

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. 7383

Тян Е.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2019

Постановка задачи.

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

В данной работ были использованы процедуры:

- Write_msg – выводит сообщение на экран
- TERT_TO_HEX – переводит из двоичной в шестнадцатеричную систему счисления
- BYTE_TO_HEX – переводит байтовое число в шестнадцатеричную систему счисления
- WRD_TO_HEX – переводит шестнадцатибитовое число в шестнадцатеричную систему счисления
- BYTE_TO_DEC – переводит байтовое число в десятичную систему счисления
- SET_PC_TYPE – устанавливает тип PC, сравнивая коды, если код не совпал, то код выводится в виде шестнадцатеричного числа, и выводит строку с названием модели
- SET_STM_VER – устанавливает версию системы
- SET_OEM_NUM – устанавливает серийный номер OEM
- SET_SRL_NUM – устанавливает серийный номер пользователя.

В данной программе использовались следующие структуры данных:

- PC_TYPE – строка, содержащая информацию для аккуратного вывода типа PC
- STM_VER – строка, содержащая версию системы
- OEM_NUM – строка, содержащая серийный номер OEM
- SRL_NUM – строка, содержащая серийный номер пользователя
- TYPE_PS2_80 – строка, содержащая тип PC, соответствующий коду F8 в шестнадцатеричной системе
- TYPE_PC_Con – строка, содержащая тип PC, соответствующий коду F9 в шестнадцатеричной системе
- TYPE_PS2_30 – строка, содержащая тип PC, соответствующий коду FA в шестнадцатеричной системе

- TYPE_PC_XT – строка, содержащая тип PC, соответствующий коду FB или FE в шестнадцатеричной системе
- TYPE_AT – строка, содержащая тип PC, соответствующий коду FC в шестнадцатеричной системе
- TYPE_PCjr – строка, содержащая тип PC, соответствующий коду FD в шестнадцатеричной системе
- TYPE_PC – строка, содержащая тип PC, соответствующий коду FF в шестнадцатеричной системе

Программа определяет код, соответствующий типу PC, и выводит строку, где указывается тип PC или шестнадцатеричное число кода. Затем программа записывает версию системы, серийный номер OEM и серийный номер пользователя в соответствующие строки. В конце программа выводит строки на экран.

Ход работы.

1. Был написан текст исходного .COM модуля, определяющего тип PC, версию системы, серийный номер OEM и серийный номер пользователя.
2. Получены «хороший» .COM модуль и «плохой» .EXE, полученный из исходного текста для .COM модуля.



Рисунок 1 — Результат работы программы «плохого» .EXE модуля

3. Был написан текст исходного .EXE модуля, выполняющего те же функции.
4. Был получен «хороший» .EXE.
5. На рис. 1 приведены результаты работы программы.

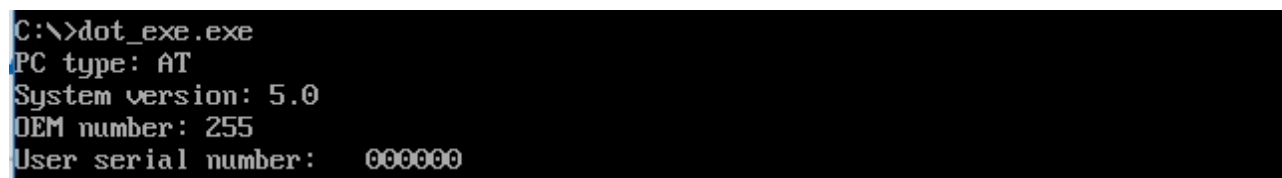


Рисунок 2 — Результат работы программы «хорошего» .EXE модуля

Ответы на контрольные вопросы.

Отличия исходных текстов COM и EXE программ

- 1) Сколько сегментов должна содержать COM-программа?

Ответ: COM-программа должна содержать один сегмент кода, где находятся данные, код.

- 2) Сколько сегментов должна содержать EXE-программа?

Ответ: EXE-программа должна содержать три сегмента – сегмент стека, сегмент данных, сегмент кода.

Исправление: EXE-программа может содержать несколько программных сегментов, включая сегмент стека, сегмент данных и сегмент кода.

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

Ответ: `assume` – директива, сообщающая транслятору, о том, какому сегментному регистру соответствует какой сегмент. Директива `org 100h` сообщает компилятору, что всю адресацию нужно сместить на 256 байт, где будет располагаться PSP.

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

Ответ: нет, т.к. отсутствует таблица настроек(`relocation table`), в которой находится соответствие фактических адресов сегментов и абсолютных ссылок на сегменты, следовательно нельзя использовать команды, которые используют адрес сегмента и дальнюю адресацию.

Отличия форматов файлов COM и EXE модулей

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

Ответ: код начинается с 0 адреса, что приведено на рис. 2.

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

Ответ: с адреса 0 располагается таблица настроек, с 300h адреса располагается код. Иллюстрацию данного файла можно увидеть на рис. 3.

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

Ответ: так же начиная с 0 адреса располагается таблица настроек, но код начинается с 200h адреса, т.к. в «плохом» EXE-файле была использована

0000000000: E9 DB 01 50 43 20 74 79	70 65 3A 20 24 0D 0A 53	éÜ@PC type: \$!\$S
0000000010: 79 73 74 65 6D 20 76 65	72 73 69 6F 6E 3A 20 20	ystem version:
0000000020: 2E 20 20 0D 0A 24 4F 45	4D 20 6E 75 6D 62 65 72	. \$!\$OEM number
0000000030: 3A 20 20 20 20 20 20 0D	0A 24 55 73 65 72 20 73	: \$!\$User s
0000000040: 65 72 69 61 6C 20 6E 75	6D 62 65 72 3A 20 20 20	erial number:
0000000050: 20 20 20 20 20 20 20 20	20 20 20 0D 0A 24 50 53	\$!\$PS
0000000060: 32 20 6D 6F 64 65 6C 20	38 30 20 24 50 43 20 43	2 model 80 \$PC C
0000000070: 6F 6E 76 65 72 74 69 62	6C 65 20 24 50 53 32 20	onvertible \$PS2
0000000080: 6D 6F 64 65 6C 20 33 30	20 24 50 43 2F 58 54 20	model 30 \$PC/XT
0000000090: 24 41 54 20 24 50 43 6A	72 20 24 50 43 20 24 B4	\$AT \$PCjr \$PC \$
00000000A0: 09 CD 21 C3 24 0F 3C 09	76 02 04 07 04 30 C3 51	oÍ!Ã\$α<ov0♦♦0ÃQ
00000000B0: 8A C4 E8 EF FF 86 C4 B1	04 D2 E8 E8 E6 FF 59 C3	ŠÄëÿtÄ±ðèèæÿYÃ
00000000C0: 53 8A FC E8 E9 FF 88 25	4F 88 05 4F 8A C7 32 E4	SŠüèëÿ~%0~♣0ŠÇ2ä
00000000D0: E8 DC FF 88 25 4F 88 05	5B C3 51 52 50 32 E4 33	èÜÿ~%0~♣[ÃQRP2ä3
00000000E0: D2 B9 0A 00 F7 F1 80 CA	30 88 14 4E 33 D2 3D 0A	0¹☒ ÷ñ€Ê0~JN3D=☒
00000000F0: 00 73 F1 3D 00 00 76 04	0C 30 88 04 58 5A 59 C3	sñ= v♦90~♦XZYÃ
0000000100: 06 53 50 BB 00 F0 8E C3	26 A1 FE FF BA 03 01 3C	♣SP» ðŽÃ&¡þÿ°♥0<
0000000110: F8 74 29 3C F9 74 31 3C	FA 74 39 3C FB 74 41 3C	øt)<ùt1<ùt9<ùtA<
0000000120: FC 74 49 3C FD 74 51 3C	FF 74 59 8A E0 E8 7F FF	ütI<ÿtQ<ÿtYŠàèøÿ
0000000130: BB 03 01 89 47 09 E8 66	FF EB 55 90 E8 60 FF BA	»♥0%GoèfÿëU☒è`ÿ°
0000000140: 5E 01 E8 5A FF EB 49 90	E8 54 FF BA 6C 01 E8 4E	^0èZÿëI☒èTÿ°l0èN
0000000150: FF EB 3D 90 E8 48 FF BA	7C 01 E8 42 FF EB 31 90	ÿë=☒èHÿ° 0èBÿë1☒
0000000160: E8 3C FF BA 8A 01 E8 36	FF EB 25 90 E8 30 FF BA	è<ÿ°Š0è6ÿë%☒è0ÿ°
0000000170: 91 01 E8 2A FF EB 19 90	E8 24 FF BA 95 01 E8 1E	‘0è*ÿë↓☒è\$ÿ°•0è▲
0000000180: FF EB 0D 90 E8 18 FF BA	9B 01 E8 12 FF EB 01 90	ÿë☒☒è↑ÿ°>0è↑ÿë0☒
0000000190: 5A 58 5B 07 C3 50 56 BE	0D 01 83 C6 12 E8 3A FF	ZX[•ÃPV%J0fÆ†è:ÿ
00000001A0: 83 C6 03 8A C4 E8 32 FF	5E 58 C3 50 53 56 8A C7	fÆ♥ŠÄè2ÿ^XÃPSVŠÇ
00000001B0: BE 26 01 83 C6 0E E8 21	FF 5E 5B 58 C3 50 53 51	%&0fÆñè!ÿ^[XÃPSQ
00000001C0: 56 8A C3 E8 E9 FE BF 3A	01 83 C7 16 89 05 8B C1	VŠÄèép¿:0fÇ=♣<Ã
00000001D0: BF 3A 01 83 C7 1B E8 E7	FE 5E 59 5B 58 C3 E8 1F	¿:0fÇ<èçb^Y[XÃè▼
00000001E0: FF B4 30 CD 21 E8 AD FF	E8 C0 FF E8 CF FF BA 0D	ÿ`0Í!è-ÿèÄÿèÏÿ°J
00000001F0: 01 E8 AB FE BA 26 01 E8	A5 FE BA 3A 01 E8 9F FE	0è«þ°&0èÿþ°:0èÿþ
0000000200: 32 C0 B4 3C CD 21 C3		2Ã<í!Ã

Рисунок 3 — Вид файла .COM в шестнадцатеричном виде

директива org 100h, резервирующая пространство для заголовка, там код располагается с 300h.

Исправление: В «плохом» EXE-файле код и данные размещены в одном сегменте, в «хорошем» – на каждый отведен отдельный сегмент. Таким образом структура «плохого» EXE-файла: PSP, код и данные, стек.

Структура «хорошего» EXE-файла: PSP, сегмент кода, сегмента данных сегмент стека. Иллюстрацию «хорошего» EXE-файла можно увидеть на рис.

4.

Загрузка COM модуля в основную память

Результаты загрузки COM модуля в основную память представлены на рис. 5.

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

000000000:	4D 5A 07 01 03 00 00 00	20 00 00 00 FF FF 00 00	MZ•@♥ ŷŷ
000000010:	00 00 00 00 00 01 00 00	3E 00 00 00 01 00 FB 71	@ > @ ŷq
000000020:	6A 72 00 00 00 00 00 00	00 00 00 00 00 00 00 00	jr
000000030:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000040:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000050:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000060:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000070:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000080:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000090:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000A0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000B0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000C0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000D0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000E0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000F0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000100:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000110:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000120:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000130:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000140:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000150:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000160:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000170:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000180:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000190:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000001A0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000001B0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000001C0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000001D0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000001E0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000001F0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000200:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000210:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000220:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000230:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000240:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000250:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000260:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000270:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000280:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000290:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000002A0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000002B0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000002C0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000002D0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000002E0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000002F0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000300:	E9 DB 01 50 43 20 74 79	70 65 3A 20 24 0D 0A 53	eÜ0PC type: \$!\$S
000000310:	79 73 74 65 6D 20 76 65	72 73 69 6F 6E 3A 20 20	ystem version:
000000320:	2E 20 20 0D 0A 24 4F 45	4D 20 6E 75 6D 62 65 72	. \$!\$OEM number
000000330:	3A 20 20 20 20 20 20 0D	0A 24 55 73 65 72 20 73	: \$!\$User s
000000340:	65 72 69 61 6C 20 6E 75	6D 62 65 72 3A 20 20 20	erial number:
000000350:	20 20 20 20 20 20 20 20	20 20 20 0D 0A 24 50 53	\$!\$PS
000000360:	32 20 6D 6F 64 65 6C 20	38 30 20 24 50 43 20 43	2 model 80 \$PC C
000000370:	6F 6E 76 65 72 74 69 62	6C 65 20 24 50 53 32 20	onvertible \$PS2
000000380:	6D 6F 64 65 6C 20 33 30	20 24 50 43 2F 58 54 20	model 30 \$PC/XT
000000390:	24 41 54 20 24 50 43 6A	72 20 24 50 43 20 24 B4	\$AT \$PCjr \$PC \$
0000003A0:	09 CD 21 C3 24 0F 3C 09	76 02 04 07 04 30 C3 51	oÍ!\$!\$<ov0♦♦0AQ
0000003B0:	8A C4 E8 EF FF 86 C4 B1	04 D2 E8 E8 E6 FF 59 C3	ŠÄëiŷtÄ±♦ÖëëŷYÄ
0000003C0:	53 8A FC E8 E9 FF 88 25	4F 88 05 4F 8A C7 32 E4	SŠüëŷ`%0*♦0\$C2ä
0000003D0:	E8 DC FF 88 25 4F 88 05	5B C3 51 52 50 32 E4 33	ëÜŷ`%0*♦[ÄQRP2ä3
0000003E0:	D2 B9 0A 00 F7 18 0A CA	30 88 14 4E 33 D2 3D 0A	Öt\$ ÷ñëÊ0`JN3Ö=
0000003F0:	00 73 F1 3D 00 00 76 04	0C 30 88 04 58 5A 59 C3	sñ= v♦90`♦XZYÄ
000000400:	06 53 50 BB 00 F0 8E C3	26 A1 FE FF BA 03 01 3C	♦SP» ðZÄ&jbŷ»♥@<
000000410:	F8 74 29 3C F9 74 31 3C	FA 74 39 3C FB 74 41 3C	øt)<üt1<üt9<ütA<
000000420:	FC 74 49 3C FD 74 51 3C	FF 74 59 8A E0 E8 7F FF	ütI<yQtQ<yTŷŠäëŷ
000000430:	BB 03 01 89 47 09 E8 66	FF EB 55 90 E8 60 FF BA	»♥0%GoëfŷÛÜÜë`ŷ»
000000440:	5E 01 E8 5A FF EB 49 90	E8 54 FF BA 6C 01 E8 4E	^0ëZŷëIÜëTŷ»10ëN
000000450:	FF EB 3D 90 E8 48 FF BA	7C 01 E8 42 FF EB 31 90	ŷë=ÜëHŷ» 0ëBŷë1Ü
000000460:	E8 3C FF BA 8A 01 E8 36	FF EB 25 90 E8 30 FF BA	ë<ŷ»Š0ë6ŷë%Üë0ŷ»
000000470:	91 01 E8 2A FF EB 19 90	E8 24 FF BA 95 01 E8 1E	^0ë*ŷë4Üë\$ŷ»•0ë▲
000000480:	FF EB 0D 90 E8 18 FF BA	9B 01 E8 12 FF EB 01 90	ŷëÜëTŷ»>0ë†ŷë0Ü
000000490:	5A 58 5B 07 C3 50 56 BE	0D 01 83 C6 12 E8 3A FF	ZX[•APV%»0fÆtë:ŷ
0000004A0:	83 C6 03 8A C4 E8 32 FF	5E 58 C3 50 53 56 8A C7	fÆ♥ŠÄë2ŷ^xÄPSVŠC
0000004B0:	BE 26 01 83 C6 0E E8 21	FF 5E 5B 58 C3 50 53 51	%&0fÆ!ŷ^[xÄPSQ
0000004C0:	56 8A C3 E8 59 FE BF 3A	01 83 C7 16 89 05 8B C1	VŠÄëëp;:0fC=»•<Ä
0000004D0:	BF 3A 01 83 C7 1B E8 E7	FE 5E 59 5B 58 C3 E8 1F	ç:0fC+ëçb^Y[xÄë♥
0000004E0:	FF B4 30 CD 21 E8 AD FF	E8 C0 FF E8 CF FF BA 0D	ŷ`0Í!è-ŷëAŷëÍŷ»»
0000004F0:	01 E8 AB FE BA 26 01 E8	A5 FE BA 3A 01 E8 9F FE	0ë«p»&0ëŷp»:0ëŷp
000000500:	32 C0 B4 3C CD 21 C3	2Ä`<Í!Ä	

Рисунок 4 — Вид «плохого» EXE файла в шестнадцатиричном виде

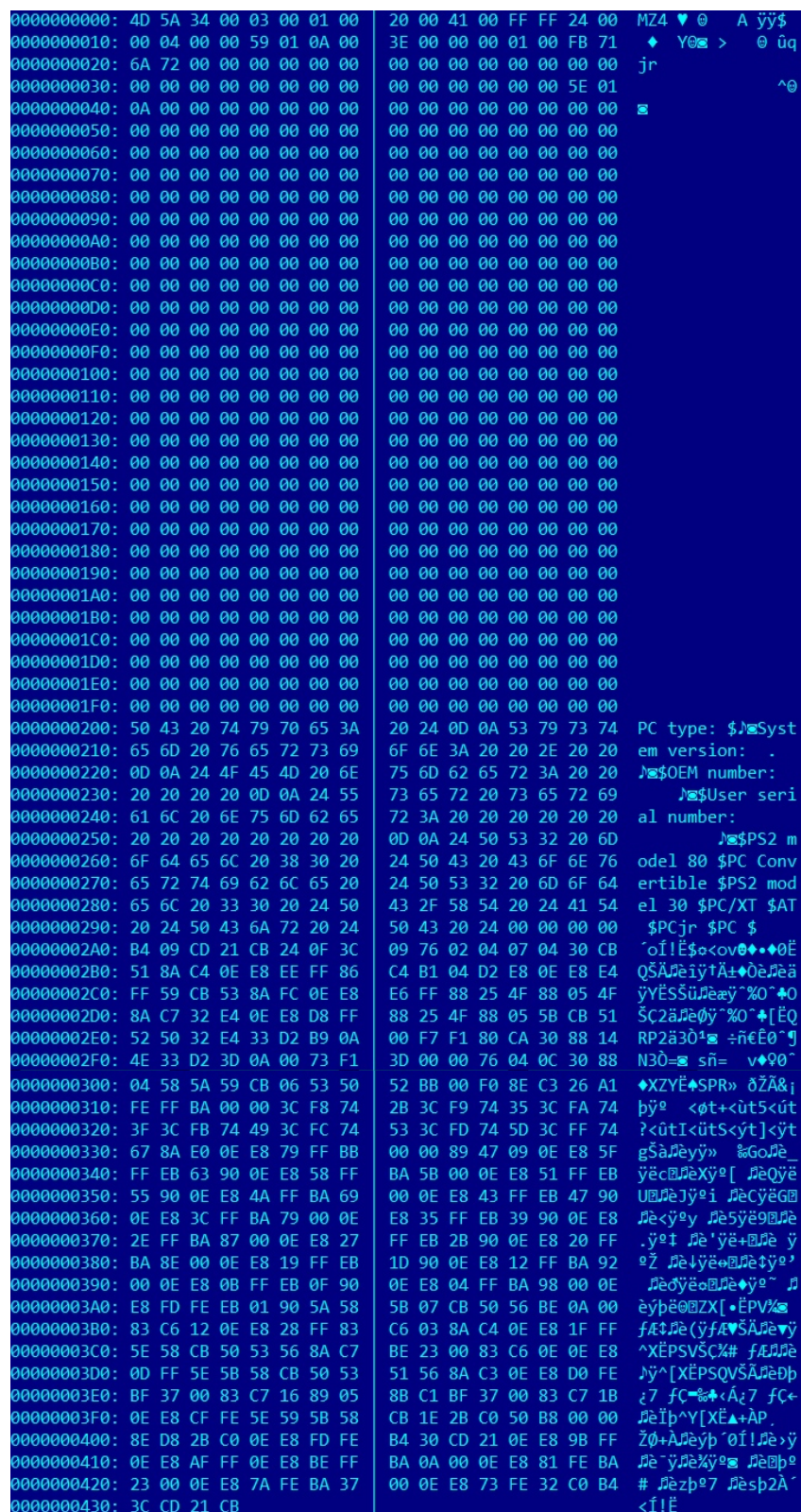


Рисунок 5 — Вид «хорошего» EXE файла в шестнадцатиричном виде

Ответ: сегментные регистры устанавливаются на начало PSP. Регистр SP устанавливается на конец PSP.

Исправление: регистр SP устанавливается не на конец PSP, а на верх стека.

2) Что располагается с адреса 0?

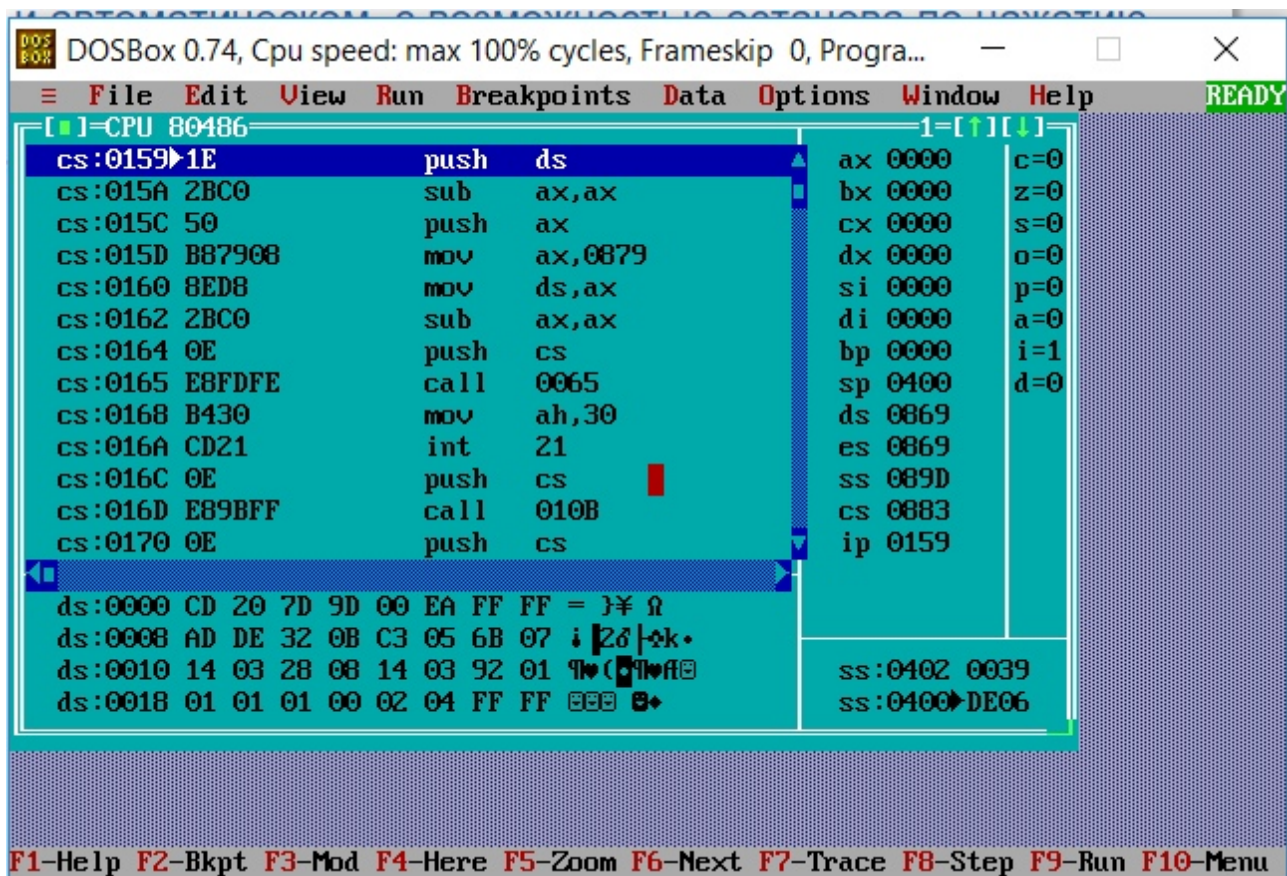


Рисунок 6 — Вид COM файла в TD.EXE отладчике

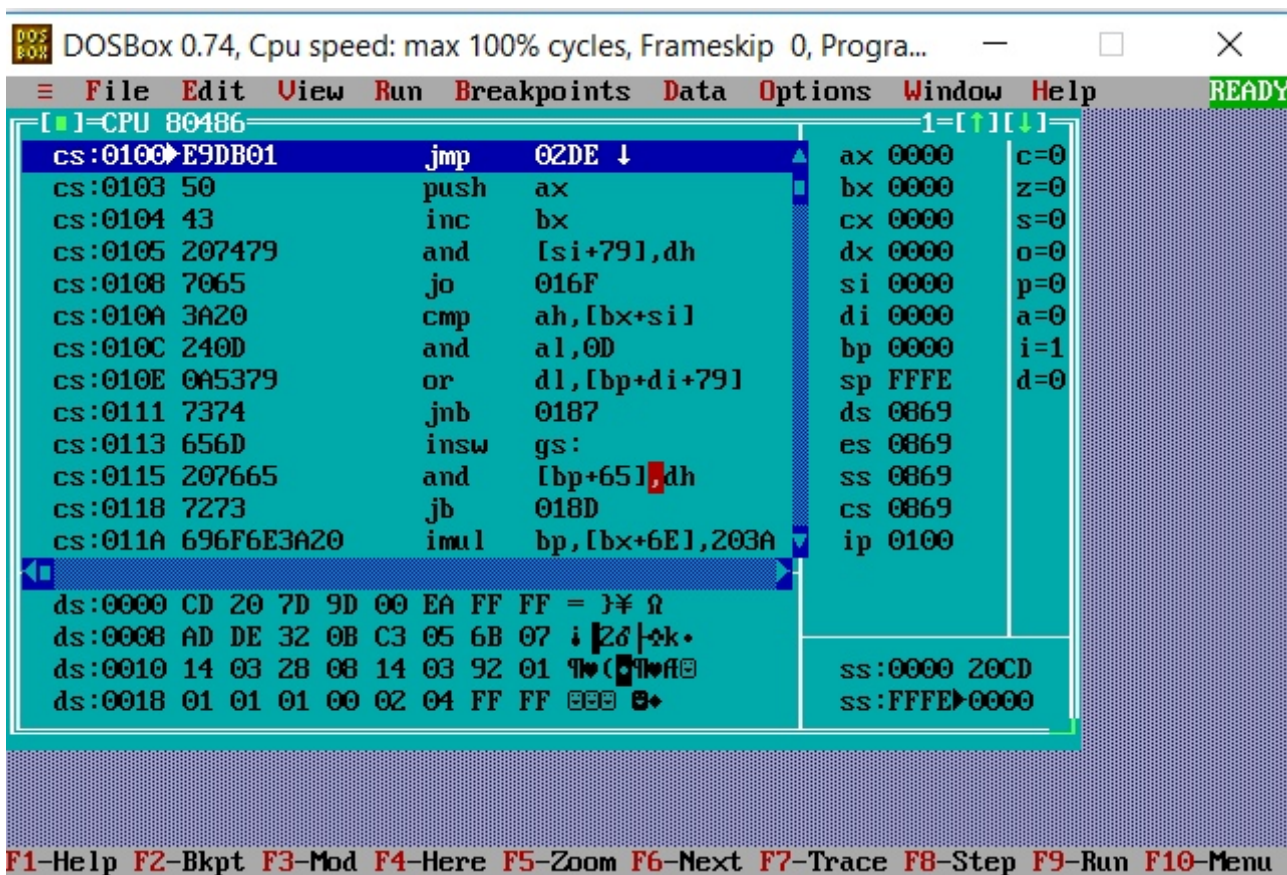


Рисунок 7 — Вид «хорошего» EXE файла в TD.EXE отладчике

Ответ: PSP.

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

Ответ: DS= 0869, ES=0869, SS=0869, CS=0869. Они указывают на начало PSP.

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

Ответ: DOS автоматически определяет стек. Если для программы размер сегмента в 64К достаточен, то DOS устанавливает SP=FFFE, что является верхом стека. В противном случае – устанавливает верх стека в конец памяти.

Загрузка «хорошего» EXE модуля в основную память

Результаты загрузки «хорошего» EXE модуля в основную память представлены на рис. 6.

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

Ответ: здесь все происходит сложнее, т.к. требуется настройка сегментных адресов. Регистры ES и DS устанавливаются на начало PSP(ES=DS=0869).

Каждому сегментному регистру в соответствие ставится адрес начала данного сегмента: SS=089D, CS=0883.

- 2) На что указывают регистры DS и ES?

Ответ: на начало PSP.

- 3) Как определяется стек?

Ответ: с помощью директивы DW 512 DUP(?).

- 4) Как определяется точка входа?

Ответ: с помощью директивы END <имя процедуры>, где имя процедуры – процедура, с которой должна начинаться программа.

Выводы.

В данной работе были исследованы различия в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способы их загрузки в основную память.