

**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

Котов Д.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Исследование возможности построения загрузочного модуля оверлейной структуры.

### **Постановка задачи.**

**Шаг 1.** 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) Файл оверлейного сегмента загружается и выполняется.
- 4) Освобождается память, отведённая 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.

### **Выполнение работы.**

#### **Шаг 1.**

Подготовлены строки для вывода требуемых сообщений.

Для выполнения задания написаны следующие процедуры:

- PUTS, нужная для вывода строк.
- FREE, освобождающая место в памяти с помощью функции 4Ah прерывания 21h.
- GET\_PATH, определяющая путь к файлу.

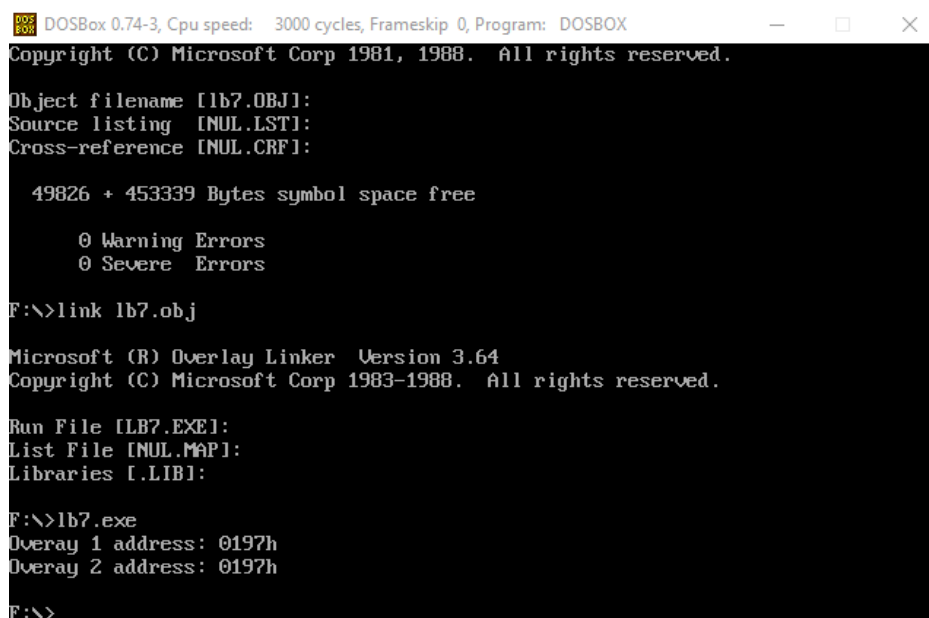
- GET\_FILESIZE, определяющая размер файла оверлея.
- MALLOC, запрашивающая необходимую память для файла оверлея.
- LOAD, выполняющая загрузку оверлея и освобождение отведённой под него памяти после выполнения.

При возникновении ошибок, выводятся соответствующие сообщения.

В процедуре MAIN для каждого из оверлеев последовательно вызываются вышеназванные процедуры, а также выполняется проверка значения регистра BX, которое равно 0 при успешном выполнении, иначе – 1. Если значение BX оказалось равно 1, то происходит переход к загрузке следующего оверлея или к завершению программы, при условии, что ошибка возникла при работе со вторым оверлеем.

**Шаг 2.** Написаны и отлажены оверлейные сегменты. Оверлейный сегмент с помощью процедуры PUTS выводит адрес сегмента, в который он загружен.

**Шаг 3.** Запустим отлаженное приложение. Оверлейные сегменты загружаются с одного и того же адреса, перекрывая друг друга.



```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lb7.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

49826 + 453339 Bytes symbol space free

0 Warning Errors
0 Severe Errors

F:\>link lb7.obj

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

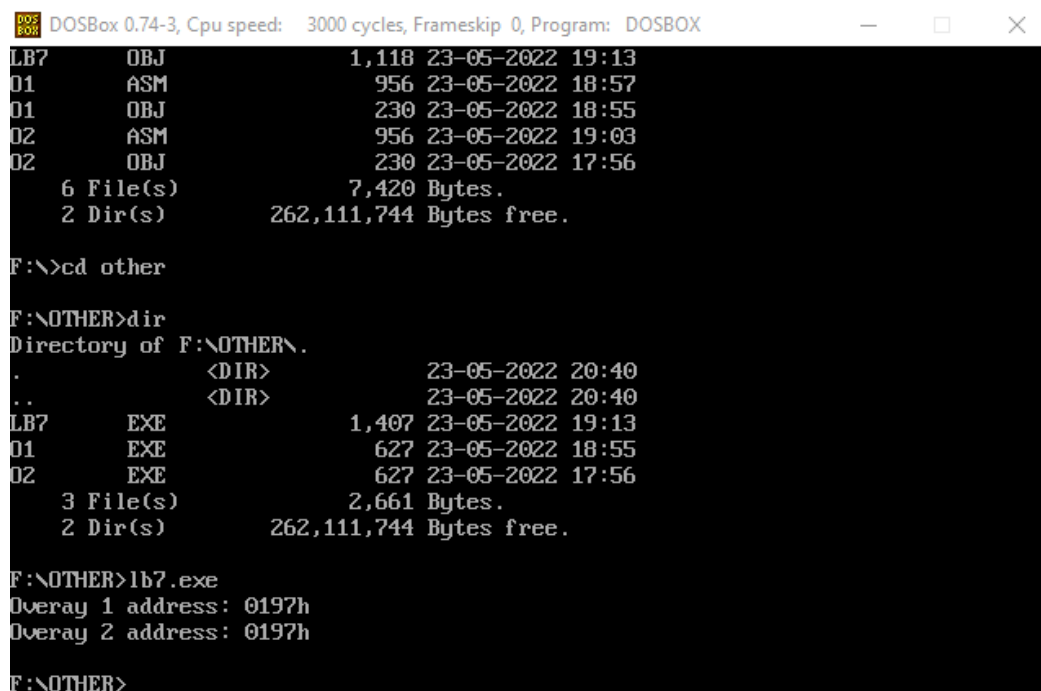
Run File [LB7.EXE]:
List File [NUL.MAP]:
Libraries [LIB]:

F:\>lb7.exe
Overlay 1 address: 0197h
Overlay 2 address: 0197h
F:\>_

```

*Рисунок 1 – Демонстрация работы программы. Оверлейные сегменты загружаются с одного и того же адреса*

**Шаг 4.** Запустим приложение из другого каталога.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
LB7      OBJ      1,118 23-05-2022 19:13
01       ASM      956 23-05-2022 18:57
01       OBJ      230 23-05-2022 18:55
02       ASM      956 23-05-2022 19:03
02       OBJ      230 23-05-2022 17:56
      6 File(s)      7,420 Bytes.
      2 Dir(s)      262,111,744 Bytes free.

F:\>cd other

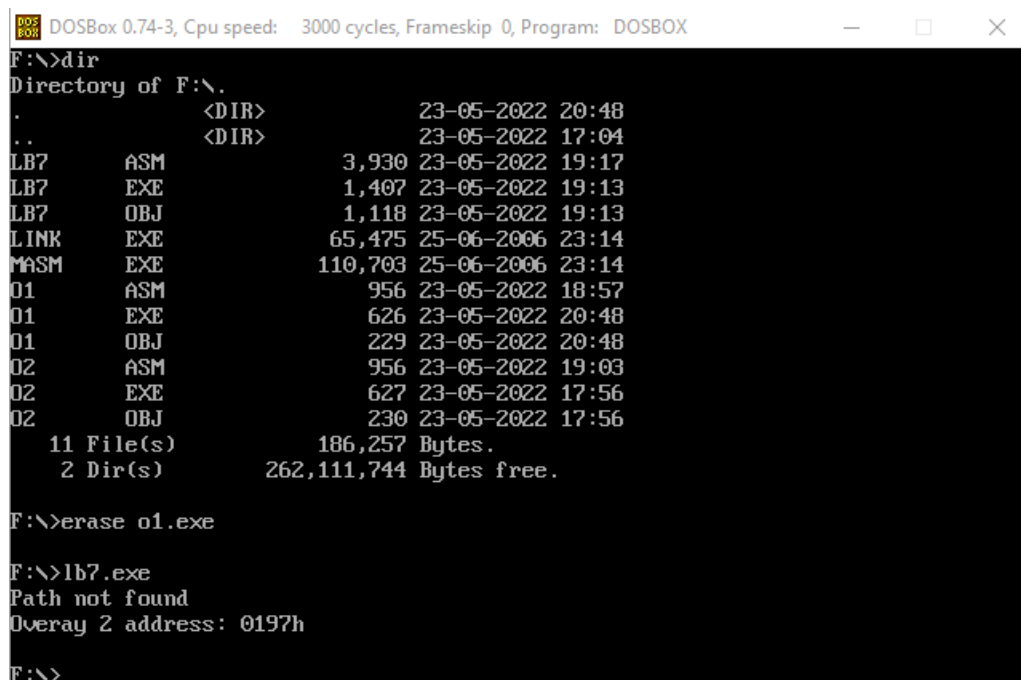
F:\OTHER>dir
Directory of F:\OTHER\
.          <DIR>      23-05-2022 20:40
..         <DIR>      23-05-2022 20:40
LB7       EXE      1,407 23-05-2022 19:13
01        EXE       627 23-05-2022 18:55
02        EXE       627 23-05-2022 17:56
      3 File(s)      2,661 Bytes.
      2 Dir(s)      262,111,744 Bytes free.

F:\OTHER>lb7.exe
Overlay 1 address: 0197h
Overlay 2 address: 0197h

F:\OTHER>_
```

*Рисунок 2 – Демонстрация работы программы. Запуск приложения из другого каталога*

**Шаг 5.** Запустим приложение в случае, когда одного оверлея нет в каталоге.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
F:\>dir
Directory of F:\
.          <DIR>      23-05-2022 20:48
..         <DIR>      23-05-2022 17:04
LB7       ASM      3,930 23-05-2022 19:17
LB7       EXE      1,407 23-05-2022 19:13
LB7       OBJ      1,118 23-05-2022 19:13
LINK      EXE     65,475 25-06-2006 23:14
MASM      EXE     110,703 25-06-2006 23:14
01        ASM      956 23-05-2022 18:57
01        EXE      626 23-05-2022 20:48
01        OBJ      229 23-05-2022 20:48
02        ASM      956 23-05-2022 19:03
02        EXE      627 23-05-2022 17:56
02        OBJ      230 23-05-2022 17:56
     11 File(s)     186,257 Bytes.
      2 Dir(s)     262,111,744 Bytes free.

F:\>erase 01.exe

F:\>lb7.exe
Path not found
Overlay 2 address: 0197h

F:\>_
```

*Рисунок 3 – Демонстрация работы программы. Первый оверлей удалён*

### **Выводы.**

Была исследована возможность построения загрузочного модуля оверлейной структуры.

### **ВОПРОСЫ**

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

Изменения не потребуются, так как оверлейные модули состоят из одного сегмента, точка входа в который находится по адресу 0, как в .COM модуле. Дополнительное смещение адреса, по которому передаётся управление не потребуется, поскольку при загрузке оверлейных модулей PSP не создаётся.