**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Операционные системы»**

Тема: Исследование структур загрузочных модулей**.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6381 |  | Токун М. С. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

# **Цель работы.**

Исследование различий в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов из 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;

# **Ход работы.**

## **Описание данных**

**IbmPC** – содержит тип IBM PC

**SystVer**– содержит версию системы

**SerOEMNum** – содержит серийный номер OEM

**UserSer**– содержит серийный номер пользователя

## **Описание функций**

**TETR\_TO\_HEX**, **BYTE\_TO\_HEX**, **WRD\_TO\_HEX** – перевод в 16ую СС

**BYTE\_TO\_DEC** – перевод в 10ую СС

**IDENT\_IBM\_PC** – определение типа IBM PC

**IDENT\_SYS\_VER** – определение версии системы

**IDENT\_OEM\_SER** – определение серийного номера OEM

**IDENT\_USER\_SER** – определение серийного номера пользователя

**Write** – вывод сообщения

## **Описание алгоритма**

Программа определяет тип IBM PC, версию системы, серийный номер OEM и серийный номер пользователя, а затем выводит их.

## **Результат выполнения**

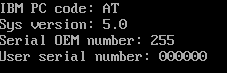


Рисунок 1 – Результат выполнения программы

# **Вывод.**

В ходе данной лабораторной работы было выполнено исследование различий в структурах исходных текстов модулей .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

## **Ответы на контрольные вопросы.**

1. Отличия исходных текстов COM и EXE программ:
   1. COM программа должна содержать только один сегмент – сегмент кода.
   2. EXE программа содержит три сегмента – кода, данных, стека.
   3. COM программа должна обязательно содержать директивы:

ASSUME (указывает, что сегменты кода и данных начинаются в одном месте) и ORG (задаёт смещение в 256 байт от нулевого адреса, т.е. служит для резервирования памяти под префикс программного сегмента).

* 1. В COМ программе нельзя использовать команды формата mov <регистр>, SEG <имя сегмента>, т.к., в отличие от EXE-файлов, там отсутствует таблица адресации сегментов.

1. Отличия форматов файлов COM и EXE модулей
   * 1. В COM файле все данные определяются в сегменте кода. Код начинается с адреса 0h, но адресация в сегменте кодов начинается со смещения 100h от начала PSP (рис. 2).
     2. В «плохом» EXE файле данные и код содержаться в одном сегменте. Код располагается с адреса 300h, а с 0h идёт Relocation table, по которой строится EXE файл (рис. 3).
     3. В «хорошем» EXE файле данные, код и стек разделены по сегментам. В отличии от «плохого» EXE файла, отсутствует директива ORG, поэтому код располагается с адреса 200h (рис. 4).

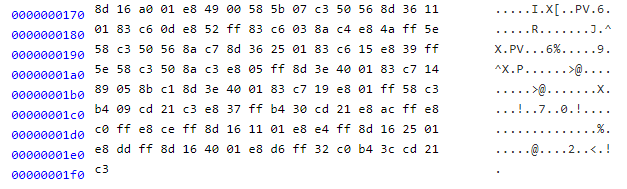
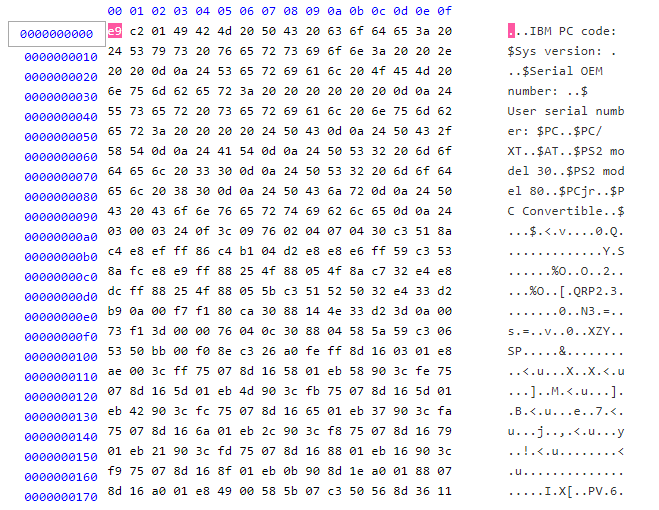


Рисунок 2 – COM файл

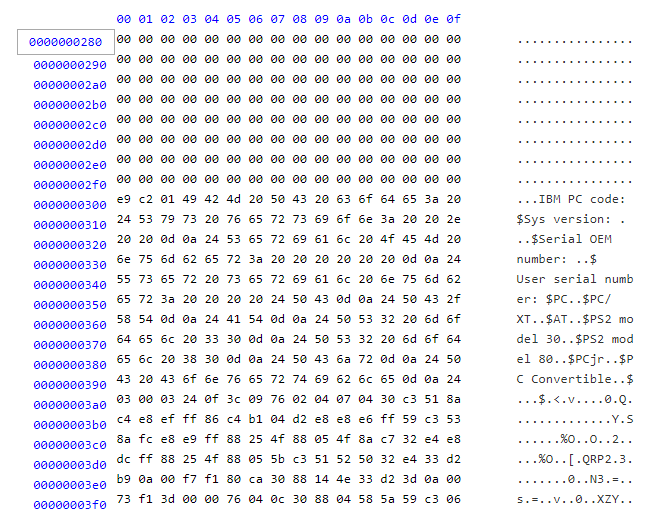
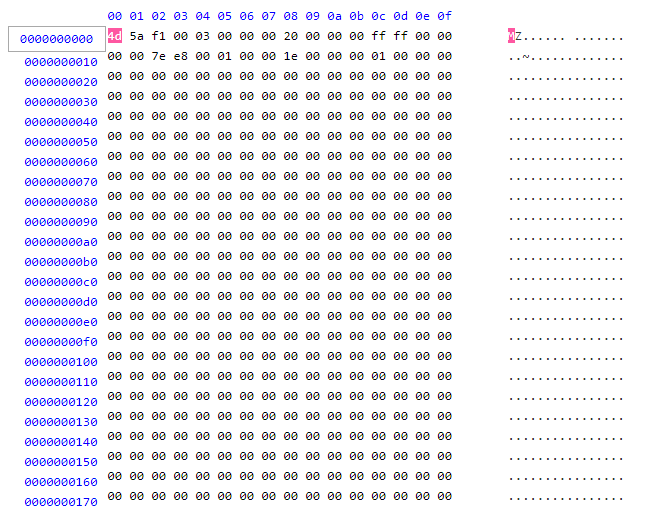
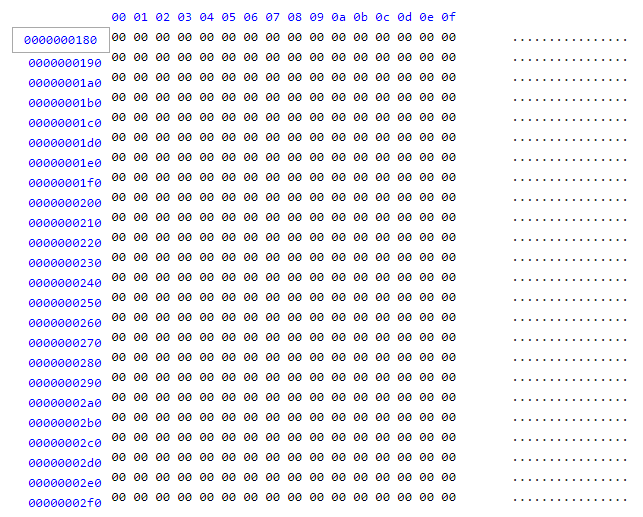
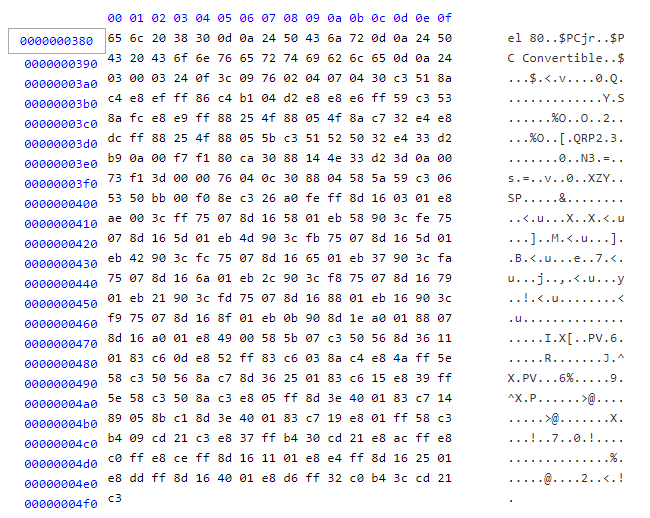
  

Рисунок 3 – «плохой» EXE файл

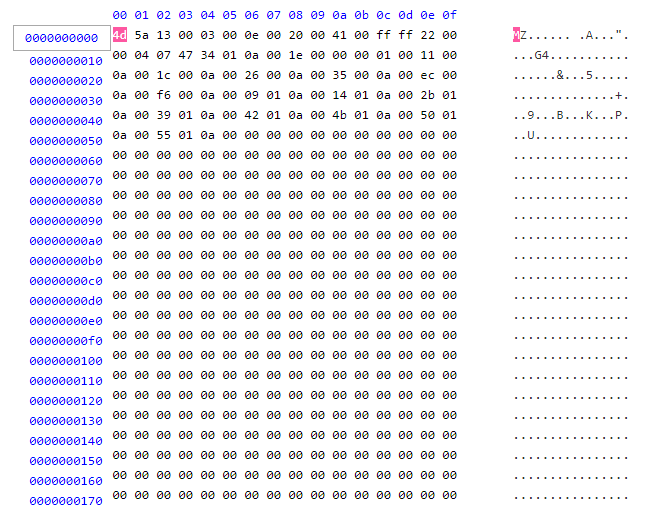
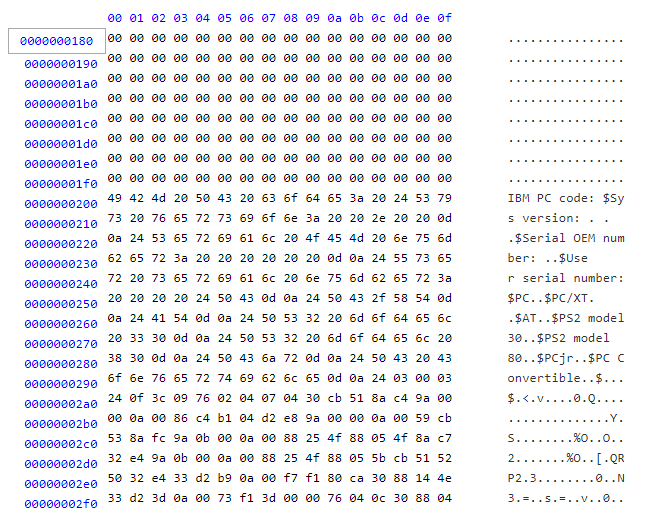
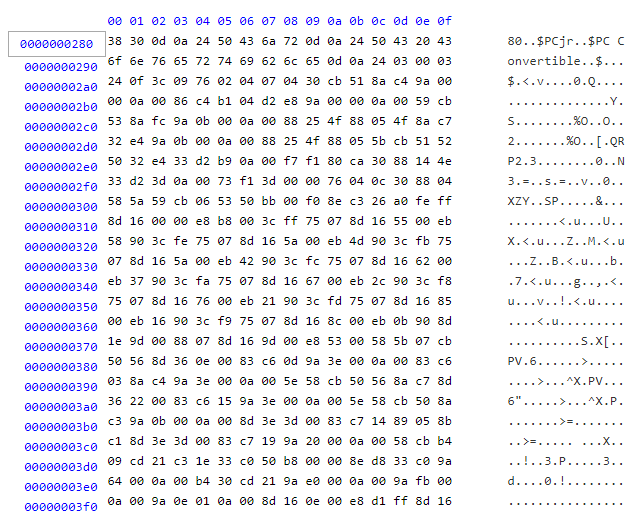
 C:\Users\mihai\YandexDisk\Скриншоты\2018-04-15_19-24-24.png

Рисунок 4 – «хороший» EXE файл

1. Загрузка COM модуля в основную память

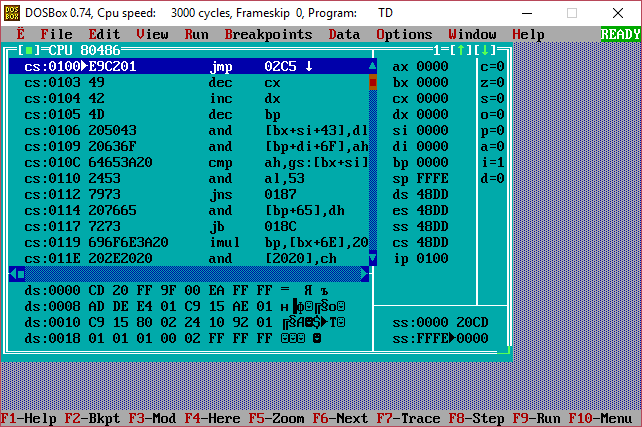


Рисунок 5 – COM файл в отладчике TD.EXE

1. Формат загрузки модуля COM - сначала PSP, затем код и данные, потом стек. Код располагается по адресу – 100h.
2. С адреса 0 располагается сегмент PSP.
3. Сегментные регистры указывают на начало PSP, и равны в этом случае 48DD.
4. Стек занимает оставшееся после загрузки кода и данных место в сегменте памяти. SP=FFFEh – вершина стека. Диапазон адресов, которые он может занимать 0h-FFFEh.
5. Загрузка «хорошего» EXE модуля в основную память

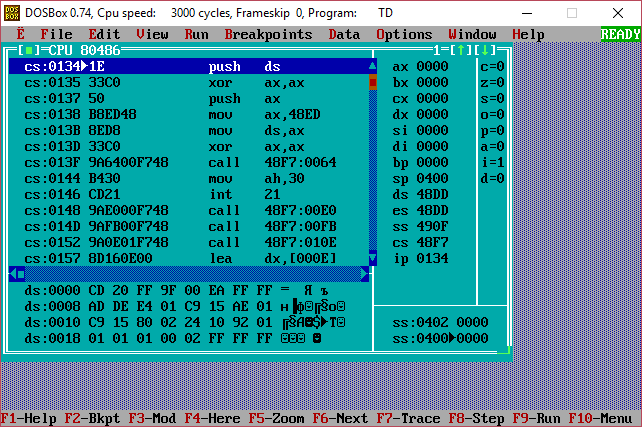


Рисунок 6 – «хороший» EXE файл в отладчике TD.EXE

1. При запуске «хорошего» EXE, загружается сначала PSP, затем данные, потом код, в конце стек. Значения сегментных регистров: DS=ES=48DD, CS=48F7, SS=490F.
2. До загрузки в DS адреса сегмента данных оба на начало PSP.
3. Стек определяется с помощью директивы «DW 512 DUP(?)» в описании сегмента стека.
4. Точка входа определяется с помощью директивы «END», которая указывает метку, с которой должна начинаться программа.