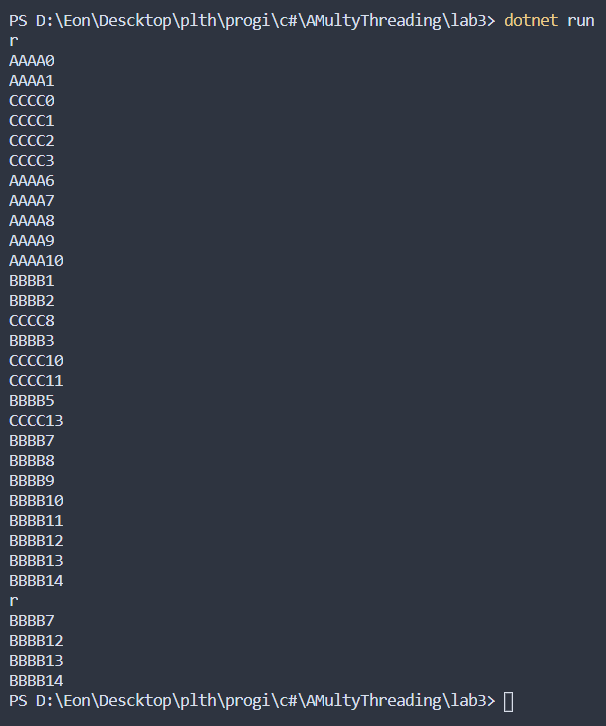
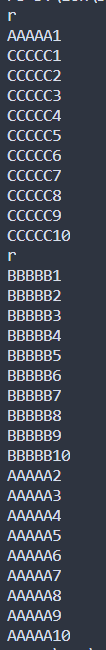
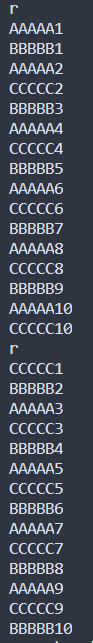
Проблема гонки данных:

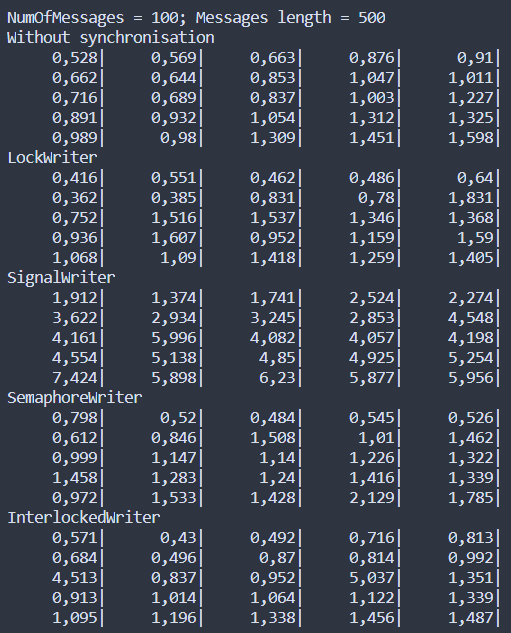
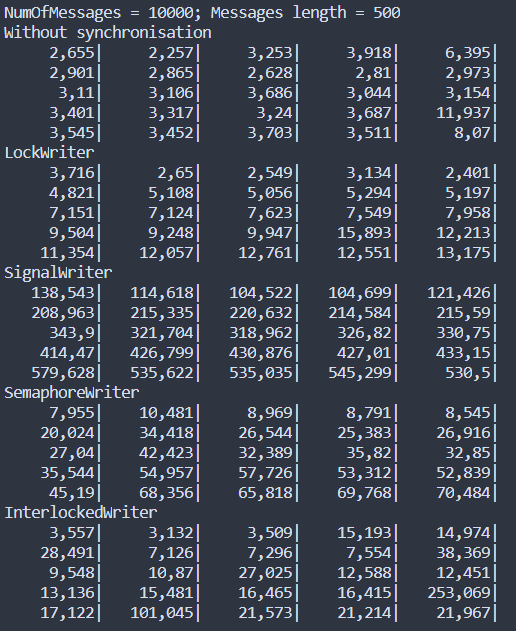
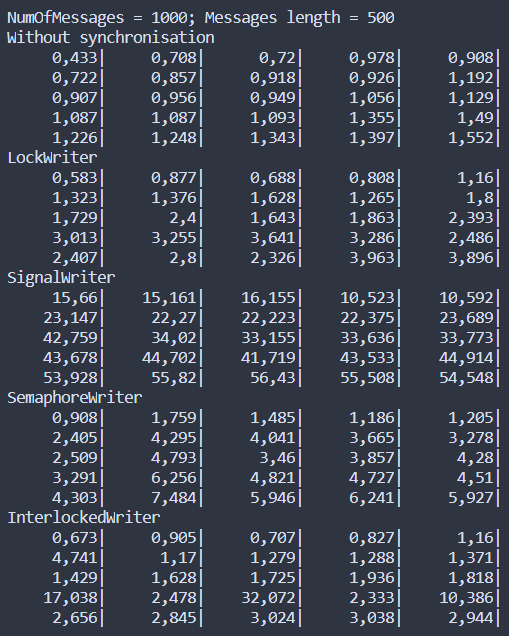


В программе работали 3 писателя и 2 читателя. У каждого писателя было 15 сообщений. Как видно из результатов, многие сообщения прочитаны не были, а некоторые дублируются (например, у писателя A были прочитаны только сообщения 0, 1, 6, 7, 8, 9, 10, а сообщения писателя B 7, 12, 13, 14 дублируются у читателей).

При использовании средств синхронизации (Lock слева, AutoResetEvent справа):

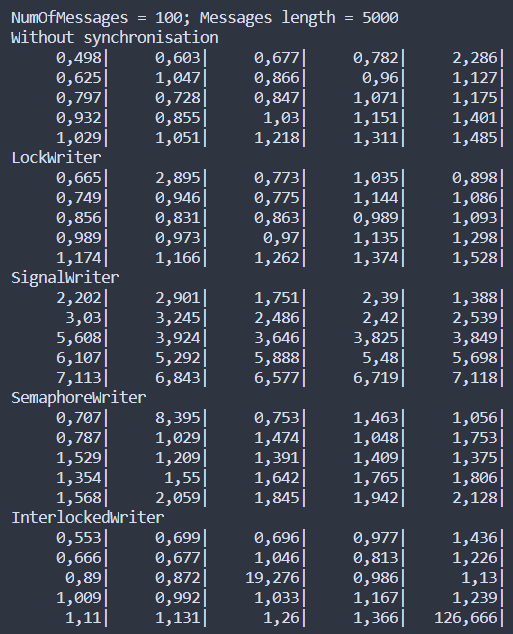
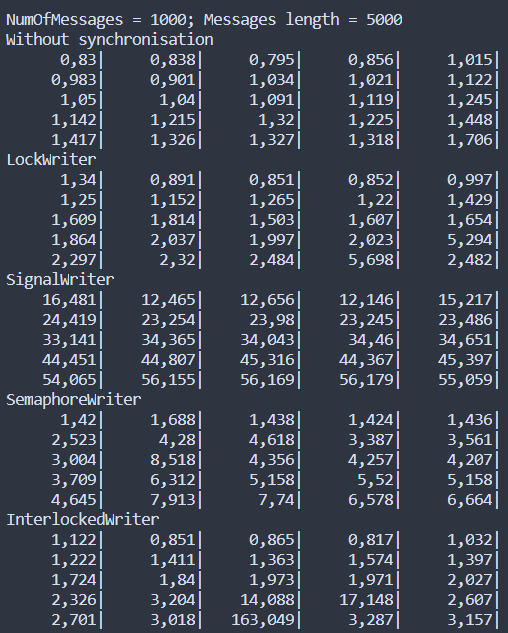
 

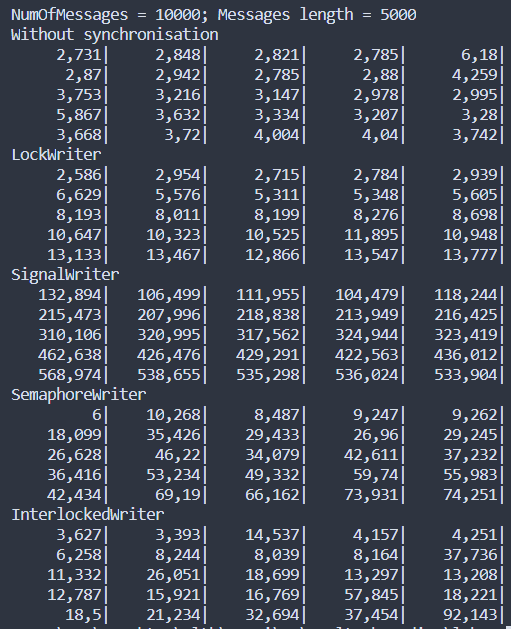
Программа выполнялась с 3 писателями и 2 читателями, у писателей по 10 сообщений. Сообщения не теряются и не дублируются. При использовании сигнальных сообщений и семафоров потоки считывают сообщения по очереди.

По результатам видно, что без средств синхронизации программа работает быстрее всех. Это происходит, потому что несколько потоков могут считывать и записывать информацию в одно и то же время, тем самым дублируя и теряя сообщения. При использовании средств синхронизации каждое сообщение будет записано и считано, на что уходит больше времени. Также видно, что сигнальные средства синхронизации выполняются дольше всех, так как задействуют ядро ОС. В работе использовался SemaphoreSlim, поэтому время его работы меньше, чем у сигнальных средств синхронизации, но на больших объёмах данных SemaphoreSlim проигрывает полностью «высокоуровневым» средствам синхронизации. На больших объёмах данных самым быстрым оказался Lock.

Для сигнальных сообщений и семафоров в главном потоке генерируется 1 сигнал вместо цикла, так как после того, как главный поток сгенерирует сигнал, он продолжит выполнение и когда закончит, то освободит ресурсы, которые могут понадобиться другим потокам.

При увеличении длины обрабатываемых сообщений результаты изменились не сильно, то есть на время работы сильнее влияет количество сообщений у писателей.