

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

Студентка гр. 9383

Лихашва А.Д.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить прерывания на языке ассемблере, а так же написать собственное прерывание.

Задание.

Задание 2А:

Цифра в шифре задает номер и назначение заменяемого вектора прерывания:

1 - 1Ch - прерывание от часов - генерируется автоматически операционной системой

18 раз в сек;

2 - 60h - прерывание пользователя - должно генерироваться в программе;

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

4 - 08h - прерывание от системного таймера - генерируется автоматически

операционной системой 18 раз в сек.

Буква определяет действия, реализуемые программой обработки прерываний:

А - Печать сообщения на экране;

В - Выдача звукового сигнала;

С - Приостановить вывод на экран (вставить цикл задержки).

Замечание: для исключения возможного взаимного влияния системных и пользовательских

прерываний рекомендуется отвести в программе под стек не менее 1К байт.

Ход работы:

Программа обработки прерывания - это отдельная процедура **interrupt**, которая выводит сообщение «This is user interruption» на экран.

Программа, использующая новые программы обработки прерываний при своем завершении должна восстанавливать оригинальные векторы прерываний. Функция 35 прерывания 21H возвращает текущее значение вектора прерывания, помещая значение сегмента в ES, а смещение в BX.

Для установки адреса нового обработчика прерывания в поле векторов прерываний используется функция 25H прерывания 21H, которая помещает заданные адреса сегмента и смещения обработчика в вектор прерывания с заданным номером, то есть 60h (прерывание пользователя).

Далее может выполняться вызов нового обработчика прерывания. В конце программы восстанавливается старый вектор прерывания.

Тестирование.

№	Входные данные	Выходные данные
1		This is user interruption

Выводы.

В результате выполнения лабораторной работы были изучены прерывания.

Файл lab5.asm находится в приложении А.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
AStack SEGMENT STACK
```

```
    DB 256 DUP(?)
```

```
AStack ENDS
```

```
DATA SEGMENT
```

```
    KEEP_CS DW 0 ; для хранения сегмента
```

```
    KEEP_IP DW 0 ; и смещения вектора прерывания
```

```
    message DB 'This is user interruption$' ;строка для сообщения
```

```
DATA ENDS
```

```
CODE SEGMENT
```

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

```
interrupt proc far ;начало процедуры
```

```
    push ax ;сохраняем изменяемые регистры
```

```
    push dx ;сохраняем изменяемые регистры
```

```
    mov dx, offset message ;в dx помещаем адрес message
```

```
    mov ah, 9h ;функция установки вектора
```

```
    int 21h ;вывод message
```

```
    pop dx ;восстанавливаем регистры
```

```
    pop ax ;восстанавливаем регистры
```

```
    mov al, 20h
```

```
    out 20h, al
```

```
    iret ;конец прерывания
```

```
interrupt endp ;конец процедуры
```

```
main proc far
```

```
    push ds ; сохранение адреса начала PSP в стеке для последую-  
щего восстановления по команде ret
```

```
    sub ax,ax ;
```

```
    push ax ;
```

```
    mov ax,data ; загрузка сегментного регистра данных
```

```
    mov ds,ax
```

```
    mov ah, 35h ;функция получения вектора
```

```
    mov al, 60h ;номер вектора
```

```
    int 21h ; возвращение текущего значения вектора прерывания
```

```
    mov KEEP_IP, bx ;запоминание смещения
```

```
    mov KEEP_CS, es ;и сегмента вектора прерывания
```

```
    push ds
```

```
    mov dx, offset interrupt ;смещение для процедуры в DX
```

```
    mov ax, seg interrupt ;сегмент процедуры
```

```
    mov ds, ax ;помещаем в ds
```

```
    mov ah, 25h ;функция установки вектора
```

```
    mov al, 60h ;номер вектора
```

```
    int 21h ;меняем прерывание
```

```
    pop ds ;восстанавливаем ds
```

```
    int 60h ; вызов нового прерывание
```

```
    cli
```

```
    push ds
```

```
    mov dx, KEEP_IP
```

```
    mov ax, KEEP_CS
```

```
    mov ds, ax
```

```
    mov ah, 25h
```

```
mov al, 60h
int 21h    ;восстанавливаем старый вектор прерывания
pop ds
sti

ret
```

```
MAIN ENDP
```

```
CODE ENDS
```

```
END MAIN
```