

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. 7383

Преподаватель

МЕДВЕДЕВ И. С.

КИРЬЯНЧИКОВ В. А.

Санкт-Петербург

2018

Цель работы

Изучение и разработка собственных прерываний на языке Ассемблер.

Постановка задачи

Написать собственное прерывание на языке Ассемблер в соответствии с вариантом задания. Вариант 1А. Разработать прерывание 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек.

Действия, реализуемые программой обработки прерываний: печать сообщения на экран.

Ход работы

Была написана программа на языке Ассемблер, в которой реализовано прерывание. В данной программе присутствует цикл, для того чтобы продлить время работы программы и показать, как работает прерывание от системного таймера. Код программы представлен в приложении.

Вывод

В результате выполнения данной лабораторной работы были получены навыки написания собственных прерываний и использования функций получения и установки вектора прерываний.

Приложения

Приложение А. Текст исходного файла программы с заданными значениями исходных данных

```
EOFLine EQU '$'

AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS

DATA SEGMENT

    KEEP_CS dw 0 ; для хранения сегмента

    KEEP_IP dw 0 ; и смещения прерывания

    Messege db 'interrupt done $'

DATA ENDS

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

ROUT PROC FAR ;начало процедуры

    push ax ;сохраняем все изменяемые регистры

    push dx ;

    mov ah,9h ;функция установки вектора

    mov dx,offset Messege; помещение смещения начала Messege в dx

    int 21h ;вывод строки на экран

    pop dx ;восстанавливаем регистры

    pop ax

    mov al, 20h
    out 20h, al
    iret ;конец прерывания

ROUT ENDP ;конец процедуры
```

```

Main    PROC FAR
        push DS
        sub  AX,AX
        push AX
        mov  AX,DATA
        mov  DS,AX
        MOV  AH, 35H
            MOV AL, 08H
            INT 21H
            MOV KEEP_IP, BX
            MOV KEEP_CS, ES

        PUSH DS
        MOV  DX, OFFSET ROUT
        MOV  AX, SEG ROUT
        MOV  DS, AX
        MOV  AH, 25H
        MOV  AL, 08H
        INT 21H
        POP  DS
        mov bp, 20
        mov si, 20

        cycle:
            dec bp
            cmp bp, 0
            jnz cycle
            dec si
            cmp si, 0
            jnz cycle

        CLI
        PUSH DS
        MOV  DX, KEEP_IP
        MOV  AX, KEEP_CS
        MOV  DS, AX
        MOV  AH, 25H
        MOV  AL, 08H
        INT 21H
        POP  DS
        STI

        RET

Main    ENDP
CODE    ENDS
        END Main

```