

**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

\_\_\_\_\_

Ильясов А.В.

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.

## **Необходимые сведения для составления программы**

Для загрузки и выполнения одной программы из другой используется функция 4B00h прерывания int 21h (загрузчик ОС). Перед обращением к этой функции необходимо выполнить следующие действия:

1) Подготовить место в памяти. При начальном запуске программы ей отводится вся доступная в данный момент память OS, поэтому необходимо освободить место в памяти. Для этого можно использовать функцию 4Ah прерывания int 21h. Эта функция позволяет уменьшить отведенный программе блок памяти. Перед вызовом функции надо определить объем памяти, необходимый программе ЛР6 и задать в регистре ВХ число параграфов, которые будут выделяться программе. Если функция 4Ah не может быть выполнена, то устанавливается флаг переноса CF=1 и в АХ заносится код ошибки:

- 7 - разрушен управляющий блок памяти;
- 8 - недостаточно памяти для выполнения функции;
- 9- неверный адрес блока памяти.

Поэтому после выполнения каждого прерывания `int 21h` следует проверять флаг переноса `CF=1`.

2) Создать блок параметров. Блок параметров - это 14-байтовый блок памяти, в который помещается следующая информация:

- `dw` сегментный адрес среды
- `dd` сегмент и смещение командной строки
- `dd` сегмент и смещение первого FCB
- `dd` сегмент и смещение второго FCB

Если сегментный адрес среды 0, то вызываемая программа наследует среду вызывающей программы. В противном случае вызывающая программа должна сформировать область памяти в качестве среды, начинающуюся с адреса кратного 16 и поместить этот адрес в блок параметров.

Командная строка записывается в следующем формате:

первый байт - счетчик, содержащий число символов в командной строке, затем сама командная строка, содержащая не более 128 символов.

На блок параметров перед загрузкой вызываемой программы должны указывать `ES:BX`.

3) Подготовить строку, содержащую путь и имя вызываемой программы. В конце строки должен стоять код ASCII 0. На подготовленную строку должны указывать `DS:DX`.

4) Сохранить содержимое регистров `SS` и `SP` в переменных. При восстановлении `SS` и `SP` нужно учитывать, что `DS` необходимо также восстановить.

Когда вся подготовка выполнена, вызывается загрузчик `OS` следующей последовательностью команд:

```
mov AX, 4B00h
int 21h
```

Если вызываемая программа не была загружена, то устанавливается флаг переноса `CF=1` и в `AX` заносится код ошибки:

1 - если номер функции неверен;

- 2 - если файл не найден;
- 5 - при ошибке диска;
- 8 - при недостаточном объеме памяти;
- 10 - при неправильной строке среды;
- 11 - если не верен формат.

Если CF=0, то вызываемая программа выполнена и следует обработать ее завершение. Для этого необходимо воспользоваться функцией 4Dh прерывания int 21h. В качестве результата функция возвращает в регистре AH причину, а в регистре AL код завершения.

Причина завершения в регистре AH представляется следующими кодами:

- 0 - нормальное завершение;
- 1 - завершение о Ctrl-Break;
- 2 - завершение о ошибке устройства;
- 3 - завершение по функции 31h, оставляющей программу резидентной.

Код завершения формируется вызываемой программой в регистре AL перед выходом в OS с помощью функции 4Ch прерывания int 21h.

В качестве вызываемой программы целесообразно использовать программу, разработанную в Лабораторной работе №2, модифицировав ее следующим образом. Перед выходом из программы перед выполнением функции 4Ch прерывания int 21h следует записать с клавиатуры символ и поместить введенный символ в регистр AL, в качестве кода завершения. Это можно сделать с помощью функции 01h прерывания int 21h.

```
mov AH, 01h  
int 21h
```

Введенный символ остается в регистре AL и служит аргументом для функции 4Ch прерывания int 21h.

## Ход работы

1) Запуск программы из того же каталога, в котором расположены разработанные модули. Для этого был создан каталог test, в который были перемещены модули lr6.exe и lr2.com. При первом запуске была нажата клавиша «а», а во-второй раз сочетание клавиш ctrl-c, которое в MS-DOS не работает, и, вместо прерывания, печатается символ сердечка. Ниже представлены результаты работы программы с разными нажатыми клавишами:

```
C:\TEST>LR6.EXE
The segment address of the inaccessible memory (from PSP): 9FFF

The segment address of the environment passed to the program: 1191

The tail of comand line:

The contents of the environment:
PATH=Z:\ COMSPEC=Z:\COMMAND.COM BLASTER=A220 I7 D1 H5 T6

The path of the loaded module:
C:\TEST\LR2.COM a
Normal termination!
End code: 61

C:\TEST>
```

рисунок 1 – запуск программы с последующим нажатием «а»

```
C:\TEST>LR6.EXE
The segment address of the inaccessible memory (from PSP): 9FFF

The segment address of the environment passed to the program: 1191

The tail of comand line:

The contents of the environment:
PATH=Z:\ COMSPEC=Z:\COMMAND.COM BLASTER=A220 I7 D1 H5 T6

The path of the loaded module:
C:\TEST\LR2.COM ♥
Normal termination!
End code: 03

C:\TEST>
```

рисунок 2 – запуск программы с вводом ctrl-c

2) Запуск программы из каталога, отличного от того, в котором содержатся разработанные модули. Для этого в каталоге test был создан подкаталог test1, из которого и была вызвана программа:

```

C:\TEST\TEST1>C:\TEST\LR6.EXE
The segment address of the inaccessible memory (from PSP): 9FFF

The segment address of the environment passed to the program: 1191

The tail of comand line:

The contents of the environment:
PATH=Z:\ COMSPEC=Z:\COMMAND.COM BLASTER=A220 I7 D1 H5 T6

The path of the loaded module:
C:\TEST\LR2.COM a
Normal termination!
End code: 61

```

рисунок 3 – запуск программы из другого каталога с нажатием «а»

```

C:\TEST\TEST1>C:\TEST\LR6.EXE
The segment address of the inaccessible memory (from PSP): 9FFF

The segment address of the environment passed to the program: 1191

The tail of comand line:

The contents of the environment:
PATH=Z:\ COMSPEC=Z:\COMMAND.COM BLASTER=A220 I7 D1 H5 T6

The path of the loaded module:
C:\TEST\LR2.COM ♥
Normal termination!
End code: 03

```

рисунок 4 – запуск программы из другого каталога с вводом ctrl-c

3) запуск программы, когда модули находятся в разных каталогах. Для этого модуль lr6.exe был помещен в подкаталог test1 и был вызван из него:

```

C:\TEST\TEST1>LR6.EXE
File not found!

```

Рисунок 5 – запуск программ из разных каталогов

## Выводы

В ходе данной лабораторной была исследована возможность построения загрузочного модуля динамической структуры: было реализовано взаимодействие между вызывающим и вызываемым модулями.

## **Ответы на контрольные вопросы.**

### **1. Как реализовано прерывание Ctrl-C?**

Ответ: прерывание 23h вызывается, если была нажата комбинация клавиш Ctrl-C или Ctrl-Break. Адрес, по которому передается управление (0000:008c). Управление передаётся тогда, когда DOS распознает, что пользователь нажал Ctrl-Break или Ctrl-C. Адрес по вектору INT 23h копируется в поле PSP Ctrl-Break Address функциями DOS 26h (создать PSP) и 4Ch (EXEC). Исходное значение адреса обработчика Ctrl-Break восстанавливается из PSP при завершении программы. Таким образом, по завершении порожденного процесса будет восстановлен адрес обработчика Ctrl-Break из родительского процесса.

### **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 прерываний int 21h.

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

Ответ: в месте, где программа ожидала ввода символа: в точке вызова функции 01h прерывания int 21h.