CSE 330 - Operating Systems, Summer 2021

Project 3, **Due: June 17th 2021, 11:59 pm**

No Late Submissions

Strictly Individual Projects

1.Readers and Writers:

Using the semaphores you have implemented, implement the Readers and Writers Problem.

2.Test case:

Have a global variable *int i* and set it to 0 initially.

The readers should read the variable value and print the current value of the variable. Each reader should print the variables twice. The writers should write their ids in the variable. The writer should write in one go and then verify that the correct Id is written into the variable.

Readers and writers do not run in infinite loop, but reader reads twice while writer writes and then again verifies.

The reader yields after one read and then finishes after the second read.

- On the *first* read the reader prints:
    - *Printf(“\n This is the %d th reader reading value i = %d for the first time \n”, readerID, i);*
- *On the second* read the reader prints:
    - *Printf(“\n This is the %d th reader reading value i = %d for the second time \n”, readerID, i);*

The writer yields after the first write and then exits after the verification step.

- On the *first* write the writer prints:
    - *Printf(“\n This is the %d th writer writing value i = %d \n”, writerID, i);*
- On the *verification* loop the writer should print:
    - *Printf(“\n This is the %d th writer verifying value i = %d \n”, writerID, i);*

The test case will be in the following format

2,2

```
1
-1
2
-2
```

The first line will have number of readers and number of writers consecutively, followed by R+W numbers showing the ready Q. The positive numbers are readers, the negative numbers are writers. Always have the initial scratch pad variable i = 0;

In this case the output will be

```
This is the 1 th reader reading value i = 0 for the first time


 This is the 1 th reader reading value i = 0 for the second time


 This is the 1 th writer writing value i = 1


 This is the 1 th writer verifying value i = 1


 This is the 2 th reader reading value i = 1 for the first time


 This is the 2 th reader reading value i = 1 for the second time


 This is the 2 th writer writing value i = 2


 This is the 2 th writer verifying value i = 2
```

Your project must consist of 5 files

1. TCB.h (uses ucontext.h)
2. q.h (includes TCB.h)

3. threads.h (includes q.h)

4. sem.h (includes threads.h) if you have written one.

5. proj-4.c (includes threads.h)
   (make sure the compile command, “gcc proj-4.c” does the correct compilation).

All 5 files are to be ZIPPED into one zip file and uploaded in Canvas. The name of this file should reflect the name of the student (abbreviated, do not make it too long).

Note: Grading is on Ubuntu. It is in your interest to make sure the program compiles and runs in the target test platform.

Testing your code on grading test cases

Grading test cases are different from the test cases provided.

To test your code against grading testcases follow these steps:

Put the four files q.h tcb.h threads.h and proj-4.c in one zip file and name the file

YOUR LAST NAMEProject4.zip (no gaps)

The resulting file should be YOUR LAST NAMEProject4.zip

Make sure there are no folders inside the zip

For example BanerjeeProject4.zip

The submission site is: http://10.218.107.121/index330FallP4.html

In order to access the submission site you need to first login to sslvpn.asu.edu using you ASU ID and password. You will be asked to download cisco vpn. Please do so. For any subsequent time you want to access the submission site you have to login to sslvpn.asu.edu.

In the site there will be an option to upload one file. Please upload YOURLASTNAMEProject3.zip file there. Then please hit upload. Wait for 5 seconds and then you can see the evaluation of your code on the grading test cases. To access the evaluation please type in

http://10.218.107.121/YOURLASTNAMEProject4.output

such as

http://10.218.107.121/ BanerjeeProject4.output

Then download the text file and open with any text editor. If your code passed one test case then the corresponding test case will show “NO ERRORS HERE”

If it failed a given test case then it will print “-----------------------------”

At the end of the file the total number of test cases you passed will be displayed.