

# Конкурс на самую быструю реализацию матричного умножения

В случаях, когда требуется достигнуть максимальной производительности приложения, часто приходится оптимизировать код с учетом архитектурных особенностей конкретных процессоров: учитывать доступные расширения ISA, размеры кэша различных уровней и многое другое.

Во многих численных алгоритмах (например, в методе нахождения обратной матрицы, описанном в лабораторной о векторизации) наиболее вычислительно сложной операцией является матричное умножение. Именно от её эффективности зависит время работы всей программы. Вам предлагается достичь максимальной производительности данной операции на процессоре Intel Core i7-10700 (2.9 ГГц) при работе с матрицами размера **4096x4096**.

Тестовый компьютер с указанным процессором доступен по протоколу SSH по адресу [10.2.70.220](https://10.2.70.220). Работать с тестовым компьютером можно ТОЛЬКО ИЗ СЕТИ НГУ (общежития также входят в сеть НГУ). Для работы из других сетей можно использовать доступ по VPN. Также можете заходить на кафедраальный сервер, а через него – на тестовый компьютер, используя команду «ssh -i *your\_priv\_key* [userXX@10.2.70.220](https://10.2.70.220)», где *your\_priv\_key* – Ваш сгенерированный приватный ключ, а *userXX* – Ваша учетная запись. На сервер все студенты ходят одной учетной записью (evmpu), поэтому Ваш ключ будет виден всем. В связи с этим особенно важно задавать парольную фразу.

Каждому желающему поучаствовать в конкурсе необходимо В СРОК ДО **10 ДЕКАБРЯ** выслать на почтовый адрес [a.vlasenko@g.nsu.ru](mailto:a.vlasenko@g.nsu.ru) письмо с темой «КОНКУРС» с указанием ФИО, номера группы и файлом с открытой частью RSA-ключа (длина по умолчанию – 2048 бит). Сгенерировать RSA-ключ, состоящий из файлов закрытой и открытой частей можно при помощи программы *puttygen* под Windows, либо *ssh-keygen* под Linux. При генерации ключа настоятельно рекомендуется задать парольную фразу (passphrase). По данной заявке для Вас будет создана индивидуальная учетная запись на тестовом компьютере, о чем будет сообщено ответным письмом по электронной почте.

На тестовом компьютере в Вашей домашней директории будет размещаться папка *matr\_bench* следующего содержания:

- *mybench.cpp* //Бенчмарка – ее модифицировать не нужно
- *multiply.cpp* //Ваше основное поле работы
- *multiply.h*
- *СmakeLists.txt* //Нужен только, если захотите использовать *СMake*

### Для участия в конкурсе Вам необходимо:

1. Заменить в файле *mybench.cpp* функцию *multiply(float \*mat1, float \*mat2, int N, float \*res)* умножения матрицы на матрицу на свою оптимизированную реализацию под процессор Intel Core i7-10700.
2. Скомпилировать программу с подключением библиотеки Google Benchmark при помощи следующей команды:  

```
g++ -march=native -O3 -fno-tree-vectorize multiply.cpp mybench.cpp -lbenchmark -pthread -o mulBench
```

*(именно с ключами “-march=native -O3 -fno-tree-vectorize” будет компилироваться код при контрольных замерах)*  
Программу также можно скомпилировать при помощи CMake (см. пункт «Сборка с использованием CMake»)
3. Протестировать реализацию. Для проведения, например, 5-ти тестов подряд можно использовать следующую команду:  

```
./mulBench --benchmark_repetitions=5
```
4. Выслать файл *multiply.cpp* с собственной реализацией функции *multiply(float \*mat1, float \*mat2, int N, float \*res)* по адресу [a.matveev3@g.nsu.ru](mailto:a.matveev3@g.nsu.ru). Тема письма «Конкурс ЭВМиПУ. НОМЕР\_ГРУППЫ ФАМИЛИЯ ИМЯ»
5. Дождаться письма с результатом на электронную почту и появления своего результата в таблице лидеров (<https://docs.google.com/spreadsheets/d/e/2PACX-1vSWPHBnrAzG52X72UmVV0lwfrH0csCCVFzsIS2itNhO0jeAki4ACXxCHK2m5aaU-Y8SGZQpL1guq3xy/pubhtml?gid=1414876569>). Результаты будут проранжированы по времени CPU.
6. Если результат Вас не устроил и Вы смогли реализовать умножение еще более эффективно, то можете выслать на [a.matveev3@g.nsu.ru](mailto:a.matveev3@g.nsu.ru) обновленный вариант.

### Сборка с использованием CMake

1. В директории *matr\_bench* создайте папку в которой будет выполняться сборка (например “*mkdir bench\_build*”) и перейдите в неё (“*cd bench\_build*”);
2. Выполните генерацию скриптов сборки на основе проекта CMake с помощью команды “*cmake ..*”
3. Выполните сборку проекта (команда “*make*”);
4. В случае успешной сборки в папке *bench\_build* будет создан исполняемый файл *mulBench*, запустите его для выполнения тестирования производительности.

### Ограничения:

1. Реализация на языке C или C++.
2. Компилятор GCC 8.5.0.
3. Метод перемножения - на ваш выбор.
4. Разрешается использовать любые расширения набора команд, представленные на процессоре тестового компьютера.
5. Допускаются только однопоточные реализации (то есть задействовать инструментарию подобные OpenMP, Pthreads и т.д., позволяющие создавать многопоточные приложения, запрещено).
6. Запрещается использовать библиотеки линейной алгебры, подобные BLAS.

Завершение конкурса и подведение итогов – **25 декабря**.

### **НАГРАДА**

**Трое студентов**, написавшие самые быстрые реализации, получают 5 "автоматом" за теоретическую часть курса ЭВМиПУ.