

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

Студентка гр. 0381

Шиняева А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы

Исследовать возможность построения загрузочного модуля оверлейной структуры, структуру оверлейного сегмента и способ загрузки и выполнения оверлейных сегментов.

Постановка задачи

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

1. освобождает память для загрузки оверлеев;
2. читает размер файла оверлея и запрашивает объём памяти, достаточный для его загрузки;
3. загружает и выполняет оверлейный сегмент;
4. освобождает память, отведённую для оверлейного сегмента;
5. выполняет аналогичные действия для следующего оверлейного сегмента.

Также необходимо написать и отладить оверлейные сегменты. Оверлейный сегмент выводит адрес сегмента, в который он загружен.

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

Запустить приложение из другого каталога. Приложение должно быть выполнено успешно.

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

Выполнение работы

Для выполнения данной работы были реализованы следующие функции:

- * `println` для вывода строки на экран;
- * `num2dec` для записи в строку числа, лежащего в регистре AL;
- * `free` для очистки неиспользуемой модулем памяти;
- * `setp` для создания блока параметров;

- * `getpath` для получения пути вызываемого модуля;
- * `callovl` для вызова оверлейного модуля;
- * `main` для выполнения поставленной в данной лабораторной работе задачи.

Для вывода информации на экран были созданы следующие строки:

- * `unknown`, хранящая в себе строку 'Unknown error\$';
- * `ferr7`, хранящая в себе строку 'Error: memory control block destroyed\$';
- * `ferr8`, хранящая в себе строку 'Error: not enough memory to execute the function\$';
- * `ferr9`, хранящая в себе строку 'Error: invalid memory block address\$';
- * `serr2`, хранящая в себе строку 'Error: file could not be find\$';
- * `serr3`, хранящая в себе строку 'Error: path could not be find\$';
- * `cerr1`, хранящая в себе строку 'Error: function is not exists\$';
- * `cerr2`, хранящая в себе строку 'Error: file could not be found\$';
- * `cerr3`, хранящая в себе строку 'Error: path could not be found\$';
- * `cerr4`, хранящая в себе строку 'Error: too many open files\$';
- * `cerr5`, хранящая в себе строку 'Error: no access\$';
- * `cerr8`, хранящая в себе строку 'Error: no memory\$';
- * `cerrA`, хранящая в себе строку 'Error: wrong environment string\$'.

Выполнение программы начинается с вызова функции `free` для освобождения неиспользуемой программой памяти. Для этого с помощью функции `4Ah` прерывания `21h` выделяется необходимое программе количество памяти. В случае, если при этом возникла какая-либо ошибка, программа выводит соответствующее сообщение об ошибке и завершается.

После этого вызывается функция `setp`, которая заполняет выделенный в сегменте данных блок параметров.

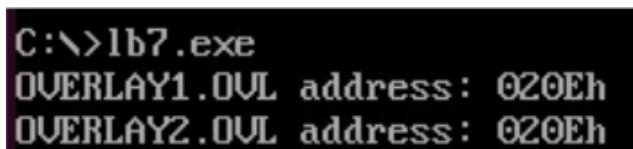
С помощью функции `getpath` считывается в строку путь до текущего файла, после чего имя файла меняется на имя вызываемых оверлейных модулей.

Функция `callovl` заполняет выделенный в памяти буфер DTA, выделяет память под оверлейный модуль, сохраняет сегментные регистры DS и ES, а также регистры, отвечающие за стек. Устанавливаются регистры ES:BX так, чтобы они указывали на блок параметров, после чего вызывается программный модуль.

После выполнения модуля регистры возвращаются, проверяется работа модуля.

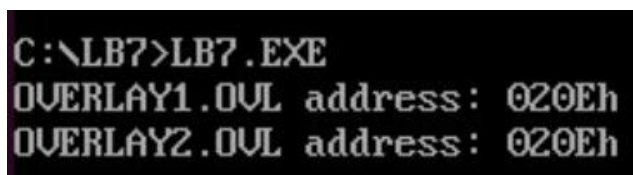
Разработанный программный код см. в приложении А.

Результаты работы программы представлены на рисунках 1-4.



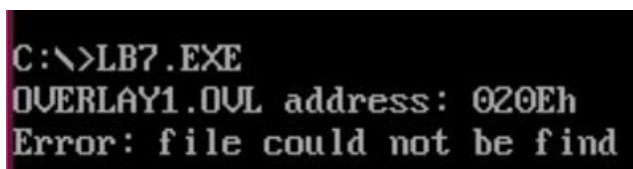
```
C:\>lb7.exe
OVERLAY1.OVL address: 020Eh
OVERLAY2.OVL address: 020Eh
```

Рис. 1 – Запуск приложения



```
C:\LB7>LB7.EXE
OVERLAY1.OVL address: 020Eh
OVERLAY2.OVL address: 020Eh
```

Рис. 2 – Запуск приложения из
другого каталога



```
C:\>LB7.EXE
OVERLAY1.OVL address: 020Eh
Error: file could not be find
```

Рис. 3 - Запуск приложения при
отсутствии overlay2.oül

Контрольные вопросы

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

При обращении к оверлейному .COM модулю необходимо обратиться к сегменту, смещённому на 100h относительно начала модуля, ввиду наличия PSP.

Вывод

В ходе работы были исследованы возможность построения загрузочного модуля оверлейной структуры, структура оверлейного сегмента и способ dosbox загрузки и выполнения оверлейных сегментов.