**Лабораторная работа № 2**

**«Введение в OpenMP. Операции с массивами. Быстрая сортировка»**

**Целью** лабораторной работы №2 является приобретение *знаний, умений и навыков* в области технологии параллельного программирования средствами библиотеки OpenMP. Для достижения поставленной цели необходимо решение следующих основных **задач**:

1. Изучить основы технологии параллельного программирования средствами библиотеки OpenMP.
2. Изучить способы разделения задач между потоками.
3. Способом параллельных циклов научиться решать задачи сложения (и других арифметических операций) элементов.
4. Способом параллельных задач научиться реализовывать параллельную работу рекурсивных функций на примере быстрой сортировки элементов.

**Задания**.

1. Написать последовательный и параллельный вариант программы вычисления суммы элементов массива (количество элементов массива более 100000). Запустить на суперкомпьютере и в своей среде программирования. Использовать разные варианты количества используемых потоков. Проанализировать результаты.
2. Написать последовательный и параллельный вариант программы быстрой сортировки массива (количество элементов массива более 100000). Запустить на суперкомпьютере и в своей среде программирования. Использовать разные варианты количества используемых потоков. Проанализировать результаты.
3. Написать последовательный и параллельный вариант программы работы с двумя одномерными массивами (количество элементов массивов одинаково и более 100000), в которой будет реализованы операции сложения, вычитания, умножения, деления элементов с одинаковыми индексами. Запустить на суперкомпьютере и в своей среде программирования. Использовать разные варианты количества используемых потоков. Проанализировать результаты.
4. Написать последовательный и параллельный вариант программы работы с двумя двумерными массивами (количество элементов массивов одинаково и более 100000, размерности совпадают): операции сложения, вычитания, умножения, деления элементов с одинаковыми индексами. Запустить на суперкомпьютере и в своей среде программирования. Использовать разные варианты количества используемых потоков. Проанализировать результаты.