**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

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6381 |  | Шевелева А.М. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

**Цель работы**

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

**Постановка задачи**

Требуется написать текст исходного .COM модуля, который определяет тип PC и версию системы. 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 (Original Equipment Manufacturer) и серийным номером пользователя. Полученные строки выводятся на экран.

Далее необходимо отладить полученный исходный модуль и получить «хороший» .COM модуль, а также необходимо построить «плохой» .EXE, полученный из исходного текста для .COM модуля.

Затем нужно написать текст «хорошего» .EXE модуля, который выполняет те же функции, что и модуль .COM, далее его построить, отладить и сравнить исходные тексты для .COM и .EXE модулей.

**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 30Н прерывания 21Н. Входным параметром является номер функции в АН:

MOV AH, 30h

INT 2Ih

Выходными параметрами являются:

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

АН - номер модификации

ВН - серийный номер OEM (Original Equipment Manufacturer)

BL:CX - 24-битовый серийный номер пользователя.

**Функции и структуры программы**

Процедуры программы представлены в таблице 1.

|  |  |
| --- | --- |
| WRITE | Вывод строки на экран. |
| OSVER | Определение версии операционной системы, номера модификации, серийного номера OEM и серийного номера пользователя. |
| PCTYPE | Определение типа IBM PC. |
| TETR\_TO\_HEX | Перевод из четверичной системы счисления в шестнадцатеричную. |
| BYTE\_TO\_HEX | Перевод байта в al в два символа шестнадцатеричного числа в ax. |
| WRD\_TO\_HEX | Перевод в шестнадцатеричную систему счисления 16-ти разрядного числа. |
| BYTE\_TO\_DEC | Перевод из двоичной системы счисления в десятичную. |

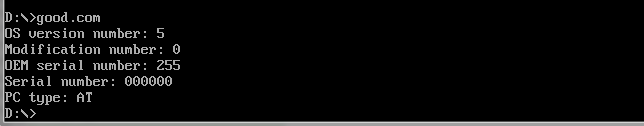
Таблица 1. Процедуры программы

**Последовательность действий, выполняемых утилитой**

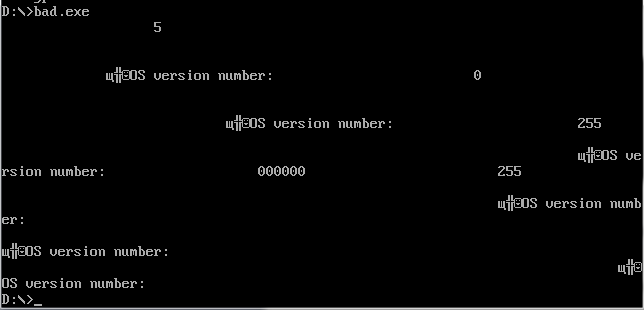
1. Определение и вывод на экран версии операционной системы.
2. Определение и вывод на экран номера модификации.
3. Определение и вывод на экран серийного номера OEM.
4. Определение и вывод на экран серийного номера пользователя.
5. Определение и вывод на экран типа IBM PC.

**Результаты запуска программ**

Результат запуска файла good.com представлен на рисунке 1.

Рисунок 1. Результат запуска файла good.com

Результат запуска файла bad.exe представлен на рисунке 2.

Рисунок 2. Результат запуска файла bad.exe

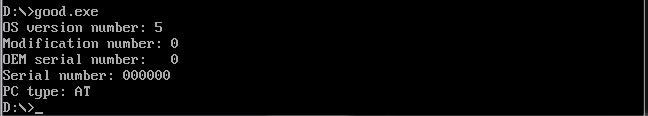
Результат запуска файла good.exe представлен на рисунке 3.

Рисунок 3. Результат запуска файла good.exe

**Ответы на контрольные вопросы**

Отличия исходных текстов COM и EXE программ:

1. Сколько сегментов может содержать COM программа?

Ответ: COM программа может содержать ровно 1 сегмент.

1. EXE - программа?

Ответ: EXE программа может иметь любое количество сегментов.

1. Какие директивы должны обязательно быть в тексте COM – программы?  
   Ответ: в программе обязательно должна быть директива ORG 100h, которая устанавливает значение программного счетчика в 100h, так как при загрузке СОМ-файла в память DOS занимает первые 256 байт блоком данных PSP и располагает код программы после этого блока.

Также в тексте COM-программы обязательно должна быть директива ASSUME, которая указывает, в значение какого сегмента установлен данный сегментный регистр. В данном случае она указывает, что сегмент кода и данных установлен в сегмент TESTPC. Если закомментировать ASSUMEи скомпилировать полученный код, то получим ошибку, которая сообщает об отсутствии или недоступности CS, то есть программа не сможет обнаружить начало сегмента кода.

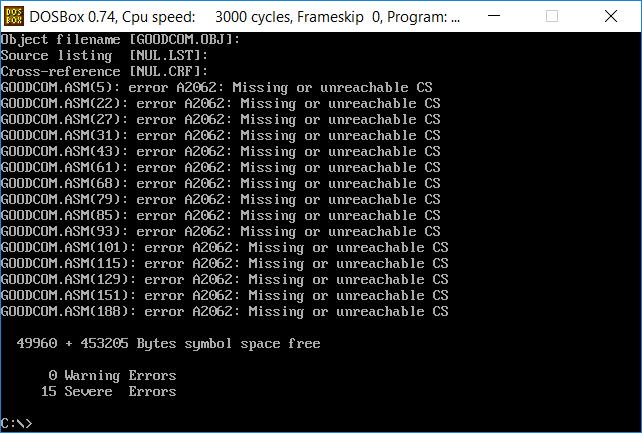


Рисунок 4. Программа без директивы ASSUME

1. Все ли форматы команд можно использовать в COM программе?  
   Ответ: Не все. В COM программах запрещены межсегментные переходы и межсегментные вызовы. Нельзя использовать команды типа mov[регистр], seg[сегмент].

Отличия форматов файлов COM. и EXE. модулей:

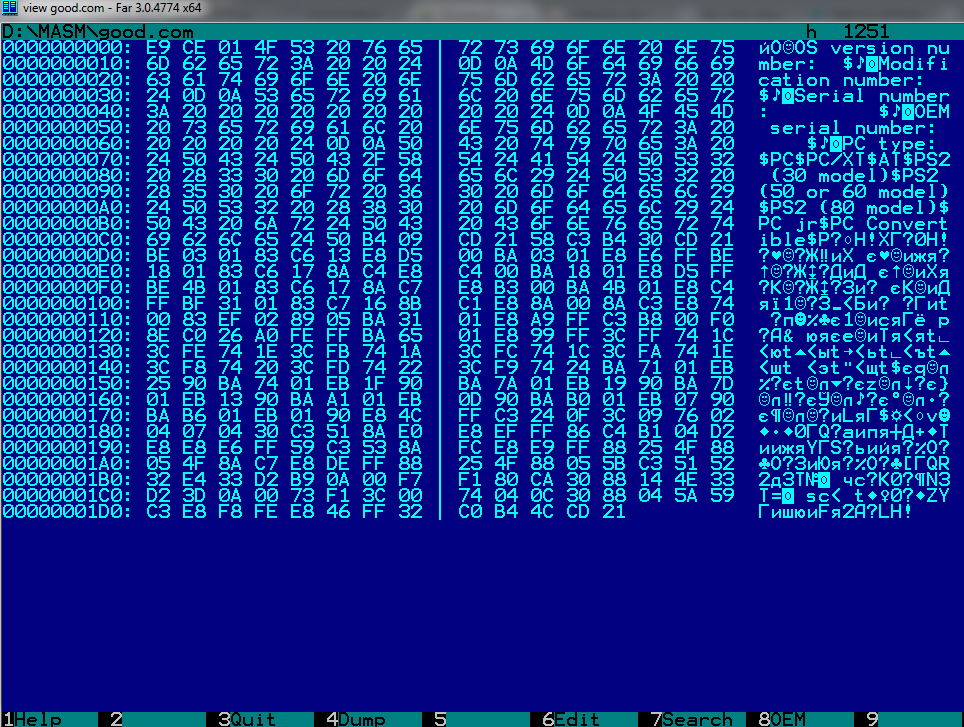
1. Какова структура файла СОМ? С какого адреса располагается код?  
   Ответ: COM файл содержит 1 сегмент, в котором размещаются данные и машинные команды. Код начинается с адреса 0h, но при загрузке модуля устанавливается смещение в 100h.

Рисунок 5. «Хороший» .COM модуль в Far

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

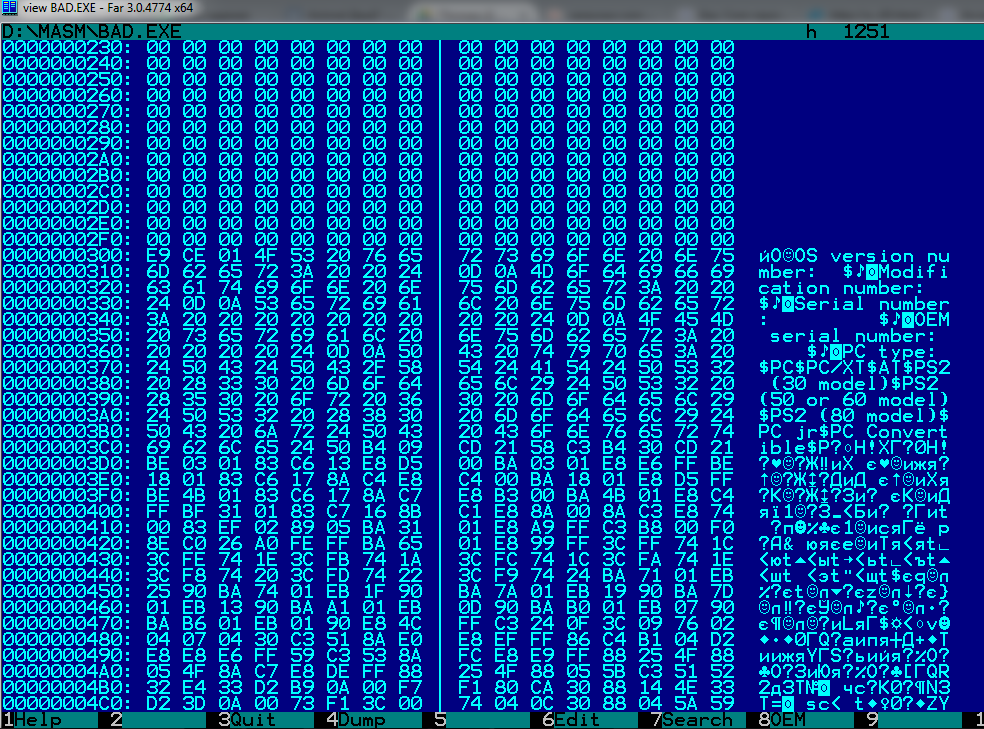
Ответ: в файле «хорошего» EXE содержится сегмент стека, сегмент данных и сегмент кода. Отличие от «плохого» в наличии 3 сегментов. Кроме того, код начинается с 200h, а не с 300h, как в «плохом» ЕХЕ-файле, так как в этом файле отсутствует директива ORG 100h, резервирующая пространство для заголовка;

Рисунок 6. «Плохой» .EXE модуль в Far

1. Какова структура «хорошего» EXE? Чем он отличается от файла «плохого»EXE?  
   Ответ: в файле «хорошего» EXE содержится сегмент стека, сегмент данных и сегмент кода. Отличие от «плохого» в наличии 3 сегментов. Кроме того, код начинается с 200h, а не с 300h, как в «плохом» ЕХЕ-файле, так как в этом файле отсутствует директива ORG 100h, резервирующая пространство для заголовка;

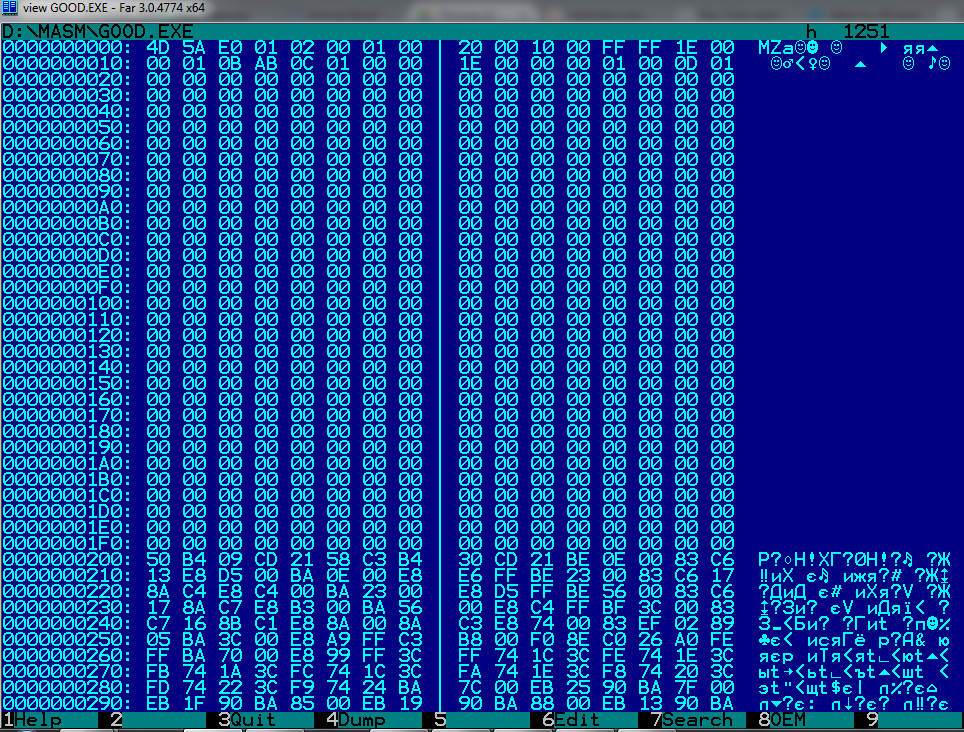
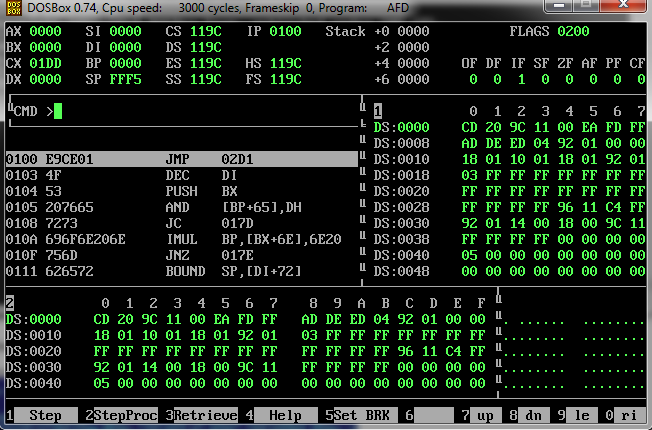


Рисунок 7. «Хороший» .EXE модуль в Far

Загрузка COM. модуля в основную память:

1. Какой формат загрузки модуля СОМ? С какого адреса располагается код?  
   Ответ: После загрузки файла для COM программы сегментные регистры устанавливаются на начало PSP. Указатель команд IP устанавливается на 100h.

Код располагается с адреса 100h.

Рисунок 8 .COM модуль в TD

1. Что располагается с адреса 0?

Ответ: Сегмент PSP.

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

Ответ: Все сегментные регистры (CS, DS, ES, SS) указывают на начало PSP.  
Они имеют значение 119C.

1. Как определяется стек? Какую область памяти он занимает? Какие адреса?  
   Ответ: Для COM-файла DOS автоматически определяет стек и устанавливает одинаковый общий сегментный адрес во всех четырех сегментных регистрах. DOS устанавливает в регистре SP адрес конца сегмента - FFFE. Это будет вершина стека. Стек пополняется снизу вверх (от старших адресов к младшим).

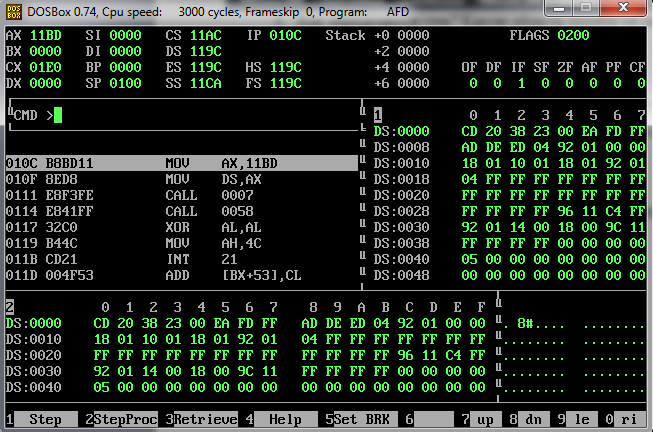
Загрузка «хорошего» EXE. Модуля в основную память:

Рисунок 9. «Хороший» .EXEмодуль в TD

1. Как загружается «хороший» ЕХЕ? Какие значения имеют сегментные регистры?  
   Ответ: Сегментные регистры DS и ES устанавливаются на начало PSP, CS – на начало сегмента команд, SS – на начало сегмента стека. IP загружается смещение точки входа в программу, которая берётся из метки после директивы END.
2. На что указывают регистры DS и ES?

Ответ: Регистры DS и ES указывают на начало сегмента PSP.

1. Как определяется стек?

Ответ: Стек определяется с помощью директивы .stack.

1. Как определяется точка входа?

Ответ: Точка входа определяется с помощью директивы END (в качестве ее операнда выступает адрес первой выполнимой программной строки).

**Заключение:**

В ходе работы было проведено исследование различий в структурах исходных текстов модулей .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.