

# Operating Systems

**Computer Science  
&  
Information Technology (CS)**



**RANK 1 GATE 2015**

**Computer Science  
Ravi Shankar Mishra**

**20 Rank under AIR 100**

## Postal Correspondence

- ✓ Examination Oriented Theory, Practice Set
- ✓ Key concepts, Analysis & Summary



## CONTENT

1. INTRODUCTION .....	03-07
2. PROCESS .....	08-22
3. CPU SCHEDULING .....	23-31
4. DEALOCK .....	32-42
5. CONCURRENCY CONTROL & PROCESS SYNCHRONIZATION .....	43-62
6. MEMORY OF MANAGEMENT .....	63-94
7. FILE SYSTEM AND I/O MANAGEMENT.....	95-117
8. PROTECTION AND SECURITY.....	118-130
9. GATE PRACTICE SET.....	131-148

### Syllabus:

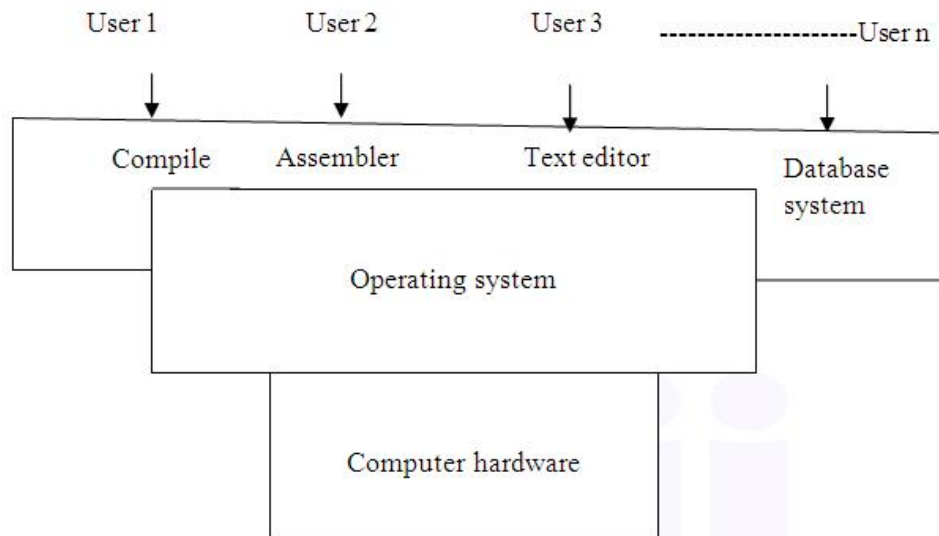
Operating System: Processes, Threads, Inter-process communication, Concurrency, Synchronization, Deadlock, CPU scheduling, Memory management and virtual memory, File systems, I/O systems, Protection and security

# CHAPTER-1

## INTRODUCTION

### Operating system:

- An operating system is a program that controls the execution of application program and acts as an interface between applications and the computer hardware.



### Component of computer system:

Abstract view of the components of a computer system

A computer system is divided into four components

1. Hardware
2. Operating system
3. Application program
4. The users

- **Hardware:** The central processing unit, the memory and the input/output etc, the Hardware devices and provide the basic computing resources.
- **Operating system:** control and co-ordinate for the use of the hardware among the various application program for various users.
- **Application program:** Such as MS-DOS, database system, games and business program define the way in which these resources are used to solve computing problems of users.

Operating system can be viewed as:

- Resource Allocator: manage and allocate resources.
- Control program: A control program manages the execution of user program to prevent errors and improper use of computer. It is especially concerned with operating and control of I/O devices.

### Kernel

- The kernel is the central module of an operating system. It is the part of the operating system that loads first, and it remains in main memory.
- The kernel is responsible for memory management, process and task management, and disk management.
- The kernel connects the system hardware to the application software. Every operating system has kernel. When a process makes request of the kernel, the request is called as system call.

### System call:

- A system call is how a program requests a service from an operation system's kernel.
- System call provides the interface between a process and the operating system.
- System call typically written in high level language (C or C++).
- System calls are mostly accessed by program via a high – level Application programming interface (API) rather than direct system call use.

The relationship between API, system call, OS is shown below.

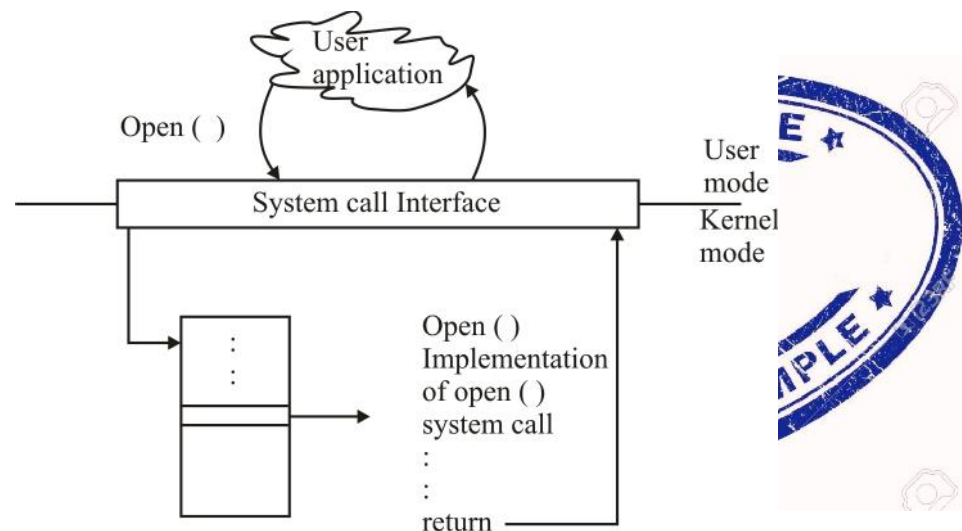


Figure: Calling open ( ) system call

### Explanation:

- First user application call open ( ) system call, using the application program interface.
- Then it is transferred to kernel.
- Kernel then transferred this call to real implementation of open ( ).
- At the end, value is returned to interface.

### System boot

- When power initialized on system, execution starts at a fixed memory location.
- Firmware ROM contains the initial boot code that is called as bootstrap loader.
- Bootstrap loader is used to locate the kernel, loads into memory, and start it.
- Sometimes, we need two step to load initial code (bootstrap leader):
- Step 1: There is boot block at fixed block that is loaded by ROM code.

- Step 2: This boot block loads the boot strap loader from disk.
- Common bootstrap leader, GRUB, allows selection of kernel from multiple disks, version's kernel options.

### Functions of operating system:

- Process management
- Main memory management
- File management
- I/O system management
- Secondary storage management
- Networking
- Security and protection

### Typical service provided by operating system

- Program creation
- Program execution
- Access to input/output devices
- Controlled access to files
- System access
- Error Detection and response
- Accounting

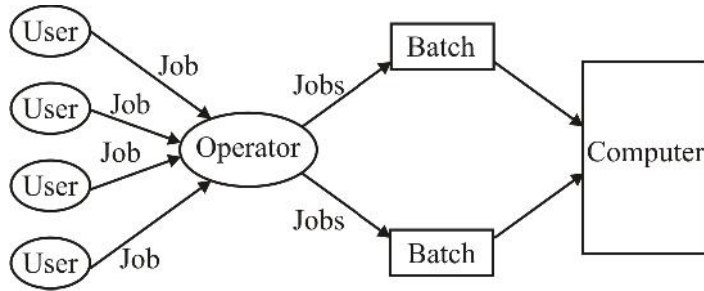


### Types of operating system

- Batch operating systems
- Multi programmed systems
- Time sharing operating system.
- Multi processor operating system
- Distributed operating system
- Real time operating system

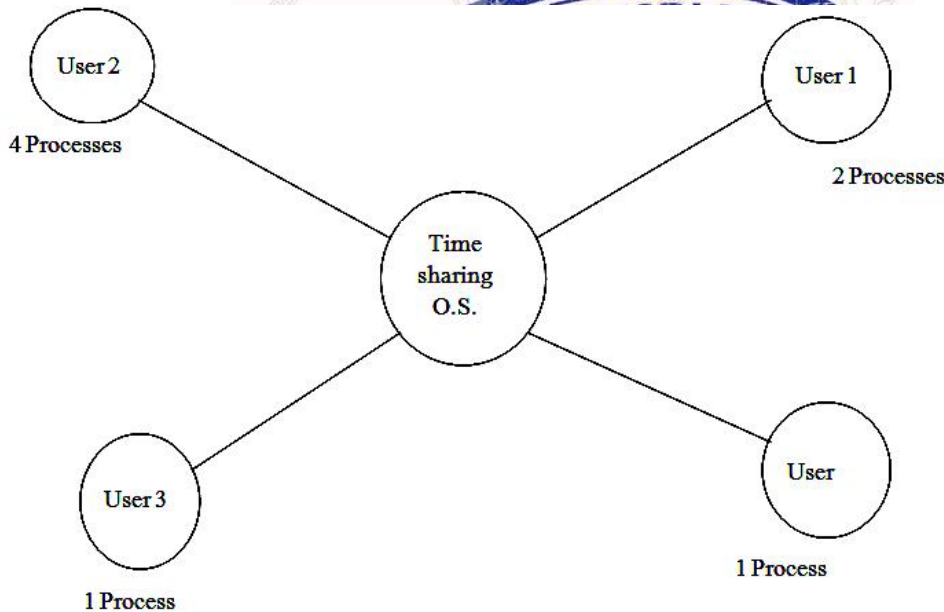
### Batch Operating System

- In the batch operating system, each user prepares his job on an offline devices like punch cards and submit it to the computer operator.
- The users of batch operating system do not interact with the computer directly.
- To speed up processing, jobs with similar needs are batched together and run as a group.



## Time sharing operating system

- Time sharing is a technique that allows many users, located at various terminals, to use a particular computer system at the same time.
- The main difference between multi programmed batch systems and time sharing systems is that in case of multi programmed batch systems, objective is to maximize processor use, where as 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. In this, a fixed time quanta of CPU is assigned to each user. In round robin fashion, control is transferred to users.



## Multi programmed Operating System

### Characteristics of multi programmed Operating System

More than one process can be loaded into main memory

- If any request for I/O activity block the process than control is switched to another process which is decided by short term scheduler.
- If any process wants to execute the I/O activity then completion of I/O activity and execution of other process is done parallel.



- In a non multi programmed system the CPU would sit idle. In multi programming system the operating system. simple switches to and execute another job.

**Drawback:**

Multi programmed operating system usually don't support multiuser.

**Multitasking**

- Multitasking is a logical extension of multi programming that allows a user to perform more than one task at a time.
- Microsoft windows 2000, IBM's OS/390, and Linux are example of operating systems that can do multitasking.

**Multiprocessor:**

- Multiprocessing is the use of two or more central processing units (CPUs) within a single computer system. These multiple CPUs are in close communication sharing the computer bus, memory and the other peripheral devices. These systems are referred as tightly coupled system.
- One advantage of building this type of system is increased throughput. By increasing the number of processor we hope to get more work done in a shorter period of Time.
- Multiprocessor can also save money because the processors can share peripheral, cabinet and power supply. If several program are to operate on same set of data, it is cheaper to store those data on the disk and to have all the processor share them, rather than to have many computer with local disk and many copies of data.
- Another advantage is increased reliability.

**Two types of multiprocessor system**

## 1. Symmetric multiprocessing model

In this model processor runs identical copy of operating system and these copies communicate with each other when needed.

## 2. Asymmetric multiprocessing:

In this model each processor is assigned a specific task. A master processor control the system, the other processor either look to master for instruct or have predefined tasks. This scheme defines a master-slave relationship.

**Distributed system:**

- Also referred as loosely coupled system i.e. do not share clock, memory instead each processor have its own local memory.
- If a number of different sites are connected to one another than a user at one site may be able to use the resources available at another.
- If one site fails in distributed system then the remaining site can potentially continue operating.
- When many sites are connected to one another by a communication network the processor at different site have the opportunity to exchange information.

**Real time system:**

- Used when there are vivid time requirement on the operating of processor or flow of date.
- System that control scientific experiments, medical imaging system, industrial control system and some display system are real time system.
- A real time operating system has well defined, fixed time constraints. Processing must be done within defined constraints or system will fail.

**Two types of real time System:-****A hard time system:**

Guarantee that critical task completed on the time. This goal requires that all delays in the system be bounded from the retrieval of stored data to the time that it takes the operating system to finish any request mode of it. Kernel delays need to be bounded.

**Soft real time system:**

A less restrictive type of real time system is soft real time system, when a critical real time gets priority over other tasks and retains that priority until it completes. Soft real time is an achievable goal that can be mixed with each other types of systems. However they have more limited utility than do hard real time system.





## CHAPTER-2

### PROCESS & THREADS

### Process

A program in execution is called as process.

→ A program is passive while a process is active. A process will need certain resources such as CPU, time, memory, files and input/output devices to accomplish its task.

### Process state diagram:

A flow chart to represent all the different states which can be achieved by any process.

#### 5 – State process model:

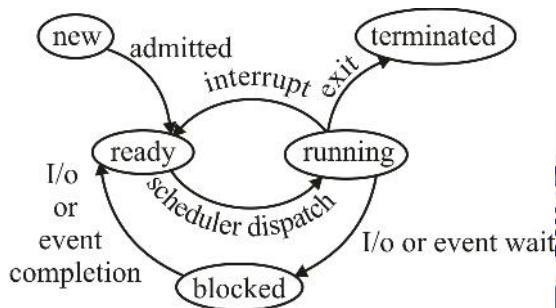


Figure: Process state

**New:** A process that has just been created but has not yet been admitted to the pool of executable processes by the operating system. Typically a new process has not yet been loaded into main memory; although its process control block has been created.

**Ready:** The process is in the main memory and ready for execution.

**Running:** The process is in the main memory and executing in the CPU.

**Blocked:** The process is in the main memory and waiting for an event to occur (waiting for I/O completed).

**Exit:** The process is halted or completed successfully.

### State transition

(i) **New – ready**

The OS will move a process from the new state to the ready state when it is prepared to take on an additional process.

(ii) **Ready – running**

When the processor requires any process to execute then it selects the process from the ready state to the running state.

(iii) **Running - exit**

The currently running process is terminated by the OS, if the process indicates that it has completed, or if it aborts.

(iv) **Running – ready**

After executing a specified amount of time, the process will moved to ready state.

(v) **Running – blocked**

A process is put in the blocked state, if it requests something for which it must wait. For example: request for input/output device.

(vi) **Blocked - ready**

A process in the blocked state is moved to ready state when the event for which it has been waiting occurs.

(vii) **Ready – exit**

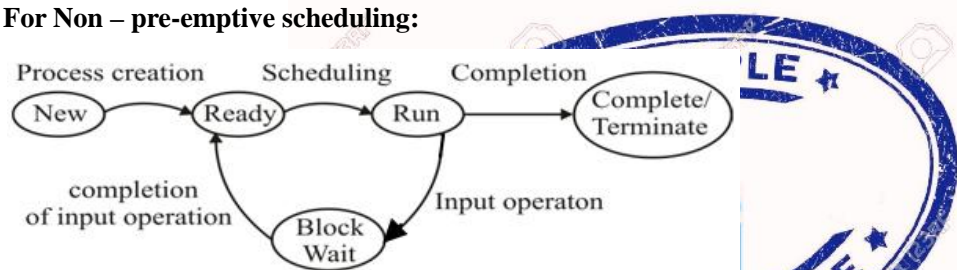
This transition happens, when a parent may terminate a child's process at any time. (If a parent terminates, all child process associated with that parent may be terminated).

(viii) **Blocked – exit**

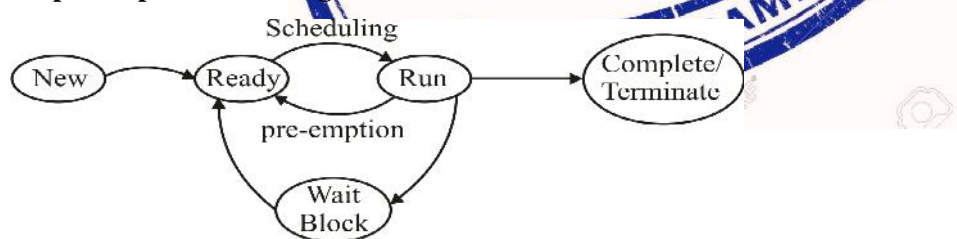
This transition happens, when a parent may terminate a child's process at any time. (If a parent terminates, all child process associated with that parent may be terminated).

### Note:

#### For Non – pre-emptive scheduling:



#### For pre-emptive scheduling:



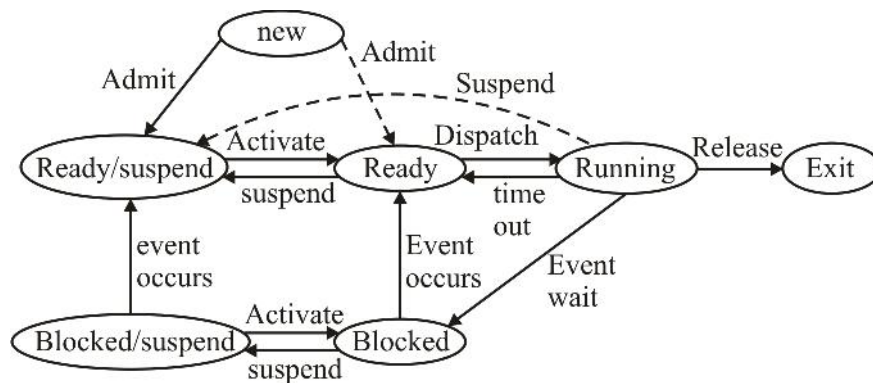
## Swapping

There is limitation on main memory so at some point in time; we required moving part or all of a process from main memory to disk.

When none of the process in main memory is in the ready state, the OS swapped out one of the blocked processes on to disk into a suspended queue.

Suspended queue is a queue of existing processes that have been temporarily kicked out of main memory, or suspended.

## Process state transition diagram with suspend state



### Suspended Process Transition:

#### Blocked -Blocked Suspended:

If there are no ready processes then at least one blocked process is swapped out to make room for another process that is not blocked. Transition can be made even if there are ready processes available, if the operating system determine that the currently running process or ready process that it would like to dispatch require more main memory to maintain adequate performance.

#### Block Suspended-Ready Suspended

A process can be sent from blocked suspended state to ready suspended state if the event for which it was waiting has occurred.

#### Ready Suspended- Ready:

When there are no ready processes in the main memory the operating system needs to bring one to continue the execution process. In addition it might be that a process in the ready suspended state may have a higher priority compared with any of processes in the ready state.

#### New- Ready Suspended & New- Ready:

Whenever a process is created it can either go to a ready queue or a ready suspended queue. In either case operating system creates tables and allocates space to their processes.

#### Ready-Ready Suspended:

Normally operating system suspends a blocked process rather than a ready process. But in a situation when some space is required, low priority ready process may be swapped with a blocked high priority process in anticipation that it will be ready soon.

#### Blocked suspended – Blocked:

Consider the situation that a process terminates, resulting space in main memory. There is a process in the

blocked suspended state with high priority that any of process in ready suspended state and O.S believes that the blocking event occur soon. Under such condition it may be reasonable to bring a blocked process into main memory in preference to ready process.

### Running-Ready Suspended:

Normally a running process is moved to the ready state when its time runs out. However if operating system preempts that a Higher priority process on the blocked suspended state has just become unblocked, it can move a running process directly to ready suspended state to free some memory.

### CPU Mode:

CPU modes are operating mode that placed restriction on type and scope of operations that can be performed by certain processes being run by the CPU.

There are two CPU modes:

1. User mode/Slave mode/problem state
  2. Kernel Mode/ Master mode/ supervision mode/ privileged mode / supervisor state
- 1) **User mode:** In user mode, the executing code has no ability to directly access hardware or reference memory. Code running in user mode must delegate to system API's to access hardware or memory. Crashes in user mode are always recoverable. Most of the codes are running on user mode.
- 2) **Kernel mode:** In kernel mode, the executing code has complete and unrestricted access to the underlying hardware. It can execute any CPU instruction and reference any memory address. Kernel mode is generally reserved for the lowest level, most trusted function of the OS. If any crash happens in kernel mode then it will halt the entire system.

O.S always executes in kernel mode and application program always run in user mode so can't perform critical operation.

### Process controls block (PCB):

Program Control Block is a kernel data structure that is used to describe a process.PCB may vary from process to process.

#### PCB Structure

<b>Pointer to various data structure</b>
<b>Process Sate</b>
<b>Process Identification Number</b>

<b>User Identification Number</b>	
<b>Process Operating Mode</b>	
<b>Scheduling Parameter</b>	
<b>Resources acquired</b>	
<b>H/W context</b>	<b>Process register value, stack pointer etc.</b>
<b>S/W context</b>	<b>Single Info, open file, Open socket, memory region</b>

**Pointer:** It is used to link one PCB to another (Since info of processes in the system are maintained as linked list)

**Process state:** State may be New, Ready, running, waiting etc.

**Process Number:** It is the number that identifies process in the system.

**Program counter:** It indicates the address of next instruction to be executed for this process.

**CPU Registers:** They vary in number and Type based on computer Architecture. They include accumulator, Index Register, stack pointer, general purpose register plus condition code information on an interrupt this state information along with program counter is saved to allow a process to continue correctly later.

**CPU scheduling Information:** It includes process priority, Pointer to scheduling queue, scheduling parameter etc.

**Memory Management information:** It includes value of base and limit registers, Page table/ segment table based on memory management scheme used.

**Accounting Information:** It include amount of CPU and Real time uses, time Limit, account number job/Process Number etc.

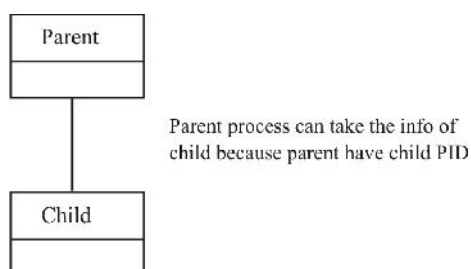
## Process creation:

The process can be created for many reasons like:

- (i) Created by a parent process to increase modularity and parallelism
- (ii) Process is created when new batch job is submitted
- (iii) Process can be created by operating system to provide services.
- (iv) Process can be created, when a new user log on to the system.

When a new process is created, operating system creates the data structure for the process and allocates address space in main memory to the process.

During execution a process can create another process also. Every process has its own ID, Known as PID.





There are many system calls that are used to create a new process for example:

- (i) **Fork ( )** system call generates two identical processes with separate memory. The new process (child process) has a unique process ID and has a different parent process ID.

**For example**

Process Creation in UNIX

```
Main ( )
```

```
{
```

```
Int i=0
```

```
Int PID;
```

```
PID=fork( ) // system call
```

```
If (PID= = 0)
```

```
{
```

```
Print f(“Inside the child Process”)
```

```
}
```

```
Else
```

```
{
```

```
Printf(“Inside the Parent Process”);
```

```
}
```

```
}
```

When execute the fork system call it return child PID to parent process and to child process. Control of execution starts after fork system call in both processes. By the n fork system call  $2^N - 1$  new process are created.

- (ii) **V fork ( )** system call generates two processes that share the same memory.
- (iii) **The exec ( )** family of function replaces the current process image with a new process image.
- (iv) **Clone:** Clone, as fork, creates a new process. These calls allow the child process to share parts of its execution context with the calling process, such as the memory space, the table of file descriptions, and the table of signal handlers.

## Inter Process Communication

- Inter process communication allows two or more processes to exchange information.
- **Independent process:** A process is independent if it cannot affect or be affected by the other process executing in the system. Any process that does not share data with any other process is independent.
- **Cooperating Process:** A process is cooperating if it can affect or be affected by the other processes executing in the system. Any process that share data with any other process is cooperative process.



→ There are two fundamental method for inter process communication:

- (i) Shared memory
- (ii) Message passing

### **Shared Memory**

- One of the simplest inter process communication method is using shared memory.
- Shared memory allows two or more processes to access the same memory.
- When one process changes the memory, all the other processes see the modification
- Basically, a shared memory region resides in the address space of the process. The processes that wish to communicate using this shared memory segment must attach it to their address space.
- The form of the data and the location are determined by these processes and are not under the operation system's control.

**New Edition with GATE 2015 Solutions is available.**

**GATE Classroom Coaching**

**GATE Postal Correspondence Coaching**

**GATE Online test Series**



**To Buy Postal Correspondence Package call at 0-9990657855**