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

ОТЧЕТ

по лабораторной работе №5

по дисциплине «Операционные Системы»

Тема: Исследование структур загрузочных модулей

Студент гр. 9382 _____ Иерусалимов Н.

Преподаватель _____ Ефремов 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.

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 формировать текстовую строку в формате xx.yy, где xx – номер основной версии, а yy - номер модификации в десятичной системе счисления, формировать 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.

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

Таблица 1 – функции в программе

Выполнение работы.

Были написаны следующие процедуры:

PRINT – Выводит строку на экран.(Какие строки были объявлены смотрите ниже)

FIND_INFO_PC – Находит информацию о компьютере, после чего происходит сравнение с таблицей(рис.4), если были совпадения с ней, заносит в регистр dx нужную строку, после чего вызывается процедура для вывода на экран PRINT.

FIND_OS_VERSION – С помощью прерывания 30h в регистр ah заносится информация о версии DOSBOX. После этого мы переводим 16-ричную запись в 10 и выводим на экран значение версии, серийный номер и серийный номер пользователя.

Были объявлены строки для вывода информации:

1. T_PC db 'Type: PC',0DH,0AH,'$'
2. T_XT db 'Type: PC/XT',0DH,0AH,'$'
3. T_AR db 'Type: AT',0DH,0AH,'$'
4. TPS2_30 db 'Type: PS2 модель 30',0DH,0AH,'$'

5. TPS2_80 db 'Type: PS2 модель 80',0DH,0AH,'$'
6. T_JR db 'Type: PCjr',0DH,0AH,'$'
7. T_CONV db 'Type: PC Convertible',0DH,0AH,'$'
8. VERSIONS db 'Version MS-DOS: . ',0DH,0AH,'$'
9. N_SERIAL db 'Serial number OEM: ',0DH,0AH,'$'
10. N_USER db 'User serial number: H $'

PC	FF
PC/XT	FE,FB
AT	FC
PS2 модель 30	FA
PS2 модель 50 или 60	FC
PS2 модель 80	F8
PCjr	FD
PC Convertible	F9

Рис.4

В результате работы программы получаем такой вывод(рис 1-3).

```
C:\>MAIN_EXE.EXE
Type: AT
Version MS-DOS: 5.0
Serial number OEM: 0
User serial number: 000000H
```

Рисунок 1 – “хороший EXE модуль”

Рисунок 2 – “Плохой EXE модуль”

```
C:\>MAIN_COM.COM
Type: AT
Version MS-DOS: 5.0
Serial number OEM: 0
User serial number: 000000H
```

Рисунок 3 – “Хороший .COM модуль”

Выводы.

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

Приложение А.

Ответы на контрольные вопросы.

Отличия исходных текстов COM и EXE программ:

1. **Сколько сегментов должна сдержать COM-программа?**

Программа содержит один сегмент. В этом сегменте находиться Код и данные, а стек генерируется автоматически.

2. **EXE-программа?**

Должна содержать не менее одного сегмента. Сегменты кода, данных и стека описываются отдельно друг от друга, но есть возможность не описывать сегмент стека, в таком случае будет использоваться стек DOS.

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

Должна быть обязательна директива ORG 100h, так как при загрузке модуля все сегментные регистры содержат адрес префикса программного сегмента (PSP), который является 256-байтовым(100H) блоком, поэтому адресация имеет смещение в 256 байт от нулевого адреса. Также необходима процедура ASSUME для того, чтобы сегмент данных и сегмент кода

указывали на один общий сегмент. (ASSUME CS:TESTPC, DS:TESTPC, ES:NOTHING, SS:NOTHING)

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

Не все форматы поддерживаются. Нельзя использовать команды вида mov <регистр>, seg <имя сегмента>, так как в .com-программе отсутствует таблица настроек (содержит описание адресов, которые зависят от размещения загрузочного модуля в ОП).

Отличия форматов файлов .COM и.EXE программ:

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

COM-файл состоит из одного сегмента, состоящего из сегмент кода и сегмент данных, сегмент стека генерируется автоматически при создании COM-программы. 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?

В «плохого» EXE данные и код располагаются в одном сегменте, что для EXE файла некорректно, так как код и данные должны быть разделены на отдельные сегменты. Код располагается с адреса 300h, а с адреса 0h идёт таблица настроек.

```
C:\Users\niki_\OneDrive\Рабочий стол\Stud\2-2\OS\Lab1\src\EXE\MAIN_COM.EXE
0000000000: 4D 5A CD 00 03 00 00 00  20 00 00 00 FF FF 00 00  MZH ♥       яя
0000000010: 00 00 95 42 00 01 00 00  1E 00 00 00 01 00 00 00    •B ☺  ▲   ☺
0000000020: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000030: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000040: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000050: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000060: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000070: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000080: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000090: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000000A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000000B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000000C0: 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
00000000E0: 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
0000000100: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000110: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000120: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000130: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000140: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000150: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000160: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000170: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000180: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000190: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000001A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000001B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000001C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000001D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000001E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000001F0: 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
0000000210: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000220: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000230: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000240: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000250: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000260: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000270: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000280: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000290: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
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 BE 01 54 79 70 65 3A  20 50 43 0D 0A 24 54 79  йs☺Type: PC♪◙$Ty
0000000310: 70 65 3A 20 50 43 2F 58  54 0D 0A 24 54 79 70 65  pe: PC/XT♪◙$Type
0000000320: 3A 20 41 54 0D 0A 24 54  79 70 65 3A 20 50 53 32  : AT♪◙$Type: PS2
0000000330: 20 EC EE E4 E5 EB FC 20  33 30 0D 0A 24 54 79 70   модель 30♪◙$Typ
0000000340: 65 3A 20 50 53 32 20 EC  EE E4 E5 EB FC 20 38 30  e: PS2 модель 80
0000000350: 0D 0A 24 54 79 70 65 3A  20 50 D1 6A 72 0D 0A 24  ♪◙$Type: PCjr♪◙$
0000000360: 54 79 70 65 3A 20 50 43  20 43 6F 6E 76 65 72 74  Type: PC Convert
0000000370: 69 62 6C 65 0D 0A 24 56  65 72 73 69 6F 6E 20 4D  ible♪◙$Version M
0000000380: 53 2D 44 4F 53 3A 20 20  2E 20 20 0D 0A 24 53 65  S-DOS:   .  ♪◙$Se
0000000390: 72 69 61 6C 20 6E 75 6D  62 65 72 20 4F 45 4D 3A  rial number OEM:
00000003A0: 20 20 0D 0A 24 55 73 65  72 20 73 65 72 69 61 6C    ♪◙$User serial
00000003B0: 20 6E 75 6D 62 65 72 3A  20 20 20 20 20 20 20 48   number:       H
00000003C0: 20 24 24 0F 3C 09 76 02  04 07 04 30 C3 51 8A E0   $$☼<○v☻♦•♦0ГQЉa
```

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

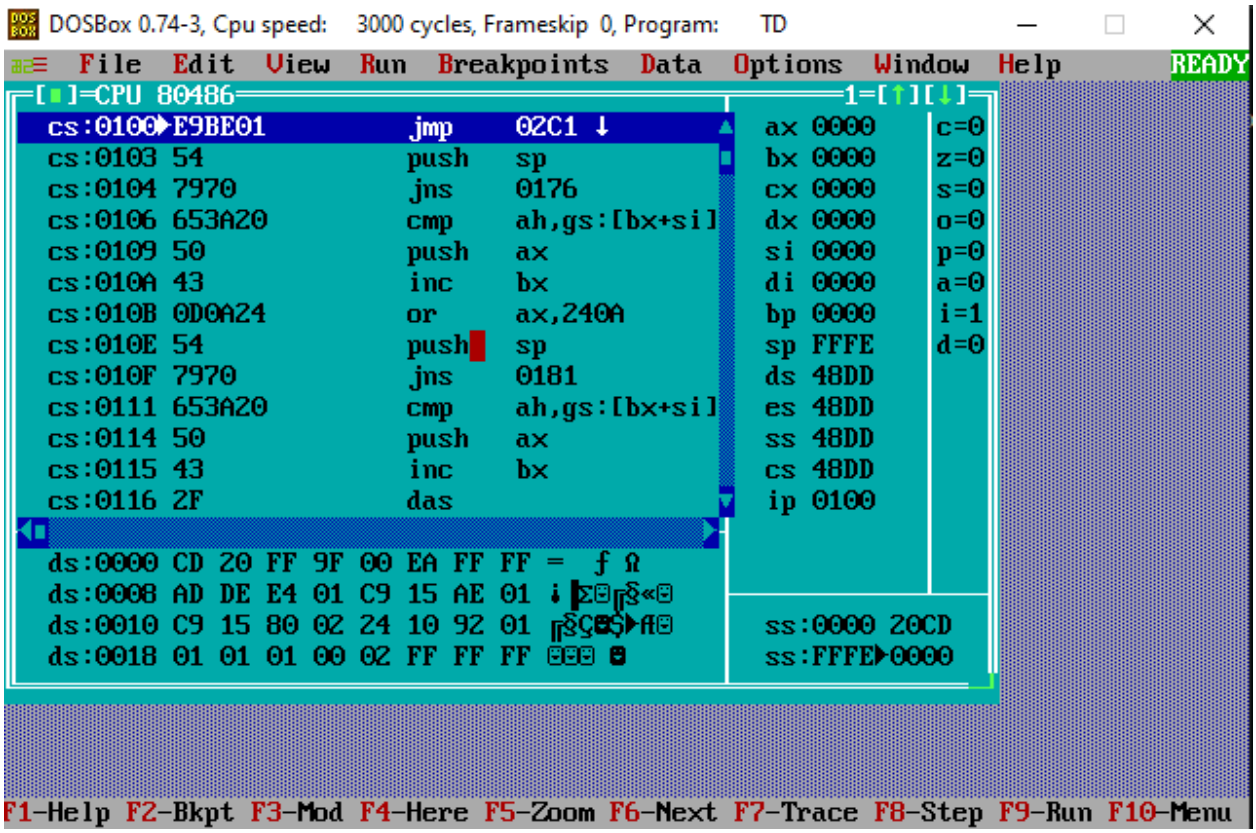
В EXE-программе код, данные и стек поделены на сегменты. Программа в формате EXE может иметь любой размер. EXE-файл имеет заголовок, который используется при его загрузке. Заголовок состоит из форматированной части, содержащей сигнатуру и данные, необходимые для загрузки EXE-файла, и таблицы для настройки адресов. В отличии от «плохого» EXE в «хорошем» EXE присутствуют три сегмента: сегмент кода, сегмент данных и сегмент стека, а «плохой» EXE содержит один сегмент, совмещающий код и данные. Также в «плохом» EXE адресация кода начинается с 300h, так как он получается из .COM файла, в котором изначально сегмент кода смещён на 100h, а при создании «плохого» EXE к этому смещению добавляется размер PSP модуля(200h). А в «хорошем» EXE присутствует только смещение для PSP модуля, поэтому код начинается с 200h. В данной случае смещение кода 400h так как выделяется память под стек (200h), память под стек находится между PSP и кодом.

```
C:\Users\niki_\OneDrive\Рабочий стол\Stud\2-2\OS\Lab1\src\EXE\MAIN_EXE.EXE
0000000000: 4D 5A F3 00 03 00 01 00  20 00 00 00 FF FF 00 00  MZy ♥ ☺     яя
0000000010: 00 01 9E AB FF 00 1E 00  1E 00 00 00 01 00 03 01   ☺h«я ▲ ▲   ☺ ♥☺
0000000020: 1E 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ▲
0000000030: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000040: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000050: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000060: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000070: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000080: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000090: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000000A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000000B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000000C0: 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
00000000E0: 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
0000000100: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000110: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000120: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000130: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000140: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000150: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000160: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000170: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000180: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000190: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000001A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000001B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000001C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000001D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000001E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000001F0: 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
0000000210: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000220: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000230: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000240: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000250: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000260: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000270: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000280: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000000290: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
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: 54 79 70 65 3A 20 50 43  0D 0A 24 54 79 70 65 3A  Type: PC♪◙$Type:
0000000310: 20 50 43 2F 58 54 0D 0A  24 54 79 70 65 3A 20 41   PC/XT♪◙$Type: A
0000000320: 54 0D 0A 24 54 79 70 65  3A 20 50 53 32 20 EC EE  T♪◙$Type: PS2 мо
0000000330: E4 E5 EB FC 20 33 30 0D  0A 24 54 79 70 65 3A 20  дель 30♪◙$Type:
0000000340: 50 53 32 20 EC EE E4 E5  EB FC 20 35 30 20 E8 EB  PS2 модель 50 ил
0000000350: E8 20 36 30 0D 0A 24 54  79 70 65 3A 20 50 53 32  и 60♪◙$Type: PS2
0000000360: 20 EC EE E4 E5 EB FC 20  38 30 0D 0A 24 54 79 70   модель 80♪◙$Typ
0000000370: 65 3A 20 50 D1 6A 72 0D  0A 24 54 79 70 65 3A 20  e: PCjr♪◙$Type:
0000000380: 50 43 20 43 6F 6E 76 65  72 74 69 62 6C 65 0D 0A  PC Convertible♪◙
0000000390: 24 56 65 72 73 69 6F 6E  20 4D 53 2D 44 4F 53 3A  $Version MS-DOS:
00000003A0: 20 20 2E 20 20 0D 0A 24  53 65 72 69 61 6C 20 6E    .  ♪◙$Serial n
00000003B0: 75 6D 62 65 72 20 4F 45  4D 3A 20 20 0D 0A 24 55  umber OEM:  ♪◙$U
00000003C0: 73 65 72 20 73 65 72 69  61 6C 20 6E 75 6D 62 65  ser serial numbe
```

Загрузка COM модуля в основную память:

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

Определяется сегментный адрес участка ОП, у которого достаточно места для загрузки программы, образ COM-файла считывается с диска и помещается в память, начиная с PSP:0100h. После загрузки двоичного образа COM-программы сегментные регистры CS, DS, ES и SS указывают на PSP(в данном случае сегментные регистры указывают на 48DD), SP указывает на конец сегмента PSP(обычно FFFE), слово 00H помещено в стек, IP содержит 100H в результате команды JMP PSP:100H.



2. **Что располагается с адреса 0?**

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

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

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

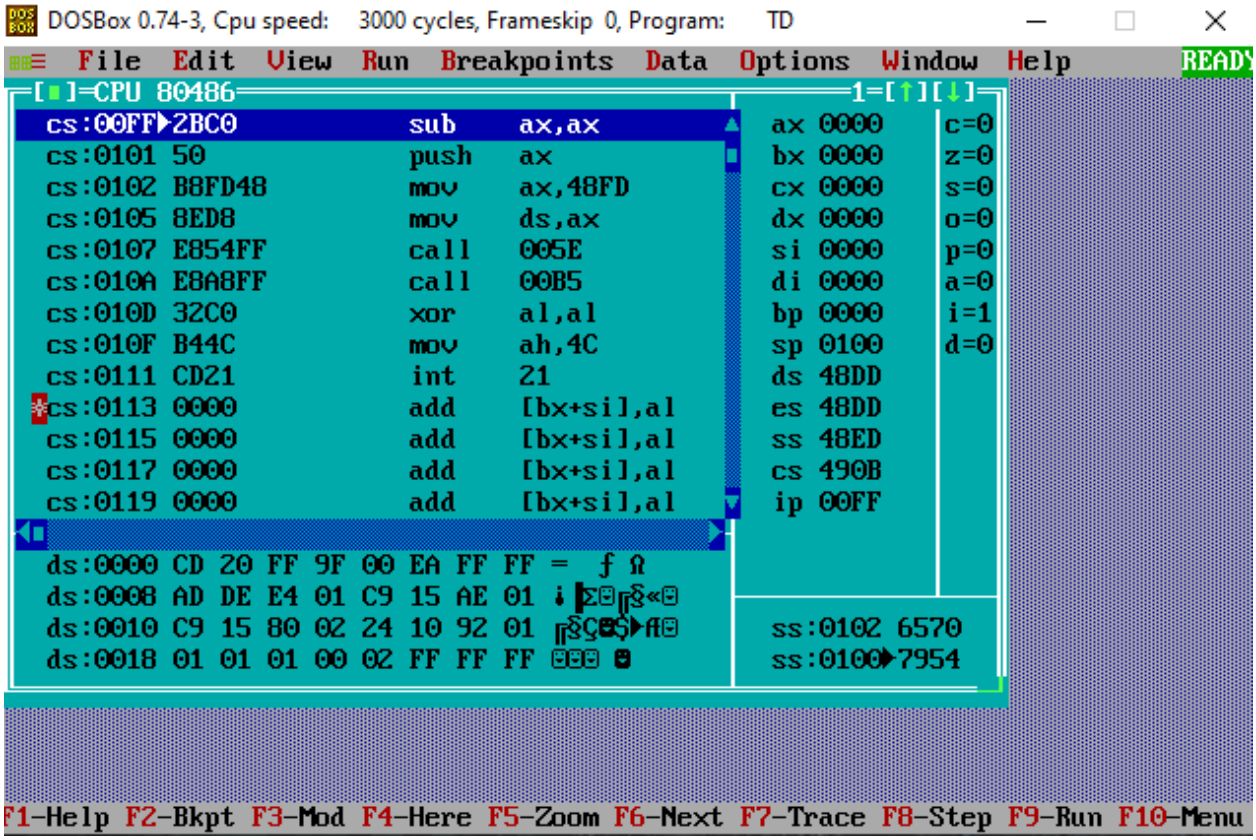
4. Как определяется стек? Какую область памяти он занимает? Какие адреса?

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:

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=490D) – на начало сегмента команд. В IP загружается смещение точки входа в программу, которая берётся из метки после директивы END. Причём дополнительный программный сегмент (PSP) присутствует в каждом EXE-файле.



2. На что указывают регистры DS и ES?

Регистры DS и ES указывают на начало сегмента PSP.

3. Как определяется стек?

Стек определяется с помощью директивы .stack, после которой задаётся размер стека. При исполнение 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.