

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

Ефимов Н.Д.

Преподаватель

Губкин А.Ф.

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 с использованием полного пути.

Задание.

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.

Были написаны следующие структуры данных:

STR_PATCH_NAME – полный путь к загружаемому модулю,

LAUNCH_PARAMETERS – загрузочные параметры, LAUNCH_ADDRESS – загрузочный адрес, DTA – буфер под загрузку функции,

Также были созданы строки для хранения информации:

STR_OVL1_NAME – имя 1 ovl модуля, STR_OVL2_NAME – имя 2 ovl модуля.

Затем были написаны строки для вывода информации:

Строки STR_ERROR_FREE_MEMORY; STR_MEMORY_7; STR_MEMORY_8; STR_MEMORY_9 были созданы для оповещения об ошибке высвобождения памяти, строки STR_ERROR_1; STR_ERROR_2; STR_ERROR_5; STR_ERROR_8; STR_ERROR_10; STR_ERROR_11 – для оповещения об ошибке запуска дочерней программы (оверлейного модуля).

Дальше были созданы следующие процедуры:

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 - головная функция.

Результаты работы программы представлены на следующих скриншотах:

```
F:\>lab7  
  
F:\OVL1.OVL  
OVL1.ovl is loaded!  
Segment address: 0197  
  
F:\OVL2.OVL  
OVL2.ovl is loaded!  
Segment address: 0197
```

А на этом скриншоте показан случай, когда одного из файлов нет в текущем каталоге.

```
F:\OVL1.OVL  
OVL1.ovl is load!  
Segment address: 0197  
  
F:\OVL2.OVL  
Route not found  
File not found
```

Выводы.

Были изучены оверлеи, многомодульная структура, а так же применены полученные знания на практике.

ПРИЛОЖЕНИЕ А

ОТВЕТЫ НА ВОПРОСЫ

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

При обращении к оверлейному сегменту необходимо обращаться к сегменту, смещённому на 100h.