

### Контрольные вопросы:

- ☐ (5 б.) Что такое контекстное переключение задач?
- ☐ (5 б.) Назовите основные подходы к организации параллелизма?
- ☐ (5 б.) Что может влиять на производительность параллельных алгоритмов?
- ☐ (5 б.) Как в стандартной библиотеке реализована концепция асинхронного исполнения?
- ☐ (5 б.) Что нужно учитывать при замене последовательной реализации алгоритма на параллельную?

### Упражнения:

- ☐ (25 б.) Реализуйте последовательную и параллельную версии алгоритма Монте-Карло для оценки числа  $\pi$ .
- ☐ (25 б.) Модифицируйте алгоритм `parallel_accumulate` так, чтобы количество потоков можно было задавать извне. Исследуйте алгоритм на масштабируемость. Для этого постройте график зависимости времени работы от количества потоков  $N$ . Определите оптимальное значение  $N$  и объясните полученные результаты.
- ☐ (25 б.) Реализуйте параллельную версию алгоритма `std::for_each`, используя рекурсивное разбиение входной последовательности, асинхронные задачи `std::async` и механизм будущих результатов `std::future`.
- ☐ (25 б.) Выберите три случайных алгоритма STL стандарта C++17, допускающих параллельное исполнение, и сравните их производительность с соответствующими последовательными реализациями алгоритмов STL.