**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 |  | Кирсанов А.Я. |
| 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.

**Ход работы.**

На языке ассемблера написаны исходные .COM и .EXE модули, определяющие тип PC и версию системы.

**Сведения о функциях и структурах данных.**

В модулях описаны следующие функции:

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

**WRD\_TO\_HEX** – перевод в 16-ти разрядного числа в шестнадцатеричную систему счисления. В AX – число, в DI – адрес последнего символа.

**BYTE\_TO\_DEC** – перевод значения регистра AL в его запись в десятичной системе счисления, SI – адрес поля младшей цифры.

**PRINT** – вызывает прерывание 21h для вывода строки на экран.

**FIND\_OS\_VERSION** – определяет версию системы в виде xx.yy, где xx – номер основной версии, а yy – номер модификации в десятичной системе счисления.

**FIND\_PC\_TYPE** – определяет тип PC из предпоследнего байта ROM BIOS.

**Последовательность действий, выполняемых утилитой.**

Ассемблерная программа читает содержимое предпоследнего байта ROM BIOS, находящегося по адресу 0F000:0FFFEh. Затем определяет тип PC по   
табл. 1.

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 |

Для определения версии MS DOS используется функция 30H прерывания 21H. Входными параметрами 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.

Программа выводит в консоль тип PC, затем версию ОС, номер OEM и номер пользователя.

Компиляция производилась с помощью компилятора TASM 5.1 и линковщика TLINK.

На рисунках 1, 2, 3 соответственно представлены результаты работы «хорошего» .EXE, «плохого» .EXE и .COM скомпилированных программ.

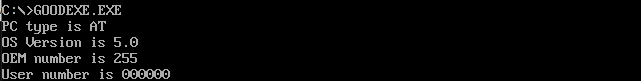


Рисунок 1 – Выполнение «хорошего» .EXE.

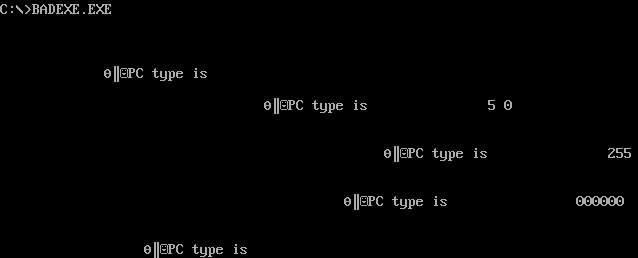


Рисунок 2 – Выполнение «плохого» .EXE.

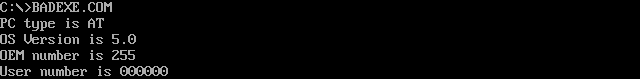


Рисунок 3 – Выполнение .COM.

**Вывод.**

Были написаны модули .COM и .EXE на языке ассемблера, выводящие информацию о типе PC, версии ОС, номеров пользователя и OEM. Исследованы различия в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

**Контрольные вопросы по лабораторной работе №1.**

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

1. Сколько сегментов должна содержать COM-программа?

* Один сегмент. Сегментные регистры CS и DS будут соответствовать этому сегменту.

1. EXE-программа?

* Любое число сегментов.

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

* Директива ORG 100H, резервирующая 256 байт для PSP, а также директива ASSUME, указывающая ассемблеру размещение сегментных регистров.

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

* Нельзя использовать команды, содержащие адреса сегментов. Это связано с тем, что в COM-программе отсутствует таблица настроек, которая указывает, какие абсолютные адреса при загрузке должны быть изменены, так как до загрузки неизвестно, куда будет загружена программа.

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

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

* В COM файле данные и код располагаются в одном сегменте. Размер файла COM не превышает 64 Кбайт. Первые 256 байт файла отведены под PSP директивой ORG 100h. При выполнении программы код начинается с адреса IP = 0100h. Оставшийся объем памяти отводится под стек. На диске код располагается с адреса 0h (см. рис. 4).

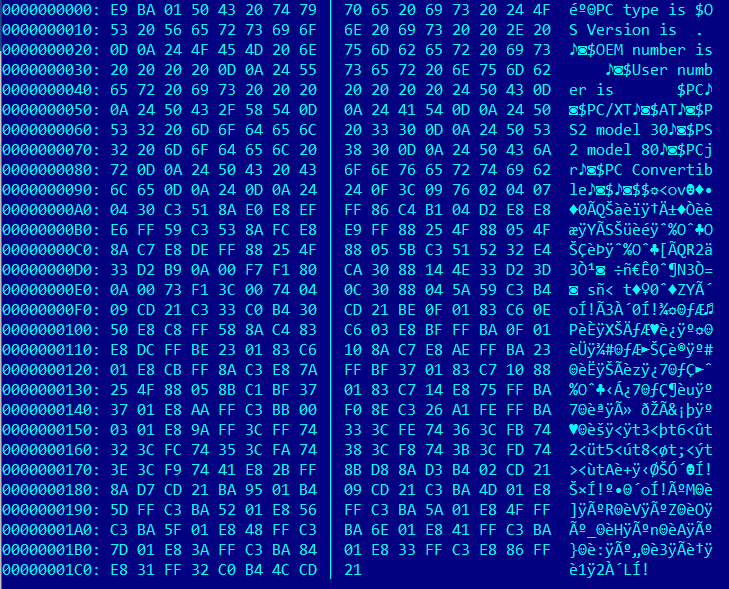


Рисунок 4 – Структура файла COM.

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

* Как и в COM файле в «плохом» EXE код и данные располагаются в одном сегменте. С адреса 0 начинается заголовок, состоящий из одного блока размером 512 байт, в котором хранится информация, необходимая системе для правильной настройки регистров процессора и самой программы при загрузке её в память   
  (см. рис. 5). Далее идет 256 байт PSP и с адреса IP = 0300h располагается код (см. рис. 6).

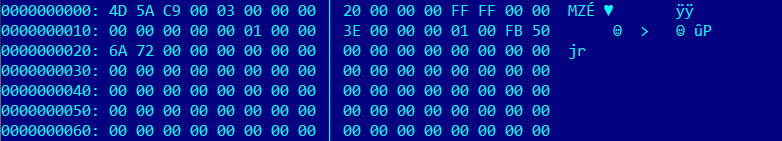


Рисунок 5 – Структура «плохого» EXE.



Рисунок 6 – Расположение кода в «плохом» EXE.

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

* В обоих файлах с адреса 0 располагается заголовок с таблицей настроек. Сегмент кода в «плохом» EXE начинается с 300h, а в «хорошем» EXE с 400h. Это связано с тем, что в «хорошем» EXE мы выделили под сегмент стека 200h байт. А стек в свою очередь располагается с адреса 200h. В «плохом» EXE с адреса 200h располагаются зарезервированные ORG 100h и нет сегмента стека.

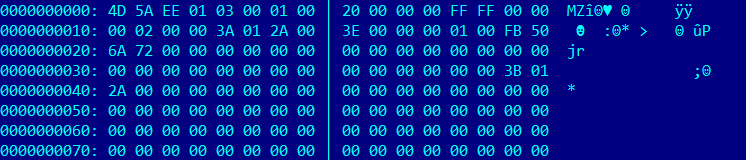


Рисунок 7 – Структура «хорошего» EXE.



Рисунок 8 – Расположение кода в «хорошем» EXE.

**Загрузка COM модуля в основную память.**

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

* При загрузке программы ей выделяется 64 Кбайт адресного пространства. Первые 256 байт под PSP, на который указывают все 4 сегментных регистра CS, DS, ES, SS. Далее идут программа и данные, далее стек. Указатель стека автоматически инициализируется числом FFFEh.

1. Что располагается с адреса 0?

* PSP

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

* Все 4 сегментных регистра указывают на начало PSP и имеют значения 50DD (см. рис. 9).

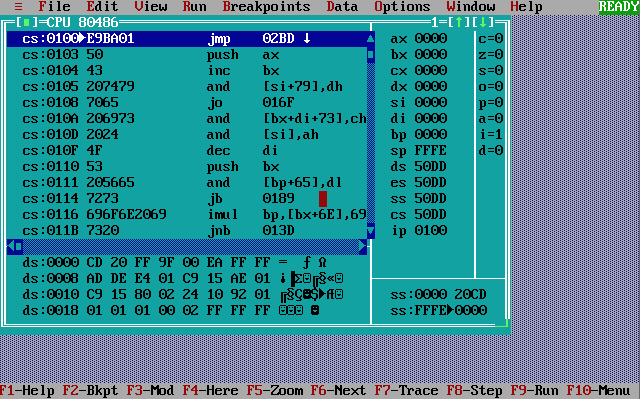


Рисунок 9 – Отладка COM модуля.

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

* Стек занимает всю память от начала сегмента данных (SS = 50DD – нашем случае) до конца выделенной памяти (SP = FFFE).

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

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

* В процессе загрузки выполнимого модуля программы в память система пристраивает к началу программы дополнительный сегмент – префикс PSP размером 256 байт. Система, загрузив программу в память, инициализирует сегментные регистры, так что регистры DS и ES указывают на начало PSP, CS – на начало сегмента команд, а SS – на начало сегмента стека. Регистры DS, ES имеют адреса 50DD, SS – 50ED, CS – 5117 (см. рис. 10).

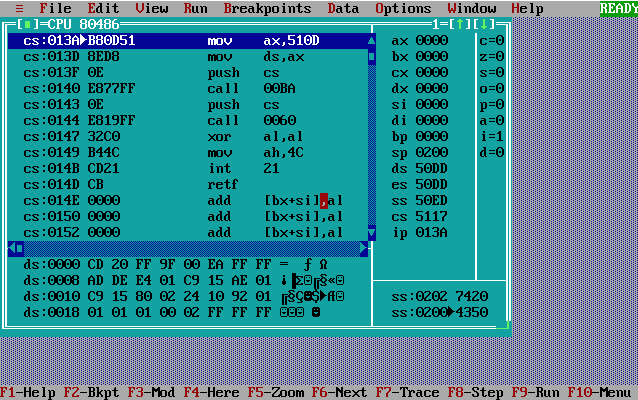


Рисунок 10 – Отладка «хорошего» EXE модуля.

1. На что указывают регистры DS и ES?

* Регистры DS и ES указывают на начало PSP

1. Как определяется стек?

* С помощью директивы SEGMENT в программе выделяется отдельный сегмент с параметром STACK. В SP хранится адрес, по которому расположена вершина стека. Регистр SS - хранит адрес сегмента стека.

1. Как определяется точка входа?

* Точка входа берется из операнда директивы END.