**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Операционные системы»**

Тема: «Исследование структур загрузочных модулей»

|  |  |  |
| --- | --- | --- |
| Студент гр. 6381 |  | Попов Н.В. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

**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.

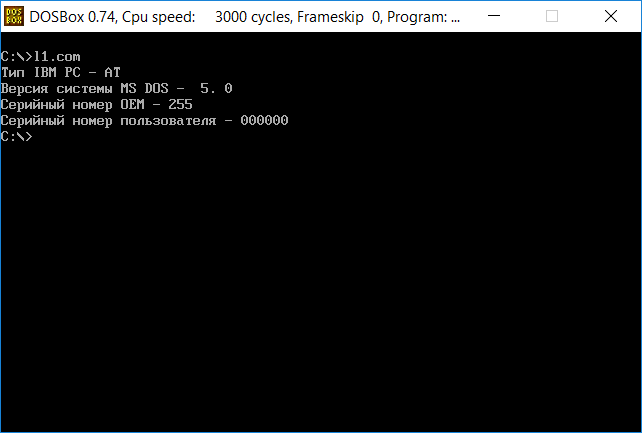
Описание функций и структур данных:

1. FIND\_TYPE\_PC: печатает тип ОС. Если определить не удалось, выводит соответствующее сообщение и полученное значение в 16ричной с/с из предпоследнего байта ROM BIOS.
2. FIND\_VERSION\_MS\_DOS: печатает версию ОС, серийный номер OEM и серийный номер пользователя
3. TETR\_TO\_HEX: вспомогательная функция для работы функции BYTE\_TO\_HEX.
4. BYTE\_TO\_HEX: переводит число AL в коды символов 16-ой с/с, записывая получившееся в bl и bh
5. WRD\_TO\_HEX: переводит число AX в строку в 16-ой с/с, записывая получившееся в di, начиная с младшей цифры
6. BYTE\_TO\_DEC: переводит байт из AL в десятичную с/с и записывает получившееся число по адресу si, начиная с младшей цифры

L1.exe – «плохой» EXE, L1exe.exe – «хороший»

Последовательность действий, выполняемых утилитой:

1. Определение и печать типа ОС с помощью функции FIND\_TYPE\_PC
2. Определение и печать версии ОС, серийного номера OEM и серийного номера пользователя с помощью функции FIND\_VERSION\_MS\_DOS



**Заключение.**

В процессе выполнения данной лабораторной работы были исследованы различия в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

**Ответы на контрольные вопросы.**

**Отличия исходных текстов COM и EXE программ**

1. Сколько сегментов должна содержать COM-программа?

Com-программа содержит один сегмент, т. е. ее размер составляет 64 Кб.

1. EXE-программа?

EXE-программа содержит не менее 1 сегмента, поэтому ее размер не ограничен 64 КБ.

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

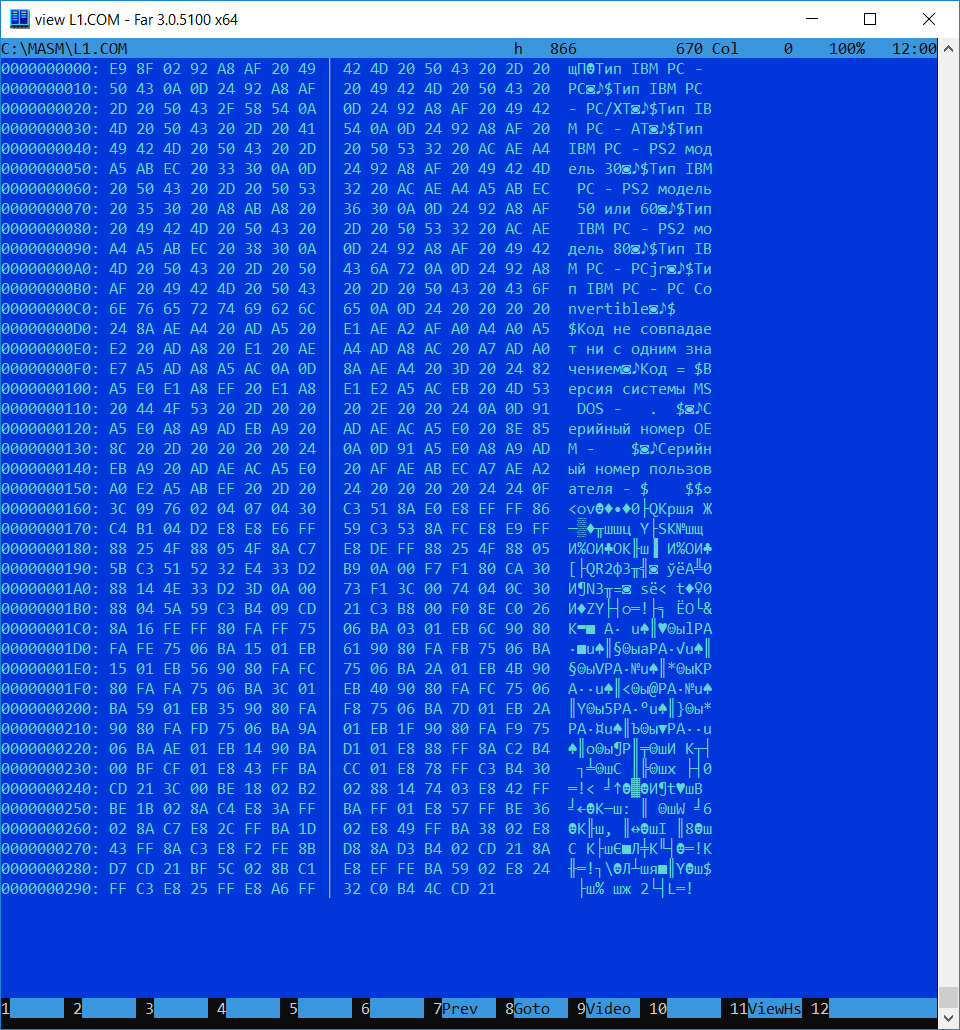
В тексте COM-программы обязательно должна быть директива ORG 100h, сдвигающая адресацию в программе на 256 байт, т.к. в первых 256 байтах программы находится PSP.

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

В COM-программе нельзя использовать команды вида mov register,segment и команды, содержащие дальнюю(far) адресацию, т.к. в этих командах используется relocation table, в которой содержатся адреса сегментов. Такой таблицы в COM-программах нет, поэтому и эти команды не действительны.

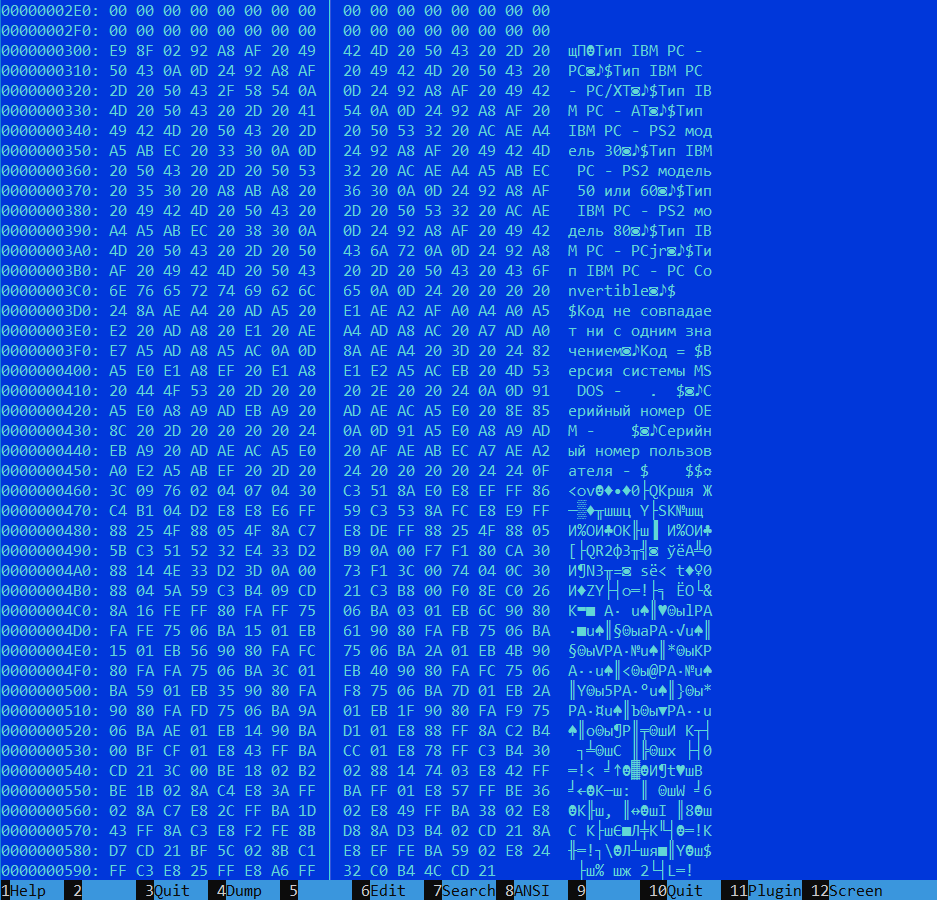
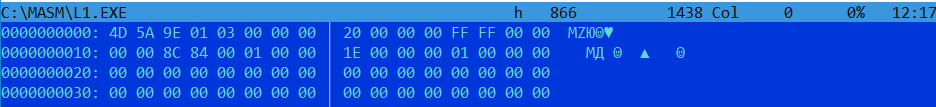
**Отличия форматов файлов COM и EXE модулей**

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



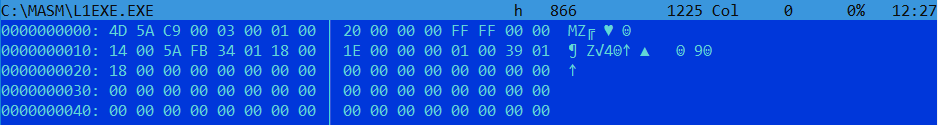
COM-файл содержит только данные и код. В файле код располагается с нулевого адреса.

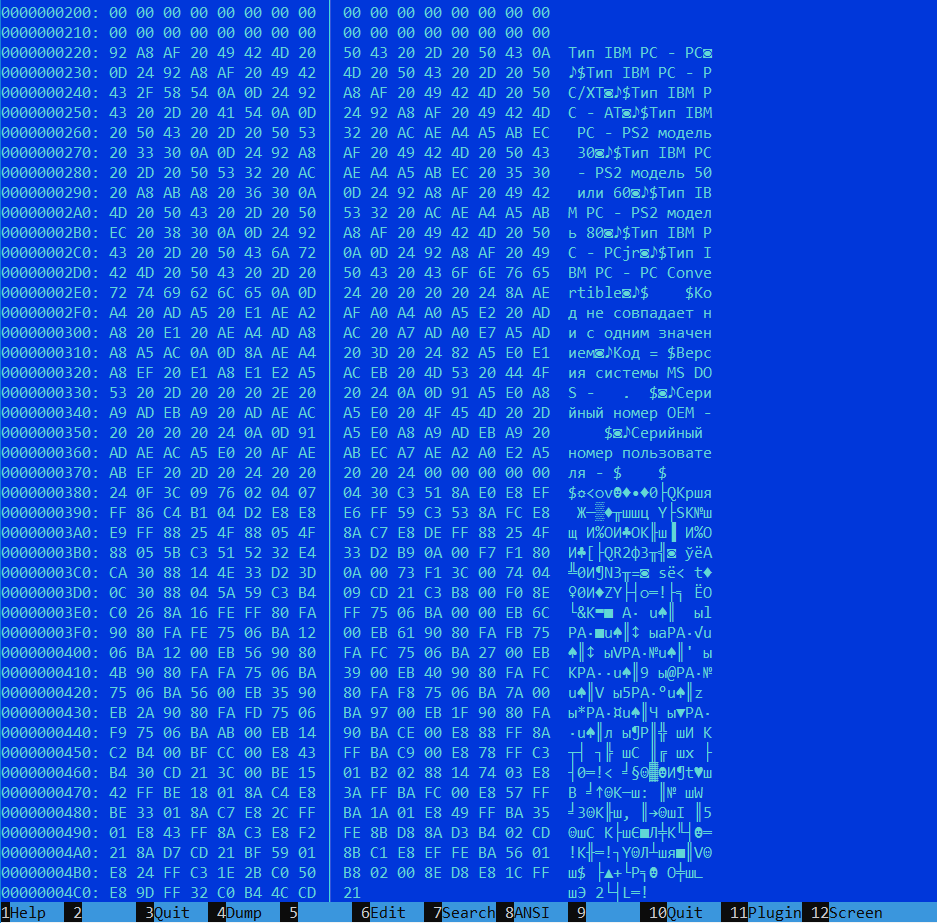
1. Какова структура файла «плохого» EXE? С какого адреса располагается код? Что располагается с адреса 0?



В «плохом» EXE-файле код и данные не разделены по сегментам, а перемешаны: перед данными по адресу 300h находится метка перехода E9 8F 02. Код располагается с адреса 300h, т.к. заголовок занимает 200h байт и команда ORG 100h «сдвигает» код на дополнительные 100h. С нулевого адреса располагается заголовок. В первых двух байтах можно увидеть символы MZ, означающие, что формат файла – 16-битный и его следует запускать в соответствии со структурой EXE-файлов. За заголовком следует relocation table.

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

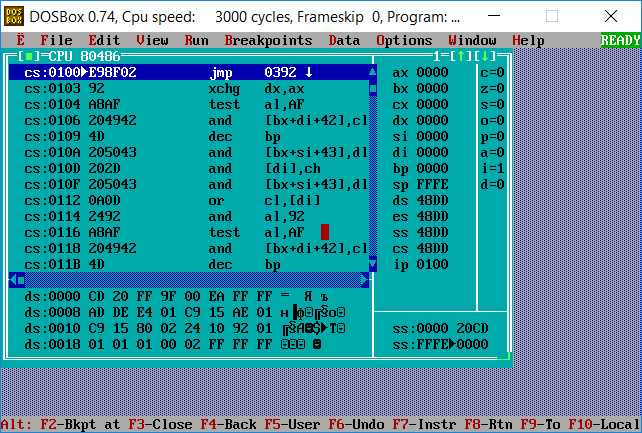




В отличие от «плохого» EXE-файла, в «хорошем» код, стек и данные выделены в отдельные сегменты. Код начинается с 220h байта, т.к. первые 200 байт содержат заголовк, а 200h-220h байты занимает сегмент стека.

Для «хорошего» EXE в директиве org 100h нет необходимости, т.к. загрузчик автоматически расположит программу после PSP.

**Загрузка COM модуля в основную память**



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

Формат загрузки модуля COM:

1. Выделение сегмента памяти для модуля
2. Установка всех сегментных регистров на начало выделенного сегмента памяти
3. Построение в первых 100h байтах памяти PSP
4. Загрузка содержимого COM-файла и присваивание регистру IP значения 100h.
5. Регистр SP устанавливается в конец сегмента
6. Что располагается с адреса 0?

С нулевого адреса располагается PSP.

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

Все сегментные регистры равны в данном случае 48DD и указывают на начало PSP.

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

Стек занимает весь сегмент COM-программы, его начало находится в конце сегмента. SS указывает на начало сегмента, а SP=FFFEh – на его конец. Стек может дойти до кода/данных программы при достаточном количестве элементов.

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

1. Как загружается «хороший» EXE? Какие значения имеют сегментные регистры? На что указывают регистры DS и ES?

DS и ES имеют значение 48DD и указывают на начало PSP, SS=48ED – начало сегмента стека, CS=4905 – начало сегмента команд.

1. Как определяется стек?

Стек выделяется в программе нужным размером.

1. Как определяется точка входа?

Точка входа в программу определяется с помощью директивы END.