

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Организация ЭВМ и систем»
Тема: «Трансляции, отладка и выполнение программ
на языке Ассемблера»

Студент гр. 9381

Авдеев Илья

Преподаватель

Ефремеов 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.

Ход работы

1. На основе шаблона был написан текст исходного .COM модуля, который определяет тип PC и версию системы. Были получены “хороший”. COM модуль и “плохой”. EXE модуль. Использовался компилятор MASM. При тестировании и ответах на вопросы также использовался TASM.

2. Был написан код программы, получен и отлажен “хороший”.

3. Структуры полученных загрузочных модулей были сравнены и проанализированы при помощи отладчика TD.exe и менеджера Far, сделаны необходимые выводы, даны ответы на поставленные вопросы.

Названия процедур	Назначение
TETR_TO_HEX	Перевод десятичной цифры в код символа.
BYTE_TO_HEX	Перевод байта в 16-ной с/с в символичный код
WRD_TO_HEX	Перевод слова в 16-ной с/с в символичный код
BYTE_TO_DEC	Перевод байта в 16-ной с/с в символичный код в 10-ной с/с
WRITE	Вывод строки
GET_PC_TYPE	Вывод типа PC
GET_VERSRION	Вывод версии, OEM и серийного номера пользователя

Последовательность действий программы

Вызывается процедура GET_PC_TYPE, которая выводит тип ПК. Её действие основано на считывании предпоследнего байта ROM BIOS и определении в зависимости от его значения типа ПК по спец. таблице. Если же соответствие не найдено, то выводится сообщение с 16-ричным кодом, который записывается в начало этого сообщения благодаря встроенной процедуре BYTE_TO_HEX.

Далее при помощи процедуры GET_VERSION выводится версия системы (задействована BYTE_TO_DEC), OEM (задействована BYTE_TO_DEC), и серийный номер пользователя (задействованы WRD_TO_HEX и BYTE_TO_HEX).

Программа завершается.

Ответы на вопросы.

Отличия исходных текстов COM и EXE программ

1. Сколько сегментов должна содержать COM-программа?

Только один

2. EXE программа?

Один и более. В зависимости от выбранной модели памяти.

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

ORG 100h, поскольку в начале COM-программы определен 256-байтовый PSP (префикс программного сегмента), а значит необходимо обеспечить смещение на 100h байт от начала.

ASSUME необходима для проверки допустимости каждого обращения к именованной ячейке памяти с учетом значения текущего сегментного регистра. Кроме того, требуется использования директивы **END** для завершения программы.

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

Нет.

COM-программа подразумевает наличие **лишь одного сегмента**, а значит, можно использовать только **near**-переходы, так как в **far**-переходах подразумевается использование нескольких сегментов.

В COM-программах в DOS не содержится таблицы настройки, которая содержит описание адресов, зависящих от размещения загрузочного модуля в оперативной памяти, поэтому нельзя использовать команды, связанные с адресом сегмента (адрес сегмента до загрузки неизвестен). Отсюда вытекает, что нельзя использовать, например, оператор **SEG NAME**, дающий доступ к началу сегмента **NAME**

Отличия форматов файлов COM и EXE модулей

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

COM-файл состоит из команд, процедур и данных, используемых в программе. Код (и данные) начинается с **нулевого** адреса. Код, данные и стек располагаются в одном сегменте.

0000000000: E9 FC 01 50 43 20 74 79	70 65 3A 20 24 50 43 0D	éüPC type: \$PC
0000000010: 0A 24 50 43 2F 58 54 0D	0A 24 41 54 0D 0A 24 50	\$PC/XT\$AT\$P
0000000020: 53 32 20 6D 6F 64 65 6C	20 33 30 0D 0A 24 50 53	S2 model 30\$PS
0000000030: 32 20 6D 6F 64 65 6C 20	35 30 20 6F 72 20 36 30	2 model 50 or 60
0000000040: 0D 0A 24 50 53 32 20 6D	6F 64 65 6C 20 38 30 0D	\$PS2 model 80
0000000050: 0A 24 50 43 6A 72 0D 0A	24 50 43 20 43 6F 6E 76	\$PCjr.\$PC Conv
0000000060: 65 72 74 69 62 6C 65 0D	0A 24 20 20 0D 0A 24 4F	ertible.\$ \$O
0000000070: 53 20 76 65 72 73 69 6F	6E 3A 20 24 20 20 2E 20	S version: \$.
0000000080: 20 0D 0A 24 4F 45 4D 20	73 65 72 69 61 6C 20 6E	\$OEM serial n
0000000090: 75 6D 62 65 72 3A 20 24	20 20 0D 0A 24 55 73 65	umber: \$ \$Use
00000000A0: 72 20 73 65 72 69 61 6C	20 6E 75 6D 62 65 72 3A	r serial number:
00000000B0: 20 24 20 20 20 20 20 20	0D 0A 24 24 0F 3C 09 76	\$ \$<ov
00000000C0: 02 04 07 04 30 C3 51 8A	E0 E8 EF FF 86 C4 B1 04	0♦♦♦AQŠàèiÿ†Ä±♦
00000000D0: D2 E8 E8 E6 FF 59 C3 53	8A FC E8 E9 FF 88 25 4F	0èèæÿYÄSŠüèéÿ~%O
00000000E0: 88 05 4F 8A C7 E8 DE FF	88 25 4F 88 05 5B C3 51	^+OŠÇèbÿ~%O^+ [ÄQ
00000000F0: 52 32 E4 33 D2 B9 0A 00	F7 F1 80 CA 30 88 14 4E	R2ä30¹ ÷ñ€Ê0^JN
0000000100: 33 D2 3D 0A 00 73 F1 3C	00 74 04 0C 30 88 04 5A	30= sñ< t♦90^♦Z
0000000110: 59 C3 50 B4 09 CD 21 58	C3 50 53 52 06 B8 00 F0	YÄP´oÍ!XÄPSR±, ð
0000000120: 8E C0 26 A0 FE FF BA 03	01 E8 E6 FF 3C FF 74 23	ŽÄ& þÿ°♥0èæÿ<ÿt#
0000000130: 3C FE 74 25 3C FD 74 21	3C FC 74 23 3C FA 74 25	<þt%<ÿt!<üt#<út%
0000000140: 3C FC 74 27 3C F8 74 29	3C FD 74 2B 3C F9 74 2D	<üt'<øt)<ÿt+<üt-
0000000150: EB 31 90 BA 0D 01 EB 38	90 BA 12 01 EB 32 90 BA	ë10° 0è80°†0è20°
0000000160: 1A 01 EB 2C 90 BA 1F 01	EB 26 90 BA 2E 01 EB 20	→0ë, 0°▼0ë&0° .0ë
0000000170: 90 BA 43 01 EB 1A 90 BA	52 01 EB 14 90 BA 59 01	0°C0ë→0°R0ëJ0°Y0
0000000180: EB 0E 90 E8 40 FF BB 6A	01 88 07 88 67 01 8B D3	ëJ0è@ÿ»j0^•^g0<Ó
0000000190: E8 7F FF 07 5A 5B 58 C3	50 53 51 52 56 57 B4 30	è∆ÿ•Z[XÄPSQRVW´0
00000001A0: CD 21 BE 7D 01 8A D4 E8	45 FF 8A C2 83 C6 03 E8	Í!%}0Š0èEÿŠÄfA♥è
00000001B0: 3D FF BA 6F 01 E8 5A FF	BA 7C 01 E8 54 FF 8A C7	=ÿ°o0èZÿ° 0èTÿŠÇ
00000001C0: E8 03 FF BF 98 01 88 05	88 65 01 BA 84 01 E8 41	è♥ÿ¿~0^+^e0°„0èA
00000001D0: FF BA 98 01 E8 3B FF 8A	C3 E8 EA FE BF B2 01 88	ÿ°~0è;ÿŠÄèèþ¿²0^
00000001E0: 05 88 65 01 8B C1 83 C7	05 E8 EB FE BA 9D 01 E8	+^e0<ÁfÇ+èèþ°00è
00000001F0: 20 FF BA B2 01 E8 1A FF	5F 5E 5A 59 5B 58 C3 E8	ÿ°²0è→ÿ_^ZY[XÄè
0000000200: 17 FF E8 93 FF 32 C0 B4	4C CD 21	‡ÿè“ÿ2À´LÍ!

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

Что располагается с адреса 0?

В хорошем EXE-файле код, данные и стек находятся в различных сегментах, а в плохом – в одном и том же сегменте. По мимо этого, изменен порядок расположения сегментов в памяти, и стек имеет другой размер – 256 байт (100h). Файл в 16-ичном представлении расположен снизу.

0000000000: 4D 5A 0B 01 03 00 00 00	20 00 00 00 FF FF 00 00	MZd0♥	ÿÿ
0000000010: 00 00 59 A4 00 01 00 00	1E 00 00 00 01 00 00 00	YH 0 ▲ 0	
0000000020: 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00		

00000002F0:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
0000000300:	E9 FC 01 50 43 20 74 79	70 65 3A 20 24 50 43 0D	éü@PC type: \$PC
0000000310:	0A 24 50 43 2F 58 54 0D	0A 24 41 54 0D 0A 24 50	\$PC/XT\$AT\$P
0000000320:	53 32 20 6D 6F 64 65 6C	20 33 30 0D 0A 24 50 53	S2 model 30\$PS
0000000330:	32 20 6D 6F 64 65 6C 20	35 30 20 6F 72 20 36 30	2 model 50 or 60
0000000340:	0D 0A 24 50 53 32 20 6D	6F 64 65 6C 20 38 30 0D	\$PS2 model 80
0000000350:	0A 24 50 43 6A 72 0D 0A	24 50 43 20 43 6F 6E 76	\$PCjr\$PC Conv
0000000360:	65 72 74 69 62 6C 65 0D	0A 24 20 20 0D 0A 24 4F	ertible\$ \$0
0000000370:	53 20 76 65 72 73 69 6F	6E 3A 20 24 20 20 2E 20	S version: \$.
0000000380:	20 0D 0A 24 4F 45 4D 20	73 65 72 69 61 6C 20 6E	\$OEM serial n
0000000390:	75 6D 62 65 72 3A 20 24	20 20 0D 0A 24 55 73 65	umber: \$ \$Use
00000003A0:	72 20 73 65 72 69 61 6C	20 6E 75 6D 62 65 72 3A	r serial number:
00000003B0:	20 24 20 20 20 20 20 20	0D 0A 24 24 0F 3C 09 76	\$ \$<ov
00000003C0:	02 04 07 04 30 C3 51 8A	E0 E8 EF FF 86 C4 B1 04	♦♦♦♦QŠàèÿ†Ä±♦
00000003D0:	D2 E8 E8 E6 FF 59 C3 53	8A FC E8 E9 FF 88 25 4F	òèèæÿYÄSŠüèéÿ~%0
00000003E0:	88 05 4F 8A C7 E8 DE FF	88 25 4F 88 05 5B C3 51	^+0ŠÇèþÿ~%0^+[ÄQ
00000003F0:	52 32 E4 33 D2 B9 0A 00	F7 F1 80 CA 30 88 14 4E	R2ä3Ô¹ ÷ñ€ÊÔ~ĴN
0000000400:	33 D2 3D 0A 00 73 F1 3C	00 74 04 0C 30 88 04 5A	3Ô= sñ< t♦♀0^♦Z
0000000410:	59 C3 50 B4 09 CD 21 58	C3 50 53 52 06 B8 00 F0	YÄP´oÍ!XÄPSR♠. š
0000000420:	8E C0 26 A0 FE FF BA 03	01 E8 E6 FF 3C FF 74 23	Ž& þÿ°♥èæÿ<ÿt#
0000000430:	3C FE 74 25 3C FD 74 21	3C FC 74 23 3C FA 74 25	<þt%<ÿt!<üt#<út%
0000000440:	3C FC 74 27 3C F8 74 29	3C FD 74 2B 3C F9 74 2D	<üt'<øt)<ÿt+<üt-
0000000450:	EB 31 90 BA 0D 01 EB 38	90 BA 12 01 EB 32 90 BA	ë1®®®è8®®®4®è2®®
0000000460:	1A 01 EB 2C 90 BA 1F 01	EB 26 90 BA 2E 01 EB 20	→0ë,®®®♥0ë&®®.0ë
0000000470:	90 BA 43 01 EB 1A 90 BA	52 01 EB 14 90 BA 59 01	®®C0ë→®®R0ëĴ®®Y0
0000000480:	EB 0E 90 E8 40 FF BB 6A	01 88 07 88 67 01 8B D3	ěĴ®è@ÿ»j0^•^Ĵg0<Ó
0000000490:	E8 7F FF 07 5A 5B 58 C3	50 53 51 52 56 57 B4 30	èøÿ•Z[XÄPSQRVW´0
00000004A0:	CD 21 BE 7D 01 8A D4 E8	45 FF 8A C2 83 C6 03 E8	Í!¼}0ŠÔèEÿŠÄf♥è
00000004B0:	3D FF BA 6F 01 E8 5A FF	BA 7C 01 E8 54 FF 8A C7	=ÿ°o0èZÿø 0èTÿŠÇ
00000004C0:	E8 03 FF BF 98 01 88 05	88 65 01 BA 84 01 E8 41	è♥ÿ¿~0^+^e0°„0èA
00000004D0:	FF BA 98 01 E8 3B FF 8A	C3 E8 EA FE BF B2 01 88	ÿ°~0è;ÿŠÄèèþ¿²0^
00000004E0:	05 88 65 01 8B C1 83 C7	05 E8 EB FE BA 9D 01 E8	♠^e0<ÁfÇ♠èèþ°®0è
00000004F0:	20 FF BA B2 01 E8 1A FF	5F 5E 5A 59 5B 58 C3 E8	ÿ°²0è→ÿ_^ZY[XÄè
0000000500:	17 FF E8 93 FF 32 C0 B4	4C CD 21	‡ÿè“ÿ2Ä´LÍ!

Данные и код расположены в одном сегменте. Данные и код начинаются с адреса 300h, с адреса 0h располагается заголовок и таблица настройки адресов.

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

0000000000:	4D 5A 20 00 03 00 01 00	20 00 10 00 FF FF 22 00	MZ ♥ @ ▶ ÿÿ"
0000000010:	00 01 2C 61 10 00 00 00	1E 00 00 00 01 00 58 01	@,a▶ ▲ @ X0
0000000020:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	

0000000200:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	éD@ \$α<ov0♦♦♦0ÃQŠ
0000000210:	E9 44 01 24 0F 3C 09 76	02 04 07 04 30 C3 51 8A	àèïÿ†Ä±♦0èèæÿYÃS
0000000220:	E0 E8 EF FF 86 C4 B1 04	D2 E8 E8 E6 FF 59 C3 53	Šüèéÿ~%0^+0ŠÇèbÿ
0000000230:	8A FC E8 E9 FF 88 25 4F	88 05 4F 8A C7 E8 DE FF	~%0^+ [ÃQR2ä30¹
0000000240:	88 25 4F 88 05 5B C3 51	52 32 E4 33 D2 B9 0A 00	÷ñ€Ê0^ĴN30= sñ<
0000000250:	F7 F1 80 CA 30 88 14 4E	33 D2 3D 0A 00 73 F1 3C	t♦Q0^♦ZYÄP´oÍ!X
0000000260:	00 74 04 0C 30 88 04 5A	59 C3 50 B4 09 CD 21 58	ÄPSR♣. ðŽÀ& þÿ
0000000270:	C3 50 53 52 06 B8 00 F0	8E C0 26 A0 FE FF BA 08	èæÿ<ÿt#<pt%<ÿt!
0000000280:	00 E8 E6 FF 3C FF 74 23	3C FE 74 25 3C FD 74 21	<üt#<üt%<üt'<øt)
0000000290:	3C FC 74 23 3C FA 74 25	3C FC 74 27 3C F8 74 29	<ÿt+<üt-ë100± ë8
00000002A0:	3C FD 74 2B 3C F9 74 2D	EB 31 90 BA 12 00 EB 38	00± ë200▼ ë,00\$
00000002B0:	90 BA 17 00 EB 32 90 BA	1F 00 EB 2C 90 BA 24 00	ë&003 ë 00H ë→00
00000002C0:	EB 26 90 BA 33 00 EB 20	90 BA 48 00 EB 1A 90 BA	W ëĴ00^ ëĴ00è@ÿ»o
00000002D0:	57 00 EB 14 90 BA 5E 00	EB 0E 90 E8 40 FF BB 6F	^•^g0<0èαÿ•Z[XÃ
00000002E0:	00 88 07 88 67 01 8B D3	E8 7F FF 07 5A 5B 58 C3	PSQRVW´0Í!%, Š0è
00000002F0:	50 53 51 52 56 57 B4 30	CD 21 BE 82 00 8A D4 E8	EÿŠÄfA♥è=ÿøt èZÿ
0000000300:	45 FF 8A C2 83 C6 03 E8	3D FF BA 74 00 E8 5A FF	00 èTÿŠÇè♥ÿ;0 ^+
0000000310:	BA 81 00 E8 54 FF 8A C7	E8 03 FF BF 9D 00 88 05	^e0%0 èAÿ00 è;ÿŠ
0000000320:	88 65 01 BA 89 00 E8 41	FF BA 9D 00 E8 3B FF 8A	Äèèþ;• ^+^e0<ÁfÇ
0000000330:	C3 E8 EA FE BF B7 00 88	05 88 65 01 8B C1 83 C7	♣èèþ0¢ è ÿ0• è→ÿ
0000000340:	05 E8 EB FE BA A2 00 E8	20 FF BA B7 00 E8 1A FF	_^ZY[XÃ,= Ž0è†ÿè
0000000350:	5F 5E 5A 59 5B 58 C3 B8	16 00 8E D8 E8 12 FF E8	Žÿ2Ä´LÍ!PC type:
0000000360:	8E FF 32 C0 B4 4C CD 21	50 43 20 74 79 70 65 3A	\$PC♣\$PC/XT♣\$A
0000000370:	20 24 50 43 0D 0A 24 50	43 2F 58 54 0D 0A 24 41	T♣\$PS2 model 30
0000000380:	54 0D 0A 24 50 53 32 20	6D 6F 64 65 6C 20 33 30	♣\$PS2 model 50
0000000390:	0D 0A 24 50 53 32 20 6D	6F 64 65 6C 20 35 30 20	or 60♣\$PS2 mode
00000003A0:	6F 72 20 36 30 0D 0A 24	50 53 32 20 6D 6F 64 65	l 80♣\$PCjr♣\$PC
00000003B0:	6C 20 38 30 0D 0A 24 50	43 6A 72 0D 0A 24 50 43	Convertible♣\$
00000003C0:	20 43 6F 6E 76 65 72 74	69 62 6C 65 0D 0A 24 20	♣\$OS version:
00000003D0:	20 0D 0A 24 4F 53 20 76	65 72 73 69 6F 6E 3A 20	\$. ♣\$OEM ser
00000003E0:	24 20 20 2E 20 20 0D 0A	24 4F 45 4D 20 73 65 72	ial number: \$ ♣
00000003F0:	69 61 6C 20 6E 75 6D 62	65 72 3A 20 24 20 20 0D	\$User serial nu
0000000400:	0A 24 55 73 65 72 20 73	65 72 69 61 6C 20 6E 75	mber: \$ ♣\$
0000000410:	6D 62 65 72 3A 20 24 20	20 20 20 20 20 0D 0A 24	

Структура хорошего EXE: стек, дата, код – отдельные сегменты, в отличие от плохого, в котором один они объединены в один сегмент.

Хороший EXE-файл, в отличие от плохого, не содержит директивы ORG 100h, выделяющей память под PSP.

Загрузка COM модуля в основную память

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

При загрузке COM-файла в память DOS занимает первые 256 байт (100h) блоком данных PSP и располагает код программы только после этого блока.

Поэтому код располагается с адреса 100h. После загрузки COM-программы в память сегментные регистры указывают на начало PSP.

The screenshot shows the DOSBox 0.74-3 interface. At the top, the CPU speed is set to 3000 cycles. Below the title bar, the CPU registers are displayed: AX=0000, SI=0000, CS=119C, IP=0100, Stack=+0 0000, FLAGS=0200; BX=0000, DI=0000, DS=119C, +2 0000; CX=0202, BP=0000, ES=119C, HS=119C, +4 0000; DX=0000, SP=FFF5, SS=119C, FS=119C, +6 0000. The flag bits are: OF=0, DF=0, IF=1, SF=0, ZF=0, AF=0, PF=0, CF=0. The command line shows 'CMD >'. Below the command line, the instruction pointer (IP) is 0100, and the instruction is 'E9F301 JMP 02F6'. The memory dump shows the first 16 bytes of memory starting at address 0000: DS:0000 CD 20 CC 46 00 EA FD FF; DS:0008 AD DE ED 04 92 01 00 00; DS:0010 18 01 10 01 18 01 92 01; DS:0018 03 FF FF FF FF FF FF FF; DS:0020 FF FF FF FF FF FF FF FF; DS:0028 FF FF FF FF 96 11 C4 FF; DS:0030 92 01 14 00 18 00 9C 11; 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. The bottom of the screen shows the DOSBox menu: 1 Step, 2 StepProc, 3 Retrieve, 4 Help, 5 Set BRK, 6, 7 up, 8 dn, 9 le, 0 ri.

2. Что располагается с 0 адреса?

С адреса 0 располагается префикс программного сегмента (PSP).

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

Сегментные регистры имеют значения, соответствующие началу сегмента, в который модуль был помещен управляющей программой. Все

сегментные регистры имеют значения 119С, т.к. указывают на один и тот же

CS	119C
DS	119C
ES	119C
SS	119C

сегмент памяти - PSP.

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

- 1.Стек создается автоматически.
- 2.Регистр ss (сегментный регистр стека) указывает на начало PSP (0h).
- 3.Регистр sp указывает на вершину стека – FFFЕh. Он занимает оставшуюся память, адреса изменяются от больших к меньшим, то есть от FFFЕh к 0000h.

План загрузки модуля .COM в основную память

- 1.В основной памяти выделяется свободный сегмент.
- 2.Первые 100h байт уходят под PSP, а в оставшаяся область выделенного сегмента под саму программу.

Загрузка «хорошего» EXE модуля в память

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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 0000	SI 0000	CS 11AC	IP 0100	Stack +0 0000	FLAGS 0200
BX 0000	DI 0000	DS 119C		+2 0000	
CX 0302	BP 0000	ES 119C	HS 119C	+4 0000	OF DF IF SF ZF AF PF CF
DX 0000	SP 0000	SS 11AC	FS 119C	+6 0000	0 0 1 0 0 0 0 0

CMD >

0100 E9F301	JMP	02F6	DS:0000	CD 20 38 23 00 EA FD FF
0103 7479	JZ	017E	DS:0008	AD DE ED 04 92 01 00 00
0105 7065	JO	016C	DS:0010	18 01 10 01 18 01 92 01
0107 206F66	AND	[BX+66],CH	DS:0018	04 FF FF FF FF FF FF FF
010A 205043	AND	[BX+SI+43],DL	DS:0020	FF FF FF FF FF FF FF FF
010D 202D	AND	[DI],CH	DS:0028	FF FF FF FF FF 96 11 C4 FF
010F 2024	AND	[SI],AH	DS:0030	92 01 14 00 18 00 9C 11
0111 50	PUSH	AX	DS:0038	FF 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

DS:0000	CD 20 38 23 00 EA FD FF	AD DE ED 04 92 01 00 00	8#.....
DS:0010	18 01 10 01 18 01 92 01	04 FF FF FF FF FF FF FF
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF FF 96 11 C4 FF
DS:0030	92 01 14 00 18 00 9C 11	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

1 Step 2 StepProc 3 Retrieve 4 Help 5 Set BRK 6 7 up 8 dn 9 le 0 ri

Считывается информация из заголовков EXE, в IP загружается смещение точки входа. Регистры DS и ES указывают на начало блока PSP, регистр CS указывает на начало сегмента кода, а регистр SS – на начало сегмента стека.

2. На что указывают регистры DS и ES?

На начало блока PSP

3. Как определяется стек?

Регистр SS указывает на начало сегмента стека, а SP – на конец сегмента стека.

4. Как определяется точка входа?

Точка входа определяется параметром после директивы END, в качестве которого нужно передать метку, с которой программа начнет выполнение команд.

Заключение

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