

**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**

Студент гр. 0381

Ибатов Н.Э.

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. Для запуска вызываемого оверлейного модуля используется функция 4B03h прерывания int 21h. Все загрузочные и оверлейные модули находятся в одном каталоге.

### **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. Оформите отчет в соответствии с требованиями.

## **Выполнение работы.**

Для того, чтобы загрузить оверлейные модули необходимо освободить память, так как программа после загрузки занимает всю доступную память. Для этого была написана функция по освобождению памяти, которая при помощи метки в конце программы определяет объем памяти, которая потребуется для выполнения программы. Далее полученный объем в байтах переводится в параграфы и передаётся функции `4ah int 21h`, которая освобождает неиспользованную память.

Далее при помощи функции `getPath` находится полный путь до оверлейного модуля, сначала она вычисляет полный путь до самого исполняемого файла, после чего отбрасывает его имя и подставляет на это место имя оверлейного модуля. Для этого оверлейный модуль, должен находиться в той же директории, что и исполняемый файл.

По полученному имени оверлейного файла, функция `fileSize` определяет размер данного оверлейного модуля, для этого используется функция `4eh int 21h`, которая предоставляет информацию о файле. Выделение памяти осуществляется функцией `malloc`, которая использует функцию `48h`.

После того, как была выделена память, вызывается функция `load`, которая загружает в данную область оверлейный модуль и передаёт ему управление. Для загрузки используется функция `4b03h int 21h`. Эта функция принимает имя файла и адрес загрузки. Управление передаётся вызовом `call` по адресу, в котором загружен оверлейный модуль.

После выполнения оверлейного модуля, память, выделенная под него, очищается и загружается другой оверлейный модуль.

Оверлейный модуль состоит из кодового сегмента, в котором точка входа находится по адресу `0`. Оверлейный модуль выводит на экран адрес, в который был загружен.

При запуске программы выводятся сообщения от оверлейных модулей, при этом загружаются они в один и тот же адрес.

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

При запуске программы из другой директории программа выполняется успешно.

```
F:\>cd lab7
F:\LAB7>lab7.exe
Overlay 1 address: 0198h
Overlay 2 address: 0198h
F:\LAB7>_
```

При запуске программы из другой директории и удаленным оверлейным модулем выводится ошибка о том, что файла нет, а второй загружается успешно.

```
F:\LAB7>lab7.exe
! : File doesn't exist
Overlay 2 address: 0198h
F:\LAB7>
```

### **Выводы.**

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

## **ПРИЛОЖЕНИЕ А**

### **ВОПРОСЫ**

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

Так как .com модуль также состоит из одного сегмента, точка входа которого находится по адресу 0, то программа не потребует изменений. При загрузке оверлейных модулей PSP не создается, поэтому дополнительное смещение адреса не требуется.