OS Assignment - 2

Submitted By -: Priyanshu Vishwakarma

Section -: C

Roll No. -: 31

Subject -: Operating System Lab (BCSC 0803)

Submitted To -: Ms. Nidhi

NOTE : GitHub Repo Link

---

1. cut - Command

```
cut   # For cutting sections from each line of files and writing the result to standard output.

SYNTAX : cut Flag Options cmd.txt(File_Name)
```

Flags :

```
-b   # To cut out a section of a line by specifying a byte position use the -b option.
```

OUTPUT :

```
-c   # To cut by character use the -c option. This selects the characters given to the -c option.
```

OUTPUT :

```
-d   # To cut using a delimiter use the -d option. This is normally used in conjunction with the -f option to specify the field          that should be cut.
-f   # select only these fields;  also print any line that contains no delimiter character, unless the -s option is specified.
```

OUTPUT :

```
-s   # Do not print lines not containing delimiters.
```

OUTPUT :

```
--complement   # The --complement option selects the inverse of the options
passed to sort.
```

OUTPUT :

2. uniq - Command

```
uniq   # It can remove duplicates, show a count of occurrences, show only
repeated lines,ignore certain characters and compare on          specific fields

SYNTAX : uniq flag option cmd.txt(File_Name)
```

OUTPUT :

Flags :

```
-c   # To output the number of occurrences of a line use the -c option in
conjunction  with uniq.This prepends a number value to        the output of
each line.

```

OUTPUT :

```
-d   # To only show repeated lines pass the -d option to uniq.
```

OUTPUT :

```
-u   # To only show lines that are not repeated pass the -u option to uniq.
```

OUTPUT :

```
-s   # To return a list of the authors numbers can be ignored by using the -s
option.This will skip the number of characters it          is given before doing
the comparison.
     # For clean output use 'sed' with piping.
```

OUTPUT :

3. chmod - Command

```
chmod   # For change the permission of files or directories.

SYNTAX : chmod Option Permission File_Name

DEFAULT PERMISSION -:

DIRECTORY : Owner -rwx,
            Group -rx,
            Others -rx
FILE      : Owner -rw,
            Group -r,
            Others -r

NOTE : r - read
       w - write
       x - eexcute
```

(i) SYMBOLIC METHOD :

```
Reference : u - owner
            g - group
            o - other
            a - all(u,g,o)

PERMISSION OPRETORS -:

ADD : '+'
remove : '-'
set(Exact) : '='
```

OUTPUT :

(ii) OCTAL METHOD :

```
rwx -:

0 - 000 - NO PERMISSION
1 - 001 - EXECUTE
2 - 010 - WRITE
3 - 011 - WRITE & EXECUTE
4 - 100 - READ
5 - 101 - READ & EXECUTE
6 - 110 - READ & WRITE
7 - 111 - READ, WRITE & EXECUTE
```

OUTPUT :

4. du - Command

```
du   # The du command is a command line utility for reporting file system disk space usage.

SYNTAX : du Flag Dir_Name(Optional)
```

OUTPUT :

Flags :

```
-a   # For all files and dir.
```

OUTPUT :

```
-h   # For human readabilty.
```

OUTPUT :

```
-c   # To view a grand total for a directory pass the -c option.
```

OUTPUT :

```
-sh  # For summary of disk usage with human readabilty.
```

OUTPUT :

```
~/dir1~/dir2   # multiple dir content disk usage.
```

OUTPUT :

---

5. diff - Command

```
diff # Diff command in Linux is used to compare the content of two files line by line.

SYNTAX : diff Flag n1.txt n2.txt(File_Name)
```

OUTPUT :

Flags :

```
-i   # By default this command is case sensitive. To make this command case in-sensitive use -i option.
```

OUTPUT :

```
-u   # To view differences in unified mode, use the -u option.
```

OUTPUT :

6. history - Command

```
history # History command is used to view the previously executed command.

SYNTAX : history Flag
```

OUTPUT :

Flags :

```
-5   # To show the limited number of commands that executed previously as follows.
```

OUTPUT :

```
-d   # History can also be removed using history -d event_number.
```

OUTPUT :

```
-c   # The whole history can be removed using history -c option.
```

OUTPUT :

```
!Event_Number:p   # To print command before executing so that a wrong command does not get executed use :p after event number of                     command.
```

OUTPUT :

7. last - Command

```
last  # The last command in Linux is used to display the list of all the users
logged in and out.

SYNTAX : last Flag
```

OUTPUT :

Flags :

```
-5   # This Flag is used to specify the number(Use Any Number) of lines to
display.
```

OUTPUT :

```
-R   # This Flag is used to hide the host-name field.
```

OUTPUT :

```
-F   # This Flag is used to display the login and logout time including the
dates.
```

OUTPUT :

```
-a   # This Flag is used is to display the host-name in the last column.
```

OUTPUT :

```
-d   # This Flag is used to translate the IP address back into its host-name.
```

OUTPUT :

---

8. sort - Command

```
sort # Sort command is used to sort a file, arranging the records in a
particular order.(Default According to ASCII)

SYNTAX : sort Flag File_Name(cmd.txt)
```

OUTPUT :

Flags :

```
-n  # To sort a file numerically used –n option.
```

OUTPUT :

```
-r  # To sort a file reverse-order used –r option.
```

OUTPUT :

```
-o  # To sort file and write it's output in the other file then we use -o Flag.
```

OUTPUT :

---

9. sed - Command

```
sed  # Sed is a powerful text stream editor. Can do insertion, deletion, search
and replace(substitution).
```

Replacing Text :

```
sed 's/Drago/coco/' cmd.txt    # Sed command is mostly used to replace the text
in a file.
```

OUTPUT :

```
sed 's/Drago/coco/2' cmd.txt   # Use the /1, /2...etc flags to replace the
first, second occurrence of a pattern in a line.
```

OUTPUT :

```
sed 's/Drago/coco/g' cmd.txt   # The substitute flag /g (global replacement)
specifies the sed command to replace all the
     occurrences of the string in the line.
```

OUTPUT :

```
sed '2 s/Drago/coco/g' cmd.txt # For only replace 2nd line word.
```

OUTPUT :

Deleting Line :

```
sed 'nd' cmd.txt   # To Delete a particular line say n(Any_Number) in this
example.
```

OUTPUT :

```
sed '$d' cmd.txt   # To Delete a last line.
```

OUTPUT :

```
sed 'x,yd' cmd.txt # To Delete line from range x to y.(x,y is any Number)
```

OUTPUT :

```
sed '/kali/d' cmd.txt # To Delete pattern matching line.
```

OUTPUT :

Specified Line :

```
sed -n '2p' cmd.txt # To see any line content.(2(Any_Number)p)
```

OUTPUT :

---

10. grep - Command

```
grep   # The grep filter searches a file for a particular pattern of characters.

SYNTAX : grep Flag Pattern File_Name(cmd.txt)
```

OUTPUT :

Flags :

```
-c  # This prints only a count of the lines that match a pattern.
```

OUTPUT :

```
-i  # Ignores, case for matching.
```

OUTPUT :

```
-n  # Display the matched lines and their line numbers.
```

OUTPUT :

```
-v  # This prints out all the lines that do not matches the pattern.
```

OUTPUT :

---

11. pipe - Command

```
|   # Pipe is a command in Linux that lets you use two or more commands such
that output of one command serves as input to the         next.

SYNTAX : Input_Command | Output_Command
```

```
ls -l | more
```

OUTPUT :

```
cat cmd.txt | head -2
```

OUTPUT :

QUERIES :

1. Total no of users connected to system currently .

```
BY Using 'last' Command.
```

OUTPUT :

2. Display only current local time of system.

```
BY Using 'date +%H:%M:%S' Command.
```

OUTPUT :

3. Arranging the files of directory on ascending order of their sizes.

```
By Using 'ls -lsh' Command. Here s for sorting , h for human readabilty & l full
details.
```

OUTPUT :

4. Display 5 th character of your name.

```
By Using 'cut -c 5 Info.txt' Command.
```

OUTPUT :

5. Display 7 th line of a file.

```
By Using 'sed -n '7p' cmd.txt' Command.
```

OUTPUT :

6. Sort a file on second column basis.

```
By Using 'sort -k 2n employee.txt' Command. Here k for Column sorting & 2n for
Column Number.
```

OUTPUT :