

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

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

Студентка гр. 8382

Кулачкова М.К.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

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.

Ход выполнения работы.

Была реализована программа, которая осуществляет вызов загрузочного модуля из того же каталога, в котором находится она сама. Перед запуском вызываемой программы осуществляется подготовка параметров для запуска. Так как изначально запущенной программе отводится вся доступная в данный момент память операционной системы, необходимо подготовить в памяти место для запуска новой программы, освободив память, не используемую вызывающей программой. Память освобождается при помощи функции 4Ah прерывания int 21h. Если память не может быть освобождена, выводится сообщение об ошибке в зависимости от кода ошибки.

В сегменте данных вызывающей программы расположен блок параметров, в который помещается следующая информация: сегментный адрес среды, сегмент и смещение командной строки, сегмент и смещение первого блока управления файлом (FCB), сегмент и смещение второго FCB. В качестве сегментного адреса среды указывается 0, и в таком случае вызываемая программа наследует среду вызывающей программы. Другим параметрам

присваиваются значения соответствующих параметров вызывающей программы.

Осуществляется подготовка строки, содержащей путь до вызываемой программы. Для этого в строку записывается путь до вызываемой программы, который расположен после переменных среды, и дополняется именем вызываемого модуля.

Программа сохраняет содержимое регистров SS, SP и DS в переменных. В регистры DS:DX заносится адрес строки с маршрутом до вызываемой программы, в регистры ES:BX – адрес блока параметров. После этого вызывается загрузчик ОС с помощью функции 4Bh прерывания int 21h. Если программа не была загружена, выводится сообщение об ошибке в соответствии с кодом ошибки. Иначе обрабатывается завершение программы.

При помощи функции 4Dh прерывания int 21h определяется причина завершения программы. Функция возвращает код причины завершения в регистре AH. Программа выводит сообщение о причине завершения. Если программа завершилась нормально, выводится код завершения, возвращаемый функцией в регистре AL.

В качестве вызываемой программы использована программа, разработанная в лабораторной работе № 2, модифицированная так, чтобы перед выходом из программы функцией 01h прерывания int 21h запрашивался символ с клавиатуры и записывался в регистр AL в качестве кода завершения.

Программа была запущена из каталога с разработанными модулями. После вызова этой программой другой программы был введен символ «А». Программа завершилась нормально с кодом завершения «А». Результат работы представлен на рисунке 1.

Программа была запущена повторно из того же каталога, при этом при запросе вызываемой программой символа с клавиатуры была введена комбинация Ctrl-C. Результат выполнения программы представлен на рисунке 2.

```
C:\>LR6.EXE
Inaccessible memory address: 9FFF
Environment address: 05B0
The command line tail is empty
Environment contents:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Program path:
C:\LR2.COM
A
Program terminated normally
Termination code:A
```

Рисунок 1 - Запуск программы с вводом символа «A»

```
C:\>LR6.EXE
Inaccessible memory address: 9FFF
Environment address: 05B0
The command line tail is empty
Environment contents:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Program path:
C:\LR2.COM
♥
Program terminated normally
Termination code:♥
```

Рисунок 2 - Запуск программы с вводом символов Ctrl-C

В каталоге с разработанными модулями был создан новый каталог, куда были скопированы полученные модули. Программа была вызвана из нового каталога с вводом символа «W». Результат выполнения программы представлен на рисунке 3.

```
C:\>LAB6\LR6.EXE
Inaccessible memory address: 9FFF
Environment address: 05B0
The command line tail is empty
Environment contents:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Program path:
C:\LAB6\LR2.COM
W
Program terminated normally
Termination code:W
```

Рисунок 3 - Запуск программы из другого каталога с вводом символа «W»

Программа была запущена еще раз из нового каталога, при этом при запросе ввода символа была введена комбинация Ctrl-C. Результат работы программы представлен на рисунке 4.

```

C:\>LAB6\LR6.EXE
Inaccessible memory address: 9FFF
Environment address: 05B0
The command line tail is empty
Environment contents:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Program path:
C:\LAB6\LR2.COM
♥
Program terminated normally
Termination code:♥

```

Рисунок 4 - Запуск программы из другого каталога с вводом символов Ctrl-C

Из нового каталога была удалена копия вызываемого модуля. Затем оттуда была запущена вызывающая программа. Результат ее работы представлен на рисунке 5.

```

C:\>LAB6\LR6.EXE
Loading error: file not found

```

Рисунок 5 - Запуск программы, когда модули находятся в разных каталогах

Ответы на вопросы.

1. Как реализовано прерывание Ctrl-C?
 - При нажатии Ctrl-C процессу посылается сигнал SIGINT. Если процесс не установил собственный обработчик сигнала, система запускает обработчик по умолчанию, который завершает выполнение процесса.
2. В какой точке заканчивается вызываемая программа, если код причины завершения 0?
 - Если код причины завершения 0, значит, была вызвана функция выхода из программы (4Ch прерывания int 21h) и программа завершилась нормально.
3. В какой точке заканчивается вызываемая программа по прерыванию Ctrl-C?
 - При нажатии Ctrl-C вызывается прерывание int 23h. Обработчик по умолчанию вызывает немедленное завершение программы.

Выводы.

Было реализовано приложение, состоящее из двух загрузочных модулей, расположенных в одном каталоге: вызывающего и вызываемого. Вызывающий модуль осуществляет подготовку к запуску вызываемого модуля, а затем вызывает его при помощи функции 4Vh прерывания int 21h. В ходе выполнения работы был исследован интерфейс между вызывающим и вызываемым модулями по управлению и по данным.