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: Создание собственных прерываний.

Студент гр. 0383	Пенкин М.В.
Преподаватель	Ефремов М.А

Санкт-Петербург 2021

Цель работы.

Вариант 8:

Создать собственное прерывание с номером 60h, вызываемое в программе, которое будет выполнять вывод сообщений на экран заданное число раз, после чего вставлять фиксированную задержку и выводить сообщение о завершении обработчика.

Выполнение работы.

Прерывание реализовано в процедуре SUBR_INT. В процедуре Маіп с помощью функции 35h/int 21h запоминается текущий вектор прерывания под номером 60h в переменные KEEP_CS, KEEP_IP. С помощью функции 25h/int 21h устанавливается новый вектор прерывания (реализованная процедура прерывания). Далее это прерывание вызывается в программе, предварительно в СХ командой MOV заносится некоторое положительное число, соответствующее количеству вывода строки на экран. В конце программы вектор прерывания под номером 60h восстанавливается с помощью переменных КЕЕР СS и KEEP IP.

Таблица 1. Проверка работы программы

Команды, вызываемые в программе для проверки работы прерывания	Результат работы программы
mov cx, 3 int 60h	Hello world! Hello world! Hello world! End!
mov cx, 10 int 60h	Hello world!

	Hello world! Hello world! Hello world! End!
--	---

Тексты исходных файлов программ см. в приложении А.

Тексты файлов диагностических сообщений см. в приложении Б.

Выводы.

В ходе выполнения данной лабораторной работы была изучена разработка собственных прерываний на языке Ассемблер.

Приложение А

Тексты исходных файлов программ

Название файла: lr5.asm

```
DATA SEGMENT
        KEEP_CS DW 0 ; для хранения сегмента
        KEEP IP DW 0 ; и смещения вектора прерывания
       HELLO DB 'Hello World!',10,13,'$'
     MESEND DB 'End!',10,13,'$'
DATA ENDS
       SEGMENT STACK
AStack
         DW 12 DUP(?) ; *: **** 12 * *****
AStack ENDS
CODE
         SEGMENT
         ASSUME CS:Code, DS:DATA, SS:AStack
SUBR_INT PROC FAR
       jmp start proc
       KEEP SS DW 0
     KEEP SP DW 0
     KEEP AX DW 0
     ; MESEND DB 'End!', 10, 13, '$'
       BStack DW 12 DUP(?)
start proc:
       MOV KEEP SP, SP
       MOV KEEP AX, AX
       MOV AX, SS
       MOV KEEP SS, AX
       MOV AX, KEEP AX
       MOV SP, OFFSET start proc
       MOV AX, seg BStack
       MOV SS, AX
        PUSH AX ; сохранение изменяемых регистров
       PUSH DX;
       MOV DX, OFFSET HELLO
       MOV AH, 9
     metka:
            21h ; Вызов функции DOS по прерыванию
       int
     loop metka ; Вывод сообщения заданное число раз
```

```
; mov ah, 86h
          ; mov cx, 0098h
          ;mov dx, 9680h
          int 15h; Фиксированная задержка
                 MOV
                        DX, OFFSET MESEND ; Вывод сообщения о
завершении обработчика
           MOV AH, 9
          int 21h
            POP DX;
            РОР АХ ; восстановление регистров
            MOV KEEP AX, AX
            MOV SP, KEEP SP
            MOV AX, KEEP SS
            MOV SS, AX
            MOV AX, KEEP AX
            MOV AL, 20H
            OUT 20H, AL
            IRET
     SUBR_INT_ENDP
    Main
         PROC FAR
             push DS ;\ Сохранение адреса начала PSP в
стеке
            sub AX,AX ; > для последующего восстановления по
                push AX
                                   ;/ команде ret, завершающей
процедуру.
                AX, DATA
                                     ; Загрузка сегментного
            mov
            mov DS, AX
                                      ; регистра данных.
            MOV АН, 35Н ; функция получения вектора
            MOV AL, 60H; номер вектора
                  INT 21H ; возвращает текущее значение вектора
прерывания
            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 АН, 25Н ; функция установки вектора
            MOV AL, 60H; номер вектора
            INT 21H ; меняем прерывание
            POP DS
```

;mov al, 0

```
mov cx, 10
     ;mov dx, 3
       int 60H; вызов измененного прерывания
        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
       ENDP
ENDS
Main
CODE
         END Main
```

Приложение Б

Тексты файлов диагностических сообщений

Название файла: lr5.lst

```
DATA SEGMENT
       KEEP\_CS DW 0 ; для хранения сегмента
       \mathtt{KEEP\_IP} DW 0 ; и смещения вектора прерывания
     TMP1 DW 0
     TMP2 DW 0
     TMP3 DW 0
       HELLO DB 'Hello World!',10,13,'$'
     MESEND DB 'End!',10,13,'$'
DATA ENDS
AStack SEGMENT STACK
        AStack
        ENDS
CODE
         SEGMENT
         ASSUME CS:Code, DS:DATA, SS:AStack
SUBR INT PROC FAR
       jmp start proc
       ST SS DW 0000
       ST AX DW 0000
       ST SP DW 0000
       IStack DW 30 DUP(?)
start proc:
       mov ST SP, SP
mov ST_AX, AX
mov AX, SS
mov ST_SS, AX
mov AX, IStack
mov SS, AX
mov AX, ST AX
push ax
push ds
MOV DX, OFFSET HELLO
```

```
MOV AH, 9
     metka:
          21h ; Вызов функции DOS по прерыванию
     int
     loop metka ; Вывод сообщения заданное число раз
     mov di,32
     mov ah, 0
     int 1Ah
     mov bx, dx; счетчик с момента сброса
     mov ah, 0
     int 1Ah
     sub dx, bx
     cmp di, dx
     ja Delay; переход, если больше
     MOV DX, OFFSET MESEND ;Выводсообщенияозавершении обработчика
     MOV AH, 9
     int 21h
     pop dx
             pop ax
     mov ST AX, AX
     mov AX,ST SS
     mov SS, AX
     mov SP,ST SP
     mov AX,ST AX
     mov al, 20h
     out 20h,al
     iret
     SUBR INT ENDP
     Main
              PROC FAR
              push DS
                                ;\ Сохранение адреса начала PSP в
стеке
             sub AX,AX ; > для последующего восстановления по
                                    ;/ команде ret, завершающей
                push AX
процедуру.
                 AX, DATA
             mov
                                       ; Загрузка сегментного
             mov DS, AX
                                       ; регистра данных.
             MOV АН, 35Н ; функция получения вектора
             MOV AL, 60H; номер вектора
                  INT 21H ; возвращает текущее значение вектора
прерывания
             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 АН, 25H ; функция установки вектора
             MOV AL, 60H; номер вектора
```

```
INT 21H ; меняем прерывание
        POP DS
     mov cx, 10
        int 60H; вызов измененного прерывания
        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
```