

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. 7383

Власов Р.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург
2019

Цель работы: исследование различий в структурах исходных текстов модулей .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

Ход работы.

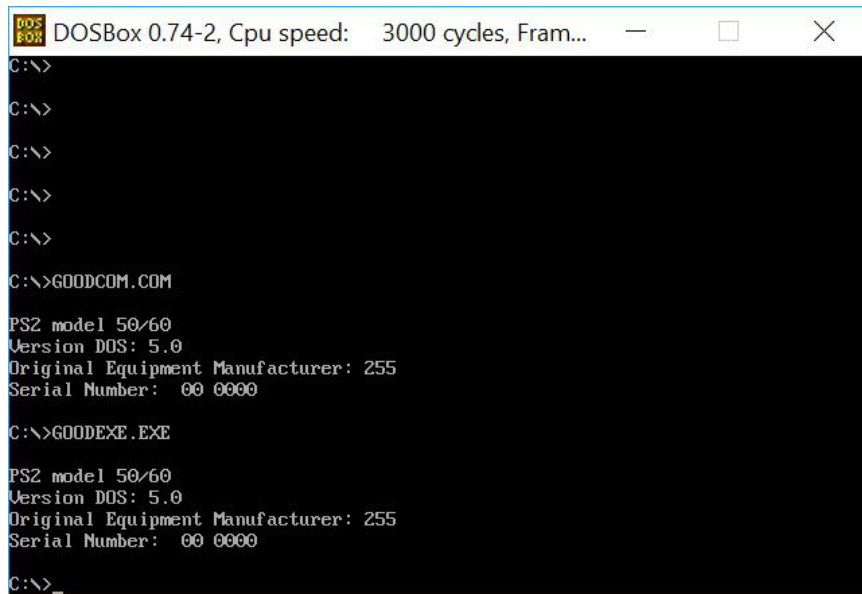
Функции и структуры данных управляющей программы:

Название	Назначение
PRINT_STRING	Вывод сообщения на экран
ENDL	Перевод каретки на новую строку
TETR_TO_HEX	Перевод числа из 2-ой в 16-ую с/с (1/2 байта)
BYTE_TO_HEX	Перевод числа из 2-ой в 16-ую с/с (1 байт)
WRD_TO_HEX	Перевод числа из 2-ой в 16-ую с/с (2 байта)
BYTE_TO_DEC	Перевод числа из 2-ой в 10-ую с/с (1 байт)
GET_PC_TYPE	Определяет тип IBM PC
GET_SYS_VER	Определяет версию системы
GET_OEM_NUM	Определяет OEM
GET_SERIAL_NUM	Определяет серийный номер пользователя

Структуры данных управляющей программы:

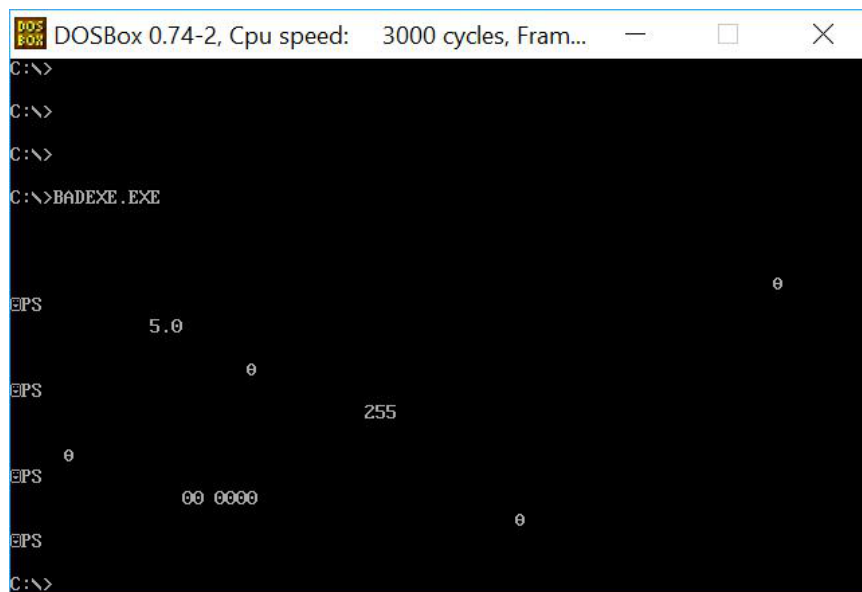
Наименование	Назначение
PC_TYPE	Хранит тип IBM PC
SYSTEM_VERSION	Хранит версию системы
OEM_NUMBER	Хранит номер OEM
SERIAL_NUMBER	Хранит серийный номер пользователя

Программа определяет и выводит на экран тип IBM PC, версию ОС, серийный номер OEM и серийный номер пользователя. Результаты работы программы приведены на рисунках 1 и 2.



```
DOS FOR DOSBox 0.74-2, Cpu speed: 3000 cycles, Fram...
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>GOODCOM.COM
PS2 model 50/60
Version DOS: 5.0
Original Equipment Manufacturer: 255
Serial Number: 00 0000
C:\>GOODEXE.EXE
PS2 model 50/60
Version DOS: 5.0
Original Equipment Manufacturer: 255
Serial Number: 00 0000
C:\>_
```

Рисунок 1 – Результат работы программ GOODCOM.COM и GOODEXE.EXE



```
DOS FOR DOSBox 0.74-2, Cpu speed: 3000 cycles, Fram...
C:\>
C:\>
C:\>
C:\>BADEXE.EXE
EPS
5.0
0
EPS
255
0
EPS
00 0000
0
EPS
C:\>
```

Рисунок 2 – Результат работы программы BADEXE.EXE

Ответы на контрольные вопросы:

Отличия исходных текстов COM и EXE программ

1. Сколько сегментов должна содержать COM-программа?

COM-программа должна содержать один сегмент.

2. Сколько сегментов должна содержать EXE-программа?

EXE-программа может содержать один или более сегментов.

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

Обязательная должна быть директива `ORG 100h`. Она нужна по той причине, что при загрузке модуля в ОП в начале COM-программы определяется 256-байтовый (100h) префикс программного сегмента, так что адресация имеет смещение в 256 байт от нулевого адреса. Это смещение мы и задаём директивой `ORG 100h` (в отличие от EXE-программы, где PSP расположен вне кодового сегмента и, следовательно, явно определять смещение там не нужно).

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

Нельзя использовать команды с дальней адресацией, поскольку в COM-программе отсутствует таблица настроек, которая указывает, какие абсолютные адреса при загрузке должны быть изменены, так как до загрузки неизвестно, куда будет загружена программа.

Отличия форматов файлов COM и EXE модулей

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

COM-файл содержит данные и машинные команды. Код начинается с адреса `0h` (но при загрузке модуля устанавливается смещение в `100h`). Структура файла показана на рисунке 3.

2. Какова структура файла «плохого» EXE? С какого адреса располагается код? Что располагается с адреса `0`?

Структура «плохого» EXE-файла менее компактна, чем у COM-файла. Код располагается с адреса 300h, с нулевого адреса располагается таблица настроек, при помощи которых строится данный EXE-файл. Структура файла показана на рисунке 4.

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

Структура «хорошего» EXE-файла несколько компактнее, чем структура «плохого» файла, так как в этом файле отсутствует директива ORG 100h, резервирующая пространство для заголовка. Именно поэтому код располагается с адреса 200h, а не с 300h, как в «плохом» EXE-файле. Структура файла показана на рисунке 5.

E:\OS\lab1\GOODCOM.COM															
00000000:	0AE9	5001	2453	5350	582F	2454	5350	2032	FFFFFFFF	'					
000000010:	6F6D	6564	206C	3033	5024	3253	6D20	646F	FFFFFFFF						
000000020:	6C65	3520	2F30	3036	5024	3253	6D20	646F	FFFFFFFF						
000000030:	6C65	3820	2430	4350	726A	5024	2043	6F43	FFFFFFFF	-0					
000000040:	766E	7265	6974	6C62	2465	6556	7372	6F69	FFFFFFFF						
000000050:	206E	4F44	3A53	2020	2020	4F24	6972	6967	FFFFFFFF	+					
000000060:	616E	206C	7145	6975	6D70	6E65	2074	614D	FFFFFFFF	4					
000000070:	756E	6166	7463	7275	7265	203A	2020	2020	FFFFFFFF	+					
000000080:	5324	7265	6169	206C	754E	626D	7265	203A	FFFFFFFF	>					
000000090:	2020	2020	2020	2020	2020	2020	2020	2420	FFFFFFFF	+					
0000000A0:	B450	CD09	5821	50C3	B452	B202	CD0A	B221	FFFFFFFF						
0000000B0:	CD0D	5A21	C358	5251	E432	D233	0AB9	F700	FFFFFFFF						
0000000C0:	80F1	30CA	1488	334E	3DD2	000A	F173	003C	FFFFFFFF	<					
0000000D0:	0474	300C	0488	595A	53C3	FC8A	12E8	8800	FFFFFFFF						
0000000E0:	4F25	0588	8A4F	E8C7	0007	2588	884F	5B05	FFFFFFFF						
0000000F0:	51C3	E08A	0BE8	8600	B1C4	D204	E8E8	0002	FFFFFFFF						
000000100:	C359	0F24	093C	0276	0704	3004	1EC3	96E8	FFFFFFFF						
000000110:	B8FF	F000	D88E	FEBB	8AFF	1F07	FF3C	0775	FFFFFFFF						
000000120:	168D	0103	48EB	3C90	75FE	8D07	0616	EB01	FFFFFFFF						
000000130:	9040	FB3C	0675	06BA	EB01	9036	FA3C	0775	FFFFFFFF						
000000140:	168D	010C	28EB	3C90	75FC	8D07	1916	EB01	FFFFFFFF						
000000150:	9020	F83C	0775	168D	0129	15EB	3C90	75FD	FFFFFFFF						
000000160:	8D07	3616	EB01	900A	F93C	168D	013B	01EB	FFFFFFFF						
000000170:	E890	FF2C	30E8	B4FF	CD30	8D21	4A36	5601	FFFFFFFF						
000000180:	8350	0FC6	C48A	2DE8	B0FF	882E	4E04	5A58	FFFFFFFF						
000000190:	23E8	E8FF	FF0A	0EE8	8DFF	5B36	5601	C683	FFFFFFFF						
0000001A0:	8A23	E8C7	FF10	E85A	FEF6	FAE8	8DFE	813E	FFFFFFFF						
0000001B0:	5701	C783	8A10	E8C3	FF37	86AA	AAC4	C783	FFFFFFFF						
0000001C0:	8B04	E8C1	FF13	E85A	FED6	DAE8	B8FE	4C00	FFFFFFFF						
0000001D0:	21CD								00000000						

Рисунок 3 – Структура COM файла

Загрузка COM-модулей в основную память

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

Порядок загрузки модуля COM: PSP, данные и код, стек. Код начинается с адреса 100h.

2. Что располагается с адреса 0?

С нулевого адреса располагается PSP.

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

Все сегментные регистры имеют значение «48DD» и указывают на начало PSP.

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

Стек занимает всё свободное пространство до конца файла (размер .COM файла не может превышать 64 кб), оставшееся после загрузки данных и кода. В данном случае значение регистра SP=FFFE.

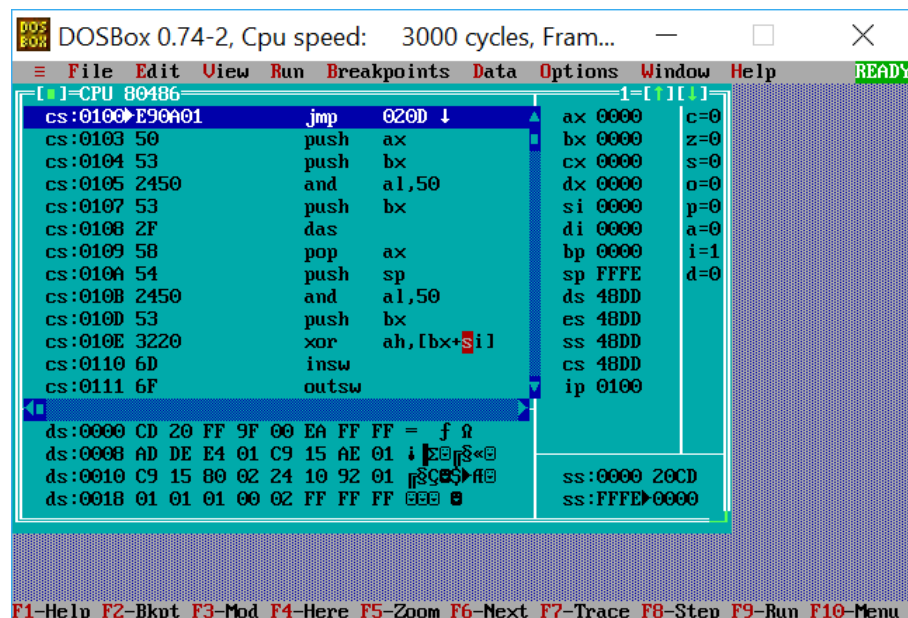


Рисунок 6 – COM модуль в отладчике TD.EXE

Загрузка «хорошего» EXE-модуля в основную память

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

Порядок загрузки EXE-модуля: PSP, сегмент кода, сегмент данных, сегмент стека. Сегментные регистры на момент загрузки программы имеют значения: ES=48DD, CS=48FB, DS=48DD, SS=48ED. На начальном этапе ES=DS, так как не были выполнены команды “mov ax, data; mov ds, ax”, т.е. в регистр данных не был помещён адрес сегмента данных.

2. На что указывают регистры DS и ES?

ES указывает на начало PSP, DS указывает на начало данных. После выполнения команд (см. предыдущий вопрос), значение DS= 48F1.

3. Как определяется стек?

Для стека в программе выделяется отдельный сегмент с параметром STACK. SS указывает на начало стека, а SP – на верхушку стека.

4. Как определяется точка входа?

Точка входа определяется с помощью директивы END <точка_входа>. Точкой входа можно указать как процедуру, так и метку.

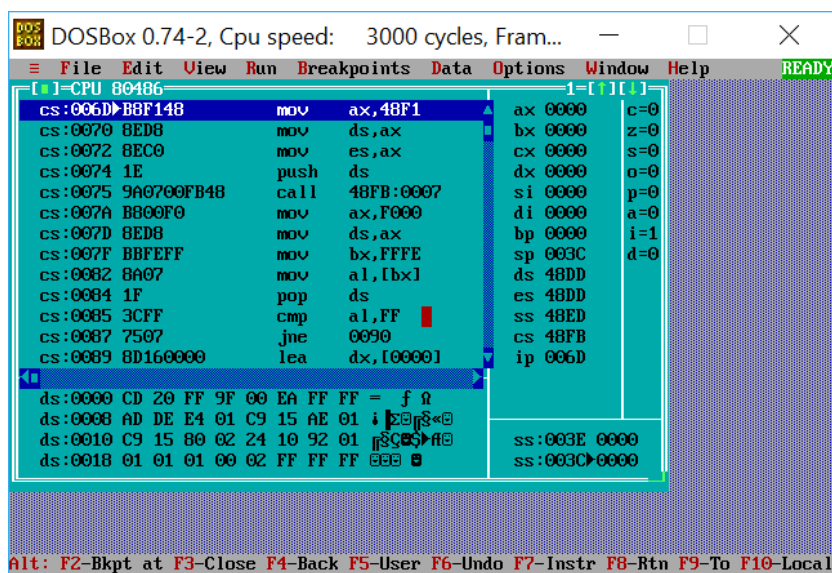


Рисунок 7 – EXE модуль в отладчике TD.EXE

Выводы:

В ходе данной лабораторной работы было проведено сравнение структуры COM и EXE файлов, исследованы различия в исходных текстах модулей COM и EXE, а также были сравнены способы их загрузки в память.