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. 0382 _____ Самулевич В.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить прерывания в языке ассемблера, рассмотреть способы и механизмы их обработки, а также написать собственный обработчик прерываний и заменить им один из встроенных.

Задание.

Вариант 16:

Изменить обработчик прерывания 23h, чтобы он выдавал звуковой сигнал с заданной частотой звука.

Выполнение работы.

Замена обработчика происходит не навсегда, а только на время выполнения программы, поэтому в ее начале происходит сохранение положения в памяти старого обработчика (для этого в сегменте данных были созданы переменные KEEP_CS и KEEP_IP), а в конце- восстановление ячейки таблицы прерываний 23h, путем загрузки в нее сохраненных ранее значений.

Новый обработчик описан в процедуре User_interruption. Его загрузка в таблицу прерываний осуществляется с помощью функции 25h из семейства int 21h.

Новый обработчик сначала сохраняет в стек все регистры, которые он будет в дальнейшем использовать. Затем он настраивает системный таймер (путем загрузки определенного значения в порт 43h), и загружает в его второй канал (порт 42), сначала младший, а затем старший байт, числа, определяющего частоту звука из динамика (Как только это число будет загружено, процессор немедленно начнет уменьшать его со скоростью 1 193 180 раз в секунду. Каждый раз, при достижении нуля, число будет возвращаться к начальному значению и в динамик будет посылаться начало следующей прямоугольной волны, заставляя его работать на установленной частоте. Чтобы получить для числа соответствующую ему частоту работы динамика, надо сначала разделить его на 1 193 180, а затем разделить единицу на получившийся результат). После этого обработчик

включает динамик (путем установки в 0 и 1 бит порта 61h значения 1), активирует паузу на пол секунды (с помощью функции 86h из семейства int 15h), в течение которой пользователь слышит звук, и выключает динамик. В конце происходит восстановление значений регистров, которые были ранее загружены в стек.

Для того, чтобы прерывание 23h генерировалось при нажатии клавиш Control + C, в программу был добавлен бесконечный цикл, который на каждом шаге просит пользователя ввести символ (функция 0, семейство int 16h). После ввода в этом цикле проверяется код полученного символа: если он равен 3, то выполняется команда int 23h.В противном случае происходит немедленное завершение программы.

Разработанный код см. в приложении А.

Выводы.

Были изучены прерывания, механизмы и способы их обработки, а также был написан собственный обработчик и помещен в таблицу векторов прерываний.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.asm

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

DATA SEGMENT
    KEEP_CS DW 0
    KEEP_IP DW 0
DATA ENDS


CODE SEGMENT
    ASSUME CS:CODE,DS:DATA, SS:AStack

User_interruption PROC FAR
    jmp start
    PITCH_OF_SOUND_HIGHT DB 01h
    PITCH_OF_SOUND_LOW  DB 0A0h

start:
    push AX
    push CX
    push DX

    mov AL,10110110b
    out 43h,AL
    mov AL, BYTE PTR PITCH_OF_SOUND_LOW
    out 42h,AL
    mov AL, BYTE PTR PITCH_OF_SOUND_HIGHT
    out 42h,AL

    in AL,61h
    or AL,00000011b
    out 61h,AL
    mov CX,0007h
    mov DX,0A120h
    mov AH,86h
    int 15h

    in AL,61h
    and AL,11111100b
    out 61h,AL

    pop DX
    pop CX
    pop AX

IRET
User_interruption ENDP


Main PROC FAR
    push DS
```

```
    sub AX,AX
    push AX
    mov  AX,DATA
    mov DS,AX

    mov AH,35h
    mov AL,23h
    int 21h
    mov KEEP_IP,BX
    mov KEEP_CS,ES

    push DS
    mov AX,SEG User_interruption
    mov DS,AX
    mov DX,OFFSET User_interruption
    mov AH,25h
    mov AL,23h
    int 21h
    pop DS
begin:
    mov ah, 0
    int 16h
    cmp al, 3
    jnz quit
    int 23h
    jmp begin

quit:

    CLI
    push DS
    mov AX,KEEP_CS
    mov DS,AX
    mov DX,KEEP_IP
    mov AH,25h
    mov AL,23h
    int 21h
    pop DS
    STI

ret
Main ENDP

CODE ENDS
    END Main
```