

# Comprehensive Linux Cheatsheet

[Download text file](#) or [Fork me on GitHub](#).



## Gnome

<b>Ctrl+Alt T</b>	Terminal
<b>Ctrl+Alt F</b>	Firefox
<b>Ctrl+Alt H</b>	Home
<b>Ctrl+Alt G</b>	Gedit
<b>Alt F12</b>	Run command

<b>Alt F1</b>	Minimize window
<b>Alt F2</b>	Toggle maximize window
<b>Alt F3</b>	Toggle full screen
<b>Ctrl+Alt D</b>	Minimize all windows

<b>Ctrl+Alt F1-F6</b>	Terminals (tty-s)
<b>Ctrl+Alt F7-F12</b>	Xwindows
<b>Ctrl+Alt Bksp</b>	Restart X
<b>Ctrl+Alt Del</b>	Log out
<b>Ctrl+Alt End</b>	Shutdown
<b>Super PgUp/PgDn</b>	Switch workspace
<b>Middle mouse button</b>	Paste selected text

### Nautilus/Nemo

<b>Ctrl L</b>	Location, show path
<b>Ctrl+Shift N</b>	New folder
<b>Ctrl H</b>	Show hidden files

### Gedit

<b>Ctrl G</b>	Find next
<b>Ctrl+Shift G</b>	Find previous
<b>Ctrl+Shift K</b>	Clear highlights

### Terminal

<b>Ctrl+Shift C</b>	Copy
<b>Ctrl+Shift V</b>	Paste
<b>Ctrl+Shift T</b>	New tab
<b>Ctrl+Shift W</b>	Close tab
<b>Ctrl PgUp/PgDn</b>	Switch tab
<b>Ctrl +/-</b>	Zoom

<b>Ctrl D</b>	Close terminal
<b>Ctrl S</b>	Scroll lock

## Bash

Keys when in emacs mode. You can switch to 'vi' mode with 'set -o vi' command.

<b>Ctrl C</b>	Interrupt, erase line
<b>Ctrl A</b>	Go to beginning of line
<b>Ctrl E</b>	End of line
<b>Ctrl U</b>	Copy line
<b>Ctrl Y</b>	Paste line
<b>Alt .</b>	Last argument
<b>Ctrl R</b>	Search trough history
<b>Alt *</b>	Show all matches for regular expression
<b>Ctrl+Alt E</b>	Show current line passed through alias, history and shell expansion
<b>Ctrl X, Ctrl E</b>	Edit command in editor
<b>Ctrl P</b>	Show last command (same as up arrow)

# Awesome Terminal Commands

## # Packages

<b>dpkg</b>	Low level package manager for Debian.
<b>-l</b>	Lists installed packages.
<b>-i &lt;package&gt; (sudo)</b>	Installs package from a package file.
<b>apt-get</b>	Advanced Package Tool built on top of 'dpkg'. New command called simply 'apt' is also available. It merges the functionalities of 'apt-get' and 'apt-cache'.
<b>update</b>	Updates local list of existing packages.
<b>-u dist-upgrade</b>	Upgrades by intelligently handling changing dependencies with new versions of packages. To regularly update put this line in 'crontab': 'apt-get update && apt-get -u dist-upgrade'.
<b>upgrade</b>	Same as dist-upgrade, but will not remove installed packages or install new ones.
<b>install &lt;package&gt;</b>	Also updates single package.
<b>remove &lt;package&gt;</b>	Removes package but leaves its configurations.
<b>remove apt-listchanges</b>	Useful when Debian can't find a package.
<b>purge &lt;package&gt;</b>	Removes package and its configurations. Run 'apt-get autoremove' after to remove all dependencies that are not needed anymore.
<b>autoremove</b>	Removes unneeded packages.
<b>source &lt;package&gt;</b>	Downloads code.
<b>build-dep &lt;package&gt;</b>	Installs the build dependencies.
<b>--yes</b>	Answers with 'yes' to most questions (Except the ones that can have potentially harmful consequences).
<b>--force-yes</b>	Answers 'yes' to all questions (Not recommended).
<b>apt-cache</b>	Queries the APT's internal database.
<b>search &lt;keyword&gt;</b>	Searches packages like 'apropos', but globally.
<b>show &lt;package&gt;</b>	Shows package info like version, dependencies, etc.
<b>showpkg &lt;package&gt;</b>	Similar, but also shows the packages that depend on the searched package (reverse dependencies).
<b>policy &lt;package&gt;</b>	Shows installed and remote version.
<b>apt-file</b>	APT package searching utility.
<b>search &lt;file&gt;</b>	Search in which package a file is included.
<b>update</b>	Updates local list of package contents.
<b>aptitude</b>	Enables package browsing (skin for apt-get).

**search '~i!~M'** Lists installed packages that were not installed as a dependency, with short description of each.

**search <package>** Package search.

**wine** Installs wine applications.

**update-alternatives** Maintains symbolic links determining default commands.

**unattended-upgrade** Automatic installation of security upgrades.

## Commands

**apropos <cmd>** Searches the manual page names and descriptions (use quotes for phrases).  
**-a** Matches all keywords.

**whatis <cmd>** Displays one-line manual page description.

**whereis <cmd>** Locates the binary, source, and manual page files for a command.

**which <cmd>** Locates only the binary of a command.

**wtf** Translates acronyms and filename suffixes.

## Install Mantra

```
./configure --help
./configure
make
sudo make install
```

## # General

**su** Switches user.  
**- <user>** Switches to user.  
**-** Switches to root.

**man** Help on commands.  
**<section>** Section numbers: 1. Programs, 2. System calls, 3. Library calls, 4. Special files, 5. File formats, 7. Miscellaneous, 8. System administration commands

**echo** Prints passed text.  
**-n** Does not add newline at the end.  
**-e** Enables interpretation of backslashed letters.

**xargs <cmd>** Passes output from one command to arguments of another:  
**'echo -a | xargs ls'**  
**-t** Echoes the command before executing it.  
**-p** Echoes command and asks for confirmation before execution.  
**-0** Input items are separated by null character instead of space.

**tee <file>** Sends output of a program to specified file and to standard output:  
**'<cmd\_1> | tee out\_1.txt | <cmd\_2>'**  
**/dev/tty** Sends output to terminal and to standard output

**expr** Evaluates passed expression.  
**1 + 1** Prints '2'.

**bc** Evaluates input. It's basically a calculator, but also provides some control commands.  
**echo 1 + 1 | bc** Prints '2'.  
**echo "scale=5;3/4" | bc** Prints '.75000'.

**sh** Runs command interpreter (shell). Can run a script even if not executable.  
**-c '<commands>'** Starts new non-interactive shell and reads commands from arguments instead of 'stdin'.  
 To append lines to system configuration file run:  
**'sudo sh -c 'echo "<text>" >> <file>''**

**bash** Runs bash command interpreter (shell).  
**-c** Reads commands from arguments instead of 'stdin'.  
**-n <script>** Checks script for errors.  
**-x** Prints commands before execution. Useful for debugging.

**gcc** Gnu C compiler. Run 'g++' for C++ code.

**-w** Suppresses warnings (Only prints errors).  
**-Wall** All warnings.  
**-g** Compile for debugging.  
**-std=<std>** Sets the standard. Supported standards for C are: 'c90', 'gnu90', 'c99', 'gnu99', 'c11' and 'gnu11'.  
 Supported standards for C++ are: 'c++98', 'gnu++98', 'c++11' and 'gnu++11'.  
 'gnu90' and 'gnu++98' are the default options.  
**-O<level>** Optimization level. '0': Reduce compilation time (default), '1-3': -  
 Level of optimization, 's' - Optimize for size, 'g' - Optimize debugging  
 experience.

**run-parts <dir>** Runs all scripts or programs in a directory.  
**date** Tells and sets date and time.  
**-s <string>** Sets date.  
**+%T -s "10:13:13"** Sets time.  
**timedatectl** Controls the system time and date.  
**set-timezone CET** Sets timezone.  
**cal** Calendar  
**xclip** Copies to clipboard.  
**mkfifo <pipe>** Creates named pipe during that shell session.  
**mkisofs** Creates a DVD/CD image.  
**genisoimage** Creates a DVD/CD image (Debian).  
**cdrecord** Writes to a CD/DVD.  
**acpi** Checks battery.  
**fdisk -l (sudo)** Shows partitions.  
**shutdown** Closes down the system at a given time.  
**now** Takes you to the single user mode.  
**-h now** Begins the shutdown procedure, same as 'halt' and 'poweroff'.  
**-h 11:50** At 11:50.  
**-r now** Same as 'reboot'.  
**make** Utility that maintains groups of programs.  
**-q** Doesn't run any commands, just returns '0' exit code if everything is up to date or non-zero otherwise.  
**-B** Unconditionally makes all targets.

## Files

**ls**

- d** List directory names instead of contents
- S** Sort by size
- t** Sort by time
- l** One file per line
- ./\*** Ls one level deep
- i** Get inode number of file (file id). Use 'sudo find / -inum <number>' to find all links that point to same file.

**cp**

- i** Interactive (Prompts before overwrite)
- v** Verbose (Explains what is being done)
- R** Copy directories recursively
- p** Preserve mode, ownership and timestamps
- preserve=all** Also preserves context, links and xattr

**rm**

- i** Interactive (Prompts before every removal)
- v** Verbose (Explains what is being done)
- f** Force remove (Does not prompt, useful if 'rm' is aliased with '-i')
- R** Removes directories and their content recursively

**mkdir**

- p** Make parents if needed

**ln** Makes links to the files
 

- s <file> <link>** Makes symbolic link. If you want to use relative paths you must be in links directory !!!!!!!!!!!!!!!!!!!!!

**df**

- h** Displays humanly readable free disk space

**du**

- s <dir>** Directory size

**mc** Midnight commander  
**Alt o** Open parent dir in another panel

	<b>Ctrl o</b> Switch to bash	
<b>find &lt;dir&gt;</b>	<b>-name &lt;file&gt;</b> Search by name <b>-regex &lt;regex&gt;</b> Use regex for name search <b>-not</b> Insert before other options to negate <b>-maxdepth &lt;levels&gt;</b> Descend only to levels deep <b>-samefile &lt;file&gt;</b> Find all hard links of a file <b>-xdev</b> Don't descend directories on other filesystems <b>-inum &lt;inum&gt;</b> Find files with the inode number <b>-type &lt;f d b ...&gt;</b> Find files of type <b>-delete</b> Delete found files <b>-exec &lt;cmd&gt; {} \;</b> Find files and execute command for every found file. '{}' is replaced with filename <b>-exec &lt;cmd&gt; {} +</b> Find files and execute command with all filenames in place of '{}' <b>-atime +/-n</b> Find files that were last accessed less or more than n days. <b>-print0   xargs -0 &lt;cmd&gt;</b> Sends found files to a command as parameters. Uses 'NUL' character as separator, necessary for filenames with spaces	
<b>locate &lt;regex&gt;</b>	Similar as 'find' but using index	
	<b>-i</b> Ignore case <b>--regex</b> Interprets all patterns as extended regex <b>-0   xargs -0 &lt;cmd&gt;</b> Sends found files to a command as parameters.	
<b>updatedb (sudo)</b>	Update locate index	
<b>md5sum</b>	Prints md5 sum hash of a file	
<b>read</b>	Read single line from standard input	
	<b>-n 1</b> Print after reading one character <b>-s</b> Do not echo input coming from terminal	
<b>shred</b>	Securely remove files	
<b>file</b>	Determine file's type	
<b>tree</b>	Ls in a tree-like (hierarchical) format	
<b>install</b>	Copy files and set attributes	
<b>gpg</b>	Decrypt file with password	
	<b>-c</b> Encrypt	
<b>mktemp</b>	Create a temporary file or directory in '/tmp' and returns it's name.	
<b>rename</b>	<b>s/&lt;from&gt;/&lt;to&gt; &lt;files&gt;</b> Renames multiple files using 'sed' syntax	
<b>rsync</b>	A fast, versatile, remote (and local) file-copying tool	
	<b>-Hbaz -e ssh 'src_dir' &lt;user&gt;@&lt;host&gt;:dest_dir'</b> - Backs up the 'src_dir': '-H' preserves hard links, '-b' renames preexisting destination files (back up), '-a' preserve everything except hard links and '-z' compresses. <b>cmp</b> Compares two files, similar to diff but also for binaries	
<b>stat</b>	Displays files status	
	<b>-c%X</b> Time of last modification of the file	
<b>readlink</b>	<b>-f</b> Follow link recursively and print files path	
<b>xdg-open</b>	Open file with default application for the file type	
<b>dialog</b>	Display dialog box from shell script	
<b>watch</b>	Execute command periodically	

## Archives

<b>dtrx</b>	<b>&lt;archive&gt;</b> Universal archive extractor
<b>tar</b>	<b>xvzf &lt;file&gt;.tar.gz (.tgz)</b> Decompress and detar <b>xvjf &lt;file&gt;.tar.bz2</b> Decompress and detar <b>-cf &lt;archive&gt;.tar &lt;files&gt;</b> Compress
<b>unzip</b>	<b>\*.zip</b> Backslash is necessary so that bash doesn't expand the '*' <b>-d &lt;dir&gt;</b> Extract into directory (create if doesn't exist)
<b>zip</b>	<b>-r &lt;archive&gt; &lt;dir&gt;</b> Compress whole directory recursively. <b>-g &lt;archive&gt; &lt;files&gt;</b> Add files to existing archive (grow).
<b>unrar</b>	<b>e</b> Extract files from rar archive
<b>zcat</b>	Cats gzipped file

## Terminal Multiplexers

<b>screen</b>	Switch between multiple virtual terminals (useful in ssh). Prefix for a command is 'Ctrl a'.
<b>c</b>	New terminal,
<b>n</b>	Next,
<b>p</b>	Previous,
<b>a</b>	Go to beginning of line,
<b> </b>	New region vertically,
<b>S</b>	New region horizontally,
<b>tab</b>	Move to next region,
<b>Q</b>	Close all but selected region,
<b>X</b>	Kill the current region,
<b>esc</b>	Enter copy/scrollback mode -> space: start/stop marking,
<b>]</b>	Paste,
<b>k</b>	Kill window,
<b>t</b>	Show time and avg CPU load
<b>tmux</b>	Terminal multiplexer, better screen. Prefix for a command is 'Ctrl b'. Most commands are the same as in 'screen'.
<b>ls</b>	Shows running sessions
<b>attach [-t &lt;no&gt;]</b>	Attach to running session
<b>d</b>	Detach from currently attached session
<b>pgup</b>	Enter in copy mode and pageup,
<b>[</b>	Copy mode,
<b>]</b>	Paste,
<b>"</b>	Split horizontally,
<b>%</b>	Split vertically

## # Bash

<b>"\$x"</b>	ALWAYS PUT DOUBLE QUOTES AROUND VARIABLE!!!!!!!!!!!!!! All variables in bash are global!!!!!!
<b>"\$*"</b>	Combines all the arguments into single word, separating them with first character of IFS variable. If IFS is not set, space is used. If IFS is null, no separator is used!!!!!!!!!! No args provided will result in one empty string being passed on!!!
<b>"\$@"</b>	Use this instead!!!! Will retain arguments as-is, so no args provided will result in no args being passed on. This is in most cases what you want to use for passing on arguments. Google: "\$@" is right almost everytime, and \$* is wrong almost everytime.
<b>"\$#"</b>	Number of arguments
<b>"\$1"</b>	First argument
<b>"\$0"</b>	Name of the script
<b>\$'\n'</b>	String literal with escape sequences (there is a backslash before n) If you want IFS to be a new line (useful with for loop) you need to: 'IFS=\$'\n'' - The dollar forces substitution!!!! Also if you want 'while read line; do...' to preserve leading spaces and tabs, you need to set IFS=""
<b>\$?</b>	Exit code of last command (0 - Success)
<b>Ctrl-Z, kill %%</b>	Kill looping bash script
<b>test &lt;expr&gt;</b>	Same as '[ <expr> ]'. Returns zero exit status if true.
<b>-n</b>	Is string non empty
<b>-z</b>	Is string empty
<b>-a</b>	And
<b>-o</b>	Or
<b>=</b>	Strings are equal
<b>-nt</b>	File newer than
<b>-ot</b>	Older than
<b>-d</b>	Directory exists
<b>-e</b>	File exists
<b>-f</b>	Is a regular file
<b>-h</b>	Is a symbolic link
<b>-r</b>	Has read permission
<b>-w</b>	Has write permission

	<b>-x</b> Has execute permission
<b>[[ &lt;expr&gt; ]]</b>	Same as '[', but without word splitting and filename expansion. And with additional operators: '&&', ' ', '<', '>' (lexicographic less, more), and also regular expression matching.
<b>=~</b>	Regex comparison operator: '[ "\$HOST" =~ ^user.* ]'
<b>let &lt;expr&gt;</b>	Executes expression: let a="\$b"+2
<b>\$(command)</b>	Same as 'command'
<b>eval &lt;variable&gt;</b>	Execute string as command
<b>\$RANDOM</b>	0 - 32767
<b>input='cat'</b>	Getting standard input
<b>-</b>	In place of a file name means standard in or out
<b>set -o vi</b>	Set line editing to vi mode
<b>pushd .</b>	Put current dir on stack
<b>popd</b>	Pop dir from stack
<b>cd -</b>	go to last dir
<b>source &lt;script&gt;</b>	Run script: for example source /etc/profile (same as . <cmd>)
<b>#!/bin/bash</b>	Good practice to insert at beginning of a bash script
<b>export</b>	<b>PATH="\$PATH:&lt;dir&gt;"</b> Adds new directory to path environment variable.
<b>read</b>	<b>-p &lt;message&gt;</b> Prompt for user input
<b>var=\${1:-"&lt;default&gt;"}</b>	Setting variable with default value if \$1 is empty
<b>getopts</b>	Parse parameters/arguments, builtin
<b>getopt</b>	GNU version is even better then getopts, not a builtin
<b>while read line; do</b> <b>&lt;commands&gt;; done &lt;</b> <b>&lt;file&gt;</b>	Read from file line by line
	<b>-r</b> Do not treat backslashes as escape characters
<b>complete -F</b> <b>&lt;completion_function&gt;</b> <b>&lt;cmd&gt;</b>	Set completion function for command
<b>complete -p &lt;cmd&gt;</b>	Print the completion function for command
<b>compgen -c &lt;pattern&gt;</b>	Print all completions for pattern
<b>help &lt;builtin&gt;</b>	Display information about builtin command
<b>wait</b>	Wait for all background processes to end

## Safety

<b>set</b>	<b>-e</b> Exit if any command fails
	<b>-u</b> Exit if referencing undefined variable
	<b>-o pipefail</b> If any command in a pipeline fails, its return code is used as the return code of the whole pipeline
<b>IFS=\$'\n\t'</b>	Remove space from the default Internal Field Separator

## History

<b>sudo !!</b>	Run the last command as root
<b>␣&lt;cmd&gt;</b>	Execute a command without saving it in the history
<b>!<b>&lt;cmd&gt;</b></b>	Run last command that starts with cmd

## Redirections

<b>&lt;cmd&gt; 2&gt;</b> <b>/dev/null</b>	Redirect error output to 'null'
<b>&lt;cmd&gt; &amp;&gt;</b> <b>/dev/null</b>	Redirect both standard and error output to 'null'
<b>&lt;cmd&gt; &gt;&amp;2</b>	Write to stderr
<b>&lt;cmd&gt; 2&gt;&amp;1  </b> <b>less</b>	Add stderr to stdout and print it with less (useful for gcc)

## Arrays And Lines

Reads line by line from variable. To preserve spaces use 'IFS='.

```
while IFS= read -r line; do
    echo "... $line ..."
done <<< "$list" -
```

**\${a[1]}** Value of the second element of the array  
**for c in \${a[@]}** Iterate over array  
**\${varname:offset:length}** Get substring: 's="aeiou"; \${s:3:1} -> o'  
**\${#var}** Length of a var  
**\${#name[subscript]}** Length of the element  
**\${#name[@]}** Length of the array

## Aliases And Functions

**alias** Print all aliases  
**<name>** Print alias  
**<name>='cmd'** Set alias  
**command <cmd>** Executes original command, bypassing any aliases or shell functions that may be defined for command  
**\<cmd>** Temporarily disable alias (call original)  
**type <cmd>** Will tell you what is command aliased to or if it is a builtin, function or a command -P just check commands  
**declare -F** Print function names  
**declare -f** Print functions

## # Text

### Print

**head** **-n-<num\_of\_lines>** Print all lines but the last n  
**-c <num\_of\_chars>** Print first c characters  
**tail** **-n+<line\_num>** Start at line number  
**-f** Do not stop printing (follow)  
**cat** **-n** Number all lines  
**>> file** Simplest text editor (great for pasting)  
**less** **&<patt>** Display only lines with pattern  
**-N** Show line numbers  
**--** Do not show '~' after 'EOF'  
**+G** Tells less to start at the end of the file  
**+F** Follow the input (to scroll up first press ctrl+c)  
**-F** Or --quit-if-one-screen  
**v** Opens editor defined in '\$VISUAL' or '\$EDITOR'  
**:n** Examine the next file  
**<, >** Go to home, end  
**wc** Count lines, words and characters

### Edit

**sudo -e <file>** Edit file as sudo  
**tr <from> <to>** Translate characters  
**-d** Delete characters  
**cut <file>** Removes columns from each line of files  
**-d ':' -f 1,7 /etc/passwd** Only show the username and the shell  
**sort** Sorts lines  
**-u** Uniq, removes duplicates  
**-t** Set delimiter for fields (default is space)  
**-k** Select by which field to sort  
**uniq** Removes adjacent duplicates  
**-d** Intersection



	<b>-u</b>	Difference
<b>column</b>	Columnate text	
	<b>-t</b>	Create a table
<b>shuf</b>	Shuffle input lines	
<b>tac</b>	Concatenate and print files in reverse (reverse 'cat')	
<b>join</b>	Join lines of two files on a common field	
<b>colrm</b>	<b>[from [to]]</b> Removes columns	
<b>seq &lt;number&gt;</b>	Output numbers from 1 to number	
<b>ispell, aspell</b>	Interactive spell checker	
<b>basename &lt;path&gt;</b>	Strips directory from path	
	<b>-s .&lt;suffix&gt;</b>	Also strip suffix
	<b>-a</b>	Process multiple filenames
<b>dirname &lt;path&gt;</b>	Strip last component from path	
<b>fmt</b>	Produce roughly uniform line lengths	
<b>fold</b>	Wrap each input line to fit in specified width	
<b>paste</b>	Glue two documents side by side	
<b>sed</b>	<b>'s///g'</b>	Substitute every occurrence in line, not just the first one
	<b>'s///I'</b>	Ignore case
	<b>-r</b>	Extended syntax, for '+', '?', ... Also you shouldn't escape the parenthesis
	<b>-r 's###e'</b>	Execute match as a command
	<b>-i &lt;file&gt;</b>	Will make changes directly to the file (in place)
	<b>-u</b>	Unbuffered mode (processes input immediately)
	<b>-n l</b>	Print escape sequence (keycode) of a pressed key
<b>expand</b>	Convert tabs to spaces	
	<b>-t &lt;number&gt;</b>	Set number of spaces (default is 8)
	<b>-i</b>	Do not convert tabs after non blanks

## Diff

<b>diff</b>	<b>-u &lt;files&gt;</b>	Unified format
	<b>--brief -r</b>	Compare two directory trees
<b>colordiff</b>	Version of diff with colors	
<b>sdiff</b>	Two files side by side	
<b>comm</b>	Compare two sorted files line by line	
<b>patch</b>	Apply a diff file to original	
	<b>patch &lt; patch.diff</b>	Apply patch
	<b>diff -u &lt;old_file&gt; &lt;new_file&gt; &gt; patch.diff</b>	Create patch

## Search

<b>grep &lt;patt&gt;</b>	<b>-v</b>	Inverse
<b>&lt;file&gt;</b>	<b>-n</b>	Line numbers
	<b>-w</b>	Whole word
	<b>-A&lt;num&gt;</b>	Print also num lines after
	<b>-B&lt;num&gt;</b>	Print also num lines before
	<b>-r</b>	Recursive
	<b>-o</b>	Print only matching part
	<b>-P</b>	Perl notation with additional operators such as: '\\t', '+' and '?' (non-greedy!!!!).
	<b>-i</b>	Ignore case
	<b>-I</b>	Do not process binary files
	<b>-l</b>	Just print files with matches
	<b>-L</b>	Just print files without matches
	<b>-e &lt;patt&gt;</b>	Necessary to put before pattern if it starts with '-' '!!!!!!' or if you want multiple patterns.
	<b>  wc -l</b>	Count occurrences
	<b>--line-buffered</b>	Processes input line by line instead of in bigger chunks
<b>look</b>	Display lines beginning with a given string	
<b>strings</b>	Print all text parts of binary file	

**Convert**

<b>todos, fromdos</b>	Convert line endings form/to windows format (package tofromdos)
<b>enscript</b>	Converts text files to postscript, rtf, HTML
<b>gs</b>	Ghostscript: postscript and PDF language interpreter and previewer
<b>pdftohtml</b>	Pdf to html
<b>pdftotext</b>	Pdf to text
<b>libreoffice</b>	New Openoffice
<b>figlet</b>	Display large characters made up of ordinary screen characters (Ascii art)
<b>toilet</b>	Similar (Ascii art)
<b>cproto</b>	Generates C function prototypes (declarations)

**Editors**

<b>nano</b>	Simple text editor. <b>/etc/nanorc</b> Config file. <b>/usr/share/nano/&lt;lang&gt;.nanorc</b> Syntax highlight files. <b>Alt + / or ?</b> Go to last line.
<b>fte</b>	Cool text editor with CUA (IBM)-shortcuts
<b>diakonos</b>	Simple terminal text editor with ctrl-c for copy
<b>pyroom</b>	Distraction free writing (gui)

**# Network**

<b>whois</b>	Info about domain
<b>host &lt;ip/hostname&gt;</b>	DNS lookup utility
<b>nslookup</b>	Same interactively
<b>dig</b>	Same, lot of options
<b>hostname</b>	Prints/sets computer name, to set it permanently edit '/etc/hostname' and '/etc/hosts'
<b>netstat</b>	Displays contents of /proc/net files, status of ports... <b>-r</b> Show routing table <b>-i</b> Show interfaces
<b>arp</b>	Manipulate the system ARP cache (IP -> mac)
<b>route</b>	Tool used to display or modify the routing table <b>add default gw &lt;ip&gt;</b> Change the default gateway should DNS not be configured correctly on your machine, you need to edit '/etc/resolv.conf' to make things work
<b>ifconfig eth0</b>	<b>down/up (sudo)</b> Turn network interface on/off <b>&lt;ip&gt; netmask &lt;mask&gt; up</b> Set ip and mask
<b>ifup eth0</b>	Will bring eth0 up if it is currently down.
<b>ip</b>	<b>link show</b> List network interfaces <b>link set dev eth0 up</b> Bring interface eth0 up or down <b>addr show</b> List addresses of interfaces <b>route add default via &lt;ip&gt;</b> Set default gateway
<b>traceroute, traceroute6, traceroute6.iputils</b>	Traces route
<b>tracpath, tracpath6</b>	Similar (iputils package)
<b>mtr</b>	Combines the functionality of the traceroute and ping
<b>findsmb</b>	List info about machines that respond to SMB name queries - Windows based machines sharing their hard disks
<b>/etc/services</b>	List of internet services with their port numbers
<b>NetworkManager</b>	Network management daemon, configuration file is in /etc/NetworkManager/NetworkManager.conf
<b>nm-tool</b>	Prints info
<b>nm-online</b>	Is network connected
<b>nmcli</b>	Command-line tool for controlling NetworkManager

<b>nc</b>	(netcat) It can open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, do port scanning...
<b>ncat</b>	Concatenate and redirect sockets
<b>ethtool eth0</b>	Show status of eth0 -S Statistics -s Change settings (speed, duplex,...)
<b>ss -tupl</b>	List internet services on a system -tuo List active connections to/from system

## Wireless

<b>iwconfig</b>	Sets the wireless configuration options basic to most wireless devices
<b>iwlist wlan0</b>	<b>&lt;option&gt;</b> Displays current status information of a device, more detailed then iwconfig <b>scan (sudo)</b> List wireless networks in range
<b>iwspy</b>	Sets the list of IP addresses in a wireless network and checks the quality of their connections
<b>iwpriv</b>	Accesses configuration options specific to a particular device
<b>rkill</b>	<b>list</b> Show wireless adapters (wifi and bluetooth) <b>block/unblock &lt;dev_num&gt;</b> Block/unblock wireless device
<b>iw dev wlan0</b>	<b>link</b> Show link status of wlan0 <b>set biterates &lt;standard&gt;</b> Manually set interface speed <b>scan (sudo)</b> List wireless networks in range
<b>wavemon</b>	Monitor wireless connection link quality

## # Internet

<b>mutt</b>	Mail client
<b>sftp</b>	Secure ftp
<b>sshd</b>	(openssh-server) ssh server daemon, on Windows service is named 'CYGWIN sshd'

**/etc/init.d/ssh restart (sudo)** Restart sshd (ssh daemon)  
**/usr/local/etc/init.d/openssh start (sudo)** Start openssh daemon  
**ssh-keygen -t rsa -C <email>** Generate rsa key pair, keys are stored in ~/.ssh

<b>ssh</b>	SSH client <b>&lt;user&gt;@&lt;address&gt; "mpg321 -" &lt; &lt;file&gt;.mp3</b> Stream audio over ssh <b>enter ~.</b> Kill unresponsive session
<b>scp</b>	Securely copy files over network. Example: 'scp <local_file> <user>@<host>:~<remote_file>'
<b>wget</b>	Download files from WWW -O Specify output file -c Continue downloading unfinished file. Can also use wildcards, but use single quotes around url. -r -ll --no-parent -A.gif Recursively to the depth of one ignoring references to the parent directory and all gifs.
<b>curl</b>	Similar -q0 - Writes to standard output + quiet -i Urls specified by standard input
<b>transmissioncli</b>	Torrent terminal client -d Download limit (kB/s) -u Upload limit
<b>youtube-dl</b>	Download from YouTube
<b>noip2</b>	Dynamic dns update client
<b>rdesktop</b>	Remote Desktop Protocol client
<b>tin, nn</b>	Usenet client
<b>nrss</b>	Rss feed reader

## Browsers

<b>lynx</b>	Terminal web browser
<b>-cmd_log=&lt;file&gt;</b>	Write keystrokes to script
<b>-cmd_script=&lt;file&gt;</b>	Read keystrokes from script
<b>-syslog-urls</b>	Log requested URLs with syslog.
<b>-dumb</b>	Prints txt to stdout
<b>-crawl</b>	Same, turns numbers off
<b>-source</b>	Prints html to stdout
<b>-l</b>	Add the current link to your bookmark file
<b>a</b>	Save the address of a document or link to a bookmark file, by default ~/lynx_bookmarks.html
<b>o</b>	Options (you can select vim mode)
<b>ctrl+n</b>	Scroll down two lines
<b>ctrl+p</b>	Scroll up two lines
<b>ctrl+p</b>	Previous line
<b>ctrl+n</b>	Next line
<b>elinks</b>	Another web browser (has menus)

## Hacking

<b>nmap</b>	<b>-sP 192.168.1.1-3</b> Network scanning <b>192.168.1.3 -p100-139</b> Port scanning <b>-O 192.168.1.3 (sudo)</b> Scanning os
<b>tcpdump (sudo)</b>	Sniffer (show network traffic) <b>-l</b> Buffered output (for piping to less, etc.)
<b>ettercap</b>	Multipurpose sniffer/interceptor/logger for switched LAN (can detect man in the middle, denial of service, DNS spoofing)
<b>driftnet</b>	Picks out and displays images from network traffic
<b>kismet</b>	Wireless 802.11b monitoring tool
<b>aircrack-ng</b>	Wireless WEP/WPA cracking utilities
<b>Cain &amp; Abel</b>	Password recovery tool for Microsoft Operating Systems
<b>dnsniff</b>	Various hacking tools: <b>arp spoof</b> For man in the middle attack <b>dsniff</b> Password sniffer for several protocols, ...
<b>ip link set eth0</b>	<b>promisc on</b> Set network interface to promiscuous mode <b>multicast off</b> Set multicast off

## # System

<b>meta, system info</b>	In cinnamon
<b>uname -a</b>	Print system info, kernel version
<b>cat /etc/issue</b>	Show name and version of distribution
<b>init</b>	Upstart init daemon job configuration
<b>telinit</b>	Change system runlevel
<b>dmesg</b>	Print the contents of your bootup (startup) messages displayed by the kernel. This is often useful when debugging problems
<b>getconf -a</b>	Print all system configuration variables
<b>nohup &lt;cmd&gt;</b>	Run a command immune to hangups, runs even after the shell is closed (writes output to nohup.out) <b>&amp;&gt;/dev/null &amp;</b> Run a command immune to hangups in background, do not save output
<b>wmctrl</b>	X Window Manager
<b>awesome</b>	Tiled window manager <b>-k</b> Check configuration script for errors

<b>busybox &lt;cmd&gt;</b>	Combines tiny versions of many common UNIX utilities into a single small executable. (1.5 Mb)
<b>mono</b>	.NET support
<b>Users</b>	
<b>users</b>	Prints logged in users
<b>who</b>	Logged in users, more data
<b>w</b>	Logged in users, also what are they running
<b>vipw</b>	Edit password file
<b>vigr</b>	Edit groups file
<b>umask</b>	the umask is a value set by the shell. It controls the default permissions of any file created
<b>usermod &lt;user&gt;</b>	Modify user account information <b>-l &lt;new_name&gt; &lt;old_name&gt;</b> <b>-c "&lt;new_real_name&gt;" &lt;user&gt;</b> <b>-d /my/new/home &lt;user&gt;</b> Change location of the users home <b>-m -d /my/new/home</b> Also move the files
<b>useradd &lt;user&gt;</b>	Add user
<b>adduser &lt;user&gt;</b>	More high level (use 'adduser <user> sudo' after to add user to sudo group) <b>--system</b> Create system user (can not log in) <b>&lt;group&gt;</b> Add user to group (only takes effect after login!!!)
<b>groups &lt;user&gt;</b>	What groups user belongs to
<b>userdel -r &lt;user&gt;</b>	Remove user and his home dir
<b>deluser</b>	<b>-remove-all-files &lt;user&gt;</b> More high level, removes also files outside home, cron jobs, itd
<b>passwd &lt;user&gt;</b>	Change password

## File Systems

<b>mkfs</b>	Build a Linux filesystem
<b>mke2fs</b>	Create a ext filesystem
<b>mkswap</b>	Set up a swap area
<b>parted</b>	Partition manipulation program <b>-l (sudo)</b> List partitions
<b>fdisk</b>	Manipulate disk partition table <b>-l (sudo)</b> List partitions
<b>disks</b>	Nice GUI partition tool
<b>mount</b>	List all mounted devices (to get list of devices use 'fdisk -l') <b>&lt;device&gt; &lt;path&gt;</b> For example mount /dev/hdc2 to /mymedia
<b>lsattr</b>	List file attributes
<b>chattr</b>	Change file attributes
<b>rkhunter -c</b>	Checks if it can find any rootkits under the system
<b>ntfsundelete</b>	Undelete files on NTFS partition

## Logging

<b>last</b>	When various users have logged in or out. This includes information on when the computer was rebooted.
<b>lastlog</b>	Displays a list of users and what day/time they logged into the system. To get all failed logins run: 'cat /var/log/auth.log   grep "failed password" -i'
<b>rsyslogd</b>	manages all the logs on your system closelog, openlog, syslog, vsyslog -> library calls that send messages to the system logger
<b>logger</b>	Makes entries in the system log
<b>zeitgeist</b>	Activity logger

## Load

<b>tlload</b>	'graphic' representation of system load
<b>top</b>	Show processes by resource consumption <, > Change resource
<b>htop</b>	Better top
<b>free -tm</b>	Displays memory statistics <b>-s &lt;seconds&gt;</b> Continuously display
<b>vmstat</b>	Performance of system components / virtual memory statistics
<b>sar</b>	System activity information
<b>iostat</b>	Disk usage
<b>time &lt;cmd&gt;</b>	Time a execution of a command
<b>perf</b>	Profiler

## Tracing

<b>auditd</b>	System call auditing (package)
<b>ausearch</b>	Querys the audit logs
<b>autrace</b>	Traces a specific process
<b>auditctl</b>	Controls the behavior of the auditd server
<b>strace &lt;cmd&gt;</b>	Trace system calls and signals. All printed system calls can be looked up by 'man'! <b>-s</b> Maximum string size we want printed (default is too short, 2000 is OK) <b>-f</b> Also follow children <b>-p &lt;pid&gt;</b> Attach to process <b>-o &lt;file&gt;</b> Write output to file <b>-c</b> Count/aggregate <b>-T</b> Time the execution of each call
<b>ltrace</b>	List library calls made by command
<b>lsuf</b>	List open files with file descriptors <b>-p &lt;pid&gt;</b> Open files by process <b>&lt;path&gt;</b> Open files in path
<b>lsmod</b>	Show which kernel modules (drivers) are loaded
<b>modinfo &lt;module&gt;</b>	Get more information about kernel module

## Hardware

<b>lshw (sudo)</b>	List all hardware
<b>lspci [-tv]</b>	Show pci info
<b>lsusb [-tv]</b>	Show usb info
<b>lscpu</b>	Print CPU information
<b>dmidecode -q</b>	Display bios/dmi information like ram size/type, max ram, computer model name, cpu information.
<b>smartctl</b>	<b>-A /dev/sda</b> Show disk usage info
<b>hdparm</b>	<b>-tT /dev/sda</b> Do a read speed test
<b>badblocks</b>	<b>-s /dev/sda</b> Check for bad blocks
<b>fstrim -v /</b>	Discard unused blocks, useful for ssd-s
<b>sensors</b>	(lm-sensors) hardware monitoring tool, temperature, fan speed
<b>pwmconfig</b>	(fancontrol) set fan speed

## Terminal

<b>tty</b>	Print the file name of the terminal connected to standard input
<b>stty</b>	Change and print terminal line settings
<b>openvt</b>	run a program on a next available tty
<b>script</b>	Makes a typescript of everything printed on your terminal. Ctrl-d to stop recording.
<b>setterm</b>	Set terminal attributes <b>-cursor off/on</b> Set cursor on/off
<b>gpm</b>	Enable mouse for tty

## Keys/Characters/Fonts

**In X**

<b>xmodmap</b>	Remap keys
<b>setxkbmap -layout us</b>	Set us keyboard layout
<b>xev</b>	Get keycodes of pressed keys
<b>xset</b>	User preference utility for X
<b>-r</b>	Turn key autorepeat off
<b>r</b>	Turn key autorepeat on

**X Not Necessary**

<b>loadkeys</b>	Load key mapping
<b>&lt;country_code&gt;</b>	
<b>showkey</b>	Get keycodes of pressed keys
<b>setfont &lt;font&gt;</b>	Set console font
	<b>/usr/share/consolefonts/Uni2-VGA16.psf.gz</b> For example
	<b>/etc/default/console-setup</b> contains the default settings
<b>jfbterm</b>	Enable unicode characters in terminal
<b>echo -en</b>	Change blue color in tty (first numeral after P means slot, and others shade)
<b>"\e]PC7373C9"</b>	

**# Processes**

<b>ps --forest</b>	View hierarchical view of processes
<b>v</b>	Virtual memory
<b>--sort &lt;field&gt;</b>	Sort by field
<b>pstree</b>	Similar
<b>pgrep &lt;pattern&gt;</b>	Prints PIDs of processes containing pattern
<b>-l</b>	Also print process name
<b>pkill &lt;pattern&gt;</b>	Kills every process that contains pattern in name
<b>kill</b>	<b>&lt;pid&gt;</b> Sends TERM signal to process
	<b>-kill &lt;pid&gt;</b> Sends KILL signal meaning force quit, data will be lost
<b>killall</b>	Uses name instead of pid
<b>skill</b>	Sends signals to command/user/tty or report process status
	<b>-stop &lt;user&gt;</b> Stop all of the users processes
	<b>-cont &lt;user&gt;</b> Continue all users processes
<b>nice</b>	Sets the priority for a process (from max of -20 to min of 20)
	<b>-20 &lt;cmd&gt;</b> Execute command with maximum priority
<b>renice</b>	Changes the priority of an existing process
	<b>+20 &lt;pid&gt;</b> Change processes priority to lowest level
<b>snice</b>	Works very similarly to skill
	<b>-10 -u root</b> Increase the priority of all root's processes
<b>pmap</b>	Report memory map of a process (mapped file)

**Jobs**

<b>Ctrl z</b>	Put job into background
<b>jobs</b>	Prints currently running job
<b>bg &lt;job_id&gt;</b>	Put job in background
<b>fg &lt;job_id&gt;</b>	Bring job to the foreground
<b>%n</b>	Job number n
<b>%s</b>	Job whose command line starts with s
<b>%%</b>	Current job
<b>%-</b>	Previous job

**Scheduled Commands**

<b>at</b>	Executes command at a particular time at 21:30 / at now + time / at -f shell_script now + 1 hour echo "ls -l"   at midnight
<b>atq</b>	List jobs currently in 'at' queue
<b>atrm</b>	Remove a job from the 'at' queue
<b>crontab -e</b>	Schedule commands for repeating execution
<b>cron</b>	Daemon that executes scheduled commands
<b>sudo service cron status</b>	Print status of cron
<b>sudo service cron [stop start restart]</b>	Stop, start, or restart cron daemon
<b>sudo vim /etc/default/cron</b>	Set logging lever
<b>cat /var/log/cron</b>	Print log
<b>anacron</b>	Like cron but it catches up with tasks next time the computer gets turned on

## Services

<b>service</b>	Allows you to start, stop or restart a service (it runs a script in /etc/init.d folder)
<b>-f sshd</b>	Restart the ssh server
<b>httpd status</b>	Get status of apache
<b>--status-all</b>	Print status of all services You can also execute the shell script directly from /etc/init.d folder like: /etc/init.d/httpd stop.

## System Runlevels

<b>runlevel</b>	Output previous and current runlevel
<b>0</b>	Shuts down the system
<b>1</b>	Administrative single-user mode
<b>2</b>	Same as 3 but without networking / multiuser with X server
<b>3</b>	Text mode state (ctrl+alt+F1) / User defined
<b>4</b>	User defined
<b>5</b>	X-window mode (ctrl+alt+F7) / User defined
<b>6</b>	Reboots
<b>S</b>	Single user mode

## # Multimedia

### Audio

<b>alsamixer</b>	Set audio level (curses)
<b>amixer</b>	Set audio levels (command line)
<b>rmmod pcspkr (sudo)</b>	Disable pc-speaker, beep
<b>cmus</b>	Music player (can be controlled from outside)
<b>mplayer</b>	Movie/music player
<b>mpg321</b>	Plays mp3
<b>ogg123</b>	Plays ogg
<b>aplay</b>	Plays audio
<b>play</b>	Plays audio
<b>arecord &lt;file&gt;</b>	Command line audio recorder and player
<b>-f &lt;format&gt;</b>	Set file format (cd)
<b>-d &lt;seconds&gt;</b>	Set duration
<b>-f cd -d &lt;seconds&gt; -t raw   lame -x -r</b>	Out.mp3 — Capture audio that is playing and convert it to mp3
<b>id3v2 -l</b>	Lists all files tags
<b>sound-juicer</b>	Cd ripper
<b>xfburn</b>	Cd burner
<b>traverso</b>	Simple daw

### Bitmap



<b>display</b>	Displays an image
<b>montage</b>	Creates a montage from images <b>&lt;input_file/s&gt; -set label '%t' &lt;output_file/s&gt;</b> Labels images
<b>convert &lt;old&gt; &lt;new&gt;</b>	Converts file format (imagemagick)
<b>import</b>	Captures screen-shots from the X server
<b>mogrify</b>	Edit image
<b>gocr</b>	Command line text recognition tool
<b>ocrad</b>	Command line text recognition tool
<b>ppmforge</b>	Creates picture of random planet or clouds
<b>gnuplot</b>	Interactive plotter: plot [-10:10] sin(x) <b>-p</b> Leave plots open after exit <b>plot &lt;file&gt;.dat</b> Plot data from dat file
<b>fbi</b>	Display images inside tty

## Video

<b>ffmpeg -i &lt;file_in&gt; &lt;file_out&gt;</b>	Video and audio format converter. Has been replaced by avconv
<b>avconv -i &lt;file_in&gt; &lt;file_out&gt;</b>	Video and audio format converter (libav-tools)
<b>openshot</b>	Gui movie editor
<b>openmovieeditor</b>	Gui movie editor
<b>aview, asciiview</b>	ASCII art image viewer and video player

# Awesome Debian Files

## Bash

<b>~/ .bashrc</b>	Executed at every shell startup, user specific
<b>/etc/bash.bashrc</b>	Executed at every shell startup, all users
<b>~/ .profile, .bash_profile, .bash_login</b>	First file found executed at login, user specific
<b>/etc/profile</b>	Executed at login, all users (put PATHS here)
<b>/etc/rc.local</b>	Last startup script executed, runs command as su

## Home

<b>~/ .Xmodmap</b>	Keyboard map
--------------------	--------------

## Bins

<b>/bin</b>	Key programs like ls, cat, bash, ...
<b>/sbin</b>	Key programs for system management like ifconfig, mkfs, fdisk, ...
<b>/usr/bin</b>	Distribution managed programs
<b>/usr/sbin</b>	Distribution managed system programs
<b>/usr/local/bin</b>	User programs not managed by the distribution package
<b>/usr/local/sbin</b>	User system management programs, not managed by the distribution package

## General Config

"Edit To Configure" or "Editable Text Configuration".

<b>/etc/mailcap</b>	Default programs for extensions
---------------------	---------------------------------

<b>/etc/passwd</b>	Users
<b>/etc/groups</b>	Groups
<b>/etc/default</b>	Boot script parameters that the end user or administrator is likely to change. <b>/console-setup</b> Set console (tty) character set, font size, ...
<b>/etc/fstab</b>	Filesystem table. To mount drive at startup, create dir in '/media' and append line like this : '/dev/sda1 /media/data ntfs user,fmask=0111,dmask=0000 0 2' (Check 'man fstab' for details).
<b>/etc/alternatives</b>	Links to default application versions (here you can change the default Java JDK)
<b>/etc/issue</b>	Name and version of distribution
<b>/etc/fstab</b>	Automatic mounts are handled by configuring the file
<b>/etc/sudoers</b>	Lists of users and the commands they can run with sudo (needs to be edited with visudo command)
<b>/etc/apt</b>	<b>/sources.list</b> List of places where to look for packages

## Services

<b>/etc/init/, ~/.init/</b>	<b>Init</b> Upstart init daemon job configuration
<b>/etc/init.d</b>	Folder with service scripts, that get executed at start and end <b>/halt</b> Runs at the end
<b>/etc/rc&lt;level&gt;.d</b>	Startup scripts for different runlevels - Links to scripts in /etc/init.d - S85httpd -> S means startup, K is for stop. (To disable service just change S for K)
<b>/etc/rc.local</b>	Last initialization file executed - Put your commands here
<b>/etc/init/ssh.conf</b>	Sshd config
<b>/etc/ssh/sshd_config</b>	Sshd config
<b>/etc/crontab</b>	System-wide crontab
<b>/etc/cron.hourly,</b> <b>/etc/cron.daily, ...</b>	Links to scripts that will execute periodically. Scripts within a cron directory are run alphabetically.
<b>/etc/rsyslog.conf</b>   <b>rsyslog.d/50-</b> <b>default.conf</b>	Log conf (need to restart rsyslogd after edit)
<b>/etc/syslog.conf</b>	Configuration information for syslogd

## Network

<b>/etc/resolv.conf</b>	Dns information
<b>/etc/sysconfig</b>	<b>/networking/devices/ifcfg-eth0</b> Use ifcfg to configure a particular interface
<b>/etc/services</b>	List of internet services with their port numbers
<b>/etc/NetworkManager</b>	<b>/NetworkManager.conf</b> Configuration file

## Proc

Various information about the system.

<b>/proc/cpuinfo</b>	Information about the CPU
<b>/proc/modules</b>	information about which kernel-modules are loaded on your system
<b>/proc/net</b>	Network related <b>/route</b> Routing table <b>/netstat</b> Displays contents of /proc/net files
<b>/proc/iomem</b>	Neat memory map
<b>/proc/partitions</b>	Partitions info
<b>/proc/acpi</b>	<b>/battery/BAT1/info</b> Battery info <b>/ac_adapter/ACAD/state</b> Adapter info <b>/wakeup</b> List of devices that can wake up your machine via acpi  <b>sudo sh -c "echo USB1 &gt; /proc/acpi/wakeup"</b> Enable device USB1 to wakeup computer from sleep/suspend
<b>/proc/net/wireless</b>	Wireless connection info

## Usr

"Unix System Resources"

<b>/usr/lib</b>	<b>/jvm</b>	Java JREs and JDKs
<b>/usr/share</b>	<b>/man</b>	Man pages
	<b>/bash-completion</b>	Bash completion functions

## Sbin

The "system-administrator's bin file". Hosts programs that would be in /bin if they didn't have "root-only" access permissions.

## Var

"Variable"

<b>/var/log</b>	System logs in here
	<b>/auth.log</b> Logins
	<b>/syslog</b> Most of the rest of the logs
<b>/var/spool</b>	Contains data which is awaiting some kind of later processing

## Boot

Kernels.

<b>/boot/grub/menu.lst</b>	Grub configuration file
<b>/etc/default/grub</b>	Grub configuration file

# Git

## Git Mantra

```
git init
git add <file> OR git add .
git status
git commit -am "<commit_message>"
```

## Clone From Github

```
git clone git@github.com:<user>/<project>
```

Download repo (later you keep refreshing with 'git pull origin master') -> You need SSH key. If you don't want, use <https://github.com/<user>/<repo>> for address.

## Generate Ssh Key

1. Check for existing keys: `cd ~/.ssh; ls -al`
2. Generate new key: `ssh-keygen -t rsa -C "your_email@example.com"`
3. Add your key to the ssh-agent: `ssh-add ~/.ssh/id_rsa`
4. Add your key to GitHub: copy contents of `~/.ssh/id_rsa.pub` and paste them into key field at 'Account settings' > 'SSH Keys' > 'Add SSH key'.

## Add To Github

```
# Create remote repository on website.
git remote add origin git@github.com:<user>/<project>.git
git pull origin master
git push origin master
# Sometimes also: git push --set-upstream origin master
```

## Remote

<b>git remote</b>	<b>update</b>	Get info about state of remote
	<b>show origin</b>	Print address of the origin
	<b>set-url origin &lt;origin&gt;</b>	'git@github.com:<user>/<repo>.git' - Change the url of origin, ssh key needed; 'https://www.github.com/<user>/<repo>.git' - Same, but without key.
<b>git status -uno</b>		Check if everything up-to-date
<b>git fetch; git checkout &lt;branch&gt;</b>		Checkout remote branch

## Tags

<b>git push --tags</b>	Push tags
<b>git fetch --tags</b>	Pulling tags (automatically if on the same branch and there is a new commit?)

## Undo

<b>git reset --hard HEAD~1</b>	Delete last commit and all of its changes
<b>HEAD~1</b>	Delete last commit but keep your changes

## Also Useful

<b>git checkout HEAD^ &lt;file&gt;</b>	Retrieve deleted file
<b>git rm --cached &lt;file&gt;</b>	Untrack file without deleting it
<b>git tag -a 0.9.1 -m "Version 0.9.1 release"</b>	Tag latest commit
<b>git tag</b>	List local tags
<b>chown -R &lt;user&gt;: &lt;group&gt; * (sudo)</b>	In .git/objects
<b>git format-patch -1 &lt;sha&gt;</b>	Generate patch file
<b>git rev-parse HEAD</b>	Get sha of head
<b>git revert</b>	
<b>git describe</b>	Print version and hash of HEAD
<b>git log --name-only --author=&lt;name&gt;</b>	Print changed files by commit
<b>git log &lt;file&gt;</b>	Print files history
<b>git ls-files</b>	List files
<b>git show &lt;revision&gt;: &lt;file&gt;</b>	Take a peek at the older revision of the file
<b>git diff &lt;commit&gt; &lt;commit&gt;</b>	Compare two commits
<b>git stash</b>	If you want to switch branches, but you don't want to commit your changes yet, you can 'stash' them
<b>git stash apply</b>	Apply the changes you stashed
<b>git stash list</b>	List all the stashes
<b>git stash show -p stash@{0}</b>	Show the diff of most recent stash
<b>git update-index --chmod=+x &lt;file&gt;</b>	Change files permissions
<b>git config -l</b>	Print repos configuration settings
<b>git config core.filemode</b>	Ignores executable bit of the files

**false**

## Checkout

**git log >** First save log to file  
**../gitLog**  
**git checkout** Then checkout previous versions  
**<hash>**  
**git checkout** Return to head  
**head**

## Github Md Format

**![Alt text]** Insert image  
**(/doc/basket-**  
**stats.png?**  
**raw=true "**  
**<Description>")**

## Bisect

**git bisect start**  
**git bisect bad** Tell git that current version is bad  
**git bisect good** Tell git the last good version you know about. Now git will checkout a version in  
**v25.0.2** between, so you can check it and tell:  
**git bisect** This will continue until the commit that introduced the bug is found  
**bad/good**  
**git bisect reset** Exit bisect mode

## Tools

**gitk** Repo explorer  
**gitg** A bit nicer version  
**tig** Text based repo explorer  
**gitstats** Generates stats for git repo, outputs HTML

## Virtual Box

**git config** Ignores the filemode changes made by the host system  
**core.filemode**  
**false**  
**git config --** If problem pulling  
**global --unset**  
**https.proxy**

# Vim

**+<linenum>** Open file at line number  
**alt+<normal mode** Escape, key !!!!!!!!!!!!!!!  
**key>**  
**ctrl+[** Escape  
**.** Execute last command again  
**;** Repeat the last character-wise search  
**\c** Case insensitive search  
**?** Search backward  
**V** Linewise visual mode  
**~** Switch case  
**>** Tab selection right  
**>>** Tab line right

<b>P</b>	Paste before cursor
<b>x</b>	Delete character
<b>gF</b>	Open file under cursor
<b>K</b>	Look up word under cursor in man pages
<b>ctrl+v</b>	Select visual box (block select)
<b>:e</b>	Reload file
<b>:sav</b>	Save file as and keep new file open (save as)

## Help

<b>ctrl+]</b>	Follow link
<b>ctrl+o</b>	Go back
<b>:q</b>	Exit help

## Movement

<b>e</b>	End of word
<b>E</b>	End of WORD
<b>W</b>	Start of WORD
<b>ge</b>	End of previous word
<b>), (</b>	Sentence
<b>{, }</b>	Paragraph
<b>]], [[</b>	Section
<b>:&lt;num&gt;</b>	Goto line number
<b>ctrl+o</b>	Go to previous location
<b>ctrl+i</b>	Go to next location
<b>%</b>	Jump to matching bracket

## Lines

<b>0</b>	Start of line
<b>^, _</b>	First non-blank of line
<b>+, -</b>	First non-blank of next/previous line
<b>Enter</b>	First non-blank of next line

## Screen Lines

<b>g0, g\$</b>	Start/end of screen line
<b>gm</b>	Middle of screen line
<b>gk, gj</b>	Up/down one screen line

## Page Up/Down

<b>H, M, L</b>	Go to top/middle/bottom of screen
<b>ctrl F, B</b>	Page up/down
<b>ctrl D, U</b>	Half page up/down
<b>ctrl E, Y</b>	One more line at bottom/top
<b>z Enter, z., z-</b>	Reposition line with cursor at top/middle/bottom

## Search

<b>*, #</b>	Search forward/backward for exact word under cursor
<b>g*, g#</b>	Same, but even when word is embedded
<b>%</b>	Find match of current brace, quote,...
<b>fx, Fx</b>	Move cursor forward/backward to x on current line
<b>tx, Tx</b>	Same, but to one char before x
<b>;/</b>	Repeat/reverse last
<b>:%s/old/new/gc</b>	Replace, like sed, c means with conformations

**Marks**

<code>''</code>	Move to position of last edit of file
<code>`.</code>	Move to last change in file
<code>`0</code>	Position where you last exited vim

**Insert Mode Commands**

<code>ctrl+h</code>	Backspace
<code>ctrl+u</code>	Delete line
<code>ctrl+w</code>	Delete previous word

**Macros**

<code>q&lt;x&gt;</code>	Record actions (macro) into x
<code>q</code>	Stop recording macro
<code>@&lt;x&gt;</code>	Execute x (macro)

**Registers**

<code>"ayy</code>	Copy line into register a
<code>"ap</code>	Paste register a
<code>:reg</code>	Access all registers

**Set Command**

<code>:set &lt;x&gt;</code>	Set x
<code>:set no&lt;x&gt;, &lt;x&gt;!</code>	Unset x
<code>:set &lt;x&gt;=value</code>	Assign x
<code>:set &lt;x&gt;-=value</code>	Remove value from <x>
<code>:set all</code>	Print all values
<code>:set &lt;x&gt;?</code>	Print x

**Set Command Operands**

<code>autoindent, ai</code>	Autoindent (noai)
<code>backup, bk</code>	Back file up before overwrite (nobackup)
<code>ignorecase, ic</code>	Ignore case in search (noic)
<code>number, nu</code>	Display line numbers (nonu)
<code>relativenumber, rnu</code>	Display relative numbers (nornu)
<code>shiftwidth, sw</code>	Number of spaces added when indenting (8)
<code>tabstop, ts</code>	Tab width (8)
<code>wrap</code>	Wrap lines (wrap)
<code>wrapscan, ws</code>	Search wraps around file (ws)
<code>mouse=a</code>	Mouse mode (use shift when selecting to copy to clipboard)
<code>linebreak</code>	Do not break words
<code>:set iskeyword=.</code>	Remove dot from words part (two words if separated with dot)

**Edit Commands**

<code>[n] operation</code>	If both n and m are specified then n x m
<code>[m] motion</code>	
<code>c, d, y</code>	Change, delete, yank
<code>C, D, Y</code>	Till the end of line
<code>cc, dd, yy</code>	Current line
<code>cf&lt;x&gt;, df&lt;x&gt;, yf&lt;x&gt;</code>	Forward up to x

<b>c), d), y)</b>	Sentence
<b>~</b>	Change case of character
<b>g~w</b>	Switch case of a word
<b>gu, gU</b>	To lower/upper case
<b>[p</b>	Paste but match current indentation
<b>r</b>	Replace character
<b>S</b>	Substitute entire line
<b>x,X</b>	Delete character/delete back
<b>.</b>	Repeat last change
<b>ctrl+a, ctrl+x</b>	Increment/decrement number under cursor

### Automatic Linebrake (Wrap)

<b>gq</b>	Formats (wraps) selected text
<b>gqq</b>	Format current line
<b>:set tw=72</b>	Set text width

### Advanced

<b>tw=72 fo=cq wm=0</b>	No automatic wrapping, rewrapping will wrap to 72
<b>tw</b>	Controls the wrap width you would like to use
<b>fo</b>	Controls whether or not automatic text wrapping is enabled, depending whether or not the t flag is set
<b>wm</b>	Controls when to wrap based on terminal size

### Colorscheme

<b>:colorscheme</b>	<b>darkblue, torte</b> Nice, darker
	<b>slate, default</b> Less contrast
<b>:highlight</b>	Set light background
<b>Normal</b>	
<b>ctermbg=grey</b>	

### Spellcheck

<b>:set spell</b>	Turn spellcheck on
<b>spelllang=en_us</b>	
<b>:set nospell</b>	Turn off
<b>:setlocal spell</b>	Set dictionary
<b>spelllang=en_us</b>	
<b>z=</b>	Show suggestions for misspelled word
<b>]s</b>	Go to next misspelled word
<b>[s</b>	Go to previous misspelled word

### Tabs

<b>:tabe &lt;file&gt;</b>	Open new tab
<b>gt, gT</b>	Go to next/previous tab
<b>ctrl+pgup/pgdn</b>	Switch tab
<b>vim -p</b>	Open one tab page per file
<b>ZZ</b>	Save and close tab (same as :wq)

### Splits

<b>:vsp</b>	Split vertically
<b>ctrl-w, direction</b>	Move to split



**Autocomplete**

**ctrl+n** Show autocomplete suggestions

**Hexdump**

**:%!xxd** Convert to hex  
**:%!xxd -r** Convert back

**Vundle**

**:PluginInstall** Install plugins

**Reformat Code**

**=** Fix indentation

# Misc

**Piratebay**

**torrents.thepiratebay.sx/7532474/Cabin.torrent** Download torrent file

**Java**

**javac -cp** Tell Java where libraries are located  
**<path>:<path>...**

**java -Xmx6g** Reserve 6 giga for process  
**myprogram**

**-jar <jar>** Execute jar  
**-cp .:<path>:<path>...** Tell java where libraries are located, you also need to pass the location of class among paths, hence .:

**export** Set heap space globally  
**\_JAVA\_OPTIONS=-Xmx1000m**

**appletviewer** Run Java applet  
**<page>.html**

**jps -lvm** List java processes

**jmap -histo:live** Memory map  
**<pid>**

**jvisualvm** Profiler

**jar cvfe** Create executable jar

**"bla.jar"**  
**<main\_class>**  
**\*.class**

**jar xf <jar>** Extract files from jar

**jar tf <jar>** Print contents of a jar

**Install Oracle Jdk**

```
sudo apt-get remove openjdk*
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java7-installer
```

**Install Oracle Jdk On Debian**

```
echo "deb http://ppa.launchpad.net/webupd8team/java/ubuntu precise main" | sudo tee /etc/apt/sources.list.
echo "deb-src http://ppa.launchpad.net/webupd8team/java/ubuntu precise main" | sudo tee -a /etc/apt/source
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys EEA14886
sudo apt-get update
```

```
sudo apt-get install oracle-java7-installer
sudo apt-get install oracle-java7-set-default
```

## Firefox

**about:config**      Layout.css.devPixelsPerPx default zoom (-1.0)

## Xrandr

<b>xrandr</b>	<b>--output VGA1 --primary</b>	Changes primary screen
	<b>--output VGA1 --auto --pos 0x0</b>	'--output LVDS1 --auto --right-of VGA1' - To change their relative positions
	<b>-q</b>	List devices
	<b>--auto</b>	Reset
	<b>--output LVDS1 --off</b>	Turn off laptop screen
	<b>--output [VGA HDMI] --mode 1600x1200</b>	24" 16x12 on

## Mount Iso

```
sudo mkdir /media/x
sudo mount -o loop <path_to_iso> /media/x
```

## Eclipse

<b>ctrl+l</b>	Quick fix
<b>alt+shift+s</b>	Source submenu
<b>ctrl+F7, esc</b>	Close pop-up console window
<b>ctrl+7, ctrl+ /</b>	Toggle comment
<b>F3</b>	Goto definition
<b>objectaid</b>	UML plugin

## Wine

<b>winecfg</b>	Drives tab to set drive
<b>wine explorer</b>	Run wine app in virtual desktop
<b>/desktop=abalaba,1024x768</b>	
<b>app.exe</b>	
<b>reason on wine: down alt</b>	Open menu
<b>down</b>	
<b>regedit</b>	Registry editor

## Cygwin

```
[cygwin] ssh-
host-cofig
[command prompt]
net start sshd
[any] ssh      Run sshd (use windows password)
<windows_username
- CASE
MATTERS!>@<host>
```

## Golang

<b>go</b>	<b>build</b>	Compile packages and dependencies
	<b>clean</b>	Remove object files
	<b>env</b>	Print Go environment information
	<b>fix</b>	Run go tool fix on packages
	<b>fmt</b>	Run gofmt on package sources
	<b>get</b>	Download and install packages and dependencies (first you need to set GOPATH to dir where packages will get downloaded)
	<b>install</b>	Compile and install packages and dependencies
	<b>list</b>	List packages

<b>run</b>	Compile and run Go program
<b>test</b>	Test packages
<b>tool</b>	Run specified go tool
<b>version</b>	Print Go version
<b>vet</b>	Run go tool vet on packages

**syntax highlight** 'https://github.com/jnwhiteh/vim-golang'

**for gedit** 'sudo cp /usr/share/gtksourceview-3.0/language-specs/go.lang  
/usr/share/gtksourceview-2.0/language-specs/'

**simple ncurses** Go get github.com/nsf/termbox-go

## Repackaging A Linux Install Iso

```
# Mount ISO
mkdir -p /mnt/linux
mount -o loop /tmp/linux-install.iso /mnt/linux

# Copy contents to a working directory
cd /mnt/
tar -cvf - Linux | (cd /var/tmp && tar -xf - )

# Make your changes and repack (on Debian use genisoimage). -c passes the name of the file that will be
cd /var/tmp/linux
mkisofs -o ../your-new.iso -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -
```

## Gdb

<b>gdb &lt;cmd&gt;</b>	Start gdb
<b>run &lt;arguments&gt;</b>	Start debugging
<b>run &lt; &lt;file&gt;</b>	Run with piped input
<b>up</b>	Follow trace
<b>print &lt;variable&gt;</b>	Print variable

## Coredump

<b>ulimit -c</b>	Set core file limit to unlimited
<b>unlimited</b>	
<b>gdb &lt;cmd&gt; core</b>	Debug core file with gdb

## Chrome

<b>F6, ctrl+l,</b> <b>alt+d</b>	Go to address bar
------------------------------------	-------------------