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(ка) гр. 9382 _____ Голубева В.П.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2020

Цель работы.

Разработать собственное прерывание.

Теоретические материалы.

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 обработки прерывания - это отдельная процедура, имеющая структуру:

```
SUBR_INT PROC FAR
    PUSH AX ; сохранение изменяемых регистров
    ...
    <действия по обработке
    прерывания> POP AX ;
    восстановление регистров
    …
    MOV AL, 20H
```

```
OUT 20H,AL IRET
SUBR_INT  ENDP
```

Две последние строки обработчика прерывания, указанные перед командой IRET выхода из прерывания, необходимы для разрешения обработки прерываний с более низкими уровнями, чем только что обработанное.

Замечание: в лабораторной работе действиями по обработке прерывания может быть вывод на экран некоторого текста, вставка цикла задержки в вывод сообщения или включение звукового сигнала.

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. В соответствии с этим, программа должна содержать следующие инструкции:

```
   ; -- в сегменте данных
KEEP_CS   DW   0;   для   хранения
сегмента
KEEP_IP DW 0 ; и смещения вектора
прерывания
   ; -- в начале программы
   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
```

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.

```
PUSH DS
MOV  DX,  OFFSET  ROUT  ;  смещение  для
процедуры в DX
MOV  AX, SEG ROUT     ;                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
```

Далее может выполняться вызов нового обработчика прерывания.

```
В  конце  программы  восстанавливается  старый
вектор прерывания 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
```

Задание.

Вариант 4 шифр 2A

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

A - Печать сообщения на экране.

Выполнение работы.

При помощи функции 35H получаем текущий вектор прерывания

При помощи 25H устанавливаем новый вектор прерывания

int 60H — вызываем наше пррывание

Наше прерывание представляет собой функцию WRITE_MSG

Тестирование.

Тестирование представлено в таблице 1.

Таблица 1. Тестирование программы

Номер	Входные данные	Выходные данные
1		Golubeva Valentina 9382

Выводы.

Было разработано собственное прерывание, печатающее сообщение на экран.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.asm

```
STACKSG SEGMENT  PARA STACK 'Stack'
        DW       1024 DUP(?)
STACKSG    ENDS

DATA      SEGMENT
      KEEP_CS DW 0 ; для хранения сегмента
      KEEP_IP DW 0 ; и смещения вектора прерывания
      message DB 'Golubeva Valentina 9382  $'

DATA  ENDS

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

WRITE_MSG  PROC FAR

      PUSH AX ;сохраняем все изменяемые регистры
      PUSH DX

      MOV AH, 9H ;функция установки вектора
      MOV DX, OFFSET message ; в dx загружаем адрес сообщения
      INT 21H; вывод строки на экран

      POP DX ;восстанавливаем регистры
      POP AX ;восстанавливаем регистры
      MOV AL, 20H
      OUT 20H, AL

      IRET ;конец прерывания

WRITE_MSG ENDP ;конец процедуры

Main      PROC  FAR
```

```
        PUSH DS
        SUB AX, AX
        PUSH AX
        MOV AX, DATA
        MOV DS, AX

        MOV  AH, 35H   ; функция получения текущего значения вектора
прерывания
        MOV  AL, 60H   ; номер вектора
        INT  21H
        MOV  KEEP_IP, BX  ; запоминание смещения
        MOV  KEEP_CS, ES  ; и сегмента вектора прерывания

        PUSH DS
        MOV DX, OFFSET WRITE_MSG
        MOV  AX, SEG WRITE_MSG    ; сегмент процедуры
        MOV  DS, AX          ; помещаем в DS сегмент процедуры
        MOV  AH, 25H         ; функция установки вектора
        MOV  AL, 60H         ; номер вектора
        INT  21H             ; меняем прерывание
        POP  DS         ; восстанавливаем зачение ds


        INT 60H         ;вызвали наше прерывание

        CLI             ;сбросили флаг прерывания
        PUSH DS
        MOV  DX, KEEP_IP
        MOV  AX, KEEP_CS
        MOV  DS, AX
        MOV  AH, 25H         ;установки адреса нового обработчика
прерывания в поле векторов прерываний
        MOV  AL, 60H
        INT  21H          ; восстанавливаем старый вектор прерывания
        POP  DS
        STI
```

```
      RET
Main      ENDP
CODE      ENDS
END Main        ;ENDS CODE
```