Assignment 1(OpenMP Programming)

Due: 2014-11-03 23:59

Assignment:

P1:

1. Dining philosophers problem

Five silent philosophers sit at a round table with bowls of spaghetti. Forks are placed between each pair of adjacent philosophers.

Each philosopher must alternately think and eat. However, a philosopher can only eat spaghetti when he has both left and right forks. Each fork can be held by only one philosopher and so a philosopher can use the fork only if it's not being used by another philosopher. After he finishes eating, he needs to put down both forks so they become available to others. A philosopher can grab the fork on his right or the one on his left as they become available, but can't start eating before getting both of them.

Eating is not limited by the amount of spaghetti left: assume an infinite supply.

2. Producer–consumer problem

The problem describes several processes, the producers and the consumers, who share a common, fixed-size buffer used as a queue. The producer's job is to generate a piece of data, put it into the buffer and start again. At the same time, the consumer is consuming the data (i.e., removing it from the buffer) one piece at a time. The problem is to make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer.

The solution for the producer is to either go to sleep or discard data if the buffer is full. The next time the consumer removes an item from the buffer, it notifies the producer, who starts to fill the buffer again. In the same way, the consumer can go to sleep if it finds the buffer to be empty. The next time the producer puts data into the buffer, it wakes up the sleeping consumer.

3. Cigarette smokers problem

Assume a cigarette requires three ingredients to smoke:

- Tobacco
- Smoking paper
- A match

Assume there are also three chain smokers around a table, each of whom has an infinite supply of one of the three ingredients — one smoker has an infinite supply of tobacco, another has an infinite supply of paper, and the third has an infinite supply of matches.

Assume there is also a non-smoking arbiter. The arbiter enables the smokers to make their cigarettes by arbitrarily (non deterministically) selecting two of the smokers, taking one item out of each of their supplies, and placing the items on the table. The arbiter then notifies the third smoker that they have done this. The third smoker removes the two items from the table and uses them (along with their own supply) to make a cigarette, which they smoke for a while. Meanwhile, the arbiter, seeing the table empty, again chooses two smokers at random and places their items on the table. This process continues forever.

The smokers do not hoard items from the table; a smoker only begins to roll a new cigarette once they have finished smoking the last one. For instance if the arbiter places tobacco and paper on the

table while the match-supply smoker is smoking, the tobacco and paper will remain untouched on the table until the match-supply smoker is finished with their cigarette and then collects the items.

4. Sleeping barber problem

The barber has one barber chair and a waiting room with a number of chairs in it. When the barber finishes cutting a customer's hair, he dismisses the customer and then goes to the waiting room to see if there are other customers waiting. If there are, he brings one of them back to the chair and cuts his hair. If there are no other customers waiting, he returns to his chair and sleeps in it.

Each customer, when he arrives, looks to see what the barber is doing. If the barber is sleeping, then the customer wakes him up and sits in the chair. If the barber is cutting hair, then the customer goes to the waiting room. If there is a free chair in the waiting room, the customer sits in it and waits his turn. If there is no free chair, then the customer leaves.

Description:

You will be allocated one of these problems above according to your student id.

[Your Problem ID]=([Last Three Digits of Student ID] Mod 4)+1

Use OpenMP to simulate your problem and deal with the synchronization problems you encountered.Your test case should include all situations referred to in your problem.

P2:

Please implement Median filter algorithm in OpenMP.

Median filter:http://en.wikipedia.org/wiki/Median_filter

Description:

Everyone should complete P2 using Open MP, use optimization technology in your code to improve efficiency as much as you can.

NOTICE:

1. Marking Criterion:We will evaluate your program from several aspects.
   - Optimize Technology(40%)
   - Execution Efficiency(30%)
   - Correctness(20%)
   - Code Style(10%)
2. Write a brief report to claify your results and thought.
3. Send your final version to TA before due date.You should archive your source code and report with name StudentID_Name_Assinment1.rar(or any archive file types).
4. If you have any questions, please feel free to contact TA.
5. OpenMP Totorials: https://computing.llnl.gov/tutorials/openMP/

Warning:

No cheating,please refer to 《上海交通大学学生学业诚信守则》,it is your responsibility to take the consequences if you violate the rule.