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: Разработка собственного прерывания.

Студентка гр. 9383 _____ Лысова А.М,

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2020

Цель работы.

Изучить работу прерываний, написать свое в соответствии с вариантом.

Задание.

Написать прерывание в соответствии с вариантом.

Вариант 4А:

Номер и назначение заменяемого вектора прерывания - 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек.

Действие, реализуемое программой обработки прерывания — печать сообщения на экране.

Ход работы.

В ходе работы была реализована программа на языке Ассемблер, которая сохраняет старый вектор прерывания, устанавливает новый, вызывает его обработку и восстанавливает старый вектор прерывания.

В программе реализована процедура MY_INT, которая и отвечает за обработку прерывания. Внутри нее устанавливается и вызывается функция вывода строки, а так же происходит работа с более низкими уровнями благодаря процедуре out.

В основной процедуре происходит сохранение старого, установка нового и восстановление старого векторов прерывания. Используя при этом 35h и 25h — функции прерывания int 21h, которые, соответственно, получают и устанавливают вектор.

Кроме того в главной процедуре использовались такие инструкции, как cli и sti — (Clear/Set Interrupt-Enable Flag) очищают или устанавливают флаг прерывания(IF).

Исходный код см. в приложении А.

Выводы.

Были изучены механизмы работы прерываний, а также реализован собственный в соответствии с заданием.

ПРИЛОЖЕНИЕ А.
ИСХОДНЫЙ КОД ПРОГРАММЫ.

Название файла: lab5.asm

```
EOF EQU '$'

AStack  SEGMENT STACK
    DB 1024 DUP(?)
AStack  ENDS

DATA    SEGMENT
    KEEP_CS DW 0    ;для хранения сегмента вектора прерывания
    KEEP_IP DW 0    ;для хранения смещения вектора прерывания
    message DB 'I am interrupt $'
DATA    ENDS

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

MY_INT PROC FAR
    push ax ;сохранение изменяемых регистров
      push dx

      mov ah,9h ;заносим функцию вывода строки
      mov dx, offset message ;заносим адрес самой строки
    int 21h ;вызываем прервывание, выводя строку

      pop dx ;восстановление регистров
      pop ax

      mov al,20h  ;разрешение обработки прерываний
      out 20h,al  ;с более низкими уровнями

      iret ;выход из прерывания
MY_INT ENDP

MAIN PROC FAR

    mov ax, DATA
    mov ds, ax
```

```
    GETVECTOR:      ;сохраняем вектор прерывания
    mov ah, 35h     ;функция получения вектора
    mov al, 08h     ;номер вектора
    int 21h
    mov KEEP_CS, es
    mov KEEP_IP, bx

    SETVECTOR:      ;устанавливаем новый вектор прерывания
    push ds
    mov dx, OFFSET MY_INT
    mov ax, SEG MY_INT
    mov ds, ax
    mov ah, 25h ;функция установки вектора
    mov al, 08h ;номер вектора
    int 21h
    pop ds


    int 08h

    RESTOREVECTOR:  ;восстанавливаем старый вектор прерывания
    cli
    push ds
    mov dx, KEEP_IP
    mov ax, KEEP_CS
    mov ds, ax
    mov ah, 25h ;функция установки вектора
    mov al, 08h ;номер вектора
    int 21h
    pop ds
    sti

    mov ah, 4ch ;завершение программы
    int 21h

MAIN ENDP

CODE ENDS

END MAIN
```