

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

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

Студентка гр. 8381

Гречко В.Д.

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 определённые действия, если скан-код совпадает с определёнными кодами, которые он должен обрабатывать. Если скан-код не совпадает с этими кодами, то управление передаётся стандартному прерыванию.

### **Необходимые сведения для составления программы.**

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.

В момент вызова прерывания скан-код будет находиться в порте 60h. Поэтому сначала надо этот код прочитать командой IN и сохранить на стеке. Затем используется порт 61h, чтобы быстро послать сигнал подтверждения микропроцессору клавиатуры. Надо просто установить бит 7 в 1, а затем сразу

изменить его назад в 0. Заметим, что бит 6 порта 61h управляет сигналом часов клавиатуры. Он всегда должен быть установлен в 1, иначе клавиатура будет выключена. Эти адреса портов применимы и к АТ, хотя он и не имеет микросхемы интерфейса с периферией 8255.

Сначала скан-код анализируется на предмет того, была ли клавиша нажата (код нажатия) или отпущена (код освобождения). Код освобождения состоит из двух байтов: сначала 0F0h, а затем скан-код. Все коды освобождения отбрасываются, кроме случая клавиш-переключателей, для которых делаются соответствующие изменения в байтах их статуса. С другой стороны, все коды нажатия обрабатываются. При этом опять могут изменяться байты статуса клавиш-переключателей. В случае же символьных кодов, надо проверять байты статуса, чтобы определить, например, что скан-код 30 соответствует нижнему или верхнему регистру буквы А. После того как введенный символ идентифицирован, процедура ввода с клавиатуры должна найти соответствующий ему код ASCII или расширенный код. Приведенный пример слишком короток, чтобы рассмотреть все случаи. В общем случае скан-коды сопоставляются элементам таблицы данных, которая анализируется инструкцией XLAT. XLAT принимает в AL число от 0 до 255, а возвращает в AL 1-байтное значение из 256-байтной таблицы, на которую указывает DS:BX. Таблица может находиться в сегменте данных. Если в AL находился скан-код 30, то туда будет помещен из таблицы байт номер 30 (31-й байт, так как отсчет начинается с нуля). Этот байт в таблице должен быть установлен равным 97, давая код ASCII для "a". Конечно для получения заглавной А нужна другая таблица, к которой обращение будет происходить, если статус сдвига установлен. Или заглавные буквы могут храниться в другой части той же таблицы, но в этом случае к скан-коду надо будет добавлять смещение, определяемое статусом клавиш-переключателей.

Номера кодов должны быть помещены в буфер клавиатуры. Процедура должна сначала проверить, имеется ли в буфере место для следующего символа. Буфер устроен как циклическая очередь. Ячейка памяти 0040:001A

содержит указатель на голову буфера, а 0040:001С - указатель на хвост. Эти словные указатели дают смещение в области данных BIOS (которая начинается в сегменте 40h) и находятся в диапазоне от 30 до 60. Новые символы вставляются в ячейки буфера с более старшими адресами, а когда достигнута верхняя граница, то следующий символ переносится в нижний конец буфера. Когда буфер полон, то указатель хвоста на 2 меньше указателя на голову - кроме случая, когда указатель на голову равен 30 (начало области буфера), а в этом случае буфер полон, когда указатель хвоста равен 60. Для вставки символа в буфер, надо поместить его в позицию, на которую указывает хвост буфера и затем увеличить указатель хвоста на 2; если указатель хвоста был равен 60, то надо изменить его значение на 30.

### **Ход работы.**

**Шаг 1.** Для выполнения лабораторной работы был написан и отлажен программный модуль типа .EXE, который выполняет такие же функции как в программе ЛР 4, а именно:

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.

Выгрузка прерывания по соответствующему значению параметра в командной строке /un. Выгрузка прерывания состоит в восстановлении стандартного вектора прерываний и освобождения памяти, занимаемой резидентом. Затем осуществляется выход по функции 4Ch прерывания int 21h.

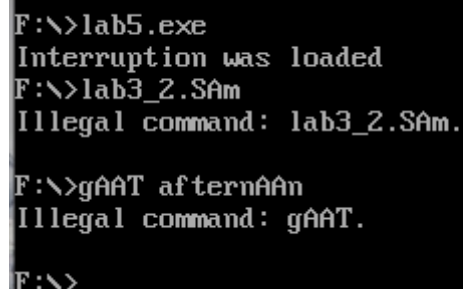
Для того, чтобы проверить установку прерывания, можно поступить следующим образом. Прочитать адрес, записанный в векторе прерывания. Предположим, что этот адрес указывает на точку входа в установленный резидент. На определённом, известном смещении в теле резидента

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. Длина кода сигнатуры должна быть достаточной, чтобы сделать случайное совпадение маловероятным.

Программа должна содержать код устанавливаемого прерывания в виде удалённой процедуры. Этот код будет работать после установки при возникновении прерывания. Он должен выполнять следующие функции:

1. Сохранить значение регистров в стеке при входе и восстановить их при выходе.
2. При выполнении тела процедуры анализируется скан-код.
3. Если этот код совпадает с одним из заданных, то требуемый код записывается в буфер клавиатуры.
4. Если этот код не совпадает ни с одним из заданных, то осуществляется передача управления стандартному обработчику прерывания.

**Шаг 2.** Запуск отлаженной программы. Как видно из рис. 1 резидентный обработчик прерывания 09h установлен. Работа прерывания проверяется введением различных символов, обрабатываемых установленным обработчиком и стандартным обработчиком. В данном случае написанный обработчик заменяет символы “С”, “О” и “D” на “S”, “А” и “Т” соответственно. Для тестирования использовалась фраза good afternoon.



```
F:\>lab5.exe
Interruption was loaded
F:\>lab3_2.SAm
Illegal command: lab3_2.SAm.

F:\>gAAT afternAAAn
Illegal command: gAAT.

F:\>
```

*Рисунок 1 — Вывод программы lab5.exe после первого запуска*

**Шаг 3.** Также необходимо проверить размещение прерывания в памяти. Для этого была запущена программа ЛР 3, которая отображает карту памяти в

виде списка блоков MCB. Полученные результаты представлены на рис. 2. Как видно из рисунка, процедура прерывания осталась резидентной в памяти.

```
F:\>lab5.exe
Interruption was loaded
F:\>lab3_2.com
Available memory: 640 kbytes
Extended memory: 15360 kbytes
MCBs:
MCB number 1
Block is occupied by MS DOS, size = 16 bytes; occupied by: no info
MCB number 2
Block is free, size = 64 bytes; occupied by: no info
MCB number 3
Block is owned by PSP = 0040, size = 256 bytes; occupied by: no info
MCB number 4
Block is owned by PSP = 0192, size = 144 bytes; occupied by: no info
MCB number 5
Block is owned by PSP = 0192, size = 704 bytes; occupied by: LAB5
MCB number 6
Block is owned by PSP = 01C9, size = 144 bytes; occupied by: no info
MCB number 7
Block is owned by PSP = 01C9, size = 1072 bytes; occupied by: LAB3_2
MCB number 8
Block is free, size = 646944 bytes; occupied by: no info
```

*Рисунок 2 — Вывод программы lab3\_2.com после выполнения lab5.exe*

**Шаг 4.** Запуск отлаженной программы и определение, что программа определяет установленный обработчик прерываний. Полученные результаты представлены на рис. 3. На рисунке показано, что при повторном запуске программа выводит сообщение о том, что резидентный обработчик уже загружен.

```
F:\>lab5.exe
Interruption was loaded
F:\>lab5.exe
Interruption is loaded
F:\>
```

*Рисунок 3 — Вывод программы lab5.exe при повторном запуске*

**Шаг 5.** Запуск отлаженной программы с ключом выгрузки и убедитесь, что резидентный обработчик прерывания выгружен, то есть сообщения на экран не выводятся, а память, занятая резидентом, освобождена. Для этого также следует запустить программу ЛР 3. Результаты представлены на рис. 4.

```

F:\>lab5.exe /un
Interruption was unloaded
F:\>lab3_2.com
Available memory: 640 kbytes
Extended memory: 15360 kbytes
MCBs:
MCB number 1
Block is occupied by MS DOS, size = 16 bytes; occupied by: no info
MCB number 2
Block is free, size = 64 bytes; occupied by: no info
MCB number 3
Block is owned by PSP = 0040, size = 256 bytes; occupied by: no info
MCB number 4
Block is owned by PSP = 0192, size = 144 bytes; occupied by: no info
MCB number 5
Block is owned by PSP = 0192, size = 1072 bytes; occupied by: LAB3_2
MCB number 6
Block is free, size = 647824 bytes; occupied by: no info
F:\>

```

*Рисунок 4 - Вывод программы lab3\_2.com после выполнения lab5.exe с ключом выгрузки*

Как видно из рис. 5, при выгрузке резидентного обработчика было выведено сообщение, а также при запросе повторной выгрузки было показано, что резидентный обработчик не загружен.

```

F:\>lab5.exe
Interruption was loaded
F:\>lab5.exe /un
Interruption was unloaded
F:\>lab5.exe /un
Interruption is not loaded
F:\>_

```

*Рисунок 5 — Вывод программы lab5.exe при повторном запуске с ключом выгрузки*

**Шаг 6.** Даны ответы на контрольные вопросы.

А) Какого типа прерывания использовались в работе?

*Ответ:* в коде использовались программные прерывания, такие, как int 21h, тогда как само обрабатываемое прерывание от клавиатуры (09h) является аппаратным.

Б) Чем отличается скан-код от кода ASCII?

*Ответ:* скан-код — код, присвоенный каждой клавише, с помощью которого драйвер клавиатуры распознает, какая клавиша была нажата. ASCII код — код символа, необходимый для хранения и печати символа.

### **Вывод.**

В результате выполнения данной лабораторной работы была изучена возможность встраивания пользовательского обработчика прерываний от клавиатуры в стандартный.