

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

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Организация ЭВМ и Систем»
Тема: Написание собственного прерывания.

Студент гр. 0383

Зенин П.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Создать собственное прерывание, выполняющее действие, описанное в задании.

Задание.

1. Краткие сведения.

Прерывание - это процесс вызова процедур для выполнения некоторой задачи, обычно связанной с обслуживанием некоторых устройств (обработка сигнала таймера, нажатия клавиши и т.д.).

Когда возникает прерывание, процессор прекращает выполнение текущей программы (если ее приоритет ниже) и запоминает в стеке вместе с регистром флагов адрес возврата(CS:IP) - места, с которого будет продолжена прерванная программа. Затем в CS:IP загружается адрес программы обработки прерывания и ей передается управление.

Адреса 256 программ обработки прерываний, так называемые векторы прерывания, имеют длину по 4 байта (в первых двух хранится значение IP , во вторых - CS) и хранятся в младших 1024 байтах памяти.

Программа обработки прерывания должна заканчиваться инструкцией IRET (возврат из прерывания), по которой из стека восстанавливается адрес возврата и регистр флагов.

Программа обработки прерывания - это отдельная процедура, имеющая структуру:

```
SUBR_INT PROC FAR
```

```
    PUSH AX ; сохранение изменяемых регистров
```

```
    <действия по обработке прерывания>
```

```
    POP AX ; восстановление регистров
```

```
    MOV AL, 20H
```

```
    OUT 20H,AL
```

```
    IRET
```

SUBR_INT ENDP

Две последние строки перед IRET необходимы для разрешения обработки прерываний с более низкими уровнями, чем только что обработанное.

Замечание: в лабораторной работе действиями по обработке прерывания может быть вывод на экран некоторого текста, вставка задержки в вывод сообщений, включение звукового сигнала и т.п.

Программа, использующая новые программы обработки прерываний, при своем завершении должна восстанавливать оригинальные векторы прерываний. Функция 35 прерывания 21H возвращает текущее значение вектора прерывания, помещая значение сегмента в ES, а смещение в BX. В этом случае программа должна содержать следующие инструкции:

; -- в сегменте данных

KEEP_CS DW 0 ; для хранения сегмента

KEEP_IP DW 0 ; и смещения прерывания

; -- в начале программы

MOV AH, 35H ; функция получения вектора

MOV AL, 1CH ; номер вектора

INT 21H

MOV KEEP_IP, BX ; запоминание смещения

MOV KEEP_CS, ES ; и сегмента

Для задания адреса собственного прерывания с заданным номером в таблице векторов прерываний используется функция 25H прерывания 21H, которая устанавливает вектор прерывания на указанный адрес нового обработчика.

PUSH DS

MOV DX, OFFSET ROUT ; смещение для процедуры в DX

MOV AX, SEG ROUT ; сегмент процедуры

MOV DS, AX ; помещаем в DS

```

MOV AH, 25H      ; функция установки вектора
MOV AL, 60H      ; номер вектора
INT 21H          ; меняем прерывание
POP DS

```

В конце программы восстанавливается старый вектор прерывания

```

CLI
PUSH DS
MOV DX, KEEP_IP
MOV AX, KEEP_CS
MOV DS, AX
MOV AH, 25H
MOV AL, 1CH
INT 21H          ; восстанавливаем вектор
POP DS
STI

```

Варианты заданий

Шифры, определяющие варианты заданий приведены в таблице.

№ студ.	Шифр задания	№ студ.	Шифр задания	№ студ.	Шифр задания	№ студ.	Шифр задания
1	1a	8	2a	15	3a	22	4a
2	1b	9	2b	16	3b	23	4b
3	1c	10	2c	17	3c	24	4c
4	1d	11	2d	18	3d	25	4d
5	1e	12	2e	19	3e	26	4e
6	1f	13	2f	20	3f	27	4f
7	1g	14	2g	21	3g	28	4g

Цифра в шифре задает номер и назначение заменяемого вектора прерывания:

1 - 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек;

2 - 60h - прерывание пользователя - должно генерироваться в программе;

3 - 23h - прерывание, генерируемое при нажатии клавиш Control+C;

4 - 16h - прерывание от клавиатуры (по заданному скан-коду клавиши делать действия A-G, указанные ниже).

Буква определяет действия, реализуемые программой обработки прерываний:

A - Выполнить вывод сообщения на экран заданное число раз, после чего вставить фиксированную задержку и вывести сообщение о завершении обработчика.

B - Выдача звукового сигнала с заданной высотой звука.

C - Выдача звукового сигнала с заданной длительностью звучания.

D - Выполнить чтение и вывод на экран отсчета системных часов (в тиках, где 1 тик = 55 мсек).

E - Выполнить чтение и вывод на экран отсчета часов реального времени из памяти CMOS (в формате BCD).

F - Вывод на экран заданного количества (3-5) сообщений, задержка между которыми возрастает в 2 раза, начиная от 1 сек.

G - Выполнить ввод и печать заданного количества символов, после чего вывести сообщение о завершении обработчика.

Замечание: для исключения возможного взаимного влияния системных и пользовательских прерываний рекомендуется отвести в программе под стек не менее 1К байт.

Вариант 3, шифр 1с:

прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек, выдача звукового сигнала с заданной длительностью звучания.

Выполнение работы.

Была написана программа на языке Assembler, которая выдаёт звуковой сигнал заданной длительности. Для работы вывода звука было написано своё прерывание. В основной части программы происходит получение вектора прерывания (08h), используя прерывания 21h (35h), затем с помощью прерывания 21h (25h) задаётся новый адрес прерывания. Прерывание устанавливается на выполнение процедуры, длительность звука устанавливается в регистре ВХ.

Работа процедуры генерации звука происходит следующим образом: сначала включается динамик, затем в регистр СХ заносится значение из ВХ. В цикле выполняется убавление регистра СХ до 0, а уже затем динамик выключается. Данная процедура выполняется снова и снова, по сигналу таймера.

В конце программы происходит восстановление старого прерывания с использования с помощью 25h.

Код программы см. в приложении А.

Выводы.

В этой работе были изучено создание прерываний и работа с ними.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ И ДРУГИЕ ФАЙЛЫ

Название файла: lab5.asm

```
AStack SEGMENT STACK
    DW 1024 DUP(?)
AStack ENDS

DATA SEGMENT
    KEEP_CS DW 0 ; для хранения сегмента
    KEEP_IP DW 0 ; и смещения вектора прерывания

DATA ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:AStack
SUBR_INT PROC FAR ; звуковое прерывание от таймера
    jmp h_start
    save_ss dw 0000h
    save_sp dw 0000h
    ind_stack dw 512 DUP(?)
    h_start:

    mov save_ss, SS
    mov save_sp, sp
    mov sp, seg ind_stack
    mov ss, sp
    mov sp, OFFSET h_start
    PUSH CX ; сохранение изменяемых регистров
    PUSH AX

    MOV AL, 10110110b ;цепочка для командного регистра 8253
    out 43h, al
    MOV AX, 500 ;высота звука
    OUT 42H, AL ;таймер

    MOV AH, AL
    OUT 42H, AL
```

```

IN AL, 61H ; получаем состояние динамика
OR AL, 00000011b
OUT 61H, AL ; включается динамик

mov cx, bx ;
l:
NOP
NOP
NOP
sub cx, 1
cmp cx, 0
jnz l ;играется звук, пока не закончится цикл

MOV AL, AH
OUT 61H, AL ;динамик выключается

POP AX ;восстановление регистров
POP CX
    mov ss, save_ss
mov sp, save_sp
MOV AL, 20H
OUT 20H, AL ;вывод на порт
IRET
SUBR_INT ENDP

Main PROC FAR
MOV AH, 35H ; функция получения вектора
MOV AL, 08h ; номер вектора прерывания в соответствии с заданием
INT 21H ; реализуется процедура прерывания
MOV KEEP_IP, BX ; запоминание смещения
MOV KEEP_CS, ES ; и сегмента вектора прерывания
MOV BX, 1500 ; длительность звучания
PUSH DS
MOV DX, OFFSET SUBR_INT ; смещение для процедуры в DX
MOV AX, SEG SUBR_INT ; сегмент процедуры
MOV DS, AX ; помещаем в DS
MOV AH, 25H ; функция установки вектора?
MOV AL, 08h ; номер вектора
INT 21H ; меняем прерывание

```



```

POP DS

mov ah, 0h
int 16h

endInput:
    CLI ; Сброс флага прерываний IF - 0
    PUSH DS
    MOV DX, KEEP_IP
    MOV AX, KEEP_CS
    MOV DS, AX
    MOV AH, 25H
    MOV AL, 08h ;
    INT 21H ; восстанавливаем старый вектор прерывания
    POP DS
    STI
    MOV AH, 4Ch
    INT 21h
Main ENDP
CODE ENDS
END Main

```