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. 0381 _____ Ионина К.С.

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.

Задание.

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 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, сформировать 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 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.

Выполнение работы.

Код исходного .COM модуля написан в файле lab1com.asm. В первую очередь в файле прописываются строки для вывода сообщений о типах PC, серийном номере OEM_NUM и серийном номере пользователя. Процедура WRITE создана для вывода данных сообщений.

Тип PC определяется с помощью процедуры TYPE_OF_PC. Код системы записан в байте, который сохраняется в AL. Затем происходит сравнение данного значения с кодами из таблицы 1, после - переход к соответствующей метке. В ней в DX заносится смещение нужного сообщения. В завершении в метке write_this происходит вызов процедуры WRITE, которая печатает сообщения на экран.

Процедура VERS определяет версию системы, серийный номер OEM_NUM и номер пользователя. Функция 30h прерывания 21h возвращает требуемую информацию. Далее выводятся сообщения про тип PC, серийный номер OEM и серийный номер пользователя.

Применив команду masm lab1com.asm был собран объектный файл lab1com.obj, далее командой link lab1com.obj собрался «плохой» .EXE-модуль. Результат запуска lab1com.exe - на рисунке 1.

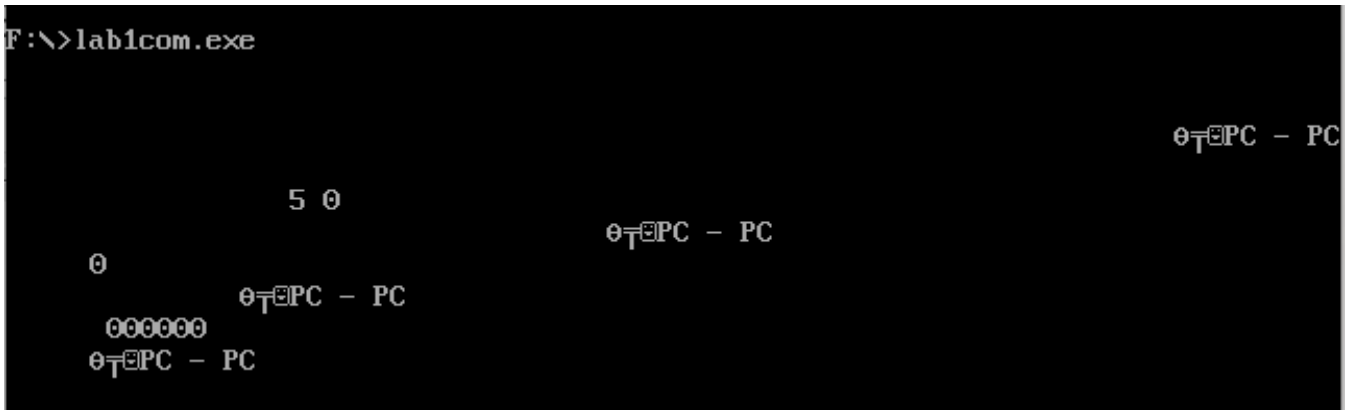


Рисунок 1. Вывод модуля lab1com.exe

Применив команду exe2bin lab1com.exe lab1com.com был получен .COM-модуль. Результат запуска lab1com.com представлен на рисунке 2.

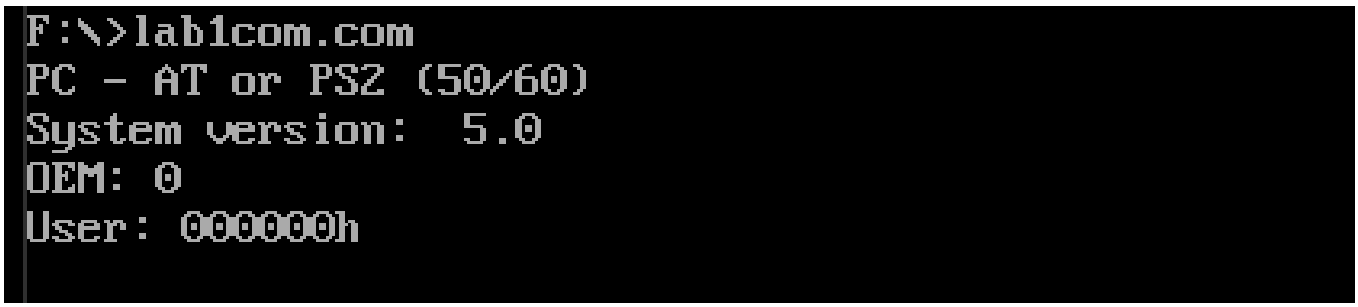


Рисунок 2. Вывод модуля lab1com.com

Код «хорошего» .EXE модуля находится в файле lab1exe.asm. Для его записи из файла lab1com.asm была извлечена вся информация, но с некоторыми изменениями: в сегмент данных вынесены строки сообщений, добавлены определения сегмента стека и данных, а также код, из которого вызывались процедуры TYPE_OF_PC и VERS вынесен в добавленную дальнюю процедуру MAIN, в ней также присутствует загрузка адреса сегмента данных.

После сборки и запуска lab1exe.exe выводятся верные сообщения.

Рисунок 3. Вывод модуля lab1exe.exe

Выводы.

В ходе выполнения лабораторной работы были исследованы различия в структурах исходных текстов модулей .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

Контрольные вопросы по лабораторной работе №1.

Отличия исходных текстов COM и EXE программ.

1) *Сколько сегментов должна содержать COM-программа?*

COM-программа должна содержать ровно один сегмент, который содержит в себе данные и код. При этом стек генерируется автоматически.

2) *EXE-программа?*

EXE-программа должна содержать по крайней мере один сегмент (сегмент кода). Вдобавок она может содержать сегменты стека и данных. Если сегмент стека не был задан, то будет использован стек DOS. Данные должны бать вынесены в отдельный сегмент.

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

В тексте COM-программы обязательно должна быть:

- Директива ORG 100h - пропускает первые 256 байт сегмента для размещения в них префикса программного сегмента PSP (отсутствие данной директивы оставляет возможным сборку и запуск модуля, но вывод будет неверным).
- Директива ASSUME – указывает, что данный сегмент будет использоваться в качестве сегмента кода и сегмента данных.

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

Не все форматы команд можно использовать в COM-программе. Команды, операндами которых являются сегменты, не могут быть выполнены, т.к. в COM-модулях отсутствует заголовок, в котором содержится таблица настройки (relocation table). По ней осуществляется поиск абсолютных адресов сегмента.

Отличия форматов файлов COM и EXE модулей.

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

Файл COM состоит из одного сегмента – сегмента кода, который содержит код и данные. COM-файл ограничен размером одного сегмента и не превышает 64 Кб. Код располагается с адреса 0h, но при запуске модуля устанавливается

смещение в 100h, так как в COM модулях используется директива 100h для выделения 256 байт под PSP.

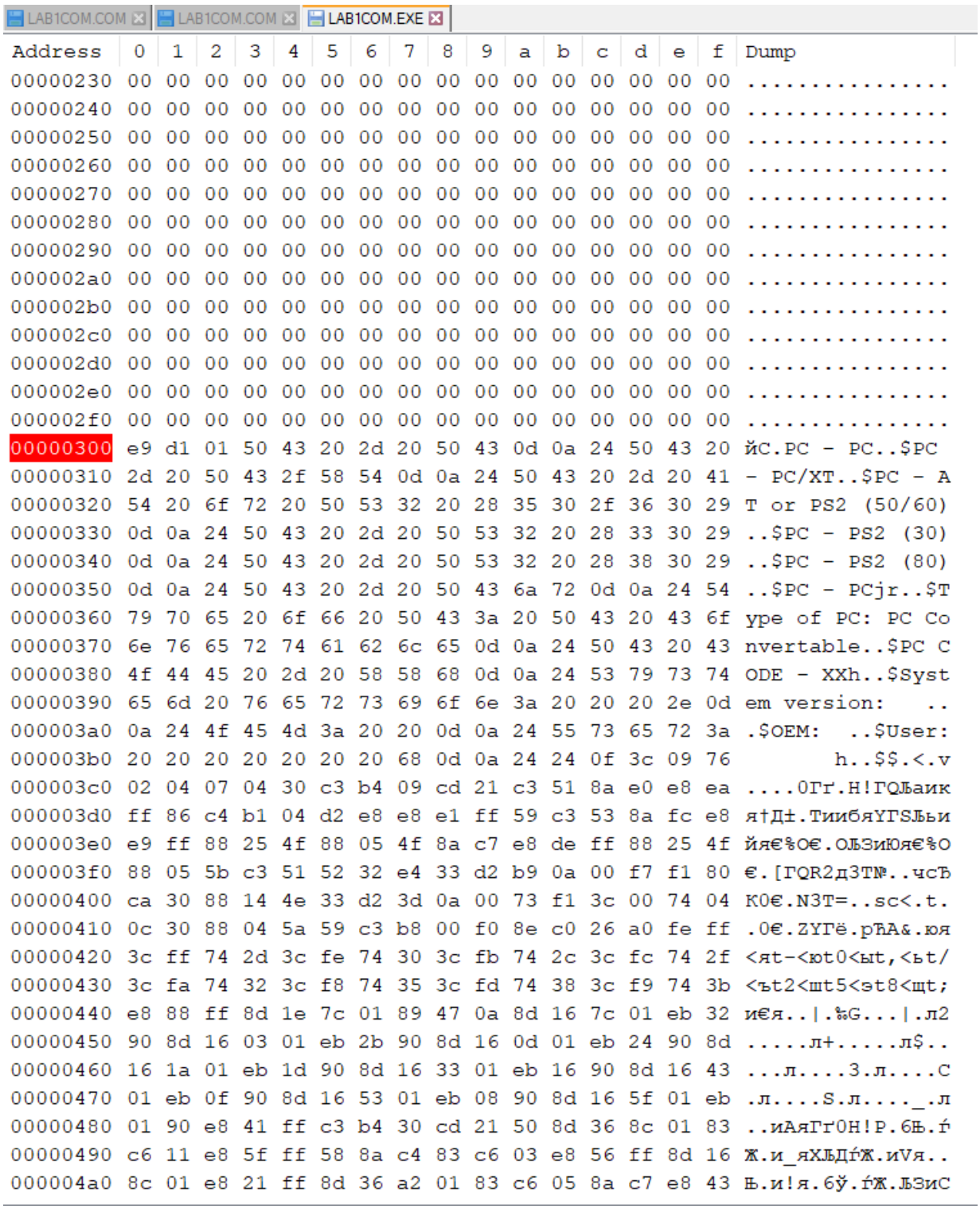
LAB1COM.COM LAB1COM.COM

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
00000000	e9	d1	01	50	43	20	2d	20	50	43	0d	0a	24	50	43	20	йС.PC - PC..$PC
00000010	2d	20	50	43	2f	58	54	0d	0a	24	50	43	20	2d	20	41	- PC/XT..$PC - A
00000020	54	20	6f	72	20	50	53	32	20	28	35	30	2f	36	30	29	T or PS2 (50/60)
00000030	0d	0a	24	50	43	20	2d	20	50	53	32	20	28	33	30	29	..$PC - PS2 (30)
00000040	0d	0a	24	50	43	20	2d	20	50	53	32	20	28	38	30	29	..$PC - PS2 (80)
00000050	0d	0a	24	50	43	20	2d	20	50	43	6a	72	0d	0a	24	54	..$PC - PCjr..$T
00000060	79	70	65	20	6f	66	20	50	43	3a	20	50	43	20	43	6f	ype of PC: PC Co
00000070	6e	76	65	72	74	61	62	6c	65	0d	0a	24	50	43	20	43	nvertable..$PC C
00000080	4f	44	45	20	2d	20	58	58	68	0d	0a	24	53	79	73	74	ODE - XXh..$Syst
00000090	65	6d	20	76	65	72	73	69	6f	6e	3a	20	20	20	2e	0d	em version: ..
000000a0	0a	24	4f	45	4d	3a	20	20	0d	0a	24	55	73	65	72	3a	.$OEM: ..$User:
000000b0	20	20	20	20	20	20	20	68	0d	0a	24	24	0f	3c	09	76	h..$$.<.v
000000c0	02	04	07	04	30	c3	b4	09	cd	21	c3	51	8a	e0	e8	ea	....0Гґ.Н!ГQЉаик
000000d0	ff	86	c4	b1	04	d2	e8	e8	e1	ff	59	c3	53	8a	fc	e8	я†Д±.ТиибяYГSЉьи
000000e0	e9	ff	88	25	4f	88	05	4f	8a	c7	e8	de	ff	88	25	4f	йя€%О€.ОЉЗиЮя€%О
000000f0	88	05	5b	c3	51	52	32	e4	33	d2	b9	0a	00	f7	f1	80	€.[ГQR2д3Т№..чсЋ
00000100	ca	30	88	14	4e	33	d2	3d	0a	00	73	f1	3c	00	74	04	К0€.N3Т=..sc<.t.
00000110	0c	30	88	04	5a	59	c3	b8	00	f0	8e	c0	26	a0	fe	ff	.0€.ZYГё.рЋA&.юя
00000120	3c	ff	74	2d	3c	fe	74	30	3c	fb	74	2c	3c	fc	74	2f	<яt-<ю t0<ы t,<ь t/
00000130	3c	fa	74	32	3c	f8	74	35	3c	fd	74	38	3c	f9	74	3b	<ъt2<ш t5<э t8<щ t;
00000140	e8	88	ff	8d	1e	7c	01	89	47	0a	8d	16	7c	01	eb	32	и€я..|.‰G...|.л2
00000150	90	8d	16	03	01	eb	2b	90	8d	16	0d	01	eb	24	90	8d	.....л+.....л$..
00000160	16	1a	01	eb	1d	90	8d	16	33	01	eb	16	90	8d	16	43	...л....3.л....C
00000170	01	eb	0f	90	8d	16	53	01	eb	08	90	8d	16	5f	01	eb	.л....S.л...._.л
00000180	01	90	e8	41	ff	c3	b4	30	cd	21	50	8d	36	8c	01	83	..иАяГґ0Н!Р.6Њ.ѓ
00000190	c6	11	e8	5f	ff	58	8a	c4	83	c6	03	e8	56	ff	8d	16	Ж.и_яХЉДѓЖ.иVя..
000001a0	8c	01	e8	21	ff	8d	36	a2	01	83	c6	05	8a	c7	e8	43	Њ.и!я.6ў.ѓЖ.ЉЗиС
000001b0	ff	8d	16	a2	01	e8	0e	ff	8d	3e	ab	01	83	c7	0b	8b	я..ў.и.я.>«.ѓЗ.‹
000001c0	c1	e8	18	ff	8a	c3	e8	02	ff	89	45	fe	8d	16	ab	01	Би.яЉГи.я‰Ею..«.
000001d0	e8	f3	fe	c3	e8	40	ff	e8	ac	ff	32	c0	b4	4c	cd	21	иююГи@яи¬я2АґLН!

Рисунок 4. Содержимое файла lab1com.com

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

«Плохой» EXE файл содержит заголовок с технической информацией и единственный сегмент, в котором одновременно располагаются данные и код. Код начинается с адреса 300h (начало кода выделено на рис.5 цветом). С адреса 0 располагается заголовок EXE файла, в нём содержатся сведения о размере модуля, относительных смещениях, об адресе точки входа и т.д.



LAB1COM.COM | LAB1COM.COM | LAB1COM.EXE

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
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	d1	01	50	43	20	2d	20	50	43	0d	0a	24	50	43	20	йC.PC - PC..$PC
00000310	2d	20	50	43	2f	58	54	0d	0a	24	50	43	20	2d	20	41	- PC/XT..$PC - A
00000320	54	20	6f	72	20	50	53	32	20	28	35	30	2f	36	30	29	T or PS2 (50/60)
00000330	0d	0a	24	50	43	20	2d	20	50	53	32	20	28	33	30	29	..$PC - PS2 (30)
00000340	0d	0a	24	50	43	20	2d	20	50	53	32	20	28	38	30	29	..$PC - PS2 (80)
00000350	0d	0a	24	50	43	20	2d	20	50	43	6a	72	0d	0a	24	54	..$PC - PCjr..$T
00000360	79	70	65	20	6f	66	20	50	43	3a	20	50	43	20	43	6f	ype of PC: PC Co
00000370	6e	76	65	72	74	61	62	6c	65	0d	0a	24	50	43	20	43	nvertable..$PC C
00000380	4f	44	45	20	2d	20	58	58	68	0d	0a	24	53	79	73	74	ODE - XXh..$Syst
00000390	65	6d	20	76	65	72	73	69	6f	6e	3a	20	20	20	2e	0d	em version: ..
000003a0	0a	24	4f	45	4d	3a	20	20	0d	0a	24	55	73	65	72	3a	.$OEM: ..$User:
000003b0	20	20	20	20	20	20	20	68	0d	0a	24	24	0f	3c	09	76	h..$$.<.v
000003c0	02	04	07	04	30	c3	b4	09	cd	21	c3	51	8a	e0	e8	ea	....0Гґ.H!ГQЉаик
000003d0	ff	86	c4	b1	04	d2	e8	e8	e1	ff	59	c3	53	8a	fc	e8	я†Д±.ТиибяYГSЉьи
000003e0	e9	ff	88	25	4f	88	05	4f	8a	c7	e8	de	ff	88	25	4f	йяЂ%О€.ОЉЗиЮяЂ%О
000003f0	88	05	5b	c3	51	52	32	e4	33	d2	b9	0a	00	f7	f1	80	€.[ГQR2д3Т№..чсЋ
00000400	ca	30	88	14	4e	33	d2	3d	0a	00	73	f1	3c	00	74	04	К0€.N3Т=..sc<.t.
00000410	0c	30	88	04	5a	59	c3	b8	00	f0	8e	c0	26	a0	fe	ff	.0€.ZYГё.рЋА&.юя
00000420	3c	ff	74	2d	3c	fe	74	30	3c	fb	74	2c	3c	fc	74	2f	<яt-<юt0<ыt,<ьt/
00000430	3c	fa	74	32	3c	f8	74	35	3c	fd	74	38	3c	f9	74	3b	<ъt2<шt5<эt8<щt;
00000440	e8	88	ff	8d	1e	7c	01	89	47	0a	8d	16	7c	01	eb	32	и€я..|.‰G...|.л2
00000450	90	8d	16	03	01	eb	2b	90	8d	16	0d	01	eb	24	90	8d	.....л+.....л$..
00000460	16	1a	01	eb	1d	90	8d	16	33	01	eb	16	90	8d	16	43	...л....3.л....C
00000470	01	eb	0f	90	8d	16	53	01	eb	08	90	8d	16	5f	01	eb	.л....S.л...._.л
00000480	01	90	e8	41	ff	c3	b4	30	cd	21	50	8d	36	8c	01	83	..иАяГґ0Н!Р.6Њ.ѓ
00000490	c6	11	e8	5f	ff	58	8a	c4	83	c6	03	e8	56	ff	8d	16	Ж.и_яXЉДѓЖ.иVя..
000004a0	8c	01	e8	21	ff	8d	36	a2	01	83	c6	05	8a	c7	e8	43	Њ.и!я.6ў.ѓЖ.ЉЗиС

Рисунок 5. Частичное содержимое файла lab1com.exe.

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

«Хороший» EXE файл содержит заголовок с технической информацией (общая длина составляет 200h), затем расположен сегмент стека (200h-600h) (так

как на стек было выделено 512 слов по 2 байта), сегмент данных (600h-6F0h), сегмент кода (6F0h – конец файла) (начало кода и сегмента данных выделено на рис.6 цветом). Отличие «хорошего» EXE от «плохого» заключается в делении на сегменты. Как написано выше, «хороший» содержит три сегмента (код, данные, стек), в то время как «плохой» - только один (сразу в одном сегменте находятся и данные и код).

LAB1EXE.EXE

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
00000560	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000570	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000580	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000590	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000005a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000005b0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000005c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000005d0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000005e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000005f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000600	50	43	20	2d	20	50	43	0d	0a	24	50	43	20	2d	20	50	PC - PC..$PC - P
00000610	43	2f	58	54	0d	0a	24	50	43	20	2d	20	41	54	20	6f	C/XT..$PC - AT o
00000620	72	20	50	53	32	20	28	35	30	2f	36	30	29	0d	0a	24	r PS2 (50/60)..$
00000630	50	43	20	2d	20	50	53	32	20	28	33	30	29	0d	0a	24	PC - PS2 (30)..$
00000640	50	43	20	2d	20	50	53	32	20	28	38	30	29	0d	0a	24	PC - PS2 (80)..$
00000650	50	43	20	2d	20	50	43	6a	72	0d	0a	24	54	79	70	65	PC - PCjr..$Type
00000660	20	6f	66	20	50	43	3a	20	50	43	20	43	6f	6e	76	65	of PC: PC Conve
00000670	72	74	61	62	6c	65	0d	0a	24	50	43	20	43	4f	44	45	rtable..$PC CODE
00000680	20	2d	20	58	58	68	0d	0a	24	53	79	73	74	65	6d	20	- XXh..$System
00000690	76	65	72	73	69	6f	6e	3a	20	20	20	2e	0d	0a	24	4f	version: ...$O
000006a0	45	4d	3a	20	20	0d	0a	24	55	73	65	72	3a	20	20	20	EM: ..$User:
000006b0	20	20	20	20	68	0d	0a	24	00	00	00	00	00	00	00	00	h..$........
000006c0	24	0f	3c	09	76	02	04	07	04	30	c3	b4	09	cd	21	c3	$.<.v....0Гґ.H!Г
000006d0	51	8a	e0	e8	ea	ff	86	c4	b1	04	d2	e8	e8	e1	ff	59	QЉаикя†Д±.ТиибяY
000006e0	c3	53	8a	fc	e8	e9	ff	88	25	4f	88	05	4f	8a	c7	e8	ГSЉьийя€%О€.ОЉЗи
000006f0	de	ff	88	25	4f	88	05	5b	c3	51	52	32	e4	33	d2	b9	Юя€%О€.[ГQR2д3Т№
00000700	0a	00	f7	f1	80	ca	30	88	14	4e	33	d2	3d	0a	00	73	..чсЪК0€.N3Т=..s
00000710	f1	3c	00	74	04	0c	30	88	04	5a	59	c3	b8	00	f0	8e	с<.t..0€.ZYГё.рЋ
00000720	c0	26	a0	fe	ff	3c	ff	74	2d	3c	fe	74	30	3c	fb	74	А&.юя<яt-<юt0<ыt
00000730	2c	3c	fc	74	2f	3c	fa	74	32	3c	f8	74	35	3c	fd	74	,<ьt/<ъt2<шt5<эt
00000740	38	3c	f9	74	3b	e8	88	ff	8d	1e	79	00	89	47	0a	8d	8<щt;и€я..y.‰G..
00000750	16	79	00	eb	32	90	8d	16	00	00	eb	2b	90	8d	16	0a	.y.л2.....л+....
00000760	00	eb	24	90	8d	16	17	00	eb	1d	90	8d	16	30	00	eb	.л$.....л....0.л
00000770	16	90	8d	16	40	00	eb	0f	90	8d	16	50	00	eb	08	90	....@.л....P.л..
00000780	8d	16	5c	00	eb	01	90	e8	41	ff	c3	b4	30	cd	21	50	..\.л..иАяГґ0Н!Р
00000790	8d	36	89	00	83	c6	11	e8	5f	ff	58	8a	c4	83	c6	03	.6‰.ѓЖ.и_яХЉДѓЖ.
000007a0	e8	56	ff	8d	16	89	00	e8	21	ff	8d	36	9f	00	83	c6	иVя..‰.и!я.6џ.ѓЖ
000007b0	05	8a	c7	e8	43	ff	8d	16	9f	00	e8	0e	ff	8d	3e	a8	.ЉЗиСя..џ.и.я.>Ё
000007c0	00	83	c7	0b	8b	c1	e8	18	ff	8a	c3	e8	02	ff	89	45	.ѓЗ.‹Би.яЉГи.я‰E
000007d0	fe	8d	16	a8	00	e8	f3	fe	c3	2b	c0	50	b8	40	00	8e	ю..Ё.иуюГ+АРё@.Ћ
000007e0	d8	e8	38	ff	e8	a4	ff	32	c0	b4	4c	cd	21				Ши8яи¤я2АґLН!

Рисунок 6. Частичное содержимое файла lab1exe.exe.

Загрузка COM модуля в основную память.

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

Выделяется свободный сегмент памяти, адрес которого заносится в сегментные регистры. Далее в первые 256 байт данного сегмента записывается PSP. Затем с диска загружается COM-файл без изменений. Сегментные регистры CS, DS, ES, SS устанавливаются на начало PSP (0h). Указатель стека устанавливается на конец данного сегмента, и в стек записывается адрес возврата 0000h (начало PSP). В регистр IP записывается значение 100h.

2) *Что располагается с адреса 0?*

С адреса 0 располагается PSP-префикс программного сегмента.

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

Сегментные регистры CS, DS, SS, SE имеют значения 48DD, в начале программы указывают на начало PSP.

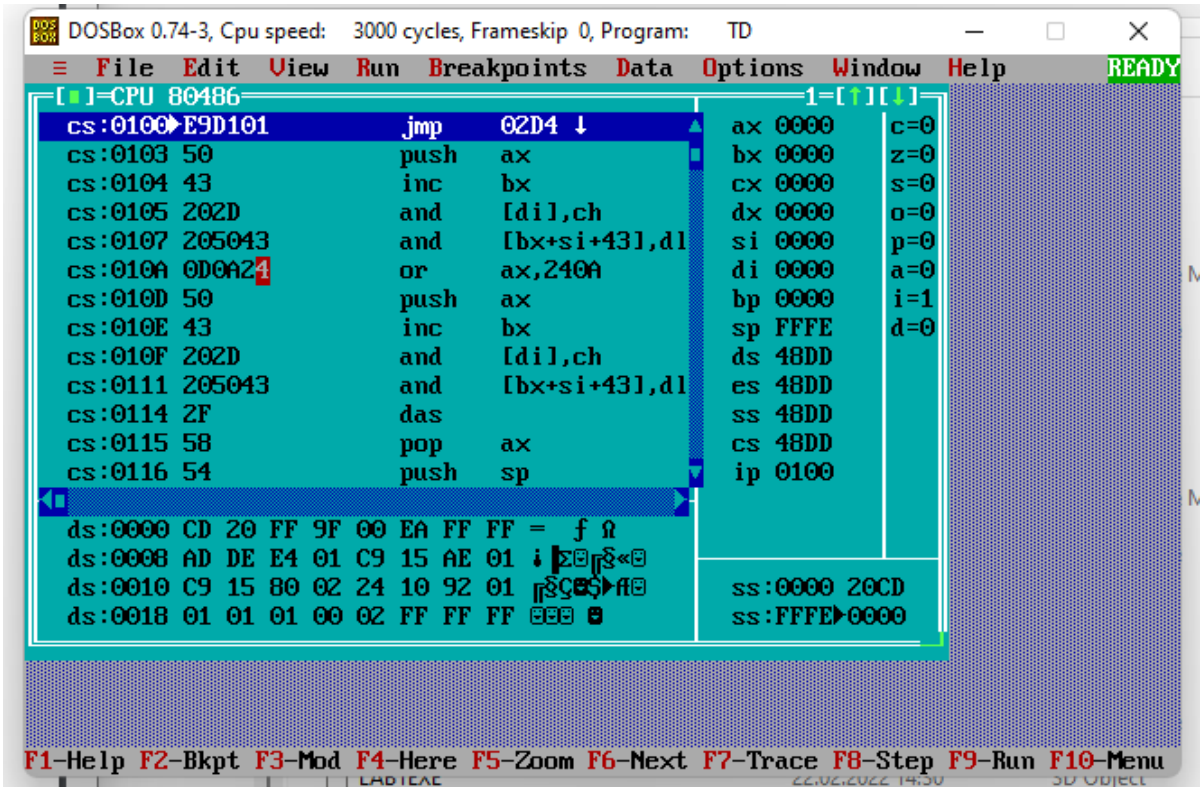


Рисунок 7. Отладчик td.exe с открытым COM-файлом.

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

Стек определяется автоматически при запуске программы и располагается в сегменте кода. Указатель стека установлен на конец сегмента FFFEh, т.е. под стек отводится оставшаяся часть сегмента после кода и данных.

Загрузка «хорошего» EXE модуля в основную память.

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

Определяется сегментный адрес свободного участка памяти, у которого достаточно места для загрузки программы. Создаются блоки памяти для переменных среды и для PSP и программы. В блок памяти переменных среды помещается путь к файлу программы, заполняется PSP. Происходит считывание форматированной части заголовка файла. На основе данных в ней определяется размер загрузочного модуля и смещение его начала.

2) *На что указывают регистры DS и ES?*

Регистры DS и ES в начале выполнения программы указывают на начало сегмента PSP.

3) *Как определяется стек?*

Стек определяется с помощью директивы .STACK, после которой задаётся размер стека. Также стек можно определить, использовав стандартную директиву SEGMENT, с помощью команды

Имя_Сегмента SEGMENT STACK

4) *Как определяется точка входа?*

Точка входа определяется при помощи директивы END.

END ИмяТочкиВхода