

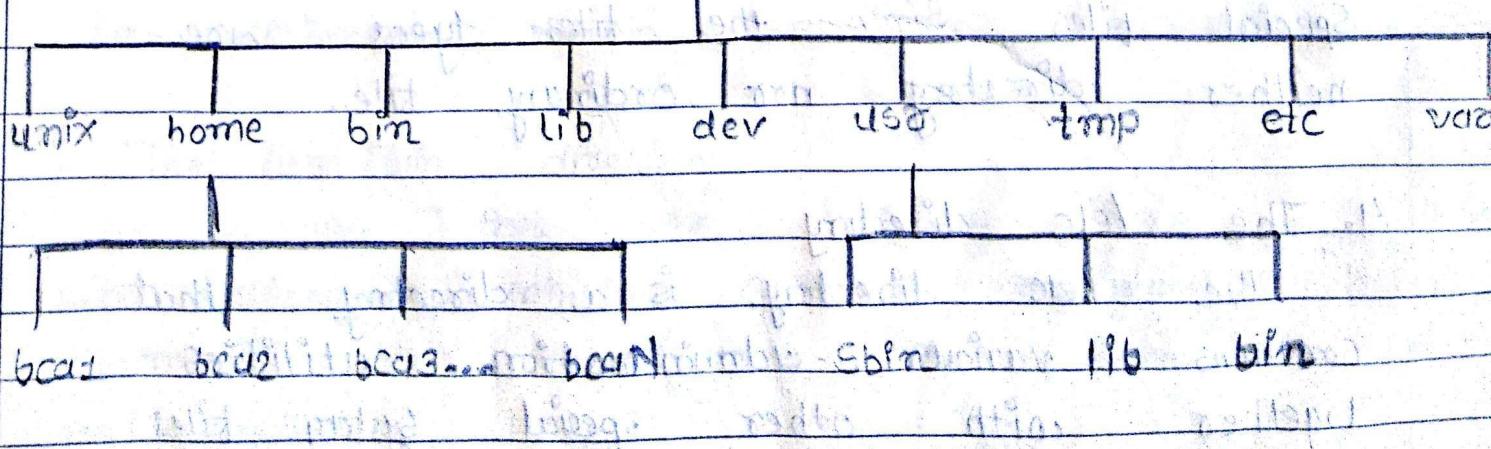
Question - Answer.

Q Explain Directory Structure of unix. [Four times]

Ans. In Unix, all utilities, applications, data and even a directory is treated as a file. A directory contains several other files or directories. Moreover, all devices connected to Unix system are also treated as file.

The unix file system has a hierarchical structure which resembles an upside down tree. Thus, at first level of file system there is a directory known as root directory. The root directory is denoted by slash (/). Under the root directory, there are several other sub-directories called bin, usr, etc, tmp and dev. The root directory is a parent of all these directories. Each of these sub-directories contains several files and directories called sub-sub-directories.

The root directory also contains a file called 'unix', which is Unix kernel itself. The basic structure of unix.



1. The root directory.

The root directory is denoted by '`/`'. It is also known as the super user's home directory. The user who logs in as root will be placed in a root (`/`) directory initially.

2. The /bin directory

The `/bin` directory contains commonly used unix commands that all users will have to use at some time. The word '`bin`' stands for binary, hence this directory contains binary files. Binary files are executable files.

3. The /dev directory

All device files are stored in `/dev` directory or in its subdirectories. Unix treats devices such as printers, disk storage devices, terminals and even areas of the computer's memory as a file. The files in `/dev` are termed as special files, since the file types are neither directory nor ordinary file.

4. The /etc directory

The `/etc` directory is a directory that contains various administration utilities together with other special system files.

Such as /etc/inittab, /etc/gettydef and so on that allow the host unix system to start up properly at bootstrap time. Utilities for handling the system's terminal devices are stored in /etc.

5. The /sbin and /usr/sbin directory

It contains commands that cannot be used by ordinary users but the system administrator can execute it. On older System, these directories may not exist where /etc could contain all administrative files and commands.

6. The /tmp directory

The tmp directory contains the temporary files created by unix. Users are allowed to create temporary files. These file are automatically deleted when the system is shutdown or restarted.

7. The /home directory

This is the home directory of all registered users of unix system. The system administrator creates these directories when he creates account for different users.

8. The /usr/bin directory

Within the usr directory there is another bin directory which contains additional Unix command files that are more important the end - user.

9. The /usr/src directory

This is a directory that contains the source code for many of the binary utilities.

10. The /lib directory

The /lib directory contains all library functions provided by unix for programmers.

11. The /var directory

The variable part of the file system. It contains all your print jobs, mail queues and incoming mail.

2. Explain Features of unix operating system. [Five Times]

Ans..

1. Multi - user Capability.

Multi - user features allocates users to use the resources of main computer. In other words, at a time same computer resource, such as hard disk, memory, printer etc..., are available to many users.

The total numbers of terminals connected to system depends on the numbers of ports available on controller card of host machine. The terminal are different types, some of the terminals are as follow:

a) Dumb Terminal :-

The terminal consists of only a keyboard and a monitor is known as dumb terminal.

This terminal has no memory or disk so it can not operate as an independent machine.

b) Terminal Emulation :

This terminal consists of keyboard and monitor with its own microprocessor, memory, and hard disk drives. The terminal is linked to the host machine through a cable. The user runs terminal emulation software from the terminal to connect with host machine.

C) Dial - In Terminal:

If the host machine is at a remote location then Dial - in terminals is used. The terminal is connected to the host machine through the telephone lines.

2. Multitasking Capability

Another main feature of unix is multitasking. This features enables user to carrying out more than one job at the same time. i.e. in Unix, a single user can also run multiple task concurrently.

In multitasking environment, user can start a job, such as print a file on the printer and then move to another job, such as waiting for an email, without leaving any of the applications.

3. Inter process Communication

This features allows user to communicate with users. This features allows user to pass on data, exchange mail, or program to another user within networks.

4. Security

Unix provide three type of securities.

System level Security :

Unix, every user has been

allocated login id and password, when the system administrator opens cm accounts for user, an entry is created in system password file, called /etc/passwd.

Directory level Security:

In Unix, everything is treated as file. Even directories or devices are also considered as file. There are read, write and execute permissions to each file, which decide who can access a particular file, who can modify it and who can execute it.

File level Security:

Lastly, there is file encryption. Encryption utility encodes your file into an unreadable format, so that even if someone succeeds in opening it, your secret information are safe.

5. Portability

Unix system is written in a high level language i.e. in C. moreover, C programs are easily moved from one hardware environment to another.

6. Open System

Unix has an open architecture; one can add to the tool kit by simply writing a

a program and storing the executable in a separate area in the file system.

7. Windowing System.

Initially, Unix has a pretty weak user interface in that it was command-driven. Faced with increasing competition from Microsoft windows, it was obvious that sooner or later Unix had to come up with its own GUI.

8. System calls and libraries

Unix is written in C languages. There are many commands available in Unix that handles specialized functions called system calls. These calls are built into the kernel, and all library functions and utilities are written using them.

9. Programming facility

Unix is highly programmable; it was designed for programmers, not a casual end user. The unix shell programming language has all the necessary ingredients like control structures, loops and variables that establish it as a programming language in its own right.

10. Networking

Unix was not originally a networking system. This concept was added to unix system after a split between BSD Unix and AT&T Unix.

3. Explain features of Shell. [Three Times]

Ans.. The Unix system's shell includes the following major features :

1. Interactive Environment :

The shell allows user to create a dialogue, i.e. communication channel, between the user and the host unix system.

2. Shell Scripts :

It is the shell that has the facility to be 'programmed'. Shell Script is a file that contains shell commands that perform a useful function.

3. Input / output redirection :

I/O redirection is a function of the shell that redirects the output from program to a destination other than the screen.

4. Piping Mechanism :

A pipe operator receives its input from standard output and sends it to the next command through standard output. Using piping, programs that perform simple functions can easily be connected to perform more complex functions, minimizing the need to develop new programs.

5. Metacharacter facilities :

Shells recognize the *, ?, or [...] as special characters when reading the arguments at a command line.

6. Background processing :

A multi-tasking facility allows the user run command in the background. This the command to be processed while can proceed with other tasks.

7. Customized environments :

The shell is your working environment. Facilities available by which the shell can be customized for your personal needs.

8. Programming language Constructs :

The shell includes features that allow it to be used as programming language.

9. Shells Variables :

The user can control the behavior of the shell, as well as other programs and utilities by storing data in variables.

- 4 Booting Sequences with init process. [Five Times]
- Ans. When the system starts, it performs following steps:
- : ROM diagnostics are run; here hardware and memory tests are performed.
 - : Boot loader is loaded: The ROM BIOS reads the boot loader from boot block and loads it into a RAM.
 - : Kernel is loaded: The kernel program is called by the user. Recall that the user needs to enter the name of the program.
 - : Kernel initialization takes places: It involves performing memory tests, initializing the devices through device drivers, Swapper scheduler process, the init process and many more.
 - : The init program resides in /etc directory which is invoked by the kernel.
 - : The init program: This process has the PID 1 which is the second process of the system. The init program reads the instruction of /etc/inittab file to carry out processes like identifying the run level i.e., the mode in which system should run single user/multi-user, maintaining files to track activities etc.
 - : The getty process: The init program also invokes the getty program which establishes a

Communication with the terminals of the Unix System. The `getty` program uses the file called `/etc/gettydefs` for instructions to provide to the `login` prompt at each terminals connected to system and goes into suspended mode and activated when user attempt to login.

The `login` program : Once the user types login name and password, `getty` transfers control to a `login` program, to verify the login-name and password entered by user.

`init` → `getty` → `login` → `shell`

Now, `init` is the only living ancestor of the shell which goes to sleep mode and waiting for the death of its children. When user logs out, shell is killed and the death is intimated to `init`. `init` then wakes up and spawns another `getty` for that time to monitor the next login.

5 Explain modes of vi editors. [Five Times]

There are three basic mode of vi editor

i) command mode

ii) insert mode

iii) last-line mode.

i) Command mode.

By default, any file opened in vi editor is in command mode. This is the default mode of vi editor. Most of the letters, or short sequences of letters typed by user in this mode will be interpreted as commands.

ii) Insert mode.

If a user wish to insert in a file whatever typed from keyboard at the cursor position then insert mode is used. To switch from command mode to insert mode, a user has to use insert mode command. e.g type `a` to enter into insert mode press `<esc>` to end insert mode, and return back to command mode.

iii) Last Line mode.

Last line mode is used for line oriented commands. In command mode, if user type a colon (`:`) then the cursor moves to the bottom line of the screen, with a colon prompt. Type a line mode command and then press `<enter>`. Each time you use a line mode command, you must type a colon to enter in line mode, then type the command at the colon prompt and then press `<enter>` when you finish typing the command.

6 Write a shortnote on grep. [Five Times]
Ans.: It is a filter utility that performs various tasks as follow:

- : It scans a file for the occurrences of a pattern and displays lines in which scanned pattern is found.
- : It scans a file for the occurrences of a pattern and displays lines in which scanned pattern does not found.
- : It scans files for the occurrences of a pattern and displays name of files which contains a pattern in them.

Syntax

Grep [Options] pattern [filename(s)]

It is used to select and extract lines from a file and print only those lines that match a given pattern. In the above syntax square bracket indicates optional part.
eg: \$ grep 'unix'

Grep Metacharacters

Character

(.) or [-] It matches any one single character within a square bracket.

^ pattern

It matches a pattern at the beginning of each line.

pattern\$

It matches a pattern at the end of each line.

(dot) It matches any single character except new-line character.

(backslash) It indicates that grep should ignore the special meaning of the characters following it in regular expression.

(pattern) It matches a pattern at the beginning of any word in a line.

(pattern) It matches a pattern at the end of any word in a line.

(ch *) It matches zero or more occurrences of character ch.

(ch | {m}) The preceding character ch is occurred m-times.

(Options):

i) -c : It prints count of matching lines for each input file.

e.g. \$ grep -c '.' f2

ii) -l (list) : It displays only the names of files in which a pattern has been found.

iii) -n (number) : It can be used to display the line numbers containing the pattern, along with the lines.
\$ grep -n 'Unix' f2

iv) -v (inverse) : The -v option select all but not the lines containing the pattern.

v) -i (ignore) : It ignores case in pattern matching
\$ grep -i 'unix' file

vi) -h (hide) : It omits filename when handling multiple files.

\$ grep -h 'unix' file

7. Write a short notes Set command. [Three Times]

Ans.. A user can customize editing environment of vi editor. A last & line mode commands can make changes to the environment of an editor.
Set Command.

Command functions for customization Significance.

:set nu or ~~Assume~~ It sets display of line numbers
:set number on.

:set nonu or ~~Assume~~ It sets display of line numbers
:set nonumber off.

:set vb or ~~Assume~~ It beeps a speaker when an error occurs.
:set errorbells

:set nobe or ~~Assume~~ It does not beep a speaker
:set noerrorbells off when an error occurs.

:set ai or ~~Assume~~ It Sets auto indent on.
:set autoindent

- :set noai or :set autoindent It sets auto indent off.
- :set ic or :set ignorecase It ignores case while searching a pattern.
- :set noic or :set noignorecase It does not ignore case while searching a pattern.
- :set showmode It displays mode in which all users are working.
- :set noshowmode It doesn't display current working mode.

[Four Times]

8. Explain Access Permissions of Unix file system.

Ans.. In Unix, all files and directories have permission. There are three file level and directory level permission : read, write and execute.

Read permission : r

Write permission : w

Execute permission : x

In Unix, file permissions are assigned numerical octal values from 0 to 7.

Read permission : 4

Write permission : 2

Execute permission : 1

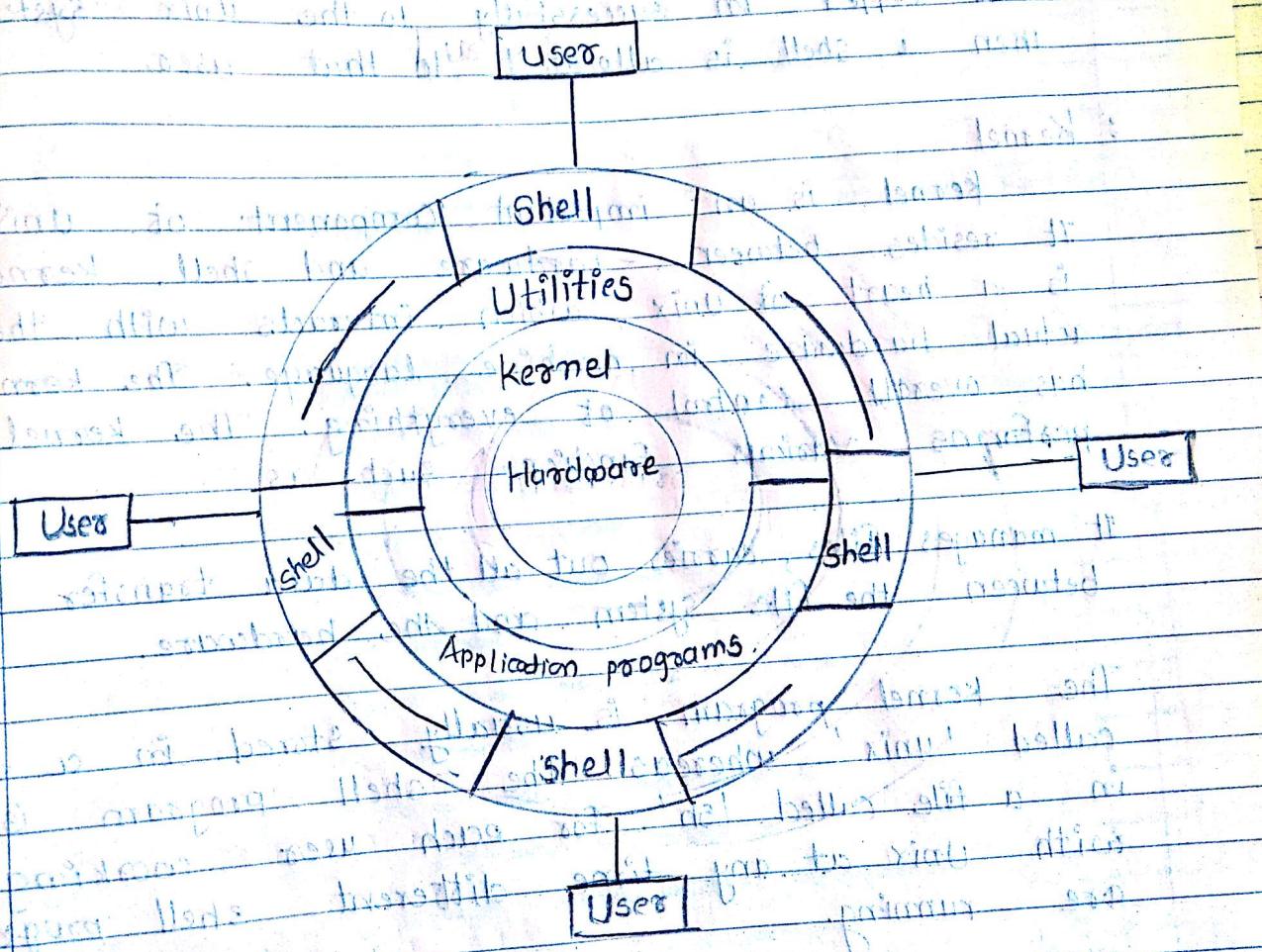
- Thus, for a file that has a permission field like `-rwxrwxrwx`, the permissions are said to be read, write and execute for the owner, groups and others.
- Similarly, a file with permission `-rw-rw-rw` is said to have permission read and write to owner, group and user. In octal notation, it would be written as `666`.
- A zero value in the permission field is treated as a complete removal of the read, write and execute permission for all.

Representation of Permission in octal and symbolic form.

Permission	Symbolic code	Octal value
No permission	<code>---</code>	<code>0</code>
Execute	<code>x</code>	<code>1</code>
Write	<code>w</code>	<code>2</code>
Write and execute	<code>wx</code>	<code>3</code>

Q. Architecture of Unix Operating System. [Four Times]

Ans. Unix is a layered operating system. There are three major components of Unix operating system.



: Hardware :

The innermost layer is the hardware that provides the services to the operating system.

: Shell

The shell is an outer layer. It is a command interpreter which interprets the commands.



Input is supplied by user at shell prompt. Then, shell converts interpreted command to the kernel which ultimately executes them. When any user logged in successfully to the unix system then a shell is allocated to that user.

• Kernel

Kernel is an important component of Unix OS. It resides between hardware and shell. Kernel is a heart of Unix, which interacts with the actual hardware in machine language. The kernel has overall control of everything. The kernel performs various functions such as

It manages files, carries out all the data transfer between the file system and the hardware.

The kernel program is usually stored in a file called 'unix' whereas the shell program is in a file called 'sh'. For each user working with Unix at any time different shell program are running.

• Utilities

Utilities are standard Unix programs that perform system functions. Utility can also refer to as a command that is used to do work such as mv is used to move files or directories. Text editor, search programs, sort program etc.

In Unix, some of the utilities are complicated applications. e.g. the unix email system, text editors such as vi, emacs and pico. All these utilities are large systems in themselves.

: Applications

The programs written by system administrators, professional programmers, or users are known as applications. Applications are not a standard part of a unix system. They provide an extended capability to the system.

10. Explain I-node file structure. [Three times]

Ans. We know that all entities in unix are treated as files. I-node (block) stores information about the file and the location of the data block where the content of file is actually stored.

The information related to all these files is stored in an I-node table on the disk. For each file, there is an I-node entry in the table. Each entry is made up to 64 byte and containing the relevant details for that file.

- : File type: regular, directory, device etc.
- : Number of links i.e. the number of aliases of the file.
- : Numeric UID of the owner.
- : Numeric GID of the owner.

- file access permission : read, write, execute
by owner, group and others.
- Size of the file.
- Date and time of last modification of file data.
- Date and time of last access of file data.
- Date and time of last changes of the i-node.
- Table of contents : disk addresses of disk blocks containing the file data.

The i-node is accessed by a number called the i-node number. Any index node is a block that holds pointers to all the blocks used by a file. This number is unique for every file in a single file system.

As with Super block, the Kernel also maintains a copy of the i-node block in its memory. Therefore when a file is opened, its i-node is copied from the hard disk to the system's i-node table which is maintained in memory.

The kernel always works with the memory copy, but it periodically updates the disk copy with the content of the memory copy.

11 Write a short notes on Sed command.

[Four Times]

The Sed supports Several command.

- 1) print
- 2) quit
- 3) Line number
- 4) Modify
- 5) Files
- 6) Substitute

i) print(p) Command : It is denoted by character p. It prints selected lines on a standard output.

Syntax \$cat file | sed -n p F1
e.g. \$sed -n p F1

ii) quit(q) command :

This command is denoted by character q. It uses single address.

\$ sed 'q' F1

iii) Line number (=) command :

It is denoted by equal sign. It writes line numbers of addressed line at the beginning of the line.

\$ sed '=' F1

e.g. \$sed -n '=' F1

iv) Modify Command :

There are different purposes of this command. It allows you to insert, append, changes or delete lines.

a) Insert Command:

It is denoted by a character i. It inserts one or more lines directly to the output before and address lines.

```
$ sed 'i' <enter>
```

b) Append Command:

It is denoted by a character a. It is similar to the insert command except that it writes the text directly to the output after the specified line.

```
$ sed '$a' <enter>
```

> unix is portable operating system

> f1

c) Change Command:

A change command is denoted by a character c.

It replace addressed /matched line with new text.

```
$ sed 'c' <enter>
```

> unix & shell programming

```
$
```

d) Delete Command:

Using the d command, we can simulate -v option of grep utility to select lines not containing a pattern.

```
$ sed '/linux/d' f1
```

> red hat linux

> Linux is open source

>

v) File Command :

File command is used to read or write data to or from other files respectively. There are two types of file command : (i) read file (ii) Write file

a) read file command :

It is denoted by r frame. When a user wants to insert common content of a file after specified line of an input file then this command is useful.

```
$ sed '1!u/nx/r f2' <f1> >f1.yub
```

b) Write file Command :

It is denoted by w frame. The write file command makes it possible to write the selected line into a separate file.

```
$ sed -n '1!u/nx/w f1.out' f2
```

vi) Substitute Command :

It is denoted by a character s. It scans a line for search pattern and substitutes it with replacement string.

[address or scanned-pattern] s/search-pattern/replace-string/[Flag(s)]

12. Write at & Batch Command. [Three times]

at Command :

It is used to execute a command at a specified time and date. The general form of this command is as follow:

Syntax:

at [option] job-name (s) or filename time - format.

It reads commands from standard input or a specified file which are to be executed at a later time.

Time Format used with at command.

Format

Meaning

HH:MM

It runs a job at a specified time of day. If that time is already past, the next day is assumed.

noon

It runs a job at 12:00 hour.

midnight

It runs a job at 00:00 hour.

teatime

It runs a job at 4:00 pm.

time MMDDyy or

It runs a job at specified time on a specified date.

time MM/DD/yy or

time DD.MM.YY

now+count time-units

It runs a job at a specified time.

time today

It runs a job today at specified time.

time tomorrow

It runs a job tomorrow at specified time.

You can give following command to send a message to user bca1 at 8:30 pm.

\$ at 18:30 <enter>

at > echo -e "Urgent call ln at TMTBCA" | write bca69 <enter>

at > (ctrl + d)

job 641 at 2014-03-25 18:30

\$

Batch Command

It allows unix to execute your job, as and when the system load is light. The batch command is useful for running a process or shell program that uses a large amount of system time.

Syntax : batch [option]

\$ batch

Sort myfile > sortfile

ctrl -d

job 855 at 2014-03-26 13:07

Any job scheduled with batch command also goes to at queue. Therefore, to display all the jobs submitted by batch will be displayed using at -l and also be removed with the command at -r job-number.

Q3. Explain communication command.

[Five Times]

Ans.: Write :

It is on-line communication command. It sends a message to another user who is currently connected to the unix system.

Syntax : write user [ttynum]

The write command provides a facility to communicate with other users, by copying lines from his terminal to other user's terminal.

e.g \$ write Bruno <enter>

1) Finger : Finger command tells you that who are currently connected to the system and who can receive messages on their terminals.

2) Who - T : The who - T command lists all the users who are currently logged in and places a '+' next to the username who have allowed messages and a '-' sign if does not allowed.

\$ who - T

: mesg : Command.

It controls write access to your terminal by others. It is used to allow or disallow other user to write to your terminal.

Syntax : mesg [y/n]

e.g \$ mesg

Options used with mesg command.

option

Meaning

y or Y

It allows write access to your terminal.

N or n

It disallows write access to your

: Inall :

It sends a message on the terminal of all the users who are currently logged in to the Unix system. This command is specially designed for the super user.

Syntax :

sudo wall [message]

\$ wall "Good morning students"
Broadcast message from Bharat (Thu Sep 5 16:53:07 2013):

: motd :

It is a special file resides in /etc directory. If a super user wishes to put an off-line message to the users then he typed message in their motd file.

: mail :

It is off-line command command. It sends and receives mail. Unlike write and talk, it allows user to send a mail to user even if they are not logged in to the unix system.

Syntax :

mail [-s subject] [-c cc-addr] [-b bcc-addr] to-addr to-addr2.

-mail -f [name]

-mail -u user

(i) -s : It specifies subject of the mail on command line.

(ii) -c (carbon copy) : It sends blind carbon copies to list.

(iv) -f : It reads the contents of your inbox or the specified file for processing.

(v) -b (blind Carbon copy) : It sends blind carbon copies to list. List should be a comma separated list of names.

(vi) -u : It is equivalent to -f option.

news command :

This command is invoked by users to read any message that is sent by the system administrator.

i) -n (name) :

It displays only the name of files whose contents are not been read before.

\$ news -n

news : mess1 mess2 mess3

ii) -g :

It lists the number of news items that have still not been read.

iii)-a :

It displays the contents of all news items regardless of whether they have been read or not.

\$ new mess

--- display news --

\$.

: talk :

It is used to talk to another user who is still connected to the system. It is superior to write command that it splits the screen horizontally into two windows.

Syntax: talk person [ttyname]