

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

Студентка гр. 0382

Рубежова Н.А.

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 выполнения.

Шаг 1. Был написан модуль `lr6.asm` типа `.EXE` со следующими процедурами:

`MEM_FREE` — процедура подготовки места в памяти. Т.к. при начальном запуске программы ей отводится вся доступная в данный момент память OS, необходимо освободить место в памяти.

`CREATE_PARAM_BLOCK` — процедура подготовки блока параметров для вызова загрузочного модуля из директории, в которой находится исполняемый модуль.

`GET_FILEPATH` — процедура получения пути к вызываемому модулю.

`TO_CALL_MODULE` — процедура запуска вызываемого модуля из основного модуля.

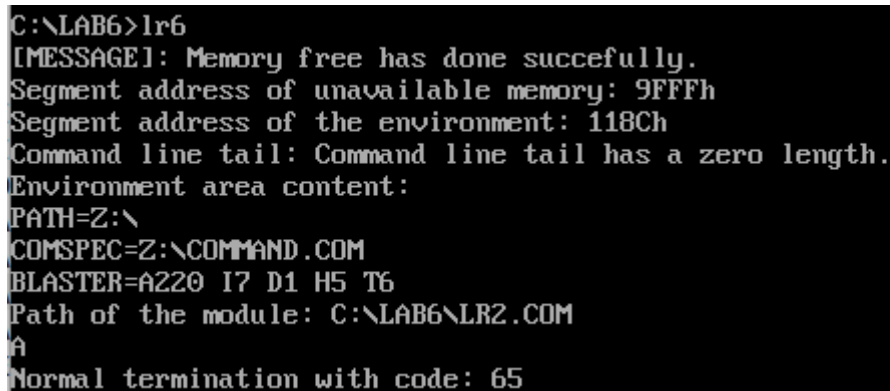
`PRINT` — вспомогательная процедура печати строки

BYTE_TO_DEC – процедура, необходимая для перевода кода завершения программы в 10с/с.

MAIN — основная функция, где происходит вызов вышеописанных функций

При запуске модуля lr6.exe освобождается «лишняя» память с выводом соответствующего сообщения, подготавливаются параметры для запуска вызываемого модуля, запускается вызываемый модуль, выводя информативные сообщения о результатах работы и ошибках. В данной лабораторной работе вызывающим модулем выступает lr6.exe, а вызываемый модуль - lab2.com, взятый из 2ой лабораторной работы, где в конце программы добавлен вызов функции DOS 01h - ввод с клавиатуры.

Шаг 2. Запустим модуль lr6.exe и введем символ А, когда текущим каталогом является каталог с разработанными модулями. Вывод программы см. на Рис. 1.



```
C:\LAB6>lr6
[MESSAGE]: Memory free has done succefully.
Segment address of unavailable memory: 9FFFh
Segment address of the environment: 118Ch
Command line tail: Command line tail has a zero length.
Environment area content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Path of the module: C:\LAB6\LR2.COM
A
Normal termination with code: 65
```

Рисунок 1 – Запуск модуля lr6.exe с вводом символа А (ASCII-код: 65), когда текущим каталогом является каталог с разработанными модулями

Видим, что программа вывела сообщение о том, что память была успешно освобождена. Также успешно был вызван модуль lab2.com с выводом необходимой информации и запросом на ввод с клавиатуры. Ввели символ А, после чего вывелось сообщение об успешном завершении программы с кодом 65, что соответствует ASCII-коду символа А. Вызывающий и вызываемый модули отработали верно.

Шаг 3. Запустим модуль lr6.exe и введем комбинацию Ctrl+C, когда текущим каталогом является каталог с разработанными модулями. Вывод программы см. на Рис. 2.

```
C:\LAB6>lr6
[MESSAGE]: Memory free has done succefully.
Segment address of unavailable memory: 9FFFh
Segment address of the environment: 118Ch
Command line tail: Command line tail has a zero length.
Environment area content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Path of the module: C:\LAB6\LR2.COM
♥
Normal termination with code: 3
```

Рисунок 2 – Запуск модуля lr6.exe с вводом комбинации Ctrl+C, когда текущим каталогом является каталог с разработанными модулями

Вывод программы аналогичен шагу 2, так как DosBox распознает комбинацию Ctrl+C за символ (♥, ASCII код = 3).

Шаг 4. Создали директорию dir_ и поместили туда загрузочные модули, а разработанные модули остались в основной папке lab6. Запустим модуль lr6.exe с вводом символа A, когда текущим каталогом является dir_, отличный от того, в котором содержатся разработанные модули.

```
C:\LAB6\DIR_>lr6
[MESSAGE]: Memory free has done succefully.
Segment address of unavailable memory: 9FFFh
Segment address of the environment: 118Ch
Command line tail: Command line tail has a zero length.
Environment area content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Path of the module: C:\LAB6\DIR_\LR2.COM
A
Normal termination with code: 65
```

Рисунок 3 – Запуск модуля lr6.exe с вводом символа A (ASCII-код: 65), когда текущим каталогом является каталог dir_

```

C:\LAB6\DIR_>lr6
[MESSAGE]: Memory free has done succefully.
Segment address of unavailable memory: 9FFFh
Segment address of the environment: 118Ch
Command line tail: Command line tail has a zero length.
Environment area content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Path of the module: C:\LAB6\DIR_\LR2.COM
♥
Normal termination with code: 3

```

Рисунок 4 – Запуск модуля lr6.exe с вводом комбинации Ctrl+C, когда текущим каталогом является каталог dir_

Сравнивая шаги 2-4 и по рисункам 3-4, видим, что выполнение программы не зависит от выбора текущего каталога.

Шаг 5. Переместим загрузочный вызываемый модуль lr2.com из dir_ обратно в основную папку lab6, а вызывающий модуль lr6.exe оставим в созданной директории dir_.

```

C:\LAB6\DIR_>lr6
[MESSAGE]: Memory free has done succefully.
[ERROR]: Failed to load the program

```

Рисунок 5 – Запуск модуля lr6.exe, когда модули находятся в разных каталогах

Видим, что произошла ошибка загрузки вызываемого модуля, так как он ищется в каталоге, из которого вызывается. То есть если загрузочные модули вызывающего и вызываемого модулей находятся в разных директориях, то вызвать второй модуль из первого невозможно.

Ответы на контрольные вопросы.

1. Как реализовано прерывание Ctrl+C?

При вводе комбинации клавиш Ctrl+C управление передается процедуре, на которую указывает вектор прерывания 23H. Стандартный обработчик прерывания 23H завершает выполнение программы. Завершение работы программы по Ctrl-Break мы использовали в данной лабораторной работе.

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

После вызова функции 4CH прерывания int 21H в конце модуля.

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

После вызова функции ввода с клавиатуры - функции 01H прерывания int 21H, поскольку вызываемая программа заканчивается на моменте ввода с клавиатуры.

Выводы.

В результате работы были исследованы возможности построения загрузочного модуля динамической структуры, а также интерфейс между вызывающим и вызываемым модулями по управлению и данным.