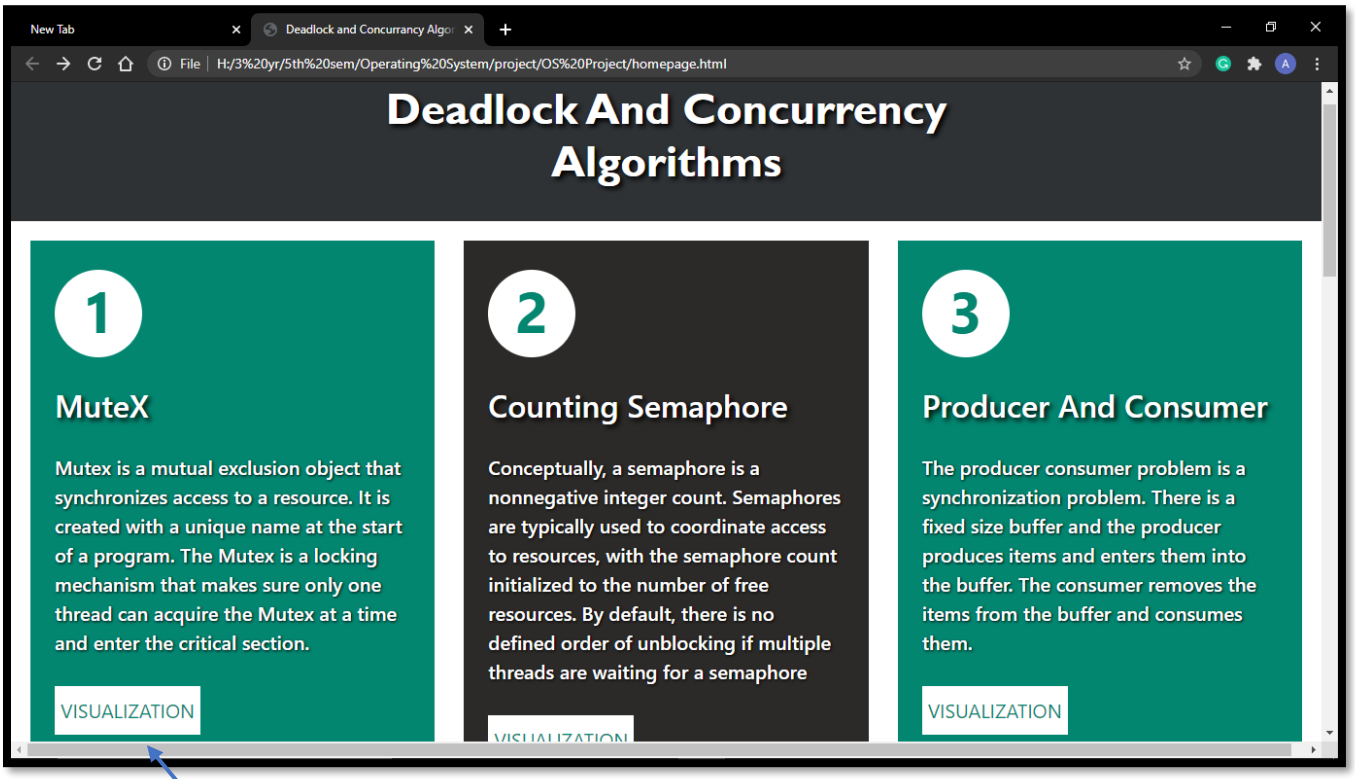
Pandit Deendayal
Petroleum University

Team 66

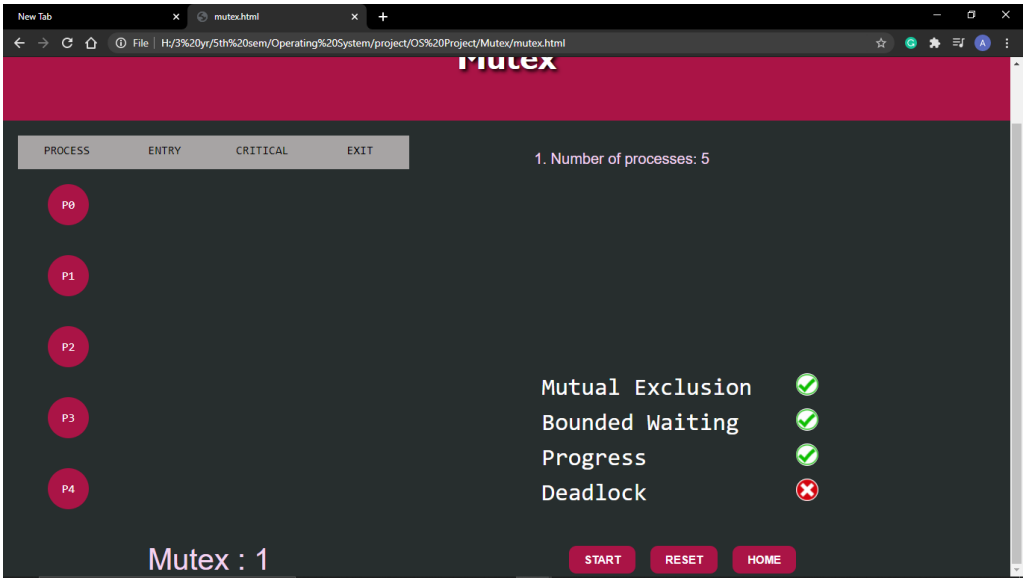
Deadlock and Concurrency Algorithms Website

There are 8 algorithms with visualization.

Home page of website



To see visualization of algorithm.

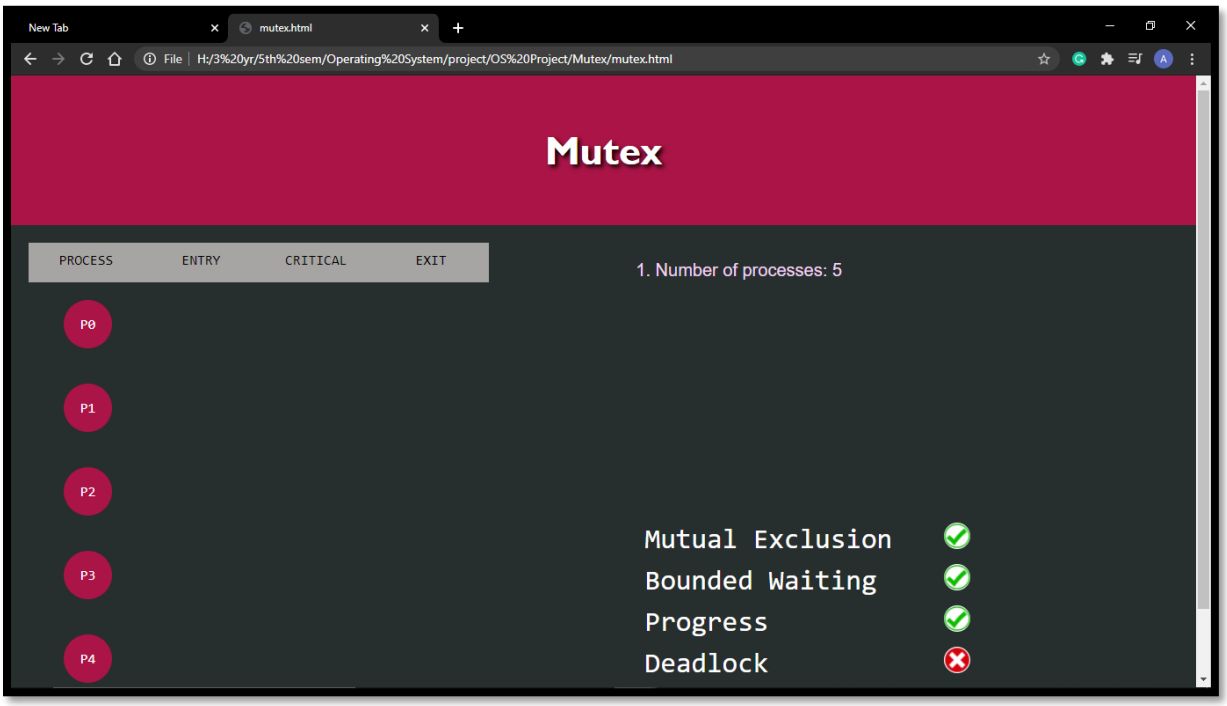


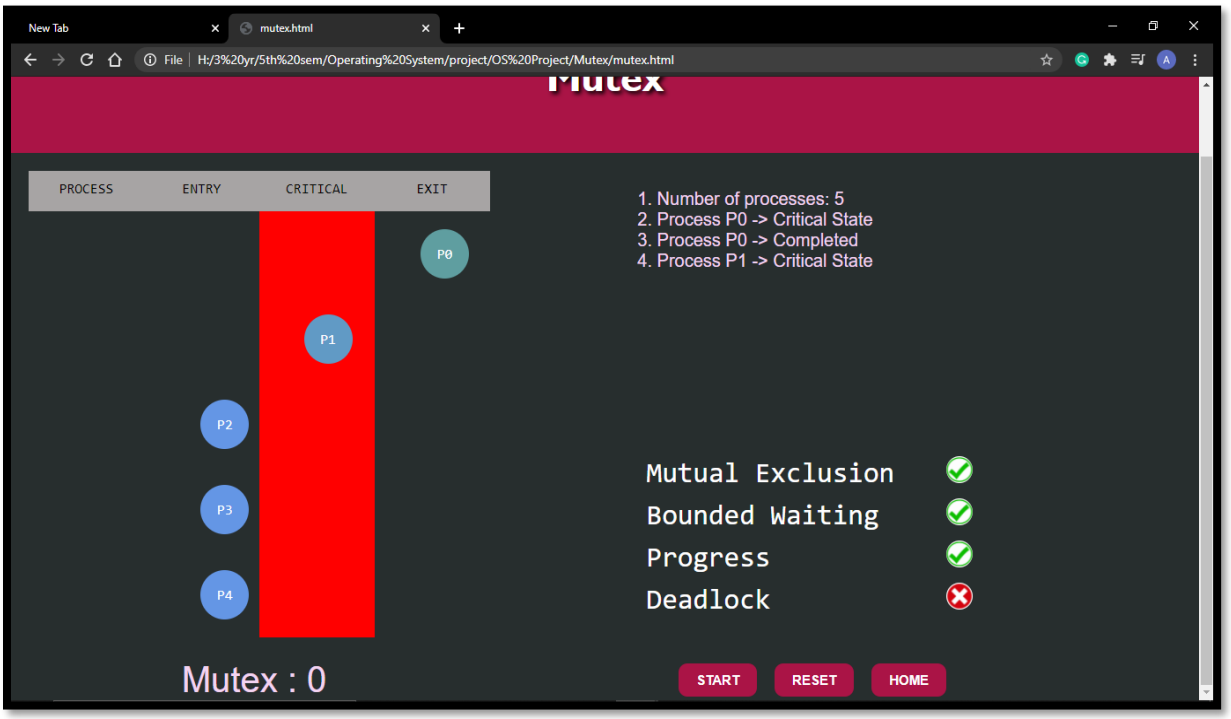
In every algorithm there is START, RESET and HOME button

1. MuteX:

Mutex is a mutual exclusion object that synchronizes access to a resource.

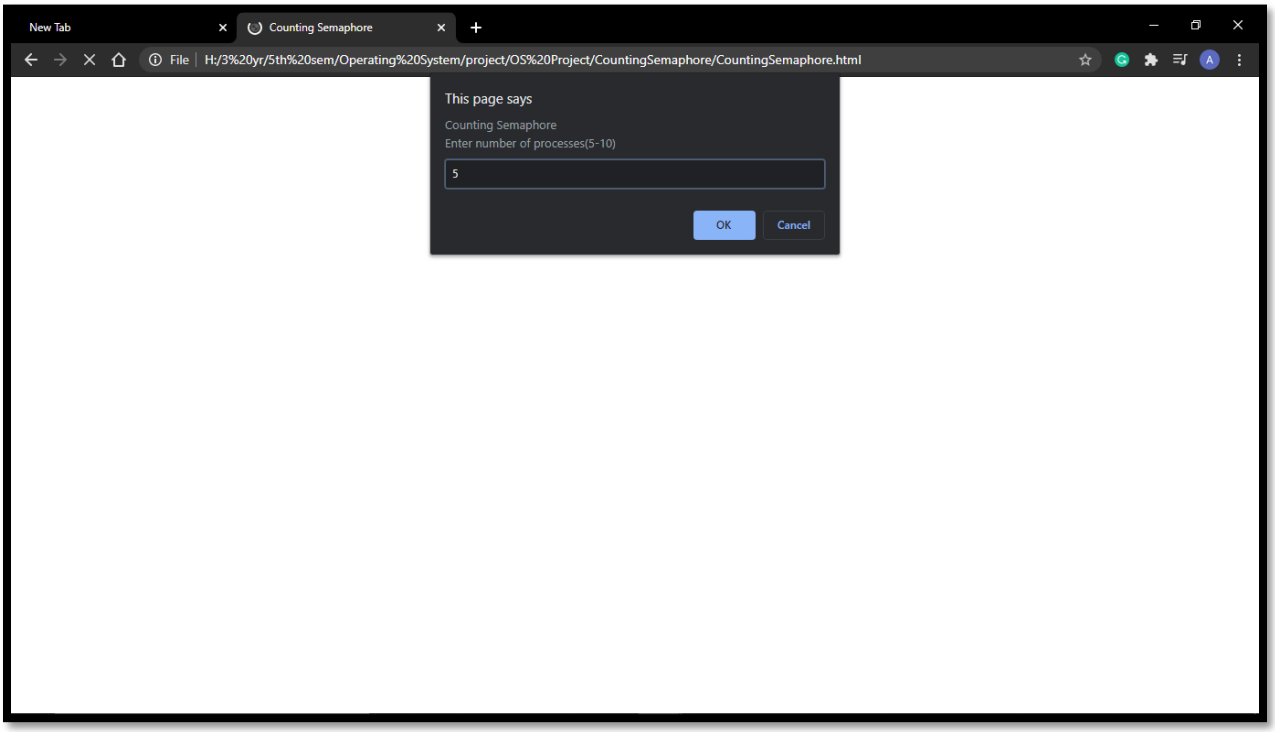
It is created with a unique name at the start of a program. The Mutex is a locking mechanism that makes sure only one thread can acquire the Mutex at a time and enter the critical section.

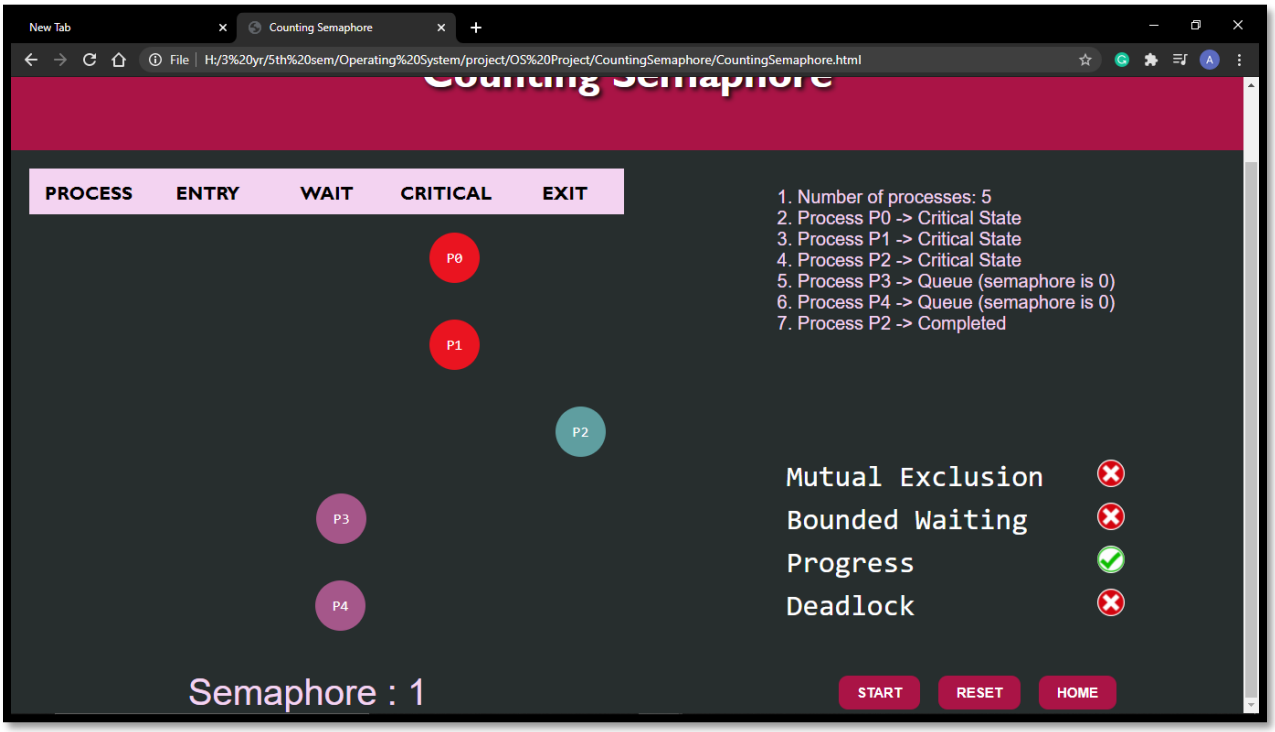




2. Counting Semaphore:

Conceptually, a semaphore is a nonnegative integer count. Semaphores are typically used to coordinate access to resources, with the semaphore count initialized to the number of free resources.
By default, there is no defined order of unblocking if multiple threads are waiting for a semaphore



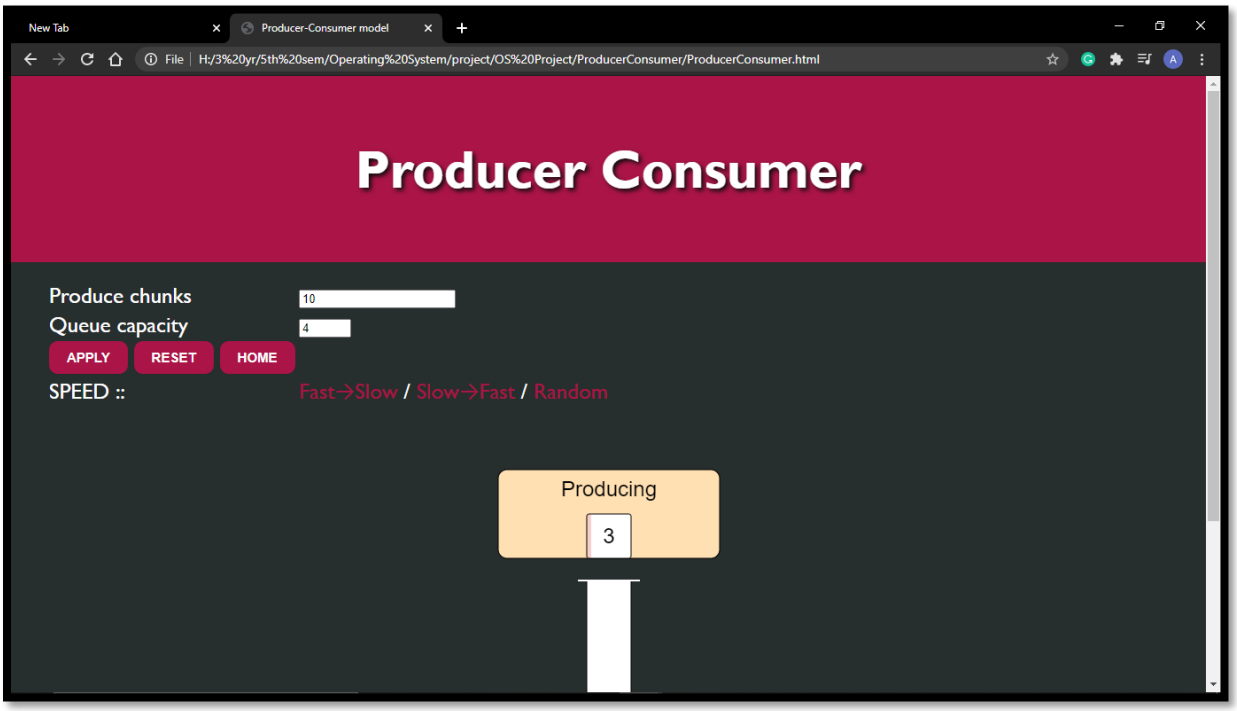


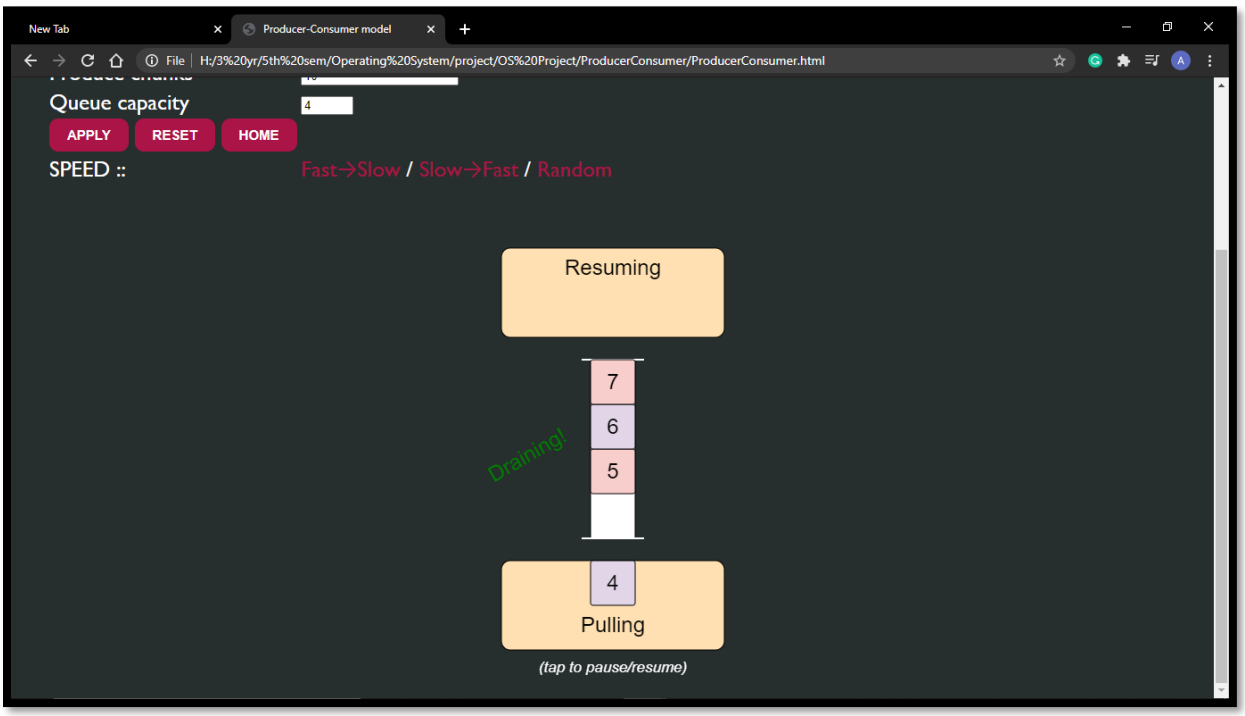
3. Producer Consumer:

The producer consumer problem is a synchronization problem.

There is a fixed size buffer and the producer produces items and enters them into the buffer.

The consumer removes the items from the buffer and consumes them.



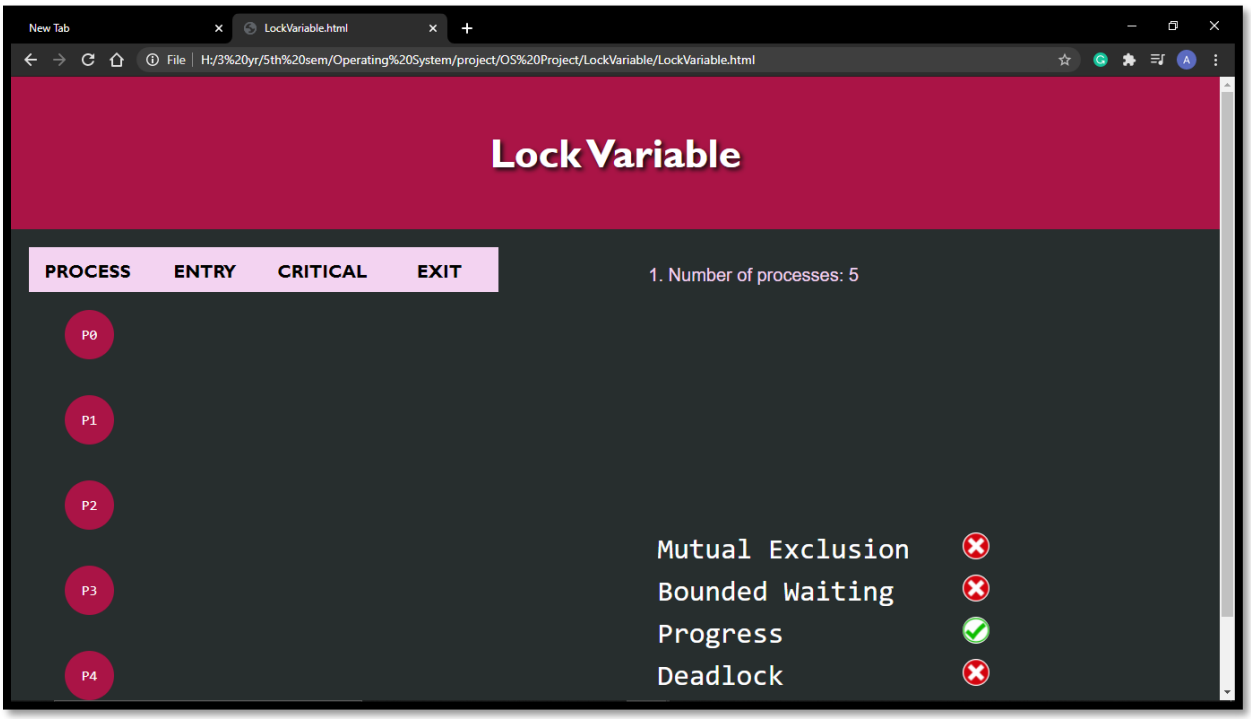


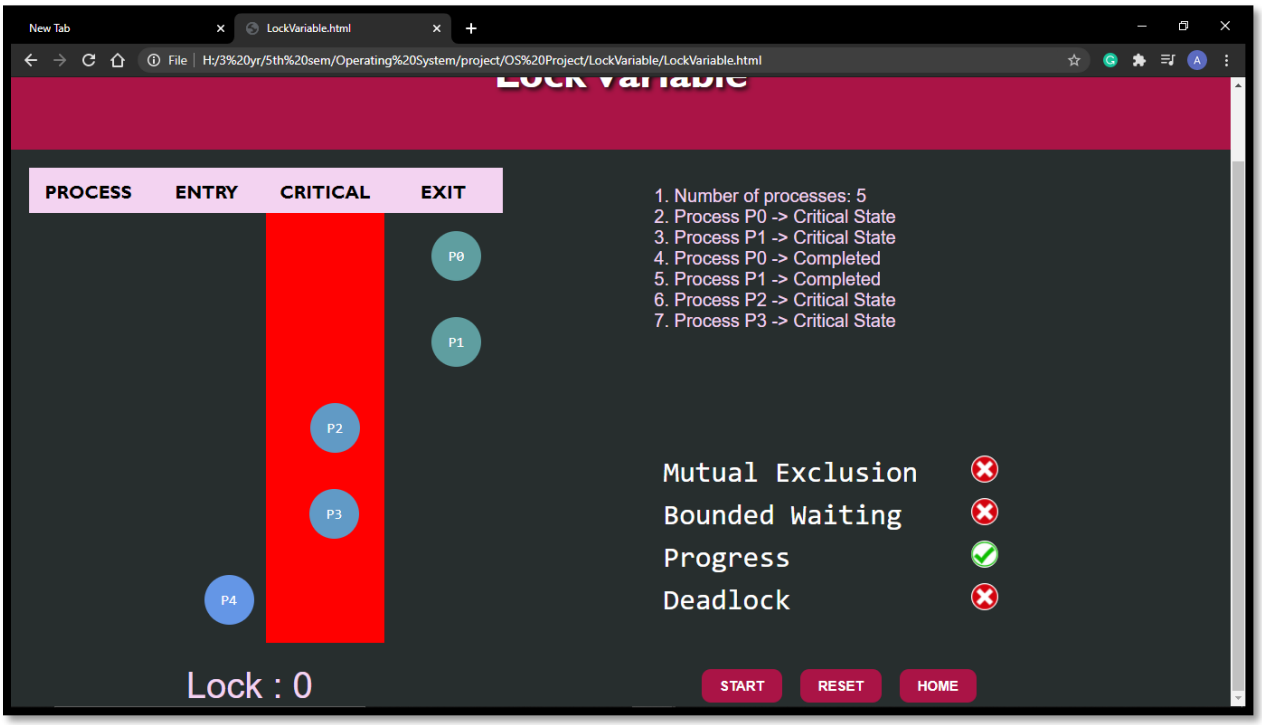
4. Lock Variable:

Two values of lock can be possible, either 0 or 1.

Lock value 0 means that the critical section is vacant while the lock value 1 means that it is occupied.

A process which wants to get into the critical section first checks the value of the lock variable.



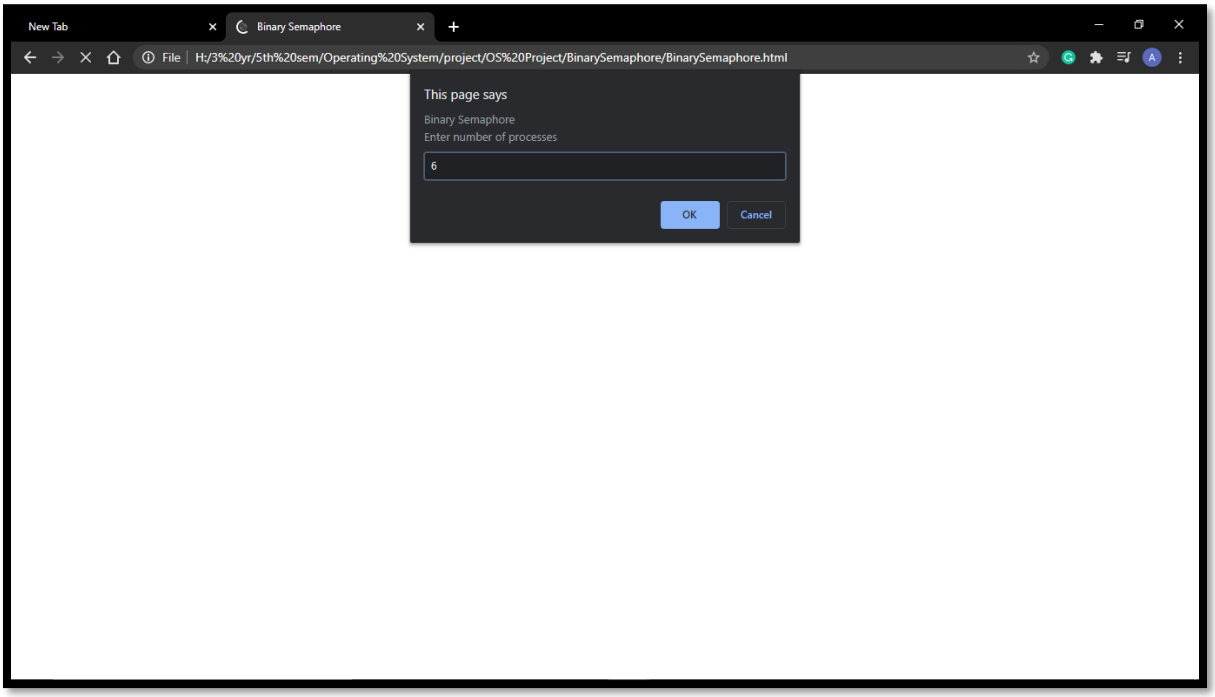


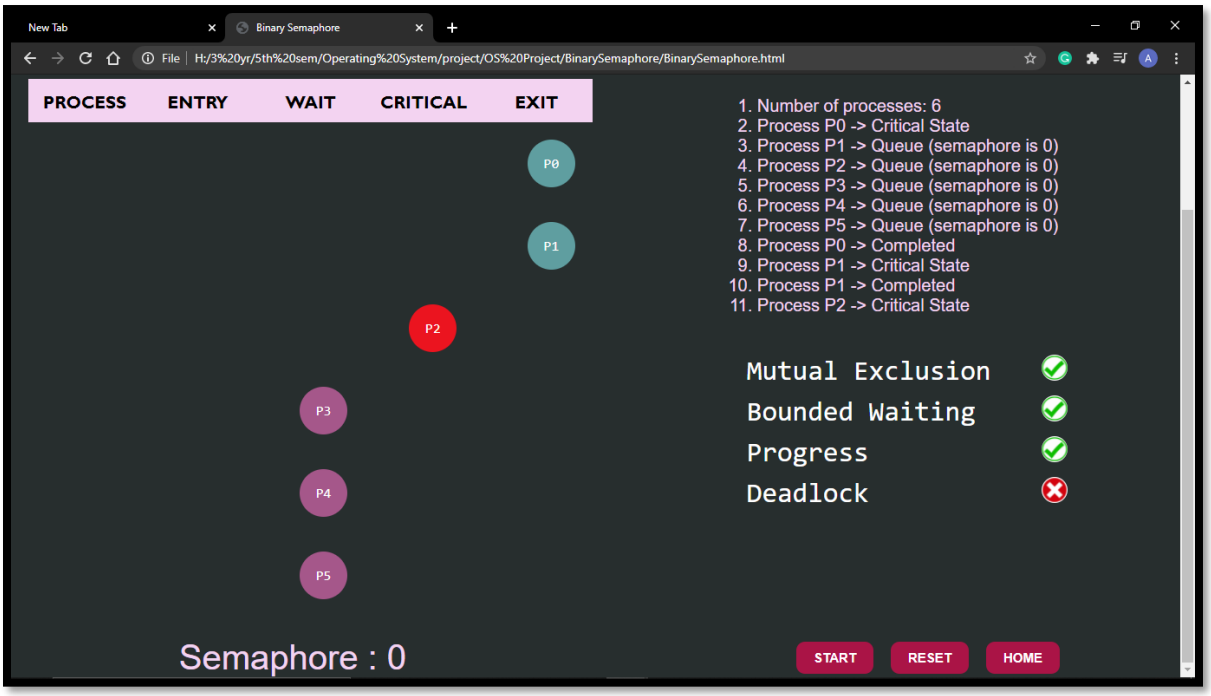
5. Binary Semaphore:

A binary semaphore is restricted to values of zero or one, while a counting semaphore can assume any nonnegative integer value.

A binary semaphore can be used to control access to a single resource.

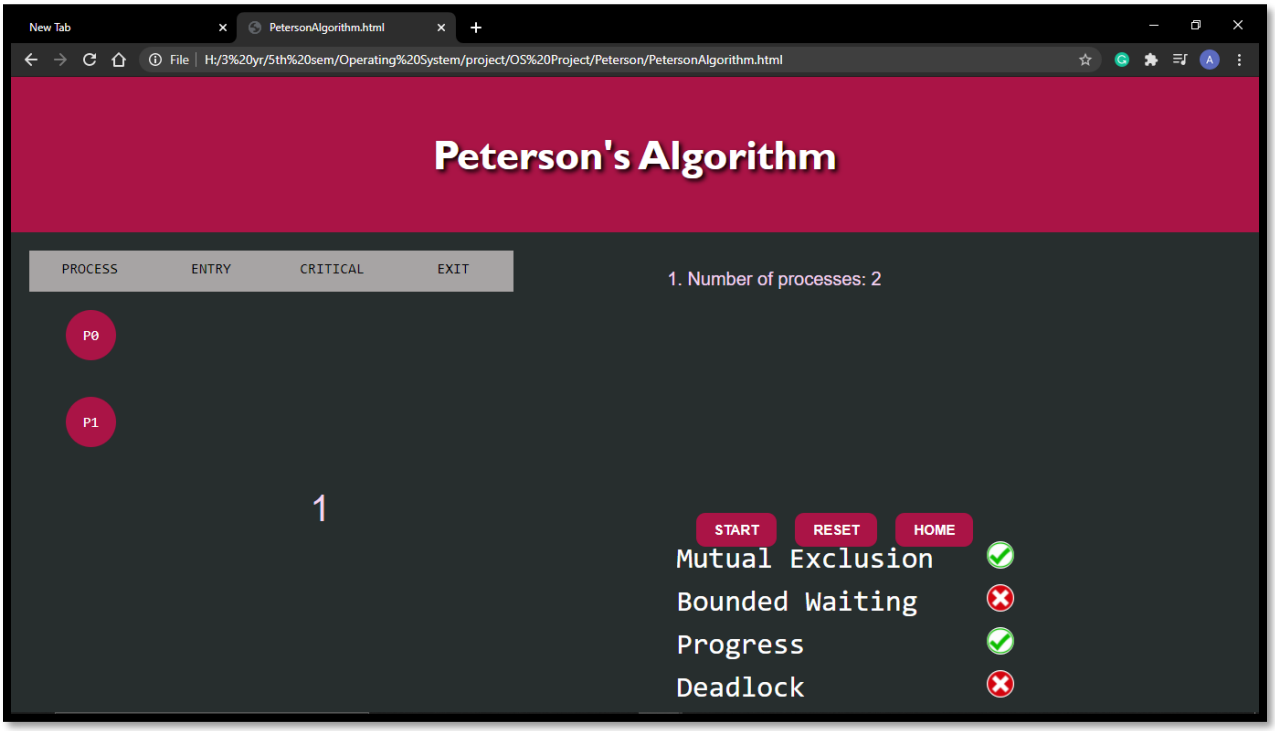
In particular, it can be used to enforce mutual exclusion for a critical section in user code.

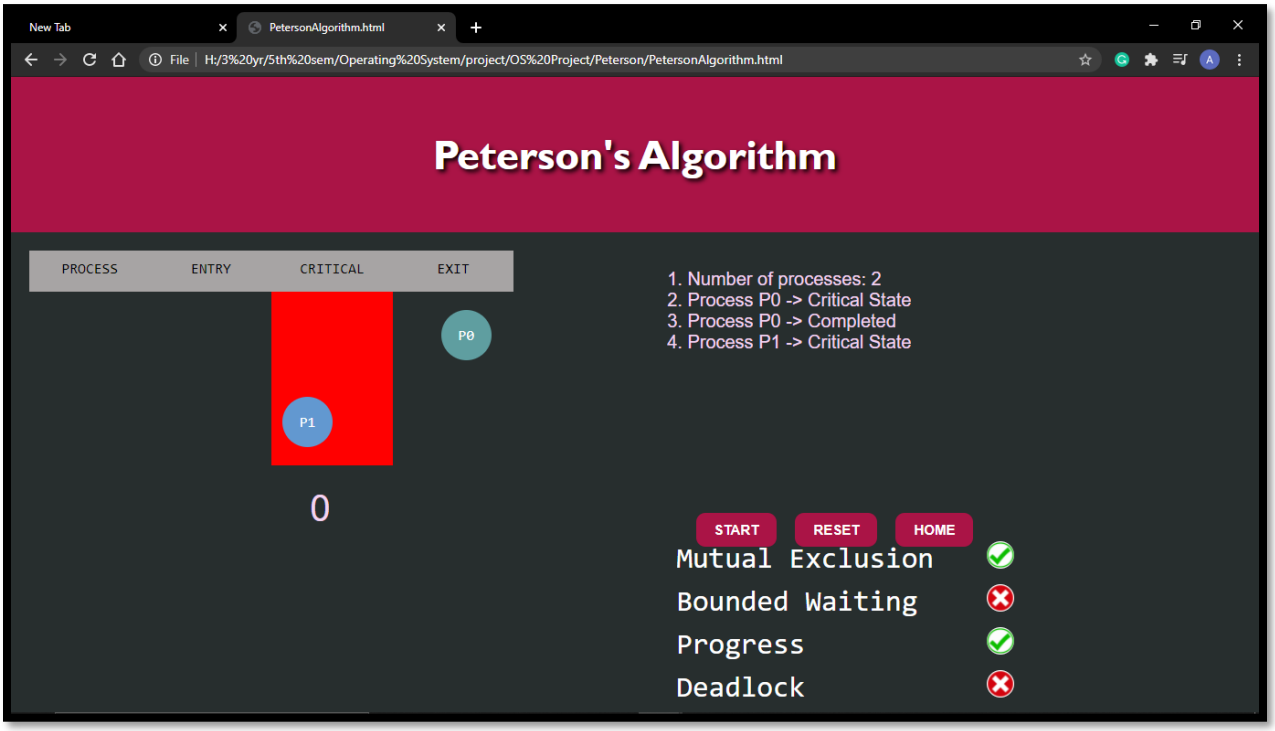




6. Peterson’s Algorithm:

Peterson's algorithm (or Peterson's solution) is a concurrent programming algorithm for mutual exclusion that allows two or more processes to share a single-use resource without conflict, using only shared memory for communication.

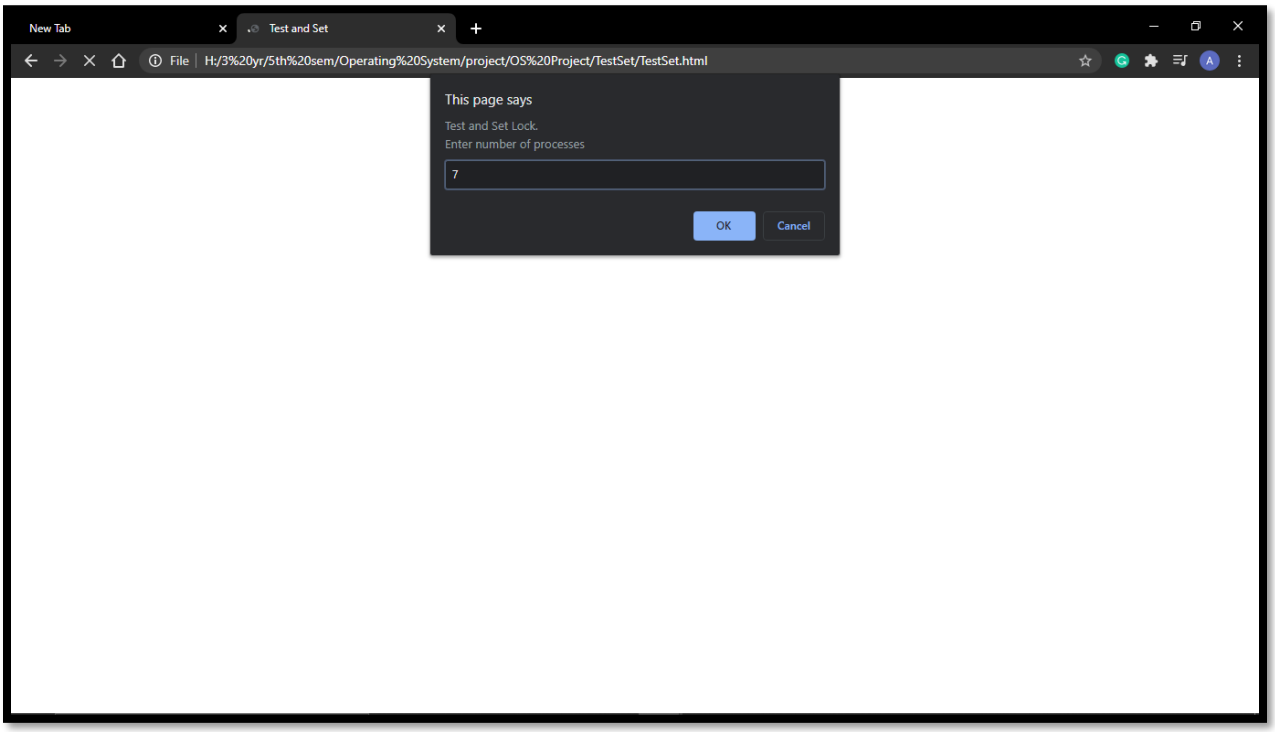


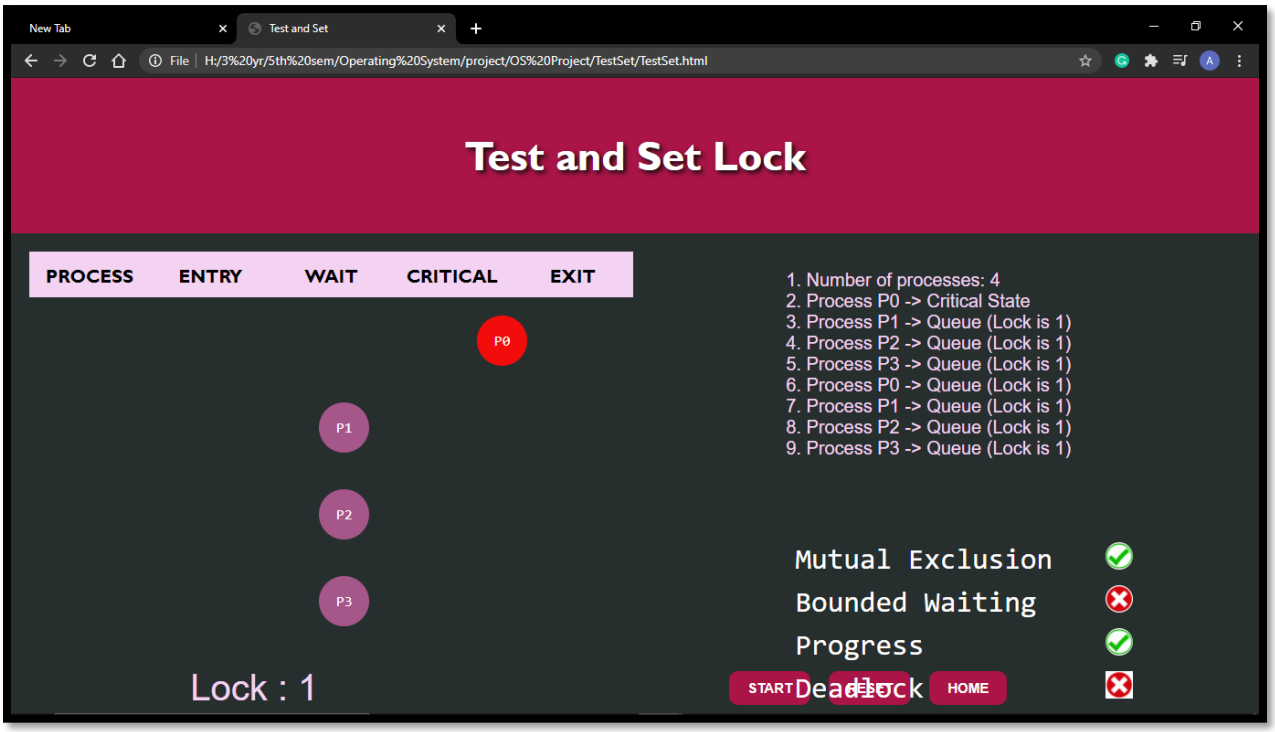


7. Test and Set:

In Process Synchronization, Test and Set Lock (TSL) is a synchronization mechanism that uses a test-and-set instruction to provide the synchronization among the processes.

It ensures mutual exclusion and freedom from deadlock.





8. Bankers Algorithm:

A binary semaphore is restricted to values of zero or one, while a counting semaphore can assume any nonnegative integer value.

A binary semaphore can be used to control access to a single resource.

In particular, it can be used to enforce mutual exclusion for a critical section in user code.