

Правительство Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»  
(НИУ ВШЭ)

Московский институт электроники и математики им. А.Н. Тихонова

ОТЧЕТ  
по Лабораторной работе №8  
по дисциплине «Системное программирование»  
по теме «Процессы ОС»

Студент гр. БИБ201  
Морин Д.А.  
«10» июня 2023 г.

Руководитель  
Преподаватель  
Д.В. Смирнов  
«\_\_\_» \_\_\_\_\_ 2023 г.

Москва 2023

## **СОДЕРЖАНИЕ**

<b>1 ВВЕДЕНИЕ.....</b>	<b>3</b>
<b>2 ОСНОВНАЯ ЧАСТЬ.....</b>	<b>4</b>
<b>3 ЗАКЛЮЧЕНИЕ.....</b>	<b>7</b>
<b>ПРИЛОЖЕНИЕ А.....</b>	<b>8</b>

# **1 ВВЕДЕНИЕ**

В операционной системе Windows процессы играют важную роль. Процесс – это экземпляр выполняющейся программы, который обладает собственным адресным пространством, ресурсами и потоками выполнения. Процессы позволяют операционной системе управлять выполнением программ и обеспечивать изоляцию и безопасность между ними.

Каждый процесс в Windows имеет свой уникальный идентификатор (PID) и может взаимодействовать с другими процессами через механизмы IPC (Inter-Process Communication). Процессы могут запускать другие процессы, называемые дочерними процессами, создавая иерархию выполнения программ.

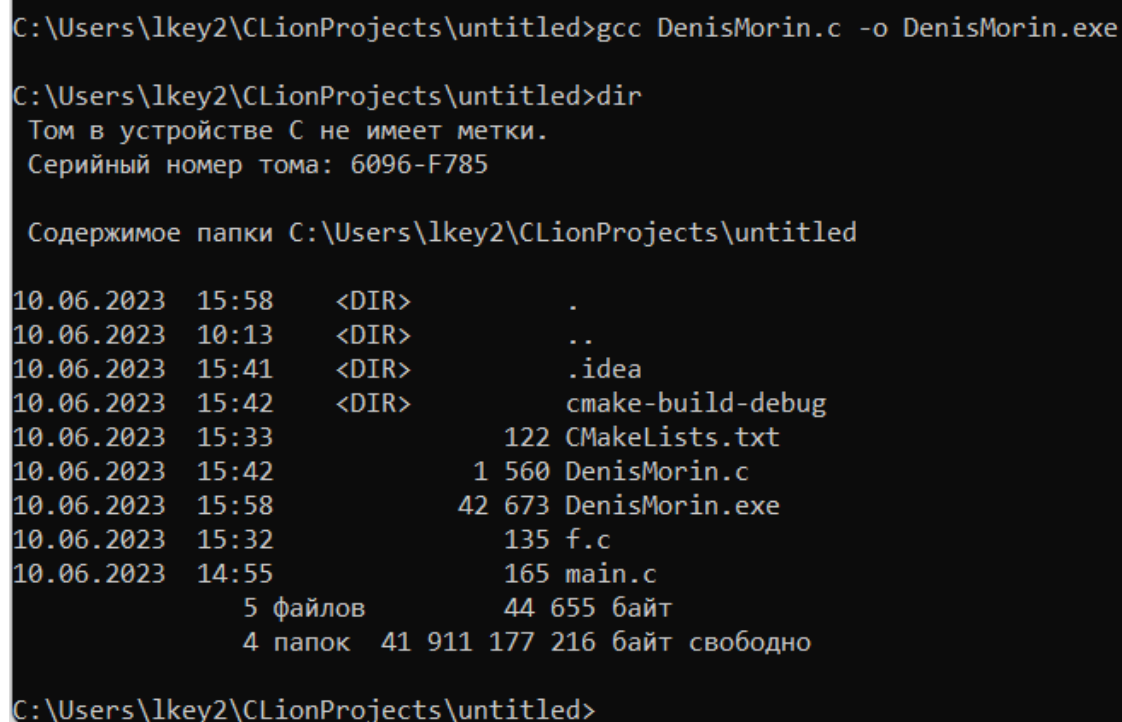
Windows предоставляет различные функции и API для управления процессами. Вы можете создавать новые процессы, получать информацию о запущенных процессах, управлять их жизненным циклом (запуск, приостановка, возобновление, завершение) и взаимодействовать с ними.

В данной практической работе будет написана программа на языке программирования Си, которая будет выводить список всех процессов с их PID, показывать свой процесс.

## 2 ОСНОВНАЯ ЧАСТЬ

По заданию для данной практической работы была написана программа на языке программирования Си (Приложение А).

Следующим этапом была компиляция в исполняемый файл (Рисунок 1).



```
C:\Users\lkey2\CLionProjects\untitled>gcc DenisMorin.c -o DenisMorin.exe

C:\Users\lkey2\CLionProjects\untitled>dir
Том в устройстве C не имеет метки.
Серийный номер тома: 6096-F785

Содержимое папки C:\Users\lkey2\CLionProjects\untitled

10.06.2023  15:58    <DIR>        .
10.06.2023  10:13    <DIR>        ..
10.06.2023  15:41    <DIR>        .idea
10.06.2023  15:42    <DIR>        cmake-build-debug
10.06.2023  15:33                122 CMakeLists.txt
10.06.2023  15:42                1 560 DenisMorin.c
10.06.2023  15:58                42 673 DenisMorin.exe
10.06.2023  15:32                135 f.c
10.06.2023  14:55                165 main.c
               5 файлов                44 655 байт
               4 папок   41 911 177 216 байт свободно

C:\Users\lkey2\CLionProjects\untitled>
```

Рисунок 1 – Создание исполняемого файла “DenisMorin.exe”

Была запущена программа, которая выводит список всех процессов (Рисунок 2).

```

C:\Users\lkey2\CLionProjects\untitled>DenisMorin.exe
Current Process ID: 7952
Process List:
0 | [System Process]
4 | System
104 | Secure System
168 | Registry
528 | smss.exe
784 | csrss.exe
948 | wininit.exe
968 | csrss.exe
600 | winlogon.exe
612 | services.exe
972 | LsaIso.exe

```

Рисунок 2 – Запуск “DenisMorin.exe”

Последним процессом оказался сам экземпляр выполняемой программы (Рисунок 3).

```

11884 | msedge.exe
5768 | msedge.exe
10608 | msedge.exe
12492 | msedge.exe
4232 | WmiPrvSE.exe
13068 | msedge.exe
7604 | ProcessHacker.exe
7952 | DenisMorin.exe          <---          This process
Для продолжения нажмите любую клавишу . . .

```

Рисунок 3 – Показ процесса запускаемого файла

Чтобы убедиться в том, что выводится идентификатор именно этого процесса, с помощью “Process Hacker” проверим это (Рисунок 4).

explorer.exe	5348	0,04	151,45 MB	LAPTOPDENIS\Ikey2	Проводник	
msedge.exe	7252	0,19 295,82 kB/s	152,52 MB	LAPTOPDENIS\Ikey2	Microsoft Edge	
SecurityHealthSystray.exe	9488		1,84 MB	LAPTOPDENIS\Ikey2	Windows Security notification ...	
Telegram.exe	2344		402,6 MB	LAPTOPDENIS\Ikey2	Telegram Desktop	
cmd.exe	132		3,43 MB	LAPTOPDENIS\Ikey2	Обработчик команд Windows	
conhost.exe	11168		6,59 MB	LAPTOPDENIS\Ikey2	Хост окна консоли	
DenisMorin.exe	7952		772 kB	LAPTOPDENIS\Ikey2		
cmd.exe	12192		2,37 MB	LAPTOPDENIS\Ikey2	Windows Command Processor	
Lightshot.exe	C:\Windows\system32\cmd.exe /c pause			3,49 MB	LAPTOPDENIS\Ikey2	Lightshot
Discord.exe	File:			2,74 MB	LAPTOPDENIS\Ikey2	Discord
Discord.exe	C:\Windows\SysWOW64\cmd.exe			2,95 MB	LAPTOPDENIS\Ikey2	Discord
Discord.exe	Windows Command Processor 10.0.22000.1			2,61 MB	LAPTOPDENIS\Ikey2	Discord
Discord.exe	Microsoft Corporation			6,62 MB	LAPTOPDENIS\Ikey2	Discord

Рисунок 4 – Процессы в программе “Process Hacker”

По рисунку 4 видно, что идентификатор запущенного процесса равен 7952, как и в выводе программы “DenisMorin.exe” в консоли на рисунке 3.

Чтобы убедиться, что и другие идентификаторы тоже совпадают у результата написанной программы и “Process Hacker”, выведем процессы с идентификаторами процессов 0 и 4 (Рисунок 5).

Processes Services Network Disk						
Name	PID	CPU	I/O total r...	Private by...	User name	Description
System Idle Process	0	93,58		60 kB	NT AUTHO... \СИСТЕМА	
System	4	0,16	32 kB/s	56 kB	NT AUTHO... \СИСТЕМА	NT Kernel & System
smss.exe	528			1,07 MB		Диспетчер сеанса Windows
Memory Compression	3004			1,16 MB		
Interrupts		0,43		0		Interrupts and DPCs

Рисунок 5 – Процессы “System Idle Process” и “System”

### **3 ЗАКЛЮЧЕНИЕ**

Понимание процессов в Windows важно для разработчиков, чтобы создавать эффективные и надежные программы. Управление процессами, взаимодействие между ними и эффективное использование ресурсов позволяют создавать сложные и мощные приложения, которые эффективно выполняют свои задачи. В ходе данной работы было знакомство с процессами в операционной системе Windows.

# ПРИЛОЖЕНИЕ А

## Код программы DenisMorin.c

```
#include <stdio.h>
#include <windows.h>
#include <tlhelp32.h>

void GetProcessList(DWORD currentPID) {

    // Take a snapshot of all processes in the system.
    HANDLE hProcessSnap = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);
    if (hProcessSnap == INVALID_HANDLE_VALUE) {
        printf("%s", "Failed to get process snapshot.");
    }

    PROCESSENTRY32 pe32;

    // Set the size of the structure before using it pe32.dwSize =
    sizeof(PROCESSENTRY32);

    // Retrieve information about the first process,
    // and exit if unsuccessful
    if (!Process32First(hProcessSnap, &pe32)) {
        printf("%s", "Failed to get first process entry.");
        CloseHandle(hProcessSnap);
    }

    // Now walk the snapshot of processes, and
    // display information about each process in turn
    printf("%s", "\nProcess List:\n");
    do {
        printf("%d%s", pe32.th32ProcessID, " | ", pe32.szExeFile);
        if (pe32.th32ProcessID == currentPID) {
            printf("%s", " <--- This process");
        }
        else{
            printf("%s", "\n");
        }
    }
    while (Process32Next(hProcessSnap, &pe32));

    CloseHandle(hProcessSnap);
}

int main() {

    // Gets current process PID
    DWORD currentPID = GetCurrentProcessId();
    printf("%s%d", "Current Process ID: ", currentPID);

    // Shows process list
    GetProcessList(currentPID);

    // Suspend till user inputs
    printf("%s", "\n");
    system("pause");
}
```



```
    return 0;  
}
```