**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

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6381 |  | Лопатина А.С. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

**Постановка задачи**

**Цель работы.**

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

**Необходимые сведения для составления программы.**

**Тип IBM PC** хранится в байте по адресу 0F000:0FFFE, в предпоследнем байте ROM BIOS. Соответствие кода и типа в таблице:

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

Для определения **версии MS DOS** следует воспользоваться функцией 30H прерывания 21H. Входным параметром является номер функции в AH:

MOV AH,30h

INT 21h

Выходными параметрами являются:

AL – номер основной версии. Если 0, то <2.0;

AH – номер модификации;

BH – серийный номер OEM (Original Equipment Manufacturer);

BL:CX – 24-битовый серийный номер пользователя;

**Сведения о функциях и структурах данных управляющей программы.**

TETR\_TO\_HEX – перевод полвины байта из 2-ой СС в 16-ую СС;

BYTE\_TO\_HEX – перевод байта, расположенного в AL, в 2 символа 16-ой числа в AX;

WRD\_TO\_HEX – перевод в 16-ую СС 16-ти разрядного числа;

BYTE\_TO\_DEC – перевод в 10-ую СС;

OUTPUT\_PROC – вывод на экран;

DET\_TYPE – получение типа РС;

DET\_VERSION – получение версии системы;

DET\_OEM\_NUM – получение серийного номера ОЕМ;

DET\_USER\_NUM – получение серийного номера пользователя;

**Последовательность действий, выполняемых программой.**

Программа определяет тип PC, версию системы, серийный номер OEM и серийный номер пользователя, затем сохраняет их в соответствующие переменные выводит полученные значения на экран.

**Результат работы программы.**

1. «Хороший» .COM модуль

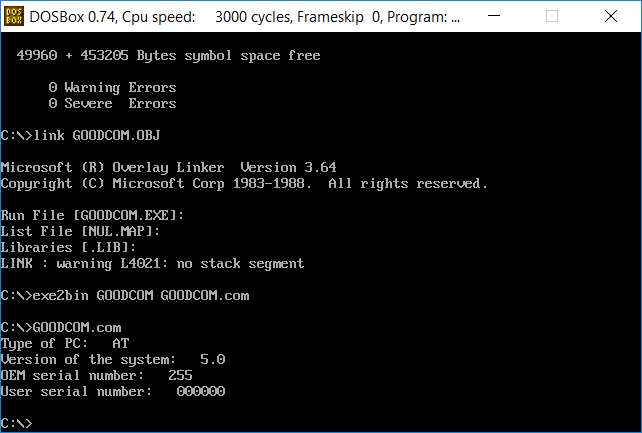


рис.1. «Хороший» .COM модуль

1. «Плохой» .EXE модуль

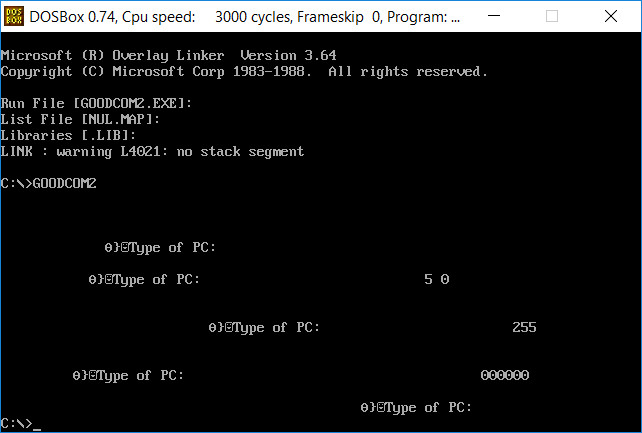


рис.2. «Плохой» .EXE модуль

1. «Хороший» .EXE модуль

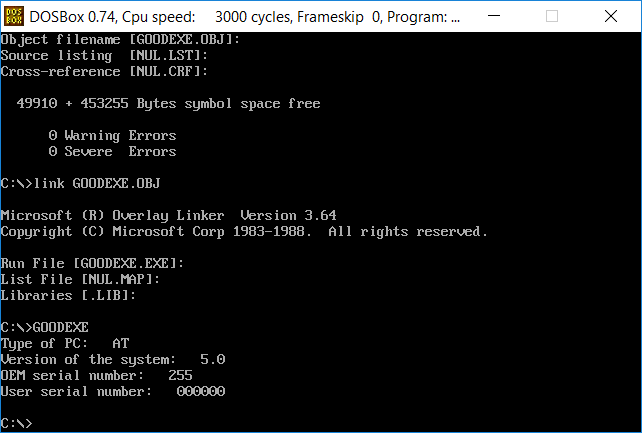


рис.3. «Хороший» .EXE модуль

**Описание результатов исследования проблем, поставленных в лабораторной работе**

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

1. Сколько сегментов должна содержать COM программа?

Ответ: СОМ программа должна содержать один сегмент;

1. EXE программа?

Ответ: ЕХЕ программа должна содержать более одного сегмента. Эта программа предусматривает содержание сегмента стека, кода и данных.

1. Какие директивы должны обязательно быть в тексте COM программы?

Ответ: в программе обязательно должна быть директива ORG 100h, которая устанавливает значение программного счетчика в 100h, так как при загрузке СОМ-файла в память DOS занимает первые 256 байт блоком данных PSP и располагает код программы после этого блока.

1. Все ли форматы команд можно использовать в COM программе?

Ответ: не все форматы команд можно использовать в СОМ программе, так как в СОМ-программе нет таблицы разметки Relocation Table, вследствие чего ей неоткуда получить информацию об адресе сегмента. А так как в СОМ-программе все сегментные регистры определяются в момент запуска на основе информации о положении полей адресов в файле из Relocation Table, то нельзя использовать команды вида mov <регистр>, seg <имя сегмента>;

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

Запускаем Far и открываем в нём .СОМ модуль, «плохой» .ЕХЕ модуль и «хороший» .ЕХЕ модуль в шестнадцатеричном виде.

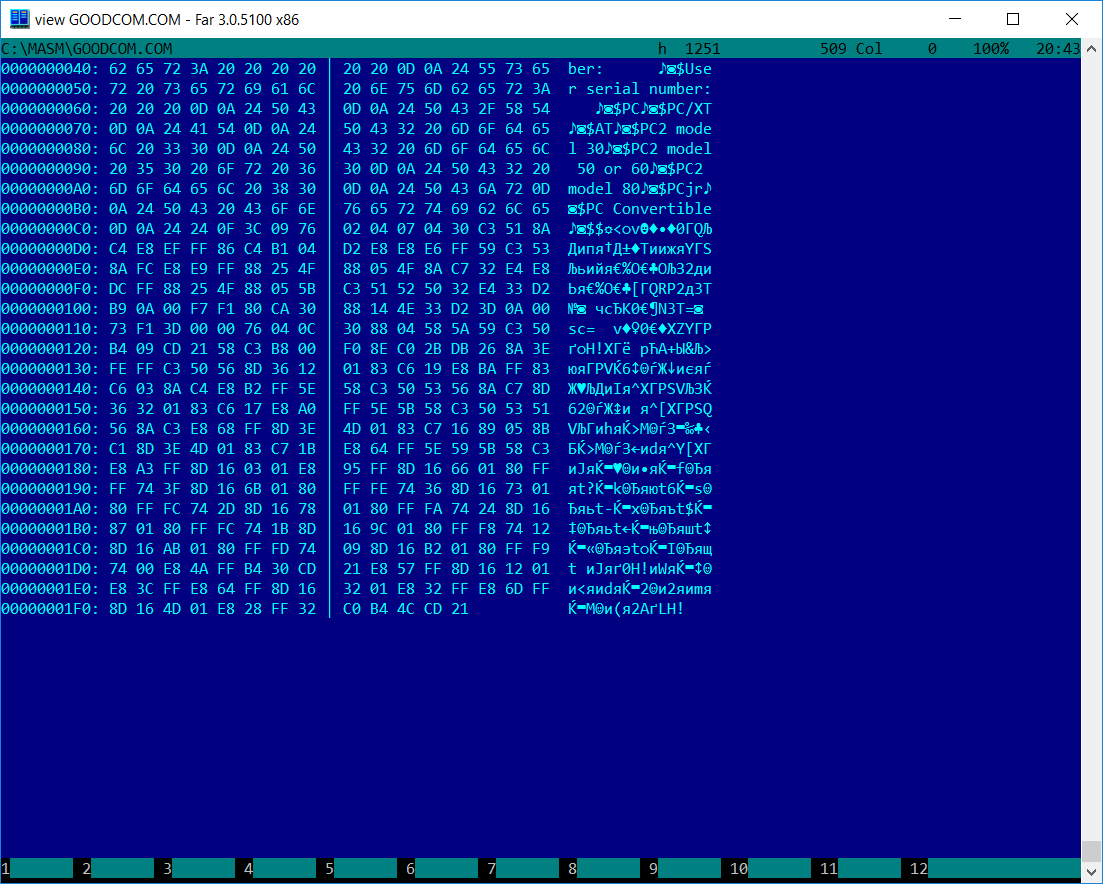


рис.4. «Хороший» .COM модуль в Far

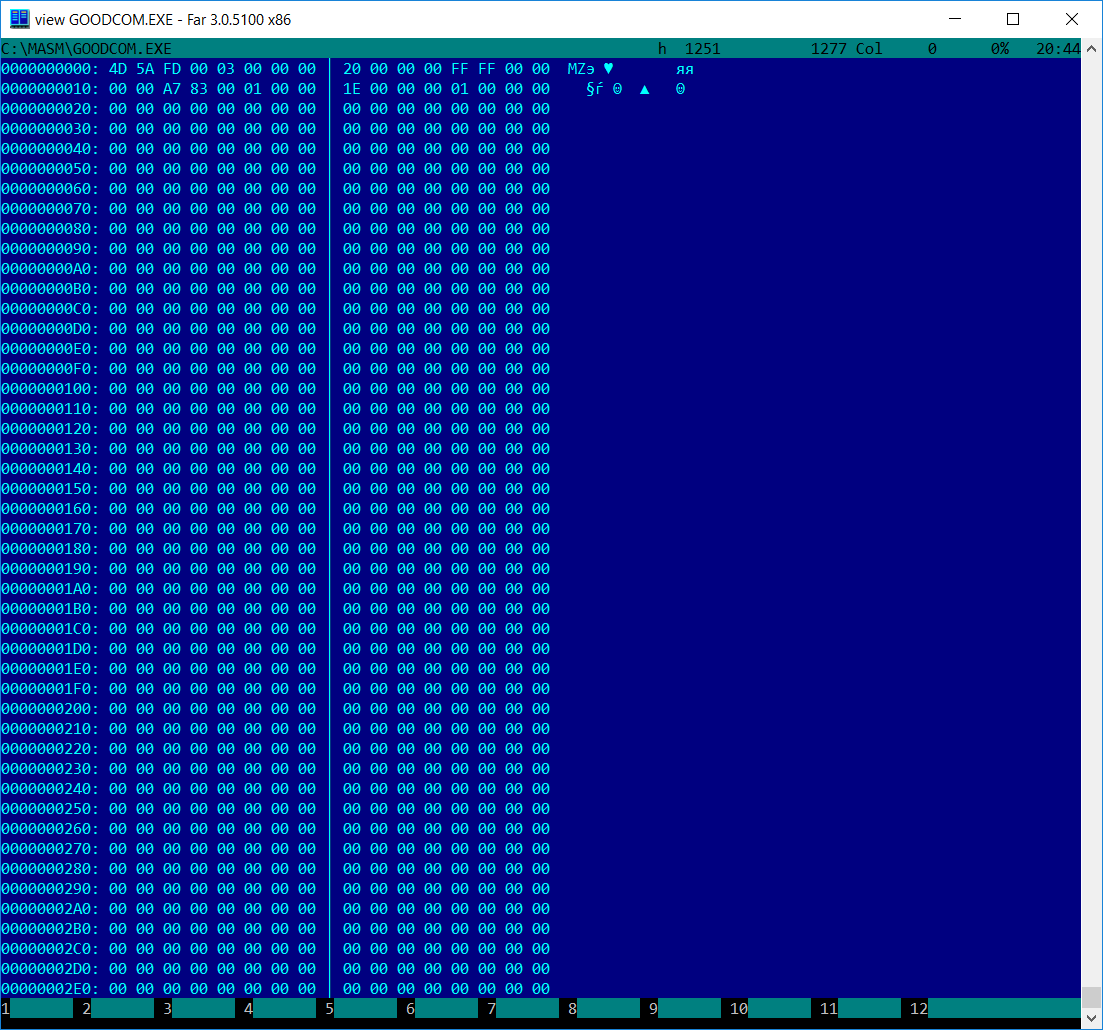


рис.5. «Плохой» .EXE модуль в Far

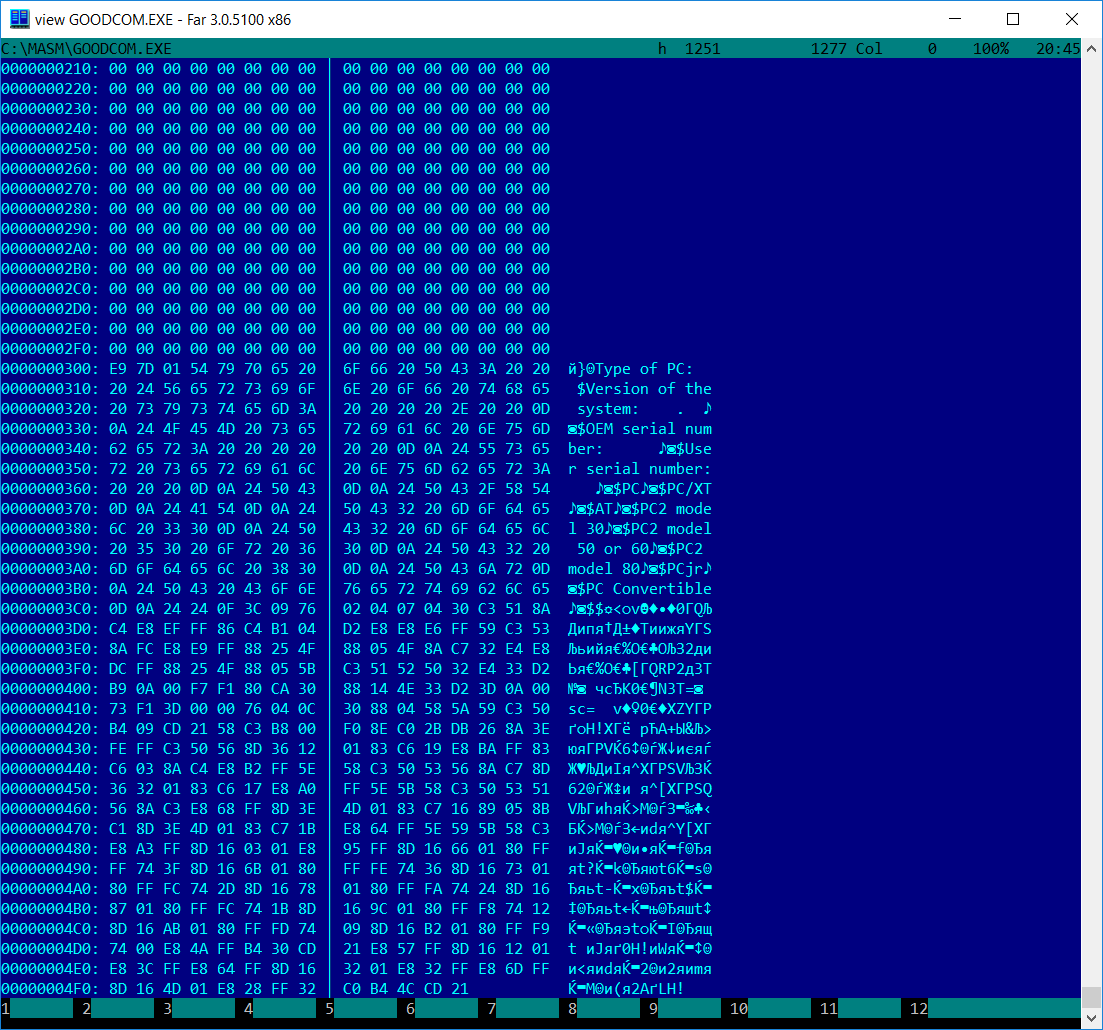


рис.6. «Плохой» .EXE модуль в Far (продолжение)

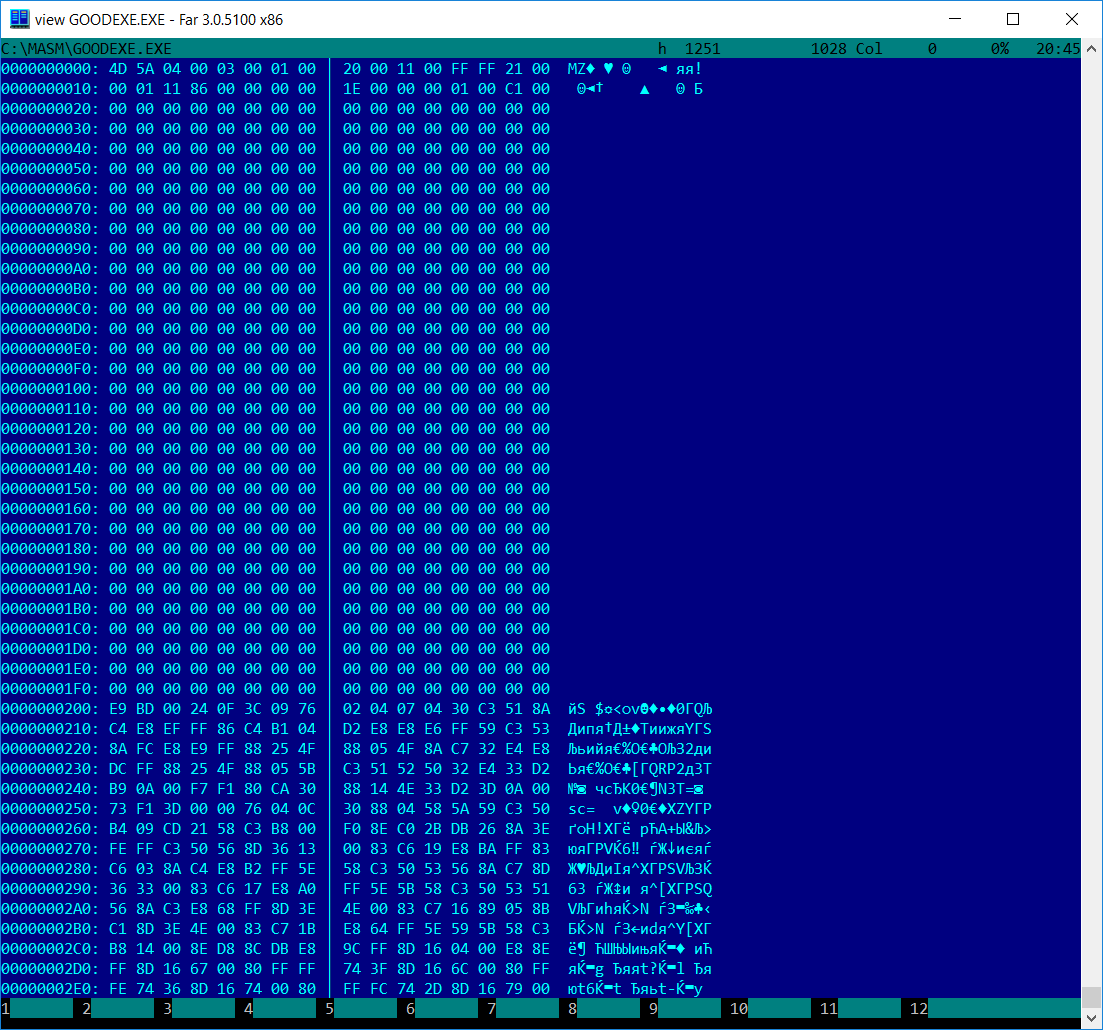


рис.7. «хороший» .EXE модуль в Far

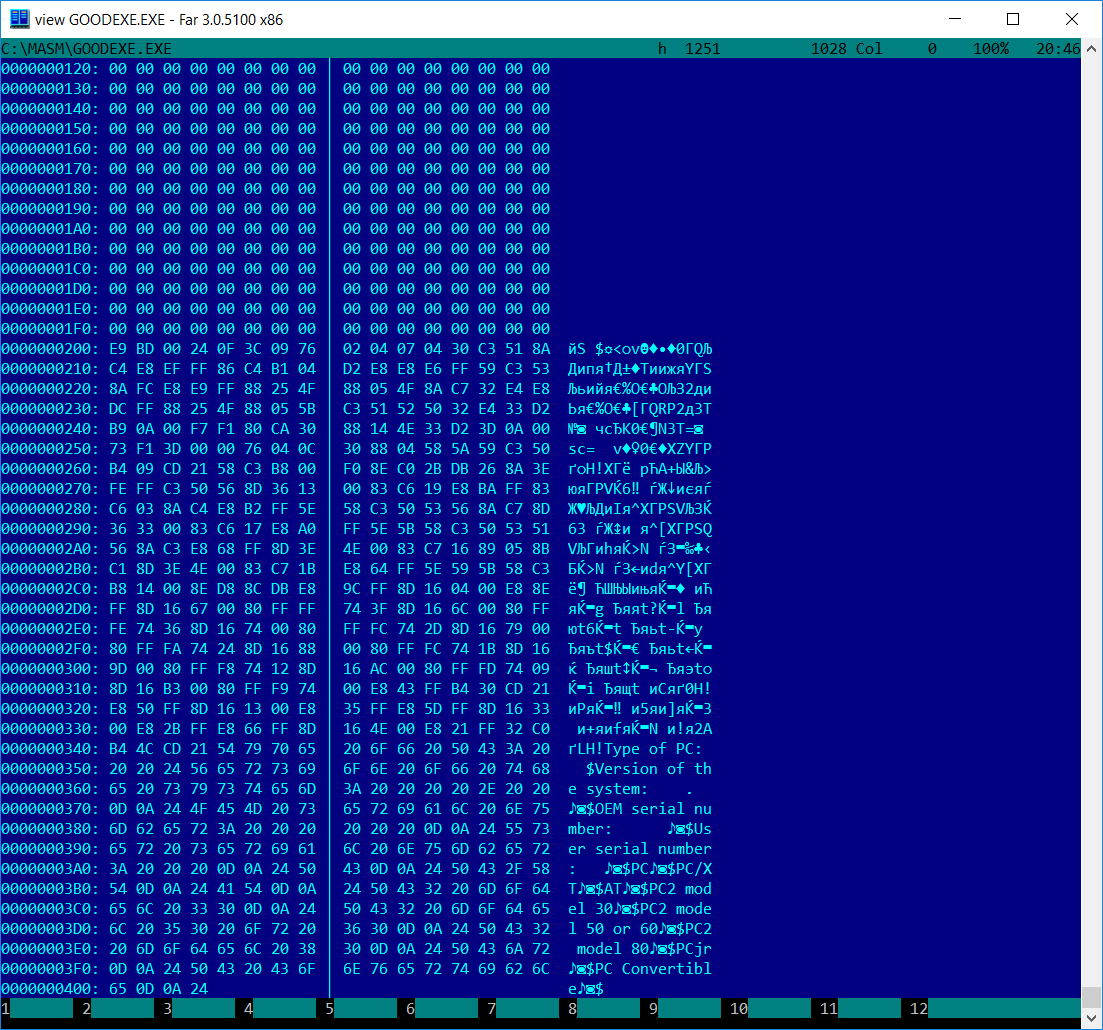


рис.8. «Хороший» .EXE модуль в Far (продолжение)

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

Ответ: файл COM состоит из данных, используемых в программе, и команд. Код располагается с нулевого адреса;

1. Какова структура файла «плохого» EXE? С какого адреса располагается код? Что располагается с 0 адреса?

Ответ: в файле «плохого» EXE в одном сегменте содержатся данные и код. Код располагается с адреса 300h. С 0 адреса располагается таблица разметки;

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

Ответ: в файле «хорошего» EXE содержится сегмент стека, сегмент данных и сегмент кода. Отличие от «плохого» в наличии 3 сегментов. Кроме того, код начинается с 200h, а не с 300h, как в «плохом» ЕХЕ-файле;

**Загрузка COM модуля в основную память.**

Откроем отладчик TD.EXE и загрузим .COM.

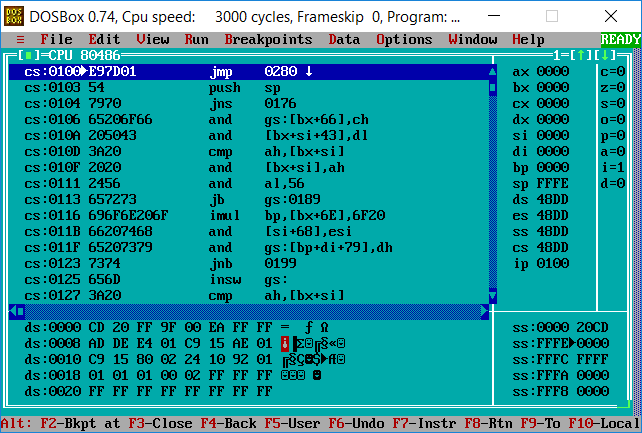


рис.9. .COM модуль в TD

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

Ответ: после загрузки COM-программы в память сегментные регистры указывают на начало PSP. Порядок загрузки: PSP, данные и код, стек. Код располагается с адреса 100h;

1. Что располагается с 0 адреса?

Ответ: с адреса 0 располагается PSP.

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

Ответ: сегментные регистры имеют значение 48DDh. Они указывают на начало PSP.

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

Ответ: стек генерируется автоматически, указатель стека в конце сегмента. Он будет занимать оставшуюся память. Стек растет от больших адресов к меньшим, то есть от FFFEh к 0000h.

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

Откроем отладчик TD.EXE и загрузим «хороший» .ЕХЕ.

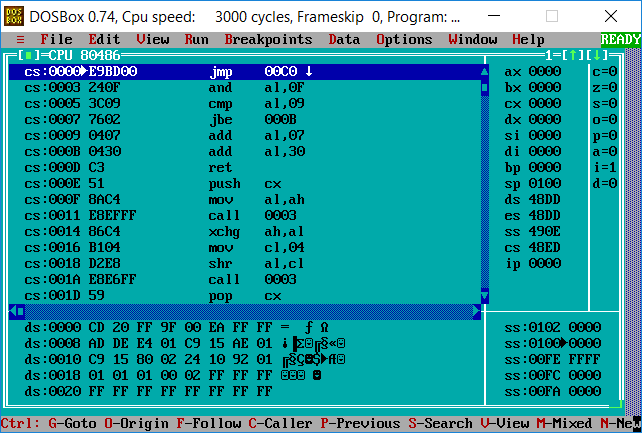


рис.10. «Хороший» .EXE модуль в TD

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

Ответ: вначале строится PSP, стандартная часть заголовка считывается в память, определяется длина тела загрузочного модуля, определяется начальный сегмент, загрузочный модуль считывается в начальный сегмент, таблица разметки считывается в рабочую память, к полю каждого сегмента прибавляется сегментный адрес начального сегмента, определяются значения сегментных регистров. При этом DS = 48DD, ES = 48DD, SS = 490E, CS = 48ED.

1. На что указывают регистры DS и ES?

Ответ: регистры DS и ES указывают на начало сегмента PSP.

1. Как определяется стек?

Ответ: стек определяется благодаря директиве STACK, а регистрам SS и SP придаются значения, указанные в заголовке, после чего к SS прибавляется сегментный адрес начального сегмента.

1. Как определяется точка входа?

Ответ: точка входа определяется благодаря директиве «END Main», где указывается имя процедуры, с которой должна начинаться программа;

**Заключение**

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