

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №5**  
**по дисциплине «Операционные системы»**  
**Тема: «Сопряжение стандартного и пользовательского обработчиков**  
**прерываний»**

Студентка гр. 8381

Бердникова А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

## **Цель работы**

Исследование возможности встраивания пользовательского обработчика прерываний в стандартный обработчик от клавиатуры. Пользовательский обработчик прерывания получает управление по прерыванию (int 09h) при нажатии клавиши на клавиатуре. Он обрабатывает скан-код и осуществляет определенные действия, если скан-код совпадает с определенными кодами, которые он должен обрабатывать. Если скан-код не совпадает с этими кодами, то управление передается стандартному прерыванию.

## **Основные теоретические положения.**

Клавиатура содержит микропроцессор, который воспринимает каждое нажатие на клавишу и посылает скан-код в порт микросхемы интерфейса с периферией. Когда скан-код поступает в порт, то вызывается аппаратное прерывание клавиатуры (INT 09H). Процедура обработки этого прерывания считывает номер клавиши из порта 60H, преобразует номер клавиши в соответствующий код, выполняет установку флагов в байтах состояния, загружает номер клавиши и полученный код в буфер клавиатуры.

В прерывании клавиатуры можно выделить три основных шага:

1. Прочитать скан-код и послать клавиатуре подтверждающий сигнал.
2. Преобразовать скан-код в номер кода или в установку регистра статуса клавиш-переключателей.
3. Поместить код клавиши в буфер клавиатуры.

Текущее содержимое буфера клавиатуры определяется указателями на начало и конец записи. Расположение в памяти необходимых данных представлено в таблице

1.

Таблица 1 – Буфер клавиатуры

Адрес в памяти	Размер в байтах	Содержимое
0040:001A	2	Адрес начала буфера клавиатуры
0040:001C	2	Адрес конца буфера клавиатуры
0040:001E	32	Буфер клавиатуры
0040:0017	2	Байты состояния

Флаги в байтах состояния устанавливаются в 1, если нажата соответствующая клавиша или установлен режим. Соответствие флагов и клавиш показано ниже на рис. 1.

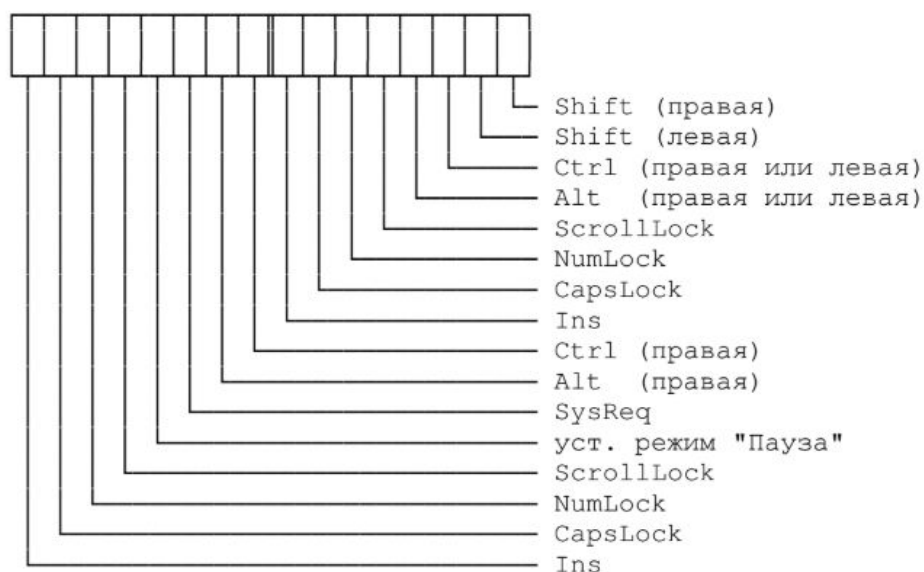


Рисунок 1 – Соответствие флагов и клавиш

## Сведения о программе

Процедуры программы:

1. INTERRUPT – процедура с прерыванием
2. INTERRUPT\_HANDLER\_REPLACEMENT - Процедура замены обработчика прерывания
3. LOAD\_RESIDENT - Процедура загрузки резидента
4. REMOVE\_RESIDENT - Процедура удаления резидента из памяти
5. FREE\_MEMORY - Процедура освобождения памяти

## Ход работы.

1) Был написан и отлажен программный модуль lab5.exe, который выполняет следующие функции:

1. Проверяет, установлено ли пользовательское прерывание с вектором 09h.

2. Устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний, если прерывание не установлено.

3. Если прерывание установлено, то выводится соответствующее сообщение.

4. Выгрузка прерывания по соответствующему значению параметра в командной строке /un.

2) Был запущен DOSBox.

3) В файле LAB5.asm была написана требуемая программа.

4) Была запущена программа LAB5.exe. Структура памяти до загрузки резидента в память (выведено с помощью программного модуля lab3.com из третьей лабораторной работы):

```
Amount of available memory (in bytes): 648912
Amount of extended memory (in kilobytes): 15360
-----
MS DOS area
Memory size (in bytes): 16
Last bytes:
-----
Free area
Memory size (in bytes): 64
Last bytes:
-----
Unknown possessor: 0040
Memory size (in bytes): 256
Last bytes:
-----
Unknown possessor: 0192
Memory size (in bytes): 144
Last bytes:
-----
Unknown possessor: 0192
Memory size (in bytes): 648912
Last bytes: LAB3
```

Рисунок 2 - Структура памяти до загрузки резидента в память

```
C:\>lab5.exe
Resident has been load!
```

Рисунок 3 - Результат выполнения lab5.exe модуля

- 5) Было проверено нахождение прерывания в памяти, для этого в консоль была введена строка “pass 012310”. Процедура прерывания заменила цифры '0' и '1' на 'x' , буквы 's' на 'c'.

```
C:\>lab5.exe
Resident has been load!

C:\>pass xx23xcc_
```

Рисунок 4 - Ввод проверочной строки

```
Free area
Memory size (in bytes): 64
Last bytes:
-----
Unknown possessor: 0040
Memory size (in bytes): 256
Last bytes:
-----
Unknown possessor: 0192
Memory size (in bytes): 144
Last bytes:
-----
Unknown possessor: 0192
Memory size (in bytes): 960
Last bytes: LAB5
-----
Unknown possessor: 01D9
Memory size (in bytes): 144
Last bytes:
-----
Unknown possessor: 01D9
Memory size (in bytes): 647776
Last bytes: LAB3
```

Рисунок 5 - Структура памяти после загрузки резидента в память

- 6) Программный модуль lab5.exe был запущен повторно, было выведено сообщение о нахождении резидента в памяти

```
C:\>lab5.exe
Resident still in memory!
```

Рисунок 6 - Сообщение о нахождении резидента в памяти

```

Free area
Memory size (in bytes): 64
Last bytes:
-----
Unknown possessor: 0040
Memory size (in bytes): 256
Last bytes:
-----
Unknown possessor: 0192
Memory size (in bytes): 144
Last bytes:
-----
Unknown possessor: 0192
Memory size (in bytes): 960
Last bytes: LAB5
-----
Unknown possessor: 01D9
Memory size (in bytes): 144
Last bytes:
-----
Unknown possessor: 01D9
Memory size (in bytes): 647776
Last bytes: LAB3

```

Рисунок 7 - Структура памяти после повторной загрузки резидента в память

- 7) Программный модуль lab5.exe был запущен с ключом выгрузки.

```

C:\>lab5.exe /un
Resident has been removed!

```

Рисунок 8 - Результат запуска модуля с ключом выгрузки

```

Amount of available memory (in bytes): 648912
Amount of extended memory (in kilobytes): 15360
-----
MS DOS area
Memory size (in bytes): 16
Last bytes:
-----
Free area
Memory size (in bytes): 64
Last bytes:
-----
Unknown possessor: 0040
Memory size (in bytes): 256
Last bytes:
-----
Unknown possessor: 0192
Memory size (in bytes): 144
Last bytes:
-----
Unknown possessor: 0192
Memory size (in bytes): 648912
Last bytes: LAB3

```

Рисунок 9 - Структура памяти после выгрузки резидента из памяти

## **Ответы на контрольные вопросы:**

### **Лабораторная работа №5**

1. Какого типа прерывания использовались в работе?

Ответ: В лабораторной работе был реализован обработчик прерываний, обрабатывающий прерывания от нажатия клавиш на клавиатуре. Также в коде программы использовались синхронные прерывания для обращения к функциям DOS.

2. Чем отличается скан код от кода ASCII?

Ответ: Скан-код в отличие от кода ASCII соответствует каждой определенной клавише клавиатуры и позволяет ее распознать с помощью драйвера клавиатуры. А код ASCII – это код конкретного символа, закрепленного за клавишей, который позволяет стандартизировать обмен данными между устройствами компьютера.

### **Выводы.**

В ходе выполнения лабораторной работы была исследована возможность встраивать пользовательский обработчик прерываний в стандартный обработчик от клавиатуры, который при нажатии на заданные клавиши вызывает пользовательский обработчик, а при нажатии на остальные вызывает стандартный.