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. 9382 _____ Савельев И.С.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

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

Задание.

Шаг 1. Напишите текст исходного .COM модуля, который определяет тип PC версию системы. Это довольно простая задача и для тех, кто уже имеет опыт программирования на ассемблере, это будет небольшой разминкой. Для тех, кто раньше не сталкивался с программированием на ассемблере, это неплохая задача для первого опыта. За основу возьмите шаблон, приведенный в разделе «Основные сведения». Необходимые сведения о том, как извлечь требуемую информацию, представлены в следующем разделе. 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 и серийным номером пользователя.Полученные строки выводятся на экран.Отладьте полученный исходный модуль.Результатом выполнения этого шага будет «хороший» .COM модуль, а такженеобходимо построить «плохой» .EXE, полученный из исходного текста для .CAM модуля.

Шаг 2. Напишите текст исходного .EXE модуля, который выполняет те же функции,что и модуль в Шаге 1 и постройте и отладьте его. Таким образом, будет получен«хороший» .EXE.

Шаг 3. Сравните исходные тексты для .COM и .EXE модулей. Ответьте на контрольные вопросы «Отличия исходных текстов COM и EXE программ».

Шаг 4. Запустите FAR и откройте (F3/F4) файл загрузочного модуля .COM и файл«плохого» .EXE в шестнадцатеричном виде. Затем откройте (F3/F4) файл загрузочного модуля «хорошего» .EXE и сравните его с предыдущими файлами. Ответьте на контрольные вопросы «Отличия форматов файлов COM и EXE модулей».

Шаг 5. Откройте отладчик TD.EXE и загрузите .COM. Ответьте на контрольные вопросы «Загрузка COM модуля в основную память». Представьте в отчете план загрузки модуля .COM в основную память.

Шаг 6. Откройте отладчик TD.EXE и загрузите «хороший» .EXE. Ответьте на контрольные вопросы «Загрузка «хорошего» EXE модуля в основную память».

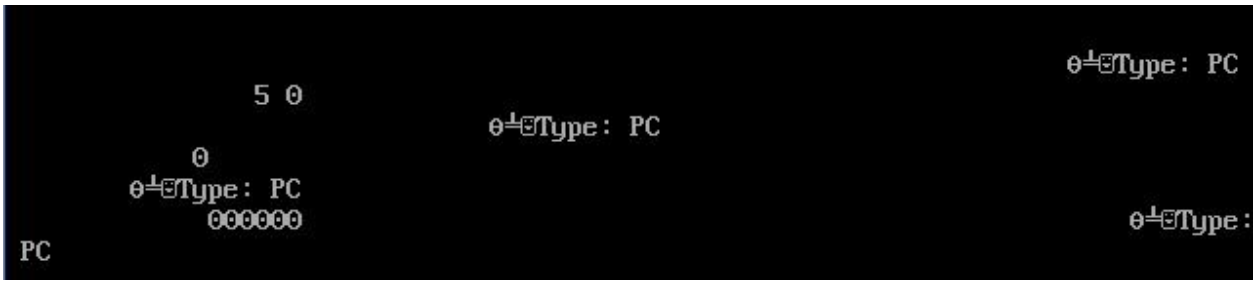
Шаг 7. Оформление отчета в соответствии с требованиями. В отчете необходимо привести скриншоты. Для файлов их вид в шестнадцатеричном виде, для загрузочных модулей – в отладчике

Ход работы.

В обоих программах в начале были объявлены константные строки для вывода информации о ПК и версии MS DOS. Также были написаны собственные функции MYPRINT - для вывода строк, DEFINE_PC_TYP - для определения типа ПК и DEFINE_OS_VER - для определения версии MS DOS.

Запуск хорошего .com модуля

Запуск плохого .exe модуля



Запуск хорошего .exe модуля

Вывод.

В результате выполнения лабораторной работы были написаны программы для определения версии PC и MS DOS. Исследованы различия в структурах исходных текстовых модулей типов .COM и .EXE, структурах файлов загрузочных модулей и способов их загрузки в основную память.

Ответы на контрольные вопросы.

Отличия исходных текстов COM и EXE программ

1) Сколько сегментов должна содержать COM-программа?

   У COM-программы 1 сегмент.

2) Сколько сегментов должна содержать EXE-программа?

У EXE-программы может быть больше одного сегмента, отдельные сегменты для данных, кода и стека.

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

В тексте COM-программы обязательно должна быть директива ORG 100h, так как в первых 256(100h) байтах находится PSP, а код программы располагается после него. Также обязательно должна присутствовать директива ASSUME, которая сообщает о том, какой сегмент к какому сегментному регистру привязан.

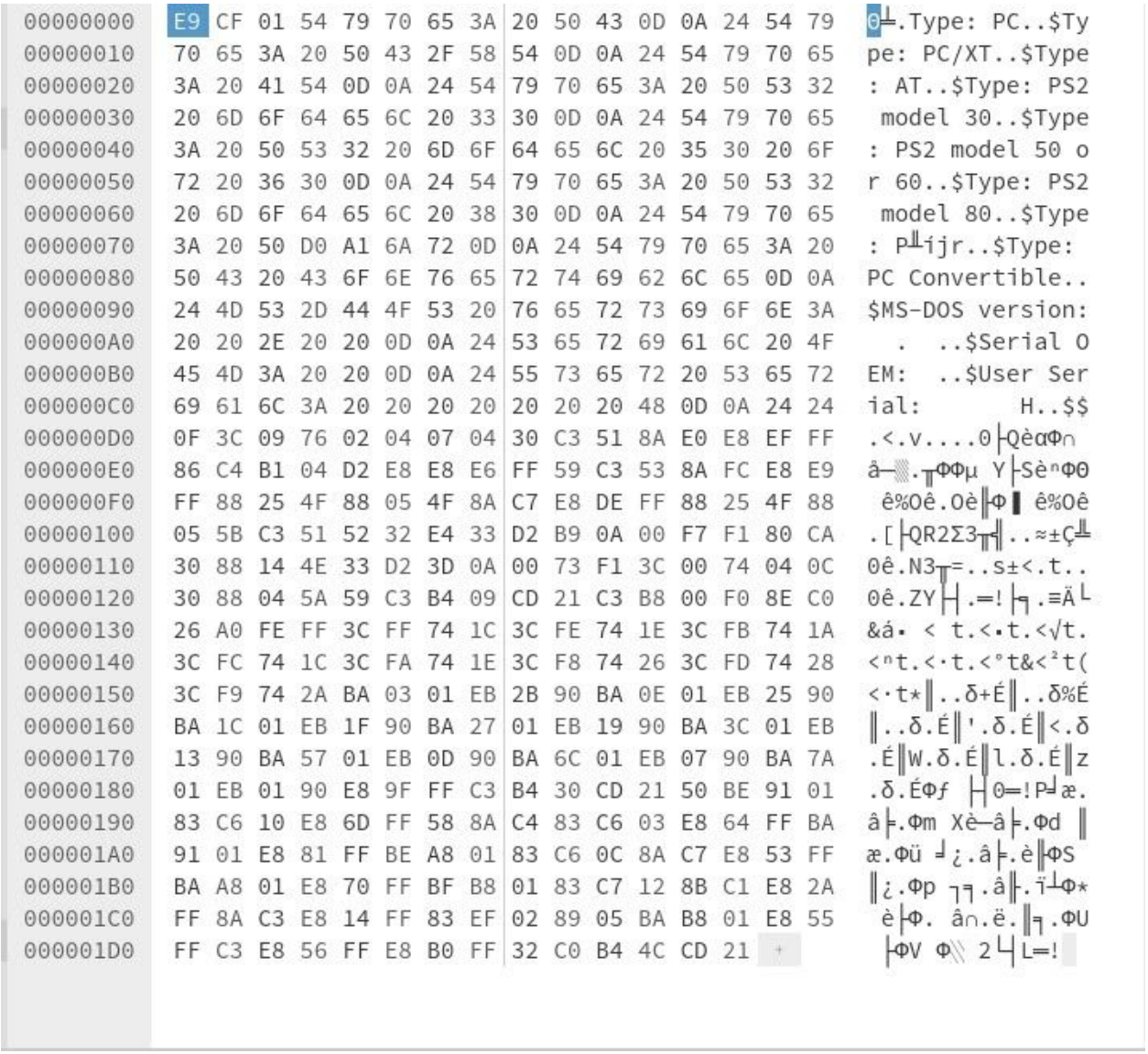
4) Все ли форматы команд можно использовать в COM-программе?

Нет, не все. Например в COM-программе нельзя использовать команды вида mov register, segment и команды, содержащие дальнюю(far) адресацию, так как для нормальной работы этим командам нужна relocation table, которой нет в COM-программах.

Отличия форматов файлов COM и EXE модулей

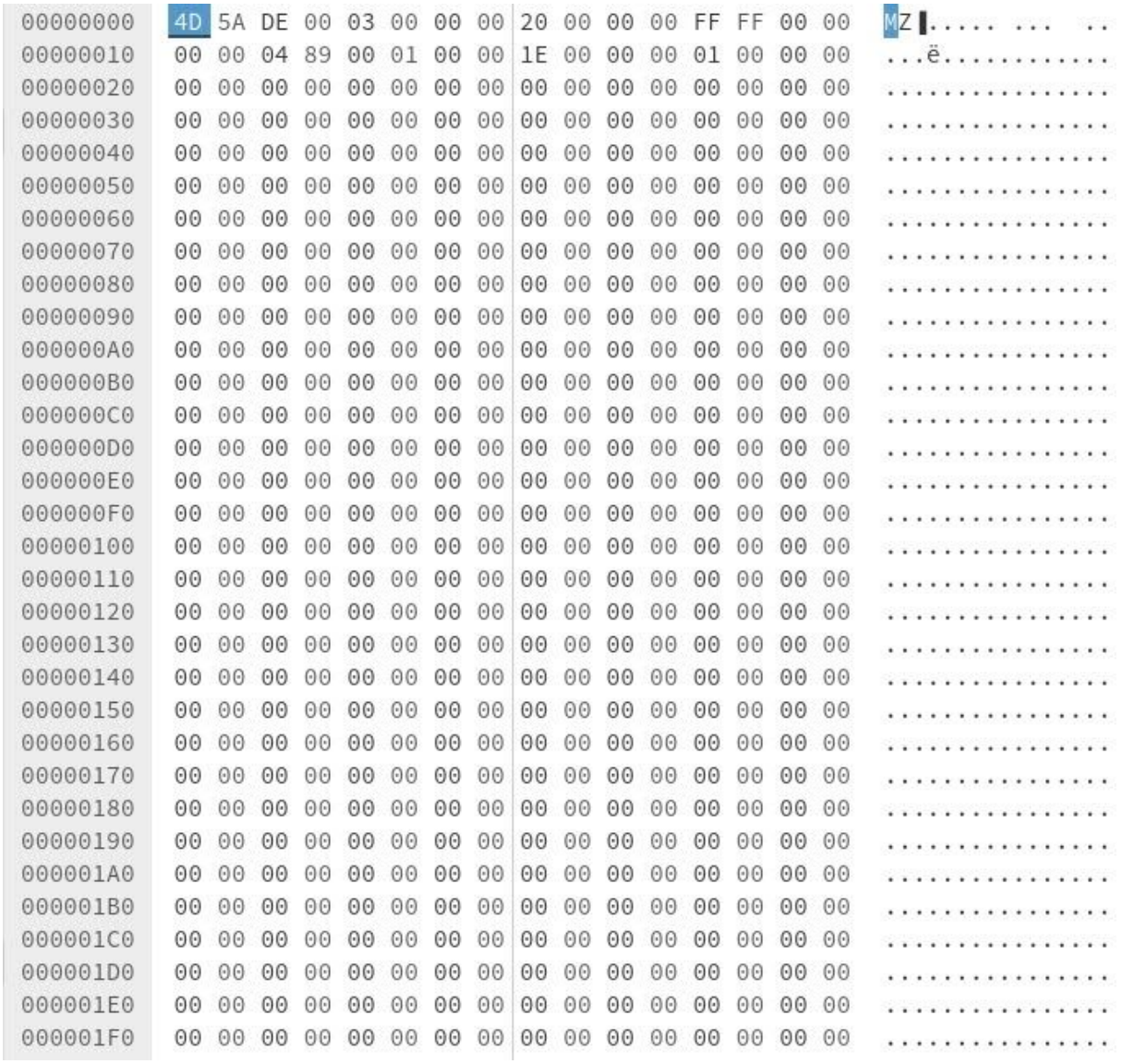
1) Какова структура файла COM? С какого адреса располагается код?

Структура состоит из одного сегмента размером не больше 64КБ. В файле код располагается с нулевого адреса, но при загрузке модуля смещается на 100h.



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

В «плохом» EXE файле данные и код содержатся в одном сегменте. Код располагается с адреса 300h. С адреса 0h располагается Relocation Table



```
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 CF 01 54 79 70 65 3A 20 50 43 0D 0A 24 54 79  Θ╧.Type: PC..$Ty
00000310  70 65 3A 20 50 43 2F 58 54 0D 0A 24 54 79 70 65  pe: PC/XT..$Type
00000320  3A 20 41 54 0D 0A 24 54 79 70 65 3A 20 50 53 32  : AT..$Type: PS2
00000330  20 6D 6F 64 65 6C 20 33 30 0D 0A 24 54 79 70 65   model 30..$Type
00000340  3A 20 50 53 32 20 6D 6F 64 65 6C 20 35 30 20 6F  : PS2 model 50 o
00000350  72 20 36 30 0D 0A 24 54 79 70 65 3A 20 50 53 32  r 60..$Type: PS2
00000360  20 6D 6F 64 65 6C 20 38 30 0D 0A 24 54 79 70 65   model 80..$Type
00000370  3A 20 50 D0 A1 6A 72 0D 0A 24 54 79 70 65 3A 20  : P╨íjr..$Type:
00000380  50 43 20 43 6F 6E 76 65 72 74 69 62 6C 65 0D 0A  PC Convertible..
00000390  24 4D 53 2D 44 4F 53 20 76 65 72 73 69 6F 6E 3A  $MS-DOS version:
000003A0  20 20 2E 20 20 0D 0A 24 53 65 72 69 61 6C 20 4F    .  ..$Serial O
000003B0  45 4D 3A 20 20 0D 0A 24 55 73 65 72 20 53 65 72  EM:  ..$User Ser
000003C0  69 61 6C 3A 20 20 20 20 20 20 20 48 0D 0A 24 24  ial:       H..$$
000003D0  0F 3C 09 76 02 04 07 04 30 C3 51 8A E0 E8 EF FF  .<.v....0├QèαΦ∩
000003E0  86 C4 B1 04 D2 E8 E8 E6 FF 59 C3 53 8A FC E8 E9  å─▒.╥ΦΦµ Y├Sèⁿ╪Θ
000003F0  FF 88 25 4F 88 05 4F 8A C7 E8 DE FF 88 25 4F 88   ê%Oê.Oè╟Φ▌ ê%Oê
00000400  05 5B C3 51 52 32 E4 33 D2 B9 0A 00 F7 F1 80 CA  .[├QR2Σ3╤╣..≈±Ç╩
00000410  30 88 14 4E 33 D2 3D 0A 00 73 F1 3C 00 74 04 0C  0ê.N3╥=..s±<.t..
00000420  30 88 04 5A 59 C3 B4 09 CD 21 C3 B8 00 F0 8E C0  0ê.ZY├┤.=!├╕.≡Ä└
00000430  26 A0 FE FF 3C FF 74 1C 3C FE 74 1E 3C FB 74 1A  &á▪ < t.<▪t.<√t.
00000440  3C FC 74 1C 3C FA 74 1E 3C F8 74 26 3C FD 74 28  <ⁿt.<·t.<°t&<²t(
00000450  3C F9 74 2A BA 03 01 EB 2B 90 BA 0E 01 EB 25 90  <·t*╢..δ+É╢..δ%É
00000460  BA 1C 01 EB 1F 90 BA 27 01 EB 19 90 BA 3C 01 EB  ╢..δ.É╢'.δ.É╢<.δ
00000470  13 90 BA 57 01 EB 0D 90 BA 6C 01 EB 07 90 BA 7A  .É╢W.δ.É╢l.δ.É╢z
00000480  01 EB 01 90 E8 9F FF C3 B4 30 CD 21 50 BE 91 01  .δ.ÉΦƒ ├┤0═!P╛æ.
00000490  83 C6 10 E8 6D FF 58 8A C4 83 C6 03 E8 64 FF BA  â╞.Φm Xè─â╞.Φd ╢
000004A0  91 01 E8 81 FF BE A8 01 83 C6 0C 8A C7 E8 53 FF  æ.Φü ╛¿.â╞.è╟ΦS
000004B0  BA A8 01 E8 70 FF BF B8 01 83 C7 12 8B C1 E8 2A  ╢¿.Φp ┐╕.â╟.ï┴Φ*
000004C0  FF 8A C3 E8 14 FF 83 EF 02 89 05 BA B8 01 E8 55   è├Φ. â∩.ë.╢┐.ΦU
000004D0  FF C3 E8 56 FF E8 B0 FF 32 C0 B4 4C CD 21        ├ΦV Φ▒ 2└┤L=!
```

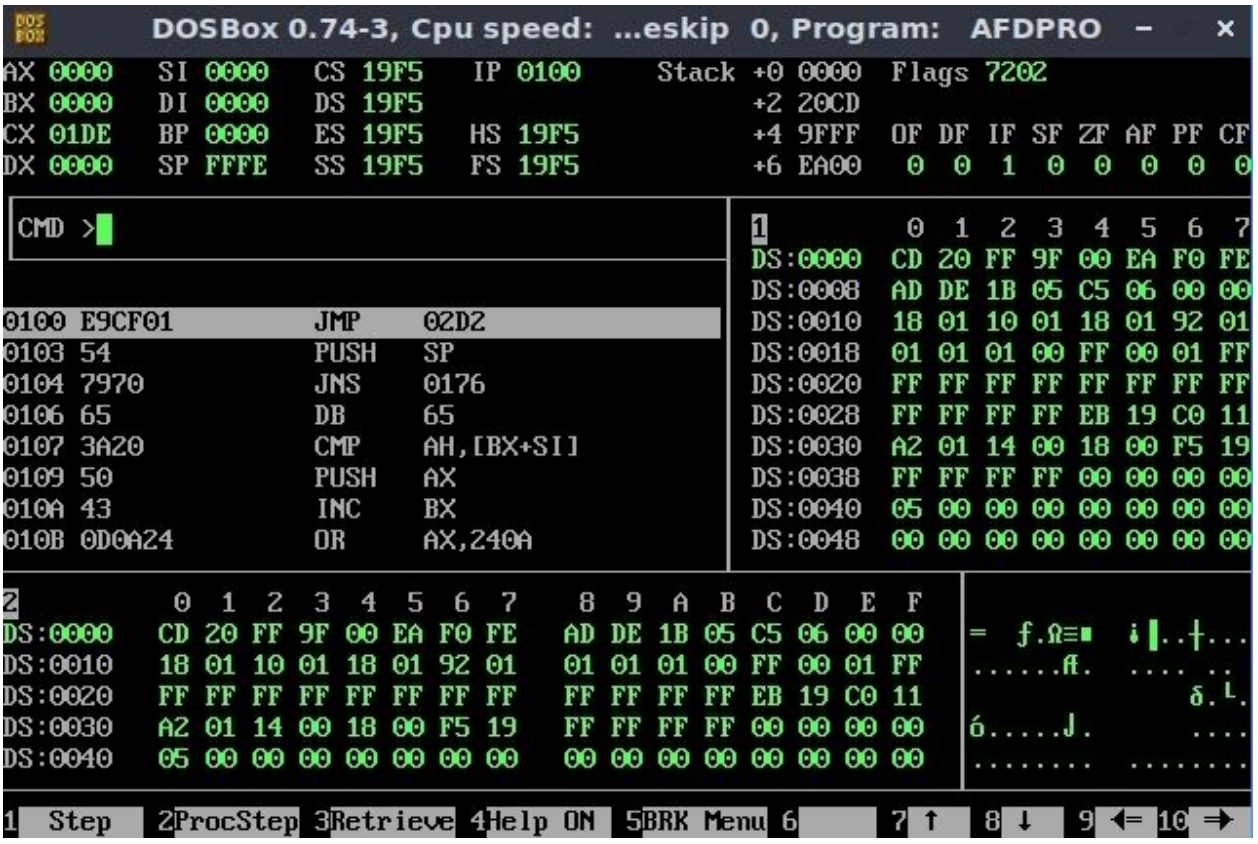
3) Какова структура файла «хорошего» EXE? Чем он отличается от файла «плохого» EXE?

В «хорошем» EXE, код, стек и данные выделены в отдельные сегменты в отличии от «плохого». В файле «плохого» EXE смещение 300h(100h изначальное смещение и 200h размер модуля PSP). У «хорошего» EXE память под стек выделяется между PSP и кодом.

```
00000000  4D 5A E7 00 03 00 01 00 20 00 00 00 FF FF 00 00  MZτ..... ...  ..
00000010  00 01 2F 99 03 01 1D 00 1E 00 00 00 01 00 07 01  ../Ö............
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  ................
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  ................
00000120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000180  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001F0  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  54 79 70 65 3A 20 50 43 0D 0A 24 54 79 70 65 3A  Type: PC..$Type:
00000310  20 50 43 2F 58 54 0D 0A 24 54 79 70 65 3A 20 41   PC/XT..$Type: A
00000320  54 0D 0A 24 54 79 70 65 3A 20 50 53 32 20 6D 6F  T..$Type: PS2 mo
00000330  64 65 6C 20 33 30 0D 0A 24 54 79 70 65 3A 20 50  del 30..$Type: P
00000340  53 32 20 6D 6F 64 65 6C 20 35 30 20 6F 72 20 36  S2 model 50 or 6
00000350  30 0D 0A 24 54 79 70 65 3A 20 50 53 32 20 6D 6F  0..$Type: PS2 mo
00000360  64 65 6C 20 38 30 0D 0A 24 54 79 70 65 3A 20 50  del 80..$Type: P
00000370  D0 A1 6A 72 0D 0A 24 54 79 70 65 3A 20 50 43 20  ╨íjr..$Type: PC
00000380  43 6F 6E 76 65 72 74 69 62 6C 65 0D 0A 24 4D 53  Convertible..$MS
00000390  2D 44 4F 53 20 76 65 72 73 69 6F 6E 3A 20 20 2E  -DOS version:  .
000003A0  20 20 0D 0A 24 53 65 72 69 61 6C 20 4F 45 4D 3A    ..$Serial OEM:
000003B0  20 20 0D 0A 24 55 73 65 72 20 53 65 72 69 61 6C    ..$User Serial
000003C0  3A 20 20 20 20 20 20 20 48 0D 0A 24 00 00 00 00  :       H..$....
000003D0  24 0F 3C 09 76 02 04 07 04 30 C3 51 8A E0 E8 EF  $.<.v....0├QèαΦ∩
000003E0  FF 86 C4 B1 04 D2 E8 E8 E6 FF 59 C3 53 8A FC E8   å─▒.╥ΦΦµ Y├Sèⁿ Φ
000003F0  E9 FF 88 25 4F 88 05 4F 8A C7 E8 DE FF 88 25 4F  0 ê%Oê.Oè╟Φ▌ ê%O
00000400  88 05 5B C3 51 52 32 E4 33 D2 B9 0A 00 F7 F1 80  ê.[├QR2Σ3╤╣..≈±Ç
00000410  CA 30 88 14 4E 33 D2 3D 0A 00 73 F1 3C 00 74 04  ╩0ê.N3╥=..s±<.t.
00000420  0C 30 88 04 5A 59 C3 B4 09 CD 21 C3 B8 00 F0 8E  .0ê.ZY├┤.=!├┐.≡Ä
00000430  C0 26 A0 FE FF 3C FF 74 1C 3C FE 74 1E 3C FB 74  └&á▪ < t.<▪t.<√t
00000440  1A 3C FC 74 1C 3C FA 74 1E 3C F8 74 26 3C FD 74  .<ⁿt.<·t.<°t&<²t
00000450  28 3C F9 74 2A BA 00 00 EB 2B 90 BA 0B 00 EB 25  (<·t*╢..δ+É╢..δ%
00000460  90 BA 19 00 EB 1F 90 BA 24 00 EB 19 90 BA 39 00  É╢..δ.É╢$.δ.É╢9.
00000470  EB 13 90 BA 54 00 EB 0D 90 BA 69 00 EB 07 90 BA  δ.É╢T.δ.É╢i.δ.É╢
00000480  77 00 EB 01 90 E8 9F FF C3 B4 30 CD 21 50 BE 8E  w.δ.ÉΦƒ ├┤0═!P╛Ä
00000490  00 83 C6 10 E8 6D FF 58 8A C4 83 C6 03 E8 64 FF  .â╞.Φm Xè─â╞.Φd
000004A0  BA 8E 00 E8 81 FF BE A5 00 83 C6 0C 8A C7 E8 53  ╢Ä.Φü ╛Ñ.â╞.è╟ΦS
000004B0  FF BA A5 00 E8 70 FF BF B5 00 83 C7 12 8B C1 E8   ╢Ñ.Φp ┐╡.â╟.ï┴Φ
000004C0  2A FF 8A C3 E8 14 FF 83 EF 02 89 05 BA B5 00 E8  * è├Φ. â∩.ë.╟╡.Φ
000004D0  55 FF C3 2B C0 50 B8 10 00 8E D8 E8 4E FF E8 A8  U ├+└P┐..Ä╪ΦN Φ¿
000004E0  FF 32 C0 B4 4C CD 21                              2└┤L═!
```

Загрузка COM модуля в основную память.



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

В начале определяется адрес сегмента оперативной памяти, в котором хватит места для загрузки программы, после чего система отводит первые 256 байт для PSP, а COM файл помещается в память со смещение 100h. Затем инициализируются сегментные регистры, которые все указывают на начало PSP. SP при этом будет указывать на конец PSP, IP станет равным 100h, в стек будет помещено 0000h. Код будет располагаться со смещением в 100h.

2) Что располагается с 0 адреса?

Сегмент PSP.

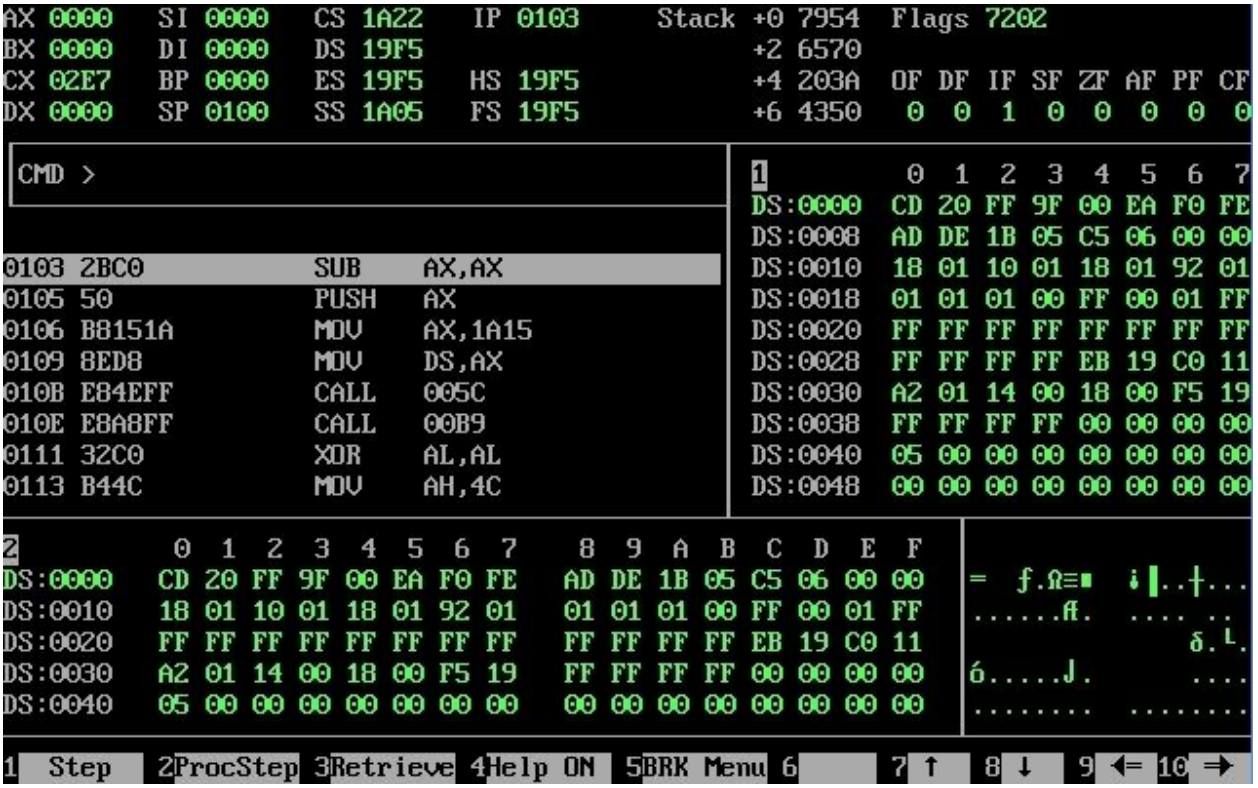
3) Какие значения имеют сегментные регистры? На какие области памяти они указывают?

Сегментные регистры будут иметь одинаковые значения. И будут указывать на начало PSP.

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

Стек определяется автоматически. Регистр SS указывает на начало сегмента, а SP=FFFEh – на его конец.

Загрузка «хорошего» EXE модуля в основную память



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

Сначала считывается информация с заголовка EXE. Затем инициализируются сегментные регистры. SS будет указывать на начало сегмента стека, PS на конец сегмента стека, CS на начало сегмента кода. В IP запишется точка входа в программу.

2) На что указывают регистры DS и ES ?

ES и DS будут указывать на начало PSP.

3) Как определяется стек?

В программе с помощью специальной директивы .STACK с указанием размера.

4) Как определяется точка входа?

Точка входа в программу определяется с помощью директивы END.