**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. 7381 |  | Судакова П.С. |
| Преподаватель |  | Ефремов М. А. |

Санкт-Петербург

2019

**Цель работы.**

Исследовать различия в структурах исходных текстов модулей типов .COM и .EXE, структур файлов загрузочных модулей и способов их загрузки в основную память.

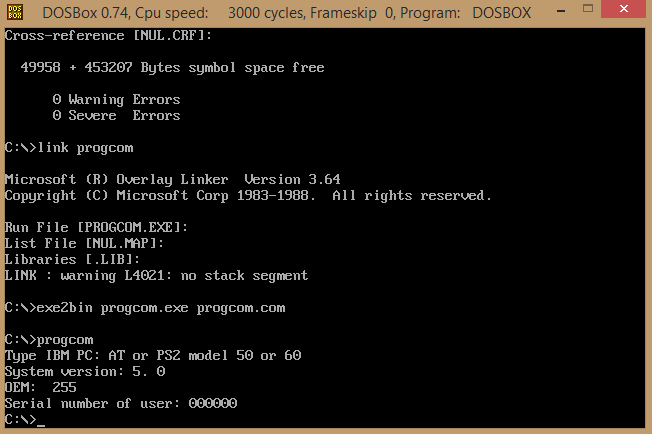
**Сведения о функциях и структурах данных, используемых в программе**

|  |  |  |
| --- | --- | --- |
| *Название поля* | *Тип* | *Назначение* |
| PrintScreen | Процедура | Выполняет вывод текстового сообщения на экран |
| TETR\_TO\_HEX | Процедура | Перевод из четверичной системы счисления в шестнадцатиричную |
| BYTE\_TO\_HEX | Процедура | Перевод байта в два символа 16-ти ричного числа |
| WRD\_TO\_HEX | Процедура | Перевод слова в 4 символа 16-ти ричного числа |
| BYTE\_TO\_DEC | Процедура | Перевод байта в 10-ти ричную систему счисления |
| TYPEPC | db | Вывод строки - Type IBM PC: |
| TYPE\_1 | db | Вывод строки - PC |
| TYPE\_2 | db | Вывод строки - PC/XT |
| TYPE\_3 | db | Вывод строки - AT or PS2 model 50 or 60 |
| TYPE\_4 | db | Вывод строки - PS2 model 30 |
| TYPE\_5 | db | Вывод строки - PS2 model 80 |
| TYPE\_6 | db | Вывод строки - PCjr |
| TYPE\_7 | db | Вывод строки - PC Convertible |
| ERR | db | Вывод строки - Noname ( ) |
| VERS | db | Вывод строки - 13,10, System version: |
| MODIF | db | Вывод строки - 13,10, Modification: |
| OEM | db | Вывод строки - 13,10, OEM: |
| SERNUM | db | Вывод строки - 13,10, Serial number of user: |

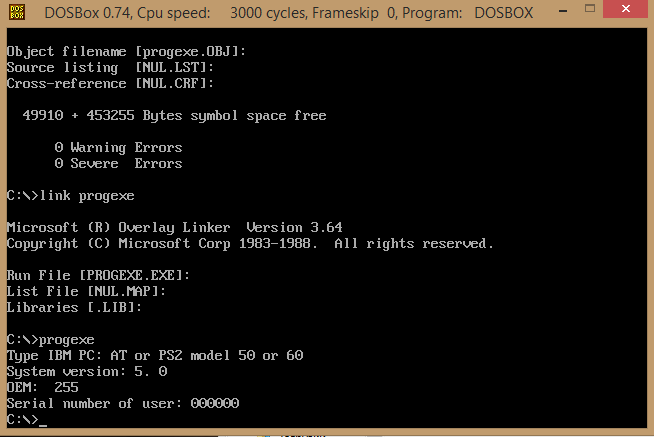
**Работа программы.**

1. Определение типа IBM PC
2. Определение номера основной версии и модификации
3. Определение ОЕМ
4. Определение серийного 24-ёх битного номера пользователя

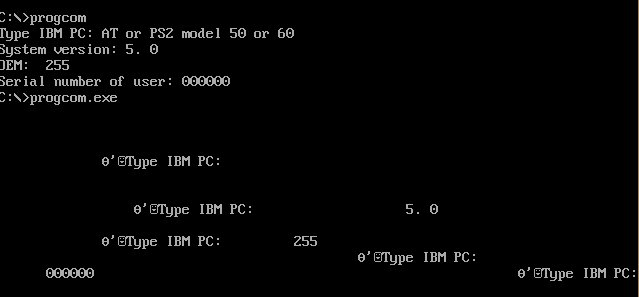
Выполнение «хорошего» .com файла



Выполнение «хорошего» .exe файла



Выполнение «плохого» .exe файла



**Ответы на контрольные вопросы**

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

1. Сколько сегментов должна содержать COM-программа?

**Ответ:** только один.

1. Сколько сегментов должна содержать EXE-программа?

**Ответ:** не менее одного сегмента.

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

**Ответ:** ORG 100h обязательно присутствует в COM файле, так как при загрузке COM-файла в память DOS занимает 256 байт блоком PSP, а код программы располагает сразу после этого блока, в EXE-файле блок данных PSP размещается вне сегмента кода, поэтому смещение программного счетчика указывать не нужно.

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

**Ответ:** не все форматы команд можно использовать.

Нельзя использовать формата mov <регистр>, SEG <имя сегмента>, т.к., в отличие от EXE-файлов, там отсутствует таблица адресации сегментов.

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

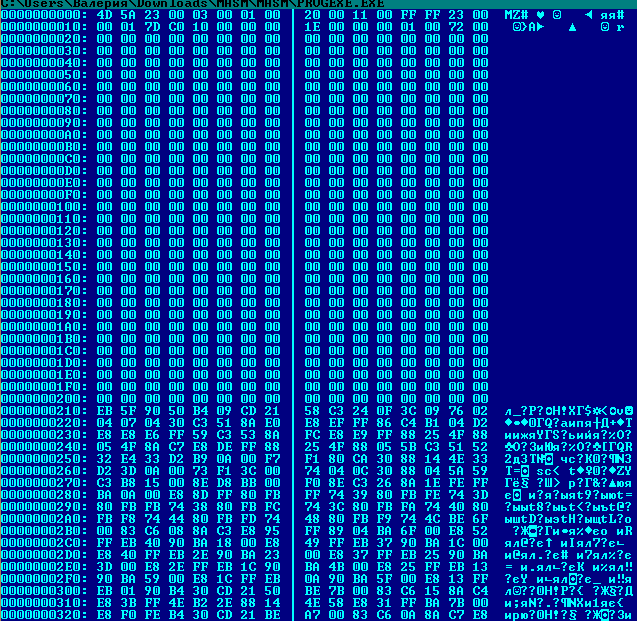
.COM файл



«плохой» .ЕХЕ файл



«хороший» .ЕХЕ файл



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

**Ответ:** COM-файл состоит из одного сегмента и содержит данные и машинные команды в одном сегменте, но код начинается с адреса 0h, а при загрузке модуля устанавливается смещение в 100h.

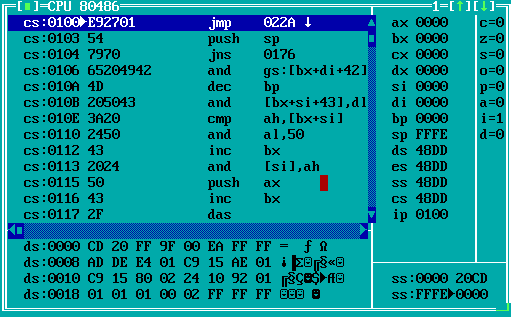
1. Какова структура файла «плохого» ЕХЕ? С какого адреса располагается код? Что располагается с адреса 0?

**Ответ:** В «плохом» EXE данные и код содержатся в одном сегменте. С адреса 0h идёт таблица настроек (Relocation table). Код располагается с адреса 300h.

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

**Ответ:** Файл .EXE должен содержать заголовок. В «плохом» EXE выделяется дополнительная память для заголовка. Код «хорошего» EXE-файла начинается с адреса 210h.

***Загрузка COM. модуля в основную память***



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

**Ответ:** сначала создаётся блок памяти для PSP и программы, затем загружается сегмент программы(в нем же лежит и код) с адреса 0100h, регистры DS,ES,SS,CS указывают на начало PSP(48DDh), указатель стека SP указывает на FFFE, в стек пишется значение 0000, в регистр IP записывается указатель на сегмент кода программы 0100h

1. Что располагается с адреса 0?

**Ответ:** сегмент PSP

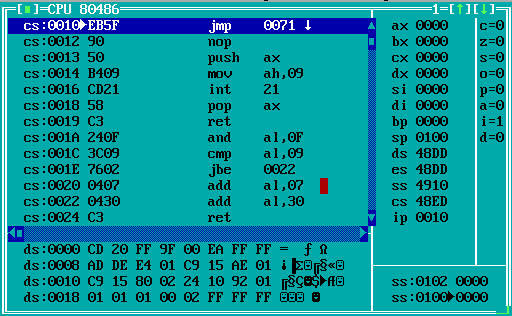
1. Какие значения имеют сегментные регистры? На какие области памяти они указывают?

**Ответ:** указывают на начало PSP

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

**Ответ:** Адрес последнего добавленного элемента– указателе стека SP (смещение относительно SS). В SS хранится селектор стека. Адреса в стеке 0h–FFFEh.

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



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

**Ответ:** .EXE файл имеет несколько сегментов в отличие от .COM файла, поэтому исходные значения регистров различаются. DS и ES указывают на начало PSP, SS – на начало стека, CS - на сегмент кода. Сегментный адрес PSP заносится в регистр DS в момент исполнения программы. DS=48DD, ES=48DD, SS=4910, CS=48ED.

1. На что указывают регистры DS и ES?

**Ответ:** и DS, и ES на начало PSP, до загрузки в DS адреса сегмента данных.

1. Как определяется стек?

**Ответ:** c помощью команды .STACK

1. Как определяется точка входа

**Ответ:** директива END завершает программу, одновременно указывая, с какой метки должно начинаться выполнение программы.

**Выводы.**

В результате выполнения данной лабораторной работы было изучено различие в структурах исходных текстов модулей типов .com и .exe, структур файлов загрузочных модулей и способов их загрузки в основную память.