Process Management in OS

A Program does nothing unless its instructions are executed by a CPU. A program in execution is called a process. In order to accomplish its task, process needs the computer resources.

There may exist more than one process in the system which may require the same resource at the same time  such as process mapping, process analysis, [process](https://www.geeksforgeeks.org/process-table-and-process-control-block-pcb/)improvement, process automation, and process control. By applying these tools and techniques, organizations can streamline their processes, eliminate waste, and improve productivity.

Overall, process management is a critical aspect of modern business operations and can help organizations achieve their goals and stay competitive in today’s rapidly changing marketplace. Therefore, the operating system has to manage all the processes and the resources in a convenient and efficient way.

Some resources may need to be executed by one process at one time to maintain the consistency otherwise the system can become inconsistent and deadlock may occur.

The operating system is responsible for the following activities in connection with Process Management

1. Scheduling processes and threads on the CPUs.
2. Create a child’s process identical to the parent’s.
3. Terminate a process
4. Wait for a child process to terminate
5. Change the priority of the process
6. Block the process
7. Ready the process
8. Dispatch a process
9. Suspend a process
10. Resume a process
11. Delay a process
12. Fork a process

**States of Process**

A process is in one of the following states:

1. **New:** Newly Created Process (or) being-created process.
2. **Ready:** After the creation process moves to the Ready state, i.e. the process is ready for execution.
3. **Run:** Currently running process in CPU (only one process at a time can be under execution in a single processor)
4. **Wait (or Block):** When a process requests I/O access.
5. **Complete (or Terminated):** The process completed its execution.
6. **Suspended Ready:** When the ready queue becomes full, some processes are moved to a suspended ready state
7. **Suspended Block:** When the waiting queue becomes full.

A diagram of a running process

Description automatically generated

**Context Switching:**The process of saving the context of one process and loading the context of another process is known as Context Switching. In simple terms, it is like loading and unloading the process from the running state to the ready state.

**When Does Context Switching Happen?**

1. When a high-priority process comes to a ready state (i.e. with higher priority than the running process)   
2. An Interrupt occurs   
3. User and kernel-mode switch (It is not necessary though)   
4. Preemptive CPU scheduling is used.

**Context Switch vs Mode Switch**

A mode switch occurs when the [CPU](https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/)privilege level is changed, for example when a system call is made or a fault occurs. The kernel works in more a privileged mode than a standard user task. If a user process wants to access things that are only accessible to the kernel, a mode switch must occur. The currently executing process need not be changed during a mode switch. A mode switch typically occurs for a process context switch to occur. Only the [kernel](https://www.geeksforgeeks.org/kernel-in-operating-system/)can cause a context switch.

**Sources :**

**1-https://www.geeksforgeeks.org/introduction-of-process-management/**

**2-https://www.javatpoint.com/os-process-schedulers**