

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 загрузочного модулей

Студент гр. 9382

Рыжих Р.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Постановка задачи

Цель работы.

Изучить основные принципы трансляции, отладки и выполнения программ на языке Ассемблера. Исследование различий в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

Сведения о функциях и структурах данных.

В данной программе используются следующие функции и структуры данных:

Процедура	Описание
TETR_TO_HEX	Перевод десятичной цифры в код символа, который записывается в AL
BYTE_TO_HEX	Перевод значений байта в число 16-ой СС и его представление в виде двух символов
WRD_TO_HEX	Перевод слова в число 16-ой СС и представление его в виде четырех символов
BYTE_TO_DEC	Перевод значения байта в число 10-ой СС и представляет его в виду символов
PRINT_STRING	Вывод строки на экран
PRINT_PC_TYPE	Печать на экран тип ПК
PRINT_OS_VERSION	Печать на экран версии ОС, серийного номера OEM и серийного номера пользователя

Последовательность действий

В ходе работы программа выполняет следующие действия:

1. Процедура PRINT_PC_TYPE, которая выводит на экран тип ПК пользователя. Информация о типе ПК находится в предпоследнем байте ROM BIOS по адресу 0F000:0FFFEh. Значение этого байта определяет тип: Ffh – PC, Feh/Fbh – PC/XT, FCh – AT, FAh – PS2 model 30, FCh –

PS2 model 50 or 60, F8h – PS2 model 80, FDh – Pcj, F9h – PC Convertible. Если значение байта не сходится со значениями типов ПК, то выводится сообщение об ошибке.

2. Процедура PRINT_OS_VERSION, которая выводит на экран версию ОС, серийный номер OEM и серийный номер пользователя. В данной процедуре используется функция 30h прерывания 21h.
3. Завершение работы программы.

Выполнение шагов лабораторной работы:

1 шаг:

Был написан текст исходного .COM модуля Lab1_COM.asm, который определяет тип ПК и версию его системы. Далее после компилирования был получен «плохой» .EXE модуль Lab1_COM.exe. При помощи EXE2BIN.EXE и «плохого» модуля был получен «хороший» .COM модуль Lab1_COM.com.

```
C:\>LAB1_COM.EXE

0|PC
5 0
0
0000000
0|PC
0|PC
0|PC
```

Рис. 1. - Пример работы "плохого" модуля .EXE Lab1_COM.exe

```
C:\>LAB1_COM.COM
AT
Version MS-DOS: 5.0
Serial Number OEM: 0
User Serial Number: 0000000H
```

Рис. 2. - Пример работы "хорошего" .COM модуля Lab1_COM.com

2 шаг:

Был написан исходный текст .EXE модуля lab1_exe.asm, который выполняет те же функции, что и модуль в Шаге 1. Далее был получен «хороший» .EXE модуль lab1_exe.exe.

```
C:\>LAB1_EXE.EXE
AT
Version MS-DOS: 5.0
Serial Number OEM: 0
User Serial Number: 0000000H
```

Рис. 3. - Пример работы хорошего .EXE модуля lab1_exe.exe

3 шаг:

«Отличия исходных текстов .COM и .EXE программ»

1) Сколько сегментов должна содержать COM-программа?

COM-программа должна содержать только один сегмент, потому что данные программы и сам хранятся в одном сегменте, а стек автоматически устанавливается на последнюю ячейку сегмента.

2) EXE-программа?

EXE-программа должна содержать один или более сегментов. Количество сегментов зависит от выбранной модели памяти.

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

В COM-программе обязательно должна быть директива `ORG 100h`. Данная директива устанавливает `CS:IP` на конец `PSP`, так как после загрузки все сегментные регистры (как и `CS`) указывают на начало `PSP`, а `IP = 0`, а это значит, что программа не будет выполняться, начиная с этого адреса. Именно эта директива смещает все относительные адреса на `100h` байт.

В COM-программе обязательно должна быть директива `ASSUME`. Данная директива указывает ассемблеру с каким сегментом или группой сегментов связаны регистры.

В COM-программе обязательно должна быть директива `END`. Данная директива завершает работу программы на ассемблере.

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

Нельзя использовать команды вида: `seg NAME`, где `NAME` – название сегмента, так как в COM-программе отсутствует таблица настройки.

Шаг 4:

Шестнадцатеричное представление модуля .COM:

00000000	E9	C6	01	50	43	0D	0A	24	50	43	2F	58	54	0D	0A	24	0F.PC..\$PC/XT..\$
00000010	41	54	0D	0A	24	50	53	32	20	D0	BC	D0	BE	D0	B4	D0	AT..\$PS2
00000020	B5	D0	BB	D1	8C	20	33	30	0D	0A	24	50	53	32	20	D0	30..\$PS2
00000030	BC	D0	BE	D0	B4	D0	B5	D0	BB	D1	8C	20	35	30	20	D0	50
00000040	B8	D0	BB	D0	B8	20	36	30	0D	0A	24	50	53	32	20	D0	60..\$PS2
00000050	BC	D0	BE	D0	B4	D0	B5	D0	BB	D1	8C	20	38	30	0D	0A	80..
00000060	24	50	D0	A1	6A	72	0D	0A	24	50	43	20	43	6F	6E	76	\$P-ijr..\$PC Conv
00000070	65	72	74	69	62	6C	65	0D	0A	24	56	65	72	73	69	6F	ertible..\$Versio
00000080	6E	20	4D	53	2D	44	4F	53	3A	20	20	2E	20	20	0D	0A	n MS-DOS: . .
00000090	24	53	65	72	69	61	6C	20	4E	75	6D	62	65	72	20	4F	\$Serial Number 0
000000A0	45	4D	3A	20	20	0D	0A	24	55	73	65	72	20	53	65	72	EM: ..\$User Ser
000000B0	69	61	6C	20	4E	75	6D	62	65	72	3A	20	20	20	20	20	ial Number:
000000C0	20	20	48	0D	0A	24	24	0F	3C	09	76	02	04	07	04	30	H..\$.<.v....0
000000D0	C3	51	8A	E0	E8	EF	FF	86	C4	B1	04	D2	E8	E8	E6	FF	Qèαφñ â-∞.πΦμ
000000E0	59	C3	53	8A	FC	E8	E9	FF	88	25	4F	88	05	4F	8A	C7	Y Sè"Φê ê%0ê.0è
000000F0	E8	DE	FF	88	25	4F	88	05	5B	C3	51	52	32	E4	33	D2	Φ ê%0ê.[QR233π
00000100	B9	0A	00	F7	F1	80	CA	30	88	14	4E	33	D2	3D	0A	00	..≈±Ç0ê.N3π=.
00000110	73	F1	3C	00	74	04	0C	30	88	04	5A	59	C3	B4	09	CD	s±<.t..0ê.ZY .=
00000120	21	C3	B8	00	F0	8E	C0	26	A0	FE	FF	3C	FF	74	1C	3C	! q.≡ÄL&á. < t.<
00000130	FE	74	1E	3C	FB	74	1A	3C	FC	74	1C	3C	FA	74	1E	3C	.t.<√t.<"t.<·t.<
00000140	F8	74	26	3C	FD	74	28	3C	F9	74	2A	BA	03	01	EB	2B	°t&<²t(<·t* ..δ+
00000150	90	BA	08	01	EB	25	90	BA	10	01	EB	1F	90	BA	15	01	É ..δ%É ..δ.É .
00000160	EB	19	90	BA	2B	01	EB	13	90	BA	4B	01	EB	0D	90	BA	δ.É +.δ.É K.δ.É
00000170	61	01	EB	07	90	BA	69	01	EB	01	90	E8	9F	FF	C3	B4	a.δ.É i.δ.ÉΦf
00000180	30	CD	21	50	BE	7A	01	83	C6	10	E8	6D	FF	58	8A	C4	0=!P z.â .Φm Xè-
00000190	83	C6	03	E8	64	FF	BA	7A	01	E8	81	FF	BE	91	01	83	â .Φd z.Φü æ.â
000001A0	C6	13	8A	C7	E8	53	FF	BA	91	01	E8	70	FF	BF	A8	01	.è ΦS æ.Φp i.
000001B0	83	C7	19	8B	C1	E8	2A	FF	8A	C3	E8	14	FF	83	EF	02	â .i Φ* è Φ. ân.
000001C0	89	05	BA	A8	01	E8	55	FF	C3	E8	56	FF	E8	B0	FF	32	ë. z.ΦU ΦV Φ\\ 2
000001D0	C0	B4	4C	CD	21	+											L =!

Рис. 4. - Шестнадцатеричное представление модуля .COM:

Шестнадцатеричное представление плохого модуля .EXE:

00000000	4D 5A D5 00 03 00 00 00	20 00 00 00 FF FF 00 00	MZ F..... ..
00000010	00 00 01 9F 00 01 00 00	1E 00 00 00 01 00 00 00	...f.....
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

Дальше идут нули до 300h строки

00000300	E9 C6 01 50 43 0D 0A 24	50 43 2F 58 54 0D 0A 24	⊖ .PC..\$PC/XT..\$
00000310	41 54 0D 0A 24 50 53 32	20 D0 BC D0 BE D0 B4 D0	AT..\$PS2
00000320	B5 D0 BB D1 8C 20 33 30	0D 0A 24 50 53 32 20 D0	î 30..\$PS2
00000330	BC D0 BE D0 B4 D0 B5 D0	BB D1 8C 20 35 30 20 D0	î î 50
00000340	B8 D0 BB D0 B8 20 36 30	0D 0A 24 50 53 32 20 D0	60..\$PS2
00000350	BC D0 BE D0 B4 D0 B5 D0	BB D1 8C 20 38 30 0D 0A	î î 80..
00000360	24 50 D0 A1 6A 72 0D 0A	24 50 43 20 43 6F 6E 76	\$P jr..\$PC Conv
00000370	65 72 74 69 62 6C 65 0D	0A 24 56 65 72 73 69 6F	ertible..\$Versio
00000380	6E 20 4D 53 2D 44 4F 53	3A 20 20 2E 20 20 0D 0A	n MS-DOS: . .
00000390	24 53 65 72 69 61 6C 20	4E 75 6D 62 65 72 20 4F	\$Serial Number 0
000003A0	45 4D 3A 20 20 0D 0A 24	55 73 65 72 20 53 65 72	EM: ..\$User Ser
000003B0	69 61 6C 20 4E 75 6D 62	65 72 3A 20 20 20 20 20	ial Number:
000003C0	20 20 48 0D 0A 24 24 0F	3C 09 76 02 04 07 04 30	H..\$.<.v....0
000003D0	C3 51 8A E0 E8 EF FF 86	C4 B1 04 D2 E8 E8 E6 FF	Qèαφη ā-тФФμ
000003E0	59 C3 53 8A FC E8 E9 FF	88 25 4F 88 05 4F 8A C7	Y Sè^nφê %Oê.Oè
000003F0	E8 DE FF 88 25 4F 88 05	5B C3 51 52 32 E4 33 D2	φ ê%Oê. [QR2Σ3т
00000400	B9 0A 00 F7 F1 80 CA 30	88 14 4E 33 D2 3D 0A 00	..≈±ç Oê.N3т=..
00000410	73 F1 3C 00 74 04 0C 30	88 04 5A 59 C3 B4 09 CD	s±<.t..Oê.ZY .=
00000420	21 C3 B8 00 F0 8E C0 26	A0 FE FF 3C FF 74 1C 3C	! η.≡Ä L&á▪ < t.<
00000430	FE 74 1E 3C FB 74 1A 3C	FC 74 1C 3C FA 74 1E 3C	▪t.<√t.<^nt.<·t.<
00000440	F8 74 26 3C FD 74 28 3C	F9 74 2A BA 03 01 EB 2B	°t&<^2t(<·t* ..δ+
00000450	90 BA 08 01 EB 25 90 BA	10 01 EB 1F 90 BA 15 01	É ..δ%É ..δ.É ..
00000460	EB 19 90 BA 2B 01 EB 13	90 BA 4B 01 EB 0D 90 BA	δ.É +.δ.É K.δ.É
00000470	61 01 EB 07 90 BA 69 01	EB 01 90 E8 9F FF C3 B4	a.δ.É i.δ.Éφf
00000480	30 CD 21 50 BE 7A 01 83	C6 10 E8 6D FF 58 8A C4	0= P↓z.â .φm Xè-
00000490	83 C6 03 E8 64 FF BA 7A	01 E8 81 FF BE 91 01 83	â .φd z.φü ↓æ.â
000004A0	C6 13 8A C7 E8 53 FF BA	91 01 E8 70 FF BF A8 01	.è φS æ.φp γι.
000004B0	83 C7 19 8B C1 E8 2A FF	8A C3 E8 14 FF 83 EF 02	â .ĩ↓φ* è φ. ân.
000004C0	89 05 BA A8 01 E8 55 FF	C3 E8 56 FF E8 B0 FF 32	ë. ι.φU φV φ\\ 2
000004D0	C0 B4 4C CD 21 +		└└L=!

Рис. 5. - Шестнадцатеричное представление плохого модуля .EXE

Шестнадцатеричное представление хорошего модуля .EXE:

00000000	4D 5A E7 00 03 00 01 00	20 00 00 00 FF FF 00 00	MZ τ..... ..
00000010	00 01 AE 23 03 01 1D 00	1E 00 00 00 01 00 07 01	..«#.....
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

00000300	50 43 0D 0A 24 50 43 2F	58 54 0D 0A 24 41 54 0D	PC..\$PC/XT..\$AT.
00000310	0A 24 50 53 32 20 D0 BC	D0 BE D0 B4 D0 B5 D0 BB	.\$PS2
00000320	D1 8C 20 33 30 0D 0A 24	50 53 32 20 D0 BC D0 BE	î 30..\$PS2
00000330	D0 B4 D0 B5 D0 BB D1 8C	20 35 30 20 D0 B8 D0 BB	î 50
00000340	D0 B8 20 36 30 0D 0A 24	50 53 32 20 D0 BC D0 BE	60..\$PS2
00000350	D0 B4 D0 B5 D0 BB D1 8C	20 38 30 0D 0A 24 50 D0	î 80..\$P
00000360	A1 6A 72 0D 0A 24 50 43	20 43 6F 6E 76 65 72 74	íjr..\$PC Convert
00000370	69 62 6C 65 0D 0A 24 56	65 72 73 69 6F 6E 20 4D	ible..\$Version M
00000380	53 2D 44 4F 53 3A 20 20	2E 20 20 0D 0A 24 53 65	S-DOS: . ..\$Se
00000390	72 69 61 6C 20 4E 75 6D	62 65 72 20 4F 45 4D 3A	rial Number OEM:
000003A0	20 20 0D 0A 24 55 73 65	72 20 53 65 72 69 61 6C	..\$User Serial
000003B0	20 4E 75 6D 62 65 72 3A	20 20 20 20 20 20 20 48	Number: H
000003C0	0D 0A 24 00 00 00 00 00	00 00 00 00 00 00 00 00	..\$.....
000003D0	24 0F 3C 09 76 02 04 07	04 30 C3 51 8A E0 E8 EF	\$.<.v....0 Qèαφη
000003E0	FF 86 C4 B1 04 D2 E8 E8	E6 FF 59 C3 53 8A FC E8	â-...Tφμ Y Sè^nφ
000003F0	E9 FF 88 25 4F 88 05 4F	8A C7 E8 DE FF 88 25 4F	θ ê%0ê.Oè φ ê%0
00000400	88 05 5B C3 51 52 32 E4	33 D2 B9 0A 00 F7 F1 80	ê. [QR2Σ3T...±Ç
00000410	CA 30 88 14 4E 33 D2 3D	0A 00 73 F1 3C 00 74 04	l0ê.N3T=..s±<.t.
00000420	0C 30 88 04 5A 59 C3 B4	09 CD 21 C3 B8 00 F0 8E	.0ê.ZY . .=! q.≡Ä
00000430	C0 26 A0 FE FF 3C FF 74	1C 3C FE 74 1E 3C FB 74	L&á. < t.<.t.<√t
00000440	1A 3C FC 74 1C 3C FA 74	1E 3C F8 74 26 3C FD 74	.<^t.<·t.<°t&<^2t
00000450	28 3C F9 74 2A BA 00 00	EB 2B 90 BA 05 00 EB 25	(<·t* ..δ+É ..δ%
00000460	90 BA 0D 00 EB 1F 90 BA	12 00 EB 19 90 BA 28 00	É ..δ.É ..δ.É (.δ.É H.δ.É ^..δ.É
00000470	EB 13 90 BA 48 00 EB 0D	90 BA 5E 00 EB 07 90 BA	f.δ.Éφf θ= Pw
00000480	66 00 EB 01 90 E8 9F FF	C3 B4 30 CD 21 50 BE 77	.â .φm Xè-â .φd
00000490	00 83 C6 10 E8 6D FF 58	8A C4 83 C6 03 E8 64 FF	w.φü Ä.â .è φS
000004A0	BA 77 00 E8 81 FF BE 8E	00 83 C6 13 8A C7 E8 53	Ä.φp γÑ.â .ï-φ
000004B0	FF BA 8E 00 E8 70 FF BF	A5 00 83 C7 19 8B C1 E8	* è φ. ân.ë. Ñ.φ
000004C0	2A FF 8A C3 E8 14 FF 83	EF 02 89 05 BA A5 00 E8	U Lpγ..Ä+φN φz
000004D0	55 FF C3 2B C0 50 B8 10	00 8E D8 E8 4E FF E8 A8	2 L=!
000004E0	FF 32 C0 B4 4C CD 21	+	

Рис. 6. - Шестнадцатеричное представление хорошего модуля .EXE

«Отличие форматов файлов COM и EXE модулей»

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

В данном файле код, данные и стек находятся в одном сегменте. Код и данные начинаются с адреса 0h (См. Рис. 4).

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

В «плохом» EXE файле код, данные и стек находятся в одном сегменте. Код и данные начинаются с адреса 300h. С адреса 0h находится управляющая информация загрузчика, которая содержит заголовок и таблицу настроек. (См. Рис. 5-6)

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

У «хорошего» EXE код, данные и стек находятся в разных сегментах, а в «плохом» - в одном сегменте. С адреса 0 в «хорошем» EXE располагается валидная таблица настроек, в отличие от «плохого» EXE. У «хорошего» EXE выделяется память под стек между PSP и кодом.

Шаг 5:

«Загрузка COM модуля в основную память»

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

В начале определяется сегментный адрес участка ОП, способного вместить загрузку программы, затем создается блок памяти для PSP и программы. После считывания COM-файл помещается в память с 100h. После сегментные регистры устанавливаются в начало PSP. SP устанавливается в конец PSP, 0000h помещается в стек, а в IP записывается 100h.

Код располагается с адреса 100h.

AX 0000	SI 0000	CS 19F5	IP 0100	Stack +0 0000	Flags 7202
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0287	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 0 0 0 0

CMD >				<div>1012000</div>															
-------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Рис. 9. - .COM в отладчике

2) Что располагается с адреса 0?

С адреса 0 располагается PSP размером в 100h байт.

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

Регистры DS, ES, CS, SS указывают на начало блока PSP.

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

Стек генерируется автоматически. Регистр SS указывает на начало блока PSP, а SP на конец стека. Стек расположен между адресами SS:0000h – SS:FFFFh и заполняется с конца модуля в сторону уменьшения адресов.

Шаг 6:

«Загрузка «хорошего» EXE модуля в основную память»

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

Данный EXE загружается со считыванием информации заголовка EXE, выполняется перемещение адресов сегментов, ES и DS устанавливаются в начало PSP, SS – на начало сегмента стека, а CS – на начало сегмента команд. В IP загружается смещение точки входа в программу.

AX 0000	SI 0000	CS 1A2B	IP 0119	Stack +0 7954	Flags 7202
BX 0000	DI 0000	DS 19F5		+2 6570	
CX 038A	BP 0000	ES 19F5	HS 19F5	+4 6F20	OF DF IF SF ZF AF PF CF
DX 0000	SP 0100	SS 1A05	FS 19F5	+6 2066	0 0 1 0 0 0 0 0

CMD >	1	0	1	2	3	4	5	6	7
	DS:0000	CD	20	FF	9F	00	EA	F0	FE
	DS:0008	AD	DE	1B	05	C5	06	00	00
	DS:0010	1B	01	10	01	18	01	92	01
	DS:0018	01	01	01	00	02	FF	FF	FF
	DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
	DS:0028	FF	FF	FF	FF	FF	FF	FF	FF
	DS:0030	A2	01	14	00	18	00	F5	19
	DS:0038	FF	FF	FF	FF	00	00	00	00
	DS:0040	05	00	00	00	00	00	00	00
	DS:0048	00	00	00	00	00	00	00	00

2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00
DS:0010	1B	01	10	01	18	01	92	01	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

= f.Ω≡ i |..†...

.....ff.

δ.L.

6.....J.

.....

Рис. 10. - .EXE в отладчике

2) На что указывают регистры DS и ES?

ES и DS указывают на начало сегмента PSP.

3) Как определяется стек?

Стек определяется на основе директивы `.stack` с указанием размера стека. SS указывает на начало сегмента стека, а SP указывает на конец.

4) Как определяется точка входа?

Точка входа определяется параметром после директивы `END`.

Заключение.

В результате выполнения лабораторной работы были изучены структурные отличия `.COM` и `.EXE` модулей и получены навыки работы с отладчиком `TD.EXE`.