**Banker’s Algorithm**

The resource-allocation-graph algorithm is not applicable to a resource-allocation system with multiple instances of each resource type. The deadlock-avoidance algorithm that we describe next is applicable to such a system but is less efficient than the resource-allocation graph scheme. This algorithm is commonly known as the banker’s algorithm. The name was chosen because the algorithm could be used in a banking system to ensure that the bank never allocated its available cash in such a way that it could no longer satisfy the needs of all its customers.

Several data structures must be maintained to implement the banker’s algorithm. These data structures encode the state of the resource-allocation system. We need the following data structures, where n is the number of processes in the system and m is the number of resource types:

**• Available**:- A vector of length m indicates the number of available resources of each type. If Available[j] equals k, then k instances of resource type Rj are available.

**• Max:-** An n × m matrix defines the maximum demand of each process. If Max[i][ j] equals k, then process Pi may request at most k instances of resource type Rj.

**• Allocation:-** An n × m matrix defines the number of resources of each type currently allocated to each process. If Allocation [i][ j] equals k, then process Pi is currently allocated k instances of resource type Rj.

**• Need:-** An n × m matrix indicates the remaining resource need of each process. If Need[i][ j] equals k, then process Pi may need k more instances of resource type Rj to complete its task. Note that Need[i][ j] equals Max[i][ j]− Allocation [i][ j].

**Safety Algorithm**

We can now present the algorithm for finding out whether or not a system is in a safe state. This algorithm can be described as follows:

1. Let Work and Finish be vectors of length m and n, respectively. Initialize Work= Available and Finish [i]=false for i = 0, 1, ..., n − 1.

2. Find an index i such that both

a. Finish[i]== false

b. Need i ≤ Work

Video Links: <https://www.youtube.com/watch?v=bYFVbzLLxfY>

<https://www.youtube.com/watch?v=-VoZvNvYt-A>

REFERENCES:

1. Galvin, Peter B., Silberchatz, A., “*Operating System Concepts*”, Addison Wesley, 8th Edition.
2. Flynn, *“Operating Systems”*, Cengage Learning.
3. Dhamdhere, D.M., "*Operating System:A Concept Based Approach*",  
   Tata Mc-Graw- Hill.
4. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-828-operating-system-engineering-fall-2012/lecture-notes-and-readings/
5. **https://www.youtube.com/watch?v=lMNrmDUJ3GY**
6. <https://computing.llnl.gov/tutorials/>
7. https://nptel.ac.in/courses/106/105/106105214
8. <https://www.guru99.com/operating-system-tutorial.html>
9. <https://www.geeksforgeeks.org/operating-systems/>