

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

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

Студент гр. 8381

Преподаватель

Почаев Н.А.

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

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

Задание.

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.

Постановка задачи.

Требуется реализовать 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 символьную 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.

Выполнение работы.

Выполнение работы производилось на базе DOSBox 0.74-3, в редакторе Notepad++. Сборка и отладка модулей производились с помощью компилятора MASM и отладчика AFD.

Был написан текст исходного .COM модуля, который определяет тип PC и информацию о системе. Полученный модуль был отлажен, и в результате были

получены «плохой» .EXE модуль и «хороший» .COM модуль (с помощью BIN2EXE).

Пример сборки и выполнения .COM модуля показан на рисунке 1.

```
C:\>exe2bin lab1com lab1com.com

C:\>LAB1COM.com
Your PC type: AT
Your version number: 5.0
Your OEM: 0
Your User ID: 0
C:\>
```

Рисунок 1 – Полученный .COM модуль и его вывод

Во время линковки «плохого» .EXE модуля было выведено предупреждение об отсутствии сегмента стека, представленное на рис. 2.

```
C:\>masm LAB1COM.ASM
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [LAB1COM.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

49958 + 451160 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>link LAB1COM.obj

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LAB1COM.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>
```

Рисунок 2 – Предупреждение во время линковки

Результаты выполнения плохого EXE. Модуля представлены на рисунке 3.

```
C:\>Lab1com.exe

      Your PC type:

      Your PC type:

      Your PC type: 5
      Your PC type: 0
                                     Your PC type: 0
      Your PC type: 0
C:\>_
```

Рисунок 3 – результат выполнения плохого EXE модуля.

Был написан текст исходного .EXE модуля, выполняющий аналогичные функции, что и .COM модуль. Результат его выполнения представлен на рис. 4.

```
C:\>link lab1exe.obj
Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LAB1EXE.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>lab1exe
Your PC type: AT
Your version number: 5.0
Your OEM: 0
Your User ID: 0
C:\>
```

Рисунок 4 – Вывод правильного EXE модуля

Отличия исходных текстов COM и EXE программ

1) Сколько сегментов должна содержать COM программа?

.COM - программы содержат только один сегмент. Модель памяти tiny - код, данные и стек объединены в один физический сегмент, максимальный размер которого не мог превышать 64 Кбайта без 256 байтов (последние требуются для создания префикса программного сегмента (PSP)).

2) EXE программа?

EXE-программа может содержать несколько сегментов. При использовании модели памяти small, в программе должен содержаться один сегмент данных и один сегмент кода в разных физических сегментах и каждый из них не может превосходить 64 Кбайта. При других моделях памяти (например large) есть возможность использования нескольких сегментов данных и (или) нескольких сегментов кода.

Помимо этого, в программе должен быть описан сегмент стека (до 64 Кбайт, содержит адреса возврата как для программы (для возврата в операционную систему), так и для вызовов подпрограмм (для возврата в главную программу), а также используется для передачи параметров в процедуры). Использует операционная система при обработке прерываний. Регистр сегмента стека (SS) адресует данный сегмент. Адрес текущей вершины стека задается регистрами SS:ESP (для 32-х разрядных регистров) и SS:RSP (для 64-х разрядных регистров).

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

Обязательными являются директивы ASSUME и ORG. Первая используется для возможности использовать директивы SEGMENT и ENDS как сегменты кода (ASSUME сообщает транслятору связь между сегментами и сегментными регистрами). Привязка осуществляется с помощью операндов директивы, где имя_сегмента — определено ключевым словом nothing или в исходном тексте программы директивой SEGMENT.

ORG, в свою очередь, устанавливает относительный адрес для начала выполнения программы. COM программе необходимо зарезервировать первые 100h байт для PSP при запуске.

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

В .COM файле отсутствует таблица настройки с информацией о типе адресов и их местоположении в коде. Поэтому нельзя использовать команды, связанные с адресом сегмента, так как адрес сегмента неизвестен вплоть до загрузки этого сегмента в память. Загрузчику необходима информация о местоположении в файле загрузочного модуля полей адресов.

Отличия форматов файлов COM и EXE модулей

1) Какова структура файла COM? С какого адреса располагается код? Вид файла COM в шестнадцатеричном формате представлен на рис. 5.

COM-файл состоит из одного сегмента, а размер файла не превышает 64 КБ. Код располагается с нулевого адреса, что видно на рис. 7. DATA Segment заканчиваются на адресе A0. С B0 с 4-го байта начинается CODE Segment. Между сегментами существует промежуток в 6 байт (20h) – связано с GPR и с 6-ти сегментным строением сегмента.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Текст декодирован
00000000	E9	D2	00	59	6F	75	72	20	50	43	20	74	79	70	65	3A	XT.Your PC type:
00000010	20	24	50	43	0D	0A	24	50	43	2F	58	54	0D	0A	24	41	\$PC..\$PC/XT..\$A
00000020	54	24	0D	0A	24	50	53	32	20	28	33	30	20	6D	6F	64	T\$..\$PS2 (30 mod
00000030	65	6C	29	0D	0A	24	50	53	32	20	28	35	30	20	6F	72	el)..\$PS2 (50 or
00000040	20	36	30	20	6D	6F	64	65	6C	29	0D	0A	24	50	53	32	60 model)..\$PS2
00000050	20	28	38	30	20	6D	6F	64	65	6C	29	0D	0A	24	50	43	(80 model)..\$PC
00000060	20	6A	72	0D	0A	24	50	43	20	43	6F	6E	76	65	72	74	jr..\$PC Convert
00000070	69	62	6C	65	0D	0A	24	2E	24	0D	0A	59	6F	75	72	20	ible..\$.\$.Your
00000080	76	65	72	73	69	6F	6E	20	6E	75	6D	62	65	72	3A	20	version number:
00000090	24	0D	0A	59	6F	75	72	20	4F	45	4D	3A	20	24	0D	0A	\$..Your OEM: \$..
000000A0	59	6F	75	72	20	55	73	65	72	20	49	44	3A	20	24	B4	Your User ID: \$r
000000B0	09	CD	21	C3	33	C9	8B	DA	33	D2	F7	F3	52	41	85	C0	.H!ГЗЙ«ЪЗТчyRA..A
000000C0	75	F6	B4	02	5A	80	FA	09	76	03	80	C2	07	80	C2	30	ущг'.ZЪъ.v.ЪВ.ЪВ0
000000D0	CD	21	E2	F0	C3	BA	03	01	E8	D4	FF	B8	00	F0	8E	C0	H!врГе..ифяё.рЪА
000000E0	B8	00	00	26	A0	FE	FF	3C	FF	74	23	3C	FE	74	25	3C	ё...& юя<ят#<ют%<
000000F0	FB	74	21	3C	FC	74	23	3C	FA	74	25	3C	FC	74	27	3C	ыт!<ьт#<ьт%<ьт'<
00000100	F8	74	29	3C	FD	74	2B	3C	F9	74	2D	EB	31	90	BA	12	шт)<эт+<шт-л!ђе.
00000110	01	EB	34	90	BA	17	01	EB	2E	90	BA	1F	01	EB	28	90	.л4ђе..л.ђе..л(ђ
00000120	BA	25	01	EB	22	90	BA	36	01	EB	1C	90	BA	4D	01	EB	е%.л"ђе6.л.ђеM.л
00000130	16	90	BA	5E	01	EB	10	90	BA	66	01	EB	0A	90	BA	10	.ђе^..л.ђеf.л.ђе.
00000140	00	E8	70	FF	EB	04	90	E8	65	FF	B4	30	CD	21	51	53	.ирял.ђиеяг'0H!QS
00000150	50	BA	79	01	E8	58	FF	B8	00	00	58	50	B4	00	BA	0A	Реу.иХяё..XPr.е.
00000160	00	E8	50	FF	BA	77	01	E8	45	FF	58	8A	E5	B4	00	8A	.иРяew.иЕяХЬег'.Ъ
00000170	C5	BA	0A	00	E8	3D	FF	BA	91	01	E8	32	FF	58	50	8A	Ее..и=яе'.и2яХРЪ
00000180	E5	B4	00	8A	C5	BA	0A	00	E8	29	FF	BA	9E	01	E8	1E	ег'.ЪЕе..и) яећ.и.
00000190	FF	58	B4	00	3C	00	74	06	BA	0A	00	E8	16	FF	58	BA	яХг'.<.т.е..и.яХе
000001A0	0A	00	E8	0F	FF	32	C0	B4	4C	CD	21						...и.я2Аг'LN!

Рисунок 5 – Вид COM файла в шестнадцатеричном виде

COM-файл состоит из одного сегмента, а размер файла не превышает 64 КБ. Код располагается с нулевого адреса, что видно на рис. 7. DATA Segment заканчиваются на адресе A0. С B0 с 4-го байта начинается CODE Segment. Между сегментами существует промежуток в 6 байт (20h) – связано с GPR и с 6-ти сегментным строением сегмента. Далее располагаются команды, отвечающие за код. Это показано на рисунке 6 и 7.

000000E0 B8 00 00 26 A0 FE FF 3C FF 74 23 3C FE 74 25 3C ё...& юя<ят#<ют%<

Рисунок 6 – Фрагмент 16-ого представления COM файла

AX 0000	SI 0000	CS 119C	IP 01E3	Stack +0 0000	FLAGS 0200																
BX 0000	DI 0000	DS 119C		+2 0000																	
CX 01AB	BP 0000	ES F000	HS 119C	+4 0000	OF	DF	IF	SF	ZF	AF	PF	CF									
DX 0103	SP FFF5	SS 119C	FS 119C	+6 0000	0	0	1	0	0	0	0	0									
CMD >				1	0	1	2	3	4	5	6	7									
FC				DS:0000	CD	20	CC	46	00	EA	FD	FF									
01E0 B80000	MOV	AX,0000		DS:0008	AD	DE	ED	04	92	01	00	00									
01E3 26A0FEFF	MOV	AL,ES:[FFFE]		DS:0010	18	01	10	01	18	01	92	01									
01E7 3CFF	CMP	AL,FF		DS:0018	03	FF	FF	FF	FF	FF	FF	FF									
01E9 7423	JZ	020E		DS:0020	FF	FF	FF	FF	FF	FF	FF	FF									
01EB 3CFE	CMP	AL,FE		DS:0028	FF	FF	FF	FF	96	11	D5	FF									
01ED 7425	JZ	0214		DS:0030	9C	11	14	00	18	00	9C	11									
01EF 3CFB	CMP	AL,FB		DS:0038	FF	FF	FF	FF	00	00	00	00									
01F1 7421	JZ	0214		DS:0040	05	00	00	00	00	00	00	00									
01F3 3CFC	CMP	AL,FC		DS:0048	00	00	00	00	00	00	00	00									
2				0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
DS:0000	CD	20	CC	46	00	EA	FD	FF	AD	DE	ED	04	92	01	00	00				.F....	
DS:0010	18	01	10	01	18	01	92	01	03	FF	FF	FF	FF	FF	FF	FF					
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	96	11	D5	FF					
DS:0030	9C	11	14	00	18	00	9C	11	FF	FF	FF	FF	00	00	00	00					
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
1 Step				2StepProc	3Retrieve	4 Help	5Set BRK	6	7 up	8 dn	9 le	0 ri									

Рисунок 7 – Адреса команд, найденные в отладчике

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

Что располагается с адреса 0?

Вид «плохого» EXE файла в шестнадцатеричном формате представлен на рис. 8.

[illegible]

```

00000300 E9 D2 00 59 6F 75 72 20 50 43 20 74 79 70 65 3A йТ.Your PC type:
00000310 20 24 50 43 0D 0A 24 50 43 2F 58 54 0D 0A 24 41 $PC..$PC/XT..$A
00000320 54 24 0D 0A 24 50 53 32 20 28 33 30 20 6D 6F 64 T$. $PS2 (30 mod
00000330 65 6C 29 0D 0A 24 50 53 32 20 28 35 30 20 6F 72 el)..$PS2 (50 or
00000340 20 36 30 20 6D 6F 64 65 6C 29 0D 0A 24 50 53 32 60 model)..$PS2
00000350 20 28 38 30 20 6D 6F 64 65 6C 29 0D 0A 24 50 43 (80 model)..$PC
00000360 20 6A 72 0D 0A 24 50 43 20 43 6F 6E 76 65 72 74 jr..$PC Convert
00000370 69 62 6C 65 0D 0A 24 2E 24 0D 0A 59 6F 75 72 20 ible..$. $.Your
00000380 76 65 72 73 69 6F 6E 20 6E 75 6D 62 65 72 3A 20 version number:
00000390 24 0D 0A 59 6F 75 72 20 4F 45 4D 3A 20 24 0D 0A $.Your OEM: $.
000003A0 59 6F 75 72 20 55 73 65 72 20 49 44 3A 20 24 B4 Your User ID: $r
000003B0 09 CD 21 C3 33 C9 8B DA 33 D2 F7 F3 52 41 85 C0 .Н!ГЗЙ«ЪЗТчуRA..А
000003C0 75 F6 B4 02 5A 80 FA 09 76 03 80 C2 07 80 C2 30 ищг'.ZЪъ.v.ЪВ.ЪВ0
000003D0 CD 21 E2 F0 C3 BA 03 01 E8 D4 FF B8 00 F0 8E C0 Н!врГе..ифяё.рЪА
000003E0 B8 00 00 26 A0 FE FF 3C FF 74 23 3C FE 74 25 3C ё...& юя<ят#<ют%<
000003F0 FB 74 21 3C FC 74 23 3C FA 74 25 3C FC 74 27 3C ыт!<ът#<ът%<ът'<
00000400 F8 74 29 3C FD 74 2B 3C F9 74 2D EB 31 90 BA 12 шт)<эт+<шт-лlђе.
00000410 01 EB 34 90 BA 17 01 EB 2E 90 BA 1F 01 EB 28 90 .л4ђе...л.ђе...л(ђ
00000420 BA 25 01 EB 22 90 BA 36 01 EB 1C 90 BA 4D 01 EB е%.л"ђеб.л.ђеМ.л
00000430 16 90 BA 5E 01 EB 10 90 BA 66 01 EB 0A 90 BA 10 .ђе^..л.ђеf..л.ђе.
00000440 00 E8 70 FF EB 04 90 E8 65 FF B4 30 CD 21 51 53 .ирял.ђиеняг'ОН!QS
00000450 50 BA 79 01 E8 58 FF B8 00 00 58 50 B4 00 BA 0A Реу.иХяё...ХРг'.е.
00000460 00 E8 50 FF BA 77 01 E8 45 FF 58 8A E5 B4 00 8A .иРяew.иЕяХЉег'.Љ
00000470 C5 BA 0A 00 E8 3D FF BA 91 01 E8 32 FF 58 50 8A Ее..и=яе'.и2яХРЉ
00000480 E5 B4 00 8A C5 BA 0A 00 E8 29 FF BA 9E 01 E8 1E ег'.ЉЕе..и)яећ.и.
00000490 FF 58 B4 00 3C 00 74 06 BA 0A 00 E8 16 FF 58 BA яХг'.<.т.е..и.яХе
000004A0 0A 00 E8 0F FF 32 C0 B4 4C CD 21 ..и.я2Аг'ЛН!

```

Рисунок 9 – Вид «плохого» EXE файла

В «плохом» EXE-файле код располагается с адреса 300h. С нулевого адреса располагается управляющая информация для загрузчика, образующая заголовок.

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

[illegible]

```

000002D0 59 6F 75 72 20 50 43 20 74 79 70 65 3A 20 24 50 Your PC type: $P
000002E0 43 0D 0A 24 50 43 2F 58 54 0D 0A 24 41 54 24 0D C..$PC/XT..$AT$.
000002F0 0A 24 50 53 32 20 28 33 30 20 6D 6F 64 65 6C 29 .$PS2 (30 model)
00000300 0D 0A 24 50 53 32 20 28 35 30 20 6F 72 20 36 30 ..$PS2 (50 or 60
00000310 20 6D 6F 64 65 6C 29 0D 0A 24 50 53 32 20 28 38 model)..$PS2 (8
00000320 30 20 6D 6F 64 65 6C 29 0D 0A 24 50 43 20 6A 72 0 model)..$PC jr
00000330 0D 0A 24 50 43 20 43 6F 6E 76 65 72 74 69 62 6C ..$PC Convertibl
00000340 65 0D 0A 24 2E 24 0D 0A 59 6F 75 72 20 76 65 72 e..$.$.Your ver
00000350 73 69 6F 6E 20 6E 75 6D 62 65 72 3A 20 24 0D 0A sion number: $.
00000360 59 6F 75 72 20 4F 45 4D 3A 20 24 0D 0A 59 6F 75 Your OEM: $.You
00000370 72 20 55 73 65 72 20 49 44 3A 20 24 00 00 00 00 r User ID: $....
00000380 B4 09 CD 21 C3 33 C9 8B DA 33 D2 F7 F3 52 41 85 r.Н!ГЗЙк'бЗТчуRA...
00000390 C0 75 F6 B4 02 5A 80 FA 09 76 03 80 C2 07 80 C2 Ауцг'.ЗЪъ.v.ЪВ.ЪВ
000003A0 30 CD 21 E2 F0 C3 1E 2B C0 50 B8 0D 00 8E D8 BA ОН!врГ. +APё..ЪШё
000003B0 00 00 E8 CB FF B8 00 F0 8E C0 B8 00 00 26 A0 FE ..иЛяё.рЪАё...& ю
000003C0 FF 3C FF 74 23 3C FE 74 25 3C FB 74 21 3C FC 74 я<ят#<ют%<ыт!<ыт
000003D0 23 3C FA 74 25 3C FC 74 27 3C F8 74 29 3C FD 74 #<ыт%<ыт'<шт)<эт
000003E0 2B 3C F9 74 2D EB 31 90 BA 0F 00 EB 34 90 BA 14 +<шт-л!ёе..л4ёе.
000003F0 00 EB 2E 90 BA 1C 00 EB 28 90 BA 22 00 EB 22 90 .л.ёе..л(ёе".л"ё
00000400 BA 33 00 EB 1C 90 BA 4A 00 EB 16 90 BA 5B 00 EB еЗ.л.ёеJ.л.ёе[.л
00000410 10 90 BA 63 00 EB 0A 90 BA 10 00 E8 67 FF EB 04 .ёеc.л.ёе..ицял.
00000420 90 E8 5C FF B4 30 CD 21 51 53 50 BA 76 00 E8 4F ъи\яг'ОН!QSPev.иО
00000430 FF B8 00 00 58 50 B4 00 BA 0A 00 E8 47 FF BA 74 яё..XPг'.е..иГяет
00000440 00 E8 3C FF 58 8A E5 B4 00 8A C5 BA 0A 00 E8 34 .и<яХЪег'.ЪЕе..и4
00000450 FF BA 8E 00 E8 29 FF 58 50 8A E5 B4 00 8A C5 BA яеЪ.и)яХРЪег'.ЪЕе
00000460 0A 00 E8 20 FF BA 9B 00 E8 15 FF 58 B4 00 3C 00 ..и яе>.и.яХг'.<.
00000470 74 06 BA 0A 00 E8 0D FF 58 BA 0A 00 E8 06 FF CB т.е..и.яХе..и.яЛ

```

Рисунок 10 – Вид «хорошего» EXE файла

В «хорошем» EXE с нулевого адреса также располагается заголовок. Также перед кодом располагается сегмент стека. Так, при размере стека 200h код располагается с адреса 400h. Если из исходного текста .EXE-программы убрать сегмент стека, то код будет располагаться с адреса 200h. Отличие от «плохого» EXE в том, что в «хорошем» не резервируется дополнительно 100h, которые в COM файле требовались для PSP, поэтому адреса начала кода отличаются на 100h + SS, где SS – размер стека.

Загрузка COM модуля в основную память

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

Запуск файла .COM в отладчике AFD.EXE представлен на рис. 12.

AX 0000	SI 0000	CS 119C	IP 0100	Stack +0 0000	FLAGS 0200																
BX 0000	DI 0000	DS 119C		+2 0000																	
CX 01AB	BP 0000	ES 119C	HS 119C	+4 0000	OF	DF	IF	SF	ZF	AF	PF	CF									
DX 0103	SP FFF5	SS 119C	FS 119C	+6 0000	0	0	1	0	0	0	0	0									
CMD >				1	0	1	2	3	4	5	6	7									
				DS:0000	CD	20	CC	46	00	EA	FD	FF									
				DS:0008	AD	DE	ED	04	92	01	00	00									
0100 E9D200	JMP		01D5	DS:0010	18	01	10	01	18	01	92	01									
0103 59	POP		CX	DS:0018	03	FF	FF	FF	FF	FF	FF	FF									
0104 6F	OUTSW			DS:0020	FF	FF	FF	FF	FF	FF	FF	FF									
0105 7572	JNZ		0179	DS:0028	FF	FF	FF	FF	96	11	C4	FF									
0107 205043	AND		[BX+SI+43],DL	DS:0030	92	01	14	00	18	00	9C	11									
010A 207479	AND		[SI+79],DH	DS:0038	FF	FF	FF	FF	00	00	00	00									
010D 7065	JO		0174	DS:0040	05	00	00	00	00	00	00	00									
010F 3A20	CMP		AH,[BX+SI]	DS:0048	00	00	00	00	00	00	00	00									
2				0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
DS:0000	CD	20	CC	46	00	EA	FD	FF	AD	DE	ED	04	92	01	00	00		.	F
DS:0010	18	01	10	01	18	01	92	01	03	FF	FF	FF	FF	FF	FF	FF			
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	96	11	C4	FF				
DS:0030	92	01	14	00	18	00	9C	11	FF	FF	FF	FF	00	00	00	00			
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			
1 Step				2 StepProc				3 Retrieve				4 Help				5 Set BRK				6	
																7 up				8 dn	
																9 le				0 ri	

Рисунок 11 – Отладка файла .COM

Определяется сегментный адрес свободного участка ОП, в который можно загрузить программу. Создается блок памяти. В поля PSP заносятся значения. Загружается COM файл со смещением 100h. Сегментные регистры устанавливаются на адрес сегмента PSP, регистр SP указывает на конец сегмента, туда записывается 0000h. С ростом стека значение SP будет уменьшаться. Счетчик команд принимает значение 100h. Программа запускается.

2) Что располагается с адреса 0?

С нулевого адреса (0h) располагается сегмент PSP.

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

Сегментные регистры имеют значения 48DDh и указывают на PSP.

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

В COM модуле нельзя объявить стек, он создается автоматически. На рис. 8 видно, что SP указывает на FFFEH. Стек занимает оставшуюся память (из 64 Кб), а его адреса изменяются от больших к меньшим, то есть от FFFEH к 0000h.

Загрузка «хорошего» EXE. Модуля в основную память

Запуск «хорошего» EXE модуля в отладчике AFD.EXE представлен на рис.

12

AX 0000	SI 0000	CS 11C4	IP 0026	Stack +0 0000	FLAGS 0200												
BX 0000	DI 0000	DS 119C		+2 0000													
CX 0280	BP 0000	ES 119C	HS 119C	+4 0000	OF	DF	IF	SF	ZF	AF	PF	CF					
DX 0000	SP 00C8	SS 11AC	FS 119C	+6 0000	0	0	1	0	0	0	0	0					
CMD >				1	0	1	2	3	4	5	6	7					
				DS:0000	CD	20	9C	11	00	EA	FD	FF					
				DS:0008	AD	DE	ED	04	92	01	00	00					
0026 1E	PUSH DS			DS:0010	18	01	10	01	18	01	92	01					
0027 2BC0	SUB AX,AX			DS:0018	03	FF	FF	FF	FF	FF	FF	FF					
0029 50	PUSH AX			DS:0020	FF	FF	FF	FF	FF	FF	FF	FF					
002A B8B911	MOV AX,11B9			DS:0028	FF	FF	FF	FF	96	11	C4	FF					
002D 8ED8	MOV DS,AX			DS:0030	92	01	14	00	18	00	9C	11					
002F BA0000	MOV DX,0000			DS:0038	FF	FF	FF	FF	00	00	00	00					
0032 E8CBFF	CALL 0000			DS:0040	05	00	00	00	00	00	00	00					
0035 B800F0	MOV AX,F000			DS:0048	00	00	00	00	00	00	00	00					
2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
DS:0000	CD	20	9C	11	00	EA	FD	FF	AD	DE	ED	04	92	01	00	00	
DS:0010	18	01	10	01	18	01	92	01	03	FF	FF	FF	FF	FF	FF	FF	
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	96	11	C4	FF		
DS:0030	92	01	14	00	18	00	9C	11	FF	FF	FF	FF	00	00	00	00	
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
1 Step 2StepProc 3Retrieve 4 Help 5Set BRK 6 7 up 8 dn 9 le 0 ri																	

Рисунок 13 – Отладка «хорошего» EXE модуля

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

Определяется сегментный адрес свободного участка Оперативной памяти, в который можно загрузить программу. Создается блок памяти для PSP и программы. После запуска программы DS и ES указывают на начало PSP (48DDh), CS – на начало сегмента команд (4919h), а SS – на начало сегмента стека (48EDh). IP имеет ненулевое значение, так как в программе есть дополнительные процедуры, расположенные до основной.

В PSP заносятся соответствующие значения. В рабочую область загрузчика

считывается форматированная часть заголовка файла. Определяется смещение начала загрузочного модуля в EXE файле. Вычисляется сегментный адрес (START_SEG) для загрузки. В память считывается загрузочный модуль. Таблица настройки порциями считывается в рабочую память. Для каждого элемента таблицы настройки к полю сегмента прибавляется сегментный адрес начального сегмента (в результате элемент таблицы указывает на нужное слово в памяти). Управление передается загруженной задаче по адресу из заголовка.

2) На что указывают регистры DS и ES?

Изначально регистры DS и ES указывают на начало сегмента PSP. Именно поэтому в начале программы для корректной работы с данными необходимо загрузить в DS адрес сегмента данных.

3) Как определяется стек?

Стек может быть объявлен при помощи директивы ASSUME, которая задает значение SP, указанное в заголовке и устанавливает сегментный регистр SS на начало сегмента стека. Также стек может быть объявлен с помощью директивы STACK. Вид программы EXE модуля без объявленного стека после команды push в отладчике представлен на рис. 13.

AX 0000	SI 0000	CS 11C4	IP 0026	Stack +0 0000	FLAGS 0200													
BX 0000	DI 0000	DS 119C		+2 0000														
CX 0280	BP 0000	ES 119C	HS 119C	+4 0000	OF	DF	IF	SF	ZF	AF	PF	CF						
DX 0000	SP 00C8	SS 11AC	FS 119C	+6 0000	0	0	1	0	0	0	0	0						
CMD >				1	0	1	2	3	4	5	6	7						
				DS:0000	CD	20	9C	11	00	EA	FD	FF						
				DS:0008	AD	DE	ED	04	92	01	00	00						
0026 1E	PUSH DS			DS:0010	18	01	10	01	18	01	92	01						
0027 2BC0	SUB AX,AX			DS:0018	03	FF	FF	FF	FF	FF	FF	FF						
0029 50	PUSH AX			DS:0020	FF	FF	FF	FF	FF	FF	FF	FF						
002A B8B911	MOV AX,11B9			DS:0028	FF	FF	FF	FF	96	11	C4	FF						
002D 8ED8	MOV DS,AX			DS:0030	92	01	14	00	18	00	9C	11						
002F BA0000	MOV DX,0000			DS:0038	FF	FF	FF	FF	00	00	00	00						
0032 E8CBFF	CALL 0000			DS:0040	05	00	00	00	00	00	00	00						
0035 B800F0	MOV AX,F000			DS:0048	00	00	00	00	00	00	00	00						
2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
DS:0000	CD	20	9C	11	00	EA	FD	FF	AD	DE	ED	04	92	01	00	00		
DS:0010	18	01	10	01	18	01	92	01	03	FF	FF	FF	FF	FF	FF	FF		
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	96	11	C4	FF			
DS:0030	92	01	14	00	18	00	9C	11	FF	FF	FF	FF	00	00	00	00		
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
1 Step	2StepProc	3Retrieve	4 Help	5Set BRK	6	7 up	8 dn	9 le	0 ri									

Рисунок 13 – Отладка EXE модуля без объявленного стека

4) Как определяется точка входа?

Смещение точки входа в программу загружается в указатель команд IP и определяется операндом директивы END <метка для входа>, который называется точкой входа.

Операндом является функция или метка, с которой необходимо начать программу.

Выводы

В ходе выполнения 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.

