# Unit 1 Fundamentals of Operating System

[**Unit 1 Fundamentals of Operating System 1**](#_uox9ifj7z7fb)

[TWO MARKS 2](#_xamfwli0iexf)

[1. What is an Operating System? 2](#_6y7iwjngzrwd)

[2. Why is the Operating System viewed as a resource allocator & control program? 2](#_hu9ur665djit)

[3. What is the Kernel? 2](#_79ziv5eyksr1)

[4. What are Batch Systems? 2](#_jwwsr4gk7fyb)

[5. What is the advantage of Multiprogramming? 2](#_75zn0zwuasnk)

[6. What is an Interactive Computer System? 2](#_kop6vf5azgbh)

[7. What do you mean by Time-Sharing Systems? 2](#_iqi5uqq0fokt)

[8. What are Multiprocessor Systems & give their advantages? 3](#_fl2unv8sv3d9)

[9. What are the different types of Multiprocessing? 3](#_1r3bi6hj9ojp)

[10. What is Graceful Degradation? 3](#_au7w21rlyem7)

[11. What is Dual- Mode Operation? 3](#_fj757wnjwhy8)

[12. What are Privileged Instructions? 3](#_ywdj0nxa475y)

[13. How can a user program disrupt the normal operations of a system? 3](#_ig68hgpnjodc)

[14. How is the protection for memory provided? 4](#_xsg4wdpyi2od)

[15. What are the various OS Components? 4](#_6z0fhlp52ub3)

[16. What is a Process? 4](#_77siu2wnp3e1)

[17. What is a Process State and mention the various States of a Process? 4](#_14dv843ypw3q)

[18. What is a Process Control Block (PCB)? 5](#_1adfvlvzweq6)

[19. What is the use of Job Queues, Ready Queues & Device Queues? 5](#_7kt1qpssohzn)

[20. What is meant by Context Switch? 5](#_1i259t46l6g8)

[21. What is Spooling? 5](#_3dgcs7vh40vq)

[22. What are System Calls? 5](#_dzsp2rsf73ph)

[23. List the services provided by an Operating System? 6](#_mm50sooshgf)

[24. What are the two types of Real Time Systems? 6](#_mylq5e6hy3ij)

[25. What is the difference between Hard Real Time System and Soft Real Time System? 6](#_qqe8xrdeeuez)

[26. Write the difference between Multiprogramming and Non - Multiprogramming? 6](#_e59fktc780da)

[27. What are the design goals of an Operating System? 6](#_ntzy2sn7kf7y)

[28. What are the five major categories of System Calls? 6](#_9b8a98y71lb4)

[29. What is the use of Fork and Execve System Calls? 7](#_26fcnpsbbbi6)

[30. Define Elapsed CPU time and Maximum CPU time? 7](#_jzy6an3dnszi)

[FIFTEEN MARKS (15) 7](#_pe4vi9zbjk6p)

[1. What are the system components of an Operating System and explain them? Common System Components 7](#_nxljfwms5d7e)

[2. Define System Calls. Write about the various System Calls. 7](#_8scpt76ecz4t)

[3. What is a Process? Explain the Process Control Block and the various Process States. 8](#_wnbstdar1fuw)

[4. Explain Process Creation and Process Termination 8](#_elmthx54ho77)

## TWO MARKS

### 1. What is an Operating System?

An operating system is a program that manages the computer hardware. It also provides a basis for application programs and acts as an intermediary between a user of a computer and the computer hardware. It controls and coordinates the use of the hardware among the various application programs for the various users.

### 2. Why is the Operating System viewed as a resource allocator & control program?

A computer system has many resources, hardware & software that may be required to solve a problem, like CPU time, memory space, file-storage space, I/O devices & so on. The OS acts as a manager for these resources so it is viewed as a resource allocator.

The OS is viewed as a control program because it manages the execution of user programs to prevent errors & improper use of the computer.

### 3. What is the Kernel?

A more common definition is that the OS is the one program running at all times on the computer, usually called the kernel, with all else being application programs.

### 4. What are Batch Systems?

Batch systems are quite appropriate for executing large jobs that need little interaction. The user can submit jobs and return later for the results. It is not necessary to wait while the job is processed. Operators batched together jobs with similar needs and ran them through the computer as a group.

### 5. What is the advantage of Multiprogramming?

Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute. Several jobs are placed in the main memory and the processor is switched from job to job as needed to keep several jobs advancing while keeping the peripheral devices in use. Multiprogramming is the first instance where the Operating system must make decisions for the users. Therefore they are fairly sophisticated.

### 6. What is an Interactive Computer System?

Interactive computer system provides direct communication between the user and the system. The user gives instructions to the operating system or to a program directly, using a keyboard or mouse, and waits for immediate results.

### 7. What do you mean by Time-Sharing Systems?

Time-sharing or multitasking is a logical extension of multiprogramming. It allows many users to share the computer simultaneously. The CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.

### 8. What are Multiprocessor Systems & give their advantages?

Multiprocessor systems also known as parallel systems or tightly coupled systems are systems that have more than one processor in close communication, sharing the computer bus, the clock and sometimes memory & peripheral devices. Their main advantages are

* Increased throughput
* Economy of scale
* Increased reliability

### 9. What are the different types of Multiprocessing?

**Symmetric multiprocessing (SMP):**In SMP each processor runs an identical copy of the OS & these copies communicate with one another as needed. All processors are peers. Examples are Windows NT, Solaris, Digital UNIX, and OS/2 & Linux.

**Asymmetric multiprocessing:** Each processor is assigned a specific task. A master processor controls the system; the other processors look to the master for instructions or predefined tasks. It defines a master-slave relationship. Example: SunOS Version 4.

### 10. What is Graceful Degradation?

In multiprocessor systems, failure of one processor will not halt the system, but only slow it down. If there are ten processors & if any one fails then the remaining nine processors pick up the work of the failed processor. This ability to continue providing service is proportional to the surviving hardware is called graceful degradation.

### 11. What is Dual- Mode Operation?

The dual mode operation provides us with the means for protecting the operating system from wrong users and wrong users from one another. User mode and monitor mode are the two modes. Monitor mode is also called supervisor mode, system mode or privileged mode. Mode bit is attached to the hardware of the computer in order to indicate the current mode. Mode bit is 0 for monitor mode and 1 for user mode.

### 12. What are Privileged Instructions?

Some of the machine instructions that may cause harm to a system are designated as privileged instructions. The hardware allows the privileged instructions to be executed only in monitor mode.

### 13. How can a user program disrupt the normal operations of a system?

A user program may disrupt the normal operation of a system by,

* Issuing illegal I/O operations
* By accessing memory locations within the OS itself
* Refusing to relinquish the CPU

### 14. How is the protection for memory provided?

The protection against illegal memory access is done by using two registers. The base register and the limit register. The base register holds the smallest legal physical address; the limit register contains the size of the range. The base and limit registers can be loaded only by the OS using special privileged instructions

### 15. What are the various OS Components?

The various system components are,

* Process management
* Main-memory management
* File management
* I/O-system management
* Secondary-storage management
* Networking
* Protection system
* Command-interpreter system

### 16. What is a Process?

A process is a program in execution. It is the unit of work in a modern operating system. A process is an active entity with a program counter specifying the next instructions to execute and a set of associated resources. It also includes the process stack, containing temporary data and a data section containing global variables.

### 17. What is a Process State and mention the various States of a Process?

As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. Each process may be in one of the following states:

* New
* Running
* Waiting
* Ready
* Terminated

### 18. What is a Process Control Block (PCB)?

Each process is represented in the operating system by a process control block also called a task control block. It contains many pieces of information associated with a specific process. It simply acts as a repository for any information that may vary from process to process. It contains the following information:

* Process state
* Program counter
* CPU registers
* CPU-scheduling information
* Memory-management information
* Accounting information
* I/O status information

### 19. What is the use of Job Queues, Ready Queues & Device Queues?

As a process enters a system, they are put into a job queue. This queue consists of all jobs in the system. The processes that are residing in main memory and are ready & waiting to execute are kept on a list called ready queue. The list of processes waiting for a particular I/O device is kept in the device queue.

### 20. What is meant by Context Switch?

Switching the CPU to another process requires saving the state of the old process and loading the saved state for the new process. This task is known as context switch. The context of a process is represented in the PCB of a process.

OPERATING SYSTEMS Page - 3 -

### 21. What is Spooling?

Spooling means Simultaneous Peripheral Operations On Line. It is a high-speed device like a disk interposed between a running program and a low speed device involved with the program in input/output. It dissociates a running program from the slow operation of devices like printers.

### 22. What are System Calls?

System calls provide the interface between a process and the Operating system. System Calls are also called Monitor calls or Operating-system function calls. When a system call is executed, it is treated by the hardware as a software interrupt. Control passes through the interrupt vector to a service routine in the operating system, and the mode bit is set to monitor mode.

### 23. List the services provided by an Operating System?

* Program execution
* I/O Operation
* File-System manipulation
* Communications
* Error detection

### 24. What are the two types of Real Time Systems?

* Hard real time system
* Soft real time system

### 25. What is the difference between Hard Real Time System and Soft Real Time System?

A hard real time system guarantees that critical tasks are completed on time. In a soft real time system, a critical real-time task gets priority over the other tasks, and retains that priority until it completes. Soft real time systems have more limited utility than do hard real-time systems.

### 26. Write the difference between Multiprogramming and Non - Multiprogramming?

The operating system picks and begins to execute one of the jobs in the memory. Eventually, the job may have to wait for some task, such as a tape to be mounted, or an I/O operation to complete. In a non-multiprogrammed system, the CPU would sit idle. In a multiprogramming system, the operating system simply switches to and executes another job. When that job needs to wait, the CPU is switched to another job, and so on. Eventually, the first job finishes waiting and gets the CPU back. As long as there is always some job to execute, the CPU will never be idle.

### 27. What are the design goals of an Operating System?

The requirements can be divided into two basic groups: User goals and System goals. Users desire that the system should be convenient and easy to use, easy to learn, reliable, safe and fast. The Operating system should be easy to design, implement, and maintain. Also it should be flexible, reliable, error free and efficient. These are some of the requirements, which are vague and have no general solution.

### 28. What are the five major categories of System Calls?

* Process Control
* File-management
* Device-management
* Information maintenance
* Communications

### 29. What is the use of Fork and Execve System Calls?

Fork is a System call by which a new process is created. Execve is also a System call, which is used after a fork by one of the two processes to replace the process memory space with a new program.

### 30. Define Elapsed CPU time and Maximum CPU time?

**Elapsed CPU Time:** Total CPU time used by a process to date.

**Maximum CPU Time:** Maximum amount of CPU time a process may use.

## FIFTEEN MARKS (15)

### 1. What are the system components of an Operating System and explain them? Common System Components

* **Process Management:**A process is an instance of a running computer program. The process management component is responsible for creating, scheduling, and terminating processes. It also ensures that processes share resources fairly and efficiently.
* **Memory Management:** Main memory is the computer's fastest memory, but it is also the most limited. The memory management component is responsible for allocating and deallocating main memory to processes as needed. It also ensures that processes do not overwrite each other's memory.
* **File Management:** A file is a collection of data stored on secondary storage, such as a hard disk drive or SSD. The file management component provides a way for users to store and organize files. It also allows processes to read and write files.
* **Device Management:** The device management component manages the system's input/output (I/O) devices, such as the keyboard, mouse, and display. It allows processes to interact with these devices.
* **Security Management:** The security management component protects the system from unauthorized access and malicious attacks. It does this by authenticating users, enforcing access control policies, and detecting and responding to security threats.
* **Networking:** The networking component allows processes to communicate with each other over a network. This includes both local networks (LANs) and wide area networks (WANs).
* **Command-Interpreter System:**The command interpreter provides a way for users to interact with the operating system, typically by issuing commands through a text-based interface. For example, the command interpreter can be used to start and stop programs, manage files, and configure the system.

### 2. Define System Calls. Write about the various System Calls.

A system call is a software mechanism that allows a user-mode program to request a service from the operating system kernel. System calls are the only way for a user-mode program to access protected system resources, such as the file system, hardware devices, and other processes.

System calls are typically implemented as special instructions that cause the processor to switch from user mode to kernel mode. When the processor is in kernel mode, it has unrestricted access to all system resources. After the kernel has completed the requested service, it switches back to user mode and the user-mode program continues execution.

**Types of System Calls**

* **File management:** creating, opening, reading, writing, and closing files
* **Process management:** creating, scheduling, and terminating processes
* **Memory management:** allocating and deallocating memory
* **Device management:** accessing and controlling hardware devices
* **Networking:** sending and receiving data over a network
* **Information:** getting information about the system, such as the current time and date, the amount of free memory, and the list of running processes

Here are some common types of system calls:

* **Process Control:**
  + fork(): Create a new process by duplicating the calling process.
  + exec(): Replace the current process image with a new one.
  + exit(): Terminate the calling process.
* File Management:
  + open(): Open a file and obtain a file descriptor.
  + read(): Read data from a file.
  + write(): Write data to a file.
  + close(): Close a file descriptor.
* Device Management:
  + ioctl(): Perform I/O control operations on devices.
  + read() and write(): These functions are also used for device I/O.
  + open() and close(): Device access is often done through file descriptors.
* Information Maintenance:
  + getpid(): Get the process ID of the current process.
  + getppid(): Get the parent process ID.
  + time(): Get the current time.
* Communication:
  + pipe(): Create an inter-process communication channel.
  + msgget(), msgsnd(), msgrcv(): Message queue operations.
  + semget(), semop(): Semaphore operations.
  + shmget(), shmat(), shmdt(): Shared memory operations.
* Memory Management:
  + brk(): Change the data segment size of the calling process.
  + malloc(), free(): Memory allocation and deallocation functions often use system calls behind the scenes.
* File System Management:
  + mkdir(), rmdir(): Create and remove directories.
  + link(), unlink(): Create and remove directory entries.
  + chmod(), chown(): Change file permissions and ownership.
* Network Communication:
  + socket(): Create a new communication endpoint (socket).
  + bind(), listen(), accept(): Functions for setting up server sockets.
  + connect(): Establish a connection to a remote socket.
* Security:
  + chmod(): Change the permissions of a file or directory.
  + chown(): Change the ownership of a file or directory.

**Importance of system calls**

System calls are an important part of operating systems because they provide a standardized way for user-mode programs to access system resources. This makes programs more portable and easier to develop. System calls also help to protect the system from unauthorized access and malicious attacks.

For example, the open() system call is used to open a file. The kernel checks the permissions on the file before allowing the program to open it. This helps to protect the file from unauthorized access.

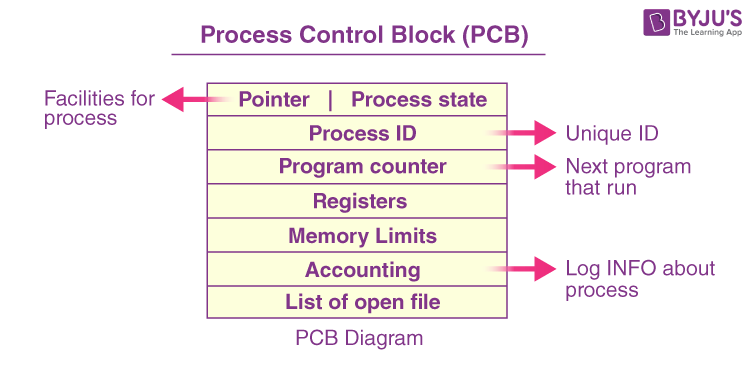
System calls are also used to implement security features such as authentication and authorization. For example, the getuid() system call returns the user ID of the current process. This information can be used to determine whether the process has permission to access a particular resource.

### 3. What is a Process? Explain the Process Control Block and the various Process States.

A process is an instance of a running computer program. It is a self-contained entity that has its own resources, such as memory, CPU time, and open files. Processes are created, scheduled, and terminated by the operating system.

**Process Control Block (PCB)**

The Process Control Block (PCB) is a data structure that is used by the operating system to store information about each process. The PCB contains information such as the process ID, process state, program counter, CPU registers, memory allocation, open file descriptions, and accounting information.

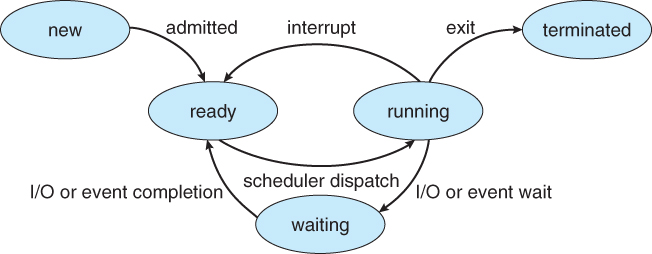


The PCB is essential for process management. It allows the operating system to track the state of each process and to switch between processes efficiently.

**Process States**

A process can be in one of the following states:

* **New:** The process has been created but has not yet been scheduled to run.
* **Ready:** The process is ready to run, but the CPU is not currently available.
* Running: The process is currently executing on the CPU.
* Waiting: The process is waiting for an event to occur, such as a file to be read or a network packet to be received.
* Terminated: The process has completed execution or has been terminated by the operating system.



### 4. Explain Process Creation and Process Termination

**Process Creation**

Process creation is the operation of creating a new process in an operating system. It is typically done by making a system call to the operating system. The operating system will then create a new PCB for the new process and allocate it the necessary resources.

The following steps are involved in process creation:

1. The operating system allocates memory for the new process.
2. The operating system creates a new PCB for the new process.
3. The operating system initializes the PCB with information about the new process, such as the process ID, parent process ID, program counter, and memory allocation.
4. The operating system places the new process in the ready state.

**Process Termination**

Process termination is the operation of ending the execution of a process. It is typically done by making a system call to the operating system. The operating system will then release the resources that were allocated to the process and remove the process from the system.

The following steps are involved in process termination:

1. The operating system deallocates the memory that was allocated to the process.
2. The operating system closes all of the files that were opened by the process.
3. The operating system removes the process's PCB from the process table.
4. The operating system returns control to the parent process of the terminated process.

**Reasons for Process Termination**

There are a number of reasons why a process may be terminated:

* The process has completed execution.
* The process has been terminated by the user.
* The process has been terminated by the operating system due to an error.
* The process has been terminated by the parent process.