Project 1: Multithreaded Programming and Synchronization

CECS 326: Operating Systems

Fall 2020

September 25, 2020

Sergio Vasquez 018663448
Kuldeep Gohil 015499534
sergio.vasquez01@student.csulb.edu
Kuldeep.Gohil@student.csulb.edu

Output for part1(assuming 4 threads):

```
sergio@sergio-HP-EliteBook-8470p:~/CECS-326/lab1$ ./part1.o 4
***thread 1 sees value 0
***thread 0 sees value 1
***thread 0 sees value 2
***thread 0 sees value 3
***thread 2 sees value 4
***thread 2 sees value 5
***thread 3 sees value 6
***thread 3 sees value 7
***thread 1 sees value 8
***thread 1 sees value 9
***thread 0 sees value 10
***thread 2 sees value 11
***thread 1 sees value 12
```

```
***thread 2 sees value 48
***thread 1 sees value 49
***thread 3 sees value 50
***thread 3 sees value 51
***thread 3 sees value 52
***thread 3 sees value 53
***thread 3 sees value 54
***thread 3 sees value 55
***thread 3 sees value 56
***thread 0 sees value 57
***thread 2 sees value 58
***thread 1 sees value 59
***thread 1 sees value 60
***thread 3 sees value 61
***thread 3 sees value 62
***thread 0 sees value 63
***thread 2 sees value 64
***thread 1 sees value 65
***thread 3 sees value 66
***thread 3 sees value 67
Thread 3 sees final value 68
***thread 0 sees value 68
***thread 0 sees value 69
***thread 2 sees value 70
***thread 1 sees value 71
***thread 0 sees value 72
***thread 0 sees value 73
***thread 2 sees value 74
Thread 2 sees final value 75
***thread 1 sees value 75
***thread 0 sees value 76
***thread 0 sees value 77
Thread 0 sees final value 78
***thread 1 sees value 78
Thread 1 sees final value 79
sergio@sergio-HP-EliteBook-8470p:~/CECS-326/lab1$
```

Output for part2(Assuming 4 threads).

```
kuldeep@kuldeep:~/Desktop/CECS 326/project1$ ./part2 4
*** thread 0 sees value 0
*** thread 0 sees value 1
*** thread 0 sees value 2
*** thread 0 sees value 3
*** thread 0 sees value 4
*** thread 0 sees value 5
*** thread 0 sees value 6
*** thread 0 sees value 7
*** thread 0 sees value 8
*** thread 0 sees value 9
*** thread 0 sees value 10
*** thread 0 sees value 11
*** thread 0 sees value 12
*** thread 1 sees value 13
*** thread 3 sees value 14
*** thread 3 sees value 15
*** thread 3 sees value 16
*** thread 3 sees value 17
*** thread 3 sees value 18
*** thread 0 sees value 19
*** thread 3 sees value 20
*** thread 1 sees value 21
*** thread 1 sees value 22
*** thread 1 sees value 23
*** thread 1 sees value 24
*** thread 0 sees value 25
*** thread 0 sees value 26
*** thread 0 sees value 27
*** thread 0 sees value 28
*** thread 0 sees value 29
*** thread 0 sees value 30
*** thread 1 sees value 31
*** thread 1 sees value 32
*** thread 1 sees value 33
*** thread 1 sees value 34
*** thread 1 sees value 35
*** thread 1 sees value 36
*** thread 1 sees value 37
*** thread 1 sees value 38
*** thread 2 sees value 39
*** thread 2 sees value 40
*** thread 2 sees value 41
*** thread 2 sees value 42
*** thread 2 sees value 43
```

```
*** thread 2 sees value 44
*** thread 2 sees value 45
*** thread 2 sees value 46
*** thread 2 sees value 47
*** thread 2 sees value 48
*** thread 2 sees value 49
*** thread 2 sees value 50
*** thread 2 sees value 51
*** thread 2 sees value 52
*** thread 2 sees value 53
*** thread 2 sees value 54
*** thread 2 sees value 55
*** thread 2 sees value 56
*** thread 2 sees value 57
*** thread 2 sees value 58
*** thread 3 sees value 59
*** thread 3 sees value 60
*** thread 3 sees value 61
*** thread 3 sees value 62
*** thread 3 sees value 63
*** thread 3 sees value 64
*** thread 3 sees value 65
*** thread 3 sees value 66
*** thread 3 sees value 67
*** thread 3 sees value 68
*** thread 3 sees value 69
*** thread 3 sees value 70
*** thread 3 sees value 71
*** thread 3 sees value 72
*** thread 1 sees value 73
*** thread 1 sees value 74
*** thread 1 sees value 75
*** thread 1 sees value 76
*** thread 1 sees value 77
*** thread 1 sees value 78
*** thread 1 sees value 79
Thread 1 sees final value 80
Thread 2 sees final value 80
Thread 3 sees final value 80
Thread 0 sees final value 80
kuldeep@kuldeep:~/Desktop/CECS 326/project1$
```

Processing of design:

Since the project was split into two parts, non-synchronization and synchronization,we split the work into two. Sergio worked on part 1 and Kuldeep worked on part 2 of the implementation. We defined a function to read in the second argument passed by the user and check if it is an integer in order to keep the main function more readable. Once the user input was validated we created both the threads and their ids as arrays since this would be a general component of both solutions. After the completion of the first portion we did the necessary addition to allow synchronization.

Discussion for difference in part 1 and part 2:

The reason for difference between the output of part 1 and part 2 is the fact that the variable that is being accessed is a shared variable among all threads. When multiple threads are running on the same application they may try to access the same variable at the same time which can cause unforeseen results. What synchronization does (part 2) is allow only one thread to access the shared variable at a given time which prevents the multiple different final values that occur in part 1.