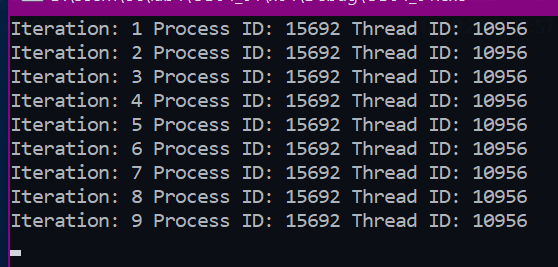
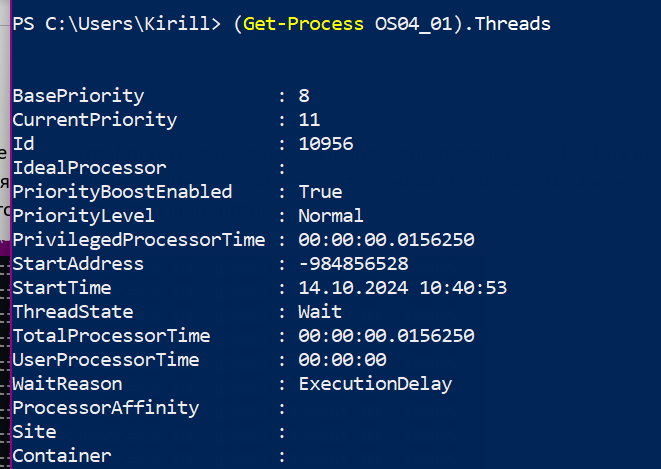
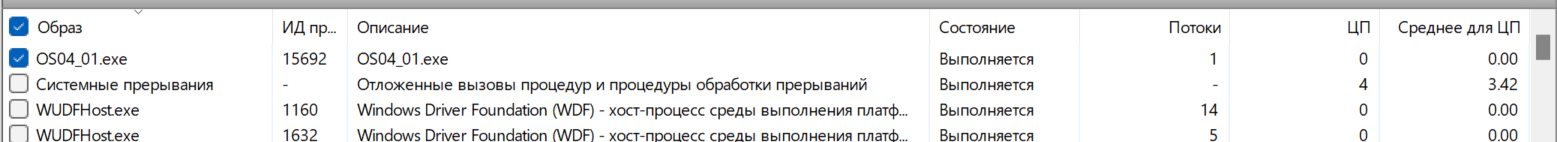
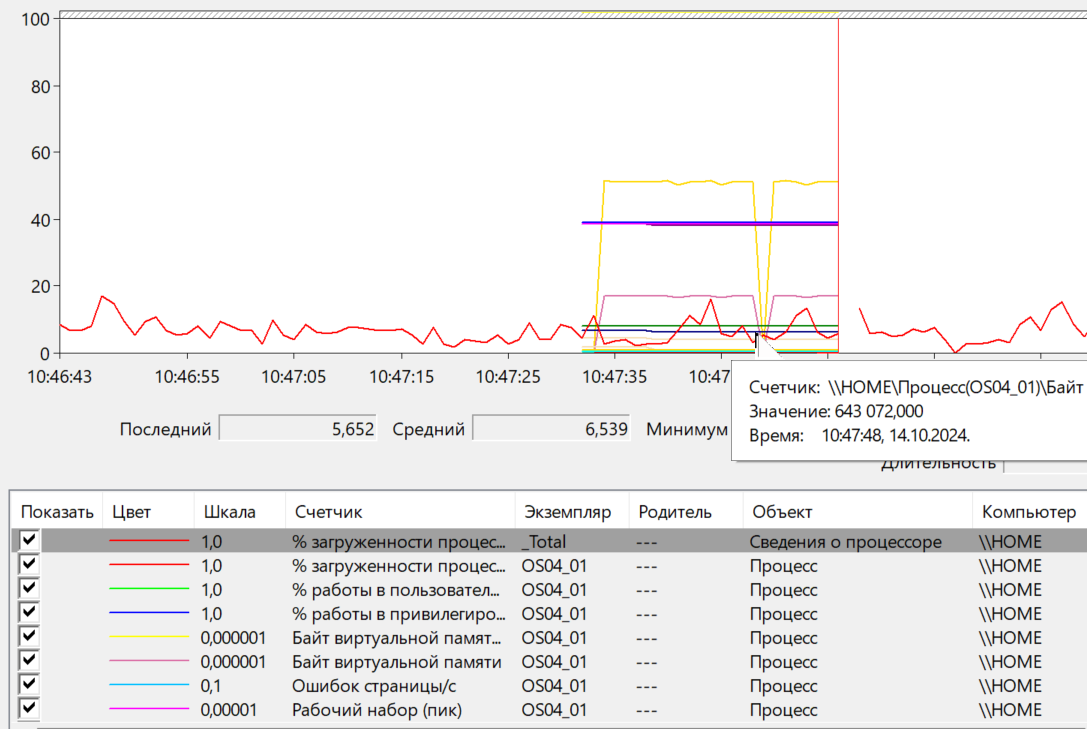
Задание 01 1. Разработайте консольное Windows-приложение OS04\_01 на языке С++, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификаторов текущего процесса и текущего потока.



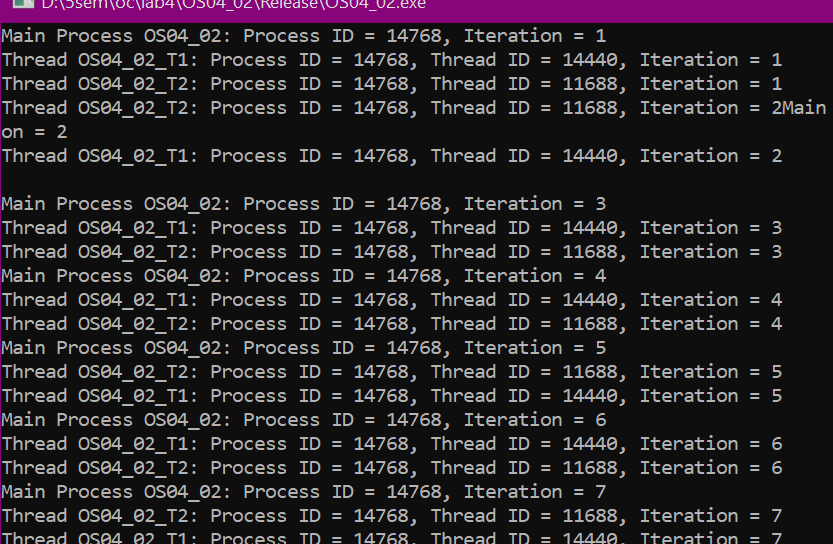
Продемонстрируйте информацию об потоках процесса OS04\_01 с помощью утилит PowerShell ISE и Performance Monitor.

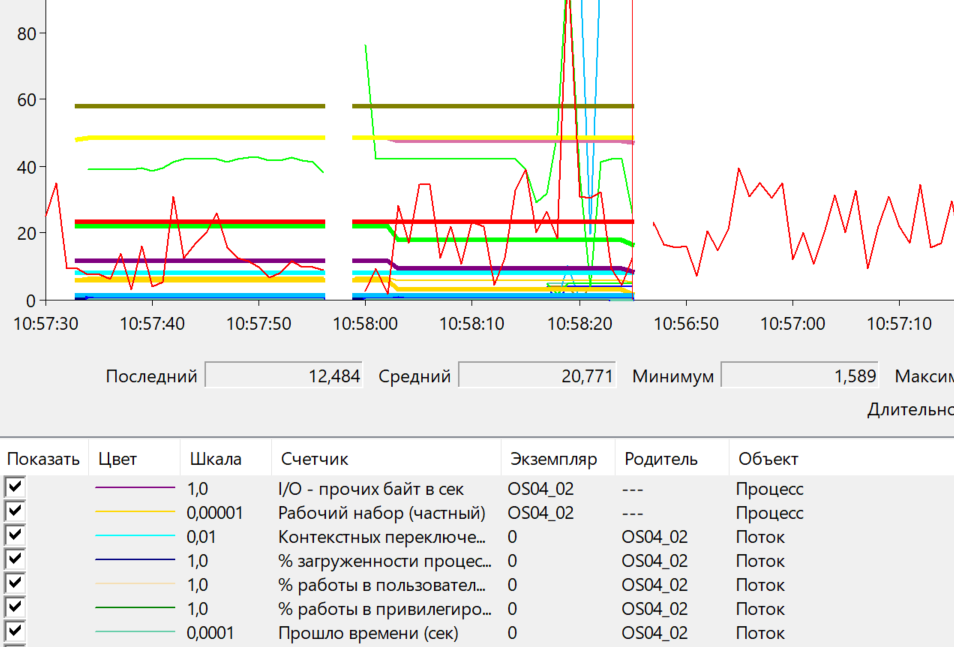
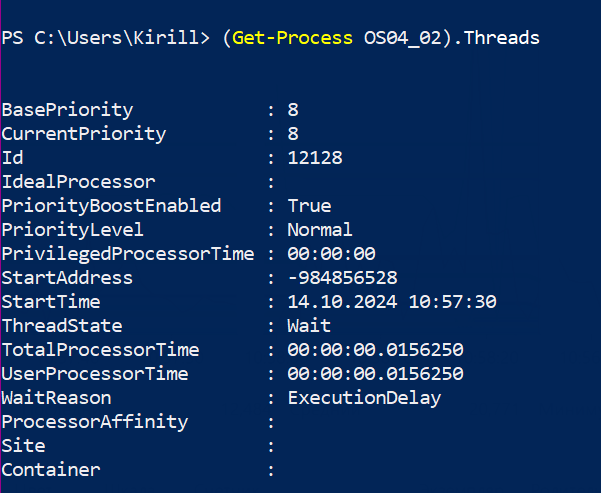






Задание 02 3. Разработайте консольное Windows-приложение OS04\_02 на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса. 4. Процесс OS04\_02 должен создать два потока: потоковые функции OS04\_02\_T1, OS04\_02\_T2. 5. Поток OS04\_02\_T1 - 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

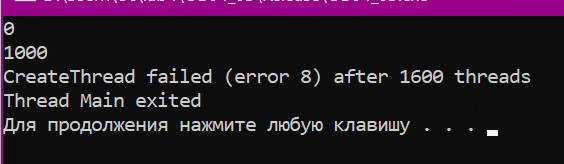




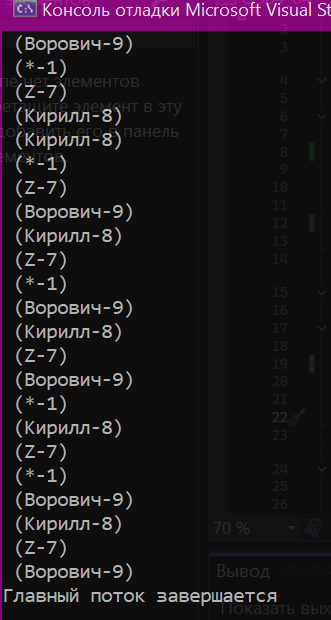
Задание 3

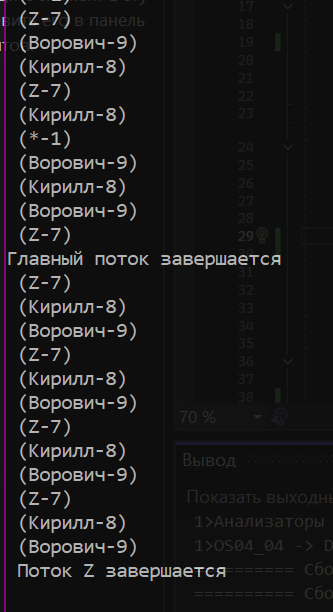
Определите, какое максимальное количество потоков можно создать в одном процессе на вашем компьютере, используя нижеприведенный программный код (или аналогичный). Сравните результаты между собой. От чего зависит 3 максимальное количество потоков? Какое ограничение накладывает операционная система.

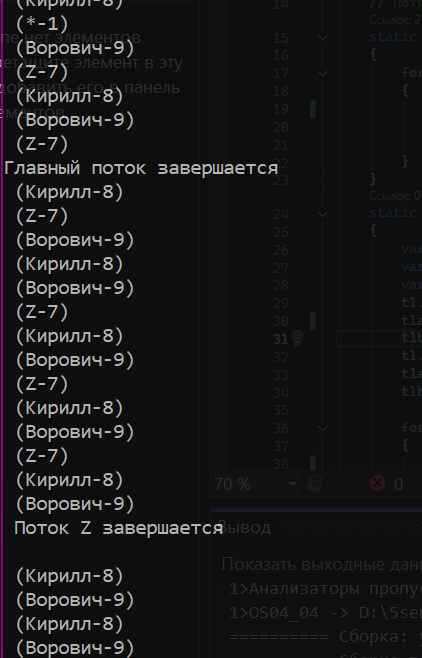


Задание 4

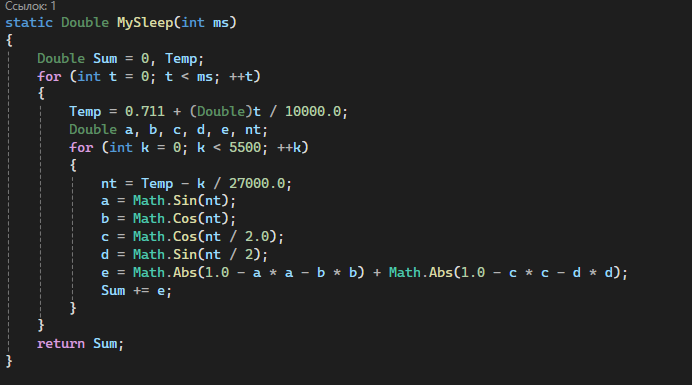
Создайте консольное приложение на C#, которое запускает три дополнительных потока и завершается через пять секунд. Один дополнительный поток завершается через 10 секунд, остальные два – через двадцать секунд. Фрагмент программного кода приведен ниже. (Вставьте Свои ИмяФамилию).



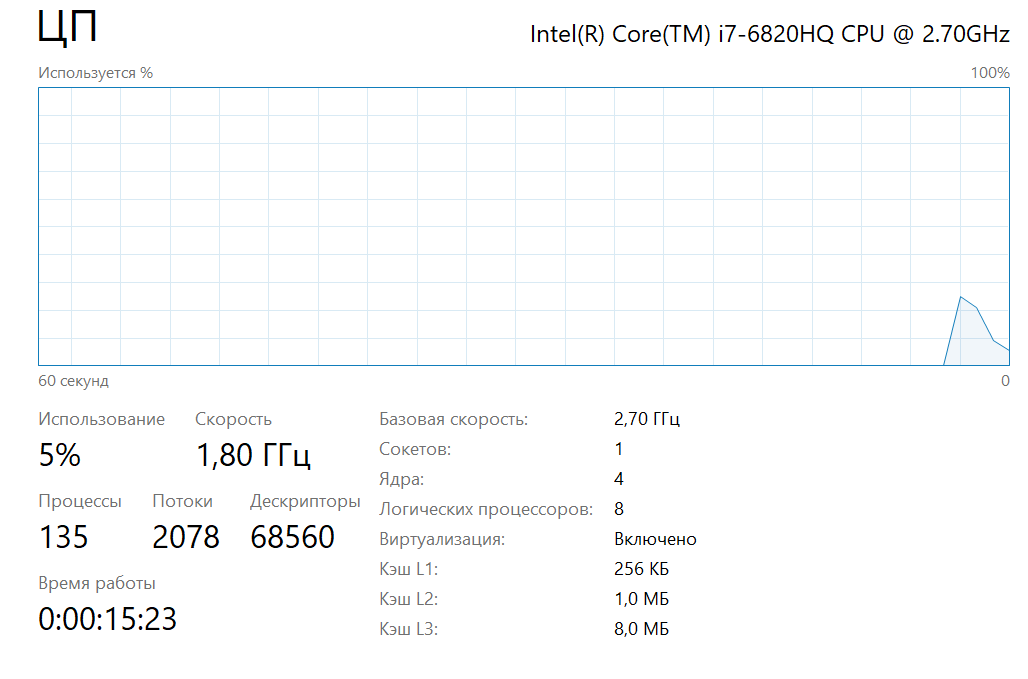




Создайте функцию, которая производит ЛЮБЫЕ вычисления длительностью n миллисекунд на вашем компьютере (для последующих заданий метод Thread.Sleep(n) не подходит, так как он освобождает центральный процессор и ничего не делает). Убедитесь, что MySleep(10000) работает ровно 10 секунд. Один из вариантов (не забывайте про оптимизатор: он может исключать вычисления, результат которых не используется; в данном примере накапливается сумма и возвращается из метода).

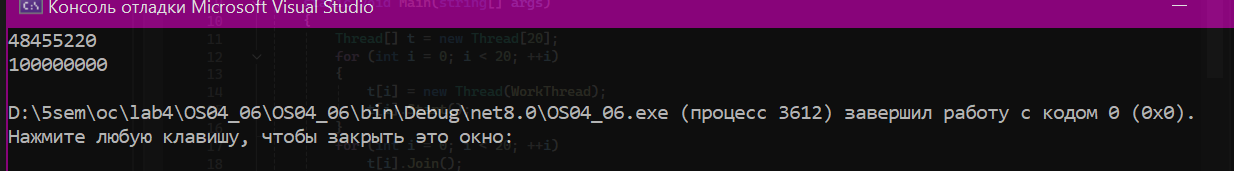


Узнайте количество ядер и логических процессоров в вашем компьютере (приложение 1).



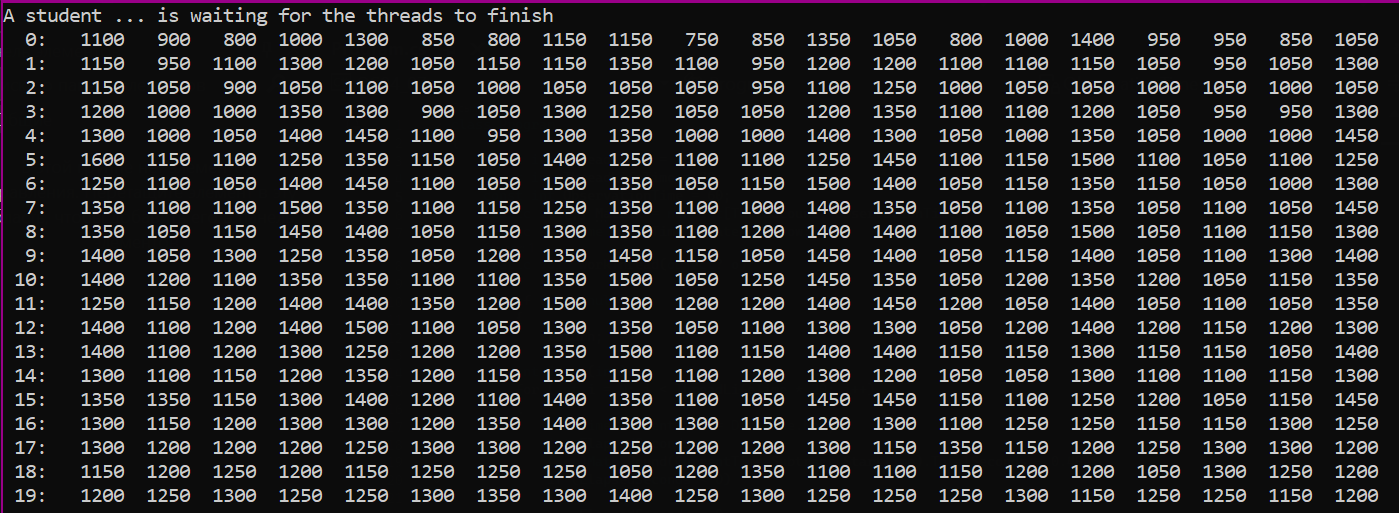
Задание 6

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



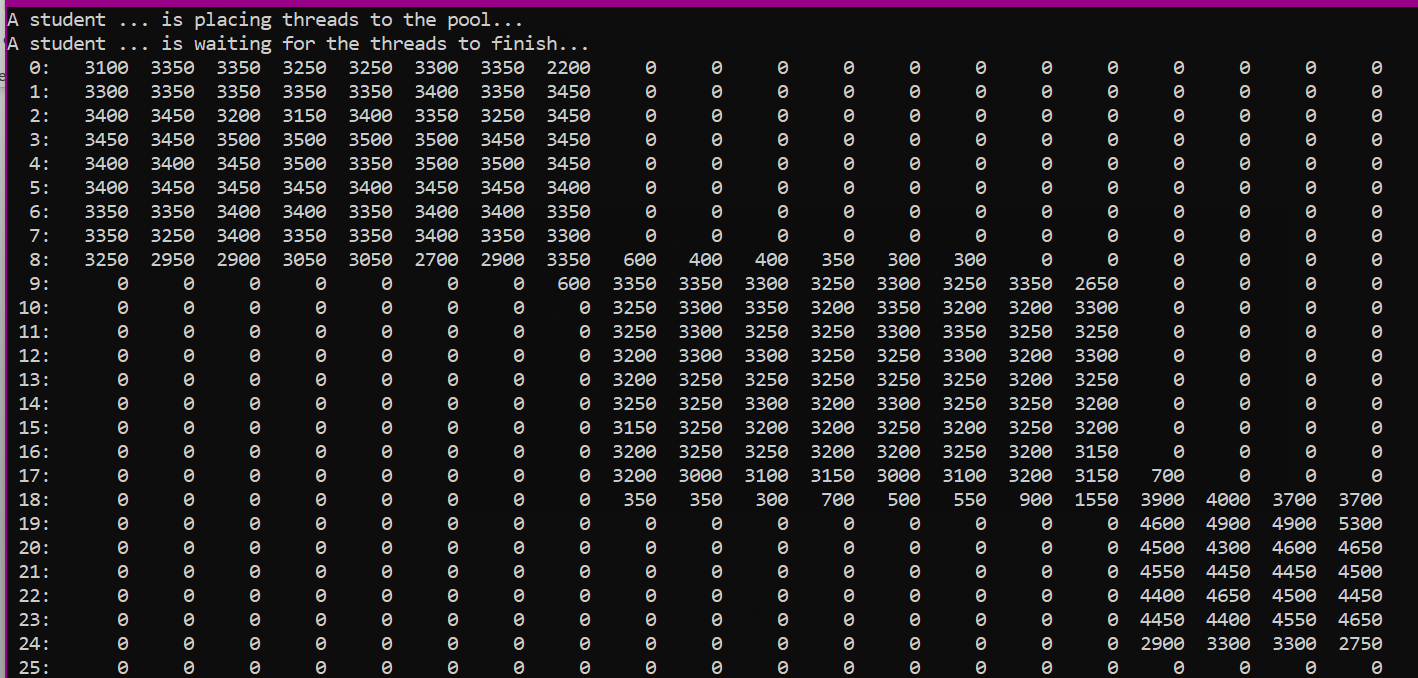
Задание 7

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).

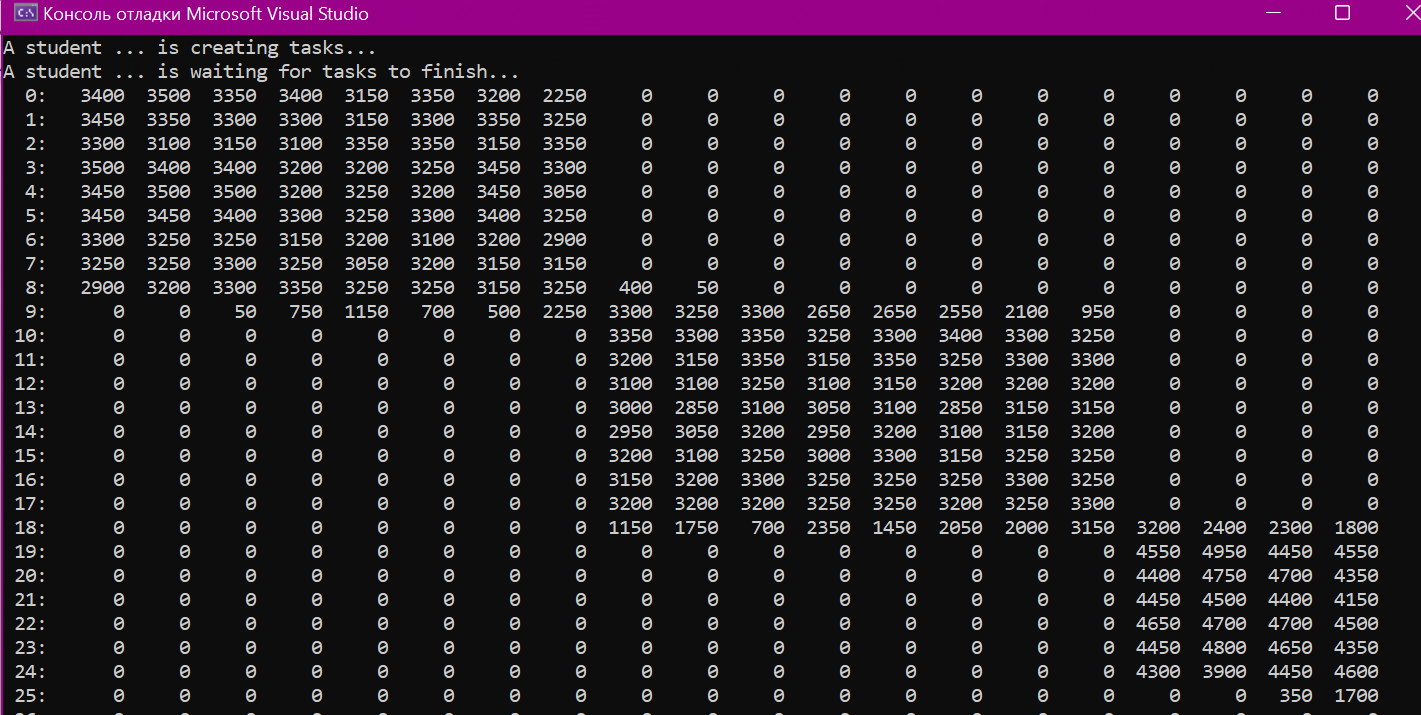


Задание 8

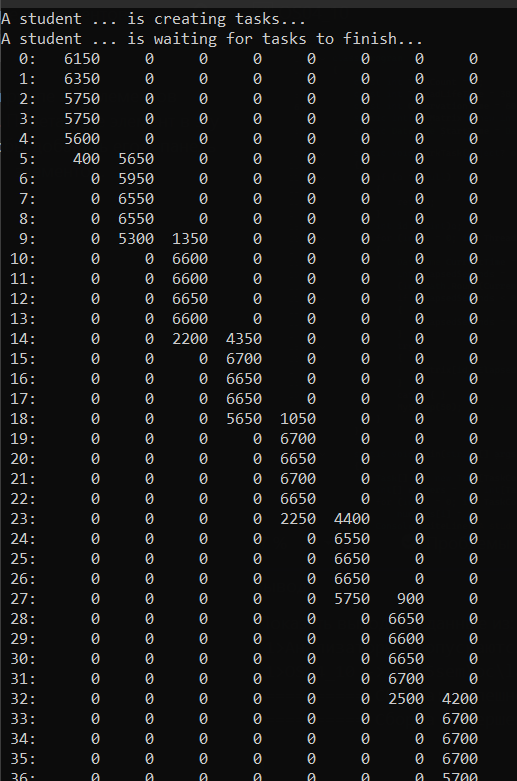
Скопируйте консольное приложение OS04\_07 как OS04\_08. Теперь используйте пул потоков. Выведите статистику работы потоков на экран в виде таблицы < >.



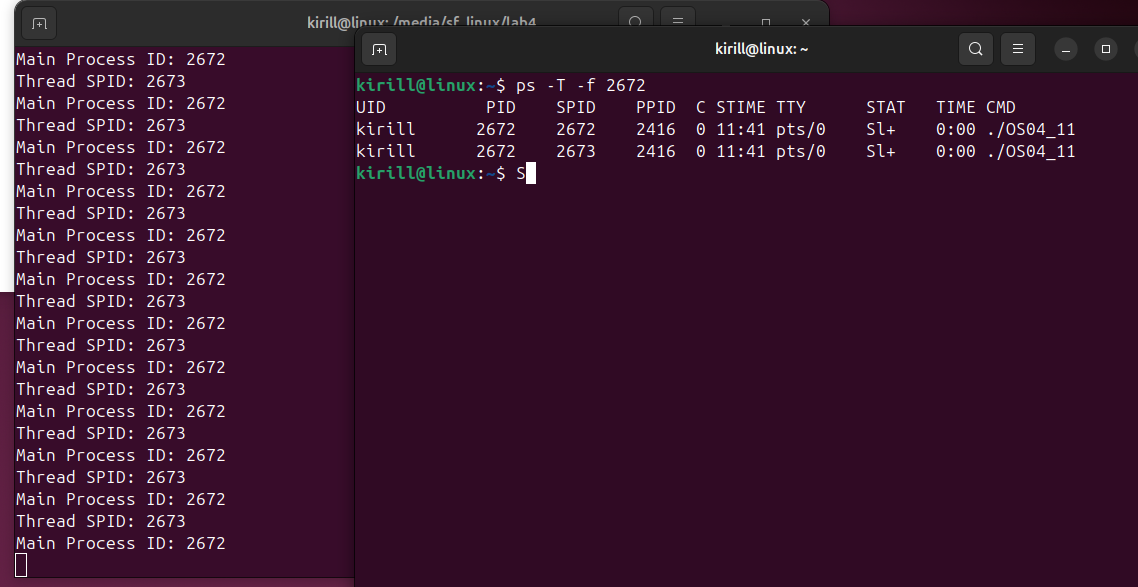
Скопируйте консольное приложение OS04\_07 как OS04\_09. На этот раз используйте System.Threading.Tasks.Task. Выведите статистику работы потоков на экран в виде таблицы



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



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.



Что такое поток управления OS?

Объект ядра операционной системы, которому ОС выделяет процессорное время, наименьшая единица работы ядра ОС.

Последовательность инструкций, выполняемых процессором в выделенные ОС интервалы времени. При создании процесса в нём есть как минимум один основной поток.

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

CreateThread (Windows)

pthread\_create (Linux)

Что такое системные и пользовательские потоки?

Системные потоки (потоки ядра ОС) – выполняют различные сервисы ОС и запускаются ядром ОС, используются для реализации пользовательских потоков.

Пользовательские потоки – потоки, служащие для решения задач пользователя, и запускаемые приложением.

Что такое многопоточность?

Это способность ЦП, позволяющая выполнять 2 или более процесса/потока с инструкциями независимо друг от друга, используя ресурсы одного процессора и разделяя между ними процессорное время.

Что такое контекст потока и для чего он нужен?

Контекст потока – данные, необходимые для возобновления работы потока при его приостановке, а именно:

Программный код, набор регистров, стек памяти, оперативная память, стек ядра и маркер доступа.

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

[5] New – поток создан

[5] Ready – готов к исполнению

[5] Running – исполняется

[12] Sleeping – заснул на некоторое время

[12] Sleeping suspended – спит и приостановлен до события

[7] Suspended ready – готов и приостановлен до события

[5] Blocked – заблокирован извне

[7] Suspended blocked – заблокирован извне и приостановлен до события

[5] Finish – поток завершил исполнение

Что такое LWP?

LWP (light-weight process) – процесс, поддерживающий работу потока пользовательского пространства. Средство достижения многозадачности.

Несколько пользовательских потоков могут быть размещены в одном или нескольких легковесных процессах, что даёт многозадачность на уровне пользователя, которая может иметь некоторые преимущества в производительности.

Что такое потокобезопасность программного кода?

Свойство программного кода (программы) корректно работать в нескольких потоках одновременно. Гарантирует, что при исполнении нескольких потоков код будет правильно себя вести.

Имеет два основных принципа (из лекции Смелова):

1. Код не должен сам себя менять

2. Не должно быть статической области памяти (общей для двух потоков)

Что такое реентерабельность кода?

Компьютерная программа в целом или её отдельная процедура называется реентерабельной, если она разработана таким образом, что одна и та же копия инструкций программы в памяти может быть совместно использована несколькими пользователями или процессами. При этом второй пользователь может вызвать реентерабельный код до того, как с ним завершит работу первый пользователь и это не должно привести к ошибке или потере данных.

Реентерабельность — свойство одной копии программного кода работать в нескольких потоках одновременно. Реентерабельный код всегда потокобезопасен. Реентерабельный код не использует статическую память и не изменяет сам себя, все данные сохраняются в динамической памяти.

Что такое Fiber?

Fiber (волокно) – механизм для ручного планирования выполнения кода в рамках потока. Находится внутри потоков (процессы –> потоки –> волокна) и является особенно легковесным потоком.

Легковесная единица выполнения, которая работает внутри процесса и предоставляет механизм для управления многозадачностью. Являются менее затратными по сравнению с потоками