

Introduction to Unix:

1. Solaris is the name of a flavor of Unix from _____
 - a. Sun Microsystems
 - b. IBM
 - c. HP
 - d. Digital Equipment Corp
2. Which helps us to access the functions provided by the kernel to do services of the OS?
 - a. System call
 - b. Command
 - c. Shell
3. Match the following
 - a. HP-UX - Sun Microsystems
 - b. Solaris - Hewlett packard
 - c. Linux - IBM
 - d. AIX - RedHat
4. The OS decides which process gets processor, when and for what time, it is called _____.
 - a. MultiTask
 - b. MultiProgramming
 - c. MultiUser\
5. What will keep track of printers and the USB connected to a computer?
 - a. I/O controller
 - b. Traffic controller
 - c. File System
 - d. Process Scheduling

What is an Operating System?

Operating system is a system software which manages hardware and software resources.

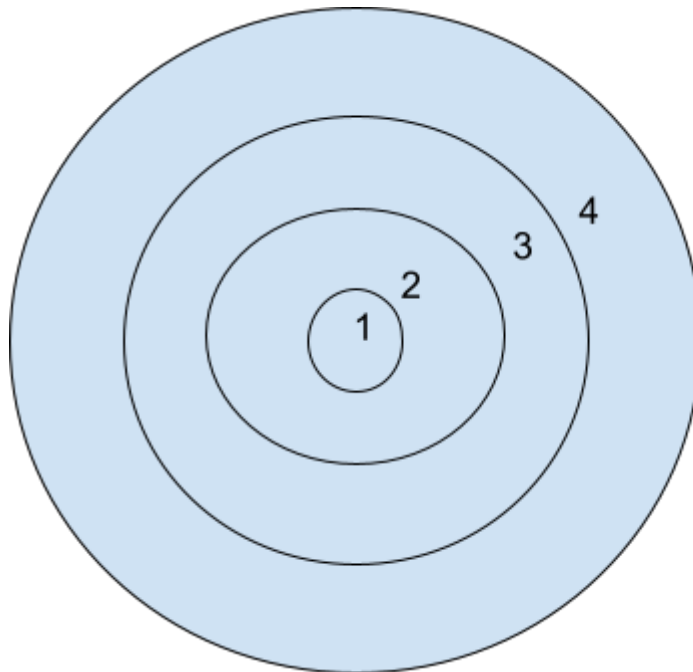
There are a number of Operating Systems in the market like Windows, Solaris, Unix, Mac OS and so on.

Unix based Operating System is the most preferred Operating System in the server side. The reason being SECURITY.

Unix is more secure when compared to the other Operating Systems.

Unix has three components

- Kernel
- Shell
- Utilities



1. Hardware
2. Kernel
3. Shell
4. Utilities

Kernel is the core part of an Operating System which directly communicates with the hardware.

Kernel has four subsystems.

1. **Memory Management - allocation and deallocation of memory for processes**
2. **Process Management - Creating, terminating and counting of a process**
3. **I/O Management - Control communication**

4. File Management - managing files

In unix like Operating System there are different shells available, the common shells are

1. Bourne Shell - Steve Bourne
2. C shell - Bill Joy
3. K shell - Dawid korn
4. Bourne again shell - FSF

As we have read all this let us now deep dive into OS

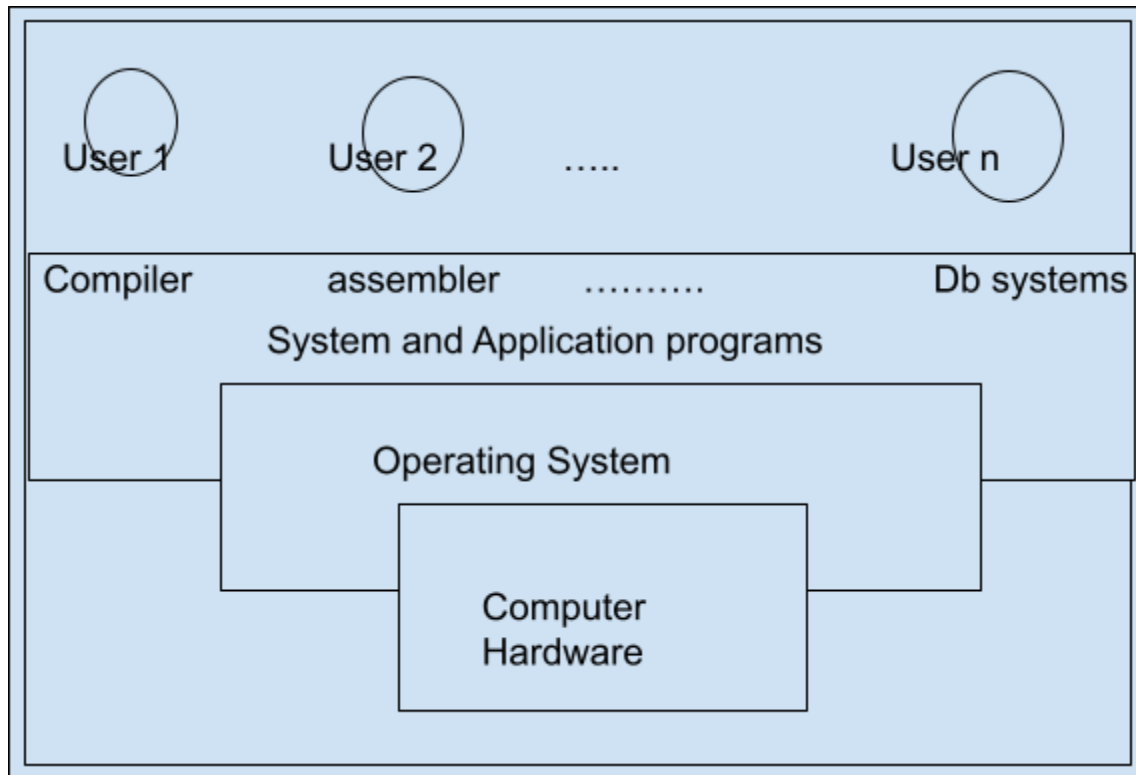
Introduction to OS

Operating System (OS) provides **interface** between computer hardware and software that runs on it.

The important functions of Operating System is

- 1. Managing memory***
- 2. Managing files***
- 3. Managing processes***
- 4. Managing devices***

And **providing security.**



Now let us start off with a UNIX OS

UNIX fundamentals

Unix OS is a ***multi tasking and multi user*** Operating system, that is, it allows two or more users to run the programmes simultaneously and it supports two or more processes running at the same time.

Unix is ***more secure*** when compared to other OS because unix has lower access rights and theoretically viruses can access local files and folders.

Unix is ***an open source software*** and is distributed freely, it follows ***console-terminal*** architecture.

Console has processing capability and many ports.

Terminal has a monitor and keyboards.

UNIX Architecture

As we know it has components like

1. Kernel
2. Shell
3. Applications

Users can not directly access the kernel but kernel can access the hardware directly.

The Kernel:

Kernel is ***the core of the operating system***, which is a collection of routines mostly written in C. It is loaded into memory when the system is booted.

It provides ***services through functions called system calls*** used by user programs to access system resources like hardware or terminal.

As we know it directly communicates with the hardware.

Features of kernel:

1. *It manages the files on the disk*
2. *It manages the memory*
3. *It manages processes*
4. *It manages the system security*
5. *It manages the network*
6. *It manages the devices.*

The Shell

Shell is ***an interface between the user and the kernel***. It is written in C and command interpreter.

When a login authentication is successful the ***shell*** is created and its life time is until the login session terminates.

It has programming capabilities.

There are lots of services provided by the shell, those are

1. *Translation(meta character)*
2. *Input/Output redirection*
3. *Background processing*
4. *Pipes*
5. *Personalizing the environment*

Now we are clear with Unix OS and its features, let's proceed working with the unix environment.

From windows if we want to access a unix machine we need *a terminal and provided with the ip address and the port number* to connect with the unix machine.

Now the user can login in using *username and password*, once the login is successful the user will be placed at the *home directory*.

The user is allowed to work in the system once the login details are verified. When logged in, the user is presented with a *prompt(\$ or #)*, it's the way the shell requests a command.

To logout from the unix machine there are three ways to do it.

1. *Ctrl+d*
2. *exit*
3. *logout*

The Command

The command is *an instruction given by a user* to the computer to do a particular task.

Once the command is given by the user *it passes to the shell*, command interpreter reads the command and executes them.

In Unix commands are *case and space sensitive*.

Syntax:

Command [options] [arguments]

The **inputs** are given as **arguments**.

The **outputs** are customised using **options**.

There are many commands in Unix. If we want to create a folder in windows we can just right click and create it. But in unix we use a command (mkdir) to create a directory. Likewise there are more commands to explore.

Let's start with the date command and discuss

date:

It is used to print or set the date and time

Syntax:

date [options][+format][arguments]

Options:

-r

--date

%d,%D,%b,%B,%m,%F,%H

passwd

Used to change the password of a user. The password should be minimum three characters.

Syntax:

\$passwd

cal

To display the calendar of current or specific month of a year

Syntax:

cal [options][arguments]

Example:

\$cal (display the current month of the calendar)

\$cal 2000(treated as year)

\$cal 7 2000(first argument treated as month and second as year)

Options:

-s,-m,-y,-j,-1,-3

bc

Stands for basic calculator

When bc is given, it expects inputs from the keyboard when given it prints the result

To exit from bc, we can use ctrl+d

Example:

```
$bc
```

```
2+4
```

```
6
```

```
ctrl+d
```

```
$
```

dc

dc is a *reverse polish* desk calculator, it stores numbers on a stack. When the numbers are given as input they are pushed into the stack and arithmetic operations pop arguments and push results into the stack.

Options

p print the top of the stack

q to quit the dc command

Explore

n, P, f

who

To display the information of all the users who are logged in.

It displays the name, date and time, terminal and with the ip address.

Syntax

who[options]

Example

```
$who
```

Options

-q, -U, -H

finger

finger command is also like a who command displays information about the users who have logged in.

It displays the name, tty, idle, login time, office, officephone

The only difference is that it displays in columns

Syntax

\$finger

whoami

This command used to print the username associated with the current effective user id

Syntax

\$whoami

who am i

This command displays the information of current user

It displays the name, date and time, terminal number of the user.

Syntax

\$ who am i

tty

It is used to know the terminal number in which the user is connected

Syntax

\$tty

df

df command displays the amount disk space available on the file system

It expects filename, if no filename is given all currently mounted filesystems is shown

Syntax

df[options][filename]

Options

-H, -i

man, info

If we want to self learn about a command these two can be helpful

Provides online manual pages as help for the unix commands

Syntax

\$man<command>

\$info<command>

cal : it displays calendar

date : it displays current date and time in the system

clear : it clears the terminal screen

df : it displays the free number of blocks and files

who am i : it displays the information for the currently logged in user

whoami : it displays effective current user name

who : it displays currently logged in users in a system

man : it displays current manual pages for the commands