**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.

|  |  |  |
| --- | --- | --- |
| Студент гр. 0381 |  | Дзаппала Д. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2022

## Цель работы.

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

## Задание.

Написать текст исходного .СОМ модуля, который определяет тип РС и версию системы. Ассемблерная программа должна читать содержимое предпоследнего байта ROM BIOS, по таблице, сравнивая коды, определять тип РС и выводить строку с названием модели. 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.

Написать текст исходного .ЕХЕ модуля, который выполняет те же функции, что и модуль .COM и постройте и отладьте его. Таким образом, будет получен «хороший» .ЕХЕ.

Сравнить исходные тексты для .СОМ и .ЕХЕ модулей. Ответить на контрольные вопросы «Отличия исходных текстов СОМ и ЕХЕ программ».

## Основные теоретические положения.

Шестнадцатеричные коды типов IBM PC: PC – **FF**, PC/XT – **FE** (**FB**), AT – **FC**, PS2 (model 30) – **FA**, PS2 (model 50 or 60) – **FC**, PS2 (model 80) – **F8**, PCjr – **FD**, PC Convertible – **F9**.

Для определения версии MS-DOS используется функция 30h прерывания 21h. Входным параметром является номер функции в AH. Выходными параметрами являются: AL – номер основной версии, AH – номер модификации, BH – серийный номер OEM, BL:CX – 24-битовый серийный номер пользователь.

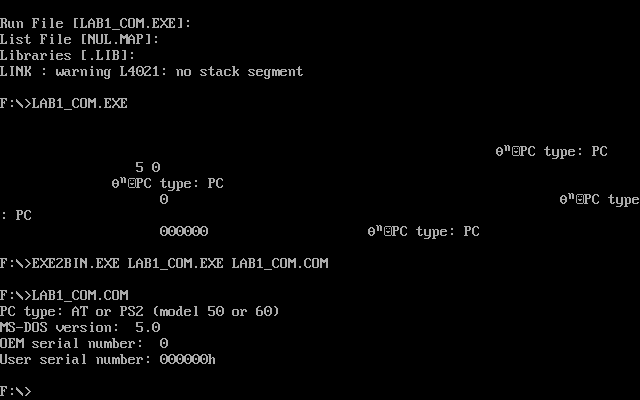
## Выполнение работы.

Для выполнения задания были определены две процедуры: для определения типа IBM PC – DefPCType, для определения характеристик MS-DOS – DefSysVer. Были объявлены строки для вывода нужной информации:

* pc db 'PC type: PC', 0dh, 0ah, '$'
* pc\_xt db 'PC type: PC/XT', 0dh, 0ah, '$'
* pc\_at db 'PC type: AT or PS2 (model 50 or 60)', 0dh, 0ah, '$'
* pc\_ps2\_30 db 'PC type: PS2 model 30', 0dh, 0ah, '$'
* pc\_ps2\_80 db 'PC type: PS2 model 80', 0dh, 0ah, '$'
* pc\_pcjr db 'PC type: PCjr', 0dh, 0ah, '$'
* pc\_conv db 'PC type: PC Convertible', 0dh, 0ah, '$'
* ms\_dos\_ver db 'MS-DOS version: . ', 0dh, 0ah, '$'
* oem\_sn db 'OEM serial number: ', 0dh, 0ah, '$'
* user\_sn db 'User serial number: h', 0dh, 0ah, '$'

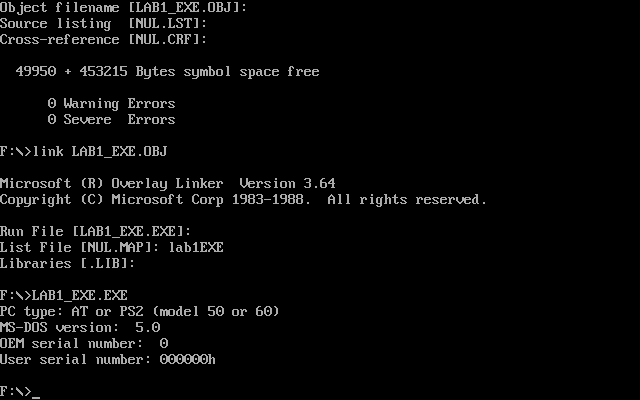
Тип PC AT и PS2 (модели 50 или 60) были объединены так как у них одинаковый код.

При запуске .COM модуля в виде «плохого» .EXE модуля получаем следующую информацию:



Использовав утилиту EXE2BIN, получаем правильный .COM модуль, который выводит корректную информацию.

Для того, чтобы написать «хороший» .EXE модуль, нужно будет отделить данные от сегмента кода, добавив сегмент данных и сегмент стека. После сборки, получаем следующую информацию:



## Выводы.

Были исследованы различия в структурах исходных текстов модулей .COM и .EXE, структур файлов загрузочных модулей и способов загрузки в основную память.

# Приложение А Ответы на контрольные вопросы по Лаб.Работе №1

**Отличия исходных текстов COM и EXE программ**

1. Сколько сегментов должна содержать COM программа?

COM модуль должен содержать только один сегмент — сегмент кода (данные также находятся в этом сегменте), а стек формируется автоматически.

1. EXE программа?

Как минимум сегмент кода, в лучшем случае, определить отдельно сегмент кода, данных и стека.

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

ORG 100h — для смещения 256 байт, в которых размещается PSP (префикс программного сегмента). Assume – для указания, что сегмент кода является общим для кода и сегмента данных.

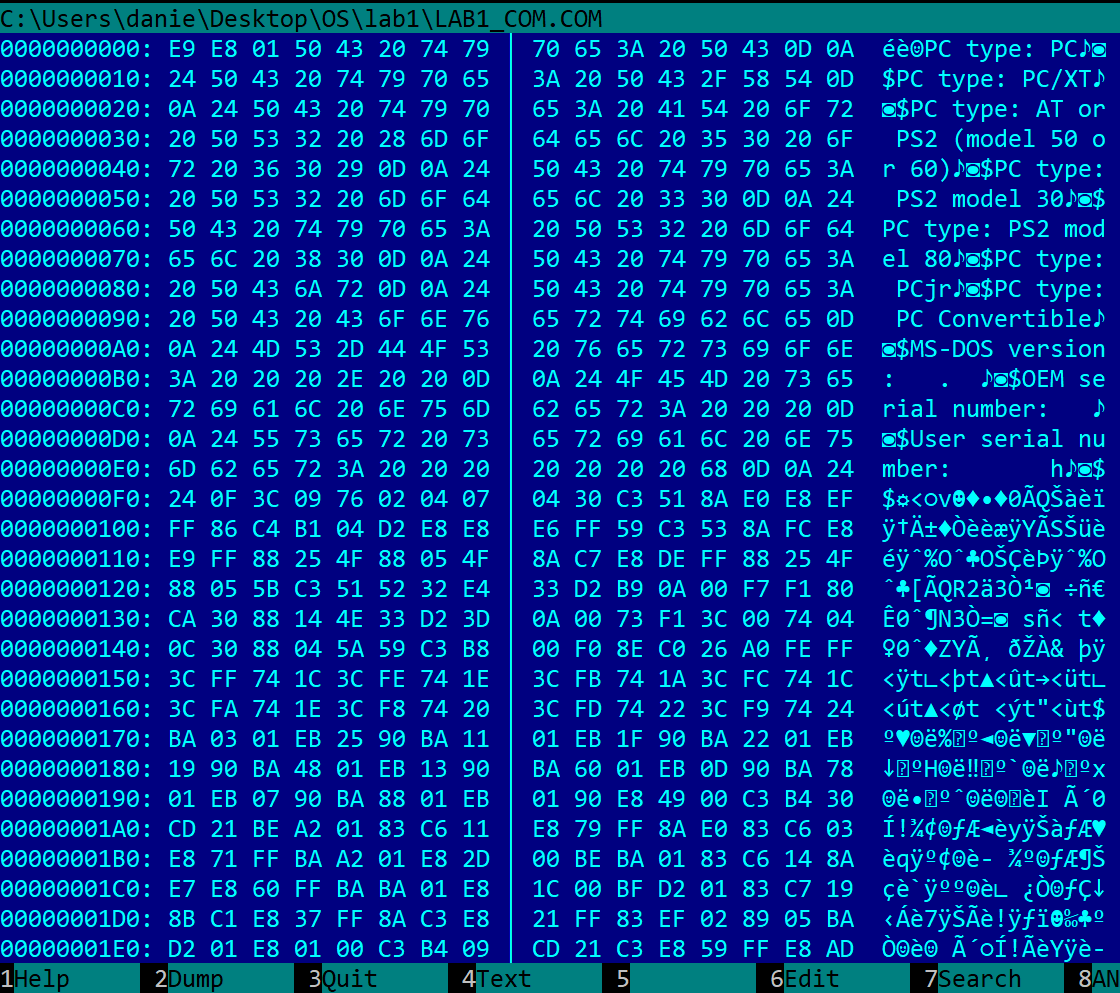
1. Все ли форматы команд можно использовать в COM программе?

Нет, так как в COM модуле нету таблицы настройки адресов (таблица, созданная транслятором, с указателями на сегменты), нельзя использовать команды, аргументы которых будут сегменты.

**Отличия форматов файлов COM и EXE модулей**

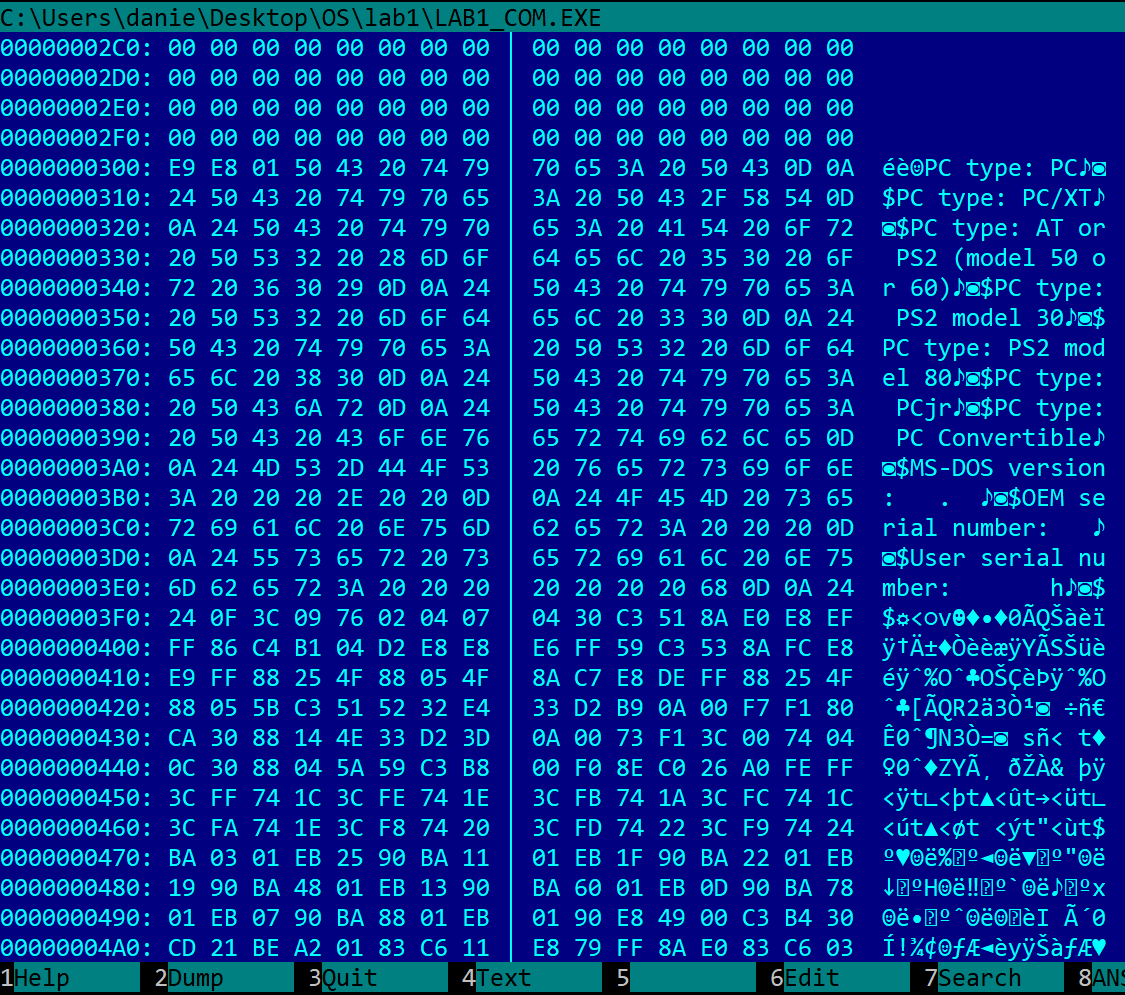
1. Какова структура файла COM? С какого адреса располагается код?

COM модуль состоит из одного сегмента — кода, совмещенного с данными. Размер исполняемого COM модуля не может превышать 64 Кб. Код начинается с адреса 0, но при загрузке программы устанавливается смещение 100h.

****

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

Код и данные находятся в одном сегменте, что неправильно. Код располагается с адреса 300h, с адреса 0h располагается заголовок .EXE модуля.

****

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

Такой файл не имеет лимита в размере; содержит три отдельных друг от друга сегмента: сегмент кода, сегмент данных и сегмент стека. Структура: заголовок (с информацией), сегмент стека (200h-280h, 128 байт), сегмент кода (280h – 386h) и сегмент данных (с 390h до конца файла).

****

**Загрузка COM модуля в основную память**

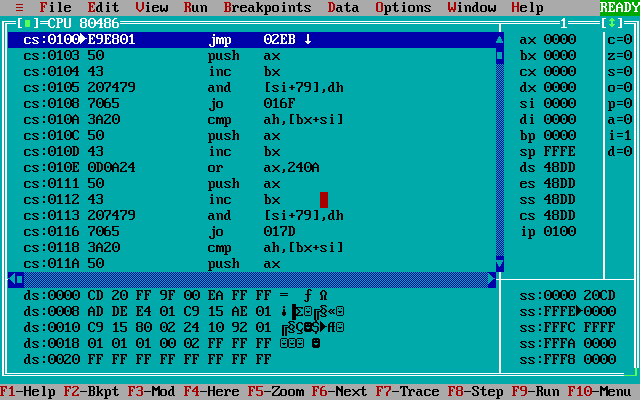
1. Какой формат загрузки COM модуля? С какого адреса располагается код?

Выделяется свободный кусок ОП и его адрес заносится в сегментные регистры. Далее содержится PSP, которое занимает 256 байт. После загружается содержимое COM модуля. Указатель на стек (SP) указывает на конец сегмента PSP, в стек записывается адрес возврата (PSP = 0000h), IP = 100h.

2. Что располагается с адреса 0? - PSP.

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

Все сегментные регистры имеют значение равное 48DD (в данном случае), это и есть адрес сегмента в который была загружена программа.

****

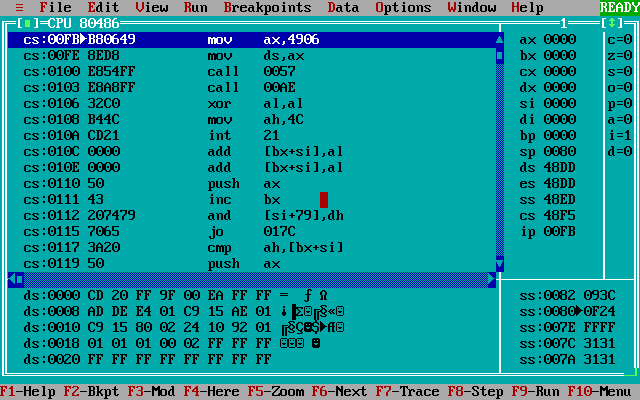
4. Как определяется стек? Какую область памяти он занимает? Какие адреса?

Стек в COM модуле формируется автоматически и находится в том же сегменте, что и остальные. SS указывает на начало PSP, SP указывает на конец стека (FFFEh).

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

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

EXE модуль загружается с адреса PSP:0100h. В процессе загрузки считывается информация заголовка (PSP) в начале файла и выполняется перемещение адресов сегментов (т.е. DS и ES указывают на начало сегмента PSP = 48DD), SS указывает на начало сегмента стека, CS на начало сегмента кода. В IP загружается смещение точки входа в программу, которая берется из метки после директивы END.

****

2. На что указывают регистры DS и ES?

Регистры DS и ES указывают на начало сегмента PSP.

3. Как определяется стек?

Стек можно определить директивой segment, написав:

segment\_name segment stack

<\*db/dw/dd/dq/dt> <count\_of\_\*> (заполнитель dup (чем\_заполнить))

segment\_name ends

Также, можно воспользоваться директивой .stack <size>

4. Как определяется точка входа?

Точка входа определяется директивой END.