

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: Написание собственного прерывания
ВАРИАНТ 14

Студент гр. 1383

Манучарова А.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить прерывания на Ассемблера, реализовать собственный обработчик прерываний.

Задание.

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

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

Выполнение работы.

В сегменте данных находятся: KEEP_CS для хранения сегмента, KEEP_IP для хранения смещения прерывания, crlf для вывода переноса строки, mes для вывода сообщения о завершении обработчика, buf для хранения сохранения входной строки.

Реализован обработчик, в процессе работы которого производится считывание строки, её вывод, а также сообщение о завершении обработчика.

В процедуре main с помощью функции 35 прерывания 21h сохраняется вектор прерывания в KEEP_CS и KEEP_IP. Далее благодаря функции 25 прерывания 21h устанавливается вектор прерывания на реализованный обработчик прерываний.

После завершения работы обработчика восстанавливается старый вектор прерывания.

Выводы.

В ходе выполнения работы были изучены прерывания и реализовано на практике прерывание пользователя, генерирующееся в программе.

ПРИЛОЖЕНИЕ А

КОД ПРОГРАММЫ

lab5.asm:

```
AStack SEGMENT STACK
```

```
    db 1024 DUP(?)
```

```
AStack ENDS
```

```
DATA SEGMENT
```

```
    KEEP_CS dw 0
```

```
    KEEP_IP dw 0
```

```
    crlf db 0dh, 0ah, '$'
```

```
    mes db 'Processing completed$'
```

```
    buf db 20
```

```
DATA ENDS
```

```
CODE SEGMENT
```

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

```
SUBR_INT PROC FAR
```

```
    PUSH AX
```

```
    push DX
```

```
    push bx
```

```
    push cx
```

```
    mov dx, offset buf
```

```
    mov ah, 0ah
```

```
    int 21h
```

```
    mov dx, offset crlf
```

```
    mov ah, 09h
```

```
    int 21h
```

```

    mov dx, offset buf
    xor ah, ah
    lea bx, buf+2
    xor cx, cx
    mov cl, buf+1
    add bx,cx
    mov byte ptr [bx], '$'
    add dx, 2
    mov ah, 9h
    int 21h

    mov dx, offset crlf
    mov ah, 09h
    int 21h

    mov ah, 9h
    mov dx, offset mes
    int 21h
    pop cx
    pop bx
    pop dx
    POP AX
    MOV AL, 20H
    OUT 20H,AL
    IRET
SUBR_INT ENDP

Main PROC FAR
    push ds

```

```

xor ax, ax
push ax
mov ax, DATA
mov ds, ax

mov ah, 35h
mov al, 60h
int 21h
mov KEEP_IP, bx
mov KEEP_CS, es

PUSH DS
MOV DX, OFFSET SUBR_INT
MOV AX, SEG SUBR_INT
MOV DS, AX
MOV AH, 25H
MOV AL, 60H
INT 21H
POP DS
int 60h

CLI
PUSH DS
MOV DX, KEEP_IP
MOV AX, KEEP_CS
MOV DS, AX
MOV AH, 25H
MOV AL, 60H
INT 21H
POP DS

```

STI

ret

Main ENDP

CODE ENDS

END Main