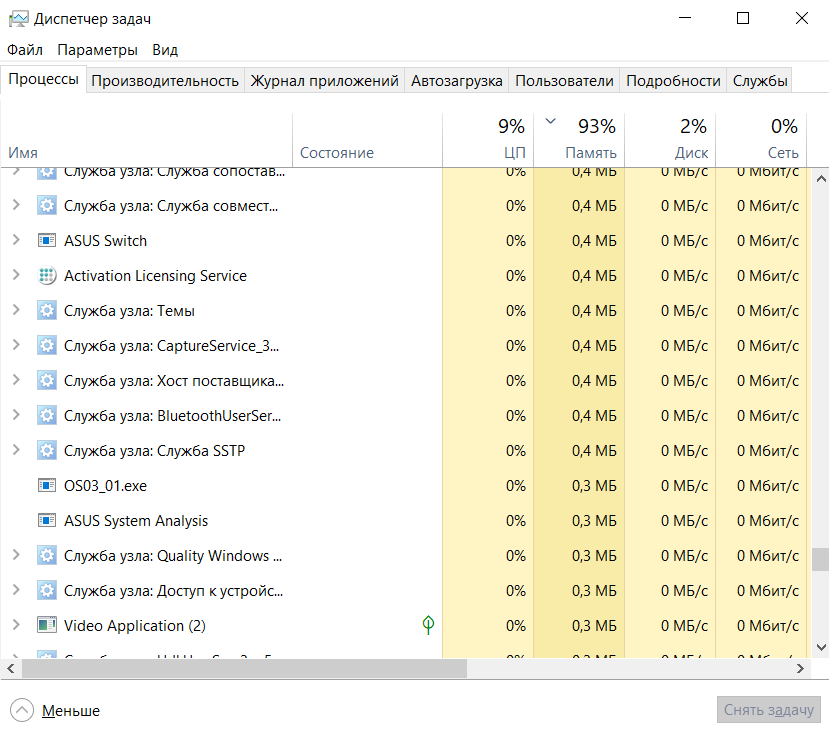
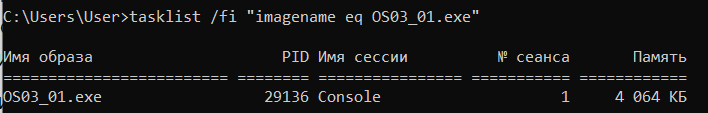
Задание 01

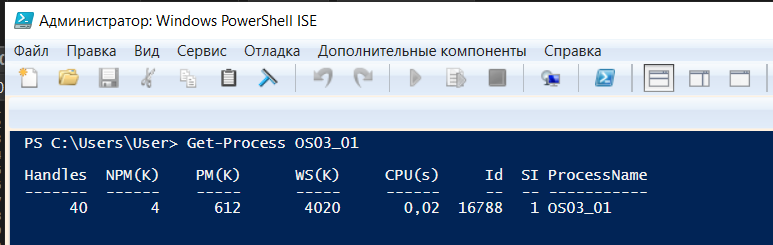
1. Разработайте консольное Windows-приложение OS03\_01 на языке С++, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификатора процесса.

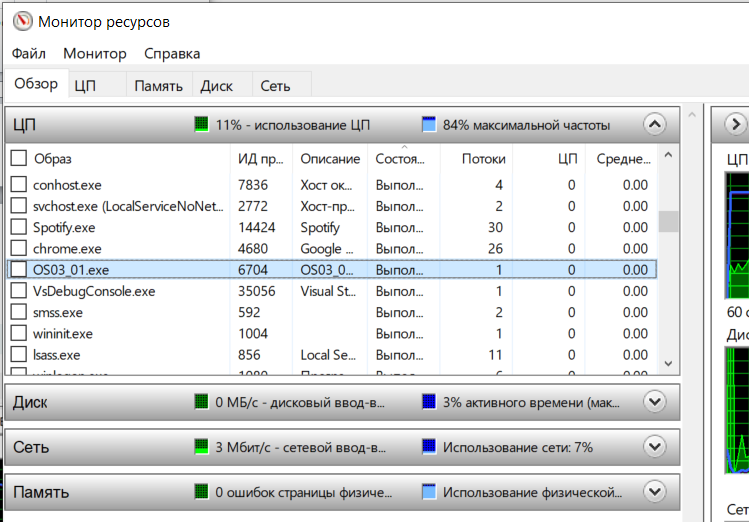
2. Продемонстрируйте информацию о процессе OS03\_01 с помощью утилит Task Manager, tasklist, PowerShell ISE и Performance Monitor.

|  |
| --- |
| #include <iostream>  #include <windows.h> //работа с ОС (ф-ция Sleep)  #include <process.h> //работа с процессами  int main()  {  for (int i = 0; i < 100; i++) {  Sleep(1000);  std::cout << \_getpid() << "\t";  }  } |



tasklist

PowerShell ISE

Win+R -> perfmon

Задание 02

1. 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.
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.

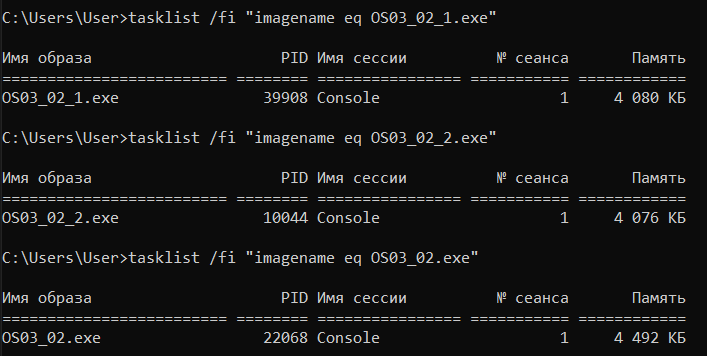
|  |
| --- |
| #include <iostream>  #include <Windows.h>  #include <process.h>  int main()  {  for (int i = 0; i < 50; i++) {  Sleep(1000);  std::cout << "OS03\_02\_1: " << \_getpid() << "\n";  }  return 0;  } |

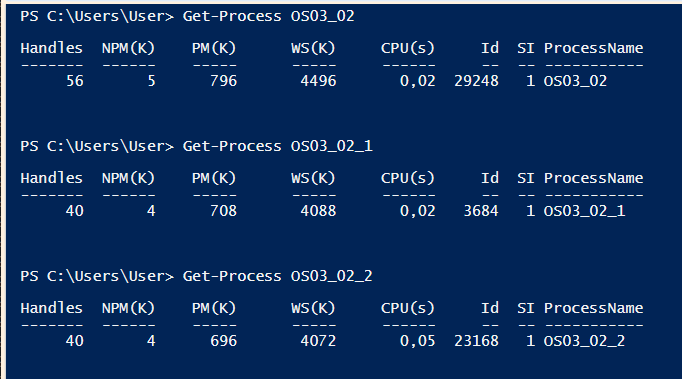
1. Процесс OS03\_02\_2 - консольное Windows-приложение выполняющее цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

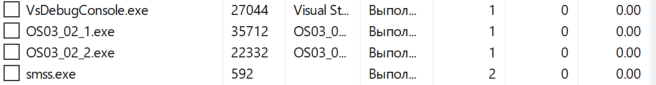
|  |
| --- |
| #include <iostream>  #include <Windows.h>  #include <process.h>  int main()  {  for (int i = 0; i < 100; i++) {  Sleep(1000);  std::cout << "OS03\_02\_2: " << \_getpid() << "\n";  }  return 0;  } |

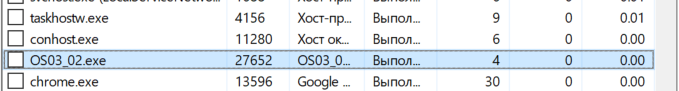
1. Продемонстрируйте информацию о процессах OS03\_02, OS03\_02\_1 и OS03\_02\_2 с помощью утилит Task Manager, tasklist, PowerShell ISE и Performance Monitor.





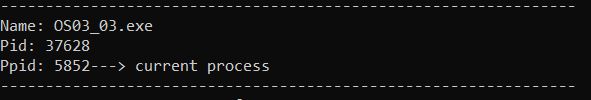
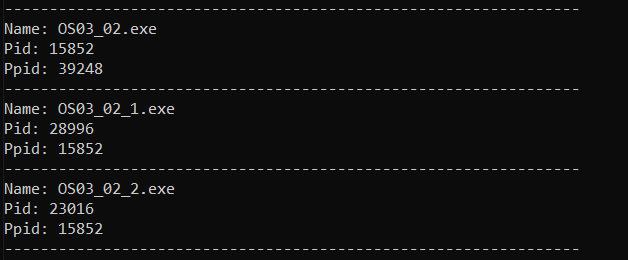






Задание 03.

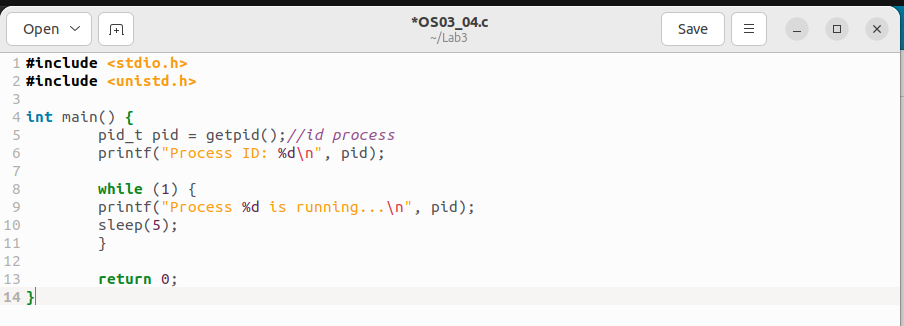
1. Разработайте консольное Windows-приложение OS03\_03 на языке С++, выводящее на консоль перечень выполняющихся процессов в данный момент в OS.
2. Запустите приложение OS03\_02 и продемонстрируйте с помощью приложения OS03\_03 в перечне процессов OS03\_02, OS03\_02\_1, OS03\_02\_2 и OS03\_03.



Задание 04

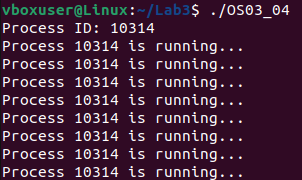
1. Разработайте консольное Linux-приложение OS03\_04 на языке С, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификатора процесса.
2. Продемонстрируйте информацию о процессе OS03\_04 с помощью файловой системы /proc.
3. Продемонстрируйте информацию о процессе OS03\_04 с помощью утилиты ps.

Создали файл OS03\_04.c с кодом:



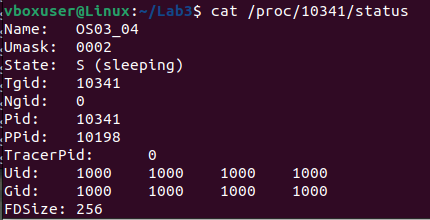
Компиляция приложения: gcc OS03\_04.c -o OS03\_04

В терминале запускаем файл - ./OS03\_04

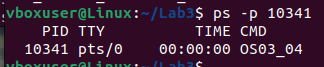


Открываем другой терминал с помощью ctrl+alt+t и пишем cat /proc/<PID>/status

Вместо <PID> пишем сам pid.



ps –p <PID>



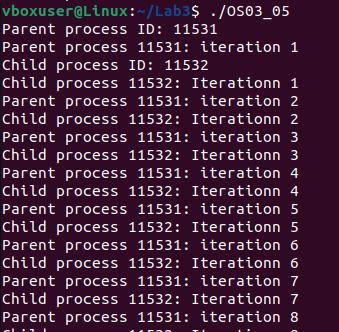
Задание 05

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 не является отдельным модулем, а встроен (fork) в программный модуль OS03\_05.
3. Процесс OS03\_05\_1 - консольное Linux-приложение, выполняющее цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
4. Продемонстрируйте информацию о процессах OS03\_05 и OS03\_05\_1 с помощью файловой системы /proc.
5. Продемонстрируйте информацию о процессах OS03\_05 и OS03\_05\_1 с помощью утилиты ps.

Создали файл OS03\_05.c с кодом:

****

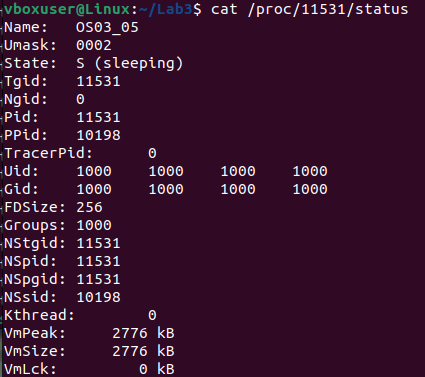
В терминале запускаем файл - ./OS03\_05



Открываем другой терминал с помощью ctrl+alt+t и пишем cat /proc/<PID>/status

Вместо <PID> пишем сам pid.

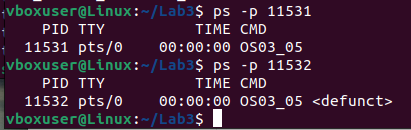
Родительский:



Дочерний:



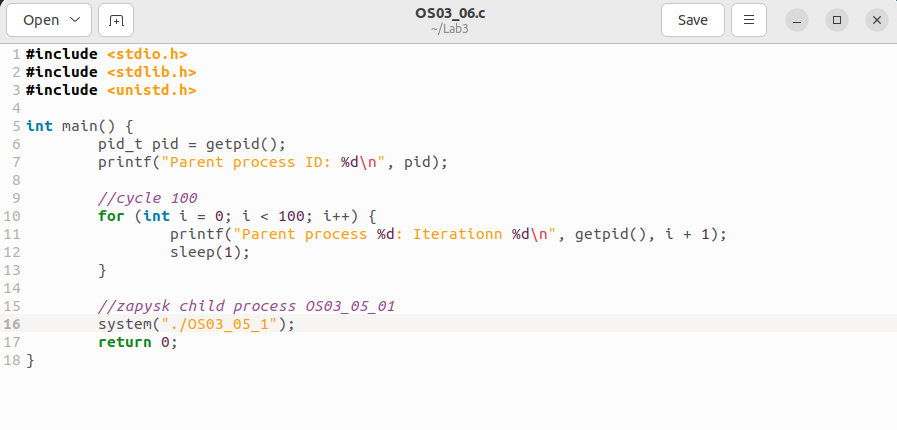
ps –p <PID>



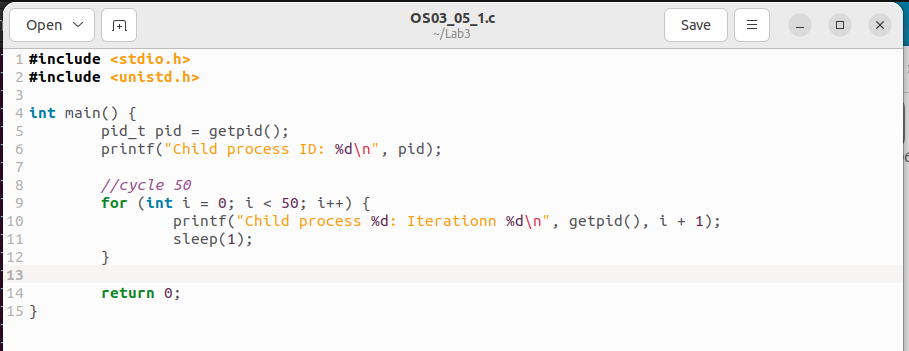
Задание 06

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.
3. Продемонстрируйте информацию о процессах OS03\_06 и OS03\_05\_1 с помощью файловой системы /proc.
4. Продемонстрируйте информацию о процессах OS03\_06 и OS03\_05-1 с помощью утилиты ps.

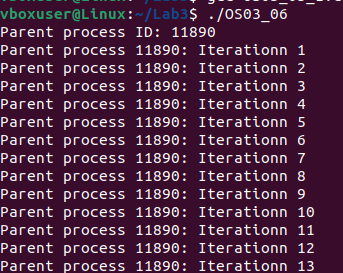
Создали файл OS03\_06.c с кодом:



И создали файл OS03\_05\_1.c с кодом:

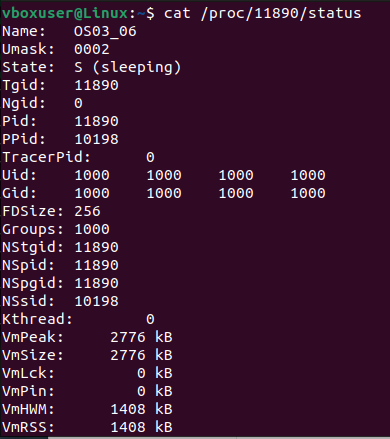


В терминале запускаем файл - ./OS03\_06

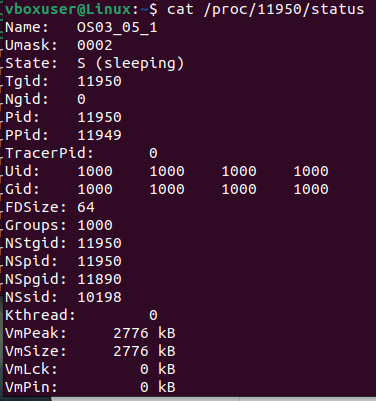


Открываем другой терминал с помощью ctrl+alt+t и пишем cat /proc/<PID>/status

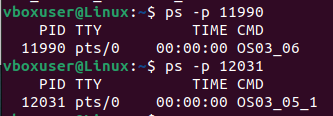
Вместо <PID> пишем сам pid.



Дочерний:



ps –p <PID>



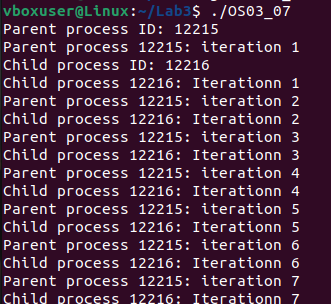
Задание 07

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.
3. Продемонстрируйте информацию о процессах OS03\_07 и OS03\_05\_1 с помощью файловой системы /proc.
4. Продемонстрируйте информацию о процессах OS03\_07 и OS03\_05-1 с помощью утилиты ps.
5. Продемонстрируйте разницу системных вызовов system и exec.

Создали файл OS03\_07.c с кодом:



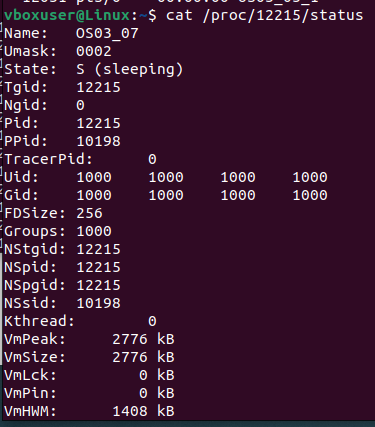
В терминале запускаем файл - ./OS03\_07



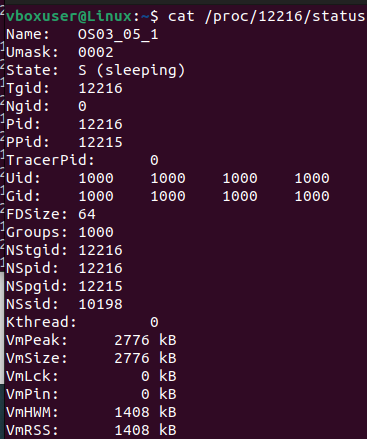
Открываем другой терминал с помощью ctrl+alt+t и пишем cat /proc/<PID>/status

Вместо <PID> пишем сам pid.

Родительский:



Дочерний:



ps –p <PID>

