



What is an Operating System?

- Operating system is an interface between the user and the computer
- In simple terms, It manages all the available resources on a computer, from the CPU, to memory, to hard disk accesses.
- Tasks the operating system must perform
 - Control Hardware (**Hardware Management**)
 - Run Applications (**Process Management**)
 - Manage Data and Files (**Memory Management**)
- Types of operating systems.....
 - UNIX, Microsoft Windows, Mac OS, Android, Linux





Why an Operating System (OS) ?

- OS interacts with hardware and manages programs.
- Allocation of memory (Memory Management)
- Process Management
- Programs are not expected to know which hardware they will run on.
- Must be possible to change hardware without changing the programs.
- Programs can't manage themselves
- OS provides a safe environment for programs to run.



What is Operating System?

- Operating system is a software between the user and the computer.
- It manages all the available resources of a computer.
- It provides a platform for application programs.
- It provides a common interface for users.
- It performs the basic management.
- It has four main components: Process Management, Memory Management, Device Management and File Management.
- It is a system program.



Why an Operating System (OS)?

- OS makes hardware and programs compatible.
- Allocation of memory (Memory Management).
- Process Management.
- Programs are not expected to know which hardware they will run on.
- Must be possible to change hardware without changing the programs.
- Programs can't manage themselves.
- OS provides a safe environment for programs to run.



How a Program Runs on a Computer



Why an Operating System (OS) ?



• OS interacts with hardware and manages programs.

• Allocation of memory (Memory Management)

• Process Management

• Programs are not expected to know which hardware they will run on.

• Must be possible to change hardware without changing the programs.

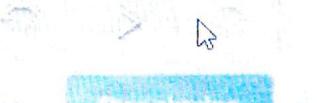
• Programs can't manage themselves

• OS provides a safe environment for programs to run.



How a Program Runs on a Computer

- OS loads program from disk and allocates memory and CPU.
- Instructions in program are run on CPU and OS keeps track of last instruction executed.
- If program needs to access the hardware, OS does the job on its behalf.
- OS saves the state of the program if program has to leave CPU temporarily.
- OS cleans up memory and registers after process has completed execution.





What is UNIX?

- ◆ UNIX is an operating system which was developed at Bell LABS in early '70s by **Ken Thompson and Dennis Ritchie**
- ◆ Originally written in assembler, later rewritten in C (allowing greater portability), a language invented by Ritchie
- ◆ UNIX stands for **Uniplexed Information and Computing System** (UNICS→UNIX)
- ◆ UNIX provides a “universal” operating system, runs on most machines being built today
- ◆ Linux is the PC version of UNIX





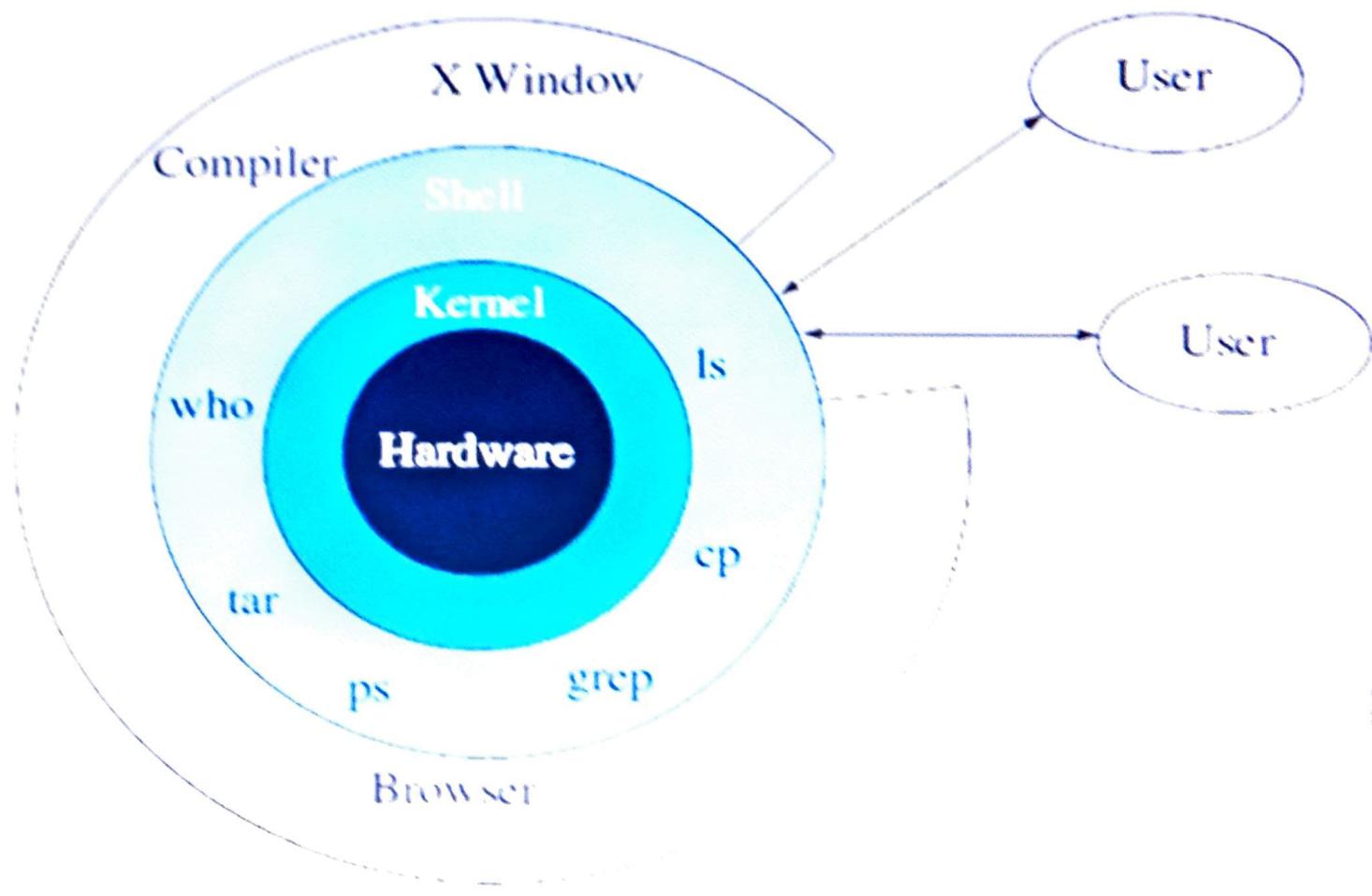
Structure/Architecture of Unix System

1. Kernel
2. Shell
3. File System
4. Commands / User Programs / Utilities



Structure of Unix System

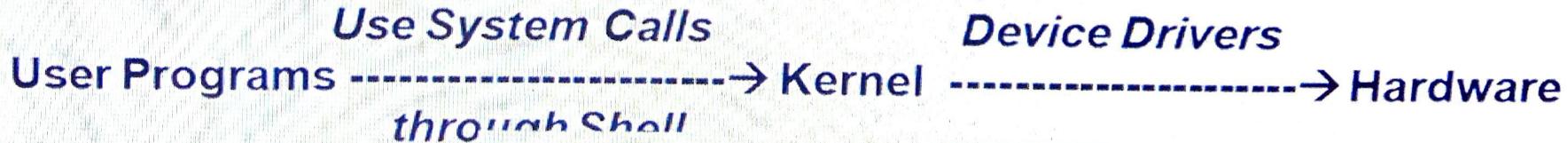
Inside UNIX



UNIX Architecture - The Kernel



- Kernel is a part of the OS (collection of routines written in C) that interacts directly with the hardware of a computer, through device drivers that are built into the kernel
- It provides a set of services that can be used by all the programs, by which it is insulating all the programs from the underlying hardware
- Kernel software always resides in memory and has direct access to the hardware (File I/O)



UNIX Architecture - The Kernel



- Major functions of **Kernel** which are executed by the program through **System Calls**
 - Managing Memory
 - Control Access to Computer
 - Maintains File System
 - Handles Interrupts/Signals
 - Handles Errors
 - I/O Services
 - Allocates Resources among Users



UNIX Architecture - The Kernel



- **Micro Kernel**

- The Kernel is grouped into several modules, the just-necessary module, consisting of a small set of kernel programs is called a micro-kernel
- Micro kernel alone is loaded into memory when booted, whereas other modules are moved in and out of memory as and when required



UNIX Architecture - The Shell

- Shell is a program that sits on the Kernel and acts as an agent or interface between the user, kernel and hence the hardware
- The shell presents a Command Line Prompt(\$ or % or #) at which the user can type any valid unix command
- The Shell reads your commands and interprets them as requests to execute a program or programs, which it then arranges to carry out, hence shell is called a Command Interpreter

**● REC****M**

MCA CLASS is presenting

UNIX Architecture - The Shell



- **A Shell**

- Presents a Prompt
- Interprets Commands
- Executes Commands
- Provides users and programs with Environment

- **After accepting a user command the Shell**

- Parses the Command
- Evaluates the Variables
- Does Command Substitution
- Interprets Meta Characters
- Identifies the Path
- Checks for Redirections

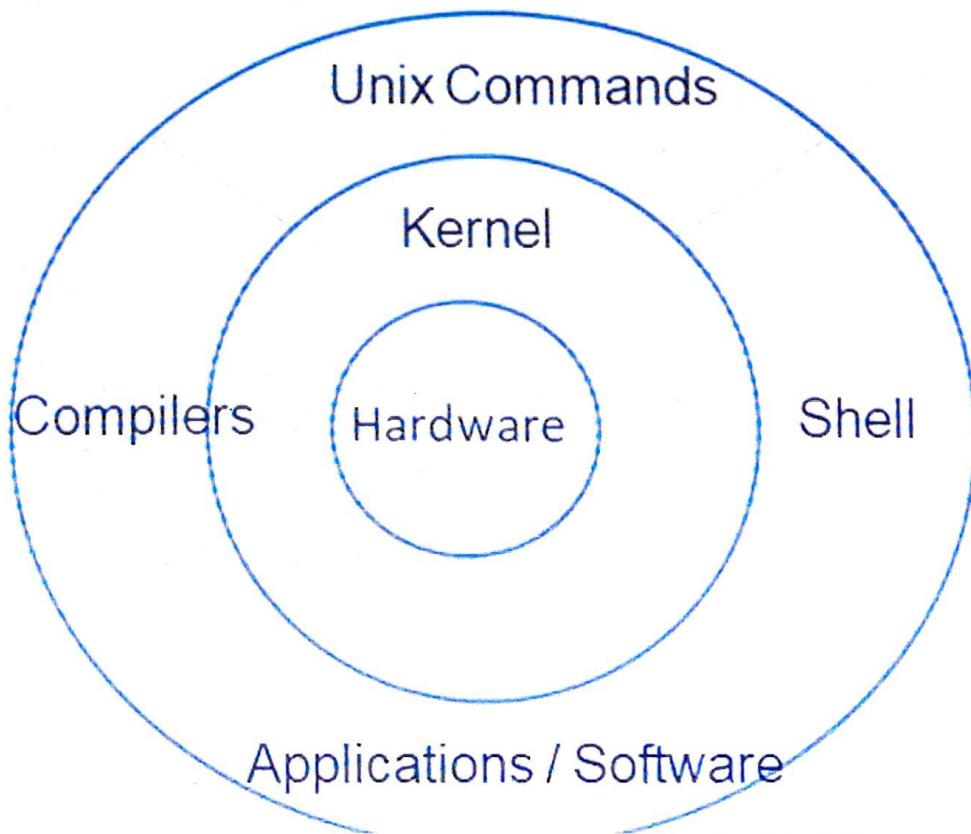
- Rebuilds the command line into a simple readily executable command, then passes this on to the Kernel for execution

• **Shell has programming capability of its own**
• **Shell Programs are called as Shell Scripts**

UNIX Architecture - The Shell



- Makes calls to the kernel for all other functions.
- At least one shell is invoked by every user.
- User has a choice of shells.





Types of Shells

- **Bourne shell (sh)** – most common shell in every UNIX system
- **C shell (csh)** – syntax is very similar to c language
- **Korn shell (ksh)** – same as bourne shell and incorporates certain features of c shell
- **Bourne again shell(bsh)** – mostly used in linux operating system as its default shell





Files

- The basic unit used to organise information in the unix system is called a **File**
- The unix file system provides a logical method for organising, storing, retrieving, manipulating and managing information
- Files are grouped into directories
- Physical devices are also treated as files, so that same commands can be used for ordinary files and for physical devices





Commands and Utilities

- There are various command and utilities which you would use in your day to day activities
- **echo, who, cat** are few examples of commands and utilities
- There are over 250 standard commands plus numerous others provided through 3rd party software
- All the commands come along with various optional options
- Commands are also called as **tools**, because they can be used separately or together in various ways to execute tasks
- They are executed by their names through the shell, hence called as **commands**



Commands and Utilities

- Many tasks can be performed using the **standard utilities** supplied with unix
 - Text editing and processing
 - E-communication and networking
 - Performing calculations
 - Developing programs
 - System administration



Basic Commands

- passwd
- ls -l
- who
- w
- tty
- lock
- stty
- script
- clear
- uname
- date
- cal
- calendar
- Bc
- echo
- cat

W





cal

- Command to see the calendar - `cal`
- `cal` with `-m` option display Monday as the first day of the week
- `cal` with `-3` option display prev/current/next month

```
$ cal June 2009
Su Mo Tu We Th Fr Sa
      1  2  3  4  5  6
    7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
```

```
[student@localhost ~]$ cal
September 2015
Su Mo Tu We Th Fr Sa
      1  2  3  4  5
    6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

```
[student@localhost ~]$ cal -3 -m 20 09 2015
August 2015     September 2015    October 2015
Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr
Sa Su
      1  2      1  2  3  4  5  6      1  2  3  4
    3  4  5  6  7  8  9  7  8  9 10 11 12 13  5  6  7  8  9 10 11
10 11 12 13 14 15 16 14 15 16 17 18 19 20 12 13 14 15 16 17  18
17 18 19 20 21 22 23 21 22 23 24 25 26 27 19 20 21 22 23 24  25
24 25 26 27 28 29 30  28 29 30      26 27 28 29 30 31
```



Features of Unix

- The Files and Process
- System Calls and Portability
- A multi-user system
- A multi-tasking system
- Building block approach (do one thing well)
- The unix tool kit
- Pattern Matching
- Programming facility
- Documentation

10



Features of Unix

- **The Files and Processes**

- Files have places and processes have life
- All files are related to each other by being a part of a single hierarchical structure, hence we can locate a file with reference to a pre-determined place
- Files can be moved from one place to another in the hierarchy
- Directories and Devices are also considered as Files only
- Process is the name given to file when it is under execution
- Process is the



Features of Unix

- **The Files and Processes**

- Processes have life they can be in the following state
 - Born
 - Dead /killed
 - Have parents
 - Have children and grand children
- Processes can be moved between foreground and background



Features of Unix

- **System calls and Portability**

- The unix system comprising of kernel, shell and applications are written in c
- All the commands in unix use a set of functions called as **system calls** to communicate with the kernel
- Unix uses write command to write to a file as well as to a device, where in the write command hides the differences in writing to a file or to a device
- All unix flavours use the same system calls (eg. Unix uses write command to write to a file or to a hardware devise which hides the innate complexities in writing to a file



Features of Unix

- **Portability**

- Once software has been written on any unix system, it can be easily ported to any other unix machine, its only the implementation of the system calls that needs to be reprogrammed and not the system calls themselves

L





Features of Unix

- **Multi-user System**

- Permits multiple programs to run simultaneously
- Multiple users can run separate jobs at the same time (Multi-User)
- A single user can run multiple jobs at the same time (Multi-tasking)
- Unlike windows, unix resources are shared between all the users, the system allocates time for all the users
- Switching happens so fast that users are not aware of it





Features of Unix

- **Multi-tasking System**

- Single user can run multiple jobs at the same time
- Like printing a file while editing another
- In this case one job is running in the foreground while others run in the background
- Users can switch jobs between foreground and background

Features of Unix



- **Building-block Approach**

- Do one thing well
- We can combine two or more commands by using pipe symbol (|)
- Eg. ls and wc
- The commands executed in this way are called as Filters, because they filter or manipulate data
- This is the reason architects of unix commands kept the output to a minimum and less interactive
- **Inter process communication** of the shell can arrange for a command's output to be passed on as input for another command

Features of Unix



- **Unix Tool Kit**
 - Unix has text manipulation utilities (Filters), compilers, interpreters, network applications(browsers), system administration tools, various shells ...



Features of Unix

- **Pattern Matching**

- Unix has various pattern matching characters called **meta characters** (* . +)
- For complex pattern matching unix uses **Regular Expression**, which is a special expression framed with characters from the meta-character set



Features of Unix

- **Documentation**

- Online help is provided by the man command
- Help command is also available
- News groups are available on internet

