mv file1 ... fileN dir`	Move a number of files to a directory
`touch file`	Create a file. If the file already exists, `touch` does not change it
`rm file`	Remove a file
`rm -r dir`	Recursively remove all files and subdirectories in *dir*
`echo`	Print `echo`'s arguments to the standard output
`pwd`	Print working directory
`pwd -P`	Print true full path, not path of symbolic link
`sudo command`	Run *command* as root

Navigating Directories

`cd dir`	Change the shell's current working directory
`mkdir dir`	Create a new directory
`rmdir dir`	Remove the directory *dir* if *dir* is empty

Linux has a directory hierarchy starts at `/` (*root directory*).
Directory separator is the slash (`/`).
Two dots (`..`) refers to the parent of a directory.
One dot (`.`) refers to the current directory.

Shell Globbing (Wildcards)

`*`	A number of any characters
`?`	A single character
`[]`	Specify a range. `[ab]` can become: `a`, `b`. `[a-c]` can become: `a`, `b`, `c`
`[!a-c]`	Any single character except *a*, *b*, *c*

Globbing is the operation that expands a wildcard pattern into the list of pathnames
It is applied on each of the components of a pathname separately. `/` in a pathname cannot be matched.
If filename starts with `.`, `.` must be matched explicitly.
Wildcard patterns are not regular expressions, they match filenames, not text.

Search files

`grep RegEx file`	Search for regular expression pattern in *file*
`grep -i`	Case-insensitive search
`grep -v`	Print all lines that don't match
`find dir -name file -print`	Find *file* in *dir* and display the pathname of it
`locate file`	Search an index that the system builds periodically

Display a file

`less file`	Display the contents of `file` one screenful at a time
`spacebar`	Go forward one screenful
`b`	Skip back one screenful
`/word`	Search forward for *word*
`?word`	Search backward for *word*
`q`	Quit `less`
`head file`	Display the first 10 lines of *file*
`tail file`	Display the last 10 lines of *file*

By **[deleted]**
cheatography.com/deleted-124743/

Published 3rd August, 2020.
Last updated 3rd August, 2020.

Linux Basic Commands Cheat Sheet

by [deleted] via cheatography.com/124743/cs/23886/

Differences between text files

`diff file1 file2`	Print differences between two text files
`diff --color`	Print differences with color
`diff -y`	Print differences side by side
`diff -c`	View differences in context mode
`diff -i`	Ignore case differences
`diff -w`	Ignore all white space

`diff` gives the instructions on how to change the first file to make it match the second file.

< denotes lines in *file1*. > denotes lines in *file2*.

Change command

`LaR`: Add the lines in range R of the second file after line L of the first file.

`FcT`: Replace the lines in range F of the first file with lines in range T of the second file.

`RdL`: Delete the lines in range R from the first file so that both the files sync up at line L.

Environment and Shell Variables

`stuff=blah`	Create a shell variable/Assign a value to a variable
`PATH=$PATH:dir`	Appends ":*dir*" to the end of `PATH` variable
`$STUFF`	Access a variable
`export STUFF`	Make `$STUFF` shell variable into an environment variable
`unset STUFF`	Delete variable `STUFF`
`env`	Prints environment variables

Shell variables are variables whose scope is in the current shell session.

Environment variables are shell variables which has been exported. Children processes get their own copy of the parent variables so they can never change the environment variables in their parent process. Environment variables must be `name=value` pair.

Command path

`PATH` is environment variable that contains *command path* (list of system directories that the shell searches when trying to locate a command).

Command-Line Editing

CTRL-B	Move the cursor left
CTRL-F	Move the cursor right
CTRL-A	Move the cursor to the beginning of the line
CTRL-E	Move the cursor to the end of the line
CTRL-W	Erase the preceding word
CTRL-U	Erase from cursor to beginning of line
CTRL-K	Erase from cursor to end of line
CTRL-Y	Paste erased text
CTRL-D	Stop the current standard input entry from the terminal

Getting Online Help

`man command`	Show manual page for *command*
`man -k keyword`	Search for a manual page by *keyword*
`man n command`	Show manual page for *command* from section *n*

Online Manual Sections

1	User commands
2	System calls
3	Higher-level Unix programming library documentation
4	Device interface and driver information
5	File descriptions (system configuration files)
6	Games
7	File formats, conventions, and encodings (ASCII, suffixes, and so on)
8	System commands and servers

Manual pages cover the essentials, but there are many more ways to get online help. Try entering a command name followed by `--help` or `-h` to look for a certain option for a command

Cheatography

Linux Basic Commands Cheat Sheet

by [deleted] via cheatography.com/124743/cs/23886/

Shell Input and Output

command > *file*	Send the output of *command* to a file
command >> *file*	Append output to the *file*
command < *file*	Channel a file to a program's standard input
command1 | *command 2*	Send the standard output of *command1* to the standard input of *command2*
command 2> *file*	Redirect the standard error (2 is standard error stream ID)
command &> *file*	Redirect the all output to *file*
command 2>&1	Send standard error to the same place as standard output

Listing and Manipulating Processes

ps	List processes owned by root
ps x	List all processes owned by you
ps ax	List all processes on the system
ps u	Include more detailed information on processes
ps w	Show full command names
top	Show real-time view of running system
kill *pid*	Send TERMinate signal to the process with ID *pid*
kill -STOP *pid* or CTRL-Z	Send STOP (freeze) signal to the process
kill -CONT *pid*	CONTinue running the process again
kill -INT *pid* or CTRL-C	End process with INTerrupt signal
kill -KILL *pid*	Terminate the process and forcibly remove it from memory

Listing and Manipulating Processes (cont)

jobs -l	List the active jobs with their status and pid (-l)
fg %*n*	Move job that have job number *n* to the foreground
bg %*n*	Move job that have job number *n* to the background
command &	Run *command* in background

The ps command has many options. Options can be specified in three different styles—Unix, BSD, and GNU. Above commands use BSD style.

File Modes and Permissions

File's mode represents the file's permission and some extra information. There is 4 parts to the mode. First character is file type. The rest contains the permissions, which break down into three sets: *user*, *group*, *other*, in that order. Each set can contain four basic representations:

r	Means that the file is readable
w	Means that the file is writable
x	Means that the file is executable
-	Means nothing

Modifying Permissions

chmod *ugo+r file*	Add (+) owner (u), group (g) and other users (o) read (r) permissions to *file*
chmod *644 file*	Set *file* mode to absolute permission mode *644*

Symbolic links

A symbolic links is a file that points to another file or a directory

ln -s *target linkname*	Create a symbolic link from *target* to *linkname*

Cheatography

Linux Basic Commands Cheat Sheet

by [deleted] via cheatography.com/124743/cs/23886/

Archiving and Compressing Files

`gzip file`	Compress `file` to `file.gz`
`gunzip file.gz` or `gzip -d`	Uncompress `file.gz` and remove the suffix
`tar cf archive.tar file1 file2`	Create (`c`) an archive name (`f`) `archive.tar` contains `file1`, `file2`
`tar xf archive.tar`	Unpack (`x`) `archive.tar`
`tar tf archive.tar`	List the contents (`t`) of `archive.tar`

`gunzip -c file.tar.gz | tar xf -`
`zcat file.tar.gz | tar xf -`
`tar zxf file.tar.gz`
	Unpack compressed archive `file.tar.gz`

`gunzip -c` uncompresses archive then sends the result to standard output.
`tar xf -` uses standard input instead of a given filename.

Some subdirectories in root

`/bin`	Contains ready-to-run programs including most of the basic Linux commands
`/dev`	Contains device files
`/etc`	Core system configuration directory that contains the user password, boot, device, networking, and other setup files
`/home`	Holds personal directories for regular users
`/lib`	Holds library files
`/proc`	Provices system statistics
`/sys`	Provides a device and system interface
`/sbin`	Place for system management programs
`/tmp`	Storage area for temporary files

Some subdirectories in root (cont)

`/usr`	Other bulk of Linux system
`/var`	Where programs record runtime information
`/boot`	Contains kernel boot loader files
`/media`	A base attachment point for removable media
`/opt`	This may contain additional third-party software
`/vmlinuz` or `/boot/vmlinuz`	Kernel location

The reason that the root directory does not contain the complete system but other parts stored in `/usr` is primarily historic—in the past, it was to keep space requirements low for the root

By **[deleted]**
cheatography.com/deleted-124743/

Published 3rd August, 2020.
Last updated 3rd August, 2020.