

Operating Systems - Homework 5

Due Date: 10.05.2019

1- In the message passing solution for the Reader/Writer problem, change the controller in such a way that the readers have higher priority than the writers.

2- Consider the following snapshot of a system:

	Allocation	Max	Available
	A B C D	A B C D	A B C D
P 0	2 0 0 1	4 2 1 2	3 3 2 1
P 1	3 1 2 1	5 2 5 2	
P 2	2 1 0 3	2 3 1 6	
P 3	1 3 1 2	1 4 2 4	
P 4	1 4 3 2	3 6 6 5	

Answer the following questions using the banker's algorithm:

- Illustrate that the system is in a safe state by demonstrating an order in which the processes may complete.
- If a request from process P1 arrives for (1,1,0,0), can the request be granted immediately?
- If a request from process P4 arrives for (0,0,2,0), can the request be granted immediately?

3- (**Bonus +15**) There is a one-way narrow bridge on a public highway. Traffic may only cross the bridge in one direction at a time. The bridge can tolerate the weight of at most 5 vehicles. Each car which arrives the bridge calls the following OneVehicle thread :

```
OneVehicle (Direction direc) {  
    ArriveBridge(direc);  
    CrossBridge(direc);  
    ExitBridge(direc);  
}
```

In the code above, direc can be East or West which gives the direction of vehicle on the bridge. Using a **Monitor**, write the procedures ArriveBridge and ExitBridge to synchronize the cars (the CrossBridge procedure just prints a message). ArriveBridge must not return until it safe for a car to cross the bridge in the given direction (it must guarantee that there will be no head-on collisions and no overweight on bridge). ExitBridge is called to indicate that the car has finished crossing the bridge. ExitBridge should take steps to let additional cars cross the bridge. For example, if there are 3 cars on bridge crossing in East direction, ArriveBridge should allow at most 2 more cars entering in East direction and stop all cars in West direction until the bridge becomes empty. No fairness or freedom from starvation is required.

4- (**Bonus +10**) Write a multithread program in Java, C++ or C# language which simulate the above Bridge problem. The program should accept the arrival ratio in each direction and show the bridge status and queues in each side.