

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. 7383

Рудоман В.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2019

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

Шаг 3. 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 4. Запустите отлаженную программу, когда модули находятся в разных каталогах. Занесите полученные данные в отчет.

Оформить отчёт и ответить на контрольные вопросы.

Процедуры, которые используются в программе.

| | |
|-------------|---|
| TETR_TO_HEX | Получение одного из двух символов 16-ричного числа в AX. |
| BYTE_TO_HEX | Перевод байта из регистра AL в два символа 16-ричного числа в AX. |
| freeMem | Функция необходимая для освобождения памяти. |
| exitProgram | Функция необходимая для выхода из программы. |
| READ | Функция, необходимая, для определения имени, вызываемого загрузочного модуля. |
| PRINT | Функция печати на экран. |

Переменные, которые используются в программе.

| | | |
|--------------|----|--|
| successPrint | db | Применяется для вывода информации о том, что процесс завершен успешно. |
| errPrint | db | Применяется для вывода информации о том, что вызываемый модуль не был найден. |
| ctrlCPrint | db | Применяется для вывода информации о том, что процесс был завершен по прерыванию ctrl + c. |
| fileName | db | Строка применяемая для хранения имени вызываемого модуля. |
| EOL | db | Флаг указывающий на конец строки. |
| stackS | dw | Применяется для запоминания регистра ss перед вызовом загрузочного модуля. |
| stackP | dw | Применяется для запоминания регистра sp перед вызовом загрузочного модуля. |
| MEMORY | db | Флаг, указывающий на то, было ли успешно освобождено место в памяти под вызываемый загрузочный модуль или нет. |

Ход выполнения работы

Шаг 1: Результаты выполнения работы представлены на рисунке №1:

```
Assembling file:   LR6.ASM
Error messages:   None
Warning messages: None
Passes:           1
Remaining memory: 472k

C:\>tlink LR6.OBJ
Turbo Link  Version 5.1 Copyright (c) 1992 Borland International

C:\>LR6.EXE
                Laboratornaya 2.  Sdelal Rudoman Vadim. Gruppa 7383
Segmentniy adress pamyati:      9FFF
Segmentniy adress sredi:       118A
Hvost kommandnoy stroki:

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

Way:
C:\LR2.COM♥♥ Process ended successful, code:  03
C:\>_
```

Рис. 1 – результат работы программы

Шаг 2: Результаты выполнения работы представлены на рисунке №2:

```
Assembling file:   LR6.ASM
Error messages:   None
Warning messages: None
Passes:           1
Remaining memory: 472k

C:\>tlink LR6.OBJ
Turbo Link  Version 5.1 Copyright (c) 1992 Borland International

C:\>LR6.EXE
                Laboratornaya 2.  Sdelal Rudoman Vadim. Gruppa 7383
Segmentniy adress pamyati:      9FFF
Segmentniy adress sredi:       118A
Hvost kommandnoy stroki:

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

Way:
C:\LR2.COM♥♥ Process ended successful, code:  03
C:\>_
```

Рис. 2 – результат выполнения программы

Шаг 3: Результаты выполнения работы представлены на рисунке №3:

```
Error messages:   None
Warning messages: None
Passes:           1
Remaining memory: 472k

C:\>tlink LR6.OBJ
Turbo Link  Version 5.1 Copyright (c) 1992 Borland International

C:\>LR6.EXE
                Laboratornaya 2.  Sdelal Rudoman Vadim. Gruppa 7383
Segmentniy adress pamyati:      9FFF
Segmentniy adress sredi:       118A
Hvost kommandnoy stroki:

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

Way:
C:\LR2.COM♥♥ Process ended successful, code:  03
C:\>LR6.EXE
                Error! No fileC:\LR2.com
C:\>_
```

Рис. 3 – результат выполнения программы без «2.com»

Контрольные вопросы

1. Как реализовано прерывание Ctrl-C?

При нажатии клавиш ctrl+c (DOS вызывает INT 23H) управление передаётся по адресу 0000:008Ch. Этот адрес копируется в PSP функциями 26h и 4Ch и восстанавливается из PSP при выходе из программы.

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

В месте вызова функции 4Ch прерывания 21h (завершение текущего процесса и передача управления процессу-родителю), так как возникли проблемы с освобождением памяти.

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

Программа завершается в любой ее точке, сразу после обработки прерывания по Ctrl-C. Конкретно в данной программе: в месте вызова функции 01h прерывания 21h.

Вывод

В результате выполнения данной лабораторной работы были исследованы организация загрузочных модулей динамической структуры. А также были получены практические навыки по построению модуля динамической структуры.