

**Linux Commands** 

**Linux Commands** 

#### **Basic Commands**

- pwd (print working directory): Prints the current directory.
- Is: Lists files and directories in the current path.
- cd: Changes the current directory.
- cd ..: Moves up one directory level.
- touch:
- touch fileName.extension: Creates a file.
- touch filename1 filename2 filename3 ...: Creates multiple files.
- touch filename{1..10}: Creates multiple files with sequential numbering.
- touch -d dateFileCreated filename.extension: Creates a file with a

specified creation date.

- echo:
- echo 'text': Prints the specified text.

## **Linux Commands**

- echo 'Hello' > main.txt: Replaces the content inside the file.

• echo 'Hello' >> main.txt: Adds the string to the content inside the file.

## **Linux Commands**

## **Editing Files**

• pico filename: Opens the file in the pico text editor.

- nano filename: Opens the file in the nano text editor.
- gedit filename: Opens the file in the gedit text editor.
- vim filename: Opens the file in the vim text editor.
- cat:
- cat filename: Displays the content of the file.
- cat -n filename: Displays the content of the file with line numbers.

#### **Linux Commands**

#### **File Management**

- shred filename: Encrypts the content inside the file.
- mkdir directoryName: Creates a new directory.
- cp src.txt dst.txt: Copies the content of the source file to the destination file.
- mv: Renames the file or moves the file to another path.
- rm: Removes the file.
- rmdir directoryName: Deletes the directory.
- rm -r directoryName: Removes the directory recursively (with all content inside the directory).
- rm -rv directoryName: Removes the directory and files recursively and prints them in the terminal

## **Linux Commands**

#### **Links and Permissions**

- In -s filename linkName: Creates a soft link (when the original file is deleted, this link is also delet
- In filename linkName: Creates a hard link (when the original file is deleted, this link keeps a copy
- chmod:
- chmod rwx filename: Gives read, write, and execute access to all (user,

STUDENTS-HUB.com

group, others).

• chmod 755 filename: Sets permissions for user (7: read, write, execute),

group (5: read, execute), and others (5: read, execute).

## Linux Commands

**Downloading and Archiving** 

- wget linkDownload: Downloads a file from the internet.
- zip file.zip fileToZip: Zips the specified file.
- unzip fileName.zip: Unzips the specified file.

#### **Linux Commands**

#### **Viewing File Contents**

- head filename: Reads the first 10 lines of the content in the file.
- tail filename: Reads the last 10 lines of the content in the file.
- head -n line\_num filename: Reads the first specified number of lines in the file.
- tail -n line\_num filename: Reads the last specified number of lines in the file.
- wc:
- wc filename: Provides the number of lines, words, and characters in the

## specified file.

- wc -I filename: Returns the number of lines.
- wc -w filename: Returns the number of words.
- wc -c filename: Returns the number of characters.

## **Linux Commands**

## **Sorting and Searching**

- sort filename: Sorts the content of the file according to ASCII values.
- sort -n filename: Sorts the content numerically.
- sort filename -o dstName: Outputs the sorted content to a new file.

STUDENTS-HUB.com

# Uploaded By: Malak Obaid

• uniq:

• uniq src.txt dst.txt: Deletes the repeated lines (used when the lines are sorted).

• grep:

- grep value filename: Searches for "value" in the specified file.
- grep value \*.txt: Searches for "value" in all files with the .txt

extension.

• grep -v value filename: Displays all lines except those containing

"value" in the specified file.

• grep -n value filename: Displays lines containing "value" along with

## **Linux Commands**

their line numbers.

• grep '[pattern]' filename: Searches for a specific pattern within the

file.

• grep "^[pattern]" filename: Finds lines starting with the specified

• grep '^word' fileName return the line which has the word at the begining of it pattern.

- grep "[pattern]\$" filename: Finds lines ending with the specified
  - grep 'word\$' fileName return the line which has the word at the end of it
  - The syntax grep '\<...>\' filename utilizes special character sequences within the regular expression to match whole words. as word before space and after space

#### pattern.

• note: dot means any character

ex: grep 'home...' fileName //This search for a word home followed any at least three character or more



- grep 'home[aA]' fileName //This search for a word home followed either by a or A
- grep '[^0-9]' fileName //This search on everything except the inside square brackets
- to move through an unknown number of characters we use .\*
  - grep '^a.\*122' fileName this will return the lines which starts with 'a' and move through all characters until find 122
- to search for a empty line:



- grep '[-0-9]' filaName used to search for the number and for -
- grep '[t\*]' fileName search for at least one t or more

note: special characters like the dot operator lose its meaning and return to be a normal character by putting it inside a square brackets ca

• to search for a word in a certain syntax

#### **Linux Commands**

#### Processes

- ps:
- ps: Displays the current processes in the computer.
- ps -ef: Displays all the processes currently running.
- kill:
- kill -I: Prints all kill signals as a manual.
- kill -9 processID: Terminates the specified process.

STUDENTS-HUB.com

# Uploaded By: Malak Obaid

## **Linux Commands**

## **Text Manipulation**

- tr:
- echo "hello" | tr 'h' 'j': Translates 'h' to 'j', resulting in "jello".
- echo "hello" | tr -d 'l': Deletes all occurrences of 'l', resulting in "heo".
- echo "hello" | tr -c '[a-z]' ' ': Replaces all non-alphabetic characters with a space.
- echo "hellooo" | tr -s 'o': Squeezes repeated occurrences of 'o' into a single 'o', resulting in "hello".
- echo "hello" | tr '[a-z]' '[A-Z]': Translates lowercase letters to uppercase, resulting in "HELLO".
  - cut:
  - cut -c1-5 filename.txt: Extracts the first five characters from each

line of the file.

• cut -d',' -f1 filename.csv: Extracts the first field from each line of

## **Linux Commands**

the CSV file, using a comma as the delimiter.

- cut -d',' -f1,3 filename.csv: Extracts the first and third fields from each line of the CSV file.
- cut -f2 filename.tsv: Extracts the second field from each line of the
- TSV file, assuming the fields are tab-separated.
- cut -c-1 filename.txt: Extracts the last character of each line from the file.
- pipe (): Used to pass the output of one command as input to another command.



• paste: Concatenates values together.

#### **Linux Commands**

#### Paths

- Relative Path: Specifies the location of a file or directory relative to the current directory.
- Example: If you are in /home/user and you want to access

documents/file.txt, you can use the relative path documents/file.txt.

- Absolute Path: Specifies the location of a file or directory from the root directory /.
- Example: The absolute path for the same file would be

## /home/user/documents/file.txt.

## **Linux Commands Opening Files**

- xdg-open:
- xdg-open filename.txt: Opens the specified file in the default text

editor.

- xdg-open http://example.com: Opens the specified URL in the default web browser.
  - open filename: Opens the specified file inside the directory.
  - diff: give the difference between two file ( according to the content ).
  - history: print the history of command in the terminal
  - top: print the most tasks take high performance from the hardware
  - xargs: to convert input from standard input into arguments to a command.
    Here's the syntax we will use:

command1 | xargs command2





Here xargs takes every file name and convert it to argument to execute it according to the command2

note: single quotes will ignore every thing inside of it

double quotes take the specials

- square brackets for range
- curly brackets: How many times the value entered should be repeated





#### Sed

- command is a single quotes
- sed command can be used to delete a while space:



## **1. Print Specific Lines**

To print specific lines using sed:

sed -n 'Np' filename

N is the line number you want to print.

For example, to print the 3rd line:

sed -n '3p' filename

## 2. Delete Specific Lines

To delete specific lines using sed, you can use:

Print specific line (e.g., line 3)

sed -n '3p' filename



#### Delete specific line (e.g., line 3)

sed '3d' filename

Print lines matching a pattern (e.g., "error")

sed -n '/error/p' filename

Delete lines matching a pattern (e.g., "error")

sed '/error/d' filename

Substitute text (e.g., replace "foo" with "bar")

sed 's/foo/bar/' filename

Substitute text globally (e.g., replace all occurrences of "foo" with "bar")

sed 's/foo/bar/g' filename

Substitute text only on a specific line (e.g., line 3)

sed '3s/foo/bar/' filename

Combine commands (e.g., delete line 3 and replace all occurrences of "foo" with "bar")

sed -e '3d' -e 's/foo/bar/g' filename

The Reference Mostly used:

https://www.freecodecamp.org/news/the-linux-commands-handbook/

@Adnan Odeh

