

**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: Разработка собственного прерывания**

Студент гр. 9382

\_\_\_\_\_

Кодуков А.В.

Преподаватель

\_\_\_\_\_

Ефремов М.А .

Санкт-Петербург

2020

**Задание:**

Разработать собственное прерывание

Вариант 4А: С помощью прерывания системного таймера напечатать сообщение на экране

**Выполнение работы:**

Использованные команды:

*iret* – возврат из прерывания

*cli/sti* – clear/set interrupt flag

Переопределенное прерывание: 08h – прерывание системного таймера

После переопределения прерывание выводит строку, определенную в GREETING.

**Вывод:**

В ходе выполнения работы были изучены принципы работы прерываний и переопределенно прерывание системного таймера.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

#### func.asm

```
STACKSG SEGMENT PARA STACK 'Stack'
    DW 1024 DUP(?)
STACKSG     ENDS

DATASG SEGMENT PARA 'Data'; SEG DATA
    KEEP_CS DW 0; для хранения сегмента
    KEEP_IP DW 0; и смещения вектора прерывания
    GREETING DB 'Kodukov Aleksandr 9382 $'
    COUNTER1 DW 0
    COUNTER2 DW 0
    crlf db 0ah, 0dh, '$'
DATASG     ENDS; ENDS DATA

CODE SEGMENT; SEG CODE
ASSUME DS:DataSG, CS:Code, SS:STACKSG

INTER_TIMER PROC FAR

    jmp s
    ST_SS DW 0000
    ST_SP DW 0000
    ST_AX DW 0000
    INT_STACK DW 20 DUP(0)

s:
    mov ST_SP,SP ; сохраняю SP

    mov ST_AX,AX

    mov AX,SS
    mov ST_SS,AX ; сохраняю SS

    mov AX,INT_STACK
    mov SS,AX ;новый стек

    mov AX,ST_AX

    PUSH AX; сохранение изменяемых регистров
    PUSH DX

; действия по обработке прерывания
MOV AH, 9; вызов того,
INT 21h; что хранится в dx

    INC COUNTER1

    cmp COUNTER1, 10
    jl less1
    INC COUNTER2
    MOV COUNTER1, 0
less1:
    MOV DX, COUNTER2
    ADD DX, 48
    MOV ah,2
    INT 21h
    MOV DX, COUNTER1
    ADD DX, 48
    MOV ah,2
```

```

INT 21h
cmp COUNTER2, 10
j1 less2
MOV COUNTER1, 0
MOV COUNTER2, 0
less2:

MOV DX, OFFSET crlf
MOV AH, 9
INT 21H

POP DX; восстановление регистров
POP AX

mov ST_AX,AX

mov AX,ST_SS ; восстанавливаю SS
mov SS,AX

mov SP,ST_SP ; восстанавливаю SP

mov AX,ST_AX

MOV AL, 20H
OUT 20H, AL

IRET

INTER_TIMER ENDP

Main PROC FAR
MOV AX, DATASG; ds setup
MOV DS, AX

MOV AH, 35H; функция получения вектора
MOV AL, 08H; номер вектора
INT 21H
MOV KEEP_IP, BX; запоминание смещения
MOV KEEP_CS, ES; и сегмента вектора прерывания

CLI
PUSH DS
MOV DX, OFFSET INTER_TIMER
MOV AX, SEG INTER_TIMER; 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
STI
MOV DX, OFFSET GREETING;

esc_loop:
mov ah, 10h
int 16h
cmp al,27
jz EXIT
next:
loop esc_loop

EXIT:
CLI
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
```