

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. 8381

Звегинцева Е.Н.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Исследование возможности построения загрузочного модуля динамической структуры. Исследование интерфейса между вызывающим и вызываемым модулями по управлению и по данным.

Описание функций и структур данных.

Таблица 1 – функции управляющей программы.

Название функции	Назначение
PRINT	Печатает строку на экран.
FREE_MEM	Освобождение неиспользуемой приложением памяти.
SET_ENV	Запись местоположения вызываемого модуля.
LOAD	Загрузка вызываемого модуля.
MAIN	Основная функция программы.

Ход работы.

Сборка и отладка модулей производилась с помощью компилятора MASM и отладчика AFDPRO.EXE в эмуляторе DOSBox.

Был написан программный .EXE модуль, который выполняет следующие функции:

1. Подготавливает параметры для запуска загрузочного модуля из того же каталога, в котором он находится сам. Вызываемому модулю передаёт новую среду и новую командную строку.
2. Вызываемый модуль запускается с использованием загрузчика.
3. После запуска проверяется выполнение загрузчика, а затем результат выполнения вызываемой программы.

В качестве вызываемого модуля был взят модифицированный модуль из ЛР2.

Была запущена программа, когда оба модуля находятся в текущем каталоге и введен символ «u». Результат работы программы представлен на рис. 1.

```
E:\>lb6
Memory was freed successfully
Inaccessible memory address: 9FFF
Program environment address: 0203
Command line tail:
Program environment content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6

Program environment content ended
Path:
E:\LB2.COM
u
Button: u
The program ended with
Normal end
```

Рисунок 1 – результат работы программы, оба модуля в текущем каталоге.

Была запущена программу, когда оба модуля находятся в текущем каталоге и введена комбинация клавиш “Ctrl+C”. Результат работы программы представлен на рис. 2.

```
E:\>lb6
Memory was freed successfully
Inaccessible memory address: 9FFF
Program environment address: 0203
Command line tail:
Program environment content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6

Program environment content ended
Path:
E:\LB2.COM
♥
Button: ♥
The program ended with
Normal end
```

Рисунок 2 – результат работы программы, оба модуля в текущем каталоге, ввод комбинации клавиш “Ctrl+C”.

Была запущена программа, когда оба модуля находятся не в текущем каталоге, и введен символ «n». Результат работы программы представлен на рис. 3.

```

E:\>TMP\lb6
Memory was freed successfully
Inaccessible memory address: 9FFF
Program environment address: 0203
Command line tail:
Program environment content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6

Program environment content ended
Path:
E:\TMP\LB2.COM
n
Button: n
The program ended with
Normal end

```

Рисунок 3 – результат работы программы, оба модуля не в текущем каталоге, ввод символа «n».

Была запущена программа, когда оба модуля находятся не в текущем каталоге и введена комбинация клавиш “Ctrl+C”. Результат работы программы представлен на рис. 4.

```

E:\>TMP\lb6
Memory was freed successfully
Inaccessible memory address: 9FFF
Program environment address: 0203
Command line tail:
Program environment content:
PATH=Z:\
COMSPEC=Z:\COMMAND.COM
BLASTER=A220 I7 D1 H5 T6

Program environment content ended
Path:
E:\TMP\LB2.COM
♥
Button: ♥
The program ended with
Normal end

```

Рисунок 4 – результат работы программы, оба модуля не в текущем каталоге, ввод комбинации клавиш “Ctrl+C”.

Была запущена программа, когда модули находятся в разных каталогах. Результат работы программы представлен на рис. 5.

```
E:\>TMP\1b6
Memory was freed successfully
Program was loaded
Z: File not found
```

Рисунок 5 – результат работы программы, модули находятся в разных каталогах.

Выводы.

В процессе выполнения данной лабораторной работы была исследована возможность построения загрузочного модуля динамической структуры.

Как видно из примеров, комбинация ctrl+c работает некорректно в dosbox и вызывает не прерывание, а символ «♥».

Ответы на контрольные вопросы.

1. Как реализовано прерывание Ctrl-C?

Ответ: При нажатии комбинации клавиш Ctrl-C происходит проверка наличия в буфере комбинации клавиш Ctrl-C с помощью функции 01h. При их обнаружении происходит вызов прерывания int 23h, завершающего программу с кодом 1.

В dosbox для ubuntu оно реализовано как замена на символ ♥.

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

Ответ: В точке вызова функции 4Ch прерывания int 21h.

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

Ответ: В точке вызова функции 01h прерывания int 21h.