**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Математического Обеспечения ЭВМ**

отчет

**по лабораторной работе №7**

**по дисциплине «Операционные системы»**

Тема: **Построение модуля оверлейной структуры.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6383 |  | Базаров И. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

# 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 4В03h прерывания int 21h. Все загрузочные и оверлейные модули находятся в одном каталоге.

В этой работе также рассматривается приложение, состоящее из нескольких модулей, поэтому все модули помещаются в один каталог и вызываются с использованием полного пути.

# Интерфейс функций и структур управляющей программы

Функции:

ERROR\_4Ah – печатает на экран сообщение об ошибке освобождения памяти;

ERROR\_4Eh – печатает на экран сообщение об ошибке, возникшей при поиске файла с оверлеем;

ERROR\_ALLOC\_MEM – пишет сообщение об ошибке выделения памяти;

ERROR\_LOAD – пишет сообщение об ошибке, возникшей при загрузке оверлея;

ERROR\_DEALLOC\_MEM – печатает сообщение об ошибке освобождения памяти из-под оверлея;

OVERLAY – устанавливает, запускает и освобождает память для оверлея (имя модуля берётся по смещению SI, длина имени CX).

Структуры:

KEEP\_PSP – переменная, сохраняющая адрес PSP;

OV\_1 – имя запускаемого оверлея №1;

OV\_2 – имя запускаемого оверлея №2;

\_4Ah\_7 – сообщение «Free memory Error: Memory control unit destroyed»;

\_4Ah\_8 – сообщение «Free memory Error: Not enough memory to perform the function»;

\_4Ah\_9 – сообщение «Free memory Error: Wrong address of the memory block»;

\_4Eh\_2 – сообщение «Error determining size: File not found»;

\_4Eh\_3 – сообщение «Error determining size: Route not found»;

\_ER\_ALLOC\_MEM – сообщение «Error: Memory not allocated»;

\_ER\_DEALLOC\_MEM – сообщение «Error: Memory not deallocated»;

\_ER\_LOAD\_1 – сообщение «Launch Error: Non-existent function»;

\_ER\_LOAD\_2 – сообщение «Launch Error: File not found»;

\_ER\_LOAD\_3 – сообщение «Launch Error: Route not found»;

\_ER\_LOAD\_4 – сообщение «Launch Error: Too many open files»;

\_ER\_LOAD\_5 – сообщение «Launch Error: No access»;

\_ER\_LOAD\_8 – сообщение «Launch Error: Low memory»;

\_ER\_LOAD\_10 – сообщение «Launch Error: Incorrect environment»;

ParBlock – смещение и сегментный адрес запускаемого оверлея;

CMD\_NUM\_CHAR – длина командной строки;

CMD\_STR – командная строка.

# Последовательность действий, выполняемых утилитой

Программа освобождает память, оставляя столько, сколько она использует, и сохраняет адрес PSP. После этого вызывается функция OVERLAY, которая создает командную строку (с адресом файла оверлея), ищет по этой строке этот файл, определяет его размер, выделяет под него память, загружает оверлей в выделенную память и, подготовив «точку возврата» вызывает его выполнение; после возврата освобождает выделенную память. Повторяется вызов функции OVERLAY для 2-го файла.

В случае, если функция освобождения памяти, функция поиска файла, функция загрузки оверлея в память или функция освобождения выделенной памяти завершились с CF==1, вызывается соответствующая функция вывода сообщения об ошибке и программа завершает свою работу.

На рис. 1 показано поведение программы, запущенной из текущего каталога, оба оверлейных файла присутствуют. На рис. 2 изображено поведение программы, запущенной из каталога TCF, расположенного в текущем; оба оверлейных файла присутствуют. На рис. 3 показано поведение программы при условии, что отсутствует файл второго оверлея.



Рис. 1 – Текущий каталог; присутствуют оба оверлея.



Рис. 2 – Каталог TCF; присутствуют оба оверлея .



Рис. 3 – Каталог TCF; присутствует только первый оверлей.

# Результаты исследования

Отличительной особенностью .COM модуля является то, что адресация смещена на 100h, а значит при его использовании в качестве оверлейного сегмента (расположен с SEG:0000h – специально отведённой под него памяти) при обращении к каким-либо данным или функциям в данном оверлее мы будем промахиваться на эти самые 100h байт. Для того, чтобы этого избежать, необходимо вызывать оверлейный модуль по смещению 100h, скомпенсировав это уменьшением его сегментного адреса на 10h. Т.о. вместо перехода к SEG:0000h мы переходим к SEG-0010h:0100h.

# Вывод

В лабораторной работе было проведено исследование возможности построения загрузочного модуля оверлейной структуры. Была исследована структура оверлейного сегмента и способ загрузки и выполнения оверлейных сегментов. Для запуска вызываемого оверлейного модуля использовалась функция 4В03h прерывания int 21h. Все загрузочные и оверлейные модули находились в одном каталоге.

Также было разработано приложение, состоящее из нескольких модулей; все модули помещались в один каталог и вызывались с использованием полного пути.