

## **ASSIGNMENT 2 – PART A**

What will the following command do?

- **echo “Hello World”**

- This command will print the string present in the double quotes i.e. Hello World. Even if quotes aren't provided echo command prints the string provided with it.

- **name="Productive"**

- This command will assign a string literal i.e. Productive to the shell variable named name.

- **touch file.txt**

- touch command will create an empty file. In the above example, touch command will create a file named file.txt.

- **ls -a**

- ls command lists the contents of a current directory. With -a option we can also list hidden files and directories.

- **rm file.txt**

- rm command is used to delete a file or directory (-r option). In the above example, rm command deletes the file named file.txt.

- **cp file1.txt file2.txt**

- cp command is used to copy files and directories. In the above example, the given command copies the contents of file1.txt, creates a file named file2.txt and pastes the content in it.

- **mv file.txt /path/to/directory/**

- mv command is used rename or move a file. In the above example, mv command moves the file (file.txt) into the specified directory (/path/to/directory/). For this command to work these directories must be present in advance.

- **chmod 755 script.sh**

- chmod stands for change modifications. This command is used to assign read, write, and execute permissions to owner, group and other users respectively. The above command gives read, write and execute permissions to the owner and read and execute permissions to group and other users respectively to script.sh file.

- **grep "pattern" file.txt**

- grep command is used to search for specific patterns or regular expressions in text files & display the matching lines. Above given command, searches for the string "pattern" from the file named file.txt.

- **kill PID**

- This command will terminate the process whose PID is mentioned in the command. Since the above command doesn't contain any process id, above command will result in an error.

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

- && (logical AND) operator is used here which enables the user to run multiple commands in single command.
- The above command produces a series of results where output of previous command acts as input for a next command. At first, mkdir command creates a mydir directory in the current directory.
- cd command is then used to change current directory to new created mydir directory. Touch file.txt creates an empty file named file.txt. Further, echo command will display the message "Hello World" on the terminal. This output of echo command is inserted into file.txt using (>) redirect operator.
- Finally, contents of file.txt are displayed using cat command.

- **ls -l | grep ".txt"**

- The above command uses piping to combine the output of both ls and grep command. ls -l is used to display the contents of current directory with details and grep ".txt" command is used to display all the files containing .txt pattern in their name.

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

- The above command uses piping to combine the output of cat sort and uniq commands. First command i.e. cat command is used to display the contents of file1.txt followed by contents of file2.txt.
- sort command is used to perform alphanumeric sort on the result of cat command. Contents of file1.txt and file2.txt are sorted separately in the result.
- Finally, uniq command is use to display only distinct lines in the result.

- **ls -l | grep "^d"**

- ls command lists the files and directories in long format. grep "^d" command filters the output to show only lines that start with "d" which in the ls -l output indicates directories.

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

- Here grep command is used to recursively search for given pattern “pattern” in the directory /path/to/directory, provided that such directory exists in first place. The output will display the lines containing the “pattern” pattern in it.

- **cat file1.txt file2.txt | sort | uniq -d**

- cat command displays the content of file1.txt followed by file2.txt. sort command is used to perform alphanumeric sort on the result of cat command. Contents of file1.txt and file2.txt are sorted separately in the result.
- uniq -d command is used to display only duplicate lines in the previous output.

- **chmod 644 file.txt**

- The above command assigns read and write permissions to owner of the file file.txt and read permission to group users and other users respectively.

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

- The above command is used to copy the source\_directory to destination directory. This is done by using -r option so that all files in source\_directory are copied recursively.

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

- find command is used for searching the files and directories. Given command searches /path/to/search directory and its subdirectories for any file ending with .txt pattern.

- **chmod u+x file.txt**

- This command is used to grant execute permissions for file.txt file to the user(owner) of the file.

- **echo \$PATH**

- This command displays the value of system environment variable that stores directories where executable programs are located.