

Вариант №1

Вычислить векторное произведение квадратных матриц A и B . Входные данные: произвольные квадратные матрицы A и B одинаковой размерности. Размер матриц задается входным параметром. Количество потоков является входным параметром, при этом размерность матриц может быть не кратна количеству потоков.

Модель.

Мною была использована итеративная модель параллелизма. Я выбрал ее, так как посчитал удобным разбить векторное произведение матриц на подсчет каждого элемента новой матрицы по отдельности. То есть, если на вход программа получает две матрицы 3×3 , то в итоге получится 9 потоков, как и количество элементов новой матрицы. И, соответственно, в каждом потоке для нахождения элемента c_{ij} понадобится провести поэлементное умножение i -ой строки первой матрицы с j -ым столбцом второй матрицы.

Таким образом у меня есть функция, которая считает один элемент векторного умножения матриц. То есть она принимает на вход результирующую матрицу и две матрицы, над которыми необходимо провести операцию матричного векторного умножения. Так же она принимает размерность матриц и номер строки и столбца искомого элемента. Далее я запускаю цикл от 0 до кол-ва элементов матрицы ($N * N$) и через `shared` делаю общими для всех потоков матрицы и их размерность. Далее через `private` я передаю копию i для каждого потока для расчета номера строки и номера столбца.

Далее через `barrier` я дожидаюсь, когда все потоки закончат исполнение и запускаю вывод матриц на экран.

Вся информация была взята с сайта softcraft.ru.

Тестирование.

В моей программе на вход подается размерность квадратных матриц и кол-во потоков. Сами матрицы генерируются с помощью `rand()` в интервале $[0..9]$ для простоты дальнейшей проверки.

Если программа получает некорректные значения, то просто завершает работу (не аварийно).

```
Enter size of quadratic matrix's: -10
Process finished with exit code 0
```

```
Enter size of quadratic matrix's: 0
Process finished with exit code 0
```

```
Enter size of quadratic matrix's: f
Process finished with exit code 0
```

Далее я провел тесты на матрицах размера от 1 до 3 (на матрицах большего размера сложно оценить работу программы) с разным кол-вом потоков, как кратным кол-ву задач, так и нет.

```

Enter size of quadratic matrix's:1
Enter quantity of threads:1
-----
First matrix
-----
1
-----
Second matrix
-----
7
-----
Result of multiplying
-----
7

```

```

Enter size of quadratic matrix's:2
Enter quantity of threads:3
-----
First matrix
-----
1      4
9      8
-----
Second matrix
-----
7      0
4      8
-----
Result of multiplying
-----
23     32
95     64
-----
Enter size of quadratic matrix's:2
Enter quantity of threads:4
-----
First matrix
-----
1      4
9      8
-----
Second matrix
-----
7      0
4      8
-----
Result of multiplying
-----
23     32
95     64

```

```

Enter size of quadratic matrix's:2
Enter quantity of threads:1
-----
First matrix
-----
1      4
9      8
-----
Second matrix
-----
7      0
4      8
-----
Result of multiplying
Enter size of quadratic matrix's:3
Enter quantity of threads:9
-----
First matrix
-----
1      4      9
8      2      5
1      1      5
-----
Second matrix
-----
7      0      4
8      4      5
7      1      2
-----
Result of multiplying
-----
102     25     42
107     13     52
50      9      19

```

Далее я приведу скриншоты с распределением потоков по подсчету элементов матрицы. Соответственно элемент матрицы $[i, j]$, где i – номер строки, а j – номер столбца.

Тесты для матриц 3x3:

```
Enter size of quadratic matrix's:3
Enter quantity of threads:1
[0, 0] - thread(0)
[0, 1] - thread(0)
[0, 2] - thread(0)
[1, 0] - thread(0)
[1, 1] - thread(0)
[1, 2] - thread(0)
[2, 0] - thread(0)
[2, 1] - thread(0)
[2, 2] - thread(0)
```

```
Enter size of quadratic matrix's:3
Enter quantity of threads:2
[0, 0] - thread(0)
[0, 1] - thread(0)
[0, 2] - thread(0)
[1, 0] - thread(0)
[1, 1] - thread(0)
[1, 2] - thread(1)
[2, 0] - thread(1)
[2, 1] - thread(1)
[2, 2] - thread(1)
```

```
Enter size of quadratic matrix's:3
Enter quantity of threads:3
[1, 0] - thread(1)
[1, 1] - thread(1)
[1, 2] - thread(1)
[0, 0] - thread(0)
[0, 1] - thread(0)
[0, 2] - thread(0)
[2, 0] - thread(2)
[2, 1] - thread(2)
[2, 2] - thread(2)
```

```
Enter size of quadratic matrix's:3
Enter quantity of threads:9
[1, 2] - thread(5)
[0, 0] - thread(0)
[1, 1] - thread(4)
[0, 1] - thread(1)
[2, 0] - thread(6)
[2, 1] - thread(7)
[1, 0] - thread(3)
[0, 2] - thread(2)
[2, 2] - thread(8)
```

```
Enter size of quadratic matrix's:3
Enter quantity of threads:5
[0, 2] - thread(1)
[1, 0] - thread(1)
[0, 0] - thread(0)
[0, 1] - thread(0)
[2, 2] - thread(4)
[1, 1] - thread(2)
[1, 2] - thread(2)
[2, 0] - thread(3)
[2, 1] - thread(3)
```