DATA 3401
Python for Data Science

Unix Shell & Version Control System

Dr. Max (Masoud) Rostami

Department of Science / College of Science
Data Science Program / College of Science
The University of Texas Arlington, Texas

Unix shell

Objectives:

- Recognize the importance of the Shell terminal for a Data Scientists.
- Operate with a Shell terminal using multiple commands.
- Practice various commands to perform different operations like navigating

directories, files organization, and ….

Origins and Development:

1. Unix was developed at **Bell Labs** in the late 1960s by a team led by Ken Thompson and Dennis Ritchie. It was initially designed to meet the needs of multitasking, multi-user computing.
2. Unix was built around a set of principles, including simplicity, modularity, and the idea of treating everything as a file. This design philosophy made Unix highly flexible and scalable.

Command-line shell offers several advantages over graphical user interfaces (GUIs).

1.Increased Efficiency:

2.Flexibility and Power:

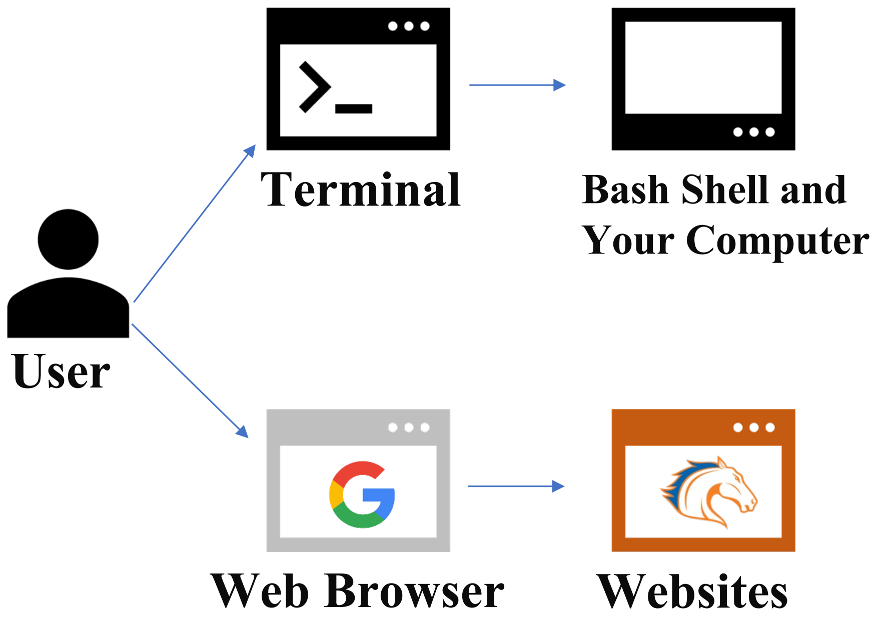
3.Automation and Scripting:

4.Remote System Management:

5.Resource Efficiency:

6.Reproducibility and Version Control:

Terminal


Terminal
Bash Shell and Your Computer
User
Web Browser
Websites

Absolute and Relative paths

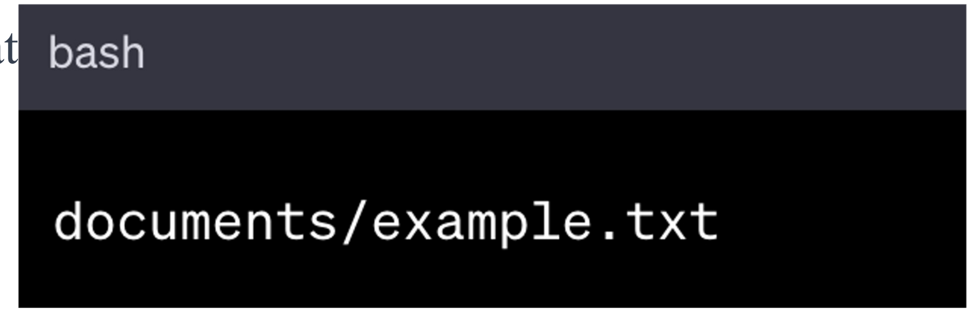
Absolute Path

An absolute path is the complete, exact location of a file or a directory, starting from the root directory.

```
/home/username/documents/example.txt
```

Relative Path

A relative path, on the other hand, is defined in relation to the current working directory. It's like giving directions from your current location. If you're already in the "/home/username" directory, you can access the "example.txt" file using the relat



bash

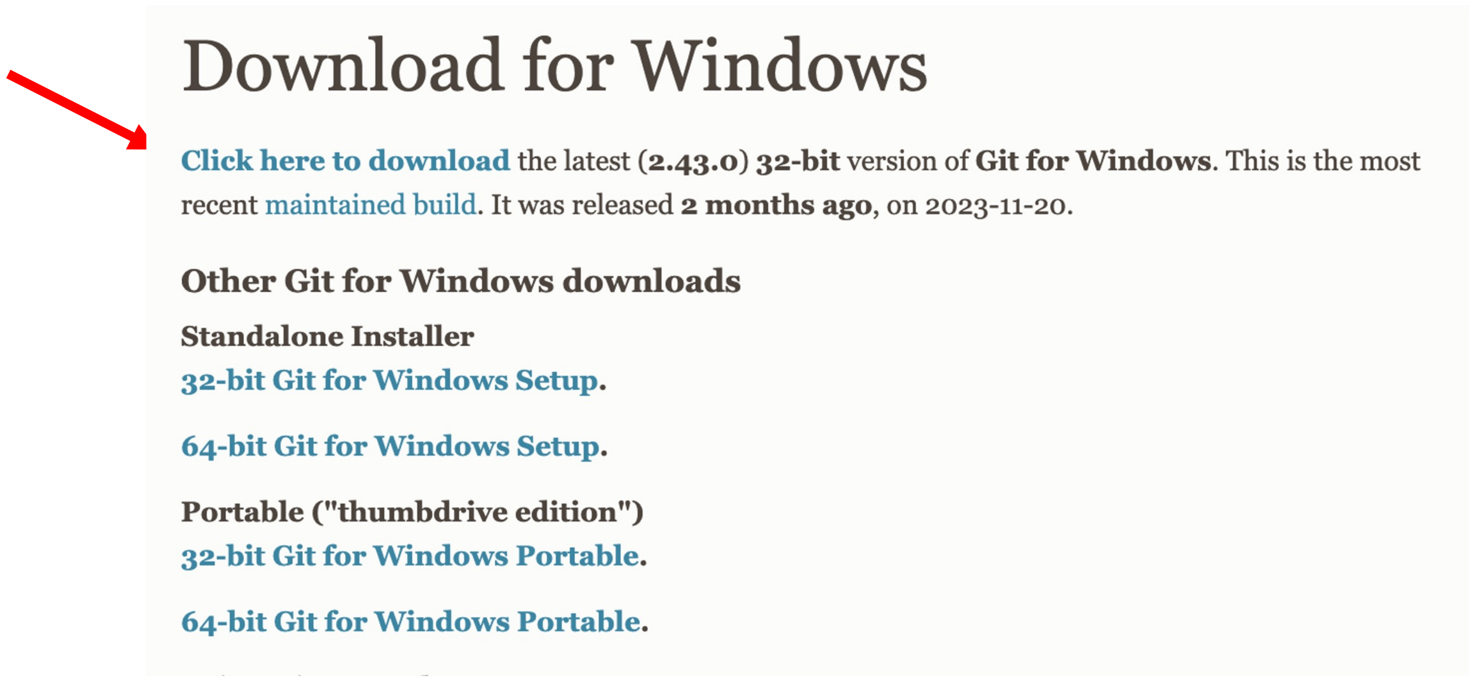
```bash
documents/example.txt
```

Let's Start

Windows: Installing Git Bash

(Windows-only! Mac and Linux users, skip this part)

https://git-scm.com/download/win



Download for Windows

Click here to download the latest (**2.43.0**) **32-bit** version of **Git for Windows**. This is the most recent maintained build. It was released **2 months ago**, on 2023-11-20.

Other Git for Windows downloads

Standalone Installer

32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Portable ("thumbdrive edition")

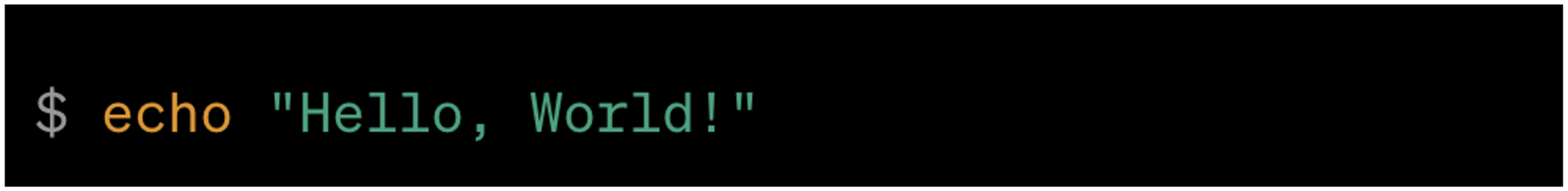
32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

git
bash

Echo (echo):

This command is used to display or print a line of text or string.



```
$ echo "Hello, World!"
```

File Operations

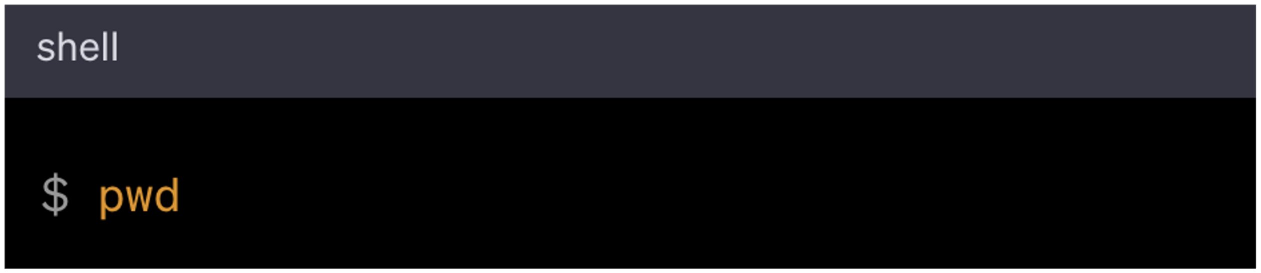
Managing your directories and file structure in your computer or servers is an important skill you need to use in the data science pipelines.

Here, I will show you how to use the ***Shell*** with multiple commands to navigate and organize your files.

1. Navigating directories

pwd (Print Working Directory):

Prints the current working directory.



```shell
$ pwd
```

cd (Change Directory):

Changes the current working directory.

Go to a specific directory: To go to a specific directory, you can specify its path. For instance, to go to the Desktop directory from your home directory, you can do:

Go to Home directory: By default, cd without any arguments will take you to your home directory.

```
cd
```

Go to the home directory with a specific path: To go to a specific directory from your home directory no matter where you currently are, you can prefix the path with ~.

```
cd ~
```

Go back to the previous directory: If you want to go back to the previous directory you were in, you can use - as an argument to cd.

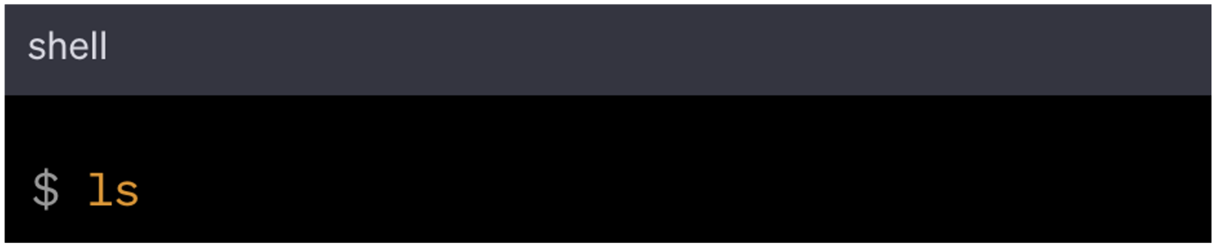
```
cd -
```

Go to the parent directory: If you want to go to the parent directory of your current location, you can use .. as an argument to cd.

```
cd ..
```

ls (List):

Lists all the files and directories in the current directory.



```shell
$ ls
```

Some commands have parameters or options to help you get more information or change the default behavior of those commands.

ls -l: Displays long format listing, which includes file/directory permissions, number of links, owner, group, size, and time of last modification.

```
ls -l
```

ls -a: Lists all files, including hidden files (those whose names start with . in Unix-like operating systems).

```
ls -a
```

ls -t: Lists files sorted by time and date.

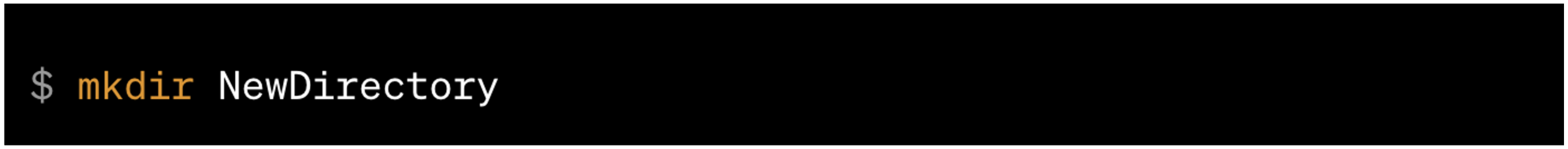
```
ls -t
```

Organizing your files

With the Shell, you can use commands to organize your files into directories, move files, copy or remove the files.

mkdir(Make Directory):

This command is used to create a new directory.



```
$ mkdir NewDirectory
```

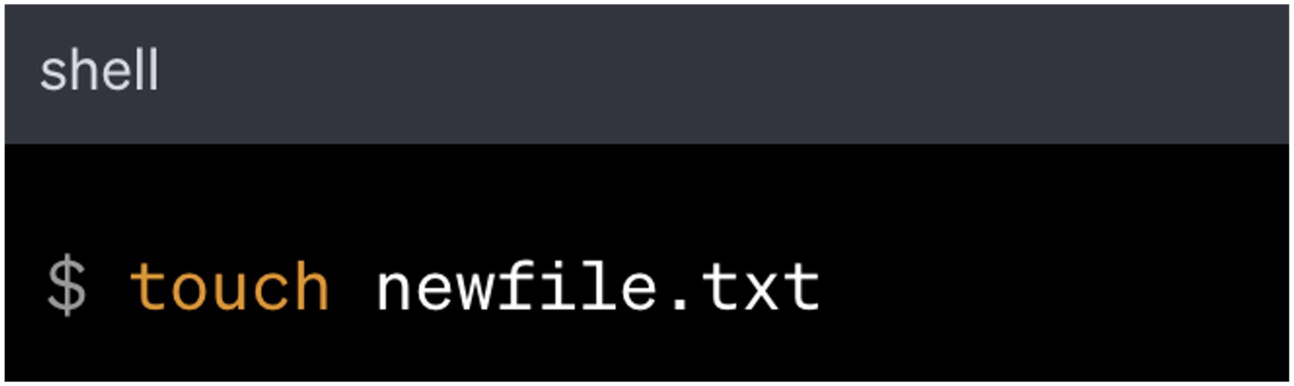
rmdir (Remove Directory):

This command is used to delete a directory.

```
$ rmdir NewDirectory
```

touch:

Creates a new empty file.

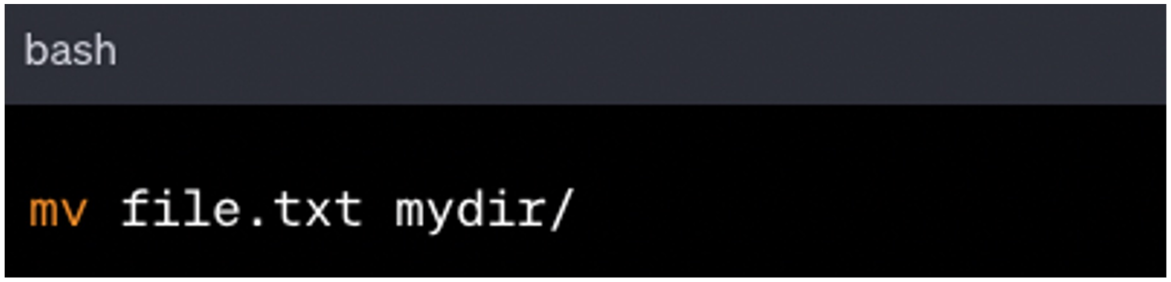


```shell
$ touch newfile.txt
```

mv (Move or Rename):

Moves or renames files and directories.

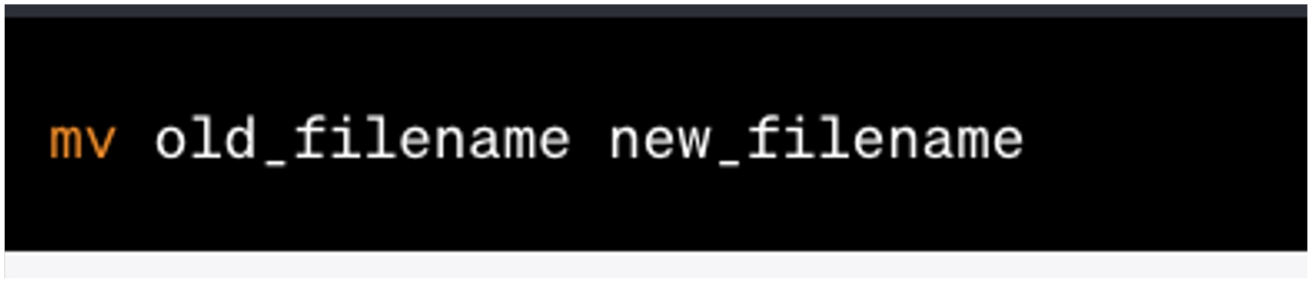
Move the file or files



bash

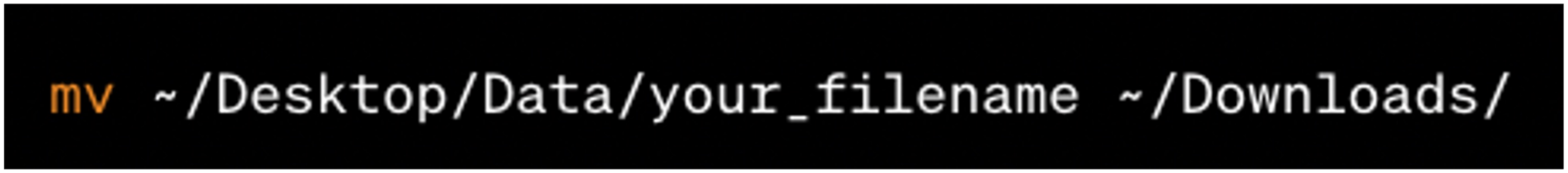
```bash
mv file.txt mydir/
```

Rename



```bash
mv old_filename new_filename
mv ~/Desktop/yourfilename.docx ~/Desktop/book/yourfilename.docx
```

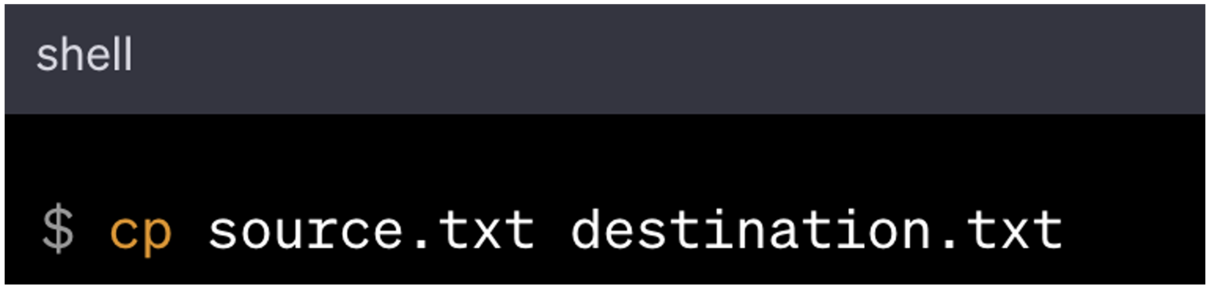
To move a file from your **Data** folder in **Desktop** to the **Downloads** directory using the terminal:



```
mv ~/Desktop/Data/your_filename ~/Downloads/
```

cp (Copy):

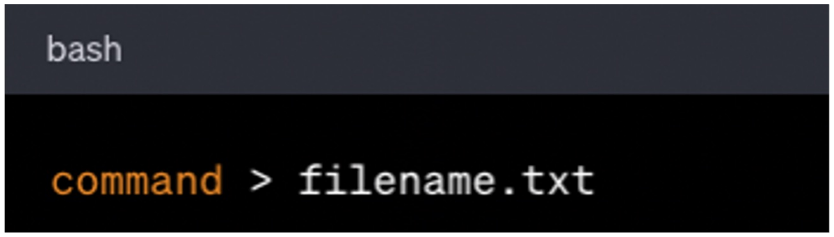
Copies files and directories.



```shell
$ cp source.txt destination.txt
```

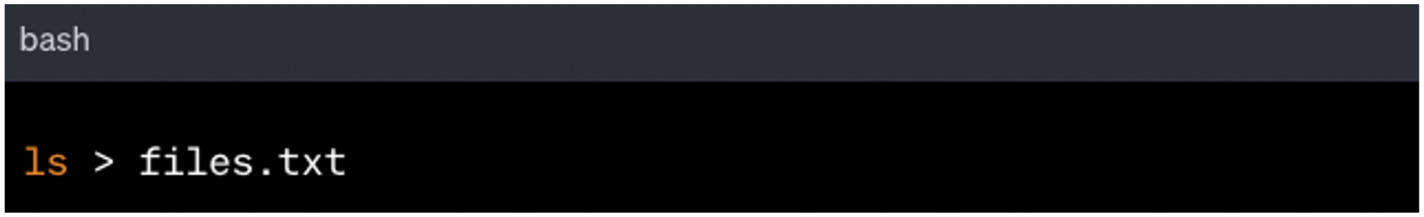
The > operator (redirection): redirect the output of a command to a file.

If you want to save the output of a command to a file, you can use the > operator followed by the desired filename.



```bash
command > filename.txt
```

For example, to save the list of files in the current directory to a file called files.txt, you'd use:



```bash
ls > files.txt
```

Quizzes

Quiz 1: **What does the pwd command do in Unix?**

A. Deletes a file

B. Prints the current working directory

C. Renames a file

D. Lists files in a directory

Quiz 2: What does the cd command do in Unix?

A. Changes the current directory

B. Shows the calendar

C. Copies a file

D. Creates a directory

Quiz 3: Which command is used to list all the files in a directory in Unix?

A. ls

B. mv

C. pwd

D. rm

Quiz 4: Which command in Unix is used to copy a file?

A. cd

B. ls

C. cp

D. pwd

Quiz 6: How would you create a new directory named "test" using a Unix command?

A. cp test

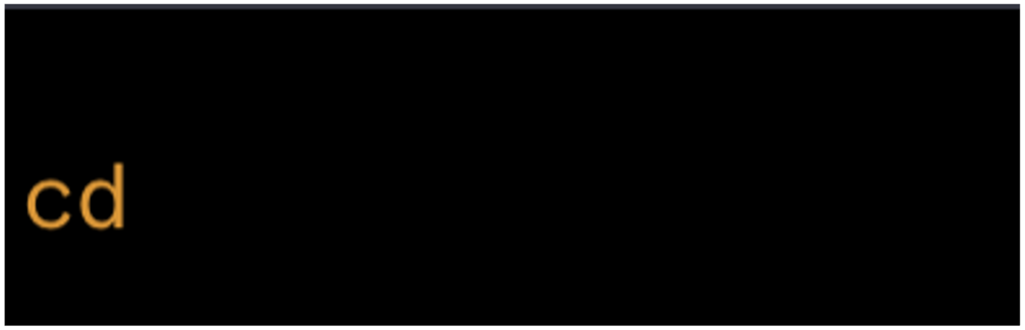
B. rm test

C. cd test
D. mkdir test

Hands on practice

Exercise 1:

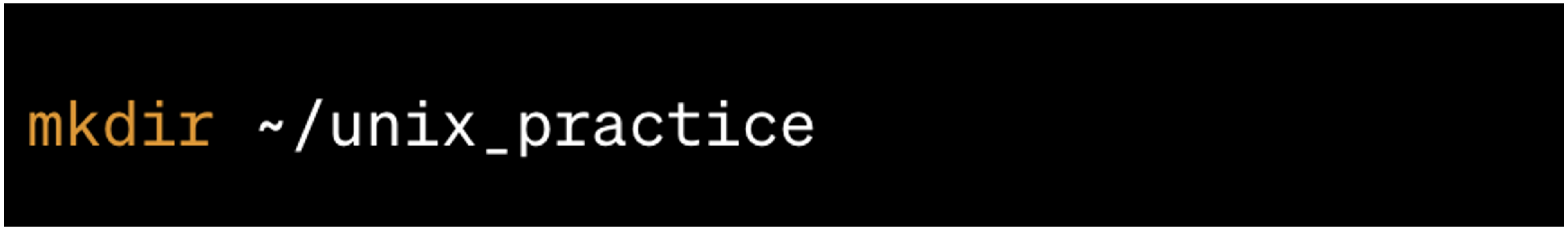
Navigate to your home directory.



```
cd
```

Exercise 2:

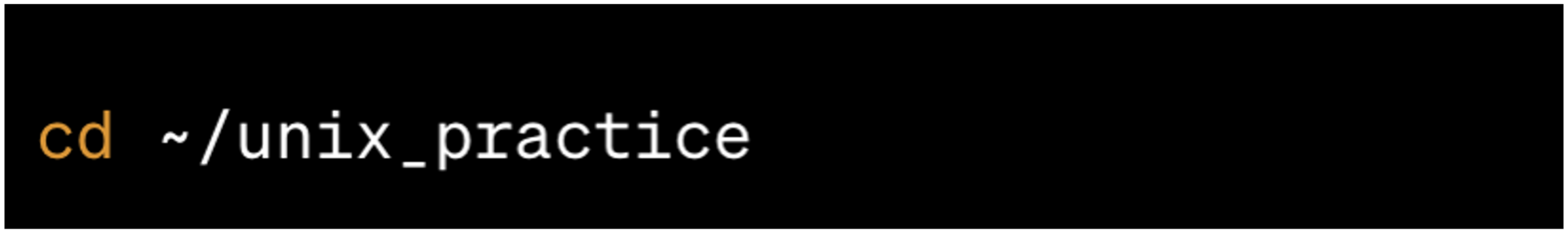
Create a new directory named "unix_practice" in your home directory.



```
mkdir ~/unix_practice
```

Exercise 3:

Navigate into the "unix_practice" directory you just created.



```
cd ~/unix_practice
```

Exercise 4:

Create a new file named "practice.txt" inside the "unix_practice"

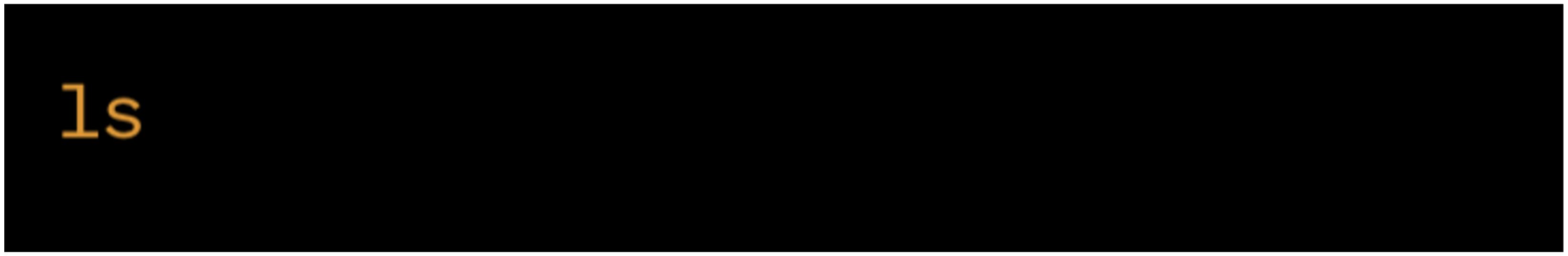
directory.

```
touch practice.txt
```

Exercise 5:

List all the files and directories in the "unix_practice" directory.

Also, show when and what times these file created.



```
ls
```

Exercise 6:

Write "Hello, Unix!" into the "practice.txt" file.

```
echo "Hello, Unix!" > practice.txt
```

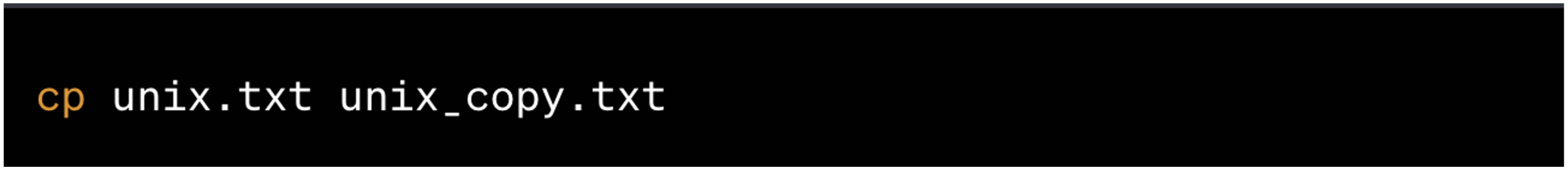
Exercise 7:

Rename the "practice.txt" file to "unix.txt".

```
mv practice.txt unix.txt
```

Exercise 8:

Make a copy of "unix.txt" and name the copy "unix_copy.txt".

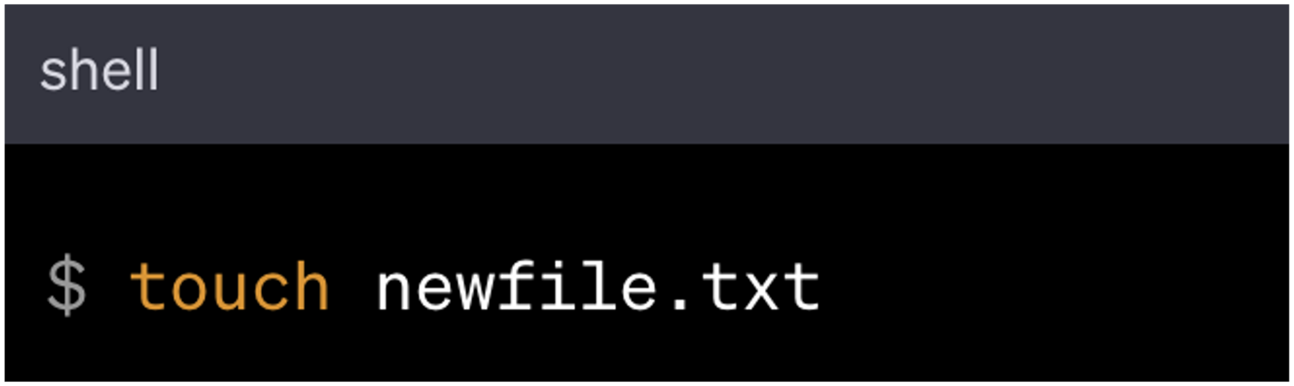


```
cp unix.txt unix_copy.txt
```

Continue on file Operations

touch:

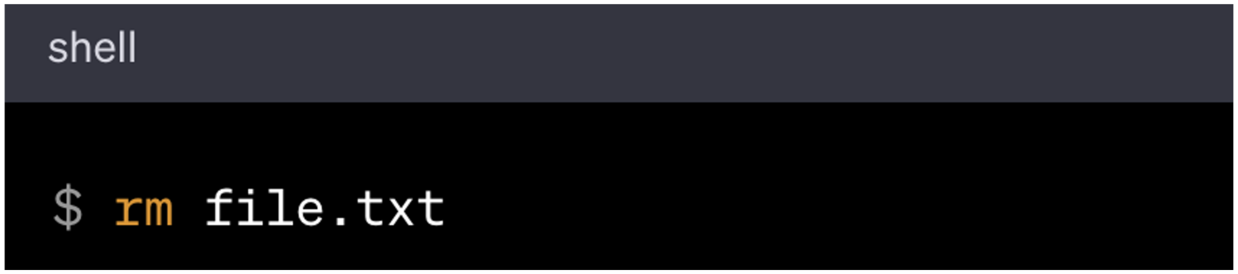
Creates a new empty file.



```shell
$ touch newfile.txt
```

rm (Remove):

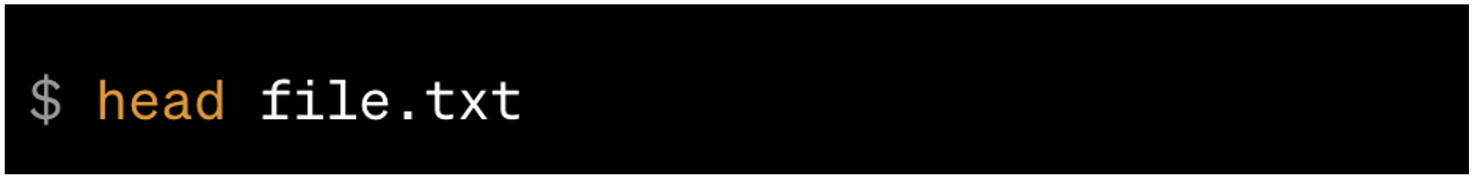
Removes files and directories.



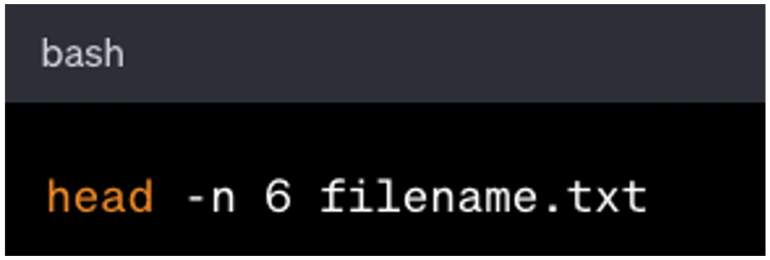
```
$ rm file.txt
```

head:

Outputs the first part of files. By default, it prints the first **10** lines of the specified files



```
$ head file.txt
```

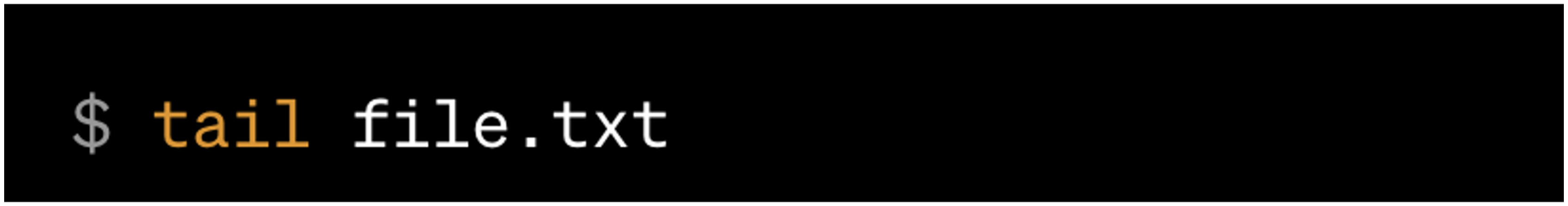


bash

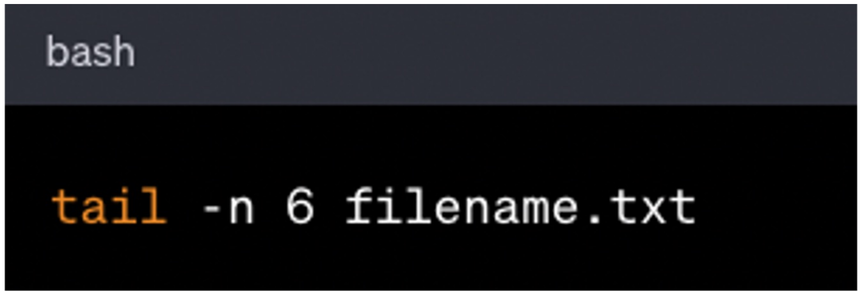
```bash
head -n 6 filename.txt
```

tail:

Outputs the last part of files. By default, it prints the last 10 lines of the specified files.



```
$ tail file.txt
```



```bash
tail -n 6 filename.txt
```

Creating and Deleting Files:

- Create three new files named file1.txt, file2.txt, and file3.txt.
- Confirm the files were created using the ls command.
- Now, use the rm command to delete file2.txt.
- Use ls again to ensure file2.txt has been deleted.

Paging and File Previews:

- Create a new file with more than 20 lines of text. You can do this manually or

paste the text from internet

- Use the head command to display the first 2 lines of the file.

- Now, use the tail command to display the last 3 lines of the file.

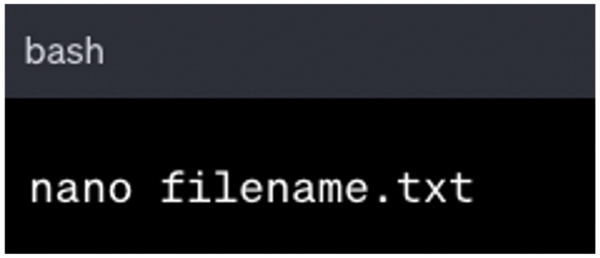
Text Processing

Text Editor

is a software program that allows you to create, view, and modify text files directly within the terminal or console interface, without the need for a graphical user interface (GUI).

nano

To open an existing file or create a new one, use:



bash

```
nano filename.txt
```

Saving Changes:

After editing:

- Press CTRL + O (that's the letter O, not zero). This is the write-out command.
- You'll be prompted at the bottom to confirm the filename. Press Enter to confirm and save.

Exiting:

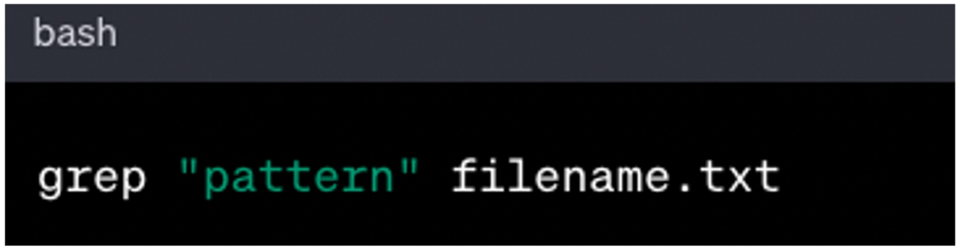
- Press CTRL + X to exit. If you have unsaved changes, nano will ask if you want to save them.

Press Y for Yes or N for No.

grep **(Global Regular Expression Print):**

Processes text line by line and prints any lines which match a specified pattern

Search for a pattern in a file:

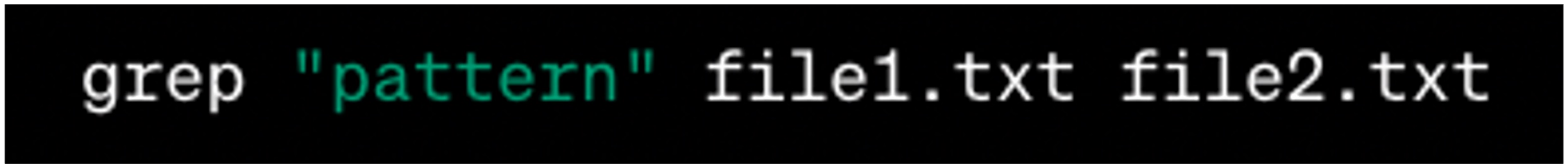


bash

```bash
grep "pattern" filename.txt
```

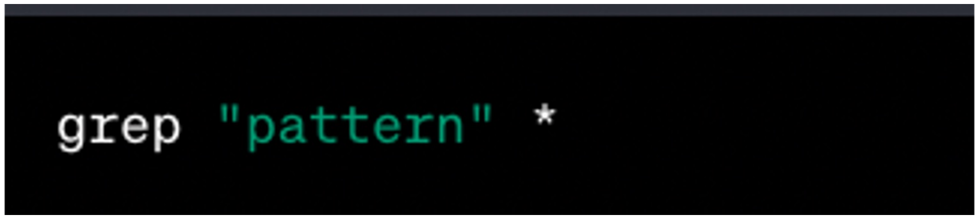
This command searches for the word "pattern" in the filename.txt file and displays all the lines that contain the word.

Search for a pattern in multiple files:



```
grep "pattern" file1.txt file2.txt
```

Search for a pattern in all files in a directory:



```
grep "pattern" *
```

- -i: Ignore case (case insensitive search).

```
grep -i "linux" example.txt
```

```
grep -ic "linux" example.txt
```

- -c: Count the number of lines that match the pattern.

```
grep -c "linux" example.txt
```

- -n: Display line numbers along with the lines that match the pattern.

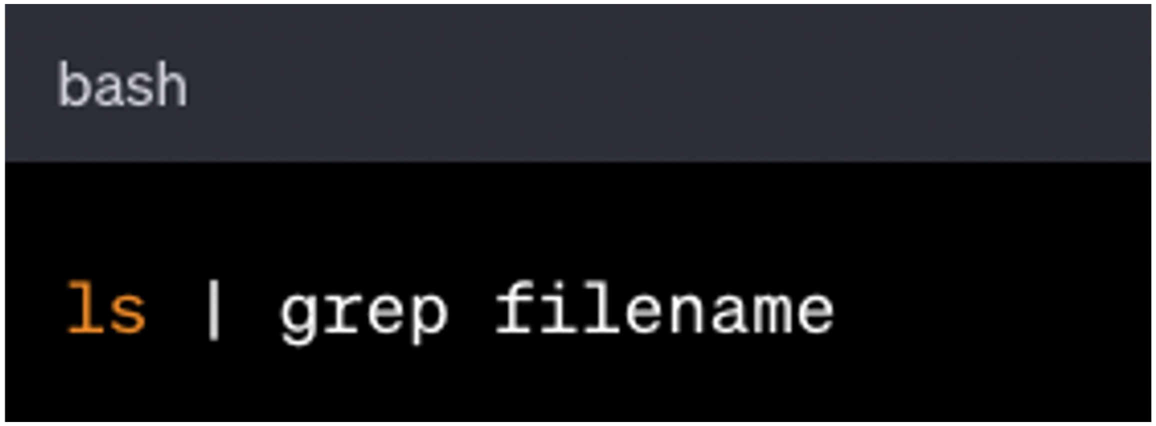
```
grep -n "Linux" example.txt
```

Pipe (|):

The pipe (|) is a powerful tool in Unix-like systems that allows for

the output of one command to be used as the input for another.

For example, to list all files in a directory and then search for a specific filename:

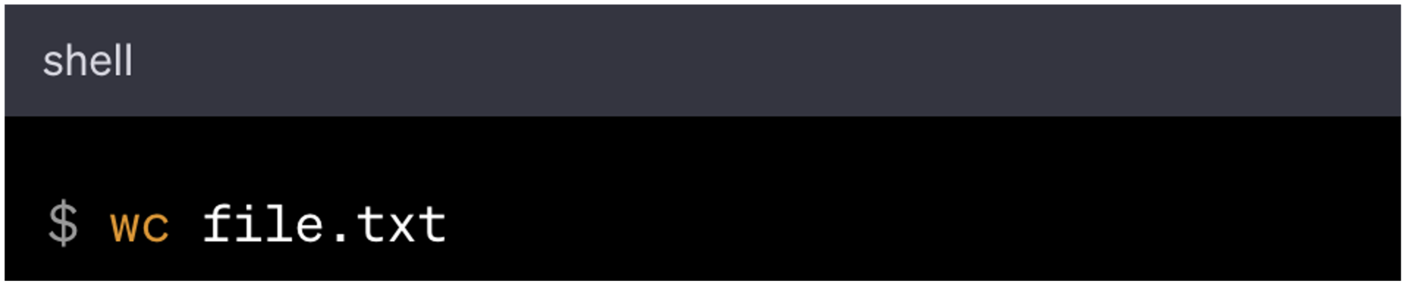


```bash
ls | grep filename
```

wc (Word Count):

Reads either standard input or a list of files and generates one or more

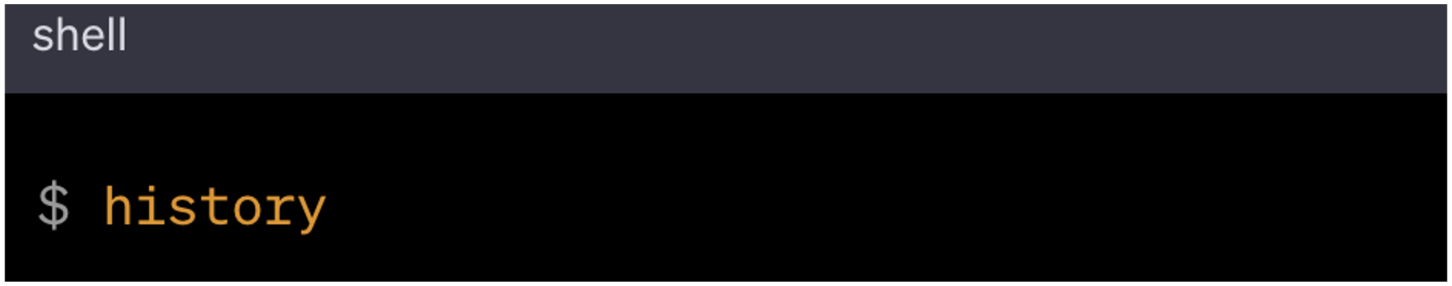
of the following statistics: newline count, word count, and byte count



```shell
$ wc file.txt
```

History (history):

This command displays the command history.



```shell
$ history
```

Hands on practice

- Create three empty files named file1.txt, file2.txt, and file3.txt using the touch command.

```
touch file1.txt file2.txt file3.txt
```

- Add the following lines to file1.txt: apple, banana, cherry

- Add the following lines to file2.txt: apple, orange

- Use grep to search for the word "apple" in file1.txt

```
grep "apple" file1.txt
```

- Use grep combined with a pipe (|) and wc to count how many times the word "apple" appears across both files.

```
grep "apple" file1.txt file2.txt | wc -l
```

- Use the history command to display your recent command history.