



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №11

«Обработка прерываний»

ДИСЦИПЛИНА: «Машинно-зависимые языки программирования»

Выполнил: студент гр. ИУК4-31Б _____ (Отрошенко Т. В.)
(Подпись)

Проверил: _____ (Амеличева К. А.)
(Подпись)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2021

Цель работы: практическое овладение навыками разработки программного кода на языке Ассемблер. Изучение основных принципов разработки пользовательских обработчиков прерываний при работе МП в реальном режиме под управлением MS DOS.

Задача: разработка программы, содержащей обработчик прерываний, дополняющий или заменяющий системный обработчик.

Вариант 10

Очистить экран. Вывести несколько строк произвольного текста (атрибут 14) в середине экрана. Перехватить прерывание 1Ch, по нажатию клавиши '1' осуществить горизонтальный скроллинг всего экрана влево на один столбец, при нажатии клавиши '2' скроллинг вправо на один столбец. Завершение программы осуществляется при вводе цифры 0.

```
.model small
.data
action      db 0
msg         db 'do not worry, it will be work$'

.stack 256
.code

start:
    mov ax,@data
    mov ds,ax

    mov ax,351Ch ; сохраняем старый вектор прерывания
    int 21h

    mov word ptr cs:Old1c,BX
    mov word ptr cs:Old1c+2,ES

    mov ax,251Ch ; устанавливаем свой вектор прерывания
```

```

push ds

push cs

pop ds

mov dx,offset Newlc

int 21h

pop ds

mov ax,3

int 10h

mov ax,0B800h ; записываем адрес графического буфера

mov es,ax

mov ah,15

xor di,di

mov al,0

mov AH,0Fh

int 10h

mov ah, 02h

mov dx, 0B20h

int 10h

mov dx, offset msg

mov ah, 09h

int 21h

enter_action:

    cmp action,-1

    jz exit

    cmp action,1

    jz left

    cmp action,2

    jz right

    jmp enter_action

```

```

left:
    mov bx,0
14:
    mov dx,es:[bx]
    mov cx,79
13:
    mov ax,es:[bx+2]
    mov es:[bx],ax
    add bx,2
    loop 13
    mov es:[bx],dx
    add bx,2
    cmp bx,4000
    jb 14
    mov action,0
    jmp enter_action
right:
    mov bx,0
114:
    add bx,79*2
    mov dx,es:[bx]
    mov cx,79
113:
    mov ax,es:[bx-2]
    mov es:[bx],ax
    sub bx,2
    loop 113
    mov es:[bx],dx
    add bx,160

```

```

    cmp bx,4000

    jb 114

    mov action,0

    jmp enter_action
exit:

    mov ax,251ch

    lds dx,cs:old1c ; возвращаем старый вектор прерывания

    int 21h

    mov ax,4c00h

    int 21h


Old1c    dd  ?
New1c    proc

    pushf

    call    dword    ptr CS:[Old1c]

    cli

    push ds

    push ax

    mov ax,@data

    mov ds,ax

    mov ah,1

    int 16h

    jz ex1c

    mov ah,0

    int 16h


    cmp al, '0' ; выход из приложения

    jnz nesc

    mov action,-1

```

```

        jmp ex1c
nesc:
        cmp al,'1'
        jb ex1c
        cmp al,'2'
        ja ex1c
        sub al,'0'
        mov action,al
ex1c: ; завершение прерывания при нажатии других клавиш
        pop ax
        pop ds
        sti
        iret
New1c endp
end start

```

Результаты выполнения:

```
do not worry, it will be work_
```

```
t will be work                                do not worry, i
```

Вывод: в ходе выполнения лабораторной работы были приобретены навыки написания программ с циклами на языке Ассемблер.