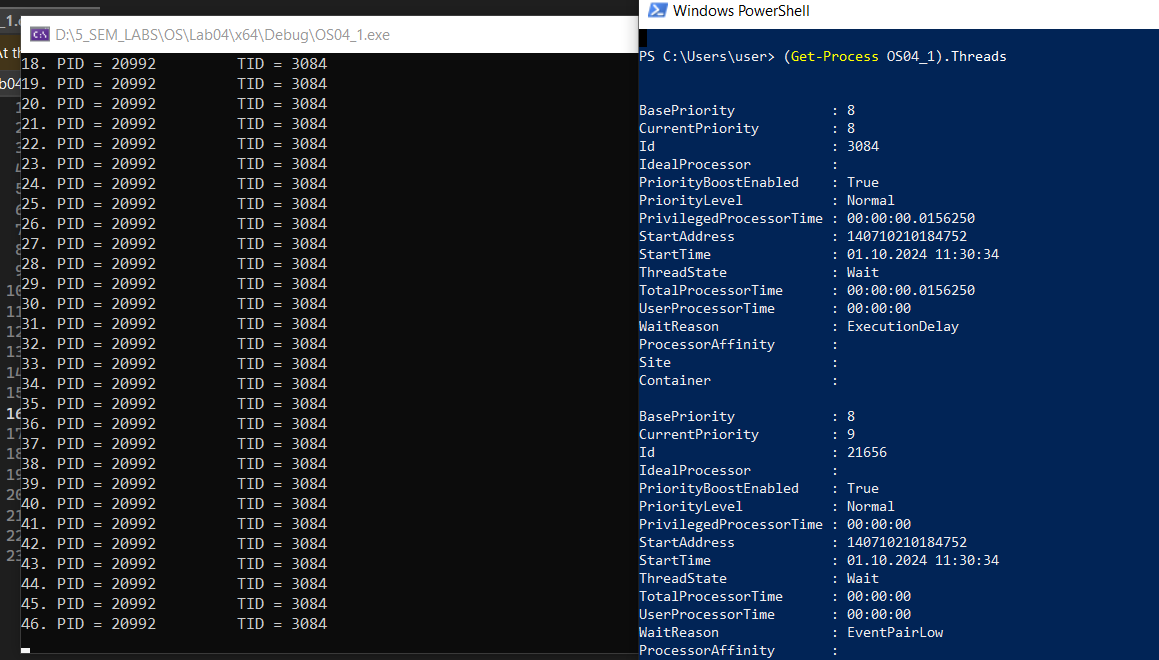
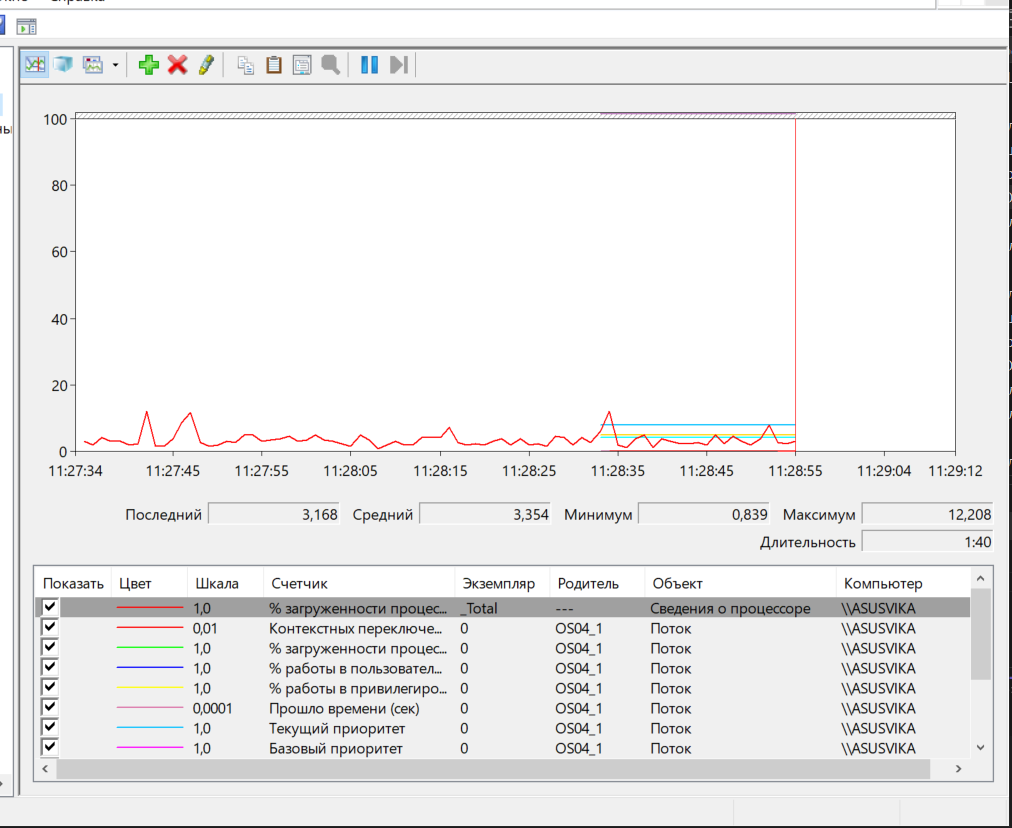
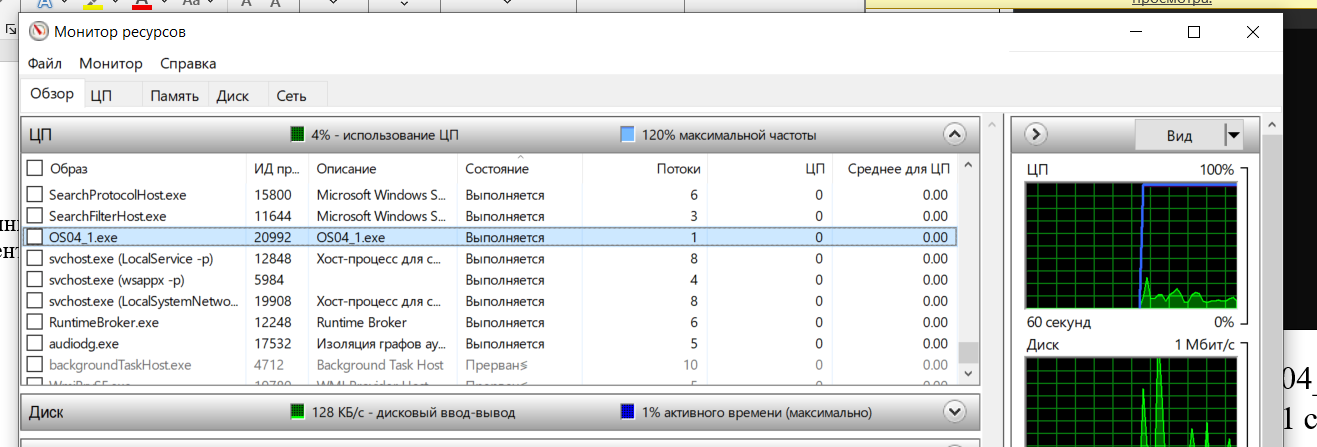
**Лабораторная работа №4**

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





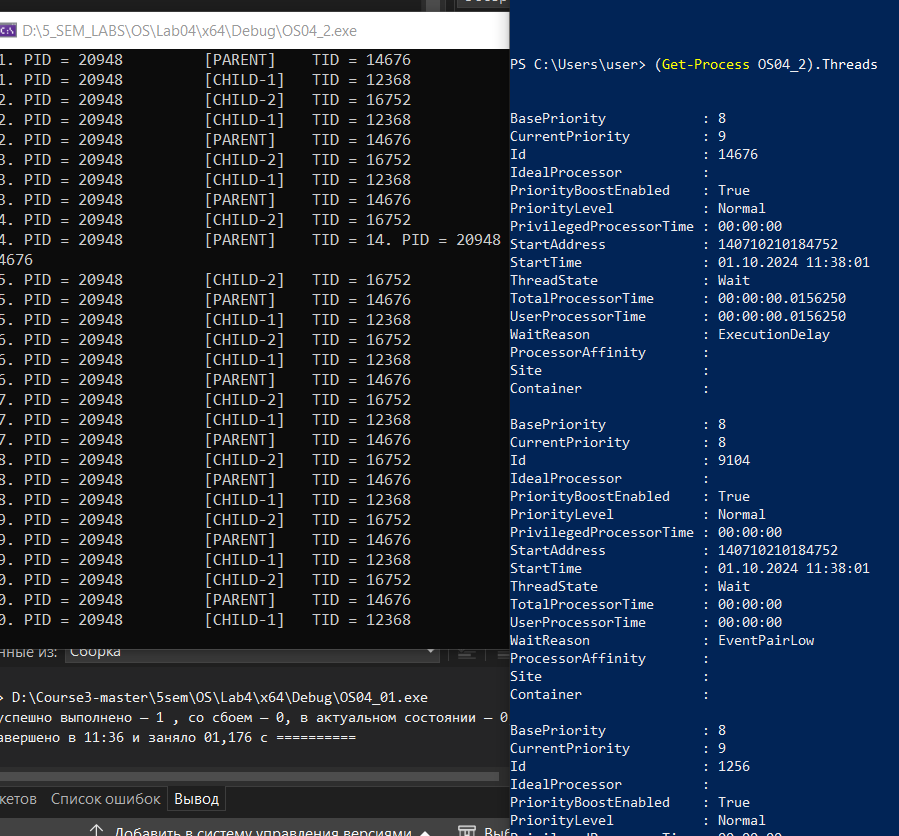


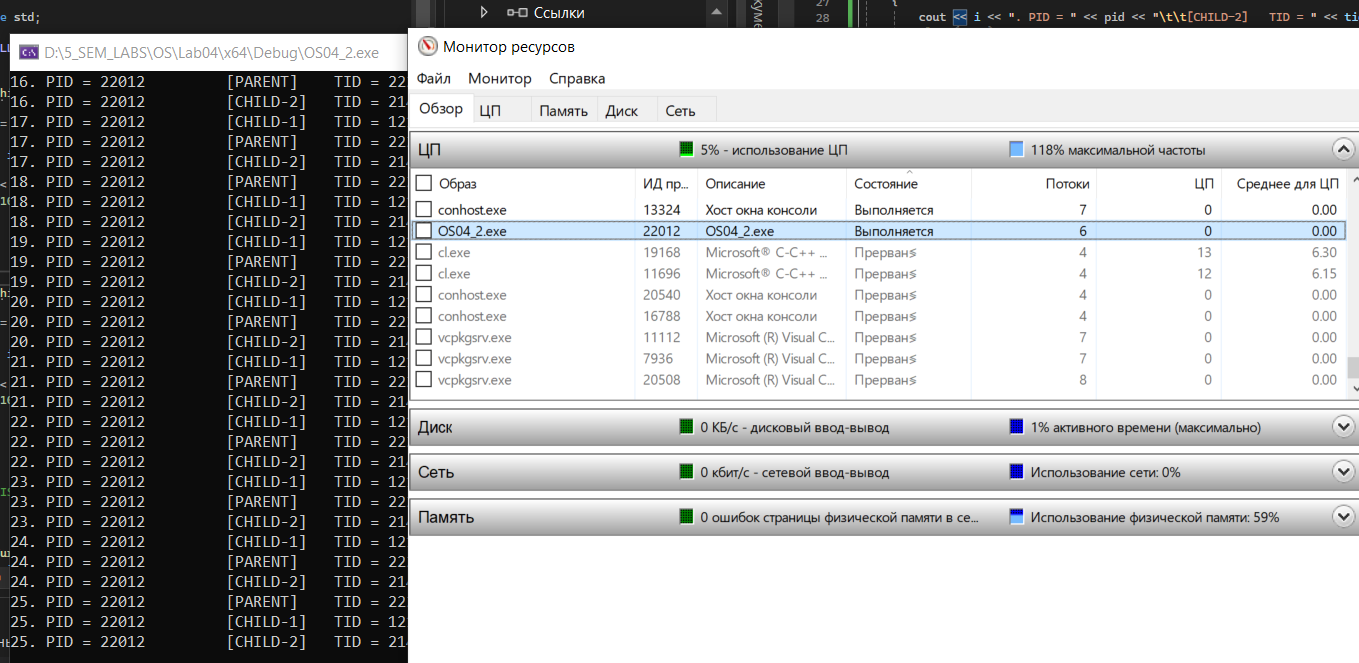
Задание 2. Windows-приложение OS04\_02 на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

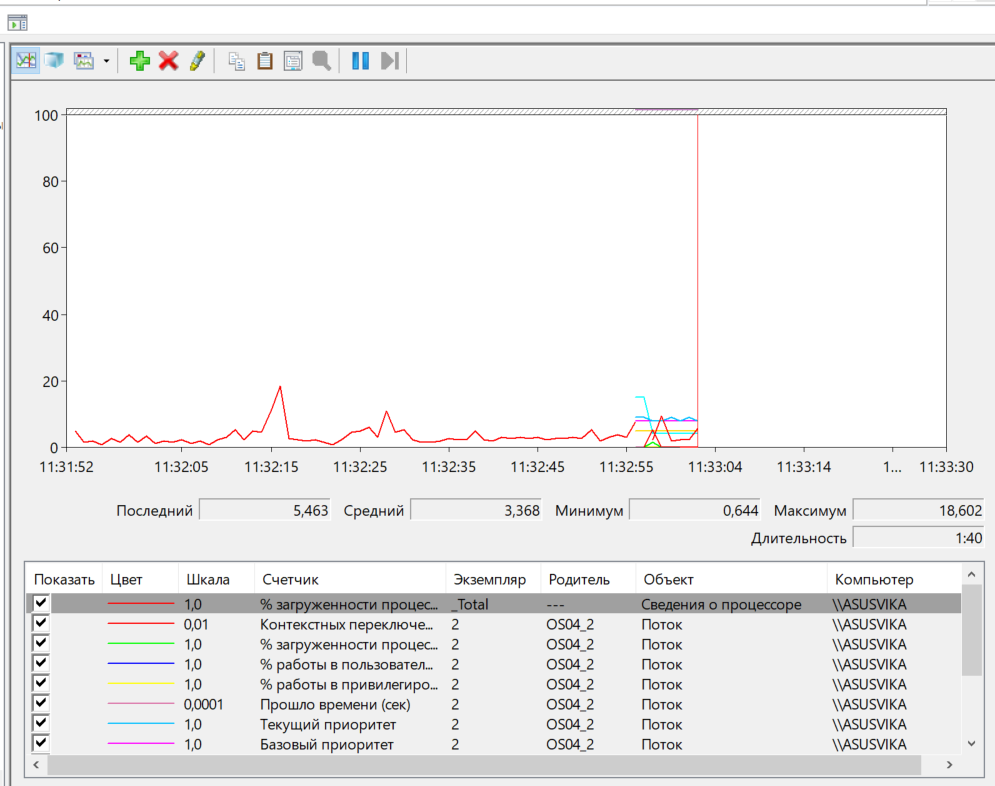
Процесс OS04\_02 должен создать два потока: потоковые функции OS04\_02\_T1, OS04\_02\_T2.

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

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







Задание 3. Windows-приложение OS04\_03 на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

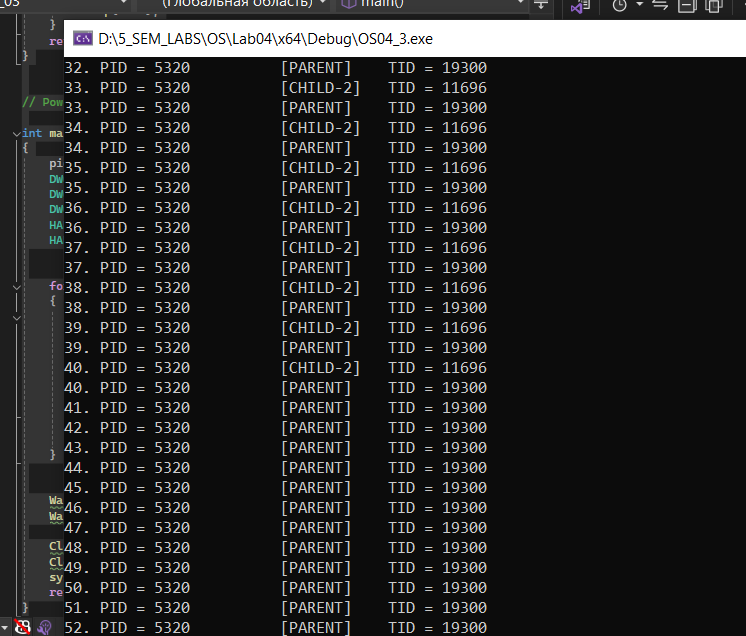
Процесс OS04\_03 должен создать два потока: потоковые функции OS04\_03\_T1, OS04\_03\_T2.

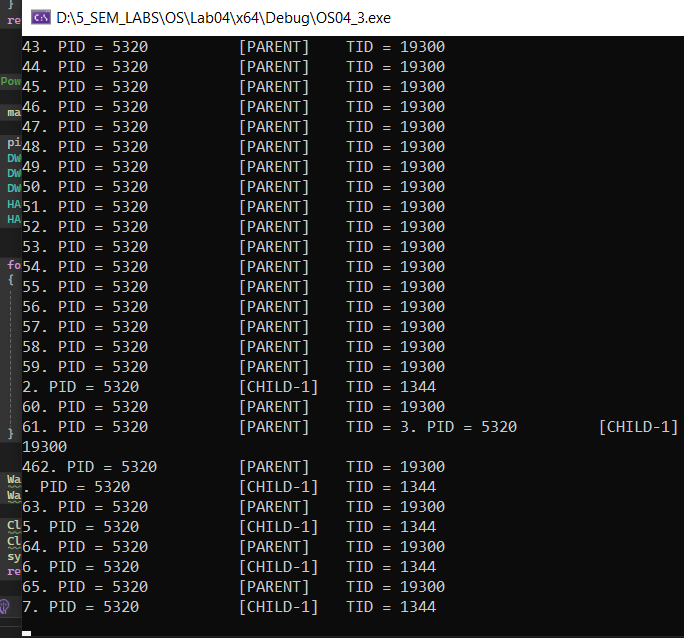
Поток OS04\_03\_T1 - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

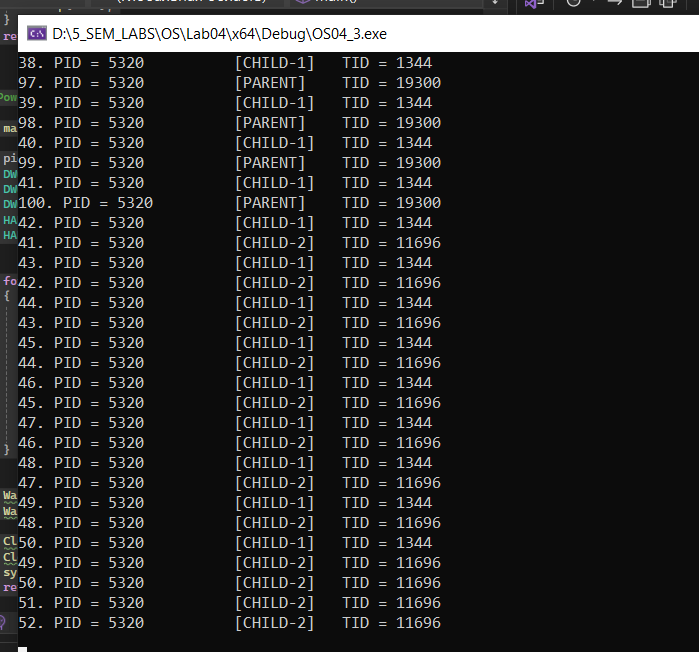
Поток OS04\_03\_T2 - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

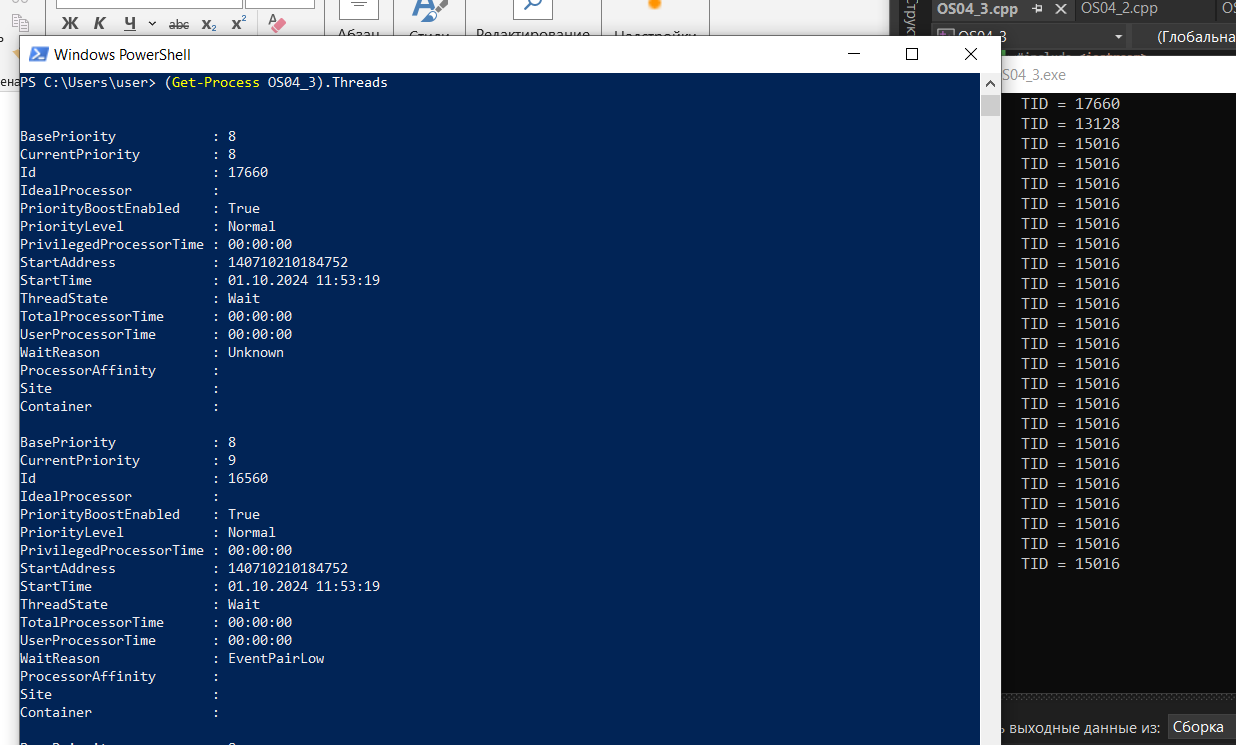
Поток main приостанавливает работу потока OS04\_03\_T1 на 20й своей итерации и возобновляют на 60й своей итерации.

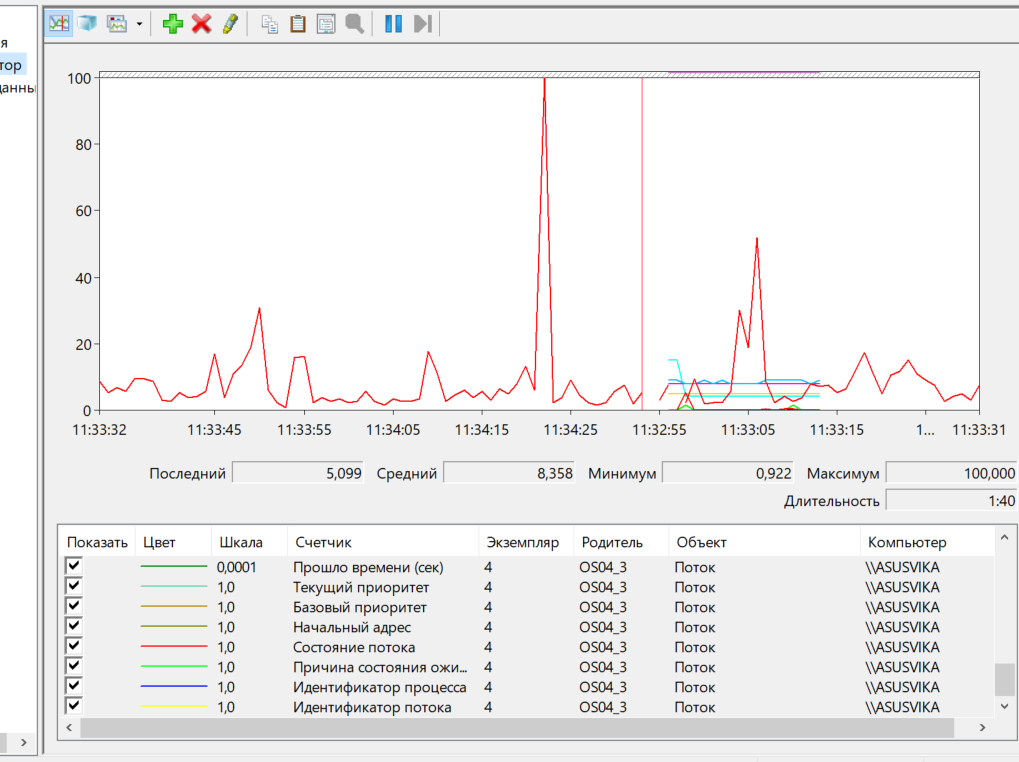
Поток main приостанавливает работу потока OS04\_03\_T2 на 40й своей итерации и возобновляют после окончания собственных итераций (по окончании собственного цикла).

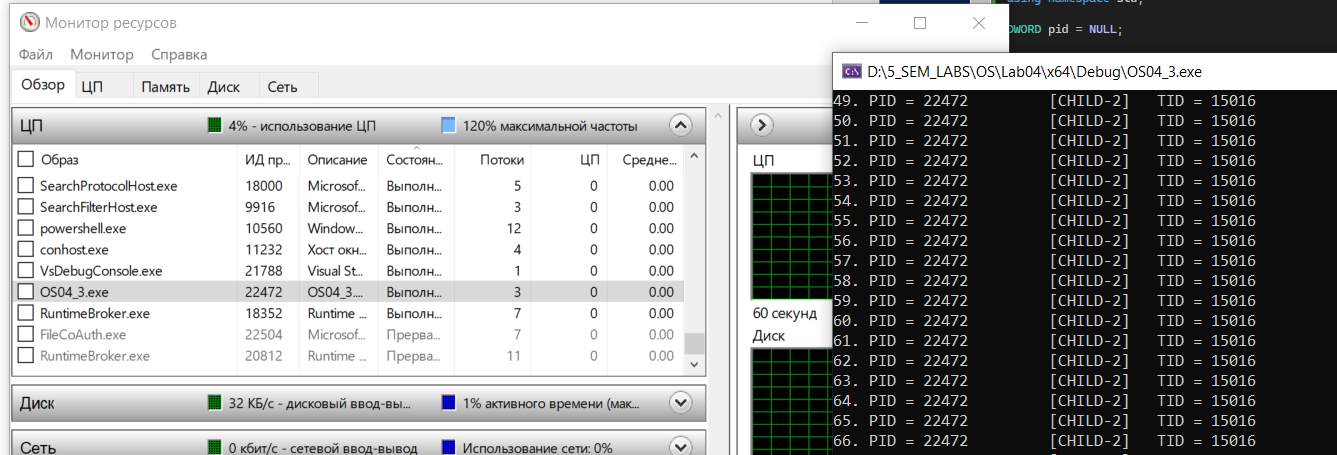












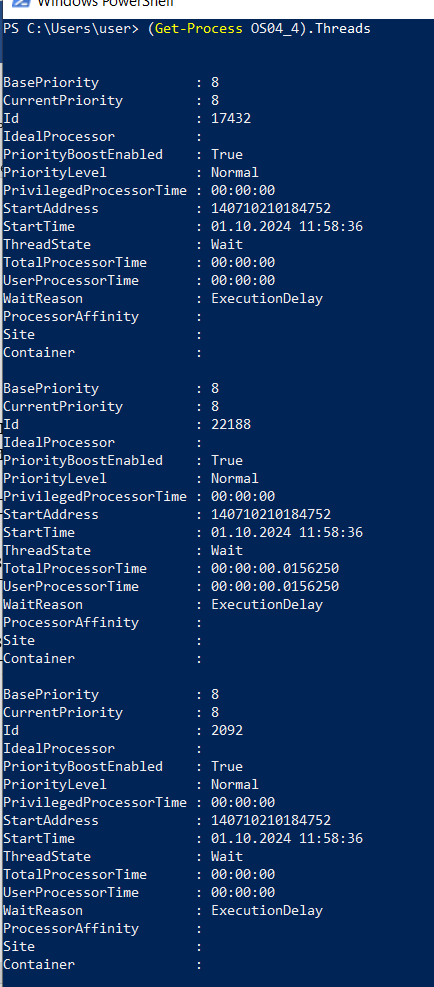
Задание 4. Windows-приложение OS04\_04 на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

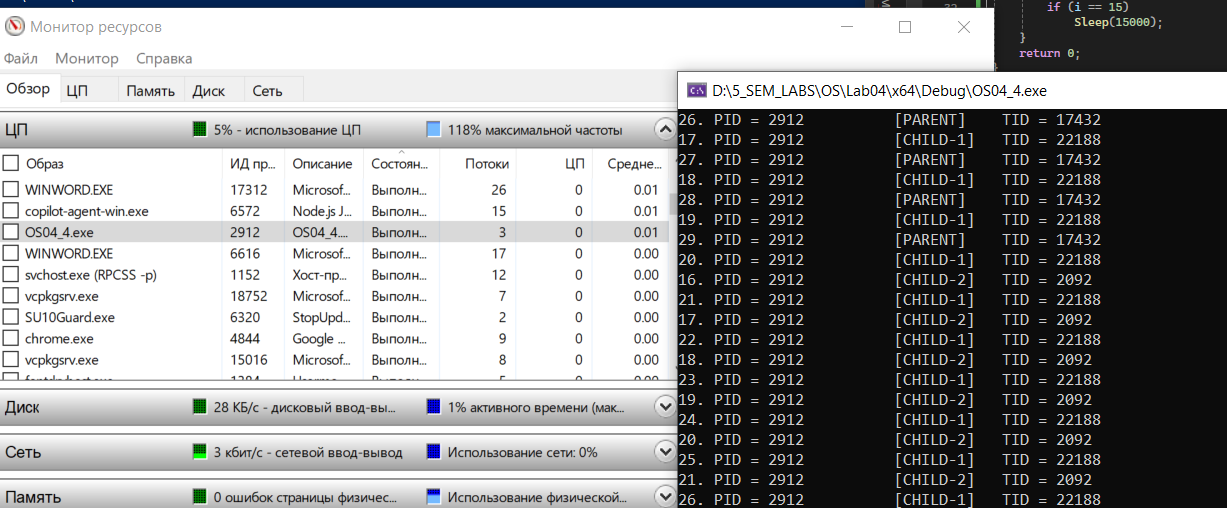
Процесс OS04\_04 должен создать два потока: потоковые функции OS04\_04\_T1, OS04\_04\_T2.

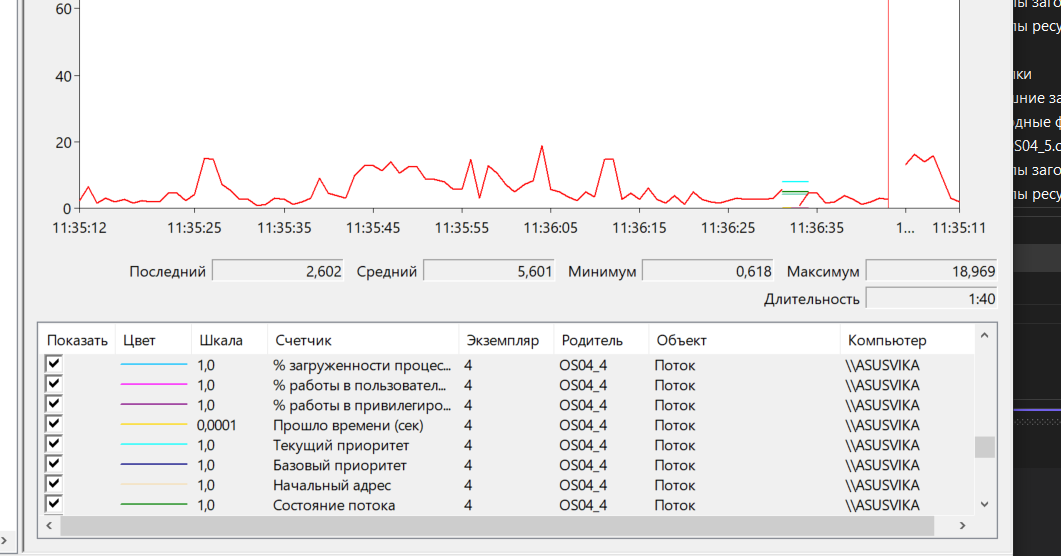
Поток OS04\_04\_T1 - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока. Поток засыпает на 25й собственный итерации на 10сек.

Поток OS04\_04\_T2 - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока. Поток засыпает на 80й собственной итерации на 15сек.

Поток main засыпает на 30й итерации на 10 сек.







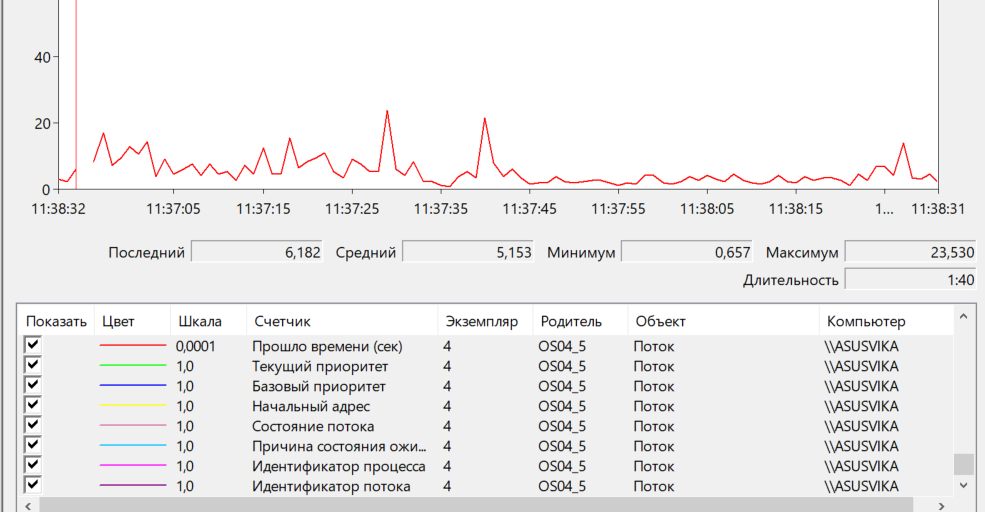
Задание 5. Windows-приложение OS04\_05 на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

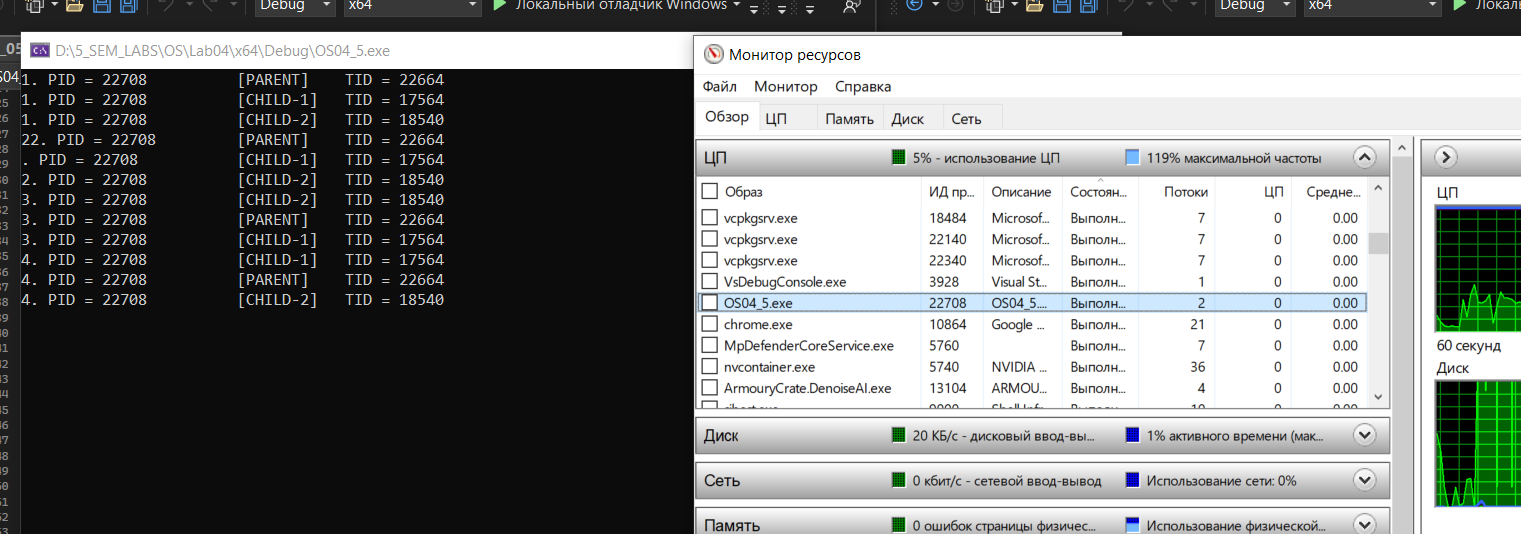
Процесс OS04\_05 должен создать два потока: потоковые функции OS04\_05\_T1, OS04\_05\_T2.

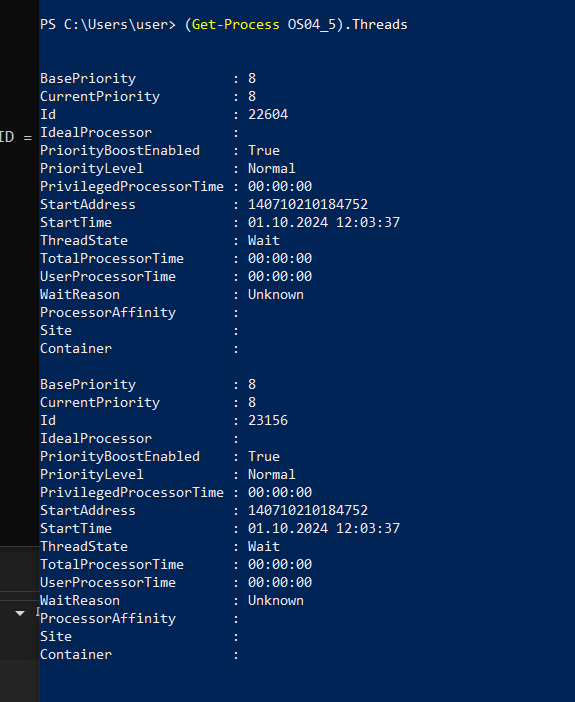
Поток OS04\_05\_T1 - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

Поток OS04\_05\_T2 - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

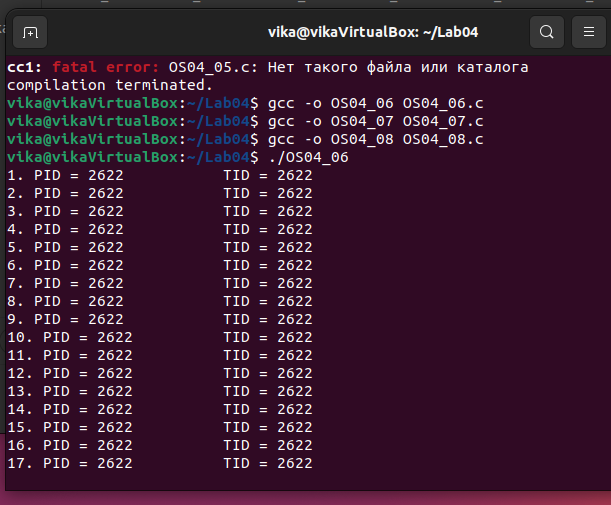
Поток main завершает поток OS04\_05\_T2 на 40й собственной итерации.

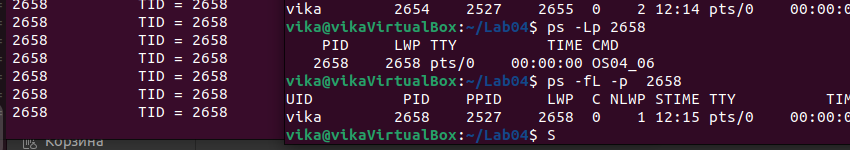






Задание 6. Linux-приложение OS04\_06 на языке С, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификатора процесса.

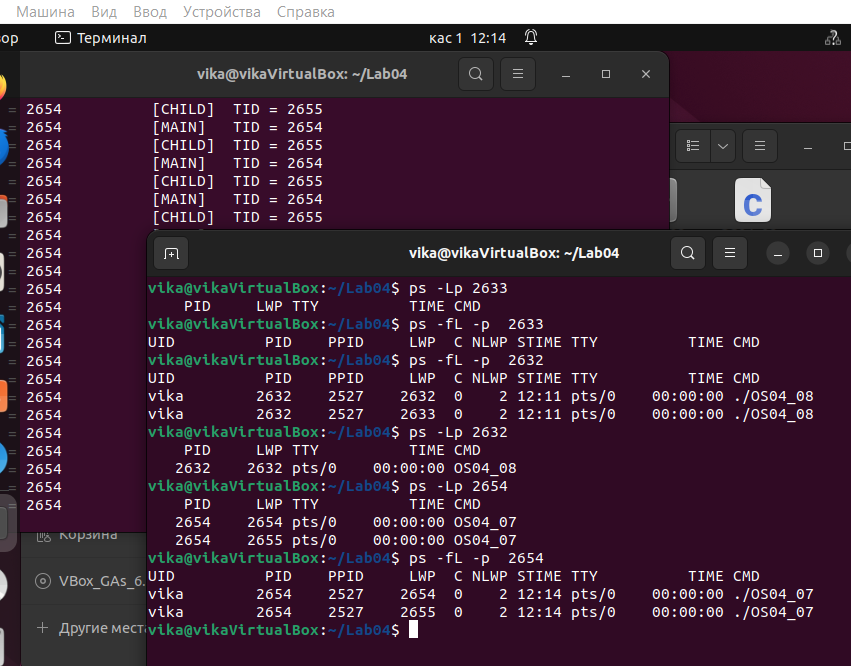




Задание 7. Linux-приложение OS04\_07 на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

Процесс OS04\_07 должен создать поток: потоковая функция OS04\_07\_T1.

Поток OS04\_07\_T1 - выполняет цикл 75 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса.

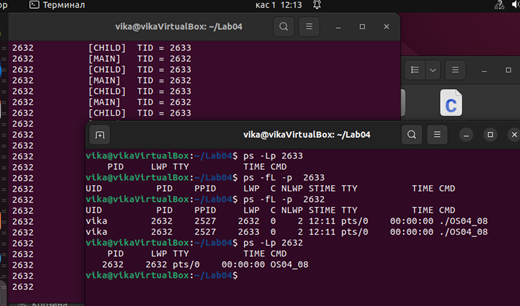


Задание 8. Linux-приложение OS04\_08 на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

Процесс OS04\_08 должен создать поток: потоковая функция OS04\_08\_T1.

Поток OS04\_08\_T1 - выполняет цикл 75 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса. Поток должен засыпать на 10сек. на 50й собственной итерации.

Поток main должен засыпать на 15 сек. на 30й собственной итерации.



**38. Что такое поток управления OS?**

Поток управления (или просто поток) — это единица выполнения, управляемая операционной системой (OS). Потоки позволяют программе выполнять несколько операций одновременно, что может повысить производительность и отзывчивость приложений. Каждый поток имеет свой собственный контекст выполнения, включая регистры, стек и состояние.

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

* **Windows**: Для создания потоков используется функция CreateThread. Также можно использовать другие функции, такие как \_beginthreadex.
* **Linux**: Потоки создаются с помощью системного вызова pthread\_create из библиотеки POSIX Threads (pthreads).

**40. Что такое системные и пользовательские потоки?**

* **Системные потоки**: Управляются операционной системой и имеют доступ ко всем системным ресурсам. Они используют механизмы ОС для планирования и управления.
* **Пользовательские потоки**: Управляются приложением или библиотекой, а не ОС. Они обычно требуют более сложного управления, но могут быть быстрее, так как не требуют переключения контекста на уровне ОС.

**41. Что такое многопоточность?**

Многопоточность — это способность программы или системы выполнять несколько потоков одновременно. Это позволяет эффективно использовать ресурсы процессора, улучшает производительность приложений и обеспечивает более отзывчивый интерфейс пользователя.

**42. Чем отличаются приоритетная многопоточность от кооперативной многопоточности?**

* **Приоритетная многопоточность**: Потоки с разными приоритетами могут быть выполнены в зависимости от их важности. ОС управляет временем выполнения потоков на основе их приоритетов.
* **Кооперативная многопоточность**: Потоки сами решают, когда они должны уступить управление другим потокам. Это требует от разработчиков явного управления временем выполнения потоков, что может привести к проблемам с блокировками.

**43. Что такое диспетчеризация потоков управления OS?**

Диспетчеризация потоков — это процесс, с помощью которого операционная система управляет выполнением потоков. Это включает в себя планирование, переключение контекста и управление состоянием потоков, чтобы обеспечить эффективное использование процессорных ресурсов.

**44. Что такое контекст потока и для чего он нужен?**

Контекст потока — это набор информации, необходимой для управления выполнением потока, включая его регистры, стек, состояние и другие параметры. Контекст необходим для переключения между потоками, чтобы ОС могла восстановить выполнение потока с того места, где оно было приостановлено.

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

* **Создан**: Поток создан, но еще не начал выполнение.
* **Готов**: Поток готов к выполнению, но ожидает, когда ОС выделит ему процессор.
* **Выполняется**: Поток в данный момент выполняется на процессоре.
* **Заблокирован**: Поток не может продолжать выполнение из-за ожидания завершения операции (например, ввода-вывода).
* **Завершен**: Поток завершил выполнение.

**46. Что такое LWP?**

LWP (Lightweight Process) — это легковесный процесс, который управляется операционной системой и может быть выполнен параллельно с другими процессами. LWP позволяет эффективно управлять многопоточностью, обеспечивая более легкое переключение контекста по сравнению с полными процессами.

**47. Что такое потокобезопасность программного кода?**

Потокобезопасность означает, что код может выполняться несколькими потоками одновременно без риска нарушения целостности данных или возникновения непредсказуемого поведения. Это достигается за счет использования механизмов синхронизации, таких как мьютексы и семафоры.

**48. Что такое реентерабельность кода?**

Реентерабельность кода — это свойство, позволяющее функции или процедуре безопасно выполняться одновременно несколькими потоками без риска изменения глобальных или статических данных. Реентерабельный код не использует общие ресурсы без синхронизации и может быть вызван в любой момент.

**49. Что такое Fiber?**

Fiber — это легковесная единица выполнения, которая управляется приложением, а не операционной системой. Фибры позволяют более эффективно управлять многопоточностью, так как они могут переключаться между собой, не требуя переключения контекста на уровне ОС.

**50. Дайте развернутое определение потока OS.**

Поток OS — это базовая единица выполнения в операционной системе, которая содержит собственный контекст выполнения и может выполняться параллельно с другими потоками. Каждый поток имеет свою собственную стековую память и регистры, что позволяет ему выполнять задачи независимо. Потоки могут взаимодействовать друг с другом и совместно использовать ресурсы, такие как память, что делает их эффективным инструментом для разработки многопоточных приложений. Операционная система управляет потоками, контролируя их создание, выполнение и завершение, а также обеспечивая диспетчеризацию и синхронизацию между ними.