

ASSIGNMENT 5

This is an implementation of PetersonLock.

Implementation.c

In this program, the `increment()` function is executed concurrently by two threads, each identified by a unique integer thread ID. The `lock()` and `unlock()` functions are used to ensure that the counter is accessed atomically.

PetersonLock.c

This code defines the implementation of the Peterson lock in C. The `lock()` and `unlock()` functions implement the mutual exclusion using two flags and a victim variable.

The `lock()` function takes the thread id as an argument, sets the flag for that thread to true, and sets the victim variable to that thread id. Then, it enters a busy-wait loop that waits for the other thread to release the lock. The loop condition checks if the flag for the other thread is set to true and the victim variable is still set to the current thread id. If both conditions are true, the thread continues to wait. Once the other thread releases the lock by setting its flag to false, the current thread can exit the loop and acquire the lock.

The `unlock()` function takes the thread id as an argument and simply sets the flag for that thread to false, releasing the lock.

Note that this implementation assumes that there are only two threads competing for the lock, and each thread is identified by a unique thread id (either 0 or 1). If there are more than two threads or the thread ids are not fixed, this implementation may not work correctly.

OUTPUT

```
(base) adityasai@ADITYASAI-MacBook-Air Assignment5 % gcc -c -o PetersonLock.o PetersonLock.c
(base) adityasai@ADITYASAI-MacBook-Air Assignment5 % gcc -pthread -o Implementation Implementation.c PetersonLock.o
(base) adityasai@ADITYASAI-MacBook-Air Assignment5 % ./Implementation
Counter value: 0
```

1000 Increment operations and 1000 decrement operations using two threads. This has led to a correct answer of 0 for the counter value.

To Run:

First, enter the folder containing all the codes with the header file, and perform the following commands:

```
> gcc -c -o PetersonLock.o PetersonLock.c
> gcc -pthread -o Implementation Implementation.c PetersonLock.o
> ./Implementation
```