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. 0381 _____ Шиняева А.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

Цель работы

Исследование различий в структурах исходных текстов модулей типов .COM и .EXE, 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 PC и выводить строку с названием модели. Если код не совпадает ни с одним значением, то двоичный код переводится 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.

Выполнение работы

Были реализованы следующие функции:

PC_TYPE - определяет тип ПК (по содержимому регистра AL) согласно таблице (Рис. 1).

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

Рис. 1 - Соответствие кода предпоследнего байта ROM BIOS и типа PC

Выполнение программы начинается с занесения в регистр AL данных предпоследнего байта ROM BIOS для определения типа PC. Происходит сравнение регистра AL с числами, соответствующих различным типам PC (Рис.1). В случае совпадения происходит вывод на экран соответствующей числу строки, а в противном случае – замена числа на код ASCII символа и вывод его на экран. Все выводы на экран в программе реализованы с помощью функции write, в которой использована функция 09h прерывание 21h.

PC_TYPE - функция OS, определяет номер основной версии MS-DOS, лежащий в регистре AL, номер её модификации в регистре AH, серийный номер OEM в регистре BH и 24-битный серийный номер пользователя в регистрах BL:CX. Программа последовательно переводит числа в соответствующие им строки и выводит на экран. Завершается программа вызовом функции 4Ch прерывания 21h с кодом 0. Были выделены строки для вывода информации:

- T_PC db 'Type: PC',0DH,0AH,'$' - для типа PC;
- T_PC_XT db 'Type: PC/XT',0DH,0AH,'$' - для типа PC/XT;
- T_AT db 'Type: AT',0DH,0AH,'$' - для типа AT;
- T_PS2_M30 db 'Type: PS2 модель 30',0DH,0AH,'$' - для типа PS2 модель 30;
- T_PS2_M50_60 db 'Type: PS2 модель 50 или 60',0DH,0AH,'$' - для типа PS2 модель 50 или 60;
- T_PS2_M80 db 'Type: PS2 модель 80',0DH,0AH,'$' - для типа PS2 модель 80;

- T_PC_JR db 'Type: PCjr',0DH,0AH,'$' - для типа PC jr;
- T_PC_C db 'Type: PC Convertible',0DH,0AH,'$' - для типа PC Convertible;
- VERSION db 'Version: . ',0DH,0AH,'$' - для версии и модификации;
- OEM db 'OEM: ',0DH,0AH,'$' - для серийного номера OEM;
- USER db 'User: $' - для серийного номера пользователя;

Результаты работы «хороших» .COM и .EXE модулей представлены на рисунках 2 и 3. Результат работы «плохого» .EXE представлен на рисунке 4.

Рис. 2 - Результат .COM модуля

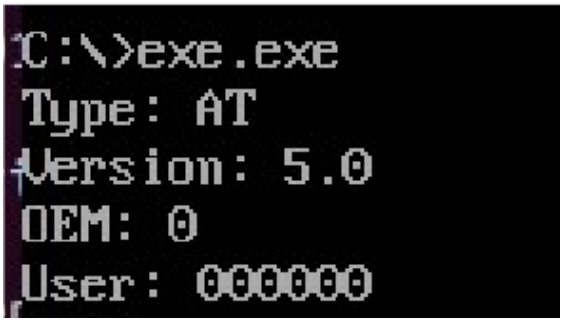

Рис. 3 - Результат .EXE модуля

Рис. 4 - Результат "плохого" .EXE модуля

Контрольные вопросы

Отличия исходных текстов COM и EXE программ

1. Сколько сегментов должна содержать COM-программа?

COM-программа должна содержать один сегмент, в котором располагаются команды и данные.

2. Сколько сегментов должна содержать EXE-программа?

EXE-программа может содержать от одного до трёх сегментов: сегмент команд, данных и стека (последние два могут отсутствовать).

3. Какие директивы должны обязательно быть в тексте COMпрограммы?

Должна быть обязательна директива ORG 100h, так как при загрузке модуля все сегментные регистры содержат адрес префикса программного сегмента (PSP), который является 256-байтовым(100H) блоком, поэтому адресация имеет смещение в 256 байт от нулевого адреса. Также необходима процедура ASSUME для того, чтобы сегмент данных и сегмент кода указывали на один общий сегмент. (ASSUME CS:TESTPC, DS:TESTPC, ES:NOTHING, SS:NOTHING)

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

Не все форматы команд можно использовать в COM-программе. Так, например, ввиду отсутствия таблицы настроек нельзя использовать команды вида mov <register>, seg . Таблица настроек состоит из элементов, число которых записано в байтах 06-07. Элемент таблицы настройки состоит из двух полей: 2-байтного смещения и 2-байтного сегмента, и указывает слова в загрузочном модуле, содержащее адрес, который должен быть настроен на место памяти, в которое загружается задача.

Отличия форматов файлов .COM и .EXE модулей

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

Файл .COM состоит из программного сегмента PSP размером 256 байт, одного сегмента, содержащего команды и данные, а также стека, генерируемого автоматически. Структура файла представлена на рисунке 5. Код располагается с адреса 100h (256).

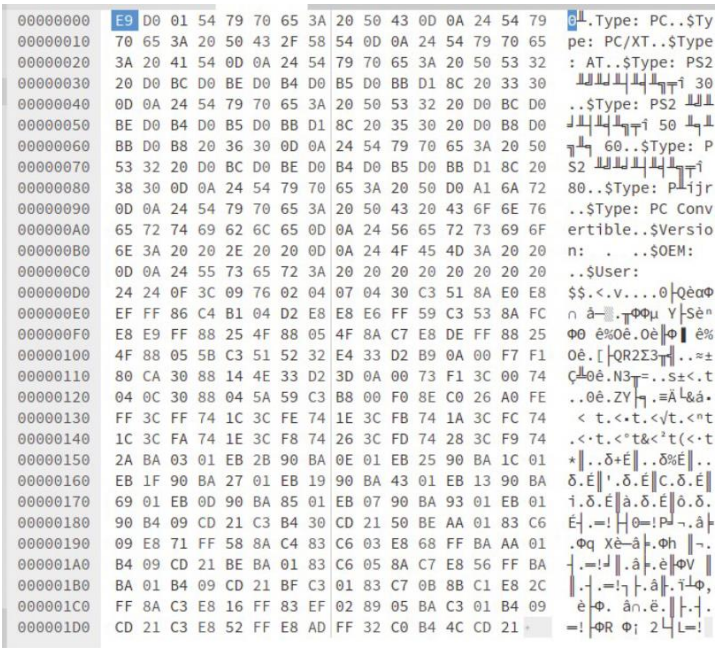

Рис. 5 - Структура файла .COM

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

«Плохой» файл .EXE состоит из единственного сегмента, что некорректно для файлов данного типа. Структура файла представлена на рисунке 6. Код и данные располагается с адреса 300h, а с адреса 0h лежат заголовок и таблица настроек.

Рис. 6 - Структура "плохого" файла .EXE

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

В «хорошем» EXE файле уже исправлено объединение всех сегментов, теперь они разнесены по разным: сегмент стека, сегмент данных и сегмент кода, в отличии от «плохого». Также отсутствие обязательной для COM файлов директивы ORG 100h приводит к тому, что память под управляющие структуры (PSP) не выделяется.

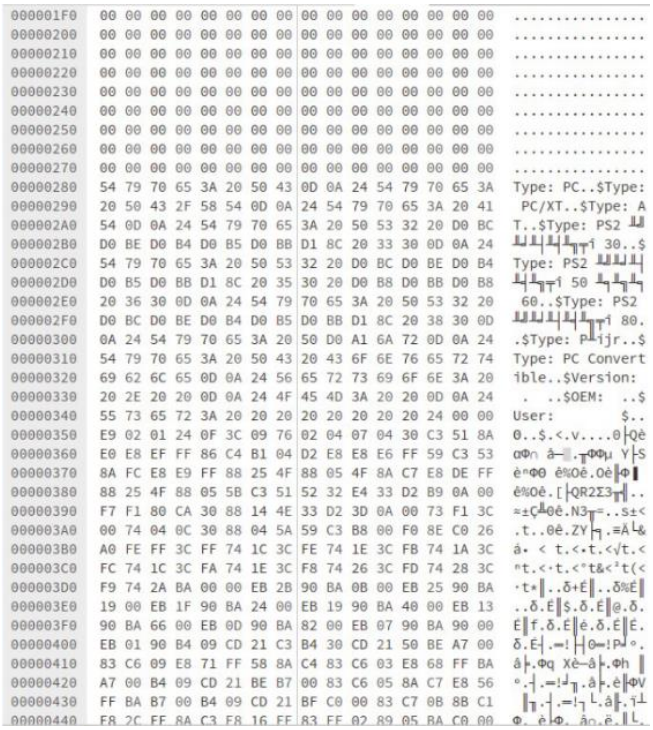

Рис. 7 - Структура «хорошего» файла .EXE

Загрузка .COM модуля в основную память

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

После загрузки программы все 4 сегментных регистра указывают на начало единственного сегмента, то есть фактически на начало PSP. Указатель стека SP автоматически инициализируется числом 0FFFEh. Таким образом, независимо от фактического размера программы, ей выделяется 64 Кбайт адресного пространства, всю нижнюю часть которого занимает стек (рисунок 8). Указатель команд на момент начала программы указывает на адрес 100h, с которого, соответственно, располагается код.

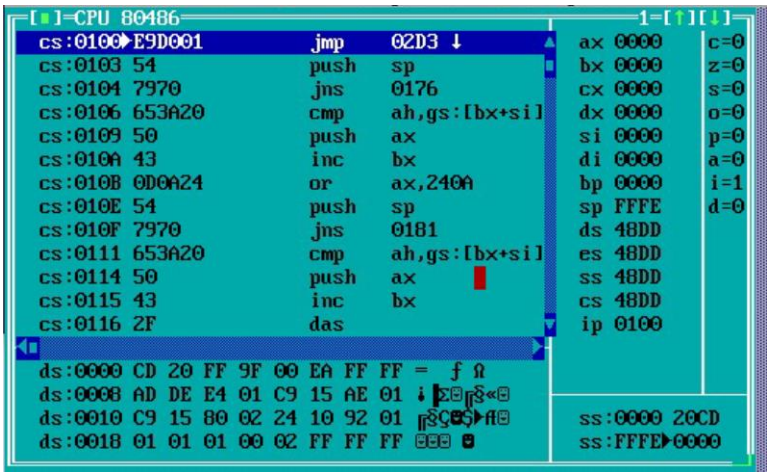

Рис. 8 - Отладка .COM модуля (с помощью TD.EXE)

2. Что располагается с адреса 0?

С адреса 0 располагается программный сегмент PSP, размером 256 байт, зарезервируемый операционной системой.

3. Какие значения имеют сегментные регистры? На какие области памяти они указывают? Сегментные регистры имеют значения 48DDh и указывают на программный сегмент PSP.

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

Стек занимает всю нижнюю часть адресного пространства. При этом сегментный регистр SS в начале программы, как и остальные регистры, указывает на PSP (48DDh), а регистр SP – на конец стека с адресом 0FFFEh.

Загрузка «хорошего» .EXE модуля в основную память

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

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. В указатель команд IP загружается смещение точки входа в программу, а в указатель стека SP – смещение конца сегмента стека. Таким образом, после загрузки программы в память адресуемыми оказываются все регистры, кроме сегментов данных.

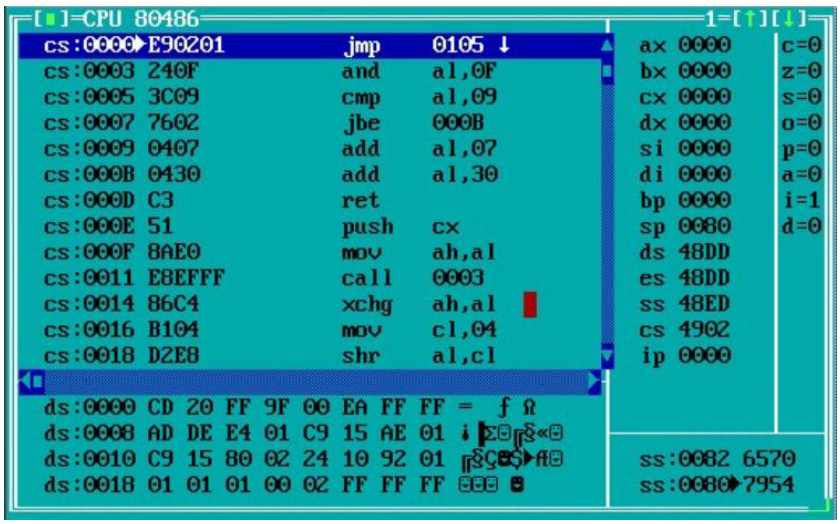

Рис. 9 - Отладка «хорошего» .EXE модуля (с помощью TD.EXE)

2. На что указывают регистры DS и ES?

При запуске программы регистры DS и ES указывают на начало PSP.

3. Как определяется стек?

Оператор segment, начинающий сегмент стека, имеет описатель stack, что приводит к тому, что при загрузке программы в память регистр SS будет настроен на начало сегмента стека, а указатель стека SP -–на его конец.

4. Как определяется точка входа?

Точка входа определяется с помощью директивы END.

Вывод

В ходе 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.