Contents

1. Introduction

As part of this programming assignment, the banker's algorithm is simulated using JAVA. The program when run requires mandatory user inputs such as number of processes, number of resources, current allocation matrix, maximum resource matrix, and available vector (A).

The report documented here captures the different aspects of the programming assignment in sections. The first section walks through the design of the program. The steps required to build, run, and terminate the programs are covered in the next few sections. Scope and conclusion are covered in the last 2 sections.

2. Design

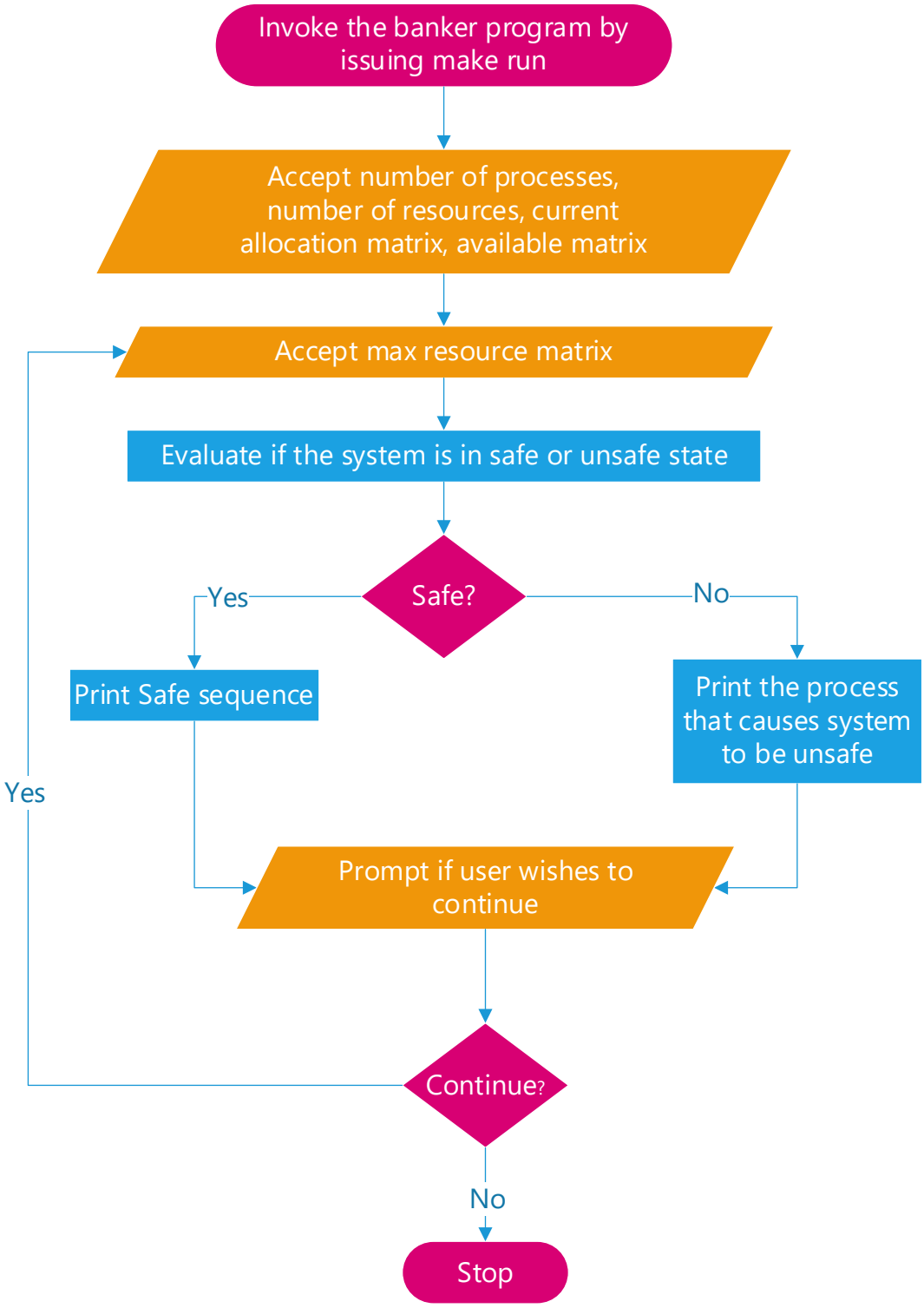
A high-level design of this program is explained in the below flow chart and the subsections that follow.

2.1. Banker Algorithm Simulation

The program prompts the user to input the number of processes, number of resources, current allocation matrix, and Available vector for a given system. The program then prompts the user to enter the max resource matrix. It then kicks off the method (check safe) that evaluates whether the system is in a safe or un-safe state. If the system is in a safe state it displays the safe sequence of processes. If the system leads to an un-safe state, then it displays the process which leads to an un-safe state.

The check safe function implementation is as follows. For every process that is not complete, the need matrix is evaluated by subtracting its max resource matrix and the current allocation matrix. If the need matrix is less than or equal to the available vector, then the process is marked as complete and added to the safe list. The resources that were allocated to the process are released and added to the available vector. If the need matrix is greater than the available matrix, then the system is in an unsafe state. The process that causes the system to be in an unsafe state is displayed. Once the system is evaluated to be safe or unsafe, the program prompts if the user wishes to continue by entering a new max resource matrix. The program completes if the user wishes to discontinue, else the new max resource matrix is evaluated for every process. Each operation throughout the program is logged into the output file called Output.txt

2.2.Flowchart



3. Files

3.1.Included as part of the submission

Filename	Description
src/Bankers.java	Bankers algorithm simulator source code
Makefile	Makefile to build the program
COEN-283 Programming Assignment #3.pdf	Design document for the programming assignment

3.2.Files that get generated after build/run

Filename	Description
src/Bankers.class	An executable that gets generated after issuing a build command `make all`
Output.txt	Output File generated to stores the program results

4. Building the program

A Makefile is provided as part of the software package. To build the program, use the below command line.

```
make all
```

```
[shivani@Shivanis-MacBook-Pro coen-283-pgm3 % make all
 ------------------------------------------------
 Building Bankers Simulation program for COEN-283 class
 mkdir -p bin
 javac -d bin src/Bankers.java
 Build successful
 © Property of Shivani Sanjay Tergaonkar
 ------------------------------------------------
```

In the case of a clean build, after an incremental change on the source code, it may be required to clean up the previously built object. In such cases, use the below command before the build.

```
make clean
```

```
[shivani@Shivanis-MacBook-Pro coen-283-pgm3 % make clean
 ---------------------------------------------
 Deleting object files
 rm -rf bin
 Object files deleted
```

5. Running the program

Once the program is built, the executable '`Bankers.class`' can be invoked directly from the command line. A user input prompt to enter the number of segments pops up as shown below.

```
make run
 ------------------------------------------------
[shivani@Shivanis-MacBook-Pro coen-283-pgm3 % make run
 ------------------------------------------------
 Running Bankers Simulation program
 java -classpath bin Bankers
 Please enter number of processes:
```

6. Verifying the program

6.1.Console Output for the system in a safe state.

```
-----------------------------------------------
shivani@Shivanis-MacBook-Pro coen-283-pgm3 % make run
-----------------------------------------------
Running Bankers Simulation program
java -classpath bin Bankers
Please enter number of processes:
3
Please enter number of resources:
1
Please enter the Allocation matrix:
3
2
2
Please enter the Available matrix:
3
Please enter the Max resource matrix:
9
4
7

Process 1 is running

Available resources are:
5
Process 1 has completed

Process 2 is running

Available resources are:
7
Process 2 has completed

Process 0 is running

Available resources are:
10
Process 0 has completed

Following is the SAFE Sequence
P1 P2 P0

Do you want to continue:
N
Bankers Simulation Program completed!
-----------------------------------------------
```

6.2. Output File generated for the system in a safe state.

```
Banker algorithm
================================================================================
Total number of processes is: 3
================================================================================

Total number of resources is: 1
================================================================================

Allocated matrix is :
3
2
2
================================================================================

Available matrix is :
3
================================================================================


Max resource matrix is :
9
4
7
================================================================================

Process 1 is running

Available resources are:

5

Process 1 has completed
================================================================================

Process 2 is running

Available resources are:

7

Process 2 has completed
================================================================================

Process 0 is running

Available resources are:

10

Process 0 has completed
================================================================================

Following is the SAFE Sequence
P1
P2
P0
================================================================================

Do you want to continue:
N
================================================================================
```

6.3. Console Output for the system in an unsafe state.

```
Please enter number of processes:
4
Please enter number of resources:
5
Please enter the Allocation matrix:
1
0
2
1
1
2
0
1
1
0
1
1
0
1
0
1
1
1
1
0
Please enter the Available matrix:
0
0
2
1
1
Please enter the Max resource matrix:
1
1
2
1
3
2
2
2
1
0
2
1
3
1
0
1
1
2
2
1

Process 3 is running

Available resources are:
1 1 3 2 1
Process 3 has completed

Process 2 is running

Available resources are:
2 2 3 3 1
Process 2 has completed

Process 1 is running

Available resources are:
4 2 4 4 1
Process 1 has completed

The System is UnSafe for process 0


Do you want to continue:
N
Bankers Simulation Program completed!
-----------------------------------------------
```

6.4.Output File generated for the system in an unsafe state.

```
Banker algorithm
================================================================================
Total number of processes is: 4
================================================================================

Total number of resources is: 5
================================================================================

Allocated matrix is :
1 0 2 1 1
2 0 1 1 0
1 1 0 1 0
1 1 1 1 0
================================================================================

Available matrix is :
0 0 2 1 1
================================================================================


Max resource matrix is :
1 1 2 1 3
2 2 2 1 0
2 1 3 1 0
1 1 2 2 1
================================================================================

Process 3 is running

Available resources are:

1 1 3 2 1

Process 3 has completed
================================================================================

Process 2 is running

Available resources are:

2 2 3 3 1

Process 2 has completed
================================================================================

Process 1 is running

Available resources are:

4 2 4 4 1

Process 1 has completed
================================================================================

The System is UnSafe for process 0
================================================================================

Do you want to continue:
N
================================================================================
```

7. Conclusion

This exercise gives a good insight into how a banker algorithm can be used to determine if a system is in a safe state or not. This also reinforces the concepts of deadlocks that may arise when multiple processes run that may require the same resources at the same time.