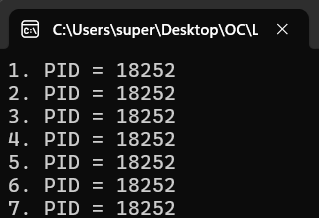
Лабораторная работа 03

OC

**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** в с помощью утилит **Task Manager**, **tasklist,**  **PowerShell ISE** и **Performance Monitor**.

****

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Изображение выглядит как текст, Шрифт, программное обеспечение, снимок экрана

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание**

**Задание 02**

1. 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.
2. Приложение **OS03\_02** должно создавать два дочерних процесса **OS03\_02\_1** и **OS03\_02\_2.**
3. Процесс **OS03\_02\_1** - консольное Windows-приложение выполняющее цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
4. Процесс **OS03\_02\_2** - консольное Windows-приложение выполняющее цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
5. Продемонстрируйте информацию о процессах **OS03\_02, OS03\_02\_1** и **OS03\_02\_2** в с помощью утилит **Task Manager**, **tasklist,**  **PowerShell ISE** и **Performance Monitor**.

Изображение выглядит как текст, снимок экрана, Шрифт, меню

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

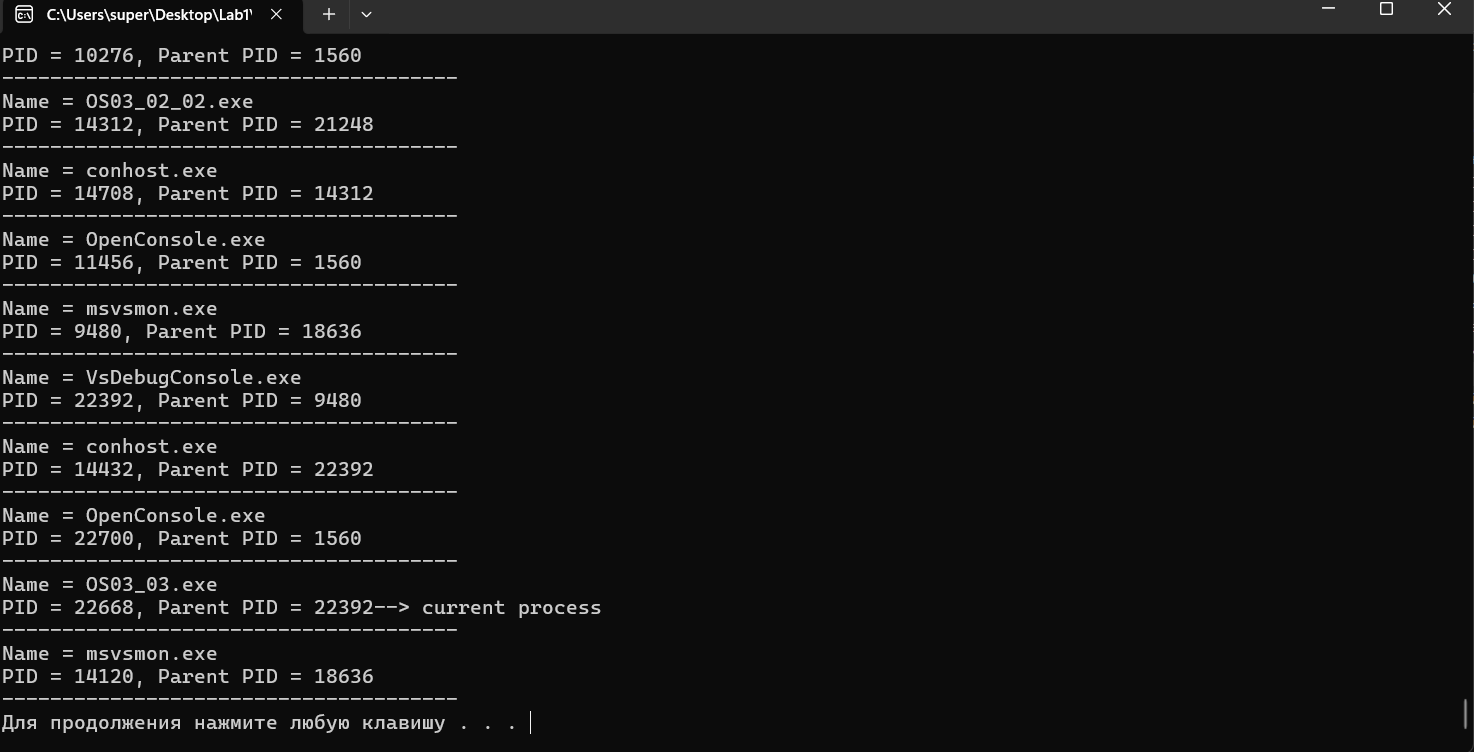
**Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание**

**Задание 03.**

1. Разработайте консольное Windows-приложение **OS03\_03** на языке С++, выводящее на консоль перечень выполняющихся процессов в данный момент в OS.
2. 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**.



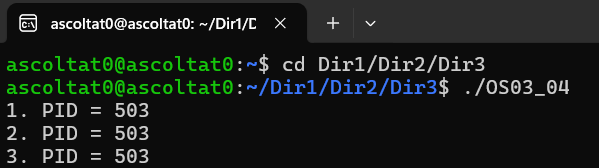


**Задание 04**

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

|  |
| --- |
| 1. #include <stdio.h> 2. #include <unistd.h> 3. int main() { 4. pid\_t pid = getpid(); 5. for (int i = 1; i <= 5000; ++i) { 6. printf("%d. PID = %d\n", i, pid); 7. sleep(1); 8. } 9. exit(0); 10. } |

|  |
| --- |
| gcc -o OS03\_04 OS03\_04.c  ./OS03\_04 |



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

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

1. Продемонстрируйте информацию о процессе **OS03\_04** с помощью утилиты **ps**.

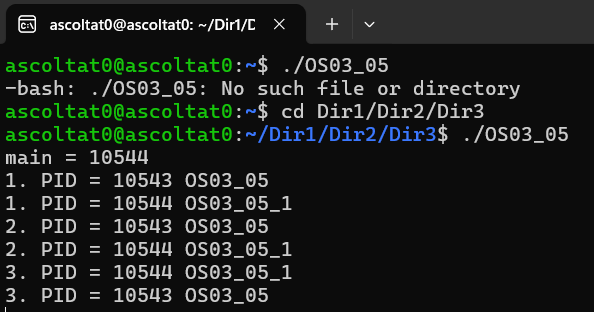
****

PID, использование процессора и памяти, время запуска и путь к исполняемому файлу

**Задание 05**

1. 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.
2. Приложение **OS03\_05** должно создавать один дочерний процесс **OS03\_05\_1** с помощью системного вызова **fork.** Процесс **OS03\_05\_1** в этом случае неявляется отдельным модулем, а встроен (fork) в программный модуль **OS03\_05**.

|  |
| --- |
| 1. #include <stdio.h> 2. #include <stdlib.h> 3. #include <errno.h> 4. #include <unistd.h> 5. #include <sys/types.h> 6. #include <sys/wait.h> 7. void cycle(int counter, char\* message) { 8. for (int i = 1; i <= counter; i++) { 9. printf("%d. PID = %d %s\n", i, getpid(), message); 10. sleep(2); 11. } 12. } 13. int main() { 14. pid\_t pid; 15. switch (pid = fork()) { 16. case -1: perror("fork error"); 17. exit(-1); 18. case 0: cycle(50, "OS03\_05\_1"); 19. exit(0); 20. default: printf("main = %d\n", pid); 21. cycle(100, "OS03\_05"); 22. wait(0); 23. } 24. exit(0); 25. } |

****

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

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

|  |
| --- |
| 1. #include <stdio.h> 2. #include <stdlib.h> 3. #include <errno.h> 4. #include <unistd.h> 5. #include <sys/types.h> 6. #include <sys/wait.h> 7. int main() { 8. for (int i = 1; i <= 50; ++i) { 9. printf("%d. PID = %d OS03\_05\_1\n", i, getpid()); 10. sleep(1); 11. } 12. exit(0); 13. } |

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

**Задание 06**

1. 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.
2. Приложение **OS03\_06** должно создавать один дочерний процесс **OS03\_05\_1** (отдельный модуль)с помощью системного вызова **system.**

|  |
| --- |
| 1. #include <stdio.h> 2. #include <stdlib.h> 3. #include <errno.h> 4. #include <unistd.h> 5. #include <sys/types.h> 6. #include <sys/wait.h> 7. int main() { 8. system("./OS03\_05\_01"); 9. for (int i = 1; i <= 100; i++) { 10. printf("%d. PID = %d OS03\_06\n", i, getpid()); 11. sleep(1); 12. } 13. exit(0); 14. } |

Изображение выглядит как текст, Шрифт, снимок экрана, типография

Автоматически созданное описание

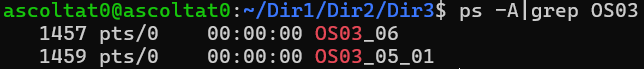
1. Продемонстрируйте информацию о процессах **OS03\_06** и **OS03\_05\_1** с помощью файловой системы **/proc**.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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



**Задание 07**

1. 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
2. Приложение **OS03\_07** должно создавать один дочерний процесс **OS03\_05\_1** (отдельный модуль)с помощью системного вызова **exec.**

|  |
| --- |
| 1. #include <stdio.h> 2. #include <stdlib.h> 3. #include <errno.h> 4. #include <unistd.h> 5. #include <sys/types.h> 6. #include <sys/wait.h> 7. int main() { 8. char\* const argv[] = { "ps", "-ef", 0 }; 9. execv("./OS03\_05\_01", argv); 10. for (int i = 1; i <= 100; ++i) { 11. printf("%d. PID = %d OS03\_07\n", i, getpid()); 12. sleep(1); 13. } 14. exit(0); 15. } |

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание

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

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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



1. Продемонстрируйте разницу системных вызовов **system** и **exec**.

**Задание 08.ответьте на следующие вопросы**

1. Что такое процесс?
2. Что такое контекст процесса?
3. Что такое родительский и дочерний процесс?
4. Что такое процесс инициализации OS?
5. Перечислите области памяти процесса и поясните их назначение.
6. Чем отличаются системные процессы от пользовательских?
7. Что такое Windows-сервисы, Linux-демоны?
8. С помощью каких системных вызовов можно создать дочерний процесс в Windows? Поясните разницу.
9. С помощью каких системных вызовов можно создать дочерний процесс в Linux? Поясните разницу.
10. Какие потоки данных доступны любому процессу автоматически?
11. Поясните назначение системного вызова WaitForSingleObject в Windows-приложении.
12. Поясните назначение системного вызова wait в Linux-приложении.
13. Дайте развернутое определение процесса OS.