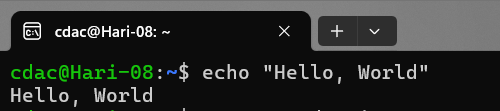
**Part A**

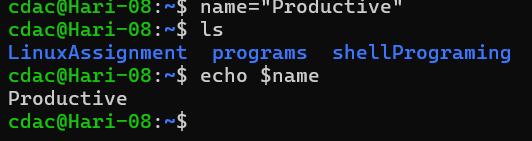
**What will the following commands do?**

**• echo "Hello, World!"**

****

Explanation = echo directly prints the ststement

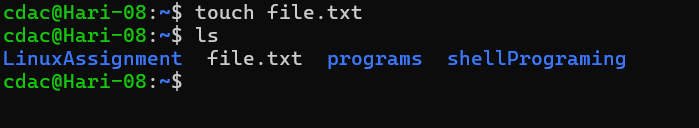
**• name="Productive"**

****

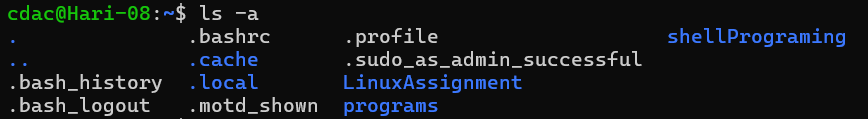
Explanation = it will assign productive to a name variable after giving command echo name it will print productive

**• touch file.txt**

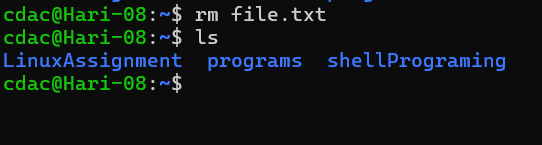
it will creates a file name file.txt



**• ls -a**

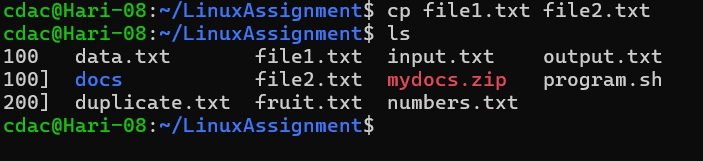
****

**• rm file.txt**

****

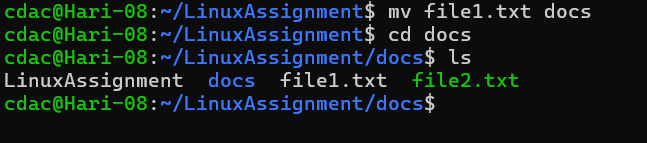
It will remove file or delete file

**• cp file1.txt file2.txt**



This will copy the contents of file 1 to file2

**• mv file.txt /path/to/directory/**

****

It will move one file to another location

**• chmod 755 script.sh**

* When you use chmod 755, it sets the following permissions:
  + Owner (user): Read ®, write (w), and execute (x) permissions.
  + Group: Read ® and execute (x) permissions.
  + Others: Read ® and execute (x) permissions.

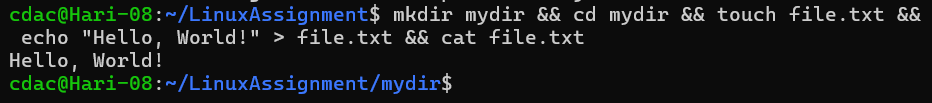
**• grep "pattern" file.txt**

It is used to find a specific word or pattern in a file

**-kill PID**

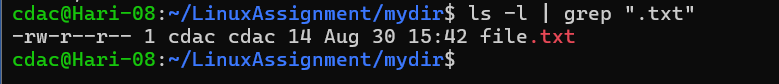
it is used to kill or terminate process by sending signal to them

• **mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt**



It will execute all the commands one by one

**• ls -l | grep ".txt"**



**• cat file1.txt file2.txt | sort | uniq**

* The cat command concatenates the contents of file1.txt and file2.txt.
* It displays the combined content to the standard output (usually your terminal).

**sort:**

The sort command sorts the lines of text alphabetically.

It takes the combined content from the previous step and arranges it in ascending order.

**uniq:**

The uniq command filters out adjacent duplicate lines from sorted input.

It ensures that only unique lines are displayed.

By default, it keeps only the first occurrence of each unique line.

•**ls -l | grep "^d"**

* + The grep command searches for lines in the input that match a specified pattern.
  + In this case, it searches for lines that start with the letter “d” (indicating directories) in the output of ls -l.

So, when you run ls -l | grep "^d", you’ll get a list of directories (folders) within the current directory. The grep command filters out only the lines that represent directories from the long-format ls output.

• **grep -r "pattern" /path/to/directory/**

* The grep command is a powerful text search utility in Linux.
* It allows you to search for specific patterns (text strings) within files.
* The -r (or --recursive) option tells grep to search recursively through directories and subdirectories
* It includes all files within the specified directory and its subfolders.
* Replace /path/to/directory/ with the actual path to the directory where you want to start the search.
* grep will search through all files within this directory and its subdirectories.

• **cat file1.txt file2.txt | sort | uniq –d**

1. sort:
   * The sort command sorts the lines of text alphabetically.
   * It takes the combined content from the previous step and arranges it in ascending order.
2. uniq -d:
   * The uniq command with the -d option filters out only the duplicated lines from the sorted input.
   * It ensures that only the common intersecting lines (duplicate lines) are displayed.

• **chmod 644 file.txt**

The numeric value 644 represents the permissions you’re setting for the file.

Each digit corresponds to a different class of users:

The first digit (6) represents the owner’s permissions.

The second digit (4) represents the group’s permissions.

The third digit (4) represents others’ (everyone else’s) permissions.

• **cp -r source\_directory destination\_directory**

1. cp Command:
   * The cp command stands for “copy.”
   * It allows you to copy files and directories.
2. -r Option (Recursive):
   * The -r (or -R) option tells cp to copy files and directories recursively.
   * When you use -r, it copies not only the specified source file or directory but also all its subdirectories and their contents.

• **find /path/to/search -name "\*.txt"**

Used to find all the file gor given path with “.txt ” extention.

**• echo $PATH**

When you type echo $ the shell prints the entire PATH variable to the terminal.

Ex: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/games:/usr/games