**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. 7383 |  | Зуев Д.В. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2019

**Постановка задачи.**

**Цель работы:** исследование различий в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

**Реализация задачи:**

В данной работе были написаны и использовались готовые следующие функции:

TETR\_TO\_HEX – вспомогательная функция для функции BYTE\_TO\_HEX

BYTE\_TO\_HEX – переводит байт AL в два символа шестнадцатеричного числа в AX.

WRD\_TO\_HEX – переводит шестнадцатиразрядное число, расположенное в AX в шестнадцатеричную систему счисления.

BYTE\_TO\_DEC – переводит число в десятеричную систему счисления.

PRINT – вызывает прерывание программы для печати строки.

FOUND\_PC\_TYPE – используя исходные данные, определяет и выводит в консоль тип PC.

FOUND\_VERSION\_OS – с помощью функции 30H определяет и выводит в консоль номер основной версии ОС, номер модификации ОС, серийный номер OEM и серийный номер пользователя

Данная утилита действует следующим образом:

1. Берет код по адресу 0F000:0FFFEh и сравнивает его с соответствующим ему по табл. 1 типом IBM PC.

Таблица 1 – Соответствие кода и типа PC.

|  |  |
| --- | --- |
| Тип PC | Код |
| PC | FF |
| PC/XT | FE, FB |
| AT | FC |
| PS2 model 30 | FA |
| PS2 model 50 or 60 | FC |
| PS2 model 80 | F8 |
| PCjr | FD |
| PC Convertible | F9 |

1. Выводит тип PC в консоль.
2. Выполняет функцию 30h с помощью прерывания int 21h. Выходными параметрами данной функции являются:

AL – номер основной версии.

AH – номер модификации.

BH – серийный номер OEM.

BL:CX – 24-битовый серийный номер пользователя.

1. Полученные на предыдущем шаге данные переводятся в шестнадцатеричную систему счисления и выводятся соответствующими сообщениями в консоль.

В ходе данной работы были написаны программы на языке Ассемблера для .COM и .EXE загрузочных модулей.

Результатом работы «плохого» и «хорошего» .EXE модулей, а также .COM модуля представлены на рисунках 1 – 3.

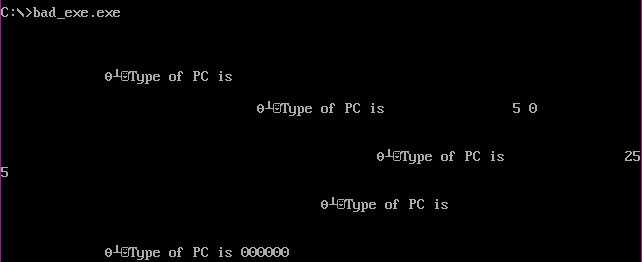


Рисунок 1 - Выполнение "плохого" .EXE модуля

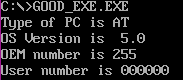


Рисунок 2 - Выполнение "хорошего" .EXE модуля

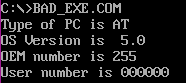


Рисунок 3 - Выполнение .COM модуля

В данной работе также были просмотрены шестнадцатеричные коды исполняемых файлов. Скриншоты представлены на рисунках 4 – 8.

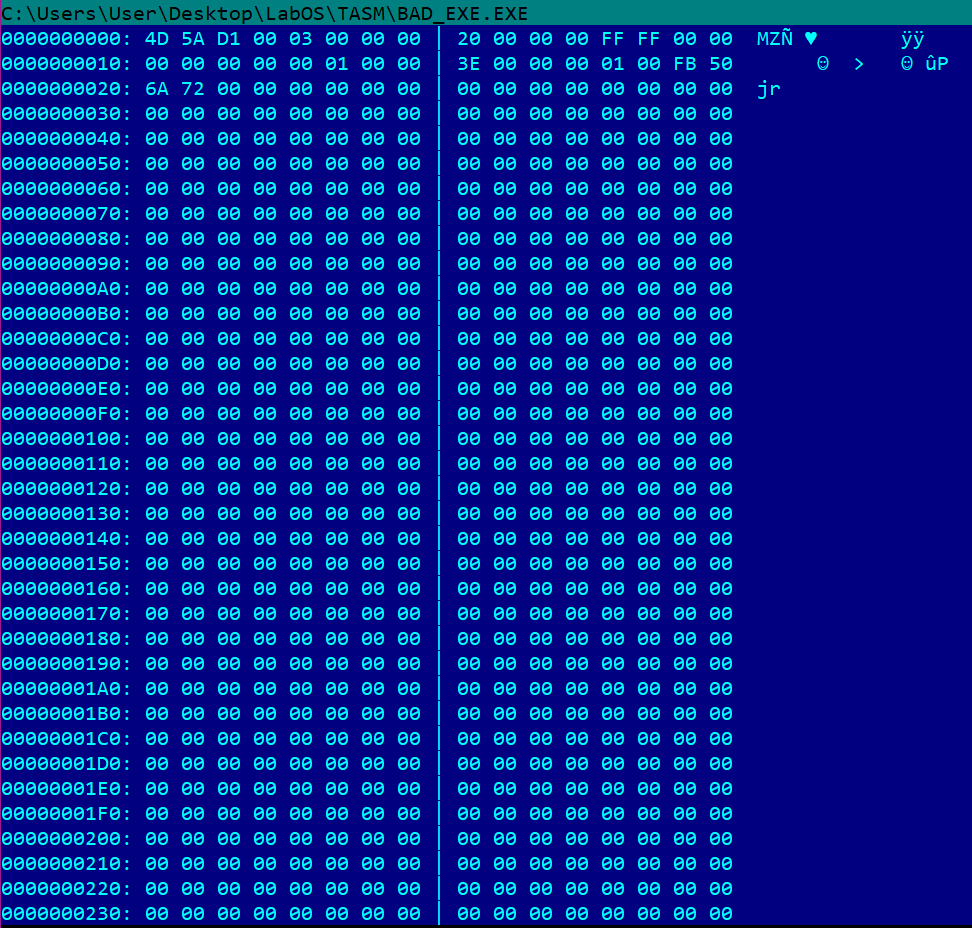


Рисунок 4 - Начало структуры "плохого" .EXE файла

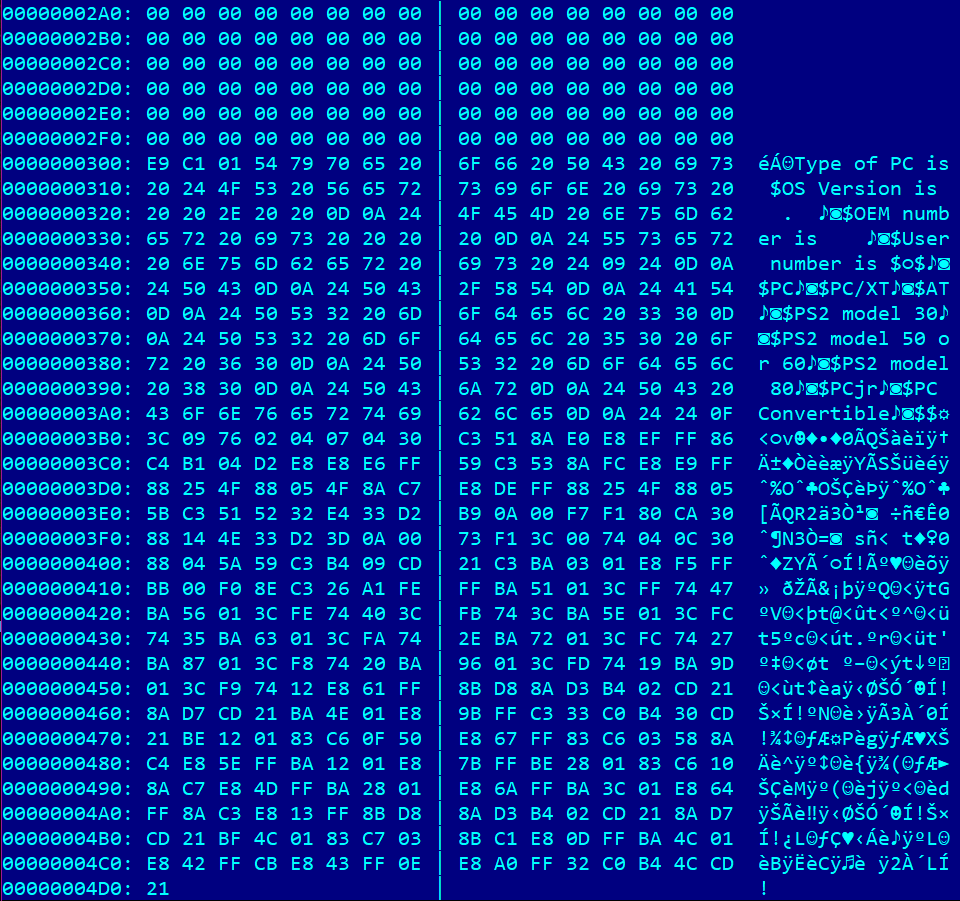


Рисунок 5 - Конец структуры "плохого" .EXE файла

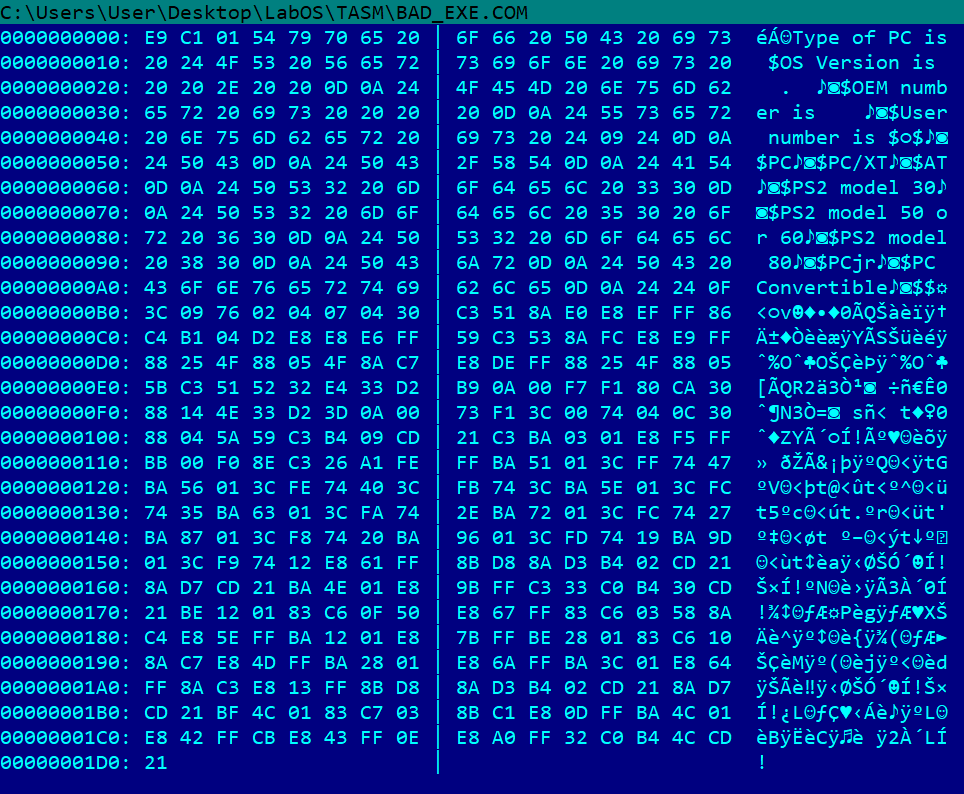


Рисунок 6 - Структура .COM файла

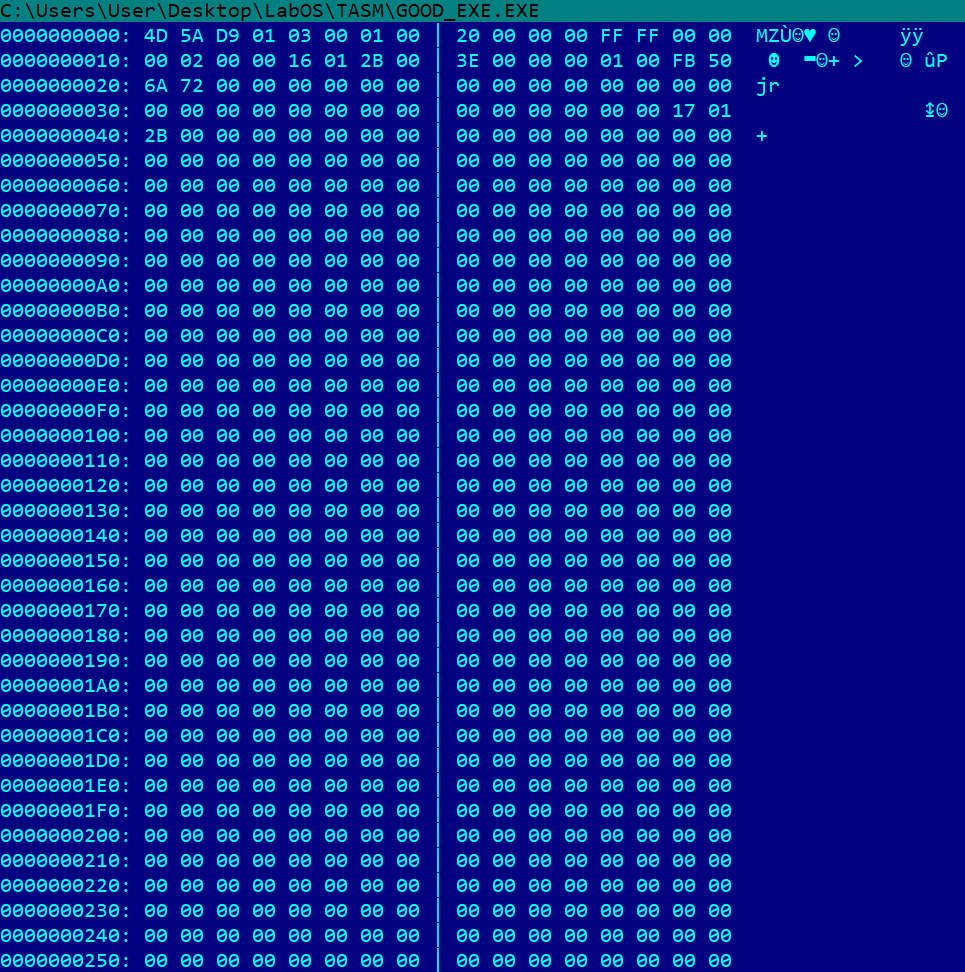


Рисунок 7 - Начало структуры "хорошего" .EXE файла

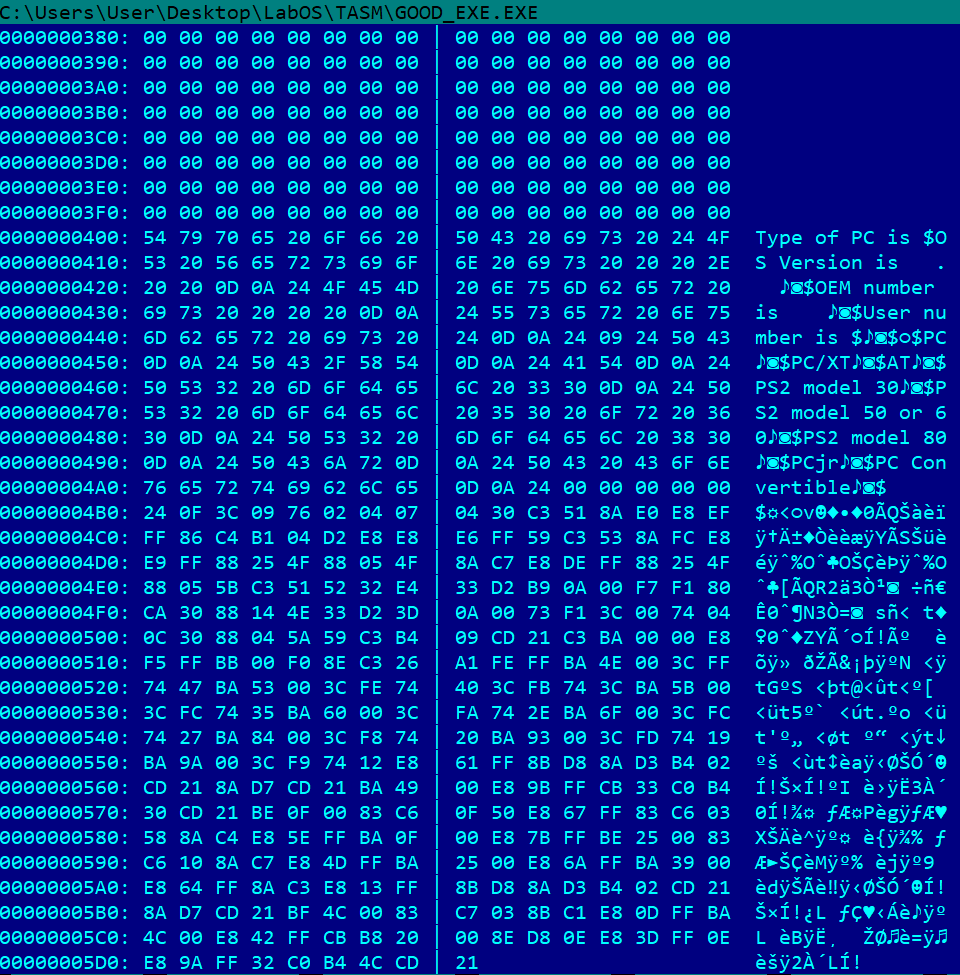


Рисунок 8 - Конец структуры "хорошего" .EXE файла

Скриншоты отладчика для исполняемых файлов представлены на рисунках 9 – 11.

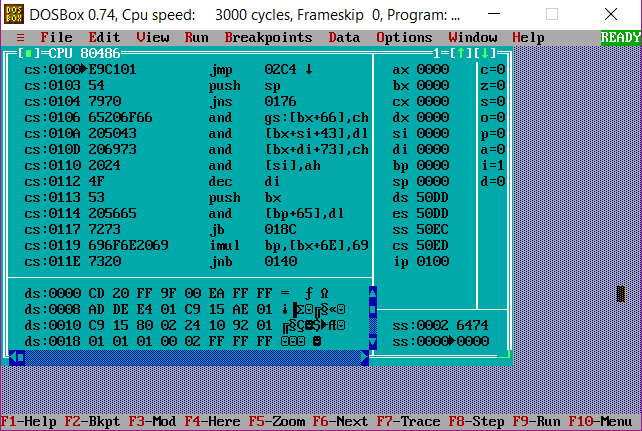


Рисунок 9 - Отладка "плохого" .EXE файла

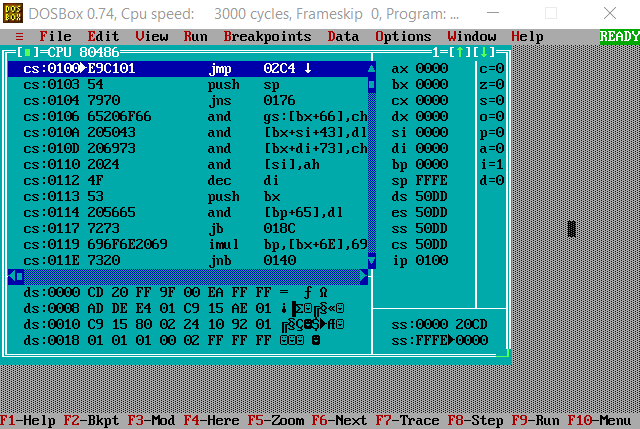


Рисунок 10 - Отладка .COM файла

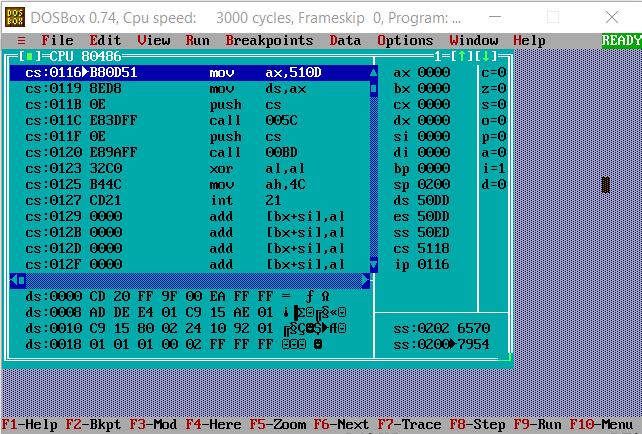


Рисунок 11 - Отладка "хорошего" .EXE файла

**Выводы.**

В ходе выполнения данной лабораторной работы была написана программа в двух версиях: для .COM и для .EXE модулей, определяющая тип IBM PC, версию операционной системы, номер OEM и серийный номер пользователя. Исследованы различия в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

**Контрольные вопросы.**

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

1. Сколько сегментов должна содержать COM-программа?

*COM-программа содержит 1 сегмент.*

1. EXE-программа?

*EXE-программа содержит любое количество сегментов.*

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

*В COM-программе обязательно должна быть директива ORG 100h, резервирующая 256 байт для PSP.*

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

*В COM-программе нельзя использовать команды, использующие адреса сегментов.*

**Отличия форматов файлов COM и EXE модулей:**

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

*Выполнимый файл .COM включает в себя программные строки вместе со строками данных. Код располагается с адреса 0h.*

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

*Так же как и в .COM файле в «плохом» .EXE данные и код располагаются в одном сегменте. Код располагается с адреса 300h. С адреса 0h располагается заголовок, состоящий из 1 блока размером 200h, хранящий в себе информацию, необходимую для правильной настройки регистров процессора и самой программы при ее загрузке в память. Так же 100h зарезервировано директивой ORG 100h.*

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

*В отличие от «плохого» EXE в «хорошем» EXE данные, стек и код расположены в разных сегментах.*

**Загрузка COM модуля в основную память:**

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

*Система выделяет для программы 64 Кбайт адресного пространства. Первые 256 байт отводятся под PSP, на который указывают регистры CS, DS, ES, SS. Далее располагается код программы и данных. По рис. 10 видно, что код располагается с адреса CS+IP=51DD. Оставшуюся память занимает стек. Регистр SP хранит значение вершины стека, равное FFFEh.*

1. Что располагается с адреса 0?

*С адреса 0 располагается PSP.*

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

*Значения сегментных регистров одинаковы и равны 50DD (см. рис. 10). Сегментные регистры указывают на PSP.*

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

*Стек определяется сегментным регистром стека SS и адресом вершины стека SP. Занимает область памяти расположенную между SS и SP, то есть 0h – FFFEh.*

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

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

*Регистры DS и ES равны 50DD и указывают на PSP. Регистр CS равен 5118 и указывает на сегмент кода (см. рис. 11). Регистр SS равен 50ED и указывает на сегмент стека.*

1. На что указывают регистры DS и ES?

*Регистры DS и ES указывают на PSP.*

1. Как определяется стек?

*Стек определяется с помощью директивы сегментации SEGMENT и атрибута комбинирования сегментов определяющего сегмент стека STACK.*

1. Как определяется точка входа?

*Точка входа определяется директивой END.*