January 30, 2019 / Jure Šorn

Comprehensive Linux Cheatsheet

Download text file or Fork me on GitHub.

Gnome

Ctrl+Alt T	Terminal
Ctrl+Alt F	Firefox
Ctrl+Alt H	Home
Ctrl+Alt G	Gedit
Alt F12	Run command
Alt F1	Minimize window
Alt F2	Toggle maximize window
Alt F3	Toggle full screen
Ctrl+Alt D	Minimize all windows
Ctrl+Alt F1-F6	Terminals (tty-s)
Ctrl+Alt F7-F12	Xwindows
Ctrl+Alt Bksp	Restart X
Ctrl+Alt Del	Log out
Ctrl+Alt End	Shutdown
Super PgUp/PgDn	Switch workspace
Middle mouse button	Paste selected text

Nautilus/Nemo

Ctrl L	Location, show path
Ctrl+Shift N	New folder
Ctrl H	Show hidden files

Gedit

Ctrl G	Find next
Ctrl+Shift G	Find previous
Ctrl+Shift K	Clear highlights

Terminal

Ctrl+Shift C	Copy
Ctrl+Shift V	Paste
Ctrl+Shift T	New tab
Ctrl+Shift W	Close tab
Ctrl PgUp/PgDn	Switch tab
Ctrl +/-	Zoom
Ctrl D	Close terminal
Ctrl S	Scroll lock

Bash

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

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

Awesome Terminal Commands

Packages

dpkg	Low level package manager for Debian.	
	-l	Lists installed packages.
	-i <package> (sudo)	Installs package from a package file.
apt-get	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`'.	
	update	Updates local list of existing packages.
	-u dist-upgrade	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'.
upgrade	Same as dist-upgrade, but will not remove installed packages or install new ones.
install <package>	Also updates single package.
remove <package>	Removes package but leaves its configurations.
remove apt-listchanges	Useful when Debian can't find a package.
purge <package>	Removes package and its configurations. Run 'apt-get autoremove' after to remove all dependencies that are not needed anymore.
autoremove	Removes unneeded packages.
source <package>	Downloads code.
build-dep <package>	Installs the build dependencies.
--yes	Answers with 'yes' to most questions (Except the ones that can have potentially harmful consequences).
--force-yes	Answers 'yes' to all questions (Not recommended).

apt-cache Queries the APT's internal database.

search <keyword>	Searches packages like 'apropos', but globally.
show <package>	Shows package info like version, dependencies, etc.
showpkg <package>	Similar, but also shows the packages that depend on the searched package (reverse dependencies).
policy <package>	Shows installed and remote version.

apt-file APT package searching utility.

search <file>	Search in which package a file is included.
update	Updates local list of package contents.

aptitude 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.

winetricks 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	Supresses warnings (Only prints errors).
-Wall	All warnings.
-g	Compile for debugging.
-std=<std>	Sets the standard. Suported standards for C are: 'c90', 'gnu90', 'c99', 'gnu99', 'c11' and 'gnu11'. Suported standarts 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 **-1** 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 **Ctrl o** Switch to bash
find <dir>	**-name <file>** Search by name **-regex <regex>** Use regex for name search **-not** Insert before other options to negate **-maxdepth <levels>** Descend only to levels deep **-samefile <file>** Find all hard links of a file **-xdev** Don't descend directories on other filesystems **-inum <inum>** Find files with the inode number **-type <f|d|b|...>** Find files of type **-delete** Delete found files **-exec <cmd> {} \;** Find files and execute command for every found file. '`{}`' is replaced with filename **-exec <cmd> {} +** Find files and execute command with all filenames in place of '`{}`' **-atime +/-n** Find files that were last accessed less or more than n days. **-print0 | xargs -0 <cmd>** Sends found files to a command as parameters. Uses '`NUL`' character as separator, necessary for filenames with spaces
locate <regex>	Similar as '`find`' but using index **-i** Ignore case **--regex** Interprets all patterns as extended regex **-0 | xargs -0 <cmd>** Sends found files to a command as parameters.
updatedb (sudo)	Update locate index
md5sum	Prints md5 sum hash of a file
read	Read single line from standard input **-n 1** Print after reading one character **-s** Do not echo input coming from terminal
shred	Securely remove files
file	Determine file's type
tree	Ls in a tree-like (hierarchical) format
install	Copy files and set attributes
gpg	Decrypt file with password **-c** Encrypt
mktemp	Create a temporary file or directory in '`/tmp`' and returns it's name.
rename	**s/<from>/<to> <files>** Renames multiple files using '`sed`' syntax
rsync	A fast, versatile, remote (and local) file-copying tool **-Hbaz -e ssh** '`<src_dir> <user>@<host>:<dest_dir>`' - 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. **cmp** Compares two files, similar to diff but also for binaries
stat	Displays files status **-c%X** Time of last modification of the file
readlink	**-f** Follow link recursively and print files path
xdg-open	Open file with default application for the file type
dialog	Display dialog box from shell script
watch	Execute command periodically

Archives

dtrx	**<archive>** Universal archive extractor
tar	**xvzf <file>.tar.gz (.tgz)** Decompress and detar **xvjf <file>.tar.bz2** Decompress and detar **-cf <archive>.tar <files>** Compress
unzip	***.zip** Backslash is necessary so that bash doesn't expand the '`*`' **-d <dir>** Extract into directory (create if doesn't exist)

zip	**-r <archive> <dir>** Compress whole directory recursively. **-g <archive> <files>** Add files to existing archive (grow).
unrar	**e** Extract files from rar archive
zcat	Cats gziped file

Terminal Multiplexers

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

Bash

"$x"	ALWAYS PUT DOUBLE QUOTES AROUND VARIABLE!!!!!!!!!!!!!!! All variables in bash are global!!!!!!!
"$*"	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!!!
"$@"	Use this instead!!!!! Will retain arguments as-is, so no args provided will result in no args being passed on. This is in most cases what you want to use for passing on arguments. Google: "$@" is right almost everytime, and $* is wrong almost everytime.
"$#"	Number of arguments
"$1"	First argument
"$0"	Name of the script
$'\n'	String literal with escape sequences (there is a backslash before n) If you want IFS to be a new line (useful with for loop) you need to: '`IFS=$'\n''` - The dollar forces substitution!!!!! Also if you want 'while read line; do...' to preserve leading spaces and tabs, you need to set IFS=""
$?	Exit code of last command (0 - Success)
Ctrl-Z, kill %%	Kill looping bash script

test <expr>	Same as '`[ <expr> ]`'. Returns zero exit status if true. **-n** Is string non empty **-z** Is string empty **-a** And **-o** Or **=** Strings are equal **-nt** File newer than **-ot** Older then **-d** Directory exists **-e** File exists **-f** Is a regular file **-h** Its a symbolic link **-r** Has read permission **-w** Has write permission **-x** Has execute permission
[[<expr>]]	Same as '`[`', but without word splitting and filename expansion. And with additional operators: '`&&`', '`||`', '`<`', '`>`' (lexicographic less, more), and also regular expression matching.
=~	Regex comparison operator: '`[[ "$HOST" =~ ^user.* ]]`'
let <expr>	Executes expression: let a="$b"+2

$(command)	Same as '`command`'
eval <variable>	Execute string as command
$RANDOM	0 - 32767
input='cat'	Getting standard input
-	In place of a file name means standard in or out
set -o vi	Set line editing to vi mode
pushd .	Put current dir on stack
popd	Pop dir from stack
cd -	go to last dir
source <script>	Run script: for example source /etc/profile (same as . <cmd>)
#!/bin/bash	Good practice to insert at beginning of a bash script
export	**PATH="$PATH:<dir>"** Adds new directory to path environment variable.
read	**-p <message>** Prompt for user input
var=${1:-"<default>"}	Setting variable with default value if $1 is empty
getopts	Parse parameters/arguments, builtin
getopt	GNU version is even better then getopts, not a builtin
while read line; do <commands>; done < <file>	Read from file line by line **-r** Do not treat backslashes as escape characters
complete -F <completion_function> <cmd>	Set completion function for command
complete -p <cmd>	Print the completion function for command
compgen -c <pattern>	Print all completions for pattern
help <builtin>	Display information about builtin command
wait	Wait for all background processes to end

Safety

set	**-e**	Exit if any command fails
	-u	Exit if referencing undefined variable

	`-o pipefail` If any command in a pipeline fails, its return code is used as the return code of the whole pipeline
`IFS=$'\n\t'`	Remove space from the default Internal Field Separator

History

`sudo !!`	Run the last command as root
`␣<cmd>`	Execute a command without saving it in the history
`!<cmd>`	Run last command that starts with cmd

Redirections

`<cmd> 2> /dev/null`	Redirect error output to '`null`'
`<cmd> &> /dev/null`	Redirect both standard and error output to '`null`'
`<cmd> >&2`	Write to stderr
`<cmd> 2>&1 | less`	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 **`-u`** Difference
`column`	Columnate text **`-t`** Create a table
`shuf`	Shuffle input lines
`tac`	Concatenate and print files in reverse (reverse '`cat`')
`join`	Join lines of two files on a common field
`colrm`	**`[from [to]]`** Removes columns
`seq <number>`	Output numbers from 1 to number
`ispell, aspell`	Interactive spell checker
`basename <path>`	Strips directory from path **`-s .<suffix>`** Also strip suffix **`-a`** Process multiple filenames
`dirname <path>`	Strip last component from path
`fmt`	Produce roughly uniform line lengths
`fold`	Wrap each input line to fit in specified width

paste	Glue two documents side by side	
sed	**'s///g'**	Substitute every occurrence in line, not just the first one
	's///I'	Ignore case
	-r	Extended syntax, for '+', '?', ... Also you shouldn't escape the parenthesis
	-r 's###e'	Execute match as a command
	-i <file>	Will make changes directly to the file (in place)
	-u	Unbuffered mode (processes input immediately)
	-n l	Print escape sequence (keycode) of a pressed key
expand	Convert tabs to spaces	
	-t <number>	Set number of spaces (default is 8)
	-i	Do not convert tabs after non blanks

Diff

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

Search

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

Convert

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

Editors

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

Network

whois	Info about domain	
host <ip/hostname>	DNS lookup utility	
nslookup	Same interactively	
dig	Same, lot of options	
hostname	Prints/sets computer name, to set it permanently edit '`/etc/hostname`' and '`/etc/hosts`'	
netstat	Displays contents of /proc/net files, status of ports...	
	-r	Show routing table
	-i	Show interfaces
arp	Manipulate the system ARP cache (IP -> mac)	
route	Tool used to display or modify the routing table	
	add default gw <ip>	Change the default gateway should DNS not be configured correctly on your machine, you need to edit '`/etc/resolv.conf`' to make things work
ifconfig eth0	**down/up (sudo)**	Turn network interface on/off
	<ip> netmask <mask> up	Set ip and mask
ifup eth0	Will bring eth0 up if it is currently down.	
ip	**link show**	List network interfaces
	link set dev eth0 up	Bring interface eth0 up or down
	addr show	List addresses of interfaces
	route add default via <ip>	Set default gateway
traceroute, traceroute6, traceroute6.iputils	Traces route	
tracepath, tracepath6	Similar (iputils package)	
mtr	Combines the functionality of the traceroute and ping	
findsmb	List info about machines that respond to SMB name queries - Windows based machines sharing their hard disks	
/etc/services	List of internet services with their port numbers	

NetworkManager	Network management daemon, configuration file is in /etc/NetworkManager/NetworkManager.conf
nm-tool	Prints info
nm-online	Is network connected
nmcli	Command-line tool for controlling NetworkManager
nc	(netcat) It can open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, do port scanning...
ncat	Concatenate and redirect sockets
ethtool eth0	Show status of eth0 **-S** Statistics **-s** Change settings (speed, duplex,...)
ss -tupl	List internet services on a system **-tuo** List active connections to/from system

Wireless

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

Internet

mutt	Mail client
sftp	Secure ftp
sshd	(openssh-server) ssh server deamon, on Windows service is named 'CYGWIN sshd'
/etc/init.d/ssh restart (sudo)	Restart sshd (ssh deamon)
/usr/local/etc/init.d/openssh start (sudo)	Start openssh deamon
ssh-keygen -t rsa -C <email>	Generate rsa key pair, keys are stored in ~/.ssh

ssh	SSH client **<user>@<address> "mpg321 -" < <file>.mp3** Stream audio over ssh **enter ~.** Kill unresponsive session
scp	Securely copy files over network. Example: '`scp <local_file> <user>@<host>:~<remote_file>`'
wget	Download files from WWW **-O** Specify output file **-c** Continue downloading unfinished file. Can also use wildcards, but use single quotes around url. **-r -l1 --no-parent -A.gif** Recursively to the depth of one ignoring references to the parent directory and all gifs.
curl	Similar **-qO -** Writes to standard output + quiet **-i** Urls specified by standard input
transmissioncli	Torrent terminal client **-d** Download limit (kB/s) **-u** Upload limit
youtube-dl	Download from YouTube
noip2	Dynamic dns update client
rdesktop	Remote Desktop Protocol client
tin, nn	Usenet client
nrss	Rss feed reader

Browsers

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

Hacking

nmap	**-sP 192.168.1.1-3** Network scanning **192.168.1.3 -p100-139** Port scanning **-O 192.168.1.3 (sudo)** Scanning os
tcpdump (sudo)	Sniffer (show network traffic) **-l** Buffered output (for piping to less, etc.)
ettercap	Multipurpose sniffer/interceptor/logger for switched LAN (can detect man in the middle, denial of service, DNS spoofing)
driftnet	Picks out and displays images from network traffic
kismet	Wireless 802.11b monitoring tool
aircrack-ng	Wireless WEP/WPA cracking utilities
Cain & Abel	Password recovery tool for Microsoft Operating Systems
dnsniff	Warious hacking tools: **arpspoof** For man in the middle attack **dsniff** Password sniffer for several protocols, ...

`ip link set eth0`	**`promisc on`** Set network interface to promiscuous mode **`multicast off`** Set multicast off

System

`meta, system info`	In cinnamon
`uname -a`	Print system info, kernel version
`cat /etc/issue`	Show name and version of distribution
`init`	Upstart init daemon job configuration
`telinit`	Change system runlevel
`dmesg`	Print the contents of your bootup (startup) messages displayed by the kernel. This is often useful when debugging problems
`getconf -a`	Print all system configuration variables
`nohup <cmd>`	Run a command immune to hangups, runs even after the shell is closed (writes output to nohup.out) **`&>/dev/null &`** Run a command immune to hangups in background, do not save output
`wmctrl`	X Window Manager
`awesome`	Tiled window manager **`-k`** Check configuration script for errors
`busybox <cmd>`	Combines tiny versions of many common UNIX utilities into a single small executable. (1.5 Mb)
`mono`	.NET support

Users

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

File Systems

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

Logging

`last`	When various users have logged in or out. This includes information on when the computer was rebooted.
`lastlog`	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`'.
`rsyslogd`	manages all the logs on your system closelog, openlog, syslog, vsyslog -> library calls that send messages to the system logger
`logger`	Makes entries in the system log
`zeitgeist`	Activity logger

Load

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

Tracing

`auditd`	System call auditing (package)

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

Hardware

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

Terminal

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

Keys/Characters/Fonts

In X

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

X Not Necessary

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

Processes

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

Jobs

`Ctrl z`	Put job into background
`jobs`	Prints currently running job
`bg <job_id>`	Put job in background
`fg <job_id>`	Bring job to the foreground

`%n`	Job number n
`%s`	Job whose command line starts with s
`%%`	Current job
`%-`	Previous job

Scheduled Commands

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

Services

`service`	Allows you to start, stop or restart a service (it runs a script in /etc/init.d folder) `-f sshd` Restart the ssh server `httpd status` Get status of apache `--status-all` 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

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

Multimedia

Audio

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

Bitmap

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

Video

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

Awesome Debian Files

Bash

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

Home

`~/.Xmodmap`	Keyboard map

Bins

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

General Config

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

`/etc/mailcap`	Default programs for extensions
`/etc/passwd`	Users
`/etc/groups`	Groups
`/etc/default`	Boot script parameters that the end user or administrator is likely to change. `/console-setup` Set console (tty) character set, font size, ...
`/etc/fstab`	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).
`/etc/alternatives`	Links to default application versions (here you can change the default Java JDK)
`/etc/issue`	Name and version of distribution
`/etc/fstab`	Automatic mounts are handled by configuring the file
`/etc/sudoers`	Lists of users and the commands they can run with sudo (needs to be edited with visudo command)
`/etc/apt`	`/sources.list` List of places where to look for packages

Services

`/etc/init/, ~/.init/`	`Init` Upstart init daemon job configuration
`/etc/init.d`	Folder with service scripts, that get executed at start and end `/halt` Runs at the end
`/etc/rc<level>.d`	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)
`/etc/rc.local`	Last initialization file executed - Put your commands here

`/etc/init/ssh.conf`	Sshd config
`/etc/ssh/sshd_config`	Sshd config
`/etc/crontab`	System-wide crontab
`/etc/cron.hourly, /etc/cron.daily, ...`	Links to scripts that will execute periodically. Scripts within a cron directory are run alphabetically.
`/etc/rsyslog.conf | rsyslog.d/50-default.conf`	Log conf (need to restart rsyslogd after edit)
`/etc/syslog.conf`	Configuration information for syslogd

Network

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

Proc

Various information about the system.

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

Usr

"Unix System Resources"

`/usr/lib`	`/jvm` Java JREs and JDKs
`/usr/share`	`/man` Man pages

/bash-completion	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"

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

Boot

Kernels.

/boot/grub/menu.lst	Grub configuration file
/etc/default/grub	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

git remote	**update**	Get info about state of remote
	show origin	Print address of the origin
	set-url origin <origin>	'git@github.com:<user>/<repo>.git' - Change the url of origin, ssh key needed; 'https://www.github.com/<user>/<repo>.git' - Same, but withouth key.
git status -uno	Check if everything up-to-date	
git fetch; git checkout <branch>	Checkout remote branch	

Tags

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

Undo

git reset	**--hard HEAD~1**	Delete last commit and all of its changes
	HEAD~1	Delete last commit but keep your changes

Also Useful

git checkout HEAD^ <file>	Retrieve deleted file
git rm --cached <file>	Untrack file without deleting it
git tag -a 0.9.1 -m "Version 0.9.1 release"	Tag latest commit
git tag	List local tags
chown -R <user>:<group> * (sudo)	In .git/objects
git format-patch -1 <sha>	Generate patch file
git rev-parse HEAD	Get sha of head
git revert	
git describe	Print version and hash of HEAD
git log --name-only --author=<name>	Print changed files by commit
git log <file>	Print files history

`git ls-files`	List files
`git show <revision>:<file>`	Take a peek at the older revision of the file
`git diff <commit> <commit>`	Compare two commits
`git stash`	If you want to switch branches, but you don't want to commit your changes yet, you can 'stash' them
`git stash apply`	Apply the changes you stashed
`git stash list`	List all the stashes
`git stash show -p stash@{0}`	Show the diff of most recent stash
`git update-index --chmod=+x <file>`	Change files permissions
`git config -l`	Print repos configuration settings
`git config core.filemode false`	Ignores executable bit of the files

Checkout

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

Github Md Format

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

Bisect

`git bisect start`	
`git bisect bad`	Tell git that current version is bad
`git bisect good v25.0.2`	Tell git the last good version you know about. Now git will checkout a version in between, so you can check it and tell:
`git bisect bad/good`	This will continue until the commit that introduced the bug is found
`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 core.filemode false`	Ignores the filemode changes made by the host system
`git config --global --unset https.proxy`	If problem pulling

Vim

`+<linenum>`	Open file at line number
`alt+<normal mode key>`	Escape, 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
`P`	Paste before cursor
`x`	Delete character
`gF`	Open file under cursor
`K`	Look up word under cursor in man pages
`ctrl+v`	Select visual box (block select)
`:e`	Reload file
`:sav`	Save file as and keep new file open (save as)

Help

`ctrl+]`	Follow link
`ctrl+o`	Go back
`:q`	Exit help

Movement

`e`	End of word
`E`	End of WORD
`W`	Start of WORD
`ge`	End of previous word

`), (`	Sentence
`{, }`	Paragraph
`]], [[`	Section
`:<num>`	Goto line number
`ctrl+o`	Go to previous location
`ctrl+i`	Go to next location
`%`	Jump to matching bracket

Lines

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

Screen Lines

`g0, g$`	Start/end of screen line
`gm`	Middle of screen line
`gk, gj`	Up/down one screen line

Page Up/Down

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

Search

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

Marks

`'"`	Move to position of last edit of file
`` `. ``	Move to last change in file
`` `0 ``	Position where you last exited vim

Insert Mode Commands

`ctrl+h`	Backspace
`ctrl+u`	Delete line
`ctrl+w`	Delete previous word

Macros

`q<x>`	Record actions (macro) into x
`q`	Stop recording macro
`@<x>`	Execute x (macro)

Registers

`"ayy`	Copy line into register a
`"ap`	Paste register a
`:reg`	Access all registers

Set Command

`:set <x>`	Set x
`:set no<x>, <x>!`	Unset x
`:set <x>=value`	Assign x
`:set <x>-=value`	Remove value form <x>
`:set all`	Print all values
`:set <x>?`	Print x

Set Command Operands

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

Edit Commands

`[n] operation [m] motion`	If both n and m are specified then n x m
`c, d, y`	Change, delete, yank

`C, D, Y`	Till the end of line
`cc, dd, yy`	Current line
`cf<x>, df<x>, yf<x>`	Forward up to x
`c), d), y)`	Sentence
`~`	Change case of character
`g~w`	Switch case of a word
`gu, gU`	To lower/upper case
`[p`	Paste but match current indentation
`r`	Replace character
`S`	Substitute entire line
`x,X`	Delete character/delete back
`.`	Repeat last change
`ctrl+a, ctrl+x`	Increment/decrement number under cursor

Automatic Linebrake (Wrap)

`gq`	Formats (wraps) selected text
`gqq`	Format current line
`:set tw=72`	Set text width

Advanced

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

Colorscheme

`:colorscheme`	`darkblue, torte`	Nice, darker
	`slate, default`	Less contrast
`:highlight Normal ctermbg=grey`	Set light background	

Spellcheck

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

Tabs

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

Splits

`:vsp`	Split vertically
`ctrl-w, direction`	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 <path>:<path>...`	Tell Java where libraries are located
`java -Xmx6g myprogram`	Reserve 6 giga for process

	`-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 _JAVA_OPTIONS=-Xmx1000m`	Set heap space globally	
`appletviewer <page>.html`	Run Java applet	
`jps -lvm`	List java processes	
`jmap -histo:live <pid>`	Memory map	
`jvisualvm`	Profiler	
`jar cvfe "bla.jar" <main_class> *.class`	Create executable jar	
`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.d/webupd8team-java.list
echo "deb-src http://ppa.launchpad.net/webupd8team/java/ubuntu precise main" | sudo tee -a /etc/apt/sources.list.d/webupd8team-java.list
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

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

Mount Iso

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

Eclipse

`ctrl+1`	Quick fix
`alt+shift+s`	Source submenu
`ctrl+F7, esc`	Close pop-up console window
`ctrl+7, ctrl+/`	Toggle comment
`F3`	Goto definition
`objectaid`	UML plugin

Wine

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

Cygwin

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

Golang

`go`	`build`	Compile packages and dependencies
	`clean`	Remove object files
	`env`	Print Go environment information
	`fix`	Run go tool fix on packages
	`fmt`	Run gofmt on package sources
	`get`	Download and install packages and dependencies (first you need to set GOPATH to dir where packages will get downloaded)
	`install`	Compile and install packages and dependencies
	`list`	List packages
	`run`	Compile and run Go program
	`test`	Test packages
	`tool`	Run specified go tool
	`version`	Print Go version
	`vet`	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 repackage (on Debian use genisoimage). -c passes the name of the file that will be created
cd /var/tmp/linux
mkisofs -o ../your-new.iso -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -J -R -V Your I
```

Gdb

gdb <cmd>	Start gdb
run <arguments>	Start debugging
run < <file>	Run with piped input
up	Follow trace
print <variable>	Print variable

Coredump

ulimit -c unlimited	Set core file limit to unlimited
gdb <cmd> core	Debug core file with gdb

Chrome

F6, ctrl+l, alt+d	Go to address bar

January 30, 2019 / Jure Šorn