

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. 0381

Шиняева А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы

Исследовать возможности построения загрузочного модуля динамической структуры и интерфейс между вызывающим и вызываемым модулями по управлению и по данным.

Постановка задачи

Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .EXE, который выполняет следующие функции:

1. подготавливает параметры для запуска загрузочного модуля из того же каталога, в котором находится сам;
2. запускает вызываемый модуль с использованием загрузчика;
3. после запуска проверяет выполнение загрузчика, а затем результат выполнения вызываемой программы.

В качестве вызываемой программы необходимо взять программу второй лабораторной работы, которая распечатывает среду и командную строку.

Запустить отлаженную программу, когда текущим каталогом является каталог с разработанными модулями. Программа вызывает другую программу, которая останавливается, ожидая символ с клавиатуры. Ввести символ из числа A-Z, пронаблюдать причину вывода и код.

Запустить отлаженную программу, когда текущим каталогом является каталог с разработанными модулями. Программа вызывает другую программу, которая останавливается, ожидая символ с клавиатуры. Ввести комбинацию символов Ctrl C, пронаблюдать причину вывода и код.

Запустить отлаженную программу, когда текущим каталогом является какой-либо другой каталог, отличный от того, в котором содержатся разработанные модули. Программа вызывает другую программу, которая останавливается, ожидая символ с клавиатуры. Повторить ввод комбинаций клавиш.

Запустить отлаженную программу, когда модули находятся в разных каталогах.

Выполнение работы

Для выполнения данной работы были реализованы следующие функции:

- `println` для вывода строки на экран;
- `num2dec` для записи в строку числа, лежащего в регистре AL;
- `free` для очистки неиспользуемой модулем памяти;
- `setp` для создания блока параметров;
- `getpath` для получения пути вызываемого модуля;
- `callp` для вызова модуля;
- `main` для выполнения поставленной в данной лабораторной работе задачи.

Для вывода информации на экран были созданы следующие строки:

- `unknown`, хранящая в себе строку 'Unknown error\$'
- `ferr7`, хранящая в себе строку 'Error: memory control block destroyed\$'
- `ferr8`, хранящая в себе строку 'Error: not enough memory to execute the function\$'
- `ferr9`, хранящая в себе строку 'Error: invalid memory block address\$'
- `cerr1`, хранящая в себе строку 'Error: function number is incorrect\$'
- `cerr2`, хранящая в себе строку 'Error: file could not be found\$'
- `cerr5`, хранящая в себе строку 'Disk error\$'
- `cerr8`, хранящая в себе строку 'Error: insufficient memory\$'
- `cerrA`, хранящая в себе строку 'Error: wrong environment string\$'
- `cerrB`, хранящая в себе строку 'Error: wrong format\$'
- `lgood`, хранящая в себе строку 'The program has ended with the code: \$'
- `lerr1`, хранящая в себе строку 'The program terminated by Ctrl-Break\$'
- `lerr2`, хранящая в себе строку 'The program terminated by device error\$'
- `lerr3`, хранящая в себе строку 'The program terminated by function 31h\$'.

Выполнение программы начинается с вызова функции `free` для освобождения неиспользуемой программой памяти. Для этого с помощью

функции 4Ah прерывания 21h выделяется необходимое программе количество памяти. В случае, если при этом возникла какая-либо ошибка, программа выводит соответствующее сообщение об ошибке.

После этого вызывается функция setp, которая заполняет выделенный в сегменте данных блок параметров.

С помощью функции getpath считывается в строку путь до текущего файла, после чего имя файла меняется на имя вызываемого модуля.

Функция callp сохраняет сегментные регистры DS и ES, а также регистры, отвечающие за стек. Устанавливаются регистры ES:BX так, чтобы они указывали на блок параметров, после чего вызывается программный модуль. После выполнения модуля регистры возвращаются, проверяется работа модуля.

Разработанный программный код см. в приложении А.

Результаты работы программы представлены на рисунках 1-4.

```
C:\>lb6.exe
Memory segment: 9FFF
Segment media address:0203
Tail of command line: [EMPTY]
Environment scope content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Path:
C:\COM.COM
The program has ended with the code: 65
```

Рис. 1 - Ввод символа 'A'

```
C:\>lb6.exe
Memory segment: 9FFF
Segment media address:0203
Tail of command line: [EMPTY]
Environment scope content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Path:
C:\COM.COM ^C
The program terminated by Ctrl-Break
```

Рис. 2 - Ввод комбинации символов
Ctrl C

```

C:\>LAB6\LB6.EXE
Memory segment: 9FFF
Segment media address:0203
Tail of command line: [EMPTY]
Environment scope content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6
Path:
C:\LAB6\COM.COMs
The program has ended with the code: 115

```

Рис. 3 - Запуск модуля из другой директории

```

C:\>lb6.exe
Error: file could not be found

```

Рис. 4 - Запуск модуля из разных директорий

Контрольные вопросы

1. Как реализовано прерывание Ctrl C?

При нажатии комбинации клавиш Ctrl-C управление передаётся по адресу 0000:008Ch, который копируется в PSP функциями 26h и 4Ch и восстанавливается при выходе из программы.

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

В точке вызова функции 4Ch прерывания 21h.

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

В точке вызова функции 01h прерывания 21h.

Вывод

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