Федеральное государственное бюджетное образовательное учреждение высшего

образования

«Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

# **Лабораторная работа 7.**

**ИЗУЧЕНИЕ ОБРАБОТЧИКА ПРЕРЫВАНИЯ КЛАВИАТУРЫ INT 9**

Вариант 9

Выполнил: Иванов В.С.

студент группы ИВТ-41-22

Проверила:

Доцент Андреева А.А.

Чебоксары, 2024

Цель работы: изучение обработчика прерывания событий и использование его на практике.

Задание. Изменить горячие клавиши блокировки/разблокировки клавиатуры.

Измененный фрагмент кода:  
Int09Hand proc

.386

push AX ;ї

push BX ;і

push CX ;іб®еа Ёвм

push DI ;іЁбЇ®«м§гҐ¬лҐ

push SI ;іаҐЈЁбвал

push DS ;і

push ES ;Щ

push CS ;їгЄ § вм DS

pop DS ;Щ иг Їа®Ја ¬¬г

in AL,60h ;Ї®«гзЁвм бЄ Є®¤ ¦ в®© Є« ўЁиЁ

cmp AL,2Dh ;сравниваем скан код нажатой клавиши с необходимой для нас

jne Exit\_09 ;Щ<L> Ё ўл©вЁ, Ґб«Ё Ґ ®

xor AX,AX ;ї

mov ES,AX ;іЇа®ўҐаЁвм д« ЈЁ Є« ўЁ вгал

mov AL,ES:[417h] ;і ¦ вЁҐ <Ctrl+Alt>

and AL,00001010b ;і

cmp AL,00001010b ;і

je Cont ;Щ

Обновленные циклы обработки выхода из локера

SWLoop1:

in al, 60h

cmp al, 2Fh

je SWLoop11

jne SWLoop1

SWLoop11:

in al,60h

cmp al, 0AFh

je SWLoop2

jmp SWLoop11

SWLoop2:

in al, 60h

cmp al, 18h

je SWLoop21

cmp al, 0AFh

je SWLoop2

jne SWLoop1

SWLoop21:

in al,60h

cmp al, 98h

je SWLoop3

jmp SWLoop21

SWLoop3:

in al, 60h

cmp al, 2Fh

je SWLoop31

cmp al, 98h

je SWLoop3

jne SWLoop1

SWLoop31:

in al,60h

cmp al, 0AFh

je SWLoop4

jmp SWLoop31

SWLoop4:

in al, 60h

cmp al, 1Eh

je SWLoop41

cmp al, 0AFh

je SWLoop4

jne SWLoop1

SWLoop41:

in al,60h

cmp al, 9Eh

je Exit

jmp SWLoop41

Exit009:

xor AX,AX ;ї

mov ES,AX ;і®зЁбвЁвм д« ЈЁ ¦ вЁп

mov AL,ES:[417h] ;і<Control+Alt> Ї® ¤аҐбг

and AL,11110101b ;і0000h:0417h Ё д« ЈЁ

mov ES:[417h],AL ;і<LeftControl+LeftAlt>

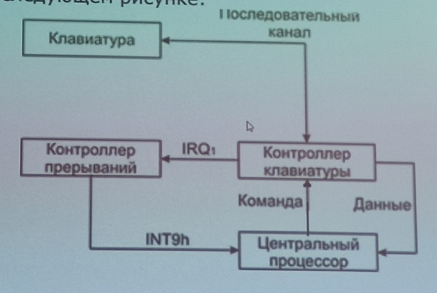
mov AL,ES:[418h] ;іЇ® ¤аҐбг 0000h:0418h

and AL,11111110b ;і

mov ES:[418h],AL ;Щ

Вывод: изучил обработчик прерывания событий и использовал его на практике.

Ответы на вопросы:

1)Схема взаимодействия клавиатуры и цп, контроллера клавиатуры и контроллера прерываний:  


Контроллер клавиатуры принимает скан-код нажатой клавиши и помещает его во внутренний буфер. Обычный скан-код представляет собой байтовое значение, младшие семь битов которого описывают код клавиши (0..127), а старший бит нажатие(0)/отпускание(1). Буфер клавиатуры доступен для чтения через порт 60h сокращенного пространства ввода-вывода. Контроллер клавиатуры также имеет и статусный и контрольный регистр.

По появлению кода в буфере контроллер формирует сигнал запроса на прерывание, который попадает в контроллер прерываний. Контроллер прерываний необходим для того, чтобы принимать сигналы от пятнадцати источников прерываний и формировать один сигнал запроса на прерывание для процессора, с последующей выдачей номера источника в специальном шинном цикле определения номера вектора. Контроллер клавиатуры подключен ко входу IRQ1 ведущего контроллера прерываний.

Контроллер прерываний состоит из двух одинаковых микросхем, включенных каскадно. Каждая микросхема обеспечивает поддержку до восьми источников. В данной лабораторной работе будет использована только входная часть контроллера прерываний которая состоит из регистра IRR (Interrupt Request Register) и IMR (Interrupt Mask Register). Регистр IRR предназначен для запоминания запросов на прерывания от устройств и будет хранить соответствующий запрос, пока он не поступит на схему генерации соответствующего прерывания или этот регистр не будет прочитан процессором. Регистр IMR предназначен для блокирования прохождения запросов на прерывания от соответствующих устройств. Единичное значение соответствующего бита в регистре IMR блокирует прохождение прерывания на дальнейшую обработку по этой линии. Регистр IMR доступен через порт 21h для чтения и записи напрямую. Регистр IRR доступен только для чтения через порт 20h, но этот порт используется для обращения к нескольким регистрам контроллера прерываний, поэтому предварительно в этот порт должна быть отправлена команда на чтение регистра IRR, которая представляет собой код 0Ah.

Для обеспечения режима программного обмена, разрабатываемая программа должна обеспечить блокирование прохождения прерывания от клавиатуры через регистр IMR. Затем в цикле должен анализироваться регистр IRR на предмет появления запроса от клавиатуры. После появления запроса необходимо выбрать очередной символ из буфера клавиатуры и приступать к его анализу и обработке.

2) как программируется контроллер прерываний. почему обработчик прерывания клавиатуры имеет номер 9, как это запрограммировано. Привести все 4 слова инициализации.

Клавиатура подключена к линии прерывания IRQ1. Этой линии соответствует прерывание INT 09h.

Клавиатурное прерывание обслуживается модулями BIOS. Драйверы клавиатуры и резидентные программы могут организовывать дополнительную обработку прерывания INT 09h.

Как работает стандартный обработчик клавиатурного прерывания, входящий в состав BIOS?

Этот обработчик выполняет следующие действия:

* читает из порта 60h скан-код нажатой клавиши;
* записывает вычисленное по скан-коду значение ASCII-кода нажатой клавиши в специальный буфер клавиатуры, расположенный в области данных BIOS;
* устанавливает в 1 бит 7 порта 61h, разрешая дальнейшую работу клавиатуры;
* возвращает этот бит в исходное состояние;
* записывает в порт 20h значение 20h для правильного завершения обработки аппаратного прерывания.
* Помимо управления содержимым буфера клавиатуры, обработчик прерывания INT 09h отслеживает нажатия на так называемые переключающие клавиши - NumLock, ScrollLock, CapsLock, Ins. Состояние этих клавиш записывается в область данных BIOS в два байта с адресами 0000h:0417h и 0000h:0418h.

Формат байта 0000h:0417h:

Биты Значение

0 Нажата правая клавиша Shift.

1 Нажата левая клавиша Shift.

2 Нажата комбинация клавиш Ctrl-Shift с любой

стороны.

3 Нажата комбинация клавиш Alt-Shift с любой

стороны.

4 Состояние клавиши ScrollLock.

5 Состояние клавиши NumLock.

6 Состояние клавиши CapsLock.

7 Состояние клавиши Insert.

Формат байта 0000h:0418h:

Биты Значение

0 Нажата левая клавиша Shift вместе с клавишей

Ctrl.

1 Нажата левая клавиша Shift вместе с клавишей

Alt.

2 Нажата клавиша SysReq.

3 Состояние клавиши Pause.

4 Нажата клавиша ScrollLock.

5 Нажата клавиша NumLock.

6 Нажата клавиша CapsLock.

7 Нажата клавиша Insert.

в общем случае для инициализации контроллера прерываний используются четыре управляющих слова (ICW1...ICW4):

ICW1. Определяет особенности последовательности приказов. Приказ посылается в порт 20h.

ICW2. Определяет адрес базового вектора. Приказ посылается в порт 21h.

ICW3. Предназначен для связи контроллеров в системе с несколькими контроллерами прерываний.

ICW4. Определяет дополнительные особенности обработки прерываний (тип микропроцессора, особенности обработки конца прерывания и т. д.).

3)Управляющие слова операции контроллера прерывания.Рассмотреть управляющее слово маски и управляющее слово конца прерываний в случае фиксированного распределения приоритетов.

### **Управляющее слово операций 1 (OCW1)**

Управляющее слово операций 1 (Operational Control Word 1, OCW1) устанавливает маску прерываний, храняющуюся в регистре IMR. Для записи OCW1 служит порт 21h в ведущем и A1h в ведомом контроллерах прерываний. Установленный разряд запрещает прерывания от соответствующей линии IRQ, сброшенный — разрешает.

Ранее записанное OCW1 может быть прочитано из того же порта.

### **Управляющее слово операций 2 (OCW2)**

Управляющее слово операций 2 (Operational Control Word 2, OCW2) используется для выдачи команды окончания обработки прерывания (EOI), выполнения вращения и установки дна приоритетов. Оно записывается в порт 20h для ведущего и A0h для ведомого контроллеров. Назначение разрядов OCW2 следующее.

|  |  |
| --- | --- |
| **Разряды** | **Назначение** |
| 7:5 | Код команды:   * 000 — отмена режима вращения приоритетов и автоматического EOI; * 001 — неспецифический EOI; * 010 — нет операции; * 011 — специфический EOI; * 100 — включение режима вращения приоритетов и автоматического EOI; * 101 — вращение приоритетов и неспецифический EOI; * 110 — установка дна приоритетов; * 111 — вращение и специфический EOI |
| 4:3 | Указывают, что это слово является OCW2. Должны быть равны нулю |
| 2:0 | В командах 011, 110 и 111 задают номер линии IRQ контроллера, к которой относится команда |

### **Управляющее слово операций 3 (OCW3)**

Управляющее слово операций 3 (Operational Control Word 3, OCW3) используется для активизации и отмены режима специального маскирования и режима опроса, а также для выбора регистра, содержимое которого может быть прочитано через порт 20h/A0h. Оно записывается в порт 20h для ведущего и A0h для ведомого контроллеров. Назначение разрядов OCW3 следующее.

|  |  |
| --- | --- |
| **Разряды** | **Назначение** |
| 7 | Должен быть равен нулю |
| 6:5 | Управление режимом специального маскирования:   * 00 — не изменять режим; * 01 — отменить режим специального маскирования; * 10 — не изменять режим; * 11 — активизировать режим специального маскирования |
| 4:3 | Указывают, что это слово является OCW3. Должны быть равны 01 |
| 2 | Режим опроса. Чтобы активизировать режим опроса, в этот разряд необходимо записать единицу, после чего считать код вектора прерывания из порта 20h/A0h. После чтения вектора режим опроса автоматически отменяется |
| 1:0 | Выбор считываемого регистра:   * 00 — не изменять; * 01 — не изменять; * 10 — считывать IRR; * 11 — считывать ISR |

В случае фиксированного распределения приоритетов для контроллера прерываний 8259A в TASM используются следующие управляющие слова:

**1. Управляющее слово маски (Mask Command Word)**

Управляющее слово маски используется для настройки разрешения или блокировки определённых прерываний. При фиксированном распределении приоритетов маска часто устанавливается следующим образом:

* **Команда маскирования**:
  + Для мастер-контроллера (IRQ0-IRQ7): OUT 0x21, AL
  + Для слейв-контроллера (IRQ8-IRQ15): OUT 0xA1, AL

**2. Управляющее слово конца прерывания (End of Interrupt Command Word)**

Управляющее слово конца прерывания используется для сигнализации контроллеру о завершении обработки прерывания. В фиксированном распределении приоритетов оно также остается тем же:

* **Команда конца прерывания**:
  + Для мастер-контроллера: OUT 0x20, AL
  + Для слейв-контроллера: OUT 0x20, AL с дополнительным параметром для указания, что обработка завершена.

4)Как осуществляется переход в подпрограмме обработки прерывания.

Инструкция INT n генерирует через программное обеспечение

обращение к обработчику прерываний. Непосредственный операнд

(значение от 0 до 255) задает номер индекса в таблице дескрипто-

ров прерываний (IDT) вызываемой обработки прерываний. В защищен-

ном режиме IDT содержит массив восьмибайтовых дескрипторов. Деск-

риптор вызываемого прерывания должен указывать прерывание,

ловушку или вентиль задачи. В реальном режиме адресации IDT

представляет собой массив из четырех указателей размером в байт.

В защищенном и реальном режиме адресации базовый линейный адрес

IDT определяется содержимым IDTR.

Инструкции прерывания INT n идентична условной программной

инструкции INTO, но номер прерывания неявно равен 4, а прерывание

выполняется, если флаг переполнения процессоров 86, 286 или 386.

Первые 32 прерывания зарезервированы фирмой Intel для сис-

темных целей. Некоторые из этих прерываний используются для внут-

ренних генерируемых исключительных ситуаций.

Инструкция INT n в общем случае ведет себя как вызов дальне-

го типа, но регистр флагов заносится в стек перед адресом

возврата. Процедуры обработки прерывания возвращают управление

через инструкцию IRET, которая извлекает из стека флаги и адрес

возврата.

В реальном режиме адресации инструкция INT n заносит в стек

флаги, регистр CS и адрес возврата (IP) в указанном порядке и пе-

реходит к длинному указателю, индекс которого указан номером

прерывания.

5)Расшифровка битов флагов состояния 0:417h и 0:418h и скан-коды использованных в программе клавиш.

mov AL,ES:[417h]

and AL,00001010b

cmp AL,00001010b

Формат байта 0000h:0417h:

Биты Значение

1 Нажата левая клавиша Shift.

3 Нажата клавиша Alt с любой

стороны.

|  |  |  |
| --- | --- | --- |
| **Key** | **Make (HEX)** | **Break (HEX)** |
| **X** | **2D** | **AD** |
| **V** | **2F** | **AF** |
| **O** | **18** | **98** |
| **A** | **1E** | **9E** |