

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. 0381

Просекин Т.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

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.

Задание.

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.

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

Выполнение работы.

Таблица 1 — Процедуры в программе.

Процедура	Описание
user_interrupt	Пользовательское прерывание
check_key	Проверка наличия ключа выгрузки
unload_int	Выгрузка прерывания
cout	Вывод строки на экран
loaded_check	Процедура проверяет загружено ли прерывание в память
load_int	Загрузка прерывания в память
Main	Процедура запускающая программу

Данные объявленные в программе:

unload_key db 0

in_use db 0

int_not_loaded db 'Interruption is not loaded.', 10, 13, '\$'

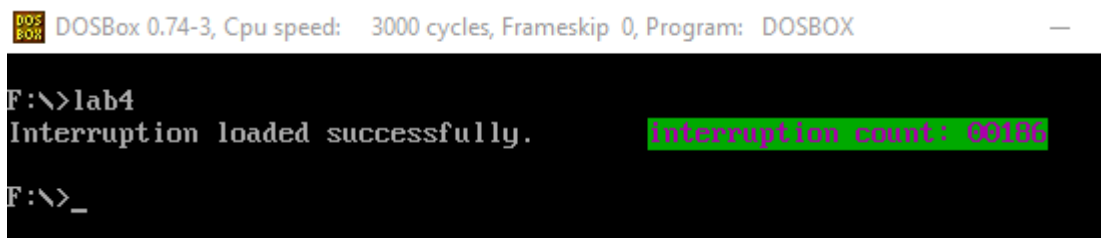
unload_inted db 'Interruption unloaded successfully.', 10, 13, '\$'

int_loaded db 'Interruption loaded successfully.', 10, 13, '\$'

int_in_use db 'Interruption is loaded already.', 10, 13, '\$'

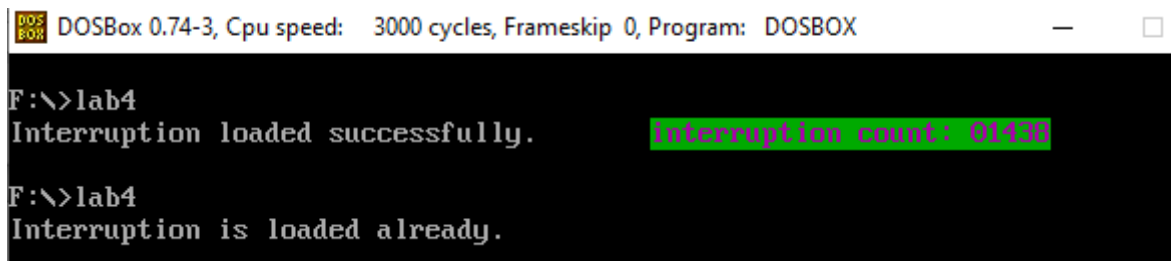
Программа проверяет наличие ключа выгрузки. Если он есть, то выполняется процедура выгрузки пользовательского прерывания из памяти. Затем процедура проверяет память на то, загружено ли в него прерывание. Если так, то выводится сообщение об успешной выгрузке и выгружается прерывание, иначе выводится соответствующее сообщение. Если флага нет, то проверяется загружено ли в память прерывание. Если так, то выводится соответствующее сообщение, иначе прерывание записывается в память.

Тестирование:




```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
F:\>lab4
Interruption loaded successfully.      interruption count: 00186
F:\>_
```

Рисунок 1. Загрузка прерывания.



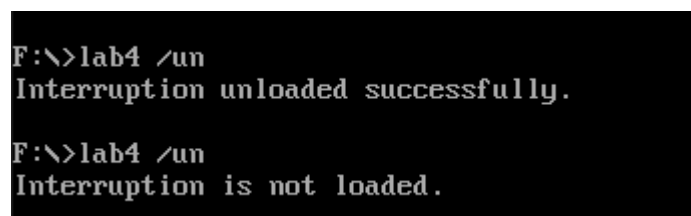
```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
F:\>lab4
Interruption loaded successfully.      interruption count: 01438
F:\>lab4
Interruption is loaded already.
```

Рисунок 2. Повторная загрузка прерывания



```
F:\>lab4 /un
Interruption unloaded successfully.
```

Рисунок 3. Выгрузка прерывания из памяти



```
F:\>lab4 /un
Interruption unloaded successfully.
F:\>lab4 /un
Interruption is not loaded.
```

Рисунок 4. Повторная выгрузка прерывания

ОТВЕТЫ НА ВОПРОСЫ

1. Как реализован механизм прерывания от часов?

Прерывание 1Ch вызывается каждые 55 миллисекунд (по каждому тикку аппаратных часов). Происходит следующее:

- 1) Сохраняется состояние регистров.
- 2) Определяется источник прерывания по номеру источника прерывания определяется смещение в таблице векторов прерываний.
- 3) Первые 2 байта помещаются в IP, вторые 2 байта в CS.
- 4) Передаётся управление по адресу CS:IP, то есть запускается обработчик прерывания.
- 5) Обработка прерывания.
- 6) Возврат управления прерванной программе.

2. Какого типа прерывания используются в работе?

В работе используются программные прерывания 10h, 21h и аппаратное прерывание 1Ch.

Выводы.

Был изучен принцип работы стандартного обработчика прерываний.
Был построен обработчик прерываний таймера.