

Задание 4. Умножение матриц.

Папченко Анастасия, группа 538

1 Постановка задачи

С помощью интринсик используемой целевой архитектуры (AVX для Intel, AMD; NEON для ARM) реализовать векторизованную версию матричного умножения ($A * B = C$). Можно предполагать, что матрицы A, B - квадратные. Обязательное требование - хранение всех матриц предполагается в едином порядке (все в row-major либо все в col-major). Тип элементов матрицы - float для 128-битных, double для 256-битных векторных расширений. Сравнить результаты и время выполнения векторизованного алгоритма с его последовательной версией. ($N = 512, 1024, 2048$).

2 Компиляция

```
gcc -o2 -fopenmp -mavx2 matrix_mult.c -o matrix_mult
```

3 Формат командной строки

```
./matrix_mult
```

4 Спецификация системы

Процессор: AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz
Число вычислительных ядер: 4

5 Результаты выполнения

Размер матрицы (N * N)	Время выполнения последовательного умножения, сек	Время выполнения векторизованного умножения, сек
512 * 512	1.040647	0.568597
1024 * 1024	28.106371	5.692754
2048 * 2048	361.262685	78.262959

Из полученных результатов можно сделать вывод, что при размере матрица 512 время выполнения векторизованного умножения почти в 2 раза быстрее, при размере 1024 быстрее в 5 раз и при размере в 2048 тоже быстрее в 5 раз.

При выполнении задания возникла проблема с распределением данных. Программа при запуске давала ошибку: Segmentation fault (core dumper). Данную проблему удалось решить с помощью явного выравнивания массива.

`__attribute__((aligned (32))) double temp[4]` - гарантирует, что массив будет выровнен по границе 32 байта в памяти.