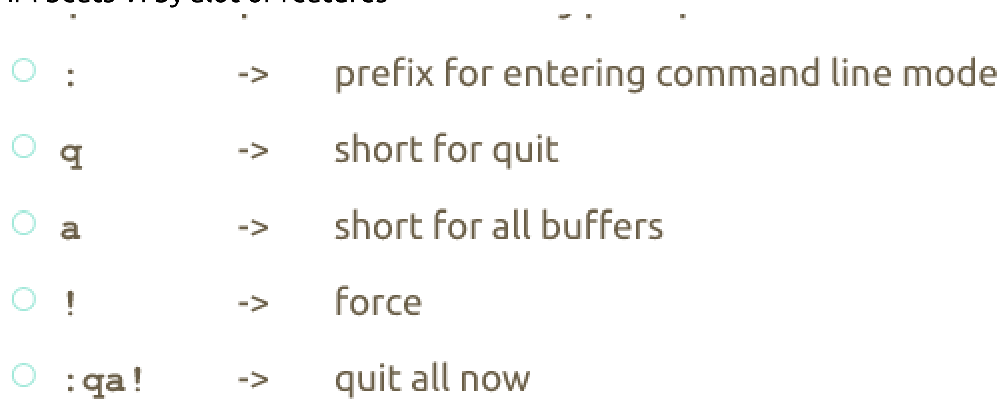
$Matthew$ $Proano$

VIM

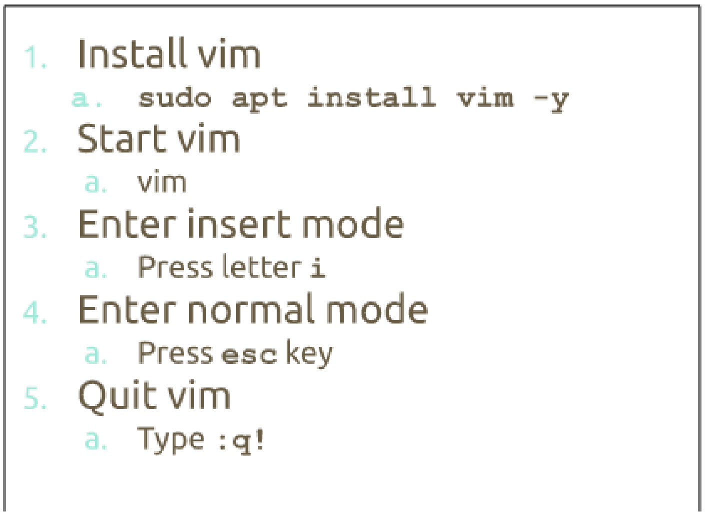
VIM beats Vi by alot of features



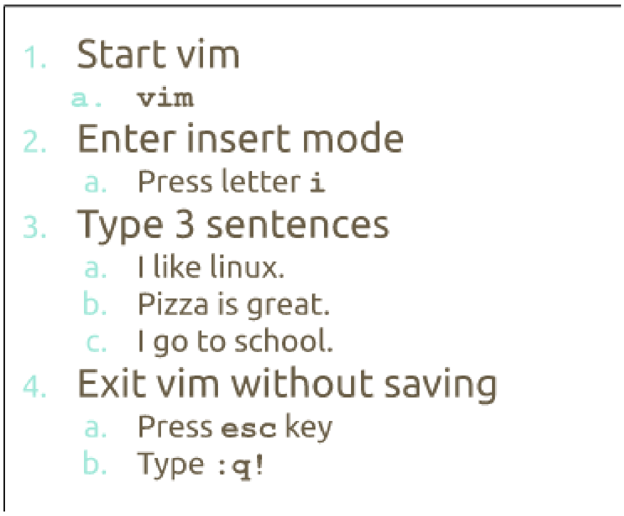
- ○ `:` -> prefix for entering command line mode
- ○ `q` -> short for quit
- ○ `a` -> short for all buffers
- ○ `!` -> force
- ○ `:qa!` -> quit all now

###insert

text pressing i is used for inserting text

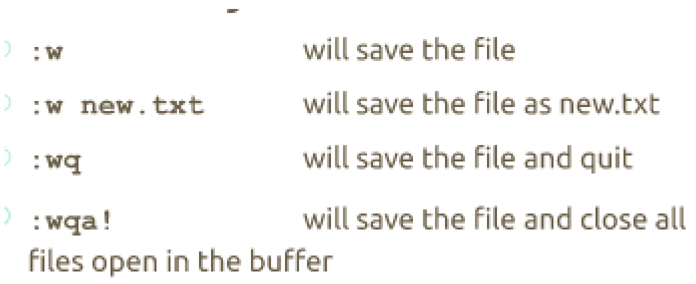


1. Install vim
   a. `sudo apt install vim -y`
2. Start vim
   a. vim
3. Enter insert mode
   a. Press letter `i`
4. Enter normal mode
   a. Press `esc` key
5. Quit vim
   a. Type `:q!`



1. Start vim
   a. `vim`
2. Enter insert mode
   a. Press letter `i`
3. Type 3 sentences
   a. I like linux.
   b. Pizza is great.
   c. I go to school.
4. Exit vim without saving
   a. Press `esc` key
   b. Type `:q!`

Saving a file & quitting

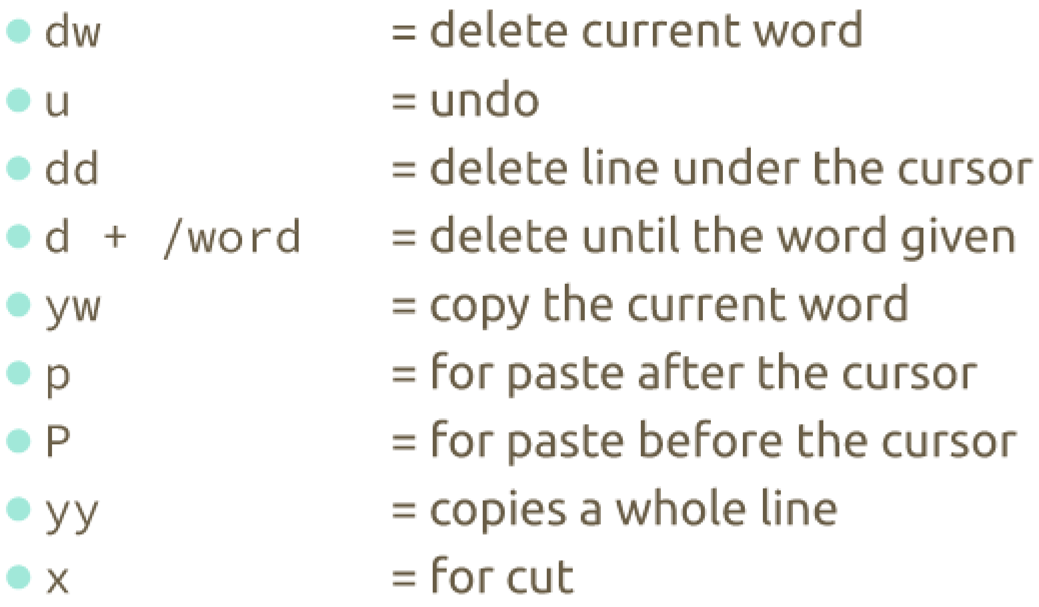


- `:w` will save the file
- `:w new.txt` will save the file as new.txt
- `:wq` will save the file and quit
- `:wqa!` will save the file and close all files open in the buffer

the command w is used for saving and quitting

moving in vim

between sentences () between paragraphs {} 10e or w will move ten words ###Searching words in vim use / to look for words ? will search backwards

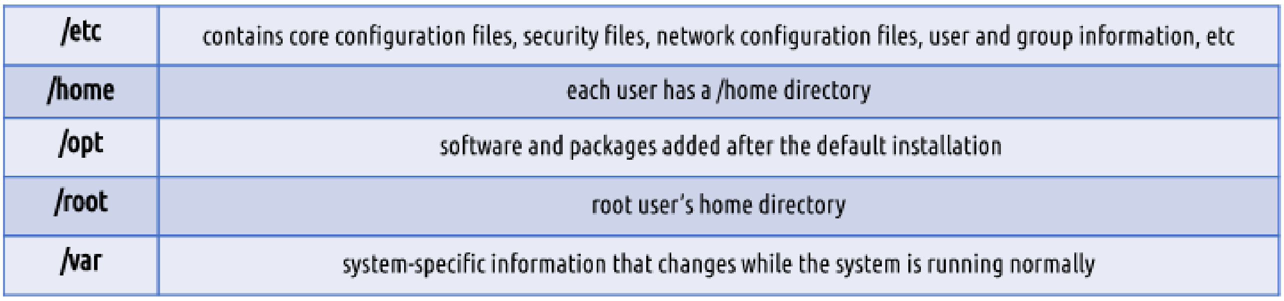


dw	= delete current word
u	= undo
dd	= delete line under the cursor
d + /word	= delete until the word given
yw	= copy the current word
p	= for paste after the cursor
P	= for paste before the cursor
yy	= copies a whole line
x	= for cut

##Managing

Data basic terms

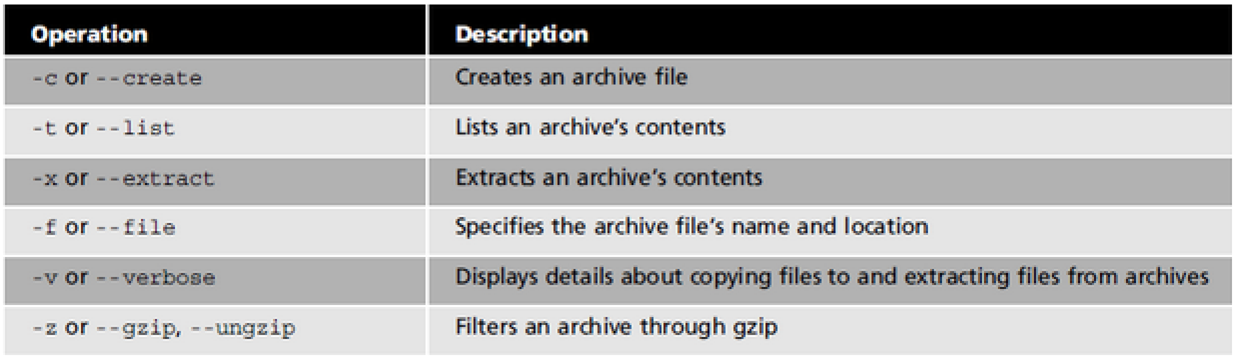
- backup- copies files and directories to archive
- system backup- used to restore data in case of a system failure.
- archive- file having other files



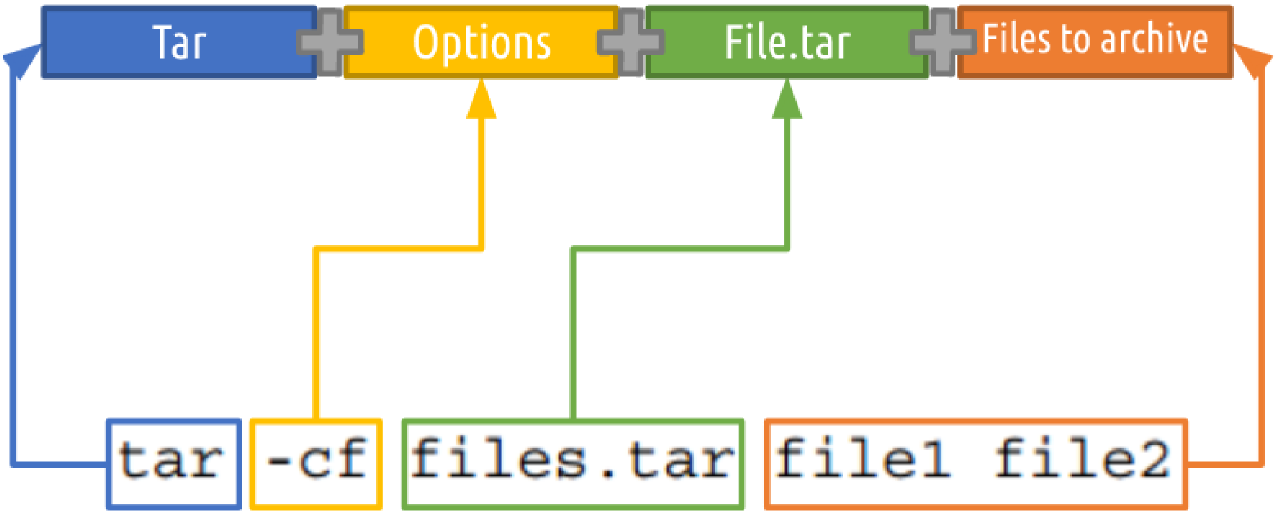
/etc	contains core configuration files, security files, network configuration files, user and group information, etc
/home	each user has a /home directory
/opt	software and packages added after the default installation
/root	root user's home directory
/var	system-specific information that changes while the system is running normally

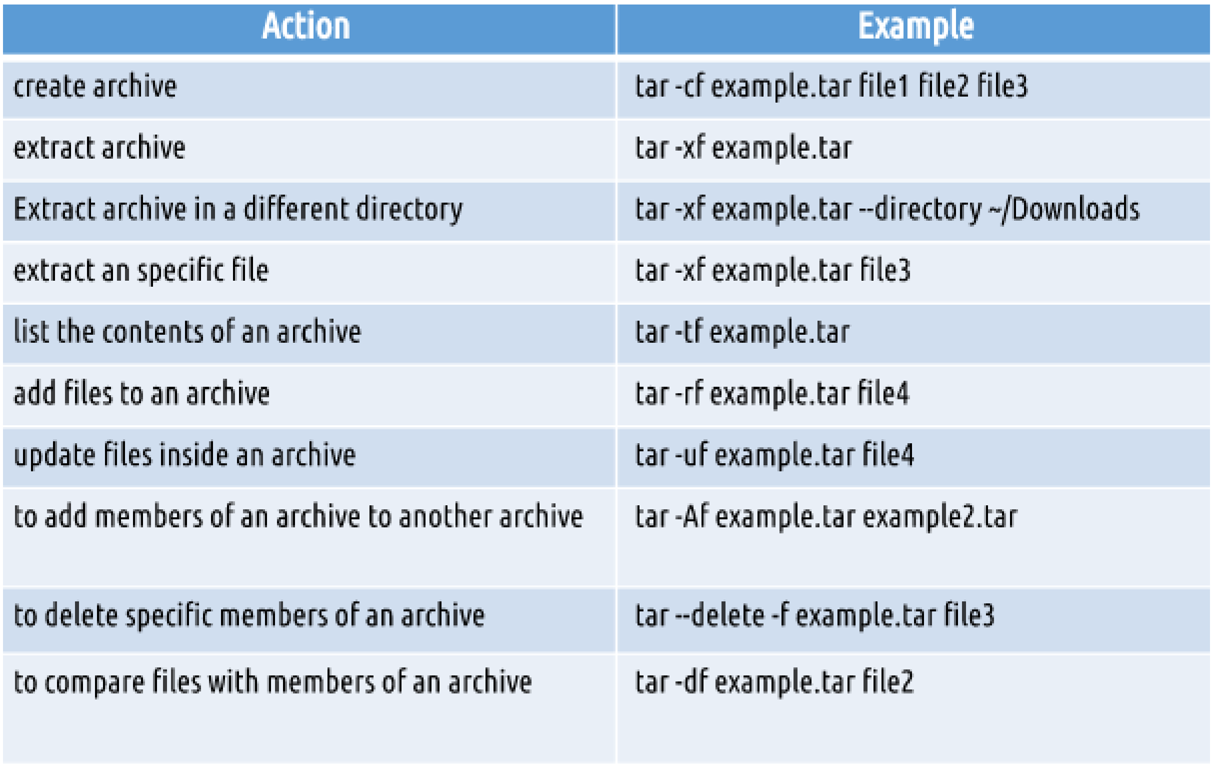
###TAR program to create an archive

- tar + options + archive name + files to add to extract an archive
- tar + option + file to extract



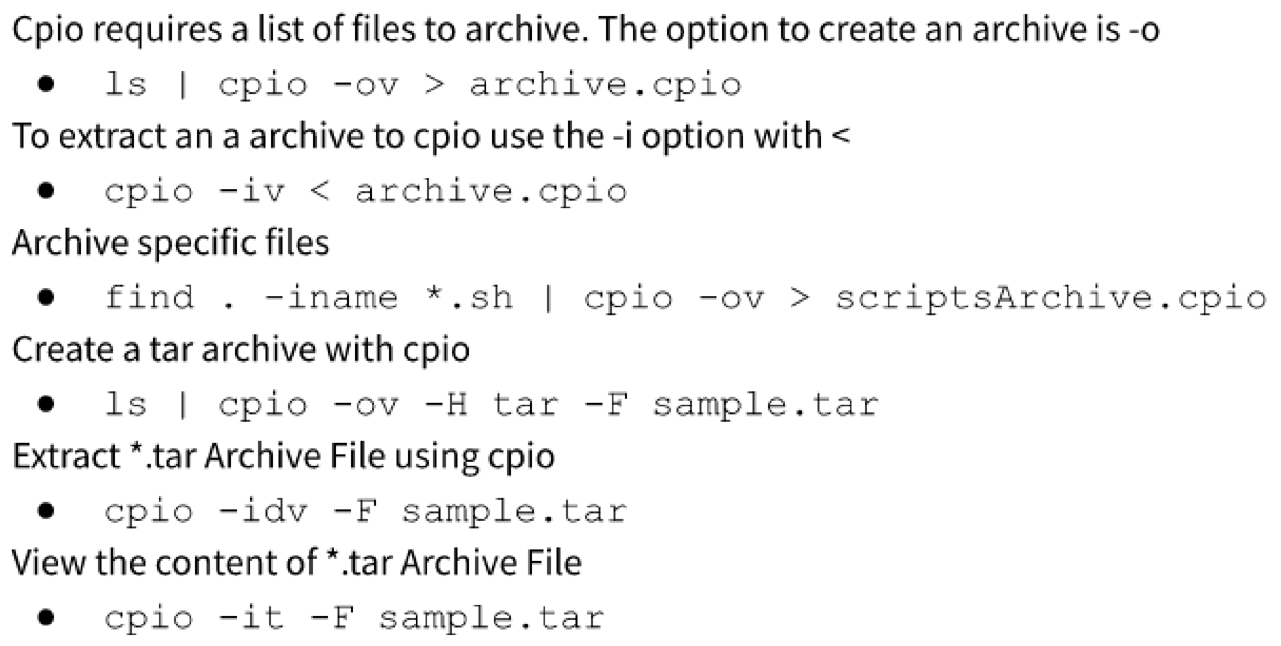
Operation	Description
-c or --create	Creates an archive file
-t or --list	Lists an archive's contents
-x or --extract	Extracts an archive's contents
-f or --file	Specifies the archive file's name and location
-v or --verbose	Displays details about copying files to and extracting files from archives
-z or --gzip, --ungzip	Filters an archive through gzip





Action	Example
create archive	tar -cf example.tar file1 file2 file3
extract archive	tar -xf example.tar
Extract archive in a different directory	tar -xf example.tar --directory ~/Downloads
extract an specific file	tar -xf example.tar file3
list the contents of an archive	tar -tf example.tar
add files to an archive	tar -rf example.tar file4
update files inside an archive	tar -uf example.tar file4
to add members of an archive to another archive	tar -Af example.tar example2.tar
to delete specific members of an archive	tar --delete -f example.tar file3
to compare files with members of an archive	tar -df example.tar file2

Cpio program



Cpio requires a list of files to archive. The option to create an archive is -o

- `ls | cpio -ov > archive.cpio`

To extract an a archive to cpio use the -i option with <

- `cpio -iv < archive.cpio`

Archive specific files

- `find . -iname *.sh | cpio -ov > scriptsArchive.cpio`

Create a tar archive with cpio

- `ls | cpio -ov -H tar -F sample.tar`

Extract *.tar Archive File using cpio

- `cpio -idv -F sample.tar`

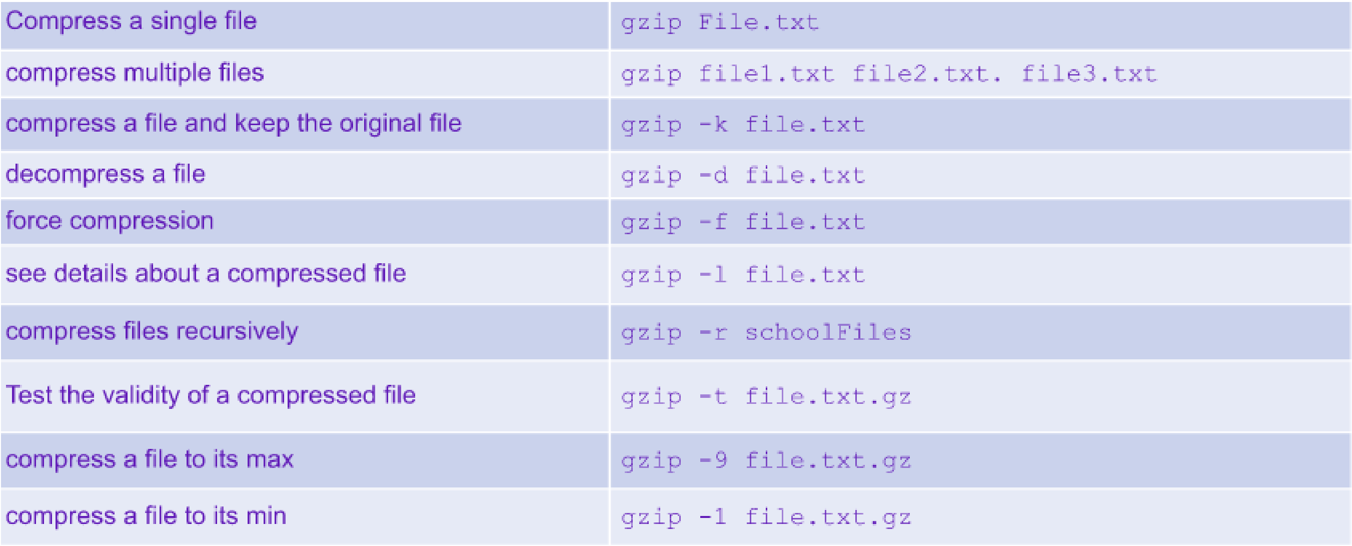
View the content of *.tar Archive File

- `cpio -it -F sample.tar`

These are just basic uses of the cpio program

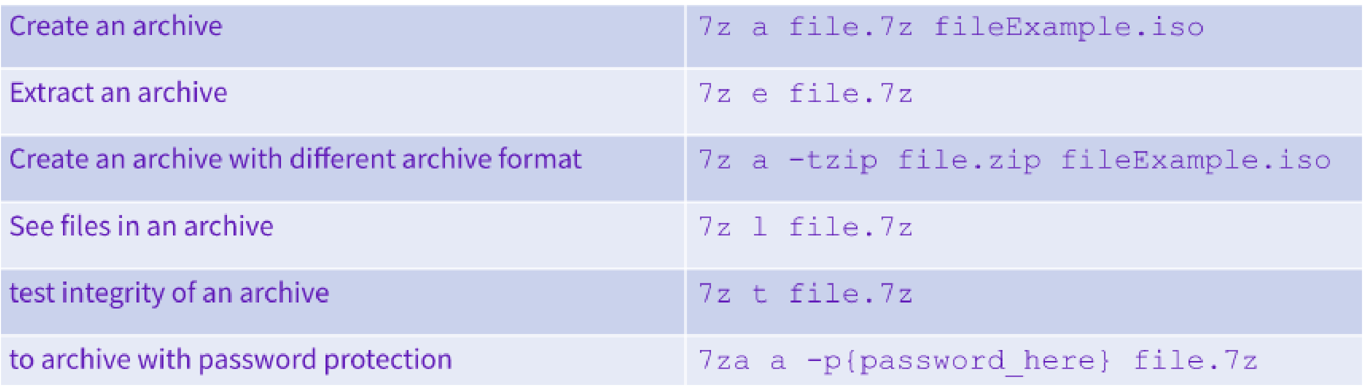
File compression

Ex. file.txt.gz =gzip or file.txt.bz2 = bzip2 or file.txt.xz =xz bzip2 offers better compression xz is the best



Compress a single file	`gzip File.txt`
compress multiple files	`gzip file1.txt file2.txt. file3.txt`
compress a file and keep the original file	`gzip -k file.txt`
decompress a file	`gzip -d file.txt`
force compression	`gzip -f file.txt`
see details about a compressed file	`gzip -l file.txt`
compress files recursively	`gzip -r schoolFiles`
Test the validity of a compressed file	`gzip -t file.txt.gz`
compress a file to its max	`gzip -9 file.txt.gz`
compress a file to its min	`gzip -1 file.txt.gz`

How to use 7zip (works the same with zip and rar) 7z + option +filename.7z + files to archive



Create an archive	`7z a file.7z fileExample.iso`
Extract an archive	`7z e file.7z`
Create an archive with different archive format	`7z a -tzip file.zip fileExample.iso`
See files in an archive	`7z l file.7z`
test integrity of an archive	`7z t file.7z`
to archive with password protection	`7za a -p{password_here} file.7z`

#Linux File Permissions ls -l shows the owner (group/person) etc/passwd had a list of all users etc/group has all groups chown is for changing group owner chmod permissions file/directories ##numeric notation

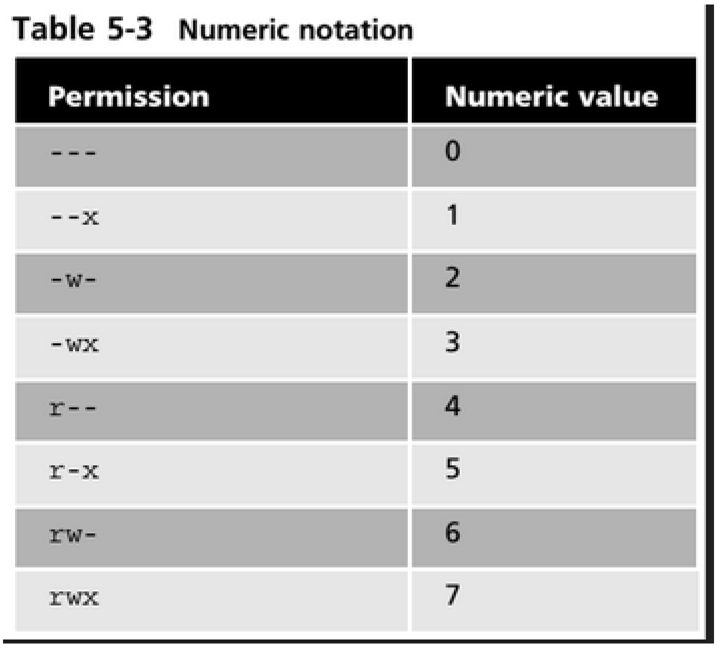
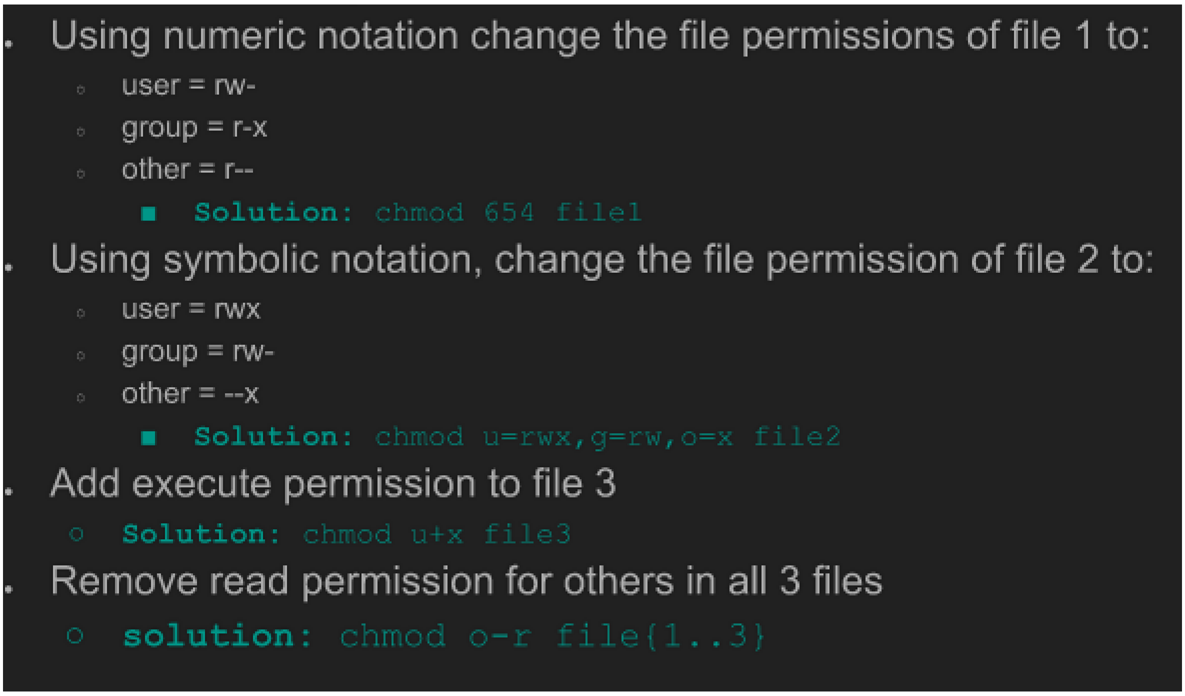


Table 5-3 Numeric notation

Permission	Numeric value
`---`	0
`--x`	1
`-w-`	2
`-wx`	3
`r--`	4
`r-x`	5
`rw-`	6
`rwx`	7

Examples



- Using numeric notation change the file permissions of file 1 to:
  - user = rw-
  - group = r-x
  - other = r--
    - **Solution:** `chmod 654 file1`
- Using symbolic notation, change the file permission of file 2 to:
  - user = rwx
  - group = rw-
  - other = --x
    - **Solution:** `chmod u=rwx,g=rw,o=x file2`
- Add execute permission to file 3
  - **Solution:** `chmod u+x file3`
- Remove read permission for others in all 3 files
  - **solution:** `chmod o-r file{1..3}`