

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра математического обеспечения и применения ЭВМ**

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

Студент гр.0382

Диденко Д. В.

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.

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

### **Постановка задачи.**

**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, отведённая 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 отчёт.  
Оформите отчёт в соответствии с требованиями.

### **Выполнение работы.**

Были написаны структуры данных:

- STR\_PATCH\_NAME –полный путь к загружаемому модулю;
- LAUNCH\_PARAMETERS – загрузочные параметры;
- LAUNCH\_ADDRESS – загрузочный адрес;
- DTA – буфер под загрузку функции.

Были написаны строки для хранения информации:

- STR\_OVL1\_NAME – имя 1 ovl модуля;
- STR\_OVL2\_NAME – имя 1 ovl модуля.

Были написаны строки для вывода информации:

- Для вывода ошибок высвобождения памяти:
  - STR\_ERROR\_FREE\_MEMORY;
  - STR\_MEMORY\_7;
  - STR\_MEMORY\_8;
  - STR\_MEMORY\_9;
- Для вывода ошибок запуска дочерней программы:
  - STR\_ERROR\_1;
  - STR\_ERROR\_2;
  - STR\_ERROR\_5;
  - STR\_ERROR\_8;
  - STR\_ERROR\_10;
  - STR\_ERROR\_11;

Были составлены функции (см. табл.1)

Таблица 1 – функции в программе

Процедура	Описание
BYTE_TO_DEC	Перевод байта в 10 систему счисления
WRITE_STRING	Вывод строки на экран
NEW_LINE	Перевод указателя на новую строку
FREE_MEMORY	Высвобождение не используемой памяти и обработка возможных ошибок
SET_FULL_FILE_NAME	Считывание пути до файла и построение полного пути
GET_OVERLAY_SIZE	Получение размера ovl модуля и проверка возможности загрузки
LOAD_OVERLAY	Загрузка ovl модуля и его запуск с обработкой ошибок
MACRO_CREATE_FULL_FILE_NAME	Универсальный макрос построения пути к файлу
MACRO_LOAD_OVERLAY	Универсальный макрос загрузки ovl модуля
MAIN	Главная функция

В результате выполнения были получены следующие значения(рис.1-3):

```

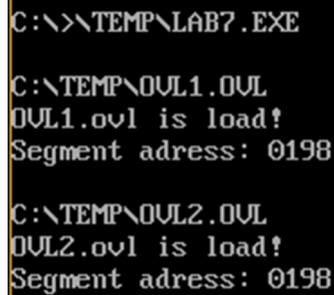
C:\>LAB7.EXE

C:\>OVL1.OVL
OVL1.ovl is load!
Segment adress: 0198

C:\>OVL2.OVL
OVL2.ovl is load!
Segment adress: 0198

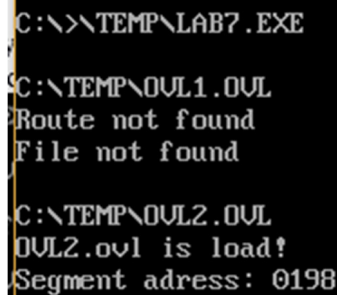
```

Рисунок 1 – Запуск отлаженной программы из каталога с разработанными модулями.



```
C:\>\TEMP\LAB7.EXE  
  
C:\TEMP\OUL1.OVL  
OUL1.ovl is load!  
Segment adress: 0198  
  
C:\TEMP\OUL2.OVL  
OUL2.ovl is load!  
Segment adress: 0198
```

Рисунок 2 – Запуск отлаженной программы из каталога не с разработанными модулями.



```
C:\>\TEMP\LAB7.EXE  
  
C:\TEMP\OUL1.OVL  
Route not found  
File not found  
  
C:\TEMP\OUL2.OVL  
OUL2.ovl is load!  
Segment adress: 0198
```

Рисунок 3 – Запуск отлаженной программы из каталога с разработанной программой, при отсутствие первого оверлея.

### **Выводы.**

В ходе лабораторной работы был построен загрузочный модуль оверлейной структуры, а также оверлеи. Изучены дополнительные функции работы с памятью и способы загрузки и выполнения оверлейных сегментов.

**ПРИЛОЖЕНИЕ А**  
**ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ**

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

Программа также имеет PSP, следовательно нужно обращаться к модулю со смещением в 100h. Также нужно сохранять регистры и восстанавливать их в конце работы программы.