**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 |  | Колованов Р.А. |
| Преподаватель |  | 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.

**Функции и структуры данных.**

Разработанная программа использует следующие функции и структуры данных:

|  |  |
| --- | --- |
| Название процедуры | Предназначение процедуры |
| INTERRUPT\_HANDLER | Обработчик прерываний. При каждом вызове инкрементирует переменную, содержащую количество вызовов прерывания, и выводит ее на экран. |
| PRINT\_STRING\_ES\_BP | Печатает на экран строку при помощи функции 13h прерывания int 10h. В регистр ES заносится адрес сегмента, в котором находится строка, а в BP – сдвиг относительно этого сегмента. |
| SET\_CURSOR | Устанавливает курсору новую позицию. Позиция берется из регистра BX. |
| GET\_CURSOR | Возвращает текущую позицию курсора в регистре BX. |
| GET\_INTERRUPT\_HANDLER | Получает адрес сегмента (записывается в ES) и сдвиг (записывается в BX), по которому находится обработчик прерываний 1Ch. |
| RESTORE\_INTERRUPT\_HANDLER | Восстанавливает старый обработчик прерываний 1Ch при помощи функции 25h прерывания 21h и очищает выделенную под новый резидентный обработчик прерываний память при помощи функции 49h прерывания 21h. |
| SET\_INTERRUPT\_HANDLER | Устанавливает новый обработчик прерываний 1Ch, оставляя его резидентным в памяти, при помощи функции 31h прерывания 21h и выходит в DOS. |
| CHECK\_INTERRUPT\_HANDLER | Проверяет, установлен ли новый обработчик прерываний 1Ch. Проверка выполняется при помощи сигнатуры прерывания, записанной на определенном смещении от начала обработчика. Если обработчик установлен, то в AX будет записано значение 1, иначе – 0. |
| PRINT | Вызывает функцию вывода строки на экран (функция 09h прерывания 21h). |

**Ход работы.**

Для начала был написан текст исходного EXE модуля lab4.asm. Далее при помощи транслятора MASM.EXE и компоновщика LINK.EXE был скомпилирован EXE модуль lab4.com с генерацией файла листинга и карты памяти, после чего была осуществлена отладка. Программа выполняет следующие функции:

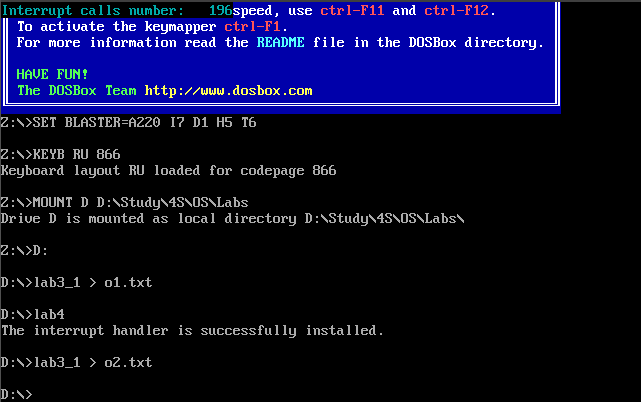
* Проверяет, установлено ли пользовательское прерывание с вектором 1Ch.
* Устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний, если прерывание не установлено, и осуществляется выход о функции 4Ch прерывания int 21h.
* Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания int 21h.
* Выгрузка прерывания о 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.

Далее программа была протестирована. Для начала проверялась установка нового резидентного обработчика прерывания 1Ch. Для проверки того, что обработчик прерываний остался резидентным в памяти, использовалась программа LAB3\_1.COM, которая осуществляет вывод списка блоков MCB. Были получены следующие результаты:

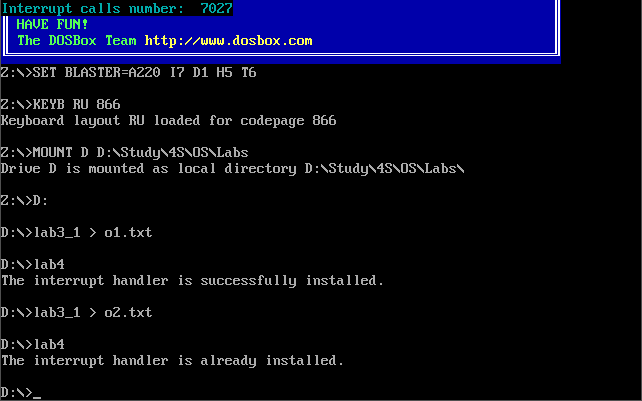
|  |
| --- |
| **Список блоков MCB до установки обработчика.**  The size of the available memory: 633 kilobytes 720 bytes  The size of the extended memory: 15360 kilobytes  MCB address: 016F  MCB type: 4D  Block owner: MS DOS  Block size: 0 kilobytes 16 bytes  Reserved:  MCB address: 0171  MCB type: 4D  Block owner: Free  Block size: 0 kilobytes 64 bytes  Reserved:  MCB address: 0176  MCB type: 4D  Block owner: 0040  Block size: 0 kilobytes 256 bytes  Reserved:  MCB address: 0187  MCB type: 4D  Block owner: 0192  Block size: 0 kilobytes 144 bytes  Reserved:  MCB address: 0191  MCB type: 5A  Block owner: 0192  Block size: 633 kilobytes 720 bytes  Reserved: LAB3\_1 |

|  |
| --- |
| **Список блоков MCB после установки обработчика.**  The size of the available memory: 633 kilobytes 80 bytes  The size of the extended memory: 15360 kilobytes  MCB address: 016F  MCB type: 4D  Block owner: MS DOS  Block size: 0 kilobytes 16 bytes  Reserved:  MCB address: 0171  MCB type: 4D  Block owner: Free  Block size: 0 kilobytes 64 bytes  Reserved:  MCB address: 0176  MCB type: 4D  Block owner: 0040  Block size: 0 kilobytes 256 bytes  Reserved:  MCB address: 0187  MCB type: 4D  Block owner: 0192  Block size: 0 kilobytes 144 bytes  Reserved:  MCB address: 0191  MCB type: 4D  Block owner: 0192  Block size: 0 kilobytes 464 bytes  Reserved: LAB4  MCB address: 01AF  MCB type: 4D  Block owner: 01BA  Block size: 0 kilobytes 144 bytes  Reserved: єg  MCB address: 01B9  MCB type: 5A  Block owner: 01BA  Block size: 633 kilobytes 80 bytes  Reserved: >‚ LAB3\_1 |

Результаты работы обработчика прерывания 1Ch:

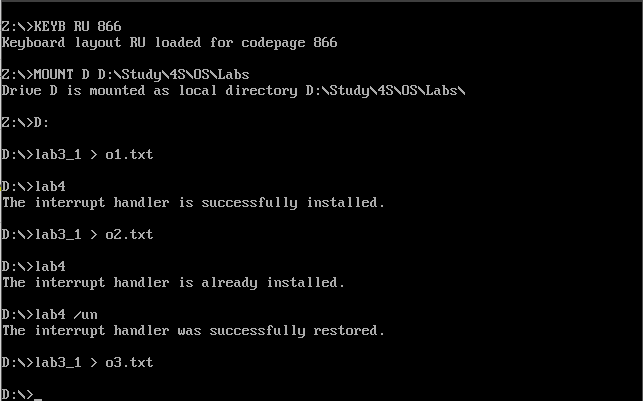


Далее проверялось определение программой установленного обработчика прерывания 1Ch. Были получены следующие результаты:



В конце проверялись выгрузка установленного обработчика прерывания 1Ch и очистка памяти, занимаемой резидентным обработчиком прерываний. Для проверки того, что обработчик прерываний был выгружен, использовалась программа LAB3\_1.COM, которая осуществляет вывод списка блоков MCB. Были получены следующие результаты:

|  |
| --- |
| **Список блоков MCB после выгрузки обработчика.**  The size of the available memory: 633 kilobytes 720 bytes  The size of the extended memory: 15360 kilobytes  MCB address: 016F  MCB type: 4D  Block owner: MS DOS  Block size: 0 kilobytes 16 bytes  Reserved:  MCB address: 0171  MCB type: 4D  Block owner: Free  Block size: 0 kilobytes 64 bytes  Reserved:  MCB address: 0176  MCB type: 4D  Block owner: 0040  Block size: 0 kilobytes 256 bytes  Reserved:  MCB address: 0187  MCB type: 4D  Block owner: 0192  Block size: 0 kilobytes 144 bytes  Reserved:  MCB address: 0191  MCB type: 5A  Block owner: 0192  Block size: 633 kilobytes 720 bytes  Reserved: LAB3\_1 |



**Результаты исследования проблем.**

1. Как реализован механизм прерывания от часов?

* При возникновении сигнала аппаратного прерывания от часов, происходит первичное аппаратное распознавание типа прерывания. Если прерывания данного типа в настоящий момент запрещены, то процессор продолжает поддерживать естественный ход выполнения команд, иначе, в зависимости от поступившей информации происходит автоматический вызов процедуры обработки прерывания, адрес которой находится в таблице векторов прерываний;
* Автоматически сохраняется некоторая часть контекста прерванного процесса, который позволит ядру возобновить его исполнение после обработки прерывания;
* Временно запрещаются прерывания данного типа, чтобы не было очереди потоков одной и той же процедуры;
* После обработки прерываний ядром ОС, контекст прерванного процесса восстанавливается, и его работа возобновляется с прерванного места; также снимается блокировка прерываний данного типа.

1. Какого типа прерывания использовались в работе?

Использовались следующие типы прерываний:

* Аппаратные (прерывание от часов – 1Ch);
* Программные (прерывания, вызываемые при помощи команды int – 21h)

**Заключение.**

Был разработан резидентный обработчик прерываний сигналов таймера.