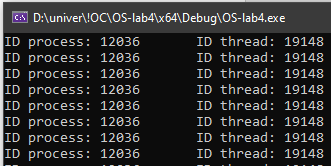
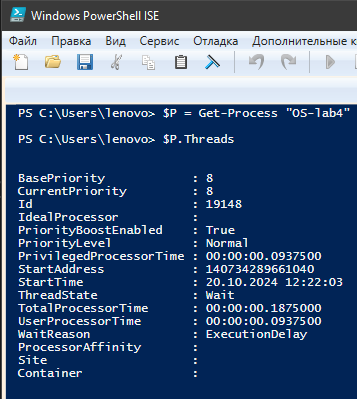
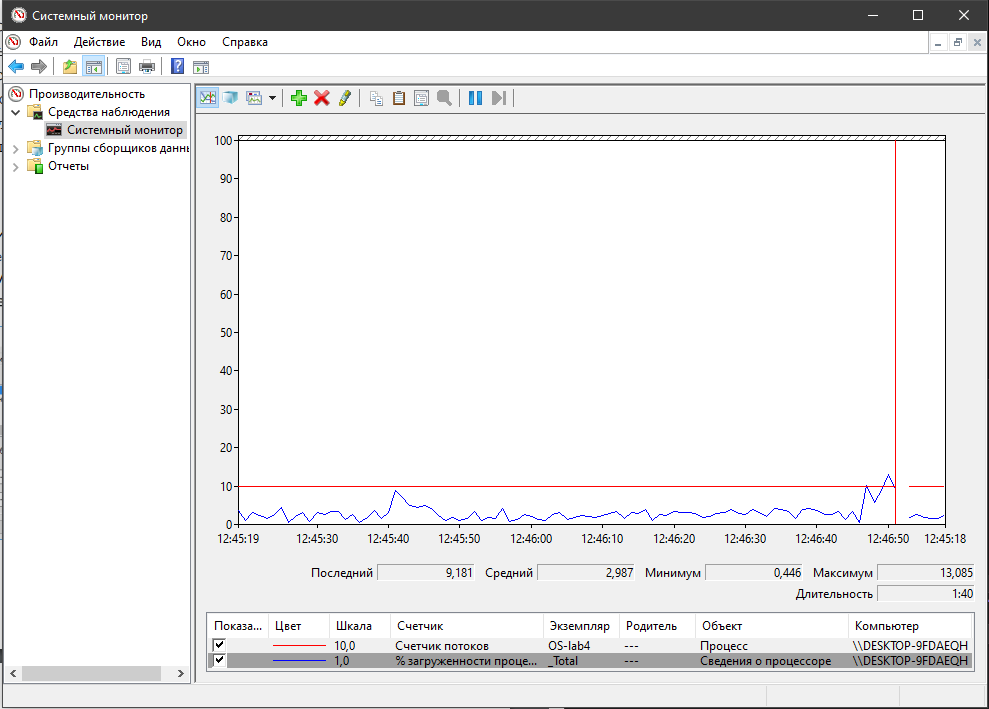
**Задание 01**

1. Разработайте консольное Windows-приложение OS04\_01 на языке С++, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификаторов текущего процесса и текущего потока.



2. Продемонстрируйте информацию об потоках процесса OS04\_01 с помощью утилит PowerShell ISE и Performance Monitor.





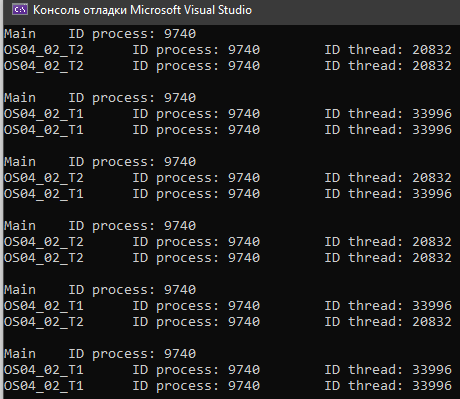
**Задание 02**

3. Разработайте консольное Windows-приложение OS04\_02 на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

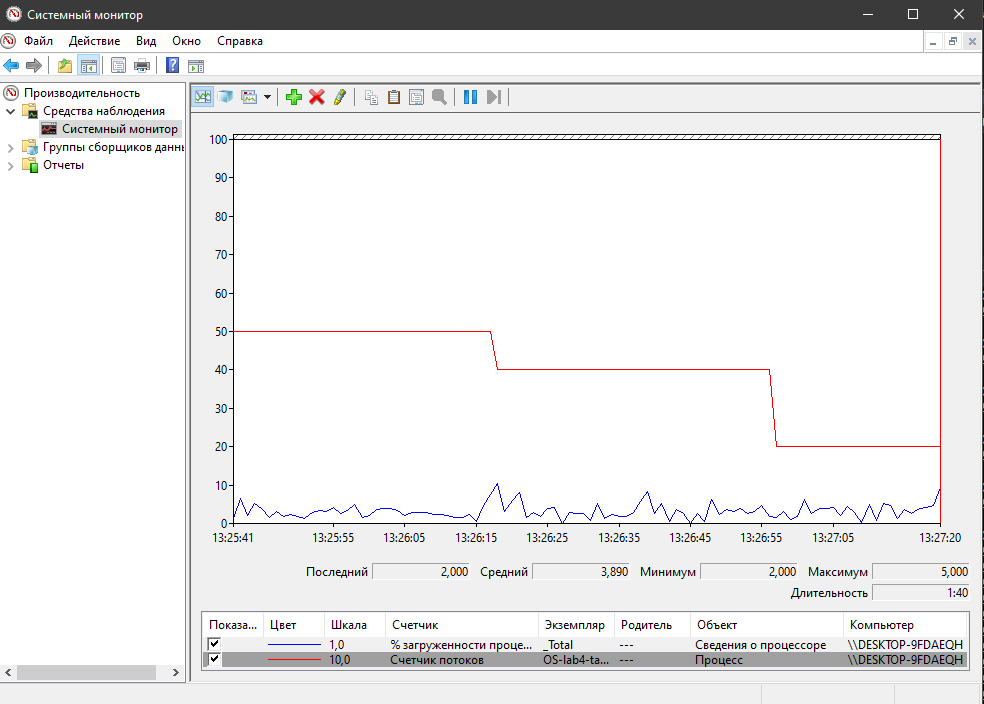
4. Процесс OS04\_02 должен создать два потока: потоковые функции OS04\_02\_T1, OS04\_02\_T2.

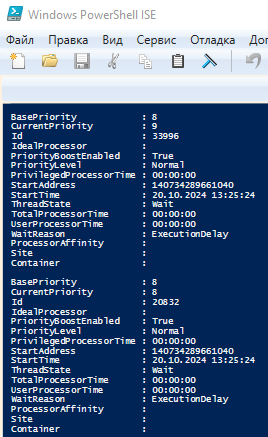
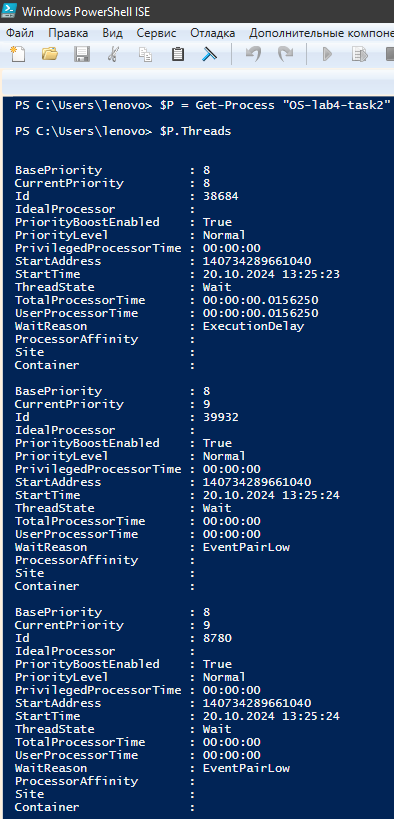
5. Поток OS04\_02\_T1 - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

6. Поток OS04\_02\_T2 - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.



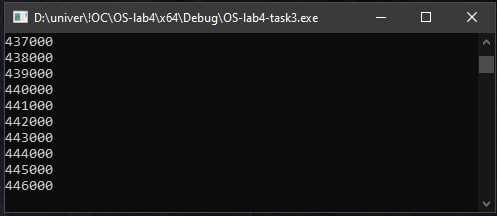
7. Продемонстрируйте информацию об потоках процесса OS04\_02 с помощью утилит PowerShell ISE и Performance Monitor.





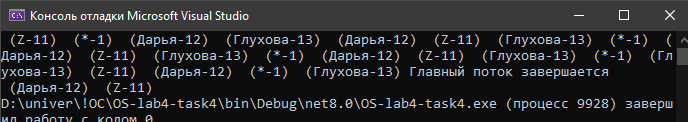
**Задание 03.**

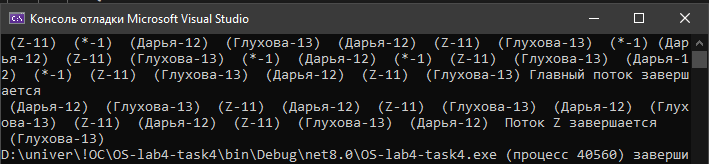
8. Определите, какое максимальное количество потоков можно создать в одном процессе на вашем компьютере, используя нижеприведенный программный код (или аналогичный). Сравните результаты между собой. От чего зависит 3 максимальное количество потоков? Какое ограничение накладывает операционная система.

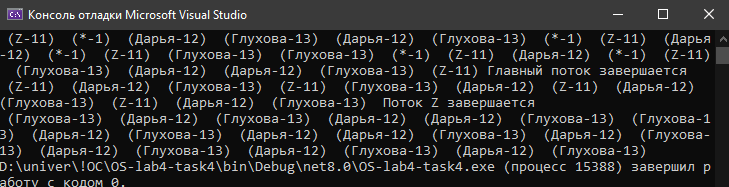


**Задание 04**

9. Создайте консольное приложение на C#, которое запускает три дополнительных потока и завершается через пять секунд. Один дополнительный поток завершается через 10 секунд, остальные два – через двадцать секунд. Фрагмент программного кода приведен ниже. (Вставьте Свои ИмяФамилию).





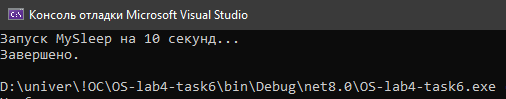


Фоновые потоки завершаются с основным потоком.

Главный поток ждет завершения НЕ фоновых, что увеличивает общее время работы приложения.

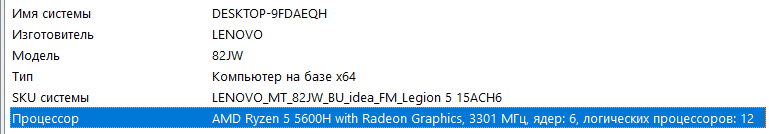
**Задание 05 (вспомогательное)**

15. Создайте функцию, которая производит ЛЮБЫЕ вычисления длительностью n миллисекунд на вашем компьютере (для последующих заданий метод Thread.Sleep(n) не подходит, так как он освобождает центральный процессор и ничего не делает). Убедитесь, что MySleep(10000) работает ровно 10 секунд.



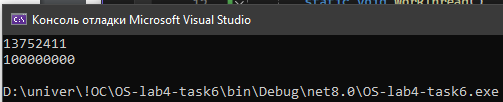
16. Узнайте количество ядер и логических процессоров в вашем компьютере (приложение 1).





**Задание 06**

17. Разработайте консольное приложение OS04\_06 на языке С#, запускающее 20 потоков, каждый из которых в цикле 5000000 раз увеличивает на единицу значение общей для всех потоков переменной. Исходное значение переменной — ноль. Выведите результат и сравните с произведением 20х5000000.



18. Сравните результаты в группе. Есть ли какая закономерность?

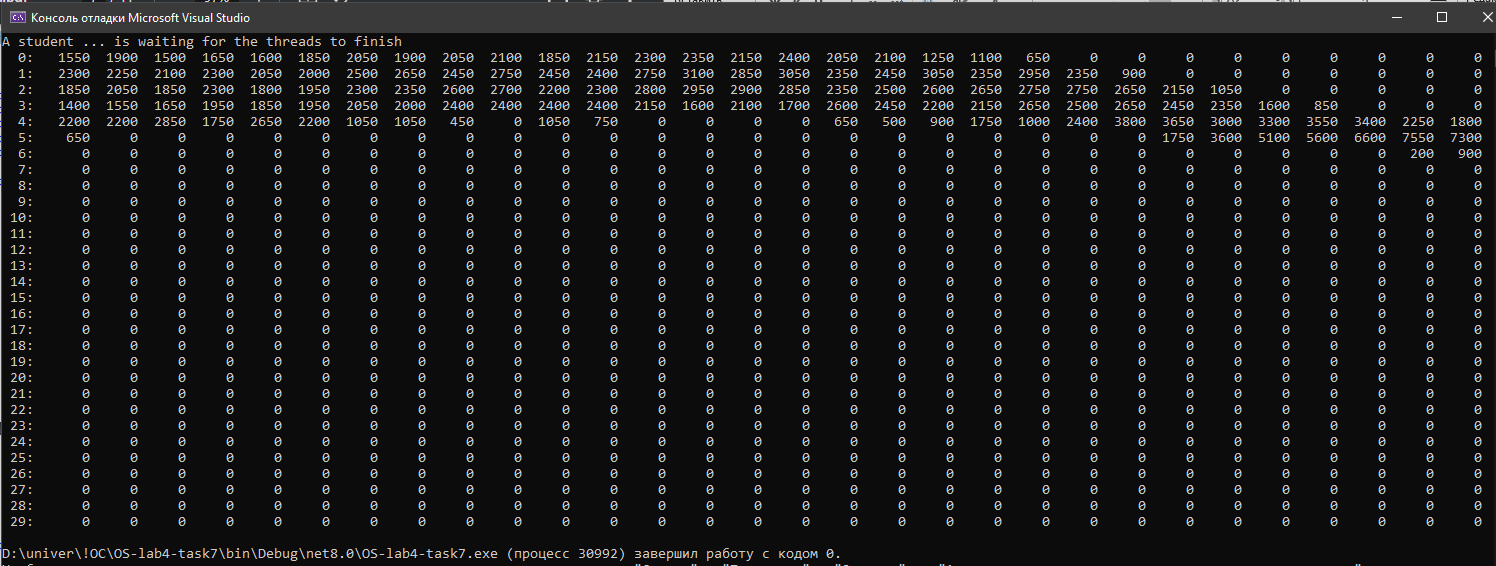
С синхронизацией - значение будет совпадать.

Без синхронизации переменная Count будет одновременно изменяться сразу несколькими потоками, увеличения могут быть потеряны.

Чем больше потоков, тем больше вероятность гонок данных.

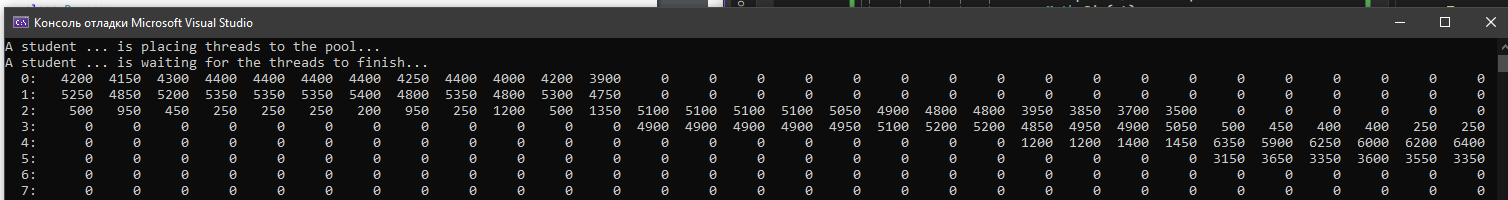
**Задание 07**

19. Разработайте консольное приложение OS04\_07 на языке С#, запускающее N потоков, каждый из которых будет производить вычисления t секунд (использовать разработанный в задании 5 метод), используя класс System.Threading.Thread. Сохраните информацию о работе потоков в течение T секунд и выведите на экран в виде таблицы < >. Подберите подходящие параметры в зависимости от количества логических процессоров в вашем компьютере (например, для четырех логических процессоров N = 10, t = 10, T=30).



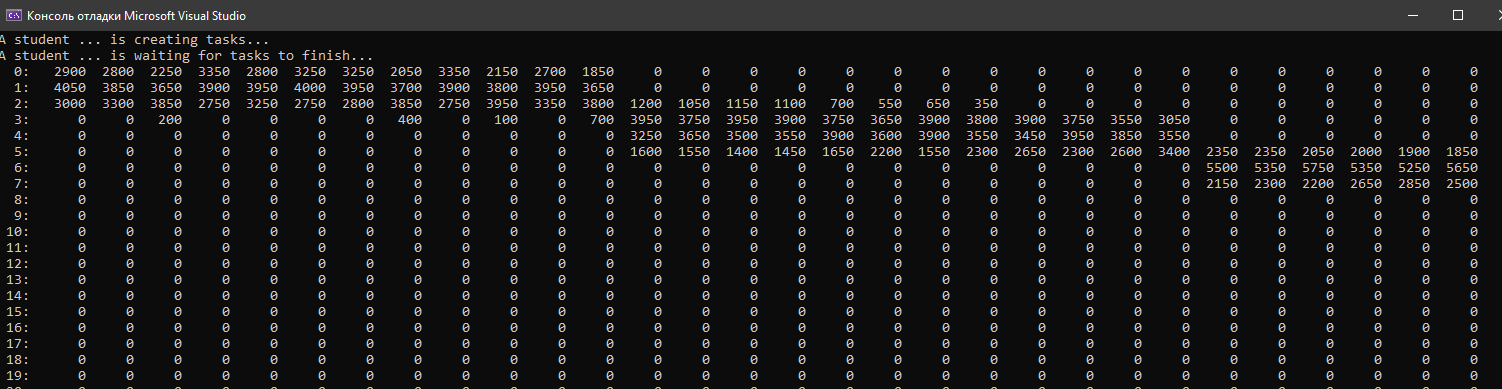
**Задание 08**

20. Скопируйте консольное приложение OS04\_07 как OS04\_08. Теперь используйте пул потоков. Выведите статистику работы потоков на экран в виде таблицы.



**Задание 09**

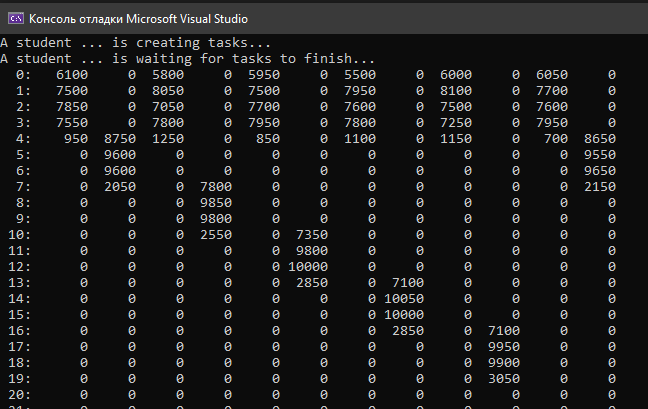
21. Скопируйте консольное приложение OS04\_07 как OS04\_09. На этот раз используйте System.Threading.Tasks.Task. Выведите статистику работы потоков на экран в виде таблицы < >. Сравните результаты заданий 7-9 и запишите вывод в отчет.



22. [For nerds only] Реализуйте запуск задач в цикле.

**Задание 10**

23. Скопируйте консольное приложение OS04\_09 как OS04\_10. Уменьшите количество задач до количества логических процессоров. Организуйте выполнение задач по очереди. Выведите статистику работы потоков на экран в виде таблицы < >.



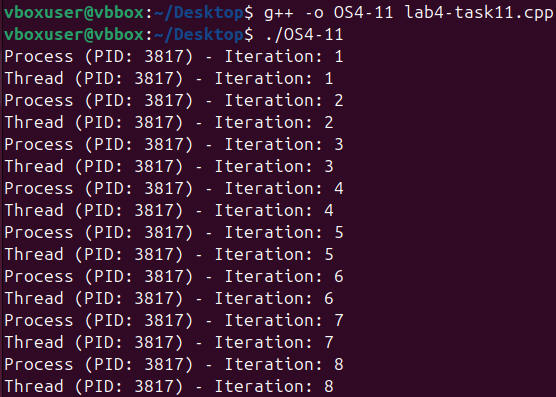
**Задание 11**

24. Разработайте на языке консольное Linux-приложение OS04\_10 на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса (использовать функции из pthread.h).

25. Процесс OS04\_10 должен создать поток: потоковая функция OS04\_10\_T1.

26. Поток OS04\_10\_T1 - выполняет цикл 75 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса.

27. Продемонстрируйте информацию о потоках процесса OS04\_10 с помощью утилиты ps.



**Задание 12. Ответьте на следующие вопросы**

28. Что такое поток управления OS?

Объект ядра операционной системы, которому OS выделяет процессорное время.

Наименьшая единица работы ядра OS

Последовательность инструкций, выполняемых процессором в выделенные OS интервалы времени.

29. С помощью каких системных вызовов создаются потоки в Windows и Linux?

CreateThread и pthread\_create:

30. Что такое системные и пользовательские потоки?

Системные потоки — это потоки, которые управляются операционной системой непосредственно через её ядро. Они имеют доступ ко всем системным ресурсам и могут быть распределены между процессорами.

Пользовательские потоки — это потоки, которые управляются на уровне приложения, а не операционной системой.

31. Что такое многопоточность?

Многопоточность – модель (парадигма) программирования (OS, системы программирования, программы) поддерживать потоки управления.

32. Что такое контекст потока и для чего он нужен?

контекст потока:

* программный код;
* набор регистров;
* стек памяти;
* стек ядра операционной системы;
* маркер доступа.

Контекст потока содержит все сведения, позволяющие потоку безболезненно возобновить выполнение, в том числе набор регистров процессора и стек потока.

33. Перечислите состояния, в которых может быть поток и поясните их назначение.

* состояние **выполнения**, когда код потока выполняется процессором; на однопроцессорных платформах в этом состоянии в каждый момент времени может находиться только один поток;
* состояние **готовности** к выполнению, когда поток готов продолжать свою работу и ждет освобождения ЦП;
* состояние **ожидания** наступления некоторого события; в этом случае поток не претендует на время ЦП, пока не наступит определенное событие (завершение операции ввода/вывода, освобождение необходимого потоку занятого ресурса, сигнала от другого потока), часто такие потоки называют блокированными. Изменение состояния потока происходит в результате соответствующих действий. Удобно для этих целей использовать следующую диаграмму состояний и переходов.

34. Что такое LWP?

легковесный процесс работает в пространстве пользователя поверх одного потока выполнения ядра, разделяет виртуальное адресное пространство и системные ресурсы потока выполнения с другими легковесными процессами в рамках того же процесса

35. Что такое потокобезопасность программного кода?

Потокобезопасность – свойство объекта или кода, которое гарантирует, что при исполнении или использовании несколькими потоками, код будет вести себя, как предполагается.

36. Что такое реентерабельность кода?

реентерабельность – функция или метод вызывается несколько раз или в потоках без конфликтов и ошибок.

37. Что такое Fiber?

Волокно (fiber) — облегченный поток, выполняемый в режиме пользователя. Волокно затребует значительно меньше ресурсов, и позволяет в некоторых случаях минимизировать количество системных вызовов и следственно увеличить производительность.