LINUX COMMANDS

BASIC COMMANDS :

1. touch - to create file

2. mkdir - to create folder

3. cd .. – go back one step

4. cd - change directory

5. pwd - present working Directory

6. whoami - to know user name

7. rmdir - remove directory

8. rm - to remove file

9. ls :- list the files or directory present in the working directory

10. ls -a :- List the hiden files along with all the files

11. ls -l :- Long List

12. nano/vi :- Are editors to modify or edit the file like notepad/editplus

vi :- (file should be give) -->{:q! – Quit vi and do not save changesvi}

synatx :- to open { vi <file-name> }

to work { change a mode to insert by click on i}

to close the editor { esc,Shift+:wq}

13. Shift+zz – Save the file and quit

:w – Save the file but keep it open

:q! – Quit vi and do not save changes

:wq – Save the file and quit

14 . Working with Nano Text Editor

1. To create and open a new file - nano new_filename

2. To save a file - press Ctrl+o (can rewrite file name)

15. cat :- I want to check or display the content present inside a file

synatx :- cat <file-name>

cat >> file :- I want to add some content into a file

syntax :- cat >> file-name (append)

to exit type :- ctrl + c

I dont want a old data but i want to add a new data inside a file

override the contnet

synatx :- cat > file-name

> :- override the old data and a new data .

>> :- Append to add a old data along with a newdata.

16. cp :- It is used to copy the data from one file to another file or one directroy to another directory.

syntax :- cp <source> <destination>

17. move :- it is used to move the data

synatx :- mv <source> <destination>

18. Remove :- it is used to delete the data

files :- rm <file-name>

rm -f <file-name> (to delete forcefully)

Directory :-

Empty :- rmdir <directory-name>

non-empty :- rm -rf <directory-name> -----rf - Recurisivly and forcefully

19. cal :- to print a calender

synatx :- cal

cal <mon> <year>

cal <year>

20.date :- print the date with time,day,year syntax :- date (tue jan 10 02:10:08 utc 2023)

SYSTEM INFORMATION :

21. uname :- to get a information about my system and operting system, machine,release,version.

22. uname -o :- to get to know about os

23. uname -r :- to know about kernal release

24. uname -v :- To know about version of kernal

25. uname -m :- to know about a machine

26. uname -a :- To get all the information

USER MANAGEMENT :-

User management will play a important role in operating system

IAM :- identity access management

like me and my brother

user :- user1

user :- user2

27. sudo su :-

sudo is a super user –Root

28. useradd :- To create a user

synatx :- useradd <user-name>

29.How to secure the user

synatx :- passwd <user-name>

30.To delete a user

synatx :- userdel -r <user-name>

31. exit – to exit from the user

32.To check whether user is created or not

vi /etc/passwd

cat /etc/passwd

nano /etc/passwd

NOTE :

Root :- switching to user account it will not ask a password Why because

Root user is a person one who created a user and he is a super user .

33.For securing the password we need to Encrypt

syntax :- openssl passwd <user-name>

34.chage :- To display a password related information

synatx :- chage -l <user-name>

GROUP MANAGEMENT :

For maiantiang a users iam going to create a supparate group for users

35.To create a group

syntax :- groupadd <group-name>

36.To check or group file

syntax :- cat /etc/group

37.To add a user into group

syntax :- usermod -a -G <group> <user-name>

38.To Rename the group

syntax :- groupmod -n <new-name> <old-name>

39. Remove a user from group

syntax :- gpasswd -d <user-name> <group>

40. View Group List for a Specific User Using groups

syntax :- groups <username>

41. to delete group

syntax :- groupdel <groupname>

42. List all members in a group

syntax :- grep <groupname> /etc/group

FILE MANAGEMENT:

43. / :- root Directory

44. /bin : binary or executable programs
(nice place for keeping persistent scripts)

45. /etc : system configuration files (an awesome place to obtain credentials)

46. /home : home directory (the default current directory when you open up the terminal)

47. /opt : optional or third-party software

48. /tmp : temporary space, usually cleared on reboot (a great place to store enumeration scripts)

49. /usr : User related programs

50. /var : log files (the perfect place to frustrate a forensic analyst)

FILE PERMISSIONS :-

owner permissions

group permissions

other permissionsFile/directory access modes

Read permissions

Write permissions

Execute permissionsPermissions mode

Numeric

4:- read
2:- write
1:- execute

Example:

51. chmod 473 <file/dir>

For owner – read , group – read write execute , other – write execute

---

Alphbets :

\+ :- to add the permssions
\- :- To remove the permissions
= :- To set a desginated permissions

r :-read
w :- write
x :- execute

---

52. chmod u=rx,g+x,o+r <file-name>

FILE COMPRESSION :

It is mainly used to compress the file by this the size of the file will get
Reduce and quality of the file will be remain same...Why we need to compress the file
size-> To save the storage space...--> Easily we can share a file

i)tar
ii)zip
iii)gzip

53. tar(tape archive):-create :- tar cvf <file.tar> <file1 file2 file3>

c :- create v :- verbosely f :- file

Example :- tar cvf jspider.tar f1.txt f2.txt f3.txt

54 . Extract :- tar xvf <file.tar>

x :- Extract
v :- verbosly
f :- file

example :- tar xvf jspider.tar

55. to remove files based on its extension

rm *.txt :- to remove similer kind of files
rm * :- to remove full files

---

zip formate :- package and compress (archive)

56. filescreate :- zip <file.zip> <file1 file2 file3>

example :- zip jspider.zip a1.txt a2.txt a3.txt

57. Extract :- unzip <file-name>

Example :- unzip jspider.zip

---

58. gzip - For compressing a individul file...

syntax :- gzip <file>

Example :- gzip devops.txt

59. Extract - For extract individul file...

syntax :- gzip -d <file.gz>

Example :-

gzip devops.txt --> devops.txt.gz

gzip -d devops.txt.gz --> devops.txt

NETWORKING COMMANDS :-

60. ifconfig :- It is used to display the ip related information.

61. ping :- send ICMP ECHO_REQUEST to network hosts

62. netstat :- It is used to check the network conncetions,routing table information
interface statics,it will be help for debugging a the servers which
all are running which port..

63. ss(socket statitics) :- It is similar to netstat but addtionally give more info about tcp.

64. host :- It is used to get a information realted to DNS servers

65. nslookup :- DNS lookup records.

66. dig(domain infomation groper) :- DNS lookup

67. last :- to display the recent login users information

INPUT OUTPUT AND ERROR REDIRECTION :

68. standard input :- It is used to collect the input from the other files

stdin :- <

syntax :-

command < fileexample :-

cat < file1

69. to change alphabets inside given file

a. syntax :- cat < <file name>
b. syntax :- tr <'olderchar'> <'newchar'> < <file name>

example :- cat < demo.txt
tr 'la' 'LA' < demo.txt
tr - translate or delete characters3.Standard

70. Output Redirection :- > or >>

syntax :- stdout

example :- echo "hello world" >> file-name

cat file1 >> file2

71. to change data from one file to another file

syntax :- cat <sourcefile> > <destinationfile>

cat file1 > file2
cat file1 >> file2

example :- cat tester.txt > demo.txt

DISK UTILITY :

Disk utility commands are used to easliy monitor the system or storage related information, and to manage disk partitions ....

72. fdisk(fixed-disk)

syntax :- fdisk <diskpath> (

double tap tab butten)it will show :- Welcome to fdisk (util-linux 2.30.2).

to quite :- ctrl + c

73. df :- it is used to get the infomration about disk usage..

74. df -h :- To display the disk usage in human readable lang...

parted :- it is used to get a information about a partitions and by using we can create a new partitions also ...

synatx :- parted –l

75. lsblk :- list block deviceslsblk lists information about all available or the specified block devices. The lsblk command reads the sysfs filesystem and udev db to gather information.

syntax :- lsblk

FILE MANAGEMENT MORE COMMADS :

76. to create multiple directory in given path

syntax :- mkdir -p sample/demo/old/devops

77. to create file inside dirctory in given path

syntax :- vi sample/demo/old/devops/file1.txt

78. ln - to make links between files. -s make symbolic link

syntax :- ln -s sample/demo/old/devops/file1.txt <link_name>

eg :- ln -s sample/demo/old/devops/file1.txt sak

79. to unlink between filessyntax :- unlink <link_name>

80. to read that linked filessyntax :- cat <link_name>

eg :- cat sak

for unlink :- call the unlink function to remove the specified file

81. to move that file from that link to another directory

syntax :- mv sample/demo/old/devops/file1.txt <directory_name>

eg :- mv sample/demo/old/devops/file1.txt demo/

82. to move that same file into previous setted path directory

syntax :- mv <path of that file> <path of that file to be moved>
eg :- mv demo/file1.txt sample/demo/old/devops/file1.txt

83. to show newest file in given ist
syntax :- ls -lt <-l --> use a long listing format>
<-t --> sort by modification time, newest first>

FILTER COMMANDS:

84. Head :- It is used to print the statring lines of file by default it will print 10 lines

syntax :- head <file-name>

i) To print a specific lines

synatx :- head -n <num> file-name

eg1 - head -n 2 sample.txt

head -<num> file-name

eg2 - head -3 sample.txt2)

85. Tail :- It will display the data from a last line(count the lines from last line) Default 10 lines

synatx :- tail <file-name>

i) Number of lines :- tail -n <num> file

eg1 - <tail -n 6 sample.txt>

86. sort :- To display the data in ascending or descending

syntax :- sort <file-name> :- ascending order

eg1 - <sort sample.txt>

87. sort -r <file-name> :- Desending order

eg2 - <sort -r sample.txt>head commands output i need to pass as an input to tail command To achive this task

88. Pipeline :- To combine a multiple commands and it will pass the output of 1st command as an input to 2nd command

syntax :- cmd1 | cmd2

example :- head -5 students.txt | tail -25.

89. Uniq :- By using this we can display/find the duplicate data

I want to count a number of occurance of a char
syntax :- uniq -c <file>

90. To print only duplicate data
syntax :- uniq -d <file-name>

91. to print duplicate data and count
syntax :- uniq -cd <file-name>

92. to print only unique lines
syntax :- uniq -u <file-name>

93. .find :- It is used to search a speific data.

94. It will search for file in user directories

syntax :- find -name <file-name>

95. It will search for in given directory

syntax :- find <from directory> -name <file name>
eg 1 :- find ./ -name sample.txt
96. search for empty files

syntax :- fnd <from directory> -empty

eg 1 :- find ./ -empty

97. search for the file with entered permissions

syntax :- find <from directory> -perm <num permission>

eg 1 :- find ./ -perm 664

98. grep :- group regular expression

syntax :- grep <search-word> <file>

command | grep <search-file/word>

99. sed (stream editor) :

syntax :- sed 's/old-data/new-data/' <file-name>

Change a data in entire file

sed 's/old-data/new-data/' <file-name>

100. Printing a SED data

sed 's/old-data/new-data/p' <file-name>

101. Replace the specific line :

sed '2 s/old-data/new-data/' <file-name>

102. Replace the data in nth line that data i want to print

sed '2 s/old-data/new-data/p' <file-name>

103. To delete the specific line :-

sed '2d' file

104. awk :- Text processing in linux

awk is referred as gawk(gnu awk)

Features :

1.It scans a file line by line
2.it formates the output file
3.it will compare

syntax :- awk '/{option}' file

example :- awk '/mech/ {print}' student.txt

105. To print a specified a column :

awk '{print $1,$2}' file-name

106. I want to print a serial number with data

awk '{print NR,$0}' file-name

107. tr :- Translate

command | tr old New

example :- echo hello | tr hello HELLO

cat student.txt | tr mech MECH