

**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 гр. 0381

Кирильцев Д.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.**

Требуется написать текст исходного .COM модуля, который определяет тип PC и версию системы. 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 сформировать 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 (Original Equipment Manufacturer) и серийным номером пользователя. Полученные строки выводятся на экран. Далее необходимо отладить полученный исходный модуль и получить «хороший» .COM модуль, а также необходимо построить «плохой» .EXE, полученный из исходного текста для .COM модуля. Затем нужно написать текст «хорошего» .EXE модуля, который выполняет те же функции, что и модуль .COM, далее его построить, отладить и сравнить исходные тексты для .COM и .EXE модулей.

Таблица 1 — Процедуры в программе.

Процедура	Описание
TETR_TO_HEX	Перевод десятичной цифры в код символа
BYTE_TO_HEX	Перевод байта в 16-ной с/с в символьный код
WRD_TO_HEX	Перевод слова в 16-ной с/с в символьный код
BYTE_TO_DEC	Перевод байта в 16-ной с/с в символьный код в 10-ной с/с
WR	Вывод строки на экран
PCTYPE	Определение модели PC
VERSION	Определение версии OS

### **Выполнение работы.**

Данные объявленные в программе:

PC db 'PC', 0DH, 0AH, '\$'

XPC db 'PC/XT', 0DH, 0AH, '\$'

TYPE\_AT db 'AT or PS2 [50 or 60]', 0DH, 0AH, '\$'

PS30\_2 db 'PS2 [30]', 0DH, 0AH, '\$'

PS80\_2 db 'PS2 [80]', 0DH, 0AH, '\$'

JR db 'PCjr', 0DH, 0AH, '\$'

PCC db 'Type PC, 0DH, 0AH, '\$'

UN db 'CODE - XXh', 0DH, 0AH, '\$'

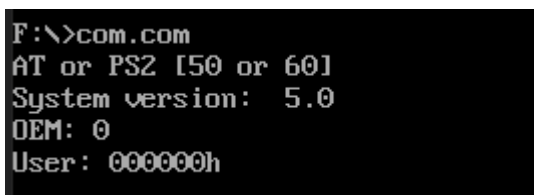
SYSTEM db 'System version: .', 0DH, 0AH, '\$'

OEN db 'OEM: ', 0DH, 0AH, '\$'

USN db 'User: h', 0DH, 0AH, '\$'

Программа последовательно выводит тип ПК, версию ОС, OEM и номер пользователя.

Далее представлены скриншоты полученных модулей.



```
F:\>com.com
AT or PS2 [50 or 60]
System version: 5.0
OEM: 0
User: 000000h
```

рис.1 - хороший COM файл.

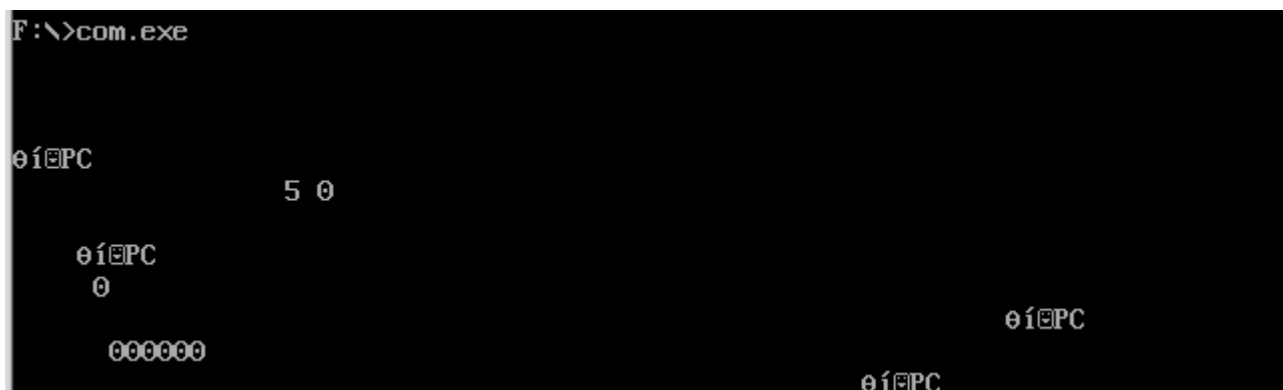


Рис.2 - плохой EXE файл

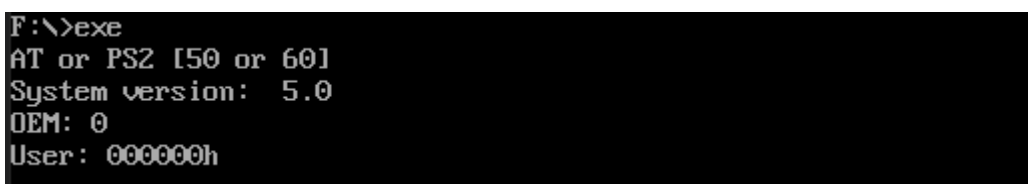


Рис.3 - хороший EXE файл

## **Отличия исходных текстов COM и EXE программ**

### *1. Сколько сегментов должна содержать COM-программа?*

Стек генерируется автоматически, код и данные в COM, находятся в одном сегменте

### *2. EXE-программа?*

EXE-модуль должен содержать сегмент кода и сегмент данных. Остальные сегменты являются опциональными. Если не объявить стек, то будет использоваться DOS-овский.

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

1. ORG 100h - происходит пропуск первых 256 байт сегмента, для размещения в них PSP (префикс программного сегмента), в случае отсутствия директивы программа будет запускаться, но вывод будет неверен.
2. ASSUME - для того, чтобы сегмент данных и сегмент кода указывали на один общий сегмент.

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

Нет, команды у которых операнды являются сегменты будут не выполнены, тк в COM-модулях отсутствует заголовок, в которой нет заголовка содержащего таблицу настройки, с помощью нее происходит поиск абсолютных адресов сегмента.

## Отличия форматов файлов .COM и .EXE программ

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

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
00000000	e9	a1	01	50	43	0d	0a	24	50	43	2f	58	54	0d	0a	24	йЎ.РC..\$PC/ХТ..\$
00000010	41	54	20	6f	72	20	50	53	32	20	5b	35	30	20	6f	72	АТ or РS2 [50 or
00000020	20	36	30	5d	0d	0a	24	50	53	32	20	5b	33	30	5d	0d	60]..\$PS2 [30].
00000030	0a	24	50	53	32	20	5b	38	30	5d	0d	0a	24	50	43	6a	.\$PS2 [80]..\$PCj
00000040	72	0d	0a	24	54	79	70	65	20	50	43	20	0d	0a	24	43	r..\$Type PC ..\$C
00000050	4f	44	45	20	2d	20	58	58	68	0d	0a	24	53	79	73	74	ODE - XXh..\$Syst
00000060	65	6d	20	76	65	72	73	69	6f	6e	3a	20	20	20	2e	0d	em version: ..
00000070	0a	24	4f	45	4d	3a	20	20	0d	0a	24	55	73	65	72	3a	.\$OEM: ..\$User:
00000080	20	20	20	20	20	20	20	68	0d	0a	24	24	0f	3c	09	76	h..\$.<.v
00000090	02	04	07	04	30	c3	b4	09	cd	21	c3	51	8a	e0	e8	ea	....0Гг.Н!ГQЪаик
000000a0	ff	86	c4	b1	04	d2	e8	e8	e1	ff	59	c3	53	8a	fc	e8	я†Д†.ТиибяҮГSЉьи
000000b0	e9	ff	88	25	4f	88	05	4f	8a	c7	e8	de	ff	88	25	4f	йя€%O€.OЉзиЮя€%O
000000c0	88	05	5b	c3	51	52	32	e4	33	d2	b9	0a	00	f7	f1	80	€. [ГQР2д3Т№..чсЪ
000000d0	ca	30	88	14	4e	33	d2	3d	0a	00	73	f1	3c	00	74	04	K0€.N3T=..sc<.t.
000000e0	0c	30	88	04	5a	59	c3	b8	00	f0	8e	c0	26	a0	fe	ff	.0€.ZYГё.рҢА&.юя
000000f0	3c	ff	74	2d	3c	fe	74	30	3c	fb	74	2c	3c	fc	74	2f	<ят-<ют0<ыт,<ьт/
00000100	3c	fa	74	32	3c	f8	74	35	3c	fd	74	38	3c	f9	74	3b	<ът2<шт5<эт8<шт;
00000110	e8	88	ff	8d	1e	4f	01	89	47	0a	8d	16	4f	01	eb	32	и€я..O.%G...O.л2
00000120	90	8d	16	03	01	eb	2b	90	8d	16	08	01	eb	24	90	8d	.....л+.....л\$..
00000130	16	10	01	eb	1d	90	8d	16	27	01	eb	16	90	8d	16	32	...л....'.л....2
00000140	01	eb	0f	90	8d	16	3d	01	eb	08	90	8d	16	44	01	eb	.л....=.л....D.л
00000150	01	90	e8	41	ff	c3	b4	30	cd	21	50	8d	36	5c	01	83	..иАяГгОН!Р.6\.ѓ
00000160	c6	11	e8	5f	ff	58	8a	c4	83	c6	03	e8	56	ff	8d	16	Ж.и_яХЉДѓЖ.иVя..
00000170	5c	01	e8	21	ff	8d	36	72	01	83	c6	05	8a	c7	e8	43	\.и!я.6г.ѓЖ.ЉзиС
00000180	ff	8d	16	72	01	e8	0e	ff	8d	3e	7b	01	83	c7	0b	8b	я...г.и.я.>{.ѓЗ.<
00000190	c1	e8	18	ff	8a	c3	e8	02	ff	89	45	fe	8d	16	7b	01	Би.яЉГи.я%Ею...{.
000001a0	e8	f3	fe	c3	e8	40	ff	e8	ac	ff	32	c0	b4	4c	cd	21	иуюГи@яи¬я2ArLH!

COM-файл состоит из одного сегмента, сегмент стека генерируется автоматически при создании COM - модуля. COM-файл ограничен размером одного сегмента и не превышает 64 Кб. Код начинается с адреса 0h, но при загрузке модуля устанавливается смещение в 100h.

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

Что располагается с адреса 0?

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
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	a1	01	50	43	0d	0a	24	50	43	2f	58	54	0d	0a	24	йЎ.PC..\$PC/ХТ..\$
00000310	41	54	20	6f	72	20	50	53	32	20	5b	35	30	20	6f	72	AT or PS2 [50 or
00000320	20	36	30	5d	0d	0a	24	50	53	32	20	5b	33	30	5d	0d	60]..\$PS2 [30].
00000330	0a	24	50	53	32	20	5b	38	30	5d	0d	0a	24	50	43	6a	..\$PS2 [80]..\$PCj
00000340	72	0d	0a	24	54	79	70	65	20	50	43	20	0d	0a	24	43	r..\$Type PC ..\$C
00000350	4f	44	45	20	2d	20	58	58	68	0d	0a	24	53	79	73	74	ODE - XXh..\$Syst
00000360	65	6d	20	76	65	72	73	69	6f	6e	3a	20	20	20	2e	0d	em version: ..
00000370	0a	24	4f	45	4d	3a	20	20	0d	0a	24	55	73	65	72	3a	..\$OEM: ..\$User:
00000380	20	20	20	20	20	20	20	68	0d	0a	24	24	0f	3c	09	76	h...\$.<.v
00000390	02	04	07	04	30	c3	b4	09	cd	21	c3	51	8a	e0	e8	ea	....0Гг.Н!ГQЉаик
000003a0	ff	86	c4	b1	04	d2	e8	e8	e1	ff	59	c3	53	8a	fc	e8	я†Д±.ТиибяҮТСЉьи
000003b0	e9	ff	88	25	4f	88	05	4f	8a	c7	e8	de	ff	88	25	4f	йя€%O€.OЉзиюя€%O
000003c0	88	05	5b	c3	51	52	32	e4	33	d2	b9	0a	00	f7	f1	80	€. [ГQR2дЗТЉ...чсЪ
000003d0	ca	30	88	14	4e	33	d2	3d	0a	00	73	f1	3c	00	74	04	K0€.N3Т=.sc<.t.
000003e0	0c	30	88	04	5a	59	c3	b8	00	f0	8e	c0	26	a0	fe	ff	.0€.ZYГё.p҃A&.юя
000003f0	3c	ff	74	2d	3c	fe	74	30	3c	fb	74	2c	3c	fc	74	2f	<ят-<ют0<ыт,<ыт/
00000400	3c	fa	74	32	3c	f8	74	35	3c	fd	74	38	3c	f9	74	3b	<ът2<шт5<эт8<шт;
00000410	e8	88	ff	8d	1e	4f	01	89	47	0a	8d	16	4f	01	eb	32	и€я..O.%G...O.л2
00000420	90	8d	16	03	01	eb	2b	90	8d	16	08	01	eb	24	90	8d	.....л+.....л\$..
00000430	16	10	01	eb	1d	90	8d	16	27	01	eb	16	90	8d	16	32	....л....'.л....2
00000440	01	eb	0f	90	8d	16	3d	01	eb	08	90	8d	16	44	01	eb	.л....=.л....D.л
00000450	01	90	e8	41	ff	c3	b4	30	cd	21	50	8d	36	5c	01	83	..иАяГг0Н!Р.6\.ф
00000460	c6	11	e8	5f	ff	58	8a	c4	83	c6	03	e8	56	ff	8d	16	Ж.и_яХЉДг҃Ж.иВя..
00000470	5c	01	e8	21	ff	8d	36	72	01	83	c6	05	8a	c7	e8	43	\.и!я.6г.ф҃Ж.ЉзиС
00000480	ff	8d	16	72	01	e8	0e	ff	8d	3e	7b	01	83	c7	0b	8b	я...г.и.я.>{.фЗ.<
00000490	c1	e8	18	ff	8a	c3	e8	02	ff	89	45	fe	8d	16	7b	01	Би.яЉГи.я%Ею...{.
000004a0	e8	f3	fe	c3	e8	40	ff	e8	ac	ff	32	c0	b4	4c	cd	21	июуГи@яи-я2Аг҃ЛН!

У «плохого» EXE файла данные и код располагаются в одном сегменте, однако это не соответствует формату EXE. Код начинается с адреса 300h, а с адреса 0h идёт настраивающая таблица (заголовок EXE файла). (Это иллюстрирует Рис. 5)



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

```

000005f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000600 50 43 0d 0a 24 50 43 2f 58 54 0d 0a 24 41 54 20 PC..$PC/XT..$AT
00000610 6f 72 20 50 53 32 20 5b 35 30 20 6f 72 20 36 30 or PS2 [50 or 60
00000620 5d 0d 0a 24 50 53 32 20 5b 33 30 5d 0d 0a 24 50 ]..$PS2 [30]..$P
00000630 53 32 20 5b 38 30 5d 0d 0a 24 50 43 6a 72 0d 0a S2 [80]..$PCjr..
00000640 24 54 79 70 65 20 50 43 20 43 6f 6e 76 65 72 74 $Type PC Convert
00000650 61 62 6c 65 0d 0a 24 43 4f 44 45 20 2d 20 58 58 able..$CODE - XX
00000660 68 0d 0a 24 53 79 73 74 65 6d 20 76 65 72 73 69 h..$System versi
00000670 6f 6e 3a 20 20 20 2e 0d 0a 24 4f 45 4d 3a 20 20 on: ...$OEM:
00000680 0d 0a 24 55 73 65 72 3a 20 20 20 20 20 20 20 68 ..$User: h
00000690 0d 0a 24 00 00 00 00 00 00 00 00 00 00 00 00 ..$.
000006a0 24 0f 3c 09 76 02 04 07 04 30 c3 b4 09 cd 21 c3 $.<.v....0Гг.Н!Г
000006b0 51 8a e0 e8 ea ff 86 c4 b1 04 d2 e8 e8 e1 ff 59 QЪаикя†Д±.ТиибяУ
000006c0 c3 53 8a fc e8 e9 ff 88 25 4f 88 05 4f 8a c7 e8 ГSЪиййя€%О€.ОЪзи
000006d0 de ff 88 25 4f 88 05 5b c3 51 52 32 e4 33 d2 b9 юя€%О€. [ГQ R2д3ТН
000006e0 0a 00 f7 f1 80 ca 30 88 14 4e 33 d2 3d 0a 00 73 ..чсЪК0€.N3Т=..s
000006f0 f1 3c 00 74 04 0c 30 88 04 5a 59 c3 b8 00 f0 8e с<.t...0€.ZYГё.pЪ
00000700 c0 26 a0 fe ff 3c ff 74 2d 3c fe 74 30 3c fb 74 А&.юя<ят-<ют0<ыт
00000710 2c 3c fc 74 2f 3c fa 74 32 3c f8 74 35 3c fd 74 ,<ът/<ът2<шт5<эт
00000720 38 3c f9 74 3b e8 88 ff 8d 1e 57 00 89 47 0a 8d 8<шт;и€я..W.%G..
00000730 16 57 00 eb 32 90 8d 16 00 00 eb 2b 90 8d 16 05 .W.л2.....л+....
00000740 00 eb 24 90 8d 16 0d 00 eb 1d 90 8d 16 24 00 eb .л$......л....$.л
00000750 16 90 8d 16 2f 00 eb 0f 90 8d 16 3a 00 eb 08 90 ..../.л.....:л..
00000760 8d 16 41 00 eb 01 90 e8 41 ff c3 b4 30 cd 21 50 ..А.л..иАяГг0Н!Р
00000770 8d 36 64 00 83 c6 11 e8 5f ff 58 8a c4 83 c6 03 .6d.ѓЖ.и_яХЪДѓЖ.
00000780 e8 56 ff 8d 16 64 00 e8 21 ff 8d 36 7a 00 83 c6 иVя...d.и!я.6z.ѓЖ
00000790 05 8a c7 e8 43 ff 8d 16 7a 00 e8 0e ff 8d 3e 83 .ЪзиСя...z.и.я.>ѓ
000007a0 00 83 c7 0b 8b c1 e8 18 ff 8a c3 e8 02 ff 89 45 .ѓЗ.<Би.яЪГи.я%Е
000007b0 fe 8d 16 83 00 e8 f3 fe c3 2b c0 50 b8 40 00 8e ю...ѓ.иуюГ+АРё@.Ъ
000007c0 d8 e8 38 ff e8 a4 ff 32 c0 b4 4c cd 21 Ши8яиця2АгЛН!

```

В EXE-модуле код и данные являются отдельными сегментами, также присутствует таблица связей, заголовок, отвечающий за настройку адресов. В «хорошем» EXE-модуле происходит разделение сегментов (кода и данных), необходимое для правильного форматирования, а в «плохом» содержится лишь один сегмент, объединяющий код и данные. «Плохой» EXE начинает код с 300h, так как он получается из COM модуля, в котором изначально сегмент кода смещен на 100h, Но, так как, происходит создание EXE-модуля, добавляется еще и сдвиг PSP (200h). В «хорошем» EXE присутствует только смещение для PSP модуля, поэтому код начинается с 200h. В данном случае смещение кода

300h так как выделяется память под стек (в размере 100h), память под стек находится между PSP и кодом. (Как показано на рис. 6)

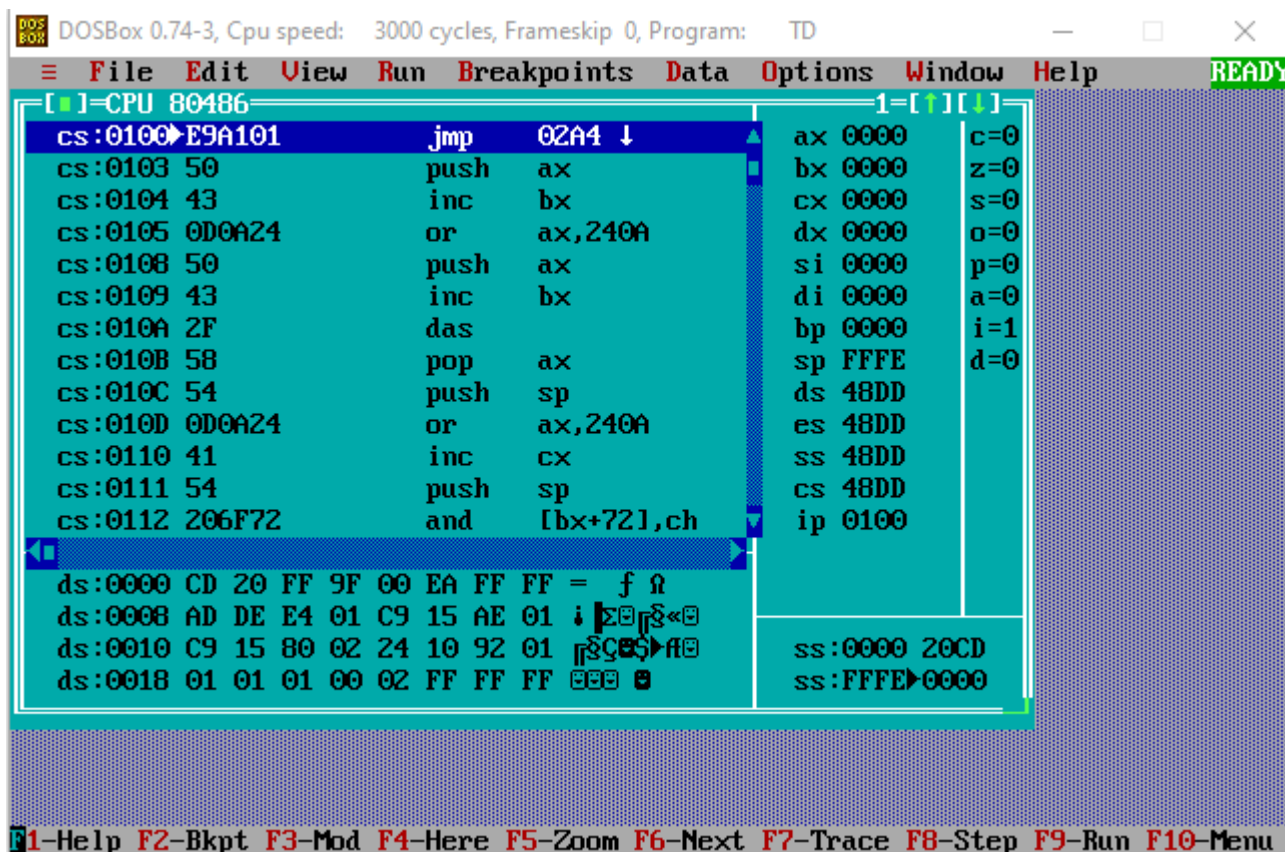
4) Как определяется стек? Какую область памяти он занимает? Какие адреса?

Стек находится между PSP и данными и занимает с 100h до 300h

### Загрузка COM модуля в основную память

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

Определяется сегментный адрес участка Основной Памяти, у которого достаточно места для загрузки программы, образ COM-файла считывается с диска и помещается в память, начиная с адреса PSP 100h. После загрузки двоичного образа COM-модуля сегментные регистры CS, DS, ES и SS указывают на PSP(в данном случае сегментные регистры указывают на 48DD), SP указывает на конец сегмента PSP (FFFE), слово 00H помещено в стек, IP содержит 100H



## 2. Что располагается с адреса 0?

Программный сегмент PSP, размером 256 байт (100h), зарезервированный операционной системой.

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

Сегментные регистры CS, DS, ES и SS указывают на PSP и имеют значения 48DD.

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

Стек генерируется автоматически при создании COM-программы. SS – на начало (0h), регистр SP указывает на конец стека (FFFEh), Адреса стека расположены в диапазоне 0h – FFFEh (FFFEh, – последний адрес, кратный двум) Загрузка «хорошего» EXE модуля в основную память

The screenshot shows the DOSBox 0.74-3 interface. The CPU register window displays the following values:

Register	Value
ax	0000
bx	0000
cx	0000
dx	0000
si	0000
di	0000
bp	0000
sp	0400
ds	48DD
es	48DD
ss	48ED
cs	4937
ip	0119

The assembly code window shows the following instructions:

Address	Instruction
cs:0119	sub ax,ax
cs:011B	push ax
cs:011C	mov ax,492D
cs:011F	mov ds,ax
cs:0121	call 005C
cs:0124	call 00CB
cs:0127	xor al,al
cs:0129	mov ah,4C
cs:012B	int 21
cs:012D	add [bx+si],al
cs:012F	add [bx+si],al
cs:0131	add [bx+si],al
cs:0133	add [bx+si],al

The memory dump window shows the following data:

Address	Data
ds:0000	CD 20 FF 9F 00 EA FF FF = f 0
ds:0008	AD DE E4 01 C9 15 AE 01 i 20 f 0
ds:0010	C9 15 80 02 24 10 92 01 f 00 f 0
ds:0018	01 01 01 00 02 FF FF FF 00 0

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

EXE-файл загружается, начиная с адреса PSP:0100h. В процессе загрузки считывается информация заголовка (PSP) EXE в начале файла и выполняется перемещение адресов сегментов, то есть DS и ES устанавливаются на начало сегмента PSP(DS=ES=48DD), SS(SS=48ED) – на начало сегмента стека, CS(CS=4905) – на начало сегмента команд. В IP загружается смещение точки входа в программу, которая берётся из метки после директивы END. (Это иллюстрирует рисунок 8)

## 2. На что указывают регистры DS и ES?

Регистры DS и ES указывают на начало PSP.

## 3. Как определяется стек?

Стек определяется с помощью Stack Segment, после которой задается размер стека. При исполнении регистр SS указывает на начало сегмента стека, а SP на конца стека(его смещение).

## 4. Как определяется точка входа?

Точка входа определяется при помощи директивы END.

## **Выводы.**

Были написаны COM и EXE модули, на основе которых производилось сравнение данных форматов. Также были выявлены недостатки и преимущества каждого из них.