

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

Студент гр. 9382

Дерюгин Д.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Научится создавать собственные прерывания.

Задание.

08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек

Печать сообщения на экране;

Теоретические положения.

Прерывание - это процесс вызова процедур для выполнения некоторой задачи, обычно связанной с обслуживанием некоторых устройств (обработка сигнала таймера, нажатия клавиши и т.д.). Когда возникает прерывание, процессор прекращает выполнение текущей программы (если ее приоритет ниже) и запоминает в стеке вместе с регистром флагов адрес возврата(CS:IP) - места, с которого будет продолжена прерванная программа. Затем в CS:IP загружается адрес программы обработки прерывания и ей передается управление. Адреса 256 программ обработки прерываний, так называемые векторы прерывания, имеют длину по 4 байта (в первых двух хранится значение IP , во вторых - CS) и хранятся в младших 1024 байтах памяти. Программа обработки прерывания должна заканчиваться инструкцией IRET (возврат из прерывания), по которой из стека восстанавливается адрес возврата и регистр флагов.

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.

Вывод.

В процессе выполнения лабораторной работы были получены знания о разработке собственного прерывания.

ПРИЛОЖЕНИЕ А.

ИСХОДНЫЙ КОД ПРОГРАММЫ.

```
ASTACK SEGMENT STACK
    DB 2000 DUP(?)
ASTACK ENDS

DATA SEGMENT
    KEEP_CS DW 0 ; для хранения сегмента
    KEEP_IP DW 0 ; и смещения вектора прерывания
    message DB 'hello', 10, 13, '$'
DATA ENDS

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

SUBR_INT PROC FAR
    PUSH AX
    PUSH DX
    MOV DX, OFFSET message
    MOV AH, 9h
    INT 21H
    POP DX
    POP AX
    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 ; 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

exit:
    SUB AH, AH
    INT 16H
    CMP AH, 01H
        JNE exit
    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
```