



**Спецкурс: системы и средства параллельного
программирования.**

Отчёт № 1.

**Анализ влияния кэша на операцию матричного
умножения.**

Работу выполнил
Гайсин Р. Р.

Постановка задачи и формат данных.

Задача: Реализовать последовательный алгоритм матричного умножения и оценить влияние кэша на время выполнения программы.

Формат командной строки: <имя файла матрицы A > <имя файла матрицы B > <имя файла матрицы C > <режим, порядок индексов>.

Режимы: 0 – ijk, 1 – ikj, 2 – kij, 3 – jik, 4 – jki, 5 – kji.

Формат файла-матрицы: Матрица представляются в виде бинарного файла следующего формата:

Тип	Значение	Описание
Число типа char	T – f (int32_t) или l (int64_t)	Тип элементов
Число типа size_t	N – натуральное число	Число строк матрицы
Массив чисел типа T	N x N элементов	Массив элементов матрицы

Элементы матрицы хранятся построчно.

Описание алгоритма.

Математическая постановка: Алгоритм матричного умножения ($A \times B = C$) можно представить в следующем виде: $c_{ij} = \sum_k (a_{ik} \cdot b_{kj})$ для каждого элемента матрицы C .

Оценка влияния кэша на время выполнения программы осуществляется за счёт перестановки индексов суммирования.

Анализ времени выполнения: Для оценки времени выполнения программы использовалась функция: clock(). Для повышения надёжности экспериментов опыты проводились несколько раз (10).

Верификация: Для проверки корректности работы программы использовались тестовые данные.

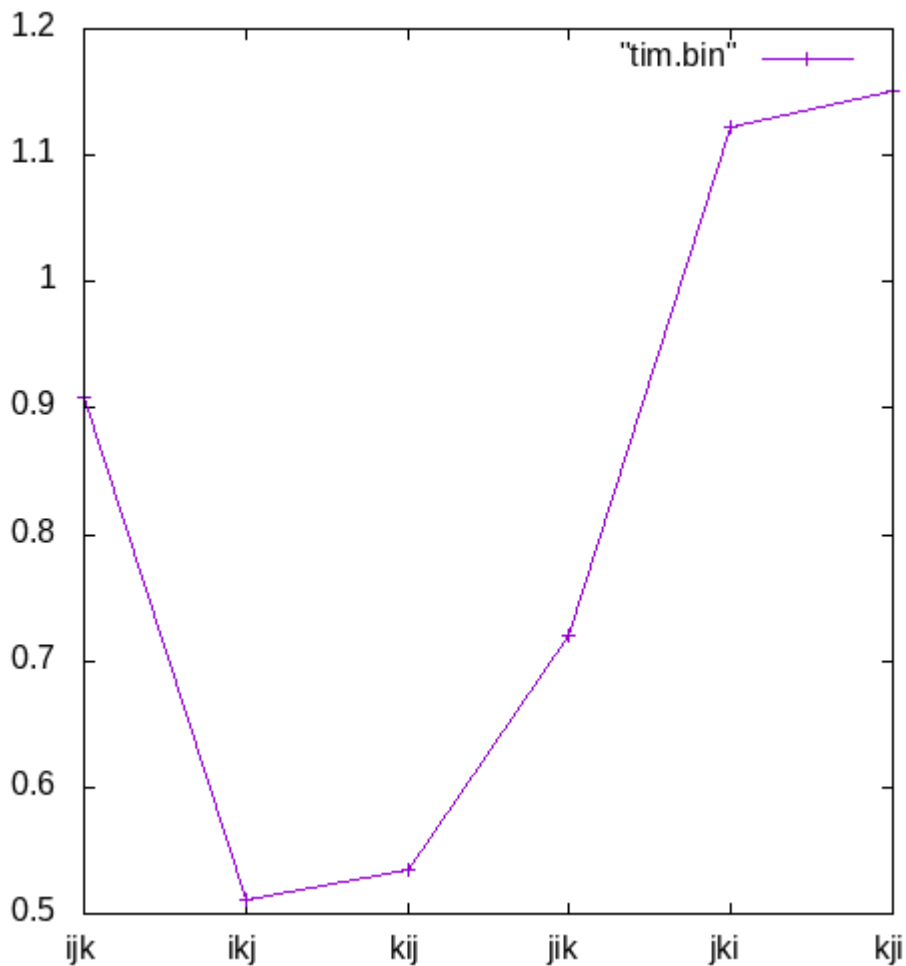
Основные функции:

- **Разбор командной строки.** В рамках функции осуществляется анализ и разбор командной строки.
- **Чтение файлов матриц.** В рамках функции осуществляется анализ совместимости входных матриц и их чтение.
- **Перемножение матриц.** В рамках функции осуществляется перемножение матриц в соответствии с выбранным порядком индексов суммирования.

Результаты выполнения.

Результаты:

Проводилось перемножение двух матриц размерами 400 x 400. Зависимость времени выполнения от порядка индексов суммирования представлена на графике (время в секундах).



Основные выводы.

Исследования показывают, что изменения порядка индексов суммирование оказывает влияние на время выполнения программы. Наименьшее время выполнения при следующем порядке индексов - ikj . При таком порядке доступ к элементам обеих входных матриц осуществляется последовательно. Наихудшее время при порядке kji . При таком подходе доступ к памяти осуществляется максимально непоследовательно.