

**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. 9381

\_\_\_\_\_

Колованов Р.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

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

### **Функции и структуры данных.**

Разработанная программа использует следующие функции и структуры данных:

Название процедуры	Предназначение процедуры
TETR_TO_HEX	Переводит значение тетрады (4-ех младших битов регистра AL) в цифру 16-ичной СС и представляет ее в виде символа, который далее записывается в регистр AL.
BYTE_TO_HEX	Переводит значение байта (регистра AL) в число 16-ичной СС и представляет его в виде двух символов, которые далее записываются в регистры AL и AH.
WORD_TO_HEX	Переводит значение слова (регистра AX) в число 16-ичной СС и представляет его в виде четырех символов, которые далее записываются по адресу, на который указывает DI.
BYTE_TO_DEC	Переводит значение байта (регистра AL) в число 10-ичной СС и представляет его в виде символов, которые далее записываются по адресу, на который указывает SI.
PRINT	Вызывает функцию вывода строки на экран (функция 09h прерывания 21h).
PRINT_PC_TYPE	Печатает на экран тип ПК. Если версия ПК не идентифицирована, выводится байт в 16-ичной СС, который содержит информацию о типе ПК.

PRINT_PC_INFO	Печатает на экран версию ОС, серийные номера OEM и пользователя.
---------------	--

### **Последовательность действий программы.**

В ходе работы программа выполняет следующие действия:

- Вызывается процедура PRINT\_PC\_TYPE, которая выводит на экран тип ПК. Информация о типе ПК получается из предпоследнего байта ROM BIOS по адресу 0F000:0FFFEh. В зависимости от значения байта определяется тип: FFh – PC, FEh или FBh – PC/XT, FCh – AT, FAh – PS2 model 30, FCh – PS2 model 50 or 60, F8h – PS2 model 80, FDh – PCjr, F9h – PC Convertible. В случае, если значение не совпадает со значениями, приведенными выше, то в качестве типа ПК выводится значения предпоследнего байта ROM BIOS.
- Вызывается процедура PRINT\_PC\_INFO, которая выводит на экран версию ОС, серийные номера OEM и пользователя. Информация получается при помощи вызова функции 30h прерывания 21h.
- Завершение работы программы.

### **Ход работы.**

1. Для начала был написан текст исходного .COM модуля lab1\_com.asm. Далее при помощи транслятора MASM.EXE и компоновщика LINK.EXE был скомпилирован плохой .EXE модуль lab1\_com.exe с генерацией файла листинга и карты памяти. При помощи EXE2BIN.EXE по плохому .EXE модулю был построен хороший .COM модуль lab1\_com.com.
2. Далее был написан текст исходного .EXE модуля lab1\_exe.asm. При помощи транслятора MASM.EXE и компоновщика LINK.EXE был скомпилирован хороший .EXE модуль lab1\_exe.exe с генерацией файла листинга и карты памяти.
3. Далее было выполнено сравнение исходных текстов lab1\_com.asm и lab1\_exe.asm.

4. После при помощи программы FAR были открыты файлы загрузочных модулей lab1\_com.com, lab1\_com.exe и lab1\_exe.exe в шестнадцатеричном виде, далее было выполнено сравнение.
5. Далее был исследован загрузочный модуль .COM при помощи отладчика AFDPRO.EXE.
6. Далее был исследован хороший загрузочный модуль .EXE при помощи отладчика AFDPRO.EXE.

### **Результаты исследования проблем.**

#### ***План загрузки модуля .COM в основную память.***

1. Определяется сегментный адрес свободного участка основной памяти для загрузки программы.
2. Для программы формируется блок PSP и загружается в начало.
3. После блока PSP по адресу PSP:0100h загружается COM модуль.

После загрузки COM-программы в основную память CS, DS, ES и SS указывают на PSP, SP указывает на конец сегмента PSP, слово 00h помещено на стек, IP имеет значение 100h.

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

1. Сколько сегментов должна содержать COM-программа?

Один сегмент.

2. Сколько сегментов должна содержать EXE-программа?

Один или более сегментов. Их количество зависит от выбранной модели памяти.

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

Директива ORG. В начальной части COM-программы размещается специальный блок PSP (префикс программного сегмента), в начале которого размещена команда вызова обработчика прерывания для завершения программы

и возврата в DOS. Так как после загрузки все сегментные регистры, включая CS, указывают на начало PSP, а IP = 0, то программа не может исполняться, начиная с этого адреса. Поэтому требуется директива ORG 100h, которая установит CS:IP на конец PSP (размер PSP равен 256 байт, значит сдвиг нужно делать на 256 = 100h байт).

Директива ASSUME. Указывает ассемблеру, с каким сегментом или группой сегментов связан тот или иной сегментный регистр. Она не изменяет значений сегментных регистров, а только позволяет ассемблеру проверять допустимость ссылок и самостоятельно вставлять при необходимости префиксы переопределения сегментов. Без этой директивы программа не скомпилируется, так как ассемблер не будет понимать, относительно чего вычислять смещения меток.

Директива END. Этой директивой завершается любая программа на ассемблере.

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

Нет. Поскольку в COM-программе отсутствует таблица настройки, использование таких форматов команд не доступно:

seg NAME, где NAME – название сегмента.

#### ***Отличия форматов файлов COM и EXE модулей.***

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

В COM-файле код, данные и стек располагаются в одном сегменте. Код (как и данные) начинается с адреса 0h. Файл в 16-ичном представлении расположен снизу.

0000000000:	E9 FC 01 50 43 20 74 79	70 65 3A 20 24 50 43 0D	éü@PC 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\$ \$0
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	♦♦♦AQŠàèiÿ†Ä±♦
00000000D0:	D2 E8 E8 E6 FF 59 C3 53	8A FC E8 E9 FF 88 25 4F	ÒèèæÿYÄSSüèéÿ~%0
00000000E0:	88 05 4F 8A C7 E8 DE FF	88 25 4F 88 05 5B C3 51	~♦OŠÇèbÿ~%0~♦[ÄQ
00000000F0:	52 32 E4 33 D2 B9 0A 00	F7 F1 80 CA 30 88 14 4E	R2ä3D¹ ÷ñ€Ê0~JN
0000000100:	33 D2 3D 0A 00 73 F1 3C	00 74 04 0C 30 88 04 5A	3D= sñ< t♦♀0~♦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	ŽĀ& þÿ°♥Oèæÿ<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	ë1®®®®®®®®®®®®®®
0000000160:	1A 01 EB 2C 90 BA 1F 01	EB 26 90 BA 2E 01 EB 20	→0ë,®®®®®®®®®®®®
0000000170:	90 BA 43 01 EB 1A 90 BA	52 01 EB 14 90 BA 59 01	®®C0ë→®®R0ë®®Y0
0000000180:	EB 0E 90 E8 40 FF BB 6A	01 88 07 88 67 01 8B D3	ë®®®®®®®®®®®®®®
0000000190:	E8 7F FF 07 5A 5B 58 C3	50 53 51 52 56 57 B4 30	èøÿ•Z[XÄPSQRVW´0
00000001A0:	CD 21 BE 7D 01 8A D4 E8	45 FF 8A C2 83 C6 03 E8	Í!%}0Š0èEÿŠÄf♥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°®0è
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“ÿ2Ä´LÍ!

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

Что располагается с адреса 0?

В плохом EXE-файле код, данные и стек располагаются в одном сегменте. Код (как и данные) начинается с адреса 300h. С адреса 0h располагается управляющая информация для загрузчика, которая содержит заголовок и таблицу настройки адресов. Файл в 16-ичном представлении расположен снизу.

0000000000:	4D 5A 0B 01 03 00 00 00	20 00 00 00 FF FF 00 00	MZó♥	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	♦♦♦♦AQŠàèiÿ†Ä±♦
00000003D0:	D2 E8 E8 E6 FF 59 C3 53	8A FC E8 E9 FF 88 25 4F	ðèèæÿYÄŠSüèéÿ~%0
00000003E0:	88 05 4F 8A C7 E8 DE FF	88 25 4F 88 05 5B C3 51	^+OŠÇèÿ~%0^+ [ÄQ
00000003F0:	52 32 E4 33 D2 B9 0A 00	F7 F1 80 CA 30 88 14 4E	R2ä3Ò¹ ÷ñÊÖ~JN
0000000400:	33 D2 3D 0A 00 73 F1 3C	00 74 04 0C 30 88 04 5A	3Ò= sñ< t♦♀0^♦Z
0000000410:	59 C3 50 B4 09 CD 21 58	C3 50 53 52 06 B8 00 F0	YÄP´oÍ!XÄPSR▲. ð
0000000420:	8E C0 26 A0 FE FF BA 03	01 E8 E6 FF 3C FF 74 23	Ž& ðÿ°♥Oèæÿ<ÿt#
0000000430:	3C FE 74 25 3C FD 74 21	3C FC 74 23 3C FA 74 25	<ðt%<ÿt!<üt#<üt%
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	ë1OèOè8Oè4Oè2Oè
0000000460:	1A 01 EB 2C 90 BA 1F 01	EB 26 90 BA 2E 01 EB 20	→Oè,Oè♥Oè&Oè.Oè
0000000470:	90 BA 43 01 EB 1A 90 BA	52 01 EB 14 90 BA 59 01	OèCOè→OèROèJOèYOè
0000000480:	EB 0E 90 E8 40 FF BB 6A	01 88 07 88 67 01 8B D3	èJèè@ÿ»jO^•^gO<Ó
0000000490:	E8 7F FF 07 5A 5B 58 C3	50 53 51 52 56 57 B4 30	èoÿ•Z[XÄPSQRVW´0
00000004A0:	CD 21 BE 7D 01 8A D4 E8	45 FF 8A C2 83 C6 03 E8	Í!¼}OŠOèEÿŠÄfA♥è
00000004B0:	3D FF BA 6F 01 E8 5A FF	BA 7C 01 E8 54 FF 8A C7	=ÿ°oOèZÿ° OèTÿŠÇ
00000004C0:	E8 03 FF BF 98 01 88 05	88 65 01 BA 84 01 E8 41	è♥ÿ¿~O^+^eO„OèA
00000004D0:	FF BA 98 01 E8 3B FF 8A	C3 E8 EA FE BF B2 01 88	ÿ°~Oè;ÿŠÄèèþ¿²O^
00000004E0:	05 88 65 01 8B C1 83 C7	05 E8 EB FE BA 9D 01 E8	+^eO<ÄfÇ+èèþ°Oè
00000004F0:	20 FF BA B2 01 E8 1A FF	5F 5E 5A 59 5B 58 C3 E8	ÿ°²Oè→ÿ_^ZY[XÄè
0000000500:	17 FF E8 93 FF 32 C0 B4	4C CD 21	±ÿè“ÿ2Ä´LÍ!

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

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 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▶ ▲ @ X@
0000000020:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	

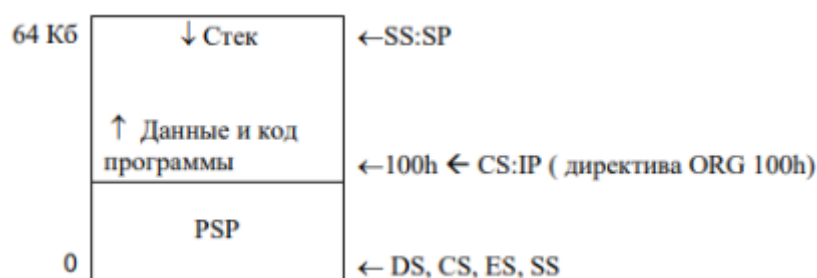


00000000200:	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	éD@ſα<ov0♦♦♦0ÃQŠ
00000000210:	E9 44 01 24 0F 3C 09 76	02 04 07 04 30 C3 51 8A	àèïÿ†Ä±♦0èèæÿYÃS
00000000220:	E0 E8 EF FF 86 C4 B1 04	D2 E8 E8 E6 FF 59 C3 53	Šüèéÿ~%0^+0ŠÇèþÿ
00000000230:	8A FC E8 E9 FF 88 25 4F	88 05 4F 8A C7 E8 DE FF	~%0^+ [ÃQR2ä30¹
00000000240:	88 25 4F 88 05 5B C3 51	52 32 E4 33 D2 B9 0A 00	÷ñ€Ê0^ſJN30= sñ<
00000000250:	F7 F1 80 CA 30 88 14 4E	33 D2 3D 0A 00 73 F1 3C	t♦Q0^♦ZYÃP´oÍ!X
00000000260:	00 74 04 0C 30 88 04 5A	59 C3 50 B4 09 CD 21 58	ÃPSR♦. ðŽÀ& þÿ°
00000000270:	C3 50 53 52 06 B8 00 F0	8E C0 26 A0 FE FF BA 08	èæÿ<ÿt#<þt%<ÿt!
00000000280:	00 E8 E6 FF 3C FF 74 23	3C FE 74 25 3C FD 74 21	<ÿt#<ÿt%<ÿt'<øt)
00000000290:	3C FC 74 23 3C FA 74 25	3C FC 74 27 3C F8 74 29	<ÿt+<ÿt-ë100± ë8
000000002A0:	3C FD 74 2B 3C F9 74 2D	EB 31 90 BA 12 00 EB 38	00± ë200▼ ë,00\$
000000002B0:	90 BA 17 00 EB 32 90 BA	1F 00 EB 2C 90 BA 24 00	ë&003 ë 00H ë→00
000000002C0:	EB 26 90 BA 33 00 EB 20	90 BA 48 00 EB 1A 90 BA	w ëſ00^ ëſ00è@ÿ»o
000000002D0:	57 00 EB 14 90 BA 5E 00	EB 0E 90 E8 40 FF BB 6F	^•^g0<0èαÿ•Z[XÃ
000000002E0:	00 88 07 88 67 01 8B D3	E8 7F FF 07 5A 5B 58 C3	PSQRVW´0Í!%, Š0è
000000002F0:	50 53 51 52 56 57 B4 30	CD 21 BE 82 00 8A D4 E8	EÿŠÃfA♥è=ÿøt èZÿ
00000000300:	45 FF 8A C2 83 C6 03 E8	3D FF BA 74 00 E8 5A FF	00 èTÿŠÇè♥ÿ;0 ^+
00000000310:	BA 81 00 E8 54 FF 8A C7	E8 03 FF BF 9D 00 88 05	^e0%0 èAÿ00 è;ÿŠ
00000000320:	88 65 01 BA 89 00 E8 41	FF BA 9D 00 E8 3B FF 8A	Ãèèþ;• ^+^e0<ÁfÇ
00000000330:	C3 E8 EA FE BF B7 00 88	05 88 65 01 8B C1 83 C7	+èèþøø è ÿø• è→ÿ
00000000340:	05 E8 EB FE BA A2 00 E8	20 FF BA B7 00 E8 1A FF	^ZY[XÃ,= Ž0è†ÿè
00000000350:	5F 5E 5A 59 5B 58 C3 B8	16 00 8E D8 E8 12 FF E8	Žÿ2Ã´LÍ!PC type:
00000000360:	8E FF 32 C0 B4 4C CD 21	50 43 20 74 79 70 65 3A	\$PCſ0\$PC/XTſ0\$A
00000000370:	20 24 50 43 0D 0A 24 50	43 2F 58 54 0D 0A 24 41	Tſ0\$PS2 model 30
00000000380:	54 0D 0A 24 50 53 32 20	6D 6F 64 65 6C 20 33 30	ſ0\$PS2 model 50
00000000390:	0D 0A 24 50 53 32 20 6D	6F 64 65 6C 20 35 30 20	or 60ſ0\$PS2 mode
000000003A0:	6F 72 20 36 30 0D 0A 24	50 53 32 20 6D 6F 64 65	l 80ſ0\$PCjrſ0\$PC
000000003B0:	6C 20 38 30 0D 0A 24 50	43 6A 72 0D 0A 24 50 43	Convertibleſ0\$
000000003C0:	20 43 6F 6E 76 65 72 74	69 62 6C 65 0D 0A 24 20	ſ0\$OS version:
000000003D0:	20 0D 0A 24 4F 53 20 76	65 72 73 69 6F 6E 3A 20	\$ . ſ0\$OEM ser
000000003E0:	24 20 20 2E 20 20 0D 0A	24 4F 45 4D 20 73 65 72	ial number: \$ ſ0\$
000000003F0:	69 61 6C 20 6E 75 6D 62	65 72 3A 20 24 20 20 0D	0\$User serial nu
00000000400:	0A 24 55 73 65 72 20 73	65 72 69 61 6C 20 6E 75	mber: \$ ſ0\$
00000000410:	6D 62 65 72 3A 20 24 20	20 20 20 20 20 0D 0A 24	

### Загрузка СОМ модуля в основную память.

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

СОМ-модуль имеет следующую структуру:



Код располагается с адреса 100h, поскольку с адреса 0h располагается блок PSP размером 256 байт.



AX	0000	SI	0000	CS	19F5	IP	0100	Stack	+0	0000
BX	0000	DI	0000	DS	19F5				+2	20CD
CX	020B	BP	0000	ES	19F5	HS	19F5		+4	9FFF
DX	0000	SP	FFFE	SS	19F5	FS	19F5		+6	EA00

2. Что располагается с адреса 0?

Специальный блок PSP (префикс программного сегмента).

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

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

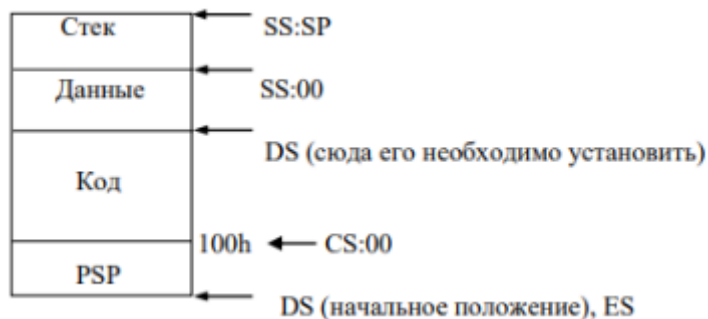
4. Как определяется стек? Какую область памяти он занимает? Какие адреса?

Регистр SS указывает на начало блока PSP (0h), а SP указывает на конец модуля (FFFFh). То есть стек расположен между адресами SS:0000h и SS:FFFFh и заполняется с конца модуля в сторону уменьшения адресов.

### ***Загрузка «хорошего» EXE модуля в основную память.***

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

Регистры DS и ES указывают на начало блока PSP, регистр CS указывает на начало сегмента кода, а регистр SS – на начало сегмента стека. Структура выглядит следующим образом:



AX	0000	SI	0000	CS	1A05	IP	0010	Stack	+0	0000
BX	0000	DI	0000	DS	19F5				+2	0000
CX	0220	BP	0000	ES	19F5	HS	19F5		+4	0000
DX	0000	SP	0100	SS	1A27	FS	19F5		+6	0000

2. На что указывают регистры DS и ES?

Регистры DS и ES указывают на начало блока PSP.

3. Как определяется стек?

Регистр SS указывает на начало сегмента стека, а SS: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.**

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