**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №7**

**по дисциплине «Операционные системы»**

Тема: **Построение модуля оверлейной структуры**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6383 |  | Гомонова А.А. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

1. **Цель работы.**

Исследование возможности построения загрузочного модуля оверлейной структуры. Исследуется структура оверлейного сегмента и способ загрузки и выполнения оверлейных сегментов. Для запуска вызываемого оверлейного модуля используется функция 4B03h прерывания int 21h. Все загрузочные и оверлейные модули находятся в одном каталоге.

1. **Постановка задачи.**

**Шаг 1.** Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .ЕХЕ, который выполняет функции:

1) Освобождает память для загрузки оверлеев.

2) Читает размер файла оверлея и запрашивает объем памяти, достаточный для его загрузки.

3) Файл оверлейного сегмента загружается и выполняется.

4) Освобождается память, отведенная для оверлейного сегмента.

5) Затем действия 1-4 выполняются для следующего оверлейного сегмента.

**Шаг 2.** Также необходимо написать и отладить оверлейные сегменты. Оверлейный сегмент выводит адрес сегмента, в который он загружен.

**Шаг 3.** Запустите отлаженную программу. Оверлейные сегменты должны загружаться с одного и того же адреса, перекрывая друг друга.

**Шаг 4.** Запустите приложение из другого каталога. Приложение должно быть выполнено успешно.

**Шаг 5.** Запустите приложение, когда одного оверлея нет в каталоге. Приложение должно закончиться аварийно.

**Шаг 6.** Занесите полученные результаты в виде скриншотов в отчёт. Оформите отчёт в соответствии с требованиями

1. **Сведения о функциях и структурах данных управляющей программы**

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| Write\_message | Вывод сообщения на экран |
| Main | Основная функция |
| MemorySize | Функция, определяющая размер оверлея при помощи функции 4Eh прерывания 21h |
| Get\_Path | Функция нахождения пути до вызываемого файла (в bp - имя файла) |
| Call\_OVL | Функция вызова оверлейной программы при помощи 4B03h прерывания 21h |

1. **Ход работы.**

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| error\_4Ah | Сообщение об ошибке в 4Ah |
| error4Ah\_7 | Сообщение об ошибке с кодом 7 |
| error4Ah\_8 | Сообщение об ошибке с кодом 8 |
| error4Ah\_9 | Сообщение об ошибке с кодом |
| error\_4B03h | Сообщение об ошибке в 4B03h |
| error\_4B03h\_1 | Сообщение об ошибке с кодом 1 |
| error\_4B03h\_2 | Сообщение об ошибке с кодом 2 |
| error\_4B03h\_3 | Сообщение об ошибке с кодом 3 |
| error\_4B03h\_4 | Сообщение об ошибке с кодом 4 |
| error\_4B03h\_5 | Сообщение об ошибке с кодом 5 |
| error\_4B03h\_8 | Сообщение об ошибке с кодом 8 |
| error\_4B03h\_10 | Сообщение об ошибке с кодом 10 |
| memory\_error\_message | Сообщение об ошибки по причине нехватки памяти для загрузки файла |
| error\_4Eh | Сообщение об ошибке в 4Eh |
| error\_4Eh\_2 | Сообщение об ошибке с кодом 2 |
| error\_4Eh\_3 | Сообщение об ошибке с кодом 3 |
| adr | Адрес, по которому производится вызов оверлея |
| DTA | Буфер DTA |
| Keep\_PSP | Переменная для хранения PSP |
| OVL\_address | Блок параметров |
| DTA\_paraghr | Путь к оверлею |
| name1 | Имя 1-ого файла |
| name2 | Имя 2-ого файла |

**Шаг 1.**

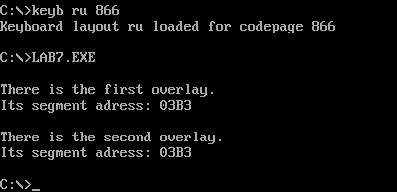
Был написан и отлажен программный модуль типа EXE, который выполняет функции:

* 1. Освобождает память для загрузки оверлеев
  2. Читает размер файла оверлея и запрашивает объём памяти, достаточный для его загрузки.
  3. Файл оверлейного сегмента загружается и выполняется.
  4. Освобождается память, отведённая для оверлейного сегмента
  5. Действия 1-4 выполняются до следующего оверлейного сегмента.

**Шаг 2.**

Также были написаны и отлажены оверлейные сегменты. Оверлейный сегмент выводит адрес сегмента, в котором он загружен.

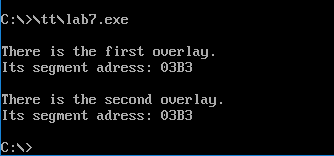
1. **Тестирование**
2. Запуск отлаженного приложения. Оверлейные сегменты загружаются с одного и того же адреса, перекрывая друг друга. (Рис.1)



**Рисунок 1**

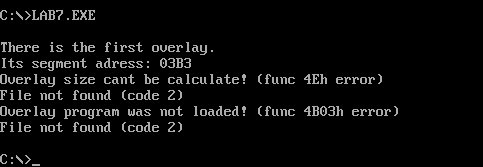
1. Запуск приложения из другого каталога. Приложение выполнено успешно. (Рис.2)

Результат:



**Рисунок 2**

1. Запуск приложения в случае, когда одного оверлея нет в каталоге. Приложение завершено аварийно. (Рис.3)

****

**Рисунок 3**

1. **Ответы на контрольные вопросы**
2. *Как должна быть устроена программа, если в качестве оверлейного сегмента использовать .COM модули?*

Так как в COM модуле присутствует PSP, то нужно обращаться к оверлейному сегменту со смещением на 100h.

1. **Вывод**

В ходе данной лабораторной работы была исследована возможность построение загрузочного модуля оверлейной структуры, а также изучена структура оверлейного сегмента и способ загрузки и выполнения оверлейных сегментов.