**Вывод:**

В результате модернизхации(?) кода лабораторной работы №3, то есть изменения редукции на синхронизацию, я получил более медленное выполнение программы, даже по сравнению с однопоточной работой. Это связано с тем, что в результате синхронизации, постоянно происходят блокировки, работа ядер сводится к постоянному ожиданиию одного из потоков.

На этом примере, хорошо видно превосходство редукции над синхронизацией.

**Контрольные вопросы**

1. Что понимается под атомарной (неделимой) операцией?

Операция в общей области памяти называется **атомарной**, если она завершается в один шаг относительно других потоков, имеющих доступ к этой памяти. Во время выполнения такой операции над переменной, ни один поток не может наблюдать изменение наполовину завершенным. Атомарная загрузка гарантирует, что переменная будет загружена целиком в один момент времени.

1. Как определяется критическая секция?

Критическая секция - блок программного кода, который может выполняться только одним потоком в каждый конкретный момент времени .

Определение критической секции в OpenMP осуществляется при помощи директивы critical, формат записи которой имеет вид:

#pragma omp critical [(name)] <block>