

Контрольные вопросы:

- ☐ (5 б.) Что есть «состояние гонки» и к чему оно может привести?
- ☐ (5 б.) Как используются мьютексы и условные переменные?
- ☐ (5 б.) На что влияет выбор гранулярности блокировки?
- ☐ (5 б.) Когда возникает взаимоблокировка и как ее предотвратить?
- ☐ (5 б.) Что есть атомарная операция и атомарный тип данных?

Упражнения:

- ☐ (25 б.) Модифицируйте параллельный алгоритм Монте-Карло для оценки числа π из предыдущего задания таким образом, чтобы потоки безопасно использовали общий разделяемый счетчик точек, попавших в круг.
- ☐ (25 б.) Реализуйте параллельный алгоритм поиска всех вхождений указанного пользователем фрагмента в нуклеотидной последовательности ДНК. Будем считать, что данная последовательность записывается как строка, состоящая только из символов А, G, Т, С. Сгенерируйте самостоятельно подобную строку достаточно большой длины. Искомый фрагмент вводится пользователем через стандартный поток ввода и также состоит из некоторого произвольного количества символов А, G, Т, С. В качестве результата предоставьте контейнер с индексами позиций в исходной последовательности, с которых начинается искомый фрагмент.
- ☐ (25 б.) Реализуйте потокобезопасную очередь с приоритетами с помощью мьютекса и условной переменной.
- ☐ (25 б.) Сравните производительность потокобезопасного стека с блокировками с семинара с производительностью потокобезопасного стека без блокировок из [Boost](#). Для этого создайте N потоков-производителей, добавляющих элементы в структуру данных, и N потоков-потребителей, извлекающих элементы из структуры данных. Каждый из них должен обрабатывать M элементов в цикле. Поэкспериментируйте с различными значениями N и M. Обеспечьте одновременное начало работы всех потоков с помощью атомарного флага типа bool. Измерьте суммарное время выполнения всех операций во всех потоках. Повторите для очередей.