UNIX Operating System:-

- ❖ UNIX is a powerful operating system which runs on every hardware and provides inspiration to the open source movement.
- ❖ UNIX is not user friendly, the user can interact with system through a command interpreter which is known as Shell.

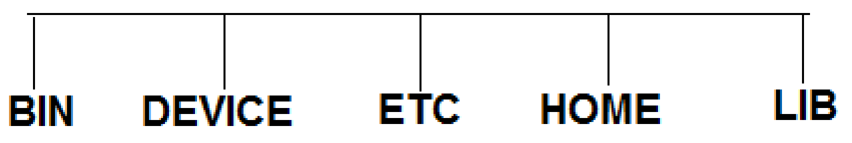
Unix & GNU

- ➢ Linux is extension of Unix, which is nothing but free version of Unix.
- ➢ GNU is a free software foundation which writes supplies of important Linux tools.
- ➢ Red Hat, Calldrak , SUSE & Mandriva are flavours of Linux.

UNIX Architecture

- ➢ The kernel interacts with the machine hardware & Shell interacts with the user this is done using special function known as system call.
- ➢ $ is regular prompt for UNIX.
- ➢ There are extra words called arguments (---) is a special argument known as option, many commands use file as argument and the command can input from file.
- ➢ Kernel is a loaded in the memory when system is booted
- ➢ The user programs access the kernel, when they need a hardware.
- ➢ This is done using special function known as system call.
- ➢ Kernel manages the system memory , schedule tasking processes and decide priority.
- ➢ Shell act as interface between users and kernel.
- ➢ Linux uses the bash shell by default although it offers the c Shell and korn shell as well.

THE DIRECTORY STRUCTURE



Features of UNIX OS:

- ➢ UNIX is multi programming with multi user system with tasking multi user.
- ➢ In UNIX jobs can run in foreground and background.
- ➢ UNIX offers features of pipe and filters.
- ➢ In pipe, output of one command goes as input to another command. Those commands that can be connected to manipulate data, so that a new job can be performed are called filters.
- ➢ Application and system administrator tools are powerful in UNIX.
- ➢ UNIX uses sophisticated pattern matching switches.
- ➢ UNIX of shell which is a programming language having loops, controls, structure and variables.
- ➢ Shell scripts of programming are made up different shell command and filters.

1. man Command:
   - This command offers online documentation of all command.
   - Eg. If we write
   - Man echo
   - It will give online documentation on echo command.
2. cal Command:
   - This command displays the calendar for the current month when given without any arguments.
   - Cal command displays calendar for any month or year between the year 1 and 9999.
   - Cal 7 2010.
   - It will displays the month july in the year 2010.
   - Cal 2010
   - It will display the calendar of all month of the year 2010.
   - Simply cal command like
   - Cal
   - It will display the calendar of June 14 i.e. current month and year.
3. Date Command:
   - This command displays the current date & time the form used on internet.
   - Date
   - We can change the system date without admin use account and password
   - There are no options with this command.
   - Date "+%a" gives abbreviated name of the week.
   - Date "+%A" gives full name of the weekday.
   - Date "+%b" gives abbreviated name of the month.
   - Date "+%B" gives full name of the month.
   - Date "+%c" gives date and time.
   - Date "+%d" or Date"+%e" gives day of the month.
   - Date "+%D" gives date in mm/dd/yy format.
   - Date "+%F" gives date in yyyy-mm-dd format.
   - Date "+%g" gives last two digits of the year.
   - Date "+%G" gives four digits of the year.
   - Date "+%h" is same as "+%b".
   - Date "+%H" give hour of the day in the time format of (00…23) in date.
   - Date "+%I" gives hour of the day in the format of (1…12) in day.
   - Date "+%j" gives day of the year.
   - Date "+%k" gives hour of the day.
   - Date "+%l" gives hour of the day in the format.
   - Date "+%m" gives the numeric value of the month.
   - Date "+%M" gives minute in time.
   - Date "+%p" gives AM/PM in time.
   - Date "+%P" gives am/pm but in lowercase in time.
   - Date "+%r" gives local 2 hours clock.
   - Date "+%R" gives time in hour and minute. ie %H%M.
   - Date "+%s" gives seconds since 01-01-1970.

> Date ”+%S” gives seconds(00...60)
> Date ”+%T” gives time same %H%M%S.
> Date ”+%U” gives week no. Of the year (00...53) with Sunday as the 1st day of the week.
> Date ”+%u” gives day of the week (1,2) where 1 is Monday.
> Date ”+%w” gives day of the week(0...6) 0 is Sunday.
> Date ”+%W” gives week number of the year(00...53) with Monday as the 1st day of the week.
> Date ”+%y” gives last two digits of the year.
> Date ”+%Y” gives year.

4. echo Command:
   > This command display the text written after the command
   > You can specify the text with or without quotes.
5. Printf Command:
   > It is an alternative command to echo command
   > The usage of this command is printf “Hello”
6. Script Command:
   > This command lets the user record the log in session in s file command is used as
   > Script
7. Who Command:
   > Displays the list of all the users who are currently connected the unix server.
   > -h option prints the result of who connected with a header line.
   > -u option gives a more detailed list of the users.
8. Uname Command:
   > Displays the name of OS.
   > -r option will give the version of the OS.
   > -n option will tell the host name or machine name.
9. Tty Command:
   > It is nothing but teletype.
   > It needs no arguments.
   > It displays the file name of the terminal you are using.
   > The location of tty command is in the directory |dev.
10. pwd Command:
    > Full form of pwd is Print Working Directory.
    > This command displays the current directory of the user .
    > This command has no arguments.
11. mkdir Command:
    > This command creates a new directory.
    > This command is followed by the name of the directory to be created.
    > Eg. mkdir abc
    > Multiple directory can be created with the command in the dame line.
    > It would created two different directory xyz & abc under the current working directory.
12. rmdir Command:
    > This command is used to remove the directory.

- $rmdir abc.
- This would be remove directory named abc.
- The rule for the removing directory should be empty and also while giving the command one should be placed in just one level above the directory.

13. ls Command:
- It is used for listing directory contents.
- -x option displays output in multiple columns.
- -f option identifier directory & executable files.
- -a option shows all the hidden begging with(.)
- -r option list all the files & subdirectory in the directory tree.
- This is also referred to as recursive listing.

14. cat Command:
- This command is used to display the content of small file on the terminal
- $ cat file name
- Option for Cat command
- -v option displays non printing characters
- -n option displays line numbers in a file.
- Cat command can also be used to create new file.
- This command used as follows
- $ cat>filename
- Cat a1>
- This will open the content to be written to file a1 when the command is terminated with ctrl+d. It is understood as end of input.

15. cp Command:
- This command is used to copy a file or a group of file.
- It creates an exact image of the file o the disk but with different name.
- This command requires at least two file names.
- Eg. Cp c1 c2
- Here file c1 is the source file & c2 is the destination file.
- The contents of file c1 are copied into c2.
- If the file already exist , it will be over written in the same directory as of file c1 , unless and until a different directory is specified.
- Cp command is used to copy more than one file in that case the last file name must be a directory.
- Here the directory must be created as the command cannot create directory.
- $cp c1 c2 c3 c4 d1
- Here the file c1 gets copied to c2, c3, c4 in directory d1.
- OPTION FOR CP COMMAND:
- -i option is used for interactive copy where user is asked over writing a file if file already exist.
- -r option behaves recursively to copy entire directory
- Eg. $ cp –r progs newprogs
- Here cp command creates a new directory new progs creates all new subdirectory of progs directory.
- If in any case cp commands faces problem one of the file will be protected or doesn't exist.

16. rm Command:

- > This command is used to delete one or more than one file.
- > $ rm file name
- > It should be used for file removal only.
- > Rm command cannot remove any directory.
- > It will remove all files from the current directory.
- > It is required that the file permission is open.
- > It is removed with rm command cannot be recovered.
- > Multiple files can be removed with single rm command and their path can be given.
- > Eg. G.m.Progs.c.progs/cz progs/c3
- > If we use * after re command it will remove all files in the current directoty.
- > The use of '*' should be done wisely.
- > OPTION FOR RM COMMAND:
- > -I option used for interactive file removal.
- > -r option will perform a recursive removal of the file & directory. When –r option is used it will partially behave like rmdir command.
- > Eg. If we write rm –r *it will remove all file and subdirectories in and below the current directory
- >

17. cd Command:

- > it is used to change the directory. Use of this command is $ cd .
- > This will take to the home directory from the current location.
- > If the command is used like $ cd.. .
- > It will take you only one level above from the current directory.

18. mv Command:

- > This command is used to rename file. Use of this command is as
- > $ mv k1 k2
- > It will rename the file k1 as k2.

19. clear Command:

- > It is gives without any argument. It clears the screen.
- > $ clear.

20. who am i Command:

- > It gives the details of the uses who has logged into the system % from the terminal.
- > This command is given as $ who am i.

A. Logout command

- ➢ This command closes the session, stops all processes and logs out of the system.
- ➢ It is given without any argument.

B. Info command

- ➢ Info gives information of the commands.
- ➢ The usage of this command is
- ➢ Info command name
- ➢ To quite man or info screen, you should press q.

C. More command

- More command is a filter for paging the screen output.
- The command is used as input to the output of first command
- Eg. Cat new1|more.
- The option –d can set a user defined page length.
- Eg. Cat new1|more -6 will display six lines at a time.
- On pressing spacebar, next six lines will be displayed and so on.
- The option –d will give instruction to be used with more command.
- Eg. Cat new1|more -6 –d.

D. Less command
- Less is filter same as more.
- While using less command, the user can scroll up or down.
- Eg. Cat new1|less.

E. tee command
- this command is used to create a new file from the standard input keyboard. At the end of the input, or can quit by giving (ctrl+d).
- if –a option is used it will append to the existing file.
- It can be written as
- Tee filename.
- Tee –a filename (to append to existing file).

21. wc Command:
- This command is used to count the number of characters words and lines in file.
- $ wc file_name
- Wc stands for word count
- OPTIONS FOR WC
- $ wc –c file_name.
- -c Option will count only char in a file
- $ wc –w file_name
- -w will give only number of words in the file.
- $ wc –l file_name
- $ wc –l option will give you number of lines in the file.
- When used with multiple file names wc produces a line for each file as well as a total count .
- Wc command like cat not only works an file but also works on a stream of data.

22. Read Command:
- This command is used to appear the inout from the user the sytem is
- $ read vari_name
- $ read val
- 111
- If we want the variable value, which we already given we have to use as follows:
- $echo $val
- It will display the value stored.
- Eg. 111.
- We can use this for strings and number as well.

23. File Command:
- This command determines the type of file given in the argument.

- One or more file name can be given as argument to the commands,There are basically three types of file
    1. Regular file
    2. Directory file
    3. Device file
- Eg. $ File abc.zip
- $ abc.zip ZIP archive
- This will show the file type abc,zip
- This file command can be used with # also.
- This will display the file type of every file.
- Eg. $ file * .
- Three modes to create shell script file:
    1. Insert mode[Press I]
    2. Append mode[Press A]
    3. Escape mode[Press Esc]

24. Od command:

- This command display the contents of the file in octal mode.
- Eg. $ od abc
- This will display the contents of file in octal
- Option with od command
- -b option will display the octal value of every character separately.
- Eg. $ od –b abc
- Hello (content)
- 110 145 154 154 157… … …
- -c option will display the charcter in ASCII value below the octal.
- Eg. Hello (content)
- 111 145 154 154 157 … … …
- H e l l o … … …

25. Cmp Command:

- This command compares two files.
- Eg. Cmp ch1 ch2
- Here in this command two file are compared byte by byte and the location of first mismatch is echoed to the screen.
- The command does not bother about any sub-sequent mismatch.
- If two files are identical cmp displays no message but returns the prompt.
- A detail list of difference in two files can be obtained with –l option.

26. Comm Command:

- This command known as common command. This command requires both the files are sorted. The use of this command is as follows:
- $ comm f1 f2
- This command will generate a 3 column output operated on filename f1 and f2.
- The first column will display the lines which are present only in file f1 and not in file f2.
- The second column will show as data represent in f2, not in f1.
- The third column will show data present in both the files. Ie. Common in both the files.

- Both the files should be sorted before applying this command.
- For example there are two files f1 and f2 as follows.
- File f1 File f2
- Hlica Jgcca
- Slica Pdpica
- Jgcca Bcca
- Cpica Hlica
- Glsica BPccs
- $ sort f1; sort f2
- Now $ comm f1 f2
- Then output will be:
- Cpica Bcca Hlica
- Glsica Bpcca Jgcca
- Ngcca pdpica
- Slica
- $ comm -13 f1 f2
- It will display only end column.
- It will not display 1st and 3rd column.

27. Sort Command:

- This command is used to arrange the content of the file alphabetically either in ascending(by default) or descending order.
- $ sort data [data is file name]
- The sort command arranges the data in ASCII COLLATING SEQUENCE [first white spaces, then numerals, uppercase letters, lowercase letters] by default the sort command sorts the entire line.
- We can sort on one or more fields (keys) or use a different ordering rule.

```
5081   Ankit     69
5087   Pratik    56
5034   Jay       89
5040   Manish    70
```

- Now $sort data will yield.

```
5034   Jay       89
5040   Manish    70
5081   Ankit     69
5087   Pratik    56
```

- Option with sort command:
- -k option is used to specify the field number on which sorting done.
- The numbering starts with the delimiter (field separator) is specified after “-t” and it is given in double quotes. If space is the delimiter it is given as “ “.

> Eg. $ sort –t “ “ –k2 data
> Output will be:

```
5081   Ankit      69

5034   Jay        89

5040   Manish     70

5087   Pratik     56
```

> -r option it will reverse the order of sorting by default the order of sorting is ascending and here it will be descending.
> Eg. $ sort –t “ “ –r –k2 data
> Output will be:

```
5087   Pratik     56

5040   Manish     70

5034   Jay        89

5081   Ankit      69
```

> Numeric sort this option used with (-n). when a file is only sorted with number the result could be unexpected.
> $ sort f1
> If we have data file as follows:
> 27
> 4
> 10
> 2
> Consisting only of number and we apply sort command the result would be:
> 10
> 2
> 27
> 4
> But we use –n option & apply the sort command
> $ sort –n f1
> Where f1 is the file name containing numeric data, the result will be
> 2
> 4
> 10
> 24
> Sorting on a secondary key. If the sorting is required on more than one key, a secondary key is provided to sort.

- If the primary key 3rd field and secondary key is the and field then –k option is specified with every key.
- Data file as follows:

5081	Ankit	Shah
5087	Pratik	Patel
5034	Jay	Sheth
5040	Manish	Patel

- If we want to sort the data as name wise then
- $ sort –t " " –k2 f1

5081	Ankit	Shah
5034	Jay	Sheth
5040	Manish	Patel
5087	Pratik	Patel

- If we want to sort the data last name and first name wise
- $ sort –t " " –k2 -k3 f1

5040	Manish	Patel
5087	Pratik	Patel
5081	Ankit	Shah
5034	Jay	Sheth

- Sorting data on columns we can also specify a character position within a field to be the begging of sort.
- In the following example sorting is to be done according to the year of birth:

5081	Ankit	Shah	22/04/90
5087	Pratik	Patel	15/08/91
5034	Jay	Sheth	18/12/91
5040	Manish	Patel	31/03/90

- $ sort –t " " -4.7,4.8 f1
- $ sort –t " " -4.7, 4.8 –k4.4, 4.5 f1
- The 1st command will sort on the 7th and 8th column position within the 4th field
- The 2nd command will primarily sort on the year and within the year it will sort on the month.
- In –k m.n format, where (n) is the character position in the mth field so 4.7, 4.8 means that sorting starts

> On 7th column of the 4th field and ends on fields 8th of the 4th field.
> Removing repeated line options:
> (-u) option which stand for unique option removes repeated lines from a file. For example
> Manager
> Clerk
> Peon
> Clerk
> Now $ sort –u f1
> Output will be as follows:
> Clerk
> Manager
> Peon
> (-o) option stands for o/p file option is used to specify the o/p file name.
> $sort –o f2 f1
> Where the o/p of vdu goes to the o/p as a file called output redirection.
> -f option sorting considers uppercase letters and lowercase letters same.
> Manager
> Clerk
> MaNager
> $ sort –f f1 will yield
> Clerk
> Manager
> MaNager
> One can always apply multiple.

28. Diff command

This command is used to display the difference between two files. this command also tell that which lines in one file have to be change so that two file become identical.

$diff f1 f2

29. gzip Command:

> This command is used to compress one file or a group of files.
> $ gzip abc.txt
> The name of the compressed file would be abc.txt.gz .
> **<u>OPTION WITH gzip COMMAND:</u>**
> -l option will show the amout of compression done for zip file.
> The usage of this command will be
> $ gzip –l abc.txt.gz
> Output will be displayed as follows

```
Compressed      Uncompressed          Ratio      Uncompressed Name

3100            4592                  35%        abc.txt
```

> If you wish to give a new name to the zip file, the command would be,
> $ gzip abc.txt newgz.gz

- Uncompressing a gzipfile command can be used as
- $ gzip –d abc.txt.gz
- Same compression can be done using unzip command. The use of this command is
- $ gunzip abc.txt.gz
- **<u>RECURSIVE COMPRESSION :</u>**
- This option will compress all the files in sub directory. It is used with (-r) option.
- $ gzip –r prog
- This will compress all files available in prog directory.
- To unzip this file the command follow as:
- $ gzip –dr prog

30. tar Command:

- It is used to create a disk archive that contains a group of files or an entire directory structure.
- **<u>OPTION WITH FOR COMMAND:</u>**
- -C option creates an archive.
- -X option extract file from archive.
- -t option display file in archive.
- -v option display while archive.
- To create an archive it is required to specify the name of the archive with (-f) option the copy or write operation with (-c) option and the file name as argument.
- The command to create and archive is
- $ tar –cf archive.tar
- The command to see the files in existing archive is
- $ tar –tf archive.tar
- To extract the file from tar command.
- $ tar –xvf archive.tar
- For single file
- $ tar –xvr archive.tar data.c

31. Chmod Command:

- Chmod stands for changing the mode of a file.
- This command is used to change the permission of a file or a directory is created with a set of permission & this default is determined by the administrator.
- Generally the default setting write protected a file from all except the user/owner.
- All other user will have read access for the file.
- The operation on file could be ot assign or remove a permission & the type of permission can be read, write, execute.
- The chmod command changes permission on a file in two different ways.
- In a relative manner by specifying the change to the current permission.
- In an absolute owner by specifying the final permission.
- This command is run only by the user/owner & the super user/administrator.
- ***<u>Relative permission:</u>***
- The command assigns permission with a (+) command and (-) command is used to remove the permission.
- The user identified by (u), group by (g), and other by (o).

- The symbol (a) refers to all that is user, group & others relative chmod by default the permission for only file is _ r w _ r _ _r_ _.
- i.e. read and write permission for the owner /user of the file and read permission for group and others.
- $ chmod u+x o-r f1.
- It will add the executable permission for the file f1 for user and remove the read permission for others.
- $chmod u-x g+w o+rw f1.
- This will remove the executable permission for the user, adds writes permissions for the group and adds read and write permission for others.
- ***Absolute permission:***
- The expression used by chmod is a string of three octal numbers.
- A set of three bits can represent on octal digit. Here permission of each category is presented by one octal digit.
- Read permission: octal 4(100)
- Write permission: octal 2(010)
- Executable permission: octal 1(001)

Binary	Total	Permission
000	0	_ _ _
001	1	_ _ x
010	2	_ w _
011	3	_w x
100	4	r_ _
101	5	r _ x
110	6	r w_
111	7	r w x

- There are three categories & three permissions for each category hence three octal digit can describe file permission completely.
- Option with chmod command:
- Recursive option of chmod command. It is indicated by
- Eg. Chmod –R a+x.
- This makes all files & subdirectories found in the three walk from the shell scripts directory executable by all the users.
- Shell scripts
- Executable by all the users
- Eg. $ ls –l f1
- _r w _ r_ _r_ _ it is by default permission.
- To give each and every permission to all
- $ chmod 777 f1

- It will give all permission to all.
- Now we give
- $ls –l f1
- O/P will be
- -r w x r w x r w x f1
- If we give the command
- $ chmod 466 f1
- It will give read permission to user, read and write permission to group and read write permission to others.

32. Find command:

- This command is used to locate the file. It is a powerful tool of the UNIX operating system.
- It recursively examines a directory tree to look for a file matching some criteria and takes some actions on the selected files.
- $find path_list selection_criteria actions
- The command operates as follows
- first, recursively examines all files in the directory specified in path list.
- It then matches each files for one or more selection criteria.
- Finally, it takes some action on those selected files.
- The path list compares of one or more sub-directory separated by white space.
- $find /-name "*.c" –print
- The above command will locate files name with *.c extension from the root directory and they will be printed. It is also possible to use relative path names like the following command.
- $find . –name "*.c" –print
- $find . –name '[a-z]*' –print
- The first command look for all c program in the directory tree.
- The second command searches for all files whose name begins with lowercase letters.
- $find . –type d –print
- $find $HOME –perm 777 –type d –print
- The above command finds directory which have all the permission , print it. Above command is logical operator (&) is used.
- $ find . ! –name "*.c" -print
- In this command it will find all the files whose extension is other than .c
- (!) sign indicates that NOT Operator.
- $ find /home\ (-name "*.sh"-o –name ".pl"\)-print
- This command print all the files which will have name have (.sh) or (.pl).
- SELECTION CRITERIA
- -inum n :- Select file having number (n).
- -type x :- selects file of type x where x can be (f) which is ordinary file,(d)directory or(l) symbolic link.
- -perm mn :- selects the file if octal permission matches with nnn completely.
- -Size + x :- selects the file if size is greater than x block.
- -user uname :- select the file if it is owned by uname.
- -mtimex :- select the file if it is modified in less than x days.

- -ls :- The output will be printed on the terminal.

33.

34. Head Command:

- This command displays the top of the file when this command is used without an option it displays the first 10 lines of the file.
- $head data.sh
- This will display 1st 10 lines of file data.sh. if we give the command:
- $head –n6 data.sh
- It will displays only first 6 lines of the file data.sh.

35. Tail Command:

- This command displays the last part of the file.
- By default it displays last 10 lines of the file.
- $ tail filename
- If command given as follows:
- $ tail –n 6 filename
- It will display last 6 lines.
- If we give
- $ tail +6 filename
- It will display all the line of file starting from line 6 to end of file.

36. Cut Command:

- This command is used to slit the file vertically.
- -c option will display from column character 6 to 22 & 25-27 character for every line of file f1.
- $ cut –c 6-22, 25-27 f1
- -f option will cut the field number 2 and 5 from file. When field is used the delimiter is also used, given after –d " " option.
- $ cut –d " " –f2, 5 f1
- If we give
- $ cut –d " " –f2 , 5, f1>> aaa
- It will save the output of the above command in aaa file.

37. Paste Command:

- It is used to paste the contents of two files side by side vertically. Tab is the default delimiter. You can specify one or more delimiter.
- $ paste filename1| filename2
- -d option will put field separator (/, \, ., ; ,etc) between both the file.
- $ paste –d"/" f1 f2.

38. Uniq Command:

- It is used to locate repeated and non repeated lines.
- $ uniq filename
- This will show only the uniq lines of the file.
- The requirement of this file is, file should be sorted.
- -d (selecting duplicate lines) option selects only one copy of the repeated lines.
- $ uniq –d f1
- -u option selects lines that are not selected.
- $ uniq –u f1
- -c option will count the frequency of occurrence of all lines in the file.

39. tr Command:

> it is used for translating filter character and words like a filter. They take input only from output. Input device does not take filename as argument.
> $tr '\' '-'<emp
> Above command will translate every backslash with a dash(-) in the file emp.
> If we give
> $head –n3 emp | tr '[a-z]' '[A-Z]'
> Here in head command output of the file emp goes as input in tr command. This will convert the first 3 lines of emp file to uppercase letters.
> -d option will delete the character mentioned in the command.
> $ tr –d '/' <emp
> aaa		12/12/1991
> bbb		04/06/1990
> ccc	09/12/1990
> output will be:
> aaa		12121991
> bbb		04061990
> ccc	09121990
> -s option will compares multiple occurrence of a character where s stands for **squeeze.**
> $str –s ' ' <emp
> This will reduce multiple occurrence space into single space into file.

40. grep Command:

> It is used to search for pattern. This command scans its input for a pattern and displays lines containing the pattern, the line number or file names where the pattern occurs.
> The syntax of grep command is
> *Grep option pattern filename(s).*
> Grep searches for pattern in one or more filename or from the standard input into filename are specified.
> $grep "sales" emp
> It will search for lines containing the string sales from the file emp.
> Grep command is also a filter and it can search its standard input for pattern and save the standard output in file.
> If matching pattern is not found the grep command silently return the prompt.
> $who | grep "Kumar" >a1
> Above command will displays who current login in the UNIX their details and whose name is Kumar will stores in file a1.
> The above example has 3 features viz. Pipeline, filtering and output redirection.
> When grep is used with multiple filenames, it displays the filenames along with the output.
> -i used to ignore the case.
> -v option placed the role of inverse it will select all the lines accept those containing pattern.
> $grep –i –iv "sales" emp f1 f2
> It will show all the content without sales.

- -n option will display line number containing the pattern.
- -c option will count number of lines containing a pattern.
- -l option will displays only name of the files containing the pattern.
- -e option is used to match multiple patterns.

41.

42. Set Command:
- The set command assign its argument to position parameters $1,$2,$3.....(system variable).
- This feature is specially used for picking up individual field from the output of the paragraph.
- Example:
- $ set 123 456 789
- This will set the value of $1= 123, $2=456, $3=789.
- If we give:
- echo $*
- Output will give the positional parameter set=123 456 789.

43. Shift Command:
- It is used to transfer the content of its positional parameters to its immediate lower numbered.
- $ shift 1
- It will give output $2=456,$2=789.

44. PS Command:
- This command gives name of the process currently run.

45. Name of the Shell Command:
- The command $ echo $ shell will give the name of the shell currently run.

46.

47. pr Command:
- This command is used to converts text files for printing. The syntax is as following :
- Pr [options] [file]
- -d double space the output.
- -h header uses a centred header instead of filename in page header.
- -l page length sets the page length specified. Default page length is 56 lines per page.
- -t omits page headers and trailers.
- -w page width sets the page width specified.

Finger command:

- This command gives information about the system users.
- **<u>OPTIONS WITH FINGER COMMAND:</u>**
- -s displays the users login name, real name, terminal name, write status, idle time, login time.
- -l produces a multi-line format displaying all of the information in –s option as well as users home directory, login shell, home phone etc.

mesg command:

- This command controls write access to your terminal.
- Usage: Mesg [y/n]
- It disallows other users to write to your terminal.
- If no option is given, it prints out the current access status of your terminal.

Wall command:

- This command send message to everybody's terminal.
- Usage: wall [-n] message
- When –n option is used to banner of "Broadcast Message" given.

Umask Command:

- This command sets default files and directory permission. This default permission is transformed by subtraction from the user mask to remove one or more permission.
- The usage of this command when done without any arguments can be as follows:
- $ umask
- That means 0222 is an octal number which when subtracted from the system default give us the actual default which 666-022=644(_rw_r_ _r_ _).
- Umask is shell built in command and the user can set their own default permission as follows:
- $ umask 111
- Will change permission for files as 666-111=555
- Permission will be like this (_r_x r_x r_x).

talk command:

- This command is a visual communication program which copies lines from your terminal to that of another user.
- ***<u>OPTIONS WITH talk COMMAND:</u>***
- -person : if you wish to talk someone on your own machine, then person is just the person's login name.
- -ttyname : if you wish to talk a user who is logged in more than once. The ttyname argument may be used to indicate the appropriate terminal name, where ttyname is of the form 'ttyxx' or 'pts/x'.

mailx command:

- Mailx can be used to send mails to users within a machine a remote machine or web users using a relay server.
- There are two ways of invoking mailx in the sending and receiving modes.
- ***<u>OPTIONS WITH mailx COMMAND:</u>***
- -s option is used to specify the subject.
- -c option is used to send copies. Multiple recipients should be enclosed in quotes.