НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №1

з дисципліни **«**Периферійні пристрої**»**

Виконав:

студент 3 курсу

ФІОТ гр. ІО-21

Кузьменко Володимир

Перевірив:

Марковський О. П.

Київ – 2015 р.

.486

.8087

;Delay before reading from timer

delaytoread equ 0FFFFh ;loops

MODEL SMALL

write macro port,symbol

mov dx,port

mov al,symbol

out dx,al

endm

read macro symbol,port

mov dx,port

in al,dx

mov symbol,al

endm

DATA SEGMENT use16

head db 0

tail db 0

res db 0

tmp db 0

sec dd 0

wassec dd 0

st db ' ',13,10,'$'

st\_pralt db 'Press ALT to start timer...',13,10,'$'

st\_pralt2 db 'Timing started, press ALT to stop timing...',13,10,'$'

st\_seconds db 'Seconds counted: ','$'

mult dw 0

laststatus db 0

DATA ENDS

CODE SEGMENT use16

ASSUME cs:CODE, ds:DATA, ss:STACK

;Delays for some time

Delay proc

push cx

mov cx,delaytoread

again:

loop again

pop cx

ret

Delay endp

;Waits timer to open and reads value from register to "res", register comes

;in "bl"

WaitAndRead proc

push ax

push dx

;wait

write 70h,0Ah

call Delay

nexttry:

read tmp,71h

test tmp,01000000b

jnz nexttry

;read

write 70h,bl

call Delay

read res,71h

pop dx

pop ax

ret

WaitAndRead endp

;Adds ("16-16" value)\*multiplier to "secs" variable, bl = value,

;edx = multiplier

AddToSec proc

push eax

push ecx

;Init all

xor eax,eax

xor cx,cx

push edx

;Higher part

mov al,bl

shr al,4

mov dl,10

mul dl

;Lower part

mov cl,bl

and cl,00001111b

add ax,cx

;Multiply

pop edx

mul edx

;Set result

mov ecx,sec

add eax,ecx

mov sec,eax

pop ecx

pop eax

ret

AddToSec endp

;Returns seconds from start of the day at "sec" variable

GetSecFromStartDay proc

push ax

push bx

push cx

push dx

;Init seconds

mov sec,0

;Add hours

;read

mov bl,4

call WaitAndRead

;add to total seconds

mov bl,res

mov edx,3600

call AddToSec

;Add minutes

;read

mov bl,2

call WaitAndRead

;add to total seconds

mov bl,res

mov edx,60

call AddToSec

;Add seconds

;read

mov bl,0

call WaitAndRead

;add to total seconds

mov bl,res

mov edx,1

call AddToSec

pop dx

pop cx

pop bx

pop ax

ret

GetSecFromStartDay endp

;Outputs number in "bx"

OutNumber proc

push ax

push bx

push cx

push dx

push di

mov mult,10000

mov di,0

mov cx,5

mov ax,bx

nextnum:

xor dx,dx

div mult

add al,'1'-1

mov st[di],al

inc di

mov ax,dx

push ax

xor dx,dx

xor ax,ax

mov ax,mult

mov bx,10

div bx

mov mult,ax

pop ax

loop nextnum

mov ah,09h

lea dx,st

int 21h

pop di

pop dx

pop cx

pop bx

pop ax

ret

OutNumber endp

;Outputs string, by offset that comes in dx

OutString proc

push ax

mov ah,09h

int 21h

pop ax

ret

OutString endp

;Counts presses of CTRL

WaitPress proc

push ax

push es

xor ax,ax

mov es,ax

;Set laststatus

mov al,es:[0417h]

and al,00001000b

mov laststatus,al

nextwait:

mov al,es:[0417h]

and al,00001000b

;Pressed ALT?

test al,00001000b

jz setzerostatus

;Was it pressed before?

mov ah,laststatus

test ah,00001000b

jnz nextwait

;Return - ALT pressed

jmp goreturn

setzerostatus:

mov laststatus,0

jmp nextwait

goreturn:

pop es

pop ax

ret

WaitPress endp

;----------Program---------------

begin:

mov dx,DATA

mov ds,dx

;clear screen

mov ax,3h

int 10h

;Out welcome message

lea dx,st\_pralt

call OutString

;Wait ALT

call WaitPress

;read seconds now

call GetSecFromStartDay

mov eax,sec

mov wassec,eax

;Out second message

lea dx,st\_pralt2

call OutString

;Wait ALT

call WaitPress

;Calc number of seconds passed

call GetSecFromStartDay

mov eax,sec

mov ebx,wassec

sub eax,ebx

mov bx,ax

;Out number of seconds

lea dx,st\_seconds

call OutString

call OutNumber

;Exit program

MOV AX,4C00h

INT 21h

CODE ends

STACK 200

END begin