File: EE599\_Lab9\_part1\_p1\_2176023892.cpp EE599\_Lab9\_part1\_p2\_2176023892.cpp EE599\_Lab9\_part1\_p3\_2176023892.cpp EE599\_Lab9\_part1\_p4\_2176023892.cpp

USCID: 2176023892

Email: yaog@usc.edu

Description:

EE599\_Lab9\_part1\_p1\_2176023892.cpp

following the given data structure to decode ingredient strings with the above rule and output the results into “part1\_q1\_out.txt” file

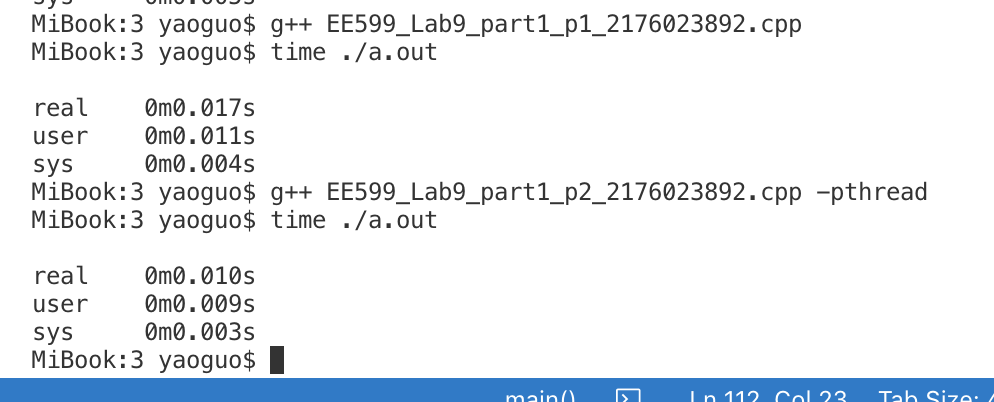
time consuming is 0.008s

EE599\_Lab9\_part1\_p2\_2176023892.cpp

Use two threads to decode ingredient strings, i.e. one thread to decode the first half of lines in the input files and the other thread to decode the last half of lines. For example, given 1000 lines, the first thread decodes the first 500 lines and the second thread decode the remaining 500 lines. Output the decoded result into “part1\_q2\_out.txt”.

If using only one thread, time consuming is 0.017

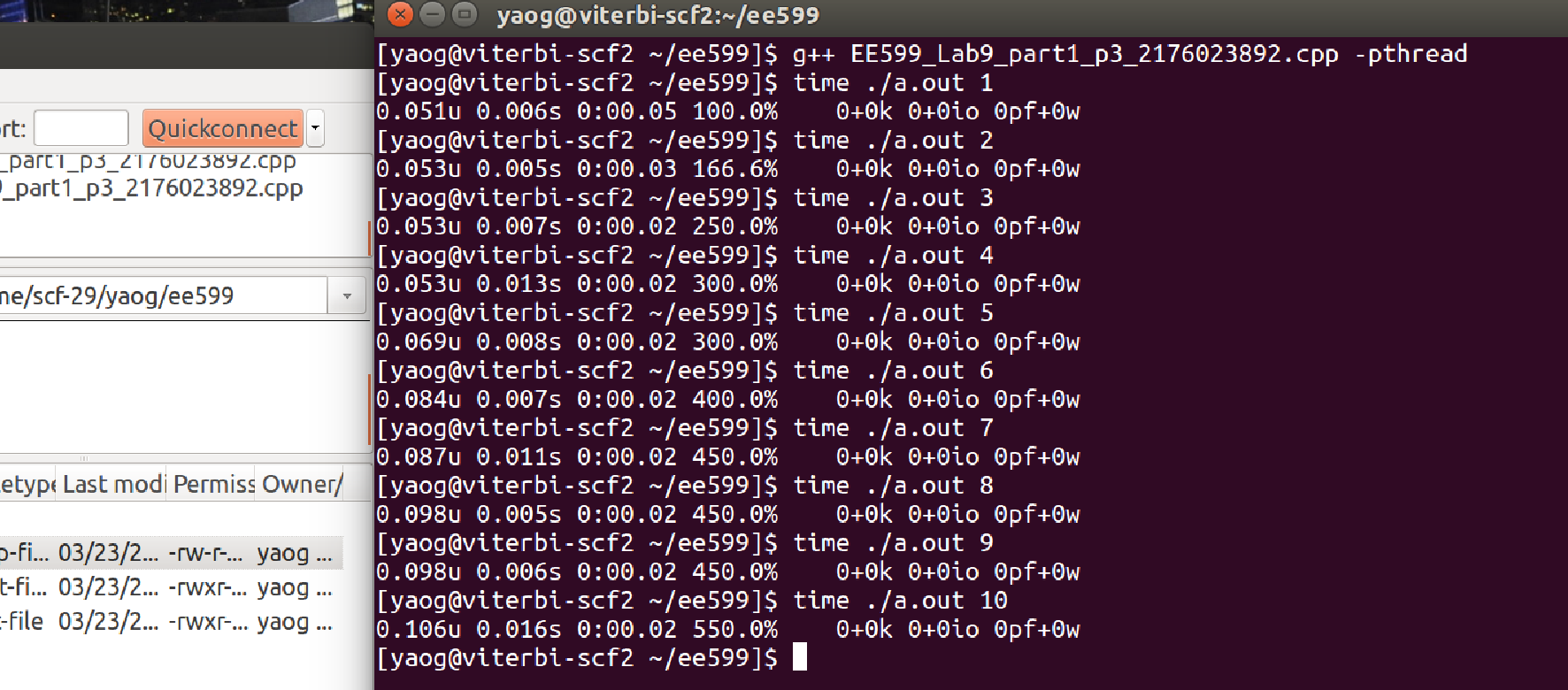
If using 2 thread, time consuming is 0.010

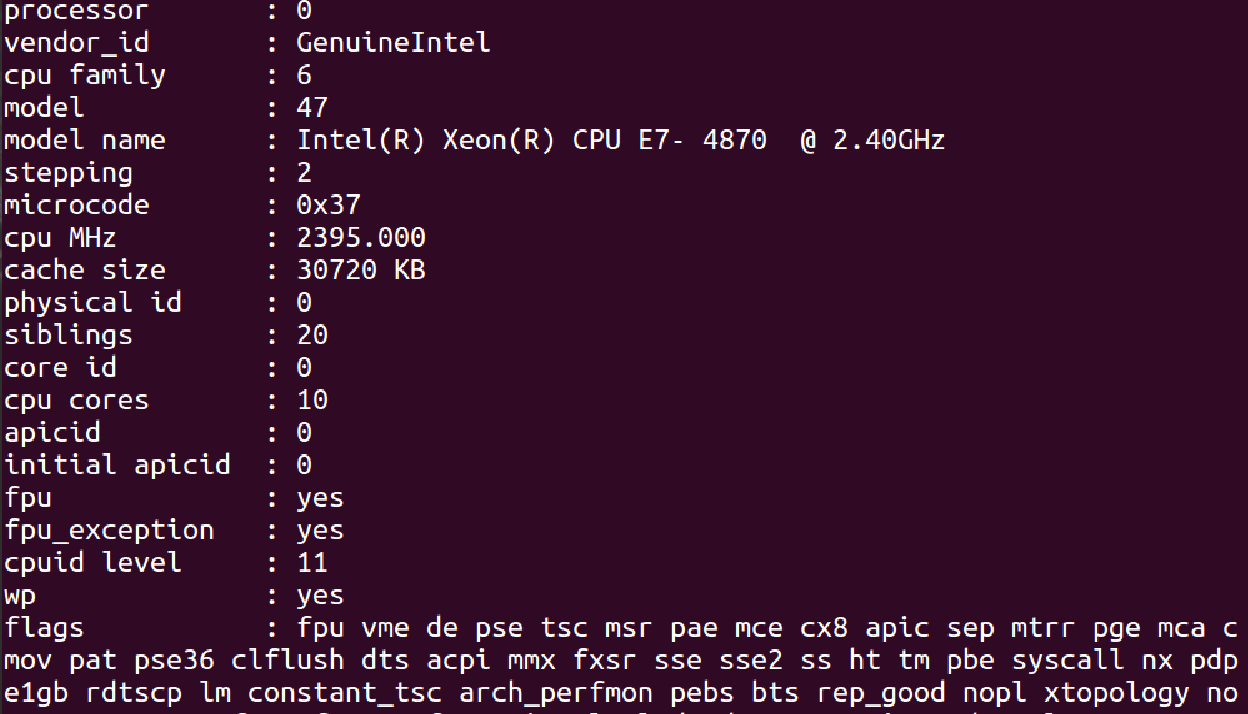


EE599\_Lab9\_part1\_p3\_2176023892.cpp

Multithreading, from 1 core to 10 cores

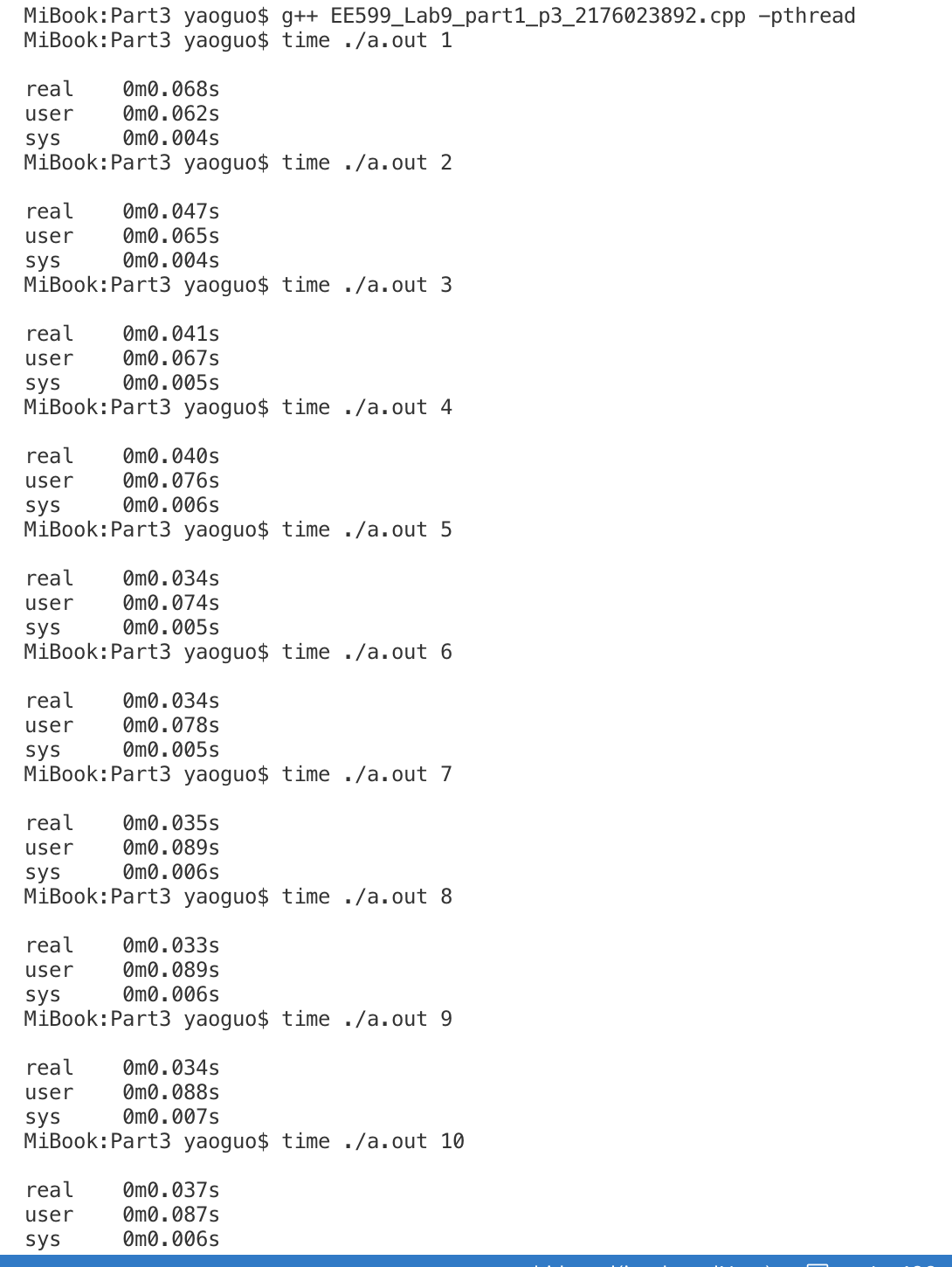
The time consuming is getting higher as cores increasing





When I used my own multi core (4cores 8threads) computer

The time consuming is showing blow.



EE599\_Lab9\_part1\_p4\_2176023892.cpp

In computing, the producer–consumer problem is a classic example of a multi-process synchronization problem. The problem describes two processes, the producer and the consumer, who share a common, fixed-size buffer used as a queue. The producer's job is to generate 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. One consideration 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. Another consideration is to make sure there would be no issues in the existence of multiple producers.

In this question, there are 2 producers and 1 consumer. Write a C/C++ program named “EE599\_Lab9\_part1\_q4\_USCID.cpp” to simulate producer-consumer problem, where the size of the buffer is passed through command line. Example: ./producer\_Consumer 10, where 10 is the size of the buffer.

Producer 1 produces 100 items and Producer 2 produces 50 items, where items are some random numbers (integers in range 0~99). The simulation should terminate after the consumer has consumed all the data.

Part of terminal output

