**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Факультет компьютерных технологий и информатики**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №7**

**по дисциплине «Операционные системы»**

**Тема: Построение модуля оверлейной структуры.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6381 |  | Федянин Н.И. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

## **Цель работы**

Целью работы является исследование возможности построения загрузочного модуля оверлейной структуры.

## **Структура данных управляющей программы**

Таблица 1 – Назначение переменных

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| err\_4Ah\_code\_7 | Хранение сведений о том, что функция 4Ah не может быть выполнена по причине разрушения управляющего блока памяти. |
| err\_4Ah\_code\_8 | Хранение сведений о том, что функция 4Ah не может быть выполнена по причине нехватки памяти для выполнения функции. |
| err\_4Ah\_code\_9 | Хранение сведений о том, что функция 4Ah не может быть выполнена по причине неверного адреса блока памяти. |
| called\_program\_err\_code\_1 | Хранение сведений о том, что вызываемая программа не была загружена по причине того, что функция не существует. |
| called\_program\_err\_code\_2 | Хранение сведений о том, что вызываемая программа не была загружена по причине того, что файл не был найден. |
| called\_program\_err\_code\_3 | Хранение сведений о том, что вызываемая программа не была загружена по причине того, что маршрут не был найден. |
| called\_program\_err\_code\_4 | Хранение сведений о том, что вызываемая программа не была загружена по причине того, что слишком много открытых файлов. |
| called\_program\_err\_code\_5 | Хранение сведений о том, что вызываемая программа не была загружена по причине отсутствия доступа. |
| called\_program\_err\_code\_8 | Хранение сведений о том, что вызываемая программа не была загружена по причине нехватки памяти. |
| called\_program\_err\_code\_10 | Хранение сведений о том, что вызываемая программа не была загружена по причине неправильной среды. |
| err\_4Eh\_code\_2 | Хранение сведений о том, что вызываемая программа не была загружена при попытке сосчитать размер файла оверлея по причине того, что файл не был найден. |
| err\_4Eh\_code\_3 | Хранение сведений о том, что вызываемая программа не была загружена при попытке сосчитать размер файла оверлея по причине того, что маршрут не был найден. |
| OVERLAY\_ADDR | Хранение сведений об адресе, по которому производится вызов оверлея. |
| DTA | Буфер DTA. |
| KEEP\_PSP | Для хранения PSP. |
| OVERLAY\_ADDRESS | Блок параметров. |
| DTA\_prgh | Путь до оверлея. |
| OVL1\_NM | Для хранения имени первого оверлея. |
| OVL2\_NM | Для хранения имени второго оверлея. |

## **Функции управляющей программы**

Таблица 2 – Назначение функций

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| PRINT\_A\_STR | Вывод строки на экран |
| DEF\_4Ah\_ERROR | Поиск ошибки в случае невозможности выполнения 4Ah |
| DEF\_PROGRAMLAUNCH\_ERROR | Поиск ошибки в случае, если вызываемая программа не была загружена |
| PathSearch | Нахождение пути до вызываемого файла |
| OverlayFileSize | Определение размера оверлея |
| OvlProcess | Вызов оверлейной программы |

## 

## **Последовательность действий, выполняемых утилитой**

* Освобождение памяти для загрузки оверлеев.
* Чтение размера файла оверлея и запрос объема памяти, достаточного для его загрузки.
* Загрузка и выполнение файла оверлейного сегмента.
* Освобождение памяти, отведенной для оверлейного сегмента.
* Предыдущие действия выполняются и для второго оверлейного сегмента.
* Оверлейные сегменты выводят сообщение о загрузке и адрес сегмента, в который они загружены.

**Ход работы**

1. Написан и отлажен **.EXE** модуль. В нем вызываются сначала первый оверлейный сегмент, затем второй. Отлаженная программа запущена в каталоге, где находятся разработанные модули. Результаты выполнения программы:

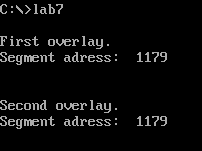


Рисунок 1 – Результаты выполнения программы

1. Далее программа вызывается из другого каталога. Результаты:

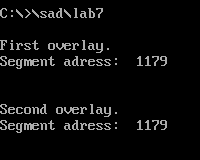


Рисунок 2 – Результаты выполнения программы

1. Теперь уберем первый оверлей из каталога. Результаты:

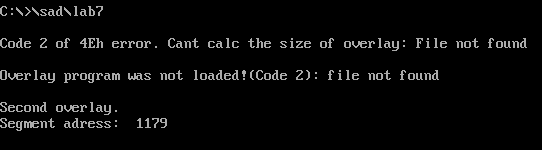


Рисунок 3 – Результаты выполнения программы

Ответы на контрольные вопросы:

*Как должна быть устроена программа, если в качестве оверлейного сегмента использовать .COM модули?*

COM-файлы имеют структуру, где код располагается с адреса 100h, а пространство от начала файла до 100h зарезервировано под PSP. Поэтому надо переписать содержимое PSP, а при обращении к оверлейному сегменту необходимо будет обращаться к сегменту, смещенному на 100h, чтобы правильно определить размер файла оверлея. Также необходимо выделить 64 Кб памяти, как это делает стандартный загрузчик для COM модулей, для корректной установки регистров стека.

# **ЗАКЛЮЧЕНИЕ**

В ходе данной лабораторной работы была исследована возможность построения загрузочного модуля оверлейной структуры. Исследованы структура оверлейного сегмента и способ загрузки и выполнения оверлейных сегментов.