KINGDOM OF SAUDI ARABIA
Ministry of Education
Al-Imam Mohammad University
College of Computer & Information Sciences

المملكة العربية السعودية
وزارة التعليم
جامعة الإمام محمد بن سعود الإسلامية
كلية علوم الحاسب والمعلومات

**First term 1442/2020**

Operating Systems (CS- 322)

Section: 173

# Banker's Algorithm

Submitted By

Abdulrahman Saleh Alghaligah (439015052)

**Instructor**

**Dr. Raza Ur Rehman Qazi**

Date: December 10, 2020.

# Table of Contents

# 1.Introduction:

This report describes Banker's algorithm and gives a snapshot output of the code that I will provide.

# 2.What is Banker's algorithm?

It is a deadlock avoidance algorithm that determine a loan can be granted or not. 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 threads 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.
- **Max:** An n x m matrix defines the maximum demand of each thread.
- **Allocation:** An n x m matrix defines the number of resources of each currently allocated to each thread.
- **Need :** An n x m matrix indicates the remaining resources need of each thread.

# 3.Safety Algorithm:

We can now present the algorithm for finding out whether 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. Needi ≤ Work
If no such i exists, go to step 4.

3. Work = Work + Allocation
Finish[i] = true
Go to step 2.

4. If Finish[i] == true for all i, then the system is in a safe state.

This algorithm may require an order of × n2 operations to determine whether a state is safe.

## 4.Resource-Request Algorithm:

Next, we describe the algorithm for determining whether requests can be safely granted.

This algorithm can be described as follows:

1.If Requesti ≤ Needi, go to step 2. Otherwise, raise an error condition, since the thread has exceeded its maximum claim.

2. If Requesti ≤ Available, go to step 3. Otherwise, Ti must wait, since the resources are not available.

3. Have the system pretend to have allocated the requested resources to thread Ti by modifying the state as follows:
Available = Available–Requesti
Allocationi = Allocationi + Requesti
Needi = Needi –Requesti

## 5.Output:

In this section I will provide a snapshot output of two examples, first one is the required one from the project instructions file , and the second one is from the text book.

### Example:1:

**Sample input file**

```
5

4

0 0 1 2
1 0 0 0
1 3 5 4
0 6 3 2
0 0 1 4

0 0 1 2
1 7 5 0
2 3 5 6
0 6 5 2
0 6 5 6

1 5 2 0

1:0 4 2 0
```

**Snapshot:**

```
There are 5 processes in the system.

There are 4 resource types.

The Allocation Matrix is...
   A B C D
0: 0 0 1 2
1: 1 0 0 0
2: 1 3 5 4
3: 0 6 3 2
4: 0 0 1 4

The Max Matrix is...
   A B C D
0: 0 0 1 2
1: 1 7 5 0
2: 2 3 5 6
3: 0 6 5 2
4: 0 6 5 6

The Need Matrix is...
   A B C D
0: 0 0 0 0
1: 0 7 5 0
2: 1 0 0 2
3: 0 0 2 0
4: 0 6 4 2

The Available vector is...
A B C D
1 5 2 0

THE SYSTEM IS IN A SAFE STATE!

The Request vector is...
  A B C D
1:0 4 2 0

THE REQUEST CAN BE GRANTED!

The Available vector is...
A B C D
1 1 0 0

------------------------------------------------------------------------
BUILD SUCCESS
```

## Example:2

**Sample input file**

```
5

3

0 1 0
2 0 0
3 0 2
2 1 1
0 0 2

7 5 3
3 2 2
9 0 2
2 2 2
4 3 3

3 3 2

4:3 3 0
```

**Snapshot:**

```
There are 5 processes in the system.

There are 3 resource types.

The Allocation Matrix is...
   A B C
0: 0 1 0
1: 2 0 0
2: 3 0 2
3: 2 1 1
4: 0 0 2

The Max Matrix is...
   A B C
0: 7 5 3
1: 3 2 2
2: 9 0 2
3: 2 2 2
4: 4 3 3

The Need Matrix is...
   A B C
0: 7 4 3
1: 1 2 2
2: 6 0 0
3: 0 1 1
4: 4 3 1

The Available vector is...
A B C
3 3 2

THE SYSTEM IS IN A SAFE STATE!

The Request vector is...
  A B C
4:3 3 0

THE REQUEST CAN NOT BE GRANTED!

The Available vector is...
A B C
0 0 2

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```

## Reference:


- Silberschatz, A., Gagne, G., & Galvin, P. B. (2018). *Operating System Concepts, 10e EPUB Reg Card Abridged Print Companion Set* (10th ed.). Wiley.