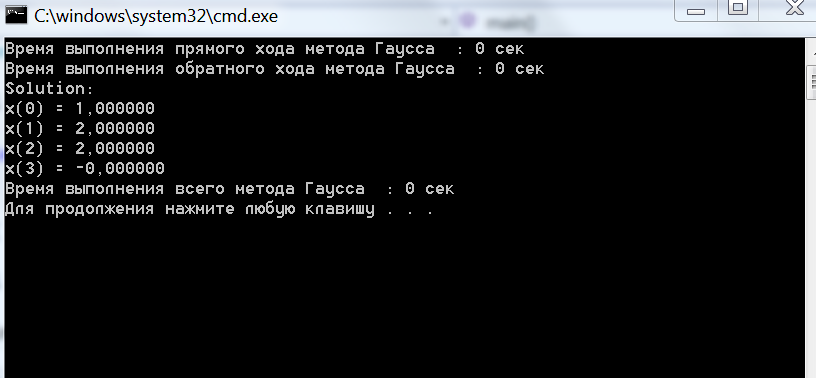
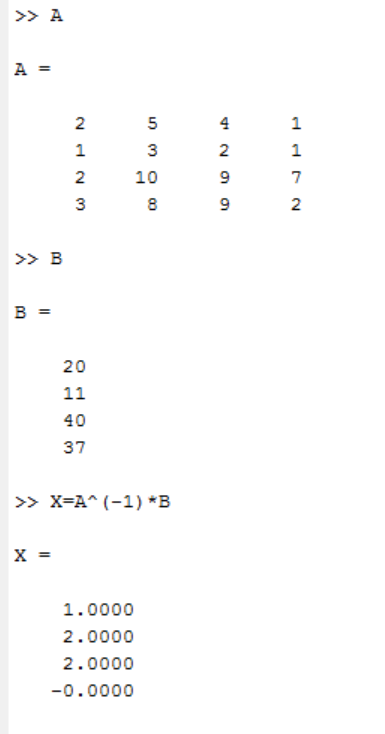
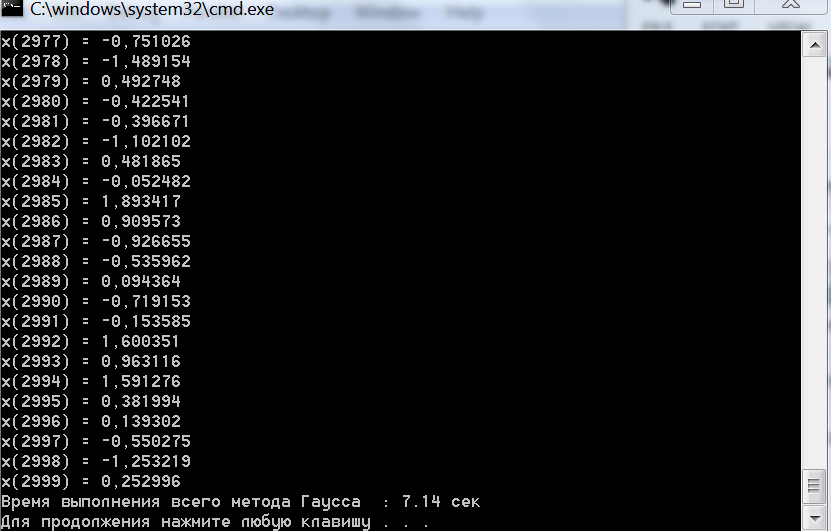
1. Запустите первоначальную версию программы и получите решение для тестовой матрицы***test\_matrix***, убедитесь в правильности приведенного алгоритма. Добавьте строки кода для измерения времени (см. задание к занятию 2) выполнения прямого хода метода Гаусса в функцию ***SerialGaussMethod()***. Заполните матрицу с количеством строк ***MATRIX\_SIZE***случайными значениями, используя функцию ***InitMatrix()***. Найдите решение СЛАУ для этой матрицы. (Закомментируйте строки кода, где используется тестовая матрица ***test\_matrix***).

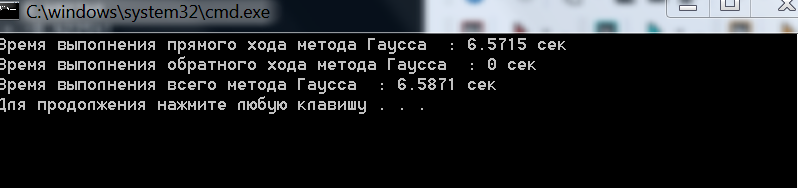




Решение тестовой матрицы верно

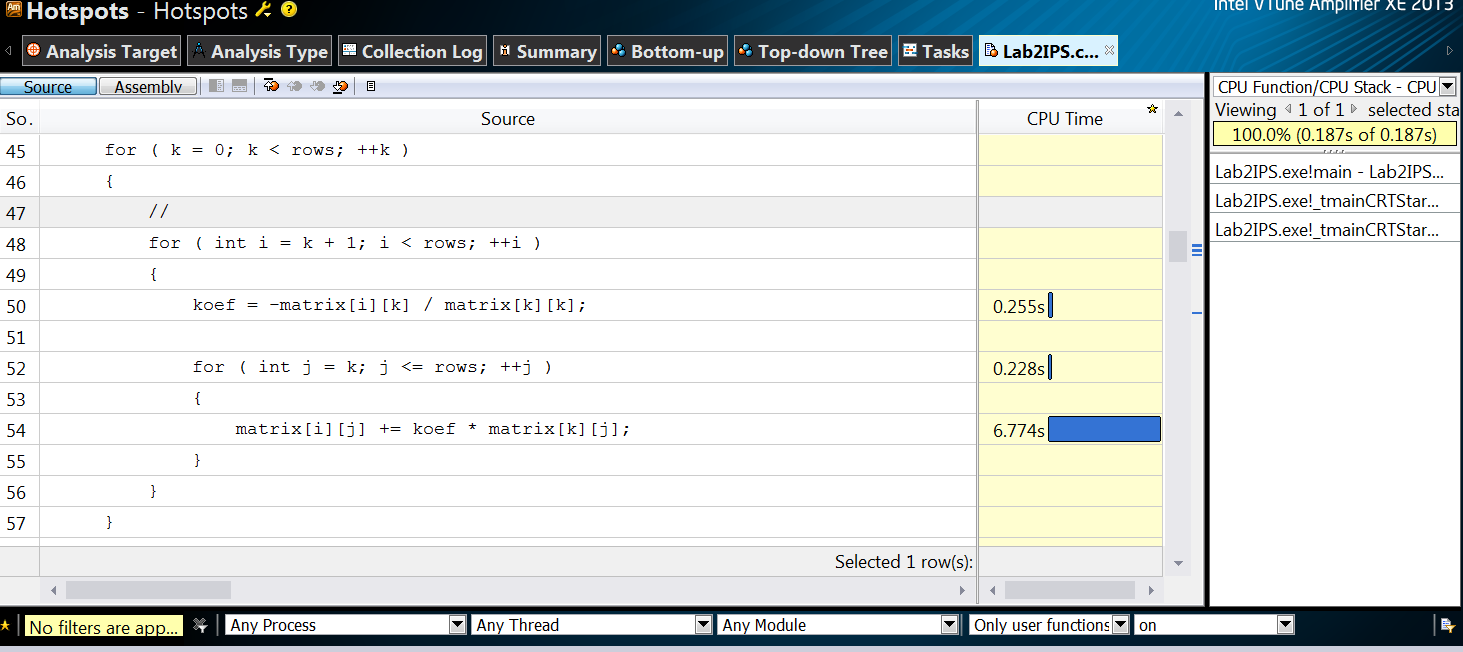
Решение произвольной матрицы ( последовательный метод Гаусса )

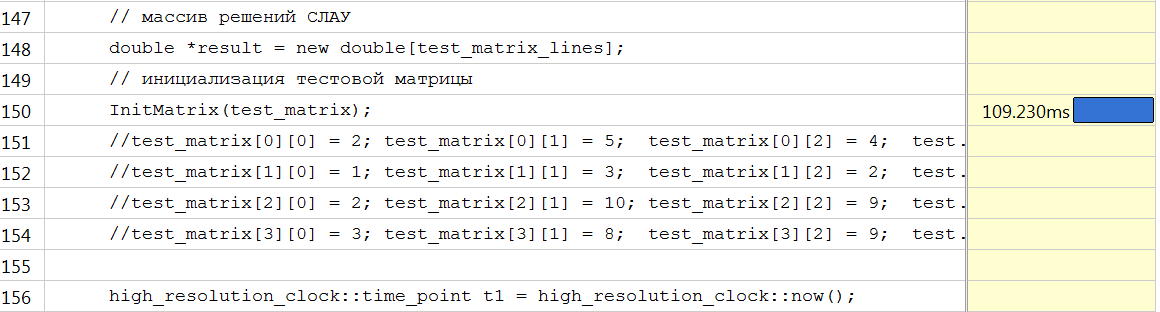


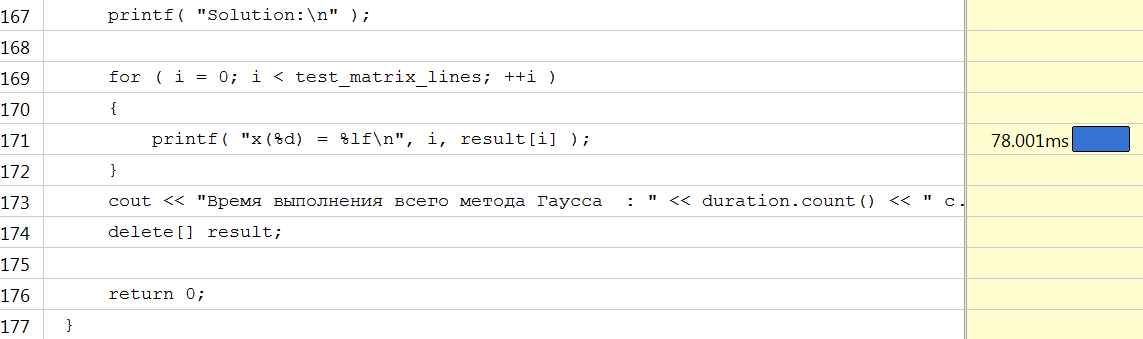


1. С помощью инструмента**Amplifier XE**определите наиболее часто используемые участки кода новой версии программы. Сохраните скриншот результатов анализа **Amplifier XE***.*Создайте, на основе последовательной функции ***SerialGaussMethod()***, новую функцию, реализующую параллельный метод Гаусса. Введите параллелизм в новую функцию, используя***cilk\_for.*Примечание:***произвести параллелизацию одного внутреннего цикла прямого хода метода Гаусса (определить какого именно), и внутреннего цикла обратного хода. Время выполнения по-прежнему измерять только для прямого хода.*

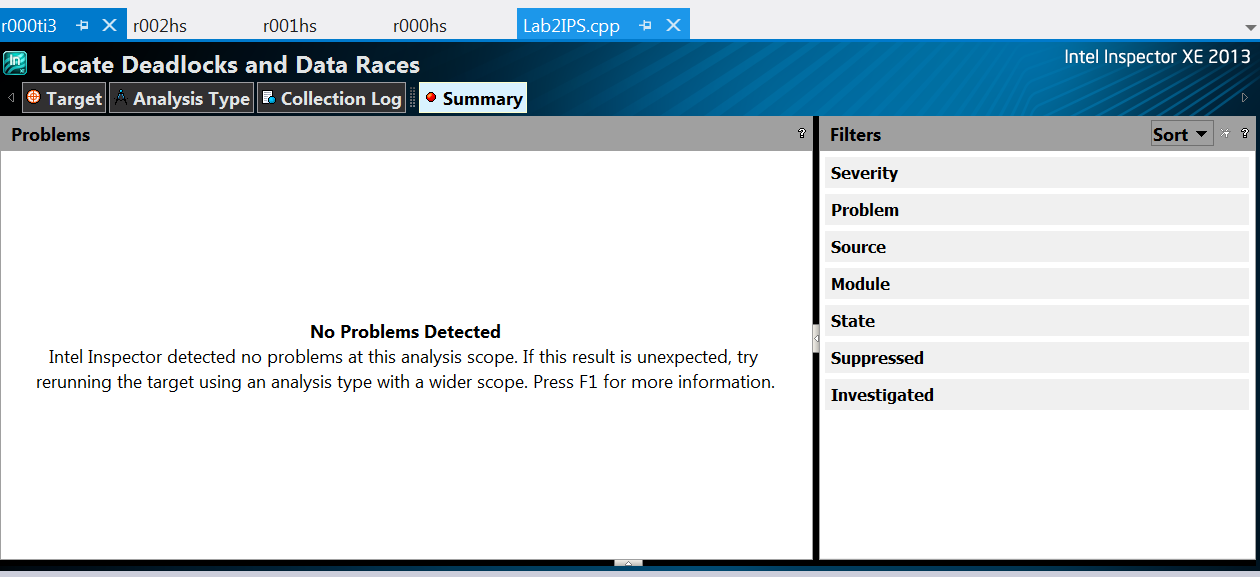
Amplifiler XE ( последовательный метод Гаусса )





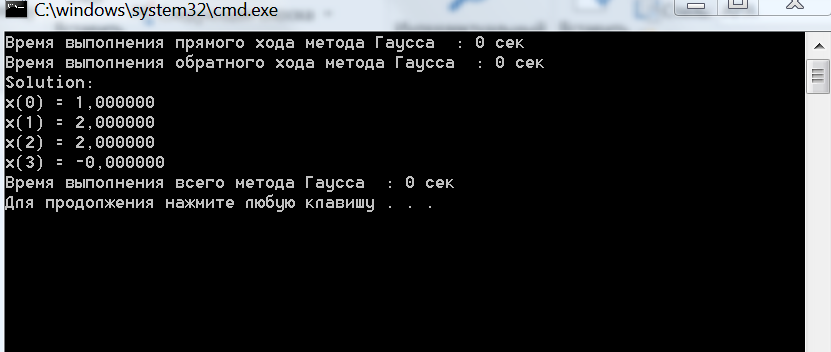


1. Далее, используя***Inspector XE*,**определите те данные (если таковые имеются), которые принимают участие в гонке данных или в других основных ошибках, возникающих при разработке параллельных программ, и устраните эти ошибки. Сохраните скриншоты анализов, проведенных инструментом ***Inspector XE***: в случае обнаружения ошибок и после их устранения.



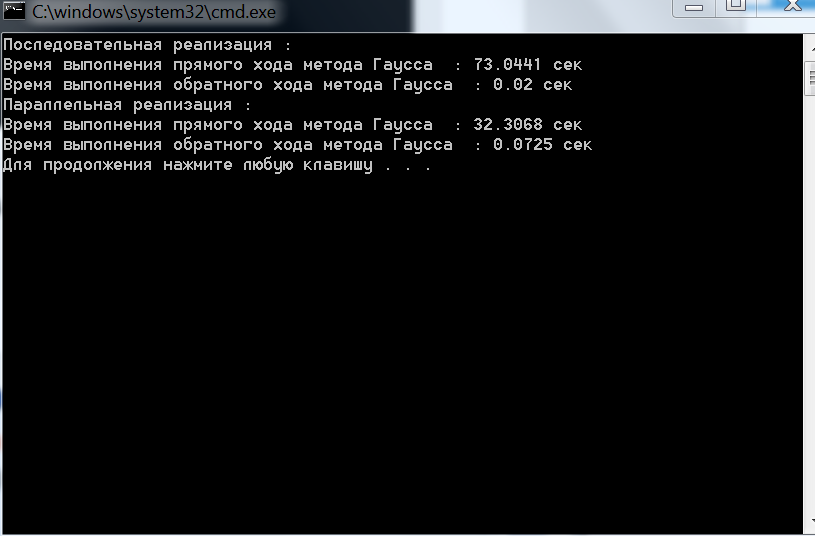
1. Убедитесь на примере тестовой матрицы ***test\_matrix***в том, что  функция, реализующая параллельный метод Гаусса работает правильно. Сравните время выполнения прямого хода метода Гаусса для последовательной и параллельной реализации при решении матрицы, имеющей количество строк***MATRIX\_SIZE,***заполняющейся случайными числами. Запускайте проект в режиме**Release**, предварительно убедившись, что включена оптимизация (***Optimization->Optimization=/O2***). Подсчитайте ускорение параллельной версии в сравнении с последовательной. Выводите значения ускорения на консоль.

Заустим тестовую матрицу с параллельным методом Гаусса



Решение совпадает

Сравним время выполнения параллельной и послед реализации при матрице размера 5000 на 5000



Ускорение программы :

