МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
“ЛЭТИ” ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

Отчет

По лабораторной работе №1

По дисциплине “Операционные системы”

Тема: Исследование структур загрузочных модулей

Студент гр. 8382 _____ Никитин А.Е.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Исследование различий структур исходных модулей типов .com и .exe, структур файлов загрузочных модулей и способов загрузки в основную память.

Необходимые сведения.

Тип IBM PC хранится в байте по адресу 0F000:0FFFEh, 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.

Постановка задачи.

Требуется реализовать 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 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.

Контрольные вопросы.

1) Сколько сегментов должна содержать COM-программа?
    - В отличии от EXE, COM программа содержит только 1 сегмент.
2) Сколько сегментов должна содержать EXE-программа?
    - Содержание сегментов в EXE программе может варьироваться в зависимости от модели памяти.
3) Какие директивы должны обязательно быть в тексте COM-программы?
    - ASSUME, которая устанавливает соответствие сегментых регистров и сегмента
    - ORG 100h, устанавливающая смещение в 100h для PSP в начале программы
4) Все ли форматы команд можно использовать в COM-программе?
    - Нельзя использовать команды, использующие адрес сегмента, так как он определяется после запуска программы. Также нельзя создать больше одного сегмента

5) Какова структура файла COM? С какого адреса располагается код?
    - С адреса 0h
6) Какова структура файла «плохого» EXE? С какого адреса располагается код? Что располагается с адреса 0?
    - С адреса 0h располагается заголовок и таблица настройки адресов
    - Плохой EXE также содержит 1 сегмент, но код начинается с адреса 300h
7) Какова структура файла «хорошего» EXE? Чем он отличается от файла «плохого» EXE?
    - Структура файла аналогична, за исключением наличия трёх сегментов вместо одного. Теперь сегмент кода расположен по адресу 400h, так как перед ним появился сегмент стека размером 100h
8) Какой формат загрузки модуля COM? С какого адреса располагается код?
    - Код с 100h
    - От 0 до 100h PSP
9) Что располагается с адреса 0?
    - См. выше
10) Какие значения имеют сегментные регистры? На какие области в памяти они указывают?
    - Сегментные регистры имеют значение 449D, указывающее на начало PSP
11) Как загружается «хороший» EXE? Какие значения имеют сегментные регистры?
    - DS, ES (449D) – начало PSP;
    - SS (44AD) – начало сегмента стека;
    - CS (44DA) – начало сегмента кода.
12) На что указывают регистры DS и ES?
    - См. выше
13) Как определяется стек?
    - Директивой STACK или с помощью ASSUME SS: <ссылка на сегмент, который будет определен как сегмент стека>
14) Как определяется точка входа?
    - Директивой END <ссылка на точку входа>

Выполнение работы.

Для определения типа PC и версии были написаны тексты .com и .exe модулей. Для определения версии MS DOS была использована функция 30h прерывания 21h. После ее вызова версия системы определяется значением регистров al, ah. В регистре bh - серийный номер OEM, в bl:cx – 24-битовый серийный номер пользователя.

Результат выполнения .com модуля виден на рис. 1. Результат выполнения “плохого” .exe модуля, полученного из исходного текста для .com модуля, виден на рис. 2. Результат выполнения “хорошего” .exe модуля - на рис. 3.

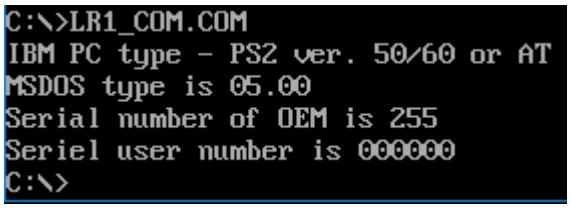

Рисунок 1 - Результат работы LR1_COM.COM

Рисунок 2 - Результат LR1_COM.EXE

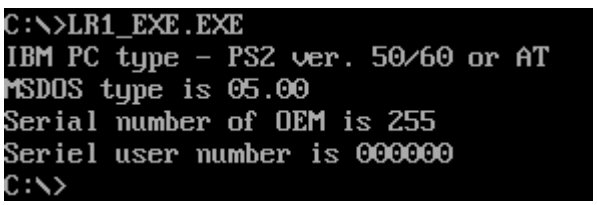

Рисунок 3 - Результат работы LR1_EXE.EXE

Представление исходных файлов в шестнадцатеричном виде представление исходных файлов в шестнадцатеричном виде:

Рисунок 4 – LR1_COM.COM в шестнадцатеричном виде.

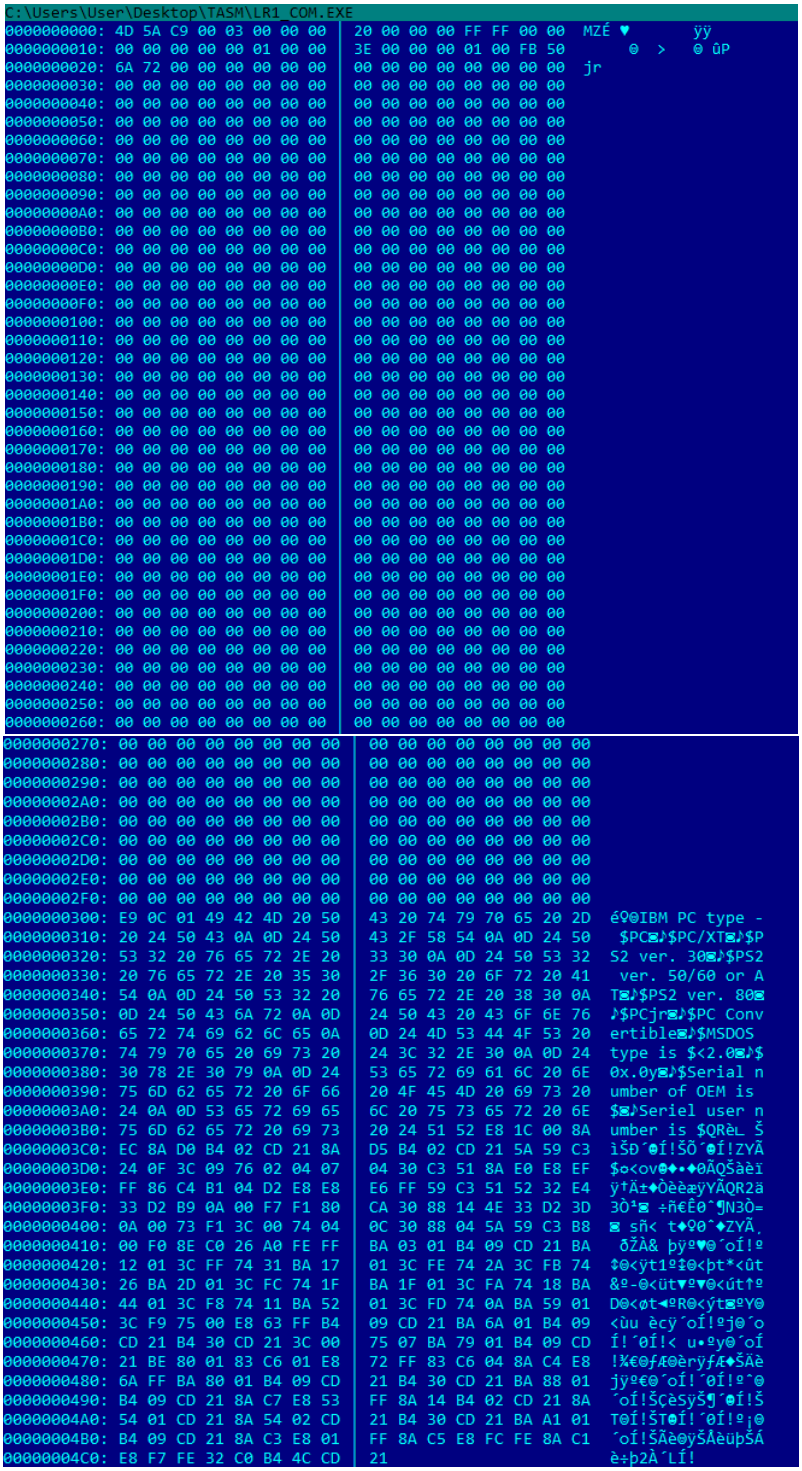

Рисунок 5 – LR1_COM.EXE в шестнадцатеричном виде.

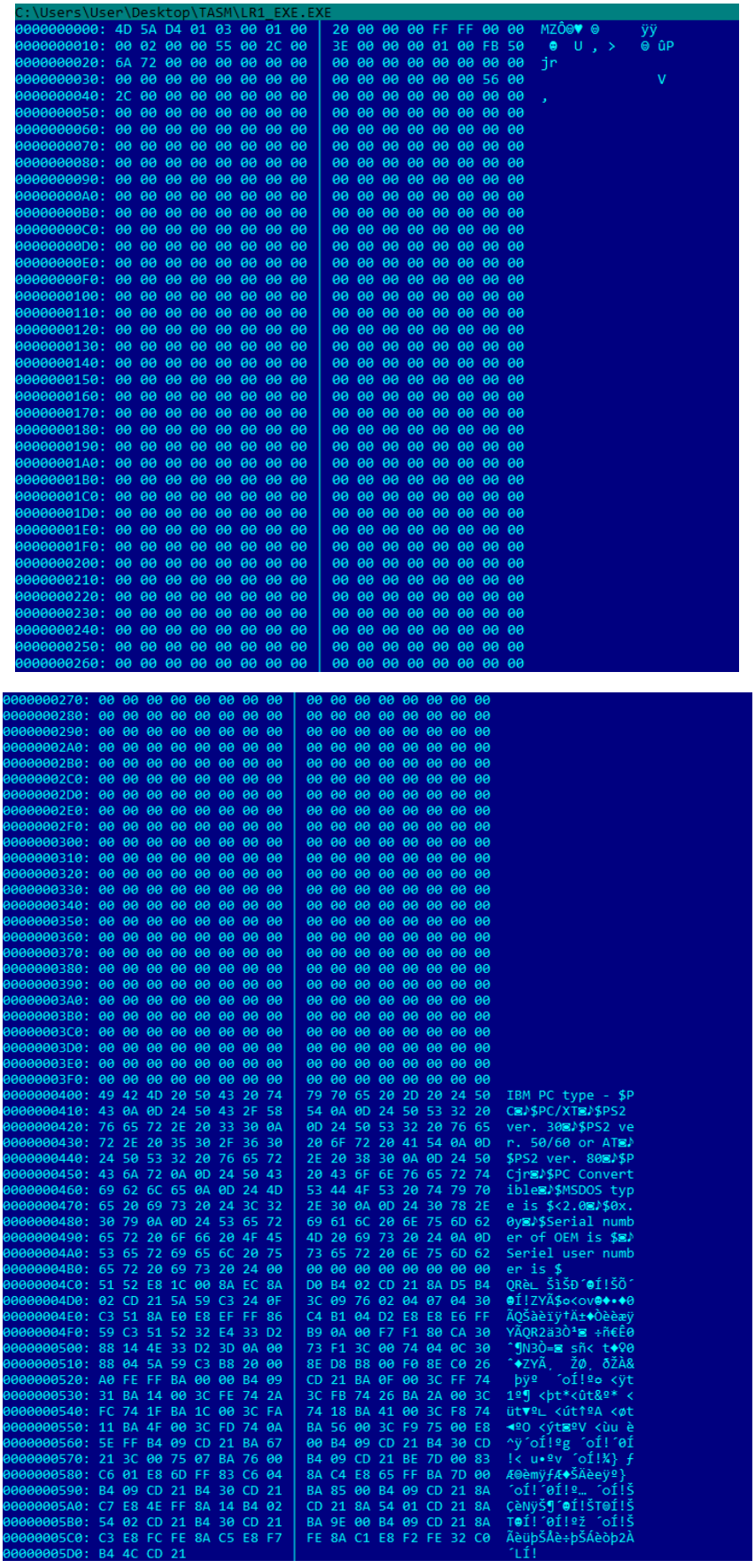

Рисунок 6 – LR1_EXE.EXE в шестнадцатеричном виде.