Linux Essentials
Session-2
Command Line Basics


Kahoot!

Basic SHELL Commands

Simple Globbing

Globbing is primarily used to match patterns in filenames or text by using a wildcard character to create the pattern.

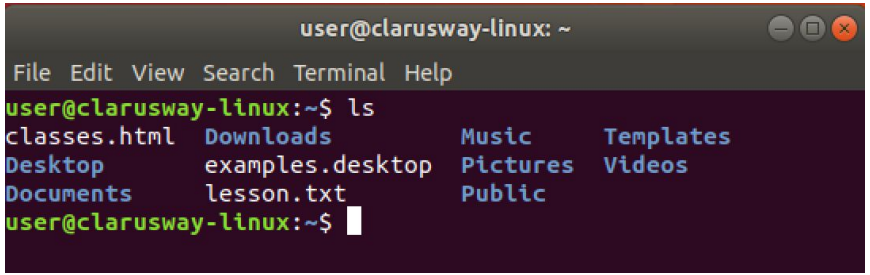
Character	Name	Function
?	**Question mark**	**Match any single character**
*****	**Asterisk**	**Match any number of character(s)**
[]	Brackets	Match character from a range
^	Caret	Used to match starting character
$	Dollar sign	Used to match ending character
{}	Curly brace	Used to match more than one pattern
|	Pipe	Used for applying more than one condition

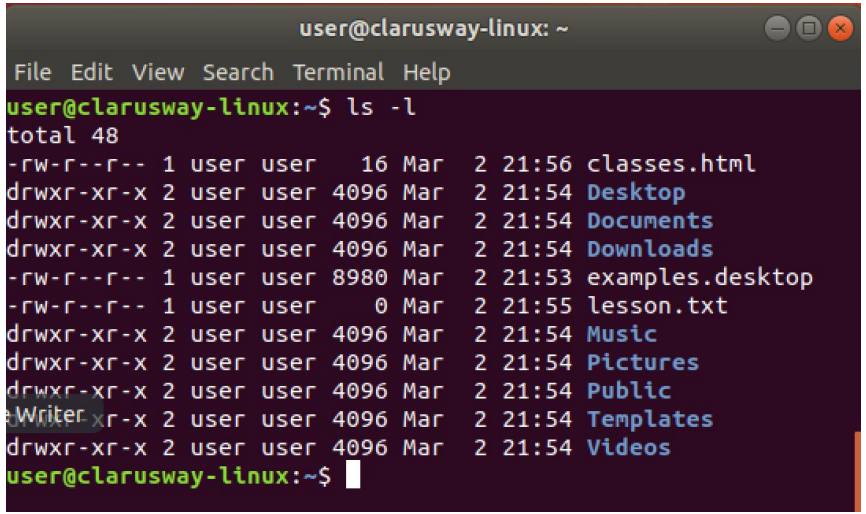
Basic Shell Commands

ls [flag] list directory contents

Flags:

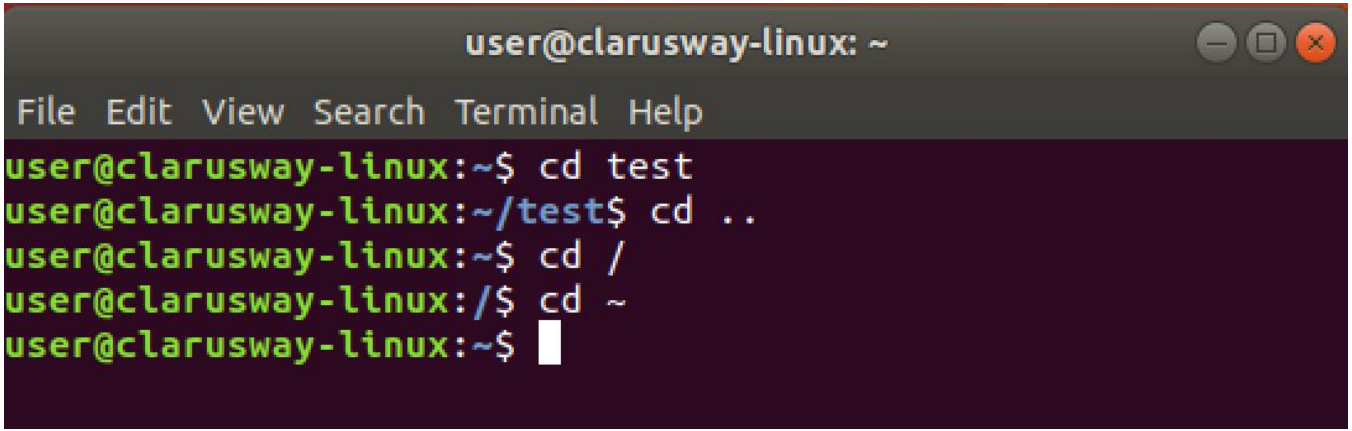
-l list details
-s sort results alphabetically
-t sort results by last modified date
-u sort result by last accessed date
-R list recursively (with subdirectories)





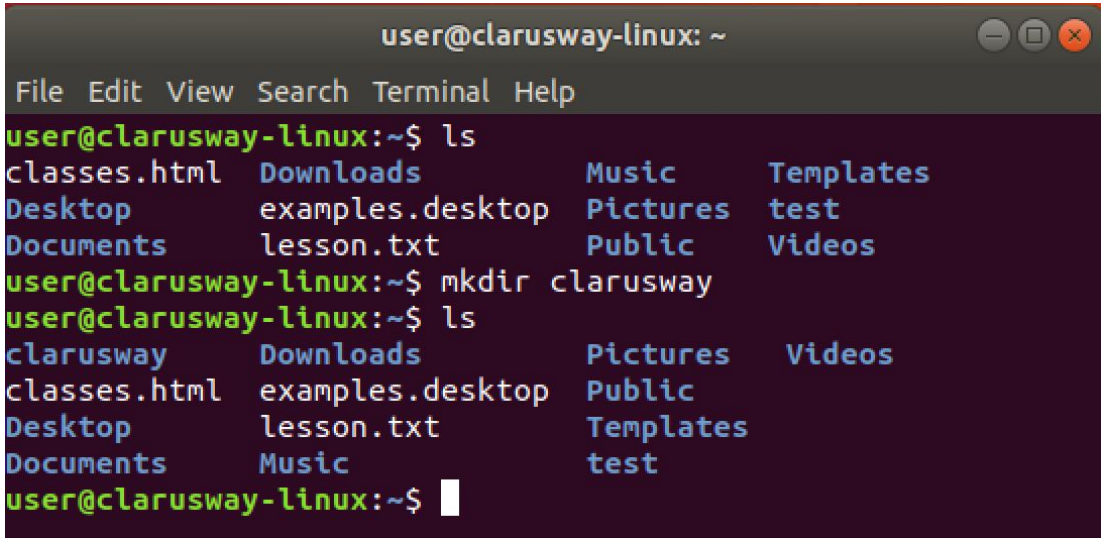
Basic Shell Commands

cd [dir] change current directory
cd .. change current directory to one level up
cd / change current directory to the root directory
cd ~ change current directory to the home directory



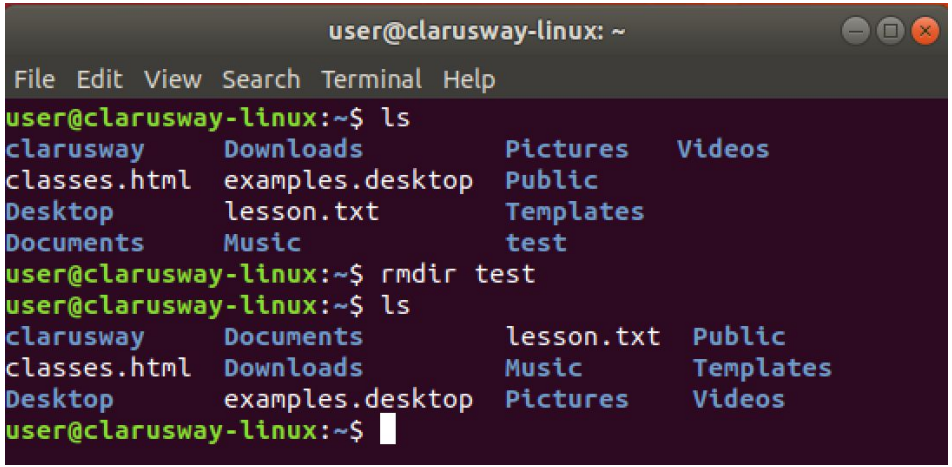
Basic Shell Commands

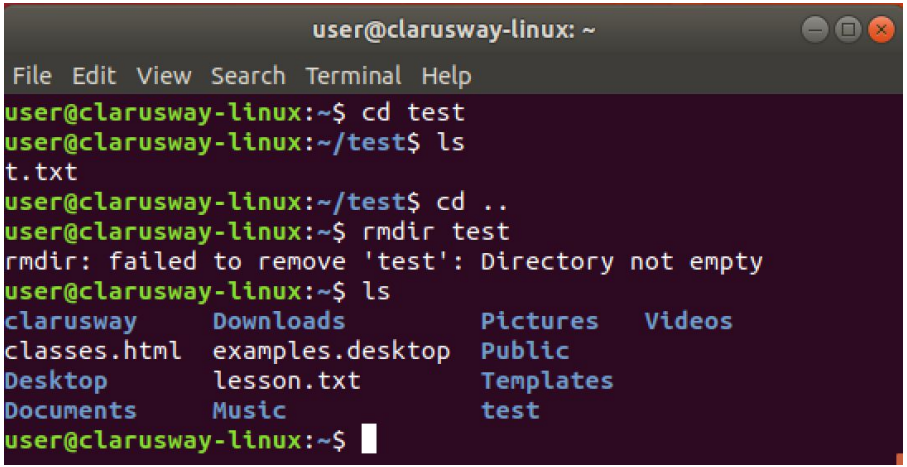
mkdir [dir] create a new directory



Basic Shell Commands

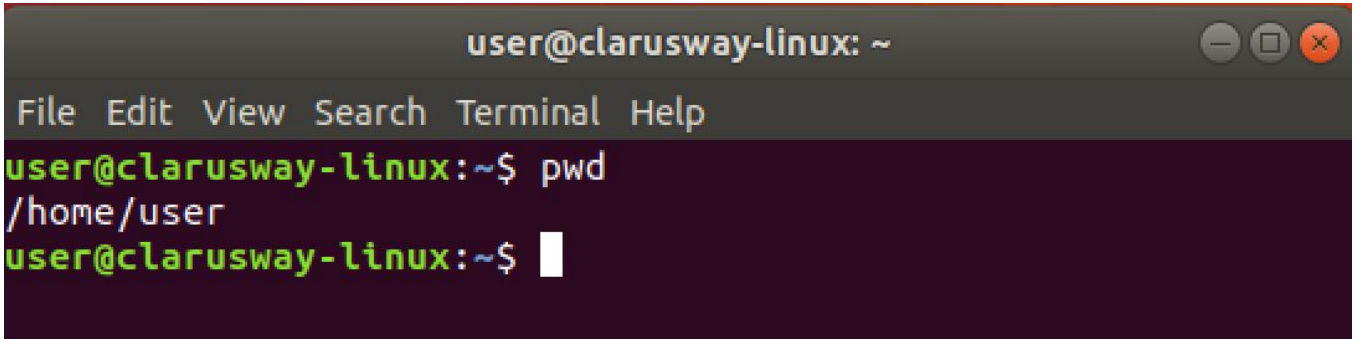
rmdir [dir] delete an empty directory





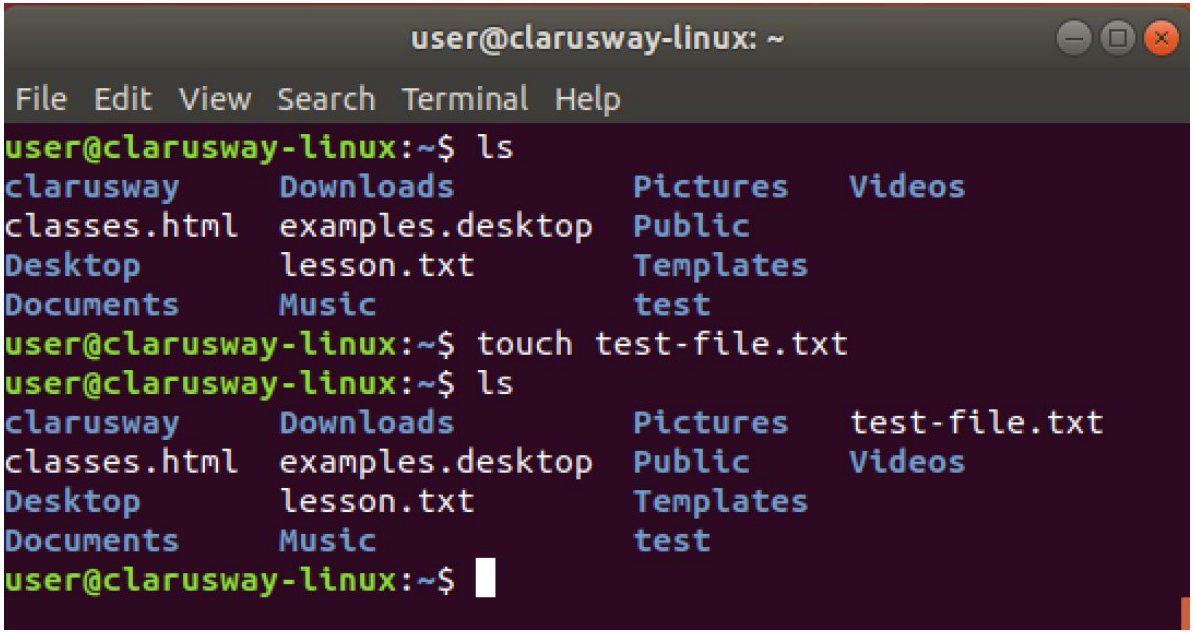
Basic Shell Commands

pwd show current path



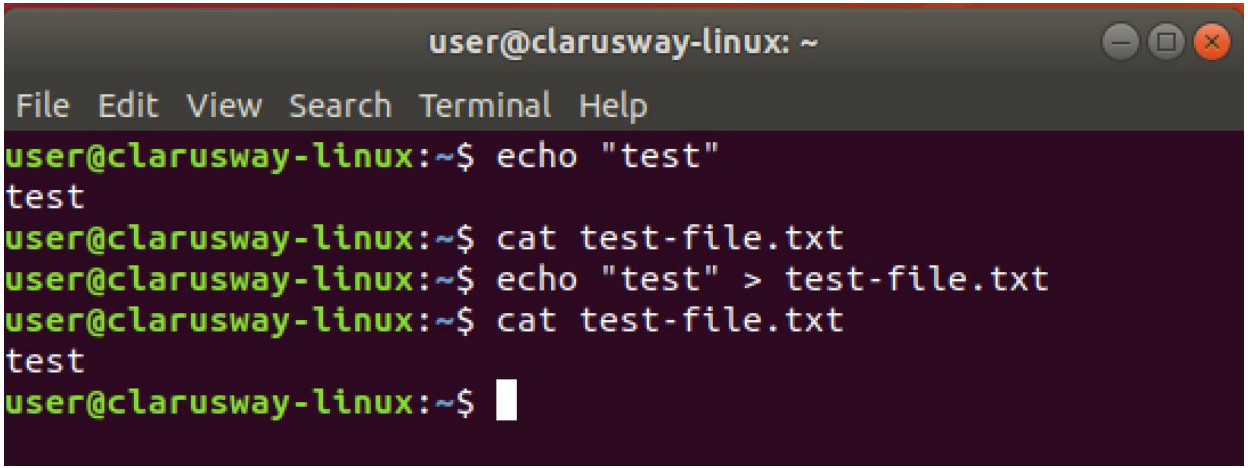
Basic Shell Commands

touch create a file



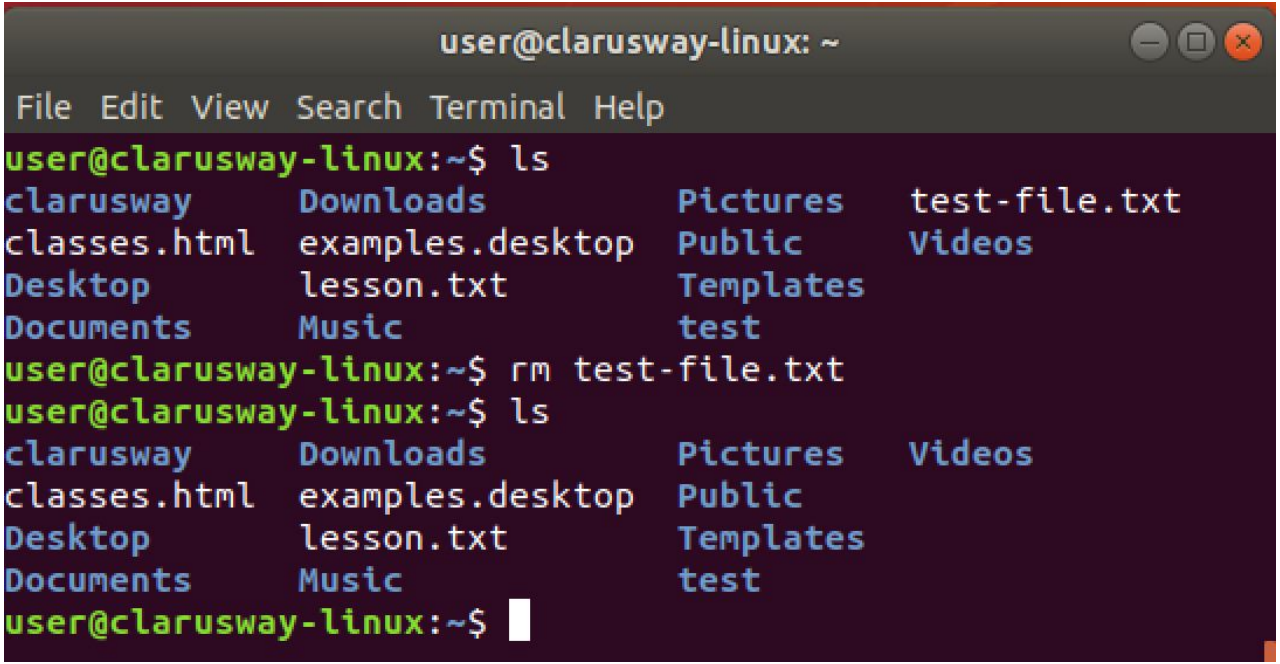
Basic Shell Commands

echo print message to screen
echo > [file] print message into a file



Basic Shell Commands

rm delete a file

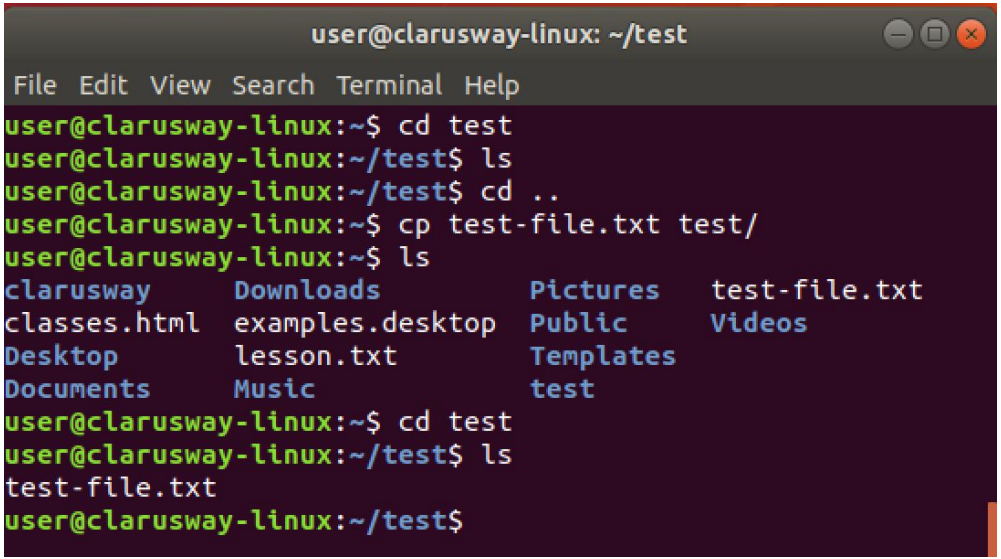


Basic Shell Commands

cp copy a file to another location

cp [flag] [source] [target]

Flags:
-R copy recursively
-i interactive, prompt before overwrite

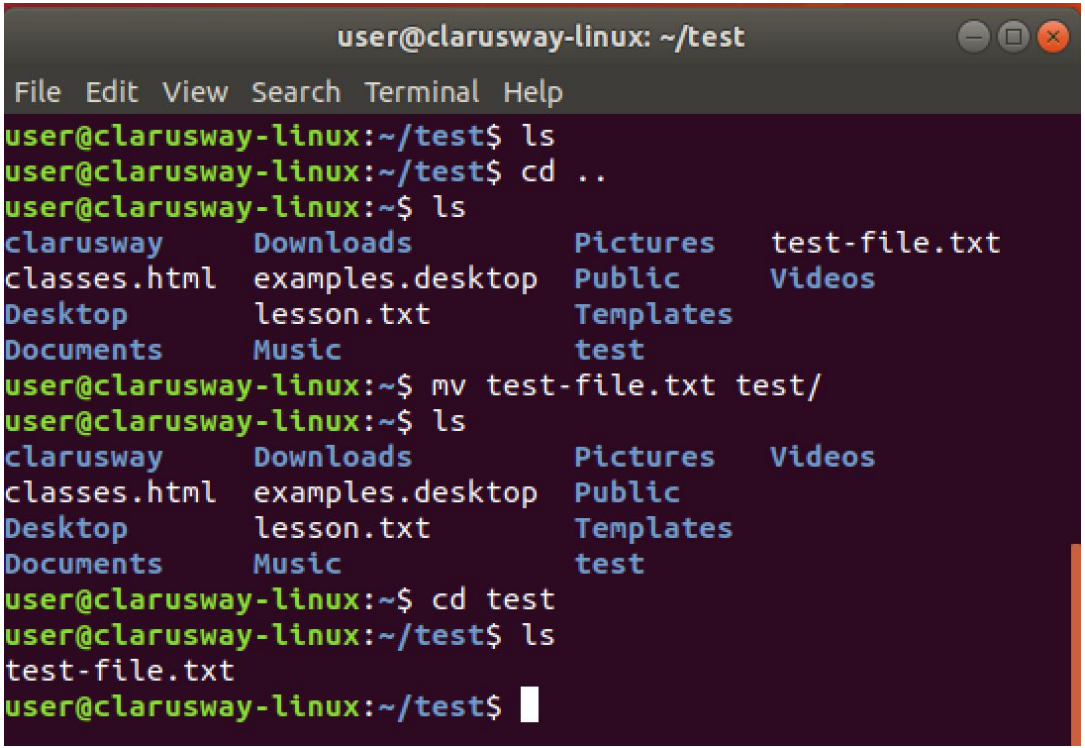


Basic Shell Commands

mv move a file to another location

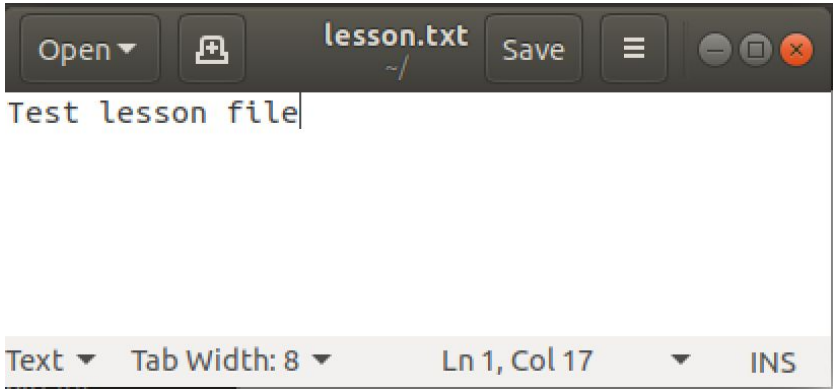
mv [flag] [source] [target]

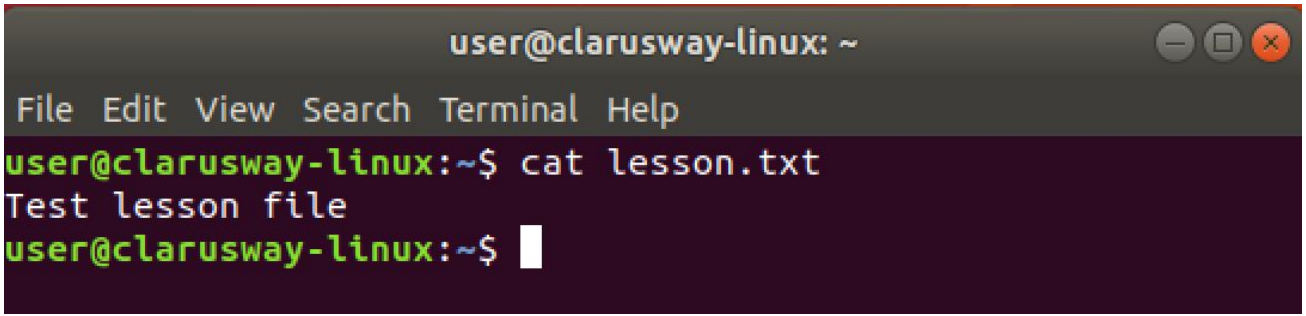
Flags:
-i interactive, prompt before overwrite



Basic Shell Commands

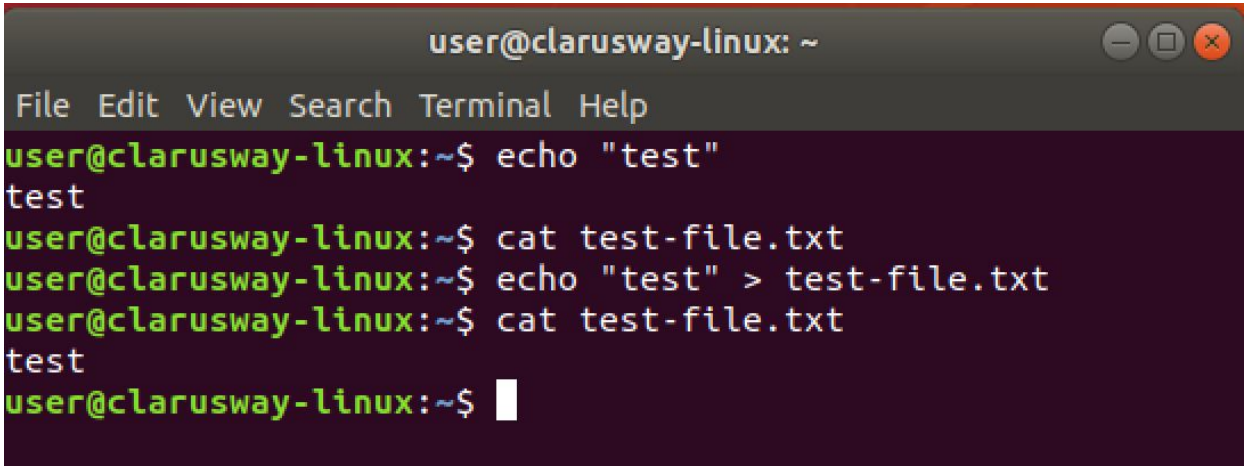
cat show file contents

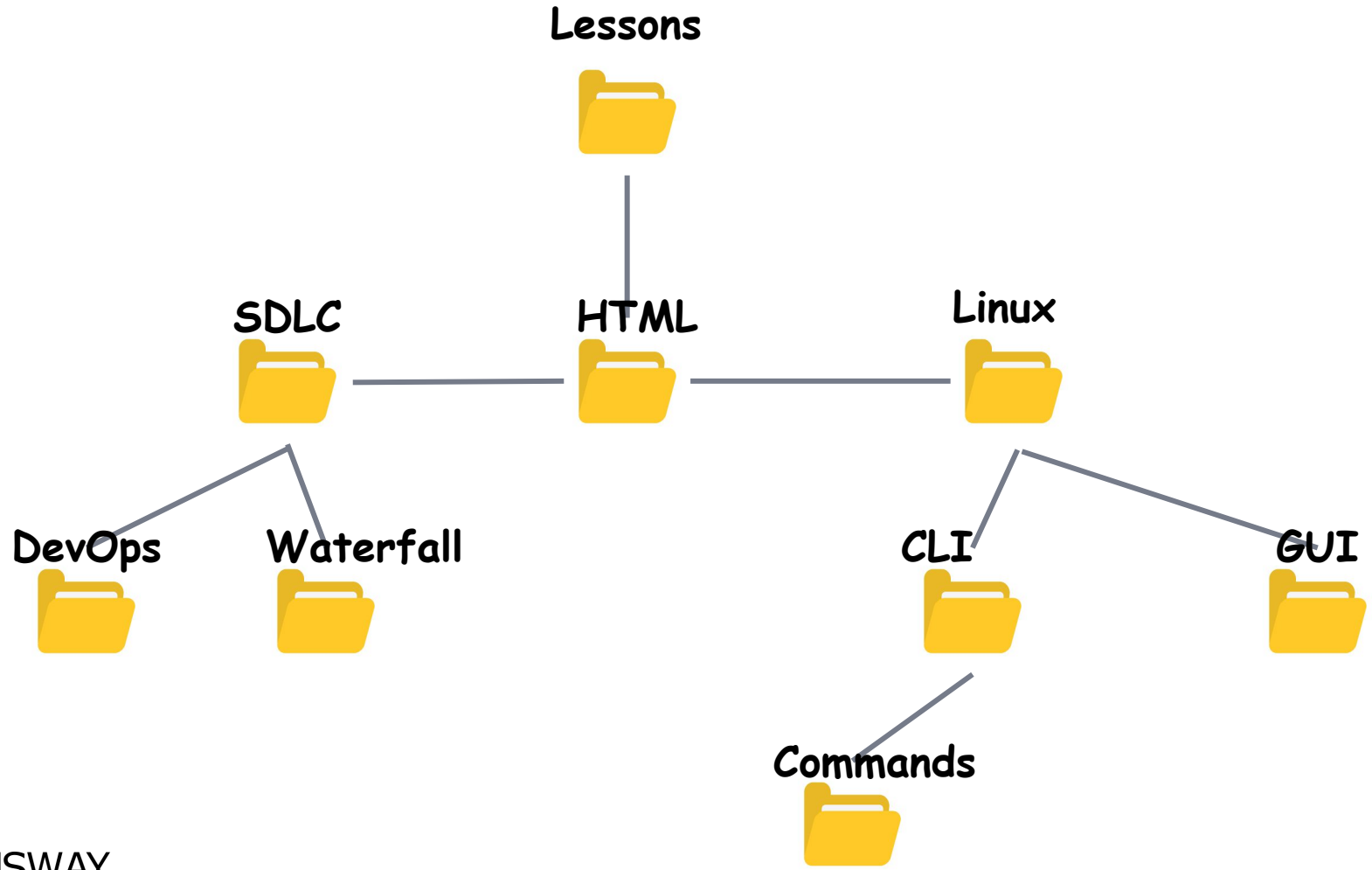




Basic Shell Commands

echo print message to screen
echo > [file] print message into a file




Lessons
SDLC
HTML
Linux
DevOps
Waterfall
CLI
GUI
Commands

Exercise

- clarusway
  - lessons
    - linux.txt
      - “I love linux”
    - html.txt
      - “I can create a website”
  - materials
    - pre-class
      - lms.txt
    - post-class
      - try-it.txt

Basic Shell Commands

~/clarusway
~/clarusway/lessons
~/clarusway/lessons/linux.txt
"I love linux"
~/clarusway/lessons/html.txt
"I can create a website"
~/clarusway/materials
~/clarusway/materials/pre-class
~/clarusway/materials/pre-class/lms.txt
~/clarusway/materials/post-class
~/clarusway/materials/pre-class/try-it.txt

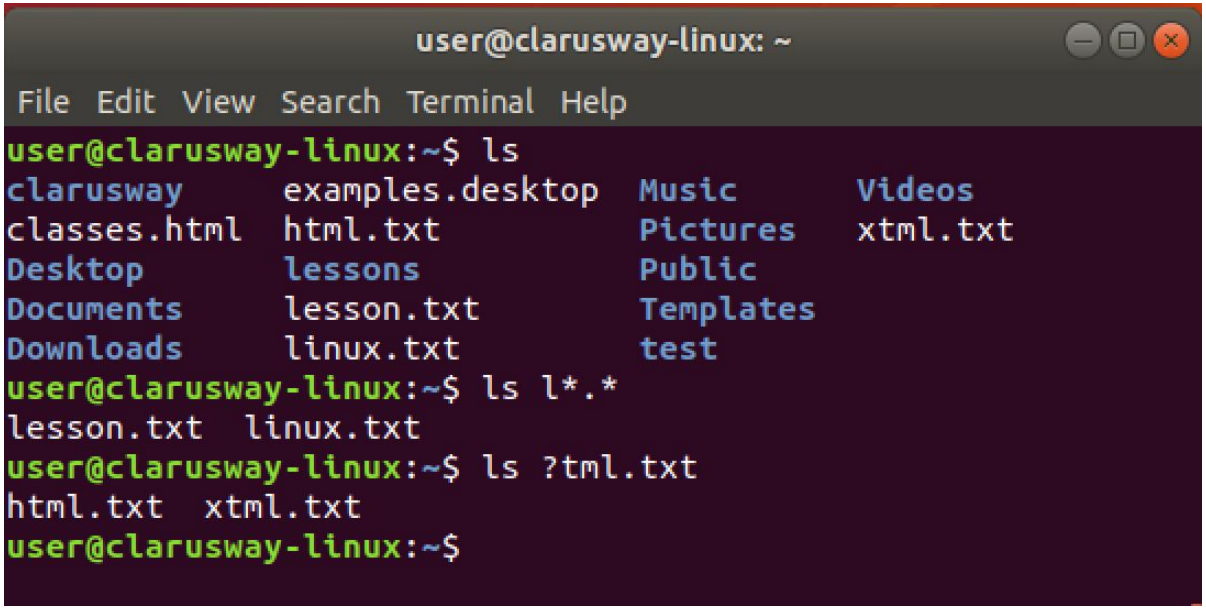
Basic Shell Commands

~/clarusway
~/clarusway/lessons
~/clarusway/lessons/linux.txt
“I love linux”
~/clarusway/lessons/html.txt
“I can create a website”
~/clarusway/materials
~/clarusway/materials/pre-class
~/clarusway/materials/pre-class/lms.txt
~/clarusway/materials/post-class
~/clarusway/materials/pre-class/try-it.tx

```
136  ls
137  mkdir clarusway
138  cd clarusway
139  mkdir lessons
140  cd lessons
141  touch linux.txt
142  echo "I love linux" > linux.txt
143  touch html.txt
144  echo "I can create a web site" > html.txt
145  cd ..
146  mkdir materials
147  cd materials
148  mkdir pre-class
149  cd pre-class
150  touch lms.txt
151  cd ..
152  mkdir post-class
153  cd post-class
154  toouch try-it.txt
```

Multiple File/Directory Operations

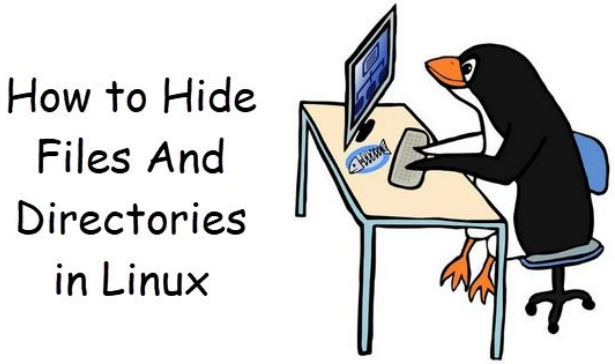
- **?** used for a single character
- ***** used for multiple characters.



Hidden Files and Directories

Any file or directory starts with period (.)
.abc

```
root@DESKTOP-4QQ1S5L:~# ls
root@DESKTOP-4QQ1S5L:~# ls -a
.  ..  .bash_history  .bashrc  .profile  .viminfo
root@DESKTOP-4QQ1S5L:~# touch .file5
root@DESKTOP-4QQ1S5L:~# ls -a
.  ..  .bash_history  .bashrc  .file5  .profile  .viminfo
root@DESKTOP-4QQ1S5L:~#
```



Basic SHELL Commands

File Commands

ls – directory listing
ls -al – formatted listing with hidden files
cd *dir* - change directory to *dir*
cd – change to home
pwd – show current directory
mkdir *dir* – create a directory *dir*
rm *file* – delete *file*
rm -r *dir* – delete directory *dir*
rm -f *file* – force remove *file*
rm -rf *dir* – force remove directory *dir* *
cp *file1 file2* – copy *file1* to *file2*
cp -r *dir1 dir2* – copy *dir1* to *dir2*; create *dir2* if it doesn't exist
mv *file1 file2* – rename or move *file1* to *file2* if *file2* is an existing directory, moves *file1* into directory *file2*
ln -s *file link* – create symbolic link *link* to *file*
touch *file* – create or update *file*
cat > *file* – places standard input into *file*
more *file* – output the contents of *file*
head *file* – output the first 10 lines of *file*
tail *file* – output the last 10 lines of *file*
tail -f *file* – output the contents of *file* as it grows, starting with the last 10 lines

Process Management

ps – display your currently active processes
top – display all running processes

System Info

date – show the current date and time
cal – show this month's calendar
uptime – show current uptime
w – display who is online
whoami – who you are logged in as
finger *user* – display information about *user*
uname -a – show kernel information
cat /proc/cpuinfo – cpu information
cat /proc/meminfo – memory information
man *command* – show the manual for *command*
df – show disk usage
du – show directory space usage
free – show memory and swap usage
whereis *app* – show possible locations of *app*
which *app* – show which *app* will be run by default

Compression

tar cf *file.tar files* – create a tar named *file.tar* containing *files*
tar xf *file.tar* – extract the files from *file.tar*
tar czf *file.tar.gz files* – create a tar with Gzip compression
tar xzf *file.tar.gz* – extract a tar using Gzip
tar cjf *file.tar.bz2* – create a tar with Bzip2 compression
tar xjf *file.tar.bz2* – extract a tar using Bzip2
gzip *file* – compresses *file* and renames it to *file.gz*

Basic SHELL Commands

File system Commands	
ls	lists directories and files
ls -a	lists all files including hidden files
ls -lh	formatted list including more data
ls -t	lists sorted by date
pwd	returns path to working directory
cd *dir*	changes directory
cd ..	goes to parent directory
cd /	goes to root directory
cd	goes to home directory
touch *file_name*	creates en empty file
cp *file file_copy*	copy a file
cp -r	copy files contained in directories
rm *file*	deletes a file
rm -r *dir*	deletes a directory and its files
mv *file1 file2*	moves or renames a file
mkdir *dir_name*	creates a directory
rmdir *dir_name*	deletes a directory

Text handling commands	
command > *file*	saves STDOUT in a file
command >> *file*	appends STDOUT in a file
cat *file*	concatenate and print files
cat *file1 file2* > *file3*	merges files 1 and 2 into *file3*
cat *fasta > all.fasta	concatenates all fasta files in the current directory
head *file*	prints first lines from a file
head -n 5 *file*	prints first five lines from a file
tail *file*	prints last lines from a file
tail -n 5 *file*	prints last five lines from a file
less *file*	view a file
less -N *file*	includes line numbers
less -S *file*	wraps long lines
grep '*pattern*' *file*	Prints lines matching a pattern
grep -c '*pattern*' *file*	counts lines matching a pattern
cut -f 1,3 *file*	retrieves data from selected columns in a tab-delimited file
sort *file*	sorts lines from a *file*

Write a command that lists all files which have 3-letter extension starting with “y”.

Using the Command Line to Get Help

Table of Contents

1 Man Pages

Man Pages

man [command]

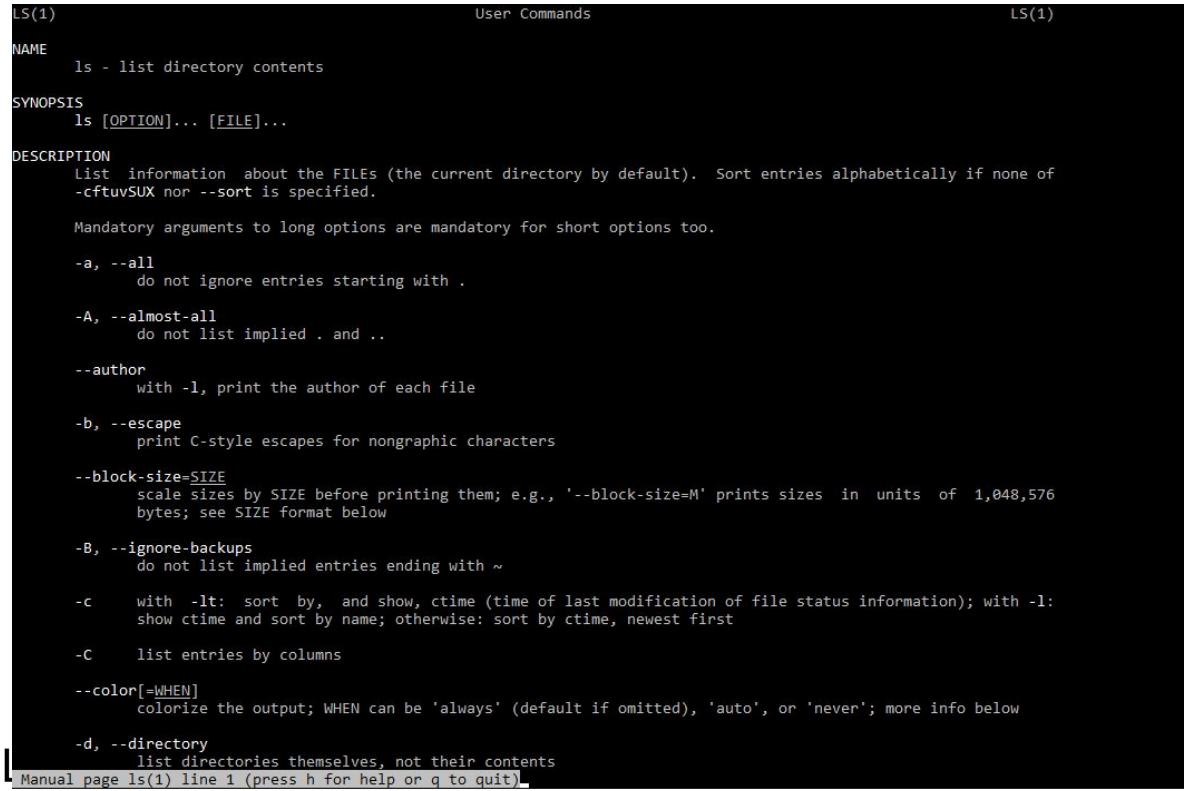
A man page (short for manual page) is a form of software documentation usually found on a Unix or Unix-like operating system.

if we install a package to do some task, the man Paige for that package will typically be installed at the same time. This gives us the ability to take a look at that documentation and make sure that we're using it in a manner consistent with its design.

The man page for a particular command is invoked by preceding the command with **man.**

Man Pages

man ls



NAME
Program or Function name(s) followed by descriptions of functionality.

SYNOPSIS
A short overview of available options

DESCRIPTION
Detailed information about arguments and options.

2 Info Pages

Info Pages

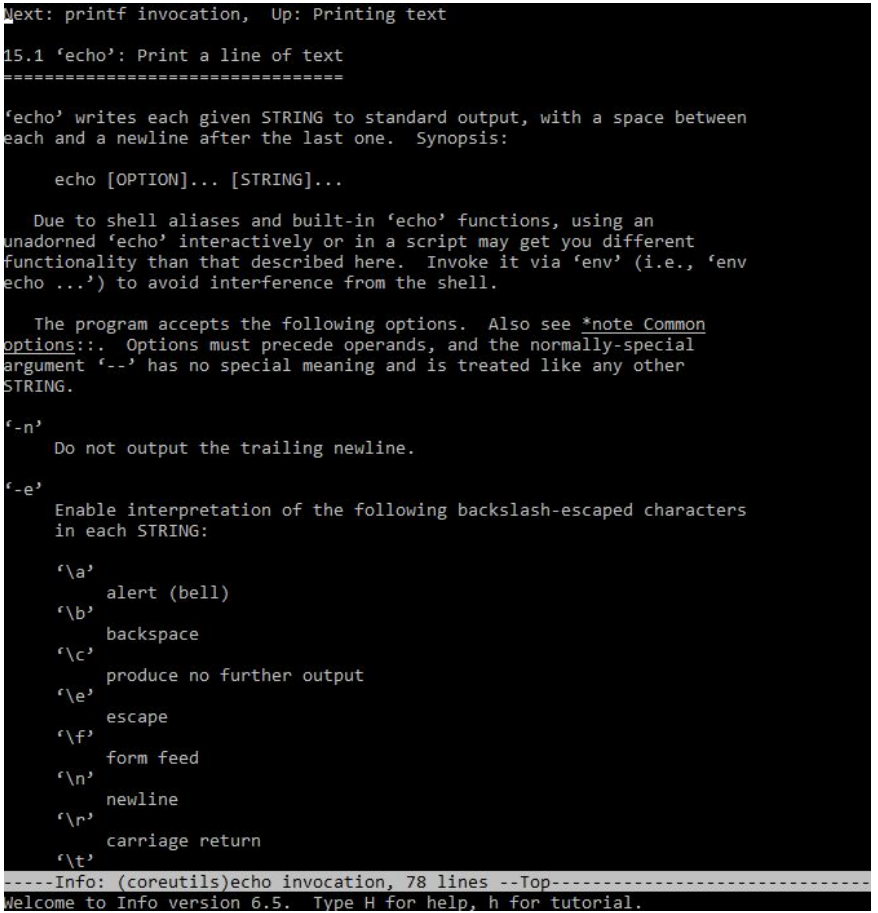
info [command]

Info pages are additional documentation with more robust capability in detail. Info Page normally provides more detailed information about a command than its respective man page. Additionally, Info uses a structure for linking these pages together, and they may be assembled into a larger collection.

The info page for a particular command is invoked by preceding the command with **info**

Info Pages

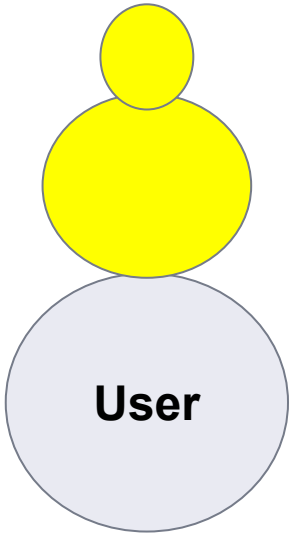
info echo

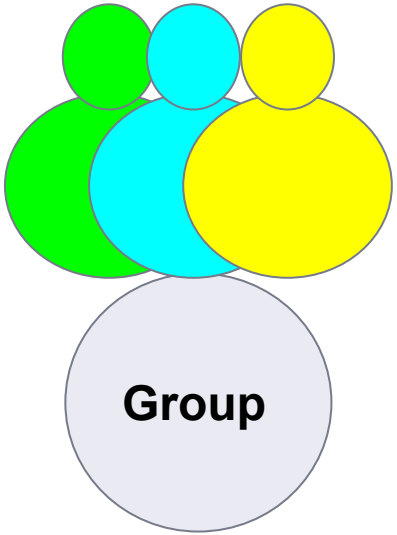


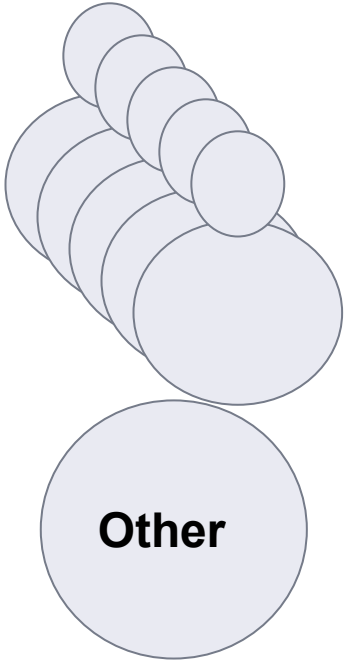
File Permission

File Permission

Ownership



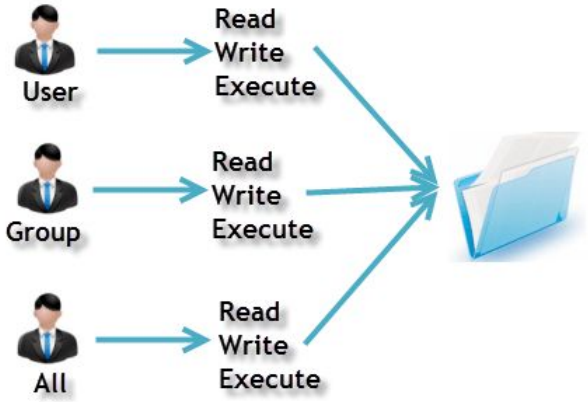




File Permission

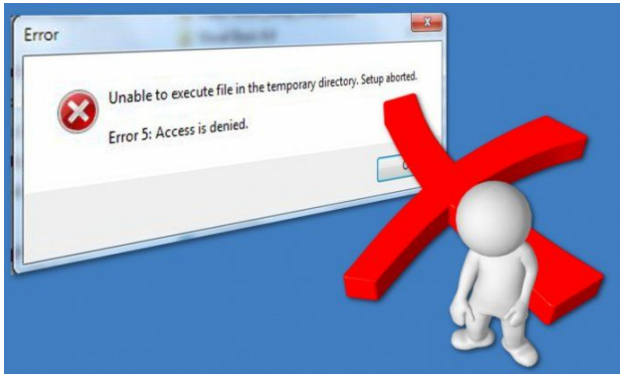
Ownership	
User	• A user is the owner of the file.
Group	• A user- group can contain multiple users.
Other	• Any other user who has access to a file.

Permission	
Read	• This permission give you the authority to open and read a file.
Write	• The write permission gives you the authority to modify the contents of a file.
Execute	• you cannot run a program unless the execute permission is set.



File Permission

Permissions



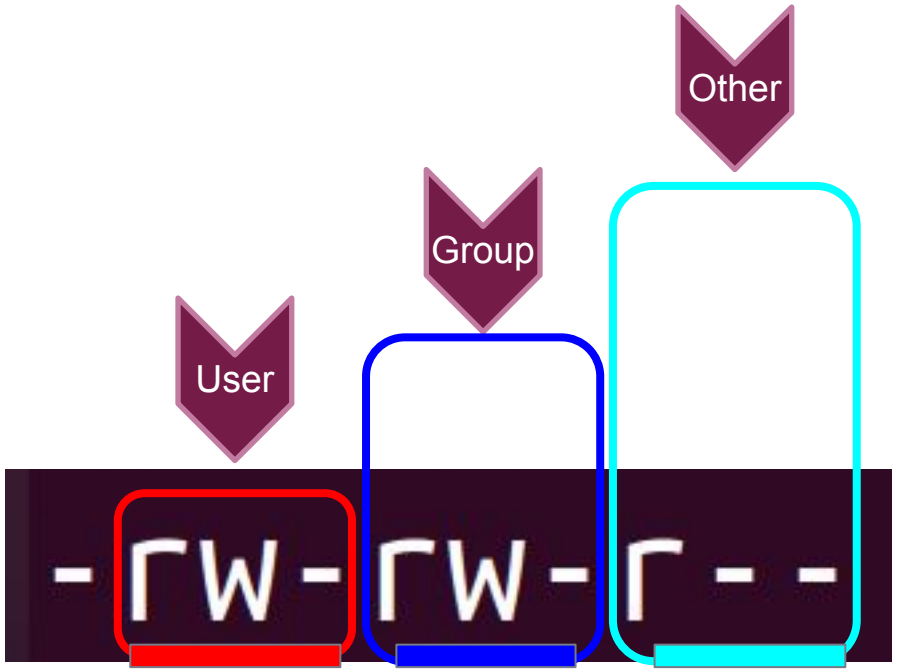
Read

Write

Execute

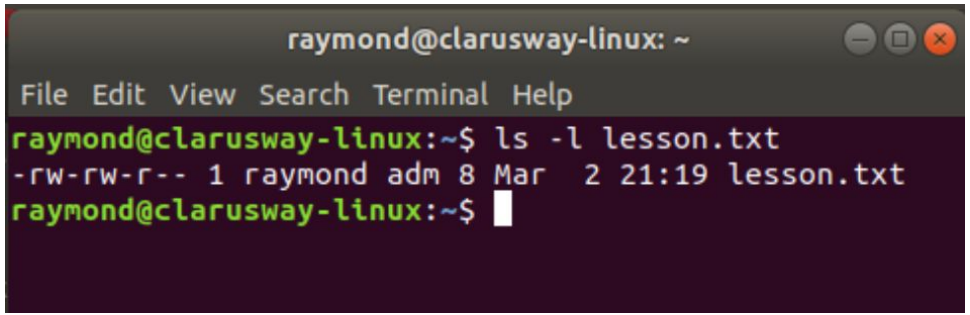
File Permission

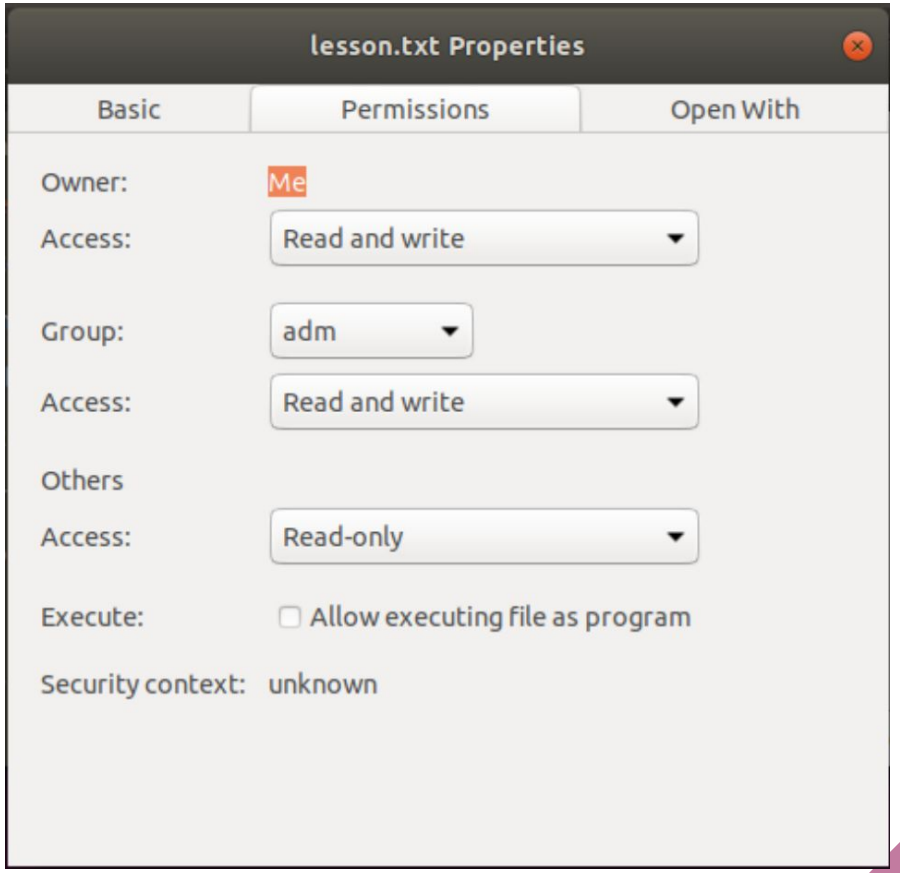
Ownership




```
-rw-rw-r-- 1 zk zk 0 Dec  7 15:39 html.txt
```

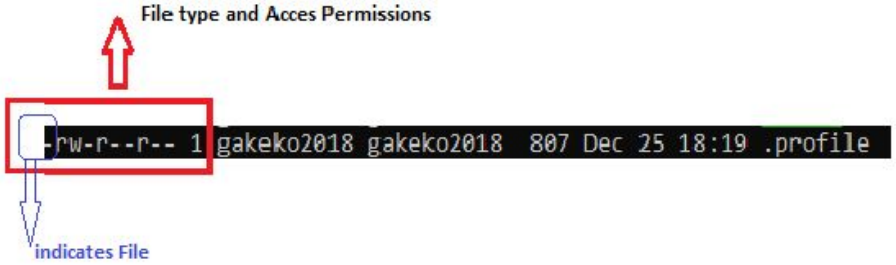
File Permission

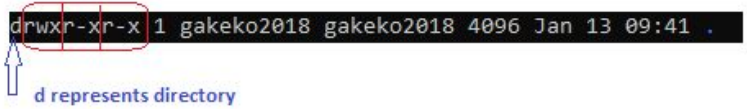


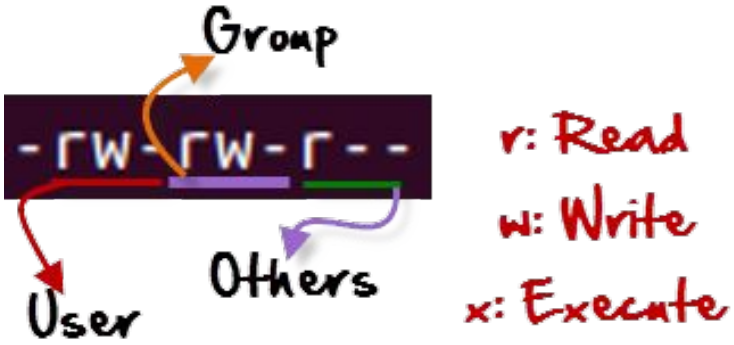


File Permission

```
gakeko2018@DESKTOP-JAO7K2U:~$ ls
cert.pem
gakeko2018@DESKTOP-JAO7K2U:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .local  .profile  .ssh  cert.pem
gakeko2018@DESKTOP-JAO7K2U:~$ ls -al
total 12
drwxr-xr-x 1 gakeko2018 gakeko2018 4096 Jan 13 09:41 .
drwxr-xr-x 1 root       root       4096 Dec 25 18:19 ..
-rw------- 1 gakeko2018 gakeko2018  236 Jan 14 12:21 .bash_history
-rw-r--r-- 1 gakeko2018 gakeko2018  220 Dec 25 18:19 .bash_logout
-rw-r--r-- 1 gakeko2018 gakeko2018 3771 Dec 25 18:19 .bashrc
drwxrwxrwx 1 gakeko2018 gakeko2018 4096 Jan 13 09:38 .local
-rw-r--r-- 1 gakeko2018 gakeko2018  807 Dec 25 18:19 .profile
drwx------ 1 gakeko2018 gakeko2018 4096 Jan 13 09:41 .ssh
-r-------- 1 gakeko2018 gakeko2018 1675 Jan 13 09:38 cert.pem
```







r: Read
w: Write
x: Execute

r = read permission
w = write permission
x = execute permission
- = no permission

File Permission

Changing Permission with chmod Command

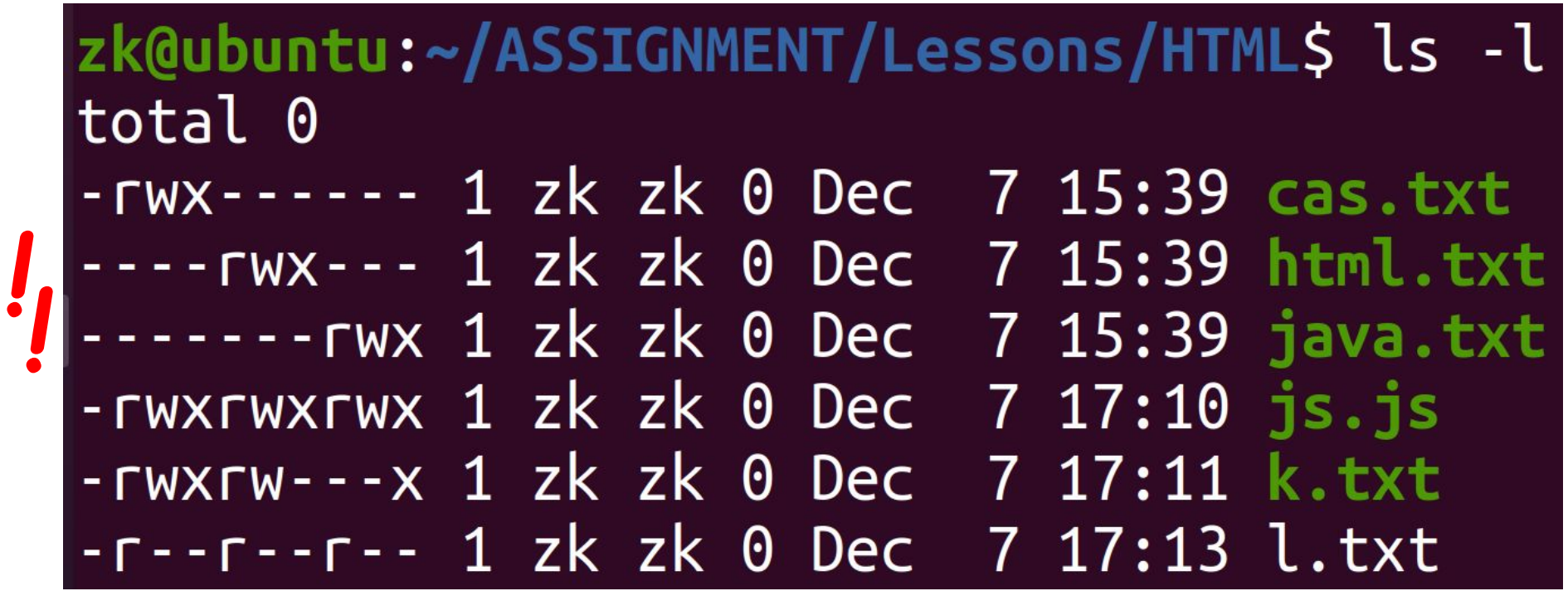
We can use the chmod command which stands for **change mode**.
we can set permissions (read, write, execute) on a file/directory for the owner, group and the world.

chmod permissions filename

```
chmod u=rwx,g=rx,o=r myfile
```

Symbol	Permission Type
---	No Permission
--x	Execute
-w-	Write
-wx	Execute+Write
r--	Read
r-x	Read+Execute
rw-	Read+Write
rwx	Read+Write+Execute

File Permission

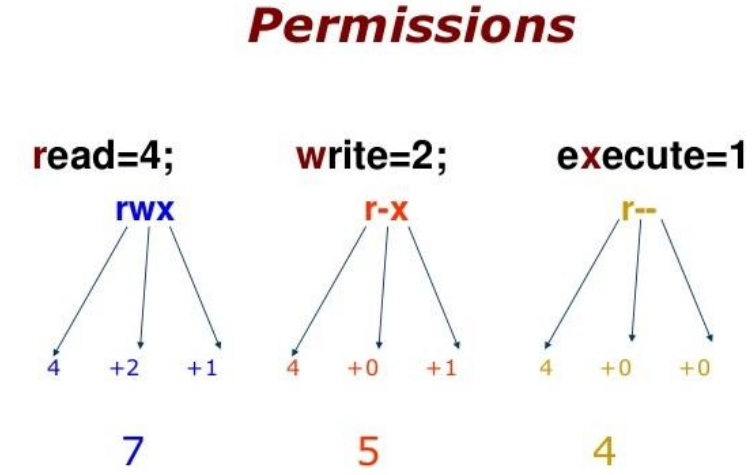


File Permission

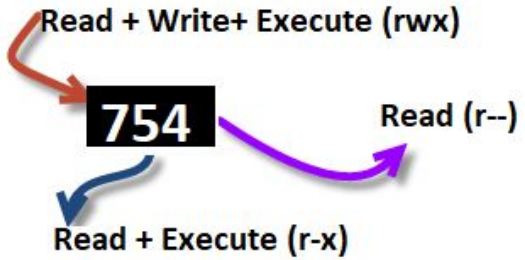
```
root@DESKTOP-4QQ1S5L:~# ls -l
total 0
-rw-rw-rw- 1 root root 0 Dec 29 17:53 file1
-r--r--rwx 1 root root 0 Dec 29 17:53 file2
root@DESKTOP-4QQ1S5L:~# chmod 754 file2
root@DESKTOP-4QQ1S5L:~# ls -l file2
-rwxr-xr-- 1 root root 0 Dec 29 17:53 file2
root@DESKTOP-4QQ1S5L:~#
```

754 code says;

- Owner can read, write and execute
- User's group can read and execute
- Other can only read

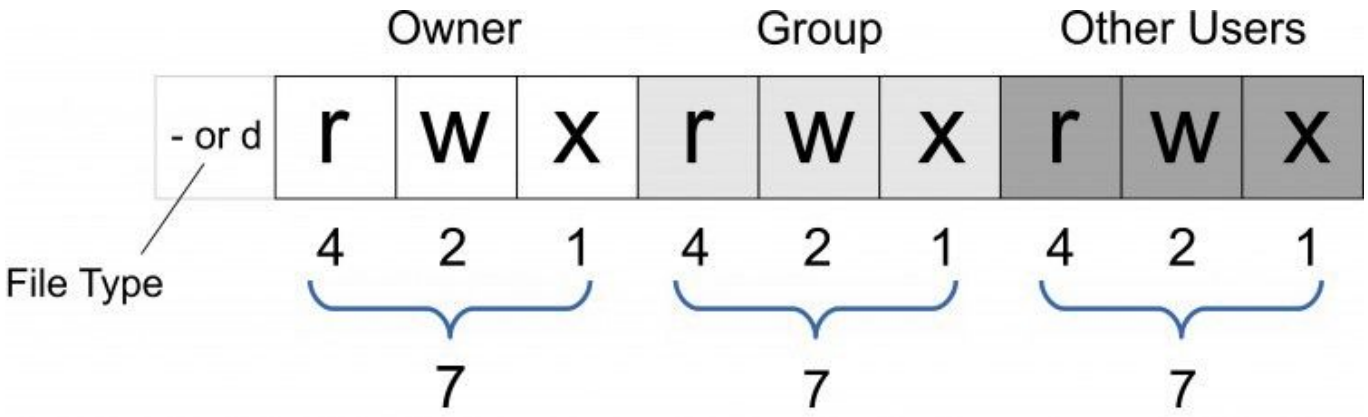


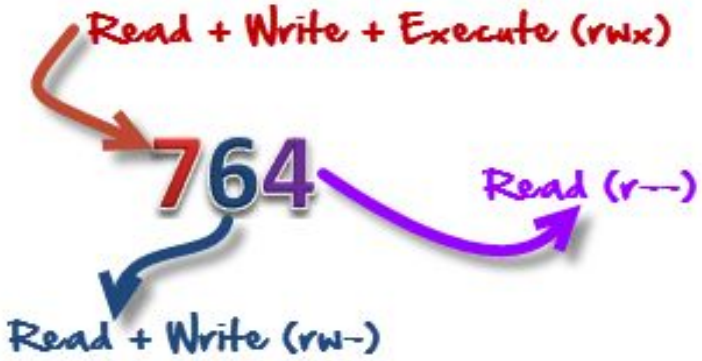

```
chmod u=rwx,g=rx,o=r myfile
chmod 754 myfile
```



CLARUSWAY
WAY TO REINVENT YOURSELF

File Permission





d	r read	w write	x exec	r read	- write	x exec	r read	- write	- exec
File type	Owner permissions			Group permissions			User permissions		
(directory)	4	2	1	4	2	1	4	2	1
	7			5			4		

Set permissions of myfile.txt to;

owner : full access
group : read and execute
others : no access

THANKS!

Any questions?