

Московский Авиационный Институт  
(Национальный Исследовательский Университет)  
Институт №8 “Компьютерные науки и прикладная математика”  
Кафедра №806 “Вычислительная математика и программирование”

**Лабораторная работа №4 по курсу**  
**«Операционные системы»**

Группа: М8О-215Б-23

Студент: Моисеев К. В.

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_

Дата: 28.12.24

Москва, 2024

## Постановка задачи

### Вариант 22.

Требуется создать динамические библиотеки, которые реализуют заданный вариантом функционал. Далее использовать данные библиотеки 2-мя способами:

1. Во время компиляции (на этапе «линковки»/linking)
2. Во время исполнения программы. Библиотеки загружаются в память с помощью интерфейса ОС для работы с динамическими библиотеками

В конечном итоге, в лабораторной работе необходимо получить следующие части:

- ☐ Динамические библиотеки, реализующие контракты, которые заданы вариантом;
- ☐ Тестовая программа (*программа №1*), которая использует одну из библиотек, используя информацию, полученную на этапе компиляции;
- ☐ Тестовая программа (*программа №2*), которая загружает библиотеки, используя только их относительные пути и контракты.

Провести анализ двух типов использования библиотек.

Описание	Сигнатура	Реализация 1	Реализация 2
Подсчёт наибольшего общего делителя для двух натуральных чисел	Int GCF(int A, int B)	Алгоритм Евклида	Наивный алгоритм. Пытаться разделить числа на все числа, что меньше A и B.
Расчет значения числа Пи при заданной длине ряда (K)	float Pi(int K)	Ряд Лейбница	Формула Валлиса

# Общий метод и алгоритм решения

## Использованные системные вызовы

Для выполнения лабораторной работы использованы системные вызовы, предоставляемые Windows API:

- **LoadLibrary(LPCSTR lpLibFileName)** – загружает динамическую библиотеку в память, возвращая дескриптор модуля.
- **FreeLibrary(HMODULE hLibModule)** – освобождает ресурсы, связанные с загруженной библиотекой.
- **GetProcAddress(HMODULE hModule, LPCSTR lpProcName)** – получает адрес экспортированной функции из загруженной динамической библиотеки.

## Алгоритм решения

### 1. Функции библиотек

Для выполнения заданий реализованы функции с двумя различными подходами для каждого задания:

#### 1. Функция вычисления наибольшего общего делителя (НОД):

- **Алгоритм Евклида:** Использует итеративный подход для вычисления НОД, основываясь на свойствах деления с остатком.
- **Наивный алгоритм:** Последовательно проверяет все числа от 1 до минимального из двух чисел, чтобы найти наибольший общий делитель.

#### 2. Функция вычисления числа $\pi$ :

- **Ряд Лейбница:** Суммирует последовательность дробей с чередующимся знаком. Формула:  $\pi = 4 \sum_{k=0}^{\infty} (-1)^k \frac{1}{2k+1}$
  - **Формула Валлиса:** Использует бесконечное произведение дробей. Формула:  $\pi = 2 \prod_{k=1}^{\infty} \frac{2k \cdot 2k}{(2k-1) \cdot (2k+1)}$
- Из-за медленной сходимости обе формулы требуют большого числа итераций для достижения точности.

## 2. Программа 1

Первая тестовая программа (Program 1) использует статическую привязку библиотек на этапе компиляции.

Алгоритм:

1. Заголовочные файлы функций подключаются в коде до `main()`.
2. Пользователь вводит команды в формате:
  - **1 arg1 arg2** – для вычисления НОД.
  - **2 arg1** – для вычисления числа  $\pi$ .
3. В зависимости от команды, вызывается соответствующая функция.
4. Цикл завершается при вводе EOF.

### 3. Программа 2

Вторая тестовая программа (Program 2) использует динамическую загрузку библиотек в процессе выполнения.

Алгоритм:

1. С помощью `LoadLibrary` загружаются первоначальные версии библиотек (например, для Евклида и Лейбница).
2. Получаются адреса функций с использованием `GetProcAddress`.
3. Пользователь вводит команды в формате:

- **0** – переключение реализаций функций:
  - Загружаются альтернативные библиотеки (например, для Наивного алгоритма и Валлиса).
  - Старые библиотеки освобождаются с помощью FreeLibrary.
- **1 arg1 arg2** – для вызова функции НОД.
- **2 arg1** – для вызова функции числа  $\pi$ .

4. Цикл продолжается, пока не введена команда EOF.

#### 4. Makefile

Makefile автоматизирует сборку программы, описывая зависимости, цели и команды компиляции.

Алгоритм:

1. Для каждой библиотеки создаются объектные файлы с флагом `-shared` для генерации динамических библиотек (`.dll`).
2. Для Program 1 линковка происходит на этапе компиляции с использованием библиотеки.
3. Для Program 2 создается исполняемый файл без статической привязки библиотек.

#### Особенности реализации

- **Program 1:** Упрощенный подход, где использование функций осуществляется через заранее скомпилированные библиотеки.
- **Program 2:** Акцент на использование системных вызовов Windows API для динамической работы с библиотеками.
- **Makefile:** Полностью автоматизирует процесс сборки, минимизируя вероятность ошибок ручной компиляции.

```
# Компилятор и флаги
CXX = gcc
CXXFLAGS = -Wall -Wextra -O2 -Iinclude

# Библиотеки
LIBS = libGCF_Euclid.dll libGCF_Naive.dll libPi_Leibniz.dll libPi_Wallis.dll

# Тестовые программы
APP1 = TestProgram1.exe
APP2 = TestProgram2.exe

# Исходные файлы для библиотек
GCF_EUCLID_SRC = src/GCF_Euclid.c
GCF_NAIVE_SRC = src/GCF_Naive.c
PI_LEIBNIZ_SRC = src/Pi_Leibniz.c
PI_WALLIS_SRC = src/Pi_Wallis.c

# Исходные файлы для тестовых программ
TEST1_SRC = test/TestProgram1.c
TEST2_SRC = test/TestProgram2.c

# Цели
all: $(LIBS) $(APP1) $(APP2)

# Правила для создания библиотек
libGCF_Euclid.dll: $(GCF_EUCLID_SRC)
    $(CXX) $(CXXFLAGS) -shared -o $@ $<

libGCF_Naive.dll: $(GCF_NAIVE_SRC)
```

```

$(CXX) $(CXXFLAGS) -shared -o $@ $<

libPi_Leibniz.dll: $(PI_LEIBNIZ_SRC)
    $(CXX) $(CXXFLAGS) -shared -o $@ $<

libPi_Wallis.dll: $(PI_WALLIS_SRC)
    $(CXX) $(CXXFLAGS) -shared -o $@ $<

# Правила для создания тестовых программ
$(APP1): $(TEST1_SRC) libGCF_Euclid.dll libPi_Leibniz.dll
    $(CXX) $(CXXFLAGS) -o $@ $< -L. -lGCF_Euclid -lPi_Leibniz

$(APP2): $(TEST2_SRC)
    $(CXX) $(CXXFLAGS) -o $@ $< -L.

# Удаление сборок
clean:
    del $(LIBS) $(APP1) $(APP2)

```

Описание используемых флагов компиляции и линковки:

Компилятор:

- CXX = gcc: Используется компилятор GCC для C.
- Флаги компиляции:
- -Wall -Wextra: Включают все стандартные и дополнительные предупреждения, помогая выявить потенциальные ошибки в коде.
- -O2: Включает уровень оптимизации O2 для повышения производительности создаваемых бинарных файлов.
- -Iinclude: Добавляет директорию "include" в список путей поиска заголовочных файлов.

Создание динамических библиотек:

- -shared: Указывает компилятору создать динамическую библиотеку (DLL).

**Линковка с библиотеками:**

- -L. -lGCF\_Euclid -lPi\_Leibniz: Указывает, что тестовая программа зависит от библиотек libGCF\_Euclid.dll и libPi\_Leibniz.dll, расположенных в текущей директории.

**Удаление сборок:**

- Команда "del" используется для удаления сгенерированных файлов (библиотек и исполняемых файлов).

## Код программы

### GCF.h

```

#ifndef GCF_H
#define GCF_H

#ifdef _WIN32
#define EXPORT __declspec(dllexport)
#else
#define EXPORT

```

```
#endif

EXPORT int GCF_Euclid(int A, int B);
EXPORT int GCF_Naive(int A, int B);

#endif // GCF_H
```

## Pi.h

```
#ifndef PI_H
#define PI_H

#ifdef _WIN32
#define EXPORT __declspec(dllexport)
#else
#define EXPORT
#endif

EXPORT float Pi_Leibniz(int K);
EXPORT float Pi_Wallis(int K);

#endif // PI_H
```

## GCF\_Euclid.c

```
#include "GCF.h"

int GCF_Euclid(int A, int B) {
    while (B != 0) {
        int temp = B;
        B = A % B;
        A = temp;
    }
    return A;
}
```

## GCF\_Naive.c

```
#include "GCF.h"

int GCF_Naive(int A, int B) {
    int gcf = 1;
    for (int i = 2; i <= (A < B ? A : B); ++i) {
        if (A % i == 0 && B % i == 0) {
            gcf = i;
        }
    }
}
```

```
    return gcf;
}
```

## Pi\_Leibniz.c

```
#include "Pi.h"

float Pi_Leibniz(int K) {
    float pi = 0.0f;
    int sign = 1;
    for (int i = 0; i < K; ++i) {
        pi += sign * 1.0f / (2 * i + 1);
        sign = -sign;
    }
    return pi * 4.0f;
}
```

## Pi\_Wallis.c

```
#include "Pi.h"

float Pi_Wallis(int K) {
    float pi = 1.0f;
    for (int i = 1; i <= K; ++i) {
        pi *= (2.0f * i) / (2 * i - 1);
        pi *= (2.0f * i) / (2 * i + 1);
    }
    return pi * 2.0f;
}
```

## TestProgram1.c

```
#include <stdio.h>
#include "GCF.h"
#include "Pi.h"

int main() {
    int cmd;
    printf("Enter the command (1 for GCF, 2 for Pi, EOF to exit): ");
    while (scanf("%d", &cmd) != EOF) {
        if (cmd == 1) {
            int a, b;
            printf("Enter two numbers: ");
            if (scanf("%d %d", &a, &b) != 2) {
                printf("Invalid input. Please enter two integers.\n");
                while (getchar() != '\n'); // Очистка буфера ввода
                continue;
            }
            printf("GCF (Euclid): %d\n", GCF_Euclid(a, b));
        }
    }
}
```

```

    } else if (cmd == 2) {
        int k;
        printf("Enter the number of terms for Pi (Leibniz): ");
        if (scanf("%d", &k) != 1 || k < 0) {
            printf("Invalid input. Please enter a non-negative integer.\n");
            continue;
        }
        printf("Pi (Leibniz): %.6f\n", Pi_Leibniz(k));
    } else {
        printf("Unknown command. Please enter 1 for GCF or 2 for Pi.\n");
    }
    printf("Enter the command (1 for GCF, 2 for Pi, EOF to exit): ");
}
return 0;
}

```

## TestProgram2.c

```

#include <stdio.h>
#include <stdlib.h>
#include <windows.h>

// Типы функций, загружаемых из динамических библиотек
typedef int (*GCF_FUNC)(int, int);
typedef float (*PI_FUNC)(int);

// Функция для загрузки библиотеки
void loadLibrary(HMODULE *lib, const char *name) {
    *lib = LoadLibrary(name);
    if (!*lib) {
        printf("Failed to load library: %s\n", name);
        exit(EXIT_FAILURE);
    }
}

int main() {
    HMODULE lib1 = NULL, lib2 = NULL;
    GCF_FUNC GCF = NULL;
    PI_FUNC Pi = NULL;

    // Изначальная загрузка библиотек
    loadLibrary(&lib1, "libGCF_Euclid.dll");
    loadLibrary(&lib2, "libPi_Leibniz.dll");

    GCF = (GCF_FUNC)(void*)GetProcAddress(lib1, "GCF_Euclid");
    if (!GCF) {
        printf("Failed to load function GCF_Euclid from libGCF_Euclid.dll\n");
        FreeLibrary(lib1);
        exit(EXIT_FAILURE);
    }

    Pi = (PI_FUNC)(void*)GetProcAddress(lib2, "Pi_Leibniz");
    if (!Pi) {
        printf("Failed to load function Pi_Leibniz from libPi_Leibniz.dll\n");
    }
}

```



```

    FreeLibrary(lib2);
    exit(EXIT_FAILURE);
}

int cmd;
printf("Enter the command (0 to switch libraries, 1 for GCF, 2 for Pi, EOF to exit):
");
while (scanf("%d", &cmd) != EOF) {
    if (cmd == 0) {
        // Переключение библиотек
        static int gcfFlag = 0, piFlag = 0;

        FreeLibrary(lib1);
        FreeLibrary(lib2);

        if (gcfFlag) {
            loadLibrary(&lib1, "libGCF_Naive.dll");
            GCF = (GCF_FUNC)(void*)GetProcAddress(lib1, "GCF_Naive");
        } else {
            loadLibrary(&lib1, "libGCF_Euclid.dll");
            GCF = (GCF_FUNC)(void*)GetProcAddress(lib1, "GCF_Euclid");
        }

        if (piFlag) {
            loadLibrary(&lib2, "libPi_Wallis.dll");
            Pi = (PI_FUNC)(void*)GetProcAddress(lib2, "Pi_Wallis");
        } else {
            loadLibrary(&lib2, "libPi_Leibniz.dll");
            Pi = (PI_FUNC)(void*)GetProcAddress(lib2, "Pi_Leibniz");
        }

        if (!GCF || !Pi) {
            printf("Failed to reload functions after switching libraries.\n");
            exit(EXIT_FAILURE);
        }

        gcfFlag = !gcfFlag;
        piFlag = !piFlag;

        printf("Switched to another library.\n");
    } else if (cmd == 1) {
        // Вычисление GCF
        int a, b;
        printf("Enter two numbers: ");
        if (scanf("%d %d", &a, &b) != 2) {
            printf("Invalid input. Please enter two integers.\n");
            while (getchar() != '\n');
            continue;
        }
        printf("GCF: %d\n", GCF(a, b));
    } else if (cmd == 2) {
        // Вычисление Pi
        int k;
        printf("Enter the number of terms for Pi: ");
    }
}

```

```

        if (scanf("%d", &k) != 1 || k < 0) {
            printf("Invalid input. Please enter a non-negative integer.\n");
            while (getchar() != '\n');
            continue;
        }
        printf("Pi: %.6f\n", Pi(k));
    } else {
        printf("Unknown command.\n");
    }
    printf("Enter the command (0 to switch libraries, 1 for GCF, 2 for Pi, EOF to
exit): ");
}

// Освобождение библиотек перед завершением
FreeLibrary(lib1);
FreeLibrary(lib2);
return 0;
}

```

## Протокол работы программы

### Тестирование

PS C:\Users\klosh\Desktop\OS4> .\TestProgram1.exe

Enter the command (1 for GCF, 2 for Pi, EOF to exit): 2

Enter the number of terms for Pi (Leibniz): 100000

Pi (Leibniz): 3.141586

Enter the command (1 for GCF, 2 for Pi, EOF to exit): 0

Unknown command. Please enter 1 for GCF or 2 for Pi.

PS C:\Users\klosh\Desktop\OS4> .\TestProgram2.exe

Enter the command (0 to switch libraries, 1 for GCF, 2 for Pi, EOF to exit): 0

Switched to another library.

Enter the command (0 to switch libraries, 1 for GCF, 2 for Pi, EOF to exit): 2

Enter the number of terms for Pi: 10000

Pi: 3.141499

Enter the command (0 to switch libraries, 1 for GCF, 2 for Pi, EOF to exit): 1

Enter two numbers: 1260 3400

GCF: 20

### Strace

### Program1

PS C:\Users\klosh\Desktop\NtTrace-main\NtTrace-main> .\NtTrace -v -pre -filter NtMapViewOfSection,NtUnmapViewOfSection,NtOpenFile,NtCreateFile,NtReadFile C:\Users\klosh\Desktop\OS4\TestProgram1.exe

Process 2704 starting at 00007FF60E3613F0 with command line:  
"C:\Users\klosh\Desktop\OS4\TestProgram1.exe"

C:\Users\klosh\Desktop\OS4\TestProgram1.exe

Loaded DLL at 00007FF9624F0000 ntdll.dll

Instrumenting NtCreateFile at: 00007FF962590C60, ssn: 0x55

Instrumenting NtOpenFile at: 00007FF962590820, ssn: 0x33

Instrumenting NtReadFile at: 00007FF962590280, ssn: 0x6

Instrumenting NtReadFileScatter at: 00007FF962590780, ssn: 0x2e

Instrumenting NtMapViewOfSection at: 00007FF9625906C0, ssn: 0x28

Instrumenting NtMapViewOfSectionEx at: 00007FF962592530, ssn: 0x11c

Instrumenting NtUnmapViewOfSection at: 00007FF962590700, ssn: 0x2a

Instrumenting NtUnmapViewOfSectionEx at: 00007FF962593D30, ssn: 0x1dc

NtOpenFile(FileHandle=0x82751ff378, DesiredAccess=SYNCHRONIZE|0x20, ObjectAttributes="\??\C:\Users\klosh\Desktop\NtTrace-main\NtTrace-main\", IoStatusBlock=0x82751ff2e8, ShareAccess=3, OpenOptions=0x21) ...

NtOpenFile(FileHandle=0x82751ff378 [0x54], DesiredAccess=SYNCHRONIZE|0x20, ObjectAttributes="\??\C:\Users\klosh\Desktop\NtTrace-main\NtTrace-main\", IoStatusBlock=0x82751ff2e8 [0/1], ShareAccess=3, OpenOptions=0x21) => 0

NtMapViewOfSection(SectionHandle=0x58, ProcessHandle=-1, BaseAddress=0x22debc149f0 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x22debc14948 [0], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF9618F0000 C:\WINDOWS\System32\KERNEL32.DLL

NtMapViewOfSection(SectionHandle=0x58, ProcessHandle=-1, BaseAddress=0x22debc149f0 [0x00007ff9618f0000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x22debc14948 [0x000c4000], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

NtMapViewOfSection(SectionHandle=0x68, ProcessHandle=-1, BaseAddress=0x22debc15130 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x22debc15088 [0], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF95FBE0000 C:\WINDOWS\System32\KERNELBASE.dll

NtMapViewOfSection(SectionHandle=0x68, ProcessHandle=-1, BaseAddress=0x22debc15130 [0x00007ff95fbe0000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x22debc15088 [0x003b9000], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

NtMapViewOfSection(SectionHandle=0x58, ProcessHandle=-1, BaseAddress=0x82751fe8d8 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x82751fe8e8 [0], InheritDisposition=2 [ViewUnmap], AllocationType=0x00500000, Protect=2) ...

NtMapViewOfSection(SectionHandle=0x58, ProcessHandle=-1, BaseAddress=0x82751fe8d8 [0x00007ff448ca0000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x82751fe8e8 [0x00100000], InheritDisposition=2 [ViewUnmap], AllocationType=0x00500000, Protect=2) => 0

NtCreateFile(FileHandle=0x82751feb08, DesiredAccess=READ\_CONTROL|SYNCHRONIZE|0x19f, ObjectAttributes=4:"\Connect", IoStatusBlock=0x82751fe4c0, AllocationSize=null, FileAttributes=0, ShareAccess=7, CreateDisposition=2, CreateOptions=0x20, EaBuffer=0x22debc165d0, EaLength=0x54b) ...

NtCreateFile(FileHandle=0x82751feb08 [0x58], DesiredAccess=READ\_CONTROL|SYNCHRONIZE|0x19f, ObjectAttributes=4:"\Connect", IoStatusBlock=0x82751fe4c0 [0/0x18], AllocationSize=null, FileAttributes=0, ShareAccess=7, CreateDisposition=2, CreateOptions=0x20, EaBuffer=0x22debc165d0, EaLength=0x54b) => 0

NtMapViewOfSection(SectionHandle=0x94, ProcessHandle=-1, BaseAddress=0x82751fe870 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x82751fe880 [0], InheritDisposition=2 [ViewUnmap], AllocationType=0, Protect=2) ...

NtMapViewOfSection(SectionHandle=0x94, ProcessHandle=-1, BaseAddress=0x82751fe870 [0x0000022debde0000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x82751fe880 [0x3000], InheritDisposition=2 [ViewUnmap], AllocationType=0, Protect=2) => 0

NtMapViewOfSection(SectionHandle=0x98, ProcessHandle=-1, BaseAddress=0x82751fe870 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x82751fe880 [0], InheritDisposition=2 [ViewUnmap], AllocationType=0, Protect=2) ...

NtMapViewOfSection(SectionHandle=0x98, ProcessHandle=-1, BaseAddress=0x82751fe870 [0x0000022debdf0000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x82751fe880 [0x3000], InheritDisposition=2 [ViewUnmap], AllocationType=0, Protect=2) => 0

NtMapViewOfSection(SectionHandle=0x9c, ProcessHandle=-1, BaseAddress=0x82751fe870 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x82751fe880 [0], InheritDisposition=2 [ViewUnmap], AllocationType=0, Protect=2) ...

NtMapViewOfSection(SectionHandle=0x9c, ProcessHandle=-1, BaseAddress=0x82751fe870 [0x0000022debe00000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x82751fe880 [0x1000], InheritDisposition=2 [ViewUnmap], AllocationType=0, Protect=2) => 0

NtMapViewOfSection(SectionHandle=0xb4, ProcessHandle=-1, BaseAddress=0x22debc14da0 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x22debc14cd8 [0], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF95FFA0000 C:\WINDOWS\System32\ucrtbase.dll

NtMapViewOfSection(SectionHandle=0xb4, ProcessHandle=-1, BaseAddress=0x22debc14da0 [0x00007ff95ffa0000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x22debc14cd8 [0x00111000], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

NtOpenFile(FileHandle=0x82751fefe0, DesiredAccess=SYNCHRONIZE|0x21, ObjectAttributes="\??\C:\Users\klosh\Desktop\OS4\libGCF\_Euclid.dll", IoStatusBlock=0x82751ff048, ShareAccess=5, OpenOptions=0x60) ...

NtOpenFile(FileHandle=0x82751fefe0 [0x68], DesiredAccess=SYNCHRONIZE|0x21, ObjectAttributes="\??\C:\Users\klosh\Desktop\OS4\libGCF\_Euclid.dll", IoStatusBlock=0x82751ff048 [0/1], ShareAccess=5, OpenOptions=0x60) => 0

NtMapViewOfSection(SectionHandle=0xb8, ProcessHandle=-1, BaseAddress=0x22debc18220 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x22debc18178 [0], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF93DC00000  
C:\Users\klosh\Desktop\OS4\libGCF\_Euclid.dll

NtMapViewOfSection(SectionHandle=0xb8, ProcessHandle=-1, BaseAddress=0x22debc18220 [0x00007ff93dc00000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x22debc18178 [0x00015000], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

NtOpenFile(FileHandle=0x82751fefe0, DesiredAccess=SYNCHRONIZE|0x21, ObjectAttributes="\??\C:\Users\klosh\Desktop\OS4\libPi\_Leibniz.dll", IoStatusBlock=0x82751ff048, ShareAccess=5, OpenOptions=0x60) ...

NtOpenFile(FileHandle=0x82751fefe0 [0x68], DesiredAccess=SYNCHRONIZE|0x21, ObjectAttributes="\??\C:\Users\klosh\Desktop\OS4\libPi\_Leibniz.dll", IoStatusBlock=0x82751ff048 [0/1], ShareAccess=5, OpenOptions=0x60) => 0

NtMapViewOfSection(SectionHandle=0xb8, ProcessHandle=-1, BaseAddress=0x22debc18530 [0], ZeroBits=0, CommitSize=0, SectionOffset=null,

ViewSize=0x22debc18488 [0], InheritDisposition=1 [ViewShare],  
AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF93BF20000 C:\Users\klosh\Desktop\OS4\libPi\_Leibniz.dll

NtMapViewOfSection(SectionHandle=0xb8, ProcessHandle=-1,  
BaseAddress=0x22debc18530 [0x00007ff93bf20000], ZeroBits=0, CommitSize=0,  
SectionOffset=null, ViewSize=0x22debc18488 [0x00015000], InheritDisposition=1  
[ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

Initial breakpoint

Enter the command (1 for GCF, 2 for Pi, 3 to exit): NtReadFile(FileHandle=0x5c,  
Event=0, ApcRoutine=null, ApcContext=null, IoStatusBlock=0x82751ff980,  
Buffer=0x22debc1efe0, Length=0x1000, ByteOffset=null, Key=null) ...

1

NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null,  
IoStatusBlock=0x82751ff980 [0/3], Buffer=0x22debc1efe0, Length=0x1000,  
ByteOffset=null, Key=null) => 0

Enter two numbers: NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null,  
ApcContext=null, IoStatusBlock=0x82751ff980, Buffer=0x22debc1efe0,  
Length=0x1000, ByteOffset=null, Key=null) ...

16 24

NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null,  
IoStatusBlock=0x82751ff980 [0/7], Buffer=0x22debc1efe0, Length=0x1000,  
ByteOffset=null, Key=null) => 0

GCF (Euclid): 8

Enter the command (1 for GCF, 2 for Pi, 3 to exit): NtReadFile(FileHandle=0x5c,  
Event=0, ApcRoutine=null, ApcContext=null, IoStatusBlock=0x82751ff980,  
Buffer=0x22debc1efe0, Length=0x1000, ByteOffset=null, Key=null) ...

3

NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null,  
IoStatusBlock=0x82751ff980 [0/3], Buffer=0x22debc1efe0, Length=0x1000,  
ByteOffset=null, Key=null) => 0

NtOpenFile(FileHandle=0x82751ff710, DesiredAccess=SYNCHRONIZE|0x21,  
ObjectAttributes="\??\C:\WINDOWS\SYSTEM32\kernel.appcore.dll",  
IoStatusBlock=0x82751ff778, ShareAccess=5, OpenOptions=0x60) ...

NtOpenFile(FileHandle=0x82751ff710 [0xb8],  
DesiredAccess=SYNCHRONIZE|0x21,  
ObjectAttributes="\??\C:\WINDOWS\SYSTEM32\kernel.appcore.dll",  
IoStatusBlock=0x82751ff778 [0/1], ShareAccess=5, OpenOptions=0x60) => 0

NtMapViewOfSection(SectionHandle=0xbc, ProcessHandle=-1,  
BaseAddress=0x22debc20190 [0], ZeroBits=0, CommitSize=0, SectionOffset=null,  
ViewSize=0x22debc200b8 [0], InheritDisposition=1 [ViewShare],  
AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF95E8E0000 C:\WINDOWS\SYSTEM32\kernel.appcore.dll

NtMapViewOfSection(SectionHandle=0xbc, ProcessHandle=-1,  
BaseAddress=0x22debc20190 [0x00007ff95e8e0000], ZeroBits=0, CommitSize=0,  
SectionOffset=null, ViewSize=0x22debc200b8 [0x00018000], InheritDisposition=1  
[ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

NtMapViewOfSection(SectionHandle=0xc0, ProcessHandle=-1,  
BaseAddress=0x22debc20410 [0], ZeroBits=0, CommitSize=0, SectionOffset=null,  
ViewSize=0x22debc20378 [0], InheritDisposition=1 [ViewShare],  
AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF960210000 C:\WINDOWS\System32\msvcrt.dll

NtMapViewOfSection(SectionHandle=0xc0, ProcessHandle=-1,  
BaseAddress=0x22debc20410 [0x00007ff960210000], ZeroBits=0, CommitSize=0,  
SectionOffset=null, ViewSize=0x22debc20378 [0x000a7000], InheritDisposition=1  
[ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

Process 2704 exit code: 0

**Program2**

C:\Users\klosh\Desktop\OS4\TestProgram2.exe

Loaded DLL at 00007FF9624F0000 ntdll.dll

Instrumenting NtCreateFile at: 00007FF962590C60, ssn: 0x55

Instrumenting NtOpenFile at: 00007FF962590820, ssn: 0x33

Instrumenting NtReadFile at: 00007FF962590280, ssn: 0x6

Instrumenting NtReadFileScatter at: 00007FF962590780, ssn: 0x2e

Instrumenting NtMapViewOfSection at: 00007FF9625906C0, ssn: 0x28

Instrumenting NtMapViewOfSectionEx at: 00007FF962592530, ssn: 0x11c

Instrumenting NtUnmapViewOfSection at: 00007FF962590700, ssn: 0x2a

Instrumenting NtUnmapViewOfSectionEx at: 00007FF962593D30, ssn: 0x1dc

NtOpenFile(FileHandle=0x2841bff188, DesiredAccess=SYNCHRONIZE|0x20, ObjectAttributes="\??\C:\Users\klosh\Desktop\NtTrace-main\NtTrace-main\", IoStatusBlock=0x2841bff0f8, ShareAccess=3, OpenOptions=0x21) ...

NtOpenFile(FileHandle=0x2841bff188 [0x54], DesiredAccess=SYNCHRONIZE|0x20, ObjectAttributes="\??\C:\Users\klosh\Desktop\NtTrace-main\NtTrace-main\", IoStatusBlock=0x2841bff0f8 [0/1], ShareAccess=3, OpenOptions=0x21) => 0

NtMapViewOfSection(SectionHandle=0x58, ProcessHandle=-1, BaseAddress=0x208d16b49f0 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16b4948 [0], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF9618F0000 C:\WINDOWS\System32\KERNEL32.DLL

NtMapViewOfSection(SectionHandle=0x58, ProcessHandle=-1, BaseAddress=0x208d16b49f0 [0x00007ff9618f0000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16b4948 [0x000c4000], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

NtMapViewOfSection(SectionHandle=0x68, ProcessHandle=-1, BaseAddress=0x208d16b5130 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16b5088 [0], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF95FBE0000 C:\WINDOWS\System32\KERNELBASE.dll



NtMapViewOfSection(SectionHandle=0x68, ProcessHandle=-1,  
BaseAddress=0x208d16b5130 [0x00007ff95fbe0000], ZeroBits=0, CommitSize=0,  
SectionOffset=null, ViewSize=0x208d16b5088 [0x003b9000], InheritDisposition=1  
[ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

NtMapViewOfSection(SectionHandle=0x58, ProcessHandle=-1,  
BaseAddress=0x2841bfe6e8 [0], ZeroBits=0, CommitSize=0, SectionOffset=null,  
ViewSize=0x2841bfe6f8 [0], InheritDisposition=2 [ViewUnmap],  
AllocationType=0x00500000, Protect=2) ...

NtMapViewOfSection(SectionHandle=0x58, ProcessHandle=-1,  
BaseAddress=0x2841bfe6e8 [0x00007ff48ed60000], ZeroBits=0, CommitSize=0,  
SectionOffset=null, ViewSize=0x2841bfe6f8 [0x00100000], InheritDisposition=2  
[ViewUnmap], AllocationType=0x00500000, Protect=2) => 0

NtCreateFile(FileHandle=0x2841bfe918,  
DesiredAccess=READ\_CONTROL|SYNCHRONIZE|0x19f,  
ObjectAttributes=4:"\Connect", IoStatusBlock=0x2841bfe2d0, AllocationSize=null,  
FileAttributes=0, ShareAccess=7, CreateDisposition=2, CreateOptions=0x20,  
EaBuffer=0x208d16b65d0, EaLength=0x54b) ...

NtCreateFile(FileHandle=0x2841bfe918 [0x58],  
DesiredAccess=READ\_CONTROL|SYNCHRONIZE|0x19f,  
ObjectAttributes=4:"\Connect", IoStatusBlock=0x2841bfe2d0 [0/0x18],  
AllocationSize=null, FileAttributes=0, ShareAccess=7, CreateDisposition=2,  
CreateOptions=0x20, EaBuffer=0x208d16b65d0, EaLength=0x54b) => 0

NtMapViewOfSection(SectionHandle=0x90, ProcessHandle=-1,  
BaseAddress=0x2841bfe680 [0], ZeroBits=0, CommitSize=0, SectionOffset=null,  
ViewSize=0x2841bfe690 [0], InheritDisposition=2 [ViewUnmap], AllocationType=0,  
Protect=2) ...

NtMapViewOfSection(SectionHandle=0x90, ProcessHandle=-1,  
BaseAddress=0x2841bfe680 [0x00000208d1640000], ZeroBits=0, CommitSize=0,  
SectionOffset=null, ViewSize=0x2841bfe690 [0x3000], InheritDisposition=2  
[ViewUnmap], AllocationType=0, Protect=2) => 0

NtMapViewOfSection(SectionHandle=0x94, ProcessHandle=-1,  
BaseAddress=0x2841bfe680 [0], ZeroBits=0, CommitSize=0, SectionOffset=null,  
ViewSize=0x2841bfe690 [0], InheritDisposition=2 [ViewUnmap], AllocationType=0,  
Protect=2) ...

NtMapViewOfSection(SectionHandle=0x94, ProcessHandle=-1, BaseAddress=0x2841bfe680 [0x00000208d1650000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x2841bfe690 [0x3000], InheritDisposition=2 [ViewUnmap], AllocationType=0, Protect=2) => 0

NtMapViewOfSection(SectionHandle=0x98, ProcessHandle=-1, BaseAddress=0x2841bfe680 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x2841bfe690 [0], InheritDisposition=2 [ViewUnmap], AllocationType=0, Protect=2) ...

NtMapViewOfSection(SectionHandle=0x98, ProcessHandle=-1, BaseAddress=0x2841bfe680 [0x00000208d1660000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x2841bfe690 [0x1000], InheritDisposition=2 [ViewUnmap], AllocationType=0, Protect=2) => 0

NtMapViewOfSection(SectionHandle=0xb0, ProcessHandle=-1, BaseAddress=0x208d16b4da0 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16b4cd8 [0], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF95FFA0000 C:\WINDOWS\System32\ucrtbase.dll

NtMapViewOfSection(SectionHandle=0xb0, ProcessHandle=-1, BaseAddress=0x208d16b4da0 [0x00007ff95ffa0000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16b4cd8 [0x00111000], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

Initial breakpoint

NtOpenFile(FileHandle=0x2841bff430, DesiredAccess=SYNCHRONIZE|0x21, ObjectAttributes="\??\C:\Users\klosh\Desktop\OS4\libGCF\_Euclid.dll", IoStatusBlock=0x2841bff498, ShareAccess=5, OpenOptions=0x60) ...

NtOpenFile(FileHandle=0x2841bff430 [0xb8], DesiredAccess=SYNCHRONIZE|0x21, ObjectAttributes="\??\C:\Users\klosh\Desktop\OS4\libGCF\_Euclid.dll", IoStatusBlock=0x2841bff498 [0/1], ShareAccess=5, OpenOptions=0x60) => 0

NtMapViewOfSection(SectionHandle=0xbc, ProcessHandle=-1, BaseAddress=0x208d16bcfd0 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16bbe98 [0], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF93AC40000 C:\Users\klosh\Desktop\OS4\libGCF\_Euclid.dll

NtMapViewOfSection(SectionHandle=0xbc, ProcessHandle=-1, BaseAddress=0x208d16bcfd0 [0x00007ff93ac40000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16bbe98 [0x00015000], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

NtOpenFile(FileHandle=0x2841bff430, DesiredAccess=SYNCHRONIZE[0x21, ObjectAttributes="\??\C:\Users\klosh\Desktop\OS4\libPi\_Leibniz.dll", IoStatusBlock=0x2841bff498, ShareAccess=5, OpenOptions=0x60) ...

NtOpenFile(FileHandle=0x2841bff430 [0xc0], DesiredAccess=SYNCHRONIZE[0x21, ObjectAttributes="\??\C:\Users\klosh\Desktop\OS4\libPi\_Leibniz.dll", IoStatusBlock=0x2841bff498 [0/1], ShareAccess=5, OpenOptions=0x60) => 0

NtMapViewOfSection(SectionHandle=0xb8, ProcessHandle=-1, BaseAddress=0x208d16bd270 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16bbe98 [0], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF939380000 C:\Users\klosh\Desktop\OS4\libPi\_Leibniz.dll

NtMapViewOfSection(SectionHandle=0xb8, ProcessHandle=-1, BaseAddress=0x208d16bd270 [0x00007ff939380000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16bbe98 [0x00015000], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

Enter the command (0 to switch libraries, 1 for GCF, 2 for Pi, 3 to exit):

NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null, IoStatusBlock=0x2841bff770, Buffer=0x208d16befc0, Length=0x1000, ByteOffset=null, Key=null) ...

1

NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null, IoStatusBlock=0x2841bff770 [0/3], Buffer=0x208d16befc0, Length=0x1000, ByteOffset=null, Key=null) => 0

Enter two numbers: NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null, IoStatusBlock=0x2841bff770, Buffer=0x208d16befc0, Length=0x1000, ByteOffset=null, Key=null) ...

6 9

NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null, IoStatusBlock=0x2841bff770 [0/5], Buffer=0x208d16befc0, Length=0x1000, ByteOffset=null, Key=null) => 0

GCF: 3

Enter the command (0 to switch libraries, 1 for GCF, 2 for Pi, 3 to exit):

```
NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null,
IoStatusBlock=0x2841bff770, Buffer=0x208d16befc0, Length=0x1000, ByteOffset=null,
Key=null) ...
```

0

```
NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null,
IoStatusBlock=0x2841bff770 [0/3], Buffer=0x208d16befc0, Length=0x1000,
ByteOffset=null, Key=null) => 0
```

```
NtUnmapViewOfSection(ProcessHandle=-1, BaseAddress=0x7ff93ac40000) ...
```

```
Unload of DLL at 00007FF93AC40000
```

```
NtUnmapViewOfSection(ProcessHandle=-1, BaseAddress=0x7ff93ac40000) => 0
```

```
NtUnmapViewOfSection(ProcessHandle=-1, BaseAddress=0x7ff939380000) ...
```

```
Unload of DLL at 00007FF939380000
```

```
NtUnmapViewOfSection(ProcessHandle=-1, BaseAddress=0x7ff939380000) => 0
```

```
NtOpenFile(FileHandle=0x2841bff470, DesiredAccess=SYNCHRONIZE|0x21,
ObjectAttributes="\??\C:\Users\klosh\Desktop\OS4\libGCF_Naive.dll",
IoStatusBlock=0x2841bff4d8, ShareAccess=5, OpenOptions=0x60) ...
```

```
NtOpenFile(FileHandle=0x2841bff470 [0xc4], DesiredAccess=SYNCHRONIZE|0x21,
ObjectAttributes="\??\C:\Users\klosh\Desktop\OS4\libGCF_Naive.dll",
IoStatusBlock=0x2841bff4d8 [0/1], ShareAccess=5, OpenOptions=0x60) => 0
```

```
NtMapViewOfSection(SectionHandle=0xc0, ProcessHandle=-1,
BaseAddress=0x208d16c1a50 [0], ZeroBits=0, CommitSize=0, SectionOffset=null,
ViewSize=0x208d16bbe98 [0], InheritDisposition=1 [ViewShare],
AllocationType=0x00800000, Protect=0x80) ...
```

```
Loaded DLL at 00007FF93BF20000 C:\Users\klosh\Desktop\OS4\libGCF_Naive.dll
```

```
NtMapViewOfSection(SectionHandle=0xc0, ProcessHandle=-1,
BaseAddress=0x208d16c1a50 [0x00007ff93bf20000], ZeroBits=0, CommitSize=0,
SectionOffset=null, ViewSize=0x208d16bbe98 [0x00015000], InheritDisposition=1
[ViewShare], AllocationType=0x00800000, Protect=0x80) => 0
```

```
NtOpenFile(FileHandle=0x2841bff470, DesiredAccess=SYNCHRONIZE|0x21,
ObjectAttributes="\??\C:\Users\klosh\Desktop\OS4\libPi_Wallis.dll",
IoStatusBlock=0x2841bff4d8, ShareAccess=5, OpenOptions=0x60) ...
```

NtOpenFile(FileHandle=0x2841bff470 [0xc4], DesiredAccess=SYNCHRONIZE[0x21, ObjectAttributes="\??\C:\Users\klosh\Desktop\OS4\libPi\_Wallis.dll", IoStatusBlock=0x2841bff4d8 [0/1], ShareAccess=5, OpenOptions=0x60) => 0

NtMapViewOfSection(SectionHandle=0xc0, ProcessHandle=-1, BaseAddress=0x208d16c1d40 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16bbe98 [0], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF93BA20000 C:\Users\klosh\Desktop\OS4\libPi\_Wallis.dll

NtMapViewOfSection(SectionHandle=0xc0, ProcessHandle=-1, BaseAddress=0x208d16c1d40 [0x00007ff93ba20000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16bbe98 [0x00015000], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

Switched to another library.

Enter the command (0 to switch libraries, 1 for GCF, 2 for Pi, 3 to exit):

NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null, IoStatusBlock=0x2841bff770, Buffer=0x208d16befc0, Length=0x1000, ByteOffset=null, Key=null) ...

1

NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null, IoStatusBlock=0x2841bff770 [0/3], Buffer=0x208d16befc0, Length=0x1000, ByteOffset=null, Key=null) => 0

Enter two numbers: NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null, IoStatusBlock=0x2841bff770, Buffer=0x208d16befc0, Length=0x1000, ByteOffset=null, Key=null) ...

6 9

NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null, IoStatusBlock=0x2841bff770 [0/5], Buffer=0x208d16befc0, Length=0x1000, ByteOffset=null, Key=null) => 0

GCF: 3

Enter the command (0 to switch libraries, 1 for GCF, 2 for Pi, 3 to exit):

NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null, IoStatusBlock=0x2841bff770, Buffer=0x208d16befc0, Length=0x1000, ByteOffset=null, Key=null) ...

3

NtReadFile(FileHandle=0x5c, Event=0, ApcRoutine=null, ApcContext=null, IoStatusBlock=0x2841bff770 [0/3], Buffer=0x208d16befc0, Length=0x1000, ByteOffset=null, Key=null) => 0

NtUnmapViewOfSection(ProcessHandle=-1, BaseAddress=0x7ff93bf20000) ...

Unload of DLL at 00007FF93BF20000

NtUnmapViewOfSection(ProcessHandle=-1, BaseAddress=0x7ff93bf20000) => 0

NtUnmapViewOfSection(ProcessHandle=-1, BaseAddress=0x7ff93ba20000) ...

Unload of DLL at 00007FF93BA20000

NtUnmapViewOfSection(ProcessHandle=-1, BaseAddress=0x7ff93ba20000) => 0

NtOpenFile(FileHandle=0x2841bff520, DesiredAccess=SYNCHRONIZE|0x21, ObjectAttributes="\??\C:\WINDOWS\SYSTEM32\kernel.appcore.dll", IoStatusBlock=0x2841bff588, ShareAccess=5, OpenOptions=0x60) ...

NtOpenFile(FileHandle=0x2841bff520 [0xc4], DesiredAccess=SYNCHRONIZE|0x21, ObjectAttributes="\??\C:\WINDOWS\SYSTEM32\kernel.appcore.dll", IoStatusBlock=0x2841bff588 [0/1], ShareAccess=5, OpenOptions=0x60) => 0

NtMapViewOfSection(SectionHandle=0xb8, ProcessHandle=-1, BaseAddress=0x208d16bbcf0 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16c1ac8 [0], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF95E8E0000 C:\WINDOWS\SYSTEM32\kernel.appcore.dll

NtMapViewOfSection(SectionHandle=0xb8, ProcessHandle=-1, BaseAddress=0x208d16bbcf0 [0x00007ff95e8e0000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16c1ac8 [0x00018000], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

NtMapViewOfSection(SectionHandle=0xc0, ProcessHandle=-1, BaseAddress=0x208d16c1d20 [0], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16c1c88 [0], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) ...

Loaded DLL at 00007FF960210000 C:\WINDOWS\System32\msvcrt.dll

NtMapViewOfSection(SectionHandle=0xc0, ProcessHandle=-1, BaseAddress=0x208d16c1d20 [0x00007ff960210000], ZeroBits=0, CommitSize=0, SectionOffset=null, ViewSize=0x208d16c1c88 [0x000a7000], InheritDisposition=1 [ViewShare], AllocationType=0x00800000, Protect=0x80) => 0

Process 18472 exit code: 0

## **Вывод**

В ходе лабораторной работы были приобретены навыки создания и использования динамических библиотек на платформе Windows. Реализованы две программы: первая использует функции библиотек через статическую линковку на этапе компиляции, а вторая загружает библиотеки динамически во время выполнения с помощью средств WINAPI. Проведен сравнительный анализ обоих подходов, выявлены их преимущества и недостатки. Работа продемонстрировала гибкость и эффективность динамических библиотек, а также важность правильного управления ресурсами при их использовании.