

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

Перелыгин Д.С.

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

    0 Your PC type:

    0 Your PC type:

    0 Your PC type: 5
0 Your PC type: 0

                                0 Your PC type: 0

    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-программа содержит только один сегмент.

2. Сколько сегментов должна содержать EXE-программа?

Один и более сегментов.

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

Обязательна директива org со значением 100h, которая задает смещение для начала выполнения программы, потому что 256-байтовый блок до этого смещения зарезервирован системой под PSP.

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

Нельзя использовать директиву seg, особенно учитывая то, что сегмент всего один.

Отличия форматов файлов .COM и .EXE модулей.

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

Файл COM представляет собой непосредственно код в виде машинных команд и данные программы, которые расположены с нулевого адреса, как показано на скриншоте.

0000000000: E9 D2 00 59 6F 75 72 20	50 43 20 74 79 70 65 3A	йт Your PC type:
0000000010: 20 24 50 43 0D 0A 24 50	43 2F 58 54 0D 0A 24 41	\$PC\$PC/XT\$A
0000000020: 54 24 0D 0A 24 50 53 32	20 28 33 30 20 6D 6F 64	T\$PS2 (30 mod
0000000030: 65 6C 29 0D 0A 24 50 53	32 20 28 35 30 20 6F 72	el)\$PS2 (50 or
0000000040: 20 36 30 20 6D 6F 64 65	6C 29 0D 0A 24 50 53 32	60 model)\$PS2
0000000050: 20 28 38 30 20 6D 6F 64	65 6C 29 0D 0A 24 50 43	(80 model)\$PC
0000000060: 20 6A 72 0D 0A 24 50 43	20 43 6F 6E 76 65 72 74	jr\$PC Convert
0000000070: 69 62 6C 65 0D 0A 24 2E	24 0D 0A 59 6F 75 72 20	ible\$. \$Your
0000000080: 76 65 72 73 69 6F 6E 20	6E 75 6D 62 65 72 3A 20	version number:
0000000090: 24 0D 0A 59 6F 75 72 20	4F 45 4D 3A 20 24 0D 0A	\$Your OEM: \$
00000000A0: 59 6F 75 72 20 55 73 65	72 20 49 44 3A 20 24 B4	Your User ID: \$r
00000000B0: 09 CD 21 C3 33 C9 8B DA	33 D2 F7 F3 52 41 85 C0	oH!ГЗЙ<b3TчyRA..A
00000000C0: 75 F6 B4 02 5A 80 FA 09	76 03 80 C2 07 80 C2 30	uцrθZЪov▼БВ•ЪБ0
00000000D0: CD 21 E2 F0 C3 BA 03 01	E8 D4 FF B8 00 F0 8E C0	H!врГe▼иФяё рfA
00000000E0: B8 00 00 26 A0 FE FF 3C	FF 74 23 3C FE 74 25 3C	ё & юя<ят#<ют%<
00000000F0: FB 74 21 3C FC 74 23 3C	FA 74 25 3C FC 74 27 3C	ьт!<ьт#<ьт%<ьт'<
0000000100: F8 74 29 3C FD 74 2B 3C	F9 74 2D EB 31 90 BA 12	шт)<эт+<шт-л1ћє\$
0000000110: 01 EB 34 90 BA 17 01 EB	2E 90 BA 1F 01 EB 28 90	0л4ћє\$0л.ћє▼0л(ћ
0000000120: BA 25 01 EB 22 90 BA 36	01 EB 1C 90 BA 4D 01 EB	є%0л"ћє60лLћєM0л
0000000130: 16 90 BA 5E 01 EB 10 90	BA 66 01 EB 0A 90 BA 10	¬ћє^0л¬ћєf0л¬ћє>
0000000140: 00 E8 70 FF EB 04 90 E8	65 FF B4 30 CD 21 51 53	ирял♦ћиєяг0H!QS
0000000150: 50 BA 79 01 E8 58 FF B8	00 00 58 50 B4 00 BA 0A	Реу0иХяё XPr є
0000000160: 00 E8 50 FF BA 77 01 E8	45 FF 58 8A E5 B4 00 8A	иРяew0иЕяХьєг ъ
0000000170: C5 BA 0A 00 E8 3D FF BA	91 01 E8 32 FF 58 50 8A	Еє и=яє'0и2яXPљ
0000000180: E5 B4 00 8A C5 BA 0A 00	E8 29 FF BA 9E 01 E8 1E	єг ъЕє и)яєћ0и▲
0000000190: FF 58 B4 00 3C 00 74 06	BA 0A 00 E8 16 FF 58 BA	яХг < тєє и=яХє
00000001A0: 0A 00 E8 0F FF 32 C0 B4	4C CD 21	и юя2ArLH!

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

Плохой exe также содержит совмещенные машинные команды и данные, но они размещены с адреса 300h. На нулевом адресе располагается заголовок exe-файла.

0000000000:	4D 5A AB 00 03 00 00 00	20 00 00 00 FF FF 00 00
0000000010:	00 00 04 C5 00 01 00 00	1E 00 00 00 01 00 00 00
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

0000000300:	E9 D2 00 59 6F 75 72 20	50 43 20 74 79 70 65 3A	йт Your PC type:
0000000310:	20 24 50 43 0D 0A 24 50	43 2F 58 54 0D 0A 24 41	\$PC\PC/XT\PC\$A
0000000320:	54 24 0D 0A 24 50 53 32	20 28 33 30 20 6D 6F 64	T\PC\$PS2 (30 mod
0000000330:	65 6C 29 0D 0A 24 50 53	32 20 28 35 30 20 6F 72	el)\PC\$PS2 (50 or
0000000340:	20 36 30 20 6D 6F 64 65	6C 29 0D 0A 24 50 53 32	60 model)\PC\$PS2
0000000350:	20 28 38 30 20 6D 6F 64	65 6C 29 0D 0A 24 50 43	(80 model)\PC\$PC
0000000360:	20 6A 72 0D 0A 24 50 43	20 43 6F 6E 76 65 72 74	jr\PC\$PC Convert
0000000370:	69 62 6C 65 0D 0A 24 2E	24 0D 0A 59 6F 75 72 20	ible\PC\$. \$PCYour
0000000380:	76 65 72 73 69 6F 6E 20	6E 75 6D 62 65 72 3A 20	version number:
0000000390:	24 0D 0A 59 6F 75 72 20	4F 45 4D 3A 20 24 0D 0A	\$PCYour OEM: \$PC
00000003A0:	59 6F 75 72 20 55 73 65	72 20 49 44 3A 20 24 B4	Your User ID: \$r
00000003B0:	09 CD 21 C3 33 C9 8B DA	33 D2 F7 F3 52 41 85 C0	оН!ГЗЙ<бЗТчыRA...A
00000003C0:	75 F6 B4 02 5A 80 FA 09	76 03 80 C2 07 80 C2 30	уцгZъovъBъBъB
00000003D0:	CD 21 E2 F0 C3 BA 03 01	E8 D4 FF B8 00 F0 8E C0	H!врГ€▼иФяё ртA
00000003E0:	B8 00 00 26 A0 FE FF 3C	FF 74 23 3C FE 74 25 3C	ё & юя<ят#<ют%<
00000003F0:	FB 74 21 3C FC 74 23 3C	FA 74 25 3C FC 74 27 3C	ыт!<ьт#<ьт%<ьт'<
0000000400:	F8 74 29 3C FD 74 2B 3C	F9 74 2D EB 31 90 BA 12	шт)<эт+<шт-л1ћє\$
0000000410:	01 EB 34 90 BA 17 01 EB	2E 90 BA 1F 01 EB 28 90	0л4ћє\$0л.ћє▼0л(ћ
0000000420:	BA 25 01 EB 22 90 BA 36	01 EB 1C 90 BA 4D 01 EB	є%0л"ћє60лLћєM0л
0000000430:	16 90 BA 5E 01 EB 10 90	BA 66 01 EB 0A 90 BA 10	¬ћє^0л►ћєf0л0ћє►
0000000440:	00 E8 70 FF EB 04 90 E8	65 FF B4 30 CD 21 51 53	ирял♦ћиеяг0H!QS
0000000450:	50 BA 79 01 E8 58 FF B8	00 00 58 50 B4 00 BA 0A	Рєу0иХяё XPrг €

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

В хорошем exe разделены код и данные.

000000002D0:	59 6F 75 72 20 50 43 20	74 79 70 65 3A 20 24 50	Your PC type: \$P
000000002E0:	43 0D 0A 24 50 43 2F 58	54 0D 0A 24 41 54 24 0D	C:\\$PC/XT\\$AT\$
000000002F0:	0A 24 50 53 32 20 28 33	30 20 6D 6F 64 65 6C 29	\$PS2 (30 model)
00000000300:	0D 0A 24 50 53 32 20 28	35 30 20 6F 72 20 36 30	\$PS2 (50 or 60
00000000310:	20 6D 6F 64 65 6C 29 0D	0A 24 50 53 32 20 28 38	model)\$PS2 (8
00000000320:	30 20 6D 6F 64 65 6C 29	0D 0A 24 50 43 20 6A 72	0 model)\$PC jr
00000000330:	0D 0A 24 50 43 20 43 6F	6E 76 65 72 74 69 62 6C	\$PC Convertibl
00000000340:	65 0D 0A 24 2E 24 0D 0A	59 6F 75 72 20 76 65 72	e\$. \$Your ver
00000000350:	73 69 6F 6E 20 6E 75 6D	62 65 72 3A 20 24 0D 0A	sion number: \$
00000000360:	59 6F 75 72 20 4F 45 4D	3A 20 24 0D 0A 59 6F 75	Your OEM: \$You
00000000370:	72 20 55 73 65 72 20 49	44 3A 20 24 00 00 00 00	r User ID: \$
00000000380:	B4 09 CD 21 C3 33 C9 8B	DA 33 D2 F7 F3 52 41 85	oÍ!Ã3É<Ú3D÷óRA...
00000000390:	C0 75 F6 B4 02 5A 80 FA	09 76 03 80 C2 07 80 C2	Àuö`ΘZ€úov♥€Â•€Â
000000003A0:	30 CD 21 E2 F0 C3 1E 2B	C0 50 B8 0D 00 8E D8 BA	0Í!âðÃ▲+ÀP. Žð°
000000003B0:	00 00 E8 CB FF B8 00 F0	8E C0 B8 00 00 26 A0 FE	èËÿ. ðŽÀ. & p
000000003C0:	FF 3C FF 74 23 3C FE 74	25 3C FB 74 21 3C FC 74	ÿ<ÿt#<pt%<ût!<üt
000000003D0:	23 3C FA 74 25 3C FC 74	27 3C F8 74 29 3C FD 74	#<ût%<üt'<øt)<ÿt
000000003E0:	2B 3C F9 74 2D EB 31 90	BA 0F 00 EB 34 90 BA 14	+<ût-ë1®° ë4®°¶
000000003F0:	00 EB 2E 90 BA 1C 00 EB	28 90 BA 22 00 EB 22 90	ë.®°L ë(®°" ë"®
00000000400:	BA 33 00 EB 1C 90 BA 4A	00 EB 16 90 BA 5B 00 EB	°3 ëL®°J ë-®°[ë
00000000410:	10 90 BA 63 00 EB 0A 90	BA 10 00 E8 67 FF EB 04	►®°c ë®°®°» ègÿë♦
00000000420:	90 E8 5C FF B4 30 CD 21	51 53 50 BA 76 00 E8 4F	®è\ÿ`0Í!QSP°v è0
00000000430:	FF B8 00 00 58 50 B4 00	BA 0A 00 E8 47 FF BA 74	ÿ. XP´ °® èGÿ°t
00000000440:	00 E8 3C FF 58 8A E5 B4	00 8A C5 BA 0A 00 E8 34	è<ÿXŠâ´ ŠÅ°® è4
00000000450:	FF BA 8E 00 E8 29 FF 58	50 8A E5 B4 00 8A C5 BA	ÿ°Ž è)ÿXPŠâ´ ŠÅ°
00000000460:	0A 00 E8 20 FF BA 9B 00	E8 15 FF 58 B4 00 3C 00	® è ÿ°> èšÿX´ <
00000000470:	74 06 BA 0A 00 E8 0D FF	58 BA 0A 00 E8 06 FF CB	t▲°® èÿÿX°® è▲ÿË

Загрузка COM-модуля в основную память.

Алгоритм загрузки:

- Система выделяет свободный сегмент памяти и заносит его адрес во все сегментные регистры (CS, DS, ES и SS).
- В первые 256 байт этого сегмента записывается PSP.
- Непосредственно за ним загружается содержимое COM-файла без изменений.
- Указатель стека (регистр SP) устанавливается на конец сегмента.
- В стек записывается 0000h (адрес возврата для команды ret).
- Управление передаётся по адресу CS:0100h, где находится первый байт исполняемого файла.

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

Загрузка описана выше. Код располагается с адреса 100h.

2. Что располагается с адреса 0?

PSP.

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

Все они указывают на начало сегмента памяти, выделенного системой.

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

Указатель стека установлен на конец выделенного сегмента (На последний кратный 2, FFFE). Он может занимать память до адреса 0.

На скриншоте показано стартовое состояние регистров.

The screenshot displays the initial state of the 80486 registers and memory segments. The CPU is identified as 80486. The register window shows the following values:

Register	Value	Flag
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	

The memory window shows the following values:

Segment	Address	Value
cs	0100	E9D200
cs	0103	59
cs	0104	6F
cs	0105	7572
cs	0107	205043
cs	010A	207479
cs	010D	7065
cs	010F	3A20
cs	0111	2450
cs	0113	43
cs	0114	0D0A24
cs	0117	50
cs	0118	43
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

Загрузка «хорошего» EXE-модуля в основную память.

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

DS и ES устанавливаются на начало сегмента PSP, SS – на начало сегмента стека, CS – на начало сегмента команд. В IP загружается смещение точки входа в программу.

2. На что указывают регистры DS и ES?

На начало сегмента PSP.

3. Как определяется стек?

Его можно определить с помощью директивы .stack или вручную определить сегмент, выделить необходимое количество памяти, и связать данный сегмент с SS с помощью директивы ASSUME.

4. Как определяется точка входа?

Точка входа в программу определяется директивой END и меткой после нее.

На скриншоте показано стартовое состояние регистров для exe программы.

```

[CPU 80486]
cs:0026>1E      push  ds
cs:0027 2BC0      sub   ax,ax
cs:0029 50        push  ax
cs:002A BBFA48     mov   ax,48FA
cs:002D 8ED8       mov   ds,ax
cs:002F BA0000     mov   dx,0000
cs:0032 E8CBFF     call  0000
cs:0035 B800F0     mov   ax,F000
cs:0038 8EC0       mov   es,ax
cs:003A B80000     mov   ax,0000
cs:003D 26A0FEFF   mov   al,es:[FFFE]
cs:0041 3CFF       cmp   al,FF
cs:0043 7423       je    0068

ds:0000 CD 20 FF 9F 00 EA FF FF = Я Ъ
ds:0008 AD DE E4 01 C9 15 AE 01 н ф р о
ds:0010 C9 15 80 02 24 10 92 01 р а с т
ds:0018 01 01 01 00 02 FF FF FF   о о о

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 00C8  d=0
ds 48DD
es 48DD
ss 48ED
cs 4905
ip 0026

ss:00CA 0000
ss:00C8>0000
```

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.

