UNIX

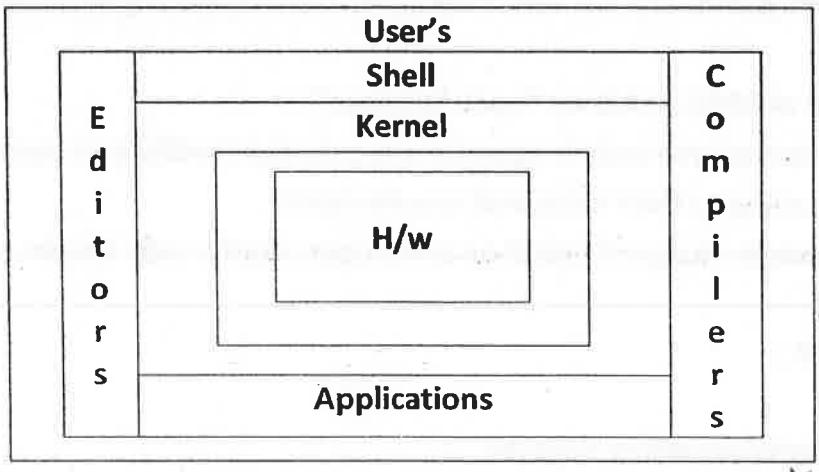
- It is an operating system.
- It is system software.
- It is a collection of system programs.
- It acts as interface between user and the hardware.
- It is classified into 2 types
    1. CUI(Character User Interface):-Ms-DOS,unix
    2. GUI(Graphical User Interface):-Windows,linux
- All CUI Operating systems are not user friendly but GUI Operating systems are user friendly.
- The main feature of UNIX is, it is an "open Sysytem".
- Open system means source code is open i.e any user can modify unix open source code according to their ideas and requirements.
- Any Operating system designed based on UNIX open source code known as "Flavour of unix".
- **Flavour's of UNIX :**
    1. Linux
    2. Sun Solaris(sun micro system)
    3. IBM-AIX(IBM)
    4. HP-UX(HP)
    5. SCO-Unix(Santa Cruz)
    6. IRIX(Silicon graphics)

The features of LINUX

- It is free software.
- It is GUI.
- It doesn't require any separate hardware setup.
- In-Built Software's like C,C++,Perl,Python,PHP,MySql,Open office, Apache web server, Mozilla firefox,.....
- Open system.

- **FLAVOURS OF LINUX**
  1. Red hat Linux(RHEL)
  2. Fedora Linux
  3. Open Suse Linux
  4. Ubuntu Linux
  5. Cent os Linux
  6. Oracle Enterprise Linux(OEL)

STRUCTURE OF LINUX



SHELL:

Interpreter converts high level instructions to machine level.

- It is a command line Interpreter(CLI).
- It takes request from user and checks command existence,if command exist then it converts into Kernal understand language(Machine Language)and it sends the given request to kernel.
- It acts as interface between user and the kernel.

KERNEL:

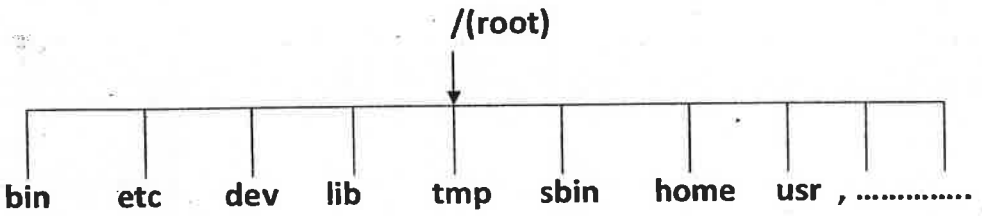
- It heart of Unix.
- It is a group of 100's of system calls.
- System call means low level programming.

- Each and Every system call written in "C" Language.
- Examples of system calls:- Open(),fork(),Proc(),copy(),Kill(),exec(),.....

Unix File System

- In UNIX, each and every thing it treats as file.
- The normal file known as Regular file.
- The Directory Known as Directory file.
- The device known as Device file.

The base of the unix file system is "/"(root) , and it is default administrator's home directory .



/bin : It maintains all user Executable commands.

/etc : It contains all systems configuration files.

/etc/passwd : It maintains each and every user information with 7 fields.

Username:passwd:uid:gid:comment:home:shell

/dev : It contains all device files information.

/temp : It contains all temporary files information.

/sbin : It contains all administrator executable commands.

/lib : It contains all library files information.

/home (or) /usr: It is default user's home directory

UnixBasicCommands

1. **$logname** : It displays current user name
2. **$pwd** : It displays present working directory path
3. **$date** : It displays system date & time
4. **$clear** : To clear the screen
5. **$cal** : It displays current month calendar
6. **$cal2000** : It displays 2000 year calendar
7. **$cal 01 2010** : It displays January month 2010 year
8. **$uname** : It displays operating system name
9. **$uname -r** : It displays kernel version
10. **$hostname** : It displays server name
11. **$hostname -i** : It displays server IP address
12. **$who** : It displays list of users who are connected to the server
    Tecno1 tty01 Aug 0112:21
    Tecno2 tty04 Aug 0112:25
13. **$who am i** : It displays current user Information
    Tecno tty01 Aug 01 12:21
14. **$tty** : It displays current terminal name
15. **$uptime** : It displays how long server is up and running , no.of users connected, and average load on the server
16. **$su** : It is used to switch from one user account to another user account
    Syntax:$su - username
    Eg: ***$su - tecno3*** ↵
17. **$whoami** : It displays switched user name (or) child user name

18. **$whichcommandname (or)$whereiscommandname** :It displays location of the given command

    Eg: $ which date ↵

19. **$exit** : To logout from current user account

20. **$man commandname** : It displays help pages of given command

Eg:$man date: *It displays manual pages of date command.*

Working with files

Cat: It is used to create new files (or) to open existing files (or) to append data to the existing file.

1. **Creating a new file:**

Syntax: $ cat > filename ---------- ---------- Ctrl d	***Eg: $cat > sample*** ↵ Tecnosoft is one of the leading institute in Software & Development Ctrl d

Note: Extension is optional for files

2. **How to open a file**

Synatx : $cat < filename (or) $cat filename	***Eg: $cat < sample*** ↵ Tecnosoft is one of the leading institutes in Software & Development sector.

3. **Appending data to the file**

$cat >> filename Ctrl d	***Eg: $cat >> sample*** ↵ *Tecnosoft is leading software training institute since a decade* Ctrl d

4. **How to open multiple files**

Syntax :$cat file1 file2 ..... filen	***Eg: $cat emp1 emp2 emp3*** It displays first emp1 file contents followed by emp2 file followed by emp3 i.e It joins multiple files vertically without space.

5. **touch:** It is used to create empty files i.e. zero byte files.Moreover, it is used to change file time stamp.

Syntax:$touch filename ↵ ***Eg: $touch a1*** ↵

6. **Creating multiple files**

$touch file1 file2.... filen

Eg: $touch x1 x2 x3 x4↵ It creates x1, x2, x3 and x4 as new empty files

7. Deleting files

a. rm : To delete a file

Syntax : $rmfilename ↵ ***Eg: $rmsample*** ↵

b. $rm -i filename ↵ It deletes a file with conformation

Eg: $rm -i sample ↵

Removesample?y (yes)
n(no)

c. $rm -f filename↵ It deletes a file forcibly

Eg: $rm -f sample

d. Deleting multiple files

$rmfile1 file2.....filen↵

Eg: $rm a1 a2 a3 ↵

Working with directories

1. **mkdir:** To create a new directory

   $mkdirdirname↵

   Eg:$mkdirabc ↵

2. **cd:** To change a directory

   $cd dirname↵

   Eg: $cdabc ↵

.→ Current Directory
.. → Parent Directory
~ → User's Home Directory

3.$cd..↵ : To come out from current working directory

4.$cd / : It changes to root directory

5.Creatingmultiple directories

$mkdirdir1dir2 .....dirn↵

Eg:$mkdir d1 d2 d3d4↵ It creates d1, d2, d3 and d4 as new directories

6. Creating directories under sub directories

$pwd↵
/home/tecno

$mkdirabc xyzabcabc/a1abc/a2abc/a1/a11 abc/a1/a12 xyz/x1 xyz/x2 ↵

or

$mkdir-pabc/a1/a11abc/a1/a12 xyz/x1 xyz/x2 ↵

7.Removing directories

$rmdir :To remove a directory but directory must be empty.

Syntax : $rmdirdirname↵ ***Eg: $rmdirabc***

a. **$rm -r directoryname**: It deletes recursively entire directory structure

Eg: $rm–rabc ↵

b. **$rm-ridirectory name**: It deletes recursively entire directory structurewith confirmation.

Eg: $rm–riabc ↵

c. **$rm–rfdirectoryname:** It deletes recursively entire directory structure with forcibly

Eg: $rm–rfabc ↵

Copying files:

1. **$cp:** To copy afile

Syntax : $cp source file target file

Source file must be existing file and target file may be a new file or may be a existing file.

Note : Any Path starts from ("/")root directory known **as Absolute path.**

If path not started with("/")root directory known **as Relative Path.**

2. Copying files from one directory to another directory

$pwd↵

/ home / tecno

Eg: Copy a11 directory emp file into x1 directory

1. $cpabc/a1/a11/ emp xyz/ x1/↵

   $cd abc/a1/a11 ↵

   $pwd↵

   / home/tecno/abc/a1/a11

2. $cpemp/home/ tecno/ xyz /x1/↵ **(or)**

   $cpemp../../../xyz/ x1/↵

3. Copying multiples into one directory

$cpfile1file2 .....filendirectory

Eg: **$cpempa1 sampleabc↵**

It copies emp, a1 and sample files into abc directory

4. Copying a directory

$cp -Rsourcedirtargetdir

Eg: **$cp-Rabc xyz** ↵ The abc directory copied into xyz directory

5. mv: To rename a file /directory (or) To move a file / directory

Syntax : $mv oldfile/olddirnewfile/newdir

Eg: **$mv file1 file2** ↵ file1 is renamed by file2

Eg: **$m dir1 dir2** ↵ dir1 is renamed by dir2

6. **Creating hidden files:** To hide a file/dir, start filename or directory name with "." Character

Syntax :$cat >.filename ↵ Ctrl d	**Eg:** $cat >.sample ↵ Ctrl d

2.$mv emp .emp↵To hidea existing file

3.$mv .emp emp↵ To unhide a file

4.$mkdir .dirname ↵ **Eg:** $mkdir .abc↵ To hide a directory

5.$mv .abc abc↵ To unhide a directory.

Viewing list of files

1.$ls ↵ It list current directory all files and sub directories in the ascending order based onASCIIvalues

2.$ls –a↵ Itlist all files along with hidden files

3. $ls -r ↵ It list all files in reverse order i.e. descending order

4. $ls -R ↵ It list all files Recursively

5. $ls -t ↵ It list all files based on date & time of creation

6. $ls -l ↵ Itlistallfiles in long list format i.e. with 9 fields

File type	File permissions	No of links	Owner name	Group Name	Size in Bytes	Date	Time	Filename
-	rw-r--r--	1	Tecno	Tecno	1521	Aug 01	12:21	Sample
d	rwxr-xr-x	2	Tecno	Tecno	4096	Aug 01	12:30	abc

Note :We can combine same unix command options in any order.i.e
$ls -at , $ls -rt , $ls -Ra , $ls -lrt,.....

Wild Card Characters

1. * : It matches O or more characters
2. ? : It matches any single character
3. [ ] : It matches any single character in the given list
4. [-] :It matches any single character in the given range

Examples

$ls d*	$ls a???	$rm *	$cp *.c ~/abc
$ls *t	$ls [vsd]*	$rm -i *	$cpabc/a1/* .
$ls b*k	$ls [abcdef]*	$rm t*	$cp xyz/[bkt]* abc
$ls a?	Or $ls [a-f]*	$rm [bkt]*	$rm -rfabc/*

1.wc: It counts total no.of lines, words and characters in a given file.

$wc filename ↵

Eg: $wc sample ↵ It counts no of lines,word and characters in a sample file

1.$wc -l filename ↵ It counts no.of lines
2. $wc -w filename ↵ It counts no.of words
3.$wc -c filename ↵ It counts no.of characters
4.$wc -lw filename ↵ It counts no.of lines & words
5.$wc -wc filename ↵ It counts no.of words & characters
6.$wc -lc filename ↵ It counts no.of lines & characters

2.cmp: It compares 2 files character by character, if no output, then files are same otherwise, the files are not same

```
1. $cat a1
Hello
2. $cat a2
Hello
3. $cat a3
Helo
```

$cmp file1 file2 ↵

Eg: 1. $cmp a1 a2 ↵
2. $cmp a2 a3 ↵
Differ a2 a3 : byte 4, line1

3.diff : It displays different lines between 2 files.

$diff file1 file2 ↵

4.file: It displays the given file type.

Syntax :$file filename ↵

Eg : 1. $file sample ↵
Sample: Ascii text file
2. $file abc↵
abc : directory
3. $file x1 ↵
x1 : empty

Redirecting Input/Output/Errors

stdin (0), stdout (1) , stderr(2)

< : Redirect Input

> (or) 1> : Redirect output

2> : Redirect errors

Eg: 1. $cat emp>file1 ↵ *It redirects emp file data into file1 file.*

$cat file1

2. $cat emp>>file1 ↵ *It appends emp file data into file1 file.*

3.$cta sample 2 > file2 ↵ *It redirects errors to file2 file*

$cat file2

4.$cat file1 file2 file3>out_file 2 >err_file ↵ *It redirects output intoout_file file and errors into err_file file.*

Note: **&1 represents output filename and &2 represents error filename**

Eg1: $cat file1 file2 file3 > out_file 2 >&1 ↵ *It redirects output and errors into same out_file file*

Eg2: $cat file1 file2 file3 2>err_file >&2 ↵ *It redirects output and errors into same err_file file*

Creating Aliasname :

alias :alias is the command to create alias name for existing commands.aliases are temporary.

Syntax : $alias <aliasname>='unix command'

Eg: $alias c='clear'

$alias d='date'

How to see aliases

$alias ↵ It displays all alias names

How to remove alias names

$unalias :To remove alias name.

Syntax : $unalias<aliasname>

Eg : $unalias c

$basename : It extracts only filename from the given path.

Eg :$basename /home/tecno/abc/a1/a11/sample.log
Sample.log

$dirname : It extracts only dirctoryname from the given path.

Eg : $basename /home/tecno/abc/a1/a11/sample.log
/home/tecno/abc/a1/a11

$history :It displays previously executed commands list.

$history n : It displays last 'n' executed commands list

$history ↵ , **$history -10** ↵

$ history 10

Filter Commands

Flat file :

* In a file data entered by using delimiter known as flat file.
* The default delimiter is **tab key.**
* **Delimiter** means **field separator.**
* **Enter key** means **record separator.**

→ Create a student flat file with student id,name,phno,course and loc with default delimiter(tab key)(delimiter flat file)

```
$cat > stud↵
1       2          3              4       5
101     Hari       9966422225     unix    Hyd↵
102     sai        66839666       oracle  Sec ↵
103     devaj      9963979390     dba     hyd↵
104     venkat     66619666       perl    sec↵
105     siva       9966322224     unix    hyd↵
106     lakshmi    66839666       oracle  sec↵
107     padma      9966422225     dwh     hyd↵
Ctrl d
```

→ Create emp flat file with empno,ename,sal and deptno with , delimiter(custom delimiter flat file)

```
$cat >emp↵
  1      2      3    4
101, lakshmi,80000,10 ↵
102, padma,75000,20↵
103,devaj,70000,30↵
104,venkat,65000,10↵
105,tejas,70000,20↵
Ctrl+d
```

1. cut : It is used for to extract specific fields and characters from a given file.

1. **$cut -f 2,4 stud ↵** It displays 2nd,4th fields from stud file
2. **$cut -f 2-5 stud ↵** It displays 2nd field to 5th field from stud file.
3. **$cut -f 3- stud ↵** It displays 3rd field to last field from stud file.
4. **$cut -d "," -f 2,4emp↵** It displays 2nd&4th fields from emp file.
5. **$cut -c 1-10 stud ↵** It displays every line 1st character to 10th character from stud file
6. **$cut -c 5-10,15-20 stud ↵** It displays every line 5th character to 10th character and 15th character to 20th character from stud file.

2. paste : It is used for to join two or more files horizontally by using delimiter(the default delimiter is tab).

Syntax : $paste file1 file2 .....

Examples :

$cat>states	$cat>cities	$paste states cities ↵	$paste -d":" states cities ↵
AP	Hyd	AP Hyd	AP:Hyd
TN	Chennai	TN Cehnnai	TN:Cehnnai
KN	Bangalore	KN Bangalore	KN:Bangalore
MS	Mumbai	MS Mumbai	MS:Mumbai
Ctrl d	Lucknow	Lucknow	:Lucknow
	Ctrl d		

3. tr: It translates character by character.

Examples

```
$cat>sample
I am learning unix
Ctrl d
```

1. **$tr "aeiou" "AEIOU" < sample ↵** — It replaces lower case vowel characters by upper case vowel characters in a sample file
   O/p : I Am lEArnIngUnIx
2. **$tr "[a-z]" "[A-Z]" < sample ↵** — converts sample file contents into upper case letters.
   O/p : I AM LEARNING UNIX
3. **$tr "[a-zA-Z]" "[A-Za-z]" < sample ↵** — converts sample file contents into reverse case letters.
   O/p : I AM LEARNING UNIX
4. **$tr "," "\t" <emp↵** — It replaces comma(,) delimiter with tab delimiter in emp file
5. **$tr "aeiou" "9"< sample** — It replaces all lower case vowels with '9' letter.
6. **$tr -d "a" < sample ↵** — It deletes 'a' character from sample file
   O/p : I mlerningunix
7. **$tr -d "aeiou" < sample ↵** — It deletes lower case "aeiou" vowel characters from sample file.
   O/p : m lrnngnx
8. **$tr-s " " <sample ↵** — It squeezes space character from sample file

4. sort : It sorts the given file contents in ascending order based on ASCII values

Syntax:

1. **$sort filename↵ It sorts file contents in ascending order**
2. **$Sort -r filename↵ It sorts file contents in descending order**
3. **$sort -u filename ↵It displays unique lines**
4. **$sort -n filename↵ It sorts file contents based on numeric comparision**

ASCII Values
0-9 -> 48-57
A-Z -> 65-90
a-z -> 97-122

Examples1)

$cat>sample↵	$sort sample↵
te---------	3---------
he----------	a----------
3---------	de--------
de--------	ha-------
ha---------	he--------
a---------	te--------
ctrl d	

Any filter command will not alter data in original file. we need to do it manually.

2) $sort -r sample ↵It sorts in descending order.

3) $sort -u sample ↵It displays only unique lines i.e, it eliminates duplicate lines.

4)

Eg: $cat>file1 ↵	$sort file1 ↵	$sort -n file1 ↵
15	1025	15
23	125	23
125	15	125
225	225	225
456	23	456
456	456	456
1025	456	1025
Ctrl d		

How to sort filed by filed

1) sort file on the basis of 2nd field

$sort -k 2,2 stud

103	devaj	9963979390	dba	hyd
101	hari	9966422225	unix	Hyd
105	hari	9966322224	unix	hyd
106	lakshmi	66839666	oracle	sec
107	padma	9966422225	dwh	hyd
102	sai	66839666	oracle	Sec
104	venkat	66619666	perl	sec

2)sort file on the basis of 3rd field

$sort -k 3n,3 stud

104	venkat	66619666	perl	sec
102	sai	66839666	oracle	Sec
106	lakshmi	66839666	oracle	sec
103	devaj	9963979390	dba	hyd
105	siva	9966322224	unix	hyd
101	hari	9966422225	unix	Hyd
107	padma	9966422225	dwh	hyd

3)sort the file alphabetically on the 2nd field, numerically on the 3rd filed

$sort -k2,2 -k3n,3 stud

103	devaj	9963979390	dba	hyd
105	hari	9966322224	unix	hyd
101	hari	9966422225	unix	Hyd
106	lakshmi	66839666	oracle	sec
107	padma	9966422225	dwh	hyd
102	sai	66839666	oracle	Sec
104	venkat	66619666	perl	sec

5. uniq : It displays uniq lines in the given file but the file contents must be in sorted order.

Syntax: $uniq filename↵

1. **$uniq -u sample ↵**It displays non duplicates lines.
2. **$uniq -d sample↵**It displays only duplicated lines.
3. **$uniq -c sample ↵** It counts how many times the lines are repeated in the file

Examples

$cat>sample↵	**$uniq sample↵**	**uniq -u sample↵**	**$uniq -d sample↵**	**$uniq -c sample↵**
aaa	aaa	ccc	**aaa**	2 aaa
aaa	ccc	ddd	**hhh**	1 ccc
ccc	ddd	ppp		1 ddd
ddd	hhh			3 hhh
hhh	ppp			1 ppp
hhh				
hhh				
ppp				
Ctrl d				

6. grep[Globally search a regular expression and print it]

Definition: It is used for to search a string and regular expressions in a given file or files.

Syntax:grep string/pattern filename(s)

Examples

$catsample

---------**tecno**----------
---
tecnosoft----------

---------**tecno soft solutions**

1. **$grep tecno sample** :- It prints the line containing tecno
2. **$grep hello sample** :- It prints the line containing hello
3. **$grep tecno file1 file2 file3** :-It searchstecno string in file1,file2 & file3 files and prints lines having tecno
   file1: -------
   file2: -------
   file3: -------
4. **$grep tecno*** :-It searches the tecno string in current directory all files.
   file1: -----
   file3: ------
   file3: -------
   sample:-----
   sample: -----
   sample:--------
5. **$grep "tecno soft" sample** :- It prints the line containing tecno soft

Grepcommand options

1) $grep –i "Tecno" sample ↵ It ignores case sensitive.

2) $grep -c "tecno" sample ↵ Itcounts number of lines having tecno

3) $grep -n "tecno" sample↵It prints lines with line numbers containing tecno

4) $grep -v "tecno soft" sample↵It displays the lines do not containing tecno soft

5) $grep -l "tecno" sample ↵ It prints only filenames containing "tecno"

6) $grep -o "tecno" sample ↵ It prints only pattern containing "tecno"

7) $grep --color "tecno" sample ↵ It highlights the pattern with color

8) $grep -An "tecno" sample ↵ It prints "tecno" line followed by after 'n' lines.

9) $grep -Bn "tecno" sample ↵ It prints "tecno" line followed by before 'n' lines.

10) $grep -Cn "tecno" sample ↵ It prints "tecno" line followed by before and after 'n' lines.

11) $grep -R "tecno" *↵ It searches recursively "tecno" and found prints those filenames along with "tecno" lines.

Regular expression: Any string contains wild card character known as regular expression or pattern.

The pattern are classified into 3 types.

1. Character pattern(default)

2. Word pattern

3. Line pattern

Character pattern examples:

1) $grep "tec *" sample↵

2) $grep "b[aeiou]ll" sample ↵

ball✓
bell✓
bill✓
boll✓
bull✓
bdll×
b3ll×

3) $grep "b. .d" sample ↵

B3$d✓
ba d✓
d35#✗

Note:. is a wild card character, it matches any single character

Word pattern characters

\<\> =>Word boundary

\< => Start of the word

\> => End of the word

1) $grep "\<tecno\>" sample ↵ (or) $grep -w "tecno" sample ↵

tecno✓
tecnosoft×
hellotecno×

2) $grep "\<tecno" sample ↵

tecno✓
tecnosoft✓
hellotecno×

3) $grep "tecno\>" sample ↵

tecno✓

tecnosoft×

hellotecno✓

4) $grep "\<[0-9][0-9][0-9][0-9]\>" sample↵

3591✓

5666✓

380792×

46×

Line pattern characters(Anchors)

^ => start of the line

$ => End of the line

1) $grep "^d" sample ↵ It displays lines starting with 'd' character

d------✓

n------×

d ------✓

t-------×

2) $grep "^the" sample ↵It displays line starting with 't,h,e' characters

there-------✓

b----------×

the---------✓

c----------×

then--------✓

3) $grep "^\<the\>" sample↵It displays lines starting with "the" word.

There---------×

b----------×

the----------✓

c----------×

then---------×

4) $grep "^[bkt]" sample ↵ It displays lines starting with b or k or t.

b----------✓

c ----------×

k----------✓

a ----------×

t----------✓

5) $grep "^[^bkt]" sample ↵It displays lines which are not starting with b or k or t.

b----------×

c ----------✓

k----------×

a ----------✓

t----------×

6) $grep "t$" sample ↵ It displays lines ending with "t'

----------t✓

----------c ×

----------t ✓

7) $grep "[0-9]$" sample ↵ It displays lines ending with digit.

----------5 ✓

----------c ×

----------3 ✓

8) $grep "^unix$" sample ↵ It displays line having only unix.

unix------unix×
unix✓
----unix---- ×
unix-----×

9) $grep "^...$" sample ↵ It displays line having exactly 3 characters.

Dev✓
hello----abc×
135 ✓
a@✓
Unix-----unix×

10) $grep "^\." Sample ↵ It displays line starting with .character.

.---------- ✓
----------n ×
.---------- ✓

Note : \.,*,\^,\$,\?,\\,\",\(,\),.....

11)$grep "\$$" Sample ↵ It displays line ending with $ character.

----------$ ✓
----------t ×
----------$ ✓

12) $grep "^$" sample ↵ It displays empty lines.

13) $grep -c "^$" sample ↵ It counts number of empty lines

14) $grep -v "^$" sample ↵ It displays non empty lines.

15) How to delete empty lines from a file

$grep -v "^$" Sample ↵temp
$mv temp sample↵

7. fgrep[faster grep]:

It is used for to search multiple strings, but it doesn't allow to search regular expressions. It searches the strings more faster than the grep

Example

$fgrep "unix↵
>oracle↵
>perl" stud↵ It displays the line containing either unix or perl or oracle from stud file

8. egrep[extended grep] :

It is combination of grep&fgrep plus some additional wild card characters.

Additional wild card characters

1) (|) :- It matches any one string in the given list

Eg : $egrep "(unix¦oracle¦perl)" stud↵ It matches either unix or oracle or perl strings

2) {m} :- It matches exact occurrence of its preceding character

Eg: ab{3}c
abbc×
abbbc✓
abbbbc×

eg: $egrep "\<[0-9]{4}\>" sample↵ It matches exact 4 digit number.

3) {m,n} :- It matches minimum 'm' occurrences and maximum 'n' occurrences of its preceding character.
Eg: ab{3,5}c
abbc✗
abbbc✓
abbbbc✓
abbbbbc✓
abbbbbbc✗
Eg: $egrep "\<[0-9]{4,7}\>" sample↵ It matches 4 or 5 or 6 or 7 digit numbers

4) {m,} :- It matches minimum 'm' occurrences of its preceding character.
Eg: ab{3,}c
abbc✗
abbbc✓
abbbbbbbbbc✓
Eg: $egrep "\<[0-9]{4,} \>" sample↵ It matches minimum 4 digit numbers
Eg : $egrep "\<[789][0-9]{9}\>" sample ↵ It searches mobile number.
Eg :$egrep -o "\<[0-9]{2}[-/][0-9]{2}[-/][0-9]{2}\>" sample ↵ It searches dd-mm-yy or dd/mm/yydate formats

9. sed [Stream Editor] :

It is used for to search and replace a string and it is multipurpose filter string

Syntax: sed "s/oldstring/newstring/g" file name
s meansSubstitution
g meansglobally & all occurrences in every line

Examples

$cat <sample
tecno----------
---------tecno------
tecno----tecno----
----------tecno

1) $sed "s/tecno/linux/g" sample ↵ all tecno's it replaces with linux word (with 'g' option)
linux----------
---------linux------
linux----linux----
----------linux

2) $sed "s/tecno/linux/ sample ↵ Every line first occurrence of "tecno" it replaces with linux.(without 'g' option)
linux----------
---------linux------
linux---tecno-----
----------linux

3)$sed "s/TECNO/linux/gi" sample↵ "i" means Ignore case sensitive.
linux----------
---------linux------
linux----linux----
----------linux

4) $sed "s/^tecno/linux/gi" sample↵
linux----------
--------tecno-----
linux---tecno----
----------tecno

5) $sed "s/^$/I like Tecnosoft/" sample ↵All empty lines are replaced by "I like Tecnosoft"

6) $sed "s/\<tecno\>//gi" sample ↵It deletes tecnoword from sample file.

7) $sed -i "s/\<tecno\>//gi" sample ↵It saves changes directly in the original file

8)$sed -i "s/\<[0-9][0-9][0-9][0-9]\>//g" ↵It deletes 4 digit numbers from sample file

8) $sed -n '3p' stud↵Itprints 3rd record from stud file

9) $sed -n '2,5p' stud↵It prints2nd record to 5threcord from stud file

10) $sed -n '$p' stud ↵ It prints last record from stud file

11) $sed -n '1p↵
>$p' stud ↵It prints 1st and last record from stud file

12) $sed '3d' stud ↵It deletes 3rd record from stud file

13) $sed '$d' stud ↵ It deletes last record from stud filei.e it deletes footer from file

14) $sed '1d' stud ↵ It deletes first record from stud filei.e it deletes header from file

15) $sed-i '1d ↵
$d' stud ↵ It deletes first and last records from stud filei.e it deletes header and footer from file

16)$sed '2,5w file1' stud ↵It copies 2nd record to 5th record from stud file to file1 file

10. head :It displays the first "n" lines from a given file. By default it displays the first 10 lines from a file.
Syntax : $head -n filename
Example : $head -30 emp : It display the first 30 lines from a empfile

11. tail :It displays the lasst "n" lines from a given file. By default it displays the last 10 lines from a file.
1) **$tail -n filename**
Example : $tail -30 emp : It display the last 30 lines from a emp file
2) **$tail +n filename** : It displays nth line to end of the file
Example : $tail +30 emp : It display 30th line to end of the file
3) **$tail -f filename** : It is used for to monitor file growth.

Piping(¦) : It is used for to combine two or more commands .
Examples
1) Count numbers of users connected to the servers.
$who ¦wc -l ↵
2) Count no of files in current direcrory.
$ls¦wc-l ↵
3) Count total number of sub directories in current directoy.
$ls -l¦grep "^d"¦wc -l

4) Display only hidden files in the current directory
$ls -a | grep "^\."

5) Display owner name,filesize and filename of all files in the current directory

$ls -l | tr -s " " | cut -d" " -f 3,5,9

6) Display today's date day

$date | cut -c 1-3

tee : It is used for to write data to the file as well as to the screen.

1) $catemp¦ tee file1↵ It copies emp file data to file1 file and also it displays emp data to the screen

2) $cat emp¦ tee file1 file2 file3 ↵ It copies emp file data to file1 , file2 and file3 files and also it displays emp data to the screen

rev

$rev : It reverses given file contents linewise

Syntax : $rev filename

Eg : $rev stud

yes

$yes "sometext" : It displays the given text infinite times,to stop press **ctrl c**

Eg : $yes "Tecnosoft"

lp (or) lpr

$lp :It is used for to take printouts.

Syntax : $lp filename

Eg : $lp sample

File permissions:

Default permissions for Regular file : rw –r--r--	**Default permissions for Directory file is :rwxr-xr-x.**
Open file → read	ls→ read
Write or modify → write	Create or delete files → write
Execute → read and execute	cd → read and execute

Chmod : It is used for to change file permissions

Syntax: Chmodwho/[+/-/=]/Permission filename/directory name.

Who		Permissions
User (or) owner →u	**+ → add permission**	**read→ r**
Group → g	**- → deny permission**	**write→w**
Other → o	**= → Assign permissions**	**execute→x**

Examples

1) $chmodg+x sample ↵It adds execute permission to group members.

$ls -l sample ↵

rw-r-xr--

2) $chmodu+x,o-r sample↵It adds execute permissions to owner and it removes read permission from others.

$ls-l sample ↵

rwxr-x---

3)$chmod g=w sample ↵It adds only write permission to group members and it deletes read & execute permission from group members

$ls –l sample ↵

rwx-w----

7 → r,w,x
6 → r,w
5 → r,x
4 → r
3 →w,x
2 → w
1 → x
0 → no permission

Numeric Code

Octal code(2nd method)
Read-4
Write-2
Execute-1

1) $chmod756 sample ↵

rwxr-xrx-

2)$chmod 642 sample ↵

rw-r---w-

3) $chmod755 sample↵

rwxr-xr-x

4)$chmod700 sample↵

rwx------

5)$chmod 777 sample↵

rwxrwxrwx

#chown :To change owner of the file.

#chgrp : To change group of the file.

Vi editor

- It is used for to create new files,to open files or to modify existing files.
- It has 3 modes.
  1. Command mode
  2. Input or Insert mode
  3. Ex command mode
- The default mode is command mode.
- Vi is command to open vi editor.

The following are the commands to shift from command mode to insert mode.

1. **A** -> It places curse at end of the current line.
2. **a**->It places cursor at right side of the cursor line.
3. **I**->It place cursor at beginning of the current line.
4. **i** -> It place cursor at left side of the cursor line.
5. **O**-> It inserts newline of the cursor.
6. **o**->It inserts new line below of the cursor.

- **"ESC"** is the key to shift from insert mode to command mode.
- **"esc shift :** " is the command to shift to Ex command mode from command mode.

Command mode commands:

k(nk) **"n" means any number**

(nh)h<- | ->l(nl)

j(nj)

1. **w(nw)** : Next word starting.
   e(ne) : Word ending.
   b(nb) : Previous Word beginning.
2. **$** : End of the current line(end key)
   ^ : Beginning of the current line(home key)
3. **H** : Beginning of the current page.
   M : Middle of the current page.
   L : End of the current page.
4. **Ctrl f** : Forward one page(pgdn)
   Ctrl b : Backward one page (pgup)
5. **x(nx)** : Deletes current character (del key).
6. **X** : Deletes previous character (backspace key)
7. **dw(ndw)** : Deletes current word.
8. **dd(ndd)** : Deletes current line
9. **d$** : Deletes current position to end of the line.

10. **d^** : Deletes current position to beginning of the line.
11. **dgg** : It deletes cursor position to beginning of the file.
12. **dG** : It deletes cursor position to end of the file.
13. . **yw(nyw)** : It copies current word.
14. **yy(nyy)** : It copies current line.
15. **y$** : It copies current position to end of the line.
16. **y^** : It copies current position to beginning of the line.
17. **P** : paste below the cursor position.
    p : paste above the cursor position.
18. **J** : It joins next line to current line.
19. **cc** : It clears a line and turns to insert mode.
20. **u** : undo

EX command mode commands (esc shift :) :

: w : Save without quit.

: wfile name : Save with given file name.

: q! : quit without save.:**wq** : Save and quit.

: n : place cursor at nth line.

: $: place cursor at last line in the file.

: set nu : Sets line numbers.

:setnonu : Removes line numbers.

: ! <unix command> : Executesunix command.

: /string/ : Top to bottom search. **n -> next occurrence**

: ?string? : Bottom to top search. **N ->previous occurrence**

Search and replace and delete

: starting line no., ending line no.s/old string/new string/gi : Search and Replace string

:1,$ s/unix/linux/gi : It replaces all "unix" words with "linux" word

: 1, $ s/^/unix/ : It adds unix at beginning of each line

: 1, $s/$/;/ : It adds ; at end of each line.

:nd : It deletes nth line

3d : It deletes 3rd line

:$d : It deleteslastline

:4, 7d : It deletes 4th line to 7th line.

COMMUNICATION COMMANDS

Ctrl+alt+F1 (terminal F7)

1. write: It is used for to write message to another user account, but he should be logged into the server.

Syntax: $write username/terminalname↵

Eg:

Tecno1	Tecno2
$write tecno2↵ Hello, I am in Tecnosoft Lab **Ctrl d**	**Message from tecno1 user** Hello , I am in Tecnosoft Lab
Message from tecno2 user Hi, I am also in Tecnosoft,but I am in the class.	**$write tecno1↵** Hi, I am also in Tecnosoft,but I am in the class. **Ctrld**
$mesg n ↵ [To Deny messages]	**$write tecno1↵** Tecno1 user disabled messages. $
$mesg y↵ [To Allow messages]	

2. wall: It is used for to send broadcast message to all users, whoever connected to the sever and mesg is y.

$wall ↵
Welcome to Tecnosoft
Ctrl d

3. mail : To send the mail

Syntax:
$mail username ↵

Ctrl d

Eg: **1. $mailtecno1 ↵** This mail is transferred to tecno1 user mail box(/var/spool/mail/tecno1)
Subject: Hello ↵
Hi, How are u ?
I am doing unix course in Tecnosoft
Ctrl d

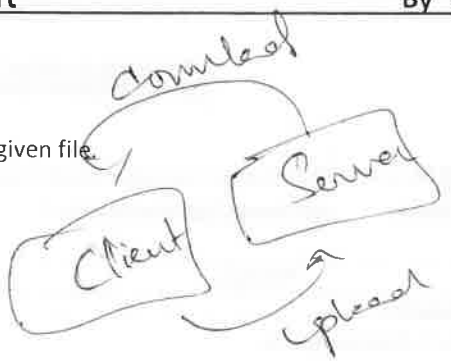
2.$mail tecno1 tecno2 tecno3 ↵ This mail is transferred to tecno1,tecno2,tecno3 user mail boxes.
Subject: From Tecnosoft↵
Tecnosoft offering unix/linux with adv.shellscripting(both normal track and fast track batches)
Ctrl d

3 . $mail -s "Student file "tecno< stud ↵ It transfers stud file contents to tecno user mail box

Note : mail is the command to open mail box.
Eg: $mail ↵

Mail No.	Sender's name	Date	Time	Subject
1	Tecno	Aug1	12:21 am	Hello
2	Venkat	Mar15	6:30pm	Resume
3	Devaj	Aug18	9:30am	Student file
4	Tecno	Feb 02	10:10am	Hi

&2 ↵ [It opens 2nd mail]
&3↵ It opens 3rd mail]
&q ↵ [To quit from mail box]



Mail box options

&w filename	:	It Writes the current mail contents to given file.
&r	:	To reply
&p	:	To print out
&d	:	Delete current mail.
&d2	:	Deletes 2nd mail.
&d 1-10	:	Deletes 1st mail to 10th mail.

Networking commands

1 .telnet: Telnet protocol is In-Built in Windows/Linux . It is used for to connect to remote servers.

Syntax: $telnet ipaddress↵

Login : tecno1 ↵

Password: xxxxx↵

$

2.ftp :ftp stands for the file transfer protocol. It is used for to transfer files from one server to another server account.

Syntax: $ftp ipaddress↵

Login : tecno1

Password: XXXXX

ftp>

ftp commands

ftp>ls↵ [It displays server side files]
ftp>!ls↵ [It displays client side files]
ftp>pwd↵ [It displays server side current directory]
ftp>!pwd↵ [It displays client side current directory]
ftp>cd dirname↵ [To change directory in server side]
ftp>lcddirname↵ [To change a directory in client side]
ftp>get filename ↵ [To download a file]
ftp>mgetfile1 file2...filen↵ [To download multiple files]
ftp>mget *↵ [To download all files from current directory]
ftp>mget *.log↵ [To download only log files]
ftp>put filename ↵ [To upload a file]
ftp>mputfile1 file2 .... Filen↵ [To upload multiple files]
ftp>? ↵ [It list all FTP commands]
ftp>bye↵ [To quit from FTP].

Job Control

Jobs are 2 types.
1)Foreground jobs
2)Background jobs

Note : By default all jobs come under foreground jobs.
Examples :
$cpfile1 file2 ↵Foreground job.

$cpfile1 file2&↵ **Background job.**
512(PID)

$sort sample>a1&↵ **Background job**
513(PID)

Note: In foreground, user can execute only one job. But in the background user can execute many jobs.

1. **How to kill foreground job?**
   Ctrl c
2. **How to suspend foreground job?**
   Ctrl Z
3. **How to resume suspended foreground job?**
   $fg(or) $fgjobid

$ps (or) $ps -f ↵ It displays currently running process list in the current user account.
$ps -ef↵ It displays currently running process list in the linux server.

1. How to kill background job?
$kill pid
$kill -9 pid↵ [100% sure kill the job]

$jobs: It displays only background jobs with jobid's

$jobs↵
[1] running sleep 3000 &
[2] running sleep 4000 &
[3] running sleep 5000 &
[4] stopped sleep 6000
[5] stopped sleep 10000

1. **How to bring background job to foreground ?**
   $fg jobid↵

2. **How to send foreground job to background ?**
   First suspend foreground job by using **ctrl Z** and execute the following command
   $bg
3. **How to kill process with processname ?**

$pkillprocessname

nohup : The nohup jobs will create in server account. So nohup jobs will execute even the user disconnects from his account.

Eg: $nohupcp file1 file2&↵ The nohup command creates one nohup.out file and it appends the nohup job output into nohup.out file

Job scheduling

1 .crontab: It executes the given jobs repeatedly in the server account and by default it sends the crontab jobs output to user mail account.

By default the crontab command accepts 5 fields

1) **Minutes(0-59)**
2) **Hours (0-23)**
3) **Day (1-31)**
4) **Month (1-12)**
5) **Weekday (0(sun) -6) (or) Weekday (1(mon) -7(sun))**

How to open crontab editor
$crontab –e

* * * * * date	Every 1 minute it executes date command and it writes output into user's mail box
30 10 * * * sh script1.sh	Every day at 10:30 am it executes script1.sh and and it writes output into user's mail box
30 22 * * 0 sh script2.sh > file1 2>file2	Every Sunday at 10:30pm it executes script2.sh and it writes output into file1 file and if any errors and it writes wrrors into file2 file.
30 22 1-15 2,3,8 * sh script3.sh	2nd,3rd and 8th months first 15 days at 10:30 pm it executes script3.sh and it writes output into user's mail box
*/10 * * * * sh script4.sh	Every 10 minutes it executes script4.sh
* */2 * * * sh script5.sh	Every 2 hours it executes script5.sh
* * */2 * * sh script6.sh	Alternative days it executes script6.sh

How to list all crotab jobs
$crotab -l↵
How to remove all crontab jobs
$crontab -r↵

2.at : It executes the given jobs only once.

$at now↵ at>ls↵ at >ctrl d	It executes ls command immediately and it writes output into user's mail box
$at now + 15 minutes↵ at>shscript1.sh > file1 ctrld	It executes script1.sh file after 15 minutes and it writes output into file1 file
$at now +2 Hours ↵ at>init 0 ctrl	After 2 hours automatically it shutdowns the system
$at now + 3 days ↵	After 3 days it executes the given jobs
$at now +1 month ↵	After 1 month it executes the given jobs
$at 4PM tomorrow ↵	Tomorrow at 4'0 clock it executes the given jobs
$at 12:20amaug01↵	1st august at 12:20am it executes the given jobs

How to list at command jobs
$atq
How to remove at command jobs
$atrmjobid

3.batch : The batch jobs will execute, when server is free or server load is less.
$batch↵
at>ls↵
at>ctrl d↵

How to Zip files

1 .gzip : It is used fot to zip the file
Syntax : $gzip filename ↵
Eg: $gzipsample ↵
Note : The extension of the zip file is **.gz**

2. zcat : It is used for to open zip files
Syntax : $zcat filename.gz
Eg : $zcatsample.gz

3 .gunzip : To unzip the file
Syntax : $gunzip ` filename.gz
Eg:$gunzipsample.gz ↵

4 .compress : To compress the file
Syntax :$compress filename ↵
Eg: $compress sample ↵

Note : The extension of compressed file is **.Z**
5 .uncompress : Touncompress the file.
Syntax : $uncompressfilename.Z
Eg: $uncompresssample.Z↵

Disk status:

1.$df↵ It displays disk space in bytes.
2.$df –h ↵ It displalys disk space in kilo bytes.
3.$du ↵ It displays directory wise disk usage in form of blocks. Each block size 1024 bytes

4.date command formats

$date
Sun aug 01 12:21:30 IST 2010
$date +%d
01
$date +%m
08
$date +%a
sun
$date +%A
sunday
$date +%b
aug
$date +%B
august
$date +%D
08/01/10
$date +%d-%b-%Y
01-aug-2010

Adding/substracting days

1)	**$date**	**-d**	**"+1**	**day"**	**:**	**It adds one day i.e tomorrows date**
2)	**$date**	**-d**	**"-1**	**day"**	**:**	**It substracts one day i.e yesterday's date**
3)	**$date**	**-d**	**"+7**	**day"**	**:**	**It adds one week**
4)	**$date**	**-d**	**"-1**	**day"**	**:**	**It substracts one week**

Adding/substracting days

1)	**$date**	**-d**	**"+1**	**month"**	**:** **It adds one month**
2)	**$date**	**-d**	**"-1**	**month"**	**:** **It substractsone month**
3)	**$date**	**-d**	**"+12**	**month"**	**:** **It adds one year**
4)	**$date**	**-d**	**"-12**	**month"**	**:** **It substracts one year**

Admin Commands

Login : root
Password :

useradd : To create a new user.
Syntax : #useradd username
Eg: #useradd tecno1

passwd: To create a new password
Syntax : #passwd username
Eg : #passwd tecno1

userdel: To delete a user
Syntax : #userdel -r username

#init0: To shutdown the system
#init6 :To restart the system