Overview

Bankers algorithm in Operating System is used to avoid deadlock and for resource allocation safely to each process in the system. As the name suggests, it is mainly used in the banking system to check whether the loan can be sanctioned to a person or not. Bankers algorithm in OS is a combination of two main algorithms: **safety algorithm** (to check whether the system is in a safe state or not) and **resource request algorithm** (to check how the system behaves when a process makes a resource request).

What is the Bankers Algorithm in OS?

A process in OS can request a resource, can use the resource, and can also release it. There comes a situation of **deadlock in OS** in which a set of processes is blocked because it is holding a resource and also requires some other resources at the same time that are being acquired by other processes. So to avoid such a situation of deadlock, we have the Bankers algorithm in Operating System.

Bankers algorithm in OS is a deadlock avoidance algorithm and it is also used to safely allocate the resources to each process within the system. It was designed by Edsger Dijkstra. As the name suggests, Bankers algorithm in OS is mostly used in the banking systems to identify whether a loan should be given to a person or not.

To understand this algorithm in detail, let us discuss a real-world scenario. Supposing there are 'n' account holders in a particular bank and the total sum of their money is 'x'. Now, if a person applies for a loan (let's say for buying a car), then the loan amount subtracted from the total amount available in the bank gives us the remaining amount and that should be greater than 'x', then only the loan will be sanctioned by the bank. It is done keeping in mind about the worst case where all the account holders come to withdraw their money from the bank at the same time. Thus, the bank always remains in a safe state.

Bankers algorithm in OS works similarly. Each process within the system must provide all the important necessary details to the operating system like upcoming processes, requests for the resources, delays, etc. Based on these details, OS decides whether to execute the process or keep it in the waiting state to avoid deadlocks in the system. **Thus, Bankers algorithm is sometimes also known as the Deadlock Detection Algorithm.**

While implementing or working with Bankers algorithm in OS, it is quite important to keep a track of three things:

1. How many resources of each type, a process can request for and it is denoted by the [MAX] array.

2. How many resources of each type, a process is currently holding or allocated and it is denoted by [ALLOCATION] array.
3. How many resources of each type are currently available within the system and it is denoted by the [AVAILABLE] array.

Example 1.

In this example, we have a process table with number of processes that contains **allocation field** (for showing the number of resources of type: A, B and C allocated to each process in the table), **max field** (for showing the maximum number of resources of type: A, B, and C that can be allocated to each process) and also, the **available field** (for showing the currently available resources of each type in the table).

Processes	Allocation	Max	Available
	A B C	A B C	A B C
P0	1 1 2	5 4 4	3 2 1
P1	2 1 2	4 3 3	
P2	3 0 1	9 1 3	
P3	0 2 0	8 6 4	
P4	1 1 2	2 2 3	

Considering the above processing table, we need to calculate the following two things:

Q.1 Calculate the need matrix? Q.2 Is the system in a safe state?

Ans.1 We can easily calculate the entries of need matrix using the formula: (Need)i = (Max)i - (Allocation)i

Process	Need

Ans.2 Let us check for safe sequence:

1. For process P0, Need = (4, 3, 2) and Available = (3, 2, 1) Clearly, the resources needed are more in number than the available ones. So, now the system will move to process the next request.

P2	6 1 2
P3	8 4 4
P4	1 1 1

2. For Process P1, Need = (2, 2, 1) and Available = (3, 2, 1) Clearly, resources needed are less than equal to the available resources within the system. Hence, request of P1 is granted.

Available = Available + Allocation$Available = Available + Allocation$ **= (3, 2, 1) + (2, 1, 2) = (5, 3, 3) (New Available)**

3. For Process P2, Need = (6, 1, 2) and Available = (5, 3, 3) Clearly, the resources needed are more in number than the available ones. So, now the system will move to process the next request.

4. For Process P3, Need = (8, 4, 4) and Available = (5, 3, 3) Clearly, the resources needed are more in number than the available ones. So, now the system will move to process the next request.

5. For Process P4, Need = (1, 1, 1) and Available = (5, 3, 3) Clearly, resources needed are less than equal to the available resources within the system. Hence, request of P4 is granted.

Available = Available + Allocation$Available = Available + Allocation$ **= (5, 3, 3) + (1, 1, 2) = (6, 4, 5) (New Available)**

6. Now again check for Process P2, Need = (6, 1, 2) and Available = (6, 4, 5) Clearly, resources needed are less than equal to the available resources within the system. Hence, request of P2 is granted. Available = Available + Allocation$Available = Available + Allocation$ = (6, 4, 5) + (3, 0, 1) = (9, 4, 6) (New Available)=(6,4,5)+(3,0,1)=(9,4,6)($NewAvailable$)

7. Now again check for Process P3, Need = (8, 4, 4) and Available = (9, 4, 6) Clearly, resources needed are less than equal to the available resources within the system. Hence, request of P3 is granted.

Available = Available + Allocation$Available = Available + Allocation$ = (9, 4, 6) + (0, 2, 0) = (9, 6, 6) (New Available)=(9,4,6)+(0,2,0)=(9,6,6)($NewAvailable$)

8. Now again check for Process P0, Need = (4, 3, 2), and Available (9, 6, 6) Clearly, the request for P0 is also granted.

Safe sequence: < P1, P4, P2, P3, P0 ><P1,P4,P2,P3,P0>

The system has allocated all the required number of resources to each process in a particular sequence. Therefore, it is proved that the system is in a safe state.

Advantages of Bankers Algorithm in OS

- Bankers algorithm in OS consists of [MAX] array attribute which indicates the maximum number of resources of each type that a process can hold. Using the [MAX] array, we can always find the need of resources for a particular process. [Need] = [MAX] - [Allocated][$Need$]=[MAX]−[$Allocated$]
- This algorithm helps in detecting and avoiding deadlock and also, helps in managing and controlling process requests of each type of resource within the system.
- Each process should provide information to the operating system about upcoming resource requests, the number of resources, delays, and about how long the resources will be held by the process before release. This is also one of the main characteristics of the Bankers algorithm.
- Various types of resources are maintained by the system while using this algorithm, that can fulfill the needs of at least one process type.
- This algorithm also consists of two other advanced algorithms for maximum resource allocation.

Disadvantages of Bankers Algorithm in OS

- Bankers algorithm in OS doesn't allow a process to change its maximum need of resources while processing.
- Another disadvantage of this algorithm is that all the processes within the system must know about the maximum resource needs in advance.
- It requires a fixed number of processes for processing and no additional process can be started in between.
- This algorithm allows all the resource requests of the processes to be granted in a fixed finite period of time, but the maximum time period is one year for allocating the resources.

Conclusion

- There comes a situation of deadlock in OS in which a set of processes is blocked because it is holding a resource and also requires some other resources at the same time that are being acquired by other processes.
- Bankers algorithm in Operating System is used to avoid such deadlocks and also, for resource allocation safely to each process in the system.
- Bankers algorithm is mostly used in the banking systems to identify whether the loan should be given to a person or not.
- There are some important data structure terms that are used to implement Bankers algorithm in OS and they are as follows: **Available, Max, Allocation, Need and Finish.**
- Bankers algorithm consists of two main algorithms used to avoid deadlock and control the processes within the system: **Safety algorithm and Resource request algorithm.**
- Safety algorithm in OS is used to mainly check whether the system is in a safe state or not.
- Resource request algorithm checks the behavior of a system whenever a particular process makes a resource request in a system. It mainly checks whether resource requests can be safely granted or not within the system.
- The disadvantage of the Bankers algorithm in OS is that it doesn’t allow a process to change its maximum need of resources while processing and all the processes within the system must know about the maximum resource needs in advance.
- Bankes algorithm in OS requires a fixed number of processes for processing and no additional process can be started in between.
- Bankers algorithm is sometimes also known as the **deadlock detection algorithm**.