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 Захаров Ф.С.

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

Написать тексты исходных .COM и .EXE модулей, которые определяют 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.

Выполнение работы

В данной программе используются следующие функции и структуры данных:

- TETR_TO_HEX - Перевод десятичной цифры в код символа, который записывается в AL
- BYTE_TO_HEX - Перевод значений байта в число 16- ой СС и его представление в виде двух символов
- WRD_TO_HEX - Перевод слова в число 16-ой СС и представление его в виде четырех символов

- BYTE_TO_DEC - Перевод значения байта в число 10- ой СС и представляет его в виду символов
- PRINT_STRING - Вывод строки на экран
- PRINT_PC_TYPE - Печать на экран тип ПК
- PRINT_OS_VERSION - Печать на экран версии ОС, серийного номера OEM и серийного номера пользователя

В ходе работы программа выполняет следующие действия:

1. Процедура PRINT_PC_TYPE, которая выводит на экран тип ПК пользователя. Информация о типе ПК находится в предпоследнем байте ROM BIOS по адресу 0F000:0FFFEh. Значение этого байта определяет тип:

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

Рисунок 1 - Соответствие кода предпоследнего байта ROM BIOS и типа PC

Если значение байта не сходится со значениями типов ПК, то выводится сообщение об ошибке.

2. Процедура PRINT_OS_VERSION, которая выводит на экран версию ОС, серийный номер OEM и серийный номер пользователя. В данной процедуре используется функция 30h прерывания 21h.

3. Завершение работы программы.

Был написан текст исходного .COM модуля lb1com.asm, который определяет тип ПК и версию его системы. С помощью команды masm lb1com.asm был получен объектный файл lb1com.obj. Командой link lb1com.obj был собран «плохой» .EXE модуль.

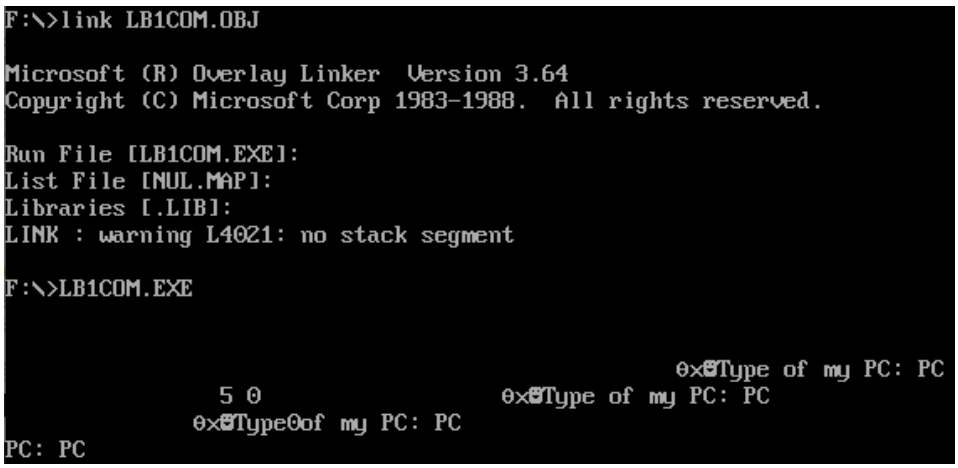

Рисунок 2 - Вывод lb1com.exe

Далее при помощи EXE2BIN был получен «Хороший» com файл.

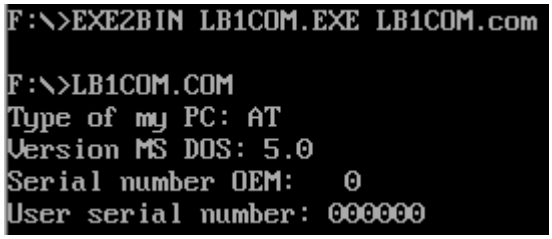

Рисунок 3 - вывод lb1com.com

Далее был написан .exe модуль lb1exe.asm, который выполняет те же функции, что и lb1com.asm, так же скомпилирован и получен «хороший» .exe модуль.

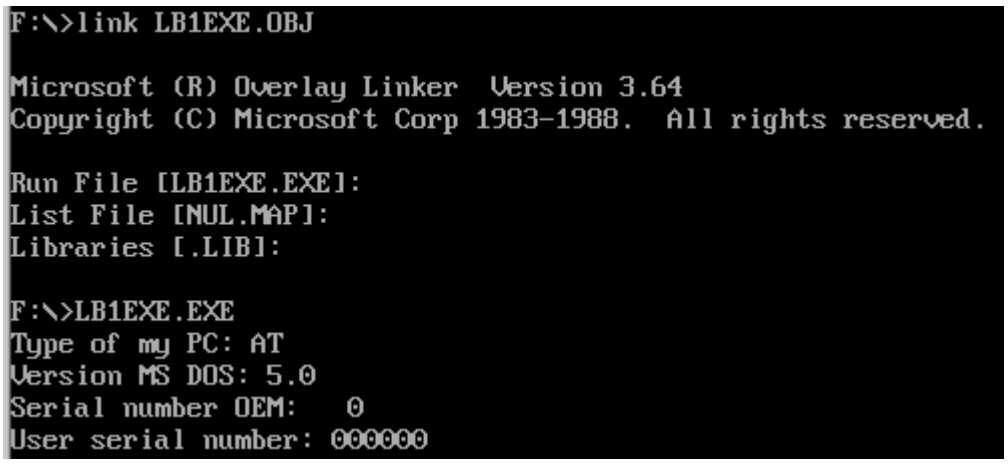

Рисунок 4 - вывод lb1exe.exe

Контрольные вопросы

Отличия исходных текстов COM и EXE программ

1. Сколько сегментов должна содержать COM-программа?

Ровно один сегмент, содержащий данные и код. Стек генерируется автоматически.

2. Сколько сегментов должна содержать EXE-программа?

EXE-программа должна содержать один или более сегментов. Количество сегментов зависит от выбранной модели памяти.

3. Какие директивы должны обязательно быть в тексте COMпрограммы?

ORG 100h – смещение кода на 256 байт от нулевого адреса (пропуск области PSP) ASSUME нужно использовать, чтобы сегментные регистры указывали на один сегмент.

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

Нельзя использовать команды вида: seg NAME, где NAME – название сегмента, так как в COM-программе отсутствует таблица настройки.

Отличия форматов файлов .COM и .EXE модулей

1. Какова структура файла .COM? С какого адреса располагается код? У COM файла есть только один сегмент, в котором располагаются код и данные. Они начинаются с нулевого адреса.

```
Address  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f Dump
00000000 e9 78 02 54 79 70 65 20 6f 66 20 6d 79 20 50 43 йx.Type of my PC
00000010 3a 20 50 43 0d 0a 24 54 79 70 65 20 6f 66 20 6d : PC..$Type of m
00000020 79 20 50 43 3a 20 50 43 2f 58 54 0d 0a 24 54 79 y PC: PC/XT..$Ty
00000030 70 65 20 6f 66 20 6d 79 20 50 43 3a 20 41 54 0d pe of my PC: AT.
00000040 0a 24 54 79 70 65 20 6f 66 20 6d 79 20 50 43 3a .$Type of my PC:
00000050 20 50 53 32 20 6d 6f 64 65 6c 20 33 30 0d 0a 24  PS2 model 30..$
00000060 54 79 70 65 20 6f 66 20 6d 79 20 50 43 3a 20 50 Type of my PC: P
00000070 53 32 20 6d 6f 64 65 6c 20 35 30 20 6f 72 20 36 S2 model 50 or 6
00000080 30 0d 0a 24 54 79 70 65 20 6f 66 20 6d 79 20 50 0..$Type of my P
00000090 43 3a 20 50 53 32 20 6d 6f 64 65 6c 20 38 30 3a C: PS2 model 80:
000000a0 20 0d 0a 24 54 79 70 65 20 6f 66 20 6d 79 20 50  ..$Type of my P
000000b0 43 3a 20 50 d0 a1 6a 72 0d 0a 24 54 79 70 65 20 C: PPЎjr..$Type
000000c0 6f 66 20 6d 79 20 50 43 3a 20 50 43 20 43 6f 6e of my PC: PC Con
000000d0 76 65 72 74 69 62 6c 65 0d 0a 24 56 65 72 73 69 vertible..$Versi
000000e0 6f 6e 20 4d 53 20 44 4f 53 3a 20 20 2e 20 20 0d on MS DOS:  .  .
000000f0 0a 24 53 65 72 69 61 6c 20 6e 75 6d 62 65 72 20 .$Serial number
00000100 4f 45 4d 3a 20 20 20 20 20 20 20 0d 0a 24 55 73 OEM:       ..$Us
00000110 65 72 20 73 65 72 69 61 6c 20 6e 75 6d 62 65 72 er serial number
00000120 3a 20 20 20 20 20 20 20 0d 0a 24 45 72 72 6f 72 :       ..$Error
00000130 21 20 54 68 65 20 62 79 74 65 20 76 61 6c 75 65 ! The byte value
00000140 20 64 6f 65 73 20 6e 6f 74 20 6d 61 74 63 68 20  does not match
00000150 74 68 65 20 50 43 20 74 79 70 65 20 76 61 6c 75 the PC type valu
00000160 65 73 24 0f 3c 09 76 02 04 07 04 30 c3 51 8a e0 es$.<.v....0ГQЉа
00000170 e8 ef ff 86 c4 b1 04 d2 e8 e8 e6 ff 59 c3 53 8a ипя†Д±.ТиижяYГSЉ
00000180 fc e8 e9 ff 88 25 4f 88 05 4f 8a c7 e8 de ff 88 ьийя€%О€.ОЉЗиЮя€
00000190 25 4f 88 05 5b c3 51 52 32 e4 33 d2 b9 0a 00 f7 %О€.[ГQR2д3Т№..ч
000001a0 f1 80 ca 30 88 14 4e 33 d2 3d 0a 00 73 f1 3c 00 сЋК0€.N3Т=..sc<.
000001b0 74 04 0c 30 88 04 5a 59 c3 50 b4 09 cd 21 58 c3 t..0€.ZYГPґ.H!XГ
000001c0 b8 00 f0 8e c0 26 a0 fe ff 3c ff 74 26 3c fe 74 ё.рЋА&.юя<яt&<юt
000001d0 28 3c fb 74 24 3c fc 74 26 3c fa 74 28 3c fc 74 (<ыt$<ьt&<ъt(<ьt
000001e0 2a 3c f8 74 2c 3c fd 74 2e 3c f9 74 30 ba 2b 02 *<шt,<эt.<щt0є+.
000001f0 eb 31 90 ba 03 01 eb 2b 90 ba 17 01 eb 25 90 ba л1.є..л+.є..л%.є
00000200 2e 01 eb 1f 90 ba 42 01 eb 19 90 ba 60 01 eb 13 ..л..єB.л..є`.л.
```

Рисунок 5 - Содержимое lb1_com.com

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

«Плохой» EXE файл содержит заголовок с технической информацией, необходимой для загрузки, таблицу настроек адресов и сегмент, в

котором находятся данные и код. Код и данные располагаются с адреса 300h, а с адреса 0h лежат заголовок и таблица настроек.

```
Address  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f Dump
00000000 4d 5a 87 01 03 00 00 00 20 00 00 00 ff ff 00 00 MZ‡..... ...яя..
00000010 00 00 fc 24 00 01 00 00 1e 00 00 00 01 00 00 00 ..ь$............
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

Рисунок 6 - Начало "плохого" EXE файла

```
000002e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000002f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000300 e9 78 02 54 79 70 65 20 6f 66 20 6d 79 20 50 43 йx.Type of my PC
00000310 3a 20 50 43 0d 0a 24 54 79 70 65 20 6f 66 20 6d : PC..$Type of m
00000320 79 20 50 43 3a 20 50 43 2f 58 54 0d 0a 24 54 79 y PC: PC/XT..$Ty
00000330 70 65 20 6f 66 20 6d 79 20 50 43 3a 20 41 54 0d pe of my PC: AT.
00000340 0a 24 54 79 70 65 20 6f 66 20 6d 79 20 50 43 3a .$Type of my PC:
00000350 20 50 53 32 20 6d 6f 64 65 6c 20 33 30 0d 0a 24  PS2 model 30..$
00000360 54 79 70 65 20 6f 66 20 6d 79 20 50 43 3a 20 50 Type of my PC: P
00000370 53 32 20 6d 6f 64 65 6c 20 35 30 20 6f 72 20 36 S2 model 50 or 6
00000380 30 0d 0a 24 54 79 70 65 20 6f 66 20 6d 79 20 50 0..$Type of my P
00000390 43 3a 20 50 53 32 20 6d 6f 64 65 6c 20 38 30 3a C: PS2 model 80:
000003a0 20 0d 0a 24 54 79 70 65 20 6f 66 20 6d 79 20 50  ..$Type of my P
000003b0 43 3a 20 50 d0 a1 6a 72 0d 0a 24 54 79 70 65 20 C: PPЎjr..$Type
000003c0 6f 66 20 6d 79 20 50 43 3a 20 50 43 20 43 6f 6e of my PC: PC Con
000003d0 76 65 72 74 69 62 6c 65 0d 0a 24 56 65 72 73 69 vertible..$Versi
000003e0 6f 6e 20 4d 53 20 44 4f 53 3a 20 20 2e 20 20 0d on MS DOS:  .  .
000003f0 0a 24 53 65 72 69 61 6c 20 6e 75 6d 62 65 72 20 .$Serial number
00000400 4f 45 4d 3a 20 20 20 20 20 20 20 0d 0a 24 55 73 OEM:       ..$Us
00000410 65 72 20 73 65 72 69 61 6c 20 6e 75 6d 62 65 72 er serial number
00000420 3a 20 20 20 20 20 20 20 0d 0a 24 45 72 72 6f 72 :       ..$Error
00000430 21 20 54 68 65 20 62 79 74 65 20 76 61 6c 75 65 ! The byte value
00000440 20 64 6f 65 73 20 6e 6f 74 20 6d 61 74 63 68 20  does not match
00000450 74 68 65 20 50 43 20 74 79 70 65 20 76 61 6c 75 the PC type valu
00000460 65 73 24 0f 3c 09 76 02 04 07 04 30 c3 51 8a e0 es$.<.v....0ГQЉа
00000470 e8 ef ff 86 c4 b1 04 d2 e8 e8 e6 ff 59 c3 53 8a ипя†Д±.ТиижяYГSЉ
```

Рисунок 7 - "Плохой" exe файл с адреса 300h

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

   У «хорошего» EXE код, данные и стек находятся в разных сегментах, а в

«плохом» - в одном сегменте. С адреса 0 в «хорошем» EXE располагается валидная таблица настроек, в отличие от «плохого» EXE. У «хорошего» EXE выделяется память под стек между PSP и кодом.

Загрузка .com модуля в основную память:

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

В начале определяется сегментный адрес участка ОП, способного вместить загрузку программы, затем создается блок памяти для PSP и программы. После считывания COM-файл помещается в память с 100h. После сегментные регистры устанавливаются в начало PSP. SP устанавливается в конец PSP, 0000h помещается в стек, а в IP записывается 100h.

Код располагается с адреса 100h.

2. Что располагается с адреса 0?

PSP.

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

Сегментные регистры имеют значение 119C. Они указывают на начало PSP.

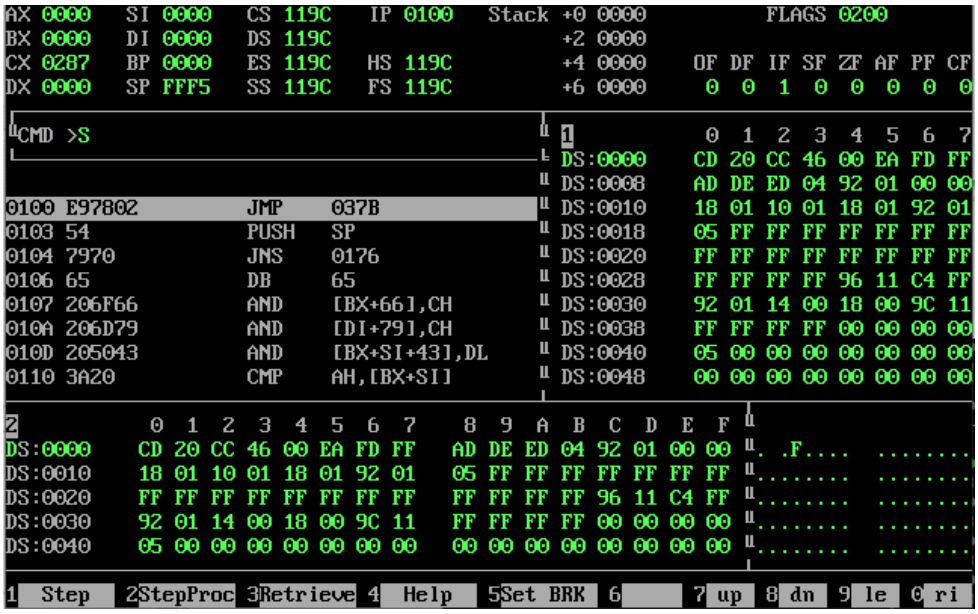

Рисунок 8 - lb1com.com в отладчике

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

Стек генерируется автоматически. Регистр SS указывает на начало блока PSP, а SP на конец стека. Стек расположен между адресами SS:0000h –

SS:FFFFh и заполняется с конца модуля в сторону уменьшения адресов.

Загрузка «хорошего» EXE модуля в основную память.

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

   Данный EXE загружается со считыванием информации заголовка EXE, выполняется перемещение адресов сегментов, ES и DS устанавливаются в начало PSP, SS – на начало сегмента стека, а CS – на начало сегмента команд. В IP загружается смещение точки входа в программу.

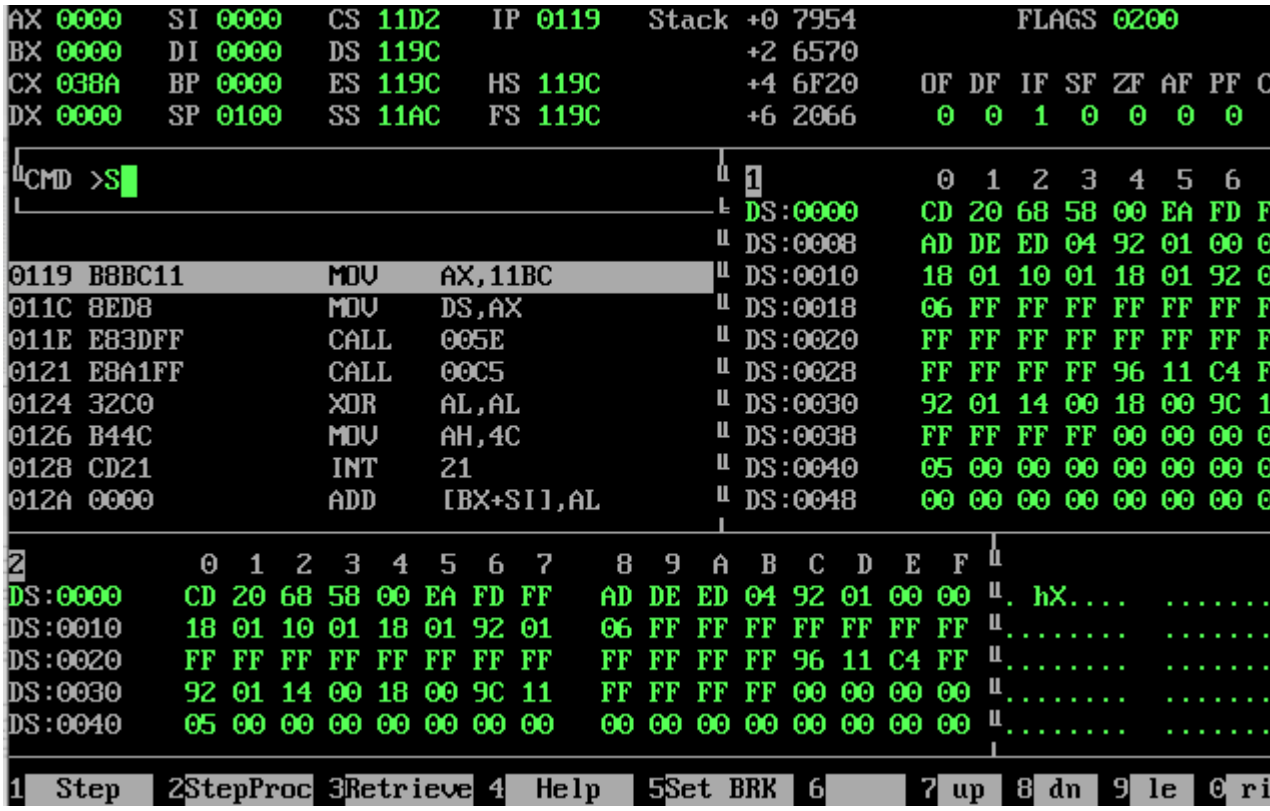

Рисунок 9 - lb1exe.exe в отладчике

2. На что указывают регистры DS и ES?

   На начало сегмента PSP.

3. Как определяется стек?

   Стек определяется на основе директивы .stack с указанием размера стека. SS указывает на начало сегмента стека, а SP указывает на конец.

4. Как определяется точка входа?

Точка входа определяется параметром после 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.