Chapter 1: Introduction

Introduction
History of Operating System
- First Generation Operating System
- Second Generation Operating System
- Third Generation Operating System
- Fourth Generation Operating System

Operating System
Services
Types of OS
- Mainframe operating system
- Server Operating Systems
- Multiprocessor Operating Systems
- Personall Computer(PC) OS
- Mobile Operating Systems
- Real Time Operating Systems

Concepts
- Processes
- Files
- Address Spaces
- Input/Output
- Protection
- Shell
- System Call

Operating System Structures
- Monolithic Systems
- Layered Systems
- Microkernels
- Client-Server Model
- Virtual Machines
- Exokernel

Introduction

Computers, laptops and smartphones are now a days common in everybody's life. One important part of this devices which is not visible is Operating system. An Operating System, or "OS," is software that communicates with the hardware and allows other programs to run.

[Operating System Definition] "An Operating System is a program that provides *convenient interface* to user and manages computer hardaware *efficiently*."

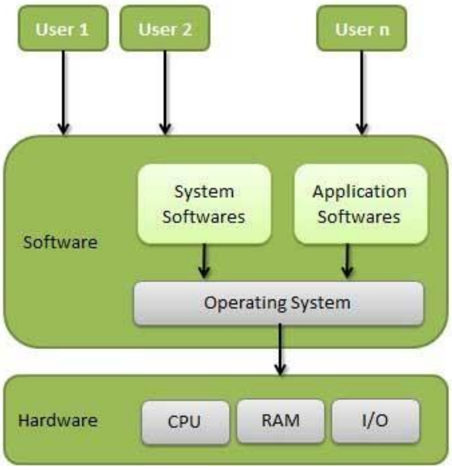


Figure 1.1: Operating System

Every general-purpose computer must have an operating system to run other programs. Operating systems perform basic tasks, such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk, and controlling peripheral devices such as disk drives and printers. Operating system provides a base on which other application programs can run. For large systems, the operating system has even greater

responsibilities and powers. The Operating System is also responsible for security, ensuring that unauthorized users do not access the system.

Every desktop computer, tablet, and smart phone includes an Operating System that provides basic functionality for the device. Common desktop operating systems include Windows, Mac OS X, and Linux. While each OS is different, they all provide a graphical user interface, or GUI, that includes a desktop and the ability to manage files and folders. They also allow you to install and run programs written for the operating system. While Windows and Linux can be installed on standard PC hardware, Mac OS X can only run on Macintosh computers. Therefore, the hardware you choose affects what operating system(s) you can run. Mobile devices, such as tablets and smartphones also include operating systems that provide a GUI and can run applications. Common mobile OS include Android, iOS, and Windows Phone. These OS are developed specifically for portable devices and therefore are designed around touchscreen input.

[1.] OS is like Brain, Brain controls all parts of the body:

1. What to think? (Use of Mind)
2. What to remember?
3. What to see,hear, smell and feel?
4. What to speak,write? . . .

In similar way OS controls computer H/W.

1. Use of Processor to Processing information
2. Use of Storage for storing Information (Storage RAM/ROM/HDD)
3. Use of Input devices for Taking Input from user (Input Devices)

4.Displaying or giving output to user (Output Devices) . . .

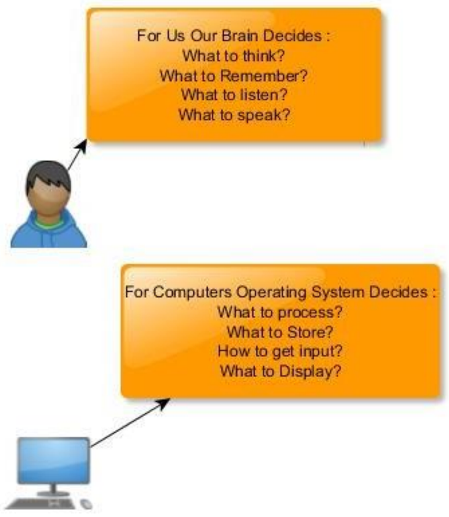


Figure 1.2: Operating System Analogy

History of Operating System

OS's are changing since the birth of computers. We can observe OS changes with change in H/W.

OS Class	Period	Objectives	Hardware
Bare Machine	First Generation	Program Execution	Vacuum Tubes
Batch Processing	Second Generation	Improve CPU Utilization	Transistors
Multiprograming/ TimeSharing	Third Generation	Resource Utilization, User Interaction	ICs
Network/Distributed/Realtime	Fourth Generation	Resource Sharing, Communication	LSI/VLSI

First Generation Operating System

In the 1940s, the earliest electronic digital systems had no operating systems. Those days Machines run from a console with display lights, toggle switches, input device, and printer. These machines consists of thousands of vacuum tubes. They were very big (placed in rooms). they were very slower than even the cheapest personal computers available today. There were no programming languages available. Typical mathematical and scientific calculation problems are solved with computers. Programs are written on sheets and implemented by wiring out(Hardwired). Programs are directly executed on computers without any system software support. This approach is called as *Bare Machine Approach*.

Second Generation Operating System

In 1950's use of transistors made computers better in speed, reliability and cost in comparison with vacume tube computers. Those computers were as mainframe computer systems. A program or set of programs which was called as a job is executed with help of Batch Monitor (Batch OS).

[Batch OS Definition] In Batch OS, major concern is to improve CPU Utilization by making ensure that processor is given some task at all time. Each set of jobs was considered as a batch.

For example there may be a batch of short Fortran jobs. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. The output for each job were available to users in a form of output tapes or papers. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. The operator then sorts programs into batches with similar requirements. The problems with Batch Systems are following:

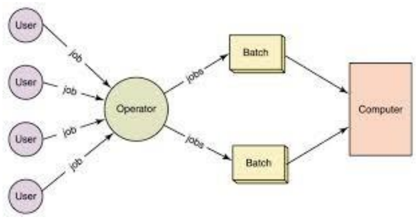


Figure 1.3: Batch Operating System

1.Lack of interaction between the user and job.
2.CPU is often idle, because the speeds of the mechanical I/O devices is slower than CPU.
3.Difficult to provide the desired priority.

Third Generation Operating System

While Second generation computers were consisting individual transistors, Third Generation Computers uses Integrated Circuits (ICs). Computers equipped with ICs were better in terms of price/performance ratio. OS/ 360 was developed by IBM introduced concept of

Multiprogramming.

Multiprogramming

Multiprogramming is very important property of operating systems. A single user or program executed by user cannot utilize CPU or the I/O devices all times. In fact program consist of portions where CPU is used (known as CPU Burst) or I/O devices are used (I/O Burst). Multiprogramming increases CPU utilization by Allocating CPU to another program, when program uses I/O and so on. CPU Scheduler is a part of OS responsible for allocating CPU to programs. When, the first program finishes waiting(I/O) and it will gets CPU back. Multiprogramming increases CPU Utilization by ensuring CPU is busy almost all times.

[Multiprogramming OS Definition] In multiprogramming system, when one program is waiting for I/O transfer, there is another program ready to utilize the CPU. It is possible to load several jobs in memory for concurrent execution.

Spooling

The technique of *Spooling(Simultaneous Peripheral Operation On Line)* is also evolved with Third-generation operating systems. Spooling allows Multiple programs to do peripheral operations concurrently. With the spooling multiple application can use I/O Devices without interfering each other. The operating system solves this problem by keeping Each application's output in separate disk file. When an application request for printing, the spooling system queues the corresponding spool file for output to the printer. The spooling system copies the queued spool files to the printer one at a time.

Benefits of Spooling:

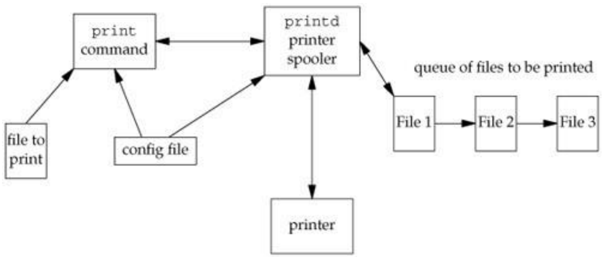


Figure 1.4: Spooling

1. Even in a simple system, the spooler may be reading the input of one job while printing

the output of a different job. During this time, still another job (or other jobs) may be executed, reading its "cards" from disk and "printing" its output lines onto the disk.

2. Spooling is also used for processing data at remote sites. The CPU sends the data via communication paths to a remote printer (or accepts an entire input job from a remote card reader). The remote processing is done at its own speed, with no CPU intervention.
3. Spooling can keep both the CPU and the I/O devices working at much higher rates.

Timesharing or Multitasking Systems

Multiprogrammed systems provide an environment in which the various system resources (for example, CPU, memory, and peripheral devices) are utilized effectively, but they do not provide user interaction with the computer system. *Time sharing* (or multitasking) is a logical

extension of multiprogramming. In time-sharing systems, the CPU executes multiple tasks concurrently by switching among them. Important difference is multitasking allows users to interact with each program while program is running. All modern operating System(for example Windows, LINUX, UNIX) uses multitasking. We all can experience multitasking on our PCs/ Laptops, where multiple task or applications concurrently running. The objective of multiprogramming is to improve CPU Utilization while timesharing focus on user response (interaction).

Time sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor’s time which is shared among multiple users simultaneously is termed as time-sharing.

Difference between Multiprogramming and Timesharing Operating System

The main difference between Multiprogrammed Batch Systems and Time-Sharing Systems is that in case of Multiprogrammed batch systems, objective is to maximize processor use, whereas in Time-Sharing Systems objective is to minimize response time. Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receives an immediate response. For example, in a transaction processing, processor execute each user program in a short burst or quantum of computation. That is if n users are present, each user can get time quantum. When the user submits the command, the response time is in few seconds at most. Operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time. Computer systems that were designed primarily as batch systems have been modified to time-sharing systems.

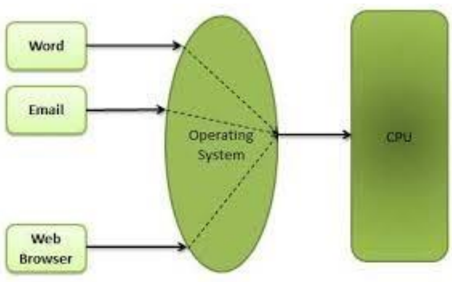


Figure 1.5: Time Sharing Operating System

Advantages of Timesharing operating systems are following:

1. Provide advantage of quick response.
2. Avoids duplication of software.
3. Reduces CPU idle time.

Disadvantages of Timesharing operating systems are following:

1. Problem of reliability.

2. Programs may interfere each other. Timesharing operating system needs mechanism for ensuring security and integrity of user programs and data.
3. Problem of data communication.

Fourth Generation Operating System

With the development of LSI (Large Scale Integration) circuits and VLSI, computers are available for more users. In fourth generation OS focuses on user interaction. Earlier OS like DOS,UNIX were simple, they provide CLI(Command Line Interface). CLI was suitable for sophisticated user but not enjoyed by naive users. GUI(Graphical User Interface) was developed to improve usability. Interaction with mouse was promoted by windows. An interesting development is the growth of networks of personal computers. Networking provides sharing of resources and communication. Operating systems which support networking were evolved as Network Operating Systems and Distributed Operating Systems .

Operating System Services

An operating system provides an environment for the execution of programs. It provides following services to programs and to the users of those programs.

1. User interface: User Interface (UI) take several forms. One is a command-line interface(CLI), which uses text commands. Another is a batch interface, in which commands stored into files, and those files are executed. Modern OS uses graphical user interface (GUI). Some systems provide two or all three of these variations.
2. Program execution: The system must be able to load a program into RAM and run that program.
3. I/O operations: A running program may read or write file or interact with I/O device. Examples are recording to a CD or DVD drive or editing a file. OS manages operation required to do I/O.
4. File Manipulation: The user generally store information in form of files. Each program need to create,delete,read and write files and directories. Program may search for a given file, and list file information. Finally, some programs include permissions management to allow or deny access to files or directories based on file ownership. OS needs to provide easy to use, reliable and consistent file system.
5. Communications: One program may interacts with another program. OS provides communication facility between programs whether they are executing on same computer or different computer systems connected by a computer network.
6. Error detection: The operating system needs to detect possible errors. Errors may occur in the CPU and memory hardware or in I/O devices and in the user program (such as an arithmetic overflow, an attempt to access an illegal memory location). Operating system should be able to detect errors and recover from those(exception handling).
7. Resource allocation: Multiple programs may request for same resources at simultaneously. OS allocates resources in a manner which resolves conflicts and provide better utilization and efficiency. Example of Computational Resources are processor, memory, files ,I/O devices. Accounting. OS keeps record of allocation of computer resources. This record keeping may be used for improving computing services.
8. Protection and security: Information stored may be used by attackers for bad purpose. OS needs to provides mechanism for protection and security. When several separate processes execute concurrently, it should not be possible for one process to interfere with the others or with the operating system itself. Protection involves ensuring that all access to system resources is controlled. Security ensures protection from outsiders. One example of security mechanism is authentication with a password, to gain access to system resources.

Types of Operating System

Operating systems keep evolving over the period of time. Following are few of the important types of operating system which are most commonly used.

Mainframe operating system

Mainframe Computers used by corporate and governmental organizations for critical applications,web servers, bulk data processing such as census, industry and consumer statistics, enterprise resource planning and transaction processing. Mainframe Computers may have more than 1000 disks and Terabytes of storage.

Mainframes are designed to handle very high volume input and output (I/O) and emphasize throughput (Number of jobs completed) computing. Mainframe OS offers three kinds of services:

1. Batch : Batch processing focuses on CPU Utilization, no interaction with jobs are required here.
2. Transaction Processing: It handles large number of small request. Example of transaction processing is check processing at a bank or airline reservation.
3. Timesharing : In Timesharing CPU switches between programs to provide interaction.

Server Operating Systems

Server Operating Systems run on servers, Which are very large personal computers or mainframes.Server Operating Systems works in client server networking environment.The serves large number of users connected with networks.Server operating systems can act as Web server, Mail server, File server, Database server, application server and Print server.

Examples of Server Operating Systems are Windows Server,UNIX Server,Sun Solaris, Mac OS X Server, Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server. Server OS

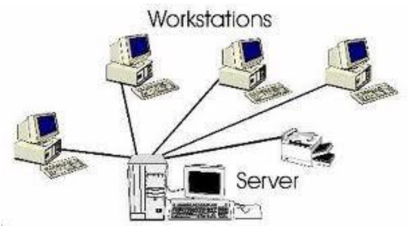


Figure 1.6: Server Operating System

provides following features:

1. GUI is not important or optional.
2. It is possible to reconfigure and update both hardware and software to some extent without restart.
3. It provides Advanced backup facilities to permit regular and frequent online backups of critical data.
4. It provides flexible and advanced networking capabilities,
5. It provides more Protection and Security features than PC Operating System.

Multiprocessor Operating Systems

Multiprocessor Operating System have two or more central processing units (CPU),sharing bus, memory and other peripheral devices. These systems are referred as tightly coupled systems.

Multiprocessor OS A computer system in which two or more CPUs share full access to Main Memory.

Benefits of Multiprocessor OS

1. Increase throughput: Multiprocessor OS uses multiple processor to complete more task per unit time. Although speed up ratio with N processors is not N but less than N.

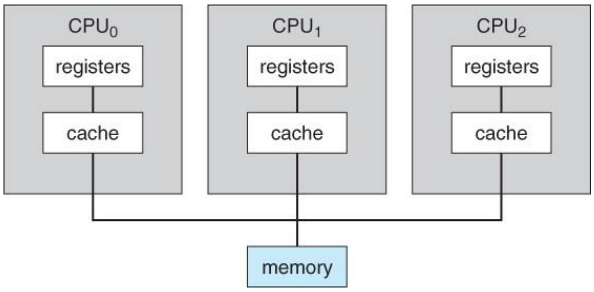


Figure 1.7: Multiprocessor Operating System

2. Economy of scale:Multiprocessor systems can save more money than multiple single processor system, because they can share peripherals, mass storage and power supplies.
3. Increased reliability: They have *Graceful Degradation* property. It means failure of single processor will reduce speed of operation, but not stop working of entire system.

Personnel Computer(PC) Operating Systems

Personnel Computers are becoming powerful and inexpensive. The Operating System for PC's emphasizes user interaction and ease of use. Modern PC Operating Systems provides multitasking and easy to use GUI. Example of PC Operating Systems are Windows(XP, Vista, 7, 8 ..), Macintosh, Linux(Fedora, Ubuntu ...).

Mobile Operating Systems

We are all aware about mobiles, PDA's(Personnel Digital Assistant),Tablets. These small devices also runs OS, which is referred as *Mobile Operating Systems*. Developers of mobile systems and applications face many challenges, most of which are due to the limited size of such devices. For example, a PDA is typically about 5 inches in height and 3 inches in width. Because of their size, most mobile devices have a small amount of memory, slow processors, and small display screens. The amount of physical memory in a handheld depends upon the device. As a result, the operating system and applications must manage memory efficiently. This includes returning all allocated memory back to the memory. Second important limitation is limited battery power. Mobile OS needs to use battery power effectively. Modern Mobile Operating Systems combine the features of a PC Operating System with other features, including a touchscreen, cellular, Bluetooth, Wi-Fi, GPS mobile navigation, camera, video camera, speech recognition, voice recorder and music player. Example of Operating Systems for mobile devices (smart phones and tablets) are Apple's iOS and Google's Android.

Figure 1.8: Mobile Operating System

Real Time Operating Systems

Real-time systems are those that are used to offer services like ATM (Automatic teller machine) or airlines reservation systems. This systems are often called transaction oriented systems. This is because the basic information processing involves a transaction. As an example, consider processing a request like withdrawing money. It is a financial transaction, which requires updating accounts within an acceptable response time. Both the process control systems and transaction oriented systems are real time systems. In this systems response time to a request is critical. The main objective is to meet deadlines. Embedded systems run real-time operating systems. A real-time system is used in control device in a dedicated application. Real time systems can be used in device like washing machines, microwaves and smart TV’s . Systems that control scientific experiments, medical imaging systems, industrial control systems, and certain display systems are realtime systems. Some automobile-engine fuel-injection systems, home-appliance controllers, and weapon systems are also real-time systems. Real time Systems can be classified as:

Hard Real time Systems

Hard real-time means you must absolutely hit every deadline. Very few systems have this requirement. Some examples are nuclear systems, some medical applications such as pacemakers, a large number of defense applications, avionics, etc.

Soft Real time Systems

Some times hard real time systems may not provide interaction while executing jobs. Soft real time systems can miss some deadlines but provide better interactiveness. We may observe performance degradation if too many deadlines are missed. A good example is the sound system in your computer. If you miss a few bits, it will not affect.

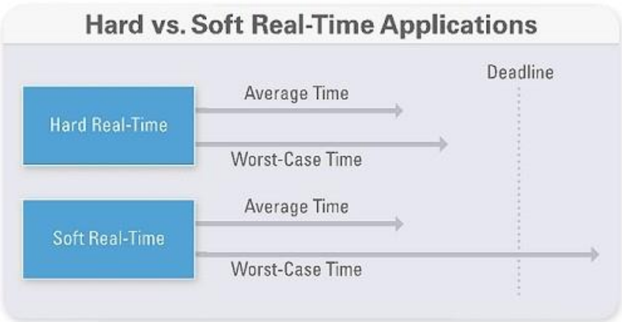


Figure 1.9: Real Time Operating System

Concepts of OS

Processes

Program consist of sequence of instruction. Program is stored in form of files in Secondary storage. A process is the actual execution of those instructions. Several processes may be associated with the same program; for example, opening up several instances of the same program often means more than one process is being executed.

Process "A process is a program in execution."

Process consists of following resources:

1. Executable machine code of program.
2. Data Area to keep variables.
3. Stack to keep track of active subroutines and/or other events).
4. Heap to hold intermediate computation data generated during run time.
5. Operating system descriptors of resources that are allocated to the process, such as file descriptors (Unix terminology) or handles (Windows),and data sources.
6. Security attributes, such as the process owner and the process’ set of permissions (allowable operations).

7. Processor state (context), such as the content of registers, physical memory addressing, etc. The state is typically stored in computer registers when the process is executing, and in memory otherwise.

The operating system holds most of this information about active processes in data structures called process control blocks.

Files

A collection of data or information is stored in form of files. There are many different types of files: data files, text files , program files, directory files, and so on. Different types of files store different types of information. For example, program files store programs, whereas text files store text. On most modern operating systems, files are organized into one-dimensional arrays of bytes.

Most operating systems have the concept of a directory. We can think directory as container that holds group of files together. A student, for example, might have one directory for each course he or she is taking (for the programs needed for that course), another directory for his electronic mail, and still another directory for his World Wide Web home page. Directory entries may be either files or other directories.Normally directories are arranged as tree form, such arrangement is called Hierarchical File System .

Address Spaces

All executing programs are stored in main memory. In a very simple operating system, only one program at a time is in memory. To run a second program, the first one has to be removed and the second one placed in memory. Multiprogramming operating systems allow multiple programs to be stored in memory at the same time. Operating system uses protection mechanism to keep them from interfering with one another A different, but equally important memory-related issue, is managing the address space of the processes. Normally, each process has some set of addresses it can use, typically running from 0 up to some maximum.

Input/Output

Input/output or I/O is the communication between a computer and user. Inputs are the signals or data received by the system and outputs are the signals or data sent from it. I/O devices are used by a human (or other system) to communicate with a computer. For example, a keyboard or mouse is an input device for a computer, while monitors and printers are output devices. Some devices for communication between computers, such as modems and network cards, perform both input and output operations. Any transfer of information to or from the CPU/memory, for example by reading data from a disk drive, is considered I/O.

Protection

Some information stored in computer is confidential. This information may include e-mail, business plans, tax returns, and much more. It is up to the operating system to manage the system security so that files are only accessible to authorized users. If a computer program is run by unauthorized user then he/she may cause severe damage to computer or data stored in it. So a computer system must be protected against unauthorized access, malicious access to system memory, viruses, worms etc. Some operating system uses dual mode operation as protection mechanism.

Dual Mode Operations In order to ensure the proper execution of the OS, Operating system distinguish between the execution of operating-system code and user-defined code. OS uses two separate modes of operation: user mode and kernel mode (also called supervisor mode, system mode, or privileged mode). A bit, called the mode bit is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1). In User mode, a subset of instructions is available. We can execute limited set of instructions.

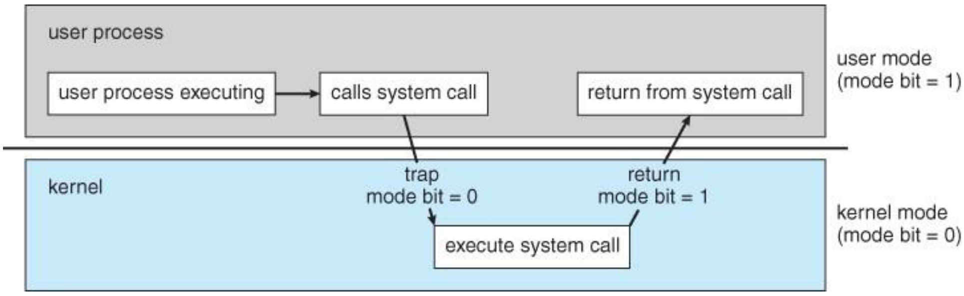


Figure 1.10: Dual Mode of Operation

Benefits

1. *I/O protection*: all I/O operations are privileged; so user programs can only access I/O by sending a request to the (controlling) OS.
2. *Memory protection*: It uses base/limit registers (in early systems), memory management unit, (MMU, in modern systems) for protection. User programs can only access the memory that the OS has allocated.

For example CPU control, timer (alarm clock), context switch. User programs can only read the time of day,It can not chang time. Uer program can only have as much CPU time as the OS allocates. When a user application requests a service from the OS (via a system call), it must request os to change mode from user to kernel mode to fulfil the request.

The Shell

In UNIX operating system consists of two major parts: kernel and shell. Kernel performs low level tasks like process management,memory management and I/O management. UNIX shell provides interface through which user gives commands. This commands are interpreted by shell. It makes heavy use of many operating system features. Many shells exist, including sh, csh, ksh, and bash. All of them support the functionality described below, which derives from the original shell (sh).

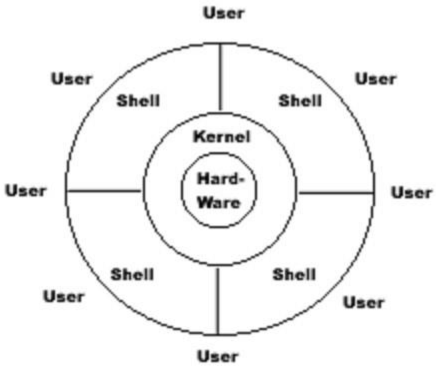


Figure 1.11: Shell

System Calls

Most user processes require a system call to seek OS services.

Systems call provide a library or API that sits between normal programs and the operating system. a system call is how a program requests a service from an operating system's kernel.

Program uses hardware-related services (for example, accessing a hard disk drive), creation and execution of new processes, and communication with integral kernel services such as process scheduling. System calls provide an essential interface between a process and the operating system. Example of System calls:

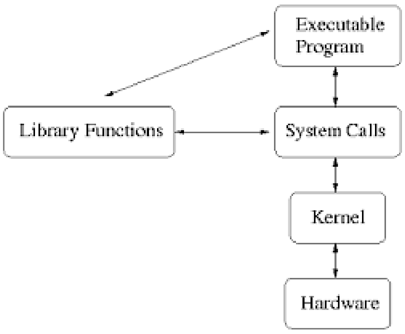


Figure 1.12: System Calls

1. To create or terminate processes.
2. To access or allocate memory.
3. To get or set process attributes.
4. To create, open, read, write files.
5. To change access rights on files.
6. To mount or un-mount devices in a file system.
7. To make network connections.
8. Set parameters for the network connection.
9. Open or close ports of communication.
10. To create and manage buffers for device or network communication.

Operating System Structures

Operating system is a very large and complex software. Operating system should be stable, flexible, robust, efficient and easy to use. To meet this requirements OS needs to be designed with software engineering principles. One of the best practice to reduce complexity is to divide large OS into small components. Each of small components are called modules. Each module should be a well-defined portion of the system, with carefully defined inputs, outputs, and functions.

Monolithic Systems

Many operating systems do not have well-defined structures. In this approach the entire operating system runs as a single program. If no systematic approach of system development is used than system is said to have monolithic structure. Small systems generally developed in same way. MS-DOS is an example of such a system. It was originally designed and implemented by a few people who had no idea that it would become so popular.

In MS-DOS, the interfaces and levels of functionality are not well separated. For instance, application programs are able to access the basic I/O routines to write directly to the display and disk drives. Such freedom leaves MS-DOS vulnerable to errant (or malicious) programs, causing entire system crashes when user programs fail. Of course, MS-DOS was also limited by the hardware of its era. Because the Intel 8088 for which it was written provides no dual mode and no hardware protection, the designers of MS-DOS had no choice but to leave the base hardware accessible.

Benefits:

1. Monolithic kernels are faster.
2. Monolithic kernels are simple to implement.

Disadvantages:

1. Monolithic systems are not flexible.Addition/removal is not possible
2. Monolithic systems are difficult to debug.
3. Since the device driver reside in the kernel space it make monolithic kernel less secure.

Layered Systems

Layering is software engineering guideline for handling large and complex systems. In Layered model, We can break operating systems into smaller pieces or components. The related components are kept in particular layer. The bottom layer is the hardware, while the highest layer is the user interface.Each layer can be changed without disturbing others.

The layers are selected so that each uses functions (operations) and services of only lower-level layers. This approach simplifies debugging and system verification. The first layer can be debugged without any concern for the rest of the system, because, by definition, it uses only the basic hardware (which is assumed correct) to implement its functions. Example of layered model is the THE system. Which is built by E. W. Dijkstra (1968) and his students.

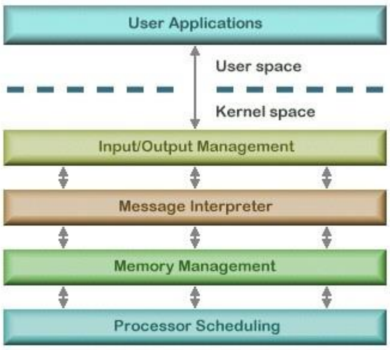


Figure 1.13: Layered Operating System

Benefits:

1. The layered operating system are flexible, modular and easy to build.
2. Each layer is independent of others that is the reason each can be developed independently. Overall development time can be reduced by dividing layers to various developer teams.

Disadvantages:

1. It requires an appropriate definition of the various layers.
2. Careful planning of the proper placement of the layer is necessary.
3. Layered OS are generally slower than monolithic and microkernel OS.

Microkernels

Microkernel OS structures the operating system by removing all nonessential components from the kernel. Only the very important parts like IPC(Inter process Communication), basic scheduler, basic memory handling and basic I/O primitives are put into the kernel. Others nonessential portions of the kernel are maintained as server processes in User Space. Communication between components of the OS is provided by message passing. Microkernel is a smaller kernel. The main function of the microkernel is to provide process management and communication facility between the client program. The other services runs in user space. Microkernel approach provides flexibility,modularity, security and reliability

Benefits:

1. Extending the operating system becomes much easier.

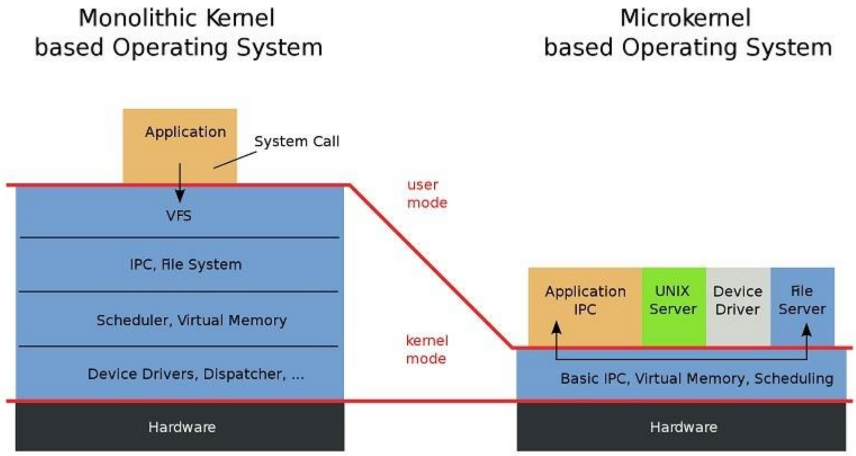


Figure 1.14: Microkernel Operating System

2. Since the kernel is smaller, we do not need to change kernel frequently.
3. The microkernel also provides more security and reliability.

Disadvantages:

1. Slower Processing due to additional Message Passing

Examples: Windows NT and MINIX are examples of microkernel OS.

Client-Server Model

With the availablity of inexpensive PCs, Terminals connected to centralized systems have been replaced. PCs at client end have more processing power than terminals. In client server architecture clients PC request server. Server systems handles requests generated by client systems. Server provides some services to clients. Client can get services by requesting for service. Server respond by doing some processing and returning some result. Examples of computer applications that use the client–server model are Email, network printing, and the World Wide Web. Servers are classified by the services they provide. For instance, a web server serves web pages and a file server serves computer files. A shared resource may be any of the server computer's software and electronic components, from programs and data to processors and storage devices. The sharing of resources of a server constitute a service. Generally clients and servers runs on different computers, connected by a local or wide-area network. But they may run on same machines also. For example, a single computer can run web server and file server software at the same time to serve different data to clients making different kinds of requests. Client software can also communicate with server software within the same computer.

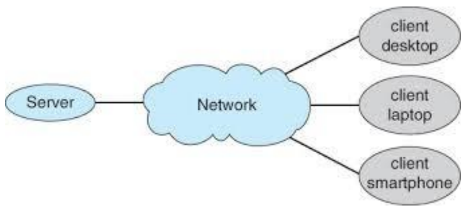


Figure 1.15: Client Server Operating System

Benefits:

1. Client server os provides Enhanced Data Sharing.
2. We can share Resources among different Platforms(different OS).
3. Administrator can manage overall system from any place.

Disadvantages:

1. High resource requirement at server end.
2. It requires more sophisticated softwares than PC operating system

Virtual Machines

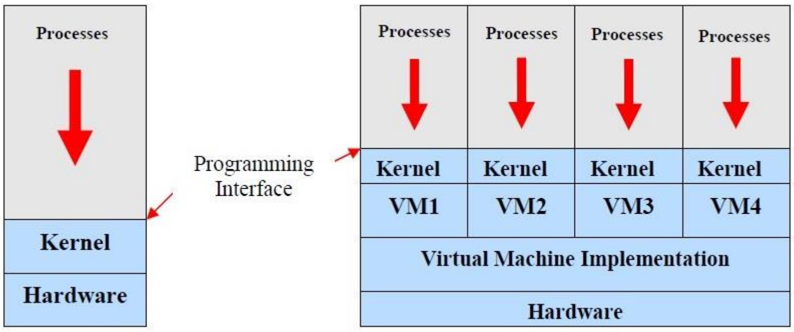
Virtual machine creates the illusion that each separate execution environment(Operating systems) is running its own private computer. A virtual machine is a software computer that, like a physical computer, runs an operating system and applications. A virtual machine (VM) is an operating system OS or application environment that is installed on software which imitates dedicated hardware. The end user has the same experience on a virtual machine as they would have on dedicated hardware. Specialized software called a hypervisor emulates the PC client or server's CPU, memory, hard disk, network and other hardware resources completely, enabling virtual machines to share the resources. The hypervisor can emulate multiple virtual hardware platforms that are isolated from each other, allowing virtual machines to run Linux and Windows server operating systems on the same underlying physical host.

Benefits:

1. Virtual-machine system can be used as testing platform for new operating-systems.Operating systems are large and complex programs. It is difficult to modify OS. Because the operating system executes in kernel mode,a wrong change could cause an error that would destroy the entire system. Thus, it is necessary to test all changes to the operating system carefully. Bug in OS will not affect actual hardware.
2. Multiple OS can be used on same computer.
3. Easy maintenance, availability and convenient recovery.

Disadvantages:

1. In Virtual Machine environment, Direct sharing of resources is not possible in some cases.
2. Virtual machine is not that efficient as a real one when accessing the hardware.



System models. (1) Non-virtual machine. (2) Virtual machine.

Figure 1.16: Virtual Machine

Example: VMware VMware is a popular virtual machines software. VMware runs as an application on a host operating system such as Windows or Linux. It allows this host system will run concurrently several different guest operating systems. Each guest OS will run independent on their virtual machines. Consider the following scenario: A developer has designed an application and would like to test it on Linux, FreeBSD, Windows NT, and Windows XP. One option is for her to obtain four different computers, each running a copy of one of these operating systems. Another alternative is for her first to install Linux on a computer system and test the application, then to install FreeBSD and test the application, and so on. This option allows her to use the same physical computer but is time-consuming, since she must install a new operating system for each test.

Virtual Machine softwares like VMware allows multiple operating system running concurrently Such testing could be accomplished concurrently on the same physical computer.

Exokernels

Exokernel is an operating system kernel developed by the MIT Parallel and Distributed Operating Systems group. Generally Kernel handles low-level responsibilities of controlling hardware (particularly memory allocation). In exokernel application programmer can directly interact with hardware. Most of the developers prefer that OS takes responsibility of this low-level tasks, because they want to focus on writing applications.In some cases developer wants to to control hardware directly without interacting with OS. An exokernel just allocates physical hardware resources to programs. This allows the program to use library operating systems.Exokernel architectures may have great performance benefits, If program is correctly written. But in some case of bugs, it may lead to crash. Exokernels are very small, since functionality is limited to ensuring protection and multiplexing of resources.

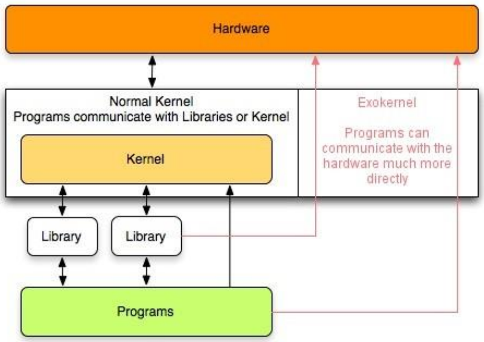


Figure 1.17: Exokernel

Benefits:

1. We can build higher performance applications by giving them flexible, extensible access to OS primitives.

Disadvantages:

1. Exokernel technology is still not thoroughly researched and tested.
2. Different applications to run simultaneously on the same system, that would also mean different look and feels for each of them.