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 заголовочных модулей

Студент гр. 0381 _____ Михайлов В.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

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

Постановка задачи.

Шаг 1. Напишите текст исходного .COM модуля, который определяет тип PC 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 вразделе «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 форматовфайлов COM и EXE модулей».

Шаг 5. Откройте отладчик TD.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.

Исходные данные.

За основу был взят предоставленный шаблон, содержащий процедуры: TETR_TO_HEX, BYTE_TO_HEX_WRD_TO_HEX, BYTE_TO_DEC.

Таблица 1 – Соответствие типа IBM PC шестнадцатеричному коду.

Тип IBM PC	Код
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

Выполнение работы.

Первым шагом создаётся исходный код .COM модуля – файл lr1com.asm. В начале объявлены строки для вывода в консоль посредством процедуры writestring. Строка для вывода определяется в процедуре pc_type. В регистре al хранится значение байта, в котором записан код системы, и он сравнивается с кодами из таблицы 1. При совпадении значений производится переход к метке, после которой в регистр dx запишется необходимое сообщение для вывода. Далее вызывается процедура writestring. Если соответствие не найдено, то выводится сообщение об этом. В свою очередь процедура os_ver определяет версию системы, серийный номер OEM и номер пользователя. Для получения соответствующих данных применяется функция 30h и прерывание 21h.

Написанный код транслируется и компонуется. Получается «плохой» .EXE модуль. Его запуск даёт следующую картину:

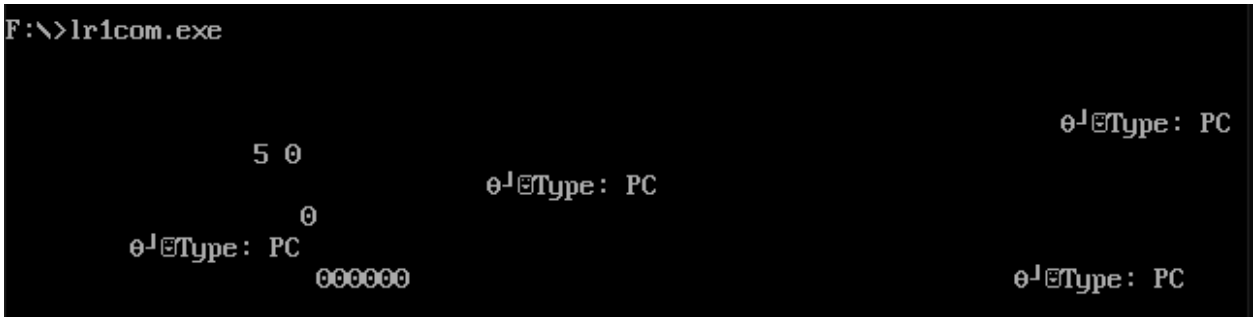

Рис.1 – «плохой» .EXE модуль

Для корректного результата необходимо сформировать .COM модуль. Для этого используется команда exe2bin lr1com.exe lr1com.com. Теперь вывод происходит «хорошо»:

Рис.2 – «хороший» .COM модуль

Затем создаётся «хороший» .EXE модуль – файл lr1exe.asm. Для этого достаточно позаимствовать код из предыдущего файла и внести некоторые изменения: добавить определения сегментов стека и данных, перенести строки в сегмент данных, реализовать процедуру main, в которой происходит загрузка адреса сегмента данных и вызов процедур pc_type и os_ver. Так, после сборки и запуска исполняемого файла получается корректное сообщение:

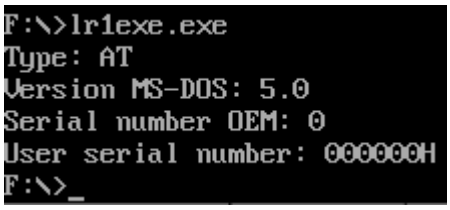

Рис.3 – «хороший» .EXE модуль

Вывод.

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

Контрольные вопросы.

Отличия исходных текстов COM и EXE программ

1) Сколько сегментов должна содержать COM-программа?

Ответ: один, содержащий данные и код, а стек генерируется автоматически.

2) EXE-программа?

Ответ: 3. Данные, код и стек, но возможно объединение сегментов.

3) Какие директивы должны обязательно быть в тексте .COM

программы?

Ответ: ORG 100h – смещение кода на 256 байт от нулевого адреса (пропуск области PSP), ASSUME – чтобы сегментные регистры указывали на один сегмент.

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

Ответ: нет. В .COM модуле нет таблицы настройки адресов, по которой осуществляется поиск абсолютных адресов сегментов, поэтому нельзя использовать команды с указанием сегментов.

Отличия форматов файлов COM и EXE модулей

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

Ответ: у COM файла есть только один сегмент, в котором располагаются код и данные. Модуль ограничен размером 64 Кб. Код размещается с нулевого адреса: сначала идёт команда jmp begin, далее располагаются данные, затем – код. Однако при загрузке модуля устанавливается смещение на 256 байт.

LR1COM.COM

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
00000000	3f	54	79	70	65	3a	20	50	43	0d	0a	24	54	79	70	65	?Type: PC..$Type
00000010	3a	20	50	43	2f	58	54	0d	0a	24	54	79	70	65	3a	20	: PC/XT..$Type:
00000020	41	54	0d	0a	24	54	79	70	65	3a	20	50	53	32	20	6d	AT..$Type: PS2 m
00000030	6f	64	65	6c	20	33	30	0d	0a	24	54	79	70	65	3a	20	odel 30..$Type:
00000040	50	53	32	20	6d	6f	64	65	6c	20	35	30	20	6f	72	20	PS2 model 50 or
00000050	36	30	0d	0a	24	54	79	70	65	3a	20	50	53	32	20	6d	60..$Type: PS2 m
00000060	6f	64	65	6c	20	38	30	0d	0a	24	54	79	70	65	3a	20	odel 80..$Type:
00000070	50	d1	6a	72	0d	0a	24	54	79	70	65	3a	20	50	43	20	PCjr..$Type: PC
00000080	43	6f	6e	76	65	72	74	69	62	6c	65	0d	0a	24	56	65	Convertible..$Ve
00000090	72	73	69	6f	6e	20	4d	53	2d	44	4f	53	3a	20	20	2e	rsion MS-DOS: .
000000a0	20	20	0d	0a	24	53	65	72	69	61	6c	20	6e	75	6d	62	..$Serial numb
000000b0	65	72	20	4f	45	4d	3a	20	20	0d	0a	24	55	73	65	72	er OEM: ..$User
000000c0	20	73	65	72	69	61	6c	20	6e	75	6d	62	65	72	3a	20	serial number:
000000d0	20	20	20	20	20	20	48	20	24	24	0f	3c	09	76	02	04	H $$.<.v..
000000e0	07	04	30	4e	3f	3f	79	3f	3f	04	3f	3f	59	4f	3f	75	..0N??y??.??YO?u
000000f0	3f	3f	25	4f	3f	05	4f	3f	4b	3f	3f	25	4f	3f	05	5b	??%O?.O?K??%O?.[
00000100	4e	52	32	3f	3f	0d	0a	20	3f	3f	3f	3f	14	4e	33	3f	NR2??.. ????.N3?
00000110	0d	0a	20	73	3f	3f	04	0c	30	3f	04	5a	59	6f	09	3f	.. s??..0?.ZYo.?
00000120	6f	20	3f	3f	a0	3f	79	3c	79	74	1c	3c	3f	74	1e	3c	o ??.?y<yt.<?t.<
00000130	75	74	1a	3c	75	74	1c	3c	75	74	1e	3c	6f	74	26	3c	ut.<ut.<ut.<ot&<
00000140	79	74	28	3c	75	74	2a	3f	03	01	3f	3f	0e	01	3f	3f	yt(<ut*?..??..??
00000150	1c	01	3f	3f	27	01	3f	3f	3c	01	3f	3f	57	01	3f	3f	..??'.??<.??W.??
00000160	6c	01	3f	3f	7a	01	3f	6f	30	3f	50	3f	3f	01	3f	3f	l.??z.?o0?P??.??
00000170	3f	58	3f	61	3f	3f	3f	3f	01	3f	3f	3f	01	3f	3f	3f	?X?a????.???.???
00000180	4b	56	79	3f	3f	01	3f	3f	3f	01	3f	55	3f	41	3f	3f	KVy??.???.?U?A??
00000190	65	17	79	3f	3f	05	3f	3f	01	3f	65	59	79	3f	32	41	e.y??.??.?eYy?2A
000001a0	3f	4c	3f														?L?

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

Ответ: данные и код в «плохом» EXE располагаются в одном сегменте, код располагается с адреса 300h, т.к. до 200h размещены заголовок и таблица настроек, далее идёт смещение 100h.

LR1COM.COM | LR1COM.EXE

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
000001e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
000001f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
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	e9	d9	01	54	79	70	65	3a	20	50	43	0d	0a	24	54	79	йЩ.Type: PC..$Ty
00000310	70	65	3a	20	50	43	2f	58	54	0d	0a	24	54	79	70	65	pe: PC/XT..$Type
00000320	3a	20	41	54	0d	0a	24	54	79	70	65	3a	20	50	53	32	: AT..$Type: PS2
00000330	20	6d	6f	64	65	6c	20	33	30	0d	0a	24	54	79	70	65	model 30..$Type
00000340	3a	20	50	53	32	20	6d	6f	64	65	6c	20	35	30	20	6f	: PS2 model 50 o
00000350	72	20	36	30	0d	0a	24	54	79	70	65	3a	20	50	53	32	r 60..$Type: PS2
00000360	20	6d	6f	64	65	6c	20	38	30	0d	0a	24	54	79	70	65	model 80..$Type
00000370	3a	20	50	d0	a1	6a	72	0d	0a	24	54	79	70	65	3a	20	: PPЎjr..$Type:
00000380	50	43	20	43	6f	6e	76	65	72	74	69	62	6c	65	0d	0a	PC Convertible..
00000390	24	56	65	72	73	69	6f	6e	20	4d	53	2d	44	4f	53	3a	$Version MS-DOS:
000003a0	20	20	2e	20	20	0d	0a	24	53	65	72	69	61	6c	20	6e	. ..$Serial n
000003b0	75	6d	62	65	72	20	4f	45	4d	3a	20	20	0d	0a	24	55	umber OEM: ..$U
000003c0	73	65	72	20	73	65	72	69	61	6c	20	6e	75	6d	62	65	ser serial numbe
000003d0	72	3a	20	20	20	20	20	20	20	48	20	24	24	0f	3c	09	r: H $$.<.
000003e0	76	02	04	07	04	30	c3	51	8a	e0	e8	ef	ff	86	c4	b1	v....0ГQЉаипя†Д±
000003f0	04	d2	e8	e8	e6	ff	59	c3	53	8a	fc	e8	e9	ff	88	25	.ТиижяYГSЉьийя€%
00000400	4f	88	05	4f	8a	c7	e8	de	ff	88	25	4f	88	05	5b	c3	O€.OЉЗиЮя€%O€.[Г
00000410	51	52	32	e4	33	d2	b9	0a	00	f7	f1	80	ca	30	88	14	QR2д3Т№..чсЪК0€.
00000420	4e	33	d2	3d	0a	00	73	f1	3c	00	74	04	0c	30	88	04	N3Т=..sc<.t..0€.
00000430	5a	59	c3	b4	09	cd	21	c3	b8	00	f0	8e	c0	26	a0	fe	ZYГґ.Н!Гё.рЋА&.ю
00000440	ff	3c	ff	74	1c	3c	fe	74	1e	3c	fb	74	1a	3c	fc	74	я<яt.<юt.<ыt.<ьt
00000450	1c	3c	fa	74	1e	3c	f8	74	26	3c	fd	74	28	3c	f9	74	.<ъt.<шt&<эt(<щt
00000460	2a	ba	03	01	eb	28	90	ba	0e	01	eb	22	90	ba	1c	01	*є..л(.є..л".є..
00000470	eb	1c	90	ba	27	01	eb	16	90	ba	3c	01	eb	10	90	ba	л..є'.л..є<.л..є

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

Ответ: «хороший» EXE файл содержит заголовок и таблицу настройки адресов, их общая длина 200h. После таблицы идут три сегмента: стека, данных и кода, а в «плохом» EXE файле есть только один сегмент.

Загрузка COM модуля в основную память

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

Ответ: происходит выделение свободного сегмента памяти, адрес заносится в сегментные регистры. Первые 256 байт этого сегмента занимает PSP, далее с адреса 100h загружается содержимое COM-файла. В стек записывается адрес возврата, SP указывает на конец сегмента.

2) Что располагается с адреса 0?

Ответ: префикс программного кода – PSP.

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

Ответ: сегментные регистры CS, DS, ES и SS указывают на сегмент кода, данных, дополнительные данные и стек соответственно. В начале выполнения программы регистры указывают на начало PSP.

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

Ответ: стек определяется автоматически, занимает место в сегменте кода, SP указывает на конец стека, а SS на начало, адреса стека находятся в диапазоне 0000h-FFFEh.

Загрузка «хорошего» EXE модуля в основную память

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

Ответ: сначала определяется сегментный адрес свободного участка памяти, размер которого достаточен для размещения программы, exe-файл загружается, начиная с адреса PSP:0100h, далее считывается стандартная часть заголовка в память и выполняется перемещение адресов сегментов, DS и ES устанавливаются на начало сегмента PSP, SS – на начало сегмента стека, CS – на начало сегмента кода, в IP загружается смещение точки входа в программу, которая берётся из метки после директивы END.

2) На что указывают регистры DS и ES?

Ответ: на PSP

3) Как определяется стек?

Ответ: директивой .stack. Ей обозначается начало сегмента стека или стандартной директивы segment с помощью следующей конструкции:

<segment_name> segment stack

…

<segment_name> ends

4) Как определяется точка входа?

Ответ: директивой END. После неё указывается метка, куда переходит программа при запуске.