

202251142_peterson

Code

```
#include <stdio.h>
#include <pthread.h>
#include <stdbool.h>
#include <unistd.h>
#include <stdint.h>
#include <inttypes.h>

#define NUM_PROCESSES 2

bool flag[NUM_PROCESSES];
int turn;
int shared_counter = 0;

void initialize_peterson() {
    flag[0] = flag[1] = false;
    turn = 0;
}

void enter_critical_section(intptr_t process_id) {
    intptr_t other = 1 - process_id;
    flag[process_id] = true;
    turn = other;

    printf("Process %" PRIIdPTR " attempting to enter critical section.\n", process_id);
    while (flag[other] && turn == other) {
    }
}

void exit_critical_section(intptr_t process_id) {
    flag[process_id] = false;
    printf("Process %" PRIIdPTR " exited the critical section.\n", process_id);
}

void critical_section_work(intptr_t process_id) {
    shared_counter++;
    printf("Process %" PRIIdPTR " in critical section. Updated shared counter to: %d\n", process_id);
    sleep(1);
}

void remainder_section(intptr_t process_id) {
    printf("Process %" PRIIdPTR " in remainder section.\n", process_id);
    sleep(1);
}

void* process(void* arg) {
    intptr_t process_id = (intptr_t)arg;
```

```

        for (int i = 0; i < 5; i++) {
            enter_critical_section(process_id);
            critical_section_work(process_id);
            exit_critical_section(process_id);
            remainder_section(process_id);
        }
        return NULL;
    }

int main() {
    pthread_t threads[NUM_PROCESSES];
    intptr_t process_ids[NUM_PROCESSES] = {0, 1};

    initialize_peterson();

    for (int i = 0; i < NUM_PROCESSES; i++) {
        pthread_create(&threads[i], NULL, process, (void*)process_ids[i]);
    }

    for (int i = 0; i < NUM_PROCESSES; i++) {
        pthread_join(threads[i], NULL);
    }

    printf("All processes complete. Final shared counter value: %d\n", shared_counter);
    return 0;
}

```

output

```

Process 0 attempting to enter critical section.
Process 0 in critical section. Updated shared counter to: 1
Process 1 attempting to enter critical section.
Process 0 exited the critical section.
Process 0 in remainder section.
Process 1 in critical section. Updated shared counter to: 2
Process 0 attempting to enter critical section.
Process 0 in critical section. Updated shared counter to: 3
Process 1 exited the critical section.
Process 1 in remainder section.
Process 0 exited the critical section.
Process 0 in remainder section.
Process 1 attempting to enter critical section.
Process 1 in critical section. Updated shared counter to: 4
Process 0 attempting to enter critical section.
Process 0 in critical section. Updated shared counter to: 5
Process 1 exited the critical section.
Process 1 in remainder section.
Process 0 exited the critical section.
Process 0 in remainder section.
Process 1 attempting to enter critical section.
Process 1 in critical section. Updated shared counter to: 6
Process 0 attempting to enter critical section.

```

```
Process 0 in critical section. Updated shared counter to: 7
Process 1 exited the critical section.
Process 1 in remainder section.
Process 0 exited the critical section.
Process 0 in remainder section.
Process 1 attempting to enter critical section.
Process 1 in critical section. Updated shared counter to: 8
Process 1 exited the critical section.
Process 1 in remainder section.
Process 0 attempting to enter critical section.
Process 0 in critical section. Updated shared counter to: 9
Process 1 attempting to enter critical section.
Process 1 in critical section. Updated shared counter to: 10
Process 0 exited the critical section.
Process 0 in remainder section.
Process 1 exited the critical section.
Process 1 in remainder section.
All processes complete. Final shared counter value: 10
```