

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

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Операционные системы»**  
**Тема: Обработка стандартных прерываний**

Студент гр. 0381

Котов Д.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Изучение принципа работы стандартного обработчика прерываний.  
Построить обработчик прерываний таймера.

### **Постановка задачи.**

**Шаг 1.** Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .EXE, который выполняет следующие функции:

- 1) Проверяет, установлено ли пользовательское прерывание с вектором 1Ch.
- 2) Устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний, если прерывание не установлено, и осуществляет выход по функции 4Ch прерывания int 21h.
- 3) Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания int 21h.
- 4) Выгрузка прерывания по соответствующему значению параметра в командной строке /un. Выгрузка прерывания состоит в восстановлении стандартного вектора прерываний и освобождении памяти, занимаемой резидентом. Затем осуществляется выход по функции 4Ch прерывания int 21h.

Программа должна содержать код устанавливаемого прерывания в виде удалённой процедуры. Этот код будет работать после установки при возникновении прерывания. Он должен выполнять следующие функции:

- 1) Сохранить значения регистров в стеке при выходе и восстановить их при выходе.
- 2) При выполнении тела процедуры накапливать общее суммарное и выводить на экран. Для вывода на экран следует использовать прерывание int 10h, которое позволяет непосредственно выводить информацию на экран.

**Шаг 2.** Запустите отлаженную программу и убедитесь, что резидентный обработчик прерывания 1Ch установлен. Работа прерывания должна отражаться на экране, а также необходимо проверить размещение прерывания в памяти. Для этого запустите программу ЛР 3, которая отображает карту памяти в виде списка блоков МСВ. Полученные результаты поместите в отчёт.

**Шаг 3.** Запустите отлаженную программу ещё раз и убедитесь, что программа определяет установленный обработчик прерываний. Полученные результаты поместите в отчёт.

**Шаг 4.** Запустите отлаженную программу с ключом выгрузки и убедитесь, что резидентный обработчик прерывания выгружен, то есть сообщения на экран не выводятся, а память, занятая резидентом освобождена. Для этого также следует запустить программу ЛР 3. Полученные результаты поместите в отчёт.

#### **Исходные данные.**

За основу был взят код, представленный в методическом пособии, содержащий процедуры: outputBP, setCurs, getCurs.

#### **Выполнение работы.**

##### **Шаг 1.**

Подготовлены строки для вывода требуемых сообщений.

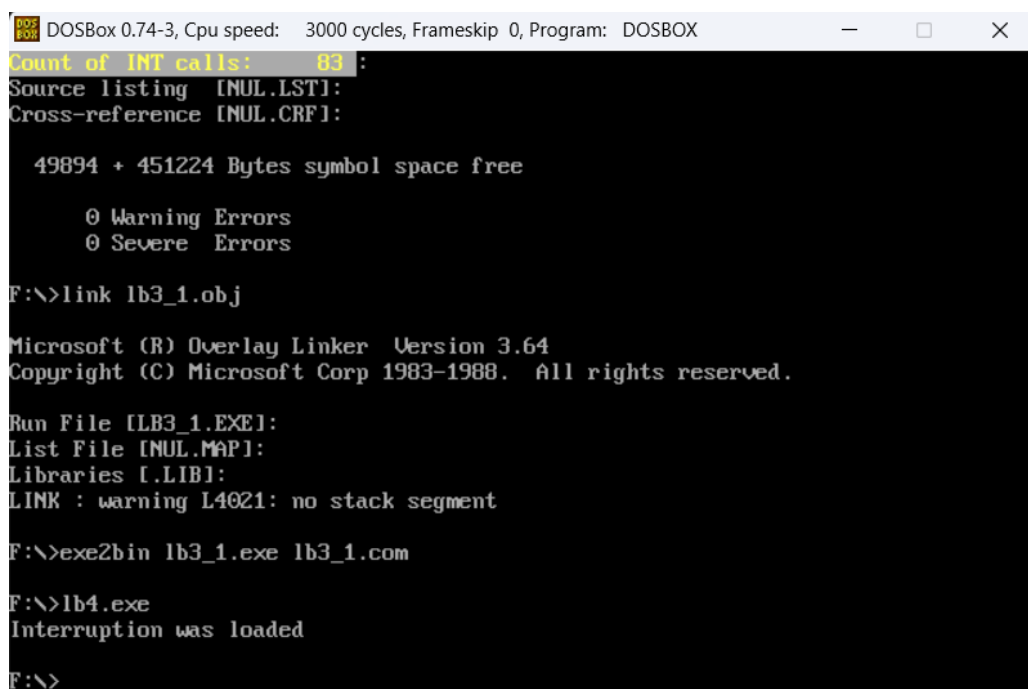
Написан обработчик прерывания 1Ch NEWINT. В обработчике организован стек на 512 слов. В слове NEW хранится идентификатор обработчика, который позволяет отличить его от стандартного. В слова KEEP\_IP и KEEP\_CS помещаются значения смещения и адреса сегмента стандартного обработчика при установке нового. В INT\_COUNT хранится строка, в которой записано, сколько раз был вызван обработчик прерываний.

Для выполнения задания написаны следующие процедуры:

- PUTS, нужная для вывода строк.
- COUNTER, записывающая количество прерываний в строку INT\_COUNT.

- LOAD\_NEWINT, устанавливающая нового обработчика.
- CHECK\_NEWINT, определяющая, установлен ли новый обработчик. Если это так, то в al заносится 1, иначе – 0.
- UNLOAD\_CHECK, проверяющая, была ли подана команда выгрузить новый обработчик. Если это так, то в bl заносится 1, иначе – 0.
- UNLOAD\_NEWINT, восстанавливающая стандартный обработчик.

**Шаг 2.** Запустим отлаженную программу и убедимся, что резидентный обработчик прерывания 1Ch установлен. Работа прерывания должна отражаться на экране, а также проверим размещение прерывания в памяти. Для этого запустим программу ЛР 3, которая отображает карту памяти в виде списка блоков MCB.



```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Count of INT calls: 83 :
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

49894 + 451224 Bytes symbol space free

  0 Warning Errors
  0 Severe Errors

F:\>link lb3_1.obj

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

Run File [LB3_1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

F:\>exe2bin lb3_1.exe lb3_1.com

F:\>lb4.exe
Interrupt was loaded

F:\>_

```

*Рисунок 1 – Установка резидентного обработчика*

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Count of INT calls: 2001 .com

F:\>lb4.exe
Interruption was loaded

F:\>lb3_1.com

Available memory size: 64 bytes
Extended memory size: 15728640 bytes
MCB Type: 4Dh Address: 016Fh Owner: MS DOS Area size: 16 bytes
SC/SD:
MCB Type: 4Dh Address: 0171h Owner: Free Area size: 64 bytes
SC/SD:
MCB Type: 4Dh Address: 0176h Owner: 0040h Area size: 256 bytes
SC/SD:
MCB Type: 4Dh Address: 0187h Owner: 0192h Area size: 144 bytes
SC/SD:
MCB Type: 4Dh Address: 0191h Owner: 0192h Area size: 1520 bytes
SC/SD: LB4
MCB Type: 4Dh Address: 01F1h Owner: 01FCh Area size: 144 bytes
SC/SD:
MCB Type: 5Ah Address: 01FBh Owner: 01FCh Area size: 647216 bytes
SC/SD: LB3_1

F:\>
```

Рисунок 2 – Список блоков MCB после установки обработчика

**Шаг 3.** Запустим отлаженную программу ещё раз и убедимся, что программа определяет установленный обработчик прерываний.

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Count of INT calls: 11646

F:\>lb3_1.com

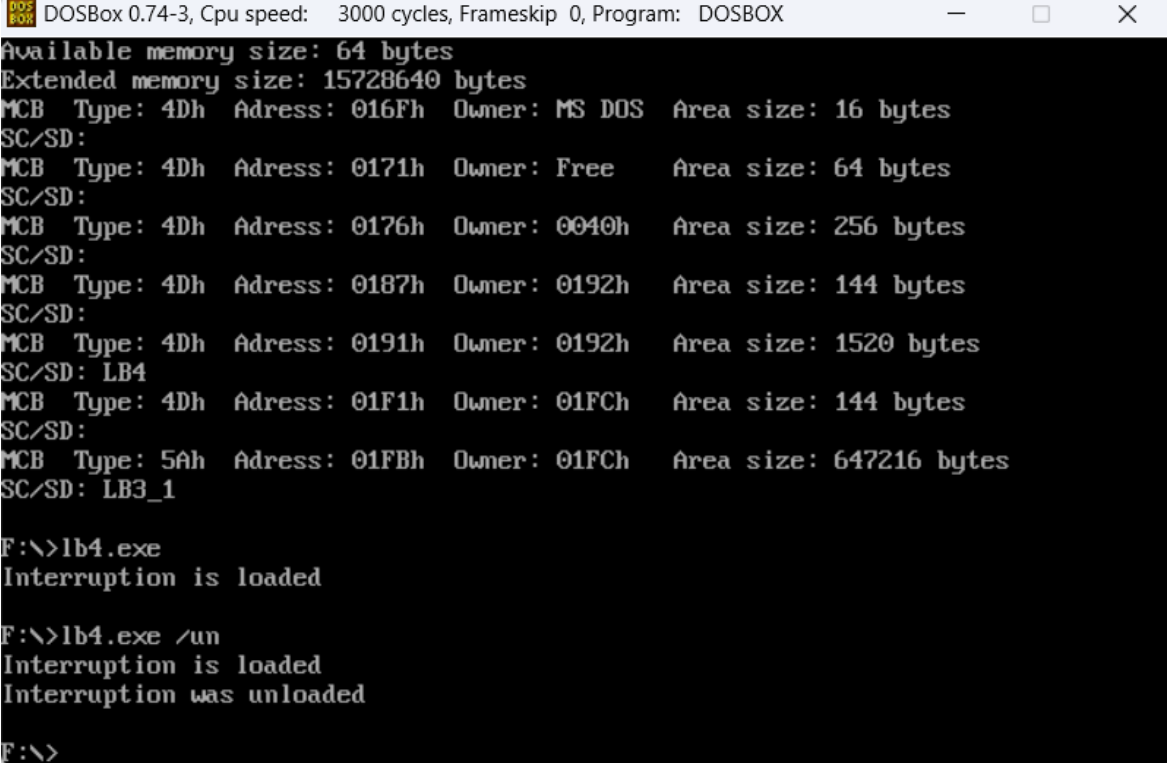
Available memory size: 64 bytes
Extended memory size: 15728640 bytes
MCB Type: 4Dh Address: 016Fh Owner: MS DOS Area size: 16 bytes
SC/SD:
MCB Type: 4Dh Address: 0171h Owner: Free Area size: 64 bytes
SC/SD:
MCB Type: 4Dh Address: 0176h Owner: 0040h Area size: 256 bytes
SC/SD:
MCB Type: 4Dh Address: 0187h Owner: 0192h Area size: 144 bytes
SC/SD:
MCB Type: 4Dh Address: 0191h Owner: 0192h Area size: 1520 bytes
SC/SD: LB4
MCB Type: 4Dh Address: 01F1h Owner: 01FCh Area size: 144 bytes
SC/SD:
MCB Type: 5Ah Address: 01FBh Owner: 01FCh Area size: 647216 bytes
SC/SD: LB3_1

F:\>lb4.exe
Interruption is loaded

F:\>_
```

Рисунок 3 – Программа определяет установленный обработчик

**Шаг 4.** Запустим отлаженную программу с ключом выгрузки и убедимся, что резидентный обработчик прерывания выгружен, то есть сообщения на экран не выводятся, а память, занятая резидентом освобождена. Для этого также запустим программу ЛР 3.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Available memory size: 64 bytes
Extended memory size: 15728640 bytes
MCB Type: 4Dh Address: 016Fh Owner: MS DOS Area size: 16 bytes
SC/SD:
MCB Type: 4Dh Address: 0171h Owner: Free Area size: 64 bytes
SC/SD:
MCB Type: 4Dh Address: 0176h Owner: 0040h Area size: 256 bytes
SC/SD:
MCB Type: 4Dh Address: 0187h Owner: 0192h Area size: 144 bytes
SC/SD:
MCB Type: 4Dh Address: 0191h Owner: 0192h Area size: 1520 bytes
SC/SD: LB4
MCB Type: 4Dh Address: 01F1h Owner: 01FCh Area size: 144 bytes
SC/SD:
MCB Type: 5Ah Address: 01FBh Owner: 01FCh Area size: 647216 bytes
SC/SD: LB3_1

F:\>lb4.exe
Interruption is loaded

F:\>lb4.exe /un
Interruption is loaded
Interruption was unloaded

F:\>
```

*Рисунок 4 – Выгрузка резидентного обработчика*

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
SC/SD: LB3_1
F:\>lb4.exe
Interruption is loaded
F:\>lb4.exe /un
Interruption is loaded
Interruption was unloaded
F:\>lb3_1.com
Available memory size: 64 bytes
Extended memory size: 15728640 bytes
MCB Type: 4Dh Address: 016Fh Owner: MS DOS Area size: 16 bytes
SC/SD:
MCB Type: 4Dh Address: 0171h Owner: Free Area size: 64 bytes
SC/SD:
MCB Type: 4Dh Address: 0176h Owner: 0040h Area size: 256 bytes
SC/SD:
MCB Type: 4Dh Address: 0187h Owner: 0192h Area size: 144 bytes
SC/SD:
MCB Type: 5Ah Address: 0191h Owner: 0192h Area size: 648912 bytes
SC/SD: LB3_1
F:\>_
```

*Рисунок 5 – Список блоков MCB после выгрузки обработчика*

## **Выводы.**

Был изучен принцип работы стандартного обработчика прерываний.  
Был построен обработчик прерываний таймера.

## ВОПРОСЫ

1) Как реализован механизм прерывания от часов?

Прерывание 1Ch вызывается каждые 55 миллисекунд (по каждому тiku аппаратных часов). Происходит следующее:

- a) Сохраняется состояние регистров.
- b) Определяется источник прерывания по номеру источника прерывания определяется смещение в таблице векторов прерываний.
- c) Первые 2 байта помещаются в IP, вторые 2 байта в CS.
- d) Передаётся управление по адресу CS:IP, то есть запускается обработчик прерывания.
- e) Обработка прерывания.
- f) Возврат управления прерванной программе.

2) Какого типа прерывания использовались в работе?

Программные прерывания 10h, 21h и аппаратное прерывание 1Ch.