

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(ка) гр. 9382

Голубева В.П.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

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

Задание.

«Истина познается в сравнении», как говорили древние. К счастью, у нас есть возможность исследовать в одной системе два различных формата загрузочных модулей, сравнить их и лучше понять как система программирования и управляющая программа обращаются с ними. Система программирования включает компилятор с языка ассемблер (часто называется, просто, ассемблер), который изготавливает объектные модули. Компоновщик (Linker) по совокупности объектных модулей, изготавливает загрузочный модуль, а также, функция ядра — загрузчик, которая помещает программу в основную память и запускает на выполнение. Все эти компоненты согласованно работают для изготовления и выполнения загрузочных модулей разного типа. Для выполнения лабораторной работы сначала нужно изготовить загрузочные модули.

Шаг 1. Напишите текст исходного .COM модуля, который определяет тип РС и версию системы. Это довольно простая задача и для тех, кто уже имеет опыт программирования на ассемблере, это будет небольшой разминкой. Для тех, кто раньше не сталкивался с программированием на ассемблере, это неплохая задача для первого опыта.

За основу возьмите шаблон, приведенный в разделе «Основные сведения».

Необходимые сведения о том, как извлечь требуемую информацию, представлены в

следующем разделе.

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 и серийным номером пользователя.

Полученные строки выводятся на экран.

Отладьте полученный исходный модуль.

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

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 TB.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.

Необходимые сведения для составления программы

Тип IBM PC хранится в байте по адресу 0F000:0FFFEh, В предпоследнем байте ROM

BIOS. Соответствие кода и типа в таблице:

PC FF

PC/XT FE, FB

AT FC

P82 модель 30 FA

PSZ модель 50 или 60 FC

PSZ модель 80 F8

Per FD

PC Convertible F9

Для определения версии MS DOS следует воспользоваться функцией
ЗОН

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

АН — номер модификации

ВН - серийный номер OEM (Original Equipment Manufacturer)

BL:СХ - 24-битовый серийный номер пользователя.

Контрольные вопросы по лабораторной работе N91

Отличия исходных текстов COM и EXE программ

- 1) Сколько сегментов должна содержать COM—программа?
- 2) EXE—программа?
- 3) Какие директивы должны обязательно быть в тексте COM-программы?
- 4) Все ли форматы команд можно использовать в COM—программе?

Отличия форматов файлов COM и EXE модулей

- 1) Какова структура файла COM? С какого адреса располагается код?
- 2) Какова структура файла «плохого» EXE? С какого адреса располагается код? Что располагается с адреса 0?
- 3) Какова структура файла «хорошего» EXE? Чем он отличается от файла «плохого» EXE?

Загрузка COM модуля в основную память

- 1) Какой формат загрузки модуля COM? С какого адреса располагается код?
- 2) Что располагается с адреса 0?
- 3) Какие значения имеют сегментные регистры? На какие области памяти они указывают?
- 4) Как определяется стек? Какую область памяти он занимает? Какие адреса?

Загрузка «хорошего» EXE модуля в основную память

- 1) Как загружается «хороший» EXE? Какие значения имеют сегментные регистры?
- 2) На что указывают регистры DS и ES?
- 3) Как определяется стек?
- 4) Как определяется точка входа?

Выполнение работы.

Были созданы и отлажены программы lab1_e.asm, который формирует «хороший» EXE и lab1_c.asm, которая формирует COM файл. Загрузочные файлы, которые из них формируются, были открыты при помощи утилиты Unhex в шестнадцатеричном виде для сравнения и ответов на вопросы.

Ответы на контрольные вопросы

Отличия исходных текстов COM и EXE программ

- 1) com программа должна содержать один сегмент
- 2) exe программа должна содержать не менее одного сегмента
- 3) В com программе должны быть директивы org, assume
- 4) Нет. Нельзя, например, использовать команды вида mov <регистр>, <имя сегмента>. Это происходит потому, что com программа не содержит таблицы настроек, в которой содержатся адреса, которые зависят от расположения загрузочного модуля в оперативной памяти. Поэтому нельзя использовать команды, которые дают доступ к началу сегмента.

Отличия форматов файлов COM и EXE модулей

- 1) Код располагается с адреса 100h, то есть сразу после префикса программы. Файл содержит данные и команды.

2) В плохом ехе данные и код хранятся в одном месте. С адреса 0 располагается PSP и таблица настроек. Код и данные располагаются с адреса 300h.

3) В хорошем ехе данные и стек разделены по сегментам. В плохом ехе имеется смещение 300h, 200h — смещение для PSP, 100h — смещение от директивы org. У хорошего ехе должна выделяться память под стек, и код начинаться с 400h.

Загрузка COM модуля в основную память

1) Ищется свободное место в памяти, там создается блок памяти для PSP, задается смещение 100h, оттуда загружается программа, сегментные регистры устанавливаются на начало PSP, указатель стека устанавливается на FFFh, в IP записывается 100h. Код располагается с адреса 100h.

2) С адреса 0 располагается префикс сегмента программы.

3) Сегментные регистры будут указывать на начало PSP, будут иметь значения 0.

4) Стек располагается с последнего байта, доступного программе — он занимает все байты от конца кода до байта FFFh. Вообще, стек определяется только SP. Мы кладём туда какой-нибудь адрес и говорим, «вот с этого места будет стек».

Загрузка «хорошего» EXE модуля в основную память

1) Система, загрузив программу в память, инициализирует сегментные регистры, так что регистры DS и ES указывают на начало PSP, CS - на начало сегмента команд, а SS - на начало сегмента стека. В указатель команд IP загружается смещение точки входа в программу.

2) DS и ES указывают на начало PSP.

3) Стек определяет с помощью директивы STACK. Затем нужно задать размер стека.

4) Точка входа берётся из операнда директивы END

В рис. 1 демонстрируются работоспособность программы.

```
C:\>lab1_e.exe
Значение в регистре AX= FC
Тип IBM: AT
Версия системы:      5.0
Серийный номер OEM в 16сс:  43
Серийный номер пользователя: 000000
```

Рис. 1. Результат работы программы lab1_e.exe

Как выглядит файл для com программы демонстрируется в Рис. 2

00000000	54 45 53 54 50 43 20 53 45 47 4D 45 4E 54 0A 20	TESTPC SEGMENT
00000010	20 20 20 41 53 53 55 4D 45 20 43 53 3A 54 45 53	ASSUME CS:TES
00000020	54 50 43 2C 20 44 53 3A 54 45 53 54 50 43 2C 20	TPC, DS:TESTPC,
00000030	45 53 3A 4E 4F 54 48 49 4E 47 2C 20 53 53 3A 4E	ES:NOTHING, SS:N
00000040	4F 54 48 49 4E 47 0A 4F 52 47 20 31 30 30 48 0A	OTHING,ORG 100H
00000050	20 20 20 20 53 54 41 52 54 3A 20 4A 4D 50 20 42	START: JMP B
00000060	45 47 49 4E 0A 53 54 52 49 4E 47 5F 41 58 20 64	EGIN,STRING_AX d
00000070	62 20 27 87 AD A0 E7 A5 AD A8 A5 20 A2 20 E0 A5	b 'Значение в ре
00000080	A3 A8 E1 E2 E0 A5 20 41 58 3D 20 27 2C 20 27 24	гистре AX= ', '\$
00000090	27 0A 73 74 72 69 6E 67 5F 69 62 6D 20 64 62 20	'string_ibm db
000000A0	27 92 A8 AF 20 49 42 4D 3A 20 27 2C 27 24 27 0A	'Тип IBM: ', '\$'
000000B0	53 54 52 49 4E 47 5F 4F 45 4D 20 64 62 20 27 91	STRING_OEM db 'C
000000C0	A5 E0 A8 A9 AD EB A9 20 AD AE AC A5 E0 20 4F 45	ерийный номер OE
000000D0	4D 20 A2 20 31 36 E1 E1 3A 20 27 2C 20 27 24 27	M в 16сс: ', '\$'
000000E0	0A 53 54 52 49 4E 47 5F 53 45 52 5F 4E 55 4D 20	STRING_SER_NUM
000000F0	64 62 20 27 91 A5 E0 A8 A9 AD EB A9 20 AD AE AC	db 'Серийный ном
00000100	A5 E0 20 AF AE AB EC A7 AE A2 A0 E2 A5 AB EF 3A	ер пользователя:
00000110	20 27 2C 20 27 24 27 0A 53 54 52 49 4E 47 5F 56	', '\$',STRING_U
00000120	45 52 53 20 64 62 20 27 82 A5 E0 E1 A8 EF 20 E1	ERS db 'Версия с
00000130	A8 E1 E2 A5 AC EB 3A 20 20 20 20 20 27 2C 20 27	истемы: ', '
00000140	24 27 0A 73 74 72 69 6E 67 5F 65 6E 74 20 64 62	'\$',string_ent db
00000150	20 27 20 27 2C 20 30 41 48 2C 20 30 44 48 2C 20	', ', 0AH, 0DH,
00000160	27 24 27 0A 73 74 72 69 6E 67 5F 64 6F 74 20 64	'\$',string_dot d
00000170	62 20 27 2E 27 2C 20 27 24 27 0A 73 74 72 69 6E	b '. ', '\$',strin

Рис. 2. Файл в lab1_c.asm шестнадцатеричном виде

Как выглядит плохой exe демонстрируется в рис. 3 и рис. 4.

00000000	4D 5A 1B 01 03 00 00 00	20 00 00 00 FF FF 00 00	MZ←☐
00000010	00 00 5B E6 00 01 00 00	1E 00 00 00 01 00 00 00	☐ Ц ☐ ▲ ☐
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	
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	

Рис. 3. Структура «плохого» ехе файла, начало

00000250	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	значение в ре гистре AX= \$Тип IBM: \$Серийный н омер OEM в 16сс: \$Серийный номер пользователя: \$ Версия системы: \$ ☐r\$. \$PC\$PC /XT\$AT\$PS2 модел ь 30\$PS2 модель 80\$PCjr\$PC Conve rtible\$Неизвестн ый тип IBM\$\$*<ou
00000260	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000270	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000280	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000290	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000002A0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000002B0	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 A5 01 87 AD A0 E7 A5	AD A8 A5 20 A2 20 E0 A5	
00000310	A3 A8 E1 E2 E0 A5 20 41	58 3D 20 24 92 A8 AF 20	
00000320	49 42 4D 3A 20 24 91 A5	E0 A8 A9 AD EB A9 20 AD	
00000330	AE AC A5 E0 20 4F 45 4D	20 A2 20 31 36 E1 E1 3A	
00000340	20 24 91 A5 E0 A8 A9 AD	EB A9 20 AD AE AC A5 E0	
00000350	20 AF AE AB EC A7 AE A2	A0 E2 A5 AB EF 3A 20 24	
00000360	82 A5 E0 E1 A8 EF 20 E1	A8 E1 E2 A5 AC EB 3A 20	
00000370	20 20 20 20 24 20 0A 0D	24 2E 24 50 43 24 50 43	
00000380	2F 58 54 24 41 54 24 50	53 32 20 AC AE A4 A5 AB	
00000390	EC 20 33 30 24 50 53 32	20 AC AE A4 A5 AB EC 20	
000003A0	38 30 24 50 43 6A 72 24	50 43 20 43 6F 6E 76 65	
000003B0	72 74 69 62 6C 65 24 8D	A5 A8 A7 A2 A5 E1 E2 AD	
000003C0	EB A9 20 E2 A8 AF 20 49	42 4D 24 24 0F 3C 09 76	

Рис. 4. Структура «плохого» ехе файла, продолжение

00000000	4D 5A 2E 01 03 00 01 00	20 00 00 00 FF FF 00 00	
00000010	00 01 29 6F DD 00 1D 00	1E 00 00 00 01 00 DE 00	
00000020	1D 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	
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	

Рис. 5. Структура «хорошего» ехе файла, начало

00000240	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000250	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000260	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000270	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000280	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000290	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000002A0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000002B0	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	87 AD A0 E7 A5 AD A8 A5	20 A2 20 E0 A5 A3 A8 E1	Значение в регистре AX= \$Тип IBM
00000310	E2 E0 A5 20 41 58 3D 20	24 92 A8 AF 20 49 42 4D	: \$Серийный номер OEM в 16сс:
00000320	3A 20 24 91 A5 E0 A8 A9	AD EB A9 20 AD AE AC A5	\$Серийный номер пользователя: \$
00000330	E0 20 4F 45 4D 20 A2 20	31 36 E1 E1 3A 20 20 20	Версия системы: \$
00000340	20 24 91 A5 E0 A8 A9 AD	EB A9 20 AD AE AC A5 E0	\$
00000350	20 AF AE AB EC A7 AE A2	A0 E2 A5 AB EF 3A 20 24	\$
00000360	82 A5 E0 E1 A8 EF 20 E1	A8 E1 E2 A5 AC EB 3A 20	\$
00000370	20 20 20 20 20 20 24 20	0A 0D 24 2E 24 50 43 24	\$
00000380	50 43 2F 58 54 24 41 54	24 50 53 32 20 AC AE A4	PC/XT\$AT\$PS2 модель 30\$PS2 модель
00000390	A5 AB EC 20 33 30 24 50	53 32 20 AC AE A4 A5 AB	\$
000003A0	EC 20 38 30 24 50 43 6A	72 24 50 43 20 43 6F 6E	\$
000003B0	26 65 22 24 69 62 6C 65	24 8D A5 A8 A2 A2 A5 E1	\$

Рис. 4. Структура «хорошего» ехе файла, продолжение

Выводы.

Была написана программа для ассемблера, которая извлекала информацию о компьютере. Было изготовлено несколько типов загрузочных модулей и изучены их различия.