

Windows disadvantages

- security
- Only one user can connect a machine (multiuser support not available)

Unix

first unix release is an enterprise one (need license to purchase)

- HP UX (HP Unix)
- Dec (IBM)
- Solaris (Sun)

Linux (Unix only but open source) → FreeBSD, CentOS, Ubuntu, Fedora, etc.

Redhat → Enterprise of Linux but need to buy. (did changes to linux and released a new version)

Shell → Shell is an environment which acts as an interface between the Unix system and user.

→ Shell gathers input from the user and execute programs.
 when a program finishes executing, it displays the programs output.

Any OS contains some files which we called as OS.

→ Kernel (Heart of OS)

Takes i/p from user & interacts with hardware and do processing.

On top of kernel, we have shell.

This is the environment to interact with kernel.

Here, env means some specific configuration which provides interaction between kernel & shell.

In the shell, we can have certain inbuilt commands, external files, OS libraries

These are called as utilities through which we will tell exactly

what is the instruction to execute to the shell,

we will execute them from shell only. They are like wrapper over a shell.

Like copy is a utility we run on shell to copy some files.

we can create these utilities and also some are provided by OS directly.



shell types available : specific env.s for specific tasks.

- Bourne shell (sh) → (o)
- Korn shell (ksh) → (o)
- Bourne Again shell (bash)
- C shell (csh) → (o)

outdated → (o)

due to some security issues and complexity.

bash is the default shell now, in many OS.

To check it, Terminal → echo \$SHELL

OS → basically we have set of files through which we can do something.

Unix File System Basics

<u>Directory</u>	<u>Description</u>
/	Root directory. (like c:\ in windows) no logical drives like in windows, they loaded as separate sub folders.
/bin	contains executable files. some times we have <u>/usr/bin</u> also.
/dev	contains device drivers
/etc	contains system related configuration files
/lib	contains shared library & kernel related files
/proc	All process related info for dynamic info of the system
/tmp	Holds Temporary files.
/var	contains variable-length files such as log files.
/sbin	contains binary for system administrator.
/mnt	used to mount temporary file system - CDROM.
/home	Home directory for user accounts. separate sub folder will be created for each user

when a new account to be created,

Admin will create

- * user id (unique)
- * password

Background → * creates a home directory. or ^{admin can} ~~user~~ change it.

* sets env. for the shell

* login scripts (runs as soon as logs in) everytime.
we can put some values here.

If we create any variable to store something in the environment (shell) then are called as environment variables.

\$env → To show all the list of env. variables, for each user available.

To create a variable. → <variable-name> = <value>

Ex: ~~MyName=Uday~~
MyName=Uday.

↳ Represents arbitrary values.

To display it, echo \$MyName → If not exists, just displays empty data.
use \$

OS also stores some default env. variables.

We can set our env. variables in login scripts to make them available everytime we login.
which is → which command shows the location of the command.
Some env. variables .profile / .bashrc

PATH → If we run a command without path, shell searches for it in the locations specified in PATH variable (separated by ;)
In the profile script add a new app location,
PATH = \$PATH; <new-path>

Unix - File Management.

- All data in Unix is organized into files.
- All files are organized into directories.
- These directories are organized into a tree-like structure called the file system.
- In Unix there are three basic types of files -
 - Ordinary files - contains data, text or program instructions
 - Directories - stores both special and ordinary files
 - Special files - files provide access to hardware such as hard drives, CD-ROM drives, modems and Ethernet adapters.

- There is no file extension concept in Unix.

- Root (r) is ^{at} the top

- Home directory → For every user a directory is there which contains user specific files & permissions
To go to home directory, cd ~ pwd → present/current working directory.

To get the position of any file is described by its pathname. Elements of pathname separated by /

Absolute path \rightarrow from root \rightarrow /c/users/uday/Desktop/a

Relative path \rightarrow from current working directory \rightarrow Desktop/a

cd .. \rightarrow goes to parent directory [from /c/users/uday \rightarrow /c/users]

cd ../.. \rightarrow goes to parent's parent. [from /c/users/uday \rightarrow /c]

. \rightarrow Refers to current directory.

mkdir dirname \rightarrow to create a directory in current ^{working} directory.

rmdir dirname \rightarrow to delete a directory.

renaming directory \rightarrow mv <name>

touch test \rightarrow creates an empty file.

Every file in unix has the following attributes.

- Owner permissions (Default owner is the creator of that file.)
- Group permissions (Group associated with the file) (default owner's group)
- Other(s) permissions (user who is not in group or owner)

To see the permission indicators, ls -l.

\$ ls -l /home/uday

file/dir access modes:

Read (r), write (w), execute (x)

Access mode numbers, octal values

\rightarrow 4 2 1

Let's say for a file we give permissions like.

Owner \rightarrow rwx

group \rightarrow r-x (no write)

others \rightarrow r-- (only read)

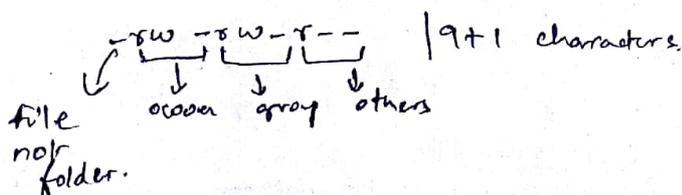
touch TEST

ls -l ~~TEST~~ \rightarrow long listing of current directory properties. it will show file permissions also.

drwxr-xr-x

First letter represents it as a directory.

If no d, like (-) then it's a file.



ls -l o/p

File size

rw-rw-r-- 1 user user

owner group

0 Feb 2 14:13 TEST

File or folder name.

Change the access levels

chmod

chmod <permissions for owner> <permissions for group> <permissions for others> filename

We want to give,

-rwx -rwx -r--
4 2 4 2 4
6 6 4

So, chmod 664 filename

777 rwx
6 6 rwx
4 rwx

su - root

To change owner, chown

chown USER FILELIST

we need to sudo if we get permission error

chown root TEST

To change group of file, chgrp

chgrp GROUP FILELIST

(or) chown root:root TEST
owner group

To act recursively, ownership of all the files -R

ls -la

to test hidden files also.

File/Folder with . in front, is a hidden file.

cp TEST1 TEST2

a new copy is created with different name.

to move or rename,

mv TEST1 TEST2
renamed 1 to 2.

mv TEST1 chef/recipe/TEST1
moved to chef/recipe folder with target name mentioned.

cat file1 → to see contents
vim file1 → vim editor
head file1 → first 10 lines printed.
head -n 2 file1 → first 2 lines
~~head~~ tail → reverse of head

rm -rf testy/
recursively &
forcibly.

ls -R
It shows recursively all files.

* → All possible values.
* → represents everything
touch 1 2 3 4
rm * → Deletes all files current directory

ls -d | head.
 ↳ pipe
 used to pass
 o/p of 1 end left side
 to right side

grep "i" DELME

search for strg "i"
 in file DELME
 and o/p the line
 matching strg

Ex: grep "h" cday.txt

hi there busy
 bello

echo "hello there" | grep "there"
 hello there

grep -r "he" DELME

↳ reverse, so displays lines
 without the strg

echo "see me"

↳ to screen/console.

redirecting

echo "Hi boys" > file.c

↳
 ↓
 for append it to existing file,
 if not, creates new file

ps

all processes running by the current user.

whoami

current user.

ps -ef

emulated
 full form.

provides processes
 run by other users
 also.

ps -ef | grep \$USER (or) ps -ef | grep USERNAME

lists any created by us.
 with full details.

widely used
 secure shell
 Remote connection
 ssh (or) rsh (remote shell)

ssh userid@password
 (or)

userid@MACHINENAME
 and prompts for
 password