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 загрузочного модулей

Студент гр. 9382 _____ Рыжих Р.В.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

Постановка задачи

Цель работы.

Изучить основные принципы трансляции, отладки и выполнения программ на языке Ассемблера. Исследование различий в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

Сведения о функциях и структурах данных.

В данной программе используются следующие функции и структуры данных:

Процедура	Описание
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

Последовательность действий

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: Ffh – PC, Feh/Fbh – PC/XT, FCh – AT, FAh – PS2 model 30, FCh –

PS2 model 50 or 60, F8h – PS2 model 80, FDh – Pcjr, F9h – PC Convertible. 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.

Выполнение шагов лабораторной работы:

1 шаг:

Был написан текст исходного .COM модуля Lab1_COM.asm, который определяет тип ПК и версию его системы. Далее после компилирования был получен «плохой» .EXE модуль Lab1_COM.exe. При помощи EXE2BIN.EXE и «плохого» модуля был получен «хороший» .COM модуль Lab1_COM.com.

Рис. 1. - Пример работы "плохого" модуля .EXE Lab1_COM.exe

Рис. 2. - Пример работы "хорошего" .COM модуля Lab1_COM.com

2 шаг:

Был написан исходный текст .EXE модуля lab1_exe.asm, который выполняет те же функции, что и модуль в Шаге 1. Далее был получен «хороший» .EXE модуль lab1_exe.exe.

Рис. 3. - Пример работы хорошего .EXE модуля lab1_exe.exe

3 шаг:

«Отличия исходных текстов .COM и .EXE программ»

1) Сколько сегментов должна содержать COM-программа?

COM-программа должна содержать только один сегмент, потому что данные программы и сам хранятся в одном сегменте, а стек автоматически устанавливается на последнюю ячейку сегмента.

2) 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 COM-программы?

В COM-программе обязательно должна быть директива ORG 100h. Данная директива устанавливает CS:IP на конец PSP, так как после загрузки все сегментные регистры (как и CS) указывают на начало PSP, а IP = 0, а это значит, что программа не будет выполняться, начиная с этого адреса. Именно эта директива смещает все относительные адреса на 100h байт.

В COM-программе обязательно должна быть директива ASSUME. Данная директива указывает ассемблеру с каким сегментом или группой сегментов связаны регистры.

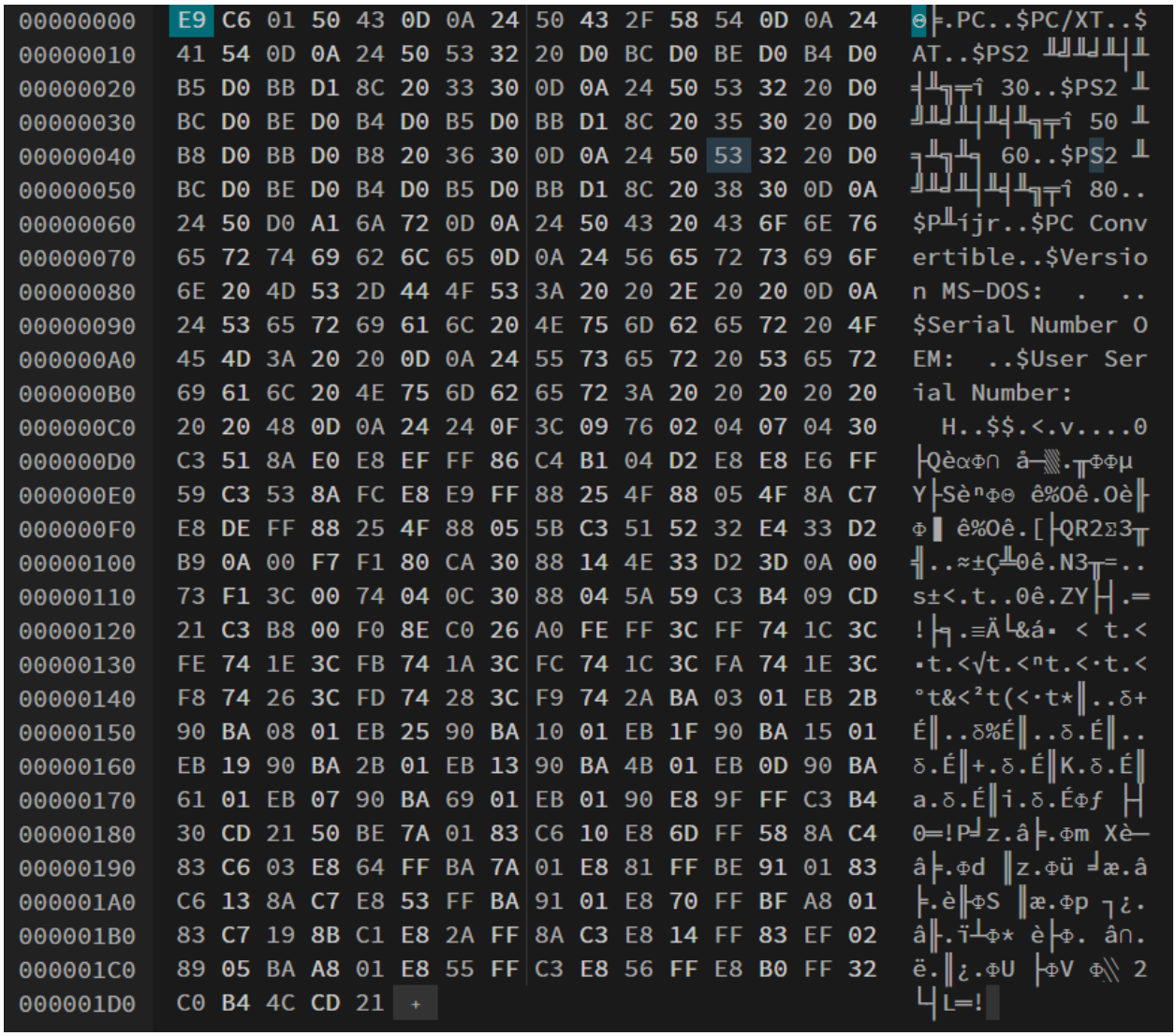
В COM-программе обязательно должна быть директива END. Данная директива завершает работу программы на ассемблере.

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.

Шаг 4:

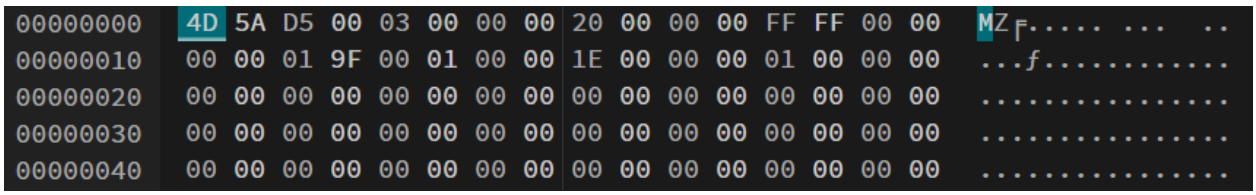
Шестнадцатеричное представление модуля .COM:



```
00000000  E9 C6 01 50 43 0D 0A 24 50 43 2F 58 54 0D 0A 24  Θ╞.PC..$PC/XT..$
00000010  41 54 0D 0A 24 50 53 32 20 D0 BC D0 BE D0 B4 D0  AT..$PS2 ╨╝╨╛╨┤╨
00000020  B5 D0 BB D1 8C 20 33 30 0D 0A 24 50 53 32 20 D0  ╡╨╗╤î 30..$PS2 ╨
00000030  BC D0 BE D0 B4 D0 B5 D0 BB D1 8C 20 35 30 20 D0  ╝╨╛╨┤╨╡╨╗╤î 50 ╨
00000040  B8 D0 BB D0 B8 20 36 30 0D 0A 24 50 53 32 20 D0  ╕╨╗╨╕ 60..$PS2 ╨
00000050  BC D0 BE D0 B4 D0 B5 D0 BB D1 8C 20 38 30 0D 0A  ╝╨╛╨┤╨╡╨╗╤î 80..
00000060  24 50 D0 A1 6A 72 0D 0A 24 50 43 20 43 6F 6E 76  $P╨íjr..$PC Conv
00000070  65 72 74 69 62 6C 65 0D 0A 24 56 65 72 73 69 6F  ertible..$Versio
00000080  6E 20 4D 53 2D 44 4F 53 3A 20 20 2E 20 20 0D 0A  n MS-DOS:  .  ..
00000090  24 53 65 72 69 61 6C 20 4E 75 6D 62 65 72 20 4F  $Serial Number O
000000A0  45 4D 3A 20 20 0D 0A 24 55 73 65 72 20 53 65 72  EM:  ..$User Ser
000000B0  69 61 6C 20 4E 75 6D 62 65 72 3A 20 20 20 20 20  ial Number:
000000C0  20 20 48 0D 0A 24 24 0F 3C 09 76 02 04 07 04 30    H..$$.<.v....0
000000D0  C3 51 8A E0 E8 EF FF 86 C4 B1 04 D2 E8 E8 E6 FF  ├Qèα╪∩ å╤▒.╥╪╪µ
000000E0  59 C3 53 8A FC E8 E9 FF 88 25 4F 88 05 4F 8A C7  Y├Sèⁿ╪Θ ê%Oê.Oè╟
000000F0  E8 DE FF 88 25 4F 88 05 5B C3 51 52 32 E4 33 D2  Φ▌ ê%Oê.[├QR2Σ3╥
00000100  B9 0A 00 F7 F1 80 CA 30 88 14 4E 33 D2 3D 0A 00  ╣..≈±Ç╨0ê.N3╥═..
00000110  73 F1 3C 00 74 04 0C 30 88 04 5A 59 C3 B4 09 CD  s±<.t..0ê.ZY├┤.═
00000120  21 C3 B8 00 F0 8E C0 26 A0 FE FF 3C FF 74 1C 3C  !├╕.≡Ä└&á▪ < t.<
00000130  FE 74 1E 3C FB 74 1A 3C FC 74 1C 3C FA 74 1E 3C  ▪t.<√t.<ⁿt.<·t.<
00000140  F8 74 26 3C FD 74 28 3C F9 74 2A BA 03 01 EB 2B  °t&<²t(<·t*╢..δ+
00000150  90 BA 08 01 EB 25 90 BA 10 01 EB 1F 90 BA 15 01  É╢..δ%É╢..δ.É╢..
00000160  EB 19 90 BA 2B 01 EB 13 90 BA 4B 01 EB 0D 90 BA  δ.É╢+.δ.É╢K.δ.É╢
00000170  61 01 EB 07 90 BA 69 01 EB 01 90 E8 9F FF C3 B4  a.δ.É╢i.δ.ÉΦƒ ├┤
00000180  30 CD 21 50 BE 7A 01 83 C6 10 E8 6D FF 58 8A C4  0═!P╛z.â╞.Φm Xè─
00000190  83 C6 03 E8 64 FF BA 7A 01 E8 81 FF BE 91 01 83  â╞.Φd ╢z.Φü ╛æ.â
000001A0  C6 13 8A C7 E8 53 FF BA 91 01 E8 70 FF BF A8 01  ╞.è╟ΦS ╢æ.Φp ┐¿.
000001B0  83 C7 19 8B C1 E8 2A FF 8A C3 E8 14 FF 83 EF 02  â╟.ï┴Φ* è├Φ. âΩ.
000001C0  89 05 BA A8 01 E8 55 FF C3 E8 56 FF E8 B0 FF 32  ë.╢¿.ΦU ├ΦV Φ╲╲ 2
000001D0  C0 B4 4C CD 21                                  └┤L═!
```

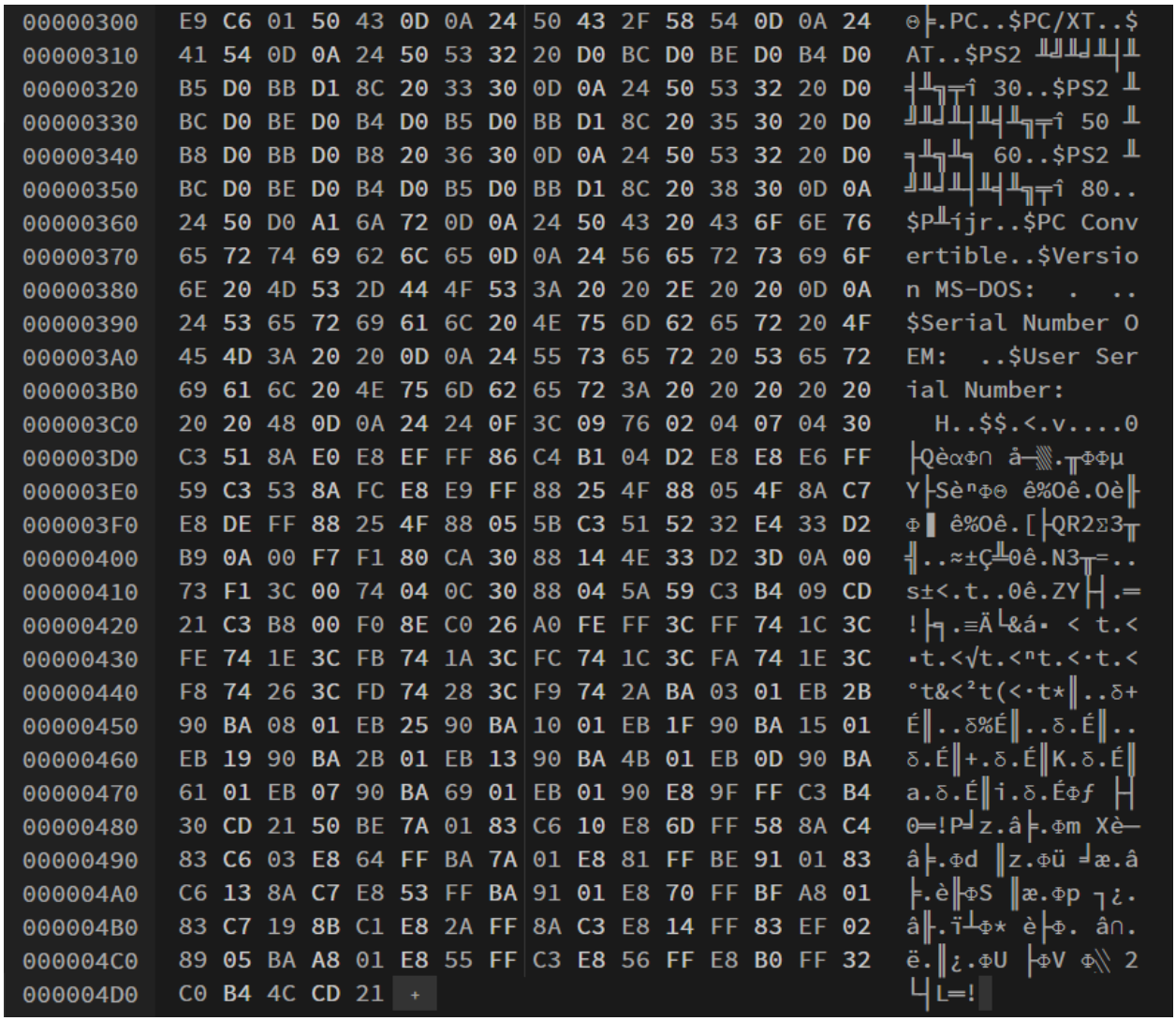
Рис. 4. - Шестнадцатеричное представление модуля .COM:

Шестнадцатеричное представление плохого модуля .EXE:



```
00000000  4D 5A D5 00 03 00 00 00 20 00 00 00 FF FF 00 00  MZ┌..... ...  ..
00000010  00 00 01 9F 00 01 00 00 1E 00 00 00 01 00 00 00  ...ƒ............
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  ................
```

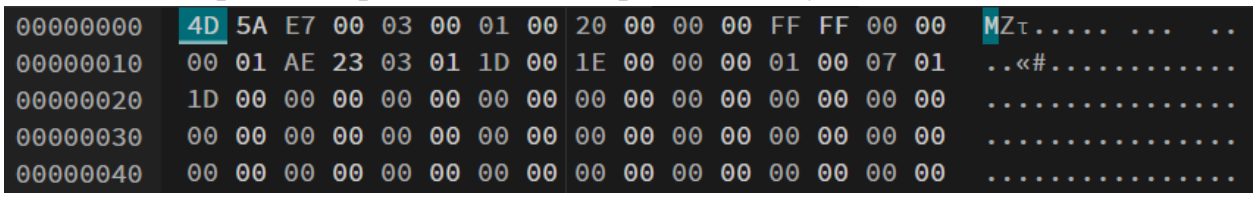
Дальше идут нули до 300h строки



```
00000300  E9 C6 01 50 43 0D 0A 24 50 43 2F 58 54 0D 0A 24  Θ╞.PC..$PC/XT..$
00000310  41 54 0D 0A 24 50 53 32 20 D0 BC D0 BE D0 B4 D0  AT..$PS2 ╨╨╨╜╨
00000320  B5 D0 BB D1 8C 20 33 30 0D 0A 24 50 53 32 20 D0  ╡╨╗╤î 30..$PS2 ╨
00000330  BC D0 BE D0 B4 D0 B5 D0 BB D1 8C 20 35 30 20 D0  ╨╨╜╨╡╨╗╤î 50 ╨
00000340  B8 D0 BB D0 B8 20 36 30 0D 0A 24 50 53 32 20 D0  ╕╨╗╨╕ 60..$PS2 ╨
00000350  BC D0 BE D0 B4 D0 B5 D0 BB D1 8C 20 38 30 0D 0A  ╨╨╜╨╡╨╗╤î 80..
00000360  24 50 D0 A1 6A 72 0D 0A 24 50 43 20 43 6F 6E 76  $P╨ijr..$PC Conv
00000370  65 72 74 69 62 6C 65 0D 0A 24 56 65 72 73 69 6F  ertible..$Versio
00000380  6E 20 4D 53 2D 44 4F 53 3A 20 20 2E 20 20 0D 0A  n MS-DOS:  .  ..
00000390  24 53 65 72 69 61 6C 20 4E 75 6D 62 65 72 20 4F  $Serial Number O
000003A0  45 4D 3A 20 20 0D 0A 24 55 73 65 72 20 53 65 72  EM:  ..$User Ser
000003B0  69 61 6C 20 4E 75 6D 62 65 72 3A 20 20 20 20 20  ial Number:
000003C0  20 20 48 0D 0A 24 24 0F 3C 09 76 02 04 07 04 30    H..$$.<.v....0
000003D0  C3 51 8A E0 E8 EF FF 86 C4 B1 04 D2 E8 E8 E6 FF  ├Qèαφ∩ å─▒.╥ΦΦµ
000003E0  59 C3 53 8A FC E8 E9 FF 88 25 4F 88 05 4F 8A C7  Y├Sèⁿφ⊖ ê%Oê.Oè╟
000003F0  E8 DE FF 88 25 4F 88 05 5B C3 51 52 32 E4 33 D2  Φ▌ ê%Oê.[├QR2Σ3╥
00000400  B9 0A 00 F7 F1 80 CA 30 88 14 4E 33 D2 3D 0A 00  ╣..≈±Ç╠0ê.N3╥=..
00000410  73 F1 3C 00 74 04 0C 30 88 04 5A 59 C3 B4 09 CD  s±<.t..0ê.ZY├┤.═
00000420  21 C3 B8 00 F0 8E C0 26 A0 FE FF 3C FF 74 1C 3C  !├╕.≡Ä└&á▪ < t.<
00000430  FE 74 1E 3C FB 74 1A 3C FC 74 1C 3C FA 74 1E 3C  ▪t.<√t.<ⁿt.<·t.<
00000440  F8 74 26 3C FD 74 28 3C F9 74 2A BA 03 01 EB 2B  °t&<²t(<·t*╢..δ+
00000450  90 BA 08 01 EB 25 90 BA 10 01 EB 1F 90 BA 15 01  É╢..δ%É╢..δ.É╢..
00000460  EB 19 90 BA 2B 01 EB 13 90 BA 4B 01 EB 0D 90 BA  δ.É╢+.δ.É╢K.δ.É╢
00000470  61 01 EB 07 90 BA 69 01 EB 01 90 E8 9F FF C3 B4  a.δ.É╢i.δ.ÉΦƒ ├┤
00000480  30 CD 21 50 BE 7A 01 83 C6 10 E8 6D FF 58 8A C4  0═!P╛z.â╞.Φm Xè─
00000490  83 C6 03 E8 64 FF BA 7A 01 E8 81 FF BE 91 01 83  â╞.Φd ╢z.Φü ╛æ.â
000004A0  C6 13 8A C7 E8 53 FF BA 91 01 E8 70 FF BF A8 01  ╞.è╟ΦS ╢æ.Φp ┐¿.
000004B0  83 C7 19 8B C1 E8 2A FF 8A C3 E8 14 FF 83 EF 02  â╟.ï┴Φ* è├Φ. â∩.
000004C0  89 05 BA A8 01 E8 55 FF C3 E8 56 FF E8 B0 FF 32  ë.╢¿.ΦU ├ΦV Φ╲╲ 2
000004D0  C0 B4 4C CD 21                                   └┤L═!
```

Рис. 5. - Шестнадцатеричное представление плохого модуля .EXE

Шестнадцатеричное представление хорошего модуля .EXE:



```
00000000  4D 5A E7 00 03 00 01 00 20 00 00 00 FF FF 00 00  MZτ..... ...  ..
00000010  00 01 AE 23 03 01 1D 00 1E 00 00 00 01 00 07 01  ..«#............
00000020  1D 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  ................
```

Рис. 6. - Шестнадцатеричное представление хорошего модуля .EXE

«Отличие форматов файлов COM и EXE модулей»

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

В данном файле код, данные и стек находятся в одном сегменте. Код и данные начинаются с адреса 0h (См. Рис. 4).

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

В «плохом» EXE файле код, данные и стек находятся в одном сегменте. Код и данные начинаются с адреса 300h. С адреса 0h находится управляющая информация загрузчика, которая содержит заголовок и таблицу настроек. (См. Рис. 5-6)

3) Какова структура «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.

Шаг 5:

«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.

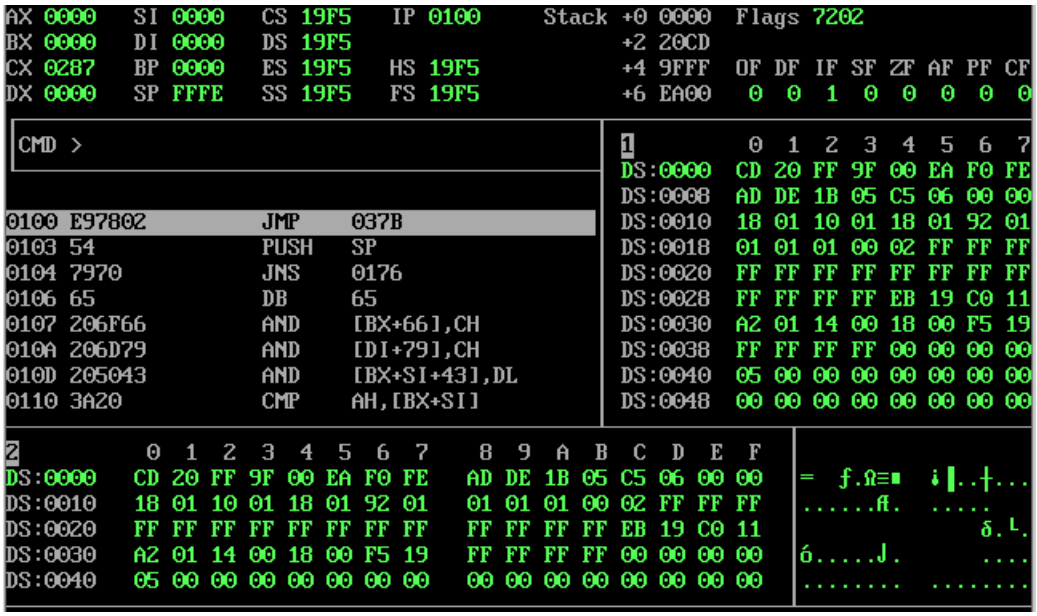


Рис. 9. - .COM в отладчике

2) Что располагается с адреса 0?

С адреса 0 располагается PSP размером в 100h байт.

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

Регистры DS, ES, CS, SS указывают на начало блока PSP.

4) Как определяется стек? Какую область 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.

Шаг 6:

«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.

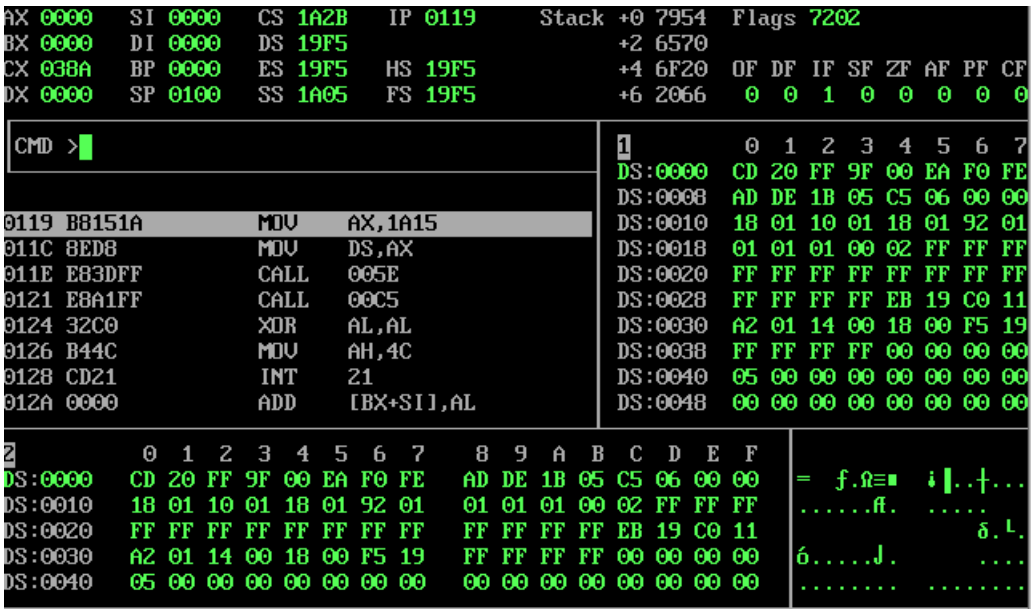


Рис. 10. - .EXE в отладчике

2) На что указывают регистры DS и ES?

ES и DS указывают 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.

Заключение.

В результате выполнения лабораторной работы были изучены структурные отличия .COM и .EXE модулей и получены навыки работы с отладчиком TD.EXE.