

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. 9382

Пя С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить команды для работы с прерываниями в ассемблере, написать собственное прерывание.

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 обработчика прерывания, указанные перед командой IRET выхода из прерывания, 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, вставка цикла задержки в вывод сообщения или включение 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 соответствии с этим, программа должна содержать следующие инструкции:

; -- в сегменте данных

KEEP_CS DW 0 ; для хранения сегмента

KEEP_IP DW 0 ; и смещения вектора 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 вектора прерывания

Для установки адреса нового обработчика прерывания в поле векторов прерываний используется функция 25H прерывания 21H, которая помещает заданные адреса сегмента и смещения обработчика в вектор прерывания с заданным номером.

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

Далее может выполняться вызов нового обработчика прерывания. 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 старый вектор прерывания

POP DS

STI

Задание:

10 Вариант – 4А

4 - 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек.

А - Печать сообщения на экране;

Ход работы:

При разработке программы были использованы следующие команды:

Инструкция OUT выводит данные из регистра AL или AX (ИСТОЧНИК) в порт ввода-вывода. Номер порта должен быть указан в ПРИЁМНИКЕ.

Тестирование.

Вводные данные	Результат
	I love you!

	I love you!
	I love you!
	I love you!
	I love you!
	I love you!
	I love you!
	I love you!

Выводы.

В результате выполнения лабораторной работы был разработан код, определяющий собственное прерывание. Были улучшены навыки письма в ассемблере.

Приложение.

Текст файла *HELLO1.LST*

__Microsoft (R) Macro Assembler Version 5.10

11/14/20 02:37:1
Page 1-1

```

0000          AStack      SEGMENT  STACK
0000  0400[          DW 1024 DUP(?)
          ????
          ]

0800          AStack      ENDS
0000          DATA SEGMENT
0000  0000          KEEP_CS DW 0
0002  0000          KEEP_IP DW 0
0004  49 20 6C 6F 76 65  message db 'I love you!',10,13,'$' ;строка
          для сообщения
          20 79 6F 75 21 0A
          0D 24

0012          DATA ENDS
0000          CODE          SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:AStack

0000          Output PROC FAR
0000  50          push ax
0001  52          push dx
0002  B4 09      mov  ah, 09h
0004  BA 0004 R  mov  dx, offset message
0007  CD 21      int  21h
0009  5A          pop  dx

```

```

000A 58                pop ax
000B B0 20            mov al,20h
000D E6 20            out 20h,al
000F CF                ired
0010                  Output ENDP

0010                  Main PROC FAR
0010 1E                push ds
0011 2B C0            sub ax,ax
0013 50                push ax
0014 B8 ---- R        mov ax,data
0017 8E D8            mov ds, ax

0019 B8 3523            mov ax,3523h
001C CD 21            INT 21H
001E 89 1E 0002 R     MOV KEEP_IP, BX ; запоминание смещения
0022 8C 06 0000 R     MOV KEEP_CS, ES ; и сегмента вектора прерывания

0026 1E                PUSH DS
0027 BA 0000 R        MOV DX, OFFSET Output ; смещение для процедуры в DX
002A B8 ---- R        MOV AX, SEG Output ; сегмент процедуры
002D 8E D8            MOV DS, AX ; помещаем в DS
002F B8 2508            mov ax,2508h
0032 CD 21            INT 21H ; меняем прерывание
0034 1F                POP DS
__Microsoft (R) Macro Assembler Version 5.10
11/14/20 02:37:1
Page 1-2

```

```

0035                  waiting:
0035 B4 01            mov ah,1h
0037 CD 21            int 21h
0039 3C 1B            cmp al,27
003B 75 13            jne nextstep

003D FA                CLI
003E 1E                PUSH DS
003F 8B 16 0002 R     MOV DX, KEEP_IP
0043 A1 0000 R        MOV AX, KEEP_CS
0046 8E D8            MOV DS, AX
0048 B8 2508            mov AX,2508h
004B CD 21            INT 21H ; восстанавливаем первый вектор прерывания
004D 1F                POP DS
004E FB                STI
004F CB                ret

0050                  nextstep:
0050 EB E3            jmp waiting

```

```

0052                  Main ENDP
0052                  CODE ENDS
                        END Main
__Microsoft (R) Macro Assembler Version 5.10
11/14/20 02:37:1
Symbols-1

```

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0800	PARA		STACK
CODE	0052	PARA		NONE
DATA	0012	PARA		NONE

Symbols:

N a m e	Type	Value	Attr	
KEEP_CS	L WORD	0000	DATA	
KEEP_IP	L WORD	0002	DATA	
MAIN	F PROC	0010	CODE	Length = 0042
MESSAGE	L BYTE	0004	DATA	
NEXTSTEP	L NEAR	0050	CODE	
OUTPUT	F PROC	0000	CODE	Length = 0010
WAITING	L NEAR	0035	CODE	
@CPU	TEXT	0101h		
@FILENAME	TEXT	HELLO1		
@VERSION	TEXT	510		

70 Source Lines
70 Total Lines
15 Symbols

48002 + 461305 Bytes symbol space free

0 Warning Errors
0 Severe Errors