

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №7**  
**по дисциплине «Операционные системы»**  
**Тема: Построение модуля оверлейной структуры**

Студент гр. 0381

Степанова Е. М.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

## **Цель работы.**

Исследование возможности построения загрузочного модуля оверлейной структуры. Исследуется структура оверлейного сегмента и способ загрузки и выполнения оверлейных сегментов.

## **Задание.**

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

**Шаг 1.** Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .EXE, 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 отлаженное приложение. 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 в случае, когда одного оверлея нет в каталоге. Приложение должно закончиться аварийно.

**Шаг 6.** Занесите полученные результаты в виде скриншотов в отчет.

## **Выполнение работы.**

Для загрузки оверлейных модулей необходимо освободить память, так как программа после загрузки занимает всю доступную память. Для этого была написана функция освобождения памяти, которая по метке в конце программы определяет объем памяти, который необходимо оставить для работы программы. Полученный объем в байтах переводится в параграфы и передается функцию `4ah int 21h`, которая освобождает лишнюю память.

Далее с помощью функции `GET_PATH` конструируется путь до первого оверлейного модуля. `GET_PATH` берет из окружения программы полный путь до текущего исполняемого файла, отсекает имя файла и заменяет его на имя оверлейного модуля. Таким образом, предполагается, что оверлейный модуль находится в том же каталоге, что и основной файл.

По полученному имени файла в функции `GET_FILESIZE` определяется размер оверлейного модуля. Для этого используется функция `4e`, которая читает данные о файле. Размер нужен для того, чтобы выделить память под загрузку модуля в память. Выделение памяти осуществляет функцию `MALLOC`, которая использует функцию `48h`.

После того, как память выделена, вызывается функция `LOAD`, которая загружает оверлейный модуль в память и передает управление ему. Для загрузки используется функция `4b03h` прерывания `int 21h`. Эта функция принимает имя файла и адрес загрузки. Управление передается с помощью инструкции `call` по адресу, в который был загружен оверлейный модуль.

После выполнения оверлейного модуля память, выделенная под него, освобождается и процедура загрузки повторяется с построения пути к второму оверлейному модулю.

Оверлейный модуль представляет собой кодовый сегмент, в котором точка входа находится по адресу 0. Оверлейный модуль выводит на экран адрес, по которому он был загружен и номер модуля.

При запуске программы выводятся сообщения от каждого оверлейного модуля, причем загружаются они по одному и тому же адресу:

```
F:\>lab7.exe
Overlay 1 loaded, address: 0198h
Overlay 2 loaded, address: 0198h
F:\>_
```

При запуске из другого каталога программа выполняется успешно:

```
F:\DIR>lab7.exe
Overlay 1 loaded, address: 0198h
Overlay 2 loaded, address: 0198h
F:\DIR>_
```

Из каталога был удален первый оверлейный модуль, теперь загрузка первого модуля выдает ошибку, а второй модуль загружается успешно:

```
F:\DIR>lab7.exe
Error: File not found
Overlay 2 loaded, address: 0198h
F:\DIR>
```

## **Выводы.**

Была исследована возможность построения загрузочного модуля оверлейной структуры. Исследована структура оверлейного сегмента и способ загрузки и выполнения оверлейных сегментов.

## ВОПРОСЫ

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

Программа не потребует изменений, так как оверлейные модули состоят из одного сегмента, точка входа в который находится по адресу 0, как и в COM файле. При загрузке оверлейных модулей PSP не создается, поэтому дополнительное смещение адреса, по которому передается управление не требуется.