

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

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

Студент гр. 0381

Соколов Д. В.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

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 по соответствующему адресу вектора прерывания. Обработчик прерываний получает управление и выполняет соответствующие действия.

В лабораторной работе 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.

Постановка задачи.

Требуется написать и отладить 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 осуществляет 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.

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 2 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.

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 3 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.

Выполнение работы.

Шаг 1. Был написан и отлажен программный модуль типа .EXE, который выполняет, выполняет требуемый функционал.

Шаг 2. Программа была отлажена и запущена. Проверено размещение прерывания в памяти (5ая строчка в таблице MCB).

```
S:\>lab_4
Interruption is loading.
Timer n0015
S:\>lab_32

Size of free memory:      578768 byte
Size of extended memory: 15728640 byte
MCB: 4Dh  Address: 016Fh  Owner: MS DOS  Area size: 16 byte
SC/SD:
MCB: 4Dh  Address: 0171h  Owner: Free    Area size: 64 byte
SC/SD:
MCB: 4Dh  Address: 0176h  Owner: 0040h   Area size: 256 byte
SC/SD:
MCB: 4Dh  Address: 0187h  Owner: 0192h   Area size: 144 byte
SC/SD:
MCB: 4Dh  Address: 0191h  Owner: 0192h   Area size: 4416 byte
SC/SD: LAB_4
MCB: 4Dh  Address: 02A6h  Owner: 02B1h   Area size: 144 byte
SC/SD:
MCB: 4Dh  Address: 02B0h  Owner: 02B1h   Area size: 65536 byte
SC/SD: LAB_32
MCB: 5Ah  Address: 12B1h  Owner: Free    Area size: 578768 byte
SC/SD:
```

Рисунок 1. Прерывание загружено в память

Шаг 3. Программа была повторно запущена, чтобы удостовериться, что программа определяет установленный обработчик прерываний.

```
Size of fTimer now56      578768 byte
Size of extended memory:  15728640 byte
MCB: 4Dh Address: 016Fh Owner: MS DOS Area size: 16 byte
SC/SD:
MCB: 4Dh Address: 0171h Owner: Free Area size: 64 byte
SC/SD:
MCB: 4Dh Address: 0176h Owner: 0040h Area size: 256 byte
SC/SD:
MCB: 4Dh Address: 0187h Owner: 0192h Area size: 144 byte
SC/SD:
MCB: 4Dh Address: 0191h Owner: 0192h Area size: 4416 byte
SC/SD: LAB_4
MCB: 4Dh Address: 02A6h Owner: 02B1h Area size: 144 byte
SC/SD:
MCB: 4Dh Address: 02B0h Owner: 02B1h Area size: 65536 byte
SC/SD: LAB_32
MCB: 5Ah Address: 12B1h Owner: Free Area size: 578768 byte
SC/SD:
S:\>lab_4
Interruption has already loaded.
```

Рисунок 2. Повторная загрузка прерывания

Шаг 4. Программа была запущена с ключом выгрузки, чтобы убедиться, что резидентный обработчик прерывания выгружен и память, занятая резидентом освобождена.

```
S:\>lab_4 /un
Interruption was unloaded.
S:\>lab_32

Size of free memory:      583360 byte
Size of extended memory:  15728640 byte
MCB: 4Dh Address: 016Fh Owner: MS DOS Area size: 16 byte
SC/SD:
MCB: 4Dh Address: 0171h Owner: Free Area size: 64 byte
SC/SD:
MCB: 4Dh Address: 0176h Owner: 0040h Area size: 256 byte
SC/SD:
MCB: 4Dh Address: 0187h Owner: 0192h Area size: 144 byte
SC/SD:
MCB: 4Dh Address: 0191h Owner: 0192h Area size: 65536 byte
SC/SD: LAB_32
MCB: 5Ah Address: 1192h Owner: Free Area size: 583360 byte
SC/SD:
```

Рисунок 3. Выгрузка прерывания

```

S:\>lab_4 /un
Interruption did not load.

S:\>lab_32

Size of free memory:      583360 byte
Size of extended memory:  15728640 byte
MCB: 4Dh  Address: 016Fh  Owner: MS DOS  Area size: 16 byte
SC/SD:
MCB: 4Dh  Address: 0171h  Owner: Free    Area size: 64 byte
SC/SD:
MCB: 4Dh  Address: 0176h  Owner: 0040h   Area size: 256 byte
SC/SD:
MCB: 4Dh  Address: 0187h  Owner: 0192h   Area size: 144 byte
SC/SD:
MCB: 4Dh  Address: 0191h  Owner: 0192h   Area size: 65536 byte
SC/SD: LAB_32
MCB: 5Ah  Address: 1192h  Owner: Free    Area size: 583360 byte
SC/SD:

```

Рисунок 4 – повторная выгрузка прерывания

Выводы.

Построен собственный обработчик прерываний сигналов таймера. Получены дополнительные знания о работе с памятью (резидентный обработчик может быть загружен и выгружен из памяти).

ПРИЛОЖЕНИЕ А

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как реализован механизм прерывания от часов?

Принимается сигнал прерывания (приходит примерно каждые 54 мс), запоминаются содержимые регистров, по номеру источника прерывания в таблице векторов определяется смещение, запоминается адрес 2 байта в IP и 2 байта в CS. Далее выполняется прерывание по сохранённому адресу и далее восстанавливается информация прерванного процесса и управление возвращается прерванной программе.

2. Какие прерывания использовались в работе?

- Int 10h – видео сервис BIOS
- Int 21h – сервисы DOS
- Пользовательское прерывание с вектором 1ch int 21h