

**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. 0382

Азаров М.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **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.

### **Задание.**

**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. Для выполнения первого шага понадобилось разработать модуль типа .EXE , который содержит следующие процедуры:

Название процедуры	Что делает
Main	Главная процедура выполняющая поставленную задачу.
Free_mem	Освобождает память, которую не использует программа с помощью процедуры Malloc и обрабатывает результат этой операции.

Malloc	Процедура которая непосредственно освобождает память, используется в процедуре Free_mem
Set_param_block	Настраивает блок параметров EPB (EXEC Parameter Block) необходимый для вызывающей программы.
Set_file_path	Подготавливает строку, содержащую путь и имя вызываемой программы.
Load_prog	Запускает дочернюю программу и обрабатывает ее завершение (или ошибки загрузки)
Handle_errors	Обработка ошибок при загрузке дочерней программы.
Handle_exit	Обработка завершения дочерней программы .

Также понадобилось немного модифицировать модуль LB2.COM, который выступает в роли дочерней программы. Изменения заключались в том, что перед выходом из программы происходит обращение к функции ввода символа с клавиатуры. Затем введенное значение записывается в регистр AL и происходит обращение к функции выхода 4Ch прерывания int 21h.

2. Родительская и дочерняя программы находятся в одном каталоге. После запуска, родительская программа действительно вызывает дочерний модуль:

```

C:\>MAIN.EXE
Success free memory

Address unavailable memory: 9FFFh
Environment address: 0221h
Content tail:parm str
Content environment:
  PATH=Z:\
  COMSPEC=Z:\COMMAND.COM
  BLASTER=A220 I7 D1 H5 T6
Path: C:\LB2.COM
A
Programm normal ended with code = A
C:\>

```

Первое сообщение «Success free memory» говорит о том что освобождение памяти произошло успешно. После выводится работа модуля LB2.COM , который был вызван родительской программой. Затем, как требует того задания дочерний модуль запрашивает ожидает ввод с клавиатуры. С клавиатуры был введен символ «А». Как и ожидалось программа завершилась нормально с кодом «А».

3. Родительская и дочерняя программы находятся в одном каталоге. После запуска и вызова дочерней программы попытаемся ввести Ctrl+C:



```
C:\>MAIN.EXE
Success free memory

Address unavailable memory: 9FFFh
Environment address: 0221h
Content tail:parm str
Content environment:
    PATH=Z:\
    COMSPEC=Z:\COMMAND.COM
    BLASTER=A220 I7 D1 H5 T6
Path: C:\LB2.COM
♥
Programm normal ended with code = ♥
C:\>_
```

К сожалению DosBox не распознает комбинацию Ctrl+C как сигнал завершения Ctrl+Break, а распознает как символ «♥». Поэтому DosBox не прервал работу программы и она завершилась нормально с введенным кодом «♥».

4. Теперь оба модуля находятся в другом каталоге под названием «OtherDir», в командной строке об этом говорит путь до директории в которой мы сейчас находимся:

```

C:\>cd \OTHERDIR

C:\OTHERDIR>MAIN.EXE
Success free memory

Address unavailable memory: 9FFFh
Environment address: 0221h
Content tail:parm str
Content environment:
    PATH=Z:\
    COMSPEC=Z:\COMMAND.COM
    BLASTER=A220 I7 D1 H5 T6
Path: C:\OTHERDIR\LB2.COM
d
Program normal ended with code = d

C:\OTHERDIR>

```

Как видим это не повлияло на корректность работы обоих модулей.

5. Теперь попробуем запустить программу когда модули находятся в разных директориях:

```

C:\>MAIN.EXE
Success free memory

File not found

C:\>

```

Как видим в таком случае родительский модуль не может найти дочерний модуль, тем самым происходит ошибка загрузки модуля и выводится соответствующее сообщение.

### Ответы на контрольные вопросы.

1) Как реализовано прерывание Ctrl-C?

**Ответ:** При нажатии Ctrl-C или Ctrl-Break выполняется прерывание INT 23h. Адрес в этом векторе (0000:008c) - это адрес, по которому передается управление. 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:** В точке вызова функции `4ch int 21h`.

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

**Ответ:** Программа завершится в точке считывания символа .

### **Вывод.**

В ходе выполнения лабораторной работы было изучено построение загрузочного модуля динамической структуры. Также был исследован интерфейс между вызывающим и вызываемым модулями по управлению и по данным. Была разработана программа, которая подготавливает параметры для запуска дочернего модуля и запускает его.