**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 заголовочных модулей**

|  |  |  |
| --- | --- | --- |
| Студент гр. 9381 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Любимов В.А. |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Ефремов М.А. |

Санкт-Петербург

2021

**Цель работы**

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

**Описание функций и структуры данных**

1. – переводит число, представляемое четырьмя младшими битами в регистре AL, в 16-ричную цифру-символ.
2. - переводит число, содержащиеся в регистре AL, в 16-ричные цифры, записывающиеся в регистры AL и AH.
3. – переводит слово в регистре AX в четыре 16-ричные цифры, записывающие по адресу, находящемуся в DI.
4. - переводит число, содержащиеся в регистре AL, в 10-ричные цифры, записывающиеся по адресу, находящемуся в SI.
5. – при помощи функции 9h из прерывания 21h выводит строку на экран.
6. - выводит на экран информацию о типе PC. Для этого считывает значение предпоследнего байта ROM BIOS в AL. Сравнивает полученное код со значениями соответствующих типов PC. В случае успеха выводит на экран строку содержащую тип PC. Если полученный код не совпал не с одним из значений, соответствующих какому-либо типу PC, то при помощи функции BYTE\_TO\_HEX переводит полученный код в 16-ричное число и выводит его на экран.
7. – выводит номер версии операционной системы серийный OEM номер и серийный номер пользователя. Для этого используется функция 30h прерывания 21h. Полученные в результате данные переводится в символьный вид при помощи функций BYTE\_TO\_DEC, BYTE\_TO\_HEX, WRD\_TO\_HEX в составе строк с пояснениями.

**Ход выполнения работы**

1. Был написан текст исходного .COM модуля lb1\_com.asm, который определяет тип PС и версию системы. Затем были получены «плохой» .EXE модуль (последовательность команд: masm lb1\_com.asm → link lb1\_com.OBJ → lb1\_com.exe) и «хороший» .COM модуль (последовательность команд: masm lb1\_com.asm → link lb1\_com.OBJ → exe2bin.exe lb1\_com.exe lb1.com → lb1.com).
2. Был написан текст программы для «хорошего» .EXE модуля, после чего он был получен при помощи следующей последовательности команд: masm lb1\_exe.asm → link lb1\_exe.OBJ → lb1\_exe.exe.
3. Структуры полученных загрузочных модулей были сравнены в 16-ричном виде при помощи файлового менеджера FAR.
4. Структуры полученных загрузочных модулей также были сравнены при помощи отладчика TD.EXE.

**Ответы на контрольные вопросы**

**Отличия исходных текстов COM и EXE программ**

1. Сколько сегментов должна содержать COM-программа?

COM-программа состоит из единственного сегмента, содержащего и данные и код.

1. EXE-программа?

EXE-программа может состоять из нескольких сегментов. Также данные, стек и код находятся в различных сегментах.

1. Какие директивы должны обязательно быть в тексте COM-программы?
2. Все ли форматы команд можно использовать в COM-программе?

**Отличия форматов файлов COM и EXE модулей**

1. Какова структура файла COM? С какого адреса располагается код?
2. Какова структура «плохого»