

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

Преподаватель

Птичкин С. А.

Ефремов М. А.

Санкт-Петербург

2021

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.

Ход работы.

- 1) Был написан и отлажен программный модуль типа EXE, который выполняет следующие функции:
 - Проверяет, установлено ли пользовательское прерывание с вектором 1Ch.
 - Устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний, если прерывание не установлено, и осуществляется выход по функции 4Ch прерывания int 21h.
 - Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания int 21h.
 - 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.

2) Далее программа была запущена:

```
C:\>lab4.exe
Interruption is loading.      Interrupts call count: 0213
C:\>
```

Было установлено что резидентный обработчик прерывания 1Ch установлен. Чтобы проверить размещение прерывания в памяти была запущена программа из 3-й лабораторной работы, которая отображает карту памяти в виде списка блоков MCB. Из скриншота видно, что MCB-блок программы располагается 5-м по порядку:

```
C:\>lab3.com
Available memory: 648128b
Extended memory: 15360kb
List of MCB:
MCB type: 4Dh    PSP address: 0000h    Size:      16 b
MCB type: 4Dh    PSP address: 0000h    Size:       64 b
MCB type: 4Dh    PSP address: 0040h    Size:     256 b
MCB type: 4Dh    PSP address: 0192h    Size:     144 b
MCB type: 4Dh    PSP address: 0192h    Size:     608 b      LAB4
MCB type: 4Dh    PSP address: 01C3h    Size:     144 b
MCB type: 5Ah    PSP address: 01C3h    Size:   648128 b      LAB3
C:\>
```

3) Затем программа вновь была запущена. Программа определила установленный обработчик прерываний, и на экран было выведено сообщение, что прерывание уже загружено в память:

```
C:\>lab4.exe
Interruption already load.      Interrupts call count: 1751
C:\>
```

4) Потом программа была запущена с ключом выгрузки, параметром /un. В результате резидентный обработчик прерывания был выгружен, память занятая резидентом освобождена, а счётчик перестал отображаться на экране. Было выведено соответствующее сообщение:

```
C:\>lab4.exe /un
Interruption was restored.
C:\>_
```

- 5) Для проверки освобождения памяти, занятой резидентом была снова запущена программа из 3-й лабораторной работы, которая показала, что MCB-блок программы lab4.exe уже отсутствует в списке:

```

Interruption was restored.

C:\>lab3.com
Available memory: 648912b
Extended memory: 15360kb
List of MCB:
MCB type: 4Dh   PSP address: 0008h   Size:      16 b
MCB type: 4Dh   PSP address: 0000h   Size:      64 b
MCB type: 4Dh   PSP address: 0040h   Size:     256 b
MCB type: 4Dh   PSP address: 0192h   Size:     144 b
MCB type: 5Ah   PSP address: 0192h   Size:   648912 b      LAB3
C:\>_

```

Описание переменных и функций.

Названия переменных	Описание
psp_adr_1	Хранит предыдущее значение ES до того, как программа была оставлена резидентной в памяти.
keep_cs	Хранит сегмент прерывания.
keep_ip	Хранит смещение прерывания.
interruption_set	Хранит количество вызванных прерываний.
count	Хранит строку: Interrupts call count: 0000 \$
not_loaded	Хранит строку: Interruption not loaded.
restored	Хранит строку: Interruption was restored.
loaded	Хранит строку: Interruption already load.
load_process	Хранит строку: Interruption is loading.

Названия функций	Описание
MY_INTERRUPT	Обработчик прерываний, который выводит количество вызванных прерываний.

IS_SET	Проверяет установлен ли написанный вектор прерывания.
IS_LOAD	Загрузка или выгрузка (проверяет параметр un)
LOAD_INTERRUPTION	Устанавливает новые обработчики прерывания, с помощью функции 25h прерывания int 21h.
UNLOAD_INTERRUPTION	Восстанавливает сохранённые обработчики прерываний и выгружает резидентную функцию.
PRINT	Выводит строку на экран.
MAIN	Главная процедура.

Ответы на контрольные вопросы.

1) Как реализован механизм прерывания от часов?

1. Поступает сигнал прерывания.
2. Запоминаются значения регистров.
3. По номеру источника прерывания в таблице векторов определяется смещение.
4. Сохраняются адреса 2 байта в IP и 2 байта в CS.
5. Выполняется прерывание по сохранённому адресу.
6. Данные прерванного процесса восстанавливаются.
7. Возвращается управление прерванной программе.

2) Какого типа прерывания использовались в работе?

1. Аппаратные прерывания.
2. Прерывания функций DOS(21h).
3. Прерывания функций BIOS(10h).

Вывод.

Была написана программа обработчика прерывания от сигналов таймера, изучена установка резидентной программы в память и её выгрузка.