**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по практической работе № 1**

**по дисциплине «Операционные системы»**

Тема: **Исследование структур загрузочных модулей**

|  |  |  |
| --- | --- | --- |
| Студент гр. 9382 |  | Русинов Д.А. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2021

**Цель работы.**

Исследование различий в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

**Задание.**

Шаг 1. Напишите текст исходного .СОМ модуля, который определяет тип РС и версию системы. Это довольно простая задача и для тех, кто уже имеет опыт программирования на ассемблере, это будет небольшой разминкой. Для тех, кто раньше не сталкивался с программированием на ассемблере, это неплохая задача для первого опыта.

За основу возьмите шаблон, приведенный в разделе «Основные сведения». Необходимые сведения о том, как извлечь требуемую информацию, представлены в следующем разделе.

Ассемблерная программа должна читать содержимое предпоследнего байта ROM BIOS, по таблице, сравнивая коды, определять тип РС и выводить строку с названием модели. Если код не совпадает ни с одним значением, то двоичный код переводиться в символьную строку, содержащую запись шестнадцатеричного числа и выводиться на экран в виде соответствующего сообщения.

Затем определяется версия системы. Ассемблерная программа должна по значениям регистров AL и AH формировать текстовую строку в формате xx.yy, где xx - номер основной версии, а yy - номер модификации в десятичной системе счисления, формировать строки с серийным номером OEM и серийным номером пользователя. Полученные строки выводятся на экран.

Отладьте полученный исходный модуль.

Результатом выполнения этого шага будет «хороший» .СОМ модуль, а также необходимо построить «плохой» .ЕХЕ, полученный из исходного текста для .СОМ модуля.

Шаг 2. Напишите текст исходного .ЕХЕ модуля, который выполняет те же функции, что и модуль в Шаге 1 и постройте и отладьте его. Таким образом, будет получен «хороший» .ЕХЕ.

Шаг 3. Сравните исходные тексты для .СОМ и .ЕХЕ модулей. Ответьте на контрольные вопросы «Отличия исходных текстов СОМ и ЕХЕ программ».

Шаг 4. Запустите FAR и откройте (F3/F4) файл загрузочного модуля .СОМ и файл «плохого» .ЕХЕ в шестнадцатеричном виде. Затем откройте (F3/F4) файл загрузочного модуля «хорошего» .ЕХЕ и сравните его с предыдущими файлами. Ответьте на контрольные вопросы «Отличия форматов файлов СОМ и ЕХЕ модулей».

Шаг 5. Откройте отладчик TD.EXE и загрузите .СОМ. Ответьте на контрольные вопросы «Загрузка СОМ модуля в основную память». Представьте в отчете план загрузки модуля .СОМ в основную память.

Шаг 6. Откройте отладчик TD.EXE и загрузите «хороший» .ЕХЕ. Ответьте на контрольные вопросы «Загрузка «хорошего» ЕХЕ модуля в основную память».

Шаг 7. Оформление отчета в соответствии с требованиями. В отчете необходимо привести скриншоты. Для файлов их вид в шестнадцатеричном виде, для загрузочных модулей – в отладчике.

Необходимые сведения для составления программы

Тип IBM PC хранится в байте по адресу 0F000:0FFFEh, в предпоследнем байте ROM BIOS. Соответствие кода и типа в таблице:

PC FF

PC/XT FE,FB

AT FC

PS2 модель 30 FA

PS2 модель 50 или 60 FC

PS2 модель 80 F8

PCjr FD

PC Convertible F9

Для определения версии MS DOS следует воспользоваться функцией 30H прерывания 21H. Входным параметром является номер функции в AH:

MOV AH,30h

INT 21h

Выходными параметрами являются:

AL - номер основной версии. Если 0, то < 2.0

AH - номер модификации

BH - серийный номер OEM (Original Equipment Manufacturer) BL:CX - 24-битовый серийный номер пользователя.

Контрольные вопросы по лабораторной работе No1

Отличия исходных текстов СОМ и ЕХЕ программ

1) Сколько сегментов должна содержать COM-программа?

2) EXE-программа?

3) Какие директивы должны обязательно быть в тексте COM-программы?

4) Все ли форматы команд можно использовать в COM-программе?

Отличия форматов файлов СОМ и ЕХЕ модулей

1) Какова структура файла COM? С какого адреса располагается код?

2) Какова структура файла «плохого» EXE? С какого адреса располагается код? Что располагается с адреса 0?

3) Какова структура файла «хорошего» EXE? Чем он отличается от файла

«плохого» EXE?

Загрузка СОМ модуля в основную память

1) Какой формат загрузки модуля COM? С какого адреса располагается код?

2) Что располагается с адреса 0?

3) Какие значения имеют сегментные регистры? На какие области памяти они указывают?

4) Как определяется стек? Какую область памяти он занимает? Какие адреса?

Загрузка «хорошего» ЕХЕ модуля в основную память

1) Как загружается «хороший» EXE? Какие значения имеют сегментные регистры?

2) На что указывают регистры DS и ES?

3) Как определяется стек?

4) Как определяется точка входа?

**Выполнение работы.**

Были объявлены константные строки для вывода информации. Была определена процедура для определения типа PC – DEFINE\_PC в соответствии с таблицей в задании. А также функция для определения ОС и прочих данных – DEFINE\_OS.

В результате выполнения были получены следующие результаты:

|  |  |
| --- | --- |
| Модуль | Результат |
| Хороший .EXE | Type: AT  Version MS-DOS: 5.0  Serial number OEM: 0  User serial number: 000000H |
| Плохой .EXE |  |
| Хороший .COM | Type: AT  Version MS-DOS: 5.0  Serial number OEM: 0  User serial number: 000000H |

**Выводы.**

Были исследованы модули .COM и .EXE, рассмотрены из различия в исходных текстах и различия готовых модулей. Также были рассмотрены способы загрузки модулей в основную память.

ПРИЛОЖЕНИЕ А

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

**Отличия исходных текстов COM и EXE программ:**

1. Сколько сегментов должна содержать COM-программа?

Один сегмент. Стек генерируется автоматически, а код с данными располагаются в одном сегменте.

1. EXE-программа?

Не менее одного сегмента, при этом сегменты стека, кода и данных отдельны друг от друга.

1. Какие директивы должны быть обязательно в тексте COM-программы?

В программе должна быть обязательно директива org 100h. Она позволяет сместить всю адресацию на 256 байт. Это необходимо, поскольку первые 256 байт занимает блок PSP, а все сегментные регистры при загрузке указывают именно на него. Также необходимо привязать сегмент данных и сегмент кода на один общий сегмент с помощью ASSUME.

1. Все ли форматы команд можно использовать в COM-программе?

Нет. Нельзя использовать, например:

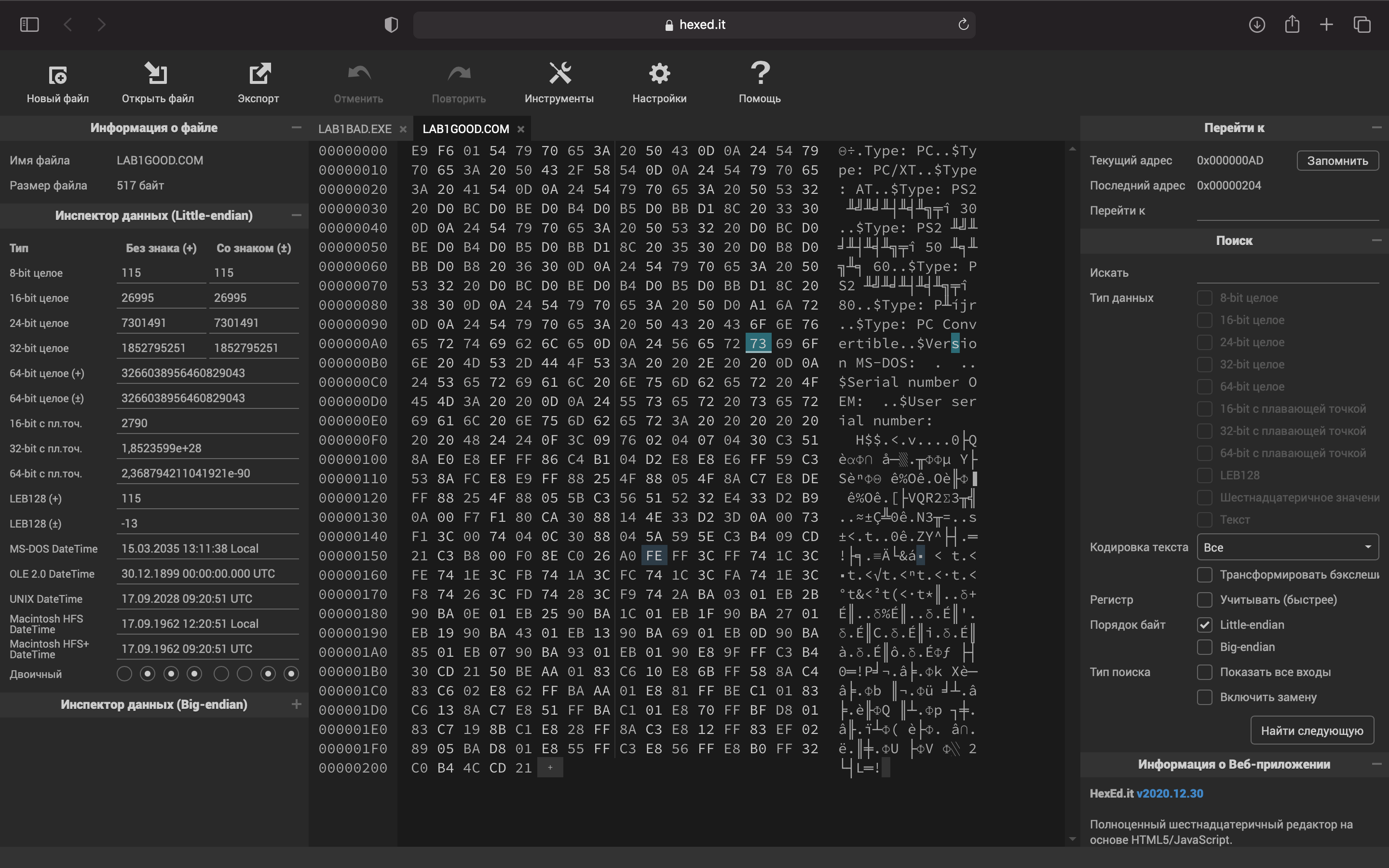
MOV REG, SEG

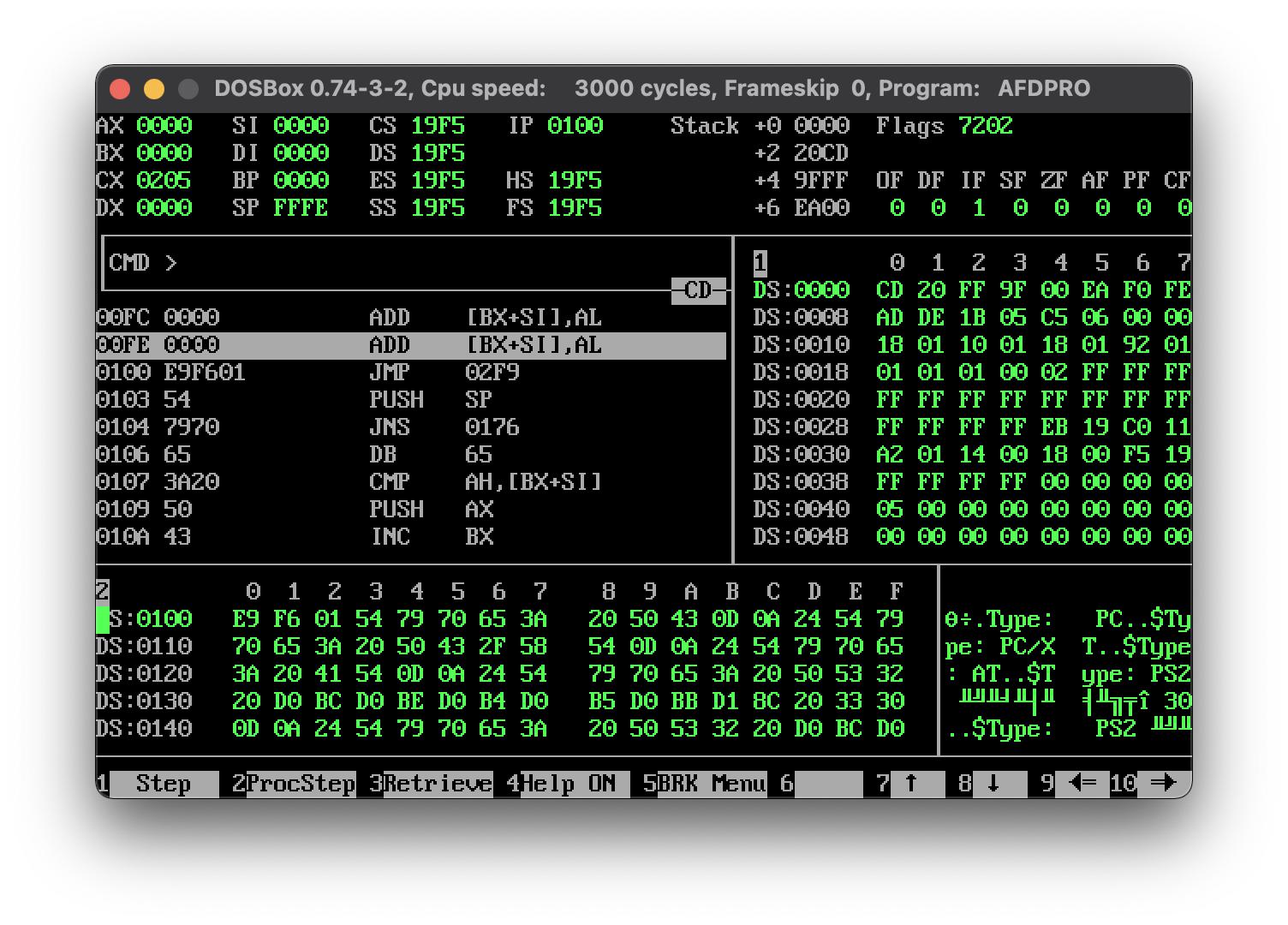
**Отличия форматов файлов .COM и.EXE программ:**

1. Какова структура файла .COM? С какого адреса располагается код?

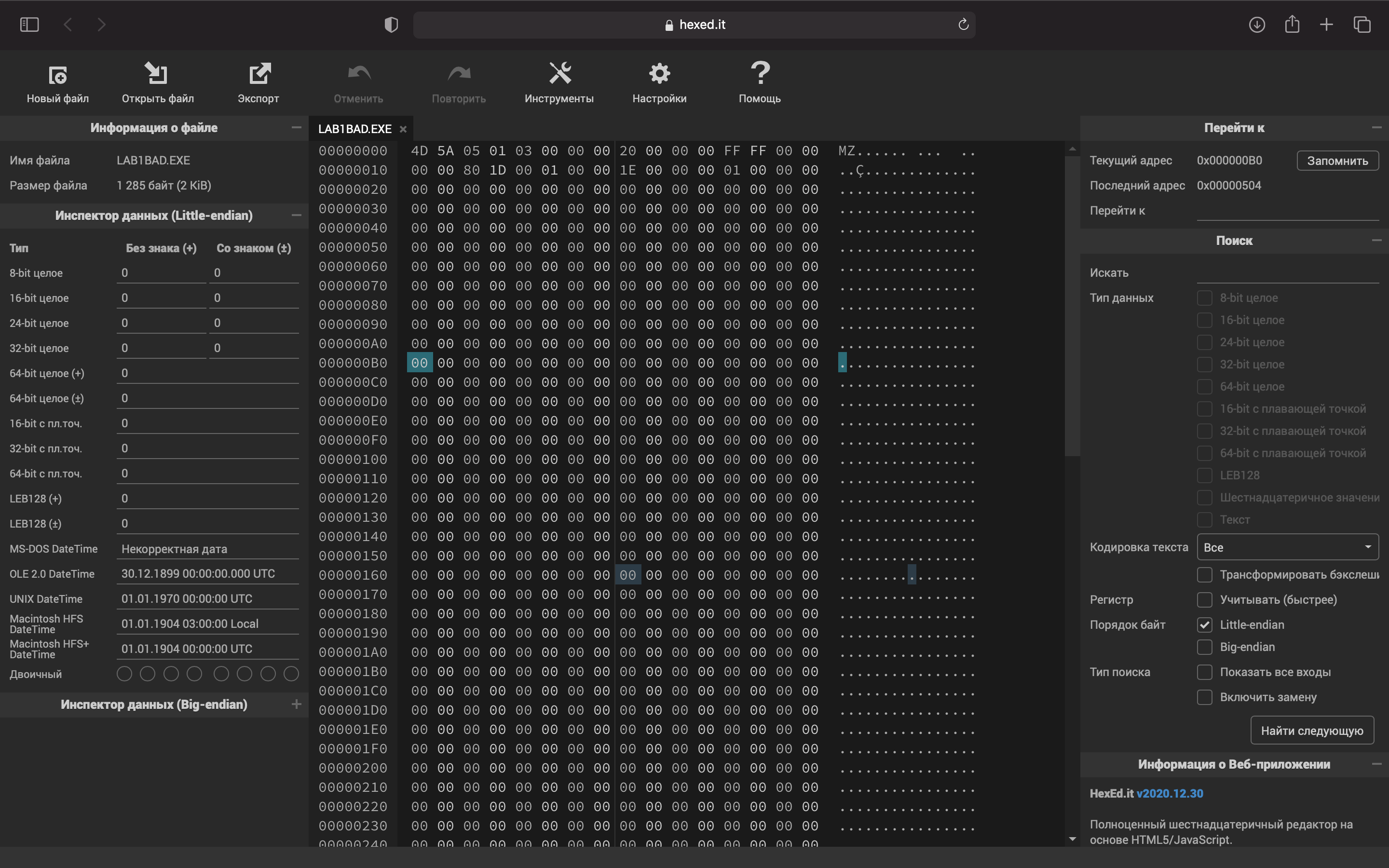
Структура состоит из одного сегмента, в который входит сегмент кода и сегмент данных, стек генерируется автоматически. Также COM-файл имеет ограничение в размере, его максимальный размер – 64 КБ – это максимальный размер одного сегмента.

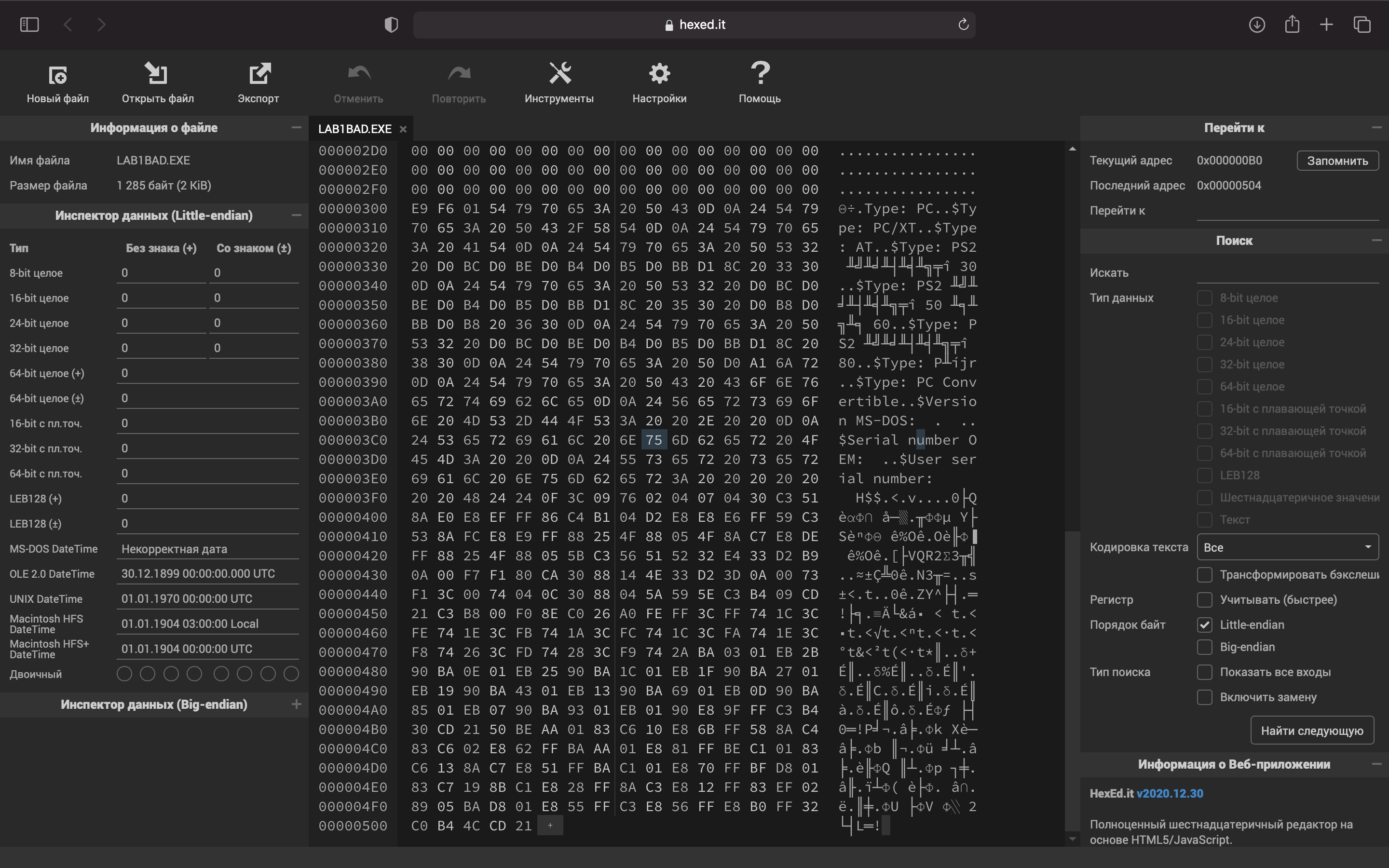
Сегмент с кодом и данными начинается с 0h. При загрузке модуля устанавливается смещение 256 байт.





1. Какова структура файла «плохого» EXE? С какого адреса располагается код? Что располагается с адреса 0?

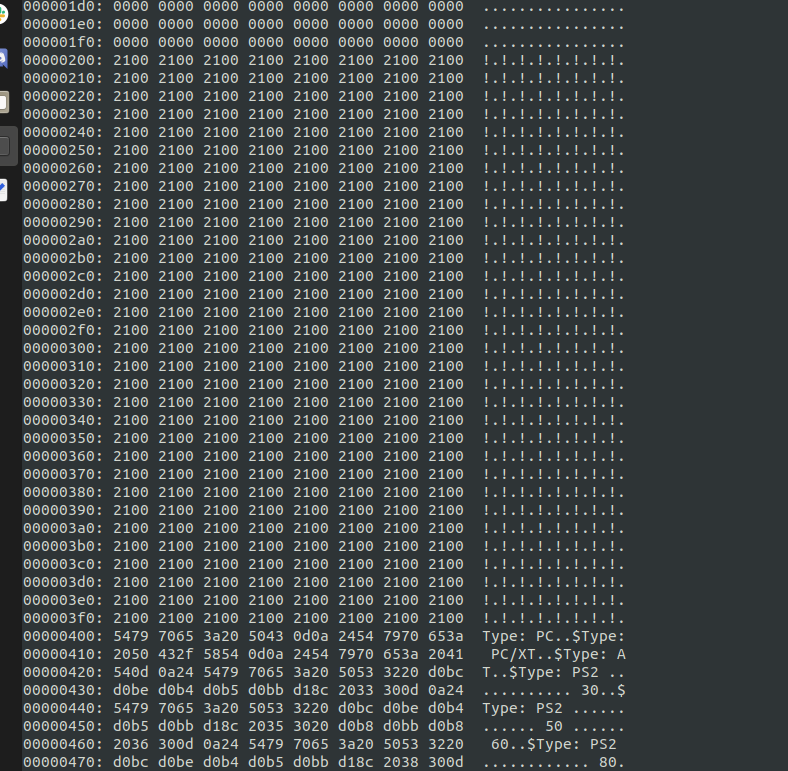
Код и данные расположены в одном сегменте. Начинается сегмент с кодом и данными с адреса 300h. С 0h идет таблица настроек.



1. Какова структура «хорошего» EXE? Чем он отличается от файла «плохого» EXE?

EXE-файл имеет поделенные на сегменты стек, данные и код, может иметь неограниченный размер, имеет в начале заголовок, который используется при загрузке этого модуля. Заголовок содержит данные и сигнатуру, таблицу для настройки адресов.

Плохой EXE-файл не имеет разделения сегментов кода и данных. Плохой EXE-файл имеет смещение 300h, поскольку изначально смещение 100h, а при создании EXE появляется смещение 200h для модуля PSP. У хорошего EXE выделяется сегмент под стек, поэтому итоговое смещение у хорошего EXE в данной программе – 400h.



**Загрузка COM модуля в основную память:**

1. Какой формат загрузки модуля COM? С какого адреса располагается код?

Сначала определяется адрес сегмента оперативной памяти, в котором достаточно места для загрузки программы, затем считывается COM-файл с диска и помещается в память, начиная с PSP:0100h. Сегментные регистры CS, DS, ES и SS будут указывать на PSP (48DD). SP будет указывать на конец PSP, 00H помещен в стек, IP содержит 100H из-за команды JMP PSP:100h.

1. Что располагается с адреса 0?

Сегмент PSP, размер сегмента 100h.

1. Какие значения имеют сегментные регистры? На какие области памяти они указывают?

Регистры CS, DS, ES и SS указывают на PSP, имеют значения 48DD.

1. Как определяется стек? Какую область памяти он занимает? Какие адреса?

Стек генерируется автоматически при загрузке COM-программы. SS – на начало – 0h, SP – на конец – FFFEh, адрес расположен в диапазоне 0h – FFFEh.

**Загрузка «хорошего» EXE модуля в основную память:**

1. Как загружается «хороший» .EXE? Какие значения имеют сегментные регистры?

Начинается загрузка с адреса PSP:0100h. Считывается информация PSP, выполняется перемещение адресов сегментов, DS и ES устанавливаются на начало PSP, SS устанавливается на начало сегмента стека, CS на начало сегмента кода. В IP загружается смещение до точки входа в программу, оно берется из метки после директивы END.

CS = 490D

SS = 48ED

ES = 48DD

DS = 48DD

1. На что указывают регистры DS и ES?

Регистры DS и ES указывают на начало сегмента PSP.

1. Как определяется стек?

При помощи директивы .STACK, также задается размер стека. Регистр SS будет указывать на начало сегмента стека, а SP на конец.

1. Как определяется точка входа?

Точка входа определяется при помощи директивы END.