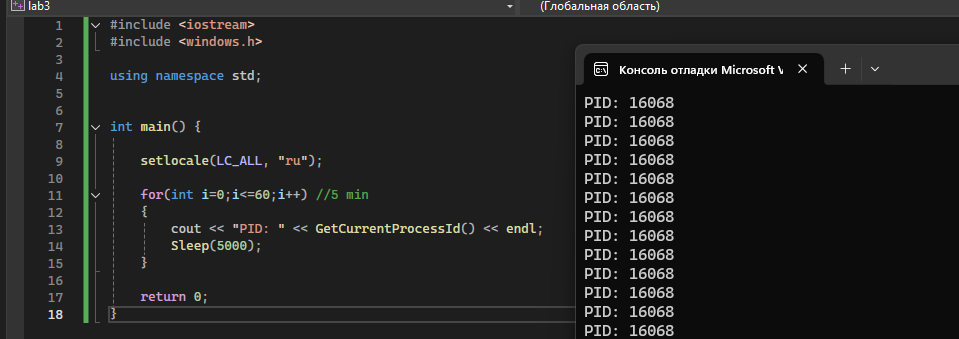
ОТЧЁТ

Операционные Системы

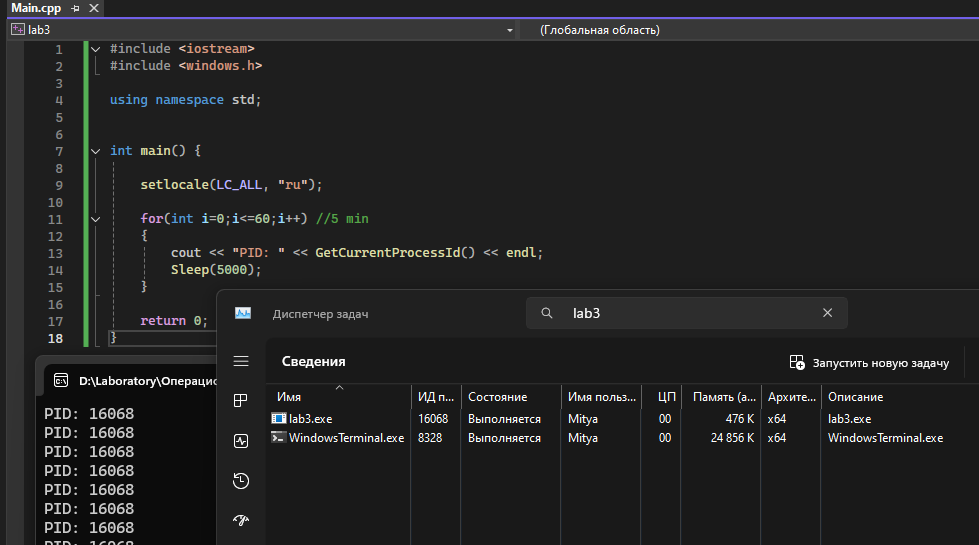
Беласин Д. А.

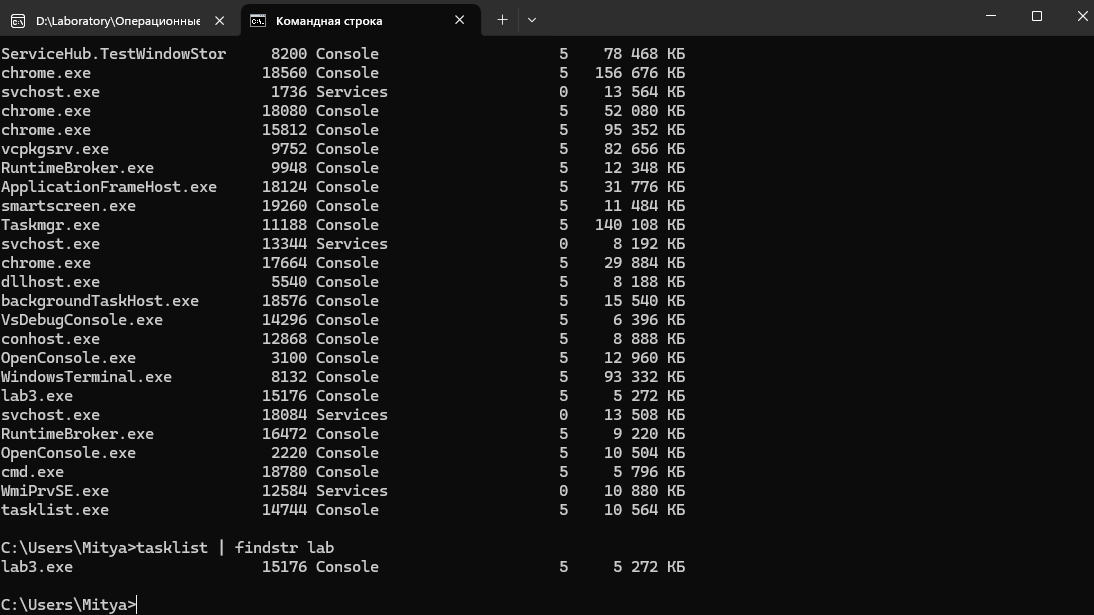
Лабораторная №3

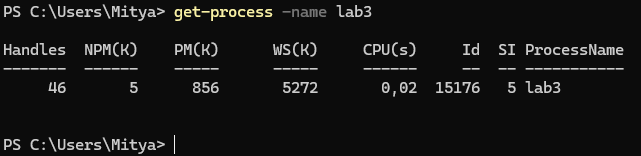
**Задание 01**

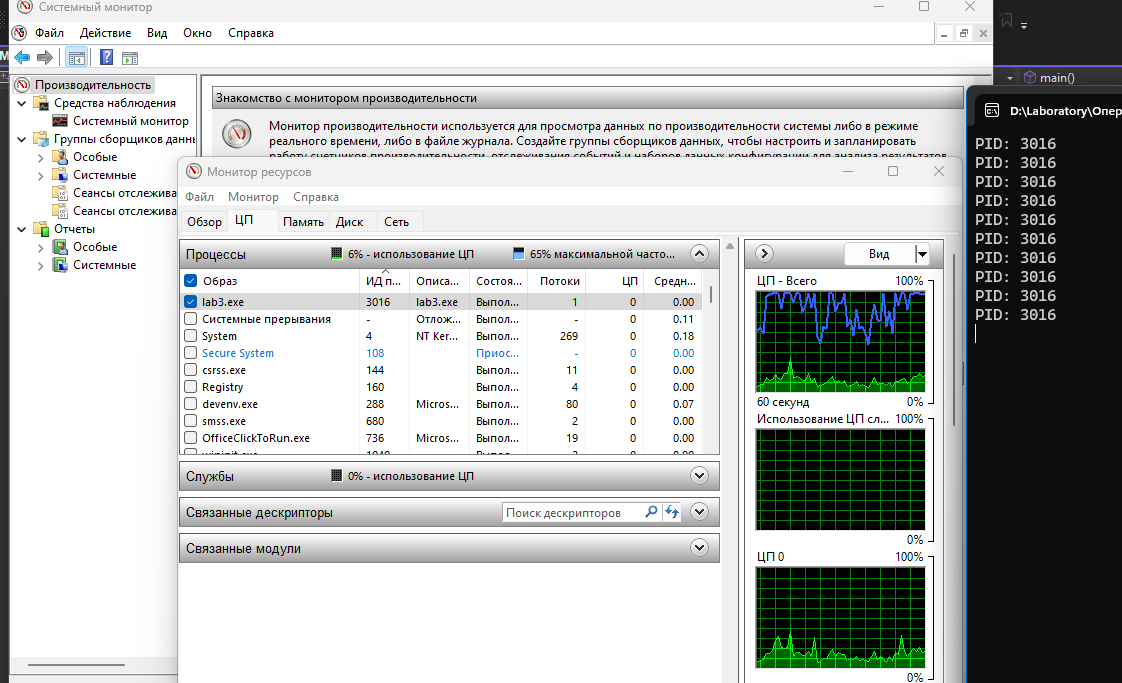
1. Разработайте консольное Windows-приложение OS03\_01 на языке С++, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификатора процесса. 

2. Продемонстрируйте информацию о процессе OS03\_01 с помощью утилит Task Manager, tasklist, PowerShell ISE и Performance Monitor.

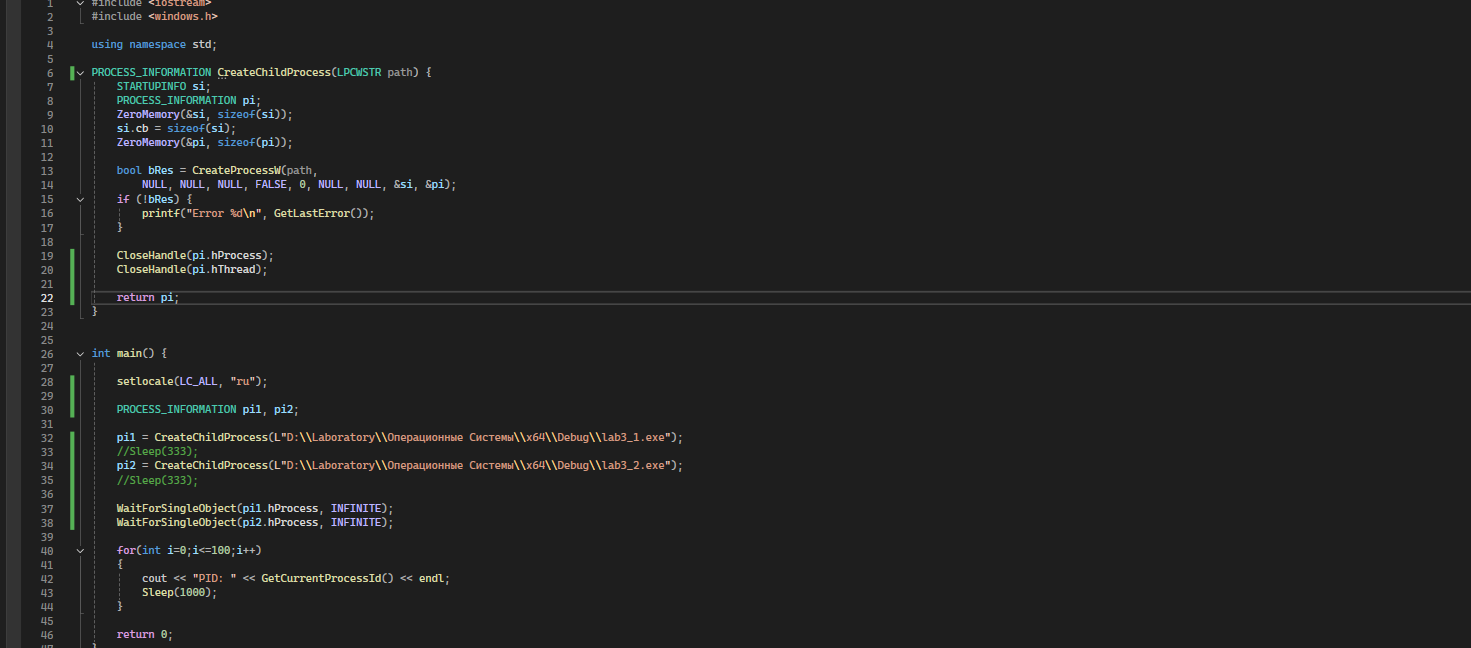
Task Manager (Диспетчер задач)

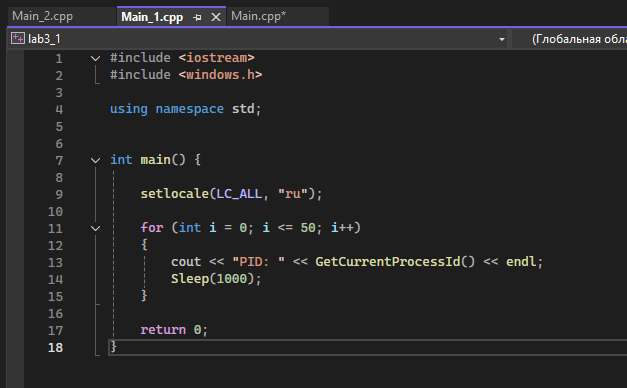
tasklist 

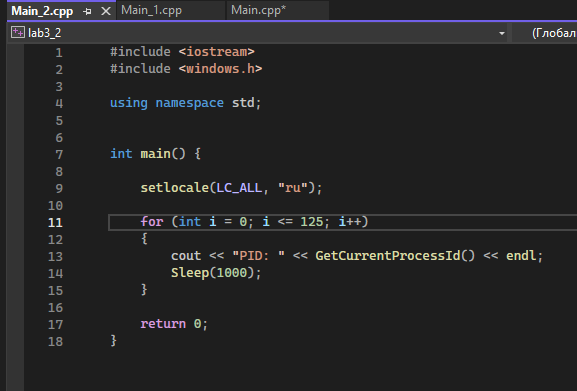
PowerShell ISE 

Performance Monitor  **Задание 02**

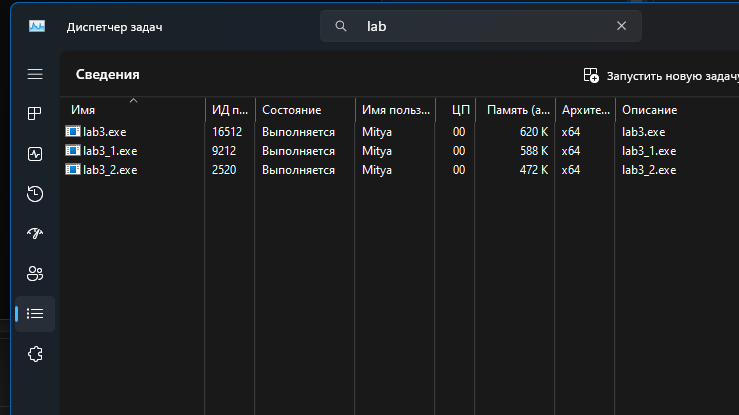
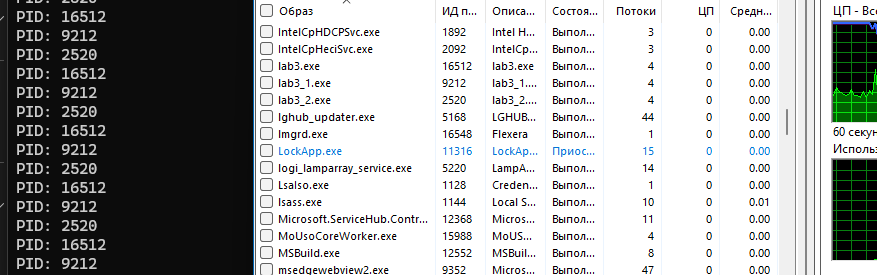
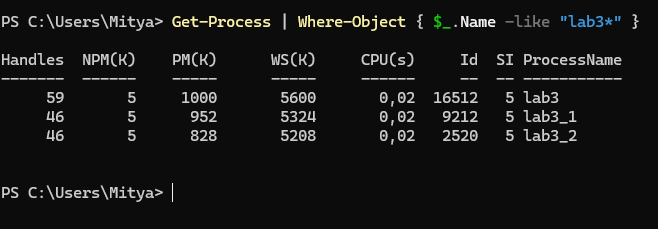
3. Разработайте консольное Windows-приложение OS03\_02 на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

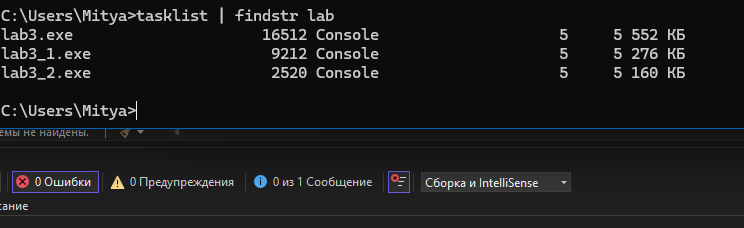
4. Приложение OS03\_02 должно создавать два дочерних процесса OS03\_02\_1 и OS03\_02\_2. 

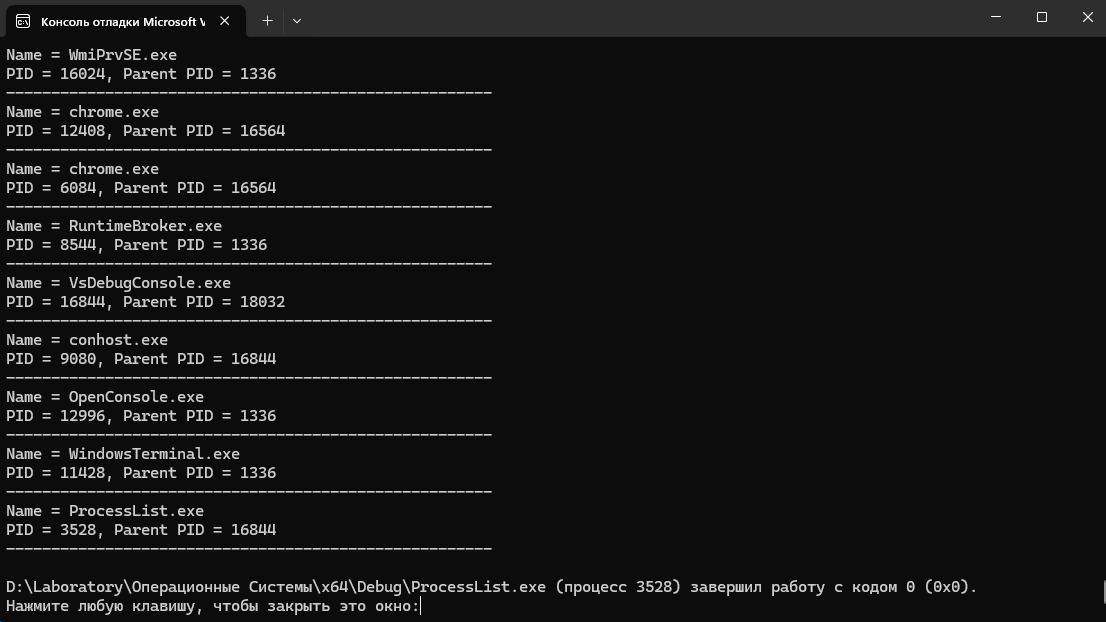
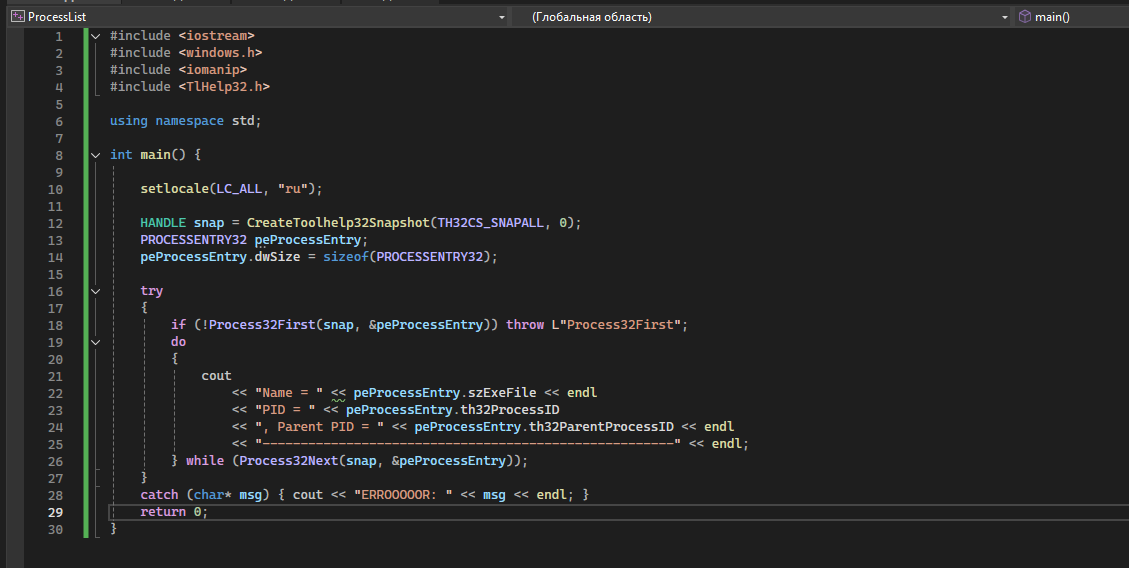
5. Процесс OS03\_02\_1 - консольное Windows-приложение, выполняющее цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса. 

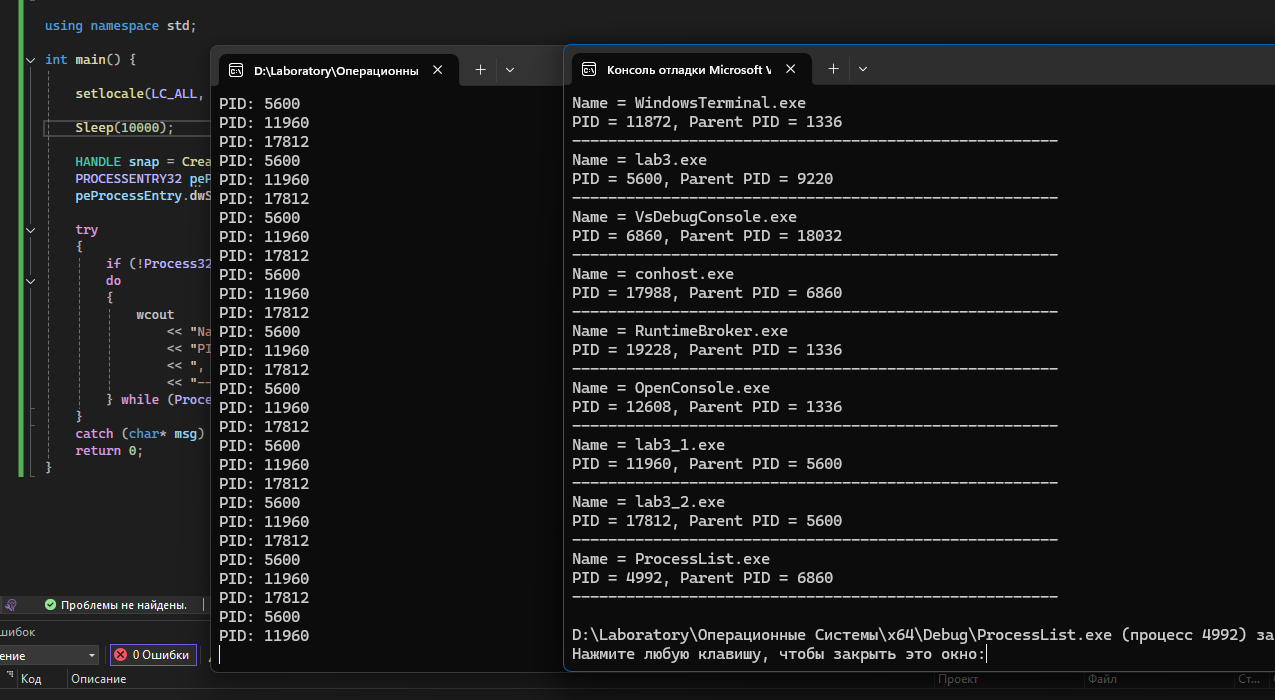
6. Процесс OS03\_02\_2 - консольное Windows-приложение выполняющее цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса. 

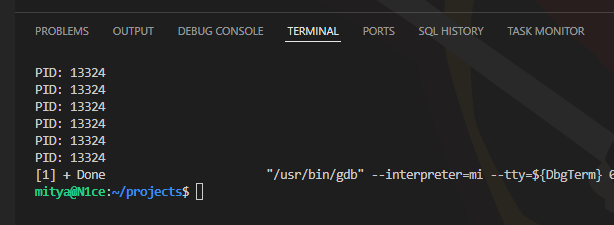
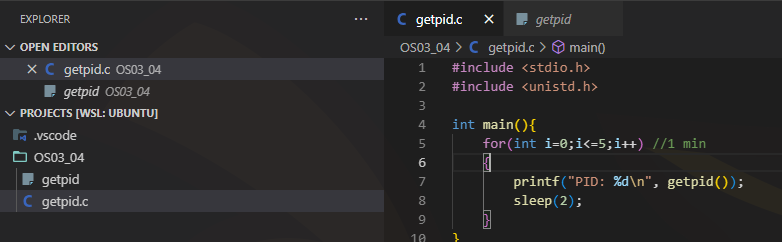
7. Продемонстрируйте информацию о процессах OS03\_02, OS03\_02\_1 и OS03\_02\_2 с помощью утилит Task Manager, tasklist, PowerShell ISE и Performance Monitor.



**Задание 03**

8. Разработайте консольное Windows-приложение OS03\_03 на языке С++, выводящее на консоль перечень выполняющихся процессов в данный момент в OS. (wcout)

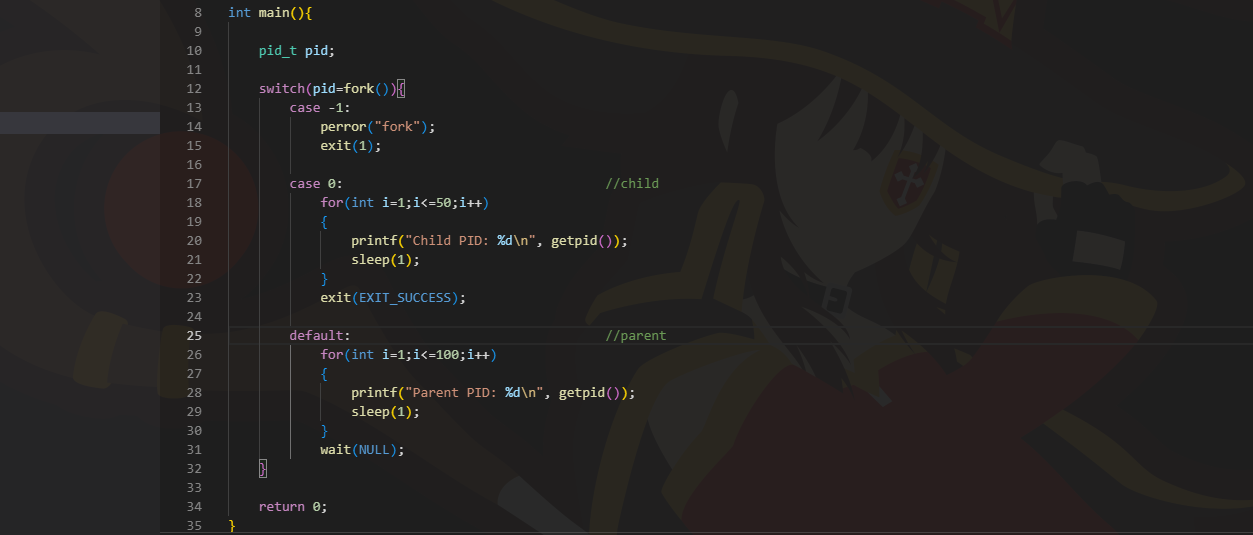
9. Запустите приложение OS03\_02 и продемонстрируйте с помощью приложения OS03\_03 в перечне процессов OS03\_02, OS03\_02\_1, OS03\_02\_2 и OS03\_03. **Задание 04**

10. Разработайте консольное Linux-приложение OS03\_04 на языке С, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификатора процесса. 

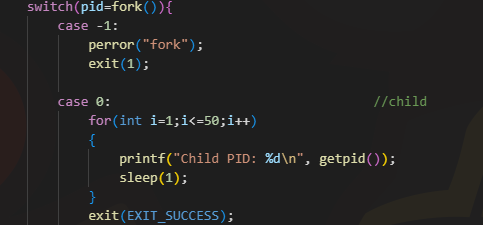
11. Продемонстрируйте информацию о процессе OS03\_04 с помощью файловой системы /proc.



12. Продемонстрируйте информацию о процессе OS03\_04 с помощью утилиты ps. **Задание 05**

13. Разработайте консольное Linux-приложение OS03\_05 на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса. 

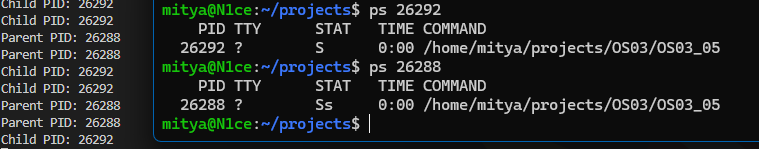
14. Приложение OS03\_05 должно создавать один дочерний процесс OS03\_05\_1 с помощью системного вызова fork. Процесс OS03\_05\_1 в этом случае не является отдельным модулем, а встроен (fork) в программный модуль OS03\_05.



15. Процесс OS03\_05\_1 - консольное Linux-приложение, выполняющее цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса. 

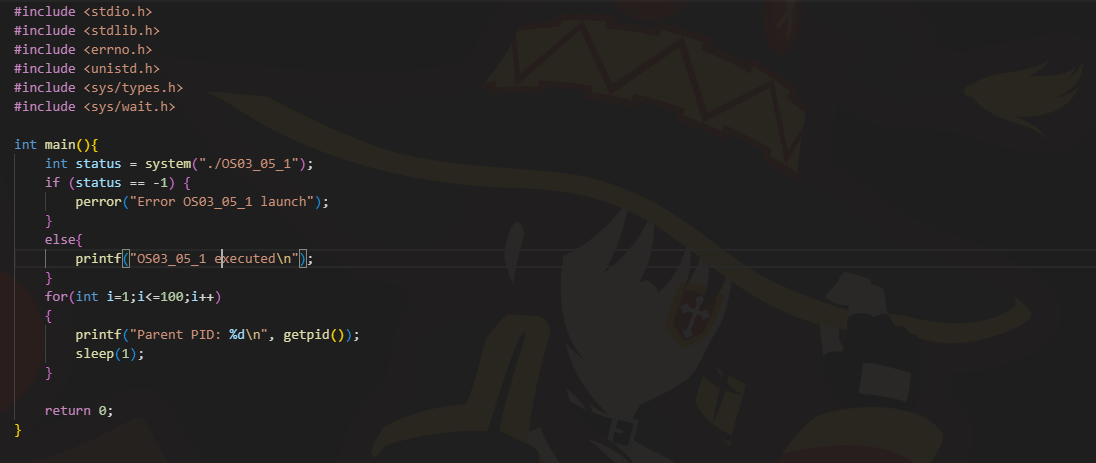
16. Продемонстрируйте информацию о процессах OS03\_05 и OS03\_05\_1 с помощью файловой системы /proc. 

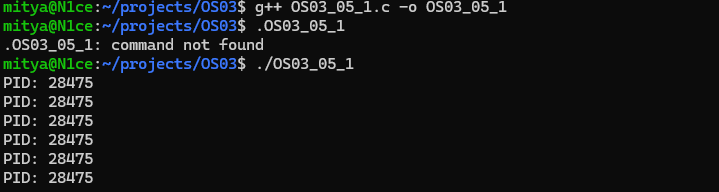


17. Продемонстрируйте информацию о процессах OS03\_05 и OS03\_05\_1 с помощью утилиты ps. 

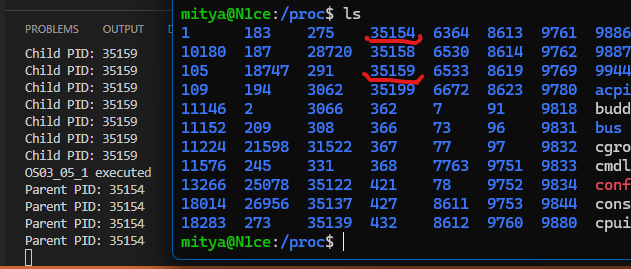
**Задание 06**

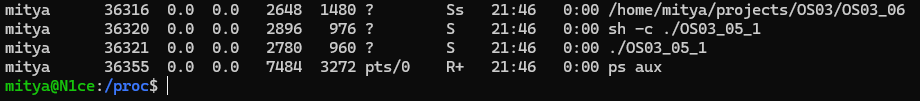
18. Разработайте консольное Linux-приложение OS03\_06 на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.



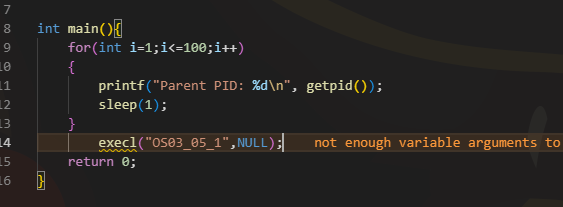
19. Приложение OS03\_06 должно создавать один дочерний процесс OS03\_05\_1 (отдельный модуль) с помощью системного вызова system. 



20. Продемонстрируйте информацию о процессах OS03\_06 и OS03\_05\_1 с помощью файловой системы /proc. 

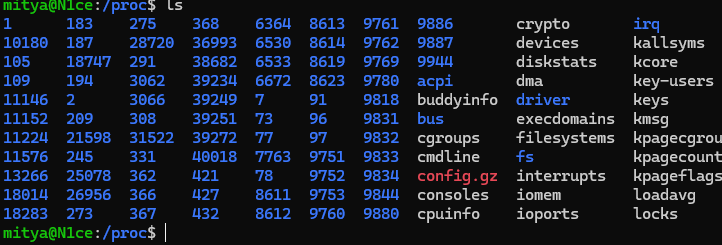
21. Продемонстрируйте информацию о процессах OS03\_06 и OS03\_05-1 с помощью утилиты ps. 

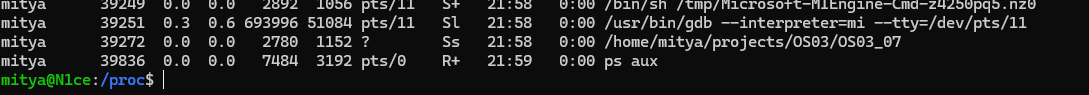
**Задание 07**

22. Разработайте консольное Linux-приложение OS03\_07 на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса 

23. Приложение OS03\_07 должно создавать один дочерний процесс OS03\_05\_1 (отдельный модуль) с помощью системного вызова exec. 

24. Продемонстрируйте информацию о процессах OS03\_07 и OS03\_05\_1 с помощью файловой системы /proc.



25. Продемонстрируйте информацию о процессах OS03\_07 и OS03\_05-1 с помощью утилиты ps. 

26. Продемонстрируйте разницу системных вызовов system и exec.

System запускает введенную команду как в консоли, а exec загружает новую программу вместо памяти старой программы.

**Задание 08**.

Ответьте на следующие вопросы

27. Что такое процесс?

Процесс — это выполняемая программа. Это единица работы, которую операционная система может управлять. Каждый процесс имеет своё состояние и ресурсы, такие как память и время процессора.

28. Что такое контекст процесса?

Контекст процесса — это вся информация, необходимая для управления и выполнения процесса. В него входят содержимое регистров процессора, состояние памяти и другие системные ресурсы.

29. Что такое родительский и дочерний процесс?

Родительский процесс — это процесс, который создает другие процессы. Дочерний процесс — это тот, который был создан другим процессом. Родительский процесс может управлять дочерним и обмениваться с ним данными.

30. Что такое процесс инициализации OS?

Процесс инициализации ОС — это первый процесс, который запускается при старте операционной системы. Он отвечает за загрузку и запуск других системных процессов.

31. Перечислите области памяти процесса и поясните их назначение.

**Код**: хранит исполняемый код программы.

**Данные**: хранит глобальные и статические переменные.

**Куча**: используется для динамического выделения памяти.

**Стек**: хранит временные данные, такие как параметры функций и адреса возврата.

32. Чем отличаются системные процессы от пользовательских?

Системные процессы выполняют задачи, необходимые для работы операционной системы, и обычно запускаются с высокими привилегиями. Пользовательские процессы запускаются пользователями и выполняют пользовательские программы.

33. Что такое Windows-сервисы, Linux-демоны?

Windows-сервисы и Linux-демоны — это фоновые процессы, которые выполняют важные системные или пользовательские задачи без взаимодействия с пользователем напрямую.

34. С помощью каких системных вызовов можно создать дочерний процесс в Windows? Поясните разницу.

В Windows дочерний процесс создается с помощью CreateProcess. Этот вызов предоставляет много параметров для настройки нового процесса, включая его приоритет и окружение.

35. С помощью каких системных вызовов можно создать дочерний процесс в Linux? Поясните разницу.

В Linux для создания дочернего процесса используется fork. Этот вызов создает точную копию текущего процесса. exec может быть использован для замены кода процесса другим исполняемым файлом.

36. Какие потоки данных доступны любому процессу автоматически?

Каждый процесс автоматически имеет доступ к стандартным потокам данных: стандартный ввод (stdin), стандартный вывод (stdout) и стандартный вывод ошибок (stderr).

37. Поясните назначение системного вызова WaitForSingleObject в Windows-приложении.

WaitForSingleObject используется для ожидания завершения конкретного события, например, окончания выполнения процесса или освобождения ресурса.

38. Поясните назначение системного вызова wait в Linux приложении.

wait ожидает завершения дочернего процесса и позволяет родительскому процессу получить информацию о его завершении, например, код возврата.

39. Дайте развернутое определение процесса OS.

Процесс ОС — это экземпляр выполняемой программы, который управляется операционной системой. Он содержит код, данные и системные ресурсы, необходимые для выполнения задач, а также имеет своё состояние выполнения и контекст.