

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

Студентка гр. 1381

Тулегенова А.О.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Написать собственное прерывание.

Задание

Вариант 25 – 4d.

- 4 - 16h - прерывание от клавиатуры (по заданному скан-коду клавиши делать действия A-G, указанные ниже).
- D - Выполнить чтение и вывод на экран отсчета системных часов (в тиках, где 1 тик = 55 мсек).

Ход выполнения работы

Выделяется 1Кб памяти под стек. В сегменте данных содержатся переменные KEEP_CS и KEEP_IP для хранения сегмента и смещения прерывания, чтобы программа при своем завершении восстанавливала оригинальные векторы прерываний.

В процедуре Main загружается в DS адрес начала сегмента данных, сохраняется сегмент и смещение текущего прерывания с помощью функции 35 прерывания 21H. Происходит считывание нажатой клавиши, пока не будет нажата нужная клавиша (пробел). Функция 25H прерывания 21H устанавливает вектор прерывания 16H. Далее созданное прерывание вызывается. В конце программы восстанавливается старый вектор прерывания.

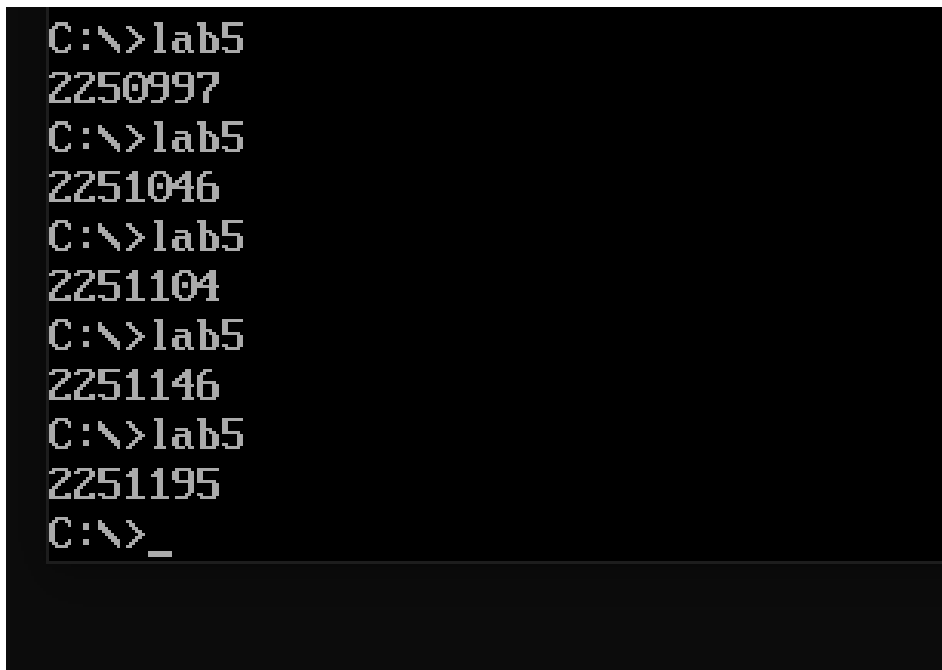
Была написана процедура пользовательского прерывания Subr_int. В стек сохраняются все изменяемые переменные. В регистр AH записывается функция чтения часов 00h, вызывается прерывание 1ah, которое записывает время в регистры cx и dx, для которых вызывается функция getTime, которая конвертирует время в символы. После выполнения функции getTime регистры восстанавливаются и происходит возврат из прерывания.

В функции getTime происходит сохранение регистров, в регистр CX записывается ноль, данный регистр будет хранить количество символов. В регистр BX записывается основание системы счисления. Пока AX, не равен нулю происходит деление на BX, при этом остаток от деления переводится в

символ и записывается в стек, значение регистра CX увеличивается на один. Далее в регистр AN записывается значения 02H для вывода на экран поочередно всех символов из стека.

Тестирование

На картинке представлены результаты работы программы.



```
C:\>lab5
2250997
C:\>lab5
2251046
C:\>lab5
2251104
C:\>lab5
2251146
C:\>lab5
2251195
C:\>_
```

Рисунок 1. Результаты работы программы.

Вывод

При выполнении лабораторной работы были изучены прерывания и была написана программа с использованием собственного прерывания.

ПРИЛОЖЕНИЕ А

КОД ПРОГРАММЫ

Название файла: lab5.asm

```
stack segment stack
    db 512 dup(?)
stack ends

data segment
    keep_cs dw 0
    keep_ip dw 0
data ends

code segment
    assume cs:code, ds:data, ss:stack

getTime proc
    push dx
    push cx
    xor cx, cx
    mov bx, 10
next:
    xor dx, dx
    div bx
    add dl, '0'
    push dx
    inc cx
    cmp ax, 0
    jnz next
    mov ah, 02h
print:
    pop dx
    int 21h
    loop print
    pop cx
    pop dx
    ret
getTime endp

Subr_int proc far
    jmp start
    keep_sp dw 0000h
    keep_ss dw 0000h
    int_stack db 50 dup(0)

    start:
    mov keep_sp, sp
    mov keep_ss, ss
    mov sp, seg int_stack
```

```

    mov ss, sp
    mov sp, offset start
    push ax
    push cx
    push dx

    mov ah, 00h
    int 1ah

    mov ax, cx
    call getTime
    mov ax, dx
    call getTime

    pop  dx
    pop  cx
    pop  ax
    mov ss, keep_ss
    mov sp, keep_sp
    mov  al, 20h

    out 20h, al
    iret
Subr_int endp

Main proc far
    push ds
    sub ax, ax
    push ax
    mov ax, data
    mov ds, ax

    mov ah, 35h
    mov al, 16h
    int 21h
    mov keep_ip, bx
    mov keep_cs, es

    check:
        mov ah, 0
        int 16h
        cmp ah, 39h
        jnz check

    push ds
    mov dx, offset Subr_int
    mov ax, seg Subr_int
    mov ds, ax
    mov ah, 25h
    mov al, 16h
    int 21h
    pop ds

```

```
    int 16h

    cli
    push ds
    mov dx, keep_ip
    mov ax, keep_cs
    mov ds, ax
    mov ah, 25h
    mov al, 16h
    int 21h
    pop ds
    sti

    mov ah, 4ch
    int 21h
Main endp
code ends
end Main
```