

Задание по курсу
«Технологии параллельного программирования»

Лабораторная работа №3

1. Одним из параллельных методов сортировки массива является четно-нечетная сортировка. Каждый шаг этого метода состоит из двух этапов. На первом этапе элементы массива разбиваются на пары, где первый элемент имеет четный номер:

$$(a_0, a_1), (a_2, a_3), (a_4, a_5), \dots, (a_{2k}, a_{2k+1}), \dots$$

В каждой паре проводится сравнение элементов и, если первый элемент больше второго, то они меняются местами.

На втором этапе элементы разбиваются на пары, где первый элемент имеет нечетный номер:

$$a_0, (a_1, a_2), (a_3, a_4), \dots, (a_{2k-1}, a_{2k}), \dots$$

и снова проводится сравнение и перестановка элементов.

При таком подходе операции перестановок элементов в каждой паре независимы, поэтому алгоритм легко распараллеливается. Для массива размера n , алгоритм отсортирует массив за n шагов.

Задание. Реализовать параллельную версию четно-нечетной сортировки при помощи директивы **sections**.

2. Ускорение параллельной программы определяется по формуле:

$$S = \frac{T_1}{T_p}$$

где T_1 — время выполнения последовательной программы; T_p — время выполнения параллельной программы.

Задание. Написать цикл, состоящий из $N = 10000$ итераций и распараллеленный при помощи директивы **for**. На k -й итерации цикла выполняются следующие действия ($k = 1, \dots, N$):

- 1) определяется массив из k элементов, заполненный случайными значениями;
- 2) выполняется сортировка этого массива (любым методом).

Определить ускорение этой программы при выполнении 2-мя и 4-мя потоками для способов распределения итераций **static**, **dynamic** и **guided**, т.е. всего для 6 случаев. Время выполнения программы для каждого случая определять как среднее арифметическое для 5 запусков.