

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

**ОТЧЕТ
по лабораторной работе №3
по дисциплине «Организация ЭВМ и системы»
Тема: Представление и обработка целых чисел. Организация
ветвящихся процессов**

Студента гр. 1383

Самулевич С.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

ЗАДАНИЕ.

Разработать собственное прерывание:

23h - прерывание, генерируемое при нажатии клавиш Control+C;

Выдача звукового сигнала с заданной высотой звука.

Выполнение работы.

В сегменте Astack выделяется 1Кбайт памяти, как и требуется в задании.

В сегменте кода определяется процедура пользовательского прерывания TIME.

С помощью команды push в стек записываются значения регистров до входа в прерывание.

При помощи определённых каналов портов 42H, 43H, 40H реализуется таймер для обеспечения определённой частоты звукового сигнала, которой выводится на динамик путём изменения 0 бита порта 62H.

В процедуре MAIN

вызывается функция 35h прерывания 21h, возвращающая значение вектора прерывания, лежащего по смещению 23h. Пользовательское прерывание SPEAKER записывается в прерывание 23h. Далее происходит вызов данного прерывания в случае считывания с клавиатуры комбинации клавиш ctrl+C.

После вызова прерывания происходит восстановление старого вектора прерывания и завершение программы.

Программный код см. в приложении А

Выводы.

В ходе работы было разработано собственное прерывание

Исходный код

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

```
DATA SEGMENT
```

```
KEEP_CS DW 0 ; для хранения сегмента
```

```
KEEP_IP DW 0 ; и смещения прерывания
```

```
DATA ENDS
```

```
AStack SEGMENT STACK
```

```
DW 512 DUP('0')
```

```
AStack ENDS
```

```
CODE SEGMENT
```

```
SPEAKER PROC FAR
```

```
sub AX, AX
```

```
sub CX, CX
```

```
in AL, 61H
```

```
and AL, 0feh
```

```
out 61h, AL
```

```
mov AL, 10110110b
```

out 43H, AL ; Set mode for 2nd channel

mov AX, 2000; Pitch of sound

out 42H, AL

mov AL, AH

out 42H, AL ; Set it to speaker port

in AL, 61H

mov AH, AL

or AL, 3

out 61H, AL ; Speaker control mode selection

sub CX, CX

KILL_TIME:

loop KILL_TIME ; Loop while speaker works

mov AL, AH

out 61H, AL ; Speaker turn off

mov al, 20h

out 20h, al

iret

SPEAKER ENDP

MAIN PROC FAR

push DS

sub AX, AX

push AX

mov AX, DATA

mov DS, AX

mov AX, 3523h

int 21h

mov KEEP_IP, BX

mov KEEP_CS, ES

```
push DS
mov DX, OFFSET SPEAKER
mov AX, SEG SPEAKER
mov DS, AX
mov AH, 25h
mov AL, 23h
int 21H
pop DS
```

```
readkey:
mov AH, 0
int 16h
cmp AL, 3
je do
jmp readkey
```

```
do:
int 23h
```

```
cli
push DS
mov DX, KEEP_IP
mov AX, KEEP_CS
mov DS, AX
mov AX, 2523h
int 21h
pop DS
sti
```

```
ret FAR
```

```
MAIN ENDP
CODE ENDS
```

END MAIN