CSC259: Operating System

Prepared By:
Ravi Maharjan
Samriddhi College
BSc CSIT 4th Semester

CSC259: Operating System

Unit 1: Operating System Overview

1.1. Definition, Two views of operating system, Evolution of operating system, Types of OS.

1.2. System Call, Handling System Calls, System Programs, Operating System Structures, The Shell, Open Source Operating Systems

CSC259: Operating System

What is Operating System?

- An Operating System (OS) is an interface between a computer user and computer hardware.
- An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
- An operating system is software that enables applications to interact with a computer's hardware. The software that contains the core components of the operating system is called the kernel.
- The primary purposes of an Operating System are to enable applications (software) to interact with a computer's hardware and to manage a system's hardware and software resources.

3

CSC259: Operating System

Some popular Operating Systems include Linux Operating System, Windows Operating System, Open Virtual Memory System (VMS), OS/400, Advanced Interactive Executive (AIX) version of unix OS , MacOS, etc. Today, Operating systems is found almost in every device like mobile phones, personal computers, mainframe computers, automobiles, TV, Toys etc.

An Operating System is the low-level software that supports a computer's basic functions, such as scheduling tasks and controlling peripherals.

4

CSC259: Operating System

Roles of OS:

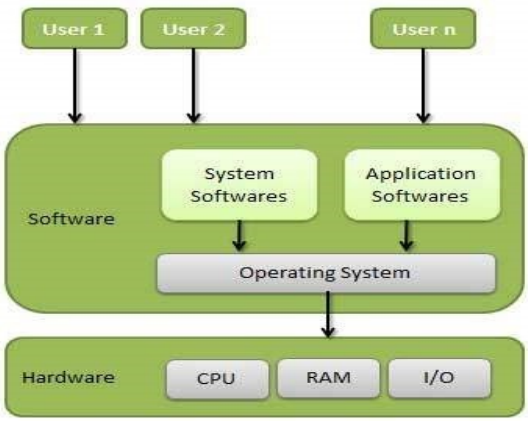
- Implementing the user interface
- Sharing hardware among users
- Allowing users to share data among themselves
- Preventing users from interfering with one another
- Scheduling resources among users
- Facilitating input/output
- Recovering from errors
- Accounting for resource usage
- Facilitating parallel operations
- Organizing data for secure and rapid access
- Handling network communications

5

CSC259: Operating System

Architecture

We can draw a generic architecture diagram of an Operating System which is as follows:



6

CSC259: Operating System

Two views of Operating system

1. **OS as an extended machine**

Today's computer comes with multiple components; users are not interested in how these components work together to perform tasks. All we need is to complete our task with no complexity and overhead.

Thus, we need is that program should hide the truth about hardware from the programmer.

This is very primitive task of OS. Users are not interested whether the file is written in CD or hard drive, not about how is is stored?

In this view, the function of OS is to present the user with equivalent of extended machine or virtual machine that is easier to program than underlying the hardware.

7

CSC259: Operating System

Two views of Operating system

1. **OS as an extended machine**

•At the Machine level the structure of a computer's system is complicated to program, mainly for input or output. Programmers do not deal with hardware. They will always mainly focus on implementing software. Therefore, a level of abstraction is supposed to be maintained.

•Operating systems provide a layer of abstraction for using disk such as files.

•This **level of abstraction** allows a program to create, write, and read files, without dealing with the details of how the hardware actually works.

•The level of abstraction is the key to managing the complexity.

•Good abstractions turn an impossible task into two manageable tasks.

•The first is to define and implement the abstractions.

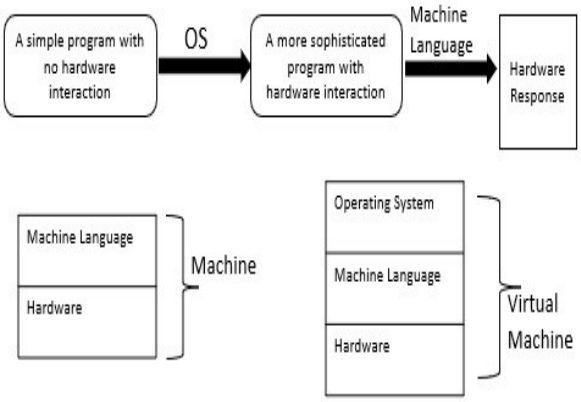
•The second is to solve the problem at hand.

8

CSC259: Operating System

For example − It is easier to deal with photos, emails, songs, and Web pages than with the details of these files on disks.

The diagram given below shows the functioning of OS as an extended machine −



9

CSC259: Operating System

2. **OS as a Resource Manager**

- What happens if three programs try to print their output on the same printer at the same time
- What happens if two network users try to update a shared document at the same time?
- OS primary function is to keep track of who is using which resource to grant resource requests, to account for usage and to mediate conflicting requests from different programs and users
- Virtualizes resources so multiple users can share it
- Protect application from one another
- Provide efficient and fair access to resource

10

CSC259: Operating System

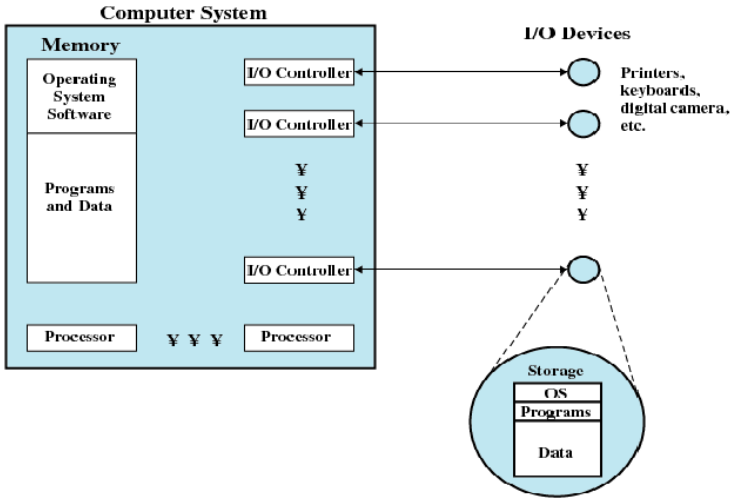


Figure: The operating system as Resource Manager

11

CSC259: Operating System

Evolution of Operating System

Operating systems have evolved through a number of distinct phases or generations which corresponds roughly to the decades.

The 1940's - First Generations (User Driven)

- The earliest electronic digital computers had no operating systems.
- Machines of the time were so primitive that programs were often entered one bit at time on rows of mechanical switches (plug boards).
- Programming languages were unknown (not even assembly languages).
- Operating systems were unheard of.

12

CSC259: Operating System

The 1950's - Second Generation (Batch Processing)

• By the early 1950's, the routine had improved somewhat with the introduction of punch cards.

• The General Motors Research Laboratories implemented the first operating systems in early 1950's for their IBM 701.

• The system of the 50's generally ran one job at a time. These were called single-stream batch processing systems because programs and data were submitted in groups or batches.

13

CSC259: Operating System

The 1960's - Third Generation (Multiprogramming)

• The systems of the 1960's were also batch processing systems, but they were able to take better advantage of the computer's resources by running several jobs at once.

• So operating systems designers developed the concept of multiprogramming in which several jobs are in main memory at once; a processor is switched from job to job as needed to keep several jobs advancing while keeping the peripheral devices in use.

• For example, on the system with no multiprogramming, when the current job paused to wait for other I/O operation to complete, the CPU simply sat idle until the I/O finished. The solution for this problem that evolved was to partition memory into several pieces, with a different job in each partition. While one job was waiting for I/O to complete, another job could be using the CPU.

14

CSC259: Operating System

• Another major feature in third-generation operating system was the technique called spooling (simultaneous peripheral operations on line). In spooling, a high-speed device like a disk interposed between a running program and a low-speed device involved with the program in input/output.

Instead of writing directly to a printer, for example, outputs are written to the disk. Programs can run to completion faster, and other programs can be initiated sooner when the printer becomes available, the outputs may be printed.

• Another feature present in this generation was time-sharing technique, a variant of multiprogramming technique, in which each user has an on-line (i.e., directly connected) terminal. Because the user is present and interacting with the computer, the computer system must respond quickly to user requests, otherwise user productivity could suffer. Timesharing systems were developed to multiprogram large number of simultaneous interactive users.

15

CSC259: Operating System

Fourth Generation (Client Server/Distributed Systems)

• With the development of LSI (Large Scale Integration) circuits, chips, operating system entered in the system entered in the personal computer and the workstation age.

• Microprocessor technology evolved to the point that it becomes possible to build desktop computers as powerful as the mainframes of the 1970s.

• Two operating systems have dominated the personal computer scene: MS-DOS, written by Microsoft, Inc. for the IBM PC and other machines using the Intel 8088 CPU and its successors, and UNIX, which is dominant on the large personal computers using the Motorola 6899 CPU family.

16

CSC259: Operating System

Types of OS:

1. **Batch Operating System**
2. **Multiprogramming Operating System**
3. **Multiprocessing Operating System**
4. **Time Sharing Operating System**
5. **Real Time Operating System**
6. **Distributed Operating System**
7. **Network Operating System**
8. **Personal Computer operating System**

17

CSC259: Operating System

1. **Batch Operating System**
   - Jobs with similar needs are grouped into batches and processed sequentially.
   - The OS automatically loads and runs each job in the batch.
   - Users do not interact with the computer system directly while their programs are running.
   - Reduced CPU idle time by automatically loading and executing jobs one after another.
   - Batch systems are not suitable for tasks requiring immediate response (e.g., user interaction, real-time control).

The problems with Batch Systems are as follows −

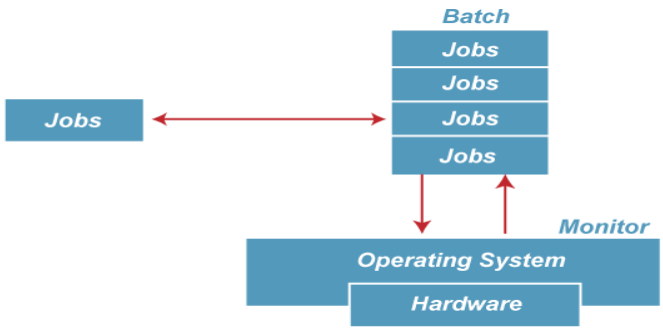
- Lack of interaction between the user and the job.
- CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.
- Difficult to provide the desired priority.

18

CSC259: Operating System

For example:

There are five jobs J1, J2, J3, J4, and J5, present in the batch. If the execution time of J1 is very high, then the other four jobs will never be executed, or they will have to wait for a very long time. Hence the other processes get starved.



19

CSC259: Operating System

Advantages:

•Efficient handling of large amounts of repetitive tasks.
•Good utilization of CPU and other resources during the batch processing era.
•Reduced setup time compared to running each job manually.

Disadvantages:

•No immediate feedback to users.
•Debugging is difficult (errors are found only after the job is complete).
•Poor interaction and flexibility.

Examples of Batch Processing Today:

Although modern OSs are interactive, batch processing is still used in:
•Payroll systems
•Bank transaction processing
•Scheduled backups and system updates

20

CSC259: Operating System

2. Multiprogramming OS:

A **Multiprogramming Operating System** is an OS that allows **multiple programs to reside in memory at the same time** and share the CPU. The OS selects one of the ready-to-run programs and allocates the CPU to it. When that program waits (e.g., for I/O), the CPU is switched to another ready program.

- Multiple programs are kept in memory and executed simultaneously (logically).
- The CPU switches rapidly between programs to maximize utilization.
- CPU is not left idle when a program waits for I/O.
- Other programs can run while one is waiting for input/output.
- The OS must manage memory so that multiple programs can be loaded at once without interfering with each other

21

CSC259: Operating System

Advantages:

•**Improved CPU Utilization** – CPU stays busy by executing another job when one is waiting.

•**Increased Throughput** – More jobs are completed in a given time.

•**Reduced Idle Time** – Minimizes idle time for both CPU and I/O devices.

Disadvantages:

•**Complex Memory Management** – Requires sophisticated memory handling techniques.

•**Difficult to Debug** – Concurrent execution can make program behavior hard to trace.

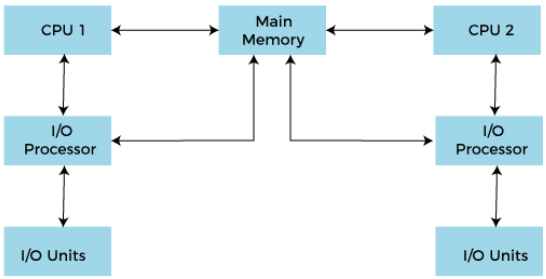
•**Risk of Starvation** – Some jobs may be delayed indefinitely if not scheduled properly.

22

CSC259: Operating System

3. Multiprocessing OS:

In Multiprocessing, Parallel computing is achieved. There are more than one processors present in the system which can execute more than one process at the same time. This will increase the throughput of the system.



Working of Multiprocessor System

CSC259: Operating System

In Multiprocessing, Parallel computing is achieved. More than one processor present in the system can execute more than one process simultaneously, which will increase the throughput of the system.

Advantages of Multiprocessing operating system:

- **Increased reliability:** Due to the multiprocessing system, processing tasks can be distributed among several processors. This increases reliability as if one processor fails, the task can be given to another processor for completion.
- **Increased throughout:** As several processors increase, more work can be done in less.

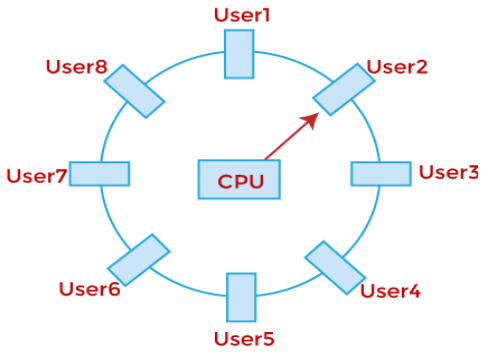
Disadvantages of Multiprocessing operating System

Multiprocessing operating system is more complex and sophisticated as it takes care of multiple CPUs simultaneously.

CSC259: Operating System

4. Time Sharing OS:

In the Time-Sharing operating system, computer resources are allocated in a time-dependent fashion to several programs simultaneously. Thus, it helps to provide a large number of user's direct access to the main computer. It is a logical extension of multiprogramming. In time-sharing, the CPU is switched among multiple programs given by different users on a scheduled basis.



Timesharing in case of 8 users

25

CSC259: Operating System

A time-sharing operating system allows many users to be served simultaneously, so sophisticated CPU scheduling schemes and Input/output management are required.

Time-sharing operating systems are very difficult and expensive to build.

Advantages of Time Sharing Operating System

•The time-sharing operating system provides effective utilization and sharing of resources.

•This system reduces CPU idle and response time.

Disadvantages of Time Sharing Operating System

•Data transmission rates are very high in comparison to other methods.

•Security and integrity of user programs loaded in memory and data need to be maintained as many users access the system at the same time.

26

CSC259: Operating System

5. Real Time Operating System:

In Real-Time Systems, each job carries a certain deadline within which the job is supposed to be completed, otherwise, the huge loss will be there, or even if the result is produced, it will be completely useless.

The Application of a Real-Time system exists in the case of military applications, if you want to drop a missile, then the missile is supposed to be dropped with a certain precision.

Advantages of Real-time operating system:

•Easy to layout, develop and execute real-time applications under the real-time operating system.

•In a Real-time operating system, the maximum utilization of devices and systems.

Disadvantages of Real-time operating system:

•Real-time operating systems are very costly to develop.

•Real-time operating systems are very complex and can consume critical CPU cycles.

27

CSC259: Operating System

Hard Real Time System:

- Well defined fixed time constraints, processing must be done within the defined constraints, or the system will fail. (e.g. use of brake in car)

Soft Real Time System:

- Less restrictive t timing constraints and does not support the dead line scheduling
- The value of job degrades as the dead line passes (e.g. late attendance in exam hall)

28

CSC259: Operating System

6. Distributed Operating System:

Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among the processors accordingly.

The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These are referred as **loosely coupled systems** or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, computers, and so on.

The advantages of distributed systems are as follows −

- With resource sharing facility, a user at one site may be able to use the resources available at another.
- If one site fails in a distributed system, the remaining sites can potentially continue operating.
- Better service to the customers.
- Reduction of the load on the host computer.
- Reduction of delays in data processing.

Disadvantages:

- Protocol overhead can dominate computation cost.

29

CSC259: Operating System

7**. Personal Computer Operating System**

A **personal computer operating system (PC OS)** is a software platform that manages the hardware and software resources of a personal computer. It acts as an intermediary between the user and the computer hardware, enabling users to run applications and perform tasks such as browsing the internet, writing documents, and playing media. Windows, Linux, MacOS are the examples.

Key Functions of a PC Operating System:

1. Provides a way for users to interact with the computer, usually through a **graphical user interface (GUI)** (e.g., Windows, macOS) or a **command-line interface (CLI)** (e.g., Linux terminal).
2. Manages the computer's **CPU**, **memory (RAM)**, **storage devices**, and **peripherals** (like printers and keyboards).
3. Organizes, stores, retrieves, and manages access to data on storage drives using file systems (e.g., NTFS, FAT32, ext4).
4. Handles **multitasking**—running multiple applications at the same time—by scheduling CPU time and managing memory allocation.
5. Protects data and system integrity through **user authentication**, **permissions**, and **firewall/antivirus support**.

30

CSC259: Operating System

8. **Handheld Operating System**:

A **handheld operating system** is a type of operating system designed specifically to run on **portable devices** such as smartphones, tablets, PDAs (Personal Digital Assistants), and other compact, mobile computing devices. Android, IOS, BlackBerry, MeeGo, Palm OS are the examples.

Key Features:

1.Lightweight and Efficient:

1. Designed to use minimal system resources (CPU, memory, storage) due to hardware limitations of handheld devices.

2.Touchscreen Interface:

1. Optimized for **touch input**, gestures, and small screen sizes.

3.Battery Management:

1. Includes **power-saving features** to extend battery life, such as sleep mode, background app control, and screen brightness adjustment.

4.Wireless Connectivity:

1. Supports wireless standards like **Wi-Fi**, **Bluetooth**, **cellular data (3G/4G/5G)**, and **GPS**.

31

CSC259: Operating System

1.2 System Call in OS:

- A system call is a way for a user program to interface with the operating system.
- The program requests several services, and the OS responds by invoking a series of system calls to satisfy the request.
- A system call can be written in assembly language or a high-level language like **C** or **Pascal**.
- System calls are predefined functions that the operating system may directly invoke if a high-level language is used.

A system call is a method for a computer program to request a service from the kernel of the **operating system** on which it is running. A system call is a request from computer software to an operating system's kernel.

The **Application Program Interface (API)** connects the operating system's functions to user programs. It acts as a link between the operating system and a process, allowing user-level programs to request operating system services. The kernel system can only be accessed using system calls. System calls are required for any programs that use resources.

32

CSC259: Operating System

How are system calls made?

When a computer software needs to access the operating system's kernel, it makes a system call.

The system call uses an API to expose the operating system's services to user programs. It is the only method to access the kernel system.

When the process is being executed and if it requires any resource, the process will create a system call (interrupt) and sends to the kernel.

33

CSC259: Operating System

Why do you need system calls in Operating System?

There are various situations where you must require system calls in the operating system. Following of the situations are as follows:

1.It must require when a file system wants to create or delete a file.

2.Network connections require the system calls to sending and receiving data packets.

3.If you want to read or write a file, you need to system calls.

4.If you want to access hardware devices, including a printer, scanner, you need a system call.

5.System calls are used to create and manage new processes.

34

CSC259: Operating System

Handling System Calls:

How system call works?

- The Applications run in an area of memory known as user space. A system call connects to the operating system's kernel, which executes in kernel space. When an application creates a system call, it must first obtain permission from the kernel. It achieves this using an interrupt request, which pauses the current process and transfers control to the kernel.

- If the request is permitted, the kernel performs the requested action, like creating or deleting a file. As input, the application receives the kernel's output. The application resumes the procedure after the input is received. When the operation is finished, the kernel returns the results to the application and then moves data from kernel space to user space in memory.
- System calls are the entry points to the kernel space. If multiple activities are there, system call makes a system call table, and based on priority, kernel mode acts.

35

CSC259: Operating System

A simple system call may take few nanoseconds to provide the result, like retrieving the system date and time. A more complicated system call, such as connecting to a network device, may take a few seconds. Most operating systems launch a distinct kernel thread for each system call to avoid bottlenecks. Modern operating systems are multi-threaded, which means they can handle various system calls at the same time.

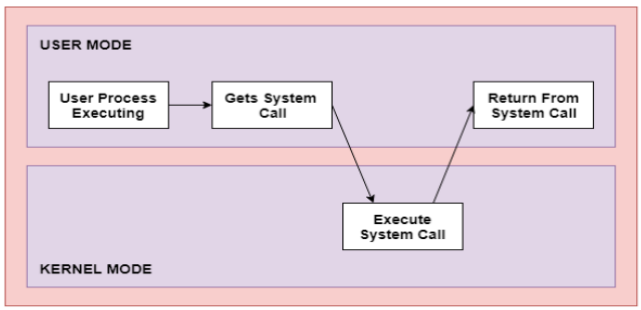


Figure: execution of system calls

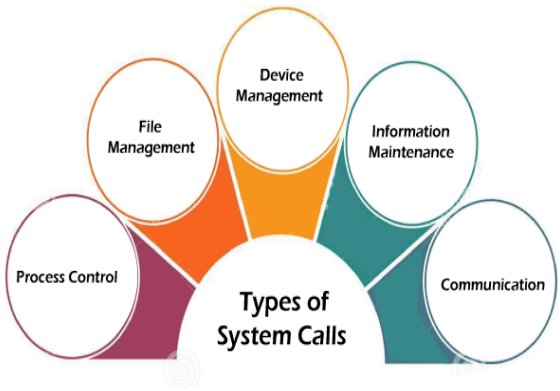
36

CSC259: Operating System

Types of System Calls

There are commonly five types of system calls. These are as follows:

1.Process Control
2.File Management
3. Device Management
4.Information Maintenance
5.Communication



37

CSC259: Operating System

1. Process Control

Process control is the system call that is used to direct the processes. Some process control examples include creating, load, abort, end, execute, process, terminate the process, etc.

2. File Management

File management is a system call that is used to handle the files. Some file management examples include creating files, delete files, open, close, read, write, etc.

3. Device Management

Device management is a system call that is used to deal with devices. Some examples of device management include read, device, write, get device attributes, release device, etc.

38

CSC259: Operating System

4. Information Maintenance

Information maintenance is a system call that is used to maintain information. There are some examples of information maintenance, including getting system data, set time or date, get time or date, set system data, etc.

5. Communication

Communication is a system call that is used for communication. There are some examples of communication, including create, delete communication connections, send, receive messages, etc.

39

CSC259: Operating System

Examples of Windows and Unix system calls

There are various examples of Windows and Unix system calls. These are as listed below in the table:

Process	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	Fork() Exit() Wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	Open() Read() Write() Close()
Device Management	SetConsoleMode() ReadConsole() WriteConsole()	Ioctl() Read() Write()

40

CSC259: Operating System

Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	Getpid() Alarm() Sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	Pipe() Shmget() Mmap()
Protection	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorgroup()	Chmod() Umask() Chown()

41

CSC259: Operating System

Some important API calls

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

42

CSC259: Operating System

System Programs in Operating System:

- **System Programming** can be defined as the act of building Systems Software using System Programming Languages.
- According to Computer Hierarchy, Hardware comes first then is Operating System, System Programs, and finally Application Programs.
- Program Development and Execution can be done conveniently in System Programs. Some of the System Programs are simply user interfaces, others are complex. It traditionally sits between the user interface and system calls. In the context of an operating system, system programs are nothing but a special software which give us facility to manage and control the computer's hardware and resources
- System programs are responsible for the ***creation*** and ***execution*** of a program

  and they can be used by the help of system calls because system calls define different types of system programs for different tasks.

CSC259: Operating System

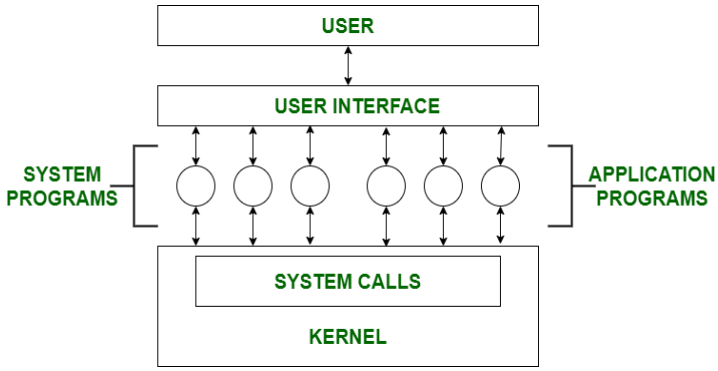


Figure: System program structure

CSC259: Operating System

Types of system programs:

1. Status Information:

The status information system programs provide required data on the current or past status of the system. They may include the system date, system time, available memory in system, disk space, logged in users etc. Terminal or other output devices or files or a window of GUI is used for showing the output of programs.

2. Communications:

Connections among processes, users, and computer systems are provided by programs. Users can send messages to another user on their screen, User can send e-mail, browsing on web pages, remote login, the transformation of files from one user to another.

CSC259: Operating System

3. File Management:

These system programs are used to manipulate system files. This can be done using various commands like create, delete, copy , rename , print etc. The files and directory are generally manipulated by these programs.

4.Program loading and execution:

When the program is ready after Assembling and compilation, it must be loaded into memory for execution. A loader is part of an operating system that is responsible for loading programs and libraries. It is one of the essential stages for starting a program. Loaders, relocatable loaders, linkage editors, and Overlay loaders are provided by the system..

5. File Modification:

This is used for modifying the content of files. Files stored on disks or other storage devices, we use different types of editors. For searching contents of files or perform transformations of files we use special commands.

CSC259: Operating System

6. Programming Language Support:

For common programming languages, we use Compilers, Assemblers, Debuggers, and interpreters which are already provided to users. It provides all support to users. We can run any programming language. All important languages are provided.

7. Application Programs:

It can perform a wide range of services as per the needs of the users. These include programs for database systems, word processors, plotting tools, spreadsheets, game , scientific applications etc.

CSC259: Operating System

Kernel of an OS:

The **kernel** is the core component of an operating system. It acts as a bridge between hardware and software, managing system resources and allowing applications to interact with hardware.

Functions of Kernel:

1.Process Management

1. Handles creation, scheduling, and termination of processes.
2. Manages multitasking and process synchronization.

2.Memory Management

1. Allocates and deallocates memory space.
2. Manages virtual memory and memory protection.

CSC259: Operating System

3. Device Management

1. Acts as an interface between hardware devices and the system.
2. Uses device drivers to control hardware.

4. File System Management

1. Manages data storage, access, and organization.
2. Provides a way to read/write files and directories.

5. System Call Handling

1. Provides an interface for user applications to request services via system calls.

CSC259: Operating System

Types of Kernels:

1.Monolithic Kernel

1. Entire OS runs in a single address space.
2. Fast but less secure (e.g., Linux).

2.Microkernel

1. Only essential functions run in the kernel; others run in user space.
2. More secure and modular (e.g., Minix).

3.Hybrid Kernel

1. Combines features of monolithic and microkernel.
2. Balances performance and modularity (e.g., Windows NT, macOS).

4.Exokernel

1. Minimalist kernel that gives applications more direct access to hardware.
2. Used in research environments.

CSC259: Operating System

Sturcture of Operating System:

The structure of an Operating System defines **how its components are organized** and how they interact with hardware and user applications. This structure affects the OS's **efficiency**, **security**, **maintainability**, and **portability**.

1. Simple Structure:

There are many operating systems that have a rather simple structure. These started as small systems and rapidly expanded much further than their scope. A common example of this is MS-DOS. It was designed for small group of people and company.

CSC259: Operating System

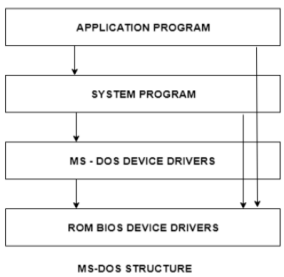


Figure: MS-DOS structure

It is better that operating systems have a modular structure, unlike MS-DOS. That would lead to greater control over the computer system and its various applications. The modular structure would also allow the programmers to hide information as required and implement internal routines as they see fit without changing the outer specifications.

CSC259: Operating System

2. **Layered Structure:**

OS is divided into a **hierarchy of layers**, each built on top of lower layers.

Characteristics:

•Lower layers manage hardware; higher layers provide user services.

•Each layer communicates only with its adjacent layers.

➕ **Advantages:**

•Modularity: Easier to debug and extend

•Encapsulation: Each layer hides its implementation

➖ **Disadvantages:**

•Performance overhead due to layered communication

•Difficult to design correct abstractions

53

CSC259: Operating System

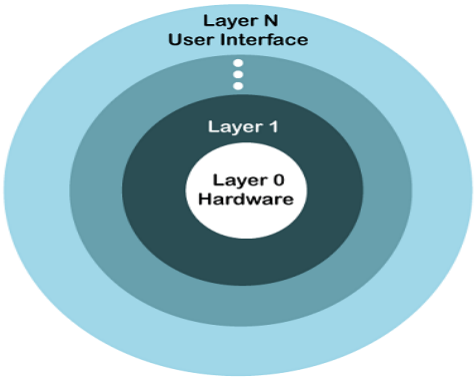


Figure: Layered Structure of OS

54

CSC259: Operating System

3. Monolithic Structure

A **single large kernel** where all OS services run in kernel space and share the same memory space.

•All components (process management, memory management, I/O, etc.) are part of a single program.

•Direct function calls between services.

•System calls are the only interface with user programs.

Examples:

•Early UNIX, MS-DOS, Linux (largely monolithic with modular support)

➕ **Advantages:**

•Simple design

•High performance due to less context switching

➖ **Disadvantages:**

•Hard to maintain and debug

•A bug in one component can crash the whole system

•Less secure (no isolation between modules)

55

CSC259: Operating System

4. Microkernel Structure

Moves as many services as possible out of the kernel into **user space**, leaving only minimal functionality in the kernel.

Kernel typically includes:

•Inter-process communication (IPC), Basic scheduling, Minimal memory management

Examples:

•QNX, Minix, L4

•Modern versions of macOS (XNU hybrid kernel has microkernel origins)

➕ **Advantages:**

•More secure and reliable (fault isolation)

•Easier to modify and extend

•Smaller codebase in kernel

➖ **Disadvantages:**

•Performance overhead due to user-kernel context switches

•Complex IPC mechanisms

56

CSC259: Operating System

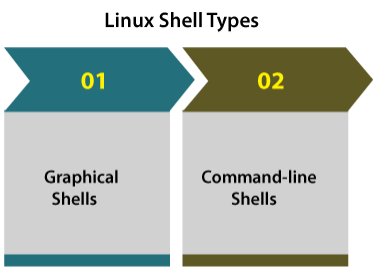
What is shell in OS?

- A shell is a program that takes commands typed by the user and call the operating system to run those commands.
- A shell is a program that acts as the interface between you and the Linux system, allowing you to enter commands for the operating system to execute.
- The shell is a command-line interface that allows the user to enter commands to interact with the operating system. It acts as an intermediary between the user and the kernel, interpreting commands entered by the user and translating them into instructions that the kernel can execute. The shell also provides various features like command history, tab completion, and scripting capabilities to make it easier for the user to work with the system

57

CSC259: Operating System

- The shell sends the result to the user over the screen when it has completed running a program which is the common output device. That's why it is known as "**command interpreter**".
- The shell is not just a command interpreter. Also, the shell is a programming language with complete constructs of a programming language such as **functions, variables, loops, conditional execution**, and many others.
- **Broadly, the shell is categorized into two main categories which are explained below:**



58

CSC259: Operating System

Graphical Shells

These shells specifies the manipulation of programs that are based on the graphical user interface (GUI) by permitting for operations like moving, closing, resizing, and opening windows and switching focus among windows as well.

Ubuntu OS or Windows OS could be examined as a good example that offers a graphical user interface to the user to interact with the program. Various users don't need for typing in any command for all the actions.

CSC259: Operating System

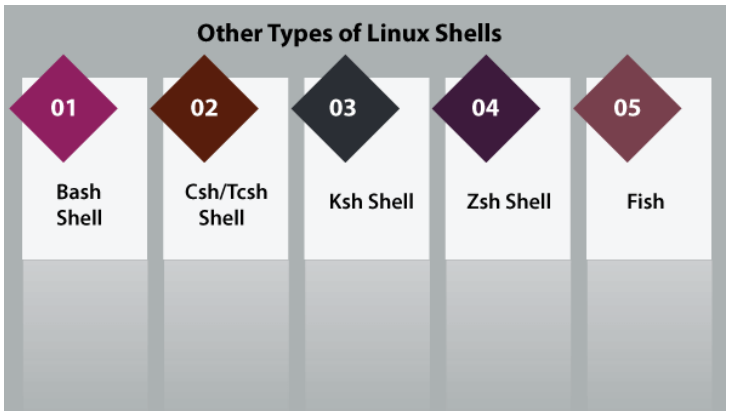
Command-line Shell

Various shells could be accessed with the help of a command-line interface by users. A unique program known as **Command prompt** in Windows or Terminal in macOS/Linux is offered for typing in the human-understandable commands and after that, it is being run. The result is further shown to the user on the terminal.

Working on a command-line shell is a complicated for many beginners due to it is hard to remember several commands. Command-line shell is very dominant and it permits users for storing commands in a file and run them together. In this way, a repetitive action could be automated easily. Usually, these files are known as Shell scripts in macOS/Linux systems and batch files in Windows.

CSC259: Operating System

Types of linux shell:



61

CSC259: Operating System

Open source operating system:

The term **"open source"** refers to computer software or applications where the owners or copyright holders enable the users or third parties to use, see, and edit the product's source code. The source code of an open-source OS is publicly visible and editable. The usually operating systems such as Apple's iOS, Microsoft's Windows, and Apple's Mac OS are closed operating systems. Open-Source Software is licensed in such a way that it is permissible to produce as many copies as you want and to use them wherever you like. It generally uses fewer resources than its commercial counterpart because it lacks any code for licensing, promoting other products, authentication, attaching advertisements, etc.

The open-source operating system allows the use of code that is freely distributed and available to anyone and for commercial purposes. Being an open-source application or program, the program source code of an open-source OS is available. The user may modify or change those codes and develop new applications according to the user requirement. Some basic examples of the open-source operating systems are **Linux, Open Solaris, Solus, React OS, Fedora, Chrome OS** etc.

62

CSC259: Operating System

Advantages of open source OS:

1. Reliable and efficient

The open-source operating systems are most reliable and efficient. Thousands of eyes monitor these because the source code is public. As a result, if there are any bugs or errors, they are fixed by the best developers worldwide.

2. Cost-efficient

Most of the open-source operating systems are free. And some of them are far less expensive than commercially closed products.

3. Flexibility

The great advantage is you may customize it as per your requirement. And there is creative freedom.

63

CSC259: Operating System

Disadvantages of open source OS:

1. Complicated

It is not as user-friendly as the ones that are closed. To use this software, you must have a basic understanding of technology.

2, Security risk

Despite the defects having been detected, there is a risk of assaults because the attackers have access to the source code.

3. No support

If you run across an issue, there is no customer support available to assist you.

64

CSC259: Operating System

THE END

Q and A