**Introduction to UNIX**

**Q. What is UNIX?**

UNIX is an open source operating system. It is widely used for networking purpose. This can be used by its copyrighters or license holder.

**Q. What is Linux?**

Linux is an open source operating system which is freely available to everyone.

**Q. What is the difference between UNIX and Linux**

|  |  |
| --- | --- |
| **Linux** | **Unix** |
| 1. It is an open-source operating system which is freely available to everyone. | 1. It is an operating system which can be only used by its copyrighters. |
| 2.It has different flavors like Ubuntu, Redhat, Fedora, etc | 2. Vendors of Unix are IBM AIX, HP-UX and Sun Solaris. |
| 3. Nowadays, Linux is in great demand. Anyone can use Linux whether a home user, developer or a student. | 3. It was developed mainly for servers, workstations and mainframes. |
| 4. Linux is used everywhere from servers, PC, smart phones, tablets to mainframes and supercomputers. | 4.It is used in servers, workstations and PCs. |
| 5. Linux is freely distributed, downloaded, and distributed through magazines And priced flavors of Linux are also cheaper than Windows. | 5. Unix copyright vendors decide different costs for their respective Unix Operating systems. |

**Q. Architecture of UNIX**

**Terminology-**

1./ Root

2. /ABC- Directory

3. F1- File

F3

F4

/d6

/d5

/d4

/d3

**F1**

**/d2**

**/d1**

**/pooja**

**/**

**/….**

**/home**

**/etc**

**user**

**/DD**

**/bin**

**Flavors:**

**Linux** - Ubuntu, Redhat, Fedora etc

**UNIX** - IBM AIX, HP-Ux, Sun Solaris etc

**Basic commands**

**1. Date:**

It will display date and time.

**Syntax:** $date

**2. Cal:**

It will display current month calendar

**Syntax:** $ cal

**3. Cal year:**

It will display specific year calendar for all months

**Syntax:** $ cal year

cal 2020

**4. Cal month year:**

It will display specific year calendar for specific month

**Syntax:** $ cal month year -cal 3 2020 or cal march 2020

**5. Who**

Display current logged users in system.

**Syntax:** $who

**7. Whoami**

It will display only user name

**Syntax**: $whoami

**8. Who all**

It will display logged user and not logged user for specific project

**Syntax**: $who all

**9. Hostname**

Display system name or host name

**Syntax:** $Hostname

**10.Hostname –i**

Display IP address

**Syntax**: $Hostname –i

**11. Whatis**

It will display command description.

**Syntax**: $whatis

Ex: $whatis cat

**12. Man**

It will provide the complete manual related to particular command with syntax and options which to be used in Linux.

**Syntax:** $man cat

**13.wc**

It will display no. of lines,words,characters

**Syntax:** $wc

**Options**- wc –l. wc –w, wc -c

**14.ls**

It will display all directories and file name.

**Syntax:** $ls

**File related commands**

***Q: Tell me any 5 basic commends you’re across?***

***Q: which commands your using with your project?***

**File related commands**

**1. Touch:**

It is use to create empty .txt file

**Syntax:** $ touch filename

**2. Cat:**

This is command is used for many purposes

1. Create a file. ($cat > file)

2. To display the content of file. ($cat file or $cat < file)

3. To append (add) data at the end of file. ($cat >> file)

4. Concatenate more than one file. ($cat file file1)

5. Copying many files into a single file. ($cat file file1 > file2)

**Syntax:** $ cat

**3. More**

To show content in a file. I will show content page by page.

**Syntax:** $more

**4. Less**

To show content in a file with 'END' keyword. It is providing the flexibility to move up and down into the file by using up and down arrow. Same like more it will show page by page data.

These commands are used to open large data/content file in real time scenarios.

**Syntax: $less**

**5. CP**

It will erase or overwrite data into file.

Syntax: $cp f1 f2

Note: it will erase or overwrite data into f2 and copy f1 data into f2.

**6. mv**

Rename a file or move the file.

**Syntax:** $mv f1 f2 - moving the data from f1 to f2

$mv d1 d2 - While using the purpose of rename but condition is that the file should exist in directory then only we can rename the file.

**Note:** f1 data copies f2 i.e. f1 will be permanently deleted by copying the data into new file i.e. f2.

**7. Head**

It will display top 10 records by default from a file.

**Syntax:** $head filename, head option filename.

Ex: head p1, head -9 p1, $head -22 p1

**8. Tail**

It will display bottom 10 records by default from file.

**Syntax:** $tail filename or tail option filename

**Ex:** tail -2 p1, tail p1, tail -22 p1

**Note**

If we want to display middle lines from file.EX- line no 3 to 7

Formula : (7-3)+1=5

Command- Head -7 F1 | tail -5

If we want to display particular one line from file. Ex- No 5

head -5 F1 | tail -1

If we want to display different number of lines from file. Ex- No 5 , No7 , No9

Head -5 F1 | tail -1 &&

Head -7 F1 | tail -1 &&

Head -9 F1 | tail -1

**9. rm**

It is used to remove file from directory. We can delete more than one file at one time by using same syntax.

**Syntax:** $rm filename,

**Ex:** rm p18 p1

***Q: how to create empty or simple .txt file?***

***Q: what is the purpose of cat?***

***Q: how we can overwrite a file?***

***Q: How to transfer or how to copy first 5 data into second file without overwriting and with overwriting?***

***Q: How to display file with line number?***

***Q: Difference between cp , mv and cat commands.***

|  |  |  |
| --- | --- | --- |
| **cp** | **mv** | **cat** |
| Copy paste | Cut paste | Copy paste |
| Overwrite | Overwrite | Doesn’t overwrite |
| First file exist with data | First file doesn’t exist with data | Both files exist with data |
| cp F1 F2 | mv F1 F2 | cat F1>> F2 |
|  | I acts as rename |  |

**Filter related commands:**

**1. Grep (global regular expression print)**

--It is used to search particular pattern or regular expression

**Syntax**: $grep ‘option’ filename

Ex-Grep ‘pune’ city

grep -n ‘pune’ city (It will display the line no in which pattern is present)

grep -i ‘pune’ city (It will ignore the upper case and lower case in pattern)

grep -c ‘pune’ city (It will count how many time the pattern is present in a file)

grep -v ‘pune’ city (It will display only those lines where the pattern is not present)

grep ‘^$’ city( It will display empty line)

grep ‘pune’ city && grep ‘city’ city (It will display 2 different pattern search)

grep –r ‘pune’.\* (suppose we don’t know file name)

grep –r ‘pune’ city F2 (I multiple file name)

**2.Sort**

To display content from a file in ascending and descending order

**Syntax**-Sort number (Ascending)

**Syntax**-Sort –r number(Descending)

**3.gzip**

The gzip command is used to truncate the file size. It is a compressing tool. It replaces the original file by the compressed file having '.gz' extension.

**Syntax gzip F1**

**4.gunzip**

The gunzip command is used to decompress a file. It is a reverse operation of gzip command.

**Syntax- gunzip F1**

**Q. How to display empty line from file?**

**Q. How to search two different patterns from 1 file?**

**Q. How to search 1 pattern from two different file?**

**Q. How to search pattern when we don’t know file name?**

**Q. For what purpose sort command is used?**

**Q. When we compress in which extension file is going to save?**

**Directory Related commands**

**1. pwd**

The pwd command is used to display the location of the current working directory.

**Syntax: pwd**

**2. mkdir**

The mkdir command is used to create a new directory under any directory.

**Syntax: mkdir <directory name>**

**3. rmdir**

The rmdir command is used to delete a directory.

By using this command we can delete only empty directory.

**Syntax: rmdir <directory name>**

**4. cd**

The cd command is used to change the current directory.

**Syntax: cd <directory name>**

**5. cd ..**

To go up to one directory back

**Syntax: cd ..**

**6. cd ~**

To go to root directory

**Syntax: cd ~**

**7.ls –l**

The ls command is used to display a list of content of a directory with details.

**Syntax: ls –l**

Columns above indicate specific things:

Column 1 - indicates information regarding file permission.

Column 2 - indicates the number of links to the file.

Column 3 & 4 - indicates the owner and group information.

Column 5 - indicates size of the file in bytes.

Column 6 - shows the date and time on which the file was recently modified.

Column 7 - shows the file or directory name.

**Q: How to reverse multiple directory back?**

**Q: What is the difference between cd and cd ..?**

**Q: How we can create sub directories?**

**Q How we can delete non empty directory?**

**Q: What is the difference between rm and rmdir?**

**Permission related commands**

**1.Chmod**

This command is used to set the permission for file or directory.

Basically there are two approaches we can use with chmod command to change the permission

1. By symbolic method

2. By using CHMOD-777 table

**1.1. Symbolic method**

To change file and directory permissions, use the command chmod (change mode). The owner of a file can change the permissions for user (u), group (g), or others (o) by adding (+) or subtracting (-) the read, write, and execute permissions.

The first way is the relative (or symbolic) method, which lets you specify permissions with single letter abbreviations. A chmod command using this method consists of at least three parts from the following list.

|  |  |  |
| --- | --- | --- |
| **Access class** | **Operator** | **Access Type** |
| u (user) | + (add access) | r (read) |
| g (group) | - (remove access) | w (write) |
| o (other) |  | x (execute) |
| a (all: u, g, and o) | | |

For example, to add permission for everyone to read a file in the current directory named **myfile**, at the UNIX prompt, enter:

**EX: chmod u+r myfile**

The ‘u’ stands for "user", the ‘+’ for "add", and the r for "read".

**chmod a+x,g+w,o-r f1**

The ‘a’ stands for "all", the ‘+’ for "add", The ‘g’ stands for "group", the ‘w’ for "write", o for “others”, - for “remove”.

**1.2. CHMOD -777**

The second way to use the chmod command is the CHMOD-777 table form, in which you specify a set of three numbers that together determine all the access classes and types. Rather than being able to change only particular attributes, you must specify the entire state of the file's permissions.

The three numbers are specified in the order: user (or owner), group, and other. Each number is the sum of values that specify read, write, and execute access.

|  |  |
| --- | --- |
| **Permission** | **Number** |
| Read (r) | 4 |
| Write (w) | 2 |
| Execute (x) | 1 |

Ex- chmod 777 F1

Chmod 000 F1

Chmod- 512 F1

**2. chown**

Different users in the operating system have ownership and permission to ensure that the files are secure and put restrictions on who can modify the contents of the files. In Linux there are different users who use the system.

Ex- chown New\_user F1

-- Each *user* has some properties associated with them, such as a user ID and a home directory. We can add users into a group to make the process of managing users easier.

-- A *group* can have zero or more users. A specified user can be associated with a “default group”. It can also be a member of other groups on the system as well.

**3. chgrp**

It is used to set permission for specific group.

Ex- chgrp Grp\_name F1

***Q. How to give write permission to group (Symbolic and numeric)?***

***Q How to remove execute permission from others (Symbolic and numeric)?***

***Q. What is chmod 251?***

***Q. How we can add or remove all permissions in one command?***

***Q. How change ownership of particular file or directory?***

***Q.What is the difference between chown and chgrp?***