Лекция 6. Запуск процесса

Архитектура ОС Windows

6 ноября 2014 г.

Секции модуля

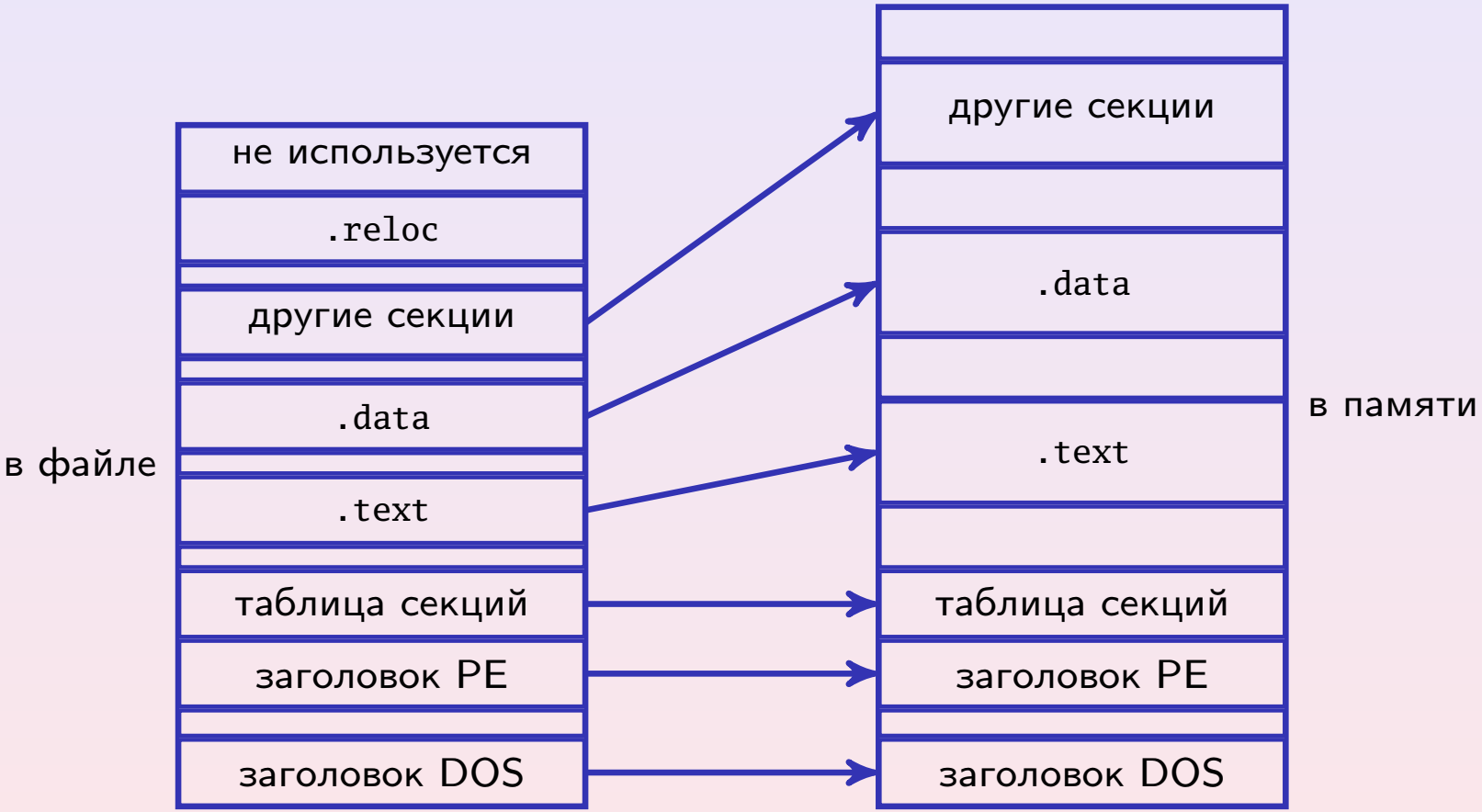

Рис. 1: секции модуля

Секции

Название	Назначение
`.text`	Секция кода
`.data`	Секция данных чтения/записи. Глобальные переменные.
`.rdata`	Секция данных только на чтение. Строковые литералы, таблицы виртуальных функций C++/COM.
`.idata`	Таблица импорта. Обычно → `.rdata`
`.edata`	Таблица экспорта (`.exp`). Обычно → `.text` или `.rdata`
`.rsrc`	Ресурсы, только для чтения. Не должна иметь другого названия и сливаться с другими секциями.
`.bss`	Неинициализированные данные (обычно вместо — `.data` с достаточным размером — `VirtualSize`).

Таблица 1: стандартные секции

Секции (окончание)

Название	Назначение
`.crt`	Данные для поддержки библиотеки времени выполнения C++.
`.tls`	Данные локального хранилища потока (TLS): начальные значения, ... (`__declspec (thread) DWORD gt_dwStartTime = 0;`)
`.reloc`	Базовые смещения в исполняемом файле. Как правило, DLL. Удаляется с ключом `/FIXED`
`.didat`	Таблица импорта для библиотек с отложенной загрузкой.

Таблица 2: стандартные секции (окончание)

Пример общих данных

Пример

```
#pragma data_seg(".shared")

int g_n = 10;     // Инициализация!

#pragma data_seg()
#pragma comment(linker, "/SECTION:.shared,RWS")

int g_n2;

// ...
```

Формат файла PE

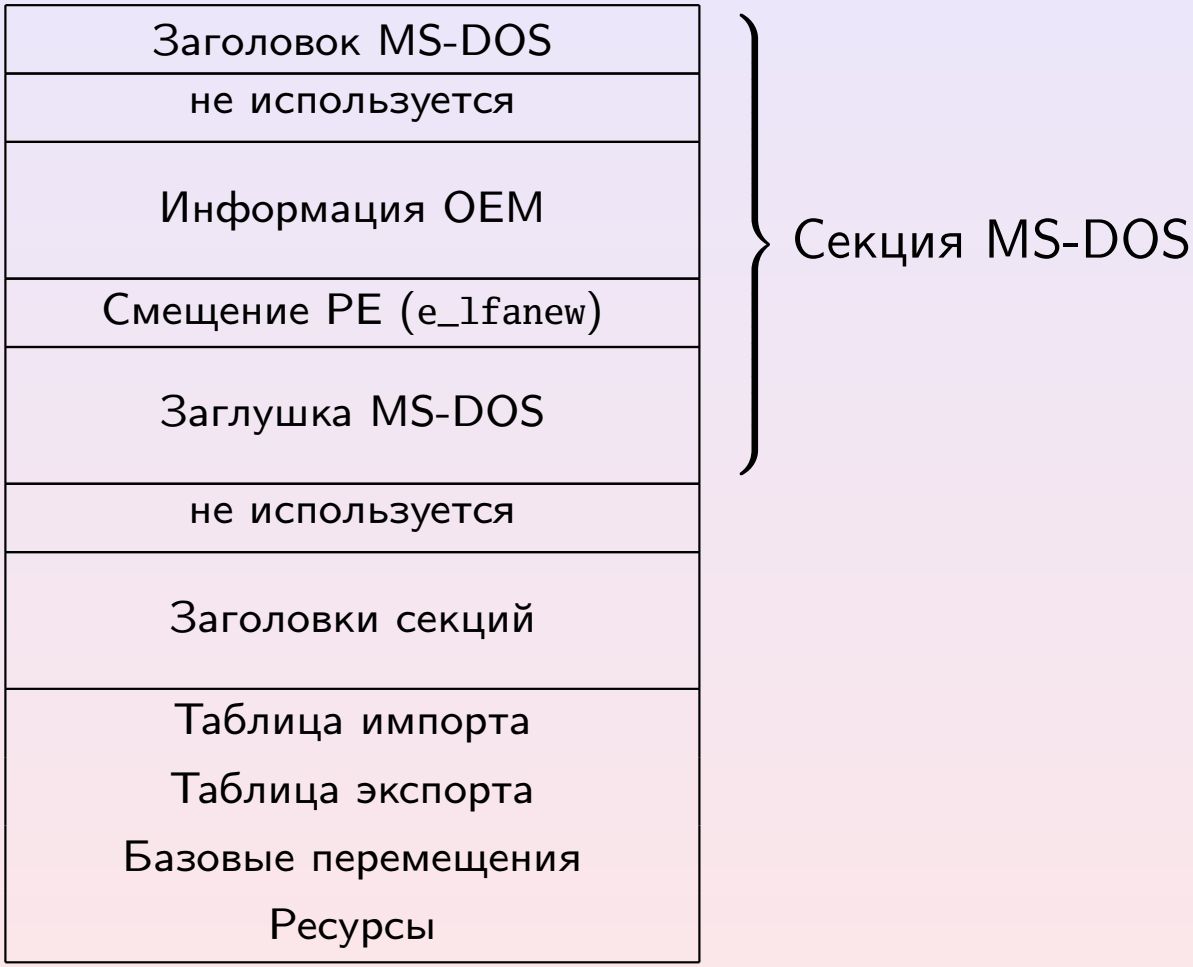

Рис. 2: формат Portable Executable

Заголовки NT

IMAGE_NT_HEADERS

```
typedef
  struct _IMAGE_NT_HEADERS
  {
    DWORD                   Signature;          // PE\

    IMAGE_FILE_HEADER       FileHeader;         // формат COFF
    IMAGE_OPTIONAL_HEADER32 OptionalHeader;
  }
    IMAGE_NT_HEADERS32, *PIMAGE_NT_HEADERS32;
```

Просмотр структуры файла

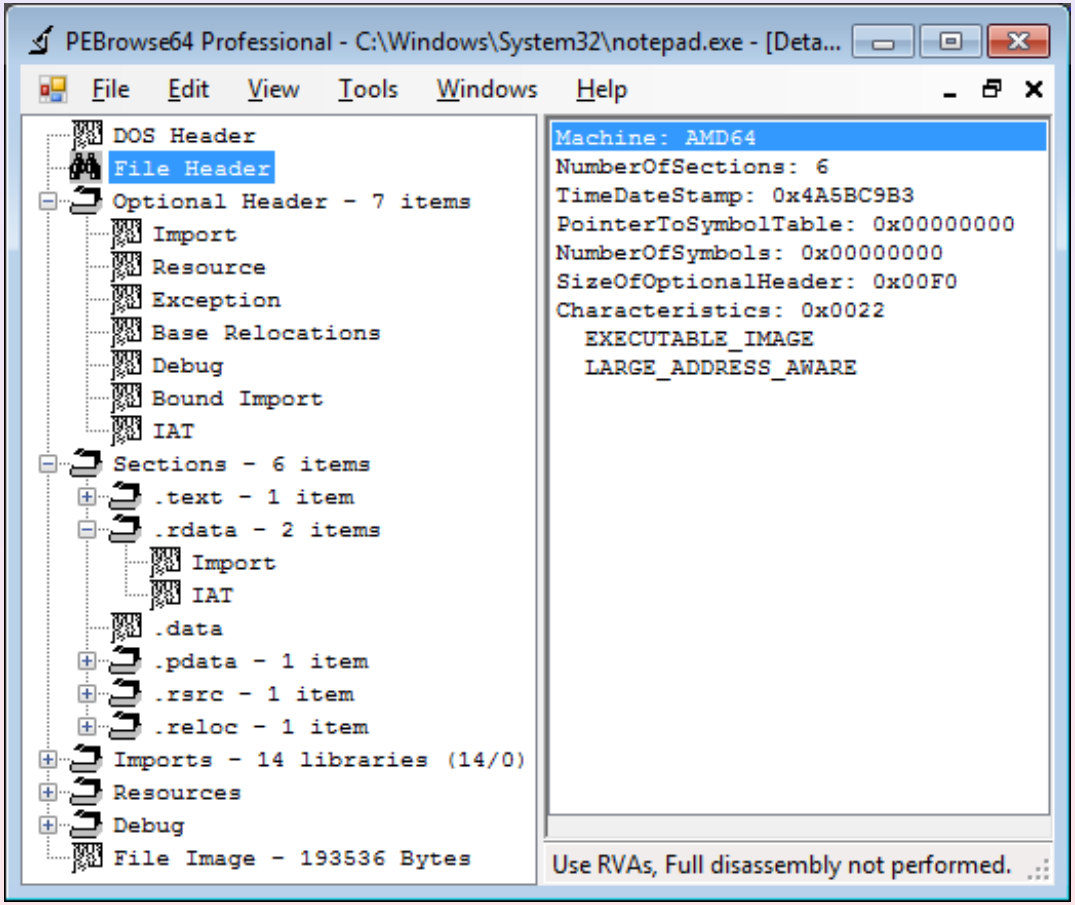

Рис. 3: окно программы PEBrowse64

Поля заголовка файла

Название	Назначение
Machine	Целевая архитектура процессора.
NumberOfSections	Количество секций (таблица секций сразу после IMAGE_NT_HEADERS)
	...
Characteristics	Набор флагов

Таблица 3: формат заголовка COFF (IMAGE_FILE_HEADER, ...)

Архитектуры

IMAGE_FILE_MACHINE_UNKNOWN
IMAGE_FILE_MACHINE_AMD64
IMAGE_FILE_MACHINE_ARM
IMAGE_FILE_MACHINE_ARMNT
IMAGE_FILE_MACHINE_ARM64
IMAGE_FILE_MACHINE_EBC
IMAGE_FILE_MACHINE_I386
IMAGE_FILE_MACHINE_IA64
IMAGE_FILE_MACHINE_THUMB
...

Таблица 4: константы типов архитектур

Характеристики

IMAGE_FILE_RELOCS_STRIPPED
IMAGE_FILE_EXECUTABLE_IMAGE
IMAGE_FILE_LARGE_ADDRESS_AWARE
IMAGE_FILE_32BIT_MACHINE
IMAGE_FILE_DEBUG_STRIPPED
IMAGE_FILE_REMOVABLE_RUN_FROM_SWAP
IMAGE_FILE_NET_RUN_FROM_SWAP
IMAGE_FILE_DLL
...

Таблица 5: константы характеристик

Пример заголовка файла (notepad.exe)

Пример (IMAGE_FILE_HEADER)

```
Machine: AMD64
NumberOfSections: 6
TimeDateStamp: 0x4A5BC9B3
PointerToSymbolTable: 0x00000000
NumberOfSymbols: 0x00000000
SizeOfOptionalHeader: 0x00F0
Characteristics: 0x0022
  EXECUTABLE_IMAGE
  LARGE_ADDRESS_AWARE
```

Поля дополнительного заголовка

Название	Назначение
	Стандартные поля
AddressOfEntryPoint	RVA первого байта точки входа или 0.
BaseOfCode	RVA первого байта кода
BaseOfData	RVA первого байта данных (нет в PE32+)
	Поля, специфичные для Windows (только файл образа)
ImageBase	Предпочтительный адрес загрузки модуля (32/64 бит, по умолчанию 0x00400000 для EXE PE32, 0x10000000 для DLL, VA = ImageBase + RVA).
SectionAlignment	Выравнивание секций в памяти (0x1000)
FileAlignment	Выравнивание секций в файле (0x200)
SizeOfImage	RVA ячейки за последней секцией (в памяти)
SizeOfHeaders	Размер всех заголовков до первой секции (в файле)

Таблица 6: основные поля IMAGE_OPTIONAL_HEADER

Поля дополнительного заголовка (окончание)

Название	Назначение
Subsystem	Подсистема
DllCharacteristics	Характеристики динамической библиотеки
SizeOfStackReserve	Зарезервированный объём стека основного потока (32/64 бит)
SizeOfStackCommit	Изначально выделенный объём стека (32/64 бит)
SizeOfHeapReserve	Зарезервированный объём кучи (32/64 бит)
SizeOfHeapCommit	Изначльно выделенный объём кучи (32/64 бит)
NumberOfRvaAndSizes	Количество элементов таблицы каталога данных за концом IMAGE_OPTIONAL_HEADER. Каждый определяет RVA и размер.

Таблица 7: основные поля IMAGE_OPTIONAL_HEADER (окончание)

Подсистема

IMAGE_SUBSYSTEM_UNKNOWN
IMAGE_SUBSYSTEM_NATIVE
IMAGE_SUBSYSTEM_WINDOWS_GUI
IMAGE_SUBSYSTEM_WINDOWS_CUI
IMAGE_SUBSYSTEM_POSIX_CUI
IMAGE_SUBSYSTEM_EFI_APPLICATION
IMAGE_SUBSYSTEM_EFI_BOOT_SERVICE_DRIVER
IMAGE_SUBSYSTEM_EFI_ROM
...

Таблица 8: константы подсистем (Subsystem)

IMAGE_DLL_CHARACTERISTICS_DYNAMIC_BASE
...

Таблица 9: константы характеристик библиотеки (DllCharacteristics)

Элементы таблицы каталога данных

IMAGE_DATA_DIRECTORY

```
typedef
  struct _IMAGE_DATA_DIRECTORY
  {
    DWORD   VirtualAddress;    // RVA
    DWORD   Size;
  }
    IMAGE_DATA_DIRECTORY, *PIMAGE_DATA_DIRECTORY;

// Массив из NumberOfRvaAndSizes структур
```

Индексы каталога данных

Индекс	Секция
IMAGE_DIRECTORY_ENTRY_EXPORT	.edata
IMAGE_DIRECTORY_ENTRY_IMPORT	.idata
IMAGE_DIRECTORY_ENTRY_RESOURCE	.rsrc
IMAGE_DIRECTORY_ENTRY_BASERELOC	.reloc
IMAGE_DIRECTORY_ENTRY_DEBUG	.debug
IMAGE_DIRECTORY_ENTRY_TLS	.tls
IMAGE_DIRECTORY_ENTRY_IAT	—
IMAGE_DIRECTORY_ENTRY_DELAY_IMPORT	.didata
...	

Таблица 10: константы индексов основных элементов каталога данных

Пример дополнительного заголовка (notepad.exe)

Пример (IMAGE_OPTIONAL_HEADER)

Magic: 0x020B
MajorLinkerVersion: 0x09
MinorLinkerVersion: 0x00
SizeOfCode: 0x0000A800
SizeOfInitializedData: 0x00025800
SizeOfUninitializedData: 0x00000000
AddressOfEntryPoint: 0x00003570
BaseOfCode: 0x00001000
ImageBase: 0x00000001‘00000000
SectionAlignment: 0x00001000
FileAlignment: 0x00000200
MajorOperatingSystemVersion: 0x0006
MinorOperatingSystemVersion: 0x0001
MajorImageVersion: 0x0006

Пример (продолжение)

MinorImageVersion: 0x0001
MajorSubsystemVersion: 0x0006
MinorSubsystemVersion: 0x0001
Win32VersionValue: 0x00000000
SizeOfImage: 0x00035000
SizeOfHeaders: 0x00000600
CheckSum: 0x0003E749
Subsystem: 0x0002
  WINDOWS_GUI
DllCharacteristics: 0x8140
  DLL can move.
  Image is NX compatible.
  TERMINAL_SERVER_AWARE

Пример дополнительного заголовка (окончание)

Пример (окончание)

```
SizeOfStackReserve:          0x0000000000080000
SizeOfStackCommit:           0x0000000000011000
SizeOfHeapReserve:           0x0000000000100000
SizeOfHeapCommit:            0x0000000000001000
LoaderFlags: 0x00000000
NumberOfRvaAndSizes: 0x00000010
IMAGE_DIRECTORY_ENTRY_EXPORT (0)
  VirtualAddress: 0x00000000
  Size: 0x00000000
IMAGE_DIRECTORY_ENTRY_IMPORT (1)
  VirtualAddress: 0x0000CFF8
  Size: 0x0000012C
...
```

Заголовки секций (после дополнительного заголовка)

Название	Назначение
Name	Название (до 8 символов).
VirtualSize	Размер в памяти (для исполняемых образов).
VirtualAddress	RVA начала секции (для исполняемых образов).
SizeOfRawData	Размер проинициализированных данных ($\leqslant$ VirtualSize).
PointerToRawData	Файловая позиция начала данных (кратно FileAlignment).
PointerToRelocations	Файловая позиция начала данных перемещения или 0.
Characteristics	Флаги секции.

Таблица 11: основные поля IMAGE_SECTION_HEADER (в количестве NumberOfSections)

Флаги секции

IMAGE_SCN_CNT_CODE
IMAGE_SCN_CNT_INITIALIZED_DATA
IMAGE_SCN_CNT_UNINITIALIZED_DATA
IMAGE_SCN_LNK_NRELOC_OVFL
IMAGE_SCN_MEM_DISCARDABLE
IMAGE_SCN_MEM_NOT_CACHED
IMAGE_SCN_MEM_NOT_PAGED
IMAGE_SCN_MEM_SHARED
IMAGE_SCN_MEM_EXECUTE
IMAGE_SCN_MEM_READ
IMAGE_SCN_MEM_WRITE
...

Таблица 12: флаги секции

Примеры характеристик основных секций

Имя	Характеристики
.text	IMAGE_SCN_CNT_CODE | IMAGE_SCN_MEM_EXECUTE | IMAGE_SCN_MEM_READ
.data, .idata, .tls	IMAGE_SCN_CNT_INITIALIZED_DATA | IMAGE_SCN_MEM_READ | IMAGE_SCN_MEM_WRITE
.rdata, .edata, .rsrc	IMAGE_SCN_CNT_INITIALIZED_DATA | IMAGE_SCN_MEM_READ
.reloc	IMAGE_SCN_CNT_INITIALIZED_DATA | IMAGE_SCN_MEM_READ | IMAGE_SCN_MEM_DISCARDABLE
.bss	IMAGE_SCN_CNT_UNINITIALIZED_DATA | IMAGE_SCN_MEM_READ | IMAGE_SCN_MEM_WRITE
...	

Таблица 13: сочетания характеристик для основных секций

Примеры заголовков секций (notepad.exe)

Пример (IMAGE_SECTION_HEADER)

```
Name: .text
VirtualSize: 0x0000A770
VirtualAddress: 0x00001000
SizeOfRawData: 0x0000A800
PointerToRawData: 0x00000600
PointerToRelocations: 0x00000000
PointerToLinenumbers: 0x00000000
NumberOfRelocations: 0x00000000
NumberOfLinenumbers: 0x00000000
Characteristics: 0x60000020
  Contains code.
  Is executable.
  Is readable.
```

Пример (окончание)

```
Name: .rdata
VirtualSize: 0x00003160
VirtualAddress: 0x0000C000
SizeOfRawData: 0x00003200
PointerToRawData: 0x0000AE00
PointerToRelocations: 0x00000000
PointerToLinenumbers: 0x00000000
NumberOfRelocations: 0x00000000
NumberOfLinenumbers: 0x00000000
Characteristics: 0x40000040
  Contains initialized data.
  Is readable.
```

Секция экспорта (.edata)

Название	Назначение
Таблица каталога экспорта	Содержит расположения и размеры остальных таблиц экспорта (1 строка).
Таблица экспорта адресов (EAT)	Массив RVA экспортируемых символов (или строк перенаправления).
Экспорт по именам и порядковым номерам	
Таблица указателей имён	Массив указателей на экспортртуемые символы, сортированные в порядке возрастания.
Таблица номеров	Массив порядковых номеров (индексам в EAT), $\sim$ элементам предыдущего массива.
Таблица экспортируемых имён	Серия строк, на которые ссылаются указатели.

Таблица 14: таблицы экспорта

Секция импорта (.idata)

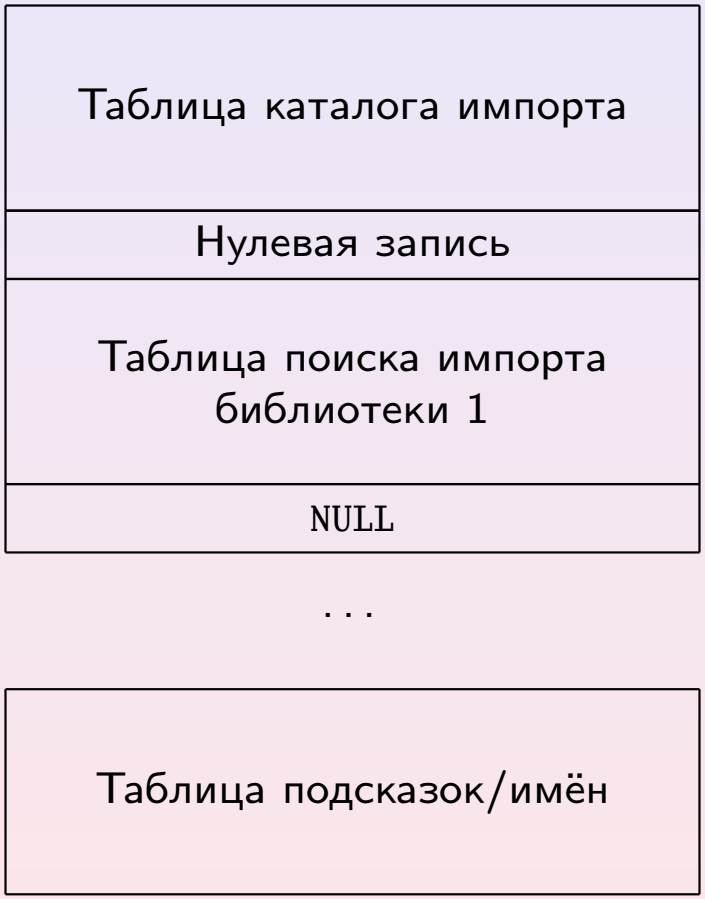

Рис. 4: структура секции импорта

Запись таблицы каталога импорта

Название	Назначение
`OriginalFirstThunk`	RVA таблицы поиска импорта (несвязанной IAT).
`Name`	RVA имени библиотеки.
`FirstThunk`	RVA IAT.

Таблица 15: основные поля `IMAGE_IMPORT_DESCRIPTOR`

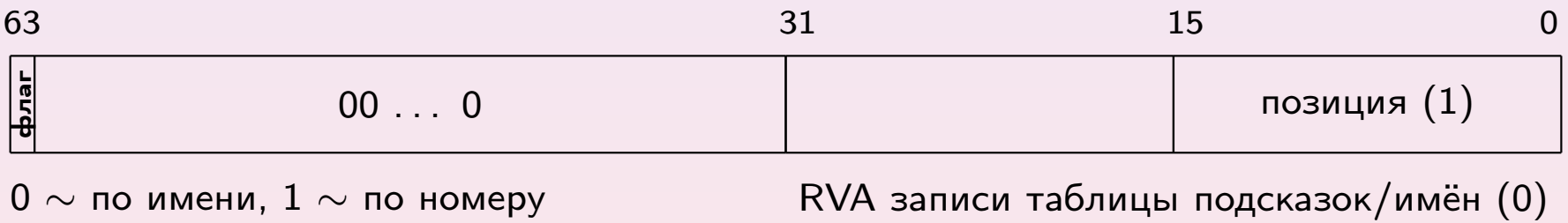

Рис. 5: формат записи таблицы поиска имени (32/64 бит)

Запись таблицы подсказок/имён

Название	Назначение
Подсказка	Индекс в таблице указателей экспортируемых имён.
Имя	ASCII-строка + '\0'.

Таблица 16: поля записи таблицы подсказок/имён

Пример таблицы импорта (notepad.exe)

Пример (таблица импорта)

```
OriginalFirstThunk: 0x0000D1E8
TimeDateStamp:      0xFFFFFFFF
ForwarderChain:     0xFFFFFFFF
Name:               0x0000D1D4 - ADVAPI32.dll
FirstThunk:         0x0000C000
0x10000D1E8: 00000000‘0000D9D8 000007FF‘7FF21ED0 (638, RegSetValueExW)
0x10000D1F0: 00000000‘0000D9EA 000007FF‘7FF2C2D0 (622, RegQueryValueExW)
0x10000D1F8: 00000000‘0000D9FE 000007FF‘7FF21F00 (572, RegCreateKeyW)
0x10000D200: 00000000‘0000DA0E 000007FF‘7FF30710 (560, RegCloseKey)
0x10000D208: 00000000‘0000DA1C 000007FF‘7FF306F0 (609, RegOpenKeyExW)
0x10000D210: 00000000‘0000DA2C 000007FF‘7FF30720 (384, IsTextUnicode)
...
```

Пример таблицы адресов импорта (notepad.exe)

Пример (таблица импорта)

```
Table: 1
0x000000010000C000: IATEntry001: 0x000007FF‘7FF21ED0
0x000000010000C008: IATEntry002: 0x000007FF‘7FF2C2D0
0x000000010000C010: IATEntry003: 0x000007FF‘7FF21F00
0x000000010000C018: IATEntry004: 0x000007FF‘7FF30710
0x000000010000C020: IATEntry005: 0x000007FF‘7FF306F0
0x000000010000C028: IATEntry006: 0x000007FF‘7FF30720
0x000000010000C030: IATEntry007: 0x000007FF‘7FF27E04
0x000000010000C038: IATEntry008: 0x000007FF‘7FF1C2A8
0x000000010000C040: IATEntry009: 0x000007FF‘7FF1C2C0
0x000000010000C048: IATEntry010: 0x000007FF‘7FF1C6FC
...
```

Пример

Пример

```
void MyFunction();
```

Пример

```
__declspec(dllimport) void MyFunction();
```

Пример

```
                call 0x0040100C                 ; ~ MyFunction
; ...
0x0040100C: jmp  dword ptr [0x00405030]         ; ~ __imp__MyFunction
```

Пример

```
                call dword ptr [0x00405030]     ; слот IAT
```

Пример

Пример

```
0x00401020: 8B 0D 34 D4 40 00   mov ecx,dword ptr [0x0040D434]
                                ;           ImageBase == 0x00400000
```

Таблица перемещений (.reloc)

Формат записи таблицы перемещений (IMAGE_BASE_RELOCATION)

```
typedef
  struct _IMAGE_BASE_RELOCATION
  {
    DWORD   VirtualAddress;     // RVA страницы
    DWORD   SizeOfBlock;        // Размер этой записи
  // WORD    TypeOffset[1];
  }
    IMAGE_BASE_RELOCATION;
```

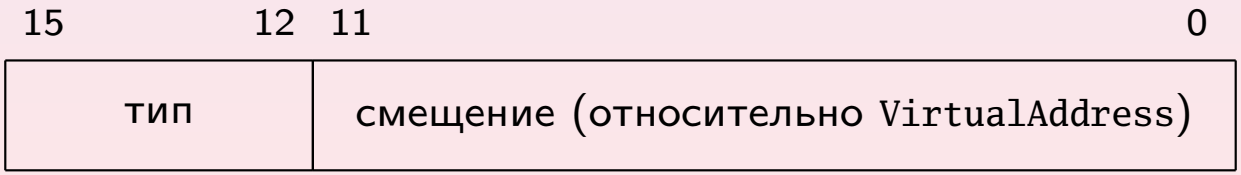

Рис. 6: формат записи таблицы «тип/смещение» (TypeOffset, 16 бит)

Типы перемещения

Константа	Назначение
IMAGE_REL_BASED_ABSOLUTE	ничего не делает.
IMAGE_REL_BASED_HIGH	прибавляет к 16-битному полю старшие 16 бит Δ.
IMAGE_REL_BASED_LOW	прибавляет к 16-битному полю младшие 16 бит Δ.
IMAGE_REL_BASED_HIGHLOW	прибавляет к 32-битному полю все 32 бит Δ.
IMAGE_REL_BASED_HIGHADJ	как IMAGE_REL_BASED_HIGH + младшие 16 бит, применяемые к младшей части, хранятся в следующей записи TypeOffset.
IMAGE_REL_BASED_DIR64	прибавляет к 64-битному полю все 64 бит Δ.
	...

Таблица 17: основные типы перемещения

Пример таблицы перемещений (notepad.exe)

Пример (таблица перемещений)

```
Table: 1
VirtualAddress:  0x0000C000 - .rdata
SizeOfBlock:     0x00000014
  0x0000CA88 DIR64 - 0x00000001‘00003300
  0x0000CAA0 DIR64 - 0x00000001‘00003D94
  0x0000CAA8 DIR64 - 0x00000001‘00003350
  0x0000CE58 DIR64 - 0x00000001‘000104A0
  0x0000CE60 DIR64 - 0x00000001‘00010540
...
```

Структуры данных процесса

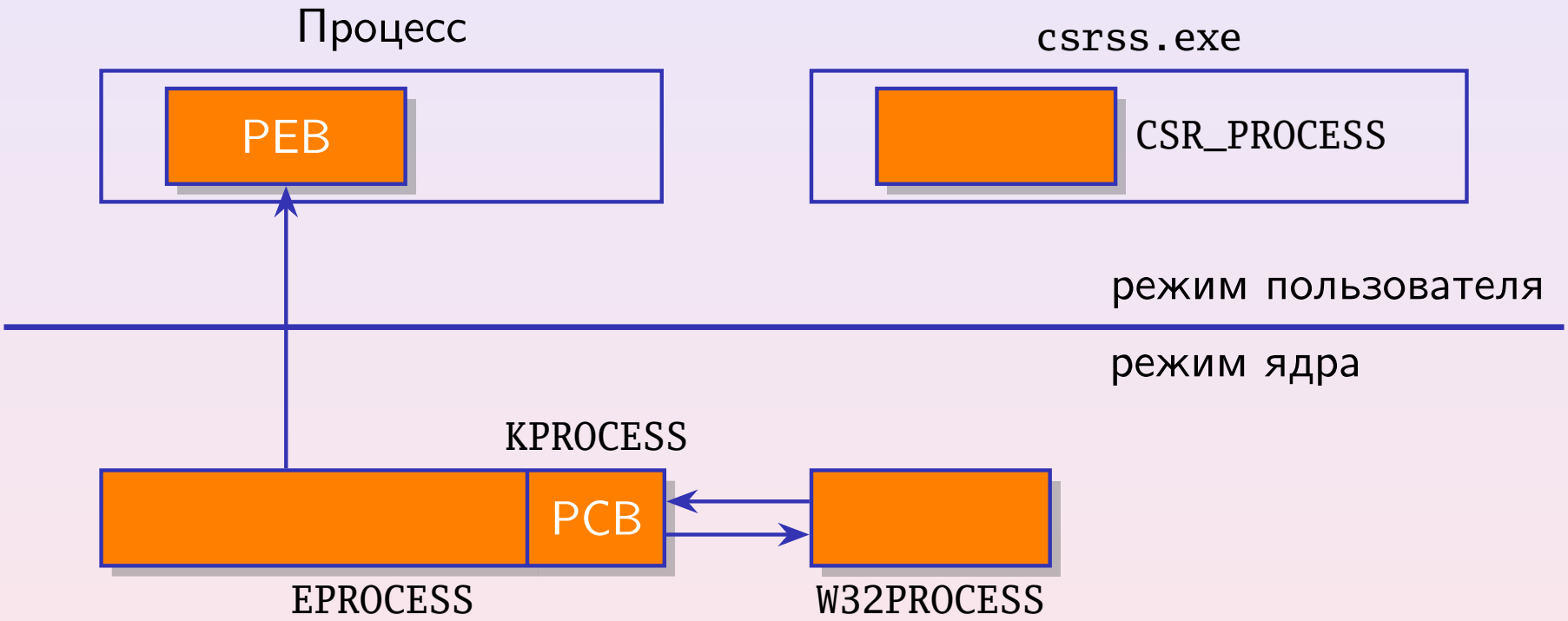

Рис. 7: основные структуры процесса

Данные структуры EPROCESS (KPROCESS)

Основные поля структуры EPROCESS

- идентификатор,
- идентификатор родителя,
- код возврата,
- маркер доступа,
- таблица дескрипторов,
- имя образа,
- базовый адрес,
- ...

Основные поля структуры KPROCESS

- ссылка на каталог страниц,
- состояние,
- время ядра,
- время пользователя,
- аффинность,
- идеальный процессор,
- список структур потоков,
- ...

Данные структур PEB, CSR_PROCESS, W32_PROCESS

Основные поля структуры PEB

- базовый адрес,
- база данных загрузчика,
- данные TLS,
- указатель на кучу,
- указатель на разделяемую таблицу дескрипторов GDI,
- маска аффинности,
- информация о совместимости,
- ...

Основные поля CSR_PROCESS

- идентификатор клиента,
- данные о сеансе,
- ...

Основные поля W32_PROCESS

- таблица дескрипторов,
- списки GDI,
- указатель на DXGPROCESS,
- ...

Порядок создания процесса

Алгоритм CreateProcess...()

1. Проверка и разбор параметров.
2. Открытие исполняемого файла.
3. Создание объекта процесса исполнительной системы.
4. Создание элементов основного потока.
5. Инициализация в подсистеме Windows.
6. Запуск на исполнение основного потока
7. Завершение инициализации в контексте потока.

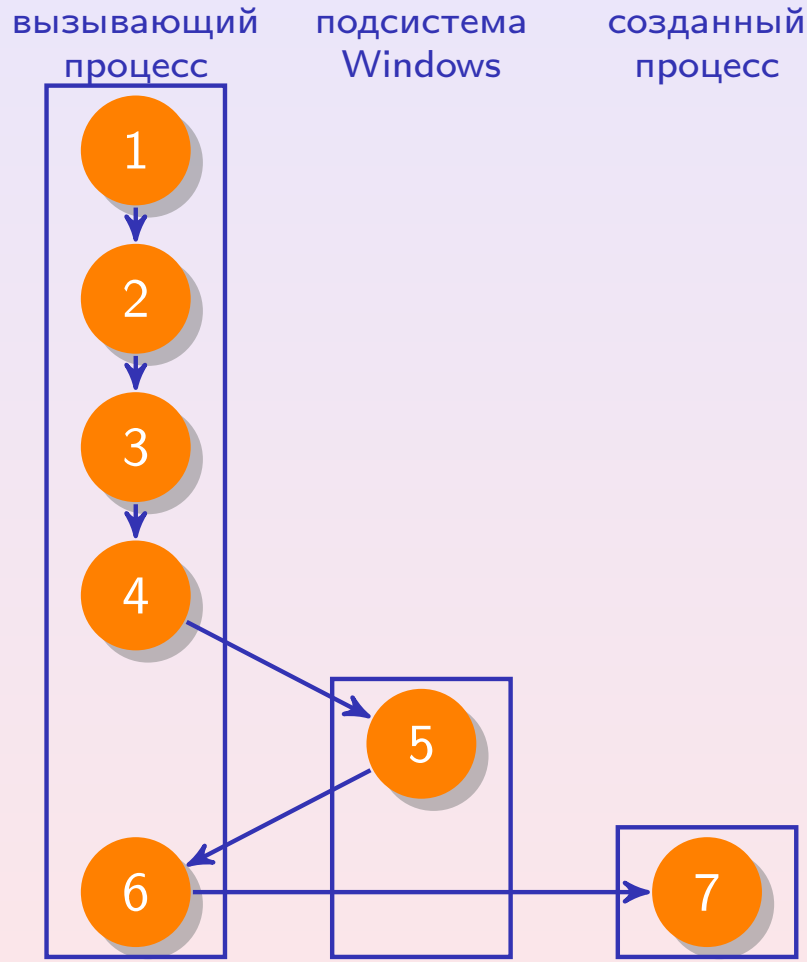

Рис. 8: порядок исполнения

Открытие файла образа

Тип файла	Исполняемый образ	Этап
Приложение Windows	Образ файла	3
Есть ключ с именем отладчика	Образ, указанный в ключе	1
Приложение Win16	`Ntvdm.exe` (общий или новый)	1
Приложение MS-DOS	`Ntvdm.exe` (общий)	1
Приложение POSIX	`Posix.exe`	1
Сценарий (`.bat`, `.cmd`)	`Cmd.exe`	1
Приложение Win64 на не поддерживающей архитектуре	—	ошибка
Неправильный формат, невозможно открыть	—	ошибка

Таблица 18: варианты алгоритма при открытии исполняемого образа в различных ситуациях

Инициализация исполнительной системы

Порядок работы

1. Заполнение структуры EPROCESS;
2. Создание начального адресного пространства;
3. Заполнение структуры KPROCESS;
4. Заполнение структуры PEB;
5. Завершение установки адресного пространства (отображение секций в адресное пространство, отображение Ntdll.dll, создание сеанса при необходимости, ...)