**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 |  | Шарипова Р. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

**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 управляющей 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\_PC\_TYPE | Определение типа IBM PC |
| GET\_SYS\_VER | Определение версии системы |
| GET\_OEM\_NUM | Определение OEM |
| GET\_SERIAL\_NUM | Определение серийного номера пользователя |

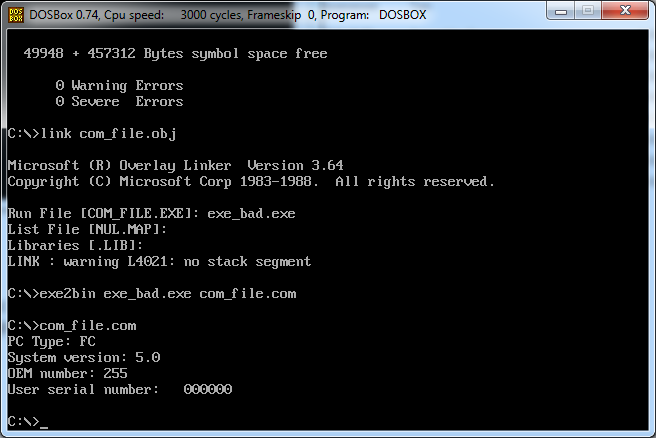
|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| PC\_TYPE | db | 'PC type: ',0dh,0ah,'$' |
| SYS\_VER | db | 'System version: . ',0dh,0ah,'$' |
| OEM\_NUM | db | 'OEM number: ',0dh,0ah, |
| SER\_NUM | db | 'User serial number: ',0dh,0ah,'$' |

**Последовательность действий, выполняемых программой:**

Программа определяет тип IBM PC, версию системы, OEM номер и серийный номер пользователя, сохраняет их в соответствующие переменные, а затем выводит эти значения на экран при помощи функции 09H и вызова прерывания int 21h.

**Результаты:**

Программа выводит на экран:



*Рисунок 1 – «Хороший» .COM*



*Рисунок 2 – «Плохой» .EXE*

**Описание результатов:**

*1)Сравним исходные тексты .COM и .EXE модулей, ответим на контрольные вопросы.*

А) *Сколько сегментов должна содержать COM-программа?*

COM-программа должна содержать один сегмент – сегмент кода.

Б) *EXE-программа?*

EXE-программа может содержать 1 и более сегментов.

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

В тексте COM-программы обязательно должна быть директива ASSUME, которая должна указывать, что сегмент кода и данных начинается с одного и того же места. Также должна быть директива ORG, которая устанавливает счётчик положения в сегменте равным заданной величине, которая передаётся как параметр.

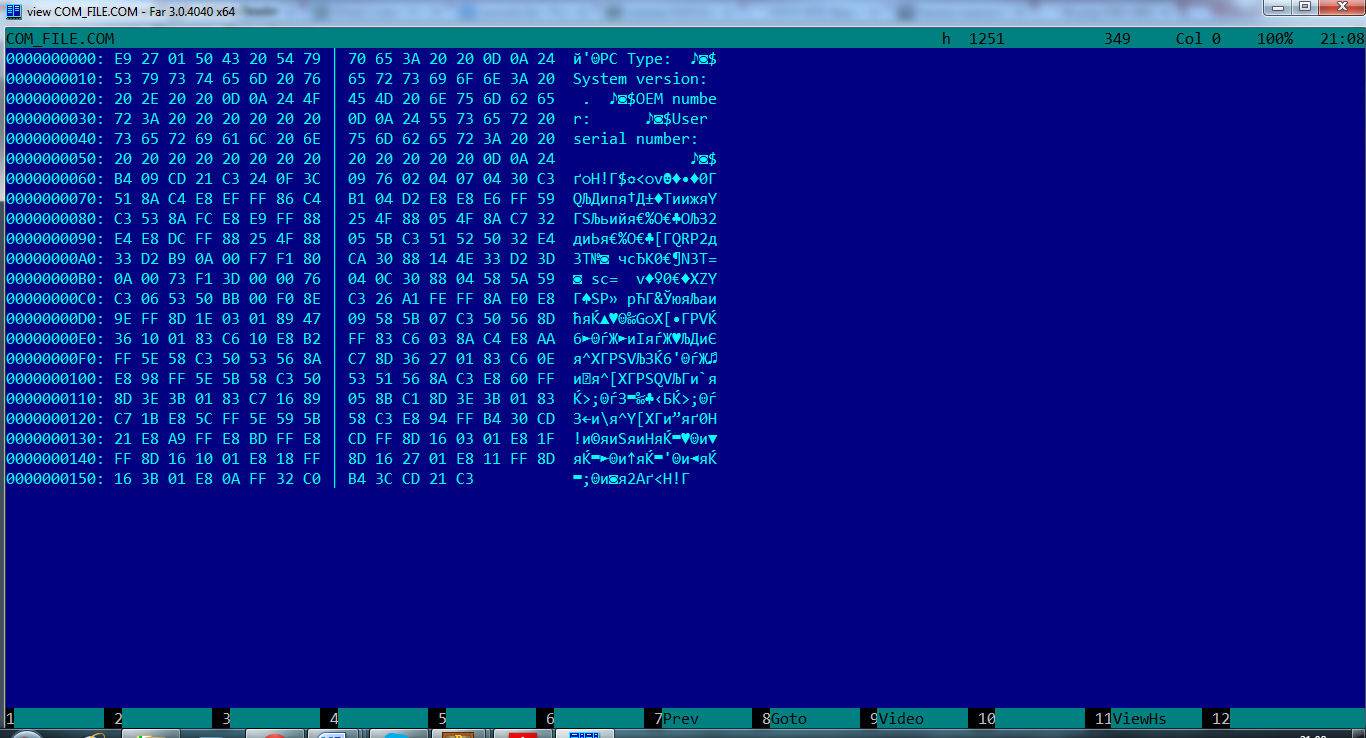
*Г) Все ли форматы команд можно использовать в COM-программе?*

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

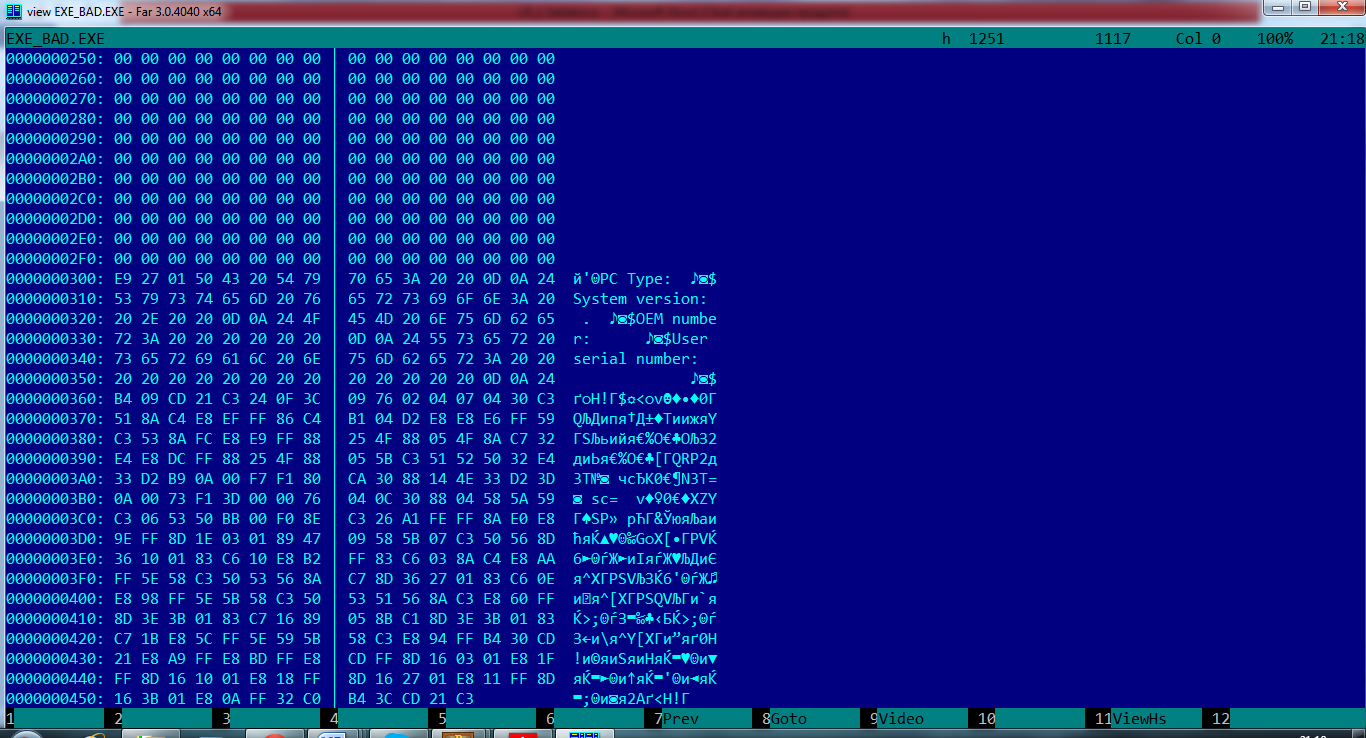
В COM файле нет relocation table, ему неоткуда получить информацию об адресе сегмента.

Мы можем записать адрес единственного сегмента в регистр, конечно, но потом не сможем создать com-файл, потому что адреса сегментов определяются загрузчиком в момент загрузки модуля в память. Но для этого загрузчику нужна информация о местоположении в файле полей адресов, требующих настройки на адреса основной памяти. Эта информация хранится в таблице настройки (relocation table), а в com-модулях она отсутствует, поэтому загрузчику негде взять информацию о сегменте.

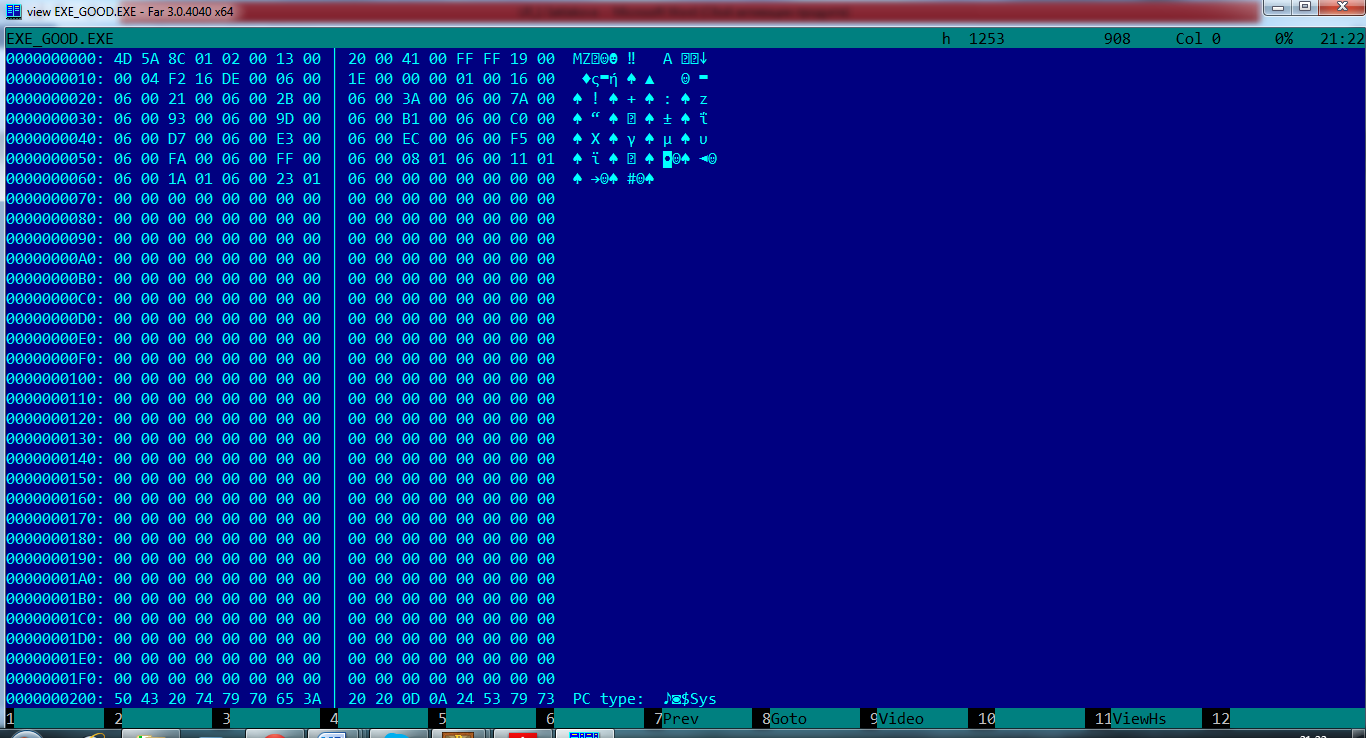
*2)Откроем файл загрузочного модуля .COM, «хорошего» и «плохого» .EXE в FAR-менеджере при помощи нажатия клавиш F3/F4.*



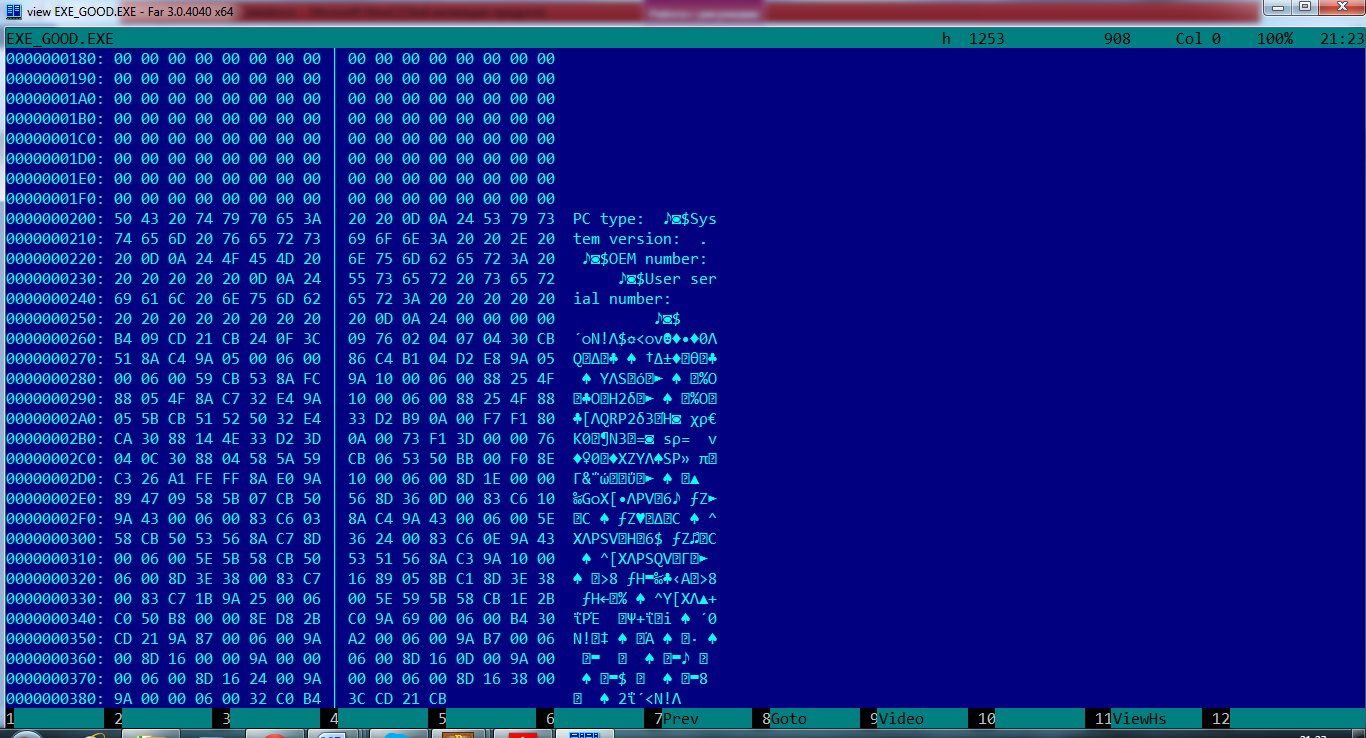
*Рисунок 3 - .COM-файл.*



*Рисунок 4 - «плохой» EXE-файл.*



*Рисунок 5 - «хороший» EXE-файл.*



*Рисунок 6 - «хороший» EXE-файл (продолжение).*

*3) Сравним форматы файлов в 16-ричном виде и ответим на контрольные вопросы:*

*А) Какова структура файла СОМ? С какого адреса располагается код?*

Структура COM-файла очень компактна (сам файл весит гораздо меньше), код располагается с нулевого адреса.

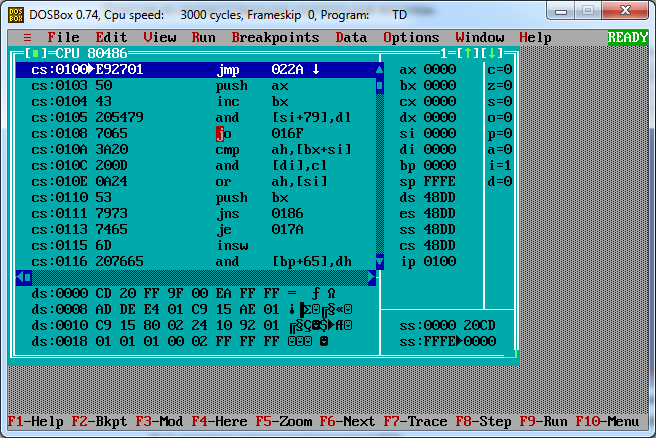
*Б) Какова структура файла «плохого» ЕХЕ? С какого адреса располагается код? Что располагается с адреса 0?*

Структура «плохого» EXE-файла менее компактна, чем у COM-файла. Код располагается с адреса 300h, с нулевого адреса располагается таблица настроек, при помощи которых строится данный EXE-файл.

*В) Какова структура файла «хорошего» ЕХЕ? Чем он отличается от файла «плохого» ЕХЕ?*

Структура «хорошего» EXE-файла несколько компактнее, чем структура «плохого» файла, так как в этом файле отсутствует директива ORG 100h, резервирующая пространство для заголовка. Именно поэтому код располагается с адреса 200h, а не с 300h, как в «плохом» EXE-файле.

*4)Откроем отладчик TD.EXE и поочерёдно прогоним через него COM-файл и «хороший» EXE-файл.*



*Рисунок 7 - .COM-файл в отладчике*

*Ответим на контрольные вопросы «Загрузка COM-модулей в основную память»:*

*А) Какой формат загрузки модуля СОМ? С какого адреса располагается код?*

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

*Б) Что располагается с адреса 0?*

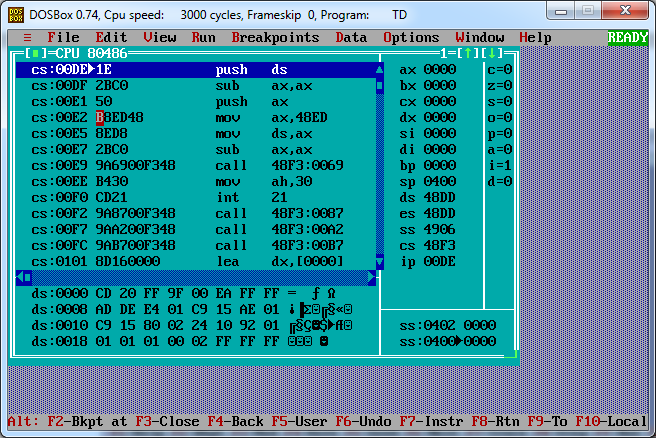
С нулевого адреса располагается PSP.

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

Все сегментные регистры имеют значение «48DD» и указывают на начало PSP.

*Г) 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), оставшееся после загрузки данных и кода. В конкретном случае значение регистра SP=FFFE.

*Рисунок 8 - «хороший» .EXE файл в отладчике*

*Ответы на контрольные вопросы «Загрузка «хорошего» EXE-модуля в основную память»:*

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

Порядок загрузки EXE-модуля: PSP, сегмент кода, сегмент данных, сегмент стека. Сегментные регистры на момент загрузки программы имеют значения: ES=48DD, CS=48F3, DS=48DD, SS=4906. На начальном этапе ES=DS, так как не были выполнены команды “mov ax, data; mov ds, ax”, т.е. в регистр данных не был помещён адрес сегмента данных.

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

ES указывает на начало PSP, DS указывает на начало данных. После выполнения команд (см. пред. вопрос), значение DS= 48ED.

*В) Как определяется стек?*

Стек определяется директивой «DW 512 DUP(?)» в описании сегмента стека.

*Г) Как определяется точка входа?*

Точка входа определяется благодаря директиве «END Main», где указывается имя процедуры, с которой должна начинаться программа.

**Вывод:**

В ходе данной лабораторной работы было проведено сравнение структуры COM и EXE файлов, исследованы различия в исходных текстах модулей COM и EXE, а также были сравнены способы их загрузки в память.