

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Операционные системы»
Тема: Исследование структур загрузочных модулей

Студент гр. 0381

Березовская В. В.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Цель работы.

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

Задание.

Ассемблерная программа должна читать содержимое предпоследнего байта ROM BIOS, по таблице, сравнивая коды, определять тип РС 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 и серийным номером пользователя. Полученные строки выводятся на экран.

Результатом выполнения этого шага будет «хороший» .COM модуль, а также необходимо построить «плохой» .EXE, полученный из исходного текста для .COM модуля.

Напишите текст исходного .EXE модуля, который выполняет те же функции и постройте его. Таким образом будет получен хороший .EXE модуль. Сравните исходные тексты для .COM и .EXE модулей. Сравните файлы .COM, «плохого» и «хорошего» .EXE модулей в шестнадцатеричном виде.

Выполнение работы.

Для написания исходного текста .COM модуля был использован шаблон из методических указаний. Были добавлены строки с названиями моделей для последующего вывода на экран.

При запуске программы выполняется переход на метку BEGIN, где происходит считывание байта, расположенного по адресу F000:FFFEh и содержащего информацию о модели компьютера. Затем этот байт последовательно сравнивается с значениями из таблицы в методических указаниях. Если обнаружено совпадение, выводится строка соответствующая данному коду модели, иначе выводится значение в шестнадцатеричном виде.

Для получения информации о версии DOS используется функция 30h прерывания 21h. Полученные значения переводятся в требуемый формат и выводятся на экран.

Результат работы программы:

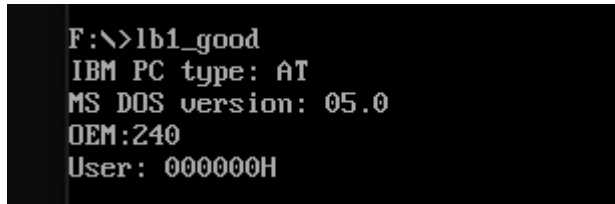
```
F:\>exe2bin lb1_bad.exe lb1_bad.com
F:\>lb1_bad.com
IBM PC type: AT
MS DOS version: 05.0
OEM:240
User: 000000H
```

Если из этого исходного кода построить .EXE модуль, он будет работать некорректно:

```
User: 000000H
F:\>lb1_bad.exe
                    IBM PC type: PC
                    5 0
                IBM PC type: PC
                240
e: PC
                000000
                    IBM PC type: PC
```

Для того, чтобы построить правильный .EXE модуль необходимо разделить программу на сегменты. Для этого в начале исходного текста добавляется

описание сегмента стека, а данные и код помещаются в собственные сегменты. Собранный из этого кода .EXE модуль выводит информацию о системе так же, как и .COM модуль.

A screenshot of a DOS command prompt window. The background is black, and the text is white. The prompt shows the command 'F:\>1b1_good' followed by several lines of system information: 'IBM PC type: AT', 'MS DOS version: 05.0', 'OEM:240', and 'User: 000000H'.

```
F:\>1b1_good
IBM PC type: AT
MS DOS version: 05.0
OEM:240
User: 000000H
```

Ответы на вопросы см. в разделе «Вопросы».

Выводы.

Были исследованы различия в структуре исходных текстов для модулей .COM и .EXE, структура загрузочных файлов этих типов и способы загрузки их в основную память.

ВОПРОСЫ

Отличия исходных текстов COM и EXE программ

1. Сколько сегментов должна содержать COM-программа?

COM-программа должна содержать только один сегмент, в котором хранятся PSP, данные, код и стек программы.

2. EXE-программа?

Как минимум один сегмент – сегмент кода, однако .EXE модуль может содержать сегменты стека и данных.

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

ASSUME, которая указывает, что этот сегмент будет использоваться в качестве сегмента кода и сегмента данных, а также ORG 100h, размещающая в первых 256 байт сегмента PSP, префикс программного сегмента.

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

Из-за того, что в .COM модуле отсутствует relocation table, таблица настройки адресов, команды с указанием сегментов не поддерживаются, поскольку в программе сегментные адреса задаются относительно начала программы и необходимо учитывать смещение начального сегмента программы, а без таблицы настройки адресов это невозможно.

Отличия форматов файлов COM и EXE модулей

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

COM файл содержит один сегмент с данными и кодом, размер не превышает 64 Кб. Так как в этом файле отсутствует заголовок и таблица настройки адресов, то код начинается с адреса 0h.

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
00000000	54	45	53	54	50	43	20	53	45	47	4d	45	4e	54	0d	0a	TESTPC SEGMENT..
00000010	20	20	20	20	41	53	53	55	4d	45	20	43	53	3a	54	45	ASSUME CS:TE
00000020	53	54	50	43	2c	20	44	53	3a	54	45	53	54	50	43	2c	STPC, DS:TESTPC,
00000030	20	45	53	3a	4e	4f	54	48	49	4e	47	2c	20	53	53	3a	ES:NOTHING, SS:
00000040	4e	4f	54	48	49	4e	47	0d	0a	20	20	20	20	4f	52	47	NOTHING... ORG
00000050	20	31	30	30	48	0d	0a	53	54	41	52	54	3a	20	4a	4d	100H..START: JM
00000060	50	20	42	45	47	49	4e	0d	0a	0d	0a	50	43	5f	54	59	P BEGIN....PC_TY
00000070	50	45	20	64	62	20	27	49	42	4d	20	50	43	20	74	79	PE db 'IBM PC ty
00000080	70	65	3a	20	50	43	27	2c	30	44	48	2c	30	41	48	2c	pe: PC',0DH,0AH,
00000090	27	24	27	20	3b	46	46	0d	0a	50	43	58	54	5f	54	59	'\$';FF..PCXT TY
000000a0	50	45	20	64	62	20	27	49	42	4d	20	50	43	20	74	79	PE db 'IBM PC ty
000000b0	70	65	3a	20	50	43	2f	58	54	27	2c	30	44	48	2c	30	pe: PC/XT',0DH,0
000000c0	41	48	2c	27	24	27	20	3b	46	45	2c	20	46	42	0d	0a	AH,'\$';FE, FB..
000000d0	50	43	4a	52	5f	54	59	50	45	20	64	62	20	27	49	42	PCJR_TYPE db 'IB
000000e0	4d	20	50	43	20	74	79	70	65	3a	20	50	43	6a	72	27	M PC type: PCjr'
000000f0	2c	30	44	48	2c	30	41	48	2c	27	24	27	20	3b	46	44	,0DH,0AH,'\$';FD
00000100	0d	0a	41	54	5f	54	59	50	45	20	64	62	20	27	49	42	..AT_TYPE db 'IB
00000110	4d	20	50	43	20	74	79	70	65	3a	20	41	54	27	2c	30	M PC_type: AT',0
00000120	44	48	2c	30	41	48	2c	27	24	27	20	3b	46	43	0d	0a	DH,0AH,'\$';FC..
00000130	50	53	54	57	4f	54	48	49	52	54	59	5f	54	59	50	45	PSTWOTHIRTY_TYPE
00000140	20	64	62	20	27	49	42	4d	20	50	43	20	74	79	70	65	db 'IBM PC type
00000150	3a	20	50	53	20	6d	6f	64	65	6c	20	33	30	27	2c	30	: PS model 30',0
00000160	44	48	2c	30	41	48	2c	27	24	27	20	3b	46	41	0d	0a	DH,0AH,'\$';FA..
00000170	50	43	43	5f	54	59	50	45	20	64	62	20	27	49	42	4d	PCC_TYPE db 'IBM
00000180	20	50	43	20	74	79	70	65	3a	20	50	43	20	43	6f	6e	PC type: PC Con
00000190	76	65	72	74	69	62	6c	65	27	2c	30	44	48	2c	30	41	vertible',0DH,0A
000001a0	48	2c	27	24	27	20	3b	46	39	0d	0a	50	53	54	57	4f	H,'\$';F9..PSTWO
000001b0	45	49	47	48	54	59	5f	54	59	50	45	20	64	62	20	27	EIGHTY_TYPE db '
000001c0	49	42	4d	20	50	43	20	74	79	70	65	3a	20	50	43	20	IBM PC type: PC
000001d0	6d	6f	64	65	6c	20	38	30	27	2c	30	44	48	2c	30	41	model 80',0DH,0A
000001e0	48	2c	27	24	27	20	3b	46	38	0d	0a	56	45	52	53	49	H,'\$';F8..VERSI
000001f0	4f	4e	54	48	49	4e	47	0d	0a	20	20	20	20	4f	52	47	NOTHING... ORG

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

располагается код? Что располагается с адреса 0?

Такой файл содержит заголовок, таблицу настройки адресов и один сегмент, в котором находятся данные и код. Код располагается с адреса 300h , потому что 200h – заголовок и таблица настроек, а 100h – смещение ORG 100h.

С адреса 0h располагается заголовок EXE файла, который содержит сигнатуру EXE файла, длину образа программы, размер таблицы настройки, сегментный адрес стека, адрес точки входа, а также ряд других параметров, необходимых для загрузки.

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
00000000	4d	5a	02	01	03	00	00	00	20	00	00	00	ff	ff	00	00	MZ..... ..яя..
00000010	00	00	38	24	00	01	00	00	1e	00	00	00	01	00	00	00	..8\$.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
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

```

000002c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000300 e9 2b 01 49 42 4d 20 50 43 20 74 79 70 65 3a 20 й+.IBM PC type:|
00000310 50 43 0d 0a 24 49 42 4d 20 50 43 20 74 79 70 65 PC..$IBM PC type
00000320 3a 20 50 43 2f 58 54 0d 0a 24 49 42 4d 20 50 43 : PC/XT..$IBM PC
00000330 20 74 79 70 65 3a 20 50 43 6a 72 0d 0a 24 49 42 type: PCjr..$IB
00000340 4d 20 50 43 20 74 79 70 65 3a 20 41 54 0d 0a 24 M PC type: AT..$
00000350 49 42 4d 20 50 43 20 74 79 70 65 3a 20 50 53 20 IBM PC type: PS
00000360 6d 6f 64 65 6c 20 33 30 0d 0a 24 49 42 4d 20 50 model 30..$IBM P
00000370 43 20 74 79 70 65 3a 20 50 43 20 43 6f 6e 76 65 C type: PC Conve
00000380 72 74 69 62 6c 65 0d 0a 24 49 42 4d 20 50 43 20 rtible..$IBM PC
00000390 74 79 70 65 3a 20 50 43 20 6d 6f 64 65 6c 20 38 type: PC model 8
000003a0 30 0d 0a 24 4d 53 20 44 4f 53 20 76 65 72 73 69 0..$MS DOS versi
000003b0 6f 6e 3a 20 30 31 2e 20 20 20 0d 0a 24 4f 45 4d on: 01. ..$OEM
000003c0 3a 20 20 20 0d 0a 24 55 73 65 72 3a 20 20 20 20 : ..$User:
000003d0 20 20 20 48 0d 0a 24 24 0f 3c 09 76 02 04 07 04 H..$$.<.v....
000003e0 30 c3 51 8a e0 e8 ef ff 86 c4 b1 04 d2 e8 e8 e6 0ГQЪаипя†Д±.Тииж

```

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

«Хороший» содержит заголовок и таблицу настройки адресов, общая длина которых 200h. В отличие от «плохого» EXE файла после таблицы идет три отдельных сегмента: сегмента стека, сегмента данных и сегмент кода.

Загрузка COM модуля в основную память

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

Выделяется свободный сегмент памяти и его адрес заносится в сегментные регистры. Затем в первые 256 байт этого сегмента записывается PSP, после этого с диска загружается содержимое COM-файла без изменений. Устанавливается на конец этого сегмента указатель, и в стек записывается адрес возврата.

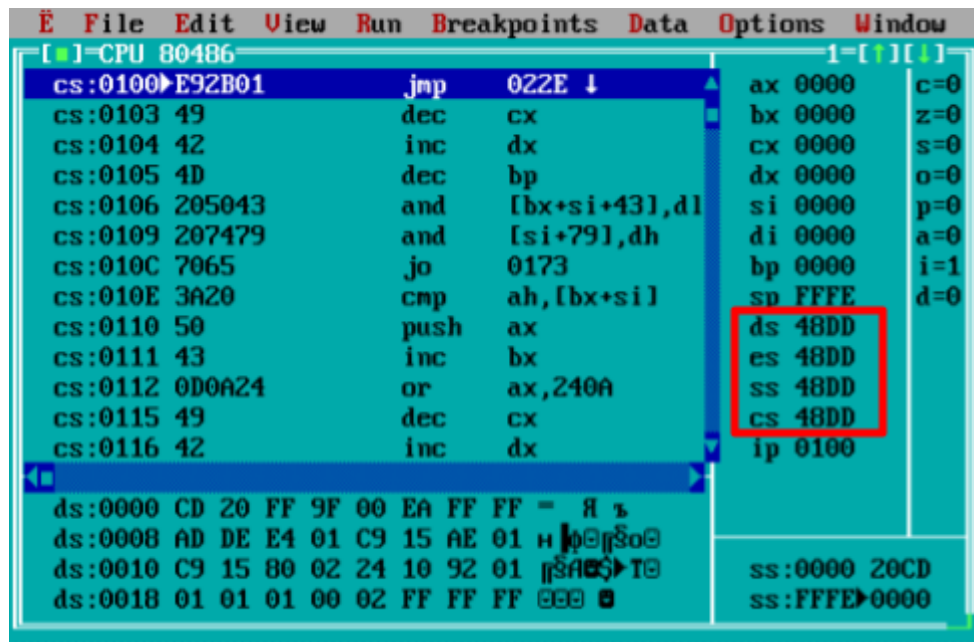
2. Что располагается с адреса 0?

Префикс программного сегмента.

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

Все сегментные регистры - CS, DS, ES, SS, - имеют одно и то же значение 48DD, это и есть сегмент в который загружена

программа.



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

Какие адреса?

При запуске программы стек в COM определяется автоматически и находится в том же сегменте, что и остальная часть программы; указатель стека установлен на конец сегмента (FFFEh). Таким образом, под стек отводится оставшаяся часть сегмента после кода и данных.

Загрузка «хорошего» EXE модуля в основную память

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

Определяется сегментный адрес свободного участка памяти, размер которого достаточен для размещения программы. Загружается с адреса PSP+0010h EXE-файл. В процессе загрузки считывается информация PSP из начала файла и выполняется перемещение сегментов (DS и ES =48DD, устанавливаются на начало PSP). SS = 48ED - на начало сегмента

стека, аналогично CS, равный 490D, - на начало сегмента команд.

Из метки после директивы END берётся смещение точки входа в программу и помещается в IP.

2. На что указывают регистры DS и ES?

После загрузки программы регистры указывают на PSP.

3. Как определяется стек?

Стек задается парой регистров SS:SP. При запуске программы в SS помещается смещение сегмента стека относительно начального сегмента программы, который содержится в заголовке, плюс адрес начального сегмента. В SP же помещается значение напрямую из заголовка.

4. Как определяется точка входа?

Она задается с помощью директивы END и помещается в заголовок файла в виде сегментного адреса сегмента кода относительно начального сегмента и значения IP.