**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по практической работе №3**

**по дисциплине «Операционные системы»**

**Тема:** **Исследование организации управления основной памятью**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 8383 |  | Ишанина Л.Н. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2020

**Цель работы.**

Для исследования организации управления памятью необходимо ориентироваться на тип основной памяти, реализованный в компьютере и способ организации, принятый в ОС. В лабораторной работе рассматривается нестраничная память и способ управления динамическими разделами. Для реализации управления памятью в этом случае строится список занятых и свободных участков памяти. Функции ядра, обеспечивающие управление основной памятью, просматривают и преобразуют этот список.

В лабораторной работе исследуются структуры данных и работа функций управления памятью ядра операционной системы.

**Необходимые сведения для составления программы.**

Учет занятой и свободной памяти ведется при помощи списка блоков управления памятью MCB (MemoryControlBlock). MCB занимает 16 байт (параграф) и располагается всегда с адреса кратного 16 (адрес сегмента ОП) и находится в адресном пространстве непосредственно перед тем участком памяти, которым он управляет.

MCB имеет следующую структуру:

|  |  |  |
| --- | --- | --- |
| Смещение | Длина поля (байт) | Содержимое поля |
| 00h | 1 | тип MCB:  5Ah, если последний в списке,  4Dh, если не последний |
| 01h | 2 | Сегментный адрес PSP владельца участка памяти, либо  0000h - свободный участок,  0006h - участок принадлежит драйверу  OS XMS UMB  0007h - участок является исключенной верхней памятью драйверов  0008h - участок принадлежит MS DOS  FFFAh - участок занят управляющим блоком 386MAX UMB  FFFDh - участок заблокирован 386MAX  FFFEh - участок принадлежит 386MAX UMB |
| 03h | 2 | Размер участка в параграфах |
| 05h | 3 | Зарезервирован |
| 08h | 8 | "SC" - если участок принадлежит MS DOS, то в нем системный код  "SD" - если участок принадлежит MS DOS, то в нем системные данные |

По сегментному адресу и размеру участка памяти, контролируемого этим MCB можно определить местоположение следующего MCB в списке.

Адрес первого MCB хранится во внутренней структуре MS DOS, называемой "ListofLists" (список списков). Доступ к указателю на эту структуру можно получить, используя функцию f52h "GetListofLists" int 21h. В результате выполнения этой функции ES:BX будет указывать на список списков. Слово по адресу ES:[BX-2] и есть адрес самого первого MCB.

Размер расширенной памяти находится в ячейках 30h, 31h CMOS. CMOS это энергонезависимая память, в которой хранится информация о конфигурации ПЭВМ. Объем памяти составляет 64 байта. Размер расширенной памяти в Кбайтахможно определить обращаясь к ячейкам CMOS следующим образом:

movAL,30h ; запись адреса ячейки CMOS

out 70h,AL

inAL,71h ; чтение младшего байта

movBL,AL ; размера расширенной памяти

movAL,31h ; запись адреса ячейки CMOS

out 70h,AL

inAL,71h ; чтение старшего байта

; размера расширенной памяти

## Выполнение работы.

Шаг 1. Был написан текст исходного .COM модуля, который выбирает и распечатывает следующую информацию:

1) Количество доступной памяти.

2) Размер расширенной памяти.

3) Выводит цепочку блоков управления памятью.

Результат, полученный программой, представлен на рисунке 1.

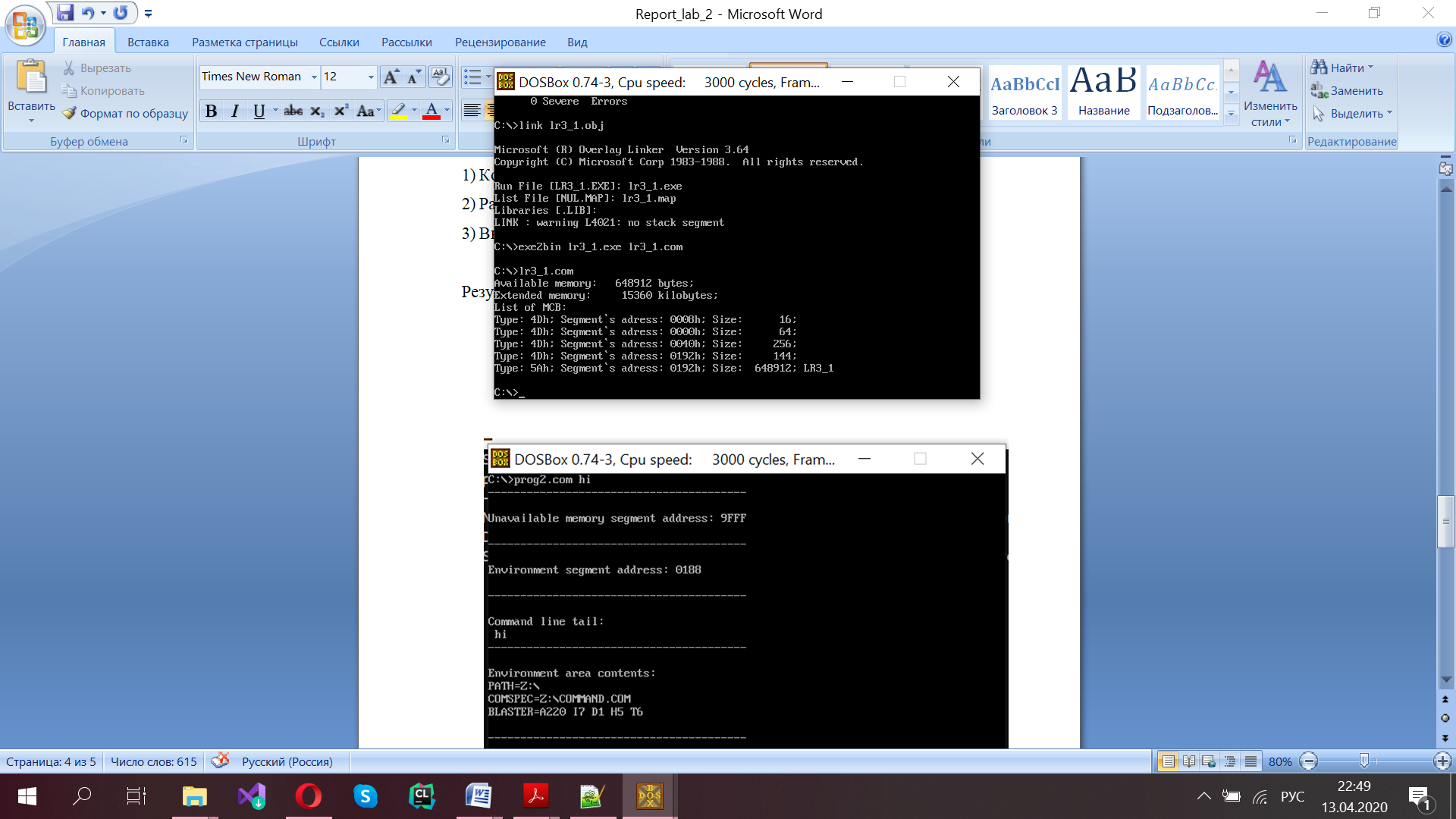


Рисунок 1 — Запуск первого варианта .COM модуля

Шаг 2. Программа была изменена так, чтобы она освобождала память, которую не занимает. Для этого была использована функция 4Ah прерывания 21h. Результат, полученный программой, представлен на рисунке 2.

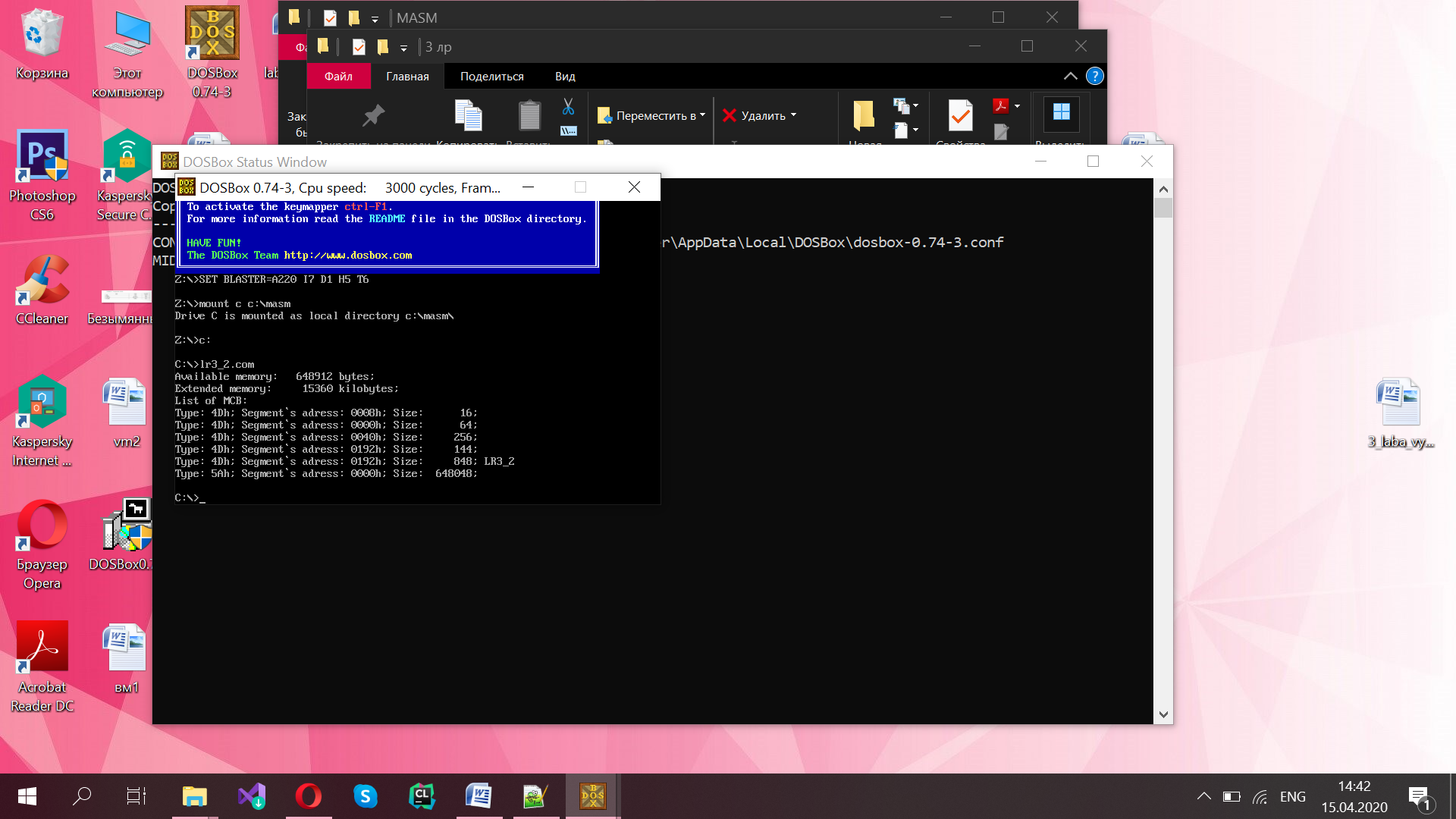


Рисунок 2 — Запуск второго варианта .COM модуля

Шаг 3. Программа была изменена так, чтобы после освобождения памяти она запрашивала 64КБ памяти функцией 48h прерывания 21h. Результат, полученный программой, представлен на рисунке 3.

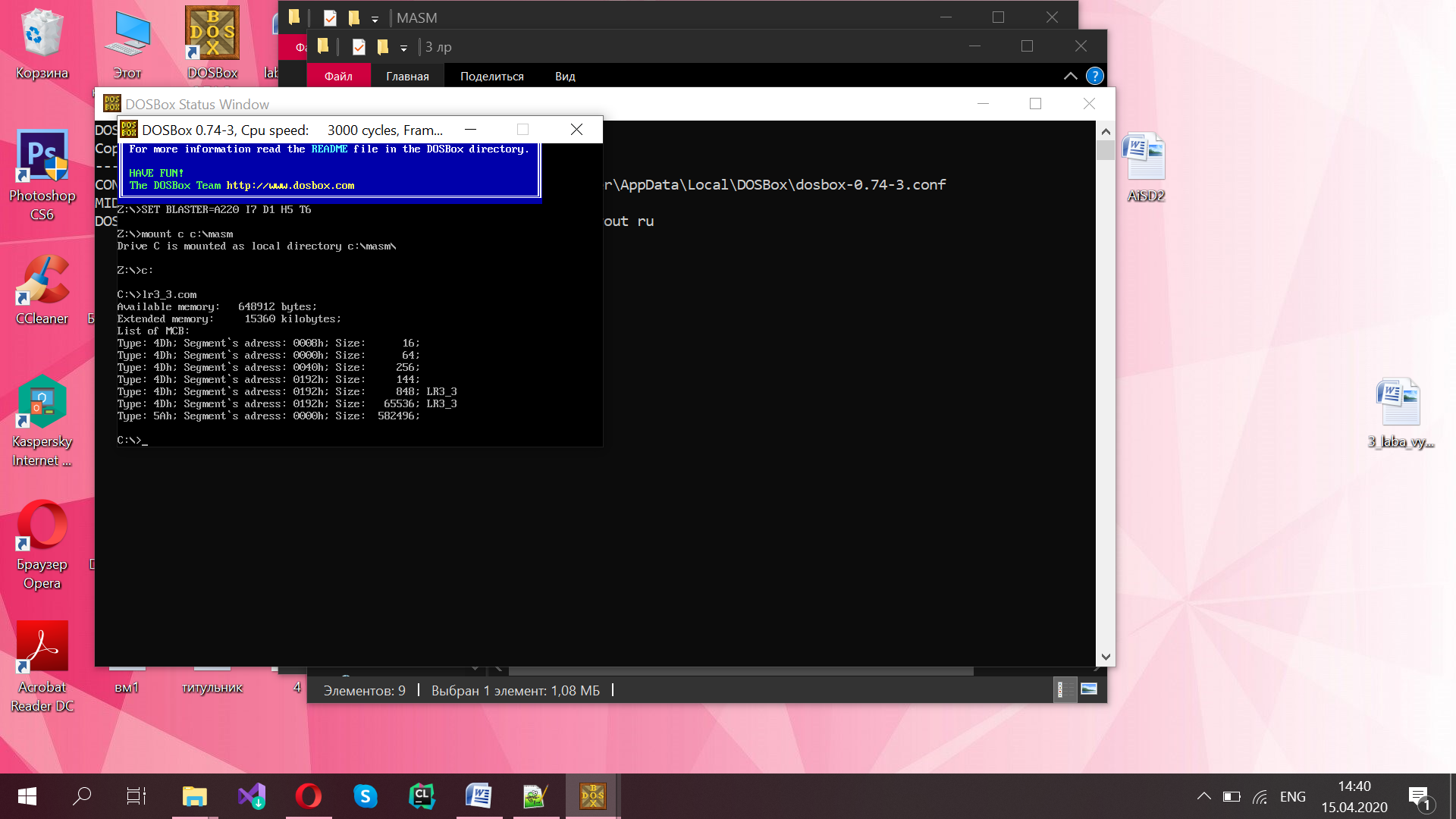


Рисунок 3 — Запуск третьего варианта .COM модуля

Шаг 4. Изначальная программа была изменена так, чтобы до освобождения памяти она запрашивала 64КБ памяти функцией 48h прерывания 21h. Результат, полученный программой, представлен на рисунке 4.

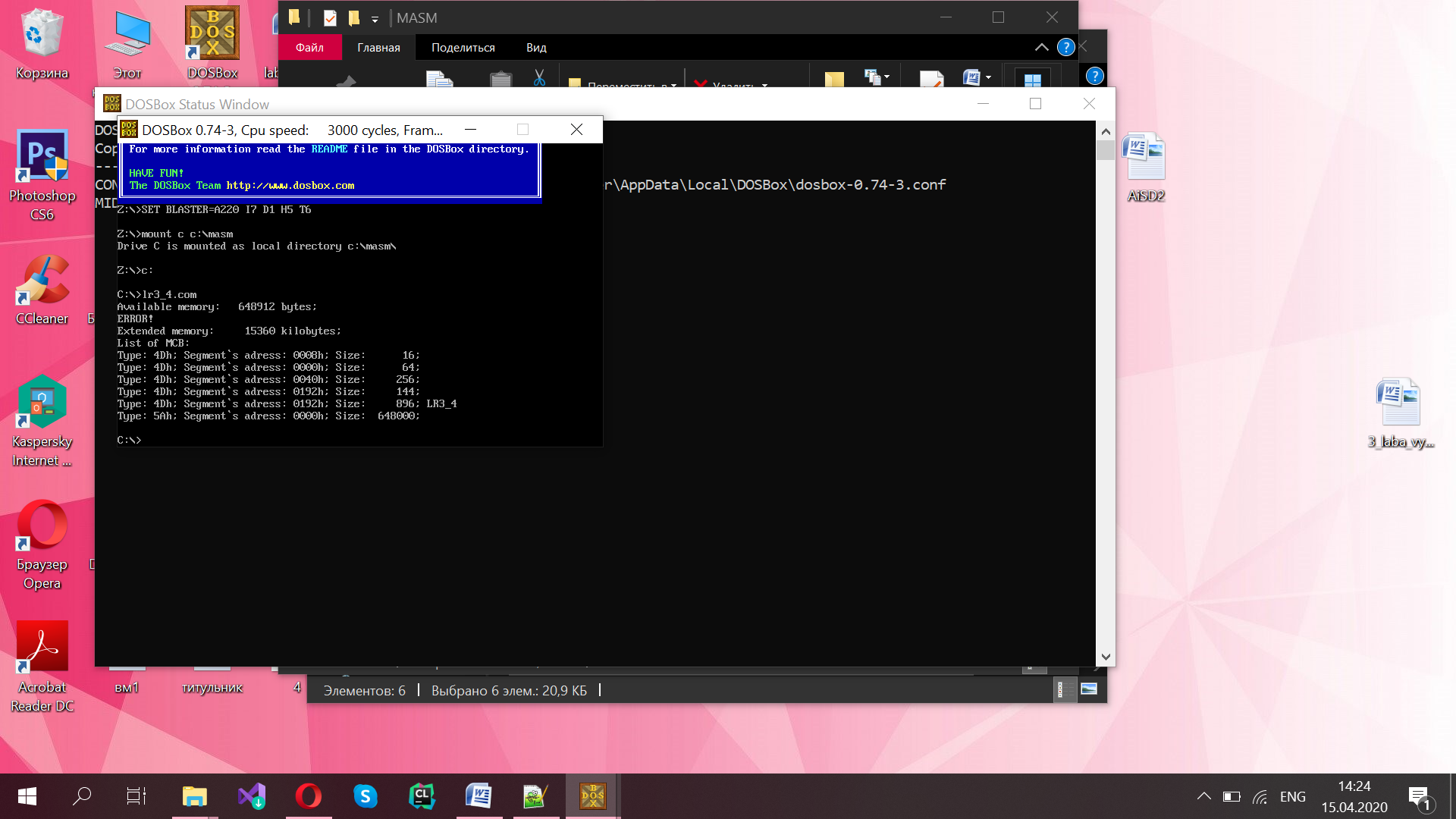


Рисунок 4 — Запуск четвертого варианта .COM модуля

**Ответы на контрольные вопросы по лабораторной работе №3:**

**Что означает «доступный объем памяти»?**

Это максимальный объем памяти, который может быть доступен программе.

**Где МСВ блок программы в списке?**

1) Последний блок, занимает всю доступную память.

2) Пятый блок (происходит освобождение неиспользованной памяти, блок с неиспользуемой ею памятью — последний)

3) Пятый блок (сначала освобождение, потом запрос)

4) Программа запрашивает память до того, как освобождает неиспользуемую – и при обработке завершения функций ядра (проверке флага CF) мы узнаём, что возникает ошибка.

**Какой размер памяти занимает программа в каждом случае?**

1) Всю свободную – 144 б + 648912 б

2) Необходимый объем – 144 б + 848 б

3) Необходимый + запрошенный объем – 144 б + 848 б + 65536 б

4) При попытке выделить дополнительную память, возникает ошибка.

– 144 б + 896 б

## Выводы.

Были исследованы структуры данных и работа функций управления памятью ядра операционной системы.