**Process in OS**

A process is defined as a sequence of instructions are executed in a **predefined order.** In simple words, any program that is executed is termed as a **process.** Processes change its state as it executes and can be either new, ready, running, waiting or terminated. A process in OS is managed by the Process Control Block (PCB).

## Scope

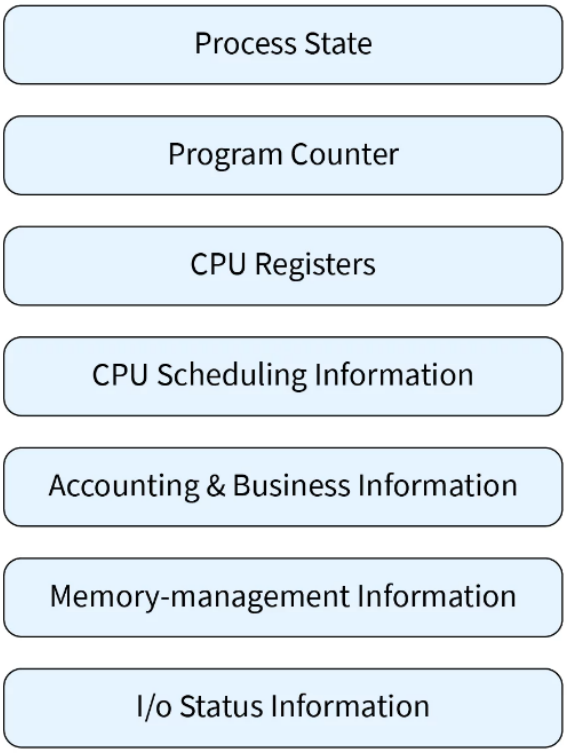
- This article covers the definition of a process in OS and its components.
- It also speaks about the working of processes in operating system and gives a deep dive into kernel space and user space and thier respective stacks.
- This article does not cover details about specific processes or how they can be troubleshooted.

## Introduction

Bill is an Engineering student in his first year of college. With a lot of assignments piling up and deadlines floating in, he realizes that he needs to do something to manage his time better. So, every night, Bill notes down the tasks that he needs to do the next day in the order of execution. Here, Bill writes a program that he would execute when the time comes. At the beginning of every day, he refers to his schedule and follows the steps in the order mentioned. This is **called a process.**

When you write a set of actions in sequential order, it is called a program. The **program, when in execution is called a process.** In a process, every course of action succeeds the previously mentioned one. A single program can host numerous processes and these processes can be run more than one time. Within an Operating System (OS), we have numerous kinds of processes.

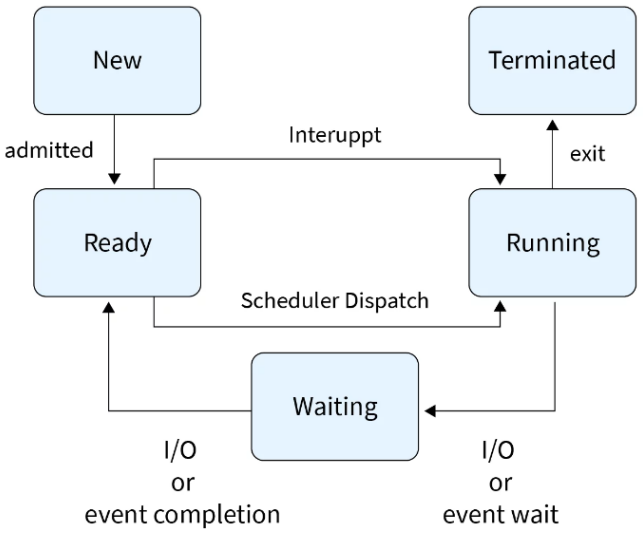
## Process Control Block



An OS serves in the creation, scheduling and termination of processes. The OS consists of Process Control Block (PCB) that helps control the functioning of processes. Every process in OS has a PCB associated with it. A PCB keeps track of processes by storing information about various things like their state, **I/O status and CPU Scheduling**.

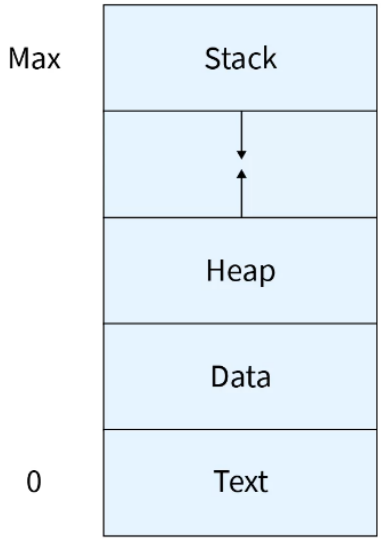
Understanding the components of a PCB will help us get to know the process information better. A PCB comprises of the following:

1. **Process ID**: An identifier that helps us in identifying and locating a process.
2. **Process state**: It identifies the state that the process is currently in. It could be a new process, ready, running, waiting or terminated.




3. **Program counter**: It holds the address of the next instruction to be executed for the process. It also stores a count of the number of instructions in the process.
4. **CPU registers**: The content of processor registers gets stored here when the process is in a running state. The different kinds of CPU registers are accumulators, index and general-purpose registers, instruction registers, and condition code registers.
5. **CPU scheduling information**: A process needs to be scheduled for execution. Based on this scheduling, it goes from ready to running. CPU Scheduling information contains process priority (to determine which process goes first), pointers for scheduling queues (to mark the order of execution), and various other scheduling parameters.
6. **Accounting and business information**: Contains details like CPU utilization, real-time used by the process, number of jobs or processes, etc.
7. **Memory-management information**: Contains the value of base and limit registers, details about the page, and segment tables. It depends on the memory system of the OS in use.
8. **I/O status information**: Comprises I/O related information including list of I/O devices allocated to the process, status, etc.

## What Process in OS Comprises of in the Memory?




**A process in OS consists of 4 important sections:**

1. **Stack** - Temporary data such as method/function arguments, return address, and local variables are stored in the stack.
2. **Heap** - Memory that is dynamically allocated to a process during its execution. Contains dynamically allocated content.
3. **Data** - This section contains the global and static variables.
4. **Text** - Consists of the current activity as reflected by the Program Counter value and the processor's registers contents.

## How do Multiple Processes Share RAM? (Single Contiguous and No Sharing Model Partition Model)

Special models are used to serve the purpose of allocating RAM to multiple processes. They are as follows: