

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. 7383

Лосев М.Л.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2018

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.

Сведения об используемых функциях и структурах данных.

BYTE_TO_HEX – переводит значение регистра AL в его запись в шестнадцатеричной с/с, помещает ее в AX

WRD_TO_HEX – переводит значение регистра AX в его запись в шестнадцатеричной с/с, помещает ее в память так, что DI указывает на младшую цифру.

BYTE_TO_DEC – переводит значение регистра AL в его запись в десятичной с/с, помещает результат в память так, что SI указывает на младшую цифру.

OUTPUT_PROC – вызывает прерывание DOS вывода строки.

Последовательность действий, выполняемых утилитой.

Утилита получает байт кода типа компьютера и сопоставляет его с некоторыми кодами. В случае совпадения с одним из них она выводит соответствующее сообщение, если же совпадения нет, то код типа компьютера переводится в символьную запись в 16 с/с и выводится на экран. С помощью функции 30h прерывания 21h определяется версия DOS, OEM, номер пользователя. Версия DOS переводится в символьное представление в 10 с/с и выводится. OEM и номер пользователя переводятся в символьное представление в 16 с/с и выводятся на экран. Происходит вызов в DOS.

Результаты исследования.

Отличия исходных текстов COM и EXE программ:

1) *Сколько сегментов должна содержать COM-программа?*

COM-программа должна содержать один сегмент.

2) *EXE-программа?*

EXE-программа может содержать один или несколько сегментов. Обычно она содержит сегменты данных, стека и кода. Может содержать дополнительный сегмент.

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

ORG 100h - директива ORG служит для резервирования 256 байт от начального адреса под PSP, устанавливает счетчик адреса в нужное абсолютное значение.

SEGMENT — определяет начало и конец сегмента памяти.

ASSUME — задает значения сегментных регистров.

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

Нет. Нельзя использовать команды, содержащие адреса сегментов, например, CODE и DATA.

Отличия форматов файлов COM и EXE модулей:

1) *Какова структура файла COM? С какого адреса располагается код?*

COM-файл содержит данные программы и машинный код. Код располагается с самого начала файла (см. рис. 1).

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

Первые два байта заняты сигнатурой EXE-файла, далее до 1Bh включительно остальная часть стандартного заголовка (заголовок у «плохого» и «хорошего» EXE-файлов одинаковый, см. рис. 3), далее следует таблица настройки адресов (в случае «плохого» EXE-файла нули). Код начинается с 300h по счету байта (см. рис. 2).

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

Он похож на «плохой», но в нем данные начинаются с 220h байта (см. рис. 4).

```
view FIRST.COM - Far 3.0.5354 x86
C:\Users\mikka\Documents\OPERATING SYST_2019\DOC_DIR\TASM\BIN\FIRST.COM
00000000: E9 FA 00 20 0D 0A 24 54 59 50 45 20 49 42 4D 5B 7E TYPE IBM
00000010: 20 50 43 3A 20 24 50 43 2E 0D 0A 24 50 43 2F 58 PC: $PC.$PC/X
00000020: 54 2E 0D 0A 24 41 54 2E 0D 0A 24 50 53 32 20 6D T.$AT.$PS2 m
00000030: 2E 33 30 2E 0D 0A 24 50 53 32 20 6D 2E 35 30 2F .30.$PS2 m.50/
00000040: 36 30 2E 0D 0A 24 50 53 32 20 6D 2E 38 30 2E 0D 60.$PS2 m.80.$
00000050: 0A 24 50 43 20 43 4F 4E 56 45 52 54 49 42 4C 45 $PC CONVERTIBLE
00000060: 2E 0D 0A 24 50 43 20 4A 52 2E 0D 0A 24 44 4F 53 .J$PC JR.J$DOS
00000070: 20 56 45 52 53 49 4F 4E 3A 20 00 2E 00 00 0A 0D VERSION: . $J
00000080: 24 4F 45 4D 3A 20 00 00 0A 0D 24 55 53 45 52 20 $OEM: $J$USER
00000090: 4E 55 4D 42 45 52 3A 20 00 00 00 00 00 24 24 NUMBER: $$
000000A0: 0F 3C 09 76 02 04 07 04 30 C3 51 8A E0 E8 EF FF оков0000Qbаипя
000000B0: 86 C4 B1 04 D2 E8 E8 E6 FF 59 C3 53 8A FC E8 E9 ТД±ТимияYTSльий
000000C0: FF 88 25 4F 88 05 4F 8A C7 E8 DE FF 88 25 4F 88 я%0000b3и0я%00
000000D0: 05 58 C3 51 52 32 E4 33 D2 B9 0A 00 F7 F1 80 CA *TQR2д3Tн чсК
000000E0: 30 88 14 4E 33 D2 3D 0A 00 73 F1 3C 00 74 04 0C 00JN3T= s<< t+q
000000F0: 30 88 04 5A 59 C3 50 B4 09 CD 21 58 C3 BB 00 F0 0002YTProHIXГ» р
00000100: 8A 47 FE BA 08 01 E8 ED FF BA 16 01 3C FF 74 40 лью000иняе=0<ят@
00000110: BA 1C 01 3C FE 74 39 3C FB 74 35 BA 25 01 3C FC е.0<ют9<ыт5е%0<ь
00000120: 74 2E BA 2B 01 3C FA 74 27 BA 37 01 3C FC 74 20 т.е+0<ьт'е70<ьт
00000130: BA 46 01 3C F8 74 19 BA 64 01 3C FD 74 12 BA 52 еF0<wtJed0<этtеR
00000140: 01 3C F9 74 0B E8 62 FF BF 03 01 89 05 BA 03 01 0<цтdнбяIY0%+eY0
00000150: E8 A3 FF BA 30 CD 21 BE 7A 01 E8 76 FF 8A C4 BE иJар0H!sz0иивяJds
00000160: 7C 01 E8 6E FF BA 6D 01 E8 8B FF 8A C7 E8 3A FF |0иияем0и<ял3и:я
00000170: BF 01 89 45 FF BA 81 01 E8 7A FF 8B C1 BF 9D I10Eеяеf0изя<B1K
00000180: 01 E8 37 FF 8A C3 E8 21 FF 89 45 FE BA 8B 01 E8 0и7ялб!я%Еюе<0и
00000190: 64 FF 32 0C B4 4C CD 21 дя2ArLH!
```

Рисунок 1 - COM-файл: данные и команды

```
view FIRST.EXE - Far 3.0.5354 x86
C:\Users\mikka\Documents\OPERATING SYST_2019\DOC_DIR\TASM\BIN\FIRST.EXE
000000170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000001A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000001B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000001C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000001D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000001E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000001F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000220: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000230: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000240: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000250: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000260: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000270: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000280: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000290: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000002F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000300: E9 FA 00 20 0D 0A 24 54 59 50 45 20 49 42 4D 5B 7E TYPE IBM
000000310: 20 50 43 3A 20 24 50 43 2E 0D 0A 24 50 43 2F 58 PC: $PC.$PC/X
000000320: 54 2E 0D 0A 24 41 54 2E 0D 0A 24 50 53 32 20 6D T.$AT.$PS2 m
000000330: 2E 33 30 2E 0D 0A 24 50 53 32 20 6D 2E 35 30 2F .30.$PS2 m.50/
000000340: 36 30 2E 0D 0A 24 50 53 32 20 6D 2E 38 30 2E 0D 60.$PS2 m.80.$
000000350: 0A 24 50 43 20 43 4F 4E 56 45 52 54 49 42 4C 45 $PC CONVERTIBLE
000000360: 2E 0D 0A 24 50 43 20 4A 52 2E 0D 0A 24 44 4F 53 .J$PC JR.J$DOS
000000370: 20 56 45 52 53 49 4F 4E 3A 20 00 2E 00 00 0A 0D VERSION: . $J
1Help 2Dump 3Quit 4Text 5 6Edit 7Search
```

Рисунок 2 – «плохой» EXE-файл

```
view FIRTEXE.EXE - Far 3.0.5354 x86
C:\Users\mikka\Documents\OPERATING SYST_2019\DOC_DIR\TASM\BIN\FIRTEXE.EXE
0000000000: 4D 5A BD 01 02 00 01 00 20 00 00 00 FF FF 00 00 MZ%00 0 yÿ
0000000010: 18 00 00 00 5E 00 0C 00 3E 00 00 00 01 00 FB 71 ↑ ^ q > 0 ûq
0000000020: 6A 72 00 00 00 00 00 00 00 00 00 00 00 00 00 00 jr
0000000030: 00 00 00 00 00 00 00 00 00 00 00 00 63 00 c
0000000040: 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 q
```

Рисунок 3 – заголовок «хорошего» EXE-файла

Рисунок 4 – данные и команды «хорошего» EXE-файла

Загрузка COM-модуля в основную память:

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

Система выделяет свободный сегмент памяти и заносит его адрес во все сегментные регистры (CS, DS, ES и SS). В первые 256 байт этого сегмента записывается PSP. Непосредственно за ним загружается содержимое COM-файла без изменений. Указатель стека (регистр SP) устанавливается на конец сегмента. В стек записывается 0000h (адрес возврата для команды ret). Управление передаётся по адресу CS:0100h, где находится первый байт исполняемого файла.

- 2) Что располагается с адреса 0?

Префикс программного сегмента.

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

Все они имеют одно и то же значение 869h (см. рис. 5). Они указывают на начало сегмента программы, то есть на начало PSP. IP имеет значение 0100h (см.

рис. 5), то есть указывает на начало программы (потому что она начинается после PSP, который занимает 256 байт).

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

Стек располагается в конце сегмента программы. Его вершина имеет адрес 0FFFFh (в начале работы программы такое значение имеет SP(см. рис. 5)). Стек растёт в сторону уменьшения адресов. Адреса теоретически могут быть от 00h до 0FFFFh, но стек не должен занимать память, в которой находятся данные и код, чтобы не переписать их.

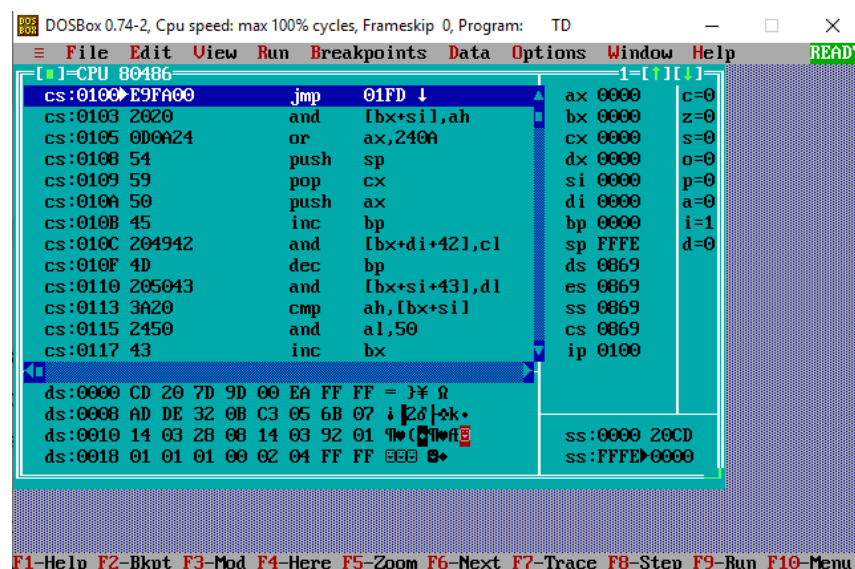


Рисунок 5 – COM-программа в отладчике

Загрузка «хорошего» EXE модуля в основную память:

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

DS и ES устанавливаются на начало PSP, SS – на начало сегмента стека, CS – на начало сегмента команд. В IP загружается смещение точки входа в программу.

2) *На что указывают регистры DS и ES?*

На начало PSP.

3) *Как определяется стек?*

В исходном коде модуля стек определяется при помощи директивы STACK. При запуске программы в SS заносится адрес сегмента стека, а в SP – адрес его вершины (см. рис. 6).

4) Как определяется точка входа?

Точка входа определяется по метке в ассемблерном коде, для которой указана директива END. 14-ый и 15-ый байты стандартного заголовка EXE-файла указывают на эту метку.

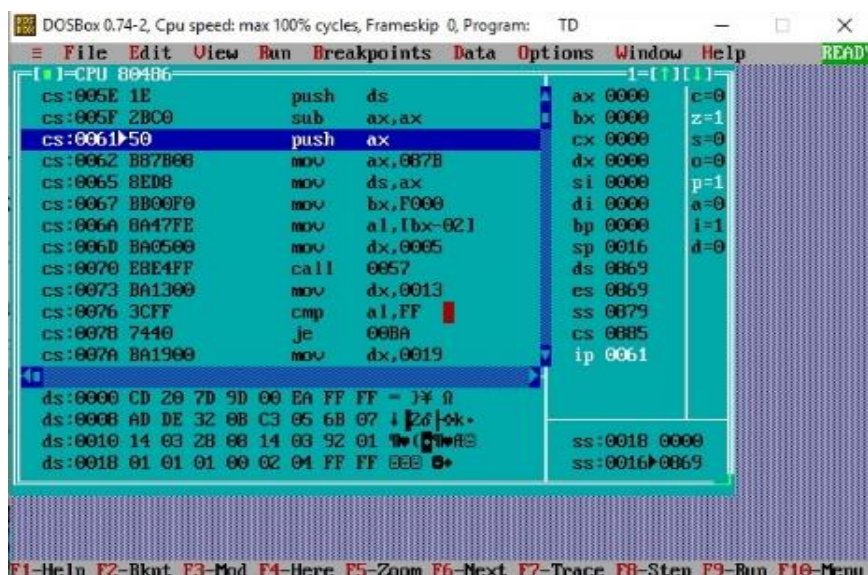


Рисунок 6 – EXE-программа в отладчике

Заключение.

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