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. 9382 _____ Докукин В.М.

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:

Вариант 7 – 4A

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

A - 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
	Sample Text Sample Text

	Sample Text Sample Text <-

Выводы.

В результате выполнения лабораторной работы были изучены принципы разработки собственных прерываний на языке Ассемблера; разработан код, определяющий собственное прерывание. Были улучшены навыки написания программ на языке Ассемблера.

Приложение.

Имя файла: INTRP.ASM

```
AStack    SEGMENT  STACK
          DW 1024 DUP(?) ; 1Kb of memory
AStack    ENDS
DATA SEGMENT
KEEP_CS DW 0
KEEP_IP DW 0
STACK_SS DW 0000
STACK_AX DW 0000
STACK_SP DW 0000
IStack DW 30 DUP(?)
mes db 'Sample Text',10,13,'$' ;строка для сообщения

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

Print PROC FAR
jmp start

   start:

   mov STACK_SP, sp
   mov STACK_AX, ax
   mov ax, ss
   mov STACK_SS, ax
   mov ax, IStack
   mov ss, ax
   mov ax, STACK_AX

   push ax
   push dx
   mov  ah, 09h
   mov  dx, offset mes
   int  21h
   pop  dx
   pop  ax

   mov STACK_AX, ax
   mov ax, STACK_SS
   mov ss, ax
   mov sp, STACK_SP
   mov ax, STACK_AX

   mov al, 20h
   out 20h, al
   iret
```

```
Print ENDP

Main PROC FAR
 push ds
 sub  ax,ax
 push ax
 mov  ax, data
 mov  ds, ax

 mov ah, 35h
 mov al, 23h
 int 21h
 mov KEEP_IP, bx ; запоминание смещения
 mov KEEP_CS, es ; и сегмента вектора прерывания

 push ds
 mov dx, offset Print ; смещение для процедуры в DX
 mov ax, seg Print    ; сегмент процедуры
 mov ds, ax           ; помещаем в DS
 mov ah, 25h
 mov al, 08h
 int 21h ; меняем прерывание
 pop ds

@wait:
 mov ah, 1h
 int    21h   ; считывание символа с клавиатуры
 cmp  al,27
 jne @wait

 cli
 push ds
 mov dx, KEEP_IP
 mov ax, KEEP_CS
 mov ds, ax
 mov ax, 2508h
 int 21h ; восстанавливаем старый вектор прерывания
 pop ds
 sti
 ret

Main ENDP
CODE ENDS
 END Main
```