

**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**

Студентка гр. 9382

\_\_\_\_\_

Балаева М.О.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2021

## **Цель работы:**

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

## **Постановка задачи:**

Требуется написать текст исходного .COM модуля, который определяет 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 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 (Original Equipment Manufacturer) и серийным номером пользователя. Полученные строки выводятся на экран.

Далее необходимо отладить полученный исходный модуль и получить «хороший» .COM модуль, а также необходимо построить «плохой» .EXE, полученный из исходного текста для .COM модуля.

Затем нужно написать текст «хорошего» .EXE модуля, который выполняет те же функции, что и модуль .COM, далее его построить, отладить и сравнить исходные тексты для .COM и .EXE модулей.

## **Необходимые сведения для составления программы:**

**Тип IBM PC** хранится в байте по адресу 0F000:0FFFE, в предпоследнем байте ROM BIOS. Соответствие кода и типа в таблице:

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

Для определения **версии MS DOS** следует воспользоваться функцией 30H прерывания 21H. Входным параметром является номер функции в AH:

MOV AH,30h

INT 21h

Выходными параметрами являются:

AL – номер основной версии. Если 0, то <2.0;

AH – номер модификации;

BH – серийный номер OEM (Original Equipment Manufacturer);

BL:CX – 24-битовый серийный номер пользователя;

В программе используются следующие процедуры:

Название процедуры	Назначение
TETR_TO_HEX	Перевод половины байта в шестнадцатеричную систему счисления
BYTE_TO_HEX	Перевод байта регистра AL в шестнадцатеричную систему счисления, помещая результат в AX
WRD_TO_HEX	Перевод двух байт регистра AX в шестнадцатеричную систему счисления, помещая результат в регистр DI

## Определение структуры данных:

Название поля данных	Тип	Назначение
PC_T	db	PC
PC_XT_T	db	PC/XT
AT_T	db	AT
PS2_M30_T	db	PS2 модель 30
PS2_M50_60_T	db	PS2 модель 50 или 60
PS2_M80_T	db	PS2 модель 80
PC_JR_T	db	PCjr
PC_CONV_T	db	PC Convertible
VERSION	db	Номер версии MS DOS
SERIAL	db	Серийный номер OEM
USER	db	Серийный номер пользователя

### Ход работы:

Шаг 1. Запуск «хорошего» .COM модуля:

```
C:\>COM.com
My type: AT
My MS-DOS versio5 0
Serial number OEM: 0
User serial number: 000000H
```

Запуск «плохого» .EXE модуля:

```
C:\>COM.EXE

5 0
0
0My type: PC
0My type: PC
0My type: PC
0My ty000000
```

Шаг 2. Запуск «хорошего» .EXE модуля:

```
C:\> exe.exe
My type: AT
Version MS-DOS: 5.0
Serial number OEM: 0
User serial number: 000000H
```

Шаг 3. Ответы на контрольные вопросы. Отличия исходных текстов COM и EXE программ:

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

**Ответ:** COM программа может содержать только один сегмент.

EXE программа?

**Ответ:** EXE программа может содержать больше одного сегмента. В программах этого типа предусматривают отдельные сегменты для кода, данных и стека.

**2.** Какие директивы должны обязательно быть в тексте COM программы?

**Ответ:** в программе должна обязательно быть директива `ORG 100h` (смещение 100h), так как при загрузке COM-файла в память DOS занимает первые 256 байт (100h) блоком данных PSP и располагает код программы только после этого блока. Кроме того, должна присутствовать директива `ASSUME`, ставящая в соответствие начало программы сегментам кода и данных, в противном случае, программа не скомпилируется из-за невозможности обнаружения начала сегмента кода.

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

**Ответ:** нет, не все, так как в отличие от EXE-программы, в которой существует таблица настроек (таблица разметки), называемая Relocation Table, COM-программа ей не располагает. Адреса сегментов определяются загрузчиком в момент запуска программы на основе информации о местоположении полей адресов в файле из Relocation Table. Следовательно, в связи с отсутствием этой таблицы в COM-программах, команды вида `mov [регистр], seg [сегмент]` недопустимы.

#### **Шаг 4.**

.COM модуль в шестнадцатеричном виде

```

00000000 E9 10 02 4D 79 20 74 79 70 65 3A 20 50 43 0D 0A 00..My type: PC..
00000010 24 4D 79 20 74 79 70 65 3A 20 50 43 2F 58 54 0D $My type: PC/XT.
00000020 0A 24 4D 79 20 74 79 70 65 3A 20 41 54 0D 0A 24 .$.My type: AT..$
00000030 4D 59 20 74 79 70 65 3A 20 50 53 32 20 D0 BC D0 MY type: PS2 1111
00000040 BE D0 B4 D0 B5 D0 BB D1 8C 20 33 30 0D 0A 24 4D 11111111i 30..$M
00000050 79 20 74 79 70 65 3A 20 50 53 32 20 D0 BC D0 BE y type: PS2 1111
00000060 D0 B4 D0 B5 D0 BB D1 8C 20 35 30 20 D0 B8 D0 BB 11111111i 50 1111
00000070 D0 B8 20 36 30 0D 0A 24 4D 79 20 74 79 70 65 3A 11 60..$My type:
00000080 20 50 53 32 20 D0 BC D0 BE D0 B4 D0 B5 D0 BB D1 PS2 11111111111111
00000090 8C 20 38 30 0D 0A 24 4D 79 20 74 79 70 65 3A 20 i 80..$My type:
000000A0 50 D0 A1 6A 72 0D 0A 24 4D 79 20 74 79 70 65 3A P11jr..$My type:
000000B0 20 50 43 20 43 6F 6E 76 65 72 74 69 62 6C 65 0D PC Convertible.
000000C0 0A 24 4D 79 20 4D 53 2D 44 4F 53 20 76 65 72 73 .$.My MS-DOS vers
000000D0 69 6F 6E 20 3A 20 20 2E 20 20 0D 0A 24 53 65 72 ion : . ..$Ser
000000E0 69 61 6C 20 6E 75 6D 62 65 72 20 4F 45 4D 3A 20 ial number OEM:
000000F0 20 0D 0A 24 55 73 65 72 20 73 65 72 69 61 6C 20 ..$User serial
00000100 6E 75 6D 62 65 72 3A 20 20 20 20 20 20 20 48 20 number: H
00000110 24 24 0F 3C 09 76 02 04 07 04 30 C3 51 8A E0 E8 $$.<.v....0|Qèαφ
00000120 EF FF 86 C4 B1 04 D2 E8 E8 E6 FF 59 C3 53 8A FC n â-µ.τφφμ Y|Sèñ
00000130 E8 E9 FF 88 25 4F 88 05 4F 8A C7 E8 DE FF 88 25 φθ è%0è.0è|φ | è%
00000140 4F 88 05 5B C3 51 52 32 E4 33 D2 B9 0A 00 F7 F1 0è. [ |QR2Σ3τ|...≈±
00000150 80 CA 30 88 14 4E 33 D2 3D 0A 00 73 F1 3C 00 74 Ç0è.N3τ=...s±<.t
00000160 04 0C 30 88 04 5A 59 C3 B8 00 F0 8E C0 26 A0 FE ..0è.ZY|γ.≡ÄL&á.
00000170 FF 3C FF 74 1C 3C FE 74 1E 3C FB 74 1A 3C FC 74 < t.<.t.<√t.<^nt
00000180 1C 3C FA 74 1E 3C F8 74 26 3C FD 74 28 3C F9 74 .<.t.<°t&<²t(<.t
00000190 2A BA 03 01 EB 2B 90 BA 11 01 EB 25 90 BA 22 01 *||..δ+É||..δ%É||".
000001A0 EB 1F 90 BA 30 01 EB 19 90 BA 4F 01 EB 13 90 BA δ.É||θ.δ.É||0.δ.É||
000001B0 78 01 EB 0D 90 BA 97 01 EB 07 90 BA A8 01 EB 01 x.δ.É||ù.δ.É||¿.δ.
000001C0 90 B4 09 CD 21 C3 B4 30 CD 21 50 BE C2 01 83 C6 É|.≡!| |θ=|P|τ.â|
000001D0 10 E8 71 FF 58 8A C4 83 C6 03 E8 68 FF BA C2 01 .φq Xè-â|.φh ||τ.
000001E0 B4 09 CD 21 BE DD 01 83 C6 13 8A C7 E8 56 FF BA |.≡!| |.â|.è|φV ||
000001F0 DD 01 B4 09 CD 21 BF F4 01 83 C7 19 8B C1 E8 2C |.|.≡!|γ|.â|.î|φ,
00000200 FF 8A C3 E8 16 FF 83 EF 02 89 05 BA F4 01 B4 09 è|φ. ân.ë. |||.|.
00000210 CD 21 C3 E8 52 FF E8 AD FF 32 C0 B4 4C CD 21 + =!|φR φi 2|L=|

```



# «плохой» .EXE модуль в шестнадцатеричном виде

COM.COM x	COM.EXE x	
00000000	4D 5A 1F 01 03 00 00 00	20 00 00 00 FF FF 00 00 MZ..... ..
00000010	00 00 DA 20 00 01 00 00	1E 00 00 00 01 00 00 00 ..Г .....
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 .....
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 .....
00000200	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 .....
00000210	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 .....
00000220	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 .....
00000230	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 .....
00000240	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 .....
00000250	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 .....
00000260	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 .....
00000270	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 .....
00000280	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 .....
00000290	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 .....
000002A0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 .....

000002B0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
000002C0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
000002D0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
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 10 02 4D 79 20 74 79	70 65 3A 20 50 43 0D 0A	ø..My type: PC..
00000310	24 4D 79 20 74 79 70 65	3A 20 50 43 2F 58 54 0D	\$My type: PC/XT.
00000320	0A 24 4D 79 20 74 79 70	65 3A 20 41 54 0D 0A 24	.\$My type: AT..\$
00000330	4D 59 20 74 79 70 65 3A	20 50 53 32 20 D0 BC D0	MY type: PS2 𐀀𐀀𐀀
00000340	BE D0 B4 D0 B5 D0 BB D1	8C 20 33 30 0D 0A 24 4D	𐀀𐀀𐀀𐀀𐀀𐀀𐀀𐀀î 30..\$M
00000350	79 20 74 79 70 65 3A 20	50 53 32 20 D0 BC D0 BE	y type: PS2 𐀀𐀀𐀀
00000360	D0 B4 D0 B5 D0 BB D1 8C	20 35 30 20 D0 B8 D0 BB	𐀀𐀀𐀀𐀀𐀀𐀀𐀀î 50 𐀀𐀀𐀀
00000370	D0 B8 20 36 30 0D 0A 24	4D 79 20 74 79 70 65 3A	𐀀𐀀 60..\$My type:
00000380	20 50 53 32 20 D0 BC D0	BE D0 B4 D0 B5 D0 BB D1	PS2 𐀀𐀀𐀀𐀀𐀀𐀀𐀀𐀀𐀀
00000390	8C 20 38 30 0D 0A 24 4D	79 20 74 79 70 65 3A 20	î 80..\$My type:
000003A0	50 D0 A1 6A 72 0D 0A 24	4D 79 20 74 79 70 65 3A	P𐀀-ijr..\$My type:
000003B0	20 50 43 20 43 6F 6E 76	65 72 74 69 62 6C 65 0D	PC Convertible.
000003C0	0A 24 4D 79 20 4D 53 2D	44 4F 53 20 76 65 72 73	.\$My MS-DOS vers
000003D0	69 6F 6E 20 3A 20 20 2E	20 20 0D 0A 24 53 65 72	ion : . ..\$Ser
000003E0	69 61 6C 20 6E 75 6D 62	65 72 20 4F 45 4D 3A 20	ial number OEM:
000003F0	20 0D 0A 24 55 73 65 72	20 73 65 72 69 61 6C 20	..\$User serial
00000400	6E 75 6D 62 65 72 3A 20	20 20 20 20 20 20 48 20	number: H
00000410	24 24 0F 3C 09 76 02 04	07 04 30 C3 51 8A E0 E8	\$\$.<.v....0 Qèαφ
00000420	EF FF 86 C4 B1 04 D2 E8	E8 E6 FF 59 C3 53 8A FC	∩ â-𐀀𐀀𐀀.𐀀φφμ Y Sèⁿ
00000430	E8 E9 FF 88 25 4F 88 05	4F 8A C7 E8 DE FF 88 25	φφ è%0è.0è φ  è%
00000440	4F 88 05 5B C3 51 52 32	E4 33 D2 B9 0A 00 F7 F1	0è.[ QR2Σ3𐀀𐀀 ..≈±
00000450	80 CA 30 88 14 4E 33 D2	3D 0A 00 73 F1 3C 00 74	Ç𐀀0è.N3𐀀=..s±<.t
00000460	04 0C 30 88 04 5A 59 C3	B8 00 F0 8E C0 26 A0 FE	..0è.ZY 𐀀.=ÃL&á.
00000470	FF 3C FF 74 1C 3C FE 74	1E 3C FB 74 1A 3C FC 74	< t.< .t.<√t.<ⁿt
00000480	1C 3C FA 74 1E 3C F8 74	26 3C FD 74 28 3C F9 74	.< .t.< °t&<²t(< .t
00000490	2A BA 03 01 EB 2B 90 BA	11 01 EB 25 90 BA 22 01	*  ..δ+É  ..δ%É  ".
000004A0	EB 1F 90 BA 30 01 EB 19	90 BA 4F 01 EB 13 90 BA	δ.É  0.δ.É  0.δ.É
000004B0	78 01 EB 0D 90 BA 97 01	EB 07 90 BA A8 01 EB 01	x.δ.É  ù.δ.É  ¿.δ.
000004C0	90 B4 09 CD 21 C3 B4 30	CD 21 50 BE C2 01 83 C6	É .=! 𐀀0=!P𐀀𐀀.â𐀀
000004D0	10 E8 71 FF 58 8A C4 83	C6 03 E8 68 FF BA C2 01	.φq Xè-â𐀀𐀀.φh   𐀀.
000004E0	B4 09 CD 21 BE DD 01 83	C6 13 8A C7 E8 56 FF BA	𐀀 .=! 𐀀.â𐀀.è φV
000004F0	DD 01 B4 09 CD 21 BF F4	01 83 C7 19 8B C1 E8 2C	.𐀀 .=! 𐀀.â𐀀.ĩ𐀀φ,
00000500	FF 8A C3 E8 16 FF 83 EF	02 89 05 BA F4 01 B4 09	è φ. ân.ë.    𐀀.
00000510	CD 21 C3 E8 52 FF E8 AD	FF 32 C0 B4 4C CD 21 +	=! φR φ; 2𐀀L=!



# «хороший» .EXE модуль в шестнадцатеричном виде

00000000	4D 5A 26 01 03 00 01 00	20 00 00 00 FF FF 00 00	MZ&..... ..
00000010	00 01 8A 3D 02 01 21 00	1E 00 00 00 01 00 06 01	..è=...!.....
00000020	21 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	.....
000002B0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
000002C0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
000002D0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
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	4D 79 20 74 79 70 65 3A	20 50 43 0D 0A 24 4D 79	My type: PC..\$My
00000310	20 74 79 70 65 3A 20 50	43 2F 58 54 0D 0A 24 4D	type: PC/XT..\$M
00000320	79 20 74 79 70 65 3A 20	41 54 0D 0A 24 4D 59 20	y type: AT..\$MY
00000330	74 79 70 65 3A 20 50 53	32 20 D0 BC D0 BE D0 B4	type: PS2 ������
00000340	D0 B5 D0 BB D1 8C 20 33	30 0D 0A 24 4D 79 20 74	������î 30..\$My t
00000350	79 70 65 3A 20 50 53 32	20 D0 BC D0 BE D0 B4 D0	ype: PS2 ��������
00000360	B5 D0 BB D1 8C 20 35 30	20 D0 B8 D0 BB D0 B8 20	������î 50 ������
00000370	36 30 0D 0A 24 4D 79 20	74 79 70 65 3A 20 50 53	60..\$My type: PS
00000380	32 20 D0 BC D0 BE D0 B4	D0 B5 D0 BB D1 8C 20 38	2 ������������î 8
00000390	30 0D 0A 24 4D 79 20 74	79 70 65 3A 20 50 D0 A1	0..\$My type: P��i
000003A0	6A 72 0D 0A 24 4D 79 20	74 79 70 65 3A 20 50 43	jr..\$My type: PC
000003B0	20 43 6F 6E 76 65 72 74	69 62 6C 65 0D 0A 24 56	Convertible..\$V
000003C0	65 72 73 69 6F 6E 20 4D	53 2D 44 4F 53 3A 20 20	ersion MS-DOS:
000003D0	2E 20 20 0D 0A 24 53 65	72 69 61 6C 20 6E 75 6D	. ..\$Serial num
000003E0	62 65 72 20 4F 45 4D 3A	20 20 0D 0A 24 55 73 65	ber OEM: ..\$Use
000003F0	72 20 73 65 72 69 61 6C	20 6E 75 6D 62 65 72 3A	r serial number:
00000400	20 20 20 20 20 20 20 48	20 24 00 00 00 00 00 00	H \$.....
00000410	24 0F 3C 09 76 02 04 07	04 30 C3 51 8A E0 E8 EF	\$.<.v....0��������
00000420	FF 86 C4 B1 04 D2 E8 E8	E6 FF 59 C3 53 8A FC E8	��������������������
00000430	E9 FF 88 25 4F 88 05 4F	8A C7 E8 DE FF 88 25 4F	��������������������
00000440	88 05 5B C3 51 52 32 E4	33 D2 B9 0A 00 F7 F1 80	��������������������
00000450	CA 30 88 14 4E 33 D2 3D	0A 00 73 F1 3C 00 74 04	��������������������
00000460	0C 30 88 04 5A 59 C3 B8	00 F0 8E C0 26 A0 FE FF	.��������������������
00000470	3C FF 74 1C 3C FE 74 1E	3C FB 74 1A 3C FC 74 1C	< t.<.t.<��t.<��t.
00000480	3C FA 74 1E 3C F8 74 26	3C FD 74 28 3C F9 74 2A	<��t.<��t.<��t(<��t*
00000490	BA 00 00 EB 2B 90 BA 0E	00 EB 25 90 BA 1F 00 EB	��������������������
000004A0	1F 90 BA 2D 00 EB 19 90	BA 4C 00 EB 13 90 BA 75	.��������������������
000004B0	00 EB 0D 90 BA 94 00 EB	07 90 BA A5 00 EB 01 90	.��������������������
000004C0	B4 09 CD 21 C3 B4 30 CD	21 50 BE BF 00 83 C6 10	��������������������
000004D0	E8 71 FF 58 8A C4 83 C6	03 E8 68 FF BA BF 00 B4	��������������������
000004E0	09 CD 21 BE D6 00 83 C6	13 8A C7 E8 56 FF BA D6	.=!������������������
000004F0	00 B4 09 CD 21 BF ED 00	83 C7 19 8B C1 E8 2C FF	.��������������������
00000500	8A C3 E8 16 FF 83 EF 02	89 05 BA ED 00 B4 09 CD	��������������������
00000510	21 C3 2B C0 50 B8 10 00	8E D8 E8 4A FF E8 A5 FF	!��������������������
00000520	32 C0 B4 4C CD 21	+	2��������������������

Ответы на контрольные вопросы. Отличия форматов файлов COM и EXE программ:

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

**Ответ:** COM файл состоит из одного сегмента и содержит данные и машинные команды. Код начинается с адреса 0h, но при загрузке модуля устанавливается смещение в 100h.

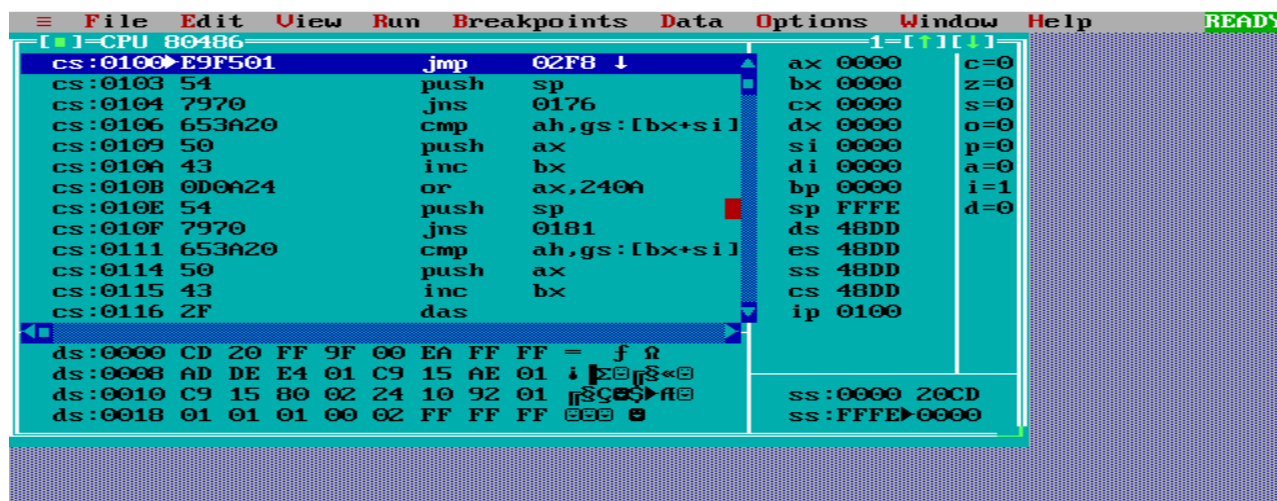
2. Какова структура файла «плохого» EXE? С какого адреса располагается код? Что располагается с 0 адреса?

**Ответ:** в «плохом» EXE файле данные и код содержатся в одном сегменте. Код располагается с адреса 300h. С адреса 0h располагается Relocation Table (таблица разметки).

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

**Ответ:** в «хорошем» файле EXE содержится информация для загрузчика, сегмент стека, сегмент данных и сегмент кода (3 сегмента вместо одного в «плохом» .EXE). Код располагается с адреса 200h в отличие от 300h в «плохом» .EXE файле. В EXE-файле присутствует специальный начальный блок (заголовок) размером не менее 200h (512 байт), кроме того в «плохом» .EXE есть смещение в 100h (256 байт), полученное после преобразования COM-файла в EXE-файл, откуда в сумме получается 300h.

**Шаг 5.** Загрузка COM модуля в основную память.



```
File Edit View Run Breakpoints Data Options Window Help
[1]-CPU 80486
cs:0100 E9F501 jmp 02F8 ↓
cs:0103 54 push sp
cs:0104 7970 jns 0176
cs:0106 653A20 cmp ah,gs:[bx+si]
cs:0109 50 push ax
cs:010A 43 inc bx
cs:010B 0D0A24 or ax,240A
cs:010E 54 push sp
cs:010F 7970 jns 0181
cs:0111 653A20 cmp ah,gs:[bx+si]
cs:0114 50 push ax
cs:0115 43 inc bx
cs:0116 2F das
ax 0000 c=0
bx 0000 z=0
cx 0000 s=0
dx 0000 o=0
si 0000 p=0
di 0000 a=0
bp 0000 i=1
sp FFFE d=0
ds 48DD
es 48DD
ss 48DD
cs 48DD
ip 0100
ds:0000 CD 20 FF 9F 00 EA FF FF = f 0
ds:0008 AD DE E4 01 C9 15 AE 01 i 0 0 0 0
ds:0010 C9 15 80 02 24 10 92 01 0 0 0 0
ds:0018 01 01 01 00 02 FF FF FF 0 0 0 0
ss:0000 20CD
ss:FFFE 0000
READY
```

Ответы на контрольные вопросы. Загрузка COM модуля в основную память:

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

**Ответ:** после загрузки COM-программы в память сегментные регистры указывают на начало PSP. Код располагается с адреса 100h (ip = 0100h).

2. Что располагается с 0 адреса?

**Ответ:** адрес начала PSP.

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

**Ответ:** 48DDh. Они указывают на начало PSP.

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

**Ответ:** стек определяется автоматически, указатель стека устанавливается на конец сегмента. Если для программы размер сегмента в 64КБ является достаточным, то DOS устанавливает в регистре SP адрес конца сегмента – FFFEH. Если 64К байтовый сегмент не имеет достаточно места для стека, то DOS устанавливает стек в конце памяти, выделяемой загрузчиком для исполнения программы, причём возможна ситуация, когда стек накладывается на данные или код. Адреса расположены в диапазоне 0000h-FFFFh.

**Шаг 6.** Загрузка «хорошего» EXE модуля в память.

The screenshot shows the DOS DEBUG program interface. The menu bar at the top includes File, Edit, View, Run, Breakpoints, Data, Options, Window, and Help. The status bar at the bottom shows function key shortcuts: F1-Help, F2-Bkpt, F3-Mod, F4-Here, F5-Zoom, F6-Next, F7-Trace, F8-Step, F9-Run, and F10-Menu. The main window is divided into several sections. The top section shows the CPU register values: CPU 80486, 1=[↑][↓]. The middle section displays assembly code with addresses, instructions, and operands. The right section shows the current values of the registers. The bottom section shows the memory dump.

Address	Instruction	Operand	Register Value
cs:0103	sub	ax,ax	ax 0000
cs:0105	push	ax	bx 0000
cs:0106	mov	ax,4BFD	cx 0000
cs:0109	mov	ds,ax	dx 0000
cs:010B	call	005C	si 0000
cs:010E	call	00B9	di 0000
cs:0111	xor	al,al	bp 0000
cs:0113	mov	ah,4C	sp 0100
cs:0115	int	21	ds 48DD
cs:0117	add	[bx+si],al	es 48DD
cs:0119	add	[bx+si],al	ss 48ED
cs:011B	add	[bx+si],al	cs 490D
cs:011D	add	[bx+si],al	ip 0103

Memory dump (ds:0000):

Address	Hex Data
ds:0000	CD 20 FF 9F 00 EA FF FF
ds:0008	AD DE E4 01 C9 15 AE 01
ds:0010	C9 15 80 02 24 10 92 01
ds:0018	01 01 01 00 02 FF FF FF

Register values (ss:0102):

Register	Value
ss:0102	6570
ss:0100	7954

Ответы на контрольные вопросы. Загрузка «хорошего» EXE модуля в память:

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

**Ответ:** в области памяти строится PSP, стандартная часть заголовка считывается в память, определяется длина тела загрузочного модуля, определяется начальный сегмент, загрузочный модуль считывается в начальный сегмент, таблица настройки считывается в рабочую память, определяются значения сегментных регистров. DS и ES устанавливаются на начало PSP, SS - на начало стека, CS - на начало сегмента кода.

2. На что указывают регистры DS и ES?

**Ответ:** в момент загрузки регистры DS и ES указывают на начало PSP. После выполнения команд `mov ax, @data` и `mov ds, ax` регистре DS содержит адрес начала сегмента данных.

3. Как определяется стек?

**Ответ:** в исходном коде модуля стек определяется при помощи директивы `STACK`, а при исполнении в регистры SS и SP записываются адрес начала сегмента стека и его вершины соответственно.

4. Как определяется точка входа?

**Ответ:** команда точки входа определяется при помощи команды `END`.

### **Заключение:**

В ходе работы было проведено исследование различий в структурах исходных текстов модулей .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.