ISE 302–Operating Systems
2016–2017, Homework 3
07.12.2016

Introduction

You will develop a simulation for the Banker's algorithm **("one type of resource" version)**. It will evaluate given state of the processes and resources to decide whether it is safe or unsafe. It will also allow user to evaluate new requests and to print execution order.

Input

Program input is a text file that represents "Max_Request" and "Has" vectors as well as number of resources. First line of the file is an integer value which is the total amount of resources. After that, each line in the document is the resource values of a process. First value is for the allocated request ("Has") and the second one is for the maximum request ("Max_Request"). These values are separated by space characters.

A sample input is provided. There are 3 processes in this input. However, another input with **different number of processes** will be used to evaluate your programs.

Program Flow

Your program should be run from the command line with the following format:

./program input.txt

Program should first read the input file and print either SAFE or UNSAFE. If the state is unsafe, then it should terminate.

If the state is safe, user can either test new requests or print correct execution order. **Testing a request does not actually update the state**, which means consequent tests are independent from each other.

Correct execution order is the order of process completion times where no deadlock occur. For the sample input, if we allocate remaining 2 resources to the process 1 it would complete and release its resources. Using these resources, either process 0 or 2 can be completed. That's why, execution orders 1->0->2 and 1->2->0 are both correct.

Output

A possible program output for the sample input is given below.

```
Input state is SAFE. Enter E to execute, R to request.
>E
A correct process execution order is 1->0->2. Enter E to execute, R to request.
>R
Which process requests the resource?
>1
How many resources does process 1 request?
>2
Request CAN be granted. Enter E to execute, R to request.
>R
Which process requests resource?
>2
How many resources does process 1 request?
>1
Request CANNOT be granted. Enter E to execute, R to request.
```

Notes

- You will not implement any actual processes or threads for this homework. **It is pure simulation.**
- Course slides contain both "one type of resource" and "multiple types of resources" versions of the algorithm. **Do not implement the "multiple types of resources" version.**
- **Testing a request does not actually update the state**. It only test whether a request can be granted, it does not actually allocate any resources.
- You will lose points if your program does not work for arbitrary number of processes.
- **Your program must compile and run in ITU SSH**. You will lose points if you show your program on your own operating system, even if it is UNIX / Linux.
- If your program needs specific instructions to compile/run or if you have any notes, indicate them as comments in the beginning of your code.

Appendix

You may make use of the following code example for File I/O operations in C and C++.

- http://rosettacode.org/wiki/Read_a_file_line_by_line