

# Параллельное программирование с использованием стандарта OpenMP

## Использование директив parallel sections и atomic

(Всего нужно набрать 8 баллов)

В предложенных задачах будут участвовать *массивы* большого размера. Их лучше сделать *глобальными* переменными, чтобы они не занимали места в стеке (если заводить очень большие массивы как локальные переменные в какой-либо функции, могут начаться проблемы, вызванные нехваткой места в стеке).

В каждой задаче изучите, как меняется ускорение от параллелизма с ростом  $n$ . Начиная с каких значений  $n$  проявляется выгода от параллельности?

Для  $n = 10^8$  занесите результаты измерений в таблицу (можно на листке бумаги):

Задача №	Время в миллисекундах		
	Последовательная программа	Параллельная (2 секции)	Параллельная (4 секции)
1			
2			

- (1 балл) Вычисление таблицы синусов. Вычислите параллельно и последовательно таблицу синусов на отрезке  $[0, \pi/2]$  с шагом  $\frac{\pi}{2n}$ , где  $n$  – параметр задачи.
- (2 балла) Вычисление числа  $\pi$  по формуле

$$\pi = 4 \arctg 1 = 4 \int_0^1 \frac{dx}{1+x^2}$$

с применением приближенного вычисления определенного интеграла по формуле средних прямоугольников

$$\int_a^b f(x)dx \approx \frac{1}{n} \sum_{i=1}^n f(a + (b-a) \frac{2i-1}{2n})$$

где  $n$  – параметр задачи. Чем больше  $n$ , тем выше точность приближенного вычисления интеграла.

- (2 балла) Вычисление числа простых чисел в интервале от 1 до  $n$ . Напишите функцию `prime_numbers`, вычисляющую число простых чисел в интервале от 1 до  $n$  при помощи перебора только нечетных чисел от 3 до  $n$ . Для проверки простоты числа можно использовать написанную ранее функцию [prime](#).

Цикл из какой функции следует распараллеливать? Если использовать два потока, будет ли самым эффективным разделить итерации цикла на два равных диапазона (от 3 до  $n/2$  и от  $n/2$  до  $n$ )? Почему?

Используя знания по математическому анализу, предложите оптимальное разбиение отрезка  $[3, n]$  на две части, чтобы нагрузка в каждом из двух потоков была одинаковой.

Для  $n=10^6$  число простых чисел будет 78497 (если считать, начиная с 3).

4. (3 балла) Напишите последовательную функцию, принимающую массив координат точек в трехмерном пространстве и вычисляющую диаметр указанного множества точек, т. е. наибольшее расстояние между двумя точками из указанного набора. Распараллельте эту функцию на а) 2 б) 4 в) 3 потока (да, правильное распараллеливание на 3 потока сложнее, чем на 2 и даже на 4).

*Для задания координат точки можно завести структуру `Point`, а набор точек хранить в переменной типа `vector <Point>`. Напишите также функцию `dist`, вычисляющую расстояние между двумя точками.*

5. (3 балла) Изобретенная Конвеем игра «Жизнь» состоит в следующем. Плоскость делится на единичные квадраты прямыми  $x = n$  и  $y = k$  ( $k$  и  $n$  пробегают независимо все целые числа). В каждом квадрате в каждый момент времени (время дискретно) может быть пусто или жить одна клетка. Состояние клетки в следующий момент времени определяется правилами:

- ✓ если у клетки число живых клеток на восьми соседних квадратах меньше двух или больше трех, клетка погибает;
- ✓ если у пустого квадрата число живых клеток на восьми соседних квадратах равно трем, в этом квадрате рождается новая клетка;
- ✓ в остальных случаях состояние клетки не меняется.

При этом состояние всех клеток меняется одновременно. Написать функцию, принимающую двумерный массив клеток и вычисляющую его состояние в следующий момент времени (клетки за пределами массива считаются мертвыми).

Работу этой функции можно ускорить двумя способами: один из них – с помощью OpenMP или аналогичных средств распределить работу по ядрам процессора. Второй способ тоже позволяет резко ускорить работу этой функции, но за счет другой идеи (найдите его; подсказка – этот способ сокращенно называется [simd](#)).

6. Предложите алгоритм распараллеливания проверки правильности скобочной структуры (исходные данные – строка из символов «(» и «(» и «)») с использованием OpenMP
- а) (3 балла) на 2 потока
  - б) (5 баллов) на 3 потока