Задание 2

Загрузите и скомпилируйте программу histogramming.cpp. Эта программа в нескольких потоках строит гистограмму искусственных данных – многократно повторяющихся чисел от 0 до 19.

Однако программа содержит ошибку – гонку данных, из-за которой она работает не правильно. Чтобы увидеть это, сначала запустите программу и укажите ей выполняться в одном потоке. Программа выведет правильный ответ – число 2621440, повторённое 20 раз. Запустите программу ещё раз, и укажите другое число потоков. Программа выдаст неправильный ответ, скорее всего 20 разных чисел, меньших, чем 2621440.

Задача 1

Исправьте проблему с гонкой данных. Для этого модифицируйте цикл в функции histogramming_thread(), так, чтобы для обновления гистограммы использовалась одна из Interlocked функций.

Задача 2

Использование Interlocked функции решает проблему с гонкой данных, однако за это придётся заплатить производительностью – скорее всего, код в потоках теперь будет часто выполняться последовательно, из-за частых попыток одновременно изменить одну и ту же ячейку гистограммы. Чтобы ускорить работу программы и получить прок от многопоточности, можно использовать следующий приём.

1) Сначала, каждый поток будет считать свою собственную локальную гистограмму по тому участку данных, который ему достался. При этом нет необходимости использовать interlocked функции, так как у каждого потока будет своя гистограмма, с которой работает только он один. На этом этапе потоки будут работать полностью параллельно и независимо друг от друга.
2) Закончив обработку своего набора данных поток должен объединить свою гистограмму с результатами других потоков. На этом этапе необходимо использовать interlocked функции, так как существует возможность одновременной модификации разными потоками одной общей гистограммы.

Оценка задания

1) Реализация задачи 1 – 30 %.
2) Реализация задачи 2 – 70 %.