

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

Студентка гр. 8381

Гречко В.Д.

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:

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

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 переводиться 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.

Ход работы.

Шаг 1. Написан текст исходного .COM модуля, определяющего тип PC и версию системы. За основу взят шаблон, приведенный в разделе “Основные сведения”. Ассемблерная программа читает содержимое предпоследнего байта ROM BIOS по таблице, сравнивая коды, определяет тип PC и выводит строку с названием модели.

Если код не совпадает ни с одним значением, то двоичный код переводится в символьную строку, содержащую запись шестнадцатеричного числа и выводится на экран в виде соответствующего сообщения.

Затем определяется версия системы. Ассемблерная программа по значениям регистров AL и AH формирует текстовую строку в формате xx.yy, где xx – номер основной версии, а yy – номер модификации в десятичной системе счисления, формирует строки с серийным номером OEM и серийным номером пользователя. Полученные строки выводятся на экран.

Отлажен полученный исходный модуль.

Результатом выполнения этого шага стал «хороший» .COM модуль, а также был построен «плохой» .EXE, полученный из исходного текста для .COM модуля.

```
C:\>new_com.com
Type of PC: FC
Version of System: 5.0
Serial number of OEM: 255
Serial number of user: 000000
```

Рисунок 1 – «Хороший» .COM модуль

```
C:\>new_com.exe
FC                    5 0                    255
000000
0*EType of PC:
5 0                    255                    0000
00
Type of PC:
255                    000000
000000
0*EType of PC:
```

Рисунок 2 – «Плохой» .EXE модуль

Шаг 2. Запуск «хорошего» .EXE модуля.

Написан текст исходного .EXE модуля, который выполняет те же функции, что и “хороший” .COM модуль. Таким образом, был получен «хороший» .EXE.

```
C:\>new_exe.exe
Type of PC: FC
Version of System: 5.0
Serial number of OEM: 255
Serial number of user: 000000
```

Рисунок 3 – «Хороший» .EXE модуль

Шаг 3. Сравнены исходные тексты для .COM и .EXE модулей. Даны ответы на контрольные вопросы «Отличия исходных текстов .COM и .EXE модулей»

1) Сколько сегментов должна содержать COM программа?

```
Object filename [new_com.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
new_com.asm(4): error A2062: Missing or unreachable CS
new_com.asm(12): error A2062: Missing or unreachable CS
new_com.asm(17): error A2062: Missing or unreachable CS
new_com.asm(22): error A2062: Missing or unreachable CS
new_com.asm(26): error A2062: Missing or unreachable CS
new_com.asm(39): error A2062: Missing or unreachable CS
new_com.asm(59): error A2062: Missing or unreachable CS
new_com.asm(67): error A2062: Missing or unreachable CS
new_com.asm(78): error A2062: Missing or unreachable CS
new_com.asm(85): error A2062: Missing or unreachable CS
new_com.asm(102): error A2062: Missing or unreachable CS
new_com.asm(116): error A2062: Missing or unreachable CS
new_com.asm(130): error A2062: Missing or unreachable CS
new_com.asm(151): error A2062: Missing or unreachable CS

49938 + 457306 Bytes symbol space free

0 Warning Errors
14 Severe Errors
```

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

Ответ: Так как адрес сегмента неизвестен до загрузки этого сегмента в память, то нельзя пользоваться командами, которые используют адрес сегмента. Из-за того, что адрес не может быть определен, возникает ошибка при попытке конвертации .EXE в .COM. В .COM и код, и данные находятся в одном сегменте, поэтому отсутствуют межсегментные переходы и межсегментные вызовы.

Загрузчику необходима информация том, где в загрузочном модуле лежат поля адресов, требующие настройки на адреса основной памяти, и какие значения в эти поля помещать. Такая информация лежит как раз в таблице настройки, которой в файле COM нет. EXE-файл имеет заголовок, который используется при его загрузке. Заголовок состоит из форматированной части, содержащей сигнатуру и данные, необходимые для загрузки EXE-файла, и таблицы для настройки адресов (Relocation Table). Таблица состоит из значений в формате сегмент: смещение. К смещениям в загрузочном модуле, на которые указывают значения в таблице, после загрузки программы в память должен быть прибавлен сегментный адрес, с которого загружена программа.

Шаг 4. Даны ответы на контрольные вопросы «Отличия форматов файлов COM и EXE модулей»

Отличия форматов файлов COM и EXE модулей.

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

Ответ: Структура COM файла проста. В файлах данного типа содержатся только машинный код и данные программы. Размер COM - файла ограничен 64 кб, т.е. размером одного сегмента памяти. Код располагается с нулевого адреса.

2) Какова структура файла «плохого» EXE? С какого адреса располагается

код? Что располагается с 0 адреса?

Ответ: По структуре EXE файл сложнее, кроме кода программы в файле также содержится: заголовок файла, таблица настройки адресов, данные. Код располагается с адреса 300h. С 0 адреса располагается заголовок EXE файла, который содержит данные необходимые для загрузки программы.

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

Ответ: EXE-программы могут состоять из нескольких сегментов (кодов, данных, стека). EXE-файл имеет заголовок, который используется при его загрузке. Заголовок состоит из форматированной части, содержащей сигнатуру и данные, необходимые для загрузки EXE-файла, и таблицы для настройки адресов. Файл может занимать больше 64 Кбайт. В “хорошем” файле отсутствует директива ORG, поэтому код начинается с адреса 200h, а не 300h.

Шаг 5. Даны ответы на контрольные вопросы «Загрузка COM модуля в основную память»

Отличия форматов файлов COM и EXE модулей.

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

Ответ: Загрузка COM модуля происходит по принципу: PSP, данные, код, стек. Сегментные регистры указывают на начало PSP. Код располагается с адреса 100h.

2) Что располагается с 0 адреса?

Ответ: С 0 адреса располагается PSP.

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

Ответ: Все сегментные регистры имеют значения 119С. Они указывают на PSP.

CS	119C
DS	119C
ES	119C
SS	119C

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

Ответ: COM-программа генерирует стек автоматически. Значение регистра SP устанавливается так, чтобы он указывал на последнюю доступную в сегменте ячейку памяти. Таким образом программа занимает начало, а стек - конец сегмента. В стек записывается 0000h.

Шаг 6. Даны ответы на контрольные вопросы «Загрузка «хорошего» EXE модуля в память»

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

Ответ: Загрузка EXE модуля происходит по принципу: PSP, сегмент кода, сегмент данных, сегмент стека. Определяются значения сегментных регистров. DS и ES =119C и указывают на начало PSP, т.к. в регистр данных еще не был помещен адрес сегмента данных. CS=11F3 – указывает на начало сегмента кода, а SS=11AC – указывает на начало сегмента стека.

CS	11F3
DS	119C
ES	119C
SS	11AC

2) На что указывают регистры DS и ES?

Ответ: На начало сегмента PSP.

3) Как определяется стек?

Ответ: Стек определяется в описании сегмента стека при помощи «DW 512 DUP(?)»

4) Как определяется точка входа?

Ответ: Точка входа определяется благодаря директиве END, которая определяет IP-адрес, с которого начинается выполнение программы.

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.

