

LINUX LIFE IN THE COMMAND LINE : NAVIGATION

NAVIGATING THE FILESYSTEM ON YOUR SHINY NEW LINUX INSTALL CAN BE DAUNTING AT FIRST HOWEVER WITH A BIT OF PRACTICE THESE COMMANDS WILL ALLOW YOU TO ZIP FROM ONE END OF THE OPERATING SYSTEM TO THE OTHER WITH BLAZING FAST SPEED. THE COMMAND LINE ALLOWS YOU TO TAKE CONTROL AND DO FAR MORE THAN THE DROPPED OWN MENUS IN YOUR GUI COULD EVER ALLOW. BUCKLE IN AND LET'S TAKE LINUX FOR A RIDE.

ls - List directory contents of the current working directory
ls /bin - List files in /bin directory (replace /bin with any other directory name)
ls -l - List files in working directory in a long listing
ls -la .. - Lists all files in parent directory (even hidden ones), from parent of working directory in long format
ls -alh - Formatted long listing with hidden files
cd DIR - Change the current directory to the DIR directory (replace DIR with the directory you want)
cd ~ - Change current directory to \$HOME
cd / - Change the current directory to the root directory
cd .. - Change to the parent of the current directory
hash - Remember and/or display program locations
pwd - Show name of current working directory
return - return from a shell function
find - Search for files in a directory hierarchy
locate - Find files by name

Where am I?

Everything in Linux is a file, from your audio drivers to your empty crypto-wallet program. This allows you to manipulate and change things in very imaginative ways, on the other hand it also makes navigation through the filesystem confusing at times. Combining the navigation commands with the understanding of the basic filesystem layout will make you a Linux wizard in no time at all. Nearly all Linux filesystems start out with the following directory tree for you to build on

/bin: contains user-executable programs and files.

/boot: contains the bootloader and the files required to boot the computer.

/etc: stores configuration files for the system.

/home: contains the home directories for the various user accounts. Each user has a subdirectory located at /home/userid. While a user is logged in, their home directory is aliased as the ~ directory.

/lib: shared system library files are located in this directory.

/root: this is the home directory for the root user. This is not the same as the root, (or /) directory.

/usr: contains shared binary, library, and documentation files for all users.

/var: data files are typically stored here.

LINUX LIFE IN THE COMMAND LINE : MANIPULATING FILES

Moving Renaming and Manipulating files seems like a relatively simple activity until your entire desktop is covered with windows and your GUI is bogging down. Using these commands you can manipulate files from all over your system right in the command line. The command line lets you manipulate files in amazing ways at lightning speeds.

```
» mkdir -p DIR – create directory named DIR
» rm FILE – Remove FILE
» rm -r DIR – Remove DIR and all its recursive contents
» rm -f FILE – Remove FILE by forcefully
» rm -rf DIR – Remove directory DIR with force and recursively
» cp FILE1 FILE2 – Copy contents of FILE1 to FILE2
» mv FILE1 FILE2 – Rename (or move) FILE1 to FILE2
» ln -s FILE link – Create a symbolic link (shortcut) “link” to FILE
» touch FILE – Create FILE or change timestamps of FILE
» cmd > FILE – Standard output (stdout) of cmd to FILE
» cmd >> FILE – Append stdout to FILE
» cat FILE1 FILE2 – Concatenate FILE1 and FILE2 and print to stdout
» less FILE – Output contents of FILE in order to read
» more FILE – page at a time version of less
» head FILE – output first 10 lines of FILE
» tail FILE – output last 10 lines of FILE
» tail -f FILE – Output content of FILE as it grows
» sed -i 's/sit/stay/g' file.txt – replace all instances of sit with stay in
file.txt
» diff – Finds the difference between two files
» cmp – Method of checking if two files are identical
» comm – Combines the functions of diff and cmp
» sort – Sorts the content of a file while outputting
» export – Export environment variables
» zip – compresses files in Zip format
» unzip – opens and extracts files compressed in Zip format
» mount – Mounts file systems such as removable drives
» chmod – Change permissions for files or folders
» chown – Change or grant ownership of files or folder
» chcon – Change file security context
» chgrp – Change file group ownership
» dd – Copies and converts a file
```

» ascii - ASCII character set encoded in octal, decimal, and hex
» aspell - Interactive spell checker
» awk - Pattern scanning and processing language
» bash - GNU Bourne-Again SHell
» bc - An arbitrary precision calculator language
» cal - Display a calendar
» curl - Transfer a URL
» file - Determine file type
» find - Search for files in a directory hierarchy
» gedit - Text editor for the GNOME Desktop
» grep - Print lines matching a pattern
» gunzip - Compress or expand files
» gzip - Compress or expand files
» less - Opposite of more
» locate - Find files by name
» look - Display lines beginning with a given string
» lpr - Print files
» man - Interface to the system reference manuals
» mc - Visual shell for Unix-like systems.
» mcedit - Internal file editor of GNU Midnight Commander.
» more - File perusal filter for crt viewing
» mount - Mount a filesystem
» nano - Nano's ANOther editor, inspired by Pico
» ps - Report a snapshot of the current processes.
» scp - OpenSSH secure file copy
» sed - Stream editor for filtering and transforming text
» sftp - OpenSSH secure file transfer
» ssh - OpenSSH remote login client
» sqlite3 - Command line interface for SQLite version 3
» su - Run a command with substitute user and group ID
» sudo - Execute a command as another user
» tar - An archiving utility
» tmux - Terminal multiplexer
» top - Display Linux processes
» tput - Initialize a terminal or query terminfo database
» uptime - Tell how long the system has been running.
» vim - Vi IMproved, a programmer's text editor
» wget - Non-interactive network downloader.
» which - Shows the full path of (shell) commands
» zcat - Compress or expand files
» zless - File perusal filter for crt viewing of compressed text

`dir` – lists file in working directory (same as `ls`) sorted in columns and vertically
`direcolor`s – Set up color for `ls`
`install` – Copies files and set attributes
`mkfifo` – Makes named pipes (FIFOs)
`mknod` – Makes block or character special files
`mkttemp` – Makes temporary file or directory
`realpath` – Returns the resolved absolute or relative path for a file
`rmdir` – Removes empty directories
`shred` – Overwrites a file to hide the contents, optionally deletes it
`sync` – Flushes file system buffers
`truncate` – Shrink or extend the size of a file to a specified size
`vdir` – Same as `ls` but listed in long format
`file` – Use to determine a file type
`find` – Search for files in directory hierarchy
`gedit` – Text editor for GNOME desktop
`grep` – Print lines matching a pattern
`gunzip` – expand compressed files
`gzip` – compress files
`locate` – Find files by name
`look` – Display lines beginning with a given string
`lpr` – Print files
`mcedit` – File editor built into Midnight Commander
`sed` – Editor for filtering and editing text
`vim` – Text editor for programming
* - Wildcard to match any characters
? - Wildcard to match any single characters
g* - Matches all filenames that begin with the character “g”
b*.txt – Matches all filenames that begin with “b” and end with “.txt”
Data??? - Matches any filename that begins with “Data” followed by 3 characters
[abc]* - Matches any filename that begins with “a”, “b” or “c”
[[upper:]]* - Matches any filename that begins with an upper case letter
*[![:lower:]] - Matches any filename that does not end with a lowercase letter
`sort` – sorts input
`uniq` – removes duplicate lines of data from a sorted stream
`fmt` – Reads standard input text and outputs formatted text
`pr` – Splits input data into pages with page breaks, headers and footers for printing
`tr` – Translates characters ie; upper/lowercase conversions, changing line termination

LINUX LIFE IN THE COMMAND LINE : NETWORKING

NETWORKING COMPUTERS IN THE HOME IS ESSENTIAL AND USING LINUX YOU CAN CREATE A HOME NETWORK WITH HARDWARE COMPUTERS VIRTUAL MACHINES DOCKER CONTAINERS MEDIA SERVERS PI HOLE ADBLOCKERS TVs LIGHTS IF CAN ALL BE CONNECTED TOGETHER AND CONTROLLED WITH LINUX USING THE FOLLOWING COMMANDS YOU WILL BE ABLE TO BUILD AND CONNECT A NETWORK OF ALL YOUR MACHINES AND ORCHESTRATE THEM ALL

- » ssh username@hostmachine - Connect to host as the username
 - » ssh -p Port username@hostmachine - Connect using port p
 - » ssh -D Port username@hostmachine - Connect and use blind port
 - » ./configure - Configure the source file
 - » make - Compile the source code
 - » make install - Install the compiled source code
 - » ping host - Ping host 'host'
 - » whois domain - Get whois info for given domain
 - » dig domain - Get the DNS record for given domain
 - » dig -x host - Reverse lookup oh host
 - » wget file - Download specific filename
 - » wget -c file - Continue a stopped download
 - » Wget -I url - Recursively download files from a url
 - » whoami - Shows your active username
 - » df - Displays disk filesystem information
 - » ifconfig - Displays network interfaces and IP addresses
 - » traceroute - Trace all the network hops to reach the destination
 - » ufw - Basic firewall command
 - » iptables - Foundational firewall for all other firewall utilities to interface with
 - » scp - OpenSSH secure file copy
 - » sftp - OpenSSH secure file transfer
 - » ip - same as ifconfig but allows for second arguments (ie; ip link show)
 - » ping - used to see if host is online
 - » iPerf - used to analyze and measure network performance between two hosts
 - » traceroute - shows the route packets take to reach destination
 - » tcpdump - used to examine all packets sent to or from a particular host
 - » netstat - Used to examine network connections, routing tables, and various network settings and stats
 - » ss - Used to display running services
 - » telnet - Connects destination host and port via telnet protocol
 - » nslookup - Used to query domain name servers and resolving IP
 - » dhclient - changes dhcp IP Address lease
 - » ethtool - allows configuration of network interfaces
 - » mtr - Allows a real time lo-cli estok at network performance
 - » iPerf - Shows detailed info about bandwidth of specific servers internal or external
 - » curl - Transfers data over network
- » ifplugstatus - Detects the link status of a local Ethernet device
 - » iftop - Monitors stats related to bandwidth
 - » firewalld - Used to configure rules of a firewall
 - » speedtest-cli - used to check internet speeds
 - » netsniff-ng - Multiple network configuration tools
 - » netplan - Network configuration
 - » snort - Network Intrusion Detection and Prevention System
 - » airownd - Ethernet activity monitoring
 - » bmon - Bandwidth Monitoring
 - » darkstat - Captures network traffic and usage stats
 - » dstat - replacement for vmstat, iostat, mpstat, netstat and ifstat.
 - » ethtool - utility for controlling network drivers and hardware.
 - » gated - gateway routing daemon.
 - » host - DNS lookup utility.
 - » hping - TCP/IP packet assembler/analyzer.
 - » ibmonitor - shows bandwidth and total data transferred.
 - » ifstat - report network interfaces bandwidth.
 - » iftop - display bandwidth usage.
 - » iproute2 - collection of utilities for controlling TCP/IP.
 - » iptables - take control of network traffic.
 - » IPtraf - An IP Network Monitor.
 - » iutils - set of small useful utilities for Linux networking.
 - » iw - a new nl80211 based CLI configuration utility for wireless devices.
 - » jwhois (whois) - client for the whois service.
 - » lsof -i" - reveal information about your network sockets.
 - » mtr - network diagnostic tool.
 - » net-tools - utilities include: arp, hostname, ifconfig, netstat, rarp, route, pipconfig, slattach, mii-tool, iptunnel and ipmaddr.
 - » nc - improved re-implementation of the venerable netcat.
 - » netcat - networking utility for reading/writing network connections.
 - » netlog - a small 'net top' tool.
 - » NetPerf - Network bandwidth Testing.
 - » netplan - Netplan is a utility for easily configuring networking on a linux system.
 - » netsniff-ng - Swiss army knife for daily Linux network plumbing.
 - » netwatch - monitoring Network Connections.
 - » ngrep - grep applied to the network layer.

- » nmap - network discovery and security auditing.
- » nmcli - a command-line tool for controlling NetworkManager and reporting network status.
- » nmcli - provides a text interface to configure networking by controlling NetworkManager.
- » nslookup - query Internet name servers interactively.
- » ping - send icmp echo_request to network hosts.
- » route - show / manipulate the IP routing table.
- » slurm - network load monitor.
- » snort - Network Intrusion Detection and Prevention System.
- » smokeping - keeps track of your network latency.
- » socat - establishes two bidirectional byte streams and transfers data between them..
- » speedometer - Measure and display the rate of data across a network.
- » speedtest-cli - test internet bandwidth using speedtest.net
- » ss - utility to investigate sockets.
- » ssh - secure system administration and file transfers over insecure networks.
- » tcpdump - command-line packet analyzer.
- » tcptrack - Displays information about tcp connections on a network interface.
- » telnet - user interface to the TELNET protocol.
- » tracepath - very similar function to traceroute.
- » traceroute - print the route packets trace to network host.
- » vnStat - network traffic monitor.
- » websockat - Connection forwarder from/to web sockets to/from usual sockets, in style of socat.
- » wget - retrieving files using HTTP, HTTPS, FTP and FTPS.
- » Wireless Tools for Linux - includes iwconfig, iwlist, iwspy, iwpriv and ifrename.
- » Wireshark - network protocol analyzer.

Many of these command line tools are not built into your Linux OS and will need to be installed. Tools such as Nmap and Wireshark are very powerful and have entire books written on using them. Networking with Linux systems gives you the ability to build worlds and interconnect machines to accomplish nearly any task you can imagine. One of the greatest tools to build into your networking arsenal is creating scripts in Bash and Python to automate tasks you find yourself repeating often. In order to run your Python automation scripts, you must install Python 3 to your system, and all needed libraries to support what you write. Useful tools for this include installing and setting up an IDE such as Codium to develop and test your scripts in a safe environment that will not cause any negative effects to your system. You can build bash scripts in more basic programs such as VIM just remember that your bash scripts must be executable and have to start with what is called a shebang (#!/bin/bash) to let the command line know that you are running a bash script. Scripts will allow you to run multiple commands and programs in a specified order all with a simple command to execute the script, but what if you want to just combine a couple commands and run them with one simple easy to remember command? Linux will do that with the simple use of aliases. You can use an alias on a temporary basis (only as long as you have the shell open) with just the command, however to make it a permanent change to your system your system you will need to add it to the bashrc file found at /home/(yourusername)~/.bashrc where you can add your desired aliases to the end of the file. An example of creating an alias for working with your iptables is as follows:

```
alias iptlist=sudo /sbin/iptables -L -n -v --line-numbers #this will display all lines of your current iptables
alias iptlistin=sudo /sbin/iptables -L INPUT -n -v --line-numbers #this will display all your INCOMING rules in iptables
alias iptlistout=sudo /sbin/iptables -L OUTPUT -n -v --line-numbers #this will display all your OUTGOING rules in iptables
```

LINUX LIFE IN THE COMMAND LINE : FUN WITH FUNCTIONS

NETWORKING MACHINES AND MOVING FILES IS GREAT BUT WOULD NOT BE USEFUL WITHOUT BEING ABLE TO PROCESS FUNCTIONS AND ACTUALLY DO THINGS WITH THE CONTENTS OF THE FILES. THESE FUNCTION COMMANDS PUT THE POWER OF PROCESSING AT YOUR FINGERTIPS FOR GETTING WORK DONE AND FOR HAVING FUN RIGHT FROM YOUR CLI.

Testing and using all these commands can be a little tiresome, so lets start with a little bit of silly commands to have some fun:

- » **sl** - Needs to be installed via your program manager of choice (spt, yum, pacman). Once installed typing the sl command will bring you an ascii visitor in the form of a steam locomotive. Not a bad reward for mis-typing the ls command!
- » **fortune** - Install fortune and run the command to see a funny fortune.
- » **rev** - This command reverses any string following the command
- » **cowsay** - Will illustrate a cow saying the string following the command in ascii
- » **Xcowsay** - A graphical version of cowsay
- » **couthink** - cowsay with a thought bubble instead of a speech bubble
- » **yes** - repeats a string following the command until ctrl+c is used to stop the loop
- » **Cmatrix** - Installing and running this command will start animation of letters like in The Matrix
- » **oneko** - Install and run this command and your mouse pointer will change to a mouse who is eventually chased by a cat. If the cat catches the mouse, it will go to sleep. There are lots of options to make the game more interactive as well.
- » **espeak** - Install and run this command to read any string following the command through the audio driver
- » **aafire** - Use this command to start an animated fire on your screen
- » **bb** - This command is useful if you want to see an ascii lava lamp
- » **man** - Install funny-manpages (some are adult themed) then run the command with one of the following words to read grown up manuals for the word you used. baby celibacy condom date echo flame flog gong grope, egrope party rescrog rm rtfm tm uubp woman (undocumented) xkill xlart sex strfry

Now that the silly stuff is out of the way, lets get to some usefull commands to get things done!

- » **((...))** - Evaluate an arithmetic expression
- » **. . .** - Execute commands from a file in the current shell
- » **:** - Null command, while not used it does have effects on other commands, used to truncate a command
- » **[]** - Evaluate a conditional expression
- » **[[...]]** - Execute a conditional command
- » **alias** - Define an/or display aliases
- » **bind** - Set Readline key bindings and variables
- » **break** - Exit for, while, or until loops
- » **caller** - Return the context of the current subroutine call
- » **case** - Execute commands based on pattern matching
- » **compgen** - Display possible completions depending on the options
- » **complete** - Modify or display completion options

- » **dirs** - Display directory stack
- » **disown** - Remove jobs from current shell
- » **echo** - Write arguments to the standard output
- » **eval** - Execute arguments as a shell command
- » **exec** - Replace the shell with the given command
- » **export** - Set export attribute for shell variables
- » **false** - Return an unsuccessful result
- » **fc** - Display or execute commands from the history list
- » **function** - Define shell function
- » **getopts** - Parse option arguments
- » **hash** - Remember or display program locations
- » **help** - Display info about builtin commands
- » **history** - Display and/or manipulate the history list
- » **if** - Execute commands based on conditional
- » **jobs** - Display status of jobs
- » **let** - Evaluate arithmetic expressions
- » **local** - Define local variables
- » **logout** - Exit a login shell
- » **popd** - Remove directories from stack
- » **pushd** - Add directories to stack
- » **read** - Read a line from the input and split it into fields
- » **readarray** - Read lines from a file into an array variable
- » **readonly** - Mark shell variables as unchangeable
- » **return** - Return from a shell function
- » **select** - Select words from a list and execute commands
- » **set** - Set or unset values of shell options and positional parameters
- » **shift** - Shift positional parameters
- » **shopt** - Set and unset shell options
- » **source** - Execute commands from a file in the current shell
- » **suspend** - Suspend shell execution
- » **test** - Evaluate conditional expressions
- » **trap** - Trap signals and other events
- » **true** - return a successful result
- » **type** - Display information about a command type
- » **typeset** - Set variable values and attributes
- » **ulimit** - Modify shell resource limit
- » **umask** - Display or set file mode mask
- » **unalias** - Remove each NAME from the list of defined aliases
- » **unset** - nset values and attributes of shell variables and functions
- » **until** - Execute commands as long as a test does not succeed
- » **variables** - Common shell variable names and usage
- » **wait** - Wait for job completion and return exit status
- » **while** - Execute commands as long as a test succeeds

- » **{ ... }** - Group commands as a unit
- » **arch** - Prints hardware name (same as command uname -m)
- » **basename** - Removes the path prefix from a given pathname
- » **chroot** - Changes the root directory
- » **date** - Prints or sets the system date and time
- » **dirname** - Strips non-directory suffix from the file name
- » **expr** - Evaluates expressions
- » **factor** - Factors numbers
- » **false** - Does nothing, but exits unsuccessfully
- » **groups** - Prints the groups of which the user is a member
- » **hostid** - Prints the numeric identifier for the current host
- » **id** - Prints real or effective UID and GID
- » **logname** - Print the user's login name
- » **nice** - Modifies scheduling priority
- » **nohup** - Allows a command to continue running after logging out
- » **nproc** - Queries the number of active processors
- » **pathchk** - Checks whether file names are valid or portable
- » **printenv** - Prints environmental variables
- » **printf** - Formats and prints data
- » **runcon** - Run command with specified security context
- » **seq** - Prints a sequence of numbers
- » **sleep** - Delays for a specified amount of time
- » **stat** - Returns data about an inode
- » **stdbuf** - Controls buffering for commands that use stdio
- » **stty** - Changes and prints terminal line settings
- » **tee** - Sends output to multiple files
- » **test** - Evaluates an expression
- » **timeout** - Run a command with a time limit
- » **true** - Does nothing, exits successfully
- » **tty** - Prints terminal name
- » **uname** - Prints system info
- » **unlink** - Removes specified file using the unlink function
- » **uptime** - Displays how long the system has been running
- » **users** - Prints the user names of users currently logged into the current host
- » **who** - Prints list of all users currently logged in
- » **whoami** - Prints the effective user ID

