**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.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 0382 |  | Диденко Д. В. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2022

**Цель работы.**

Исследование возможности построения загрузочного модуля динамической структуры. 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):



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



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.



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

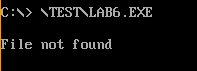


Рисунок 4 – Запуск отлаженной программы при условие, что модуль лабораторной работы 6 и модуль лабораторной работы 2.

**Выводы.**

В ходе лабораторной работы был построен загрузочный модуль динамической структуры, а также модифицирован ранее построенный программный модуль. Изучены дополнительные функции работы с памятью и исследованы возможности использования интерфейса между вызывающим и вызываемым модулями по управлению и по данным.

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 (0000:008С), 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.

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

В точке вызова функции 4ch int 21h.

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

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