============================================================================================

**Week: 1**

1. **cd**  -> change directory
2. **pwd**  -> present working directory
3. **ls**  -> list the files in the current directory

**ls -l**  -> list the files but in long listing (detailed manner)

**ls -a**  -> list the files including hidden files.

**ls -la**  -> list the files including hidden file but in long listing.

1. **cal**  -> shows the calender (current month).

**cal 2 2023** -> shows the calender but of specific month and year.

**cal 2023** -> shows the whole calender of 2023 year.

**cal 1**  ->

#Bonus calender in matrix range from from 1 - 9999

1. **date**  -> current date and time (respect to server)
2. **who** -> people connected to the same node.
3. **w**  -> detailed view of the who command including the server up-time.
4. **whoami** -> current username
5. **who am i**  -> same thing as who but just for us.
6. **clear**  -> clears the screen.
7. **exit**  -> logout from matrix.

**logout** -> same thing

============================================================================================

**Week: 2**

1. **tree** -> show the tree view of current present working directory.

**tree -C** -> it shows the tree but in color.

1. **mkdir** -> make directory.

**mkdir** **-p** -> create parent and sub/child directories both.

1. **rmdir** -> remove directory (empty).
2. **rm** -> remove file.

**rm -r** -> remove directory (including non-empty) and file.

1. **cp** -> copy files.

**cp -r**  -> copy directories (including empty and non-empty) and files.

1. **mv** -> move files or directories (both empty and non-empty).

**# -i** Prompt for confirmation if the destination path exists. can be used with cp mv rm.

1. **touch**  -> create text file.
2. **vi** -> text editor.

**ESC**  -> command mode.

**i**  -> insert mode.

**:w** -> Save.

**:x**  -> Save and exit.

**:q**  -> exit if no changes made.

**:q!**  -> exit & discard any changes.

1. **nano**  -> text editor.
2. **cat** -> text reader.

**more**

**less** -> similar to more but also allows forward and backward movement.

**head**

**head -5** -> shows the first five lines.

**tail**

**tail -5** -> show the last five lines .

1. **file** -> tell the type of file. (Helpful for the files which don’t have any e
2. **find**  -> search for the files in a directory hierarchy.

Ex: find ~ -name “output.txt”

1. **diff**  -> compare files line by line.

**diff f1 f2 -y** -> compare two files side by side.

1. **sort** -> sort lines of text file.
2. **uniq**  -> omits same lines.
3. **grep**  -> matches the pattern.

========================================================================

**Week: 3**

1. **Absoulute path ->** /home/pgarg13/slg/output.txt
2. **Relative path** -> output.txt
3. **Relative to home path** -> ~/slg/output.txt
4. **File name expansions**
   1. **\*** -> 0 or more characters
   2. **?** -> exactly one character(any)
   3. **[ ]** -> one character from the range
   4. **[! ]** -> except those characters in the range
5. **echo** -> display a line of text.

**echo red** -> display text red.

**echo red\*** -> show the files starting with red.

**echo $USER** -> show current username.

**echo '\* $USER \*'** -> display \* $USER \* //single quotes strong quotes

**echo “\* $USER \*”** -> display \* pgarg13 \* // double quotes weak quotes

1. **fc** -l -> list previous 16 commands
2. **history** -> list all the commands
3. **who | wc -l** -> gives the number of users or we can say lines in who command.

**Creating a variable:**

1. **number=5** -> variable name is number, and its value is 5(no spaces before or after = sign)

**#** The variables thatyou create exist for the lifetime of terminal if you close it is gone

**school=”Seneca”**

**myseneca=“My school is $school”** -> it will print: My school is Seneca

**echo $myseneca**

**myseneca=’My school is $school’** -> it will print: My school is $school

**37. echo $number**  -> prints the value of number on the screen.

**To store commands in a variable:**

1. **longListing=$(ls -l)**

**longListing=`ls -l`** -> both are same thing

**39. echo $longListing** -> it will execute the ls -l command, but output will be messy

**echo `**

**# To prevent that**

**echo “$longListing”** -> put it in double inverted commas for organised output.

**To print something and execute a command inside it:**

**40. echo “Today’s date is `date`”** -> it will print out : Today’s date is Thu Feb 9 15:04:54 EST 2023

===========================================================================

**Week 4:**

**File Permissions:**

**41. chmod -> change mod to change the file permisson**

**symbolic method:**

**42. chmod -r file.txt** -> removes the read command from user, group and others for file.txt

**43. chmod -x file.txt** -> removes the execute command from user, group and others for file.txt

**44. chmod -w file.txt** -> removes write command from only user

===========================================================================

**Week 5**

**45. grep**  -> print lines that matches the pattern

ex: **grep ‘Mark’ file.txt** -> shows the line which has mark in it.

**grep -i**  -> show the pattern ignoring the case.

**grep -v**  -> reverse searchs

**grep -c** -> prints the number of lines that matches the pattern.

**grep -w** -> prints the line that matches the pattern exactly nothing less nothing more.

**grep -n** -> to get the line number which has the pattern matched.

**46. sort**  -> sort lines of text file(by default first column and ascending order)

**sort -k3**  -> sort the lines of text file based on third column.(but this will not run as if we mention column then also mention delimiter)

# we have to mention by which delimiter column is seperated so,

**sort -t’,’ -k3** -> sort the file based on the third column.

**sort -n**  `-> sort numerically

**sort -u** -> sort but get rid of duplications.

**sort -R** -> sort in random order.

**sort -f**  -> sort file but ignore case.

**52. tr**  -> translate or delete character

# but it only works if we use < symbol before the file name but with pipline it is ok

For ex:

tr ‘,’ ‘-‘ < data.txt -> it replaces all , with -

**53. cut** -> cut the output from the file (doesn’t change the original file)

**For ex**:

**cut -d’,’ -f1,2 data.txt** -> print the first and second field from data.txt which is delimited by ‘,’ .

**54. wc**  -> gives the count for the file.

**wc -m** -> gives the char count for the file.

**wc -l** -> gives the line count for the file.

**wc -c** -> prints the byte count for the file.

**55. >** -> redirect from left to right.

**56. <**  -> redirect from right to left.

**57**. 2> -> redirect error to file(this will overwrite the file)

**58**. 2>> -> redirect error to the file but at the end

# EX:

wc nofile file1 > wordcount 2>> errorlog (combination of redirection of standard output and error )

59. >& -> redirect both error and output in same file

60. | -> redirect standard output.

61. tee -> split the flow to file and standard output,

tee -a -> when splitting to file append to the end of