11210CS 543200 Advanced UNIX Programming

Assignment 9 (5 pts)

Due: Oct 14, 2023, at 23:59:59

**Notice**

1. Please name your file properly; 2 pts will be taken out for incorrectly-named files.
2. Late submission is not allowed. You will receive 0 point in that case.
3. Plagiarism is not allowed. We will pass your source codes through a tool called “moss”, which is an anti-plagiarism service provided by Stanford University to determine the similarity of programs. If your program has > 20% similarity with your classmates, you will receive 0 point; no exception will be made. (Hint: Please do not share your code with fellow students. Discuss with them instead.)

**What to hand in**

1. Your implementation code. You should name it “**assignment9.c**”.
2. A makefile that can compile your code. You should name it “**Makefile**”.
3. A report that contains the answer to the questions, screenshots of your results, and explanations of how you implemented your code. You should name it “***Group\_id*.pdf**”.
4. Please upload all the code files separately and the pdf (do not zip them) to eeclass.

**Specifications**

What synchronization primitives would you need to implement a barrier? Provide an implementation of the **pthread\_barrier\_wait** function.

In your implementation, you should:

1. **(1 pt)** Use **pthread** to create **5** threads and print “*Starting thread i*” for each thread before calling the thread-create function.
2. **(2 pts)** Implement your barrier by using synchronization primitives to block the threads and await the completion of the creation process for all threads.
3. **(1 pt)** Print “*Thread # running*” after calling the barrier.
4. **(1 pt)** Make sure that your program function correctly. All threads can only run after the other threads finish starting. (see the sample output shows)

**Additional Notes**

1. **Printing the words directly is forbidden**! We will check your source codes.
2. All of your code should be implemented within one .c file.
3. **Every output of your codes should follow the same format as the sample outputs** given below, otherwise, you will not get the points.

**Sample Output**

Starting thread 0

Starting thread 1

Starting thread 2

Starting thread 3

Starting thread 4

Thread 123145321193472 running

Thread 123145321730048 running

Thread 123145322266624 running

Thread 123145320656896 running

Thread 123145320120320 running