

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

**ЛАБОРАТОРНАЯ РАБОТА № 7 по**  
**дисциплине «Операционные системы»**  
**ТЕМА: Построение модуля оверлейной**  
**структуры.**

Студентка гр. 9381

Андрух И.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### **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.

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

### **Ход работы.**

1) Написать и отладить программный модуль типа .EXE, который выполняет функции:

- Проверяет, установлено ли пользовательское прерывание с вектором 09h.
- Если прерывание не установлено, то устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний. Адрес точки входа в стандартный обработчик прерывания находится в теле пользовательского обработчика. Осуществляется выход по функции 4Ch прерывания int21h.
- Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания int 21h.

2) Написать и отладить оверлейные сегменты, которые выводят адрес сегмента, в который они загружены.

3) Запустить отлаженное приложение. Оверлейные сегменты загружаются с одного и того же адреса, перекрывая друг друга.

4) Запустить приложение из другого каталога.

5) Запустить приложение в том случае, когда одного оверлея нет в каталоге.

6) Получившиеся результаты показать в виде скриншотов.

### Выполнение работы.

Отладим два оверлейных сегмента и саму программу. При выполнении программы выводится следующее:

```
C:\>lab7
Memory was freed successfully!
Allocation of memory was successfully!
Loaded successfully!

OVERL1 address:  020A

Allocation of memory was successfully!
Loaded successfully!

OVERL2 address:  020A
```

Запустим программу из другого каталога. Запуск программы также прошел успешно:

```
C:\TASK>lab7
Memory was freed successfully!
Allocation of memory was successfully!
Loaded successfully!

OVERL1 address:  020A

Allocation of memory was successfully!
Loaded successfully!

OVERL2 address:  020A

C:\TASK>_
```

Уберём modul2.ovl из каталога и запустим программу:

```
C:\TASK>lab7
Memory was freed successfully!
Allocation of memory was successfully!
Loaded successfully!

OVERL1 address: 020A

ALLOCATION MEMOTY ERROR: file not found
LOAD ERROR: file not found

C:\TASK>_
```

Уберем из каталога modul1.ovl и modul2.ovl и запустим программу:

```
C:\TASK>lab7
Memory was freed successfully!
ALLOCATION MEMOTY ERROR: file not found
LOAD ERROR: file not found

ALLOCATION MEMOTY ERROR: file not found
LOAD ERROR: file not found

C:\TASK>_
```

### Ответы на контрольные вопросы:

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

В COM-модуле необходимо поместить значение регистра CS в регистр DS после записи значений в стек (адрес сегмента данных = адрес сегмента кода). Также при обращении к оверлейному сегменту необходимо обращаться к сегменту, смещённому на 100h.

## **Выводы.**

В ходе данной работы были исследованы возможности построения загрузочного модуля оверлейной структуры. Рассмотрено приложение, состоящее из нескольких модулей. Реализовано приложение в возможностью запуска модуля оверлейной структуры из любого каталога.