

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

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Организация ЭВМ и систем»
Тема: Написание собственного прерывания
Вариант 20

Студент гр.1381

Сагидуллин Э. Р.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Цель работы.

Реализовать собственное прерывание на языке Ассемблера

Задание.

В соответствии с 20 вариантом шифр задания – 3f, где

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

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

Выполнение работы.

В сегменте стека *STACK*, как и требуется по заданию, выделяется 1 Кбайт памяти, то есть *DW 512*.

В сегменте данных *DATA* содержится две переменных для хранения старого прерывания *23h* (его смещения и сегмента), – *KEEP_CS*, *KEEP_IP*.

В сегменте кода определяем процедуру пользовательского прерывания *FUNC*. Нынешнее состояние регистров сохраняется в стек, и восстанавливаются в конце процедуры. Далее, программа заходит во вложенный цикл. Во внешнем цикле происходит вызов процедуры печати сообщения в консоль. Во внутреннем цикле происходит вызов процедуры, которая вызывает задержку. Длительность задержки увеличивается в два раза по сравнению с предыдущей итерацией. Задержка в нужное количество секунд происходит с помощью вызова процедуры задержки в одну секунду необходимое число раз.

Процедура *MAIN*. Записывается в *DS* смещение до *DATA*. Вызывается прерывание *21h* (при *AH = 35h* и *AL = 23h*) получается прерывание, хранящееся по смещению *23h*. Значения регистров сохраняются в переменные (*BX* - в *KEEP_IP*, *ES* - в *KEEP_CS*). Новое прерывание *FUNC* записывается в прерывание *23h*, с помощью прерывания *21h* (при *AH = 25h*, *AL = 23h*). Далее, происходит вызов прерывания *23h* при вводе пользователем CTRL + C. После вызова нового прерывания происходит восстановление старого прерывания и выход из программы. Исходный код см. в приложении А.

Выводы.

В ходе выполнения лабораторной работы были изучены виды прерываний и работа с ними. Создано собственное прерывание. Реализована программа, выводящая сообщение на экран с возрастающими временными промежутками.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММ

Название файла: *lab5.asm*

```
AStack  SEGMENT STACK
        DB 512 DUP(?)
AStack  ENDS
```

```
DATA    SEGMENT
        KEEP_CS DW 0
        KEEP_IP DW 0
MESSAGE DB 'MESSAGE!', 0dh, 0ah, '$'
DATA    ENDS
```

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

```
WriteMsg PROC NEAR
        push ax
        push bx
        push cx
        push dx
```

```
        mov AH, 9
        int 21h
```

```
        pop dx
        pop cx
        pop bx
        pop ax
```

```
        ret
```

```
WriteMsg ENDP
```

```
delay  PROC NEAR
        push ax
        push bx
        push cx
        push dx
```

```
        mov cx, 0fh
        mov dx, 4240h
        mov ah, 86h
        int 15h
```

```
        pop dx
        pop cx
        pop bx
        pop ax
        ret
```

```
delay  ENDP
```

```

FUNC PROC FAR
    push ax
    push bx
    push cx
    push dx

    mov dx, OFFSET MESSAGE
    mov cx, 5
    mov ax, 1
lp1:
    mov bx, cx
    call WriteMsg
    mov cx, ax

    lp2:
        call delay
    loop lp2

    shl ax, 1
    mov cx, bx
    loop lp1

    pop dx
    pop cx
    pop bx
    pop ax
    mov al, 20h
    out 20h, al
    iret
FUNC 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 dx, OFFSET FUNC
    mov ax, SEG FUNC
    mov ds, ax
    mov ah, 25h
    mov al, 23h
    int 21h
    pop ds

```

```

;int 23h
begin:
    mov ah, 0
    int 16h
    cmp al, 3
    jnz begin
    int 23h

    cli
    push ds
    mov dx, KEEP_IP
    mov ax, KEEP_CS
    mov ds, ax
    mov ah, 25h
    mov al, 23h
    int 21h
    pop ds
    sti
        mov ah, 4ch
    int 21h
MAIN ENDP
CODE ENDS
END MAIN

```