

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №7
по дисциплине «Операционные сети»
Тема: Построение модуля оверлейной структуры

Студентка гр. 0381

Сарычева А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Исследование возможности построения загрузочного модуля оверлейной структуры.

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.

В файле lab7.asm написан исходный код .EXE модуля. В начале данного модуля прописаны строки для вывода запрашиваемой информации, для вывода которых была создана процедура OUTPUT.

Написана процедура CALL_PATH, которая сохраняет полный путь до загрузочного модуля и подготавливает параметры для его загрузки.

Процедура CALL_FREE_MEMORY освобождает место в памяти с помощью функции 4Ah прерывания 21h, если же это сделать не получается, то выводится сообщение о соответствующей ошибке.

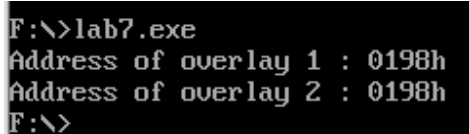
Затем с помощью процедуры CALL_OVERSIZE узнается размер файла оверлея с помощью функции 48h прерывания 21h. После чего, если получается определить размер модуля, процедура CALL_ALLOCATION_MEMORY запрашивает необходимый для модуля объем памяти с помощью функции 48h прерывания int21h. Иначе выводится сообщение о соответствующей ошибке.

Далее написана процедура CALL_PROGRAM, которая выполняет вызов оверлея с помощью функции 4B03h прерывания 21h. Если при вызове модуля произошла ошибка, то выводится соответствующее сообщение и программа завершается. Иначе выводится результат выполнения модуля.

В головной процедуре вызываемого модуля для каждого оверлея вызываются процедуры CALL_PATH, CALL_FREE_MEMORY, CALL_OVERSIZE, CALL_ALLOCATION_MEMORY, CALL_PROGRAM, кроме того после вызова каждой функции производится проверка на ее успешные выполнения. Если одна из функций завершится некорректно, то остальные функции выполняться не будут и произойдет завершение программы с помощью функции 4Ch прерывания 21h.

Также было написано два оверлейных сегмента, каждый из которых выводит адрес сегмента, в который он был загружен.

Запуск написанной программы, когда все модули находятся в директории F:\ (см. рис. 1). Программа работает корректно, т.к. оверлейные сегменты загружаются с одного и того же адреса, перекрывая друг друга.



```
F:\>lab7.exe
Address of overlay 1 : 0198h
Address of overlay 2 : 0198h
F:\>
```

Рисунок 1 – Результат работы программы

Запуск написанной программы, когда все модули находятся в каталоге lab7 директории F:\ (см. рис. 1). Программа работает корректно, т.к. оверлейные сегменты загружаются с одного и того же адреса, перекрывая друг друга (см. рис. 2).

```
F:\LAB7>lab7.exe
Address of overlay 1 : 0198h
Address of overlay 2 : 0198h
F:\LAB7>_
```

Рисунок 2 – Результат работы программы

Запуск написанной программы, когда одного оверлея нет в каталоге (см. рис. 3-4).

```
F:\LAB7>lab7.exe
Sorry, route not found
Address of overlay 2 : 0198h
F:\LAB7>_
```

Рисунок 3 – Результат работы программы

```
F:\LAB7>lab7.exe
Address of overlay 1 : 0198h
Sorry, route not found
F:\LAB7>_
```

Рисунок 4 – Результат работы программы

Выводы.

Была исследована возможность построение загрузочного модуля оверлейной структуры.

Ответы на контрольные вопросы.

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

Структурно модуль будет устроен идентично, так как .COM модуль содержит только 1 сегмент. Но задавать смещение адреса с помощью директивы ORG в начале модуля не нужно, т.к. при загрузке оверлейного модуля PSP не создается.