Date : 29th - Oct- 2020
Morning Session : 9am – 11.00 PM
By ~ Rohan Kumar

Topics: Operating System -2

Process is the execution of a program that performs the actions specified in that program. It can be defined as an execution unit where a program runs. The OS helps you to create, schedule, and terminates the processes which are used by the CPU. A process created by the main process is called a child process.

Process operations can be easily controlled with the help of PCB(Process Control Block). You can consider it as the brain of the process, which contains all the crucial information related to processing like process id, priority, state, CPU registers, etc.

What is Process Management?

Process management involves various tasks like creation, scheduling, termination of processes, and a deadlock. Process is a program that is under execution, which is an important part of modern-day operating systems. The OS must allocate resources that enable processes to share and exchange information. It also protects the resources of each process from other methods and allows synchronization among processes.It is the job of the OS to manage all the running processes of the system. It handles operations by performing tasks like process scheduling and such as resource allocation.

Process Architecture

Here, is an Architecture diagram of the Process

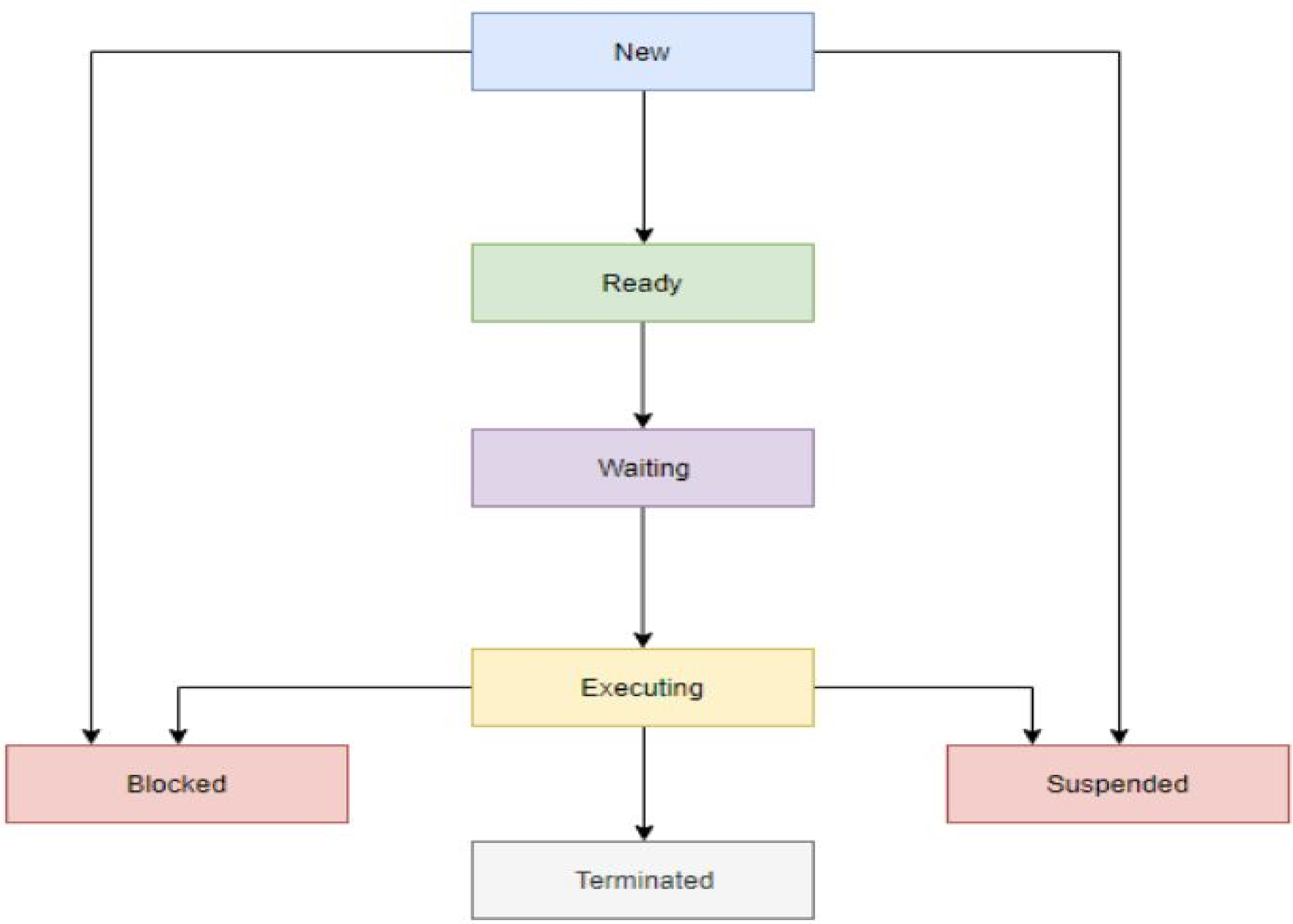
- **Stack:** The Stack stores temporary data like function parameters, returns addresses, and local variables.
- **Heap** Allocated memory, which may be processed during its run time.
- **Data:** It contains the variable.
- **Text:** Text Section includes the current activity, which is represented by the value of the Program Counter.

Process Control Blocks

The PCB is a full form of Process Control Block. It is a data structure that is maintained by the Operating System for every process. The PCB should be identified by an integer Process ID (PID). It helps you to store all the information required to keep track of all the running processes.

It is also accountable for storing the contents of processor registers. These are saved when the process moves from the running state and then returns back to it. The information is quickly updated in the PCB by the OS as soon as the process makes the state transition.

Process States



process state is a condition of the process at a specific instant of time. It also defines the current position of the process.

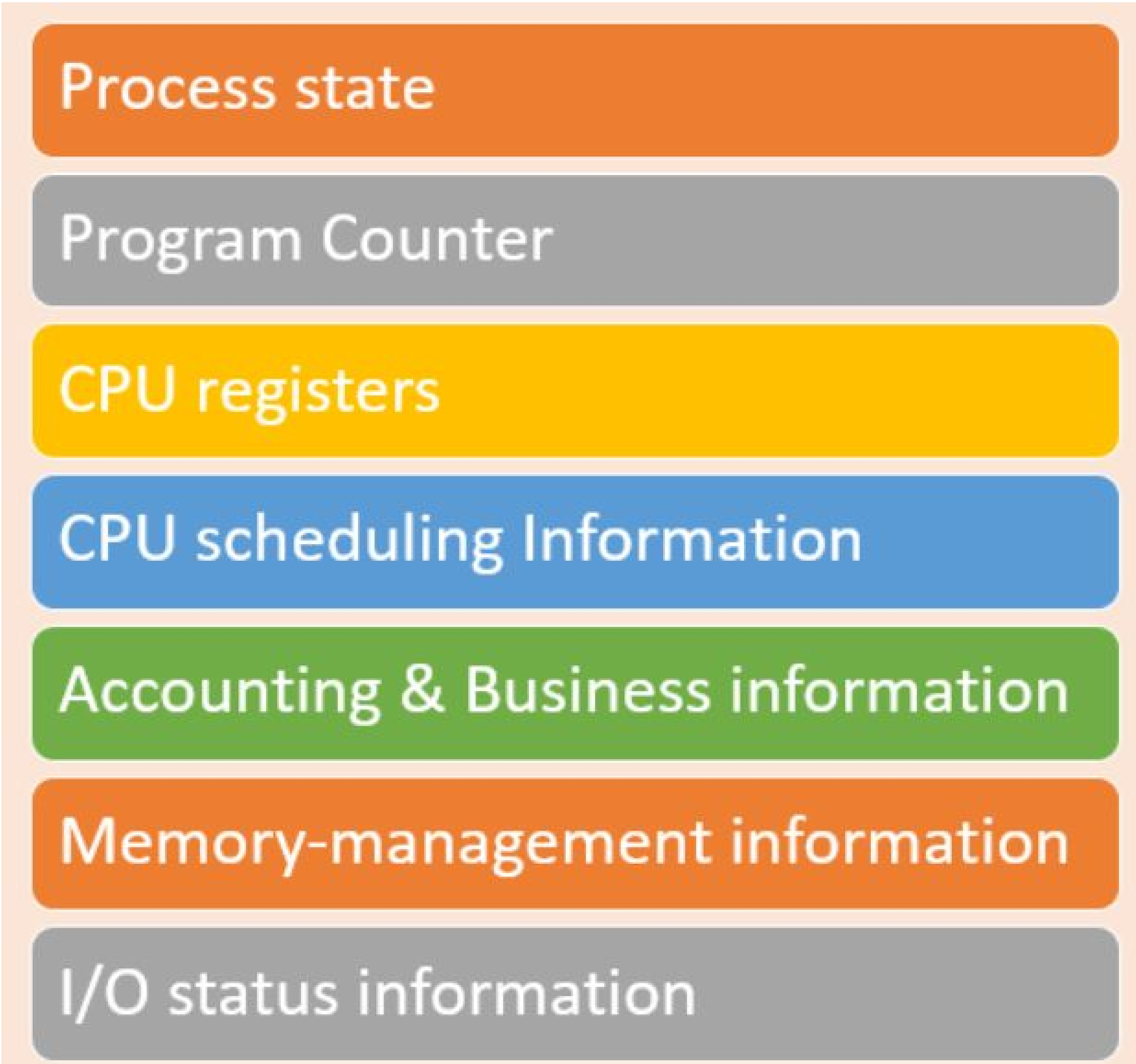
There are mainly seven stages of a process which are:

- New: The new process is created when a specific program calls from secondary memory/ hard disk to primary memory/ RAM a
- Ready: In a ready state, the process should be loaded into the primary memory, which is ready for execution.
- Waiting: The process is waiting for the allocation of CPU time and other resources for execution.
- Executing: The process is an execution state.
- Blocked: It is a time interval when a process is waiting for an event like I/O operations to complete.
- Suspended: Suspended state defines the time when a process is ready for execution but has not been placed in the ready queue by OS.
- Terminated: Terminated state specifies the time when a process is terminated

After completing every step, all the resources are used by a process, and memory becomes free.

Process Control Block(PCB)

Every process is represented in the operating system by a process control block, which is also called a task control block.



- **Process state:** A process can be new, ready, running, waiting, etc.
- **Program counter:** The program counter lets you know the address of the next instruction, which should be executed for that process.
- **CPU registers:** This component includes accumulators, index and general-purpose registers, and information of condition code.
- **CPU scheduling information:** This component includes a process priority, pointers for scheduling queues, and various other scheduling parameters.
- **Accounting and business information:** It includes the amount of CPU and time utilities like real time used, job or process numbers, etc.
- **Memory-management information:** This information includes the value of the base and limit registers, the page, or segment tables. This depends on the memory system, which is used by the operating system.

- **I/O status information:** This block includes a list of open files, the list of I/O devices that are allocated to the process, etc.

Interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention. It alerts the processor to a high priority process requiring interruption of the current working process. In I/O devices one of the bus control lines is dedicated for this purpose and is called the *Interrupt Service Routine (ISR)*.

When a device raises an interrupt at lets say process i, the processor first completes the execution of instruction i. Then it loads the Program Counter (PC) with the address of the first instruction of the ISR. Before loading the Program Counter with the address, the address of the interrupted instruction is moved to a temporary location. Therefore, after handling the interrupt the processor can continue with process i+1.

While the processor is handling the interrupts, it must inform the device that its request has been recognized so that it stops sending the interrupt request signal. Also, saving the registers so that the interrupted process can be restored in the future, increases the delay between the time an interrupt is received and the start of the execution of the ISR. This is called Interrupt Latency.

Hardware Interrupts:

In a hardware interrupt, all the devices are connected to the Interrupt Request Line. A single request line is used for all the n devices. To request an interrupt, a device closes its associated switch. When a device requests an interrupt, the value of INTR is the logical OR of the requests from individual devices.

Sequence of events involved in handling an IRQ:

1. Devices raise an IRQ.
2. Processor interrupts the program currently being executed.
3. Device is informed that its request has been recognized and the device deactivates the request signal.
4. The requested action is performed.
5. Interrupt is enabled and the interrupted program is resumed.

Handling Multiple Devices:

When more than one device raises an interrupt request signal, then additional information is needed to decide which device to be considered first. The following methods are used to decide which device to select: Polling, Vectored Interrupts, and Interrupt Nesting. These are explained as following below.

1. **Polling:**
   In polling, the first device encountered with with IRQ bit set is the device that is to be serviced first. Appropriate ISR is called to service the same. It is easy to implement but a lot of time is wasted by interrogating the IRQ bit of all devices.
2. **Vectored Interrupts:**
   In vectored interrupts, a device requesting an interrupt identifies itself directly by sending a special code to the processor over the bus. This enables the processor to identify the device that generated the interrupt. The special code can be the starting address of the ISR or where the ISR is located in memory, and is called the interrupt vector.
3. **Interrupt Nesting:**
   In this method, I/O device is organized in a priority structure. Therefore, interrupt request from a higher priority device is recognized whereas request from a lower priority device is not. To implement this each process/device (even the processor). Processor accepts interrupts only from devices/processes having priority more than it