

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

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

Студент гр. 0381

Магнитов С.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

В архитектуре компьютера существуют стандартные прерывания, за которыми закреплены определенные вектора прерываний. Вектор прерываний хранит адрес подпрограммы обработчика прерываний. При возникновении прерывания, аппаратура компьютера передает управление по соответствующему адресу вектора прерывания. Обработчик прерываний получает управление и выполняет соответствующие действия.

В лабораторной работе № 4 предлагается 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.

Постановка задачи.

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. Для того, чтобы проверить установку прерывания, можно поступить следующим образом. Прочитать адрес, записанный в векторе прерывания. Предположим, что этот адрес указывает на точку входа в установленный резидент. На определенном,

известном смещении в теле резидента располагается сигнатура, некоторый код, который идентифицирует резидент. Сравнив известное значение сигнатуры с реальным кодом, находящимся в резиденте, можно определить, установлен ли резидент. Если значения совпадают, то резидент установлен. Длину кода сигнатуры должна быть достаточной, чтобы сделать случайное совпадение маловероятным. 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.

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.

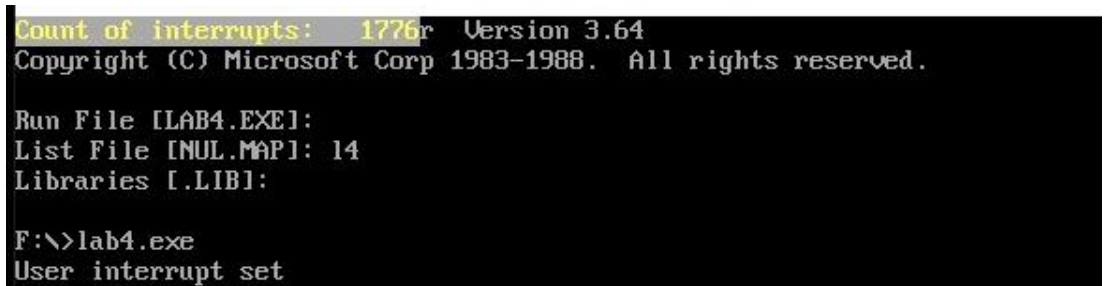
3. Запустите отлаженную программу еще раз и убедитесь, что программа определяет установленный обработчик прерываний. Полученные результаты поместите в отчет.

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 начале программы прописаны строки для выводов сообщений: `ALREADY_LOAD`, `WAS_STR`, `RESTORE_STR`. Также были написаны прерывания: `INTERRUPT` (загружается в память и выполняет накопление и вывод числа накопленных прерываний на экран), `WRITE_STRING` (печать строки), `LOAD_FLAG` (проверка наличия флага `/un`), `IS_LOAD` (проверка загрузки пользовательского прерывания в память), `LOAD_ITERRAPT` (сохранение первоначального прерывания и загрузка пользовательского прерывания в память), `UNLOAD_INTERRAPT` (выгрузка пользовательского прерывания из памяти и освобождение памяти с восстановлением первоначальных прерываний).

При запуске `lab4.exe` программа устанавливает пользовательское прерывание, после чего счётчик отображается на экране. Результат представлен на рисунке 1.



```
Count of interrupts: 1776r Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

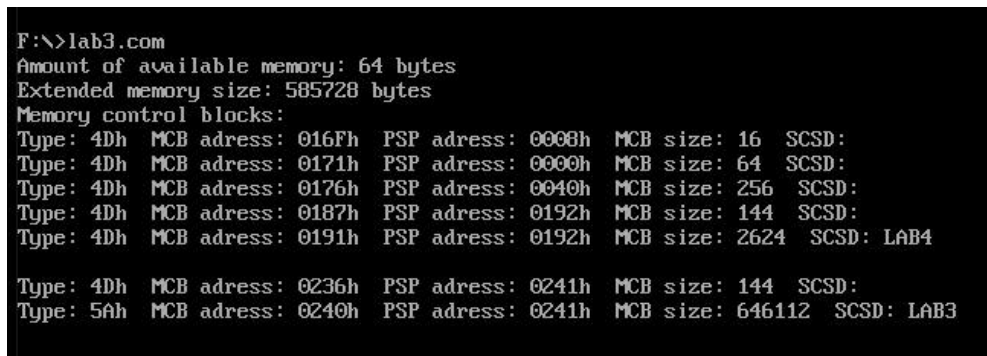
Run File [LAB4.EXE]:
List File [NUL.MAP]: 14
Libraries [.LIB]:

F:\>lab4.exe
User interrupt set
```

Рисунок 1 – Запуск `lab4.exe`

При запуске программы из лабораторной работы №3 счётчик продолжает работать, а программа из лабораторной работы №4 загружена в память.

Результаты представлены на рисунке 2.

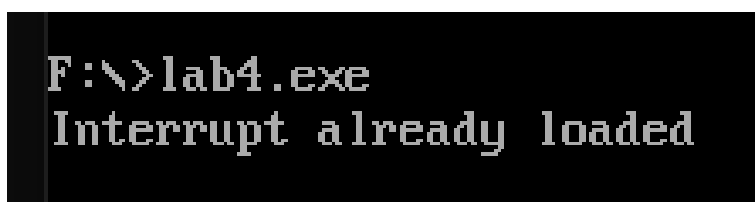


```
F:\>lab3.com
Amount of available memory: 64 bytes
Extended memory size: 585728 bytes
Memory control blocks:
Type: 4Dh MCB address: 016Fh PSP address: 0008h MCB size: 16 SCSD:
Type: 4Dh MCB address: 0171h PSP address: 0000h MCB size: 64 SCSD:
Type: 4Dh MCB address: 0176h PSP address: 0040h MCB size: 256 SCSD:
Type: 4Dh MCB address: 0187h PSP address: 0192h MCB size: 144 SCSD:
Type: 4Dh MCB address: 0191h PSP address: 0192h MCB size: 2624 SCSD: LAB4

Type: 4Dh MCB address: 0236h PSP address: 0241h MCB size: 144 SCSD:
Type: 5Ah MCB address: 0240h PSP address: 0241h MCB size: 646112 SCSD: LAB3
```

Рисунок 2 – Запуск `lab3.com`

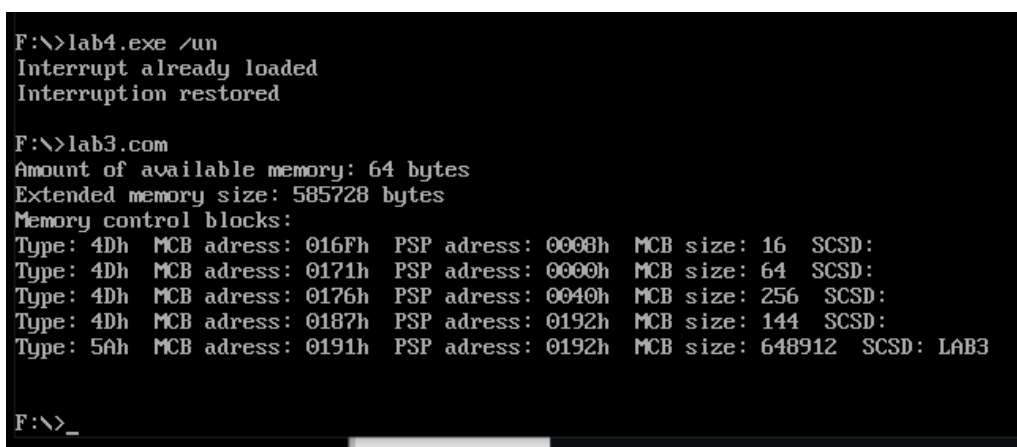
При повторном запуске программы выводится сообщение о том, что обработчик уже установлен. Результат представлен на рисунке 3.



```
F:\>lab4.exe
Interrupt already loaded
```

Рисунок 3 – Печать сообщения

При запуске программы с /un появляется сообщение о восстановлении стандартного прерывания, происходит очистка памяти, а счётчик прерываний исчезает. Результат представлен на рисунке 4.



```
F:\>lab4.exe /un
Interrupt already loaded
Interruption restored

F:\>lab3.com
Amount of available memory: 64 bytes
Extended memory size: 585728 bytes
Memory control blocks:
Type: 4Dh MCB adress: 016Fh PSP adress: 0008h MCB size: 16 SCSD:
Type: 4Dh MCB adress: 0171h PSP adress: 0000h MCB size: 64 SCSD:
Type: 4Dh MCB adress: 0176h PSP adress: 0040h MCB size: 256 SCSD:
Type: 4Dh MCB adress: 0187h PSP adress: 0192h MCB size: 144 SCSD:
Type: 5Ah MCB adress: 0191h PSP adress: 0192h MCB size: 648912 SCSD: LAB3

F:\>_
```

Рисунок 4 – Запуск программы

Вывод.

В ходе лабораторной работы был построен обработчик прерываний сигналов таймера.

Ответы на контрольные вопросы.

1. Как реализован механизм прерывания от часов?

Примерно каждые 55 миллисекунд по каждому тиксу часов подаётся сигнал аппаратными средствами и вызывается прерывание 1Ch.

2. Какого типа прерывания использовались в работе?

Int 10h и int 21h – программные прерывания. 1Ch – аппаратное прерывание.