

Universidad San Francisco

Sistemas Operativos

Adrián Duque y Pablo Llanes

Proyecto #3

Read Me

Implementar el “Banker’s Algorithm” con una aplicación que acepta como input los siguientes

parámetros:

1. Numero de procesos
2. Numero de tipos de recursos
3. Número total de cada tipo de recurso que el sistema tiene
4. Número máximo de cada tipo de recurso que cada proceso puede pedir
5. Número actual de recursos de cada tipo que cada proceso ha sido asignado por el sistema

Basándose en estos parámetros, tu implementación debe determinar si el estado del sistema es seguro o inseguro

1.- Descripción de Ejecución

Se adjunta junto con el código en java un archivo llamado test.txt, el mismo que consta de varios ejemplos que pueden ser ejecutados en el programa para comprobar su funcionalidad. A continuación, se va a mostrar una secuencia de ejecución esperada.

- Ingresar en primer lugar el numero de procesos y el número de recursos.
- Ingresar el vector de numero total de recursos disponibles por proceso, puede hacerse de uno en uno o como un vector
- Ingresar ya sea de elemento en elemento o toda la matriz de valores máximos
- Ingresar la matriz de recursos Allocated con la misma idea que el caso anterior

```
Banker x
/usr/lib/jvm/java-1.8.0-openjdk/bin/java ...
Please type the following values
Number of processes
5
Number of resource types
4
Number of total resources
6 3 4 3
Max number of each resource type
6 2 1 1
0 2 1 2
3 2 1 0 |
1 1 1 1
2 1 1 1
Number of allocated resources by the system
3 0 1 1
0 1 0 0
1 1 1 0
1 1 0 1
0 0 0 0
```

2.- Descripción de Resultados

Se muestran todas las tablas solicitadas, como lo son: los recursos disponibles y la matriz de necesidad, a partir de los cuales se determina la safe sequence.

```
Total system resources are:
A  B  C  D
6  3  4  3

Available system resources are:
A  B  C  D
1  0  2  1

Processes (maximum resources):
    A  B  C  D
P1  6  2  1  1
P2  0  2  1  2
P3  3  2  1  0
P4  1  1  1  1
P5  2  1  1  1

Processes (currently allocated resources):
    A  B  C  D
P1  3  0  1  1
P2  0  1  0  0
P3  1  1  1  0
P4  1  1  0  1
P5  0  0  0  0

Processes (possibly needed resources):
    A  B  C  D
P1  3  2  0  0
P2  0  1  1  2
P3  2  1  0  0
P4  0  0  1  0
P5  2  1  1  1

Determining safe sequence
(P4, P2, P3, P1, P5)
```

3.- Referencias Bibliográficas

- <https://www.geeksforgeeks.org/program-bankers-algorithm-set-1-safety-algorithm/>