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

Преподаватель Ефремов І

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 16 с/с 16-ти разрядного числа.
- BYTE_TO_DEC — перевод в 10с/с.

Ход выполнения работы

1) На основе шаблона, приведенного в методических указаниях, был написан текст исходного .COM модуля, который определяет тип PC 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


           04@Type PC:


             04@Type PC:             5  0
                                                                      04
@Type PC:       0
                                    04@Type PC:              000000
                                                                04@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

D:\>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]:

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

3) Загрузочные модули были рассмотрены в шестнадцатеричном виде:

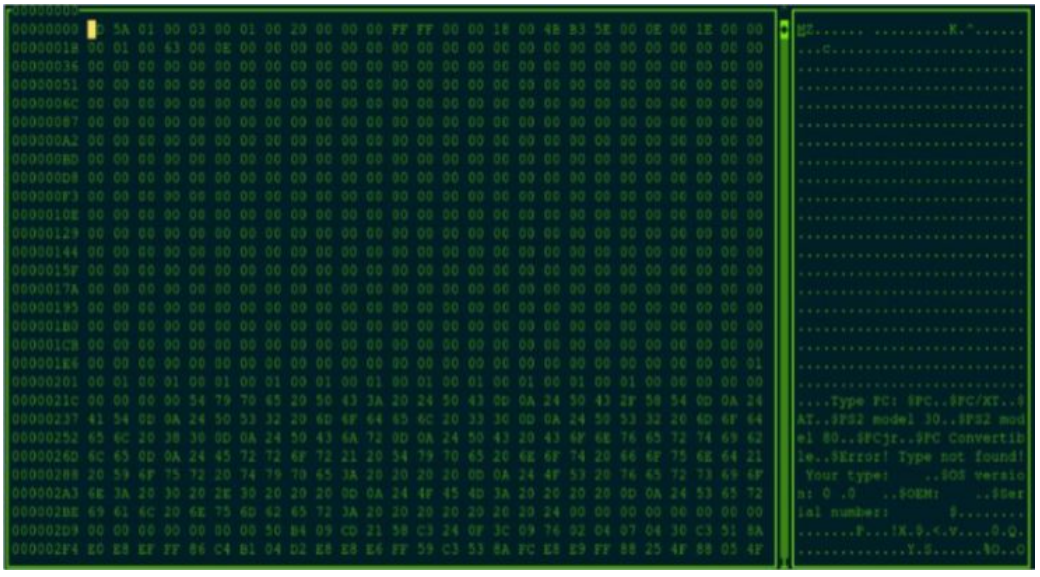

Рисунок 5 - Lab1_E.exe

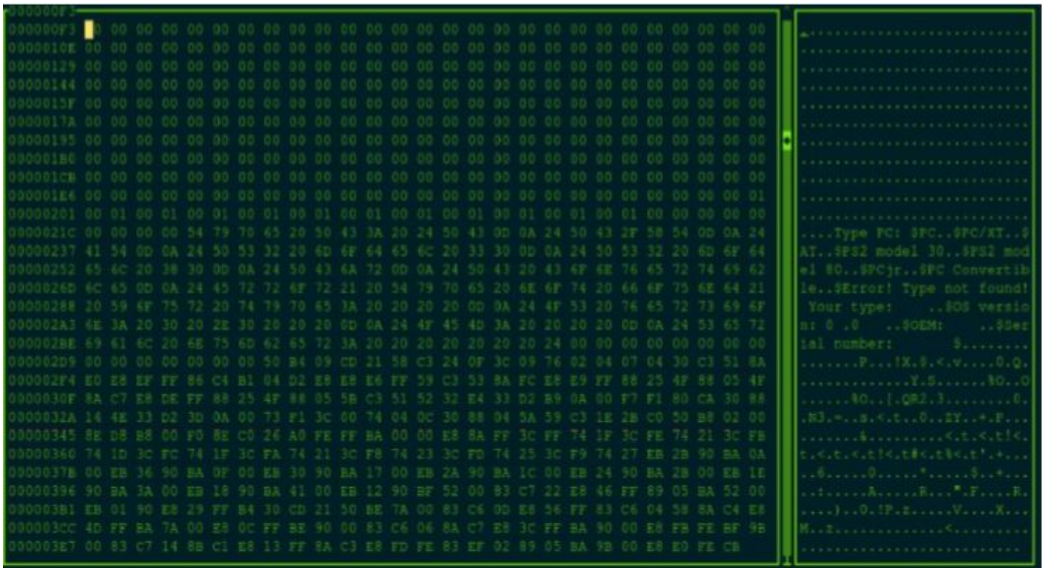

Рисунок 6 - Lab1_E.exe

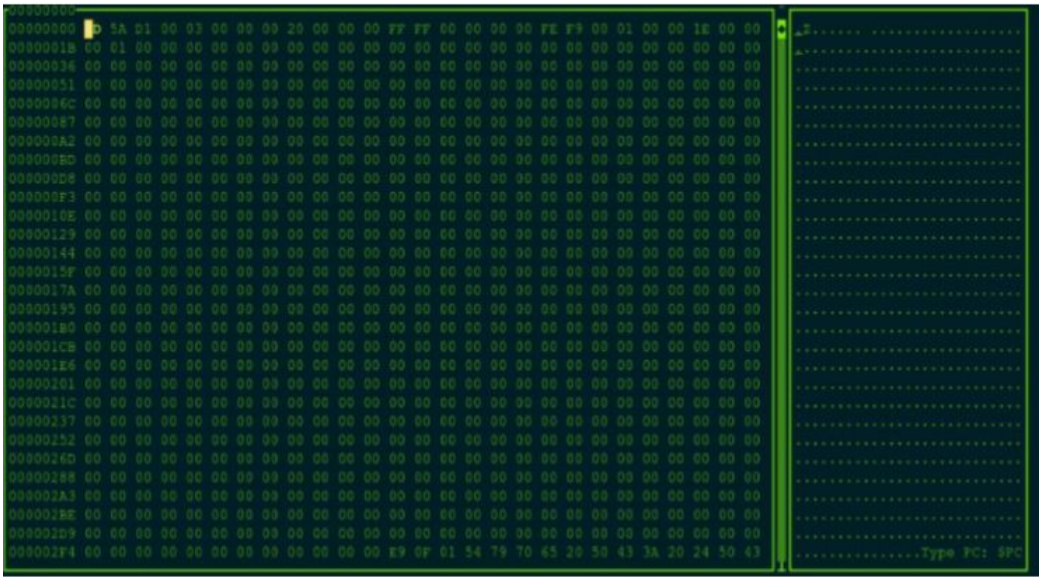

Рисунок 7 - Lab1_C.exe

Рисунок 8 - Lab1_C.exe

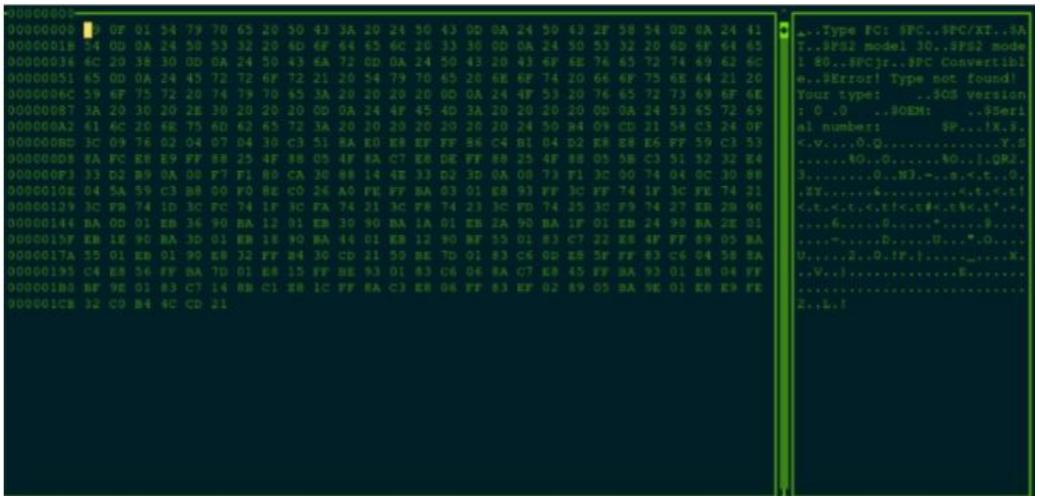

Рисунок 9 - Lab1_C.com

4) Файлы были открыты в отладчике TD.exe:

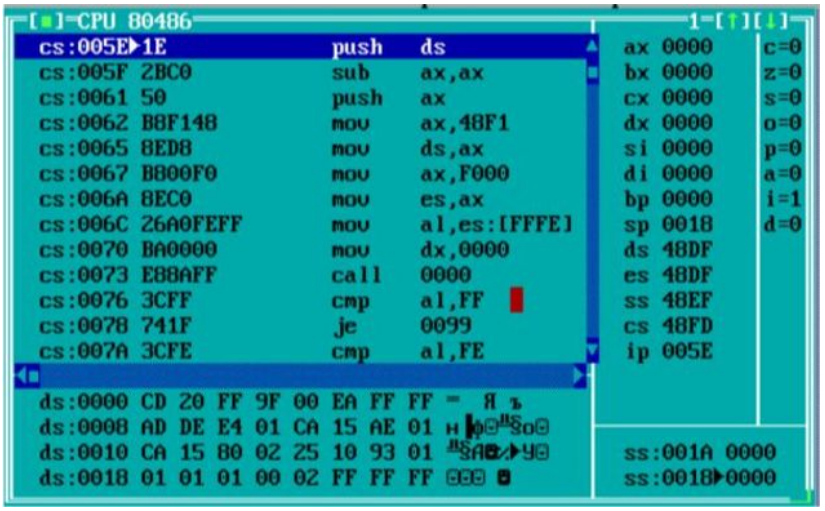

Рисунок 10 - Lab1_E.exe

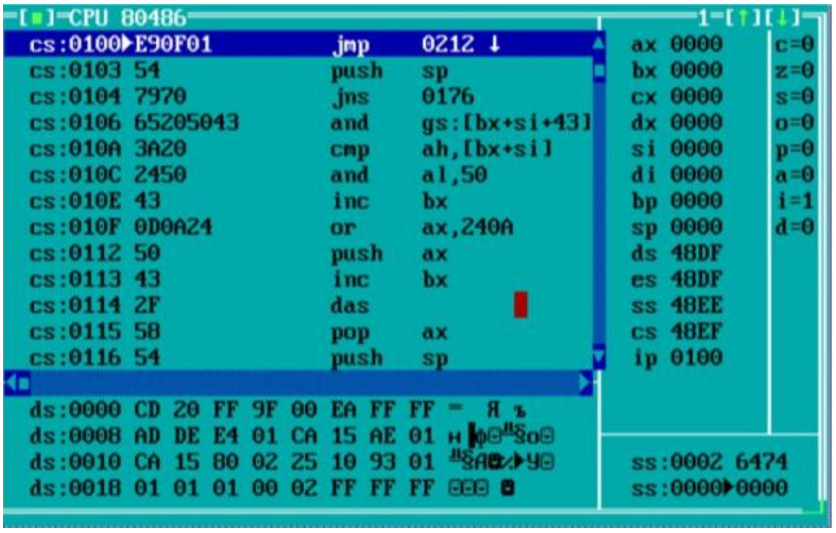

Рисунок 11 - Lab1_C.exe

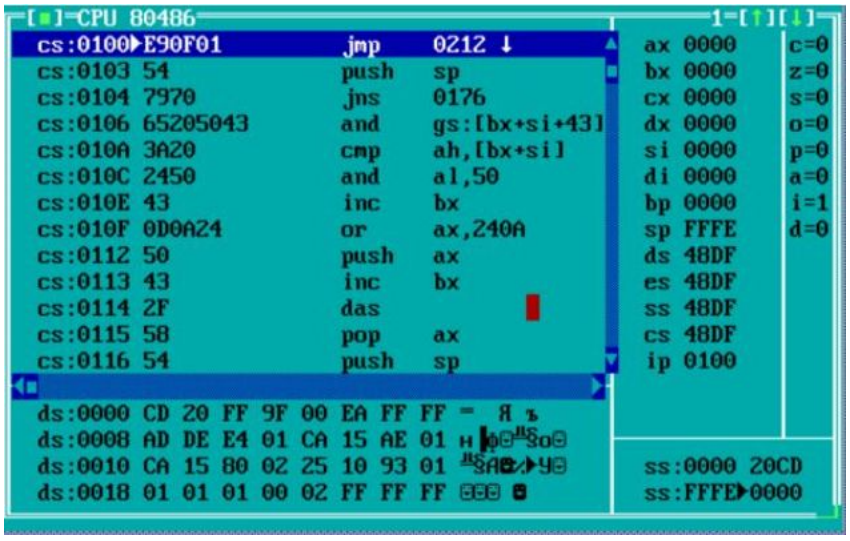

Рисунок 12 - Lab1_C.com

Ответы на контрольные вопросы:

Лабораторная работа №1

1. Отличия исходных текстов COM и EXE программ

1) Сколько сегментов должна содержать COM-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 неизвестно. 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. В «хорошем» EXE с нулевого адреса также располагается управляющая информация для загрузчика. Также перед кодом располагается сегмент стека. Так, при размере стека 200h код располагается с адреса 400h. Отличие от «плохого» EXE в том, что в «хорошем» не резервируется дополнительно 100h, которые в ком файле требовались для PSP.

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 занимаются 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 FFFEh 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.