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 _____ Парфенов В.М.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2021

Цель работы.

Цель данной лабораторной работы заключается в том, чтобы изучить методики создания собственных прерываний на языке Ассемблер. Необходимо создать собственное прерывание, записать его вместо стандартного прерывания 8h. Прерывание должно выводить определенное количество символов введенной строки.

Задание.

Вариант №7. 1G.

1 – прерывание 08h.

G – Выполнить ввод и печать заданного количества символов, после чего вывести сообщение о завершении обработчика.

Выполнение работы.

- Прерывание реализуется в процедуре SUBR_INT.
- Для прерывания отдельно создан собственный стек.
- Происходит передача необходимых параметров из главной программы в прерывание.
- Выводиться строка, которая была введена в основной части программы.
- Выводиться сообщение о завершении работы прерывания.
- В главной части происходит замена стандартного прерывания на собственное с помощью команды 35h прерывания 21p.
- Ввод строки, которую впоследствии необходимо будет вывести в прерывании.
- Возращение стандартного прерывания на место.

Тестирование.

Здесь результаты тестирования, которые помещаются на одну страницу.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	10 символов, hello worl	hello worlEnd!	Верно
2.	10 символов, aaaaaaaaaa	aaaaaaaaaaaaEnd!	Верно
3.	10 символов, what is go	what is goEnd!	Верно

Выводы.

В ходе данной лабораторной работы была изучен механизм создания собственных прерываний на языке Ассемблер. Была создана программа, которая с помощью собственно написанного прерывания выводит в поток вывода определенное количество символов.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb5.asm

```
ASSUME CS:Code, DS:DATA, SS:AStack
DATA SEGMENT
        KEEP_CS DW 0 ; для хранения сегмента
        KEEP_IP DW 0 ; и смещения вектора прерывания
           TMP1 DW 0
           TMP2 DW 0
           TMP3 DW 0
        size1 DB 10
        string DB 30 dup("$")
           MESEND   DB 'End!',10,13,'$'

DATA ENDS


AStack    SEGMENT  STACK
          DW 12 DUP(?)    ; ⌂:⌂⌂⌂⌂⌂ 12 □⌂ ⌂⌂⌂⌂⌂⌂
AStack    ENDS



CODE      SEGMENT




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 string
     ;mov ah, 9
     ;metka:
       ;int   21h  ; Вызов функции DOS по прерыванию
       ;loop metka ; Вывод сообщения заданное число раз
```

```
      ;MOV   DX, OFFSET HELLO
      ;MOV   AH,9
      ;metka:
      ;int   21h  ; Вызов функции DOS по прерыванию
      ;loop metka ; Вывод сообщения заданное число раз




    mov ah,9h
    mov dx, offset string
    int 21h

    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
        push  AX       ;/  команде ret, завершающей процедуру.
        mov   AX,DATA             ; Загрузка сегментного
        mov   DS,AX               ; регистра данных.

        MOV AH, 35H ; функция получения вектора
        MOV AL, 08h ; номер вектора
        INT 21H ; возвращает текущее значение вектора прерывания
        MOV KEEP_IP, BX ; запоминание смещения
        MOV KEEP_CS, ES ; и сегмента вектора прерывания

        mov cx, 10

        lea bx,string
        mov ah, 1
        n1:

        int 21h
        mov [bx],al
```

```
        inc bx
        loop n1

        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, 08H ; номер вектора
        INT 21H ; меняем прерывание
        POP DS


        int 08h


        CLI
        PUSH DS
        MOV DX, KEEP_IP
        MOV AX, KEEP_CS
        MOV DS, AX
        MOV AH, 25H
        MOV AL, 08H
        INT 21H ; восстанавливаем старый вектор прерывания
        POP DS
        STI

        RET
Main      ENDP
CODE      ENDS
          END Main
```