

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.7383

Рудоман В.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2019

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

Функции и структуры данных программы:

Название	Назначение
Write_msg	Вывод сообщения на экран
TETR_TO_HEX	Перевод числа из 2-ой в 16-ую с/с (1/2 байта)
BYTE_TO_HEX	Перевод числа из 2-ой в 16-ую с/с (1 байт)
WRD_TO_HEX	Перевод числа из 2-ой в 16-ую с/с (2 байта)
BYTE_TO_DEC	Перевод числа из 2-ой в 10-ую с/с (1 байт)
GET_TYPE_OF_PC	Определение типа IBM PC
GET_VER_OF_SYS	Определение версии системы
GET_NUM_OF_OEM	Определение OEM
GET_SERIAL_NUM	Определение серийного номера пользователя

Название	Тип	Назначение
PC_TYPE	db	'PC type: ',0dh,0ah,'\$'
SYSTEM_VERSION	db	'System version: . ',0dh,0ah,'\$'
OEM_NUMBER	db	'OEM number: ',0dh,0ah,
SERIAL_NUMBER	db	'User serial number: ',0dh,0ah,'\$'

Действия, выполняемые программой:

Программа определяет:

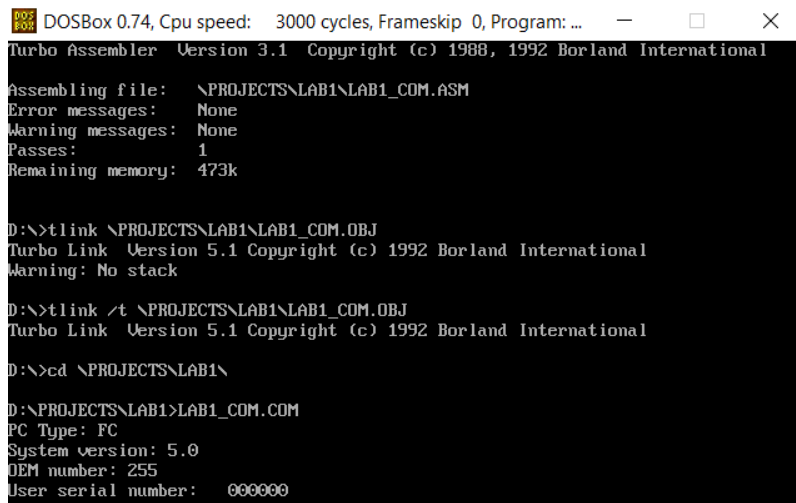
- 1) Тип IBM PC;
- 2) Версию системы;
- 3) OEM номер;
- 4) Серийный номер пользователя.

Сохраняет их в соответствующие переменные, а затем — выводит эти значения на экран при помощи функции 09H и вызова прерывания int 21h.

Ход работы:

1) Написание текста исходного LAB1_COM.ASM файла, компиляция и компоновка LAB1_COM.COM модуля, который определяет тип РС, версию системы, OEM и серийный номер пользователя.

Результат работы LAB1_COM.COM:



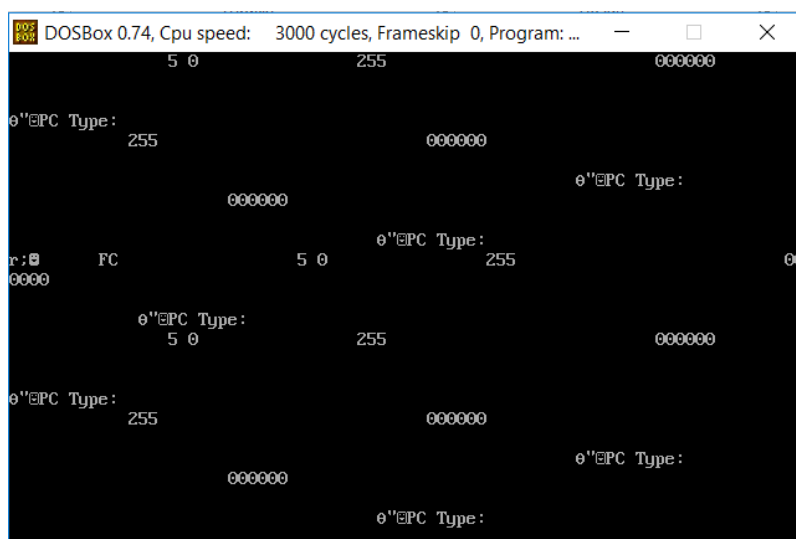
```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
Turbo Assembler Version 3.1 Copyright (c) 1988, 1992 Borland International
Assembling file:  \PROJECTS\LAB1\LAB1_COM.ASM
Error messages:   None
Warning messages: None
Passes:          1
Remaining memory: 473k

D:\>tlink \PROJECTS\LAB1\LAB1_COM.OBJ
Turbo Link Version 5.1 Copyright (c) 1992 Borland International
Warning: No stack

D:\>tlink /t \PROJECTS\LAB1\LAB1_COM.OBJ
Turbo Link Version 5.1 Copyright (c) 1992 Borland International

D:\>cd \PROJECTS\LAB1\
D:\PROJECTS\LAB1>LAB1_COM.COM
PC Type: FC
System version: 5.0
OEM number: 255
User serial number: 000000
```

Так же был скомпилирован «плохой» LAB1_COM.EXE. Результат запуска данного модуля представлен ниже:



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
5 0 255 000000
PC Type: 255 000000 PC Type:
000000 PC Type:
r: 0 FC 5 0 PC Type: 255 00
0000
PC Type: 5 0 255 000000
PC Type: 255 000000 PC Type:
000000 PC Type:
PC Type:
```

2) После этого был написан LAB1_EXE.ASM файл, построен и отлажен. В итоге получился «хороший» LAB1_EXE.EXE. Результат запуска данного модуля представлен ниже:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
Z:\>D:
D:\>tasm \PROJECTS\LAB1\LAB1_EXE.ASM \PROJECTS\LAB1\
Turbo Assembler Version 3.1 Copyright (c) 1988, 1992 Borland International

Assembling file: \PROJECTS\LAB1\LAB1_EXE.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 473k

D:\>TLINK \PROJECTS\LAB1\LAB1_EXE.OBJ
Turbo Link Version 5.1 Copyright (c) 1992 Borland International

D:\>CD \PROJECTS\LAB1
D:\PROJECTS\LAB1>LAB1_EXE.EXE
PC type: PC
System version: 5.0
OEM number: 255
User serial number: 000000
```

3) С помощью Far откроем все файлы вышеперечисленных модулей в шестнадцатеричном виде.

Представление LAB1_COM.COM в шестнадцатеричном виде:

```
view LAB1_COM.COM - Far 3.0.5354 x64
D:\TASM\projects\lab1\LAB1_COM.COM h 1252 340 Col 0 100% 23:02
00000000: E9 22 01 50 43 20 54 79 70 65 3A 20 20 0D 0A 24 é"0PC Type: 0$
00000010: 53 79 73 74 65 6D 20 76 65 72 73 69 6F 6E 3A 20 System version:
00000020: 20 2E 20 20 0D 0A 24 4F 45 4D 20 6E 75 6D 62 65 . 0$OEM numbe
00000030: 72 3A 20 20 20 20 20 20 0D 0A 24 55 73 65 72 20 r: 0$User
00000040: 73 65 72 69 61 6C 20 6E 75 6D 62 65 72 3A 20 20 serial number:
00000050: 20 20 20 20 20 20 0D 0A 24 20 20 20 20 20 0D 0A 24 0$
00000060: B4 09 CD 21 C3 24 0F 3C 09 76 02 04 07 04 30 C3 "oi!A$<ov0+0A
00000070: 51 8A C4 E8 EF FF 86 C4 B1 04 D2 E8 E8 E6 FF 59 Q5ÄiÿtA±0ëëÿY
00000080: C3 53 8A FC E8 E9 FF 88 25 4F 88 05 4F 8A C7 32 ÅSüëÿ"0"0SC2
00000090: E4 E8 DC FF 88 25 4F 88 05 5B C3 51 52 50 32 E4 äëÿ"0"0[ÄQRP2ä
000000A0: 33 D2 B9 0A 00 F7 F1 80 CA 30 88 14 4E 33 D2 3D 3D+ ñëÿ0"JN3D=
000000B0: 0A 00 73 F1 3D 00 00 76 04 0C 30 88 04 58 5A 59 sñ= v90"0XZY
000000C0: C3 06 53 50 BB 00 F0 8E C3 26 A1 FE FF 8A E0 E8 ÅSP» ðŽÄ&_bÿŠaè
000000D0: 9E FF BB 03 01 89 47 09 58 5B 07 C3 50 56 BE 10 žÿ»0GoX[•APV%>
000000E0: 01 83 C6 10 E8 B4 FF 83 C6 03 8A C4 E8 AC FF 5E 0fA-èÿfAÿŠAè-ÿ^
000000F0: 58 C3 50 53 56 8A C7 BE 27 01 83 C6 0E E8 9B FF XÄPSVŠCÿ'0fAè>ÿ
00000100: 5E 5B 58 C3 50 53 51 56 8A C3 E8 63 FF BF 3B 01 ^[XÄPSQVŠÄecÿ;0
00000110: 83 C7 16 89 05 8B C1 BF 3B 01 83 C7 1B E8 61 FF fC-0+Ä;0fC+eaÿ
00000120: 5E 59 5B 58 C3 E8 99 FF B4 30 CD 21 E8 AD FF E8 ^Y[XÄè"ÿ0í!è-ÿè
00000130: C0 FF E8 CF FF BA 03 01 E8 25 FF BA 10 01 E8 1F Äÿèÿÿ»0ëÿ»0ëÿ
00000140: FF BA 27 01 E8 19 FF BA 3B 01 E8 13 FF 32 C0 B4 ÿè'0èÿÿ;0è!!ÿ2A`
00000150: 3C CD 21 C3 <í!Ä
```

Представление «хорошего» LAB1 EXE.EXE:

```
view LAB1_EXE.EXE - Far 3.0.5354 x64
D:\TASM\projects\lab1\LAB1_EXE.EXE          h 1252          881 Col    0    100%  23:06
00000000C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000000D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000000E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000000F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000000200: 50 43 20 74 79 70 65 3A 20 20 0D 0A 24 53 79 73 PC type: 0x5$Sys
0000000210: 74 65 6D 20 76 65 72 73 69 6F 6E 3A 20 20 2E 20 tem version: .
0000000220: 20 0D 0A 24 4F 45 4D 20 6E 75 6D 62 65 72 3A 20 0x$OEM number:
0000000230: 20 20 20 20 20 0D 0A 24 55 73 65 72 20 73 65 72 0x$User ser
0000000240: 69 61 6C 20 6E 75 6D 62 65 72 3A 20 20 20 20 20 ial number:
0000000250: 20 20 20 20 20 0D 0A 24 20 0D 0A 24 00 00 00 00 0x$
0000000260: B4 09 CD 21 CB 24 0F 3C 09 76 02 04 07 04 30 CB 0i!E$<ov0+0E
0000000270: 51 8A C4 0E E8 EE FF 86 C4 B1 04 D2 E8 0E E8 E4 Q$A$e$y$T$A$+0e$e$
0000000280: FF 59 CB 53 8A FC 0E E8 E6 FF 88 25 4F 88 05 4F y$E$S$u$e$y$~%0`+0
0000000290: 8A C7 32 E4 FC 0E D8 FF 88 25 4F 88 05 5B CB 51 $C2A$e$0$y$~%0`+[$EQ
00000002A0: 52 50 32 E4 33 D2 B9 0A 00 F7 F1 80 CA 30 88 14 RP2A301$ ÷nE$0`J
00000002B0: 4E 33 D2 3D 0A 00 73 F1 3D 00 00 76 04 0C 30 88 N30=$ sñ= v+00`
00000002C0: 04 58 5A 59 CB 06 53 F0 BB 00 F0 8E C3 26 A1 FE 0XZYE$+SP» ðZ$A$jb
00000002D0: FF 8A E0 0E E8 99 FF BB 00 00 89 47 09 58 5B 07 y$A$e$~y» $GoX[.
00000002E0: CB 50 56 BE 0D 00 83 C6 10 0E E8 B2 FF 83 C6 03 ÉPV%$ fA$-e$e$y$FAV
00000002F0: 8A C4 0E E8 A9 FF 5E 58 CB 50 53 56 8A C7 BE 24 ŠA$e$0$XEP$SV$C$S
0000000300: 00 83 C6 0E 0E E8 97 FF 5E 58 58 CB 50 53 51 56 fA$e$e$-y`[XEP$SQV
0000000310: 8A C3 0E E8 5A FF BF 38 00 83 C7 16 89 05 8B C1 ŠA$e$Zy$8 fC$e$+A
0000000320: BF 38 00 83 C7 1B 0E E8 59 FF 5E 59 58 58 CB 1E $8 fC$e$e$Y$~Y[XE$
0000000330: 2B C0 50 B8 00 00 8E D8 2B C0 0E E8 87 FF BA 30 +AP. Ž0+A$e$y`0
0000000340: CD 21 0E E8 9B FF 0E E8 AF FF 0E E8 BE FF BA 00 ĩ!$e$y$e$e$y$e$y$e$
0000000350: 00 0E E8 0B FF BA 0D 00 0E E8 04 FF BA 24 00 0E e$e$y$e$e$y$e$e$y$e$
0000000360: E8 FD FE BA 38 00 0E E8 F6 FE 32 C0 B4 3C CD 21 èy$e$8 e$e$0p2A`<ĩ!
0000000370: CB E$
```

Представление «плохого» LAB1 COM.EXE:

```
view LAB1_COM.EXE - Far 3.0.5354 x64
D:\TASM\projects\lab1\LAB1_COM.EXE                               h 1252                1108 Col      0      100%  23:04
00000002A0: 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00
00000002B0: 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00
00000002C0: 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00
00000002D0: 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00
00000002E0: 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 22 01 50 43 20 54 79      70 65 3A 20 20 0D 0A 24  é"OPC Type:  J$
0000000310: 53 79 73 74 65 6D 20 76      65 72 73 69 6F 6E 3A 20  System version:
0000000320: 20 2E 20 20 00 0A 24 4F      45 4D 20 6E 75 6D 62 65  .  J$OEM numbe
0000000330: 72 3A 20 20 20 20 20 20      0D 0A 24 55 73 65 72 20  r:  J$User
0000000340: 73 65 72 69 61 6C 20 6E      75 6D 62 65 72 3A 20 20  serial number:
0000000350: 20 20 20 20 20 20 0A 24      20 20 20 20 20 0D 0A 24  J$
0000000360: B4 09 CD 21 C3 24 0F 3C      09 76 02 04 07 04 30 C3  'oif!A$<ov♦♦♦0Ä
0000000370: 51 8A C4 E8 EF FF 86 C4      B1 04 D2 E8 E8 E6 FF 59  Q5Äeiy†Ä†♦0ëeayY
0000000380: C3 53 8A FC E8 EF FF 88      25 4F 88 05 4F 8A C7 32  ÄSüeeÿ"%0*†ÄQRP2ä
0000000390: E4 E8 DC FF 88 25 4F 88      05 5B C3 51 52 50 32 E4  äeüÿ"%0*†ÄQRP2ä
00000003A0: 33 D2 B9 0A 00 F7 F1 80      CA 30 88 14 4E 33 D2 3D  30†ä ÷ñë0"¶N30=
00000003B0: 0A 00 73 F1 3D 00 00 76      04 0C 30 88 04 58 5A 59  ä sn=  v♦00"XZY
00000003C0: C3 06 53 50 BB 00 F0 E8      C3 26 A1 FE FF 8A E0 E8  Ä†SP» 0ZÄ&¡by5äe
00000003D0: 9E FF BB 03 01 89 47 09      58 5B 07 C3 50 56 BE 10  žÿ"♥0ëGoX[♦ÄPV%>
00000003E0: 01 83 C6 10 E8 BA FF 83      6C 03 8A C4 E8 AC FF 5E  0f†e-ë'ÿf†ÄSÄe-ÿ†
00000003F0: 58 C3 50 53 56 8A C7 BE      27 01 83 C6 0E E8 9B FF  XÄPSV$CK'0f†Äe>ÿ
0000000400: 5E 5B 58 C3 50 53 51 56      8A C3 E8 63 FF BF 3B 01  ^[XÄPSQV$Äecÿz;0
0000000410: 83 C7 16 89 05 8B C1 BF      3B 01 83 C7 1B E8 61 FF  fC-ë†♦Äz;0fC+ëaÿ
0000000420: 5E 59 58 C3 E8 99 FF      B4 30 CD 21 E8 AD FF E8  ^ÿ[XÄe"ÿ'oifë-ÿë
0000000430: C0 FF E8 CF FF BA 03 01      E8 25 FF BA 10 01 E8 1F  Äÿëÿë"♥0ë%ÿë-0ë♥
0000000440: FF BA 27 01 E8 19 FF BA      3B 01 E8 13 FF 32 C0 B4  ÿë"0ë†ÿë;0ëllÿ2Ä`
0000000450: 3C CD 21 C3
1Help      2Text      3Out      4Dump      5      6Edit      7Search      8ANSI      9      10Quit      11Plugins 12Screen
```

4)Открытие LAB1_COM.COM и LAB1_EXE.EXE в отладчике TD
LAB1_COM.COM:

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...

File Edit View Run Breakpoints Data Options Window Help READY

[]-CPU 80486

cs:0100	E92201	jmp	0225 ↓	ax	0000	c=0
cs:0103	50	push	ax	bx	0000	z=0
cs:0104	43	inc	bx	cx	0000	s=0
cs:0105	205479	and	[si+79],dl	dx	0000	o=0
cs:0108	7065	jo	016F	si	0000	p=0
cs:010A	3A20	cmp	ah,[bx+si]	di	0000	a=0
cs:010C	200D	and	[di],cl	bp	0000	i=1
cs:010E	0A24	or	ah,[si]	sp	FFFE	d=0
cs:0110	53	push	bx	ds	50DD	
cs:0111	7973	jns	0186	es	50DD	
cs:0113	7465	je	017A	ss	50DD	
cs:0115	6D	insw		cs	50DD	
cs:0116	207665	and	[bp+65],dh	ip	0100	

ds:0000 CD 20 FF 9F 00 EA FF FF = f Ω
ds:0008 AD DE E4 01 C9 15 AE 01 i 20 73 < Ω
ds:0010 C9 15 80 02 24 10 92 01 73 05 > Ω
ds:0018 01 01 01 00 02 FF FF FF 000 0

ss:0000 20CD
ss:FFFE 0000

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

LAB1_EXE.EXE:

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...

File Edit View Run Breakpoints Data Options Window Help READY

[]-CPU 80486

cs:00CF	1E	push	ds	ax	0000	c=0
cs:00D0	2BC0	sub	ax,ax	bx	0000	z=0
cs:00D2	50	push	ax	cx	0000	s=0
cs:00D3	B8ED50	mov	ax,50ED	dx	0000	o=0
cs:00D6	8ED8	mov	ds,ax	si	0000	p=0
cs:00D8	2BC0	sub	ax,ax	di	0000	a=0
cs:00DA	0E	push	cs	bp	0000	i=1
cs:00DB	E887FF	call	0065	sp	0400	d=0
cs:00DE	B430	mov	ah,30	ds	50DD	
cs:00E0	CD21	int	21	es	50DD	
cs:00E2	0E	push	cs	ss	5105	
cs:00E3	E89BFF	call	0081	cs	50F3	
cs:00E6	0E	push	cs	ip	00CF	

ds:0000 CD 20 FF 9F 00 EA FF FF = f Ω
ds:0008 AD DE E4 01 C9 15 AE 01 i 20 73 < Ω
ds:0010 C9 15 80 02 24 10 92 01 73 05 > Ω
ds:0018 01 01 01 00 02 FF FF FF 000 0

ss:0402 0000
ss:0400 0000

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Ответы на контрольные вопросы:

- **Отличия исходных текстов COM и EXE программ:**

1) Сколько сегментов должна содержать COM-программа?

COM-программа должна содержать один сегмент – сегмент кода

2) Сколько сегментов должна содержать EXE-программа?

EXE-программа может содержать 1 и более сегментов: сегмент стека, сегмент данных и сегмент кода.

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

В тексте COM-программы обязательно следующие директивы:

- a) ASSUME, которая должна указывать на то, что сегмент кода и данных начинается с одного и того же места.
- b) ORG, которая устанавливает счётчик положения в сегменте равным заданной величине, которая передаётся как параметр.

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

В COM-программе все сегментные регистры определяются в момент запуска программы, а не в момент компиляции (ассемблирования), поэтому в ней нельзя использовать команды вида `mov<регистр>, seg<имя сегмента>`.

В COM файле нет relocation table (таблицы настройки), поэтому он не может получить информацию об адресе сегмента. Следовательно нельзя использовать команды, которые используют адрес сегмента и дальнюю адресацию.

- **Отличия форматов файлов COM и EXE модулей:**

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

COM-файл содержит данные и машинные команды. Структура очень компактна (сам файл весит гораздо меньше). 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?*

В «плохом» EXE код и данные не разделены по сегментам, а перемешаны. Код располагается с адреса 300h, так как заголовок занимает 200h байт, а команда ORG 100h сдвигает код еще на 100h. С нулевого адреса располагается заголовок. В первых двух байтах можно увидеть символы MZ, означающие, что формат файла – 16-битный и его следует запускать в соответствии со структурой .EXE файлов. После заголовка идет таблица настройки. Без которой, файл загружался бы в память как .COM файл.

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

В «хорошем» EXE данные, стек и код разделены по сегментам. Структура несколько компактнее, чем структура «плохого» файла, так как в этом файле отсутствует директива ORG 100h, резервирующая пространство для заголовка. Из-за этого код располагается с адреса 200h, а не с 300h, как в «плохом» EXE-файле.

- **Загрузка COM-модуля в основную память:**

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

- a) *Определяется сегментный адрес участка ОП с достаточным местом для загрузки программы*
- b) *Создается блок памяти для PSP и программы;*
- c) *COM файл начинает свою загрузку с адреса 100h;*
- d) *Сегментные регистры CS, DS, ES, SS устанавливаются на начало PSP(0h), а регистр SP на конец PSP;*
- e) *В стек записывается – 0000, в регистр IP – 100h.*

Порядок загрузки модуля COM: PSP, данные и код, стек. Код начинается с адреса 100h.

2) *Что располагается с адреса 0?*

С нулевого адреса располагается PSP.

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

Все сегментные регистры имеют значение «50DD» и указывают на начало PSP.

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

Стек занимает всё свободное пространство до конца сегмента памяти (размер .COM файла не может превышать 64 кб), которое осталось после загрузки данных и кода. В данном случае значение регистра SP=FFFE.

- **Загрузка “хорошего” EXE-модуля в основную память:**

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

Порядок загрузки EXE-модуля:

- a) PSP;
- b) Сегмент кода;
- c) Сегмент данных;
- d) Сегмент стека.

Сегментные регистры на момент загрузки программы имеют значения: ES=50DD, CS=50F3, DS=50DD, SS=5105.

В начале выполнения программы, значения ES и DS совпадают, так как не были выполнены команды “mov ax, data; mov ds, ax”, т.е. в регистр данных не был занесен адрес сегмента данных.

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

ES - на начало PSP;

DS - на начало данных.

После выполнения команд (“mov ax, data; mov ds, ax”), значение DS=50ED.

3) *Как определяется стек?*

Стек определяется с помощью директивы «DW 512 DUP(?)» в описании сегмента стека.

4) *Как определяется точка входа?*

Точка входа определяется директивой END, после которой следует адрес, куда переходит программа при запуске.

Вывод: В ходе лабораторной работы было проведено сравнение текстов COM и EXE программ, их структуры, определены отличия файлов COM и EXE модулей, а так же исследованы способы загрузки их в память.