Отчет

по лабораторной работе №5

по дисциплине “Архитектура компьютера”

Тема: “Написание собственного прерывания”

Вариант №4

(Шифр 2А)

Студент гр. 5383 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Допира В. Е.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кирьянчиков В.А.

**Цель работы**

Написать собственное прерывание.

Шифр 2А.

Цифра в шифре задает номер и назначение заменяемого вектора прерывания:

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

Буква определяет действия, реализуемые программой обработки прерываний:

А - Печать сообщения на экране.

**Теоретические сведения**

Прерывание - это процесс вызова процедур для выполнения некоторой задачи, обычно связанной с обслуживанием некоторых устройств (обработка сигнала таймера, нажатия клавиши и т.д.). Когда возникает прерывание, процессор прекращает выполнение текущей программы (если ее приоритет ниже) и запоминает в стеке вместе с регистром флагов адрес возврата(CS:IP) - места, с которого будет продолжена прерванная программа. Затем в CS:IP загружается адрес программы обработки прерывания и ей передается управление. Адреса 256 программ обработки прерываний, так называемые векторы прерывания, имеют длину по 4 байта (в первых двух хранится значение IP , во вторых - CS) и хранятся в младших 1024 байтах памяти. Программа обработки прерывания должна заканчиваться инструкцией IRET (возврат из прерывания), по которой из стека восстанавливается адрес возврата и регистр флагов.

Программа обработки 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

Две последние строки необходимы для разрешения обработки прерываний с более низкими уровнями, чем только что обработанное.

Замечание: в лабораторной работе действиями по обработке прерывания может быть вывод на экран некоторого текста или включение/выключение звукового сигнала. Программа, использующая новые программы обработки прерываний при своем завершении должна восстанавливать оригинальные векторы прерываний. Функция 35 прерывания 21H возвращает текущее значение вектора прерывания, помещая значение сегмента в ES, а смещение в BX. В этом случае программа должна содержать следующие инструкции:

; -- в сегменте данных

KEEP\_CS DW 0 ; для хранения сегмента

KEEP\_IP DW 0 ; и смещения прерывания

; -- в начале программы

MOV AH, 35H ; функция получения вектора

MOV AL, 1CH ; номер вектора

INT 21H

MOV KEEP\_IP, BX ; запоминание смещения

MOV KEEP\_CS, ES ; и сегмента

Для установки написанного прерывания в поле векторов прерываний используется функция 25H прерывания 21H, которая устанавливает вектор прерывания на указанный адрес.

PUSH DS

MOV DX, OFFSET ROUT ; смещение для процедуры в DX

MOV AX, SEG ROUT ; сегмент процедуры

MOV DS, AX ; помещаем в DS

MOV AH, 25H ; функция установки вектора

MOV AL, 60H ; номер вектора

INT 21H ; меняем прерывание

POP DS

В конце программы восстанавливается старый вектор прерывания

CLI

PUSH DS

MOV DX, KEEP\_IP

MOV AX, KEEP\_CS

MOV DS, AX

MOV AH, 25H

MOV AL, 1CH

INT 21H ; восстанавливаем вектор

POP DS

STI

**Код программы**

DOSSEG

.MODEL SMALL

.STACK 100h

.DATA

KEEP\_CS DW 0 ; для хранения сегмента

KEEP\_IP DW 0 ; и смещения прерывания

Greeting DB 'Hello. Interruption is actuated.',13,10,'$'

.CODE

begin:

mov bx,@data

mov ds,bx

MOV AH, 35H ; функция получения вектора

MOV AL, 60H ; номер вектора

INT 21H

MOV KEEP\_IP, BX ; запоминание смещения

MOV KEEP\_CS, ES ; и сегмента

PUSH DS

MOV DX, OFFSET ROUT ; смещение для процедуры в DX

MOV AX, SEG ROUT ; сегмент процедуры

MOV DS, AX ; помещаем в DS

MOV AH, 25H ; функция установки вектора

MOV AL, 60H ; номер вектора

INT 21H ; меняем прерывание

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 ; восстанавливаем вектор

POP DS

STI

mov ah,4ch

int 21h

ROUT proc far ;начало процедуры

push ax ;сохраняем все изменяемые регистры

push dx ;сохраняем все изменяемые регистры

mov ah,9h ;функция установки вектора

mov dx,offset Greeting ;в dx загружаем адрес сообщения

int 21h ;вывод строки на экран

pop dx ;восстанавливаем регистры

pop ax ;восстанавливаем регистры

ret

ROUT endp

END begin

**Файл листинга**

DOSSEG

.MODEL SMALL   
.STACK 100h   
.DATA   
0000 0000 KEEP\_CS DW 0 ;   
0002 0000 KEEP\_IP DW 0 ;   
  
0004 48 65 6C 6C 6F 2E Greeting DB 'Hello. Interruption is actuated.',13,10,'$'  
20 49 6E 74 65 72   
72 75 70 74 69 6F   
6E 20 69 73 20 61   
63 74 75 61 74 65   
64 2E 0D 0A 24   
  
.CODE  
0000 begin:   
0000 BB ---- R mov bx,@data   
0003 8E DB mov ds,bx   
  
0005 B4 35 MOV AH, 35H ;   
  
0007 B0 60 MOV AL, 60H ;   
0009 CD 21 INT 21H  
000B 89 1E 0002 R MOV KEEP\_IP, BX ;   
  
000F 8C 06 0000 R MOV KEEP\_CS, ES ;   
0013 1E PUSH DS  
0014 BA 003C R MOV DX, OFFSET ROUT ; DX  
0017 B8 ---- R MOV AX, SEG ROUT ;   
  
001A 8E D8 MOV DS, AX ; DS  
001C B4 25 MOV AH, 25H ;   
  
001E B0 60 MOV AL, 60H ;   
0020 CD 21 INT 21H ;   
0022 1F POP DS  
  
0023 CD 60 int 60h  
  
0025 FA CLI  
0026 1E PUSH DS  
0027 8B 16 0002 R MOV DX, KEEP\_IP  
002B A1 0000 R MOV AX, KEEP\_CS  
002E 8E D8 MOV DS, AX  
0030 B4 25 MOV AH, 25H  
  
0032 B0 60 MOV AL, 60H  
0034 CD 21 INT 21H ;   
  
0036 1F POP DS  
0037 FB STI  
  
0038 B4 4C mov ah,4ch   
003A CD 21 int 21h   
003C ROUT proc far ;

003C 50 push ax ;   
003D 52 push dx ;   
003E B4 09 mov ah,9h ;   
  
0040 BA 0004 R mov dx,offset Greeting ; dx   
  
0043 CD 21 int 21h ;   
  
0045 5A pop dx ;   
  
0046 58 pop ax ;   
  
0047 CB ret  
0048 ROUT endp  
END begin  
Microsoft (R) Macro Assembler Version 5.10 11/22/16 22:49:2  
Symbols-1  
  
Segments and Groups:  
  
N a m e Length Align Combine Class  
  
DGROUP . . . . . . . . . . . . . GROUP  
\_DATA . . . . . . . . . . . . 0027 WORD PUBLIC 'DATA'  
STACK . . . . . . . . . . . . 0100 PARA STACK 'STACK'  
\_TEXT . . . . . . . . . . . . . 0048 WORD PUBLIC 'CODE'  
  
Symbols:   
  
N a m e Type Value Attr  
  
BEGIN . . . . . . . . . . . . . L NEAR 0000 \_TEXT  
GREETING . . . . . . . . . . . . L BYTE 0004 \_DATA  
  
KEEP\_CS . . . . . . . . . . . . L WORD 0000 \_DATA  
KEEP\_IP . . . . . . . . . . . . L WORD 0002 \_DATA  
  
ROUT . . . . . . . . . . . . . . F PROC 003C \_TEXT Length = 000C  
  
@CODE . . . . . . . . . . . . . TEXT \_TEXT   
@CODESIZE . . . . . . . . . . . TEXT 0   
@CPU . . . . . . . . . . . . . . TEXT 0101h   
@DATASIZE . . . . . . . . . . . TEXT 0   
@FILENAME . . . . . . . . . . . TEXT l5   
@VERSION . . . . . . . . . . . . TEXT 510   
  
62 Source Lines  
62 Total Lines  
22 Symbols  
  
47976 + 461315 Bytes symbol space free  
  
0 Warning Errors  
0 Severe Errors

**Карта памяти**

Start Stop Length Name Class

00000H 00047H 00048H \_TEXT CODE

00048H 0006EH 00027H \_DATA DATA

00070H 0016FH 00100H STACK STACK

Origin Group

0004:0 DGROUP

Program entry point at 0000:0000

**Вывод**

В данной лабораторной работе написано прерывание пользователя, которое генерируется в программе на языке Assembler, и при реализации обработки выводит сообщение «Hello. Interruption is actuated.» на экран.

Также были изучены процесс прерывания и назначение заменяемого вектора.