Topics to study for midterm

What is Linux?

Linux is a multitasking, multi user and multi purpose operating system.

What is Ubuntu?

Ubuntu is a Linux distribution, freely available with community and professional support.

What is a Linux Distribution?

Any operating system that uses the linux kernel.

Advantages and disadvantages of using Ubuntu

Advantages

- software is generally available for free
- the user can modify the code
- General more reliable
- here is an article about it

Disadvantages

- less hardware support than Windows
- can be risky
- less user friendly than Windows
- lack of corporate support.

How to work with multiple terminal windows

- Snap one terminal to the right and another to the left
- Open multiple tabs

How to use flameshot to highlight screenshots

- see the video in youtube

How to download files from the internet using wget.

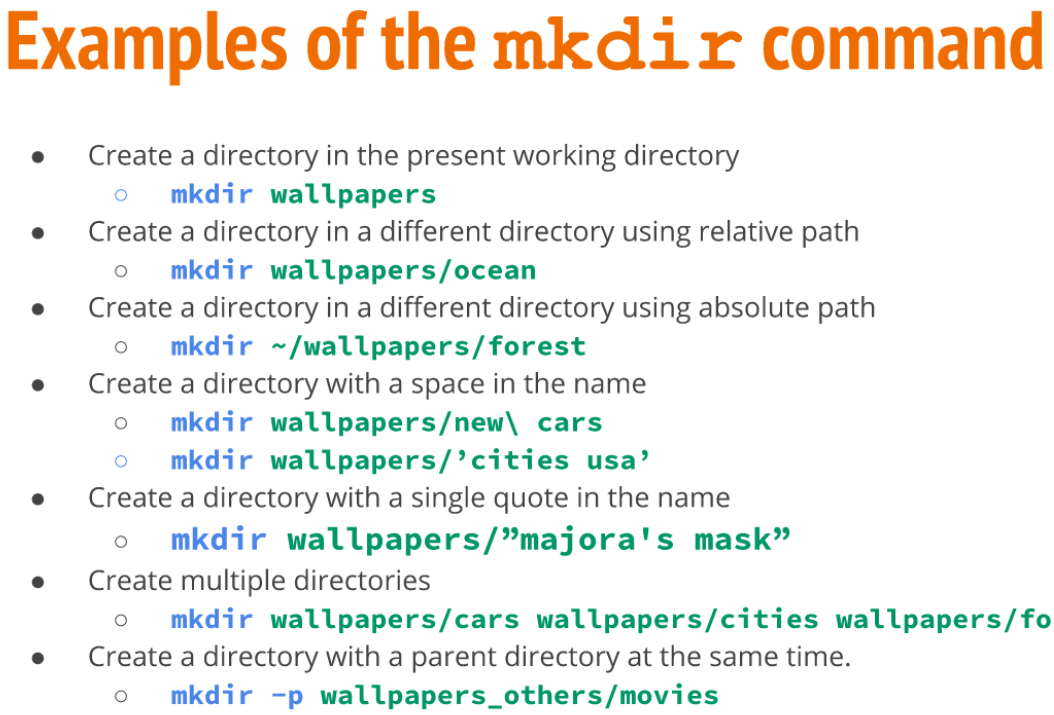
- To download files with wget use the following formula: `wget + url`. **Examples from the labs:**
  - This example downloads the file lab4.zip then unzips the lab4.zip and then removes the lab4.zip file
    - `wget https://robertalberto.com/public/lab4.zip && unzip lab4.zip && rm lab4.zip`

How to download and run scripts using curl

- To download and run a script with curl use: curl + url | bash
  - This example downloads the script lab4.1.sh and then pipes it to bash for execution
    - `curl https://robertalberto.com/cis106/lab4.1.sh | bash`

How to use the following commands:

- **mkdir**
  - used for making directories
  - formula: `mkdir + option + directory name`



- more examples in the professors website
  - **tree**
    - used: for displaying a tree of a given directory. Similar to ls but more pretty!
    - formula: `tree + option + directory name or file`
    - common options of the tree command:
      - use command: `tree --help` or `man tree`

How to create multiple files and directories using brace expansion

- What is brace expansion? a feature to expand a command. For examplem, you can create multiple files or directories in a single command:
  - Examples:

    To create a whole directory structure in a single command:
    - `mkdir -p music/{jazz,rock}/{mp3files,vidoes,oggfiles}/new{1..3}`

    To create a N number of files use:
    - `touch website{1..5}.html`
    - `touch file{A..Z}.txt`
    - `touch file{001..10}.py`
    - `touch file{{a..z},{0..10}}.js`

    Remove multiple files in a single directory
    - `rm -r {dir1,dir2,dir3,file.txt,file.py}`

- Examples from lab:
- Create the following directory structure:

```
wallpapers/
├── cars
│   ├── 1080p
│   ├── 2k
│   └── 4k
└── ocean
    ├── 1080p
    ├── 2k
    └── 4k
```

- Solution: `mkdir -p wallpapers/{cars,ocean}/{1080p,2k,4k}`

How to move and remove files and directories using absolute and relative path

Absolute path

- States the file name starting from the root (/). For example: `/home/user/Downloads/game.exe` is the absolute path of the game.exe file.
- Absolute path works from anywhere in the filesystem

Relative path

- Specifies a pathname starting from the current working directory. For example, asuming that my pwd is home, the relative path of game.exe is `Downloads/game.exe`

move a file

to move a file we use the mv command. The basic formula is: `mv + source + destination` where source and destination can be absolute path or relative path.

Examples of moving files and directories

- To move a file from a directory to another using relative path
  - `mv Downloads/homework.pdf Documents/`
- To move a directory from one directory to another using absolute path
  - `sudo mv ~/Downloads/theme /usr/share/themes`
    - *Notice that in this command I am using sudo since the destination is owned by root.*
- To move a file from one directory to another combining absolute path and relative path
  - `mv Downloads/english_homework.docx /media/student/flashdrive/`
    - *Notice that in this command I am moving the file "english_homework.docx" to the directory where the flash drive is mounted.*
- To move multiple directories/files to a different directory
  - `mv games/ wallpapers/ rockmusic/ /media/student/flashdrive/`

- mv can also rename. the formula is the same. Example:

Examples of renaming files and directories

- To rename a file
  - `mv homework.docx cis106homework.docx`
- To rename a file using absolute path
  - `mv ~/Downloads/homework.docx ~/Downloads/cis106homework.docx`
- To move and rename a file in the same command
  - `mv Downloads/cis106homework.docx Documents/new_cis106homework.docx`

How to move and remove directories using:

- ~ is a shorthand for the currents user home directory.
  - This pathname: `~/Downloads` is the same as `/home/user/Downloads`
- ../ represents one directory back. In otherwords, it can be used to navigate backwards. For example, if my pwd is `~/Downloads/games/fps` I can go back two directories to `~/Downloads` using `cd ../../`

How to save the output of a command to a file

- to save the ouput of a command to a file use: `>` for example, to save the output of `ls ~/Downloads/games` to a file name `list-of-games.txt` use this command: `ls ~/Downloads/games > list-of-games.txt`

How to append the output of a command to a file

- append means: add (something) as an attachment or supplement. To append the ouput of a command to a file that already has data in it use `>>` for example:

- To redirect standard output and append the output to a file, we use: >>
  - **Example:**
    - `ls -1alh >> list_of_files.txt`

How to use basic commands (mkdir, mv, cp, ls) to organize files in directories and subdirectories

- mkdir = creates directories
- mv = moves and rename files
- cp = copies files
- ls = list files Examples of these commands can be found here:
- presentation managing files mv cp
- presentation ls
- professors website

How to use vim to add text to a file

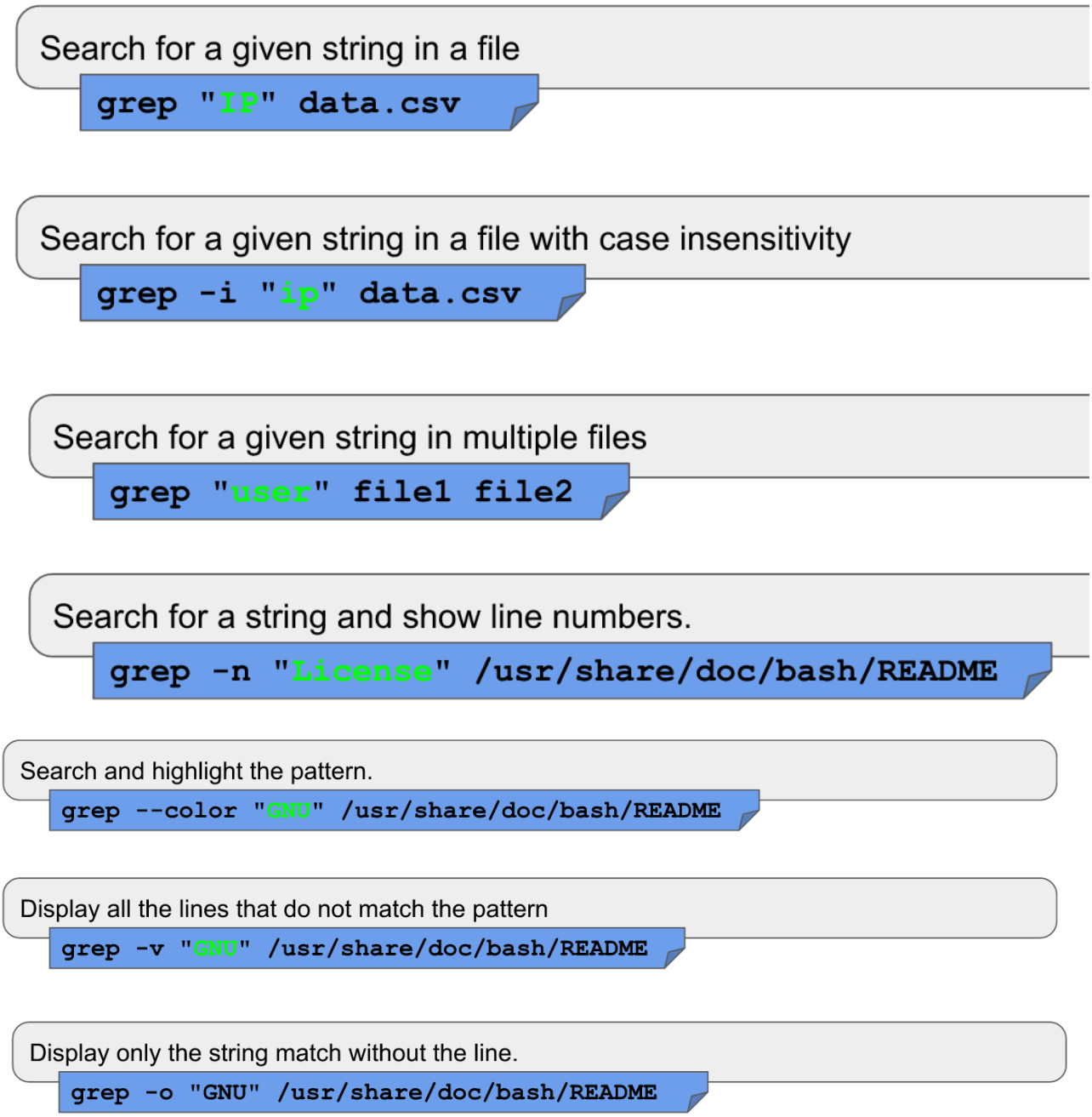
- Presentation on vim
- cheat sheet vim

How to use grep to search for strings inside a file

- grep is a command used for matching strings:
- formula 1 working with a file: `grep + option + string to search + file`
- formula 2 working with a command output: `command output | grep + option + string to search`
- common options:

Option	Explanation
-i	Turns case sensitivity off
-n	Displays line number of the each line matched
-E	Treats the pattern as an extended regular expression
-G	Treats the pattern as a basic regular expression
-v	Inverts the search
-o	Only display the string matched

- Examples of grep:



Check: https://robertalberto.com/linuxcommands/commands/grep.html

How to rename files and directories

- to rename files use the mv command:
- formula: `mv + current file name + new file name`
- Examples:

Examples of renaming files and directories

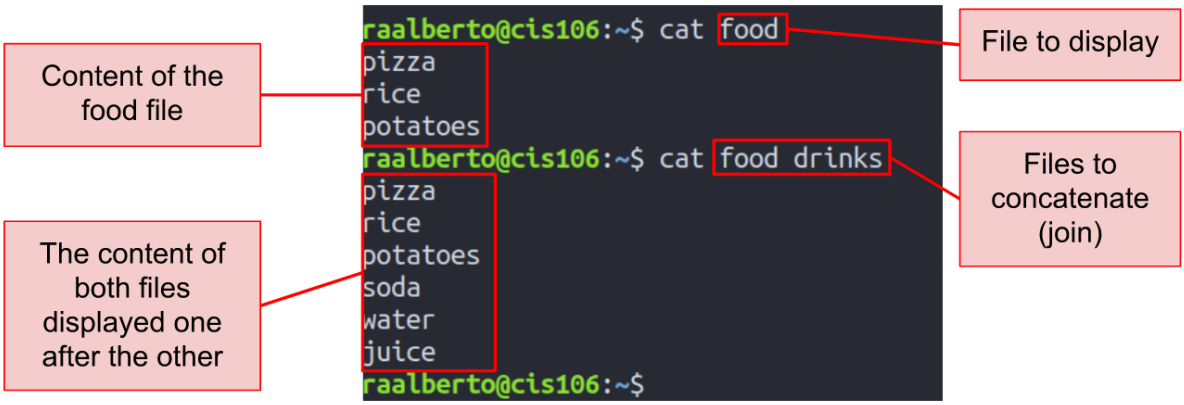
- To rename a file
  - `mv homework.docx cis106homework.docx`
- To rename a file using absolute path
  - `mv ~/Downloads/homework.docx ~/Downloads/cis106homework.docx`
- To move and rename a file in the same command
  - `mv Downloads/cis106homework.docx Documents/new_cis106homework.docx`

How to know the size of file

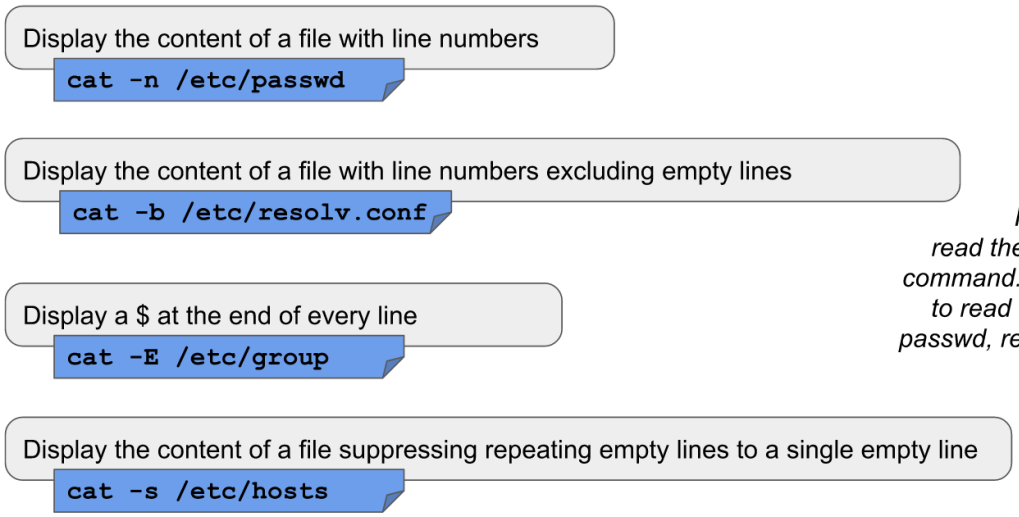
- The stat command can show this information. Example: `stat file`
- `ls -lh file` or `exa -l file` or `tree -h file` shows the file size with human readable format.
- if the file is a text file, a word count with wc does the job.

How to use the cat command

- The cat command is used for **displaying the content of a file**.
- Cat is short for **concatenate** which is the the command intended use.
- Concatenation means joining two strings together.
- Usage:
  - **cat + file to display**
  - **cat + file 1 + file 2**



- Examples:



For more information, read the man page of the cat command. Also take some time to read the man pages of the passwd, resolv.conf, group, and hosts files.

Difference between an absolute path and relative path

Absolute path

- States the file name starting from the root (/). For example: `/home/user/Downloads/game.exe` is the absolute path of the game.exe file.
- Absolute path works from anywhere in the filesystem

Relative path

- Specifies a pathname starting from the current working directory. For example, asuming that my pwd is home, the relative path of game.exe is `Downloads/game.exe`

How to get the inode number of a file

- The stat command can show this information. Example: `stat file`
- `ls -i file` or `exa -i file` or `tree --inodes file` shows the file inode number.

How to know when was a file last modified

- the `ls -l` or `stat` command provide this information.

How to display the absolute path of a file

- ls + file using absolute path

How to use the cut, head, and tail commands with the pipe to filter the output of a command

- cheat sheet of head
- cheat sheet of tail
- cheat sheet of cut
- **Head** = displays the first 10 lines of a file or a given number
- **Tail** = displays the last 10 lines of a file or a given number

- **Cut** = the cut command is used to extract a specific section of each line of a file and display it to the screen

the pipe

Allows you to send the output of a command as input to another command. Examples:

Use grep to look for a string in a particular man page

```
man ls | grep "human-readable"
```

Display only the options of the of any command from its man page

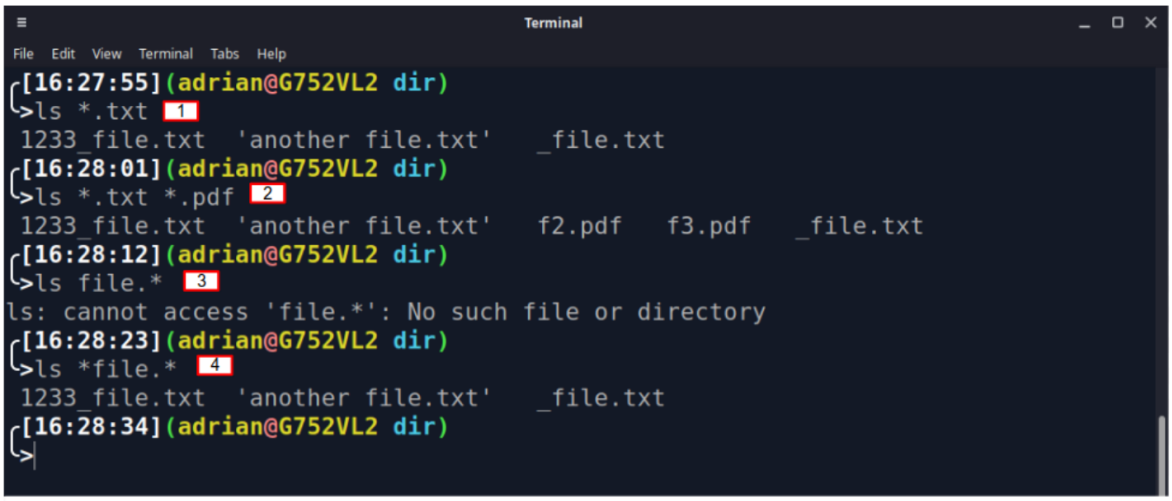
```
man ls | grep "^[[:space:]]*[[:punct:]]"
```

Display all IP addresses from the output of the ip command

```
ip addr | grep -Eo '[[:digit:]]{1,3}\.[[:digit:]]{1,3}\.[[:digit:]]{1,3}\.[[:digit:]]{1,3}'
```

How to use wildcards to match specific file names: *,?,[],{}

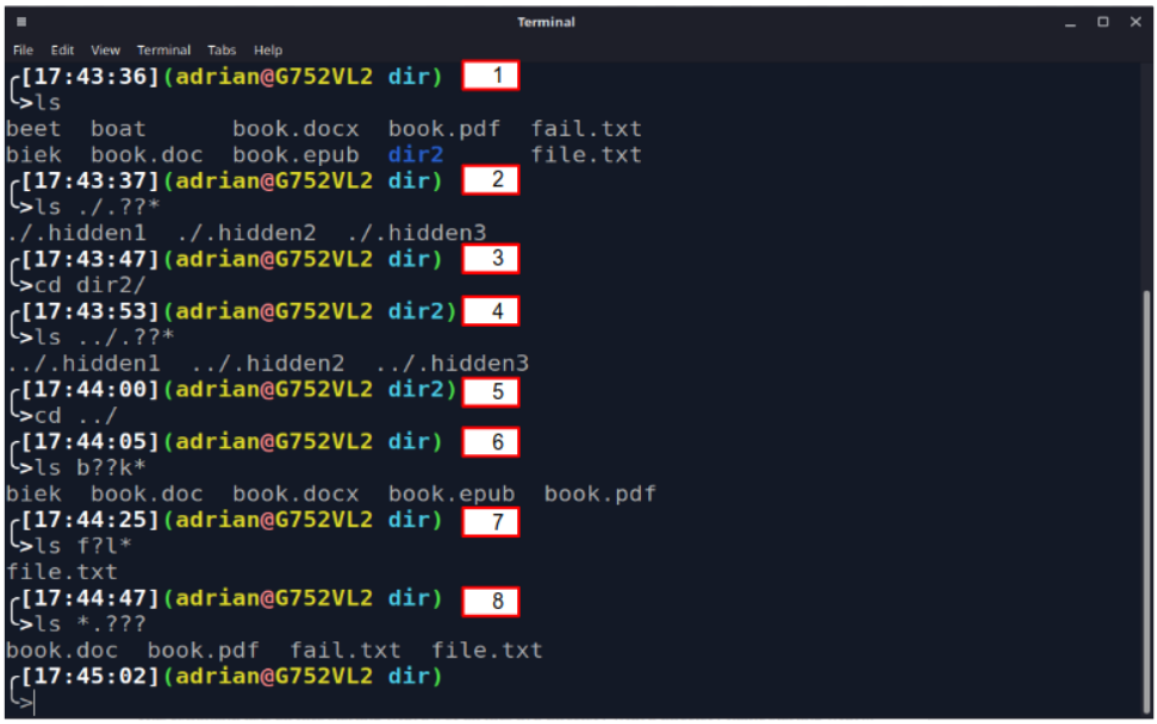
- The * wildcard
  - matches anything and nothing - any number of characters.
  - Examples:



1. ls *.txt lists all the files that end in .txt
2. ls *.txt *.pdf list all the files that end in .txt and .pdf
3. ls file.* lists all the files that start with the string “file.” regardless of their file extension. In this example, there were no files in the directory that matched this criteria.
4. ls *file.* list all the files that have any letter before the string “file.” and after as well.

- The ? wildcard
  - Matches only one character at a time.

- Examples:



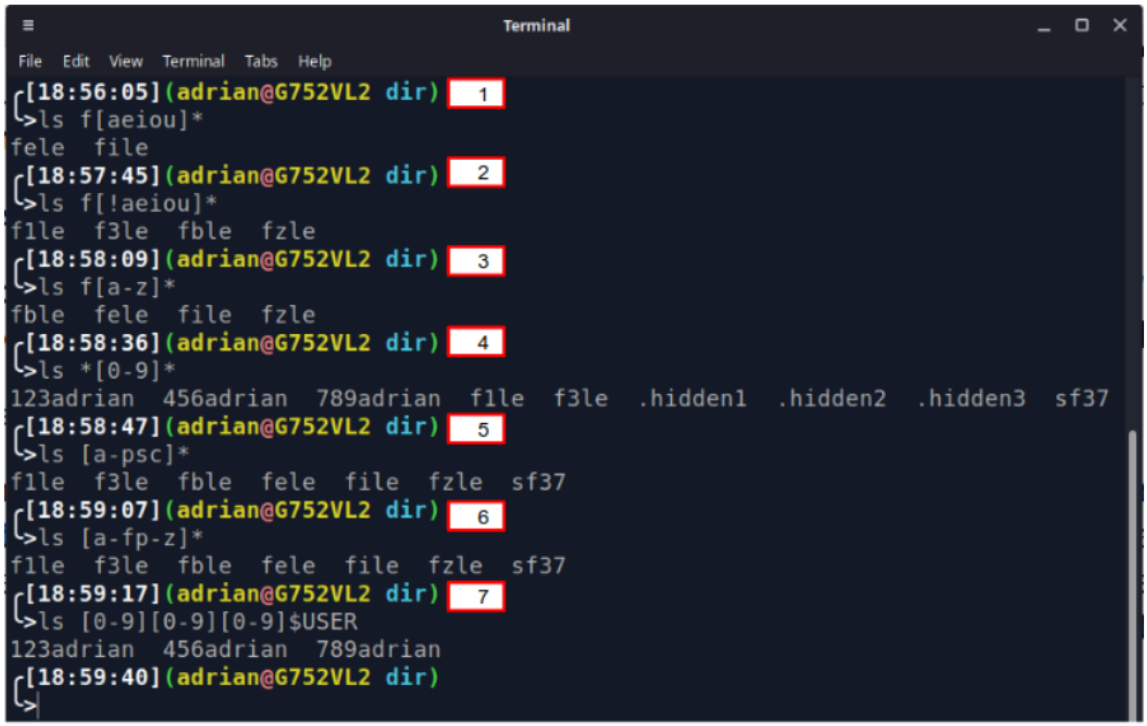
1. List all the files in the current directory (excluding hidden files)
2. List all the hidden files in the current directory
3. Changes the current working directory to dir2
4. List all the hidden files in the parent directory
5. Changes the current working directory to the previous directory (dir)
6. List all the files that have a two character between letter b and k.
7. List all the files that have a single character between letter f and l.
8. List all the files that have a 3 letter file extension.

- The [] wildcard
  - Matches a single character in a range of characters
  - The ! mark is used to reverse the match. For example, match every letter except vowels

    `[!aoiou]`
  - Examples:

Examples:

- To match all files that have a vowel after letter f:
  - `ls f[aeiou]*`
- To match all files that do not have a vowel after letter f:
  - `ls f[!aeiou]*`
- To match all files that have a range of letters after f:
  - `ls f[a-z]*`
- To match all files whose name has at least one number:
  - `ls *[0-9]*`
- To match all the files whose name does not have a number in their file name:
  - `ls *[!0-9].*`
- To match all files whose name begins with a letter from a-p or start with letters s or c:
  - `ls [a-psc]*`
- To match all files whose name begins with any of these two sets of characters: letters from a-f or p-z:
  - `ls [a-fp-z]*`
- To match all files whose name begins with any 3 combination of numbers and the current user's username:
  - `ls [0-9][0-9][0-9]$USER`



1. To match all files that have a vowel after letter f
2. To match all files that do not have a vowel after letter f
3. To match all files that have a range of letters after f
4. To match all files whose name has at least one number
5. To match all files whose name begins with a letter from a-p or start with letters s or c
6. To match all files whose name begins with any of these two sets of characters: letters from a-f or p-z
7. To match all files whose name begins with any 3 combination of numbers and the current user's username

Wildcard	Description
*	Matches zero or more characters in a filename
?	Matches any one character in a filename
[acf]	Matches one of multiple characters in a filename; in this example, a, c, or f
[a-f]	Matches one of a range of characters in a filename; in this example, any character from a through f
[!a-f]	Matches filenames that don't contain a specified range of characters; in this example, filenames that don't contain a through f

- {} Expansion
- What is brace expansion? a feature to expand a command. For example, you can create multiple files or directories in a single command:

- Examples:
  To create a whole directory structure in a single command:
  - `mkdir -p music/{jazz,rock}/{mp3files,vidoes,oggfiles}/new{1..3}`

  To create a N number of files use:
  - `touch website{1..5}.html`
  - `touch file{A..Z}.txt`
  - `touch file{001..10}.py`
  - `touch file{{a..z},{0..10}}.js`

  Remove multiple files in a single directory
  - `rm -r {dir1,dir2,dir3,file.txt,file.py}`

- Examples from lab:
- Create the following directory structure:

```
wallpapers/
├── cars
│   ├── 1080p
│   ├── 2k
│   └── 4k
└── ocean
    ├── 1080p
    ├── 2k
    └── 4k
```

- Solution: `mkdir -p wallpapers/{cars,ocean}/{1080p,2k,4k}`

How to manage data with tar and compression tools

- Tar command examples:

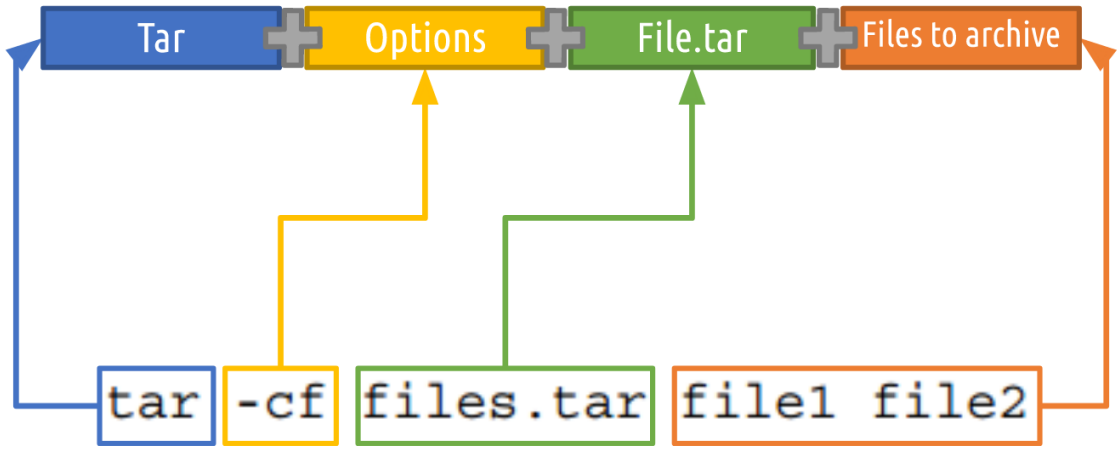
The tar program

Usage:

- To create an archive:
  - `tar + options + archive name + files to add to archive`
- To extract an archive:
  - `tar + options + 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

Tar command example explained



Examples of the tar command

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

- The option -f is always required.
- The -v option displays the details of the operation. It is not required to use it but it is good practice
- Files inside an archive are called members.

- Compression utilities:
  - The gzip, bzip2, and xz commands are used for compression.
  - Examples of file extensions:
    - **Example:**
      - file.txt ----> **file.txt.gz**
      - file.txt ----> **file.txt.bz2**
      - file.txt ----> **file.txt.xz**

File Compression | GZIP

Action description	Example
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`

File Compression | Other useful commands

- gunzip = gzip -d
- bunzip = bzip2 -d
- unxz = xz -d
- To decompress in a different directory use:
  - **Compression utility -options < path of compressed file > file with the same name without extension**
  - Example:
    - **`gzip -dkc < ~/compressedfile.iso.gzip > ~/Downloads/compressedfile.iso`**

bzcat - displays a file that has been compressed with bzip2

- **Example:** `bzcat file.bz2`

bzip2recover - limited data recovery from media errors.

- **Example:** `bzip2recover file.bz2`

zcat - displays a file that has been compressed with gzip

- **Example:** `zcat file.gz`