

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

**Кафедра Математического обеспечения электронно-вычислительных
машин**

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Операционные системы»
ТЕМА: ИССЛЕДОВАНИЕ СТРУКТУР ЗАГРУЗОЧНЫХ МОДУЛЕЙ.

Студентка гр. 0382

Рубежова Н.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить и исследовать различия в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

Задание.

1. Написать текст исходного .COM модуля, который определяет тип РС и версию системы. Результатом будет “хороший” .COM модуль, а также “плохой” .EXE, полученный из исходного текста для .COM модуля.

2. Написать текст исходного .EXE модуля, который выполняет те же функции, что и модуль в Шаге 1. Результатом будет “хороший” .EXE.

3. Сравнить исходные тексты для .COM и .EXE модулей. Ответить на контрольные вопросы “Отличия исходных текстов COM и EXE программ”.

4. Запустить FAR и открыть файл загрузочного модуля .COM и файл “плохого” .EXE в шестнадцатеричном виде. Затем открыть файл загрузочного модуля “хорошего” .EXE и сравнить его с предыдущими файлами. Ответить на контрольные вопросы “Отличия форматов файлов COM и EXE модулей”.

5. Открыть отладчик TD.EXE и загрузить .COM. Ответить на контрольные вопросы “Загрузка COM модуля в основную память”. Представить в отчете план загрузки .COM модуля в основную память.

6. Открыть отладчик TD.EXE и загрузить “хороший” .EXE. Ответить на контрольные вопросы “Загрузка “хорошего” EXE модуля в основную память”.

7. Оформить отчет в соответствии с требованиями. В отчете необходимо привести скриншоты. Для файлов их вид в шестнадцатеричном виде, для загрузочных модулей – в отладчике.

Ход выполнения.

1. Был написан исходный текст COM-модуля, который определяет тип РС и версию системы. Процедура PC_TYPE реализовывает определение типа РС, обращаясь к предпоследнему байту ROM BIOS и сравнивая коды по таблице, представленной в методическом пособии. В результате сравнения

выводится строка с названием модели. Процедура DOS_VER формирует текстовые строки с информацией о версии системы, серийным номером OEM и серийным номером пользователя, а затем полученные строки выводит на экран вызовом процедуры PRINT, которая отвечает за вывод строки на экран. В результате компиляции и сборки имеем «хороший» .COM модуль и «плохой» .EXE файл, полученный из исходного текста для .COM модуля. .EXE-модуль «плохой», поскольку имеются различия в структуре .COM и .EXE модулей. Например, в .COM-модуле код и данные должны быть определены в одном сегменте, в то время как в .EXE-модуле определяются отдельно сегмент данных, сегмент кода и сегмент стека.

Результаты выполнения см. на рисунках 1-2.

```
C:\>com_file.com
IBM PC TYPE: AT
MS DOS Version: 5.0
OEM number:255
User number: 000000h
```

Рисунок 1 – Запуск «хорошего» .COM модуля: com_file.com

```
C:\>com_file.exe
5 0
255 0
0.0IBM PC TYPE:
0.0IBM PC TYPE:
0.0IBM PC TYPE: PC
```

Рисунок 2 – Запуск «плохого» .EXE модуля: com_file.exe

2. Был построен «хороший» .EXE модуль, для этого определили сегменты данных, стека и кода, с помощью директивы ASSUME сопоставили сегменты и сегментные регистры выделили главную процедуру MAIN – точку старта выполнения программы.

```
C:\>exe_file.exe
IBM PC TYPE: AT
MS DOS Version: 5.0
OEM number:255
User number: 000000h
```

Рисунок 3 – Запуск «хорошего» .EXE модуля: exe_file.exe

3. Сравним исходные тексты для .COM и .EXE модулей:

Контрольные вопросы по лабораторной работе:

1) Сколько сегментов должна содержать COM-программа?

Ответ: 1 сегмент, в .COM модуле код и данные определяются в ОДНОМ сегменте, а стек генерируется автоматически.

2) EXE-программа?

Ответ: один или больше, код и данные должны быть разделены на отдельные сегменты. В EXE-модуле обязательно определяется сегмент кода. Также по необходимости определяются сегменты данных и стека.

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

Ответ: Директива `ORG 100h`, так как когда COM-программа начинает работать, все сегментные регистры содержат адрес префикса программного сегмента (PSP), 256-байтового блока, который резервируется операционной системой DOS непосредственно перед COM или EXE программой в памяти. Так как адресация начинается с шест. смещения 100 от начала PSP, то в программе кодируется директива `ORG 100H`. А с помощью директивы `ASSUME` делаем так, чтобы сегмент данных и кода указывали на общий сегмент.

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

Ответ: Т.к. в COM-программе все сегментные регистры определяются в момент запуска программы, а не в момент компиляции, то невозможно использование команд с указанием сегментов.

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

Ответ: .COM модуль состоит из одного общего сегмента, который содержит и код, и данные. Код располагается с адреса `0100h`, так как при загрузке COM-модуля директива `ORG 100H` (из-за блока PSP в 256 байт) устанавливает смещение `100h`.

Модуль в шестнадцатеричном виде см. на рис. 4.

Начало сегмента → 0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
 0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
 0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 CC 11
 0030 A2 01 14 00 18 00 F5 19 FF FF FF FF FF 00 00 00
 0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00
 0050 CD 21 CB 00 00 00 00 00 00 00 00 00 00 20 20 20
 0060 20 20 20 20 20 20 20 20 00 00 00 00 00 20 20 20
 0070 20 20 20 20 20 20 20 20 00 00 00 00 00 00 00
 0080 00 0D 00 00 00 00 00 00 00 00 00 00 00 00 00
 0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 Смещение на 100h → 0100 E9 2E 02 49 42 4D 20 50 43 20 54 59 50 45 3A 20 ...IBM PC TYPE:
 0110 0D 0A 24 49 42 4D 20 50 43 20 54 59 50 45 3A 20 ...\$IBM PC TYPE:
 0120 50 43 0D 0A 24 49 42 4D 20 50 43 20 54 59 50 45 PC..\$IBM PC TYPE
 0130 3A 20 50 43 2F 58 54 0D 0A 24 49 42 4D 20 50 43 : PC/XT..\$IBM PC
 0140 20 54 59 50 45 3A 20 41 54 0D 0A 24 49 42 4D 20 TYPE: AT..\$IBM
 0150 50 43 20 54 59 50 45 3A 20 50 53 32 20 6D 6F 64 PC TYPE: PS2 mod
 0160 65 6C 20 33 30 0D 0A 24 49 42 4D 20 50 43 20 54 el 30..\$IBM PC T
 0170 59 50 45 3A 20 50 53 32 20 6D 6F 64 65 6C 20 35 YPE: PS2 model 5
 0180 30 20 6F 72 20 36 30 0D 0A 24 49 42 4D 20 50 43 0 or 60..\$IBM PC
 0190 20 54 59 50 45 3A 20 50 53 32 20 6D 6F 64 65 6C TYPE: PS2 model
 01A0 20 38 30 0D 0A 24 49 42 4D 20 50 43 20 54 59 50 80..\$IBM PC TYP
 01B0 45 3A 20 50 43 6A 72 0D 0A 24 49 42 4D 20 50 43 E: PCjr..\$IBM PC
 01C0 20 54 59 50 45 3A 20 50 43 20 43 6F 6E 76 65 72 TYPE: PC Conver
 01D0 74 69 62 6C 65 0D 0A 24 4D 53 20 44 4F 53 20 56 tible..\$MS DOS V
 01E0 65 72 73 69 6F 6E 3A 20 20 2E 20 20 0D 0A 24 4F ersion: . ..\$O
 01F0 45 4D 20 6E 75 6D 62 65 72 3A 20 20 20 0D 0A 24 EM number: ..\$
 0200 55 73 65 72 20 6E 75 6D 62 65 72 3A 20 20 20 20 User number:
 0210 20 20 20 68 0D 0A 24 4F 3C 09 76 02 04 07 04 h..\$.<.v....
 0220 30 C3 51 8A E0 E8 EF FF 86 C4 B1 04 D2 E8 E8 E6 0.Q.....

Рисунок 4 – 16-ый вид «хорошего» .COM модуля: com_file.com

6) Какова структура файла “плохого” EXE? С какого адреса располагается код? Что располагается с адреса 0?

Ответ: ...

Модуль в шестнадцатеричном виде представлен на рисунке 5.

Рисунок 5 – “плохой” .EXE в 16-м виде

7) Какова структура файла “хорошего” EXE? Чем он отличается от файла “плохого” EXE?

Ответ: ...

Модуль в шестнадцатеричном виде представлен на рисунке 6.

Рисунок 6 – “хороший” .EXE в 16-м виде

Результат загрузки программы в отладчик представлены на рисунке 7.

Рисунок 7 – отладка “хорошего” .COM модуля

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

Ответ:

9) Что располагается с адреса 0?

Ответ:

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

Ответ:

11) Как определяется стек? Какую область памяти он занимает? Какие адреса?

Ответ:

Результат загрузки программы в отладчик представлены на рисунке 8.

Рисунок 8 – загруженный в отладчик “хороший” .COM модуль

12) Как загружается “хороший” EXE? Какие значения имеют сегментные регистры?

Ответ:

13) На что указывают регистры DS и ES?

Ответ: на начало PSP

14) Как определяется стек?

Ответ:

15) Как определяется точка входа?

Ответ:

Исходный код программы см. в приложении А.

Выводы.

Были изучены и исследованы различия в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.