**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: Написание собственного прерывания.

|  |  |  |
| --- | --- | --- |
| Студент гр. 1303 |  | Бутыло Е.А. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2022

## **Цель работы.**

Изучить особенности прерываний на языке Ассемблера, написать собственное прерывание.

**Задание.**

Написать прерывание 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек. Выполнить вывод сообщения на экран заданное число раз, после чего вставить фиксированную задержку и вывести сообщение о завершении обработчика. Отвести в программе под стек не менее 1К байт.

**Ход работы.**

В сегменте данных DATA содержится две переменных для хранения старого прерывания, содержавшегося по смещению 08h.

В сегменте Astack выделяется 1Кбайт памяти, то есть db 1024.

В сегменте кода сначала определяем процедуру для печати времени print\_cmos и процедуру пользовательского прерывания FUNC. В процедуре FUNC сначала сохраняются в стеке значения регистров до входа в прерывание. Далее выполняется чтение системного времени в порядке:

* Год
* Месяц
* День
* Час
* Минута
* Секунда

Формат вывода: “<год>-<месяц>-<день> <часы>h <минуты>:<секунды>”.

Далее из стека извлекаются сохранённые значения регистров.

Вызов прерывания происходит в процедуре MAIN. В переменных, указанных в сегменте данных, сохраняется старое прерывание. Новое прерывание FUNC записывается по смещению 08h, с помощью функции 25h прерывания 21h.

Исходный код программы см. в приложении А.

**Тестирование.**

Работа программы с заданными условиями представлена на рис. 1

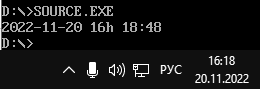


Рис.1 – Результат работы программы

**Вывод.**

В ходе выполнения лабораторной работы были изучены виды прерываний и работа с ними. В соответствии с задание было разработано собственное прерывание.

# **Приложение А Исходный код программы**

Название файла: source.asm

AStack SEGMENT STACK

DB 1024 DUP(?)

AStack ENDS

DATA SEGMENT

KEEP\_CS DW 0

KEEP\_IP DW 0

DATA ENDS

CODE SEGMENT

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

print\_cmos proc near

out 70h,al

in al,71h

push ax

mov cl, 4

shr al,cl

add al,'0'

int 29h

pop ax

and al,0Fh

add al,30h

int 29h

ret

print\_cmos endp

FUNC PROC FAR

push ax

push bx

push cx

push dx

push ds

mov al,0Bh

out 70h,al

in al,71h

and al,11111011b

out 71h,al

mov al,32h

call print\_cmos

mov al,9

call print\_cmos

mov al,'-'

int 29h

mov al,8

call print\_cmos

mov al,'-'

int 29h

mov al,7

call print\_cmos

mov al,' '

int 29h

mov al,4

call print\_cmos

mov al,'h'

int 29h

mov al,' '

int 29h

mov al,2

call print\_cmos

mov al,':'

int 29h

mov al,0h

call print\_cmos

pop ds

pop dx

pop cx

pop bx

pop ax

mov al, 20h

out 20h, al

iret

FUNC ENDP

MAIN PROC FAR

push ds

sub ax, ax

push ax

mov ax, DATA

mov ds, ax

mov ah, 35h

mov al, 08h

int 21h

mov KEEP\_IP, bx

mov KEEP\_CS, es

push ds

mov dx, OFFSET FUNC

mov ax, SEG FUNC

mov ds, ax

mov ah, 25h

mov al, 08h

int 21h

pop ds

int 08h

cli

push ds

mov dx, KEEP\_IP

mov ax, KEEP\_CS

mov ds, ax

mov ah, 25h

mov al, 08h

int 21h

pop ds

sti

mov ah, 4ch

int 21h

MAIN ENDP

CODE ENDS

END MAIN

# **Приложение Б Листинг программы**

Название файла: source.lst

Microsoft (R) Macro Assembler Version 5.10 11/20/22 16:24:1

Page 1-1

0000 AStack SEGMENT STACK

0000 0400[ DB 1024 DUP(?)

??

]

0400 AStack ENDS

0000 DATA SEGMENT

0000 0000 KEEP\_CS DW 0

0002 0000 KEEP\_IP DW 0

0004 DATA ENDS

0000 CODE SEGMENT

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

0000 print\_cmos proc near

0000 E6 70 out 70h,al ; поэ

лать AL в индексный порт CMO

S

0002 E4 71 in al,71h ; прЍ

¾читать данные

0004 50 push ax

0005 B1 04 mov cl, 4

0007 D2 E8 shr al,cl ; выЍ

´елить старшие четыре битЍ

°

0009 04 30 add al,'0' ; доЍ

±авить ASCII-код цифры 0

000B CD 29 int 29h ; выЍ

²ести на экран

000D 58 pop ax

000E 24 0F and al,0Fh ; выЍ

´елить младшие четыре битЍ

°

0010 04 30 add al,30h ; доЍ

±авить ASCII-код цифры 0

0012 CD 29 int 29h ; выЍ

²ести на экран

0014 C3 ret

0015 print\_cmos endp

0015 FUNC PROC FAR

0015 50 push ax

0016 53 push bx

0017 51 push cx

0018 52 push dx

0019 1E push ds

001A B0 0B mov al,0Bh

; CMOS OBh - управляющий регис

тр В

001C E6 70 out 70h,al ; поэ

т 70h - индекс CMOS

001E E4 71 in al,71h ; поэ

Microsoft (R) Macro Assembler Version 5.10 11/20/22 16:24:1

Page 1-2

т 71h - данные CMOS

0020 24 FB and al,11111011b ; обЍ

½улить бит 2 (форма чисел - B

CD)

0022 E6 71 out 71h,al ; и з

аписать обратно

0024 B0 32 mov al,32h ; CMOS

32h - две старшие цифры года

0026 E8 0000 R call print\_cmos ; выЍ

²од на экран

0029 B0 09 mov al,9 ; CMOS

09h - две младшие цифры года

002B E8 0000 R call print\_cmos

002E B0 2D mov al,'-' ; миЍ

½ус

0030 CD 29 int 29h ; выЍ

²од на экран

0032 B0 08 mov al,8 ; CMOS

08h - текущий месяц

0034 E8 0000 R call print\_cmos

0037 B0 2D mov al,'-' ; ещЍ

µ один минус

0039 CD 29 int 29h

003B B0 07 mov al,7 ; CMOS

07h - день

003D E8 0000 R call print\_cmos

0040 B0 20 mov al,' ' ; прЍ

¾бел

0042 CD 29 int 29h

0044 B0 04 mov al,4 ; CMOS

04h - час

0046 E8 0000 R call print\_cmos

0049 B0 68 mov al,'h' ; буЍ

ºва "h"

004B CD 29 int 29h

004D B0 20 mov al,' ' ; прЍ

¾бел

004F CD 29 int 29h

0051 B0 02 mov al,2 ; CMOS

02h - минута

0053 E8 0000 R call print\_cmos

0056 B0 3A mov al,':' ; двЍ

¾еточие

0058 CD 29 int 29h

005A B0 00 mov al,0h ; CMOS

00h - секунда

005C E8 0000 R call print\_cmos

005F 1F pop ds

0060 5A pop dx

0061 59 pop cx

0062 5B pop bx

0063 58 pop ax

0064 B0 20 mov al, 20h

Microsoft (R) Macro Assembler Version 5.10 11/20/22 16:24:1

Page 1-3

0066 E6 20 out 20h, al

0068 CF iret

0069 FUNC ENDP

0069 MAIN PROC FAR

0069 1E push ds

006A 2B C0 sub ax, ax

006C 50 push ax

006D B8 ---- R mov ax, DATA

0070 8E D8 mov ds, ax

0072 B4 35 mov ah, 35h

0074 B0 08 mov al, 08h

0076 CD 21 int 21h

0078 89 1E 0002 R mov KEEP\_IP, bx

007C 8C 06 0000 R mov KEEP\_CS, es

0080 1E push ds

0081 BA 0015 R mov dx, OFFSET FUNC

0084 B8 ---- R mov ax, SEG FUNC

0087 8E D8 mov ds, ax

0089 B4 25 mov ah, 25h

008B B0 08 mov al, 08h

008D CD 21 int 21h

008F 1F pop ds

0090 CD 08 int 08h

0092 FA cli

0093 1E push ds

0094 8B 16 0002 R mov dx, KEEP\_IP

0098 A1 0000 R mov ax, KEEP\_CS

009B 8E D8 mov ds, ax

009D B4 25 mov ah, 25h

009F B0 08 mov al, 08h

00A1 CD 21 int 21h

00A3 1F pop ds

00A4 FB sti

00A5 B4 4C mov ah, 4ch

00A7 CD 21 int 21h

00A9 MAIN ENDP

00A9 CODE ENDS

END MAIN

Microsoft (R) Macro Assembler Version 5.10 11/20/22 16:24:1

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

ASTACK . . . . . . . . . . . . . 0400 PARA STACK

CODE . . . . . . . . . . . . . . 00A9 PARA NONE

DATA . . . . . . . . . . . . . . 0004 PARA NONE

Symbols:

N a m e Type Value Attr

FUNC . . . . . . . . . . . . . . F PROC 0015 CODE Length = 0054

KEEP\_CS . . . . . . . . . . . . L WORD 0000 DATA

KEEP\_IP . . . . . . . . . . . . L WORD 0002 DATA

MAIN . . . . . . . . . . . . . . F PROC 0069 CODE Length = 0040

PRINT\_CMOS . . . . . . . . . . . N PROC 0000 CODE Length = 0015

@CPU . . . . . . . . . . . . . . TEXT 0101h

@FILENAME . . . . . . . . . . . TEXT SOURCE

@VERSION . . . . . . . . . . . . TEXT 510

115 Source Lines

115 Total Lines

13 Symbols

48006 + 457204 Bytes symbol space free

0 Warning Errors

0 Severe Errors