Linux Command Line for Beginners

Commands, Syntax, and Description in Ubuntu Terminal

Compiled by: drmumair

These handouts are structured on an online course “Learning Linux Command Line by Scott Simpson” linked HERE. The text is not reviewed/proofread, errors and omissions excepted!!!

Description	**Command**	**Syntax & #Comments**
Understanding How Commands Are Structured		
Commands Examples	ls, du, cat, df, grep …	**i.** Command **ii.** Options **iii.** Arguments
List Directory Contents	ls	ls -lah usr/bin ls -l-a-h usr/bin ls #--Long_Syntax usr/bin #Double dash “--" with long syntax
Sort Lines Of Text Files	sort	sort -u users.text
Print Lines That Match Patterns	grep	grep -i “needle” haystack #“needle” is an argument to the option -i
Finding Help For Commands		
Manual Pages	man	man ls
Manual Pages Search	/	/search term
Help	--help	ls --help #Option to Command
Search Manual Page Names And Descriptions	apropos	apropos list apropos “search for files” #apropos will search the exact match for phrase in “ ”
Helpful Keyboard Shortcuts In The Terminal		
Tab Completion 1X	⇥ 1X	Do + tab_1x #File or directory name
Tab Completion 2X	⇥ 2X	apr + tab_2x

		#Command completion or multiple command suggestions
Clear Terminal	clear	#Clear the terminal screen
Move Cursor	^ + A (^ Caret)	#Move to the beginning of line
Move Cursor	^ + E	#Move to the end of line
Move Cursor	^ + →	#Move one word forward
Move Cursor	^ + ←	#Move one word backward
Deletion	^ + U	#Deletes from cursor position to the beginning of line
Deletion	^ + K	#Deletes from cursor position to the end of line
Copy	^ + ⇧ + C (⇧ Shift)	#Copies selection
Paste	^ + ⇧ + V	#Paste selection
Recall Command	↑	#Recall previous command
Recall Command	↑ ↓	#Scroll among command history
Command History	^ + R	#Search command history
Abandon Command	^ + C	#Abandon a command you don’t want to run
Escape Character	\	Exercise\ Files OR “Exercise Files” #Escape the Space
The Linux File System		
Determine File Type	file	file Documents
Display File Or File System Status	stat	stat Documents
File System Root	/	#The highest level of the organizational hierarchy represented by /
Home Directory	~ (Tilde)	#Represents each user’s home directory i.e., home/umair/Documents OR ~/Documents
Directories	bin, sbin	#Contain programs the system relies on

Directories	lib	#Contain shared libraries and modules
Directories	mnt	#Local or network file systems are mounted to overall file system
Directories	dev	#System keeps references to all of the hardware
Directories	proc	#Contain references to processes that are running on the system
Directories	sys	#Holds files representing different kernel parameters and system information
Understanding File Paths		
Absolute Path	/	/home/umair/ Documents #Starting with the file system root "/"
Absolute Path	~	~/Documents #~ represents absolute path as it expands to "/home/user"
Relative Path	..	#Represents the parent directory
Relative path	.	#Represents the current directory
Navigating The File System		
Change Directory	cd	cd Documents/Exercise\ Files
Print Working Directory	pwd	pwd
List Subdirectories Recursively	-R -- recursive	ls -R Exercise\ Files
Move To Parent Directory	..	cd Documents/..
Stepping Back Parent Directories	..	../../finance/documents #Moving two setups up the in the directory structure/hierarchy.
Switching Back and Forth Between Two Directories	-	cd -
Home Directory	cd	cd
Exploring The Output Of The ls Command		

Do not Print Group Names	ls	ls -G
ls Output Description Column 1	ls	ls -l #d -> directory #l -> link #- -> file #Dark blue -> directories #Light blue -> links
ls Output Description Column 2	ls	#File permissions: user-group-others -> rwx-rwx-rwx (rwx read write execute)
ls Output Description Column 3	ls	#Owner of the file
ls Output Description Column 4	ls	ls -lh #Size of the file – human readable format M, G etc.
ls Output Description Column 5	ls	#Date and time on with the file is modified
ls Output Description Column 5	ls	#Name of file or link
Create And Remove Directories		
Create A New Directory	mkdir	mkdir New\ Folder
Create A New Directory Inside An Existing Directory	mkdir	mkdir Documents/New\ Folder
Create A New Directory Inside A New Directory	mkdir	mkdir -p departments/legal/contracts #-p option make parent directories as needed
Remove Directory – Empty	rmdir	rmdir departments/legal/contracts #This will remove contracts directory if the directory is empty
Remove Directory And Its Parents	rmdir	rmdir -p a/b/c #This will remove a, b, and c directories at once
Remove Files or Directories – With Contents	rm	rm -R departments #-r OR -R removes directories and their contents recursively

Copy, Move, And Delete Files And Directories		
Copy – Same Location	cp	cp poems.txt poems2.txt
Copy – Different Location	cp	cp simple_data.txt departments/hr
Copy And Rename – Different Location	cp	cp simple_data.txt departments/hr/simple_data2.txt
Move – From Current Directory To Other	mv	mv poems2.txt departments/marketing
Move – From Other Directory to Current	mv	mv departments/marketing/poems2.txt . #“.” Represents current working directory
Rename	mv	mv Exercise\ Files/poems2.txt Exercise\ Files/literature.txt #Renaming from poems2.txt to literature.txt
Move – Multiple Files Using Wildcard *	mv	mv *.txt departments/marketing #This will move all the .txt files in current directory to marketing directory
Move – Multiple Files Using Wildcard *	mv	mv departments/marketing/* . #This will move all the contents of marketing directory to the current working directory
Remove Files	rm	rm literature.txt #Be careful!!! No undo!!!
Remove – Multiple Files Using Wildcard ?	rm	rm poems?.txt #This will remove all the files with any character at “?” but not poems.txt
Remove Files or Directories – With Contents	rm	rm -R departments/customer\ service #-r OR -R removes directories and their contents recursively
Find Files From The Command Line		
Search Files	find	**i.** Command **ii.** Directory **iii.** Test For Find **iv.** Matching Pattern man find find . -name “poe*” find . -name “do*”

		#This will search the files or directories names with poe* or do* (“poe” or “do” with any number of characters) in their names in the current directory
Search Files – In A Specific Directory	find	find ~/Documents -name “*d” #This will search files/directories with letter “d” in their names in Documents Directory
Understand User Roles sudo (Set User / Switch User / Substitute User)		
Substitute User	su	su umair #Switch users
Borrow Superuser’s OR Root’s Powers	sudo	ls /root #Access denied!!! sudo ls /root #Command run with superuser’s privileges. There is no undo on deleting or changing the files. Whatever users do as root is final. So be careful!!!
Passwords		#Will display no placeholders. Three (3) chances for typing correct password.
Grace Period		ls /root #The command will work as after logging in as superuser the system will not ask for password for some time even if we use root’s privileges
Giving Up Root’s Privileges	sudo	sudo -k
Completely Switch Over To Using Root User	sudo	sudo -s #Log into root’s shell. The “$” at prompt will change to “#” or “£” sign as a visual cue that you working as root
Switch Back to Normal user		exit
Understanding File Permissions		

a. **Octal Mode**

	Read (4)	**Write (2)**	**Execute (1)**	**Result**
User	r	w	x	7
Group	r	-	x	5

Others	r	-	-	4

Octal Value	**Mode**	**Octal Value**	**Mode**
0	---	4	r--
1	--x	5	r-x
2	-w-	6	rw-
3	-wx	7	rwx

2. Symbolic Mode

	Read I	**Write (w)**	**Execute (x)**	**Mode**
User (u)	+	+	+	u+rwx
Group (g)	=			g=r
Others (o)	-			o-rwx
All (a)	=	=	=	a=rwx

Operators: + adds permissions; - removes permissions; = resets permissions to match new mode and removes previous mode

Original Mode	**Symbolic Value**	**Result**
rw-r—r--	+x #No prefix -> adds to all	rwxr-xr-x
rwxrwxrwx	g=w, o=r	rwx-w-r--
rwxrw-----	g+w, o+r	rwxrw-r--
rwxrwxrwx	a-x	rw-rw-rw-

Octal Vs Symbolic Options

Octal	**Symbolic**
754	u=rwx, g=rx, o=r
755	u=rwx, g=rx, o=rx
644	u=rw, g=r, o=r

Modify File Permissions

Look For File Permissions	ls, stat	ls -l test.sh stat test.sh #test.sh is a small bash script. This is an executable file i.e., the file can be run as a program without having to be loaded into another program first.
Change File Mod Bits – Symbolic Mode	chmod	chmod -x text.sh OR chmod a-x test,sh #This will remove ‘execute’ permission from user, group, and others.
Change File Mod Bits – Octal Mode	chmod	chmod 644 test.sh #This will set “u=rw, g=r, o=r” permissions for test.sh

Change File Mod Bits	chmod	chmod u-r test.sh chmod 244 test.sh chmod 755 test.sh #755, 644, or 700 are most often used file permissions
Look Into File Contents	cat	cat test.sh
Create Blank File	touch	touch new\ file stat new\ file chmod +x #Touch command change file timestamps
Change File Owner	chown	sudo chown root test.sh sudo chown umair test.sh man chown #This will change the test.sh ownership to root and then back to umair
Change File Group	chgrp	sudo chgrp umair test.sh
Create Hard And Symbolic Links		
Inode		#The inode (index node) is a data structure in a Unix-style file system that describes a file-system object such as a file or a directory
Soft Link Or Symbolic Link Or Symlink	ln	ln -s poems.txt writing.txt cat writing.txt ls -l #Symbolic links points to a file on disk. The first command creates writing.txt symbolic link to the file poems.txt. This is represented by “->” in cat writing.txt and editing writing.txt will edit the original file poems.txt #The link created is relative if we move this link somewhere else or the original file the link will break. If we create a symlink using absolute path the link can be moved anywhere else but will break if the original file is moved
Hard Link	ln	ln poems.txt words.txt cat words.txt ls -l #Hard link points to a specific data on the disk. Every file is a hard link to the data that makes up the file. A hard link will

		appear as a regular file on the system, it’s like a room with multiple doors
The Unix Philosophy #The tool or program should do one thing and do it well. We must have specialized tools configured to work in incredible number of ways – pipelines.		
Use Pipes To Connect Commands Together		
Pipes	| (Pipe or Bar) OR ¦ (Broken bar)	cat users.txt | sort -u | so on… #Commands are little processing units that do one particular thing and pipes are connections between those units. Output of one command is the input of next command in pipeline. echo “Hello” | wc echo “Hello World From The Command Line” | wc #The wc (word count) command counts invisible character at the end of line called “a new line”
View Text Files With cat, head, tail, and less		
Concatenate Files And Print Out The Standard Output	cat	cat poems.txt #This is used to stick together two or more files but often used to print out the contents of file on screen. Also helpful in get the contents of a text file into a series of piped commands
Output The First Part Of Files	head	head poems.txt head -n5 poems.txt #This will output the first 10 and 5 lines of poems.txt, respectively
Output The Last Part of Files	tail	tail poems.txt tail -n3 poems.txt #This will output the last 10 and 3 lines of poems.txt, respectively
Piping Commands	cat, tail	cat poems.txt | cat -n cat poems.txt | cat -n | tail -n8 cat poems.txt | tail-n5 | cat -n #Fist command pipeline will output the contents of poems.txt with numbered lines. #Second command pipeline will output the contents the contents of first command but last 8 lines.

		#The third command pipeline will output the last 5 five lines of poems.txt and will number than starting from 1. Order matter in pipes!!!
View Longer Text Files	less (opposite to more)	less poems.txt cat poems.txt | less #less provides minimal interface to move around the file. ↑ ↓ to scroll up and down, ↩ move down one line at a time, and ⎵ to go down one screen at a time. Use E, B, and F keys to navigate forwards and backwards. Press H for help and Q to quit
Search For Text In Files And Streams With grep #tText streams are divided into lines which are terminated by newline ('\n') characters.		
Print Lines That Match Patterns – Case Sensitive	grep	grep “the” poems.txt grep -n “the” poems.txt man grep #grep searches a specific character, explicit group of characters, or regular expressions (Regexes). -n option will number the output lines. “the” is argument to -n option. This is helpful if you are looking through logs, etc.
Print Lines That Match Patterns – Case Insensitive	grep	grep -In “The” poems.txt #This command will output “the” and “The” searches with line numbers. -I option process a binary file as if it did not contain matching data
Omit Lines That Match Patterns	grep	grep -vi “the” poems.txt #-v option will omit the lines that contain “the” word and is case sensitive. Use -i for case insensitive
Prints Lines – Regular Expressions (Regexes)	grep	grep -e “[hijk]” poems.txt grep -e “[HIJK]” poems.txt grep -ien “[hijk]” poems.txt #-e option use PATTERNS as the patterns. #This command is a regular expression notation for the occurrence of lowercase, uppercase, and case insensitive with line numbering occurrence of “hijk” letters
		grep -E “\w{6,}” poems.txt

		#-E option interpret PATTERNS as extended regular expressions. #This command will output all of the text strings with six characters or longer
Manipulate Text With awk, sed, And sort #Reaching in, extracting data, and presenting it in different ways.		
gawk – Pattern Scanning And Processing Language	awk	awk '{print $2}' simple_data.txt cat simple_data.txt awk '{print $2 "\t" $1}' simple_data.txt awk '{print $2 "\t" $1}' simple_data.txt | sort-n #awk is used for pulling data out of a file according to a rule. To use awk we write a program that considers data in terms of delimiters or field separators i.e., spaces or tabs #This command will output the 2nd column of simple_data.txt #The third command will output two columns 2nd and 1st with tab in between "\t" (escaped t) represents tab character #The pipe in fourth command will list data by the value in ID column
Stream Editor For Filtering And Transforming Text	sed	sed s/orange/red/ simple_data.txt #sed is used for modifying information from a file or stream #The command will change every occurance of "orange" string in the original file to "red" in the output. s represents the string
Sort Lines Of Text Files	sort	sort simple_data.txt sort -k2 simple_data.txt sort -k2 -n simple_data.txt sort -k2n simple_data.txt sort -nk2 simple_data.txt #First command will sort the data on 1st character of each row #Second command will sort data on 2nd column but based on the 1st character of each string #Third, fourth, and fifth commands will give the same output. -n option will do the numeric sorting
Sort Lines Of Text Files – Unique	sort	sort -u dupes.txt #This command will output the unique (-u) lines from a file containing duplicates

Reverse Line Characterwise	rev	cat poems.txt | head -n5 | rev man rev #This command pipeline will output the first 5 lines of poems.txt with reverse character order in line
Concatenate And Print Files In Reverse	tac	cat poems.txt | head -n5 | tac man tac #tac is cat backwards #This command will output the first 5 lines of poems.txt with reverse line order
Translate Or Delete Characters	tr	man tr #To work with individual characters
Edit Text With vim		
vim – Vi IMproved Editor, A Programmer’s Text Editor	vim OR vi	vi poems.txt #Insertion mode: where you type and make manual changes to text and is initiated by “i” key #Command mode: where you issue commands as save, search, etc., and is initiated by “esc” key
vim – Shortcut Keys	vim	Insertion mode (at cursor location): i Command mode: esc To navigate use : ↑ ↓ ← → Move backward and forward by sentence: esc ( ) Navigate to the beginning of line: esc ⇧ + i Open up a new line after the current line: esc o move to the bottom of file: esc ⇧ + g Find/search: /search word ↩ press n for next occurrence and ⇧ + n (N) for previous occurrence Save file: esc :w New\ File.txt (“w” for write) Save and quit: esc :wq Quit without saving: esc :q!
Edit Text With nano		
Nano’s ANOther Editor, Inspired By Pico	nano	nano poems.txt Help: ^ + G Save: ^ + O (New File.txt) Move to the beginning of line: ^ + A Move to the end of line: ^ + E

		Cut: ^ + K Paste: ^ + U Exit: ^ + X Find/search: ^ + W use the same command to look for the second match Move down a screen: ^ + V Move up a screen: ^ + Y *nano is simpler than vim and to an extent more use friendly
Working With tar And Zip Archives		
TApe Archive – An Archiving Utility	tar	#Common method for archiving on linux systems #tar file with gzip compression: .tar.gz OR .tar.tgz #tar file with bzip compression: .tar.bz2
TApe Archive – An Archiving Utility	tar	tar -cvf myfiles.tar Exercise\ Files #This command will output myfiles.tar archive file of the folder Exercise Files #-c (create) option tells to create archive #-v (verbose) option tells to list each file that gets added to the archive, useful to create index #-f (file) tells tar to output the archive to a file
TApe Archive – An Archiving Utility	tar	tar -caf myfiles.tar.gz Exercise\ Files tar -caf myfiles.tar.bz2 Exercise\ Files #-a (auto compress) use the archive suffix to determine the compression program in this case .gz or .bz2 tar -czf myfiles.tar.gz Exercise\ Files tar -czf myfiles.tar.tgz Exercise\ Files #-z option filter the archive through gzip
TApe Archive – Unpack	tar	tar -xf myfiles.tar.bz2 tar -xf myfiles.tar.gz -C unpack2/ #First command will unpack myfiles.tar.bz2 in the working directory #Second command will unpack myfiles.tar.gz to the directory unpack2/. -C option represents change to directory before preforming any operation
Zip – cross platefrom friendly	zip	zip -r exfiles.zip Exercise\ Files unzip exfiles.zip unzip unpack3/exfiles.zip -d unpack4/

		#-r (recursion) option is required for a folder to create a zip archive otherwise it will create an empty archive #Second command will unzip exfiles.zip #Third command will unzip exfiles.zip from unpack3 to the current directory unpack4. -d provides an optional directory to which to extract files
Output Redirection		

Stream	Descriptor / Number	Usage
Standard Input (Stdin)	0	Text input
Standard Output (Stdout)	1	Text output
Standard Error (Stderr)	2	Error Text

Stdout	>	ls 1> filelist.txt OR ls > filelist2.txt cat filelist.txt #These commands will redirect the ls output to filelist.txt
Sterr	>	ls notreal 2> filelist3.txt #This command will redirect the error to filelist3.txt as notreal is not a directory. Here we need to specify the error redirection by typing “2”. We can use more than one redirection at a time i.e., redirecting standard output of successfully copied files to one list and standard error that failed to copy to the other
Appending text to a file	>>	>filelist4.txt #Caution!!! This command will overwrite the existing filelist4.txt to a blank file ls filelist5.txt echo “some appended text” >> filelist5.txt cat filelist5.txt #This command will append the text in “” to the bottom of filelist5.txt
Redirecting file to standard input	<	#Redirecting standard input using “<” sign to send input to a command

Exploring Environment Variables and PATH		
Environment Variables	env	
PATH Variable	echo	echo $PATH #PATH is a list of paths or directories the shell is told to look for programs or executable files outside of working directory
Where A Command Is Really Located	where	where ls #This command will output the directory “/usr/bin/ls” whre the command “ls” is located. So this is one of the places where the shell looks when we ask to rum commands
Edit Shell Profile		~/.bash_profile ls -a ~ #Files starting with “.” are usually hidden .bash_profile might not exist #Use the second command to list all contents, including hidden, of your home directory. If you don’t have .bash_profile create an empty nano ~/.bash_profile ~/.bash_profile #Type: PATH=“$PATH” to set the path to whatever the path is. This will not make an changes to path but will allow to use the existing path variable. To add more directories to the path add them inside the quotes separated by colons as: PATH=“$PATH:/my/first/path:/my/second/path” if you want changes save the file and exit
Chapter 4: Challenge 2		
Create A File Containing The Usernames Our Would-Be Hacker Tried To Login		tar -xvf log.tar.gz less auth.log cat auth.log | grep “input_userauth_request” | awk ‘{print$9}’ | sort -u > users.txt #grep will print the line that matches pattern #awk ‘{print$9}’ -> user name is the 9th item in the auth.log file #-u option will sort unique terms

Find Information About Your Linux Distribution		
Linux Distribution – Information	etc Directory	ls -l /etc/*release #Using wild card to match the name of files. This command will result in two files “lsb-release” and “os-release” wich is a link to file in directory “/usr/lib/os-release” cat /etc/lsb-release cat /etc/os-release cat /etc/*release #These commands will give the details on Linux Distribution you are currently running. Last command will output the contents of both the files one after the other
Linux Kernel – Information	uname	uname -s uname -r #First command will output all the information about kernel you are using #Second command will output just the version of kernel. The information is helpful if you are troubleshooting something or need to ask for help
Find System Hardware And Disk Information		
Memory		free -h #This command will return how much memory the machine have in human readable format (-h)
Processor Resources		cat /proc/cpuinfo lscpu #First command will give more detailed information #Second command will give CPU information along with known vulnerabilities
Hard Drive		df -h #This will output human readable sizes (M: mb; G: gb; K; kb) across different volumes. Most interesting is the one starting with “/” or “root” where the system files are, where we are likely to install softwares and download files

Disk Usage	du	du sudo du -hd1 / #First command will estimate file space usage across all the system #Second command will output the how much space is taken up across whole system in human readable format (-h) at the level of detail (-d1) to show i.e., print the total for a directory or a file 1 level deep from the root -> man du. This command will give summary. We are using sudo as a general user we will have the permission to access some of the files
Hardware	lshw lspci lsusb	lshw sudo lshw | less sudo lspci | less sudo lsusb | less #These commands will the hardware piped to less or you can pipe to a file to browse easily #lspci and lsusb will output the devices attached to the PCI and USB buses
Networking Info	ip	ip a #This command will output the ip address information for each of the network adapters
Install And Update Software With A Package Manager		
Package Managers In Different Distributions		# Debian and distros derived from it e.g., Ubuntu and Mint uses -> apt package manager (Advanced Package Tool) #Red Hat and CentOS -> DNF / Yum #Fedora -> DNF #SUSE -> YaST #Arch -> pacman
Install A Package Or Tool	apt	Name of package/tool + Command/Argument for search that package/tool OR Install/Remove apt search tree apt show tree tree -> Error!!! sudo apt update sudo apt install tree tree man tree #First command will search Ubuntu repository with apt, this command will search all packages whose name or

		description matched that term #Second command will output more about the "tree" package #Running an uninstalled package will result in error message #Fourth command will get an updated list of the packages form the repositories before installing something new #Fifth command will install the package tree #Run tree -> shows folder structure #Go through manual "tree" manual pages
Update Packages	apt	sudo apt update sudo apt upgrade #Pakage update is accomplished in two steps by "apt" some package mangers do it in one step, see manual pages for your package manager for more details #First command will get the updated list of available packages from repository mirrors. apt will get to know what needs to be changed #Second command will upgrade the identified packages #It's good to this every once in a while, to make sure we are getting security patches and bug fixes #For important systems we can configure the package manager to install critical security updates
Next Steps		
	#Courses on different Linux distributions e.g., Ubuntu desktop, Fedora, SUSE, and Red Hat #Learning Bash Scripting #Learning system administration, courses on Ubuntu and Red Hat administration. You can explore multi-tasking at the command line, configuring networking, web services, file sharing, and more #Series on Linux Tips #Keep exploring the command line using "apropos" and "man" to find new command and give yourself tasks to accomplish #Online searching -> knowing what information to look for and how to apply the hint you found online is a very important skill	