# 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 системы»

Тема: Написание собственного прерывания

Студента гр. 1383		Сардинов М.Г.
Преподаватель		Ефремов М.А.
	Санкт-Петербург	
	2022	

### Цель работы.

Изучить прерывания на языке Ассемблера.

### Задание.

Вариант 3d

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

D - Выполнить чтение и вывод на экран отсчета системных часов (в тиках, где 1 тик = 55 мсек).

### Выполнение работы.

Определяется процедура пользовательского прерывания Clock. При вызове она читает системное время в тактах и выводит его посимвольно через процедуру preobr.

В процедуре MAIN с помощью функции 35h мы получаем оригинальный вектор прерываний и заменяем прерывание по смещению 23h на необходимое нам. При нажатии Control+C происходит вызов написанного прерывания, после чеговосстанавливается оригинальный вектор прерываний.

### Выводы.

В ходе работы были изучены методы работы с прерываниями и разработано собственное прерывание.

### Исходный

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

### **DATA SEGMENT**

KEEP\_CS DW 0; для хранения сегмента KEEP\_IP DW 0; и смещения прерывания

### **DATA ENDS**

AStack SEGMENT STACK DW 512 DUP('0') AStack ENDS

### CODE SEGMENT

preobr PROC FAR push dx div bl

mov dh,ah cmp al,10

JGE cont3 mov dl,al add dl,'0' mov ah,02h int 21h

jmp cont4 cont3: mov dl,al add dl,'A' sub dl,10 mov ah,02h int 21h

### cont4:

cmp dh,10 JGE cont5 mov dl,dh add dl,'0' mov ah,02h int 21h jmp cont6 cont5: mov dl,dh add dl,'A'

sub dl,10

mov ah,02h

int 21h

cont6: mov ax,0

pop dx

MOV AL, 20H

OUT 20H,AL

ret

preobr ENDP

# Clock PROC FAR PUSH ax

mov ah,00 int 1ah

mov bl,16

mov ax,0

mov al,ch

call preobr

mov al,cl

call preobr

mov al,dh

call preobr

mov al,dl call preobr

pop ax

MOV AL, 20H

OUT 20H,AL

ret

Clock ENDP

### MAIN PROC FAR

push ds

sub ax, ax

push ax

mov ax, DATA mov ds, ax

mov AX, 3523h

int 21h

mov KEEP\_IP, BX

mov KEEP\_CS, ES

push DS

mov DX, OFFSET Clock

mov AX, SEG Clock

mov DS, AX

mov AH, 25h

mov AL, 23h

int 21H

pop DS

cicle:

mov ah,0

int 16h

cmp ah,2eh

je cont1

jmp cicle

cont1:

cmp al,3h

je cont2

jmp cicle

cont2:

call Clock

cli

push DS

mov DX, KEEP\_IP

mov AX, KEEP\_CS

mov DS, AX

mov AX, 2523h

int 21h

pop DS

sti

ret FAR

MAIN ENDP

END MAIN