

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Организация ЭВМ и систем»
Тема: Написание собственного прерывания

Студент гр. 9383

Лапина А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Научиться создавать собственные прерывания.

Текст задания.

Разработать прерывание:

Шифр 1А.

1Ch – прерывание от часов — генерируется автоматически операционной системой 18 раз в сек;

А - Печать сообщения на экране

Ход работы.

Создаем строку str для хранения сообщения о прерывании «breaking». Далее создаем процедуру с прерыванием: сохраняем все изменяемые регистры, устанавливаем вектор, загружаем адрес сообщения str, выводим строки на экран, с помощью 1Ch с частотой 18 раз в секунду, восстанавливаем регистры, заканчиваем процедуру прерывания.

Затем, в main получаем вектор, запоминаем смещение, вызываем процедуру прерывания, затем восстанавливаем старый вектор прерывания.

Тестирование.

При запуске программы выводится сообщение о прерывании: «breaking» 18 раз в секунду.

Выводы.

В результате выполнения работы был получен опыт написания своего прерывания, использовать функции получения и установки вектора прерывания.

Содержимое файла lb5.asm представлено в приложении А.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.asm

```
AStack SEGMENT STACK
    DB 32 DUP(?)
AStack ENDS
DATA    SEGMENT
    KEEP_CS DW 0    ;для хранения сегмента
    KEEP_IP DW 0    ;и смещения прерывания
    str db 'breaking',10, 13,'$' ;строка для сообщения
DATA    ENDS
CODE    SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:AStack
subr_int proc far    ;начало процедуры
    push ax          ;сохраняем все изменяемые регистры
    push dx          ;сохраняем все изменяемые регистры

    mov ah,9h        ;функция установки вектора
    mov dx,offset str ;в dx загружаем адрес сообщения str

    int 21h          ;вывод строки на экран

    mov cx, 0Fh
    mov dx, 4240h
    mov ah, 86h
    int 15h;

    pop dx           ;восстанавливаем регистры
    pop ax           ;восстанавливаем регистры
    mov al,20h
    out 20h,al

    iret             ;конец прерывания
subr_int endp        ;конец процедуры

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

    MOV AH, 35H      ; функция получения вектора
    MOV AL, 1CH      ; номер вектора
    INT 21H
    MOV KEEP_IP, BX  ; запоминание смещения
    MOV KEEP_CS, ES  ; и сегмента вектора прерывания

    push ds
    mov dx,offset subr_int ;смещение для процедуры в DX
```

```

mov ax,seg subr_int    ;сегмент процедуры
mov ds,ax              ;помещаем в ds
mov ah,25h             ;функция установки вектора
mov al,1Ch             ;номер вектора
int 21h                ;меняем прерывание
pop ds                 ;восстанавливаем ds

int 1Ch                ;вызов нового прерывания 1Ch

CLI
PUSH DS
MOV DX, KEEP_IP
MOV AX, KEEP_CS
MOV DS, AX
MOV AH, 25H
MOV AL, 1CH
INT 21H                ;восстанавливаем старый вектор прерывания
POP DS
STI
ret
Main endp
code ends
end Main

```