

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ
по практической работе № 1
по дисциплине «Операционные системы»
Тема: Исследование структур загрузочных модулей

Студент гр. 9383

Чумак М.А.

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.

Постановка задачи.

Требуется написать текст исходного .COM модуля, который определяет тип РС и версию системы. Ассемблерная программа должна читать содержимое предпоследнего байта ROM BIOS, по таблице, сравнивая коды, определять тип РС 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.

Таблица 1 – функции в программе

Процедура	Описание
TETR_TO_HEX	Перевод десятичной цифры в код символа

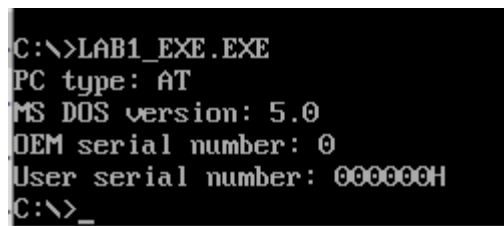
BYTE_TO_HEX	Перевод байта в 16-ной с/с в символьный код
WRD_TO_HEX	Перевод слова в 16-ной с/с в символьный код
BYTE_TO_DEC	Перевод байта в 16-ной с/с в символьный код в 10-ной с/с
PRINT_STRING	Вывод строки на экран
PC_TYPE	Определение типа PC
OS_VERSION	Определение характеристик OS

Выполнение работы.

Были объявлены следующие строки для вывода информации:

1. PC_TYPE_PC db 'PC type: PC', 0dh, 0ah, '\$'
2. PC_TYPE_PC_XT db 'PC type: PC/XT', 0dh, 0ah, '\$'
3. PC_TYPE_AT db 'PC type: AT', 0dh, 0ah, '\$'
4. PC_TYPE_PS2_30 db 'PC type: PS2 30', 0dh, 0ah, '\$'
5. PC_TYPE_PS2_50_60 db 'PC type: PS2 50 or 60', 0dh, 0ah, '\$'
6. PC_TYPE_PS2_80 db 'PC type: PS2 80', 0dh, 0ah, '\$'
7. PC_TYPE_PC_JR db 'PC type: PCjr', 0dh, 0ah, '\$'
8. PC_TYPE_PC_CONVERTIBLE db 'PC type: PC Convertible', 0dh, 0ah, '\$'
9. DOS_VERSION db 'MS DOS version: . ', 0dh, 0ah, '\$'
10. OEM_NUMBER db 'OEM serial number: ', 0dh, 0ah, '\$'
11. USER_NUMBER db 'User serial number: H \$'

Была объявлена функция, описанная выше, для определения типа ПК (PC_TYPE) в соответствии с таблицей:



```
C:\>LAB1_EXE.EXE
PC type: AT
MS DOS version: 5.0
OEM serial number: 0
User serial number: 000000H
C:\>_
```

Рисунок 3 — “хороший” .EXE модуль

Ответы на вопросы.

Отличия исходных текстов COM и EXE программ:

1. Сколько сегментов должна содержать COM-программа?

Один сегмент. Данные вместе с кодом находятся в одном сегменте, стек же генерируется автоматически.

2. EXE-программа?

Не менее одного сегмента. Сегменты кода, данных и стека описываются отдельно друг от друга. Есть возможность не описывать сегмент стека, в это случае будет использоваться стек DOS.

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

Должна быть обязательна директива `ORG 100h`, потому что при загрузке модуля все сегментные регистры содержат адрес `PSP`, который является 256-байтовым блоком, поэтому адресация имеет смещение в 256 байт от нулевого адреса. Также необходима процедура `ASSUME`, для того чтобы сегмент данных и сегмент кода указывали на один общий сегмент.

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

Нет, не все. Нельзя использовать команды вида `mov <регистр>, seg <имя сегмента>`, потому что в программе `.com` отсутствует таблица настроек, содержащая описание адресов, которые зависят от размещения загрузочного модуля в ОП.

Отличия форматов файлов .COM и .EXE программ:

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

COM-файл состоит из одного сегмента, состоящего из сегмента кода и сегмента данных. Сегмент стека генерируется автоматически при создании COM-программы. COM-файл ограничен размером одного сегмента и не превышает 64 Кб. Код начинается с адреса 0h. При загрузке модуля устанавливается смещение в 100h.

view LAB1_COM.COM - Far 3.0.5700.0 x86 Administrator

C:\labs\OS_labs\lab1\LAB1_COM.COM																							
00000000:	E9	E1	01	50	43	20	74	79	70	65	3A	20	50	43	2F	58	54	0D	0A	24	50	53	32
00000001:	24	50	43	20	74	79	70	65	3A	20	50	43	2F	58	54	0D	0A	24	50	53	32	20	33
00000002:	0A	24	50	43	20	74	79	70	65	3A	20	50	43	2F	58	54	0D	0A	24	50	53	32	20
00000003:	50	43	20	74	79	70	65	3A	20	50	43	2F	58	54	0D	0A	24	50	53	32	20	33	30
00000004:	0A	24	50	43	20	74	79	70	65	3A	20	50	43	2F	58	54	0D	0A	24	50	53	32	20
00000005:	30	20	6F	72	20	36	30	0D	0A	24	50	43	20	74	79	70	65	3A	20	50	43	2F	58
00000006:	65	3A	20	50	53	32	20	38	30	0D	0A	24	50	43	20	74	79	70	65	3A	20	50	43
00000007:	79	70	65	3A	20	50	D0	A1	6A	72	0D	0A	24	50	43	20	74	79	70	65	3A	20	50
00000008:	74	79	70	65	3A	20	50	43	20	43	6F	6E	76	65	72	74	53	20	44	4F	53	20	76
00000009:	69	62	6C	65	0D	0A	24	4D	2E	20	0D	0A	24	4F	45	4D	6E	75	6D	62	65	72	3A
0000000A:	72	73	69	6F	6E	3A	20	20	6E	75	6D	62	65	72	3A	20	20	73	65	72	69	61	6C
0000000B:	20	73	65	72	69	61	6C	20	20	73	65	72	69	61	6C	20	20	20	20	20	20	20	48
0000000C:	20	0D	0A	24	55	73	65	72	07	04	30	C3	51	8A	E0	E8	E8	E6	FF	59	C3	53	8A
0000000D:	6E	75	6D	62	65	72	3A	20	E8	E6	FF	59	C3	53	8A	FC	4F	8A	C7	E8	DE	FF	88
0000000E:	24	24	0F	3C	09	76	02	04	E4	33	D2	B9	0A	00	F7	F1	3D	0A	00	73	F1	3C	00
0000000F:	EF	FF	86	C4	B1	04	D2	E8	B4	09	CD	21	C3	B8	00	F0	74	09	CD	21	C3	B8	00
00000010:	E8	E9	FF	88	25	4F	88	05	74	1C	3C	FE	74	1E	3C	FB	4F	8A	C7	E8	DE	FF	88
00000011:	4F	88	05	5B	C3	51	52	32	74	1E	3C	F8	74	26	3C	FD	4F	8A	C7	E8	DE	FF	88
00000012:	80	CA	30	88	14	4E	33	D2	01	EB	2B	90	BA	11	01	EB	3D	0A	00	73	F1	3C	00
00000013:	04	0C	30	88	04	5A	59	C3	01	EB	2B	90	BA	11	01	EB	74	09	CD	21	C3	B8	00
00000014:	8E	C0	26	A0	FE	FF	3C	FF	74	1C	3C	FE	74	1E	3C	FB	00	09	CD	21	C3	B8	00
00000015:	74	1A	3C	FC	74	1C	3C	FA	74	1E	3C	F8	74	26	3C	FD	00	09	CD	21	C3	B8	00
00000016:	74	28	3C	F9	74	2A	BA	03	01	EB	2B	90	BA	11	01	EB	74	09	CD	21	C3	B8	00
00000017:	25	90	BA	22	01	EB	1F	90	01	EB	2B	90	BA	11	01	EB	BA	30	01	EB	19	90	BA
00000018:	01	EB	13	90	BA	5A	01	EB	0D	90	BA	6C	01	EB	07	90	0D	90	BA	6C	01	EB	07
00000019:	BA	7D	01	EB	01	90	E8	9F	FF	C3	B4	30	CD	21	50	BE	0D	90	BA	6C	01	EB	07
0000001A:	97	01	83	C6	10	E8	6D	FF	58	8A	C4	83	C6	03	E8	64	FF	C3	B4	30	CD	21	50
0000001B:	FF	BA	97	01	E8	81	FF	BE	AD	01	83	C6	13	8A	C7	E8	58	8A	C4	83	C6	03	E8
0000001C:	53	FF	BA	AD	01	E8	70	FF	BF	C4	01	83	C7	19	8B	C1	AD	01	83	C6	13	8A	C7
0000001D:	E8	2A	FF	8A	C3	E8	14	FF	83	EF	02	89	05	BA	C4	01	BF	C4	01	83	C7	19	8B
0000001E:	E8	55	FF	C3	E8	56	FF	E8	B0	FF	32	C0	B4	4C	CD	21	83	EF	02	89	05	BA	C4

Рисунок 4 — Структура файла “хорошего” .COM модуля

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

В “плохом” файле EXE в нашем случае данные и код располагаются в одном сегменте, что для EXE файла некорректно: код и данные должны быть разделены на отдельные сегменты. Код располагается с адреса 300h, а с адреса 0h идёт таблица настроек.

C:\labs\OS_labs\lab1\LAB1_COM.EXE

0000000000: 4D 5A F0 00 03 00 00 00	20 00 00 00 FF FF 00 00	MZđ ♥	ÿÿ
0000000010: 00 00 D0 C8 00 01 00 00	1E 00 00 00 01 00 00 00	ĐỀ ☹ ▲ ☹	
0000000020: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000030: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000040: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000050: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000060: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000070: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000080: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000090: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000A0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000B0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000C0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000D0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000E0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000F0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000100: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000110: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000120: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000130: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000140: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000150: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000160: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000170: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000180: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000190: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001A0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001B0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001C0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001D0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001E0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001F0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000200: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000210: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000220: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000230: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000240: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000250: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000260: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000270: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000280: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000290: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002A0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002B0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002C0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002D0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002E0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002F0: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		

Рисунок 5 — Структура файла “плохого” .EXE модуля (1)

0000000300:	E9 E1 01 50 43 20 74 79	70 65 3A 20 50 43 0D 0A	éá@PC type: PC
0000000310:	24 50 43 20 74 79 70 65	3A 20 50 43 2F 58 54 0D	\$PC type: PC/XT
0000000320:	0A 24 50 43 20 74 79 70	65 3A 20 41 54 0D 0A 24	\$PC type: AT
0000000330:	50 43 20 74 79 70 65 3A	20 50 53 32 20 33 30 0D	PC type: PS2 30
0000000340:	0A 24 50 43 20 74 79 70	65 3A 20 50 53 32 20 35	\$PC type: PS2 5
0000000350:	30 20 6F 72 20 36 30 0D	0A 24 50 43 20 74 79 70	0 or 60 \$PC typ
0000000360:	65 3A 20 50 53 32 20 38	30 0D 0A 24 50 43 20 74	e: PS2 80 \$PC t
0000000370:	79 70 65 3A 20 50 D0 A1	6A 72 0D 0A 24 50 43 20	ype: P0ijr \$PC
0000000380:	74 79 70 65 3A 20 50 43	20 43 6F 6E 76 65 72 74	type: PC Convert
0000000390:	69 62 6C 65 0D 0A 24 4D	53 20 44 4F 53 20 76 65	ible \$MS DOS ve
00000003A0:	72 73 69 6F 6E 3A 20 20	2E 20 0D 0A 24 4F 45 4D	rsion: . \$OEM
00000003B0:	20 73 65 72 69 61 6C 20	6E 75 6D 62 65 72 3A 20	serial number:
00000003C0:	20 0D 0A 24 55 73 65 72	20 73 65 72 69 61 6C 20	\$User serial
00000003D0:	6E 75 6D 62 65 72 3A 20	20 20 20 20 20 20 48 20	number: H
00000003E0:	24 24 0F 3C 09 76 02 04	07 04 30 C3 51 8A E0 E8	\$ \$ov ♦ ♦ ÅQŠ àè
00000003F0:	EF FF 86 C4 B1 04 D2 E8	E8 E6 FF 59 C3 53 8A FC	ÿ†Ä± ðèæÿYÄSŭ
0000000400:	E8 E9 FF 88 25 4F 88 05	4F 8A C7 E8 DE FF 88 25	èéÿ`%0`†0ŠÇèÿ`%
0000000410:	4F 88 05 5B C3 51 52 32	E4 33 D2 B9 0A 00 F7 F1	O`+ [ÅQR2ä30¹ ÷ ñ
0000000420:	80 CA 30 88 14 4E 33 D2	3D 0A 00 73 F1 3C 00 74	€Ê0`JN30= sñ< t
0000000430:	04 0C 30 88 04 5A 59 C3	B4 09 CD 21 C3 B8 00 F0	♦Q0`♦ZYÄ`oÍ!Ä, ð
0000000440:	8E C0 26 A0 FE FF 3C FF	74 1C 3C FE 74 1E 3C FB	ŽÀ& þÿ<ÿtL<þt▲<û
0000000450:	74 1A 3C FC 74 1C 3C FA	74 1E 3C F8 74 26 3C FD	t-><ütL<út▲<øt&<ý
0000000460:	74 28 3C F9 74 2A BA 03	01 EB 2B 90 BA 11 01 EB	t(<ùt*°♥0ë+0°◀0ë
0000000470:	25 90 BA 22 01 EB 1F 90	BA 30 01 EB 19 90 BA 42	%0°"0ë▼0°00ë↓0°B
0000000480:	01 EB 13 90 BA 5A 01 EB	0D 90 BA 6C 01 EB 07 90	0ë!!0°Z0ëJ0°l0ë•0
0000000490:	BA 7D 01 EB 01 90 E8 9F	FF C3 B4 30 CD 21 50 BE	°J0ë00èÿÿÄ`0Í!P%
00000004A0:	97 01 83 C6 10 E8 6D FF	58 8A C4 83 C6 03 E8 64	-0fÆ-èmyXŠÄfÆ♥èd
00000004B0:	FF BA 97 01 E8 81 FF BE	AD 01 83 C6 13 8A C7 E8	ÿ°-0èÿÿ%-0fÆ!!ŠÇè
00000004C0:	53 FF BA AD 01 E8 70 FF	BF C4 01 83 C7 19 8B C1	Sÿ°-0èÿÿ;Ä0fÇ↓<Á
00000004D0:	E8 2A FF 8A C3 E8 14 FF	83 EF 02 89 05 BA C4 01	è*ÿŠÄèJÿfi0%†°Ä0
00000004E0:	E8 55 FF C3 E8 56 FF E8	B0 FF 32 C0 B4 4C CD 21	èUÿÄèVÿè°ÿ2Ä`LÍ!

Рисунок 6 — Структура файла “плохого” .EXE модуля (2)

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

В «хорошем» EXE код, данные и стек разделены на сегменты. EXE файл может иметь любой размер. У «хорошего» EXE код, данные и стек находятся в разных сегментах, а в «плохом» - в одном сегменте.

C:\labs\OS_labs\lab1\LAB1_EXE.EXE

0000000000:	4D 5A F4 00 03 00 01 00	20 00 00 00 FF FF 00 00	MZô ♥ @	ÿÿ
0000000010:	00 01 47 DB 03 01 1E 00	1E 00 00 00 01 00 04 01	@GŪ♥@▲ ▲	@ ♦@
0000000020:	1E 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	▲	
0000000030:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000040:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000050:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000060:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000070:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000080:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000090:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000A0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000B0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000C0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000D0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000E0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000000F0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000100:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000110:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000120:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000130:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000140:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000150:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000160:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000170:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000180:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000190:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001A0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001B0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001C0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001D0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001E0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000001F0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000200:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000210:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000220:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000230:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000240:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000250:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000260:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000270:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000280:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
0000000290:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002A0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002B0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002C0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002D0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002E0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		
00000002F0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		

Рисунок 7 — Структура файла “хорошего” .EXE модуля (1)

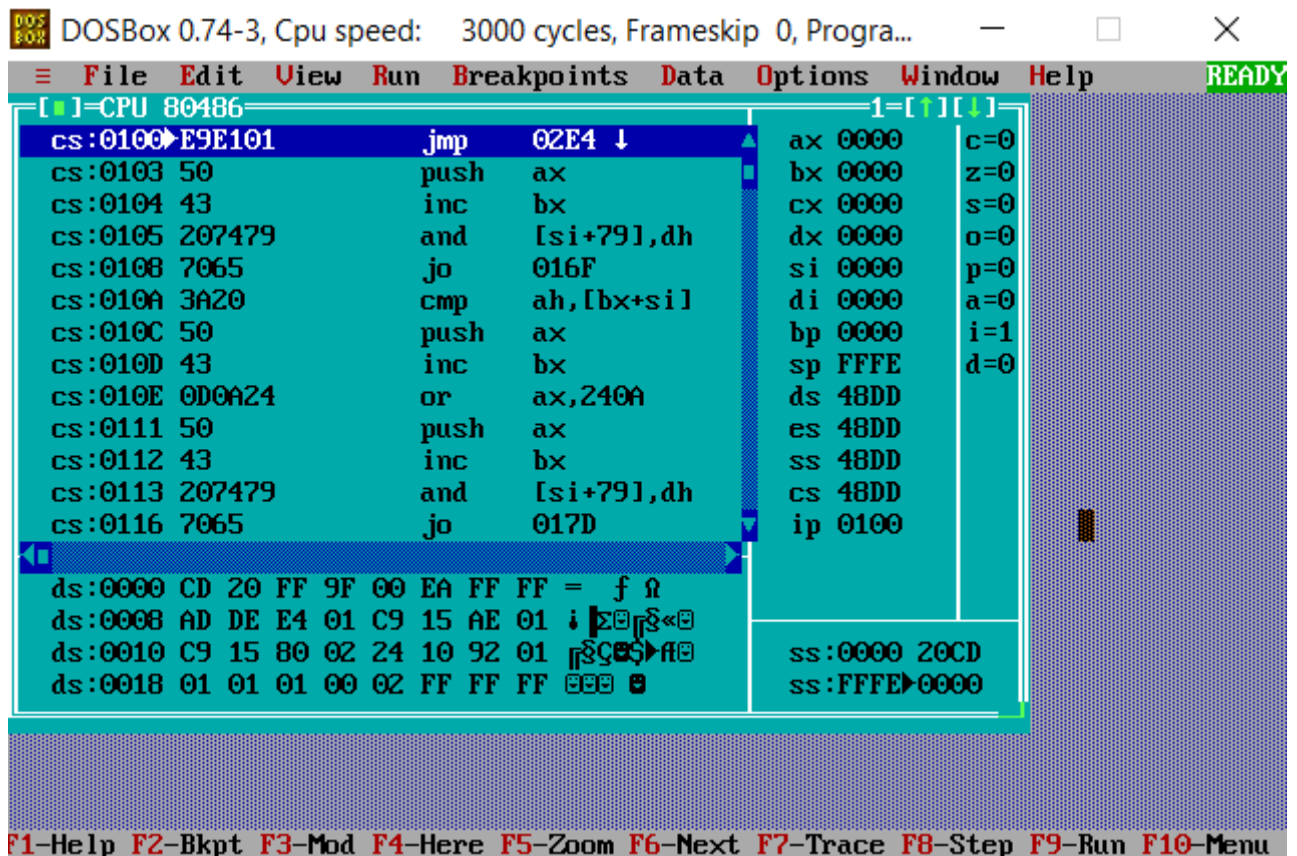
000000300:	50 43 20 74 79 70 65 3A	20 50 43 0D 0A 24 50 43	PC type: PC
000000310:	20 74 79 70 65 3A 20 50	43 2F 58 54 0D 0A 24 50	type: PC/XT
000000320:	43 20 74 79 70 65 3A 20	41 54 0D 0A 24 50 43 20	C type: AT
000000330:	74 79 70 65 3A 20 50 53	32 20 33 30 0D 0A 24 50	type: PS2 30
000000340:	43 20 74 79 70 65 3A 20	50 53 32 20 35 30 20 6F	C type: PS2 50 o
000000350:	72 20 36 30 0D 0A 24 50	43 20 74 79 70 65 3A 20	r 60
000000360:	50 53 32 20 38 30 0D 0A	24 50 43 20 74 79 70 65	PS2 80
000000370:	3A 20 50 D0 A1 6A 72 0D	0A 24 50 43 20 74 79 70	: PDijr
000000380:	65 3A 20 50 43 20 43 6F	6E 76 65 72 74 69 62 6C	e: PC Convertibl
000000390:	65 0D 0A 24 4D 53 20 44	4F 53 20 76 65 72 73 69	e
0000003A0:	6F 6E 3A 20 20 2E 20 0D	0A 24 4F 45 4D 20 73 65	on: .
0000003B0:	72 69 61 6C 20 6E 75 6D	62 65 72 3A 20 20 0D 0A	rial number:
0000003C0:	24 55 73 65 72 20 73 65	72 69 61 6C 20 6E 75 6D	\$User serial num
0000003D0:	62 65 72 3A 20 20 20 20	20 20 20 48 20 24 00 00	ber: H \$
0000003E0:	24 0F 3C 09 76 02 04 07	04 30 C3 51 8A E0 E8 EF	\$o<ov0♦♦♦0ÃQŠaèi
0000003F0:	FF 86 C4 B1 04 D2 E8 E8	E6 FF 59 C3 53 8A FC E8	ÿtÃ±♦0èèæyYÄSŠüè
000000400:	E9 FF 88 25 4F 88 05 4F	8A C7 E8 DE FF 88 25 4F	éÿ~%0~*0ŠÇèbÿ~%0
000000410:	88 05 5B C3 51 52 32 E4	33 D2 B9 0A 00 F7 F1 80	~*+[ÃQR2ä3D¹ ÷ñ€
000000420:	CA 30 88 14 4E 33 D2 3D	0A 00 73 F1 3C 00 74 04	Ê0~!N3D= sñ< t♦
000000430:	0C 30 88 04 5A 59 C3 B4	09 CD 21 C3 B8 00 F0 8E	90~♦ZYÃ´oÍ!Ã, ðŽ
000000440:	C0 26 A0 FE FF 3C FF 74	1C 3C FE 74 1E 3C FB 74	À& bÿ<ÿtL<pt▲<ût
000000450:	1A 3C FC 74 1C 3C FA 74	1E 3C F8 74 26 3C FD 74	-><ûtL<út▲<øt&<ÿt
000000460:	28 3C F9 74 2A BA 00 00	EB 2B 90 BA 0E 00 EB 25	(<ût*º è+º è%
000000470:	90 BA 1F 00 EB 1F 90 BA	2D 00 EB 19 90 BA 3F 00	ºº▼ è▼º- è↓º?
000000480:	EB 13 90 BA 57 00 EB 0D	90 BA 69 00 EB 07 90 BA	è!!ºW è!ºi è•º
000000490:	7A 00 EB 01 90 E8 9F FF	C3 B4 30 CD 21 50 BE 94	z èºèÿÿÃ´OÍ!P%”
0000004A0:	00 83 C6 10 E8 6D FF 58	8A C4 83 C6 03 E8 64 FF	fAèemÿXŠÄfA▼èdÿ
0000004B0:	BA 94 00 E8 81 FF BE AA	00 83 C6 13 8A C7 E8 53	º” èºÿ%ª fA!!ŠÇèS
0000004C0:	FF BA AA 00 E8 70 FF BF	C1 00 83 C7 19 8B C1 E8	ÿºª èpÿ¿Á fC↓<Áè
0000004D0:	2A FF 8A C3 E8 14 FF 83	EF 02 89 05 BA C1 00 E8	*ÿŠÃè!ÿÿfi0%*ºÁ è
0000004E0:	55 FF C3 B8 10 00 8E D8	E8 51 FF E8 AB FF 32 C0	UÿÃ, ŽøèQÿè«ÿ2Ã
0000004F0:	B4 4C CD 21		´LÍ!

Рисунок 8 — Структура файла “хорошего” .EXE модуля (2)

Загрузка COM модуля в основную память:

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

Сначала операционная система ищет подходящее по размеру место в оперативной памяти для COM модуля, после чего ОС помещает в это место PSP и по смещению в 100h помещает модуль.



2. Что располагается с адреса 0?

Программный сегмент PSP, размером 256 байт (100h).

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

Сегментные регистры CS, DS, ES и SS указывают на PSP и имеют значения 48DD.

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

Стек генерируется автоматически при создании COM-программы. SS – на начало (0h), регистр SP указывает на конец стека (FFFEh). Адреса стека расположены в диапазоне 0h – FFEh.

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.

2. На что указывают регистры DS и ES?

Регистры DS и ES указывают на начало сегмента PSP.

3. Как определяется стек?

Стек определяется с помощью директивы `.stack`, после которой задаётся размер стека. При исполнении регистр SS указывает на начало сегмента стека, а SP на конец стека.

4. Как определяется точка входа?

Точка входа определяется при помощи директивы `END`.

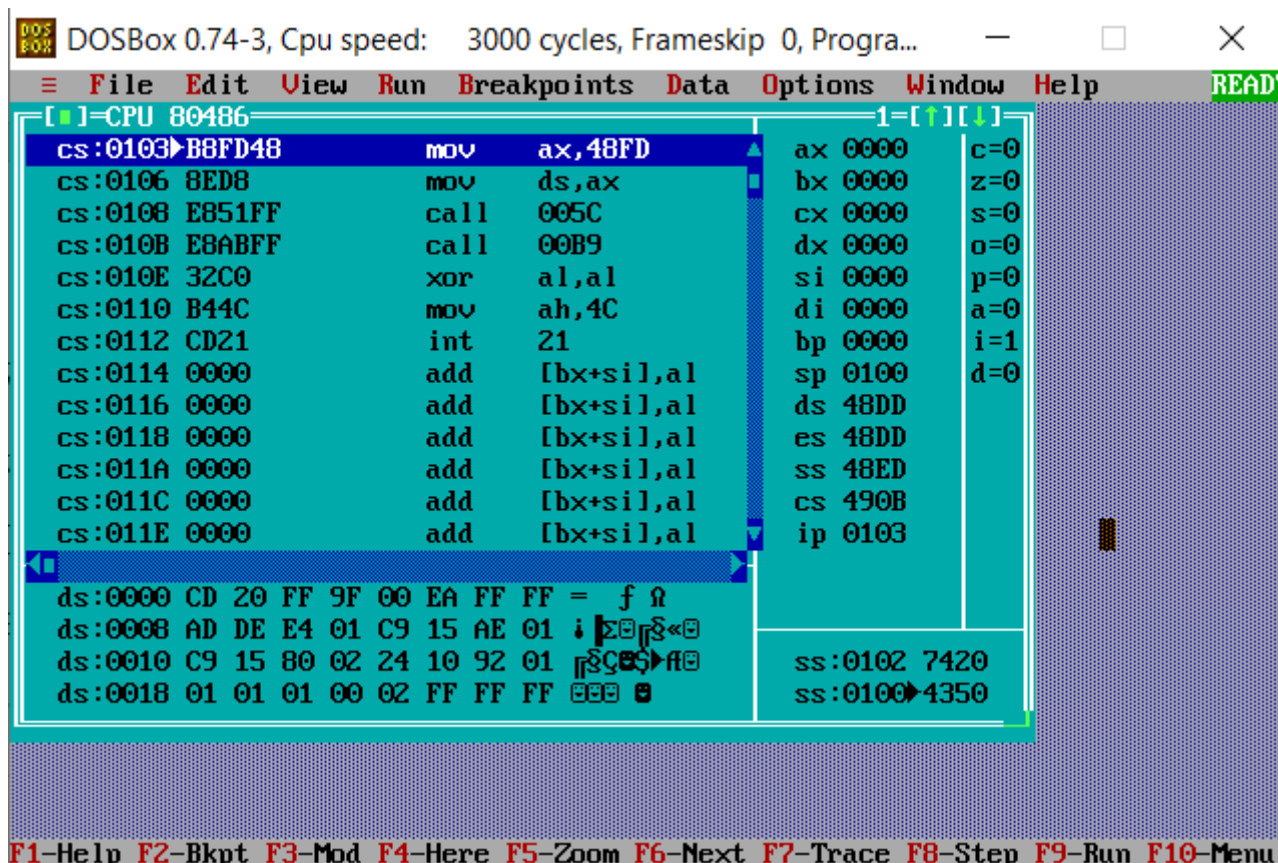


Рисунок 10 — EXE модуль в отладчике TD.EXE

Выводы.

В ходе лабораторной работы были исследованы различия в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.