Name: David Fantin
ID: 1525813
Class: CS 4373
Date: 1/31/2022

Chapter 4 and 5 Programming Assignment

Problem 4.3:

Unfortunately I kept running into segmentation errors when compiling the program. I know where the issues are coming from (see the screenshot below) however I could not seem to overcome this issue when using the pthread_create() method. I believe my design should work though, if It wasn't for this issue.

```
4.3-TrapRule.c: In function 'main':
4.3-TrapRule.c:93: warning: cast to pointer from integer of different size
4.3-TrapRule.c:101: warning: cast to pointer from integer of different size
4.3-TrapRule.c:109: warning: cast to pointer from integer of different size
```

My make-statement:

gcc -Wall -g -pthread -std=c99 4.3-TrapRule.c -o 4.3-TrapRule.o

Semaphore:

At the start of the program I initialized a semaphore (simply called "semaphore") and before the global sum was calculated in each thread, I simply called sem_wait(), then after the sum was calculated I called sem_post().

Advantage:
No busy waiting, so this option does not waist system resources.

Disadvantage:
A bit harder to work with than Mutex or Busy Waiting

Busy Waiting:

Busy waiting was very easy, and simply had a while loop before the global sum was calculated that keeps looping until the other threads finished calculating the sum.

Advantage:
Extremely easy to implement and does not require anything special.

Disadvantage:
Not as efficient as other options since busy waiting waists resources.

Mutex:

At the start of the program I initialized a mutex (simply called “mutex”) and much like the semaphore method, before the global sum was calculated in each thread, I simply locked the mutex, then after the sum was calculated I called unlocked it.

Advantage:
Extremely easy to implement (a mutex is just a lock).

Disadvantage:
Causes busy waiting, which waists resources.

Output: (Incorrect)

```
[djf3095@login 4.3]$ cat job.33849.err
/var/spool/slurm/slurmd/job33849/slurm_script: line 9: 28067 Segmentation fault      /home/djf3095/project3/4.3/4.3-TrapRule.o
[djf3095@login 4.3]$ ls
```

Problem 5.6:

My code did not work, and it also had a segmentation error fault like the previous question.

My make-statement:

gcc -Wall -g -fopenmp 5.6-P-C.c -o 5.6-P-C.o

For parallelizing I attempted to use:

pragma omp parallel num_threads( ) before the producer and consumer methods

Output: (Incorrect)

```
[djf3095@login 5.6]$ cat job.33851.err
/var/spool/slurm/slurmd/job33851/slurm_script: line 9: 28671 Segmentation fault      /home/djf3095/project3/5.6/5.6-P-C.o
[djf3095@login 5.6]$
```