Индивидуальное задание № 3

«Применение OMP»

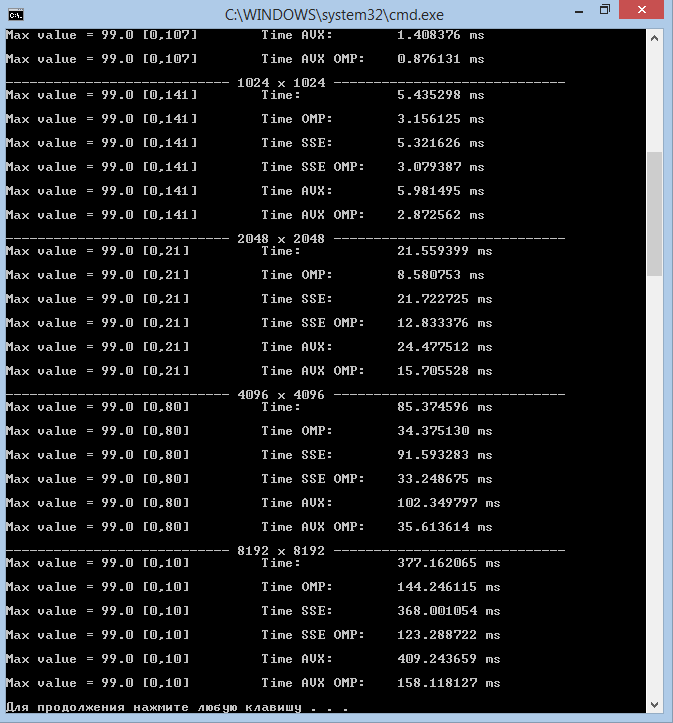
Выполнил:

Ст. гр. ПИ-13-6

Кравченко Александр

1. Нахождение максимального элемента в матрице (индекс строки, индекс столбца)

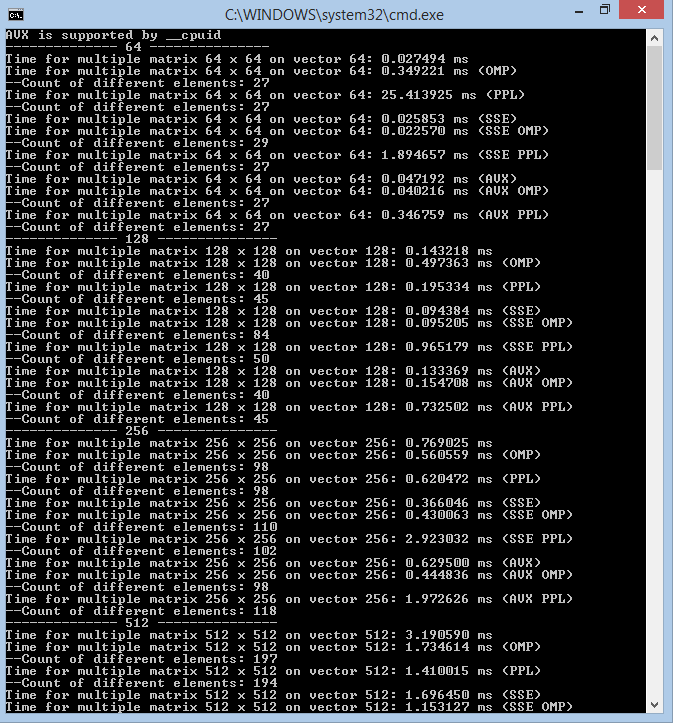
Debug 32-bit SIMPLE, OMP



Вывод: при увеличении размерности матриц видно ускорение алгоритма с применением OMP директив (до 3 раз с SSE). Идеальное ускорение равно 4 (кол-во ядер), однако теряется время под нагрузку построения самой директивы.

1. Умножение матрицы на вектор

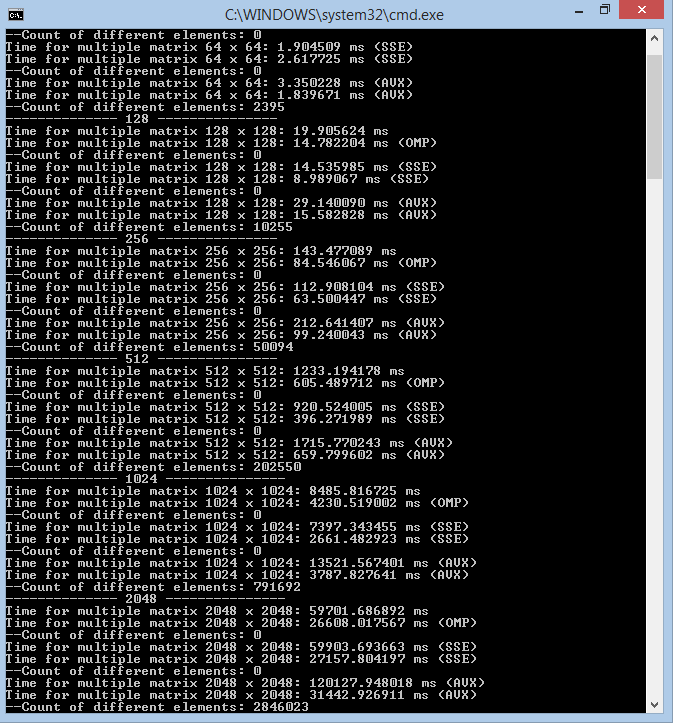
Release X64 SIMPLE, OMP, PPL



Вывод: с ростом размерности наблюдается рост разных значений. Это связано с тем, что при разной очереди подсчета значений с плавающей точкой получаются разные результаты. Лучший результат по времени выполнения показал алгоритм с применением SSE и OMP (ускорение 2.5), после него AVX и OMP (ускорение 2), и на третьем месте SSE и PPL (1.9).

1. Умножение матрицы на матрицу

Debug 32-bit SIMPLE, OMP



Вывод: с увеличением размерности увеличивается ускорение алгоритма с применением OMP директив (до 3.25 раз в режиме Release 32bit). Кол-во разных элементов не равно нулю, так как при разной очереди подсчета значений с плавающей точкой, имеем разные результаты.

1. Вывод: по ходу выполнения индивидуального домашнего задания я убедился, что переход на использование команд директивы OMP и PPL для распараллелирования при обработке массивов дает очень хорошую прибавку к экономии времени. Испытания были проведены на компьютере с 4 ядрами, значит, что максимальное ускорения равно 4. Я получал максимальное ускорение 3.25 раза (для OMP) и 2.0 (для PPL). Это связано с нагрузками самой директивы, а также внешними условиями в виде работающих пользовательских процессоров.