Multi-Instance Resource Allocation Graph – M04 P03

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.

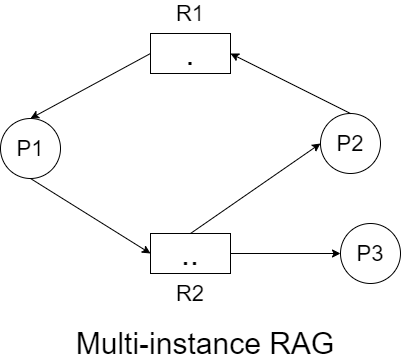
In the previous article we discussed about resource allocation graph for single instance, now in this article we are going to see about multi-instance resource allocation graph.

**Multi-instance resource allocation:**

* It means that resource has multiple instances.
* In simple words we can say that a particular resource can fulfil the need of multiple process requests.

To get a clear understanding of the topic let’s see an example and try to understand.

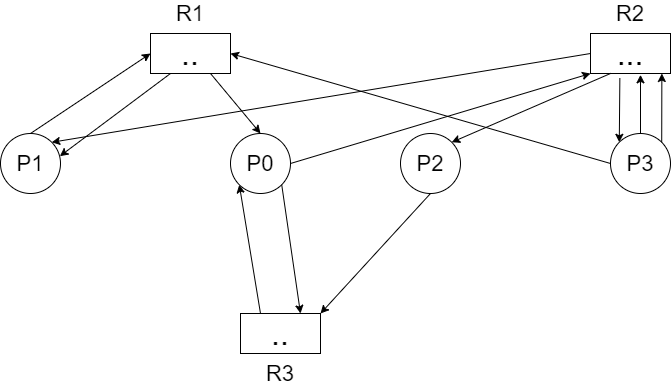
**Example:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Process No. | Allocated R1 | Allocated R2 | Request R1 | Request R2 |
| P1 | 1 | 0 | 0 | 1 |
| P2 | 0 | 1 | 1 | 0 |
| P3 | 0 | 1 | 0 | 0 |

* Current availability: (0,0)
* With the current availability we can fulfil the request of P3, because P3 is demanding nothing. So, P3 will get terminated after executing completely. Now the availability has changed as resource allocated to P3 has been released by P3.
* Current availability: (0,1)
* Now, with the current availability we can fulfil the request of P1, because it requires 1 resource of R2 and we have it. So, P1 will get terminated after getting executed completely and the resource allocated to P1 will be released.
* Current availability: (1,1)
* With this availability we can fulfil the request of P2.
* So, no deadlock is present in the system.
* This example contains circular wait but no deadlock is present, because this is multi-instance that only happen in the case of single-instance.

**Example:**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Process No. | Allocated R1 | Allocated R2 | Allocated R3 | Request R1 | Request R2 | Require R3 |
| P0 | 1 | 0 | 1 | 0 | 1 | 1 |
| P1 | 1 | 1 | 0 | 1 | 0 | 0 |
| P2 | 0 | 1 | 0 | 0 | 0 | 1 |
| P3 | 0 | 1 | 0 | 1 | 2 | 0 |

* Current availability: (0,0,1)
* With current availability we can fulfil the request of P2, it only require 1 resource of R3. It will get terminated and after executing completely.
* Current availability: (0,1,1)
* With current availability we can fulfil the request of P0, it require 1 resource of R2 and 1 resource of R3
* Current availability: (1,1,2)
* With current availability we can fulfil the request of P1 it requires 1 resource of R1.
* Current availability: (2,2,2)
* With current availability we can fulfil the request of P3 it require 1 resource of R1 and 2 resource of R2.
* No deadlock is present in this example.
* Current availability: (2,3,2), all resources are free.

So this was all about multi-instance resource allocation graph. 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 new with me, than please feel free to contact me.