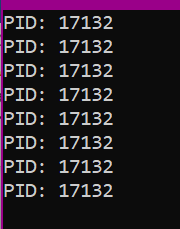
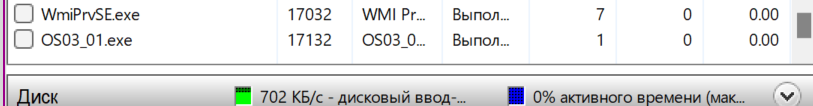
Задание 01

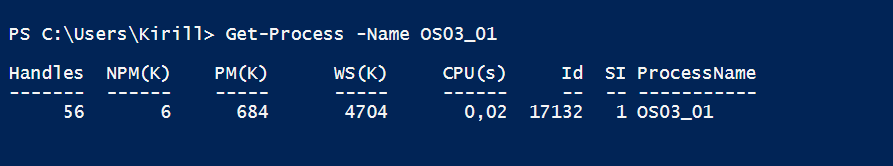
1. Разработайте консольное Windows-приложение OS03\_01 на языке С++, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификатора процесса.

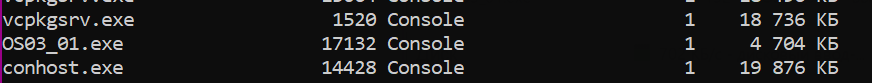
2. Продемонстрируйте информацию о процессе OS03\_01 с помощью утилит Task Manager, tasklist, PowerShell ISE и Performance Monitor.











Задание 02

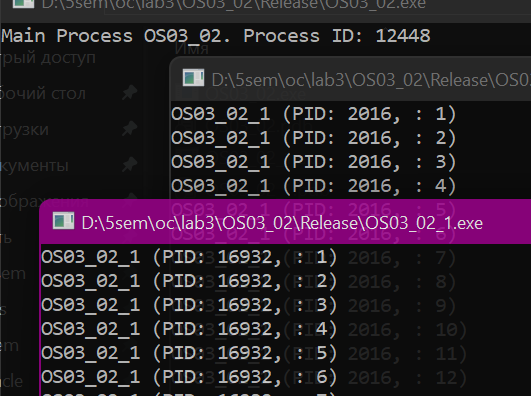
3. Разработайте консольное Windows-приложение OS03\_02 на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

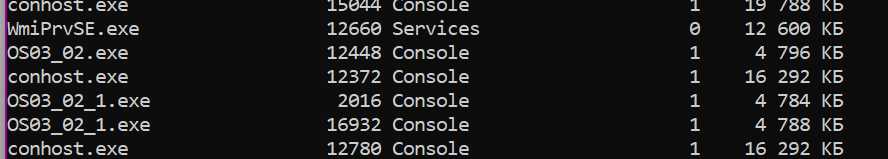
4. Приложение OS03\_02 должно создавать два дочерних процесса OS03\_02\_1 и OS03\_02\_2.

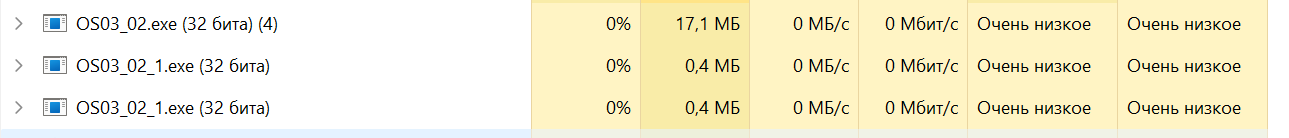
5. Процесс OS03\_02\_1 - консольное Windows-приложение, выполняющее цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

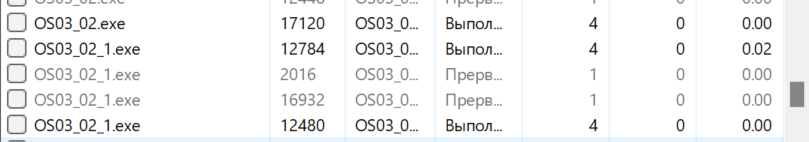
6. Процесс OS03\_02\_2 - консольное Windows-приложение выполняющее цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

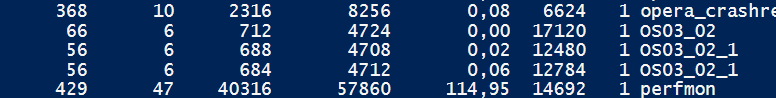
7. Продемонстрируйте информацию о процессах OS03\_02, OS03\_02\_1 и OS03\_02\_2 с помощью утилит Task Manager, tasklist, PowerShell ISE и Performance Monitor.







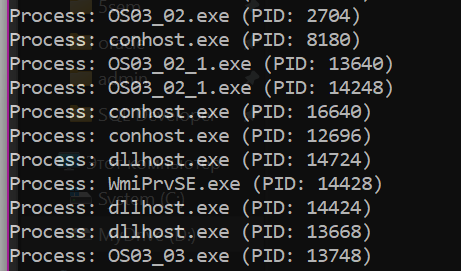




Задание 03.

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

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.

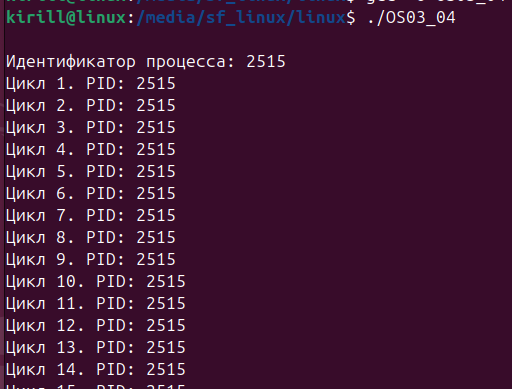


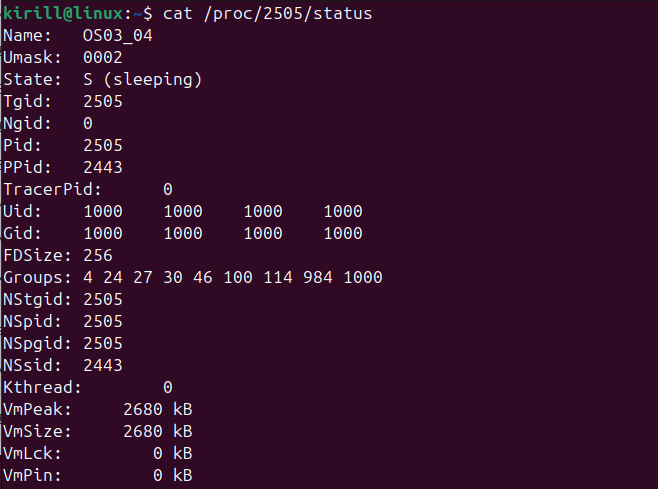
Задание 04

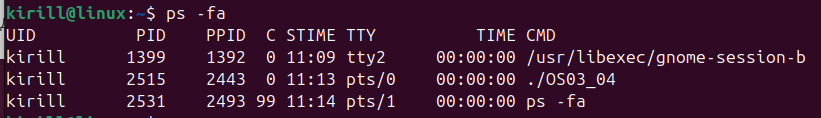
10. Разработайте консольное Linux-приложение OS03\_04 на языке С, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификатора процесса.

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

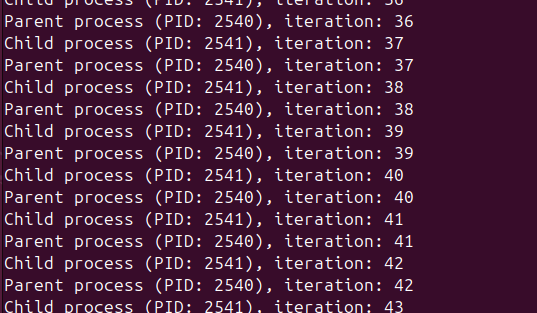
13. Разработайте консольное Linux-приложение OS03\_05 на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

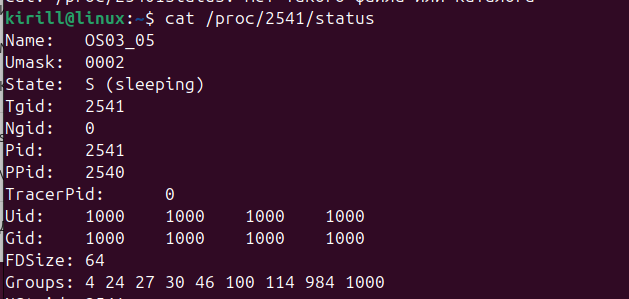
14. Приложение OS03\_05 должно создавать один дочерний процесс OS03\_05\_1 с помощью системного вызова fork. Процесс OS03\_05\_1 в этом случае не является отдельным модулем, а встроен (fork) в программный модуль OS03\_05.

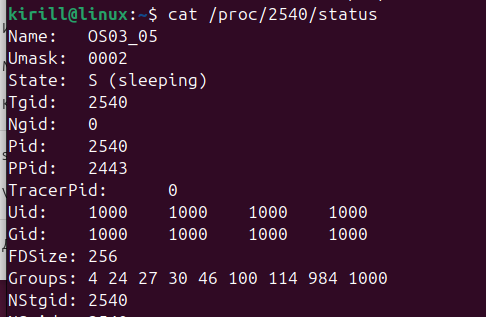
15. Процесс OS03\_05\_1 - консольное Linux-приложение, выполняющее цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

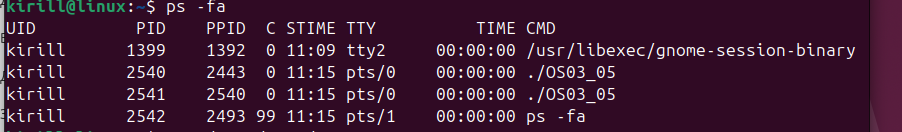
16. Продемонстрируйте информацию о процессах OS03\_05 и OS03\_05\_1 с помощью файловой системы /proc.

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









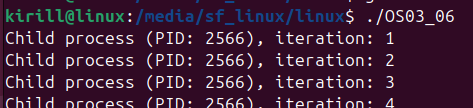
Задание 06

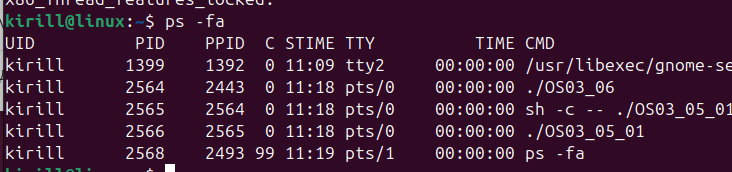
18. Разработайте консольное Linux-приложение OS03\_06 на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

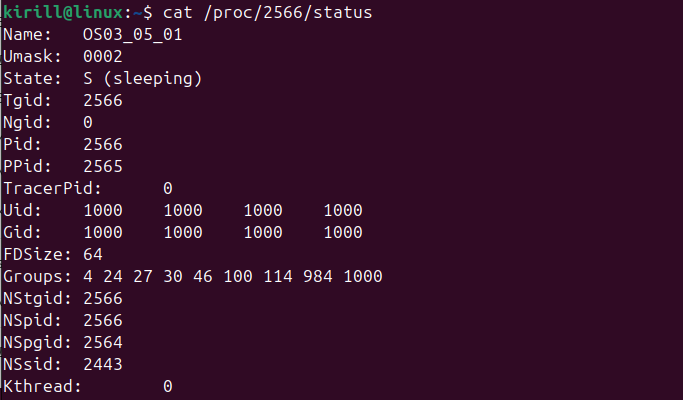
19. Приложение OS03\_06 должно создавать один дочерний процесс OS03\_05\_1 (отдельный модуль) с помощью системного вызова system.

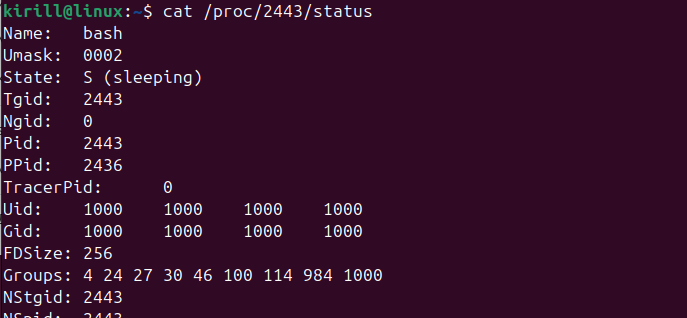
20. Продемонстрируйте информацию о процессах OS03\_06 и OS03\_05\_1 с помощью файловой системы /proc.

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









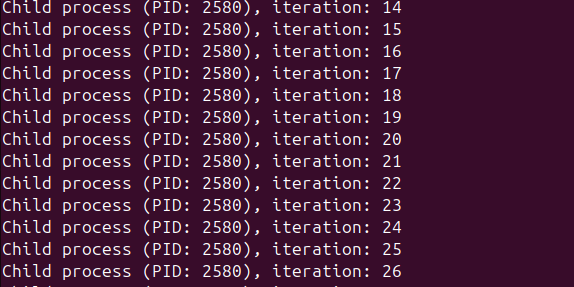
Задание 07

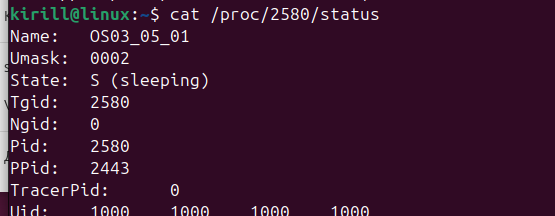
22. Разработайте консольное Linux-приложение OS03\_07 на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса

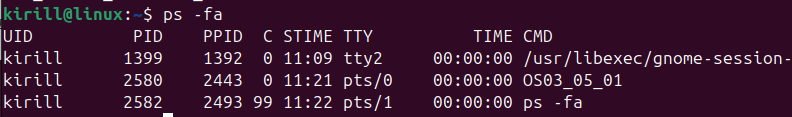
23. Приложение OS03\_07 должно создавать один дочерний процесс OS03\_05\_1 (отдельный модуль) с помощью системного вызова exec.

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 создаёт новый процесс, а exec заменяет текущий процесс.

 system возвращает управление после выполнения, а exec не возвращается.