МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Операционные системы»

Тема: Исследование структур загрузочных модулей

Студентка гр. 9383 _____ Карпекина А.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

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.

TETR_TO_HEX - перевод десятичной цифры в код символа (записывается в AL)

BYTE_TO_HEX - переводит байт из AL в два символа шестнадцатеричного числа в AX

WRD_TO_HEX - перевод в 16 с/с 16-ти разрядного числа (в AX число, DI - адрес последнего символа)

BYTE_TO_DEC - перевод в 10 с/с (SI - адрес поля младшей цифры)

PC_T - вывод в консоль типа PC

OS_T - вывод в консоль номера основной версии, серийный номер OEM и серийный номер пользователя

Выполнение работы.

Изначально написан код исходного модуля .COM, для реализации основной задачи - определения типа PC и версии системы. После линковки был получен “плохой” .EXE модуль из которого был получен “хороший” .COM модуль. Был написан код .EXE модуля с теми же структурами и функциями для получения “хорошего” .EXE модуля.

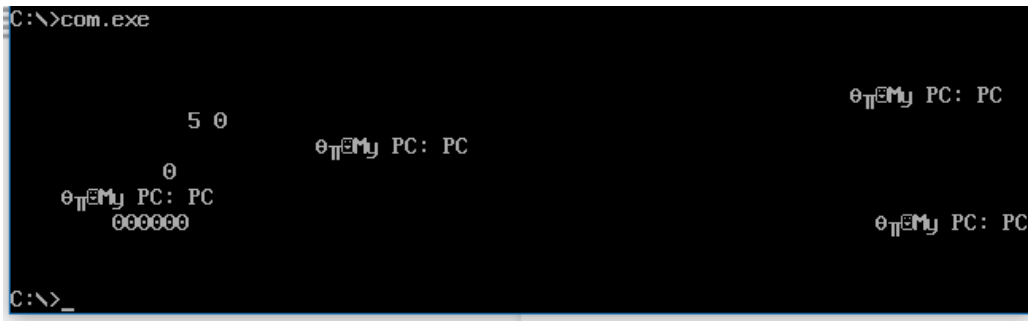


Рисунок 1 - Пример работы “плохого” .EXE модуля

Рисунок 2 - Пример работы “хорошего” .COM модуля

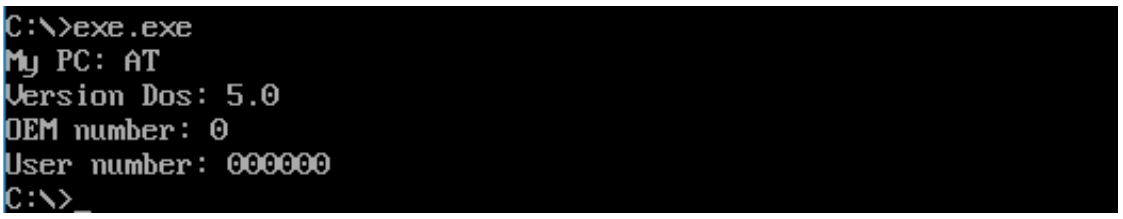


Рисунок 3 - Пример работы “хорошего” .EXE модуля

Ответы на вопросы.

“Отличия исходных текстов .COM и .EXE программ”

1) Сколько сегментов должна содержать COM программа?

COM-программы содержат единственный сегмент (или, во всяком случае, не содержат явных ссылок на другие сегменты).

2) EXE-программа?

EXE-программы содержат несколько программных сегментов, включая сегмент кода, данных и стека.

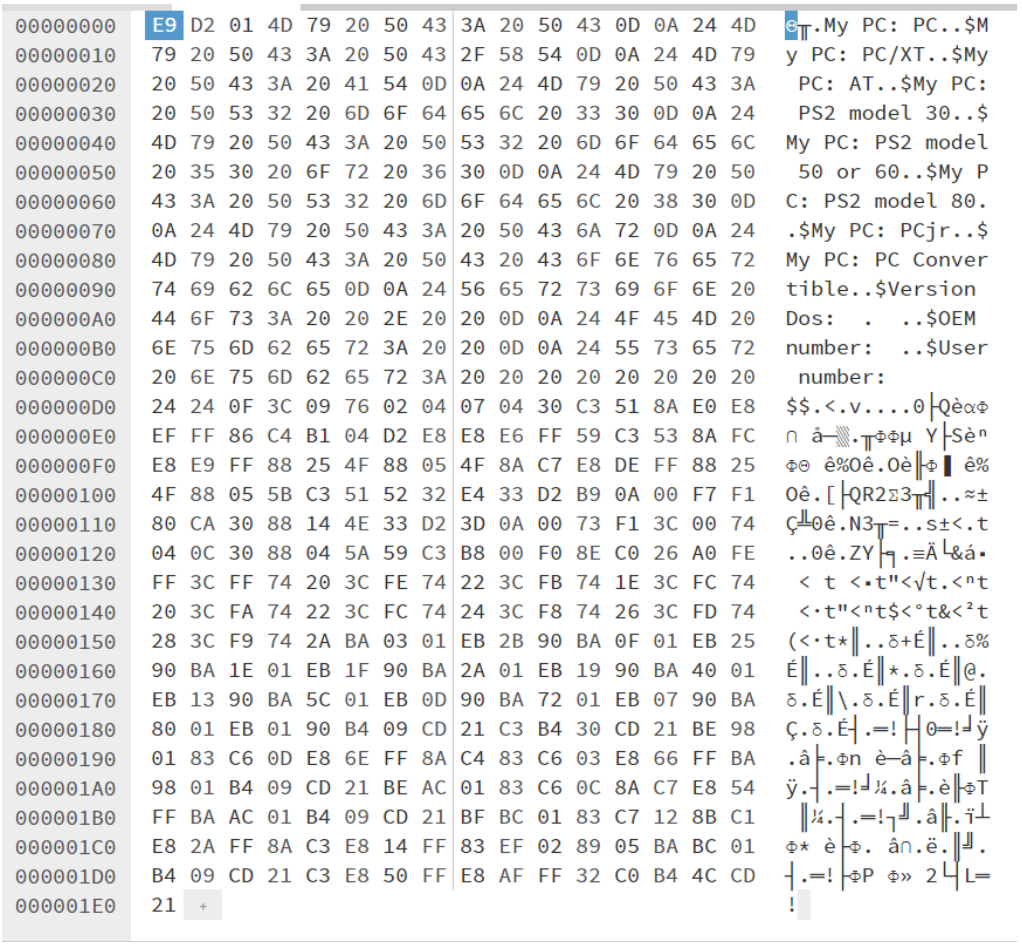
3) Какие директивы должны обязательно быть в тексте COM-программы?

Должна быть обязательна директива ORG 100h из-за смещения в 256 байт от нулевого адреса(COM программа грузится после PSP размеров в 100h). Также необходима директива ASSUME для того, чтобы сегмент данных и сегмент кода указывали на один общий сегмент. Кроме этого необходима директива END для завершения работы программы.

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

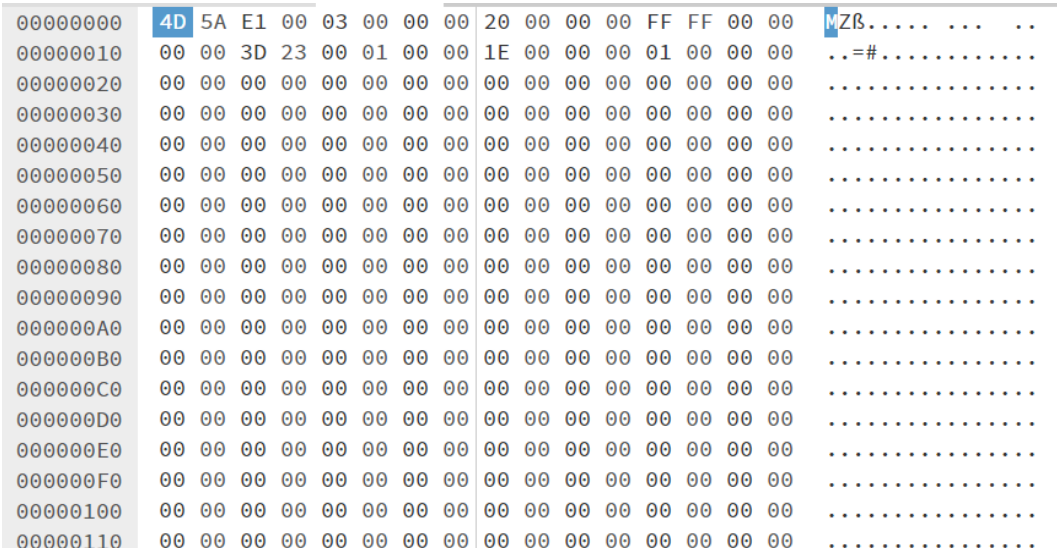
Нельзя использовать команды вида mov <регистр>, seg <имя сегмента>. Это происходит из-за того, что в .COM файле нет таблицы настроек в которой хранится информация о типе адресов и их расположении в коде. Таблица настроек подключается на этапе линковки.

Отличия форматов файлов COM и EXE модулей.



```
00000000  E9 D2 01 4D 79 20 50 43 3A 20 50 43 0D 0A 24 4D  ...My PC: PC..$M
00000010  79 20 50 43 3A 20 50 43 2F 58 54 0D 0A 24 4D 79  y PC: PC/XT..$My
00000020  20 50 43 3A 20 41 54 0D 0A 24 4D 79 20 50 43 3A   PC: AT..$My PC:
00000030  20 50 53 32 20 6D 6F 64 65 6C 20 33 30 0D 0A 24   PS2 model 30..$
00000040  4D 79 20 50 43 3A 20 50 53 32 20 6D 6F 64 65 6C  My PC: PS2 model
00000050  20 35 30 20 6F 72 20 36 30 0D 0A 24 4D 79 20 50   50 or 60..$My P
00000060  43 3A 20 50 53 32 20 6D 6F 64 65 6C 20 38 30 0D  C: PS2 model 80.
00000070  0A 24 4D 79 20 50 43 3A 20 50 43 6A 72 0D 0A 24  .$My PC: PCjr..$
00000080  4D 79 20 50 43 3A 20 50 43 20 43 6F 6E 76 65 72  My PC: PC Conver
00000090  74 69 62 6C 65 0D 0A 24 56 65 72 73 69 6F 6E 20  tible..$Version
000000A0  44 6F 73 3A 20 20 2E 20 20 0D 0A 24 4F 45 4D 20  Dos:  .  ..$OEM
000000B0  6E 75 6D 62 65 72 3A 20 20 0D 0A 24 55 73 65 72  number:  ..$User
000000C0  20 6E 75 6D 62 65 72 3A 20 20 20 20 20 20 20 20   number:
000000D0  24 24 0F 3C 09 76 02 04 07 04 30 C3 51 8A E0 E8  $$.<.v....0├Qèα╪
000000E0  EF FF 86 C4 B1 04 D2 E8 E8 E6 FF 59 C3 53 8A FC  ∩ å▒.╥╪╪µ Y├Sèⁿ
000000F0  E8 E9 FF 88 25 4F 88 05 4F 8A C7 E8 DE FF 88 25  ╪Θ ê%Oê.Oè╟╪▌ ê%
00000100  4F 88 05 5B C3 51 52 32 E4 33 D2 B9 0A 00 F7 F1  Oê.[├QR2Σ3╥╣..≈±
00000110  80 CA 30 88 14 4E 33 D2 3D 0A 00 73 F1 3C 00 74  Ç╩0ê.N3╥=..s±<.t
00000120  04 0C 30 88 04 5A 59 C3 B8 00 F0 8E C0 26 A0 FE  ..0ê.ZY├╕.≡Ä└&á▪
00000130  FF 3C FF 74 20 3C FE 74 22 3C FB 74 1E 3C FC 74   < t <▪t"<√t.<ⁿt
00000140  20 3C FA 74 22 3C FC 74 24 3C F8 74 26 3C FD 74   <·t"<ⁿt$<°t&<²t
00000150  28 3C F9 74 2A BA 03 01 EB 2B 90 BA 0F 01 EB 25  (<·t*╢..δ+É╢..δ%
00000160  90 BA 1E 01 EB 1F 90 BA 2A 01 EB 19 90 BA 40 01  É╢..δ.É╢*.δ.É╢@.
00000170  EB 13 90 BA 5C 01 EB 0D 90 BA 72 01 EB 07 90 BA  δ.É╢\.δ.É╢r.δ.É╢
00000180  80 01 EB 01 90 B4 09 CD 21 C3 B4 30 CD 21 BE 98  Ç.δ.É┤.═!├┤0═!╜ÿ
00000190  01 83 C6 0D E8 6E FF 8A C4 83 C6 03 E8 66 FF BA  .â╞.Φn è═â╞.Φf ╢
000001A0  98 01 B4 09 CD 21 BE AC 01 83 C6 0C 8A C7 E8 54  ÿ.┤.═!╜¼.â╞.è╟ΦT
000001B0  FF BA AC 01 B4 09 CD 21 BF BC 01 83 C7 12 8B C1   ╢¼.┤.═!┐╝.â╟.ï┴
000001C0  E8 2A FF 8A C3 E8 14 FF 83 EF 02 89 05 BA BC 01  Φ* è╞Φ. â∩.ë.╢╝.
000001D0  B4 09 CD 21 C3 E8 50 FF E8 AF FF 32 C0 B4 4C CD  ┤.═!├ΦP Φ» 2└┤L═
000001E0  21                                               !
```

Рисунок 4 - Шестнадцатеричное представление .COM модуля



```
00000000  4D 5A E1 00 03 00 00 00 20 00 00 00 FF FF 00 00  MZß..... ...  ..
00000010  00 00 3D 23 00 01 00 00 1E 00 00 00 01 00 00 00  ..=#............
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  ................
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

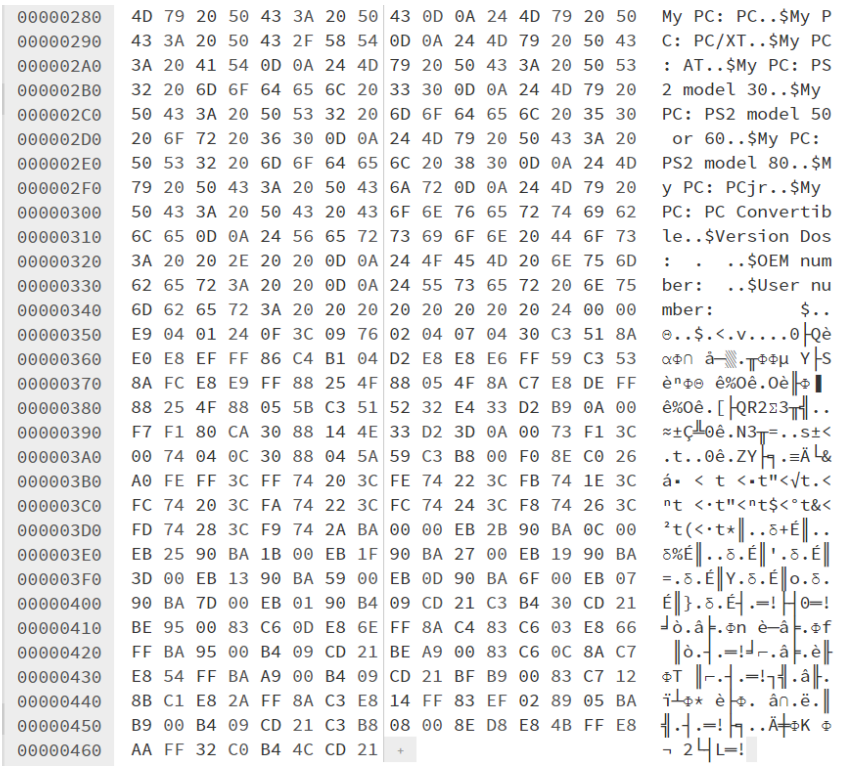
Рисунок 5 - представление “плохого” .EXE модуля

```
00000300  E9 D2 01 4D 79 20 50 43 3A 20 50 43 0D 0A 24 4D  ⊖╥.My PC: PC..$M
00000310  79 20 50 43 3A 20 50 43 2F 58 54 0D 0A 24 4D 79  y PC: PC/XT..$My
00000320  20 50 43 3A 20 41 54 0D 0A 24 4D 79 20 50 43 3A   PC: AT..$My PC:
00000330  20 50 53 32 20 6D 6F 64 65 6C 20 33 30 0D 0A 24   PS2 model 30..$
00000340  4D 79 20 50 43 3A 20 50 53 32 20 6D 6F 64 65 6C  My PC: PS2 model
00000350  20 35 30 20 6F 72 20 36 30 0D 0A 24 4D 79 20 50   50 or 60..$My P
00000360  43 3A 20 50 53 32 20 6D 6F 64 65 6C 20 38 30 0D  C: PS2 model 80.
00000370  0A 24 4D 79 20 50 43 3A 20 50 43 6A 72 0D 0A 24  .$My PC: PCjr..$
00000380  4D 79 20 50 43 3A 20 50 43 20 43 6F 6E 76 65 72  My PC: PC Conver
00000390  74 69 62 6C 65 0D 0A 24 56 65 72 73 69 6F 6E 20  tible..$Version
000003A0  44 6F 73 3A 20 20 2E 20 20 0D 0A 24 4F 45 4D 20  Dos:  .  ..$OEM
000003B0  6E 75 6D 62 65 72 3A 20 20 0D 0A 24 55 73 65 72  number:  ..$User
000003C0  20 6E 75 6D 62 65 72 3A 20 20 20 20 20 20 20 20   number:
000003D0  24 24 0F 3C 09 76 02 04 07 04 30 C3 51 8A E0 E8  $$.<.v....0├Qèα╪
000003E0  EF FF 86 C4 B1 04 D2 E8 E8 E6 FF 59 C3 53 8A FC  ∩ â─▒·╥╪╪µ Y├Sèⁿ
000003F0  E8 E9 FF 88 25 4F 88 05 4F 8A C7 E8 DE FF 88 25  ╪Θ ê%Oê.Oè╟╪ ▌ ê%
00000400  4F 88 05 5B C3 51 52 32 E4 33 D2 B9 0A 00 F7 F1  Oê.[├QR2Σ3╥╣..≈±
00000410  80 CA 30 88 14 4E 33 D2 3D 0A 00 73 F1 3C 00 74  Ç╩0ê.N3╥=..s±<.t
00000420  04 0C 30 88 04 5A 59 C3 B8 00 F0 8E C0 26 A0 FE  ..0ê.ZY├╕.≡Ä└&á▪
00000430  FF 3C FF 74 20 3C FE 74 22 3C FB 74 1E 3C FC 74   < t <▪t"<√t.<ⁿt
00000440  20 3C FA 74 22 3C FC 74 24 3C F8 74 26 3C FD 74   <·t"<ⁿt$<°t&<²t
00000450  28 3C F9 74 2A BA 03 01 EB 2B 90 BA 0F 01 EB 25  (<·t*╢..δ+É╢..δ%
00000460  90 BA 1E 01 EB 1F 90 BA 2A 01 EB 19 90 BA 40 01  É╢..δ.É╢*.δ.É╢@.
00000470  EB 13 90 BA 5C 01 EB 0D 90 BA 72 01 EB 07 90 BA  δ.É╢\.δ.É╢r.δ.É╢
00000480  80 01 EB 01 90 B4 09 CD 21 C3 B4 30 CD 21 BE 98  Ç.δ.É┤.=!─┤0=!╜ÿ
00000490  01 83 C6 0D E8 6E FF 8A C4 83 C6 03 E8 66 FF BA  .â╞.╪n è─â╞.╪f ╢
000004A0  98 01 B4 09 CD 21 BE AC 01 83 C6 0C 8A C7 E8 54  ÿ.┤.=!╜¼.â╞.è╟╪T
000004B0  FF BA AC 01 B4 09 CD 21 BF BC 01 83 C7 12 8B C1   ╢½.┤.=!┐╝.â╟.ï┴
000004C0  E8 2A FF 8A C3 E8 14 FF 83 EF 02 89 05 BA BC 01  ╪* è├╪. â∩.ë.╢╝.
000004D0  B4 09 CD 21 C3 E8 50 FF E8 AF FF 32 C0 B4 4C CD  ┤.=!├╪P ╪» 2└┤L═
000004E0  21                                               !
```

Рисунок 6 - представление “плохого” .EXE модуля

```
00000000  4D 5A 68 00 03 00 01 00 20 00 00 00 FF FF 00 00  MZh..... ...  ..
00000010  80 00 9B 71 00 00 15 00 1E 00 00 00 01 00 08 01  Ç.¢q............
00000020  15 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  ................
```

Рисунок 7 - представление “хорошего” .EXE модуля



```
00000280  4D 79 20 50 43 3A 20 50 43 0D 0A 24 4D 79 20 50  My PC: PC..$My P
00000290  43 3A 20 50 43 2F 58 54 0D 0A 24 4D 79 20 50 43  C: PC/XT..$My PC
000002A0  3A 20 41 54 0D 0A 24 4D 79 20 50 43 3A 20 50 53  : AT..$My PC: PS
000002B0  32 20 6D 6F 64 65 6C 20 33 30 0D 0A 24 4D 79 20  2 model 30..$My
000002C0  50 43 3A 20 50 53 32 20 6D 6F 64 65 6C 20 35 30  PC: PS2 model 50
000002D0  20 6F 72 20 36 30 0D 0A 24 4D 79 20 50 43 3A 20   or 60..$My PC:
000002E0  50 53 32 20 6D 6F 64 65 6C 20 38 30 0D 0A 24 4D  PS2 model 80..$M
000002F0  79 20 50 43 3A 20 50 43 6A 72 0D 0A 24 4D 79 20  y PC: PCjr..$My
00000300  50 43 3A 20 50 43 20 43 6F 6E 76 65 72 74 69 62  PC: PC Convertib
00000310  6C 65 0D 0A 24 56 65 72 73 69 6F 6E 20 44 6F 73  le..$Version Dos
00000320  3A 20 20 2E 20 20 0D 0A 24 4F 45 4D 20 6E 75 6D  :  .  ..$OEM num
00000330  62 65 72 3A 20 20 0D 0A 24 55 73 65 72 20 6E 75  ber:  ..$User nu
00000340  6D 62 65 72 3A 20 20 20 20 20 20 20 20 24 00 00  mber:        $..
00000350  E9 04 01 24 0F 3C 09 76 02 04 07 04 30 C3 51 8A  ⊖..$.<.v....0├Qè
00000360  E0 E8 EF FF 86 C4 B1 04 D2 E8 E8 E6 FF 59 C3 53  αΦ∩ ä─▒.╥ΦΦµ Y├S
00000370  8A FC E8 E9 FF 88 25 4F 88 05 4F 8A C7 E8 DE FF  èⁿΦΘ ê%Oê.Oè╟Φ▌
00000380  88 25 4F 88 05 5B C3 51 52 32 E4 33 D2 B9 0A 00  ê%Oê.[├QR2Σ3╥╣..
00000390  F7 F1 80 CA 30 88 14 4E 33 D2 3D 0A 00 73 F1 3C  ≈±Ç╩0ê.N3╥=..s±<
000003A0  00 74 04 0C 30 88 04 5A 59 C3 B8 00 F0 8E C0 26  .t..0ê.ZY├╕.≡Ä└&
000003B0  A0 FE FF 3C FF 74 20 3C FE 74 22 3C FB 74 1E 3C  á▪ < t <▪t"<√t.<
000003C0  FC 74 20 3C FA 74 22 3C FC 74 24 3C F8 74 26 3C  ⁿt <·t"<ⁿt$<°t&<
000003D0  FD 74 28 3C F9 74 2A BA 00 00 EB 2B 90 BA 0C 00  ²t(<·t*╟..δ+É╟..
000003E0  EB 25 90 BA 1B 00 EB 1F 90 BA 27 00 EB 19 90 BA  δ%É╟..δ.É╟'.δ.É╟
000003F0  3D 00 EB 13 90 BA 59 00 EB 0D 90 BA 6F 00 EB 07  =.δ.É╟Y.δ.É╟o.δ.
00000400  90 BA 7D 00 EB 01 90 B4 09 CD 21 C3 B4 30 CD 21  É╟}.δ.É┤.=!├┤0═!
00000410  BE 95 00 83 C6 0D E8 6E FF 8A C4 83 C6 03 E8 66  ╛ò.â╞.Φn è─â╞.Φf
00000420  FF BA 95 00 B4 09 CD 21 BE A9 00 83 C6 0C 8A C7   ╟ò.┤.═!╛⌐.â╞.è╟
00000430  E8 54 FF BA A9 00 B4 09 CD 21 BF B9 00 83 C7 12  ΦT ╟⌐.┤.═!┐╣.â╟.
00000440  8B C1 E8 2A FF 8A C3 E8 14 FF 83 EF 02 89 05 BA  ï┴Φ* è┼Φ. â∩.ë.╟
00000450  B9 00 B4 09 CD 21 C3 B8 08 00 8E D8 E8 4B FF E8  ╣.┤.═!├╕..Ä╪ΦK Φ
00000460  AA FF 32 C0 B4 4C CD 21                          ¬ 2└┤L═!
```

Рисунок 8 - представление “хорошего” .EXE модуля

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

COM-программа содержит лишь один сегмент – сегмент кода, который включает в себя инструкции процессора, директивы и описания переменных. Образ COM-файла помещается в память, начиная с 0h.

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

В «плохом» EXE данные и код располагаются в одном сегменте (это неправильно так как код и данные должны быть разделены на отдельные сегменты). Код располагается с адреса 300h, а с адреса 0h идёт таблица настроек.

3) Какова структура “хорошего” EXE? Чем он отличается от файла “плохого” EXE?

В “хорошем” EXE код, данные и стек находятся в разных сегментах, в отличии от “плохого” EXE. Отсутствует директива ORG 100h, поэтому память под PSP не выделяется.

Загрузка COM модуля в основную память.

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

Образ COM-файла считывается с диска и помещается в память, начиная с PSP:0100h.

После загрузки двоичного образа COM-программы:

- CS, DS, ES и SS указывают на PSP;
- SP указывает на конец сегмента PSP (обычно 0FFFEH, но может быть меньше, если полный 64K сегмент недоступен);
- слово по смещению 06H в PSP (доступные байты в программном сегменте) указывает, какая часть программного сегмента доступна;
- вся память системы за программным сегментом распределена программе; слово 00H помещено (PUSH) в стек.
- IP содержит 100H (первый байт модуля) в результате команды JMP PSP:100H.

Код с адреса 100H.

2) Что располагается с адреса 0?

PSP размером в 100H байт.

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

Регистры CS, DS, ES и SS указывают на PSP.

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

Стек генерируется автоматически между PSP и кодом. Он расположен между адресами SS:0000h и SS:FFFFh.

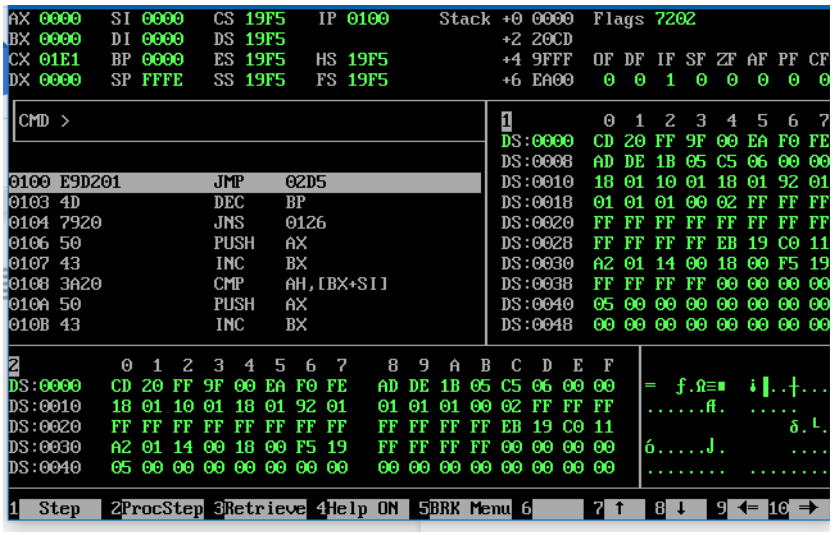


Рисунок 9 - модуль .COM в отладчике

Загрузка “хорошего” EXE модуля в основную память.

1) Как загружается “хороший” EXE? Какие значения имеют сегментные регистры?

EXE-файл загружается, начиная с адреса PSP:0100h. В процессе загрузки считывается информация заголовка EXE в начале файла и выполняется перемещение адресов сегментов.

В момент получения управления программой EXE -формата:

- DS и ES указывают на начало PSP
- CS, IP, SS и SP инициализированы значениями, указанными в заголовке EXE
- поле PSP MemTop (вершина доступной памяти системы в параграфах) содержит значение, указанное в заголовке EXE. Обычно вся доступная память распределена программе.

2) На что указывают регистры DS и ES?

DS и ES указывают на начало PSP.

3) Как определяется стек?

Стек определяется с помощью директивы .stack, после которой задается размер стека. SS и SP указывают на начало и конец стека соответственно.

4) Как определяется точка входа?

Точка входа определяется директивой END. После этой директивы указывается метка (адрес), куда перейдет программа при запуске.

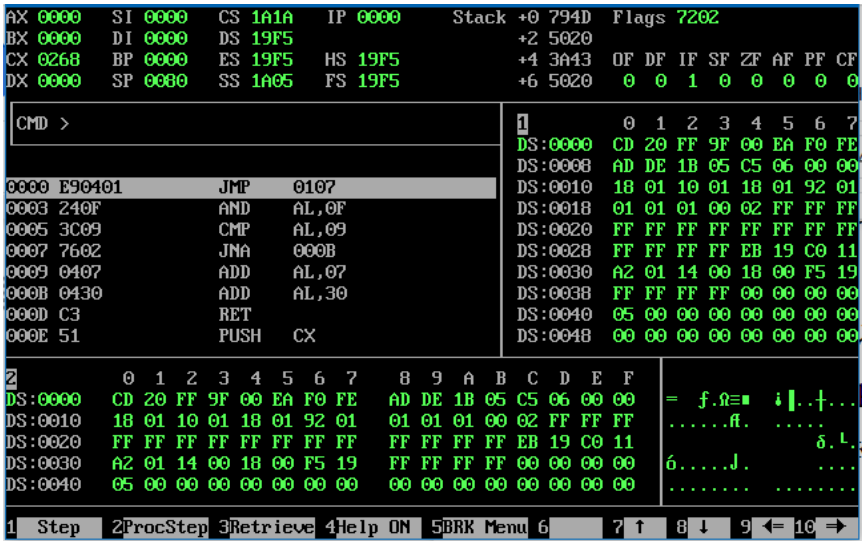


Рисунок 10 - “хороший” .EXE модуль в отладчике

Вывод.

В результате выполнения работы были написаны .COM и .EXE модули, а также изучены их структурные различия.