Training Notes – RedHat Linux

|  |  |
| --- | --- |
| **Legend** | |
| **Blue** | **Terminals Commands** |
| **Black** | **Theory** |
| **Green** | **Codes** |

A Program under execution is called a Process

**How to use vim: INSERT MODE COMMANDS**:

:q -> quit command

:w -> save/write command

:wq -> save and quit command

:q! -> forcefully quit command

! is used to forcefully perform any command.

dd -> deletes the whole line

yy -> copy a whole line

p -> paste

gg -> cursor will be placed at the top

u -> undo

shift+G -> cursor

**redirections**:

1. input redirections:

creates file if not exists

echo 'hello' > file1 > it overwrites the old content

echo 'everyone' >> file1 >> it appends the data to the file

2. output redirections: the command output will be stored in a file.

cal -3 > test2 > overwrites the old content

cal -3 >> test2 >> appends the data to the file

3. error redirections: redirects the error generated by a command to a file

2> overwrite 2>> append

ls test 2> file1 [file 'test' doesn't exist, so the error message will be written in file 'file1']

ls test 2>> file1 [appends the error message to the file]

4. Error and output redirection

&> &>>

5. output in one file and error in another file:

grep devops jenkins file1.txt **>** ./output.txt **2>** ./error.txt

**Command separators** -> denoted by ; (semicolon), – (double hyphen), && (double &)

**grep command:**

used to find a given keyword or a file in a file system.

Syntax -> grep string filename

Eg:

grep root /etc/passwd

grep -c root /etc/passwd = -c is used to count the keyword frequency

grep -i root /etc/passwd = -i is used to represent ‘ignore case’ which eliminates the case sensitivity

grep -r root /etc/ = -r is recursively. In this example, it will find the whole etc directory recursively to find the keyword ‘root’.

**lsattr and chattr command:**

**lsattr :** know the attributes of a certain file

**chattr: a**dd attributes to a certain file

**for example :-**

[root@devops ~]# lsattr file1.txt

---------------------- file1.txt

[root@devops ~]# man chattr

[root@devops ~]# chattr +a ./file1.txt

[root@devops ~]# lsattr file1.txt

-----a---------------- file1.txt

a – can only append by redirection

i – cannot be deleted by the root user

**Command separators** -> denoted by

; (semicolon), -- (double hyphen), && (double &)

**head command -** used to display given number of lines of content to the screen

**tail command -** used to display given number of last lines of content to the screen

**Pipeline (denoted by” | “)** - used to fetch the mid content.

E.g. total lines = 40

Head 20 | tail -10 => then 11 to 20 will be fetched out.

head /etc/passwd | tail -2

**link commands**

**soft link**: *ln -s /root/test1 Videos/test1*

the main difference between soft link and hard link is that if we delete one file, then we won't be able to fetch the data from the other file, but in hard link, we can fetch the data even if a file gets deleted.

**hard link**: *In /root/test2 Videos/test1*

we don't have to specify the type of linking method because by default, hard link is created

check **inode** of a file: ls -i test2.

**linking** is the method to link the inode number of multiple files

**Users**

3 Type of Users

|  |  |
| --- | --- |
| Admin / Root | UID – 0 |
| Regular User | UID – 1000 to 60000 |
| System or Application User | UID – 1 to 999 |

Commands –

useradd <name> - add new user to the system

id <name> - check ids of the user

chage -l root

User Data –

/etc/passwd

**Permissions**

Global Permissions – Read (r -> 4), Write (w -> 2), Execute (x -> 1)

Users - user (u), group (g), other (o)

ls -l <filename>

ls -ld <directory name>

**ACL – Access Control List**

setfacl -m u:user100:rwx /tata – to add specific permission to a file

getfacl /tata – to view the access list of that file

setfacl -b /dir – to remove all permissions