

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.9383

Преподаватель

Поплавский И.

Ефремов М.А.

г. Санкт-Петербург

2021 г.

1. Постановка задачи

1.1. Цель работы

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

1.2. Задание

Шаг 1. Напишите текст исходного .COM модуля, который определяет тип РС и версию системы.

Шаг 2. Напишите текст исходного .EXE модуля, который выполняет те же функции, что и модуль в Шаге 1 и постройте и отладьте его. Таким образом, будет получен «хороший» .EXE.

Шаг 3. Сравните исходные тексты для .COM и .EXE модулей. Ответьте на контрольные вопросы «Отличия исходных текстов COM и EXE программ».

Шаг 4. Запустите FAR и откройте (F3/F4) файл загрузочного модуля .COM и файл «плохого» .EXE в шестнадцатеричном виде. Затем откройте (F3/F4) файл загрузочного модуля «хорошего» .EXE и сравните его с предыдущими файлами. Ответьте на контрольные вопросы «Отличия форматов файлов COM и EXE модулей».

Шаг 5. Откройте отладчик TD.EXE и загрузите .COM. Ответьте на контрольные вопросы «Загрузка COM модуля в основную память». Представьте в отчете план загрузки модуля .COM в основную память.

Шаг 6. Откройте отладчик TD.EXE и загрузите «хороший» .EXE. Ответьте на контрольные вопросы «Загрузка «хорошего» EXE модуля в основную память».

Шаг 7. Оформление отчета в соответствии с требованиями. В отчете необходимо привести скриншоты. Для файлов их вид в шестнадцатеричном виде, для загрузочных модулей – в отладчике. 2

1.3. Последовательность действий, выполняемых утилитой

Программа определяет и выводит на экран следующие значения в заданном порядке: тип PC, версия ОС, серийный номер OEM, серийный номер пользователя.

2. Ход работы

2.1. Был написан текст исходного .COM модуля, который определяет тип PC и версию системы, а так же серийный номер OEM и серийный номер пользователя. В результате выполнения был получен «хороший» .COM модуль.

```
C:\>LR1.COM
Type PC: AT
Version MS DOS: 5.0
OEM serial number: 255
User serial number: 000000
C:\>
```

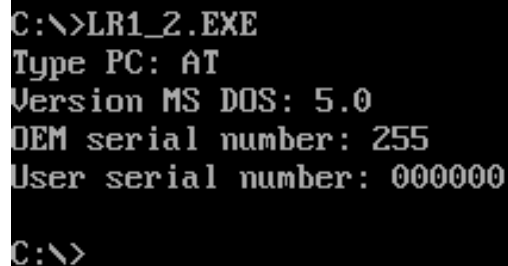
Рисунок 1 – «хороший».COM модуль

2.2. Был построен «плохой».EXE модуль, полученный из исходного текста для .COM модуля.

```
C:\>LR1.EXE
5 0
Type PC: PC
255
000000
Type PC: PC
Type PC: PC
C:\>
```

Рисунок 2 – «плохой».EXE модуль

2.3. Был написан текст исходного .EXE модуля, который выполняет те же функции. В результате был получен «хороший».EXE модуль.



```
C:\>LR1_2.EXE
Type PC: AT
Version MS DOS: 5.0
OEM serial number: 255
User serial number: 000000
C:\>
```

Рисунок 3 – «хороший».EXE модуль

3. Ответы на контрольные вопросы

3.1. Отличия исходных текстов COM и EXE программ

3.1.1. Сколько сегментов должна содержать COM-программа?

Ответ: COM-программа должна содержать один сегмент.

3.1.2. EXE-программа?

Ответ: EXE-программа может содержать более одного сегмента. В программе описываются три сегмента: команд, данных и стека.

3.1.3. Какие директивы должны обязательно быть в тексте COM-программы?

Ответ: В тексте COM-программы обязательно должна быть директива `ORG 100H`, которая резервирует 256 байт для PSP. Так же обязательно должна быть директива `ASSUME`, которая устанавливает соответствие сегментного регистра CS сегменту команд, а сегментного регистра DS – сегменту данных.

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

Ответ: В COM-программе нельзя использовать команды вида `mov c rvalue` в виде адресов сегментов и команды, содержащие дальнюю адресацию. Это связано с тем, что в COM-программе отсутствует таблица настроек (Relocation Table), с помощью которой в момент

Ответ: «плохой» .EXE модуль не разделен на сегменты. Данные и код содержатся в одном сегменте. Код «плохого» .EXE модуля располагается с адреса 300h. С адреса 0h располагается заголовок.

```
view LR1.EXE - Far 3.0.5700.0 x64 Administrator
C:\Users\didges\Desktop\spbetu_os_2021_9383\poplavskiy\lab1\LR1.EXE
00000000: 4D 5A 30 01 03 00 00 00 20 00 00 00 FF FF 00 00 MZ00▼ yy
00000010: 00 00 00 00 00 01 00 00 3E 00 00 00 01 00 FB 50  @ >  @ ūP
00000020: 6A 72 00 00 00 00 00 00 00 00 00 00 00 00 00 00 jr
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
000001E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000220: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000230: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000240: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1Help 2Text 3Quit 4Dump 5 6Edit 7Search 8ANSI
```

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

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

Ответ: «хороший» .EXE модуль, в отличие от «плохого» .EXE модуля содержит 3 отдельных сегмента – сегмент стека, сегмент данных, сегмент кода. Код «хорошего» .EXE модуля располагается с адреса 400h.

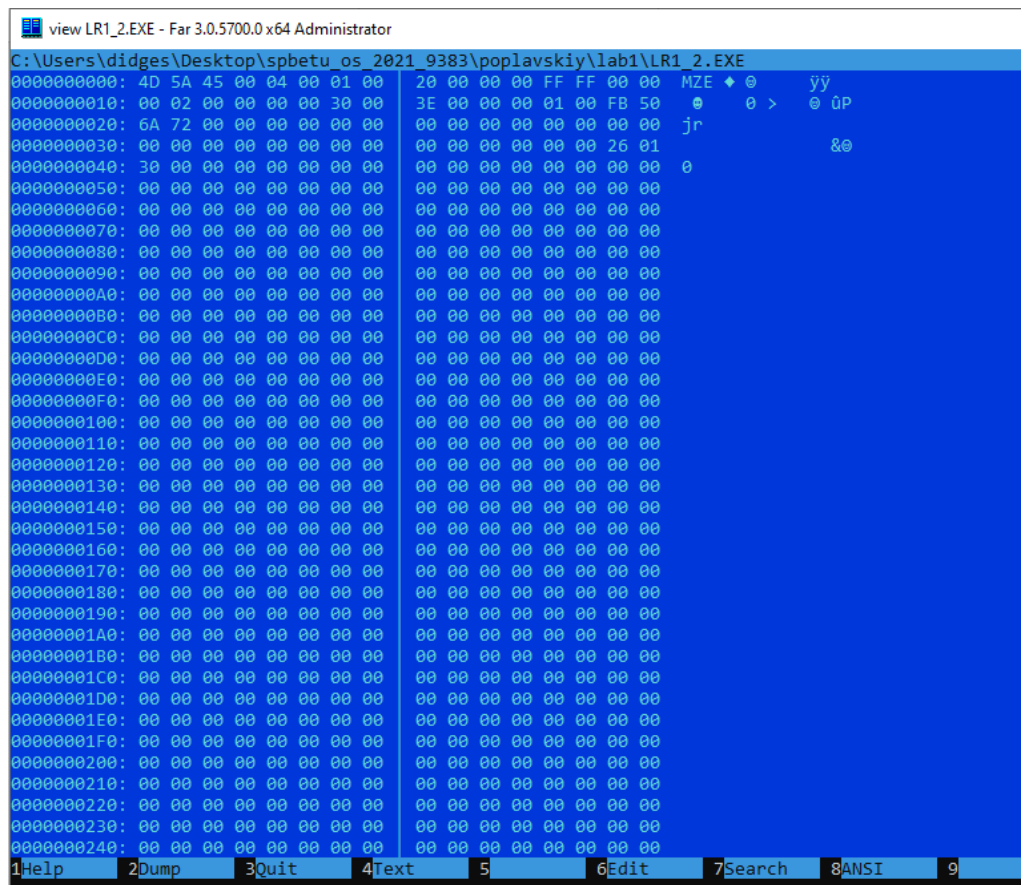


Рисунок 6 - «хороший» .EXE модуль в шестнадцатеричном виде.

3.3. Загрузка COM модуля в основную память

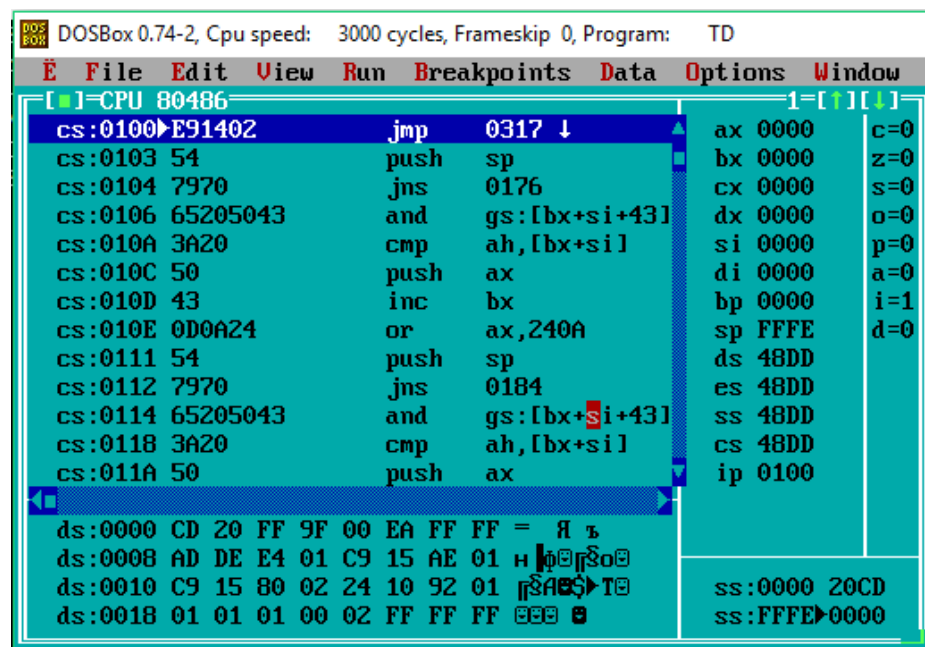


Рисунок 7 – отладчик TD.EXE для файла LR1.COM

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

Ответ: Формат загрузки модуля COM:

- 1)Выделение сегмента памяти для модуля.
- 2)Установка всех сегментных регистров на начало выделенного сегмента памяти.
- 3)Построение в первых 100h байтах памяти PSP.
- 4)Загрузка содержимого COM-файла и присваивание регистру IP значения 100h.
- 5)Регистр SP устанавливается в конец сегмента. Код начинается с адреса, содержащимся в CS, в нашем случае это 48DD.

3.3.2. Что располагается с адреса 0?

Ответ: С нулевого адреса располагается адрес начала PSP.

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

Ответ: Сегментные регистры DS, ES, SS, CS имеют значение 48DD. Они указывают на PSP.

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

Ответ: Стек генерируется автоматически. Стек занимает весь сегмент .COM программы. Сегментный регистр SS указывает на начало сегмента, а SP=FFFE на конец сегмента.

3.4. Загрузка «хорошего» EXE модуля в основную память

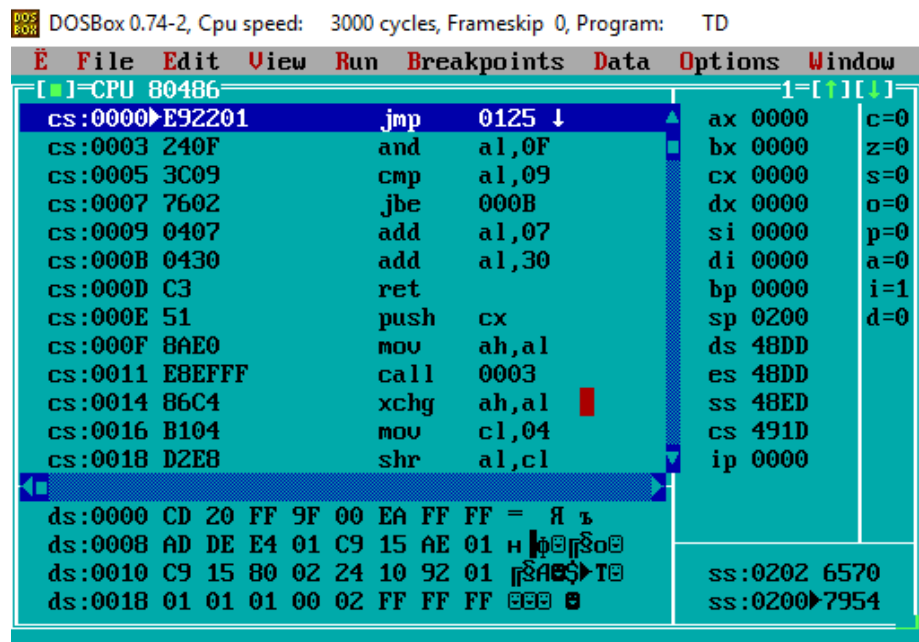


Рисунок 8 - TD.EXE для файла LR1_2.EXE

3.4.1. Как загружается «хороший» EXE? Какие значения имеют сегментные регистры?

Ответ: В процессе загрузки .EXE модуля в память система пристраивает к началу программы дополнительный сегмент PSP. Система, загрузив программу в память, инициализирует сегментные регистры DS и ES, CS, SS. В данном случае DS=ES=48DD, CS=491D, SS=48ED.

3.4.2. На что указывают регистры DS и ES?

Ответ: Сегментные регистры DS и ES указывают на начало PSP.

3.4.3. Как определяется стек?

Ответ: Стек определяется с помощью директивы STACK. В момент исполнения выделяется указанный блок памяти. В регистр SS записывается адрес начала сегмента. Регистр SP указывает на вершину стека.

3.4.4. Как определяется точка входа?

Ответ: Программа содержит директиву END. В качестве операнда этой директивы указывается точка входа в программу, т.е. адрес первой выполняемой строки. В данном случае это метка START.

4. Заключение

В результате выполнения лабораторной работы были исследованы различия в структурах исходных текстов .COM и .EXE модулей, структур файлов загрузочных модулей и способов их загрузки в основную память.