Practical assignment 1

Basic operations with programming threads

The purpose of the work

To consider basic operations with threads of execution, to learn how to use non-blocking parallelization to solve the simplest mathematical problems, using the chosen programming language. Learn how to research and evaluate the effectiveness of algorithm parallelization.

Task

1. Determine the main characteristics of a PC that affect the efficiency of performing parallel calculations. Record the value of these characteristics for the student's PC.

2. Create a mechanism that can be used to measure the execution time of the program, or other parameters that the student considers relevant.

3. Solve the problem selected according to the variant, without using parallelization. Measure the time to solve the problem, or other parameters that the student considers relevant.

4. Solve the problem chosen according to the option, using parallelization. Measure the time to solve the problem, or other parameters that the student considers relevant. To justify the choice of parallelization algorithm.

5. Repeat point 4 using a different number of execution threads. It is necessary to check the execution of the task on a fixed number of threads: 2 times less than the number of physical cores, on a number equal to the number of physical cores, on a number equal to the number of logical cores, on a number 2, 4, 8, 16 times greater than the number of logical cores.

6. Repeat point 5 using different dimensions of the data, depending on the chosen task.

7. Fill in the table and make a graph of the task execution time from the number of threads for different dimensions.

Variants

n = 50, 100, 500, 1000, 10000

1. Create a square matrix n×n and a vector of the same dimension; write a parallel program for multiplying a matrix by a vector.

2. Create two square matrices of size n×n; write a program for finding the sum of two matrices multiplied by a value ($M = k_1*A + k_2*B$).

3. Create two square matrices of size n×n; write a program for finding the difference of two matrices ($M = k_1*A - k_2*B$).

4. Transpose the n×n matrix.

5. Fill the square matrix with random numbers. On the main diagonal, place the sums of the elements that lie in the same row.

6. Fill the square matrix with random numbers. On the main diagonal, place the sums of the elements that lie in the same column.

7. Fill the square matrix with random numbers. On the lateral diagonal, place the product of the elements that lie in the same row.

8. Fill the square matrix with random numbers. On the side diagonal, place the product of the elements that lie in the same column.

9. Fill the square matrix with random numbers. Place the maximum row element on the main diagonal.

10. Fill the square matrix with random numbers. Place the maximum column element on the main diagonal.

Control Questions

1. Define the concepts: process, thread. Explain how these concepts are related to each other. The main differences between the process and the thread.

2. Explain threads mechanisms in the programming language chosen by the student, explain parameters transferring. Join operations. Memory management.

3. List the categorization according to Flynn. Explain each of the individual categories.

5. Amdal's law. Coefficient of acceleration and efficiency of the parallel algorithm.

6. List the types of parallelism, describe each of them.

Evaluation

Report	2
Code correctness	6
Charts	2
Total	**10**

Absence of Git will lead to 0 points. https://classroom.github.com/a/ZylsG9YB
Incorrect answers on theoretical questions will lead to 0 points.