

Operating Systems (PG)

Assignment 4 - Banker's Algorithm

Deadline : 11th November 2017 11:55 pm. No submissions will be accepted after deadline.

In this assignment you are supposed to implement Banker's Resource Request algorithm for deadlock avoidance.

A process may or may not request all resource at same time. Banker's safety algorithm should be executed before allocating resources to the process. If the request is denied the process goes into waiting queue till the request is not satisfied. Print "Deny" when this scenario occurs. Print "Grant" when the request is granted. At the end of the execution print the safe sequence in which the resources were allocated to processes. Print the safe sequence in terms of <process_id>.

Write a C/C++ program which takes following inputs :

- N R Q (number of processes, resources, queries respectively)
- Process ids
- Respective arrival times
- Available instances of resources
- Maximum R that can be allocated to a process
- Resources allocated to processes
- 'q' lines containing q queries. Each query will be a request from process <pid> for given resources

E.g: Assuming there are 3 resources in the system

0 1 1 1 (i.e. P0 is requesting 1 instance of each available resource)

1 1 0 1 (i.e. P1 is requesting 1 instance of 1st and 3rd resource)

Note: request from a process will never exceed the total requirement for the same process.

Example: Input:

5 3 5

0 1 2 3 4

0 0 0 0 0

3 3 2

7 5 3

3 2 2

9 0 2

2 2 2

4 3 3

0 1 0

2 0 0
3 0 2
2 1 1
0 0 2
0 7 4 3
1 1 2 2
2 6 0 0
3 0 1 1
4 4 3 1

Output:

Deny
Grant
Deny
Grant
Grant
Grant
Grant
1 3 4 0 2

Explanation: All processes are arriving at same time. Hence process with smaller pid is scheduled first.

Note: Assume burst time of each process as 1 time unit. If system is not in safe state after any allocation the print "Not safe".

Upload Format:

Put source code in folder named RollNo_Assignment4. Also create a shell script RollNo.sh to compile and execute your code. Finally create RollNo_Assignment4.tar.gz and upload on moodle.

Example [folder structure] :

2017XXXXX_Assignment4

|_1.cpp

|_2017XXXX.sh

create 2017XXXXX_Assignment4.tar.gz

Shell script file:

#!/bin/bash

g++ file.cpp

./a.out <\$1> <\$2>

\$1 is input file and \$2 is output file.

Instructions:

1. Follow upload format strictly. Follow input and output format strictly. Evaluation will be

automated for this assignment. Make sure you upload script file as well and test your script file before uploading.

2. Copying code from friends/internet/seniors will lead to straight zero. No arguments will be entertained later for copy cases.

3. You are allowed to use STLs in this assignment.