

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

Студентка гр. 8383

Ишанина Л.Н.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2018

Цель работы.

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

Необходимые сведения для составления программы.

Тип IBM PC хранится в байте по адресу 0F000:0FFFEh, 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:CH – 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 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.

Процедуры используемые в программе.

TETR_TO_HEX – процедура для перевода половины байта в шестнадцатеричную систему счисления.

BYTE_TO_HEX – процедура для перевода байта регистра AL в шестнадцатеричную систему счисления, помещая результат в AX.

WRD_TO_HEX – процедура для перевода двух байт регистра AX в шестнадцатеричную систему счисления, помещая результат в регистр DI.

BYTE_TO_DEC – процедура для перевода байта регистра AL в десятичную систему счисления, помещая результат в SI.

Ход работы.

1. Запуск «хорошего» .COM модуля.

```

C:\>exe2bin prog_c.exe prog_c.com

C:\>prog_c.com
Type your PC: PC
OS version: 05.00
OEM: 0
Serial number: 000000
C:\>

```

Рисунок 1 – «Хороший» .COM модуль

Запуск «плохого» .EXE модуля.

```

C:\>prog_c.exe

      0≡ Type your PC:

                                0≡ Type your PC:          5 0

                                                                0≡ Type your PC
:          0

0≡ Type your PC:          000000

      0≡ Type your PC:
C:\>_

```

Рисунок 2 – «Плохой» .EXE модуль

2. Запуск «хорошего» .EXE модуля.

```

C:\>prog_e.exe
Type your PC: PC
OS version: 05.00
OEM: 0
Serial number: 000000
C:\>_

```

Рисунок 3 – «Хороший» .EXE модуль

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

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

Один сегмент.

2) **EXE программа?**

EXE программа может содержать больше одного сегмента.

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

В тексте COM программы обязательно должны быть следующие директивы:

- ORG 100h

Данная директива необходима, так как она задает смещение для всех адресов программы на 256 байт для PSP.

- ASSUME

Эта директива ставит в соответствие начало программы сегментам кода и данных.

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

Нет, не все форматы команд можно использовать в COM-программе, так как COM-программа не располагает таблицей настроек(Relocation Table). Адреса сегментов определяются загрузчиком в момент запуска программы на основе информации о местоположении полей адресов в файле из этой таблицы. А значит, из-за отсутствия Relocation Table в COM-программах, такие команды как mov [регистр], seg [сегмент] невозможны.

4. COM модуль в шестнадцатеричном виде.

Рисунок 4 – .COM модуль в шестнадцатеричном виде

«Плохой» .EXE модуль в шестнадцатеричном виде.


```
view PROG.E.EXE - Far 3.0.5100 x86 Administrator
D:\4 сем\Лабы ОС\Мои\1 лаба\PROG E.EXE
00000000: 4D 5A D9 01 02 00 01 00 20 00 00 00 FF FF 00 00 MZU00 0  yy
00000010: 18 00 EC 35 57 00 0C 00 1E 00 00 00 01 00 5C 00 t 15W 9  ▲  0 \
00000020: 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
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
1Help 2 3Quit 4Dump 5 6Edit 7Search 8ANSI
```

```
view PROG.E.EXE - Far 3.0.5100 x86 Administrator
D:\4 сем\Лабы ОС\Мои\1 лаба\PROG E.EXE
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: 54 79 70 65 20 79 6F 75 72 20 50 43 3A 20 24 50 Type your PC: $P
000000230: 43 24 50 43 2F 58 54 24 41 54 24 50 53 32 20 6D C$PC/XT$AT$PS2 m
000000240: 6F 64 65 6C 20 33 30 24 50 53 32 20 6D 6F 64 65 odel 30$PS2 mode
000000250: 6C 20 38 30 24 50 43 6A 72 24 50 43 20 43 6F 6E l 80$PCjr$PC Con
000000260: 76 65 72 74 69 62 6C 65 24 0D 0A 4F 53 20 76 65 vertible$)$OS ve
000000270: 72 73 69 6F 6E 3A 20 30 20 2E 30 20 20 20 24 0D rsion: 0 .0 $)
000000280: 0A 4F 45 4D 3A 20 20 20 20 20 24 0D 0A 53 65 72 OEM: $)$Ser
000000290: 69 61 6C 20 6E 75 6D 62 65 72 3A 20 20 20 20 20 ial number:
0000002A0: 20 20 20 24 45 52 52 4F 52 20 45 52 52 4F 52 20 $ERROR ERROR
0000002B0: 45 52 52 4F 52 21 0D 0A 24 00 00 00 00 00 00 00 ERROR!)$
0000002C0: 24 0F 3C 09 76 02 04 07 04 30 C3 51 8A E0 E8 EF $<ov00000A0$aeI
0000002D0: FF 86 C4 81 04 D2 E8 E8 E6 FF 59 C3 53 8A FC E8 ytA+00eeyYAS$ue
0000002E0: E9 FF 88 25 4F 88 05 4F 8A C7 E8 DE FF 88 25 4F ey`%0 *0$CebY`%0
0000002F0: 88 05 58 C3 51 52 32 E4 33 D2 B9 0A 00 F7 F1 80 `+[AQ2B30+ +nE
000000300: CA 30 88 14 4E 33 D2 3D 0A 00 73 F1 3C 00 74 04 E0`9N30= sñ< t+
000000310: 0C 30 88 04 5A 59 C3 1E 2B C0 50 88 02 00 8E D8 00`ZYA+AP.0 Z0
000000320: B8 00 F0 8E C0 26 A0 FE FF BA 00 00 50 B4 09 CD . 0ZA& py° P'oi
000000330: 21 58 3C FF BA 0F 00 EB 47 90 3C FE BA 12 00 EB lX<y° eGk<p°t e
000000340: 3F 90 3C FB BA 12 00 EB 37 90 3C FC BA 18 00 EB ?k<ú°+ e7k<ú°t e
000000350: 2F 90 3C FA BA 18 00 EB 27 90 3C F8 BA 28 00 EB /k<ú°+ e'k<ú°( e
000000360: 1F 90 3C FD BA 35 00 EB 17 90 3C F9 BA 3A 00 EB vk<y°5 eik<ú°: e
000000370: 0F 90 BF 84 00 83 C7 24 E8 50 FF 89 05 BA 84 00 oú,, fC$ePy%+°,
000000380: 50 B4 09 CD 21 58 B4 30 CD 21 50 BE 49 00 83 C6 P'oiIX'0iIPXI fA
000000390: 0F E8 60 FF 83 C6 04 58 8A C4 E8 57 FF BA 49 00 eè`yFA+X$Aewy°I
0000003A0: 50 B4 09 CD 21 58 BE 5F 00 83 C6 08 8A C7 E8 43 P'oiIX% fA$Cec
0000003B0: FF BA 5F 00 50 B4 09 CD 21 58 BF 68 00 83 C7 16 y° P'oiIXk fC=
1Help 2 3Quit 4Dump 5 6Edit 7Search 8ANSI
```

```
0000003C0: 8B C1 E8 17 FF 8A C3 E8 01 FF 83 EF 02 89 05 BA 7Aeiy$Aeoyfio%+°
0000003D0: 6B 00 50 B4 09 CD 21 58 CB k P'oiIXE
1Help 2 3Quit 4Dump 5 6Edit 7Search 8ANSI
```

Рисунок 6 - «Хороший» .EXE модуль в шестнадцатеричном виде

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

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

Структура COM файла 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?

Структура “плохого” EXE файла содержит данные и код в одном сегменте. Код располагается с адреса 300h. С адреса 0 располагается Relocation Table.

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

Структура “хорошего” EXE файла содержит информацию для загрузчика, сегмент стека, сегмент данных и сегмент кода, а именно 3 сегмента вместо одного как в “плохом” EXE. Так же код располагается с адреса 200h в отличии от 300h в “плохом” EXE файле

6. Загрузка COM модуля в основную память.

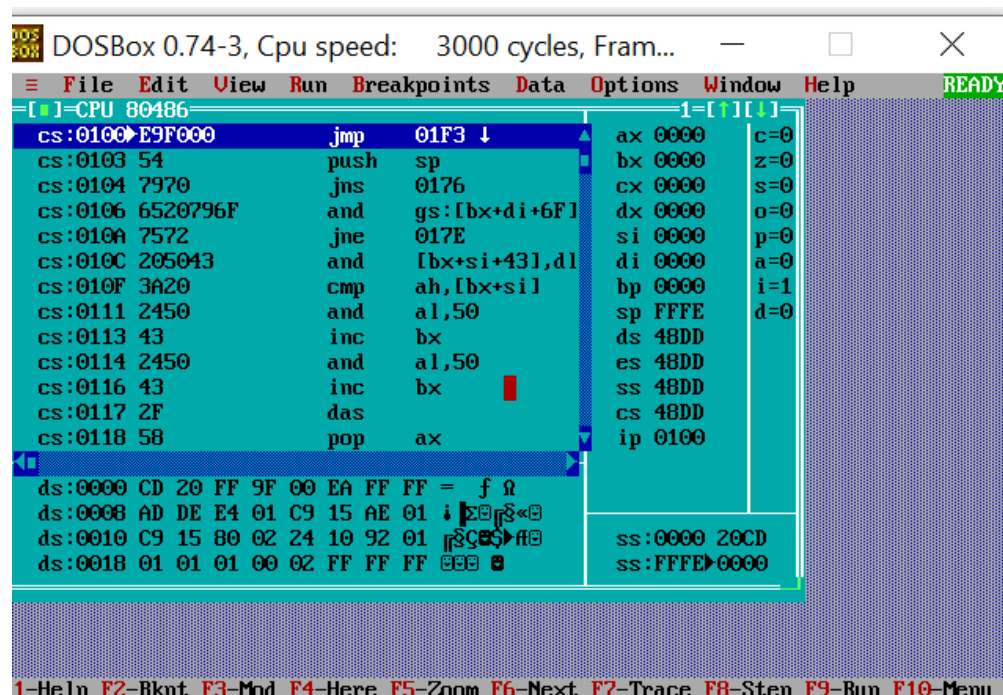


Рисунок 7 – Загрузка COM модуля в основную память

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

The screenshot shows the DOSBox 0.74-3 interface. At the top, the title bar reads "DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...". Below the title bar is a menu bar with options: File, Edit, View, Run, Breakpoints, Data, Options, Window, Help. The main window is divided into several panes. The left pane shows assembly code with the following instructions:

Address	Instruction	Comment
cs:0057	1E	push ds
cs:0058	2BC0	sub ax,ax
cs:005A	50	push ax
cs:005B	B8EF48	mov ax,48EF
cs:005E	8ED8	mov ds,ax
cs:0060	B800F0	mov ax,F000
cs:0063	8EC0	mov es,ax
cs:0065	26A0FEFF	mov al,es:[FFFE]
cs:0069	BA0000	mov dx,0000
cs:006C	50	push ax
cs:006D	B409	mov ah,09
cs:006F	CD21	int 21
cs:0071	58	pop ax

The right pane shows the state of the registers:

Register	Value
ax	0000
bx	0000
cx	0000
dx	0000
si	0000
di	0000
bp	0000
sp	0018
ds	48DD
es	48DD
ss	48ED
cs	48F9
ip	0057

At the bottom, there is a status bar with the following text: "F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu".

Рисунок 8 – Загрузка «хорошего» EXE модуля в память

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

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

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(48DD) устанавливаются на начало PSP, SS(48ED) - на начало стека, CS(4918) - на начало сегмента кода.

2) **На что указывают регистры DS и ES?**

Регистры DS и ES указывают на начало PSP.

3) **Как определяется стек?**

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?**

Точка входа определяется при помощи директивы END.

Вывод.

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.