# ASSIGNMENT NUMBER: 1(a)

TITLE: Execution of basic & advanced Unix commands OBJECTIVE:

Study of some Basic & Advance Unix commands on Linux/ Unix Operating System.

# THEORY:

To execute a command, type its name, options and arguments at the shell prompt.

ls -l /etc

Command name Options (flags) Arguments

General Purpose Commands

1. date: To display current date & time of the system.
2. cal :To display calendar of current month.
3. who:List who is currently logged on to the system.
4. Whoami:Report what user you are logged on as.
5. echo :Echo a string (or list of arguments) to the terminal
6. bc:To perform mathematical operations
7. clear:To clear the screen
8. alias : Used to tailor commands Ex alias erase=rm

alias grep=”grep -i” alias cp=“cp –i”

1. man <cmd name>: To get help for any command
2. passwd: To change the passward
3. exit: To logout from the terminal

File & Directory Related Commands

1. cp <fromfile> <tofile>: Copy from the <fromfile> to the <tofile>
2. mv *<fromfile> <tofile>* : Move/rename the <fromfile> to the <tofile>
3. rm *<file>*:Remove the file named <file>
4. mkdir *<newdir>*:Make a new directory called <newdir>
5. rmdir *<dir>*:Remove an (empty) directory
6. cd *<dir>* :Change the current working directory to *dir*
7. pwd : Print (display) the working directory
8. cat > *<file>* :*T*o create new file n save it by pressing ^d
9. cat >> <file>: To append contents into file
10. *cat <file>*:To see the contents of existing file
11. more *<file>*:*Paging out the contents of file*
12. file *<file>*:*To check the type of file*
13. wc *<file>*:To count lines,words,charaters of file
14. cmp *<file1> <file2>*:To compate two files
15. comm *<file1> <file2>*:To display common values between two files
16. diff *<file1> <file2>*:To convert one file to another
17. *gzip <file>*:*To compress the file*
18. gunzip *<file>*:To unzip the contents of
19. ls :List the files in the current working directory
20. ls <dir>:List all files & directories in given directory
21. ln <fromfile><tofile>: Creates a symbolic link to a file

Simple Filters

1. pr <file> :Paginating the file Ex pr –h “test” –d –n fname
2. head <file>:Display first 10 lines of file Ex head –n -3 fname
3. tail <file> :To display last 10 lines of file Ex tail -3 fname ; tail –c 100 fname
4. cut <file> :Splitting file vertically Ex cut –c 2-10,12-14 fname
   1. cut –d “|” –f 2,4 fname
5. paste <file1> <file2> :To combine two file vertically rather than horizontally

Ex paste –d “|” fname1 fname2

1. sort <file>:To sort file in order by field wise Ex sort –t”|” –k 2 fname
   1. sort –r fname
2. uniq <file> :Locate repeated & nonrepeated lines Ex uniq fname; uniq –d fname
3. tr ch1 ch2 < <file1>:To translate occurrence of ch1 by ch2 Ex tr ‘|’ ‘+’ < fname1
4. tee: read from standard input and write to standard output and files Ex. ls \*.txt | wc -l | tee count.txt

File permission: Use the chmod command to change file permissions

1. Changing permission relative manner

|  |  |  |
| --- | --- | --- |
| Category | Operation | Perm. |
| u-user | + assign | r-read |
| g-group | - removal | w-write |
| o-other | = assign abs perm. | x-execute |
| a-all |  |  |

Syntax: chmod category operation perm. <file>

Ex chmod u+x fname chmod a+x fname chmod u-x fname chmod a-x,go+r fname

1. Changing permission absolute manner Read=4

Write =2 Execute=1

Ex chmod 666 fname chmod 644 fname

chmod -R 644

Change owner & group

Syntax: chown options owner files Ex chown “xyz” fname

Syntax: chgrp options group files Ex chgrp “xyz” fname

Redirection: Provide a powerful command line controls

Most Linux commands read input, such as a file or another attribute for the command, and write output. By default, input is being given with the keyboard, and output is displayed on your screen. Your keyboard is your *standard input* (stdin) device, and the screen or a particular terminal window is the *standard output* (stdout) device

There are 3 types of redirection available in linux

* 1. Standard input redirection: It is used to redirect standard input. Ex. cat < fname
  2. Standard output redirection : It is used to redirect standard output. Ex cat >fname
  3. Standard error redirection: It is used to redirect standard error. Ex cat fname 2>Errorfile

Pipe

Connects commands so the output of one becomes input for the second Vertical bar(|) is the pipe operator.

Ex. ls -l | more

cat file1 file2 | sort > file3 Concatenates file1 and file2

Sends the result to the sort command

Store the alphabetized, concatenate result as a new file called file3

Grep: Global regular expression print Searching and pattern matching tools

Searches files for one or more pattern arguments. It does plain string, basic regular expression, and extended regular expression searching

Following are some of the options for grep

-i ignore case for matching

-v doesn’t display lines matching expression

-n display line numbers along of occurrences

-c counting number of occurrences

-l display list of file names

-e exp for matching

-f file take patterns from file

-E treat pattern as an extended reg. exp

-F matches multiple fixed strings (fgrep)

# Problems to be solved in the lab:

1. Change your password to a password you would like to use for the remainder of the semester.
2. Display the system’s date.
3. Count the number of lines in the /etc/passwd file.
4. Find out who else is on the system.
5. Direct the output of the man pages for the date command to a file named *mydate*.
6. Create a subdirectory called *mydir*.
7. Move the file *mydate* into the new subdirectory.
8. Go to the subdirectory *mydir* and copy the file *mydate* to a new file called *ourdate*
9. List the contents of *mydir*.
10. Do a long listing on the file *ourdate* and note the permissions.
11. Display the name of the current directory starting from the root.
12. Move the files in the directory *mydir* back to your home directory.
13. Display the first 5 lines of *mydate*.
14. Display the last 8 lines of *mydate*.
15. Remove the directory *mydir*.
16. Redirect the output of the long listing of files to a file named *list*.
17. Select any 5 capitals of states in India and enter them in a file named *capitals1*. Choose 5 more capitals and enter them in a file named *capitals2*. Choose 5 more capitals and enter them in a file named *capitals3*. Concatenate all 3 files and redirect the output to a file named *capitals*.
18. Concatenate the file *capitals2* at the end of file *capitals*.
19. Give read and write permissions to all users for the file *capitals*.
20. Give read permissions only to the owner of the file *capitals*. Open the file, make some changes and try to save it. What happens ?
21. Create an alias to concatenate the 3 files *capitals1*, *capitals2*, *capitals3* and redirect the output to a file named *capitals*. Activate the alias and make it run.
22. Find out the number of times the string “the” appears in the file *mydate*.
23. Find out the line numbers on which the string “date” exists in *mydate*.
24. Print all lines of *mydate* except those that have the letter “i” in them.
25. List the words of 4 letters from the file *mydate*.
26. List 5 states in north east India in a file *mystates*. List their corresponding capitals in a file *mycapitals*. Use the *paste* command to join the 2 files.
27. Use the *cut* command to print the 1st and 3rd columns of the /etc/passwd file for all students in this class.
28. Count the number of people logged in and also trap the users in a file using the *tee*

command.

1. Convert the contents of *mystates* into uppercase.
2. Create any two files & display the common values between them.

# OUTPUT:

Observe and note down result displayed /generated after execution of above commands.

# APPLICATIONS:

* 1. To enable the user to communicate with the kernel through the command interpreter.
  2. Useful in Shell Programming.

# FAQs:

* + 1. What is command Interpreter?
    2. How will you find current working directory?
    3. What is hierarchical file structure?
    4. What are various functions of OS?
    5. How will you sort the file which contains numeric data?