

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

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

Студент гр. 0381

Соколов Д. В.

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.

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 передаётся 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 lab2, которая распечатывает среду и командную строку. Эту программу следует немного модифицировать, вставив перед выходом из неё обращение к функции ввода символа с клавиатуры. Введённое 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 отчёт.

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 отчёт.

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 отчёт.

Шаг 5. Запустите отлаженную программу, когда модули находятся в разных каталогах. Занесите полученные данные в отчёт.

Выполнение работы.

Были написаны структуры данных:

- PARAMETR_BLOCK – Параметры для запуска дочерней программы;
- KEEP_SS – сохранение ss;
- KEEP_SP – сохранение sp.

Были написаны строки для хранения информации:

- STR_FILE_NAME – хранение имени файла;
- STR_PATCH_NAME хранение пути до файла.

Были написаны строки для вывода информации:

- Для вывода ошибок высвобождения памяти:
 - STR_MEMORY_7;

- STR_MEMORY_8;
- STR_MEMORY_9;
- Для вывода ошибок запуска дочерней программы:
 - STR_ERROR_1;
 - STR_ERROR_2;
 - STR_ERROR_5;
 - STR_ERROR_8;
 - STR_ERROR_10;
 - STR_ERROR_11;
- Для вывода состояний завершения:
 - STR_COMPLETION_0;
 - STR_COMPLETION_1;
 - STR_COMPLETION_2;
 - STR_COMPLETION_3.

Были составлены функции (см. табл.1)

Таблица 1 – функции в программе

Процедура	Описание
BYTE_TO_DEC	Перевод байта в 10 систему счисления
WRITESTRING	Вывод строки на экран
NEW_LINE	Перевод указателя на новую строку
FREE_MEMORY	Высвобождение не используемой памяти и обработка возможных ошибок
SET_PARAMETERS	Создание parameter block
SET_FULL_FILE_NAME	Считывание пути до файла и построение полного пути
START_LAB2	Загрузка дочерней программы и её запуск
COMMENT_LOAD	Вывод результатов загрузки программы
MAIN	Главная функция

В результате выполнения были получены следующие значения(рис.1-4):

```
C:\>LAB6.EXE
Locked memory address: 9FFFh
Environment address: 1193h
In Command tail no sybmols
Content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Patch:
C:\LAB2.COM
Q
Normal completion. Code = 81
```

Рисунок 1 – Запуск отлаженной программы из каталога с разработанными модулями.

```
C:\>LAB6.EXE
Locked memory address: 9FFFh
Environment address: 1193h
In Command tail no sybmols
Content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Patch:
C:\LAB2.COM
♥
Normal completion. Code = 3
```

Рисунок 2 – Запуск отлаженной программы из каталога с разработанными модулями и при ожидании ввода символа, нажимается комбинация символов Ctrl-C. Т.к. такой комбинации нет, программа принудительно завершается и выводится скан код 03h.

```
C:\> \TEST\LAB6.EXE
Locked memory address: 9FFFh
Environment address: 1193h
In Command tail no sybmols
Content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Patch:
C:\TEST\LAB2.COM
Q
Normal completion. Code = 81
```

Рисунок 3 – Запуск отлаженной программы из каталога не с разработанными модулями.



```
C:\> \TEST\LAB6.EXE  
File not found
```

A screenshot of a black command prompt window with white text. The first line shows the command prompt 'C:\>' followed by the command '\TEST\LAB6.EXE'. The second line shows the error message 'File not found'.

Рисунок 4 – Запуск отлаженной программы при условии, что модуль лабораторной работы 6 и модуль лабораторной работы 2.

Выводы.

Был построен пользовательский обработчик прерывания сигналов клавиатуры, использующий оригинальный обработчик.

ПРИЛОЖЕНИЕ А

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как реализовано прерывание Ctrl-C?

При обнаружении нажатия комбинации клавиш Ctrl-C, выполняется команда int 23h. По передаваемому адресу (0000:008C), DOS распознает, что пользователь нажал Ctrl-C. Этот адрес копируется в поле PSP (Ctrl-Break Address) функциями 26H и 4Ch, а затем восстанавливается из PSP при выходе из программы.

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

В точке вызова функции 4ch int 21h.

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

Программа завершится в точке считывания символа 01h int 21h.