

- In the Operating System, there are cases when you have to bring back the process that is in the running state to some other state like ready state or wait/block state.
- If the running process wants to perform some I/O operation, then you have to remove the process from the running state and then put the process in the I/O queue.
- Sometimes, the process might be using a round-robin scheduling algorithm where after every fixed time quantum, the process has to come back to the ready state from the running state.
- So, these process switchings are done with the help of Context Switching. In this blog, we will learn about the concept of Context Switching in the Operating System and we will also learn about the advantages and disadvantages of Context Switching. So, let's get started.

## What is Context Switching?

A context switching is a process that involves switching of the CPU from one process or task to another. In this phenomenon, the execution of the process that is present in the running state is suspended by the kernel and another process that is present in the ready state is executed by the CPU.

It is one of the essential features of the multitasking operating system. The processes are switched so fastly that it gives an illusion to the user that all the processes are being executed at the same time.

But the context switching process involved a number of steps that need to be followed. You can't directly switch a process from the running state to the ready state. You have to save the context of that process. If you are not saving the context of any process P then after some time, when the process P comes in the CPU for execution again, then the process will start executing from starting. But in reality, it should continue from that point where it left the CPU in its previous

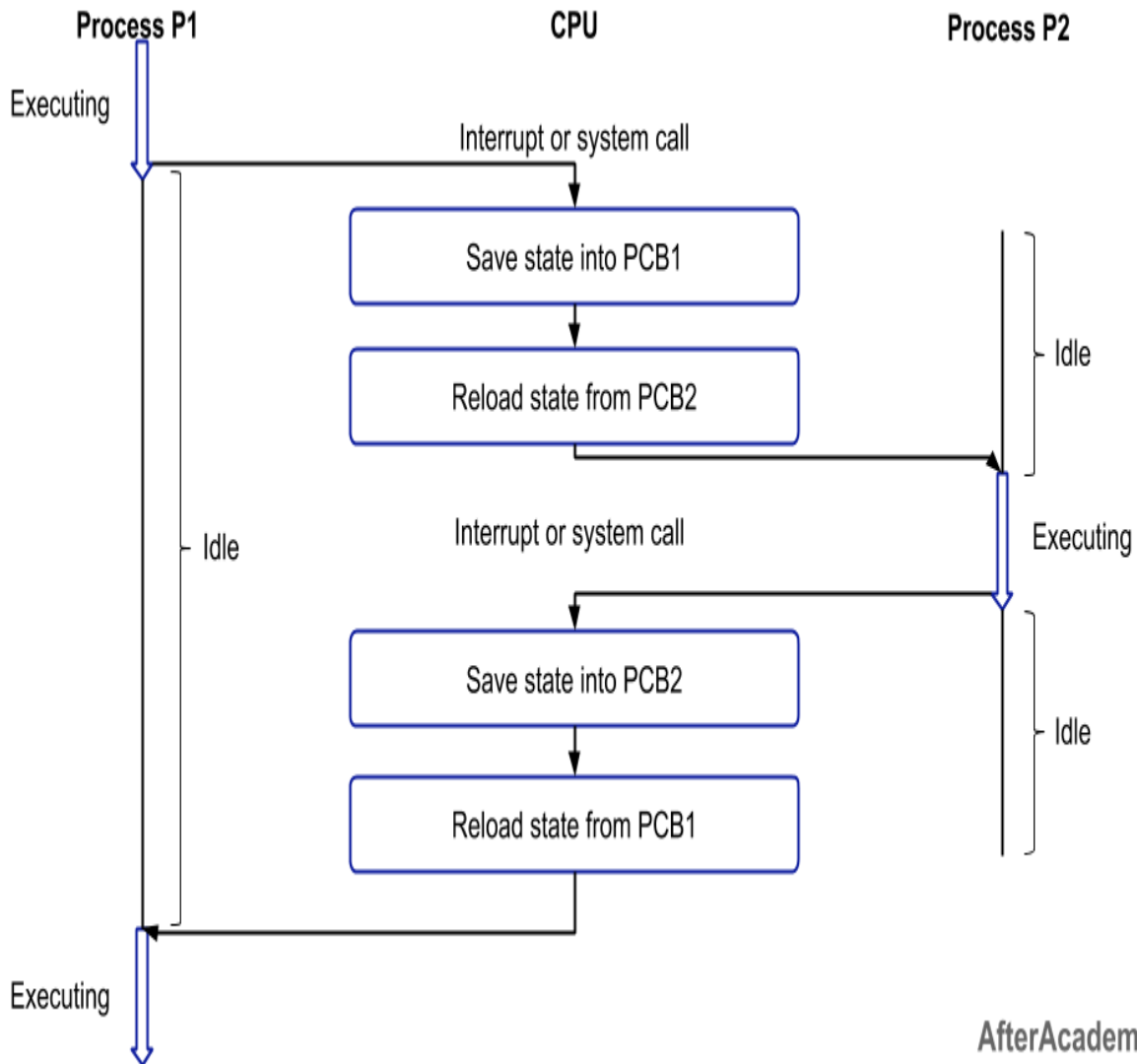
execution. So, the context of the process should be saved before putting any other process in the running state.

A context is the contents of a CPU's registers and program counter at any point in time. Context switching can happen due to the following reasons:

- When a process of high priority comes in the ready state. In this case, the execution of the running process should be stopped and the higher priority process should be given the CPU for execution.
- When an interruption occurs then the process in the running state should be stopped and the CPU should handle the interrupt before doing something else.
- When a transition between the user mode and kernel mode is required then you have to perform the context switching.

## Steps involved in Context Switching

The process of context switching involves a number of steps. The following diagram depicts the process of context switching between the two processes P1 and P2.



AfterAcademy

In the above figure, you can see that initially, the process P1 is in the running state and the process P2 is in the ready state. Now, when some interruption occurs then you have to switch the process P1 from running to the ready state after saving the context and the process P2 from ready to running state. The following steps will be performed:

1. Firstly, the context of the process P1 i.e. the process present in the running state will be saved in the Process Control Block of process P1 i.e. PCB1.
2. Now, you have to move the PCB1 to the relevant queue i.e. ready queue, I/O queue, waiting queue, etc.
3. From the ready state, select the new process that is to be executed i.e. the process P2.
4. Now, update the Process Control Block of process P2 i.e. PCB2 by setting the process state to running. If the process P2 was earlier executed by the CPU, then you can get the position of last executed instruction so that you can resume the execution of P2.
5. Similarly, if you want to execute the process P1 again, then you have to follow the same steps as mentioned above(from step 1 to 4).

For context switching to happen, two processes are at least required in general, and in the case of the round-robin algorithm, you can perform context switching with the help of one process only.

**The time involved in the context switching of one process by other is called the Context Switching Time.**

## **Advantage of Context Switching**

- Context switching is used to achieve multitasking i.e. multiprogramming with time-sharing.
- Multitasking gives an illusion to the users that more than one process are being executed at the same time.
- But in reality, only one task is being executed at a particular instant of time by a processor. Here, the context switching is so fast that the user feels that the CPU is executing more than one task at the same time.

## **The disadvantage of Context Switching**

- The disadvantage of context switching is that it requires some time for context switching i.e. the context switching time.
- Time is required to save the context of one process that is in the running state and then getting the context of another process that is about to come in the running state.
- During that time, there is no useful work done by the CPU from the user perspective. So, context switching is pure overhead in this condition.