

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе № 5
по дисциплине «Организация ЭВМ и систем»
Тема: Разработка собственного прерывания.

Студент гр. 9382

Михайлов Д.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Знакомство с организацией прерываний в компьютере. Написание собственного обработчика прерываний.

Задание.

Вариант 4А

Разработать обработчик прерывания с вектором: 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек. Вывести сообщение на экран.

Ход работы.

Используемые функции сервиса DOS:

- **25H: установить вектор прерывания**

Вход: AH = 25H, AL = номер прерывания, DS:DX = вектор прерывания:
адрес программы обработки прерывания

- **35H: дать вектор прерывания**

Вход: AH = 35H, AL = номер прерывания (00H до 0FFH)

Выход: ES:BX = адрес обработчика прерывания

- **09H: Выдать строку на дисплей**

Вход: AH = 09H, DS:DX = адрес строки, заканчивающейся символом '\$'

Используемые прерывания:

- **INT 21H: сервис DOS**

- **INT 16H: сервис клавиатуры**

Вход: AH = 00H читать (ожидать) следующую нажатую клавишу

Выход: AL = ASCII символ (если AL=0, AH содержит расширенный код ASCII) AH = сканкод или расширенный код ASCII

1. Обработчик прерывания – это отдельная процедура, имеющая структуру:

```
SUBR_INT PROC FAR
```

```
    PUSH AX ; сохранение изменяемых регистров
```

```
    ...
```

```

<действия по обработке прерывания>
POP AX ; восстановление регистров
...
MOV AL, 20H
OUT 20H,AL
IRET
SUBR_INT ENDP

```

Выделенные строки необходимы для разрешения обработки прерываний с более низкими уровнями, чем только что обработанное. Чтобы не создавать нового сегмента, можно поместить обработчик в сегмент кода.

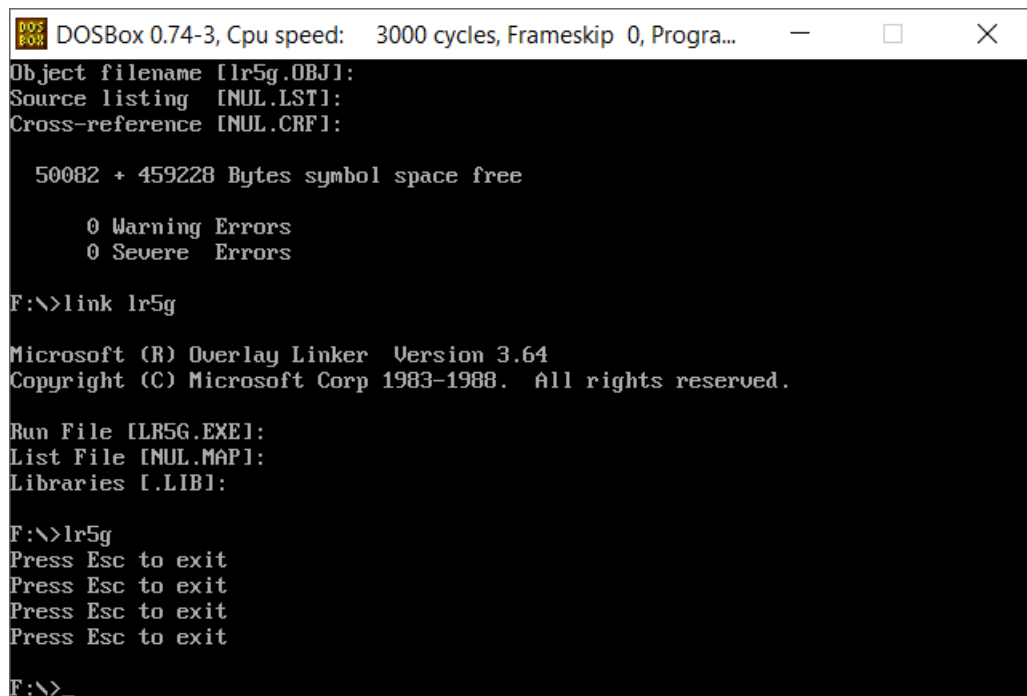
2. В программе изменяется изначальный вектор прерывания на новый, поэтому в конце программы необходимо восстановить оригинальный вектор прерывания. С помощью функции 35H прерывания 21H возможно получить информацию о обработчике прерывания, номер которого находится в AL. В регистры ES и BX помещаются значения сегмента и смещения прерывания в сегменте соответственно.

3. Далее с помощью функции 25H прерывания 21H устанавливается адрес нового прерывания. Для этого необходимо в DS поместить адрес сегмента с новым обработчиком, а в DX – смещение обработчика в этом сегменте.

4. В конце программы с помощью той же функции 25H восстанавливаются изначальные значения вектора прерывания, сохраненные ранее.

5. При запуске программа сразу же завершится и не напечатает никаких строк на экран. Необходимо добавить код для того, чтобы программа оставалась открытой. Для этого удобно воспользоваться функцией 0H прерывания 16H. Данная функция ожидает нажатия клавиши и при нажатии помещает в AL ASCII код символа, а в AH скан-код клавиши. Выход из программы будет осуществляться с помощью клавиши Esc, которая имеет скан-код 01H. Реализация задержки представлена после метки STOPPER в приложении.

6. Результат работы программы



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Object filename [lr5g.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

50082 + 459228 Bytes symbol space free

0 Warning Errors
0 Severe Errors

F:\>link lr5g

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LR5G.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

F:\>lr5g
Press Esc to exit
Press Esc to exit
Press Esc to exit
Press Esc to exit

F:\>_
```

Рисунок 2. Результат работы программы

Вывод.

В результате выполнения данной лабораторной работы были получены навыки написания собственного обработчика прерывания.

ПРИЛОЖЕНИЕ

ЛИСТИНГ ПРОГРАММЫ

#Microsoft (R) Macro Assembler Version 5.10

11/18/20 19:15:1
Page 1-1

```
0000          ASTACK SEGMENT STACK
0000 0400[          DB 1024 DUP(0)
      00
      ]

0400          ASTACK ENDS

0000          DATA SEGMENT
0000 0000          SYSTEM_CS DW 0
0002 0000          SYSTEM_IP DW 0
0004 50 72 65 73 73 20 MESSAGE DB 'Press Esc to exit', 10, 13,
      '$'
      45 73 63 20 74 6F
      20 65 78 69 74 0A
      0D 24
0018 00          COUNT DB 0
0019          DATA ENDS

0000          CODE SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:ASTACK

0000          MAIN PROC FAR
          ; Инициализация
0000 1E          PUSH DS
0001 2B C0          SUB AX, AX
0003 50          PUSH AX
0004 B8 ---- R      MOV AX, DATA
0007 8E D8          MOV DS, AX
          ; Сохранение системно
          го обработчика прерывани

0009 B8 3508          MOV AX, 3508H ; Dos fn 35h get interrup
t, interrupt 08
000C CD 21          INT 21H ; ES:BX = system interrupt hand
ler adress
000E 89 1E 0002 R    MOV SYSTEM_IP, BX
0012 8C 06 0000 R    MOV SYSTEM_CS, ES
          ; Установка прерывани
          я
0016 1E          PUSH DS ; Сохранение DS
0017 BA 0041 R      MOV DX, OFFSET MY_INT ; Установ

001A B8 ---- R      MOV AX, SEG MY_INT
001D 8E D8          MOV DS, AX ; Установка сег
```

```

001F B8 2508          MOV AX, 2508H ; Dos fn 25h set interrup
t, interrupt 08, DS:DX - my inturrept
0022 CD 21          INT 21H
0024 1F             POP DS ; Восстановление D
S
; Бесконечный цикл
0025 STOPPER: ; Считывание символа с клавиатуры
0025 2A E4          SUB AH, AH ; AH = 0
0027 CD 16          INT 16H ; AX = полученный с
0029 80 FC 01       CMP AH, 01H
002C 75 F7          JNE STOPPER ; Бесконеч
; Восстановление стартового прерывания
002E FA          CLI ; Clear interrupt flag (disable int
errupts)
002F 1E          PUSH DS ; Сохранение DS
0030 8B 16 0002 R   MOV DX, SYSTEM_IP ; Установка Оффсета
прерывания
0034 A1 0000 R     MOV AX, SYSTEM_CS
0037 8E D8          MOV DS, AX ; Установка сег
0039 B8 2508          MOV AX, 2508H ; Dos fn 25h set interrup
t, interrupt 08, DS:DX - my inturrept
003C CD 21          INT 21H
003E 1F             POP DS ; Восстановление DS
003F FB          STI ; Set interrupt flag (enable interr
upts)
0040 CB          RET
0041 MAIN ENDP
0041 MY_INT PROC FAR ; обработчик прерываний
0041 80 3E 0018 R 10  cmp COUNT,16 ; каждую секунду выводить 4
сообщения
0046 75 10          jne sec
0048 52          PUSH DX ; сохранение регистров
0049 50          PUSH AX
004A BA 0004 R     MOV DX, OFFSET MESSAGE ; вывод с
004D B4 09          MOV AH, 9H ; Dos fn 09h print string, D
S:DX - my string
004F CD 21          INT 21H
0051 C6 06 0018 R 00  mov COUNT, 0 ; COUNT = 1
0056 5A          POP DX ; восстановление
0057 58          POP AX
0058 sec:

```

```
0058 FE 06 0018 R      inc COUNT ; COUNT += 1
005C B0 20             MOV AL, 20H ; Low-level interruptions
005E E6 20             OUT 20H, AL ; Low-level interruptions
0060 CF               IRET
0061                 MY_INT ENDP

0061                 CODE ENDS

                        END MAIN
```

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0400	PARA	STACK
CODE	0061	PARA	NONE
DATA	0019	PARA	NONE

Symbols:

N a m e	Type	Value	Attr
COUNT	L BYTE	0018	DATA
MAIN	F PROC	0000	CODE Length = 0041
MESSAGE	L BYTE	0004	DATA
MY_INT	F PROC	0041	CODE Length = 0020
SEC	L NEAR	0058	CODE
STOPPER	L NEAR	0025	CODE
SYSTEM_CS	L WORD	0000	DATA
SYSTEM_IP	L WORD	0002	DATA
@CPU	TEXT	0101h	
@FILENAME	TEXT	lr5	
@VERSION	TEXT	510	

80 Source Lines
80 Total Lines
16 Symbols

48010 + 459247 Bytes symbol space free

0 Warning Errors
0 Severe Errors