

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»

Студентка гр. 8381

Бердникова А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы

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

Сведения о программе

Функции:

- TETR_TO_HEX — записывает код символа в шестнадцатеричной системе счисления в AL
- BYTE_TO_HEX — байт в AL переводится в два символа шестнадцатеричного числа в AX
- WRD_TO_HEX — перевод в шестнадцатеричную систему счисления 16-ти разрядного числа
- BYTE_TO_DEC — перевод в десятичную систему счисления

Ход выполнения работы

1) На основе шаблона, приведенного в методических указаниях, был написан текст исходного .COM модуля, который определяет тип РС и версию системы. Был получен и “хороший” .COM модуль и “плохой” .EXE модуль. Результаты работы программ представлен ниже.

Файл Lab1_C.asm

Полученный Lab1_C.com

```

C:\>tasm lab1_c.asm
Turbo Assembler Version 4.0 Copyright (c) 1988, 1993 Borland International

Assembling file:   lab1_c.asm
*Warning* lab1_c.asm(9) Reserved word used as symbol: STR
*Warning* lab1_c.asm(17) Reserved word used as symbol: ERR
Error messages:    None
Warning messages:  2
Passes:            1
Remaining memory:  466k

C:\>tlink /t lab1_c.obj
Turbo Link Version 4.01 Copyright (c) 1991 Borland International

C:\>lab1_c.com
Type PC: AT
OS version: 05.00
OEM: 0
Serial number: 000000
C:\>

```

Полученный Lab1_C.exe

```

B:\>masm lab1_c.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lab1_c.OBJ]: lab1_c
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

      49956 + 453209 Bytes symbol space free

      0 Warning Errors
      0 Severe Errors

B:\>link lab1_c.obj

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

Run File [LAB1_C.EXE]: lab1_c
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

B:\>lab1_c.exe

```

```

B:\>lab1_c.exe

      0*Type PC:

      0*Type PC:          5  0

      0*Type PC:          0          000000
      0*Type PC:
B:\>

```

- 2) Был написан текст программы, построен и отлажен исходный .EXE модуль, который выполняет те же функции, что и модуль .COM. Таким образом, был получен “хороший” .EXE модуль. Результат работы программы представлен ниже.

Файл Lab1_C.asm

Полученный Lab1_E.exe

```
Object filename [lab1_e.OBJ]: lab1_e
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

49956 + 455257 Bytes symbol space free

0 Warning Errors
0 Severe Errors

B:\>link lab1_e.obj

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

Run File [LAB1_E.EXE]: lab1_e
List File [NUL.MAP]:
Libraries [.LIB]:

B:\>lab1_e.exe
Type PC: AT
OS version: 05.00
OEM: 0
Serial number: 000000
B:\>
```

3) Загрузочные модули были рассмотрены в шестнадцатеричном виде:

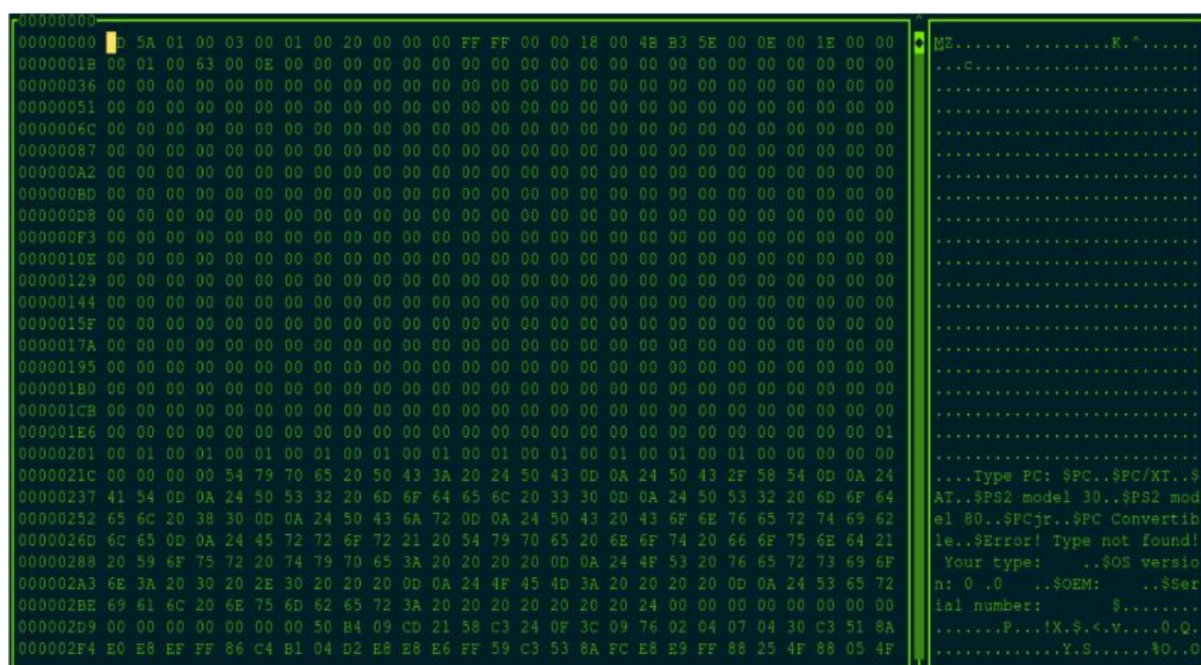


Рисунок 5 - Lab1_E.exe

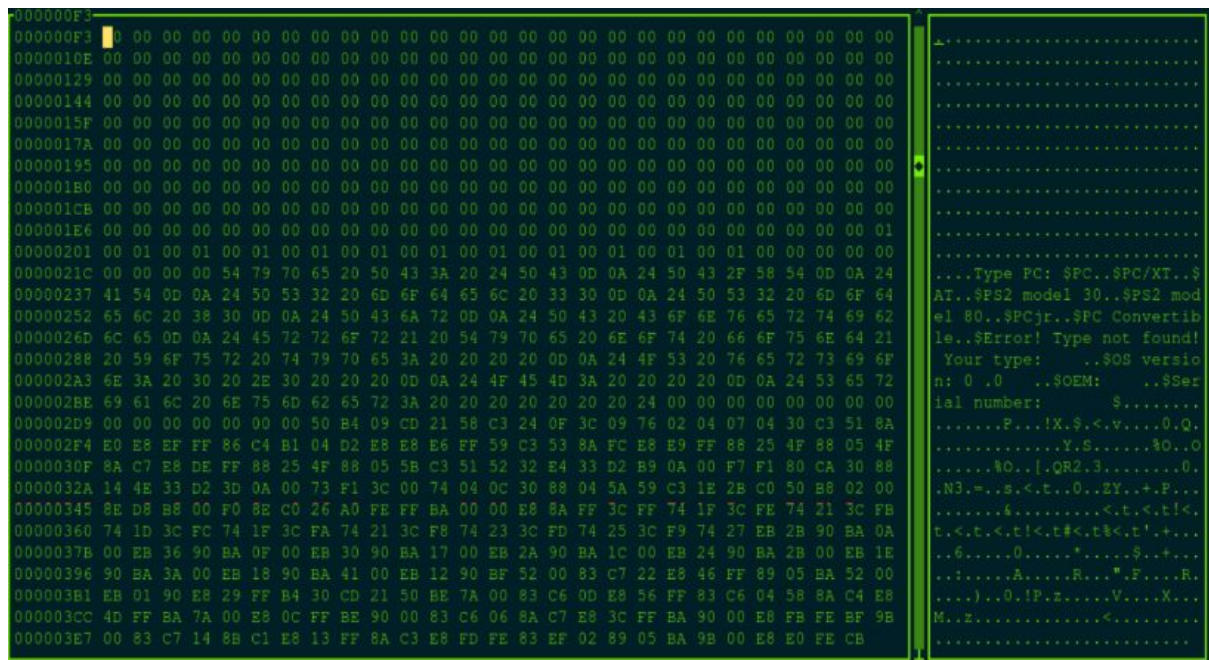


Рисунок 6 - Lab1_E.exe

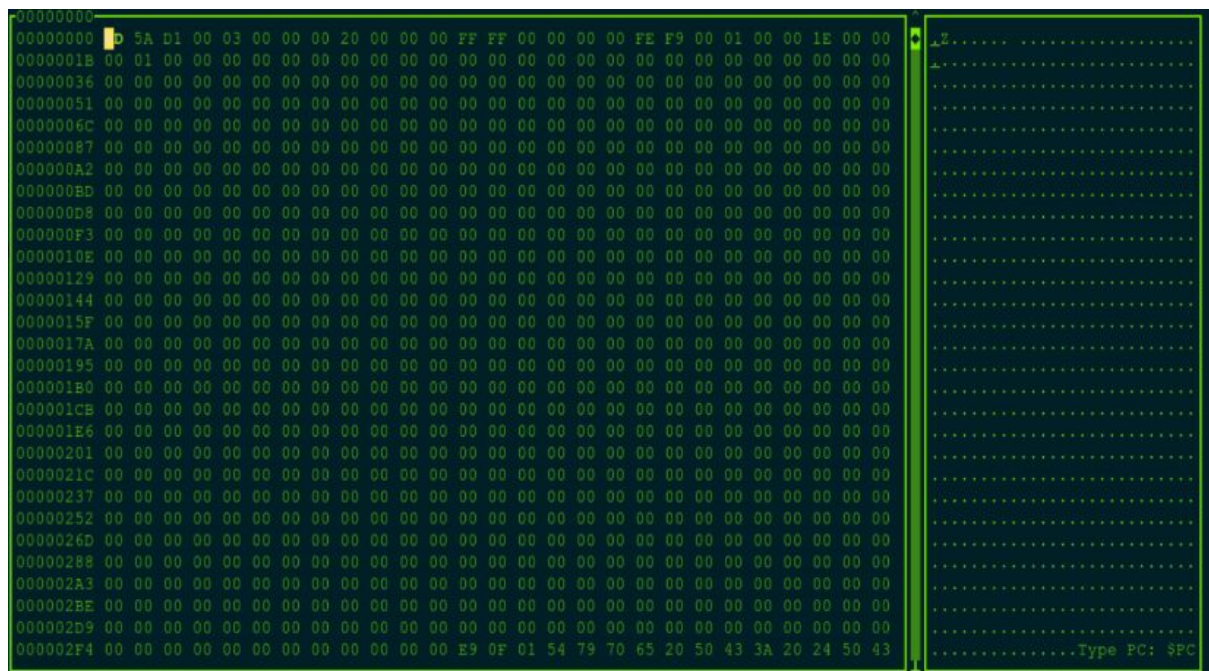


Рисунок 7 - Lab1_C.exe


```

[CPU 80486]
cs:005E>1E      push    ds
cs:005F 2BC0      sub     ax,ax
cs:0061 50        push    ax
cs:0062 B8F148     mov     ax,48F1
cs:0065 8ED8      mov     ds,ax
cs:0067 B800F0     mov     ax,F000
cs:006A 8EC0      mov     es,ax
cs:006C 26A0FEFF   mov     al,es:[FFFE]
cs:0070 BA0000     mov     dx,0000
cs:0073 E8BAFF     call    0000
cs:0076 3CFF      cmp     al,FF
cs:0078 741F      je      0099
cs:007A 3CFE      cmp     al,FE

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 0018  d=0
ds 48DF
es 48DF
ss 48EF
cs 48FD
ip 005E

ds:0000 CD 20 FF 9F 00 EA FF FF = Я %
ds:0008 AD DE E4 01 CA 15 AE 01 H 00 00 00
ds:0010 CA 15 80 02 25 10 93 01 00 00 00 00
ds:0018 01 01 01 00 02 FF FF FF 00 00 00 00

ss:001A 0000
ss:001B 0000

```

Рисунок 10 - Lab1_E.exe

```

[CPU 80486]
cs:0100>E90F01    jmp     0212 ↓
cs:0103 54        push    sp
cs:0104 7970      jns     0176
cs:0106 65205043  and     gs:[bx+si+43]
cs:010A 3A20      cmp     ah,[bx+si]
cs:010C 2450      and     al,50
cs:010E 43        inc     bx
cs:010F 0D0A24    or      ax,240A
cs:0112 50        push    ax
cs:0113 43        inc     bx
cs:0114 2F        das
cs:0115 58        pop     ax
cs:0116 54        push    sp

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 0000  d=0
ds 48DF
es 48DF
ss 48EE
cs 48EF
ip 0100

ds:0000 CD 20 FF 9F 00 EA FF FF = Я %
ds:0008 AD DE E4 01 CA 15 AE 01 H 00 00 00
ds:0010 CA 15 80 02 25 10 93 01 00 00 00 00
ds:0018 01 01 01 00 02 FF FF FF 00 00 00 00

ss:0002 6474
ss:0003 0000

```

Рисунок 11 - Lab1_C.exe

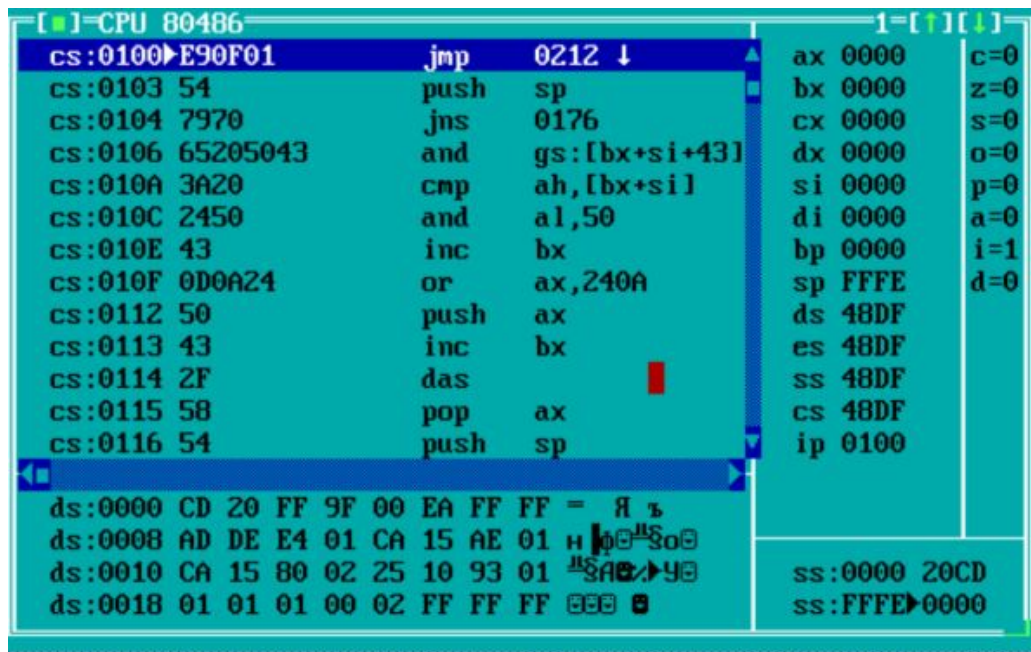


Рисунок 12 - Lab1_C.com

Ответы на контрольные вопросы:

Лабораторная работа №1

- ## 1. Отличия исходных текстов COM и EXE программ

- 1) Сколько сегментов должна содержать СОМ-программа?

Весь код, данные и стек такой программы располагаются в одном сегменте и не могут превышать 64 килобайта.

- ## 2) EXE-программа?

.EXE программа может содержать больше одного сегмента. В программах этого типа предусматривают отдельные сегменты для кода, данных и стека.

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

Директива ORG 100h, потому что при загрузке COM-файла в память DOS занимает первые 256 байт (100h) сегментом

данных PSP, а после него располагает код программы. Еще необходима директива ASSUME, с помощью директивы ASSUME ассемблеру сообщается информация о соответствии между сегментными регистрами, и программными сегментами.

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

Нельзя использовать команды, связанные с адресом сегмента, потому что адрес сегмента до загрузки неизвестен. В итоге загрузчик не сможет его определить. В модуле типа .com в DOS не содержится таблицы настройки, которая содержит описание адресов, которые зависят от размещения загрузочного модуля в ОП, поскольку подобные адреса в нём запрещены. Поэтому оно и не известно. Также нельзя использовать оператор FAR - переход на метку возможен только в результате межсегментной передачи управления, а так как в .com-файле только один сегмент, то никаких межсегментных переходов и быть не может.

2. Отличия форматов файлов COM и EXE модулей

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

.COM-файл состоит из команд, процедур и данных, используемых в программе. Код начинается с нулевого адреса.

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

Структура: управляющая информация для загрузчика(заголовок, таблица настройки адресов) и сегмент(код, данные).

Код располагается с 300h байта.

С адреса 0 располагается заголовок.

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

EXE-файл состоит из информации для загрузчика, сегмента стека, сегмент данных и сегмент кода. Отличается количеством сегментов (в «плохом» .EXE – 1 сегмент, в хорошем - 3), а также набором разрешённых команд.

Код «хорошего» EXE-файла начинается с адреса 200h, в отличие от «плохого».

Это из-за того, что в хорошем мы прописали, что сохраняем адрес данных в DS, а в плохом нет, потому что он предназначен для преобразования в com, а в коме мы не можем задать DS.

3. Загрузка COM модуля в основную память

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

- Выделяется свободный сегмент памяти, адрес которого заносится в сегментные регистры SS, CS, DS, ES
- Адреса от 0 до 100h занимают PSP
- С адреса 100h располагается содержимое COM файла
- Указатель стека (SP) устанавливается на конец сегмента
- Адрес возврата (0x0000) заносится в стек

- Управление передается по адресу CS:0100h

2) Что располагается с адреса 0?

С нулевого адреса располагается заголовок PSP.

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

Регистр SS – указывает на адрес начала сегмента стека;

Регистр DS – указывает на адрес начала сегмента данных;

Регистр CS – указывает на адрес начала сегмента кода;

При загрузке .COM модулей все сегментные регистры имеют значения 119C. Они указывают на PSP.

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

Какие адреса?

Стек создается автоматически, указатель стека в конце сегмента. Из этого следует, что он занимает оставшуюся память и адреса изменяются от больших к меньшим, то есть от FFFh к 0000h.

4. Загрузка «хорошего» EXE модуля в основную память

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

Сначала создается PSP. Затем определяется длина тела загрузочного модуля, определяется начальный сегмент.

Загрузочный модуль считывается в начальный сегмент, таблица настройки считывается в рабочую память, к полю каждого сегмента прибавляется сегментный адрес начального сегмента, определяются значения сегментных регистров. DS и

ES указывают на начало PSP (119C), CS – на начало сегмента команд (11F2h), а SS – на начало сегмента стека (11AC).

2) На что указывают регистры DS и ES?

Изначально регистры DS и ES указывают на начало сегмента PSP.

3) Как определяется стек?

Регистры SS и SP принимают значения, указанные в заголовке, потом к SS прибавляется сегментный адрес начального сегмента.

4) Как определяется точка входа?

Смещение точки входа в программу загружается в указатель команд IP. IP, а именно адрес, с которого начинается выполнение программы, определяется операндом директивы END, который называется точкой входа.