**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по практической работе №7**

**по дисциплине «Операционные системы»**

Тема: Построение модуля оверлейной структуры

|  |  |  |
| --- | --- | --- |
| Студент гр. 7383 |  | Зуев Д.В. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2019

**Постановка задачи.**

**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.

**Реализация задачи:**

В данной работе были написаны следующие функции:

TETR\_TO\_HEX – вспомогательная функция для функции BYTE\_TO\_HEX

BYTE\_TO\_HEX – переводит байт AL в два символа шестнадцатеричного числа в AX.

PRINT – вызывает прерывание программы для печати строки.

PRINT\_ERROR\_1 – функция, выводящая в соответствии со значением регистра AX сообщение об ошибке выполнения функции 4Ah и завершающая выполнение.

PRINT\_ERROR\_2 – функция, выводящая в соответствии со значением регистра AX сообщение об ошибке выполнения функции 4B03h и завершающая выполнение.

PRINT\_MEM\_ERROR – функция, выводящая в соответствии со значением регистра AX сообщение об ошибке выполнения функции 4Eh и завершающая выполнение.

PREPARE\_PLACE – функция, освобождающая место в памяти.

MAKE\_PATH – функция, строящая строку, содержащую путь и имя вызываемой программы.

OVL\_MEM – Определяет необходимое количество памяти для оверлейного модуля и выделяет её.

EXECUTION – функция, инициализирующая выполнение оверлейного модуля программы. Выполняет функцию 4B03h прерывания int 21h и очищает память от оверлейного модуля

Результаты работы программы представлены на рисунках 1 – 3.



Рисунок 1 – Результат загрузки оверлейных модулей из той же директории



Рисунок 2 – Результат загрузки оверлейных модулей из внешней директории



Рисунок 3 – Результат загрузки одного модуля

**Выводы.**

В ходе выполнения данной лабораторной работы была написана программа, запускающая выполнение оверлейных модулей из того же каталога что и у вызывающего модуля. Была исследована структура оверлейного сегмента и способ загрузки и выполнения оверлейных сегментов.

**Контрольные вопросы.**

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

*Программа должна в начале выделенной оверлейному модулю памяти формировать блок PSP размером 100h и выделять память под стек. При переходе в этот модуль должна смещать точку входа на 100h*