

**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**

Студентка гр. 0381

\_\_\_\_\_

Короткина Е.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Исследование возможности построения загрузочного модуля оверлейной структуры.

### **Постановка задачи.**

1. Написать и отладить программный модуль типа .EXE, который выполняет следующие 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

2. Написать и отладить оверлейные сегменты. Оверлейный сегмент выводит адрес сегмента, в который он загружен.

3. Запустите отлаженное приложение. Оверлейные сегменты должны загружаться с одного и того же адреса, перекрывая друг друга.

4. Запустите приложение из другого каталога. Приложение должно быть выполнено успешно.

5. Запустите приложение в случае, когда одного оверлея нет в каталоге. Приложение должно закончиться аварийно.

### **Выполнение работы.**

1. Для вывода сообщений написана процедура WRITEMESSAGE.

Созданы переменные с сообщениями различных ошибок и результатов завершения программы, переменная с названиями оверлейных модулей OVERLAY1\_NAME и OVERLAY2\_NAME, переменная, в которую будет занесен путь к файлу PATH\_TO\_FILE, буфер DTA\_BUFFER на 43 байта.

Для подготовки блока параметров написана процедура GET\_PARAMS. В этой же процедуре составляется правильный путь к файлу PATH\_TO\_FILE.

Для освобождения места в памяти функцией 4Ah прерывания int 21h написана процедура SPREAD\_MEMORY. В случае, если при выделении памяти произошла ошибка, выводится соответствующее сообщение.

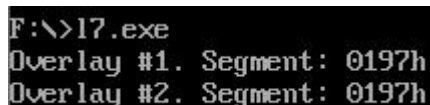
Процедура GET\_FILESIZE определяет размер файла оверлея, а процедура MEMORY\_FOR\_OVERLAY запрашивает необходимую под него память. Если определить размер файла или выделить память под него не удалось, выводится сообщение об ошибке.

В процедуре LOAD\_OVERLAY выполняет загрузку оверлея, а после его выполнения освобождает отведенную под него память. Если загрузка не удалась, по коду, возвращенному функцией 4B03h прерывания 21h, определяется причина неудавшейся загрузки и выводится сообщения об ошибке.

В процедуре Main для каждого из оверлеев последовательно вызываются SPREAD\_MEMORY, GET\_PARAMS, GET\_FILESIZE, MEMORY\_FOR\_OVERLAY, LOAD\_OVERLAY, после вызова каждой функции выполняется проверка значения BX, в который при успешном выполнении функции занесен 0, иначе - 1. Если значение BX на каком-либо этапе оказалось не равно 0, то одна из процедур не была выполнена успешно. В таком случае либо происходит переход к загрузке следующего оверлея (если ошибка возникла с первым оверлеем) или к концу программы.

2. Написаны оверлейные сегменты. Для вывода сообщений написана процедура WRITEMESSAGE, с помощью которой выводится строка, сообщающая адреса сегмента, в который был загружен оверлей.

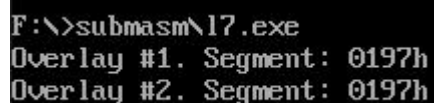
3. Программа запущено. Как видно из рис.1., оверлейные модули загружаются с одного адреса и перекрывают друг друга.



```
F:\>I7.exe
Overlay #1. Segment: 0197h
Overlay #2. Segment: 0197h
```

Рис.1. Результат работы программы для п.3.

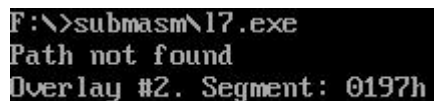
4. Создан подкаталог submasm, программа запущена из него. Программа работает, вывод аналогичен п.3.



```
F:\>submasm\l7.exe
Overlay #1. Segment: 0197h
Overlay #2. Segment: 0197h
```

Рис.2. Результат работы программы для п.4.

5. Из подкаталога удален первый оверлей. Программа выводит сообщение о том, что он не найден, но для второго оверлея по-прежнему работает успешно.



```
F:\>submasm\l7.exe
Path not found
Overlay #2. Segment: 0197h
```

Рис.3. Результат работы программы для п.5.

### **Ответы на контрольные вопросы.**

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

Основная программа должна быть устроена аналогично - в .COM модуле также только один сегмент, как и в оверлее. В коде модуля типа .COM задавать смещение директивой ORG 100h не нужно, т.к. при передаче управления оверлею PSP не создается.

### **Вывод.**

Была исследована возможность построения загрузочного модуля оверлейной структуры.