

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

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

Студентка гр. 0381

Сарычева А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Построить обработчик прерываний сигналов таймера. Эти сигналы генерируются аппаратурой через определенные интервалы времени и, при возникновении такого сигнала, возникает прерывание с определенным значением вектора. Таким образом, управление будет передано функции, чья точка входа записана в соответствующий вектор прерывания.

### **Задание.**

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 осуществляется 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 удаленной процедуры. Этот код будет работать после установки при возникновении прерывания. Он должен выполнять следующие функции:

- 1) Сохраняет стек прерванной программы (регистры SS и SP) в рабочих переменных и восстановить при выходе.
- 2) Организовать свой стек.
- 3) Сохранить значения регистров в стеке при входе и восстановить их при выходе.

4) При выполнении тела процедуры накапливать общее суммарное число прерываний и выводить на экран. Для вывода на экран следует использовать прерывание `int 10h`, которое позволяет непосредственно выводить информацию на экран.

5) Функция прерывания должна содержать только переменные, которые она использует.

Шаг 2. Запустите отлаженную программу и убедитесь, что резидентный обработчик прерывания `1Ch` установлен. Работа прерывания должна отображаться 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 отчет.

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

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 отчет.

### **Выполнение работы.**

В файле `lb4.asm` написан исходный код `.EXE` модуля. В начале данного модуля прописаны строки для вывода запрашиваемой информации, для вывода которых была создана процедура `OUTPUT`. Кроме того, были использованы процедуры `OUTPUT_STR` для вывода строки на экран, `SET_CURS` для установления позиции курсора, `GET_CURS` для считывания позиции курсора.

Написана процедура `CHECK_TAIL`, которая проверяет был ли введен параметр `/un` при запуске программы с помощью побайтного сравнения ячеек памяти, отвечающих за хвост командной строки и его размер с необходимой строкой `UNLOAD_MESSAGE`.

Процедура CHECK\_INSTALL проверяет установлено ли пользовательское прерывания с помощью функции получения вектора 35h int 21h и сравнения поля вектора и строки SIGNATURE по необходимому адресу в этом векторе, если они равны, то процедура в регистре cl сохраняет 1, иначе 0.

Процедура UNLOAD, если пользовательское прерывание установлено, выгружает его, восстанавливая старый обработчик прерывания с помощью функции 25h int 21, при этом освобождая занятые блоки памяти с помощью функции 49h int 21. В конце выводится сообщение о выполненном действии с помощью процедуры OUTPUT.

Далее написана процедура DON'T\_UNLOAD, которая сначала с помощью CHECK\_INSTALL проверяет установлен ли обработчик прерывания, если да, то просто выводится сообщение, что он уже установлен, иначе устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний с помощью функций 31h и 25h int 21h. В конце выводится сообщение о установке обработчика прерывания.

Процедура NEW\_INTERRUPTION является реализованным прерыванием, которое имеет свой стек, в который сохраняются значения всех используемых далее регистров для их восстановления при выходе. После загрузки в память, прерывание начинает подсчет количества прерываний, вызванных таймером, данное значение выводится в виде строки с помощью процедур GET\_CURS, SET\_CURS и OUTPUT\_STR.

Для преобразований числа прерываний в строку в десятичном виде используется процедура CONVERSION, которая сохраняет полученное число в строку MESSAGE.

В головной процедуре происходит вызов процедуры CHECK\_TAIL, от результата которой вызывается либо процедура UNLOAD, либо DONT\_UNLOAD, после которых происходит выход из программы с помощью функции 4Ch прерывания 21h.

Написанная программа при ее первом вызове, устанавливает обработчик прерывания и выводит об этом сообщение. После его установки в левом верхнем углу появляется счетчик количества прерываний таймера (см. рис. 1).

```
Count timer signals: 129  lter Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lb4.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

49900 + 449168 Bytes symbol space free

0 Warning Errors
0 Severe Errors

F:\>link lb4.obj

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

Run File [LB4.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

F:\>lb4.exe
User interruption is installed now

F:\>
```

Рисунок 1 – Запуск модуля lb4.exe

При запуске программы лабораторной работы №3 счетчик продолжает свою работу, а в выводе лаб. работы №3 видно, что программа из текущей лабораторной работы загружена в память (см. рис. 2).

```
Count timer signals: 527

F:\>lb4.exe
User interruption is installed now

F:\>lb3_1.com
Amount of available memory: 64 bytes
Extended memory size: 585728 bytes
Memory control blocks:
1 Type: 4Dh MSB adress: 016Fh PSP address: 0008h MSB size: 16 text:
2 Type: 4Dh MSB adress: 0171h PSP address: 0000h MSB size: 64 text:
3 Type: 4Dh MSB adress: 0176h PSP address: 0040h MSB size: 256 text:
4 Type: 4Dh MSB adress: 0187h PSP address: 0192h MSB size: 144 text:
5 Type: 4Dh MSB adress: 0191h PSP address: 0192h MSB size: 1104 text: LB4
6 Type: 4Dh MSB adress: 01D7h PSP address: 01E2h MSB size: 144 text:
7 Type: 5Ah MSB adress: 01E1h PSP address: 01E2h MSB size: 647632 text: LB3
1
F:\>
```

Рисунок 2 – Запуск модуля lb3\_1.com после загрузки обработчика прерывания

При повторной попытке загрузить созданный модуль программы выводится сообщения о том, что обработчик уже установлен.

```
Count timer signals: 1136 lled now
F:\>lb3_1.com
Amount of available memory: 64 bytes
Extended memory size: 585728 bytes
Memory control blocks:
 1 Type: 4Dh MSB adress: 016Fh PSP adress: 0008h MSB size: 16 text:
 2 Type: 4Dh MSB adress: 0171h PSP adress: 0000h MSB size: 64 text:
 3 Type: 4Dh MSB adress: 0176h PSP adress: 0040h MSB size: 256 text:
 4 Type: 4Dh MSB adress: 0187h PSP adress: 0192h MSB size: 144 text:
 5 Type: 4Dh MSB adress: 0191h PSP adress: 0192h MSB size: 1104 text: LB4
 6 Type: 4Dh MSB adress: 01D7h PSP adress: 01E2h MSB size: 144 text:
 7 Type: 5Ah MSB adress: 01E1h PSP adress: 01E2h MSB size: 647632 text: LB3
_1
F:\>lb4.exe
User interruption has already installed.
F:\>_
```

Рисунок 3 –Повторный запуск модуля lb4.exe

При попытке же запуска данного модуля с ключом /un, происходит восстановление стандартного вектора прерывания и освобождение памяти, занимаемой резидентом, после которого происходит исчезновение счетчика с экрана консоли, вывод соответствующего сообщения и завершение работы программы.

```
Amount of available memory: 64 bytes
Extended memory size: 585728 bytes
Memory control blocks:
 1 Type: 4Dh MSB adress: 016Fh PSP adress: 0008h MSB size: 16 text:
 2 Type: 4Dh MSB adress: 0171h PSP adress: 0000h MSB size: 64 text:
 3 Type: 4Dh MSB adress: 0176h PSP adress: 0040h MSB size: 256 text:
 4 Type: 4Dh MSB adress: 0187h PSP adress: 0192h MSB size: 144 text:
 5 Type: 4Dh MSB adress: 0191h PSP adress: 0192h MSB size: 1104 text: LB4
 6 Type: 4Dh MSB adress: 01D7h PSP adress: 01E2h MSB size: 144 text:
 7 Type: 5Ah MSB adress: 01E1h PSP adress: 01E2h MSB size: 647632 text: LB3
_1
F:\>lb4.exe
User interruption has already installed.
F:\>lb4.exe /un
Unload interruption
F:\>
```

Рисунок 4 – Запуск модуля lb4.exe с ключом /un

В освобождение памяти можно убедиться, повторно запустив программу лабораторной работы №3.

```
7 Type: 5Ah MSB adress: 01E1h PSP adress: 01E2h MSB size: 647632 text: LB3
_1

F:\>lb4.exe
User interruption has already installed.

F:\>lb4.exe /un
Unload interruption

F:\>lb3_1.com
Amount of available memory: 64 bytes
Extended memory size: 585728 bytes
Memory control blocks:
1 Type: 4Dh MSB adress: 016Fh PSP adress: 0008h MSB size: 16 text:
2 Type: 4Dh MSB adress: 0171h PSP adress: 0000h MSB size: 64 text:
3 Type: 4Dh MSB adress: 0176h PSP adress: 0040h MSB size: 256 text:
4 Type: 4Dh MSB adress: 0187h PSP adress: 0192h MSB size: 144 text:
5 Type: 5Ah MSB adress: 0191h PSP adress: 0192h MSB size: 648912 text: LB3
_1

F:\>
```

Рисунок 5 – Повторный запуск модуля lb3.com

### **Выводы.**

Был построен обработчик прерываний сигналов таймера, для которого была создана резидентная функция для обработки прерывания, вызываемая после настройки вектора прерывания.

### **Ответы на контрольные вопросы.**

1) Как реализован механизм прерывания от часов?

Прерывания 1Ch вызывается при получении сигнала по каждому тикку аппаратных часов (каждые 55 миллисекунд). Первоначально он указывает на IRET, но может быть изменен пользовательской программой, чтобы адресовать фоновую программу пользователя, базирующуюся на таймере.

2) Какого типа прерывания использовались в работе?

В работе использовались программные прерывания: int 10h и int 21h, аппаратное прерывание 1Ch.