

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 гр. 1381

Таргонский М. А.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Цель работы.

Изучение и разработка собственного прерывания.

Задание.

Вариант 21, Шифр 3g. Цифра в шифре задает номер и назначение заменяемого вектора прерывания: 3 - 23h - прерывание, генерируемое при нажатии клавиш Control+C; Буква определяет действия, реализуемые программой обработки прерываний: G - Выполнить ввод и печать заданного количества символов, после чего вывести сообщение о завершении обработчика. Замечание: для исключения возможного взаимного влияния системных и пользовательских прерываний рекомендуется отвести в программе под стек не менее 1К байт.

Выполнение работы.

Функция 35 прерывания 21h возвращает текущее значение вектора прерывания (а именно 023h), помещая значение сегмента и смещение в переменные KEEP_CS и KEEP_IP (каждая по 2 байта).

Для задания адреса собственного прерывания с заданным номером в таблицу векторов прерываний используется функция 25h прерывания 21h, которая устанавливает вектор прерывания на указанный адрес нового обработчика (процедура SUBR_INT).

Для выполнения ввода и печати заданного количества символов (действия по обработке прерывания) реализована процедура SUBR_INT. В конце программы восстанавливается старый вектор прерывания при помощи переменных KEEP_CS и KEEP_IP.

Работа программы и процедуры описана в комментариях в коде.

Тексты исходных файлов программ см. в приложении А.

Тексты файлов диагностических сообщений см. в приложении В.

Тестирование.

№	Исходные данные.	Ожидаемый результат.	Полученный результат.
1	Нажатие других клавиш (НЕ Ctrl+C)	Без изменений (ничего не происходит)	Все верно
2	Нажатие Ctrl+C Dcvbgfdc21485!@R	Dcvbgfdc21485!@R Finish	Все верно

Выводы.

Были изучено и разработано собственное прерывание.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ
ФАЙЛ: lab5.asm

```
NUM_SYM 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
            ; возврат каретки с кодом 13 (0Dh), перевод строки с кодом 10 (0Ah).
            MES DB 0Dh, 0Ah, NUM_SYM dup("$"), '$'
            MES_END DB 0Dh, 0Ah, 'Finish', '$'
DATA      ENDS

; Код программы
CODE      SEGMENT
            ASSUME CS:CODE, DS:DATA, SS:AStack

SUBR_INT  PROC FAR
            JMP start
            KEEP_SS DW 0000h
            KEEP_SP DW 0000h
            KEEP_STACK DB 40 DUP(?)
start:
            MOV KEEP_SP, SP
            MOV KEEP_SS, SS
            MOV SP, SEG KEEP_STACK
            MOV SS, SP
            MOV SP, offset start

            PUSH AX ; сохранение изменяемых регистров
            PUSH CX
            PUSH DX

            ; действия по обработке прерывания
```

```

mov cx, NUM_SYM
mov di, offset MES ; получаем смещение на начало сообщения
add di, 2
mov ah, 01h ; ввод с клавиатуры
lp:
int 21h
mov [di], al ; помещаем символ в строку
inc di
loop lp

; вывод строк (сообщений)
mov ah, 09h
mov dx, offset MES
int 21h
mov dx, offset MES_END
int 21h

POP AX ; восстановление регистров
POP CX
POP DX
MOV SS, KEEP_SS
MOV SP, KEEP_SP
MOV AL, 20H ; для разрешения обработки прерываний
OUT 20H,AL ; с более низкими уровнями, чем только что обработанное
IRET
SUBR_INT ENDP

```

; Главная 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

```

```

;ожидание нажатия ctrl_c

```

```

ctrl_c:
    mov ah, 0
    int 16h ;Клавиатурный ввод (чтение клавиш)
    cmp al, 3
    jne ctrl_c

```

```

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

```

ПРИЛОЖЕНИЕ Б
ИСХОДНЫЙ КОД ПРОГРАММЫ
ФАЙЛ: lab5.lst

#Microsoft (R) Macro Assembler Version 5.10

12/5/22 13:20:37

Page 1-1

= 0005 NUM_SYM equ 5 ; кол-во символов в
 строке для ввода

 ; Стек программы

0000 AStack SEGMENT STACK

0000 0400[DW 1024 DUP(?) ; Отводится
 1024 слов памяти

 ????

]

0800 AStack ENDS

 ; Данные программы

0000 DATA SEGMENT

0000 0000 KEEP_CS DW 0 ; для хранения
 сегмента

0002 0000 KEEP_IP DW 0 ; и смещения
 рывания

 ; возврат каретки с к
 одом 13 (0Dh), перевод строки

с кодом 10 (0Ah).

```
0004 0D 0A MES DB 0Dh, 0Ah, NUM_SYM dup("$"),  
'$
```

,

```
0005[
```

```
24
```

```
]
```

```
24
```

```
000C 0D 0A 46 69 6E 69 MES_END DB 0Dh, 0Ah, 'Finish', '$'  
73 68 24
```

```
0015 DATA ENDS
```

; Код программы

```
0000 CODE SEGMENT
```

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

```
0000 SUBR_INT PROC FAR
```

```
0000 EB 2D 90 JMP start
```

```
0003 0000 KEEP_SS DW 0000h
```

```
0005 0000 KEEP_SP DW 0000h
```

```
0007 0028[ KEEP_STACK DB 40 DUP(?)
```

```
??
```

```
]
```

```
002F start:
```

```
002F 2E: 89 26 0005 R MOV KEEP_SP, SP
```

```
0034 2E: 8C 16 0003 R MOV KEEP_SS, SS
```

```
0039 BC ---- R MOV SP, SEG KEEP_STACK
```

```
003C 8E D4 MOV SS, SP
```


003E BC 002F R MOV SP, offset start

0041 50 PUSH AX ; сохранение из
 еняемых регистров

0042 51 PUSH CX

0043 52 PUSH DX

#Microsoft (R) Macro Assembler Version 5.10

12/5/22 13:20:37

Page 1-2

 ; действия по обрабо
 ке прерывания

0044 B9 0005 mov cx, NUM_SYM

0047 BF 0004 R mov di, offset MES ; получае
 смещение на начало сообщ
 ения

004A 83 C7 02 add di, 2

004D B4 01 mov ah, 01h ; ввод с клав
 атуры

004F lp:

004F CD 21 int 21h

0051 88 05 mov [di], al ; помещаем си
 мвол в строку

0053 47 inc di

0054 E2 F9 loop lp

 ; вывод строк (сообщ
 ний)

0056 B4 09	mov ah, 09h
0058 BA 0004 R	mov dx, offset MES
005B CD 21	int 21h
005D BA 000C R	mov dx, offset MES_END
0060 CD 21	int 21h
0062 58	POP AX ; восстановлен
	е регистров
0063 59	POP CX
0064 5A	POP DX
0065 2E: 8E 16 0003 R	MOV SS, KEEP_SS
006A 2E: 8B 26 0005 R	MOV SP, KEEP_SP
006F B0 20	MOV AL, 20H ; для разреше
	ния обработки прерываний
0071 E6 20	OUT 20H,AL ; с более низк
	ими уровнями, чем только ч
	то обработанное
0073 CF	IRET
0074	SUBR_INT ENDP
	; Головная процедура
0074	Main PROC FAR
0074 1E	push DS ;\ Сохранени
	е адреса начала PSP в стеке
0075 2B C0	sub AX,AX ;> для после
	ующего восстановления по
0077 50	push AX ;/ команде ret
	, завершающей процедуру.
0078 B8 ---- R	mov AX,DATA ; Загрузка с

```

007B 8E D8                mov  DS,AX    ; регистра д
                                анных.

                                ; Запоминание текуще
                                го вектора прерывания

#Microsoft (R) Macro Assembler Version 5.10                12/5/22 13:20:37
                                Page    1-3

007D B4 35                MOV  AH, 35H    ; функция по
                                учения вектора

007F B0 23                MOV  AL, 23H    ; номер вект
                                ра

0081 CD 21                INT  21H

0083 89 1E 0002 R        MOV  KEEP_IP, BX ; запомина
                                ие смещения

0087 8C 06 0000 R        MOV  KEEP_CS, ES ; и сегмент
                                а

                                ; Установка вектора
                                рерывания

008B 1E                PUSH DS

008C BA 0000 R        MOV  DX, OFFSET SUBR_INT ; смеще
                                ние для процедуры в DX

008F B8 ---- R        MOV  AX, SEG SUBR_INT    ; сегме
                                нт процедуры

0092 8E D8                MOV  DS, AX      ; помещае
                                м в DS

```

0094 B4 25	MOV AH, 25H	; функция
		установки вектора
0096 B0 23	MOV AL, 23H	; номер в
		ктора
0098 CD 21	INT 21H	; меняем
		прерывание
009A 1F	POP DS	
		;ожидание нажатия ctrl
		_c
009B	ctrl_c:	
009B B4 00	mov ah, 0	
009D CD 16	int 16h	;Клавиатурны
		й ввод (чтение клавиш)
009F 3C 03	cmp al, 3	
00A1 75 F8	jne ctrl_c	
00A3 CD 23	int 23h	
		; Восстановление изн
		ачального вектора прерыв
		ния
00A5 FA	CLI	
00A6 1E	PUSH DS	
00A7 8B 16 0002 R	MOV DX, KEEP_IP	
00AB A1 0000 R	MOV AX, KEEP_CS	
00AE 8E D8	MOV DS, AX	
00B0 B4 25	MOV AH, 25H	
00B2 B0 23	MOV AL, 23H	

00B4 CD 21 INT 21H ; ВОССТАНА

ливаем вектор

00B6 1F POP DS

00B7 FB STI

00B8 CB RET

#Microsoft (R) Macro Assembler Version 5.10

12/5/22 13:20:37

Page 1-4

00B9 Main ENDP

00B9 CODE ENDS

END Main

#Microsoft (R) Macro Assembler Version 5.10

12/5/22 13:20:37

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0800	PARA		STACK
CODE	00B9	PARA		NONE
DATA	0015	PARA		NONE

Symbols:

N a m e	Type	Value	Attr
---------	------	-------	------

CTRL_C	L NEAR	009B	CODE	
KEEP_CS	L WORD	0000	DATA	
KEEP_IP	L WORD	0002	DATA	
KEEP_SP	L WORD	0005	CODE	
KEEP_SS	L WORD	0003	CODE	
KEEP_STACK	L BYTE	0007	CODE	Length = 0028
LP	L NEAR	004F	CODE	
MAIN	F PROC	0074	CODE	Length = 0045
MES	L BYTE	0004	DATA	
MES_END	L BYTE	000C	DATA	
NUM_SYM	NUMBER	0005		
START	L NEAR	002F	CODE	
SUBR_INT	F PROC	0000	CODE	Length = 0074
@CPU	TEXT	0101h		
@FILENAME	TEXT	LAB5		
@VERSION	TEXT	510		

115 Source Lines

115 Total Lines

21 Symbols

48016 + 455147 Bytes symbol space free

0 Warning Errors

0 Severe Errors