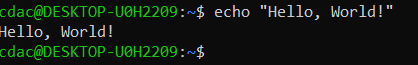
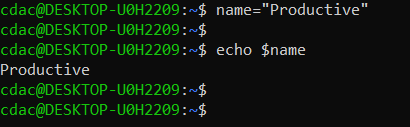
echo "Hello, World!"

* The command echo is used to display anything (string, variables, special characters) on the shell

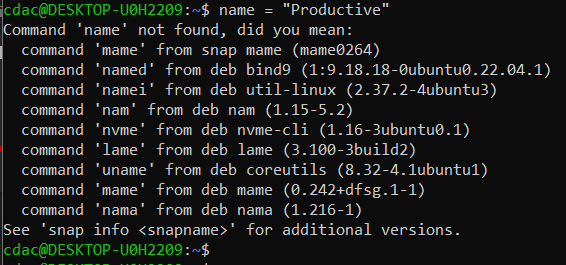


name="Productive"

* This will store the string ‘Productive’ in the variable ‘name’. This variable will keep this value until the shell is not dead (not closed). To display the value of the variable name, we can use echo $name

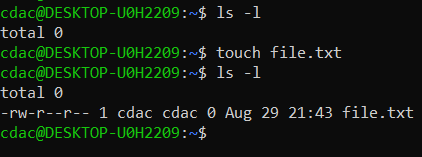


NOTE: There should be no spaces while assigning a value to a variable or below command not found error will occur



touch file.txt

* This will create a file named file.txt



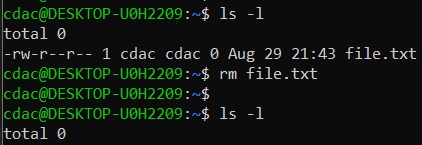
ls -a

* This will list all the hidden files present in the directory as well



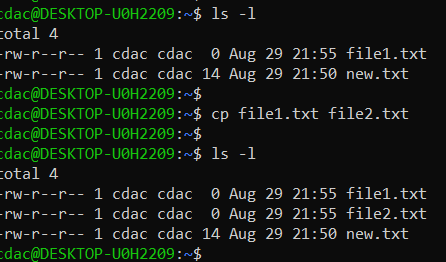
rm file.txt

* This will remove the file that we created above file.txt



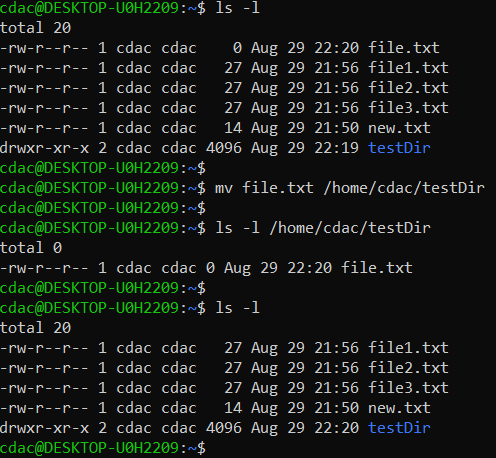
cp file1.txt file2.txt

* This will create a copy of file1.txt named file2.txt (Note: -p option can be used to preserve the timestamp)



mv file.txt /path/to/directory/

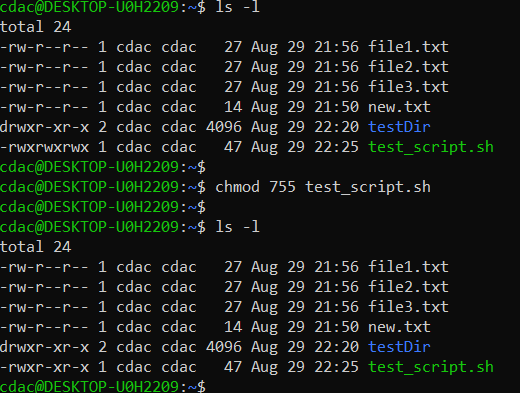
* mv file.txt /home/cdac/testDir will move file.txt to /home/cdac/testDir



chmod 755 script.sh

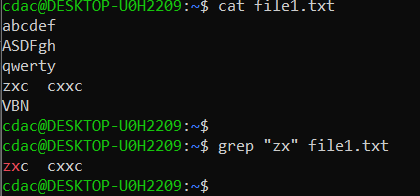
* this will give rwx permission to owner (u), rx permission to group (g) and other (o). 5: r-x, 6: r-w, 7: rwx on script.sh

r – read, w – write, x – execute



grep “pattern” file.txt

* it will search for the provided string in the file and return the line in which the pattern is found

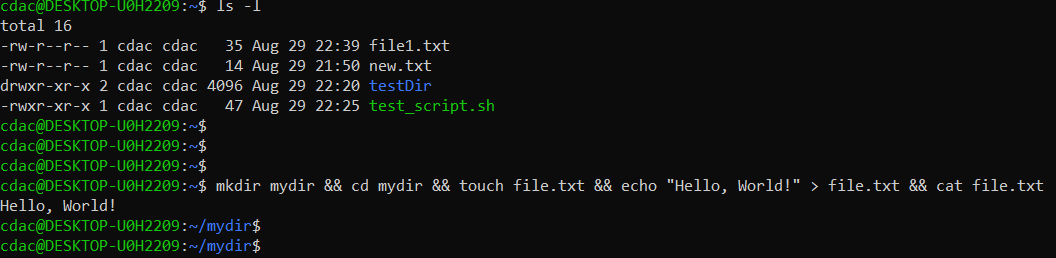


kill PID

* kill command is used to terminate/kill a process by providing the pid (process id)

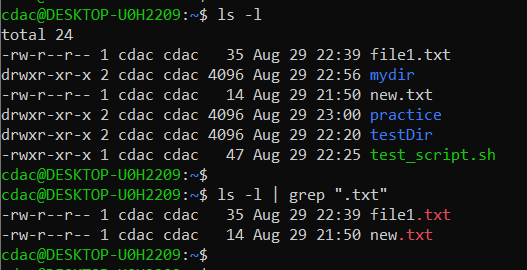
mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt

* multiple commands can be executed one after another in a single line by adding **&&** between them. (note – we can also use ;)
* Here, first mydir directory will be created. Then, cd will change directory to mydir. touch file.txt will create a new file. Hello, World! string is then redirected using > into file.txt. Then cat command will display the contents of file.txt.



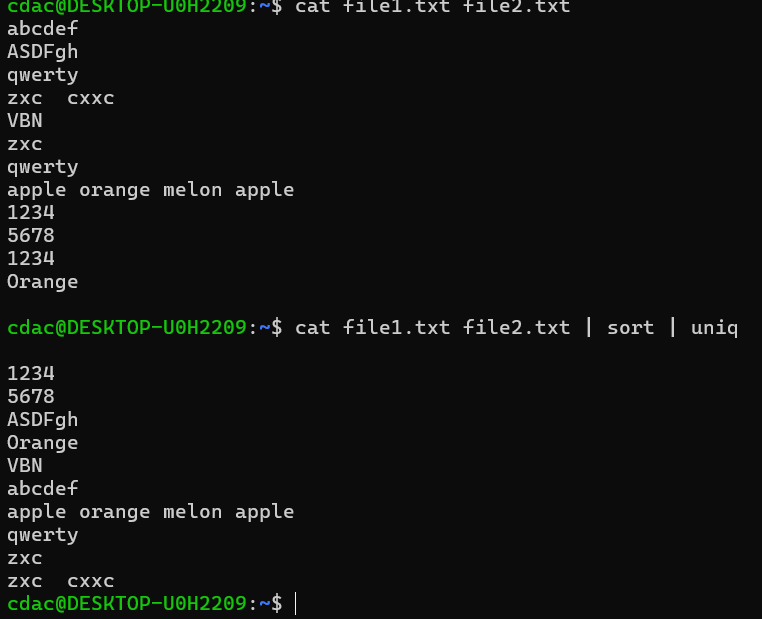
ls -l | grep ".txt"

* the output of ls -l is given to grep command as an input.
* Thus, ls -l will return all the available files and directories in the current path and then grep “.txt” will search for string “.txt” and display all the files (lines) ending with .txt



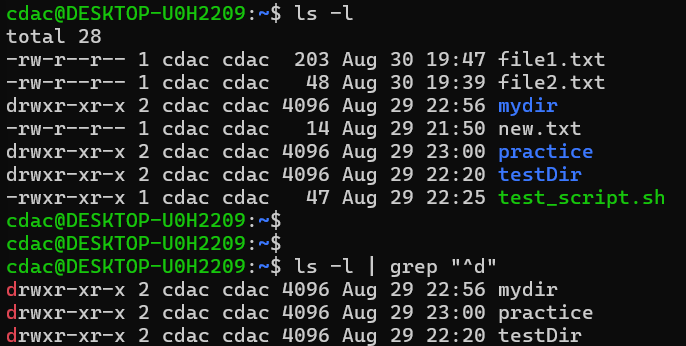
cat file1.txt file2.txt | sort | uniq

* cat command will concatenate the text of both files and output it to sort command which will sort all the lines in ascending order and **uniq** will print only the **unique lines**.



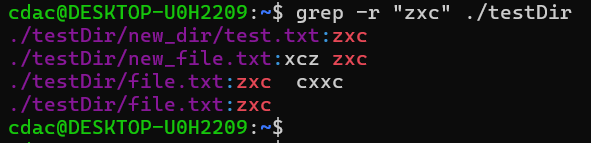
ls -l | grep "^d"

* ls -l will list all the files and directories as output which will be given to grep command. grep will search for all lines **starting with d** (^ means starts with) and return directories only as their 1st permission is d at the (normal files have – as their 1st permission)



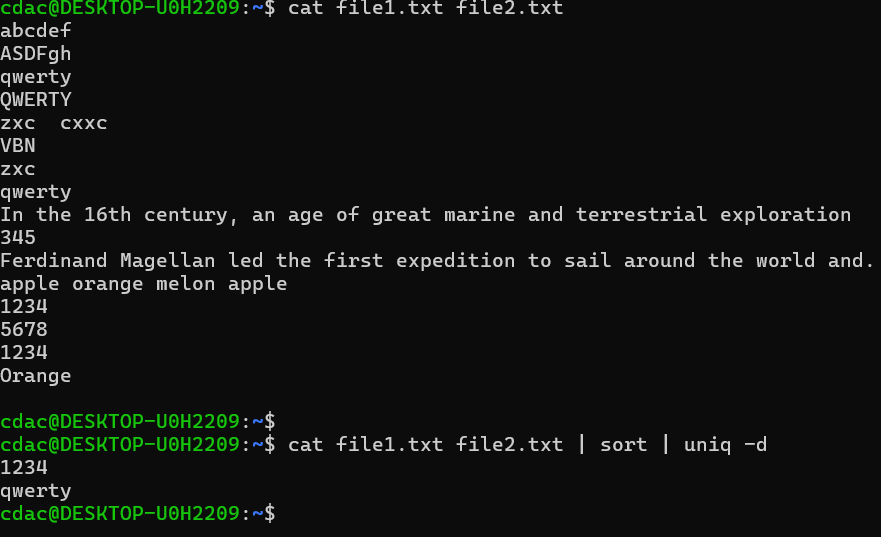
grep -r "pattern" /path/to/directory/

* grep with -r option will search the given pattern of string in all the files available in the given path and all the sub directories



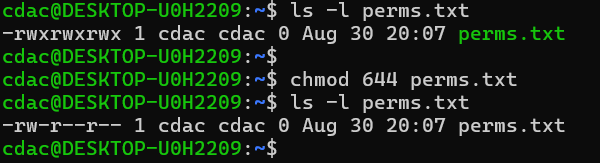
cat file1.txt file2.txt | sort | uniq –d

* cat command will concatenate the text of both files which will be given as input to sort which will sort all the **lines** in ascending order starting with special characters, then numbers, then upper- and lower-case letters. This will be given as input to uniq -d which will display only the duplicated/repeated lines due to -d option.



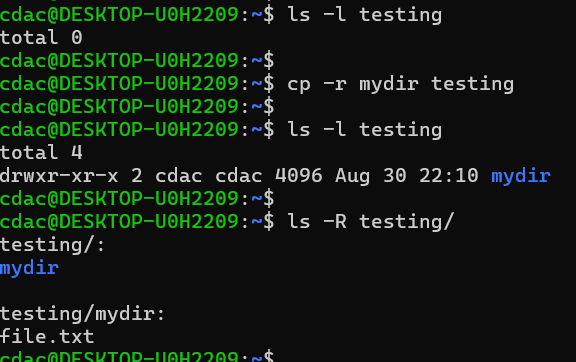
chmod 644 file.txt

* it will change permission of owner to 6 i.e. rw (read write only), change permission of group and others to 4 i.e. r (read only)



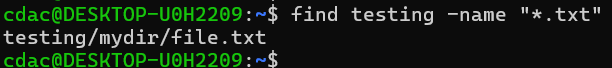
cp -r source\_directory destination\_directory

* -r option is used with cp when entire directory (with all the subdirectories and files) needs to be copied



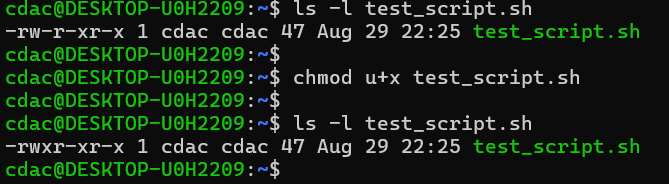
find /path/to/search -name "\*.txt"

* it will find all the files with “.txt” in their name at the given directory and its sub directories



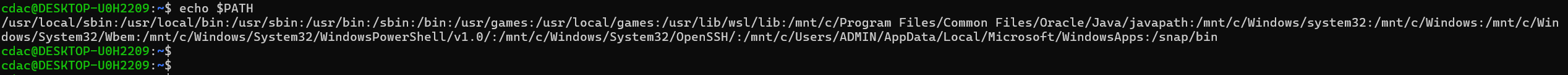
chmod u+x file.txt

* will give executable permission to owner (u) of the file

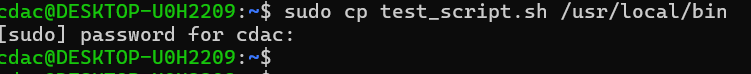


echo $PATH

* it gives all the names of the **directories where if we keep a script file, we can execute that script from anywhere just by using its name** instead of entering the entire path to execute it.



Copied a script to one of the directories set in $PATH



Ran the script by entering only the name of the script file without giving its entire address.

