**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Математического Обеспечения ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Операционные системы»**

Тема: **Исследование структур загрузочных модулей.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6383 |  | Азаревич А.Д. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

# Цель работы

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

# Краткие теоретические сведения

Тип IBM РС хранится в байте по адресу 0Е000:0ЕЕЕЕ11, в предпоследнем байте ROM BIOS. Соответствие кода и типа в таблице:

РС FF

РС/ХТ FE, FB

АТ FC

РS2 модель 30 FA

РS2 модель 50 или 60 FC

РS2 модель 80 F8

PSjr: FD

РС Сonvertible F9

Для определения версии МS DOS следует воспользоваться функцией 30Н

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

МОV АH, 30h

INT 21h

Выходными параметрами являются:

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

АН - номер модификации

ВН - серийный номер ОЕМ (Оriginal Equipment Маnufacturer)

ВL:СХ - 24-битовый серийный номер пользователя.

# Интерфейс функций и структур управляющей программы

Функции:

TETR\_TO\_HEX – переводит число из младшей половины регистра AL в его символьное обозначение (помещается в AL);

BYTE\_TO\_HEX – переводит число из регистра AL в его символьное обозначение (помещается в AX);

WRD\_TO\_HEX – переводит число из регистра AX в его символьное обозначение (помещается в память, на конец которой указывает DI);

BYTE\_TO\_DEC – переводит шестнадцатеричное число из регистра AL в 10-ое (помещается в память, на конец которой указывает SI).

Структуры:

MODEL – строка, в которой хранится тип ПК;

PC, PCXT, ModAT, PS23, PS28, PCjr, PCC, El – типы IBM PC;

Version – версия MS DOS;

OEM – серийный номер OEM;

Number – серийный номер пользователя.

# Последовательность действий, выполняемых утилитой

Ассемблерная программа читает содержимое предпоследнего байта ROM ВIOS и определяет тип РС, сравнивая коды по таблице, и выводит строку с названием модели. Если код не совпадает ни с одним значением, то двоичный код переводиться в символьную строку, содержащую запись шестнадцатеричного числа и выводится на экран в виде соответствующего сообщения.

Затем определяется версия системы. Ассемблерная программа по значениям регистров АL и АН формирует текстовую строку в формате хх.уу, где хх — номер основной версии, а уу - номер модификации в десятичной системе счисления, формирует строки с серийным номером ОЕМ и серийным номером пользователя. Полученные строки выводятся на экран.

# Результаты исследования

Отличия исходных текстов COM и EXE программ:

1)COM-программа содержит ровно 1 сегмент.

2)EXE-программа содержит от 1 и более сегментов. Обычно выделяют 3 сегмента: под код, данные и стэк.

3)Обязательными директивами при разработке COM-программы являются директивы SEGMENT, ENDS, ASSUM, ORG.

SEGMENT и ENDS — определяют начало и конец сегмента памяти.

ASSUME — задает значения сегментных регистров перед началом работы программы.

ORG - устанавливает значение программного счетчика в 100h, потому что при загрузке СОМ-файла в память, DOS выделяет под блок данных PSP первые 256 байт (десятичное число 256 равно шестнадцатеричному 100h). Код программы располагается только после этого блока.

Также используется директива END, которая завершает программу, одновременно указывая, с какой метки должно начинаться ее выполнение.

4) В COM — программе нельзя использовать команды использующие адреса сегментов, например, mov ax,NAME\_SEGMENT. Это связано с тем, файл типа .COM не может предоставить загрузчику перечня всех сегментных ссылок.

Отличия форматов файлов COM и EXE модулей

1. Файл типа .COM представляет собой исходный код из файла типа .asm, из которого удалили все директивы, а команды перевели в 16-ный вид. Код располагается с 0-го байта (см. рис. 1).



Рис. 1 - .COM-файл в 16-ном редакторе.

1. Плохой .EXE-файл содержит 1 сегмент, который начинается с адреса 300h. На нулевом адресе распологаются символы MS, показывающие, что это – EXE (см. рис. 2).



Рис. 2 – «Плохой» .EXE-файл в 16-ном редакторе.

1. «Хороший» EXE файл содержит 3 сегмента – стек, данные и код (см. рис. 3).



Рис. 3 – «Хороший» .EXE-файл в 16-ном редакторе.

Загрузка COM модуля в основную память

1. При загрузке .COM файла система помещает PSP перед кодом, а сам код – после, по смещению 100h (размер PSP).
2. При загрузке «плохого» .EXE файла PSP помещается перед сегментом кода. В сегменты кода первые 100h байт, зарезервированные директивой ORG 100h, заполнены 0 и не используются .
3. В .COM файле все сегментные регистры указывают на начало PSP. В «плохом» .EXE – DS и ES указывают на PSP, SS – на байт перед началом сегмента кода.
4. «Начало» стека (куда указывает SS) располагается на PSP, однако его вершина расположена на смещении FFFEh, т.е. он занимает весь сегмент. При загрузке «плохого» .COM-файла вершина стека находится там же, где и его начало.

Загрузка «хорошего» EXE модуля в основную память

1. Сначала выделяется память под PSP, на который указывают регистры DS и ES, потом выделяется память под стек (256 байт), на который указывает SS, после располагается сегмент данных и сегмент кода, на который указывает CS.
2. Сегментные регистры DS и ES после загрузки .EXE указывают на PSP.
3. Стек определяется регистрами SS, который указывает на его начало, и SP, который указывает на его вершину.
4. Точка входа определяется по имени процедуры, которое указывается после директивы END.

Состояние сегментных регистров .COM и .EXE файлов смотри на рис. 4.

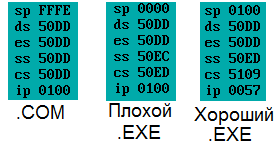


Рис. 4 – состояние сегментных регистров после запуска соответствующего файла.

# Вывод

В ходе работы была разработана программа для определения типа PC и версии OC. Были исследованы различия в структурах исходных текстов модулей типов .СОМ и .ЕХЕ, структур файлов загрузочных модулей и способов их загрузки в основную память.