Project 6: 银行家算法

Chentao Wu吴晨涛
Professor
Dept. of CSE, SJTU
wuct@cs.sjtu.edu.cn

课程目标

- 编写程序实现银行家算法，支持客户请求资源和释放资源
  - 当请求使系统处于安全状态时，银行家会批准请求
  - 当请求使系统处于不安全状态时，银行家会拒绝请求

8.6.3.1 Safety Algorithm

We can now present the algorithm for finding out whether or not 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 $i = 0, 1, \ldots, n - 1$.
2. Find an index i such that both
   a. ***Finish***[i] == ***false***
   b. $Need_i \leq Work$

   If no such i exists, go to step 4.
3. $Work = Work + Allocation_i$
   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 $m \times n^2$ operations to determine whether a state is safe.

全局变量声明

- available数组：长度为m的向量，表示每种资源的可用实例数量
- maximum数组：n*m的矩阵，表示每个进程的最大需求
- allocation数组：n*m的矩阵，表示每个进程现在分配的每种资源类型的实例数量
- need数组：n*m的矩阵，表示每个进程还需要的剩余资源

```c
#define NUMBER_OF_CUSTOMERS 5
#define NUMBER_OF_RESOURCES 4
/* the available amount of each resource */
int available[NUMBER_OF_RESOURCES];
/*the maximum demand of each customer */
int maximum[NUMBER_OF_CUSTOMERS][NUMBER_OF_RESOURCES];
/* the amount currently allocated to each customer */
int allocation[NUMBER_OF_CUSTOMERS][NUMBER_OF_RESOURCES];
/* the remaining need of each customer */
int need[NUMBER_OF_CUSTOMERS][NUMBER_OF_RESOURCES];
/* user request resources */
int request_resources(int customer_num, int request[]);
/* user release resources */
void release_resources(int customer_num, int release[]);
```

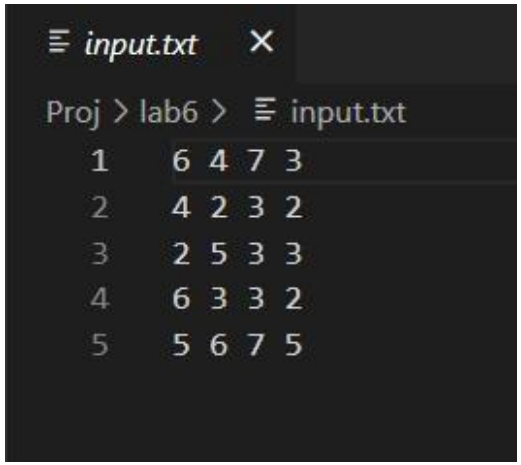
全局变量初始化

- available数组：从命令行读取，e.g. ./banker 5 6 7 8
- maximum数组：从input.txt读取
- allocation数组：全0
- need数组：与maximum数组一致

Your program will initially read in a file containing the maximum number of requests for each customer. For example, if there are five customers and four resources, the input file would appear as follows:

```
6,4,7,3
4,2,3,2
2,5,3,3
6,3,3,2
5,6,7,5
```

where each line in the input file represents the maximum request of each resource type for each customer. Your program will initialize the `maximum` array to these values.



测试

- ./banker 5 6 7 8：available初始化为(5, 6, 7, 8)
- RQ 0 1 1 1 2：用户0请求资源(1, 1, 1, 2)
- RL 0 1 0 0 0：用户0释放资源(1, 0, 0, 0)
- *：显示当前所有数组的值，exit：退出程序

```
>*
available array is:
6 6 7 8
maximum matrix is:
6 4 7 3
4 2 3 2
2 5 3 3
6 3 3 2
5 6 7 5
allocation matrix is:
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
need matrix is:
6 4 7 3
4 2 3 2
2 5 3 3
6 3 3 2
5 6 7 5
```

```
>RQ 0 3 3 3 3
Successfully allocate the resources!
>RQ 1 2 2 2 2
The state is not safe!
>RL 0 5 5 5 5
0 customer doesn't have this much resources!
>RL 0 1 2 1 2
Successfully release the resources!
>*
available array is:
4 5 5 7
maximum matrix is:
6 4 7 3
4 2 3 2
2 5 3 3
6 3 3 2
5 6 7 5
```

```
allocation matrix is:
2 1 2 1
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
need matrix is:
4 3 5 2
4 2 3 2
2 5 3 3
6 3 3 2
5 6 7 5
>quit
```

作业及评分

自行阅读课本第八章的 Programming Projects 部分，并完成以下任务，完成后共计10分。

- （课本习题 8分）编写程序实现银行家算法
  - (2分) 初始化全局变量，available数组从命令行读取，allocation矩阵初始为0，maximum和need矩阵从input.txt文件读取
  - (2分) 指令*打印 available, maximum, allocation 和 need 数组的值
  - (2分) 支持指令 RQ 请求资源
  - (2分) 支持指令 RL 释放资源

- （报告 2分）做一个简单的报告解释你的代码，报告建议不超过2页（防内卷）。