

**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**

Студент гр. 8382

\_\_\_\_\_

Вербин К.М.

Преподаватель

\_\_\_\_\_

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.

### **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:

<b>PC</b>	<b>FF</b>
<b>PC/XT</b>	<b>FE,FB</b>
<b>AT</b>	<b>FC</b>
<b>PS2 модель 30</b>	<b>FA</b>
<b>PS2 модель 50 или 60</b>	<b>FC</b>
<b>PS2 модель 80</b>	<b>F8</b>
<b>PCjr</b>	<b>FD</b>
<b>PC Convertible</b>	<b>F9</b>

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 переводится 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 – 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.

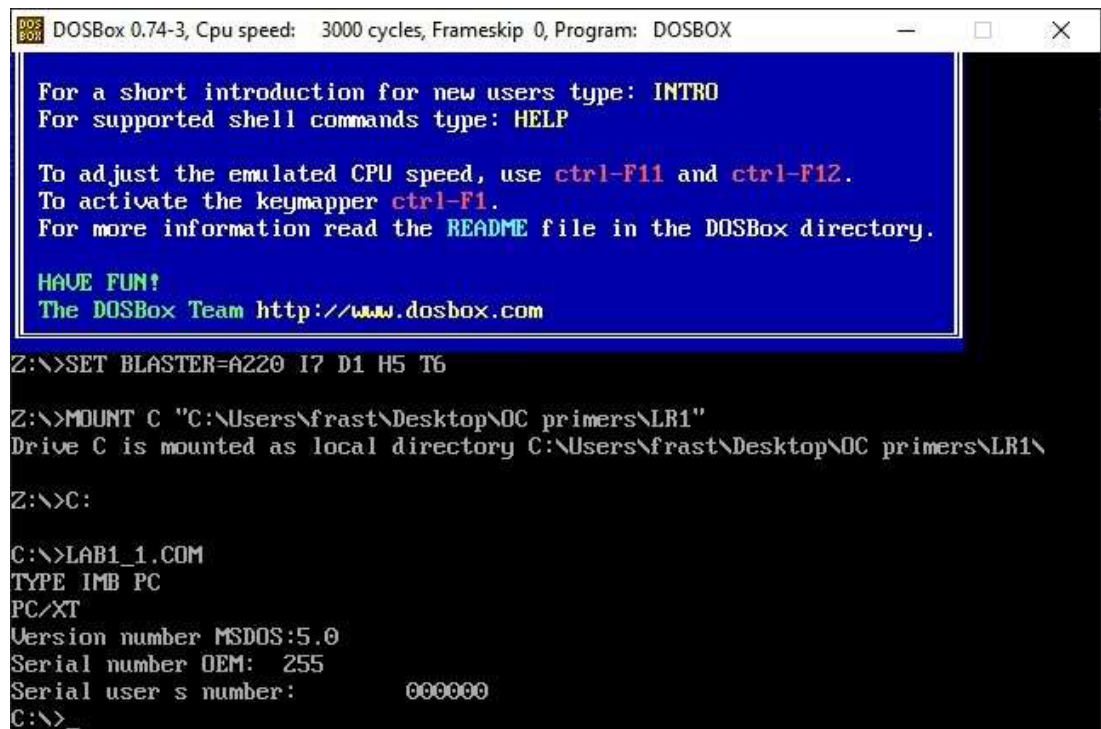
### **Структуры данных.**

Таблица 1 – Структуры данных

Название поля данных	Тип	Назначение
_type	db	Тип IBM PC
_PC	db	PC
_PC_XT	db	PC/XT
_AT	db	AT
_PS2_30	db	PS2 модель 30
_PS2_50_60	db	PS2 модель 50 или 60
_PS2_80	db	PS2 модель 80
_PCjr	db	PCjr
_PC_Conv	db	PC Convertible
_ver	db	Номер версии MS DOS
_oem	db	Серийный номер OEM
_user	db	Серийный номер пользователя

## Ход работы.

### Шаг 1. Запуск «хорошего» .COM модуля.



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

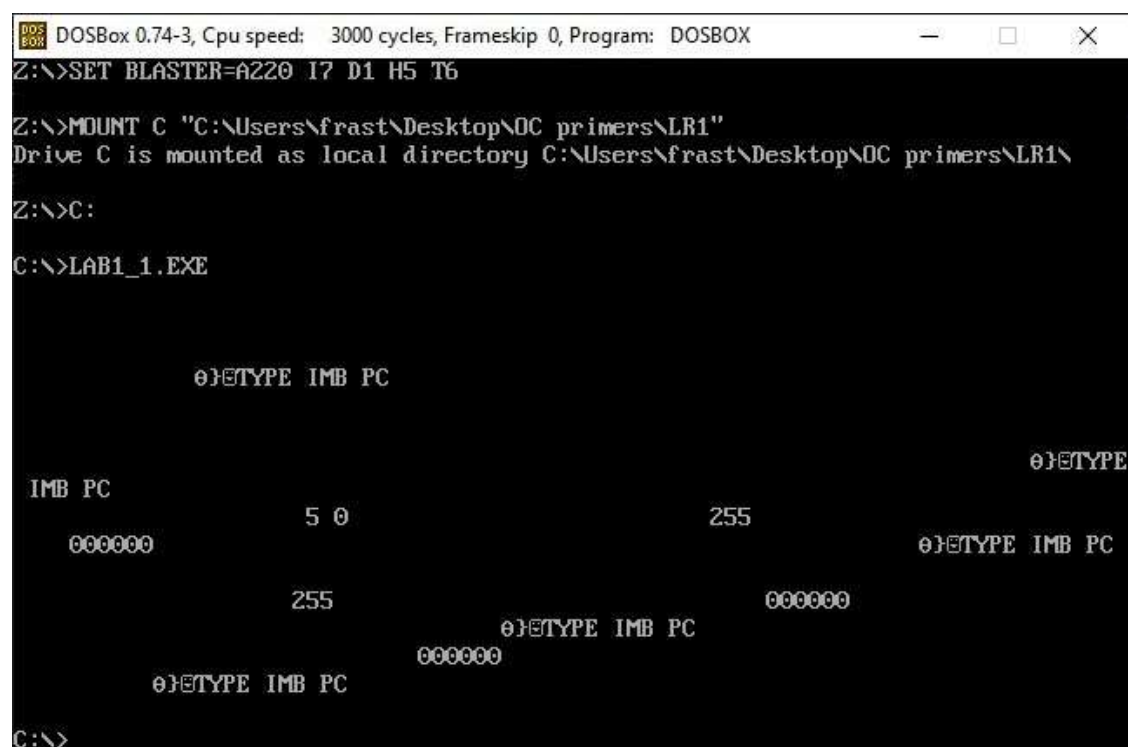
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>MOUNT C "C:\Users\frast\Desktop\OC primers\LR1"
Drive C is mounted as local directory C:\Users\frast\Desktop\OC primers\LR1\

Z:\>C:

C:\>LAB1_1.COM
TYPE IMB PC
PC/XT
Version number MSDOS:5.0
Serial number OEM: 255
Serial user s number: 000000
C:\>_
```

Рисунок 1 – «Хороший» .COM модуль



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>MOUNT C "C:\Users\frast\Desktop\OC primers\LR1"
Drive C is mounted as local directory C:\Users\frast\Desktop\OC primers\LR1\

Z:\>C:

C:\>LAB1_1.EXE

        0}E)TYPE IMB PC

IMB PC                                     0}E)TYPE

        5 0                               255
        000000                                0}E)TYPE IMB PC

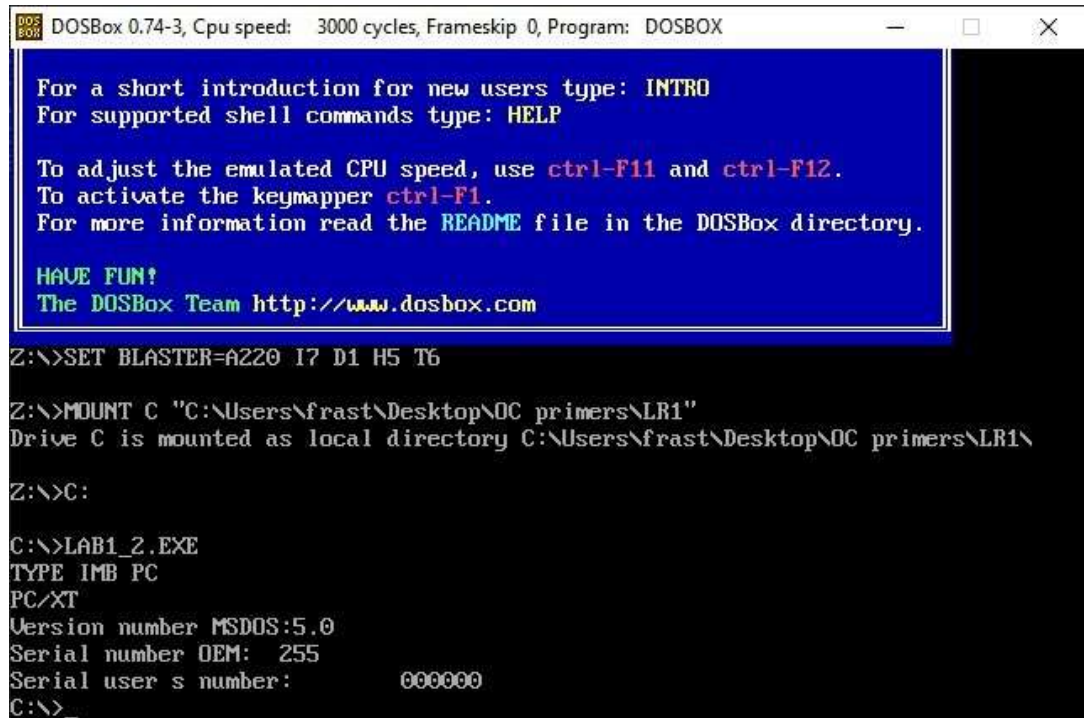
        255                                000000
        0}E)TYPE IMB PC
        000000
        0}E)TYPE IMB PC

C:\>
```

Запуск «плохого» .EXE модуля.

Рисунок 2 – «Плохой» .EXE модуль

## Шаг 2. Запуск «хорошего» .EXE модуля.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>MOUNT C "C:\Users\frast\Desktop\OC primers\LR1"
Drive C is mounted as local directory C:\Users\frast\Desktop\OC primers\LR1\

Z:\>C:

C:\>LAB1_2.EXE
TYPE IMB PC
PC/XT
Version number MSDOS:5.0
Serial number OEM: 255
Serial user s number: 000000
C:\>_
```

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.

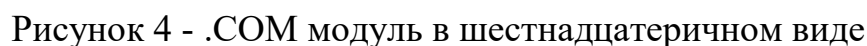
**3) Какие директивы должны обязательно быть в тексте COM программы?**

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-

#### Шаг 4. .COM модуль в шестнадцатеричном виде.





## «Плохой» .EXE модуль в шестнадцатеричном виде.

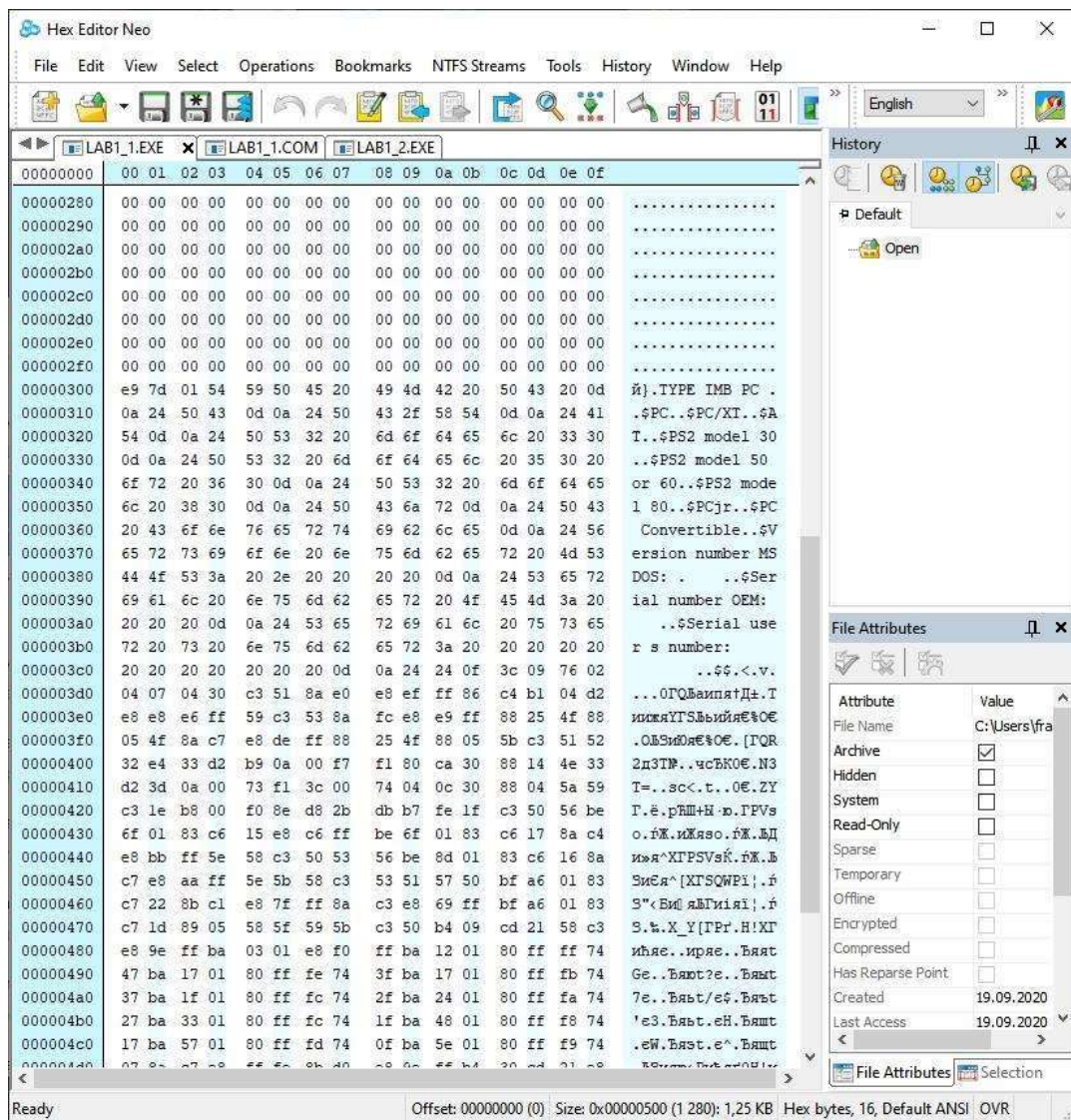


Рисунок 5 - «Плохой» .EXE модуль в шестнадцатеричном виде

«Хороший» .EXE модуль в шестнадцатеричном виде.



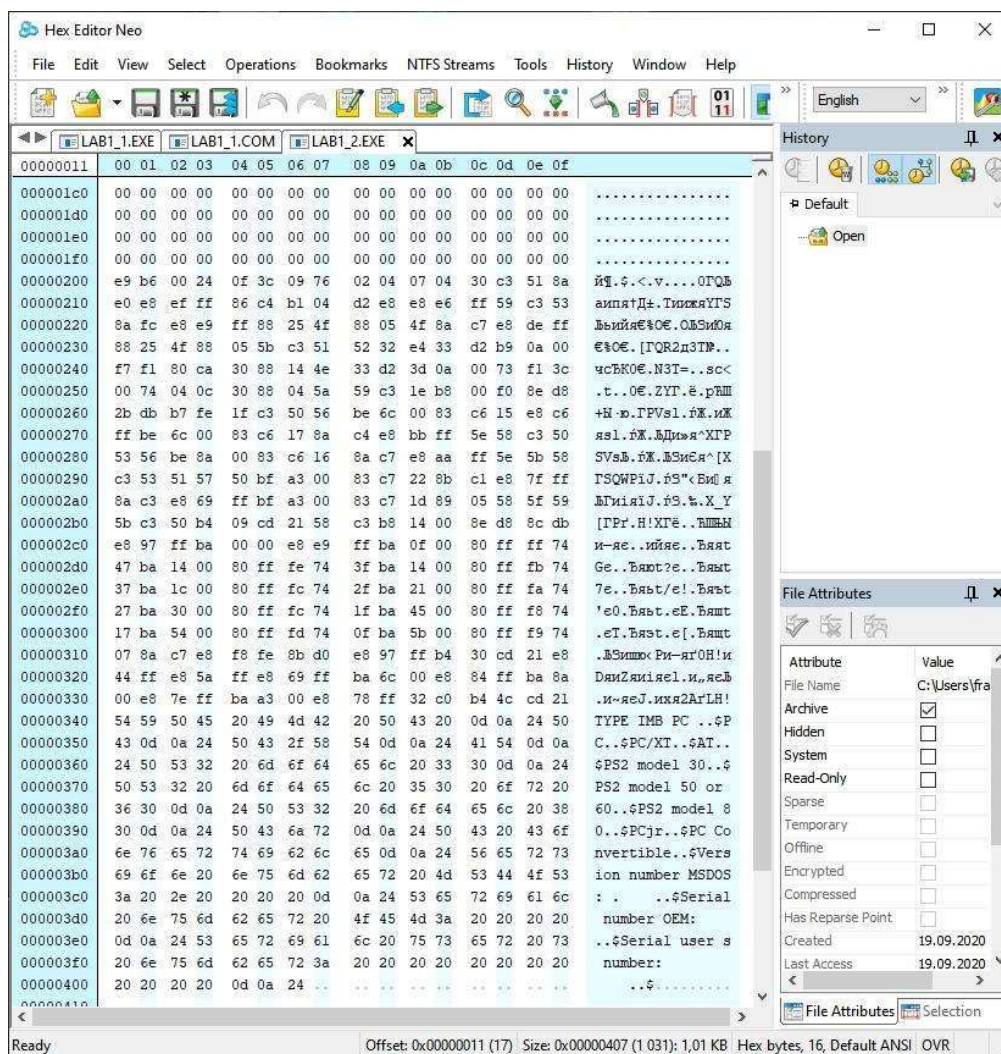


Рисунок 6 - «Хороший» .EXE модуль в шестнадцатеричном виде

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

**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 DOS header.

**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 (в плохом EXE директива ORG делает дополнительное смещение 100h).

**Шаг 5. Загрузка COM модуля в основную память.**



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFDPRO
AX 0000 SI 0000 CS 19F5 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0200 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0

CMD >

0100 E97D01 JMP 02B0
0103 54 PUSH SP
0104 59 POP CX
0105 50 PUSH AX
0106 45 INC BP
0107 20494D AND [BX+DI+4D],CL
010A 42 INC DX
010B 205043 AND [BX+SI+43],DL

DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
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 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = f.n= i | . + ...
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF .....ff. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 .....δ.L.
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 б.....J. ....
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 ? ↑ 8 ↓ 9 ← 10 ⇒
```

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?**

19F5h. 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.

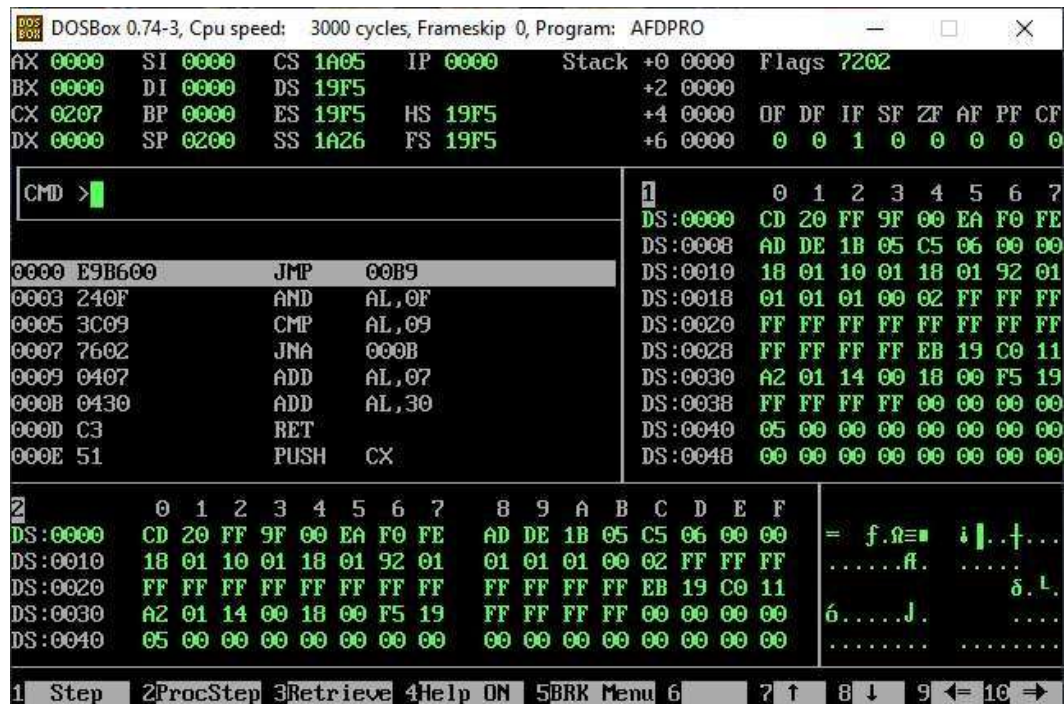


Рисунок 8 – Загрузка «хорошего» EXE модуля в память

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 директива `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.