

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №6**  
**по дисциплине «Операционные системы»**  
**Тема: Построение модуля динамической структуры.**

Студент гр. 7381

\_\_\_\_\_

Лукашев Р.С.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2019

## **Цель работы.**

Исследование возможности построения загрузочного модуля динамической структуры. В отличие от предыдущих лабораторных работ в этой работе рассматривается приложение, состоящее из нескольких модулей, а не из одного модуля простой структуры. В этом случае разумно предположить, что все модули приложения находятся в одном каталоге и полный путь в этот каталог можно взять из среды, как это делалось в работе 2. Понятно, что такое приложение должно запускаться в соответствии со стандартами ОС.

В работе исследуется интерфейс между вызывающим и вызываемым модулями по управлению и по данным. Для запуска вызываемого модуля используется функция 4B00h прерывания int 21h. Все загрузочные модули находятся в одном каталоге. Необходимо обеспечить возможность запуска модуля динамической структуры из любого каталога.

## **Постановка задачи.**

Шаг 1. Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .EXE, который выполняет функции:

- 1) Подготавливает параметры для запуска загрузочного модуля из того же каталога, в котором находится он сам. Вызываемому модулю передается новая среда, созданная вызывающим модулем и новая командная строка.
- 2) Вызываемый модуль запускается с использованием загрузчика.
- 3) После запуска проверяется выполнение загрузчика, а затем результат выполнения вызываемой программы. Необходимо проверять причину завершения и, в зависимости от значения, выводить соответствующее сообщение. Если причина завершения 0, то выводится код завершения.

В качестве вызываемой программы необходимо взять программу ЛР 2, которая распечатывает среду и командную строку. Эту программу следует немного модифицировать, вставив перед выходом из нее обращение к функции ввода

символа с клавиатуры. Введенное значение записывается в регистр AL и затем происходит обращение к функции выхода 4Ch прерывания int 21h.

Шаг 2. Запустите отлаженную программу, когда текущим каталогом является каталог с разработанными модулями. Программа вызывает другую программу, которая останавливается, ожидая символ с клавиатуры.

Введите произвольный символ из числа A-Z. Посмотрите причину завершения и код. Занесите полученные данные в отчет.

Шаг 3. Запустите отлаженную программу, когда текущим каталогом является каталог с разработанными модулями. Программа вызывает другую программу, которая останавливается, ожидая символ с клавиатуры.

Введите комбинацию символов Ctrl-C. Посмотрите причину завершения и код. Занесите полученные данные в отчет.

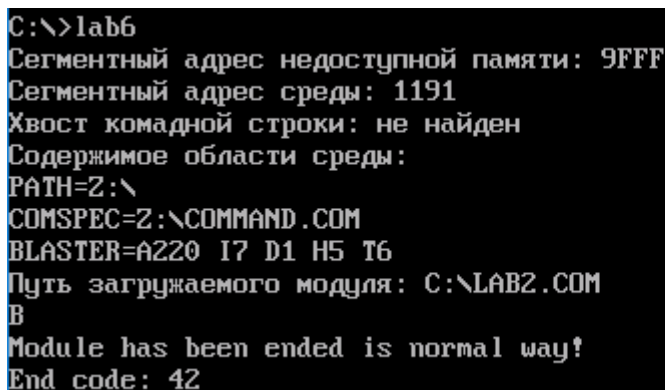
Шаг 4. Запустите отлаженную программу, когда текущим каталогом является какой-либо другой каталог, отличный от того, в котором содержатся разработанные программные модули.

Повторите ввод комбинаций клавиш. Занесите полученные данные в отчет.

Шаг 5. Запустите отлаженную программу, когда модули находятся в разных каталогах. Занесите полученные данные в отчет.

### Ход работы.

- 1) Запуск программы из каталога с разработанными модулями и ввод символа B



```
C:\>lab6
Сегментный адрес недоступной памяти: 9FFF
Сегментный адрес среды: 1191
Хвост командной строки: не найден
Содержимое области среды:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Путь загружаемого модуля: C:\LAB2.COM
B
Module has been ended is normal way!
End code: 42
```

Рисунок 1– запуск программы в том-же каталоге

- 2) Запуск программы из каталога с разработанными модулями и завершение через Ctrl + C.

```
C:\>lab6
Сегментный адрес недоступной памяти: 9FFF
Сегментный адрес среды: 1191
Хвост командной строки: не найден
Содержимое области среды:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Путь загружаемого модуля: C:\LAB2.COM
♥
Module has been ended is normal way!
End code: 03
```

Рисунок 2 – запуск программы в том-же каталоге 2

- 3) Запуск программы во время нахождения в другом каталоге


```
C:\ANT>..\lab6
Сегментный адрес недоступной памяти: 9FFF
Сегментный адрес среды: 1191
Хвост командной строки: не найден
Содержимое области среды:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Путь загружаемого модуля: C:\LAB2.COM
В
Module has been ended is normal way!
End code: 42
```

Рисунок 3 – запуск программы из другого каталога

```
C:\ANT>..\lab6
Сегментный адрес недоступной памяти: 9FFF
Сегментный адрес среды: 1191
Хвост командной строки: не найден
Содержимое области среды:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Путь загружаемого модуля: C:\LAB2.COM
♥
Module has been ended is normal way!
End code: 03
```

Рисунок 4 – запуск программы из другого каталога 2

- 4) Запуск программы при условии, что программный и загрузочный модуль находятся в разных каталогах



```
C:\>lab6  
File not found!
```

Рисунок 5 – запуск программы при условии, что модули находятся в разных каталогах

### **Ответы на контрольные вопросы.**

1. Как реализовано прерывание Ctrl-C?

**Ответ:** При нажатии сочетания клавиш Ctrl-C или Ctrl-Break вызывается прерывание INT 23h, которое завершает текущий процесс, при этом управление передается по адресу 0000:008с.

2. В какой точке заканчивается вызываемая программа, если код причины завершения 0?

**Ответ:** Если код причины завершения 0, то вызываемая программа заканчивается в месте вызова функции 4Ch прерывания INT 21h.

3. В какой точке заканчивается вызываемая программа по прерыванию Ctrl-C?

**Ответ:** При нажатии сочетания клавиш Ctrl+C программа завершает работу в том месте, где программа ожидала ввода символа, т.е. в точке вызова функции 01h прерывания int 21h.

### **Заключение.**

В ходе лабораторной работы был построен загрузочный модуль динамической структуры, а также модифицирован ранее построенный программный модуль. Изучены дополнительные функции работы с памятью и исследованы возможности использования интерфейса между вызывающим и вызываемым модулями по управлению и по данным.