Deadlock Avoidance (Banker’s Algorithm) | Operating System – M04 P05

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 forth module which consists of 8 articles.

In this article we will discuss about banker’s algorithm which is a deadlock avoidance method.

**Banker’s algorithm**

* We have to tell the system that what process are coming, which process will request for which resource, how many resource and for how much time.
* All this information we have to give to the system before process arrives, then system will decide that which process is going to wait or execute and for how much time it will wait or execute.
* It is also used for deadlock detection which means it can find that deadlock can occur in future or not.

Let’s see an example to get better understanding about banker’s algorithm.

**Example:** Find the safe sequence. We have resource A = 10, resource B = 5 and resource C = 7.

*Safe sequence means that the sequence of execution process and allocation resources which will do not cause deadlock.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Process No. | Allocation of A | Allocation of B | Allocation of C | Max need of A | Max need of B | Max need of C | Current availability of A | Current availability of B | Current availability of C | Remaining need of A | Remaining need of B | Remaining need of C |
| P1 | 0 | 1 | 0 | 7 | 5 | 3 | 3 | 3 | 2 | 7 | 4 | 3 |
| P2 | 2 | 0 | 0 | 3 | 2 | 2 | 5 | 3 | 2 | 1 | 2 | 2 |
| P3 | 3 | 0 | 2 | 9 | 0 | 2 | 7 | 4 | 3 | 6 | 0 | 0 |
| P4 | 2 | 1 | 1 | 4 | 2 | 2 | 7 | 4 | 5 | 2 | 1 | 1 |
| P5 | 0 | 0 | 2 | 5 | 3 | 3 | 7 | 5 | 5 | 5 | 3 | 1 |

* Current availability: (3,3,2)
* With this current availability we can fulfill the request of P2
* Current availability: (5,3,2)
* With this current availability we can fulfill the request of P4
* Current availability: (7,4,3)
* With this current availability we can fulfill the request of P5
* Current availability: (7,4,5)
* With this current availability we can fulfill the request of P1
* Current availability: (7,5,5)
* With this current availability we can fulfill the request of P3
* Current availability: (10,5,7) = Total resources.
* Safe sequence: P2 -> P4 -> P5 -> P1 -> P3
* No deadlock will occur so it is safe.
* More than one safe sequence is possible.

So this was all about deadlock avoidance (banker’s algorithm). Hope you liked it and learned something new from it.

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