

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

Студент гр. 9382

Сорокумов С.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Исследование различий в структурах исходных текстов модулей .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 0F000:0FFFE, 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:CH – 24-битовый серийный номер пользователя.

Задание.

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.

Функции управляющей программы:

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

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

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

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

TYPE_IBM_PC – Определяет тип IBM PC.

VERS_DOS – Определяет версию MS DOS.

OEM_DOS – Определяет серийный номер OEM.

USER_DOS – Определяет серийный номер пользователя.

PRINT – Вывод на экран.

Выполнение работы.

Шаг 1. Запуск «хорошего» .COM модуля.

```
C:\>LAB1_1.COM
Тип IMB PC
PC/XT
Номер версии MS DOS: 5.0
Серийный номер OEM: 255
Серийный номер пользователя: 000000
```

Рисунок 1 - «Хороший» .COM модуль

Запуск «плохого» .EXE модуля.

```
C:\>LAB1_1.EXE

щВѢТип IMB PC

IMB PC
000000 5 0 255
255 000000 щВѢТип IMB PC
щВѢТип IMB PC 000000 щВѢТип IMB PC
```

Рисунок 2 – «Плохой» .EXE модуль

Шаг 2. Запуск «хорошего» .EXE модуля.

```
C:\>LAB1_2.EXE
Тип IMB PC PC/XT
Номер версии MS DOS: 5.0
Серийный номер OEM: 255
Серийный номер пользователя: 000000
```

Рисунок 3 – «Хороший» .EXE модуль

Шаг 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?

Директива ORG 100h (смещение 100h), так как при загрузке COM-файла в память DOS занимает первые 256 байт (100h) блоком данных PSP и располагает код программы только после этого блока. Директива ASSUME, ставящая в соответствие начало программы сегментам кода и данных.

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

Нет, не все, так как в отличие от EXE-программы, в которой существует таблица настроек (таблица разметки), называемая Relocation Table, COM-программа ею не располагает. Адреса сегментов определяются загрузчиком в момент запуска программы на основе информации о местоположении полей адресов в файле из Relocation Table. Следовательно, в связи с отсутствием этой таблицы в COM-программах, команды вида mov [регистр], seg [сегмент] недопустимы.

Шаг 4.

.COM модуль в шестнадцатеричном виде.



Рисунок 4 - .COM модуль в шестнадцатеричном виде

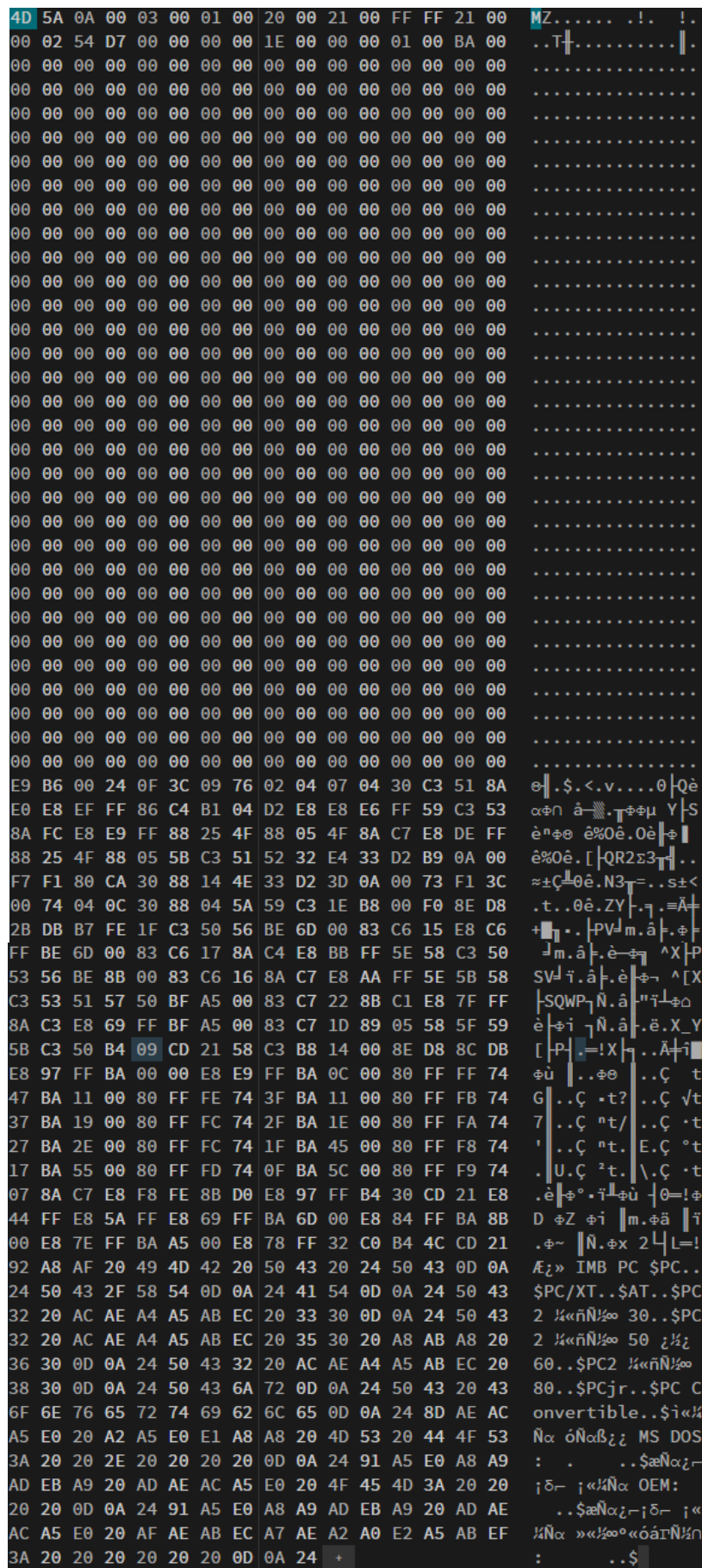
«Плохой» .EXE модуль в шестнадцатеричном виде.

[illegible]

00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
E9 82 01 92 A8 AF 20 49	4D 42 20 50 43 20 0D 0A	0é.Æ¿» IMB PC ..
24 50 43 0D 0A 24 50 43	2F 58 54 0D 0A 24 41 54	\$PC..\$PC/XT..\$AT
0D 0A 24 50 53 32 20 AC	AE A4 A5 AB EC 20 33 30	..\$PS2 ¼«ñÑ½° 30
0D 0A 24 50 53 32 20 AC	AE A4 A5 AB EC 20 35 30	..\$PS2 ¼«ñÑ½° 50
20 A8 AB A8 20 36 30 0D	0A 24 50 53 32 20 AC AE	¿½¿ 60..\$PS2 ¼«
A4 A5 AB EC 20 38 30 0D	0A 24 50 43 6A 72 0D 0A	ñÑ½° 80..\$PCjr..
24 50 43 20 43 6F 6E 76	65 72 74 69 62 6C 65 0D	\$PC Convertible.
0A 24 8D AE AC A5 E0 20	A2 A5 E0 E1 A8 A8 20 4D	.\$i«¼Ñα óÑαß¿¿ M
53 20 44 4F 53 3A 20 20	2E 20 20 20 20 0D 0A 24	S DOS: . ..\$
91 A5 E0 A8 A9 AD EB A9	20 AD AE AC A5 E0 20 4F	æÑα¿¿-¿δ- ¿¼Ñα 0
45 4D 3A 20 20 20 20 0D	0A 24 91 A5 E0 A8 A9 AD	EM: ..\$æÑα¿¿-¿
EB A9 20 AD AE AC A5 E0	20 AF AE AB EC A7 AE A2	δ- ¿¼Ñα »«½°«ó
A0 E2 A5 AB EF 3A 20 20	20 20 20 20 0D 0A 24 24	áTÑ½°: ..\$
0F 3C 09 76 02 04 07 04	30 C3 51 8A E0 E8 EF FF	.<.v....θ QèαφN
86 C4 B1 04 D2 E8 E8 E6	FF 59 C3 53 8A FC E8 E9	â-¿¿.¿φμ Y Sè"φθ
FF 88 25 4F 88 05 4F 8A	C7 E8 DE FF 88 25 4F 88	è%0è.0è φ è%0è
05 5B C3 51 52 32 E4 33	D2 B9 0A 00 F7 F1 80 CA	. QR2Σ3T ...≈±ÇL
30 88 14 4E 33 D2 3D 0A	00 73 F1 3C 00 74 04 0C	0è.N3¿=..s±<.t..
30 88 04 5A 59 C3 1E B8	00 F0 8E D8 2B DB B7 FE	0è.ZY ¿.¿.≡A + ¿¿.
1F C3 50 56 BE 72 01 83	C6 15 E8 C6 FF BE 72 01	. PV ¿.â ¿.φ=¿¿r.
83 C6 17 8A C4 E8 BB FF	5E 58 C3 50 53 56 BE 90	â ¿.è-φ¿ ^X PSV É
01 83 C6 16 8A C7 E8 AA	FF 5E 5B 58 C3 53 51 57	.â ¿.è φ¿ ^X SQW
50 BF AA 01 83 C7 22 8B	C1 E8 7F FF 8A C3 E8 69	P¿¿.â ¿"¿¿¿¿ è φ¿i
FF BF AA 01 83 C7 1D 89	05 58 5F 59 5B C3 50 B4	¿¿¿.â ¿.ë.X_Y ¿P
09 CD 21 58 C3 E8 9E FF	BA 03 01 E8 F0 FF BA 11	.=!X φP ...φ≡ .
01 80 FF FF 74 47 BA 16	01 80 FF FE 74 3F BA 16	.Ç tG ...Ç .t? .
01 80 FF FB 74 37 BA 1E	01 80 FF FC 74 2F BA 23	.Ç √t7 ...Ç "t/ #
01 80 FF FA 74 27 BA 33	01 80 FF FC 74 1F BA 4A	.Ç .t' 3.Ç "t. J
01 80 FF F8 74 17 BA 5A	01 80 FF FD 74 0F BA 61	.Ç °t. Z.Ç ²t. a
01 80 FF F9 74 07 8A C7	E8 FF FE 8B D0 E8 9E FF	.Ç .t.è φ .¿L φP
B4 30 CD 21 E8 4B FF E8	61 FF E8 70 FF BA 72 01	0=!φK φa φp r.
E8 8B FF BA 90 01 E8 85	FF BA AA 01 E8 7F FF 32	φ¿ É.φà ¿.φ¿ 2
C0 B4 4C CD 21 +		¿L =!

Рисунок 5 - «Плохой» .EXE модуль в шестнадцатеричном виде


```
4D 5A 0A 00 03 00 01 00 20 00 21 00 FF FF 21 00 MZ..... .!. !.
```



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

COM файл 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?

В «плохом» EXE файле данные и код содержатся в одном сегменте. Код располагается с адреса 300h. С адреса 0h располагается Relocation Table (таблица разметки).

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

В «хорошем» файле EXE содержится информация для загрузчика, сегмент стека, сегмент данных и сегмент кода (3 сегмента вместо одного в «плохом» .EXE). Код располагается с адреса 200h в отличии от 300h в «плохом» .EXE файле.

Шаг 5. «Загрузка COM модуля в основную память»

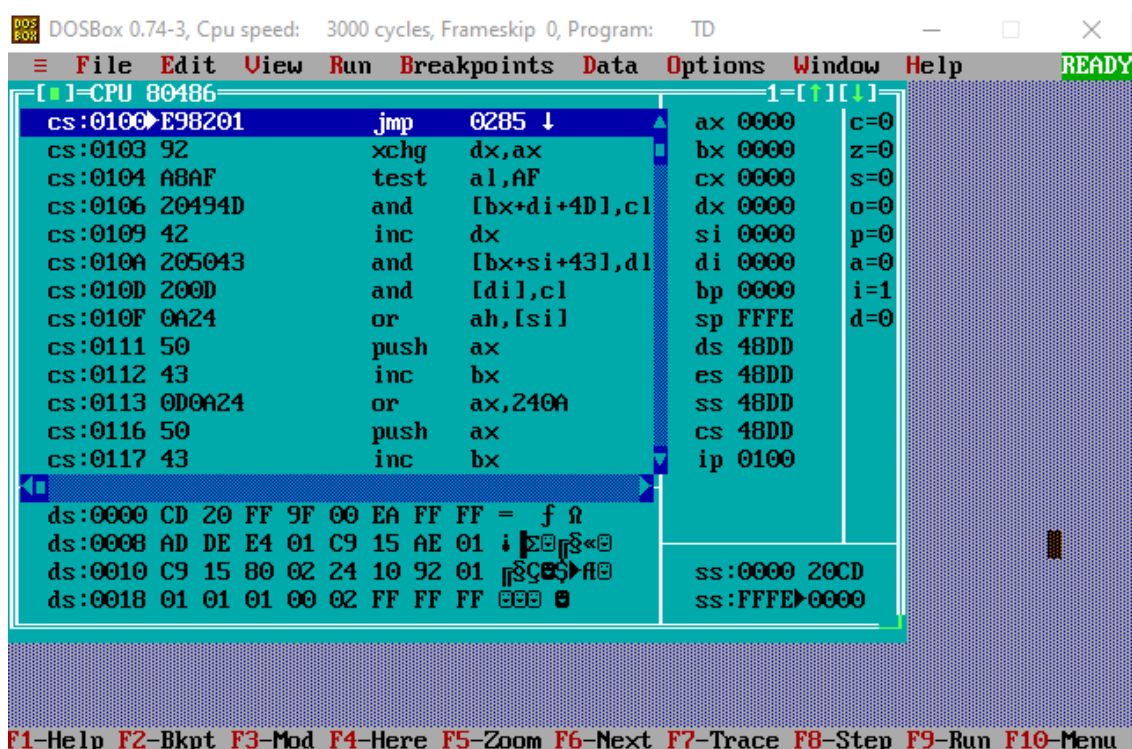


Рисунок 7 – Загрузка COM модуля в основную память

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

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

После загрузки COM-программы в память сегментные регистры указывают на начало PSP. Код располагается с адреса 100h (ip = 0100h).

2. Что располагается с 0 адреса?

Адрес начала PSP.

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

48DDh. Они указывают на начало PSP.

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

Стек определяется автоматически, указатель стека устанавливается на конец сегмента. Если для программы размер сегмента в 64КБ является достаточным, то DOS устанавливает в регистре SP адрес конца сегмента – FFFEH. Адреса расположены в диапазоне 0000h-FFFFh.

Шаг 6. Загрузка «хорошего» EXE модуля в память.

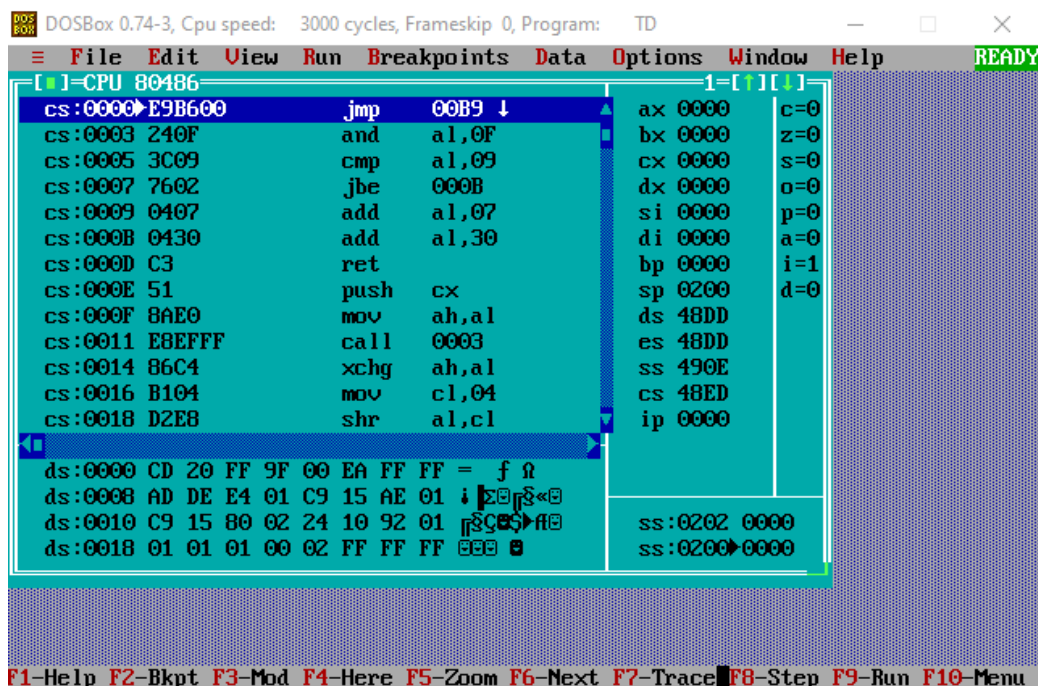


Рисунок 8 – Загрузка «хорошего» EXE модуля в память

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 устанавливаются на начало PSP, SS - на начало стека, CS - на начало сегмента кода.

2. На что указывают регистры DS и ES?

DS и ES указывают на начало PSP. После выполнения команд `mov ax, @data` и `mov ds, ax` регистре DS содержит адрес начала сегмента данных.

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?

При помощи команды `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.