Machine Translated by Google

Task Series 3

3.1 Binary semaphores (with controller)

Implement binary semaphores with the desired functionality achieved through an appropriate synchronization framework in the spirit of a controller / monitor. Test your implementation using it in one of the previous tasks 2.3 or 2.4 (choose whichever you like – if it had any problems, they should be fixed first).

3.2 Recognition of prime numbers (with controller)

Modify the solution you developed in task 2.2 so that the desired functionality is achieved through appropriate synchronization frameworks in the spirit of a controller / monitor (you will not use the binary semaphores you developed in 3.1).

3.3 Narrow bridge (with controller)

Modify the solution you developed in task 2.3 so that the desired functionality is achieved through appropriate synchronization frameworks in the spirit of a controller / monitor (you will not use the binary semaphores you developed in 3.1). Test your implementation through a suitable
simulation program (the one you already made in task 2.3).

3.4 Roller coaster (with controller)

Modify the solution you developed in task 2.4 so that the desired functionality is achieved through appropriate synchronization frameworks in the spirit of a controller / monitor (you will not use the binary semaphores you developed in 3.1). Test your implementation through a suitable
simulation program (the one you already made in task 2.4).

The implementation of tasks can be done using a combination of mutexes and conditions of the pthread library to make synchronization frameworks in the spirit of a controller. For tasks 3.2, 3.3 3.4 you can alternatively write your implementation in Java using the synchronized functionality and
wait/notify provided by the language. Your solutions should assume an open model and work regardless of whether the wake-up mode follows the signal-block or signal-continue model.

Delivery: Wednesday 10 January 2024, 23:59