FACULTY OF ENGINEERING

AIN SHAMS UNIVERSITY

CSE 223 : OPERATING SYSTEMS

BANKER’S ALGORITHM

DECEMBER 7, 2018
AIN SHAMS UNIVERSITY – FACULTY OF ENGINEERING
1 Al-Sarayyat st, Abbassiya, Cairo 11517, Egypt

Banker's Algorithm

Submitted by:

Abdelrahman Ibrahim ELGhamry
Ghamry98@hotmail.com
16P3043@eng.asu.edu.eg

Hossam ELDin Khaled Mohmed
hossampen97@gmail.com
16P3025@eng.asu.edu.eg

Abdelrahman Amr Issawi
aid-issawi@hotmail.com
16P6001@eng.asu.edu.eg

Submitted to:

Prof. Dr. Gamal Abdelshafy

December 2018

A report for Operating Systems Course codded CSE223 with the requirements
of Ain Shams University

Table of Contents

1.0 BRIEF DESCRIPTION

- This project is a console application providing the implementation of Banker's algorithm.
- Several processes request and release resources randomly.
- The algorithm will consider requests from *n* processes for *m* resources types.
- The system takes the available amount of each resource from the user at the beginning.
- The algorithm will grant a request only if it leaves the system in a safe state.
- A request that leaves the system in an unsafe state will be denied.
- Processes will continually loop requesting and releasing resources from the system.
- The processes request and then release random numbers of resources, which are bounded by their respective values in the need array.

2.0 IMPLEMENTATION

Methods:

1. **init()**
   - Void method that initiates the Available, Max, Need, Allocation and Finish arrays.
   - Prints the Max and Allocation arrays.
   - Calls the function start() to start the algorthm.

2. **start()**
   - Void method that contains the main loop of the program.
   - It iterates till all processes finish their work.
   - In each iteration, it selects a random process to request a random part of its needs, checks if it's a valid request and starts execution.
     It also calls randomRelease() method, to release random part of a random process allocation.

3. **request(int p)**
   - Void methods that generates a random request for the process p.
   - The random resources requested have a maximum of the process p need.

4. **canBeSatisfied()**
   - Boolean method that checks if the requested resources can be fulfilled within the currently available resources or not.
   - Returns the result of the check as a boolean value.

5. **isSafe()**
    - Boolean method checks if there is a sequence to fulfill all processes' needs.
    - Create a temporary available array used in the process of finding a sequence without modifying the original available array.
    - Return true if the system is in safe state, returns false otherwise.

6. **check(int p, int[] availableTemp)**
    - Boolean method checks if the need of the process p can be satisfied with the temporary available array availableTemp.
    - Called by isSafe method while finding a sequence to fulfill the needs of all processes.
    - Returns true if the availableTemp array can provide enough resources for the need of the process p, returns false otherwise.

7. **allocate(int p)**
    - Void method used to allocate the requested resources to the system by a global variable reqArr.
    - It basically subtracts the reqArr from the available array and adds the reqArr to the allocation array of the process p.
    - It also updates the need of the process p.

8. **revAllocate(int p)**
    - Void method used to reverse the allocation of a denied request.
    - It basically adds the reqArr from the available array and subtracts the reqArr to the allocation array of the process p.
    - It also updates the need of the process p.

9. **updateFinish(int p)**
    - Void method that checks if the passed process had finished its working or not.
    - Updates the finish array with the result.

10. **isAllocationEmpty()**
    - Boolean method that checks if the Allocation array is empty or not.
    - Returns the result of the check in a Boolean value.

11. **isNeedEmpty()**
    - Boolean method that checks if the Need array is empty or not.
    - Returns the result of the check in a Boolean value.

12. **release(int p)**
    - Void method that releases all allocated resources to the process p.
    - It achieves that by calling releaseResource() method m times which is the number of resources the system.

13. releaseResource(int p, int m)

- Void method that releases the allocated part of the given resource from the given process.

14. randomRelease()

- Void method that selects a random process and a random resource.
- Then it calls releaseResource() method to start releasing the allocated part of the selected resource from the selected process.

15. display(int p)

- Void methods that prints the Max and Need arrays of a given process.
- It also prints the currently available resources and the names of the finished processes.

16. printMaxArray()

- Prints the Max 2D array to the console.

17. printAllocationArray()

- Prints the Allocation 2D array to the console.

18. printNeedArray()

- Prints the Need 2D array to the console.

19. printAvailableArray()

- Prints the available resources to the console.

3.0 TEST CASES

3.1 Test Case (1) :

```
How Many Processes and Resources?
  3 4
  Fill Available Matrix (1x4):
  4 1 9 0

  The Max Array:
  P0 : 3 1 6 0
  P1 : 1 1 9 0
  P2 : 3 1 6 0
  Allocation Array:
  P0 : 0 0 0 0
  P1 : 0 0 0 0
  P2 : 0 0 0 0

  Finished Processes:

  --------------------------------------------------
  P1 max:
  1 1 9 0
  P1 need:
  1 1 9 0
  Available Resources:
  4 1 9 0
  P1 requested:
  1 1 3 0
  Valid Request: true
  Request Accepted...
  P1 released: 1 of R0

  Finished Processes:

  --------------------------------------------------
  P2 max:
  3 1 6 0
  P2 need:
  3 1 6 0
  Available Resources:
  4 0 6 0
  P2 requested:
  0 0 5 0
  Valid Request: false
  This Request Will Lead The System To Unsafe State
  The Need Array After Accepting the Request:
  P0 : 3 1 6 0
  P1 : 0 0 6 0
  P2 : 3 1 1 0
  The Available Array After Accepting The Request:
  Available Resources:
  4 0 1 0
  The Allocation Array After Accepting The Request:
  Allocation Array:
  P0 : 0 0 0 0
  P1 : 0 1 3 0
  P2 : 0 0 5 0

  Can't Fullfill The Needs of Any Process
  Unsafe State, Request Rejected...
  P1 released: 3 of R2

  Finished Processes:

  --------------------------------------------------
```

```
P2 max:
3 1 6 0
P2 need:
3 1 6 0
Available Resources:
4 0 9 0
P2 requested:
2 1 4 0
Valid Request: false
No Enough Resources, Request Rejected...
P1 released: 1 of R1

Finished Processes:

--------------------------------------------------
P0 max:
3 1 6 0
P0 need:
3 1 6 0
Available Resources:
4 1 9 0
P0 requested:
2 1 2 0
Valid Request: true
Request Accepted...
P0 released: 2 of R2

Finished Processes:

--------------------------------------------------
P1 max:
1 1 9 0
P1 need:
0 0 6 0
Available Resources:
2 0 9 0
P1 requested:
0 0 2 0
Valid Request: true
Request Accepted...
P0 released: 1 of R1

Finished Processes:

--------------------------------------------------
P0 max:
3 1 6 0
P0 need:
1 0 4 0
Available Resources:
2 1 7 0
P0 requested:
1 0 0 0
Valid Request: true
Request Accepted...
P0 released: 3 of R0

Finished Processes:

--------------------------------------------------
```

```
P0 max:
3 1 6 0
P0 need:
0 0 4 0
Available Resources:
4 1 7 0
P0 requested:
0 0 2 0
Valid Request: true
Request Accepted...
P0 released: 2 of R2

Finished Processes:

--------------------------------------------------
P0 max:
3 1 6 0
P0 need:
0 0 2 0
Available Resources:
4 1 7 0
P0 requested:
0 0 2 0
Valid Request: true
Request Accepted...
Process P0 Finished Working Successfully...
P1 released: 2 of R2

Finished Processes:
P0
--------------------------------------------------
P1 max:
1 1 9 0
P1 need:
0 0 4 0
Available Resources:
4 1 9 0
P1 requested:
0 0 3 0
Valid Request: true
Request Accepted...
P1 released: 3 of R2

Finished Processes:
P0
--------------------------------------------------
P1 max:
1 1 9 0
P1 need:
0 0 1 0
Available Resources:
4 1 9 0
P1 requested:
0 0 1 0
Valid Request: true
Request Accepted...
Process P1 Finished Working Successfully...

Finished Processes:
P0 P1
--------------------------------------------------
P2 max:
3 1 6 0
P2 need:
3 1 6 0
Available Resources:
4 1 9 0
P2 requested:
1 1 4 0
Valid Request: true
Request Accepted...
P2 released: 1 of R0

Finished Processes:
P0 P1
--------------------------------------------------
P2 max:
3 1 6 0
P2 need:
2 0 2 0
Available Resources:
4 0 5 0
P2 requested:
2 0 0 0
Valid Request: true
Request Accepted...
P2 released: 1 of R1

Finished Processes:
P0 P1
--------------------------------------------------
P2 max:
3 1 6 0
P2 need:
0 0 2 0
Available Resources:
2 1 5 0
P2 requested:
0 0 1 0
Valid Request: true
Request Accepted...
P2 released: 5 of R2

Finished Processes:
P0 P1
--------------------------------------------------
P2 max:
3 1 6 0
P2 need:
0 0 1 0
Available Resources:
2 1 9 0
P2 requested:
0 0 1 0
Valid Request: true
Request Accepted...
Process P2 Finished Working Successfully...

--------------------------------------------------
All Processes Finished Working Successfully
```

3.2 Test Case (2) :

```
How Many Processes and Resources?
5 3
Fill Available Matrix (1x3):
2 9 7

The Max Array:
P0 : 0 6 6
P1 : 2 9 0
P2 : 1 3 5
P3 : 1 2 1
P4 : 2 4 2
Allocation Array:
P0 : 0 0 0
P1 : 0 0 0
P2 : 0 0 0
P3 : 0 0 0
P4 : 0 0 0

Finished Processes:

--------------------------------------------------
P4 max:
2 4 2
P4 need:
2 4 2
Available Resources:
2 9 7
P4 requested:
1 2 0
Valid Request: true
Request Accepted...
P4 released: 1 of R0

Finished Processes:

--------------------------------------------------
P0 max:
0 6 6
P0 need:
0 6 6
Available Resources:
2 7 7
P0 requested:
0 3 0
Valid Request: true
Request Accepted...
P0 released: 3 of R1

Finished Processes:

--------------------------------------------------
P2 max:
1 3 5
P2 need:
1 3 5
Available Resources:
2 7 7
Valid Request: true
Request Accepted...
P4 released: 2 of R1

Finished Processes:

--------------------------------------------------
P3 max:
1 2 1
P3 need:
1 2 1
Available Resources:
1 9 3
P3 requested:
1 0 0
Valid Request: true
Request Accepted...
P2 released: 1 of R0

Finished Processes:

--------------------------------------------------
P0 max:
0 6 6
P0 need:
0 3 6
Available Resources:
1 9 3
P0 requested:
0 0 6
Valid Request: false
No Enough Resources, Request Rejected...
P3 released: 1 of R0

Finished Processes:

--------------------------------------------------
 P4 max:
 2 4 2
 P4 need:
 1 2 2
 Available Resources:
 2 9 3
 P4 requested:
 1 0 1
 Valid Request: true
 Request Accepted...
 P2 released: 4 of R2

 Finished Processes:

 -------------------------------------------------
```

```
P3 max:
1 2 1
P3 need:
0 2 1
Available Resources:
1 9 6
P3 requested:
0 2 1
Valid Request: true
Request Accepted...
Process P3 Finished Working Successfully...
P4 released: 1 of R0

Finished Processes:
P3
--------------------------------------------------
P2 max:
1 3 5
P2 need:
0 3 1
Available Resources:
2 9 6
P2 requested:
0 2 1
Valid Request: true
Request Accepted...
P2 released: 2 of R1

Finished Processes:
P3
--------------------------------------------------
P4 max:
2 4 2
P4 need:
0 2 1
Available Resources:
2 9 5
P4 requested:
0 2 1
Valid Request: true
Request Accepted...
Process P4 Finished Working Successfully...
P2 released: 1 of R2

Finished Processes:
P3 P4
--------------------------------------------------
P1 max:
2 9 0
P1 need:
2 9 0
Available Resources:
2 9 7
P1 requested:
0 2 0
Valid Request: true
Request Accepted...
P1 released: 2 of R1

Finished Processes:
P3 P4
--------------------------------------------------
P0 max:
0 6 6
P0 need:
0 3 6
Available Resources:
2 9 7
P0 requested:
0 3 4
Valid Request: true
Request Accepted...
P0 released: 4 of R2

Finished Processes:
P3 P4
--------------------------------------------------
P2 max:
1 3 5
P2 need:
0 1 0
Available Resources:
2 6 7
P2 requested:
0 1 0
Valid Request: true
Request Accepted...
Process P2 Finished Working Successfully...
P0 released: 3 of R1

Finished Processes:
P2 P3 P4
--------------------------------------------------
P0 max:
0 6 6
P0 need:
0 0 2
Available Resources:
2 9 7
P0 requested:
0 0 2
Valid Request: true
Request Accepted...
Process P0 Finished Working Successfully...

Finished Processes:
P0 P2 P3 P4
--------------------------------------------------
P1 max:
2 9 0
P1 need:
2 7 0
Available Resources:
2 9 7
P1 requested:
1 3 0
Valid Request: true
Request Accepted...
P1 released: 1 of R0

Finished Processes:
P0 P2 P3 P4
--------------------------------------------------
```

```
P1 max:
2 9 0
P1 need:
1 4 0
Available Resources:
2 6 7
P1 requested:
0 3 0
Valid Request: true
Request Accepted...
P1 released: 6 of R1

Finished Processes:
P0 P2 P3 P4
--------------------------------------------------
P1 max:
2 9 0
P1 need:
1 1 0
Available Resources:
2 9 7
P1 requested:
0 1 0
Valid Request: true
Request Accepted...
P1 released: 1 of R1

Finished Processes:
P0 P2 P3 P4
--------------------------------------------------
P1 max:
2 9 0
P1 need:
1 0 0
Available Resources:
2 9 7
P1 requested:
1 0 0
Valid Request: true
Request Accepted...
Process P1 Finished Working Successfully...

--------------------------------------------------
All Processes Finished Working Successfully
```