

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 15

Студент гр. 1303

Кузнецов Н.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Создать собственное прерывание согласно варианту и составить программу, обрабатывающую его.

Задание.

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

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

Выполнение работы.

В начале выполнения главной процедуры с помощью прерывания 21h и его функции 35h получаем адрес оригинального прерывания 23h и записываем его в KEEP_IP и KEEP_CS. Далее с помощью функции 25h устанавливаем свое прерывание FUNC вместо оригинального 23h. Затем реализовано зацикливание, которое нужно для ожидания ввода команды от пользователя и обработки полученной команды. Если было нажато Cntrl+C, то цикл прекращается. Далее оригинальное прерывание восстанавливается. В процедуре FUNC все регистры записываются на стек, чтобы в ходе выполнения данные не были потеряны. После этого печатаются нужные строки, а затем данные регистров восстанавливаются.

Тестирование.

| № Теста | Ввод | Вывод | Результат |
|---------|-------------|--------------------------------------|-----------|
| 1 | Отсутствует | MESSAGE MESSAGE MESSAGE END | Верно |

Вывод.

В результате лабораторной работы было создано собственное прерывание, вызывающиеся при нажатии Cntrl+C.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.asm

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

DATA SEGMENT
    KEEP_CS DW 0 ; для хранения сегмента вектора прерывания
    KEEP_IP DW 0 ; для смещения вектора прерывания
    COUNTER DW 3
    MESSAGE DB 'MESSAGE', 0dh, 0ah, '$'
    ENDING DB 'END', '$'
    EMPTY DB ' ', '$'
DATA ENDS

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

WriteMsg PROC NEAR
    mov AH, 9
    int 21h
    ret
WriteMsg ENDP

FUNC PROC FAR
    push ax
    push bx
    push cx
    push dx
    push ds
    mov dx, OFFSET MESSAGE
    mov bx, COUNTER
print:
    call WriteMsg
    sub bx, 1
    cmp bx, 0
    jne print

    mov cx, 002Eh
    mov dx, 0000h
    mov ah, 86h
    int 15h
    mov dx, OFFSET ENDING
    call WriteMsg

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

```

        mov al, 20h
        out 20h, al
        iret
FUNC ENDP

MAIN PROC FAR
    push ds
    mov ax, DATA
    mov ds, ax

    mov ah, 35h ; функция получения вектора
    mov al, 23h ; номер вектора
    int 21h
    mov KEEP_IP, bx ; запоминание смещения
    mov KEEP_CS, es ; и сегмента вектора прерывания

    push ds
    mov dx, OFFSET FUNC ; смещение для процедуры в DX
    mov ax, SEG FUNC ; сегмент процедуры
    mov ds, ax ; помещаем в DS
    mov ah, 25h ; функция установки вектора
    mov al, 23h ; номер вектора
    int 21h ; меняем прерывание
    pop ds

begin:
    mov ah, 0
    int 16h
    cmp al, 3
    jne begin
    mov dx, OFFSET EMPTY
    call WriteMsg

quit:
    cli
    push ds
    mov dx, KEEP_IP
    mov ax, KEEP_CS
    mov ds, ax
    mov ah, 25h
    mov al, 23h
    int 21h ; восстанавливаем старый вектор прерывания
    pop ds
    sti
    mov ah, 4ch
    int 21h
MAIN ENDP
CODE ENDS
    END MAIN

```