

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. 0383

Арсентьева. Д.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

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 1K 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.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
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

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

Название файла: 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
           PUSH AX ; сохранение изменяемых регистров
           PUSH BX
           PUSH CX
           PUSH DX

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

        ;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 В **ТЕКСТЫ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ**

Название файла: Lab5.lst

Microsoft (R) Macro Assembler Version 5.10

12/19/21 20:15:2

Page 1-1

```

= 0005                                NUM_SYM equ 5 ; ミコミセミサ-ミイミセ ム・クミシミイミセミサ
ミセミイ ミイ

                                ム・ビ€ ミセミコミオ ミエミサム・ミイミイミセミエミ-

                                ; ミ。ムヒオミコ ミソム€ ミセミウム€ ミ-ミシミシム・
0000                                AStack  SEGMENT STACK
0000 0400[                                DW 1024 DUP(?) ; ミ榎ヒイミセミエミクムビ・・
                                1024 ム・サミセミイ ミソミ-ミシム肖ヒク
                                ???
                                ]

0800                                AStack  ENDS

                                ; ミ頻-ミスミスム巾オ ミソム€ ミセミウム€ ミ-ミシミシム・
0000                                DATA  SEGMENT
0000 0000                                KEEP_CS DW 0 ; ミエミサム・ム・€ ミ-ミスミオミスミク
                                ム・ム・オミウミシミオミスムヒ-

0002 0000                                KEEP_IP DW 0 ; ミク ム・シミオム禍オミスミクム・ミ
                                ソム€ ミオム€ ム巾イミ-ミスミクム・
                                ; ミイミセミキミイム€ ミ-ム・ミコミ-ム€ ミオムヒコミク ム・ミコ
                                ミセミエミセシ 13 (0Dh), ミソミオム€ ミオミイミセミエ ム・ビ€ ミセミコミク
                                ム・ミコミセミエミセシ 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

; ミ墟セミエ ミソム€ ミセミウム€ ミーシシシム・
0000      CODE    SEGMENT
            ASSUME CS:CODE, DS:DATA, SS:AStack

0000      SUBR_INT PROC FAR
0000 50      PUSH AX ; ム・セム・€ ミーミスミオミスミクミオ ミクミキ
            ミシミオミスミ紹オミシム錦・ム€ ミオミウミクム・ビ€ ミセミイ

0001 53      PUSH BX
0002 51      PUSH CX
0003 52      PUSH DX

            ; ミエミオミケム・ヒイミクム・ミソミセ ミセミアム€ ミーミアミセム
            ヒコミオ ミソム€ ミオム€ ム巾イミーミスミクム・

0004 B9 0005      mov cx, NUM_SYM
0007 BB 0004 R      mov bx, offset MES ; ミソミセミサム τ・ーミオ
            ミシム・シミオム禍オミスミクミオ ミスミー ミスミーム・ーミサミセ ム・セミセミアム
            禍オミスミクム・

000A 83 C3 02      add bx, 2
000D B4 01      mov ah, 01h ; ミイミイミセミエ ム・ミコミサミーミイ
            ミクミームビ τ€ ム・

000F      lp:
000F CD 21      int 21h
0011 88 07      mov [bx], al ; ミソミセミシミオム禍ーミオミシム・

```

```
0013 43          inc bx
0014 E2 F9          loop lp
```

;ミイム巾イセシエム・び€ミセミコ(ム・セセミアム禍オ
ミスミクミケ)

```
0016 B4 09          mov ah, 09h
0018 BA 0004 R      mov dx, offset MES
001B CD 21          int 21h
001D BA 000C R      mov dx, offset MES_END
0020 CD 21          int 21h
```

```
0022 58            POP AX ;ミイセム・・ひーミスミセミイサオミスミ  
クミオム€ミオミウミクム・び€ミセミイ
```

```
0023 5B            POP BX
0024 59            POP CX
0025 5A            POP DX
```

```
0026 B0 20          MOV AL, 20H ;ミエミサム・ム€ミーミキム€ミオム
```

威オ

ミスミクム・ミセミアム€ミーミアミセミコミクミソム€ミオム€ム巾イミミスミクミ

ケ

```
0028 E6 20          OUT 20H,AL ;ム・ミアミセミサオミオミスミクミキミ
```

コ

ミクミシミクムτ€ミセミイミスム紹シミク,ム・オミシムひセミサム糊コミセム・
ムひセミセミアム€ミーミアミセムひーミスミスミセミオ

```
002A CF            IRET
002B              SUBR_INT ENDP
```

;ミ寅セミサミセミイミスミーム・ミソム€ミセム・オミエムτ€ミー

```
002B              Main      PROC FAR
```

```
002B 1E            push DS ;\ミ。ミセム・€ミーミスミオミスミク
```

ミオミーミエム€ミオム・ーミスミーム・ーミサミーPSPミイム・ひオミコミオ

```
002C 2B C0          sub AX,AX ;>ミエミサム・ミソミセム・サミオミ
```

エムτ 紗禍オミウミセミイミセム・・ひーミスミセミイミサオミスミクム・ミソミセ

```
002E 50            push AX ;/ミコミセミシミミスミエミオret
```

,ミキミミイミオム€ム威ーム紗禍オミケミソム€ミセム・オミエムτ€ム・


```

002F B8 ---- R          mov  AX,DATA ;ミ厘-ミウム€ムσキミコミ-ム・
                           ミオミウシミオミスムヒスミセミウミセ
0032 8E D8              mov  DS,AX ;ム€ミオミウミクム・ヒ€ミ-ミエ
                           ミ-ミスミス錦・

                           ;ミ厘-ミソミセシシクミスミ-ミスミクミオムヒオミコムτ禍オ
                           ミウミセミイミオミコムヒセム€ミ-ミソム€ミオム€ム巾イミ-ミスミクム・
0034 B4 35              MOV  AH, 35H ;ム・σスミコム・クム・ミソミセミ
                           サムτ・オミスミクム・ミイミオミコムヒセム€ミ-
0036 B0 23              MOV  AL, 23H ;ミスミセシシオム€ミイミオミコム
        ひ
                           セム€ミ-
0038 CD 21              INT  21H
003A 89 1E 0002 R      MOV  KEEP_IP, BX ;ミキミ-ミソミセシシクミス
        ミ-ミ
                           スミクミオム・シミオム禍オミスミクム・
003E 8C 06 0000 R      MOV  KEEP_CS, ES ;ミクム・オミウシミオミス
        ム・
                           ミ-

                           ;ミ]ム・ヒ-ミスミセミイミコミ-ミイミオミコムヒセム€ミ-ミ
                           ソム€ミオム€ム巾イミ-ミスミクム・
0042 1E                PUSH DS
0043 BA 0000 R          MOV  DX, OFFSET SUBR_INT ;ム・シミオム禍オ

```

Microsoft (R) Macro Assembler Version 5.10

12/19/21 20:15:2

Page 1-3

```

                           ミスミクミオミエミサム・ミソム€ミセム・オミエムτ€ム・ミイDX
0046 B8 ---- R          MOV  AX, SEG SUBR_INT ;ム・オミウシミオ
                           ミスム・ミソム€ミセム・オミエムτ€ム・

```

0049	8E D8	MOV DS, AX	; ミソミセシミオム禍ーミオ シシイDS
004B	B4 25	MOV AH, 25H	; ム・σ スミコム・クム・ ムτ・ひーミスミセイミコミク ミイオミコムヒセム€ ミー
004D	B0 23	MOV AL, 23H	; ミスミセシミオム€ ミイ オミコムヒセム€ ミー
004F	CD 21	INT 21H	; ミシミオミスム紹オシシ ソム€ ミオム€ ム巾イミーミスミクミオ
0051	1F	POP DS	
			; ミセミカミクミエミーミスミクミオ ミスミーミカミームヒクム・ctrl _c
0052		ctrl_c:	
0052	B4 00	mov ah, 0	
0054	CD 16	int 16h	; ミ墟サミーミイミクミームヒ ^g τ€ ミスム・ ミケ ミイミイセミエ (ム・ひオミスミクミオ ミコミサミーミイミクム・
0056	3C 03	cmp al, 3	
0058	75 F8	jne ctrl_c	
005A	CD 23	int 23h	
			; ミ漬セム・・ひーミスミセイミサミオミスミクミオ ミクミキミス ミーム・ーミサム糊スミセミウミセ ミイミオミコムヒセム€ ミー ミソム€ ミオム€ ム 巾イ ーミスミクム・
005C	FA	CLI	
005D	1E	PUSH DS	
005E	8B 16 0002 R	MOV DX, KEEP_IP	
0062	A1 0000 R	MOV AX, KEEP_CS	
0065	8E D8	MOV DS, AX	
0067	B4 25	MOV AH, 25H	
0069	B0 23	MOV AL, 23H	
006B	CD 21	INT 21H	; ミイミセム・・ひーミスミーミ イミサミクミイミーミオシシ ミイミオミコムヒセム€
006D	1F	POP DS	
006E	FB	STI	

```

006F CB                RET

0070                Main  ENDP
0070                CODE  ENDS
                        END Main

```

```

Microsoft (R) Macro Assembler Version 5.10      12/19/21 20:15:2
                        Symbols-1

```

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0800	PARA		STACK
CODE	0070	PARA		NONE
DATA	0015	PARA		NONE

Symbols:

N a m e	Type	Value	Attr	
CTRL_C	L NEAR	0052	CODE	
KEEP_CS	L WORD	0000	DATA	
KEEP_IP	L WORD	0002	DATA	
LP	L NEAR	000F	CODE	
MAIN	F PROC	002B	CODE	Length = 0045
MES	L BYTE	0004	DATA	
MES_END	L BYTE	000C	DATA	
NUM_SYM	NUMBER	0005		
SUBR_INT	F PROC	0000	CODE	Length = 002B
@CPU	TEXT	0101h		
@FILENAME	TEXT	LAB5		
@VERSION	TEXT	510		

104 Source Lines

104 Total Lines
17 Symbols

48016 + 455147 Bytes symbol space free

0 Warning Errors
0 Severe Errors