

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.0382

Рубежова Н.А.

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 оверлейные модули находятся в одном каталоге.

Задание.

Шаг 1. Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .EXE, который выполняет функции:

- а) Освобождает память для загрузки оверлеев;
- б) Читает размер файла оверлея и запрашивает объем памяти, достаточный для его загрузки;
- в) Файл оверлейного сегмента загружается и выполняется;
- г) Освобождается память, отведенная для оверлейного сегмента;
- д) Затем действия а) - г) 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.

Шаг 1-2. Были разработаны программные модули типа .EXE:

lab7.asm - основной модуль, сначала подготавливающий память и всё необходимое для запуска оверлеев, а затем загружающий сами оверлеи.

overlay1.asm и overlay2.asm – оверлейные сегменты, выводящие адрес сегмента, в который оверлей загружен.

В модуле lab7.asm использованы следующие процедуры:

- MEM_FREE – освобождает неиспользуемую основной программой память через 4Ah прерывания int 21h для последующей загрузки оверлеев, в случае неудачи – выводит сообщение об ошибке;

- GET_PATH – записывает строку – полный путь к загружаемому оверлею, используя переменные среды и добавляя к нему имя загружаемого оверлея

- BEFORE_LOADING – освобождает память перед запуском оверлея через 1Ah прерывания int 21h, получает размер загружаемого оверлея через 4Eh прерывания int 21h, если буфер DTA в 43 байта не был заполнен, значит файл оверлея или маршрут к нему не был найден и выводится соответствующее сообщение об ошибке.

- LOADING_OVL – загружает оверлейный сегмент и запускает его, используя функцию 4B03h прерывания int 21h. После выполнения оверлея память освобождается через функцию 49h прерывания int 21h.

- PRINT – процедура печати строки через 09h прерывания int 21h для вывода информативных сообщений

Шаг 3. Запустим программу в каталоге с разработанными модулями.

См. Рисунок 1.



```
C:\>lab7
C:\>OVERLAY1.OVL Segment address of overlay1.ovl: 118Eh
C:\>OVERLAY2.OVL Segment address of overlay2.ovl: 118Eh
```

Рисунок 1 – Результат запуска программы из каталога с разработанными модулями

Видим, что оверлеи запускались друг за другом, причем сегментный адрес один и тот же, то есть оверлеи загружались с одного и того же адреса, перекрывая друг друга.

Шаг 4. Запустим программу из другого каталога. См. Рисунок 2.

```
C:\LAB7>lab7
C:\LAB7\OVERLAY1.OVL Segment address of overlay1.ovl: 118Eh
C:\LAB7\OVERLAY2.OVL Segment address of overlay2.ovl: 118Eh
```

Рисунок 2 – Результат выполнения программы из каталога, отличного от того, где лежат разработанные модули

Видим, что вывод программы аналогичен выводу в шаге 3. Следовательно, выполнение программы не зависит от выбора текущего каталога.

Шаг 5. Запустим программу из каталога, в котором одного оверлея нет. См. Рисунок 3.

```
C:\>lab7
C:\OVERLAY1.OVL[ERROR]: The overlay file or route to him could not be found.
[ERROR]: Failed to load overlay.
C:\OVERLAY2.OVL Segment address of overlay2.ovl: 118Eh
```

Рисунок 3 – Результат выполнения программы из каталога, в котором отсутствует первый оверлейный сегмент

Видим, что первый оверлей был завершен аварийно, вывелись сообщения об ошибках, поскольку файл с оверлеем не был найден. Второй оверлей исправно запустился, поскольку присутствует в каталоге.

Ответы на вопросы.

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

Ответ: Поскольку оверлейный сегмент оформляется в ассемблере как функция с точкой входа по адресу 0, а в .COM модуле есть блок PSP в памяти, который занимает 256(100h) байт, то нужно учесть это смещение 100h.

Выводы.

В результате работы были исследованы возможности построения загрузочного модуля оверлейной структуры, а также изучена сама структура оверлейного сегмента, способ загрузки и выполнения оверлейных сегментов.