Example of Mix Burst Time (CPU & I/O Both) | Operating System – M02 P09

This is a multipart blog article series, and in this series I am going to explain you the concepts of operating system. This article series is divided into multiple modules and this is the second module which consists of 11 articles.

In this article we will see a question in mix burst, which means that the process will access CUP and then perform some I/O operation.

**Mix Burst Time:**

* The mode is pre-emptive.
* A criterion is pre-emptive.

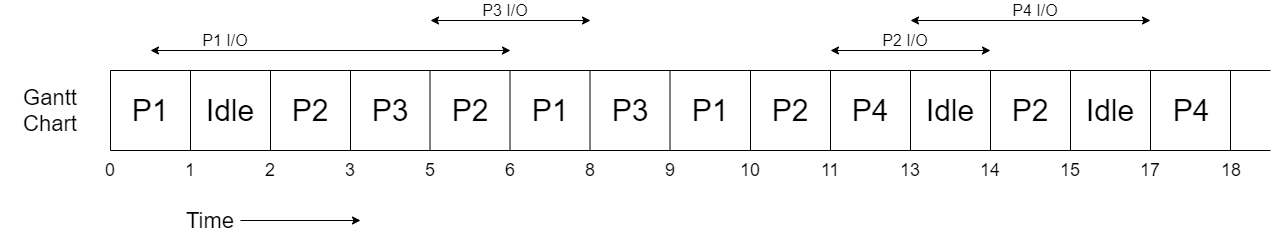
Let us understand about mix burst time scheduling with an example. We have to complete this table by solving this question by using gantt chart.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Process No. | Arrival time | Priority | CPU | I/O | CPU | Completion time |
| P1 | 0 | 2 | 1 | 5 | 3 |  |
| P2 | 2 | 3 | 3 | 3 | 1 |  |
| P3 | 3 | 1 | 2 | 3 | 1 |  |
| P4 | 3 | 4 | 2 | 4 | 1 |  |

***Lower the number higher the priority.***

Before solving this question you need to keep few points in mind.

* We have to find the completion time of every process.
* The preemption of the processes are related to the priority.



* So, at time unit 0 process P1 arrives at CPU and get executed for 1 unit of time, because after 1 unit of time it has to perform I/O operation for 5 unit of time.
* As, P1 got preempted at time unit 1 and next process P2 arrives at time unit 2 that’s why CPU remain idle for 1 unit of time.
* P2 arrive at CPU at time unit 2, but got executed for only 1 unit of time because P3 arrives at time unit 3 and have higher priority than P2.
* P3 starts its execution and got executed for 2 unit of time and then to perform I/O operation it got preempted.
* P2 and P4 are present in ready queue but P2’s priority is higher than that of P4’s priority so P2 starts its execution. When P2 executed for 1 unit for time at that time P1’s I/O operation got completed.
* So, P2 got preempted as P1 has higher priority and P1’s execution got started.
* P1 got executed for 2 units of time and then the I/O operation of P3 got completed so P1 got preempted.
* P3 got terminated after executing for 1 unit of time. And P1 again start executing, after executing for 1 unit of time it also gets terminated.
* Again P2 has higher priority so P2’s execution started and after executing for 1 unit of time P2 got preempted to perform I/O operation.
* P4 is the only process remaining in the ready queue so its execution starts and after 2 unit of time P4 get preempted to perform I/O operation.
* CPU will remain idle as P2 and P4 are busy in I/O for 1 unit of time.
* P2’s I/O is completed so it got executed for 1 unit of time and got terminated.
* Then again as only P4 is remaining and that too is busy in I/O, CPU will remain idle for 2 unit of time.
* When P4’s I/O completed it got executed for 1 unit of time and then terminated.

All the processes are now completed. Now we will find the completion time of each process.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Process No. | Arrival time | Priority | CPU | I/O | CPU | Completion time |
| P1 | 0 | 2 | 1 | 5 | 3 | 10 |
| P2 | 2 | 3 | 3 | 3 | 1 | 15 |
| P3 | 3 | 1 | 2 | 3 | 1 | 9 |
| P4 | 3 | 4 | 2 | 4 | 1 | 18 |

Some observations from the table.

* Ratio of CPU idleness = 4/18
* Ratio of CPU usage = 14/18

So, this was all about mix burst time scheduling. Hope you like it and learned something new from it.

If you have any doubt, question, queries related to this topic or just want to share something with me, then please feel free to contact me.