

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.

Студентка гр. 1383

Федорова О.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Разработать программу на языке ассемблер с обработкой прерываний в соответствии с вариантом

Задание.

3 - 23h - прерывание, генерируемое при нажатии клавиш Control+C .

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

Выполнение работы.

Длина выводимой строки задается в начале программы, в начале сегмента данных были созданы KEEP_CS и KEEP_IP для хранения сегмента и смещения прерывания. Была разработана функция SUBR_INT, вызываемая во время прерывания, в которой в начале в стек записываются значения регистров AX, BX, DX, после чего происходит считывание символов в строку для вывода, затем происходит вызов функции WriteMsg для вывода считанных символов и строки завершения обработки, после чего происходит восстановление значений изменяемых в функции регистров.

В головной функции после выбора вектора прерывания в метке expt происходит считывание нажатий клавиш до тех пор, пока считанная клавиша не будет соответствовать 3. Так нужно потому что считанные символы записываются в младший байт регистра ах, сравнение с числом равносильно сравнению с номером в таблице ascii символа. Для выбранного прерывания cntrl+C номер в таблице соответствует 3, поэтому как только полученное значение равняется 3, будет вызвано прерывание int 23h.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Hdj dj + cntrlC	Ничего не происходит	После нажатия комбинации клавиш, вызывается прерывание, которое ожидает на вход определенное количество символов
2.	cntrl+C jdjfk dle	jdjfk Student from 1383 - Fedorova Oksana	Выбранное число символов — 5, после ввода 5 символа программа выведет их и строку о завершении, так что ввести все эти символы не получится
3.	cntrl+c 123345	12334 Student from 1383 - Fedorova Oksana	

Выводы.

Изучены прерывания на языке ассемблер, разработана собственная программа, Меняющая вектор прерывания и обрабатывающая его.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Lb5.asm

STR_LEN equ 5 ; кол-во символов в строке для ввода

; Стек программы

AStack SEGMENT STACK

DW 1024 DUP(?) ; Отводится 1024 слов памяти

AStack ENDS

; Данные программы

DATA SEGMENT

KEEP_CS DW 0 ; для хранения сегмента

KEEP_IP DW 0 ; и смещения прерывания

INPUT DB 0Dh, 0Ah, STR_LEN dup("\$"), '\$'

; ,0dh,0ah - нова стока

GREETING DB 0dh,0ah,'Student from 1383 - Fedorova Oksana \$'

DATA ENDS

; Код программы

CODE SEGMENT

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

WriteMsg PROC NEAR

mov AH,9

int 21h ; Вызов функции DOS по прерыванию

ret

WriteMsg ENDP

SUBR_INT PROC FAR

PUSH AX ; сохранение изменяемых регистров

PUSH CX

PUSH DX

; действия по обработке прерывания

mov cx, STR_LEN

mov di, offset INPUT ; получаем смещение на начало сообщения

add di, 2

mov ah, 01h ; ввод с клавиатуры

write_loop:

int 21h

mov [di], al ; помещаем символ в строку

add di, 1

loop write_loop

; вывод строк (сообщений)

mov dx, offset INPUT

call WriteMsg

mov dx, offset GREETING

call WriteMsg

; восстановление регистров

POP AX

POP CX

POP DX

MOV AL, 20H ; для разрешения обработки прерываний

OUT 20H,AL ; с более низкими уровнями, чем только что

обработанное

IRET

SUBR_INT ENDP

; Головная процедура

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.
```

```
; Запоминание текущего вектора прерывания
```

```
MOV AH, 35H   ; функция получения вектора
```

```
MOV AL, 23H   ; номер вектора
```

```
INT 21H
```

```
MOV KEEP_IP, BX ; запоминание смещения
```

```
MOV KEEP_CS, ES ; и сегмента
```

```
; Установка 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, 23H           ; номер вектора
```

```
INT 21H               ; меняем прерывание
```

```
POP DS
```

```
expt:
```

```
mov ah, 0
```

```
int 16h ;Клавиатурный ввод (чтение клавиш)
```

```
cmp al, 3;код ctrlC
```

```
jne expt;переход, если не то нажато
```

```
int 23h
```

```
; Восстановление изначального вектора прерывания
```

```
CLI
PUSH DS
MOV DX, KEEP_IP
MOV AX, KEEP_CS
MOV DS, AX
MOV AH, 25H
MOV AL, 23H
INT 21H      ; восстанавливаем вектор
POP DS
STI
RET
```

```
Main    ENDP
CODE    ENDS
        END Main
```

