

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. 9383

Гладких А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить основы работы с прерываниями. Применить на практике полученные знания о прерываниях, написав свое собственное.

Краткие 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

Текст задания.

Вариант 1В.

Разработать собственное прерывание.

1 - 1Ch - прерывание от часов - генерируется автоматически операционной системой

18 раз в сек;

В - Выдача звукового сигнала;

Ход работы.

В ходе работы была разработана программа, которая по вызову прерывания 1Ch воспроизводит звуковой сигнал.

Для этого было произведено сохранение сегмента и смещения изначального прерывания 1Ch в переменные keep_cs и keep_ip соответственно. После прерывание 1Ch было переопределено под вызов функции SUBR_INT, в которой происходит воспроизведение звука через встроенный динамик IBM PC.

После воспроизведения звука происходит восстановление старого вектора прерывания — обратно записываются сегмент и смещение, сохраненные в переменных программы.

Функция MAIN завершается операндом `ret`.

Исходный код программы представлен в приложении А.

Выводы.

Были изучены основы работы с прерываниями. Были применены на практике полученные знания о прерываниях, было написано собственное прерывание, воспроизводящее звуковой сигнал.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: source.asm

```

AStackSEGMENT      STACK
DB 256 DUP(?)
AStack  ENDS

DATA  SEGMENT
keep_cs DW 0 ; для хранения сегмента
keep_ip DW 0 ; и смещения прерывания
DATA      ENDS

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

SUBR_INT PROC FAR ;начало процедуры

push ax ;сохранение изменяемого регистра
push dx ;сохранение изменяемого регистра

mov al, 10110110b
out 43h, al
mov ax, 880 ; sound pitch
out 42h, al
mov al, ah
out 42h, al
in al, 61h
mov ah, al
or al, 3
out 61h, al
sub cx, cx
kill_time:
loop kill_time
mov al, ah
out 61h, al ; turn off

pop dx ;восстановление регистра
pop ax ;восстановление регистра
```

```
mov al, 20h ;разрешаем обработку прерываний
out 20h, al ;с более низкими уровнями
iret ;конец прерывания
```

```
SUBR_INT ENDP ;конец процедуры
```

```
MAIN PROC FAR
```

```
push ds
sub ax, ax
push ax
mov ax, DATA
mov ds, ax
```

```
mov ah, 35h
mov al, 1ch
int 21h
mov keep_ip, bx
mov keep_cs, es
```

```
push ds
mov dx, offset SUBR_INT
mov ax, seg SUBR_INT
mov ds, ax
mov ah, 25h
mov al, 1ch
int 21h
pop ds
```

```
int 1ch
```

```
;восстанавливаем 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
```

```
pop ds  
sti
```

```
ret
```

```
MAIN ENDP
```

```
CODE ENDS
```

```
END MAIN
```