

**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. 9381

Камакин Д.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

В архитектуре компьютера существуют стандартные прерывания, за которыми закреплены определённые вектора прерываний. Вектор прерываний хранит адрес подпрограммы обработчика прерываний. При возникновении прерывания, аппаратура компьютера передаёт управление и выполняет соответствующие действия. В данной лабораторной работе №4 предлагается построить обработчик прерываний сигналов таймера. Эти сигналы генерируются аппаратурой через определённые интервалы времени и, при возникновении такого сигнала, возникает прерывание с определённым значением вектора. Таким образом, управление будет передано функции, чья точка входа записана в соответствующий вектор прерывания.

### **Функции.**

INTERRUPTION\_TASK — прерывания, написанное по условию задания.

SAVE\_CURSOR — через BX возвращает текущее положение курсора

CHECK\_INTERRUPT — проверяет, установлен ли наш вектор прерывания

CHECK\_ARGUMENT — проверяет, была ли программа запущена с опцией /un

INTERRUPTION\_UNLOAD — восстанавливает сохранённый обработчик прерываний и выгружает резидентную программу

INTERRUPTION\_LOAD — устанавливает новый обработчик прерывания

OUTPUT — выводит строку, адрес которой записан в dx, на экран

## Ход работы

Был написан модуль типа .EXE, который содержит следующий функционал: проверка, установлено ли написанное прерывание с вектором 1Ch.

Если прерывание не установлено, то программа устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний.

В ином случае выводится соответствующее сообщение.

Если программа была запущена с аргументом /up, то прерывание будет выгружено.

В любом случае будет выполнен выход по функции 4Ch прерывания int 21h.

Программа была протестирована:

1. Запуск программы. Выводится строка с количеством вызовов прерывания:

```
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.
Interrupts call count: 0029
Object filename [lab4.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

49892 + 453273 Bytes symbol space free

0 Warning Errors
0 Severe Errors

F:\>link lab4.obj

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

Run File [LAB4.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

F:\>lab4.exe
The interruption handler is loading
```

2. Повторный запуск программы. Выводится сообщение о том, что прерывание уже загружено.

```
Object filename [lab4.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]: Interrupts call count: 0601

49892 + 453273 Bytes symbol space free

0 Warning Errors
0 Severe Errors

F:\>link lab4.obj

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

Run File [LAB4.EXE]:
List File [NUL.MAP]:
Libraries [LIB]:

F:\>lab4.exe
The interruption handler is loading

F:\>lab4.exe
The interruption handler is already loaded
```

3. Вывод списка блоков MCB. Обработчик прерываний остался резидентным в памяти.

```
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LAB4.EXE]: Interrupts call count: 1533
List File [NUL.MAP]:
Libraries [LIB]:

F:\>lab4.exe
The interruption handler is loading

F:\>lab4.exe
The interruption handler is already loaded

F:\>LAB3_1.COM
Available memory: 648112 bytes
Extended memory: 15370 bytes
MCB table:
PSP address: 0008 Size: 16 SC/SD:
PSP address: 0000 Size: 64 SC/SD:
PSP address: 0040 Size: 256 SC/SD:
PSP address: 0192 Size: 144 SC/SD:
PSP address: 0192 Size: 624 SC/SD: LAB4
PSP address: 01C4 Size: 144 SC/SD:
PSP address: 01C4 Size: 648112 SC/SD: LAB3_1
```

4. Программа запущена с аргументом /un (выгрузка прерывания). В блоках MCB видно, что обработчик прерываний выгружен.

```
Available memory: 648112 bytes
Extended memory: 15370 bytes
MCB table:
PSP address: 0008 Size: 16 SC/SD:
PSP address: 0000 Size: 64 SC/SD:
PSP address: 0040 Size: 256 SC/SD:
PSP address: 0192 Size: 144 SC/SD:
PSP address: 0192 Size: 624 SC/SD: LAB4
PSP address: 01C4 Size: 144 SC/SD:
PSP address: 01C4 Size: 648112 SC/SD: LAB3_1

F:\>LAB4.EXE /un
The interruption handler is restored

F:\>LAB3_1.COM
Available memory: 648912 bytes
Extended memory: 15370 bytes
MCB table:
PSP address: 0008 Size: 16 SC/SD:
PSP address: 0000 Size: 64 SC/SD:
PSP address: 0040 Size: 256 SC/SD:
PSP address: 0192 Size: 144 SC/SD:
PSP address: 0192 Size: 648912 SC/SD: LAB3_1
```

### **Результаты исследования проблем.**

1. Как реализован механизм прерывания от часов?

После поступления сигнала прерывания (вызывается каждые 55 мс) проверяется, не запрещены ли прерывания такого типа в данный момент. Если не разрешены, то ничего не происходит. Иначе запоминается контекст прерванного процесса, запрещаются прерывания данного типа, происходит обработка прерывания, снимается запрет, восстанавливается контекст.

2. Какого типа прерывания использовались в работе?

Программные, вызываемые командами int 21h и int 10h.

Аппаратные (от часов, 1Ch).

## **Выводы**

В лабораторной работе была исследована обработка стандартных прерываний, построен обработчик прерываний сигналов таймера.