

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. 7381

Габов Е.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2019

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 лабораторной работе номер 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.

Ход работы.

1. Описание данных

SIGNATURE – подпись пользовательского прерывания.

UNLOAD – флаг для выгрузки прерывания.

KEEP_IP – для сохранения регистра ip.

KEEP_CS – для сохранения регистра cs.

KEEP_ES – для сохранения регистра es.

KEEP_SP – для сохранения регистра sp.

KEEP_SS – для сохранения регистра ss.

OUT_STRING – строка для вывода сообщения о количестве прерываний.

ONES – разряд единиц.

DOZENS – разряд десятков.

HUNDREDS – разряд сотен.

THOUSANDS – разряд тысяч.

LOADED – сообщение о загрузке прерывания.

UNLOADED – сообщение о выгрузке прерывания.

LOADED_BEFORE – сообщение о том что прерывание уже установлено.

2. Описание функций

MY_INT – пользовательский обработчик прерываний

CALCULATE_INT – процедура подсчета количества вызванных прерываний.

OUTPUT_BP – процедура для вывода строки на консоль в указанную позицию.

CHECK – проверяет установлено ли пользовательское прерывание и введен ли параметр /up.

SET_MY_INT – устанавливает пользовательское прерывание.

SET_DEFAULT_INT – восстанавливает исходное прерывание.

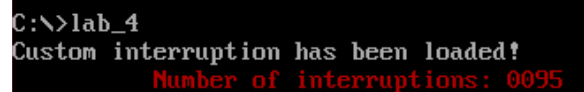
WRITE – процедура для вывода строки на консоль.

3. Последовательность действий, выполняемых программой:

Был написан и отлажен программный модуль типа EXE, который проверяет установлено ли пользовательское прерывание с вектором 1ch, если нет – то устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерывания и осуществляется выход по функции 4ch прерывания int 21h его, после этого программа завершается. Если программа запущена с параметром /up, то пользовательское прерывание выгружается, и загружается исходное прерывание.

4. Результат выполнения

1. Запуск программы



```
C:\>lab_4
Custom interruption has been loaded!
Number of interruptions: 0095
```

Рис.1 – Результат работы программы.

2. Результат работы вместе с программой из ЛР3.

```

C:\>lab_4
Custom interruption has been loaded!
      Number of interruptions: 0082
C:\>3
Available memory: 647888 bytes
Extended memory: 15360 Kbytes
List of Lists:
MCB Type: 4Dh - Address: 0008h - MCB Size: 16byte - Name:
MCB Type: 4Dh - Address: 0000h - MCB Size: 64byte - Name:
MCB Type: 4Dh - Address: 0040h - MCB Size: 256byte - Name:
MCB Type: 4Dh - Address: 0192h - MCB Size: 144byte - Name:
MCB Type: 4Dh - Address: 0192h - MCB Size: 848byte - Name: LAB_4
MCB Type: 4Dh - Address: 01D2h - MCB Size: 144byte - Name:
MCB Type: 4Dh - Address: 01D2h - MCB Size: 864byte - Name: 3
MCB Type: 4Dh - Address: 01D2h - MCB Size: 65536byte - Name: 3
MCB Type: 5Ah - Address: 0000h - MCB Size: 581456byte - Name: LAB_4

      Number of interruptions: 0122

```

Рис.2 – Размещение прерывания в памяти

3. Запуск программы еще раз

```

C:\>lab_4
Custom interruption has been loaded!
      Number of interruptions: 0247
C:\>lab_4
Custom interruption has already been loaded!
      Number of interruptions: 0287

```

Рис.3 – Результат повторной установки прерывания.

4. Запуск с ключом выгрузки.

```

C:\>lab_4
Custom interruption has been loaded!
      Number of interruptions: 0073
C:\>lab_4/un
Custom interruption has been unloaded!

```

Рис.4 – Результат работы программы с ключом выгрузки.

Вывод.

В ходе данной лабораторной работы был создан обработчик прерываний сигналов таймера.

Ответы на контрольные вопросы.

1. Как реализован механизм прерывания от часов?

Механизм прерывания часов реализован следующим образом:
Прерывание от часов (1ch) вызывается при каждом тике системного таймера (примерно 18 раз в секунду). Затем сохраняется состояние регистров, управление передается по смещению 70, затем выполняется код обработчика прерываний, далее восстанавливаются сохраненные регистры и выполняется возврат к управлению прерванной программой.

2. Какого типа прерывания использовались в работе?

В работе использованы аппаратные (1ch) и программные (10h, 21h) прерывания.