

What is
hardware

Physical component which we can see, touch or feel is called hardware.

What is software

Software refers to a program or set of instruction and application used to manage and control various functions of a computer is called software.

Types of software

1)application software

2)system software

Application software performs information processing task for end users.

eg word, excel, c++ etc

System software manage and support operation of computer it is basically two types

1)operating system

2)System utilities

Utility software is used to perform basic maintenance task on a computer

1)disk cleanup

Disk cleanup allows for you to scan your entire hard disk drive to search extra room by deleting any unnecessary files from the internet and cookies.

2)Disk Defragmenters

The main function of the disk defragmenters is to reassemble fragmented files. Designed to increase access speed by rearranging files stored on a disk to occupy contiguous storage

3)disk compressor and Archivers

Disk compressor is a type of function that a program to search your hard drive and compress file particularly old or unused files. This greatly improves your computer's functionality and performance because it does not have to keep track of many files.

4)System restore

System restore is great for fixing problems that a virus has caused after you use antivirus software to rid your

computer of the malware.

5) Registry cleaners

Registry cleaners are programs that allow for you to scan your computer for any errors in the registry which is a collection of the core computer files that are essential to performance and functionality and repairs them if needed.

6) File Splitters

are programs that allow you to break a file into smaller pieces in order to store or send files.

functions of os

- 1) file management
- 2) memory management
- 3) I/O system management
- 4) command interpreter system
- 5) program execution
- 6) file system management communication
- 7) protection system
- 8) networking

broadly divide into

memory management

security

multitasking

peripheral management

What is unix

Unix is CLI(command line interface) operating system

features of unix

- 1) multiuser capability
- 2) multiprogramming
- 3) portability
- 4) security
- 5) machine independent
- 6) strong networking support
- 7) unix shell programming
- 8) pipes and filters

basic unix commands

who

whoami

date

cal

pwd

ls

ls-l

for zooming \$ ctrl ++

cd

cd ..

cd home

man ls

man cd

man date

cat > filename (create a file) > standard output redirection operator

cat < filename (display the file not necessary to use <) standard input redirection operator

cat >> filename (append the file)

ctrl+d used to save and exit from the file

man cat

unix is case sensitive

command should start with alphabet and with lower case and special symbols and

commands are reserved words

date +%a

date +%A

date +%b

date +%B

man date

date +%y

date +%Y

date +%T

date +%F

cal

cal -j

who -d

who -b

file * will show the empty files

cd ~ default directory

```
cd - current directory
cd ../../ this will come out from two directories
mkdir
```

```
vi file1.txt
aaa
:wq
```

```
# super user(administrator) prompt su - password
# su - oracle
$ su - password
```

```
$ normal user prompt
```

```
$ whoami
oracle
```

```
$ su - password
```

Collection
of unix
commands
is called
shell
scripts

Some time one command or 2 or 3 command is not sufficient. We have to run
collection of commands then use shell
script

.sh files

```
$echo$0
bash prompt $ sh
$ksh korn shell
$echo$0
ksh
$exit
```

Feature

- == lump sum commands is executed
- == shell repeatedly execute command

shell introduction

shell script file means a file which contains a set of commands with in it .
If any file contains commands
than that file can be executable file. Shell scripts can be useful to execute
the set of commands at a single moment of time.

use of shells

- 1)customizing your work environment
- 2)automating your daily task
- 3)executing important system procedure like shutting down the system
- 4)performing same operting on many files
- 5)automating repetitive task

types of shells

- 1)Bourne shell
- 2)C shell
- 3) Korn shell

The Bourne shell

The original unix shell is known as sh short for shell or the Bourne shell,
named for Steven Bourne, the
creator of sh.Bourne shell has been considered a standard part of unix for
decades. The shell prompt is \$,
Execution command sh

The C shell

Designed by Bill Joy at the unversity of California at Berkeley, the C shell
was so named much of its syntax parallels that of the C programming language.
The shell prompt is % and execution command csh.

The Korn shell

The Korn shell became one the main salvos in AT&T response to the growing
popularity of BSD unix created by
David Korn at AT&T Bell Laboratories. The Korn shell or ksh , prompt is \$

Responsbility of shell

- 1)program execution
- 2)variable and file substitution
- 3)I/O redirection

- 4) pipeline hookup
- 5) environmental control
- 6) interpreted programming language

shell variables

Variable is data name and it is used to store value. Variable value can be changed during execution of the program

variables are of two types

- 1) system variable
- 2) user defined variables

system defined variables

Environmental variables used to provide information to the programs you use.

You can have both global

environment and local shell variables. Global environment variables are set by your login shell and new programs and shells inherit the environment of their parent

shell

some common ones are

- 1) display the graphical display to use
- 2) editor the path to your default editor
- 3) group your login group
- 4) Home path to your home directory
- 5) host the hostname of your system
- 6) IFS (internal field separators (tab spacebar enter etc))
- 7) logname the name you login with
- 8) path paths to be searched for commands

`$x=33` press enter

`$echo $x` press enter

output 33

`$echo $0`

output /bin/bash

`$ksh` press enter

`echo $0` press enter

output ksh

`$echo $x`

so it is not possible to use because where it is declared there only we can use.

`$export x` to make global

```
$echo $x  press enter
output 33
$echo $0   press enter
/bin/bash
$ksh  press enter
$echo $x
output 33
$exit  press enter    to come out of it
```

shell keywords

```
echo
case
while
if
wait
do
read
done
ulimit
set
shift
break
exit
unset
exec
umask
until
continue
trap
read only
```

shell scripts are of two types interactive and non interactive shell script.
Interactive means input
required from user at execution time and non interactive means just opposite.

```
$pwd
$ls
1st shell program
$vi myscript.sh
#it is my first script
echo "welcome to the world of shell scripting"
echo "default script is bash"
echo Thank you!
```

```
:wq
```

```
$sh myscript.sh
```

```
2nd shell program
```

```
$vi users1.sh
```

```
#Number of users
```

```
echo currently logged in users are who | wc -l
```

```
echo currently logged in users are who -q
```

```
echo logged in users count is users | wc -
```

```
echo success
```

```
:wq
```

```
$ sh users1.sh
```

```
3rd shell program
```

```
$ vi files.sh
```

```
# Number of files
```

```
echo "number of files in the current directory"
```

```
ls | wc -l
```

```
echo "are existed"
```

```
#display the domain name
```

```
hostname -d
```

```
#display IP address
```

```
hostname -i
```

```
#display current week number
```

```
date +%u
```

```
#display linux flavor name
```

```
uname -s
```

```
#display name of the processor
```

```
uname -p
```

```
#display date in yyyy/mm/dd
```

```
date +%F
```

```
#display day number in a year
```

```
cal -j
```

```
echo all commands successfully compiled
```

```
press esc
```

```
:wq
```

```
$ sh files.sh (this command is for execution of the shell )
```



```
4th shell program
$ vi noextension
#script without .sh extension
echo "welcome BASH shell programming"
echo "Thank u"
press esc key
:wq press enter
$ sh noextension
```

```
5th shell program (interactive shell script type)
$ vi read1.sh
#reading your name
echo -n "enter your name"
read name
echo you entered Mr/Ms $name
:wq
$sh read1.sh
Enter your name rajesh
you entered mr/Ms rajesh
```

```
6th shell program (interactive shell script type)
$ vi add.sh
#reading any two numbers
echo -n "enter any number"
read num1
echo -n "enter any number"
read num2
echo entered numbers are : $num1 $num2

:wq
$sh add.sh
```

```
Entered numbers are : 5 6
```

```
-----
$a=200
$b=300
$echo $a+$b
200+300
$echo $[a+b]
500
```

```

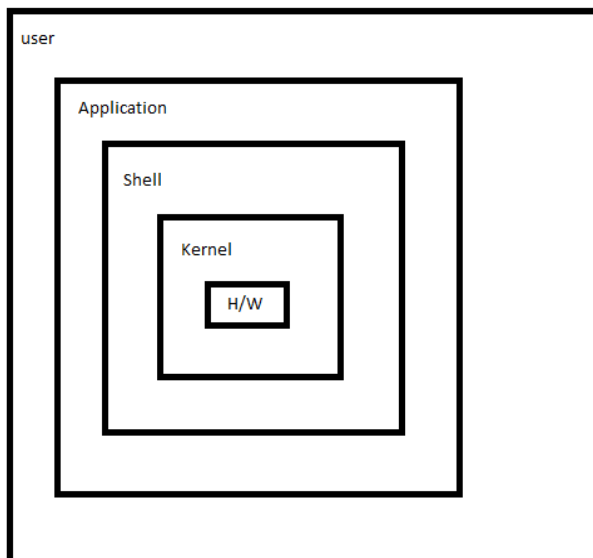
$ vi read2.sh
#Arithmetic operations
echo -n "enter any number"
read fno
echo -n "enter any number"
read sno
echo "sum of two number is : " echo $((fno+sno))
echo "Diff of two number is : " echo $((fno-sno))
echo "product of two number is : " echo $((fno*sno))

```

```

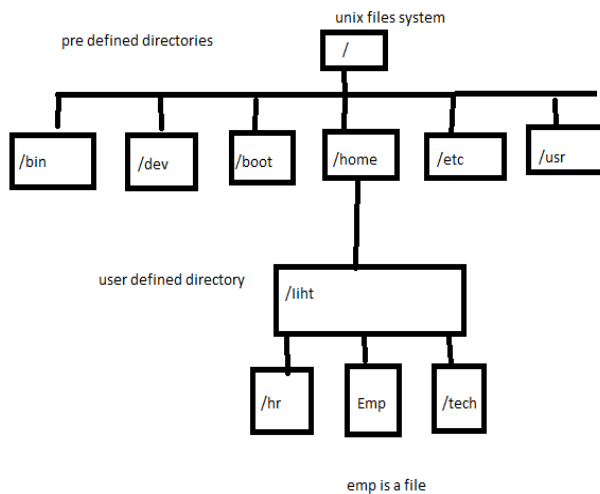
press esc key
:wq press enter
$ sh read2.sh
5
6
sum of numbers :11
Diff of two number is : -1
product of two number is : 30

```



kernel is interface between shell and h/w and kernel's main task is device management, memory management

Kernel is a low level programming
Shell is a high level programming



/bin binary contains all executable files
 /dev it contains all device drivers
 /boot contains all objects required for booting the system
 /etc contains all configuration file and disk file and information of super user
 /home contains all users and sub user details
 /usr any software installed first it will install by default in usr

unix flavours are
 IBM ----- Aix
 Oracle ----- sun solaris
 HP----- unix

unix file system organization are of two types
 physical and logical block

logical block is divided into 4 types
 boot block super block inode block data block

super block it contains file system related how many blocks are allocated

boot block bootable object required to boot the system

inode block here i stands for index and index will be unique number it contains
 type of file and type of mode

data block where actual data is stored

unix files colors
 blue or dark blue : Directory
 green : Executable or recognized data file
 sky blue : Linked file
 yellow with black background : Device
 pink : Graphic file or image file
 red : Archive file or zip file

unix files notations
 - ordinary or regular file
 d directory
 b block special file
 l symbolic link
 p named pipe
 s socket
 @ linkd file

Command : A command is a specific instruction given to a computer application to perform some kind of task or
 function