Programming Assignment 3 (Deadlock) [**30 pts**]

COMP 4270/6270: Operating Systems

Spring 2019

Due: 04/18/2019

Objectives

By completing this programming assignment, you will be familiarized with the basic concept of deadlock, the Banker's algorithm for deadlock avoidance, and recovery from deadlock. **For this assignment, you are free to use ANY programming language of your choice**.

Overview

In this assignment, you will write a program that simulates deadlock avoidance based on the Banker's algorithm and recovery from deadlock.

Task 1 [10 pts]

Review slides 25 – 30 of Chapter 7 that explain the Banker's algorithm. Implement the Banker's algorithm which includes the Safety algorithm. Your program should read the system configuration information from a data file (*i.e.*, sys_config.txt). More specifically, this data file contains the 'Available' vector, 'Max' matrix, 'Allocation' matrix in the following format.

Allocation

Process 0: 0 1 0

Process 1: 2 0 0

Process 2: 3 0 2

Process 3: 2 1 1

Process 4: 0 0 2

Max

Process 0: 7 5 3

Process 1: 3 2 2

Process 2: 9 0 2

Process 3: 2 2 2

Process 4: 4 3 3

Available

3 3 2

NOTE that the number of processes and the number of resource types specified in the data file may vary (*e.g.*, in the above example system config file, there are 5 processes and 3 resource types). Your program should be able to handle arbitrary number of processes and resource types. After reading the data file, your program should print out either 'SAFE' or 'UNSAFE' to indicate whether the system specified in the input data file is in the safe state or not. For example,

Prog2>./a.out sys_config.txt

SAFE

Task 2 [15 pts]

Your program should take a 'Request' vector as user input and print out either 'GRANTED' or 'NOT GRANTED' to indicate whether the resource request can be granted or not. For example,

Prog2>./a.out sys_config.txt

SAFE

Request vector: 1 0 2 <------- take user input

GRANTED

Request vector: 1 0 2 3 <-------- continue to take another user input

Wrong input! <-------- print out 'Wrong input' if the length of the request vector does not match.

Request vector: 12 2 2

NOT GRANTED

Task 3 [5 pts]

Implement a simple deadlock recovery mechanism. More specifically, if requested resources cannot be granted (*i.e.*, the output is 'NOT GRANTED), then identify the minimum number of processes that should be forced to be terminated such that the requested resources can be granted. For example,

Prog2>./a.out sys_config.txt

SAFE

Request vector: 12 2 2

NOT GRANTED

Process 0, Process 1 should be terminated to grant the requested resources. <--------

Evaluation criteria

- Your assignment will be evaluated based on the following:

  Documentation 10% - your code should be easy to read and well commented. For each function used in your program, the use of function, its parameters, and return values should be well described.

  Compilation 20% - your program should compile with no errors and/or warnings (base points)

  Correctness 70% - To grade your work, we will run your program with some test cases. You will get full credits for correctness if your program prints out correct output for our input test cases.

References

[1] Operating Systems Concepts 9th Edition by Silberscharz, Yale, and Gagne, Wiley