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. 0381 _____ Кирильцев Д.А.

Преподаватель _____ Ефремов М. А.

Санкт-Петербург

2022

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 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.

Таблица 1 — Процедуры в программе.

Процедура	Описание
TETR_TO_HEX	Перевод десятичной цифры в код символа
BYTE_TO_HEX	Перевод байта в 16-ной с/с в символьный код
WRD_TO_HEX	Перевод слова в 16-ной с/с в символьный код
BYTE_TO_DEC	Перевод байта в 16-ной с/с в символьный код в 10-ной с/с
model_print	Вывод строки на экран
PC_ver	Определение модели PC
OS_ver	Определение версии OS
OEM_num	Определение OEM
USER_num	Определение серийного номера пользователя

Выполнение работы.

Данные объявленные в программе:

```
PC_m db  'PC',0Dh,0Ah,'$'
XT_m db 'PC/XT',0Dh,0Ah,'$'
AT_m db  'AT',0Dh,0Ah,'$'
PS2_model30_m db 'PS2 model 30',0Dh,0Ah,'$'
PS2_model80_m db 'PS2 model 80',0Dh,0Ah,'$'
PCjr_m db 'PCjr',0Dh,0Ah,'$'
PC_convertible_m db 'PC Convertible',0Dh,0Ah,'$'
PC_custom_m db '  ', 0Dh, 0Ah, '$'
DOS_ver db '  .  ', 0Dh, 0Ah, '$'
OEM db '   ', 0Dh, 0Ah, '$'
USER db '                    ', '$'
```

Программа последовательно выводит тип пк, версию ОС, OEM и номер пользователя.

Далее предоставлены скриншоты полученных модулей.

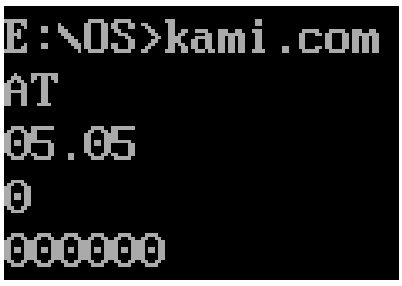

Рис.1- "хороший" COM модуль

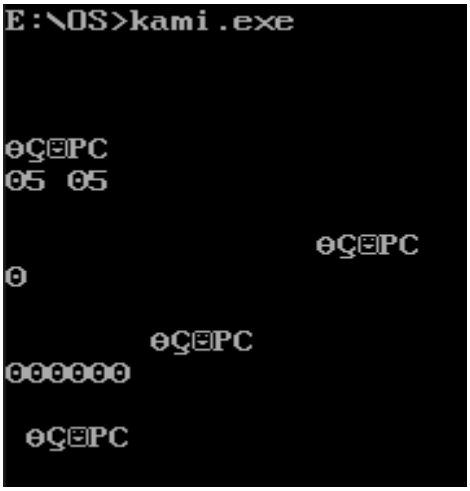

Рис. 2 - "плохой" EXE модуль

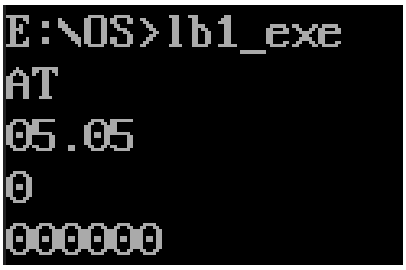

Рис.3-"хороший" EXE модуль

Отличия исходных текстов COM и EXE программ

1. Сколько сегментов должна сдержать COM-программа?

- Один, код и данные в COM-модуле располагаются в одном сегменте, а стек генерируется автоматически.

2. EXE-программа?

- EXE-модуль должен содержать сегмент кода и сегмент данных. Остальные сегменты являются опциональными. Если не объявить стек, то будет использоваться DOS-овский.

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

- 1. ORG 100h - Так как адресация начинается с шест. смещения 100 от начала PSP, то в программе после оператора SEGMENT кодируется директива ORG 100H.

- 2. ASSUME - для того, чтобы сегмент данных и сегмент кода указывали на один общий сегмент.

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

- Нет. Нельзя использовать команды вида mov register, segment , т. к. В момент ассемблирования и редактирования связей сегментное значение для сегмента неизвестно. Оно определяется только при загрузке программы. Поскольку файл типа .COM не может предоставить загрузчику перечня всех сегментных ссылок (информация для перемещения), то в данном случае программа будет выполняться неправильно.

Отличия форматов файлов .COM и .EXE программ

Рис. 4 - "Хороший" COM модуль

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

- COM-файл состоит из одного сегмента, сегмент стека генерируется автоматически при создании COM - модуля.
- COM-файл ограничен размером одного сегмента и не превышает 64 Кб.
- Программа, записанная в файле типа .COM может сразу выполняться (из-за постоянного смещения).
- Код начинается с адреса 0h, но при загрузке модуля устанавливается смещение в 100h. (в доказательство приведен рис. 4)

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

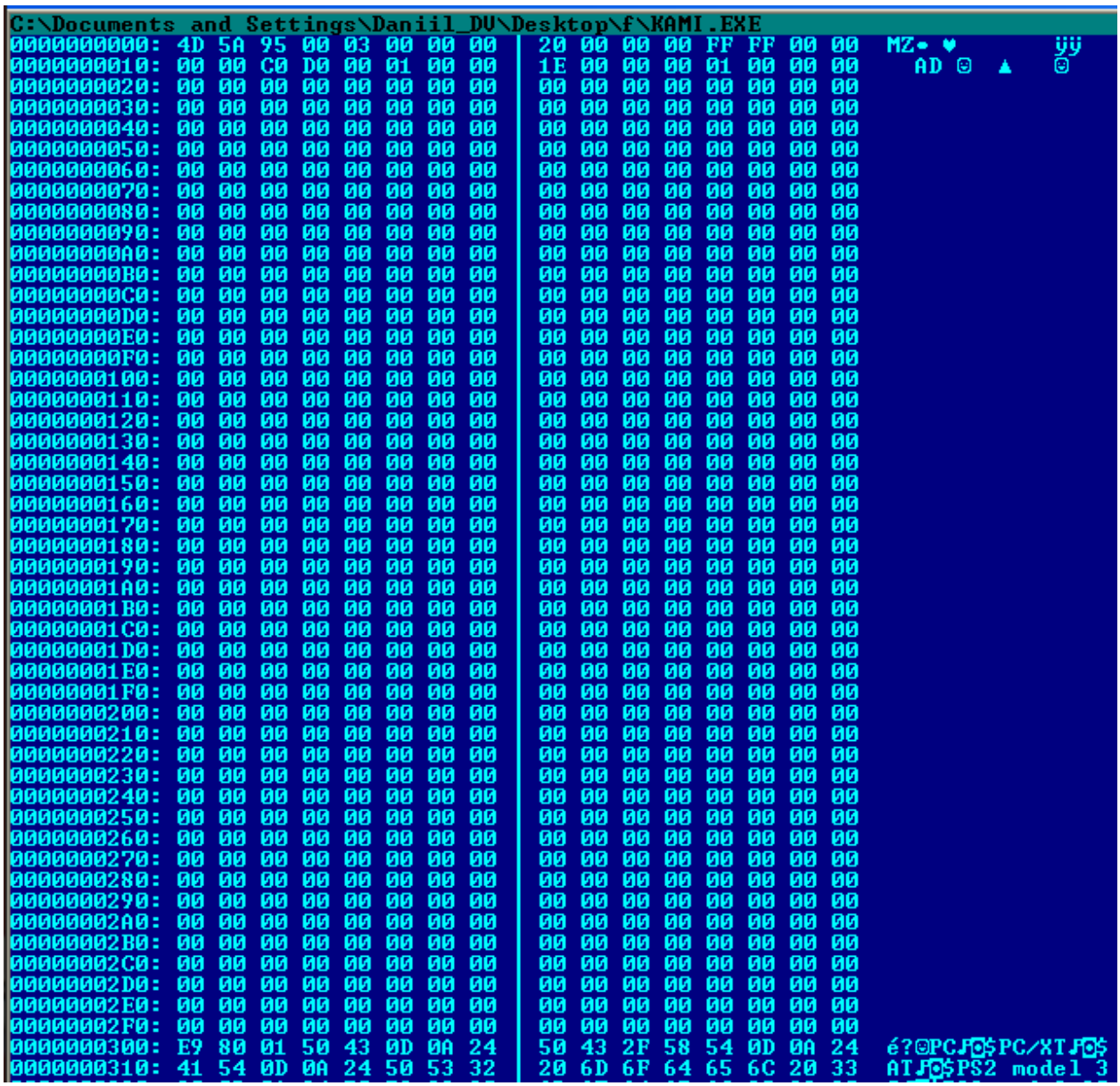

Рис. 5 - "плохой" EXE модуль

- У «плохого» EXE файла данные и код располагаются в одном сегменте, однако это не соответствует формату EXE.
- Код начинается с адреса 300h, а с адреса 0h идёт настраивающая таблица (заголовок EXE файла). (Это иллюстрирует Рис. 5)

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

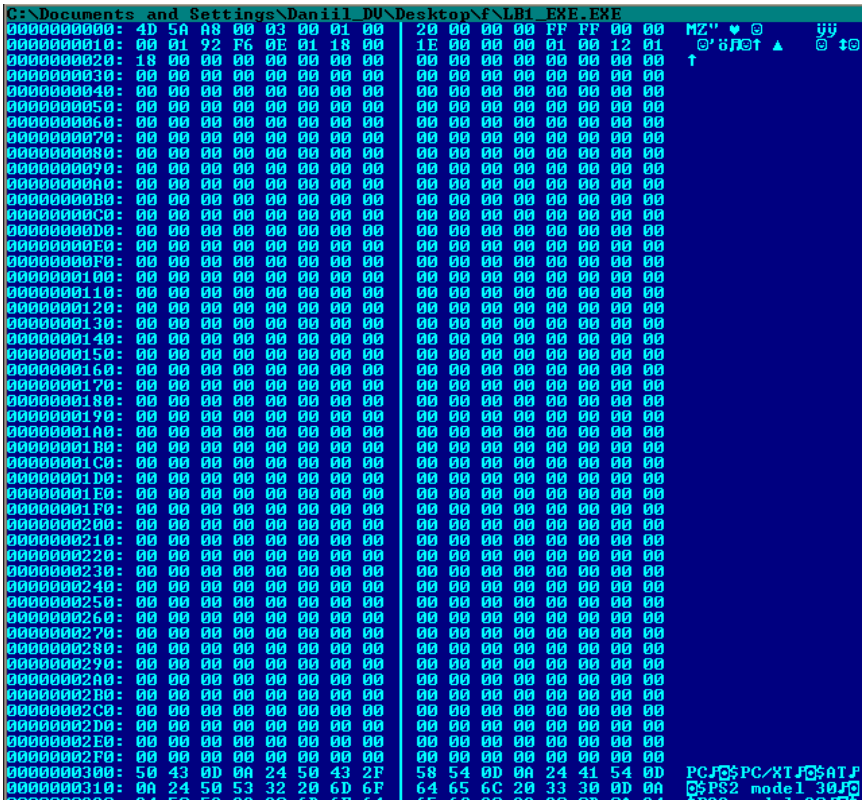

Рис. 6 - "хороший" EXE модуль

- В EXE-модуле код и данные являются отдельными сегментами, также присутствует таблица связей, заголовок, отвечающий за настройку адресов.

- В «хорошем» EXE-модуле происходит разделение сегментов (кода и данных), необходимое для правильного форматирования, а в «плохом» содержится лишь один сегмент, объединяющий код и данные. «Плохой» EXE начинает код с 300h, так как он получается из COM модуля, в котором изначально сегмент кода смещён на 100h, Но, так как, происходит создание EXE-модуля, добавляется еще и сдвиг PSP (200h). В «хорошем» EXE

присутствует только смещение для PSP модуля, поэтому код начинается с 200h.

- В данном случае смещение кода 300h так как выделяется память под стек (в размере 100h), память под стек находится между PSP и кодом. (Как показано на рис. 6)

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

- Стек находится между PSP и данными и занимает с 100h до 300h

Загрузка COM модуля в основную память

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

- Определяется сегментный адрес участка Основной Памяти, у которого достаточно места для загрузки программы, образ COM-файла считывается с диска и помещается в память, начиная с адреса PSP 100h. После загрузки двоичного образа COM-модуля сегментные регистры CS, DS, ES и SS указывают на PSP(в данном случае сегментные регистры указывают на 48DD), SP указывает на конец сегмента PSP (FFFE), слово 00H помещено в стек, IP содержит 100H. (Это можно увидеть на Рис. 7)

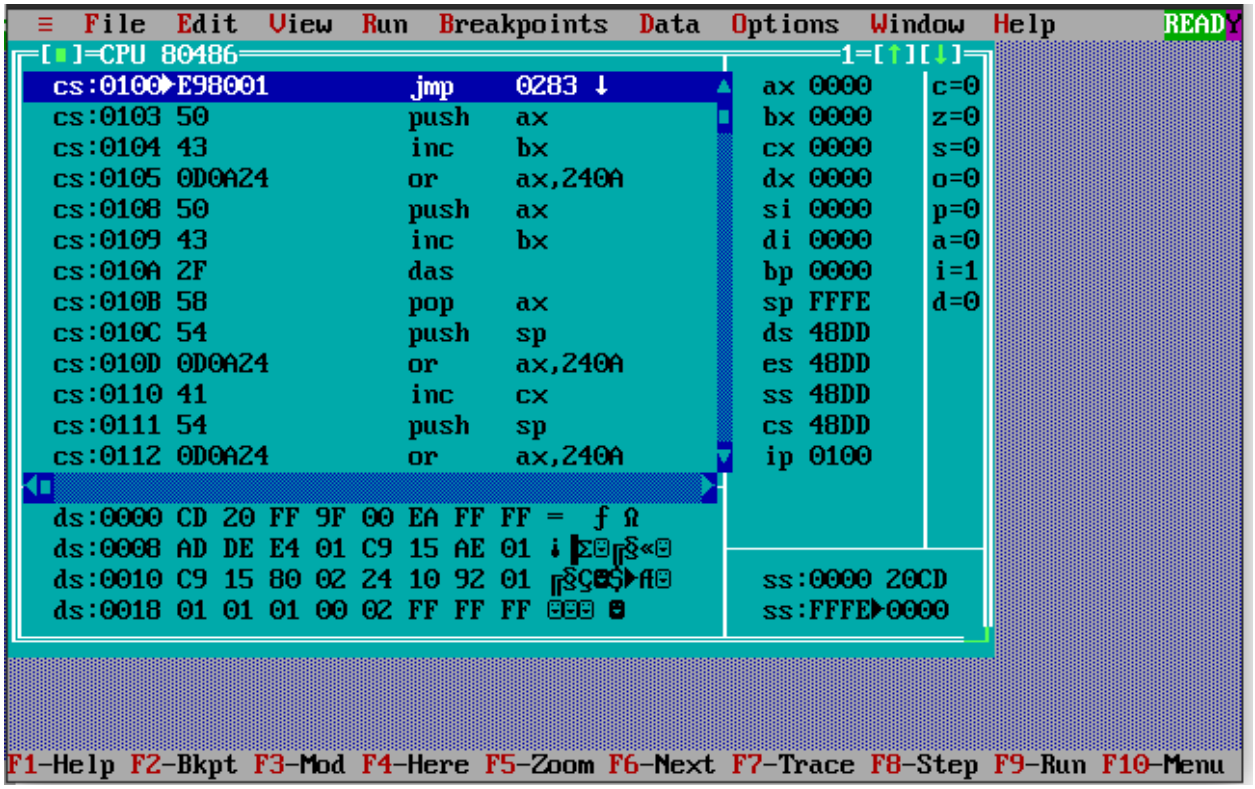

Рис. 7 - Отладчик для "Хорошего" COM-модуля

2. Что располагается с адреса 0?

- Программный сегмент PSP, размером 256 байт (100h), зарезервируемый операционной системой.

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

- Сегментные регистры CS, DS, ES и SS указывают на PSP и имеют значения 48DD. (Это можно увидеть на Рис. 7)

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

- Стек генерируется автоматически при создании COM-программы. SS – на начало (0h), регистр SP указывает на конец стека (FFFEh), Адреса стека расположены в диапазоне 0h – FFFEh (FFFEh, – последний адрес, кратный двум). (Это можно увидеть на Рис. 7)

Загрузка «хорошего» EXE модуля в основную память

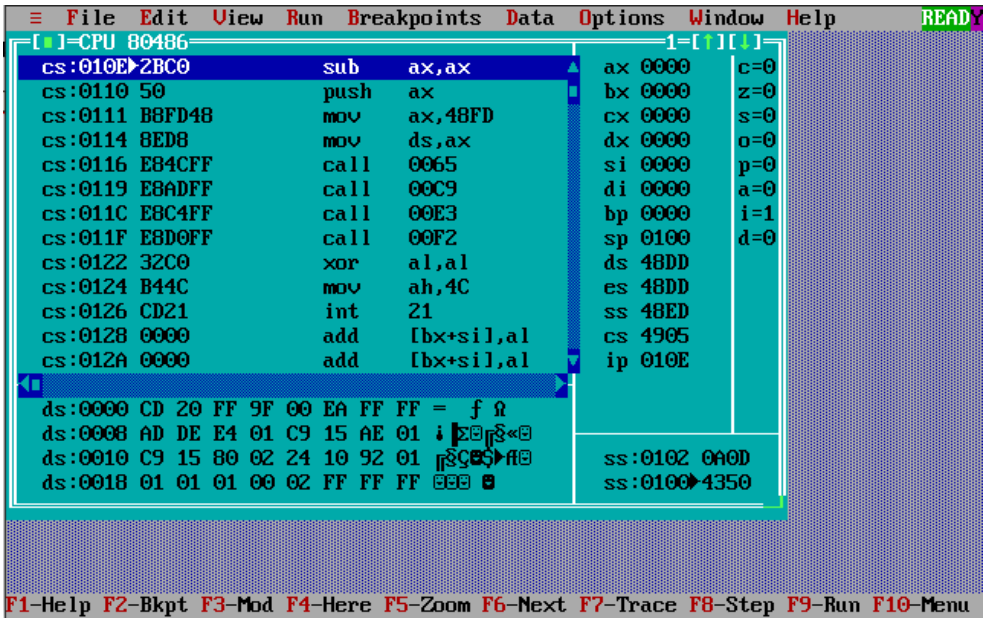

Рис. 8 - Отладчик "хорошего" EXE-модуля

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

- EXE-файл загружается, начиная с адреса PSP:0100h. В процессе загрузки считывается информация заголовка (PSP) EXE в начале файла и выполняется перемещение адресов сегментов, то есть DS и ES устанавливаются на начало сегмента PSP(DS=ES=48DD), SS(SS=48ED) – на начало сегмента стека, CS(CS=4905) – на начало сегмента команд. В IP загружается смещение точки входа в программу, которая берётся из метки после директивы END. (Это иллюстрирует рисунок 8)

2. На что указывают регистры DS и ES?

- Регистры DS и ES указывают на начало PSP.

3. Как определяется стек?

- Стек определяется с помощью Stack Segment, после которой задаётся размер стека. При исполнение регистр SS указывает на начало сегмента стека, а SP на конца стека(его смещение).

4. Как определяется точка входа?

- Точка входа определяется при помощи директивы END.

Выводы.

Были написаны COM и EXE модули, на основе которых производилось сравнение данных форматов. Также были выявлены недостатки и преимущества каждого из них.