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. 7381 _____ Ильясов А.В.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2019

Цель работы

Исследование возможности построение 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 загруженные 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.

Необходимые сведения для составления программы

Для организации программы, имеющей оверлейную структуру, используется функция 4B03h прерывания 21h. Эта функция позволяет в отведённую область памяти, начинающуюся с адреса сегмента, загрузить программу, находящуюся в файле на диске. Передача управления загруженной программе этой функцией не осуществляется и префикс сегмента программы (PSP) не создаётся.

Если флаг переноса $CF = 1$ после выполнения функции, то произошли ошибки и регистр AX содержит код ошибки. Значение регистра AX характеризует следующие ситуации:

1 – несуществующая функция;

2 – файл не найден;

3 – маршрут не найден;

4 – слишком много открытых файлов;

5 – нет доступа;

8 – мало памяти;

10 – неправильная среда.

Если флаг переноса $CF = 0$, то оверлей загружен в память.

Перед загрузкой оверлея вызывающая программа должна освободить память по функции 4Ah прерывания 21h. Затем определить размер оверлея. Это можно сделать с помощью функции 4Eh прерывания 21h. Перед обращением к функции необходимо определить область памяти размером в 43 байта под буфер DTA, которую функция заполнит, если файл будет найден. Функция использует следующие параметры: CX – значение байта атрибутов, которое для файла имеет значение 0; $DS{:}DX$ – указатель на путь к файлу, который записывается в формате строки ASCIIZ.

Если флаг переноса $CF = 1$ после выполнения функции, то произошли ошибки и регистр AX содержит код ошибки. Значение регистра AX характеризует ситуации:

2 – файл не найден;

3 – маршрут не найден.

Если $CF = 0$, то в области памяти буфера DTA со смещением 1Ah будет находиться младшее слово размера файла, а в слове со смещением 1Ch – старшее слово размера памяти в байтах.

Полученный размер файла следует перевести в параграфы, причём следует взять большое целое числа параграфов. Затем необходимо отвести память с помощью функции 48h прерывания 21h. После этого необходимо сформировать параметры для функции 4B03h и выполнить её.

После отработки оверлея необходимо освободить память с помощью функции 49h прерывания 21h.

Оверлейный сегмент не является загрузочным модулем типов .COM или .EXE. Он представляет собой кодовый сегмент, который оформляется в ассемблере как функция с точкой входа по адресу 0 и возврат осуществляется командой retf. Это необходимо сделать, потому что возврат управления должен быть осуществлён в программу, выполняющую оверлейный сегмент.Если использовать функции выхода 4Ch прерывания 21h, то программа закончит свою работу

Ход работы

1) Напишем и отладим программный модуль типа **.EXE**, который выполняет функции, указанные в разделе «Алгоритм работы программы» данной лабораторной работы. Напишем и отладим оверлейные сегменты. Оверлейный сегмент выводит адрес сегмента, в который он загружен. Запустим программу lab7.exe:

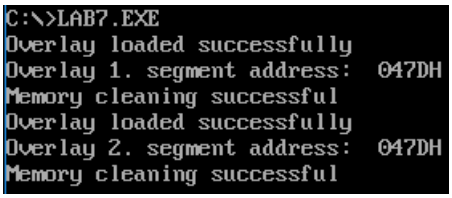

Рис.1. Результат работы программы lab7.exe

На Рисунке 1 видно, что оверлейные сегменты были загружены с одного и того же адреса.

2) Запустим приложение из другого каталога:

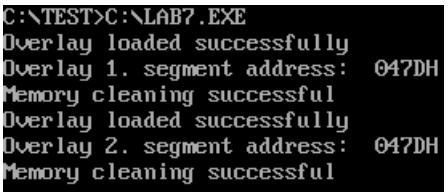

Рис.2. Результат запуска программы lab7.exe из другого каталога

Приложение было выполнено успешно.

3) Запустим приложение в случае, когда одного оверлея нет в каталоге:

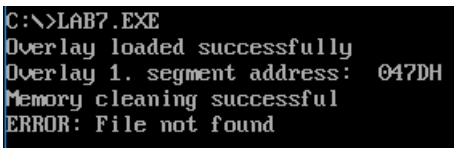

Рис.3. Результат запуска программы lab7.exe, когда в каталоге нет второго оверлея

Приложение закончилось аварийно, так как второй оверлей не был найден.

Выводы

В ходе выполнения данной лабораторной работы была исследована возможность построения загрузочного модуля оверлейной структуры, исследована структура оверлейного сегмента и способ загрузки и выполнения оверлейных сегментов. Было создано приложение, состоящее из нескольких модулей, все модули которого помещаются в один каталог и вызываются с использованием полного пути.

Ответы на контрольные вопросы.

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

Ответ: При использовании в качестве оверлейного сегмента .COM модуля, необходимо вызывать его по смещению 100h, так как в .COM файлах код располагается с адреса 100h.