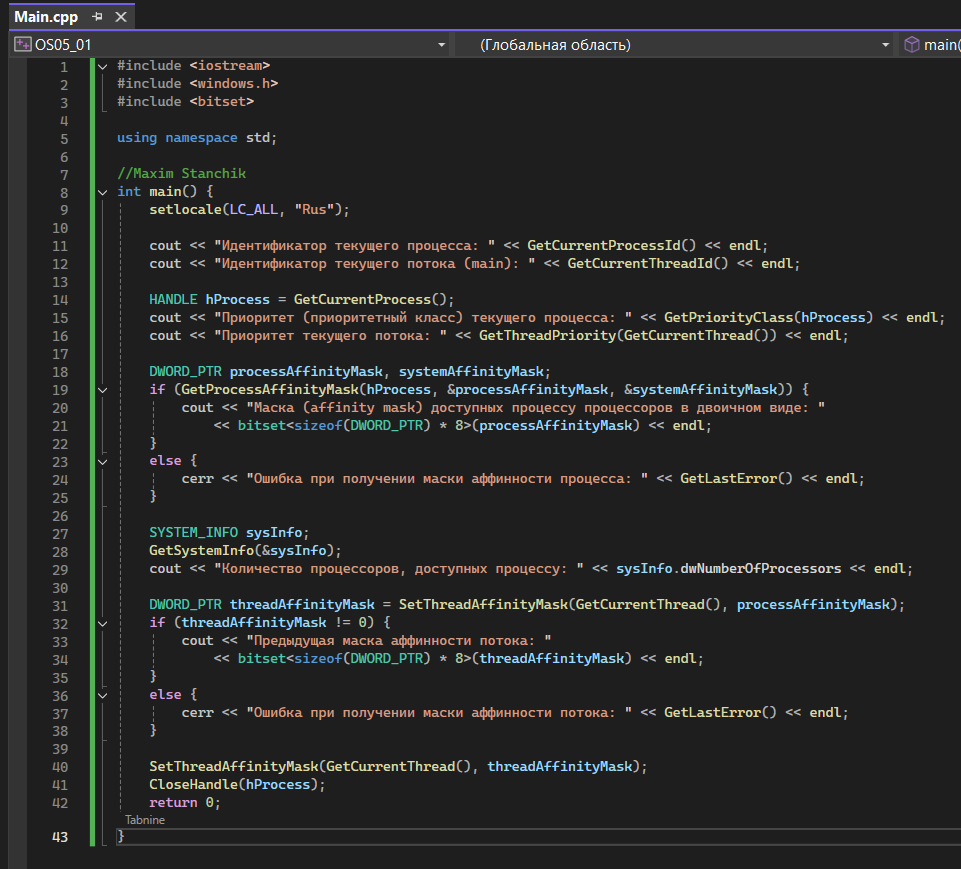
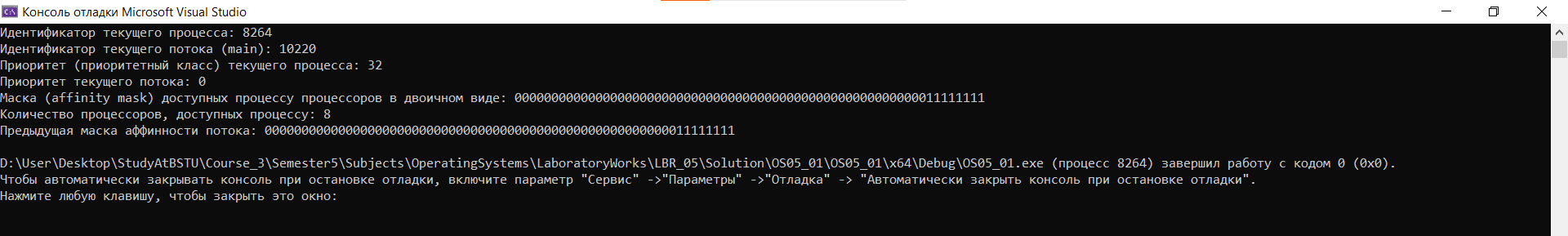
**Задание 01**

Код:

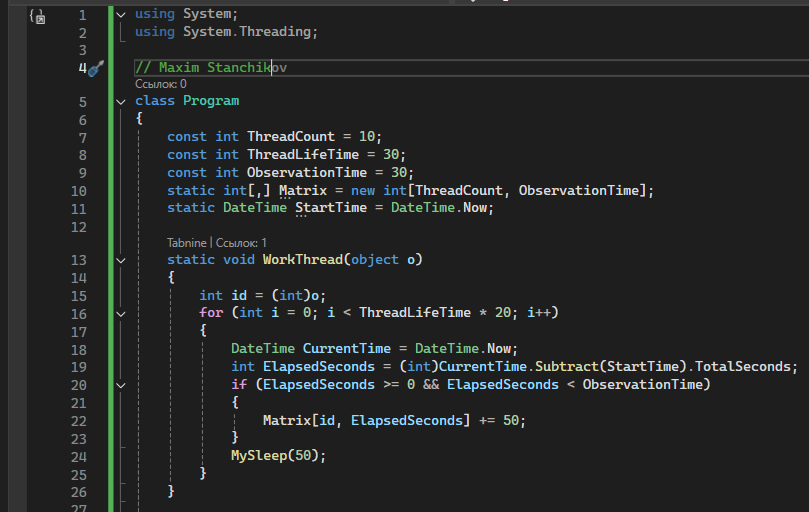
****

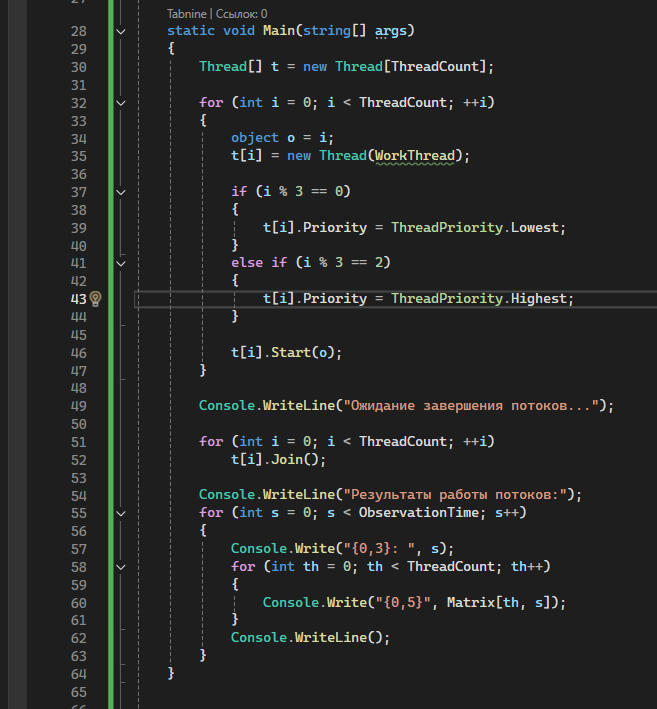
Результат:

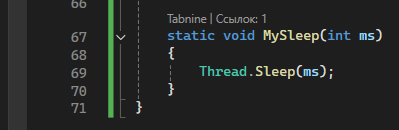


**Задание 02**

Код:

****

****

****

Результат:

****

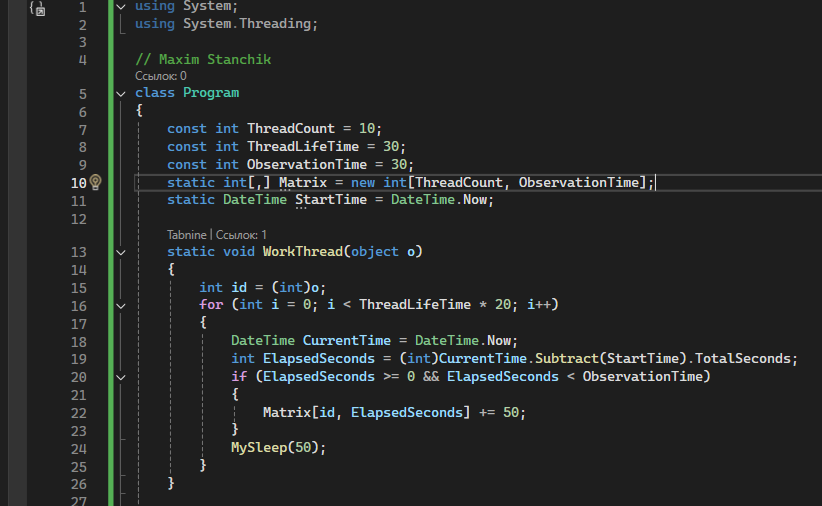
Объяснение результатов:

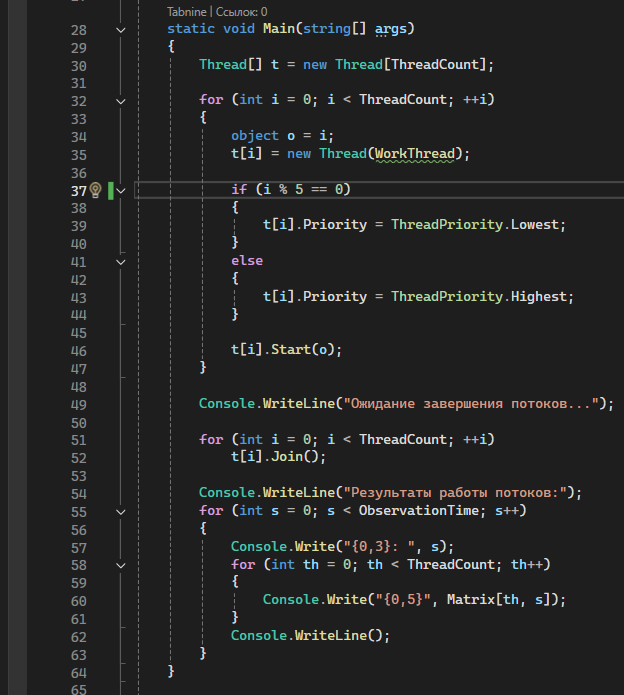
Потоки с индексом 0,3,6,9… : имеют более низший приоритет, что означает, что ОС будет выделять им меньше процессорного времени для их выполнения, чем потокам с высшим приоритетам. Однако данным потокам все равно смогли выполнить задачу, что показывает, что система имеет достаточно ресурсов для их выполнения.

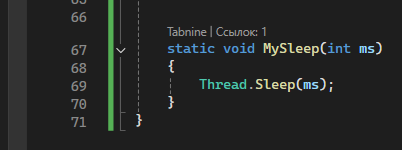
Высокоприоритетные потоки работают стабильно, а у низкоприоритетных потоков наблюдаются падения, так как они реже получают процессорное время.

**Задание 03**

Код:

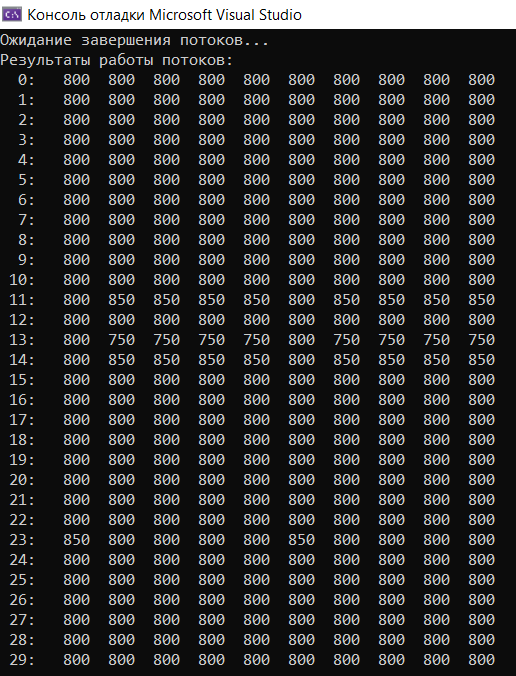
****

****

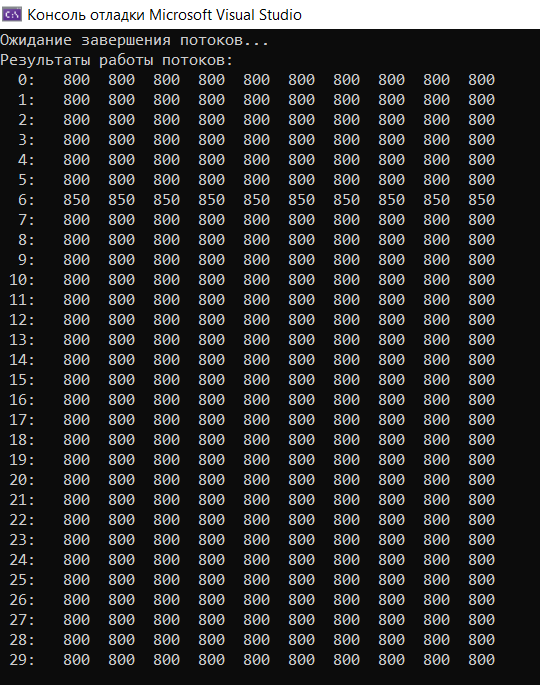
****

Результат:

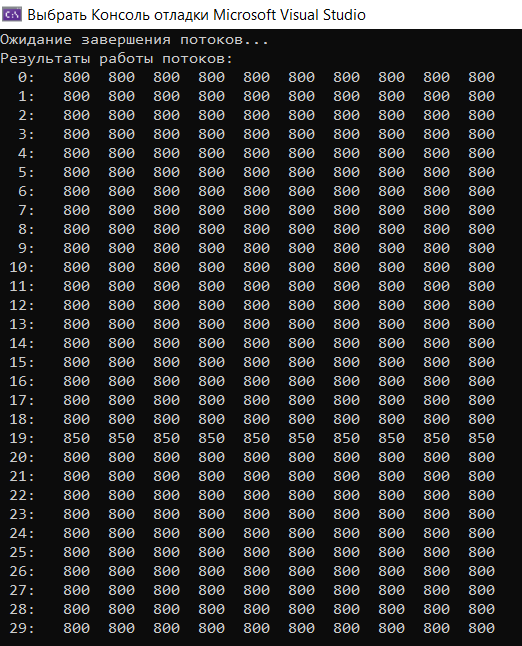
(Запустили в первый раз)



(Запустили во второй раз)



(Запустили в третий раз)

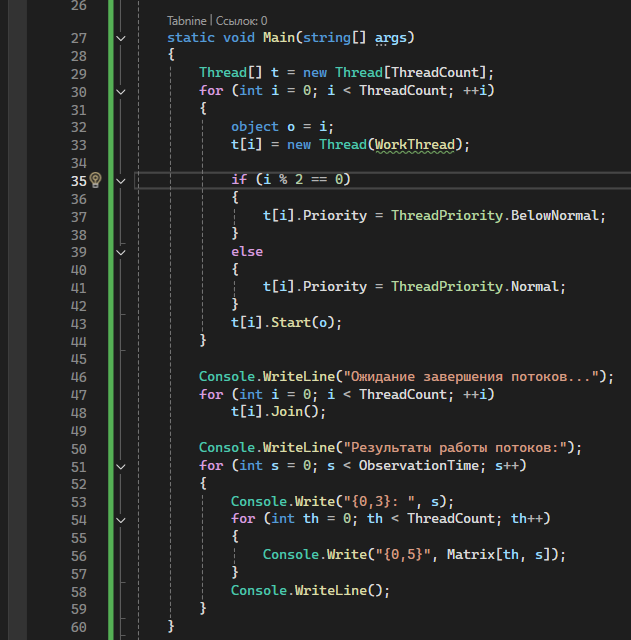


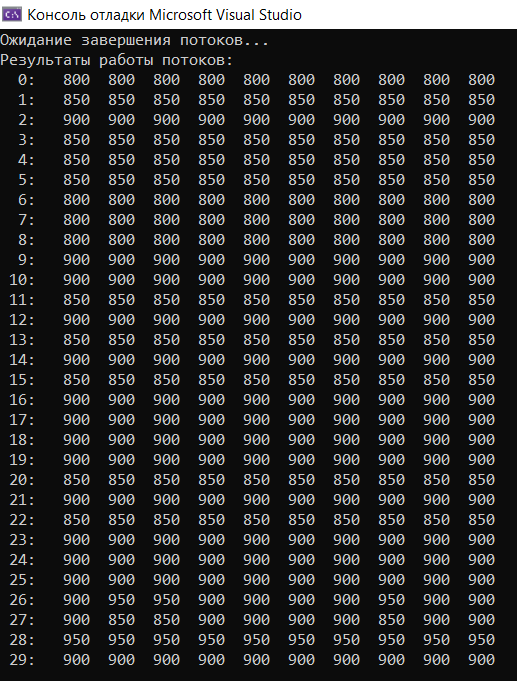
Все потоки выполнились примерно одинаково. Все потоки получили достаточное кол-во времени, чтобы завершиться. Скорее всего, нагрузка на систему невелика, поэтому все потоки завершаются в любом случае.

Теперь выполним это же приложение, но с другими приоритетами у потоков. У каждого второго потока теперь приоритет BelowNormal, у всех остальных Normal.

Потоки с приоритетом BelowNormal (-1) будут выполнять свою работу медленнее, так как их приоритет ниже. Они могут получить меньше процессорного времени, если система загружена другими задачами.

Потоки с приоритетом Normal (0) будут получать больше времени на выполнение по сравнению с потоками с приоритетом BelowNormal.



****

Объяснение результатов:

Потоки с меньшим приоритетом работают чуть медленнее, что проявляется в некоторых колебаниях результатов. Например, на секундных метках 26 и 27 видно, что потоки с приоритетом BelowNormal внесли меньше данных (например, 850 против 900 для потоков с Normal). Однако разница незначительна, что может быть связано с тем, что система не была сильно нагружена другими задачами, и потоки с более низким приоритетом всё же успевали получать достаточно времени для выполнения.

**Задание 04**

Разработайте консольное Linux-приложение OS05\_04 на языке С++, выводящее на консоль следующую информации:

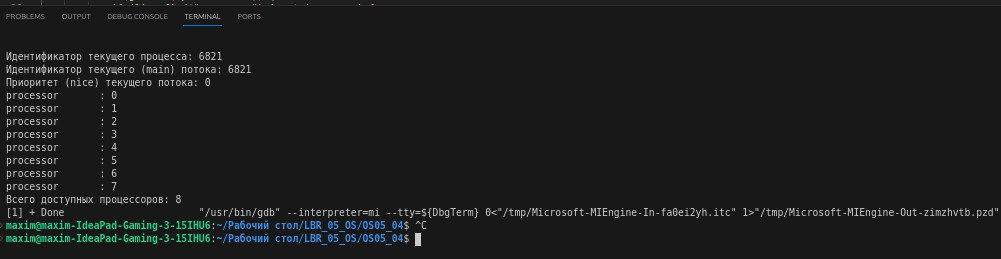
- идентификатор текущего процесса;

- идентификатор текущего (main) потока;

- приоритет (nice) текущего потока;

- номера доступных процессоров.





**Задание 05**

