**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: **Написание собственного прерывания.**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1303 |  | Хабибуллина А.М. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2022

## **Цель работы.**

* 1. Написать собственное прерывание в соответствии с вариантом на языке Ассемблер.
  2. Задание.

Вариант 25: 4D

4 - 16h - прерывание от клавиатуры (по заданному скан-коду клавиши делать действия A-G, указанные ниже).

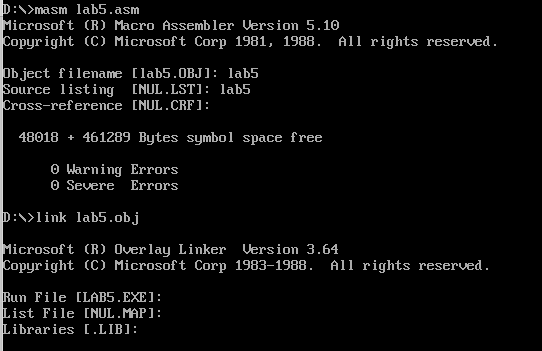
D - Выполнить чтение и вывод на экран отсчета системных часов (в тиках, где 1 тик = 55 мсек).

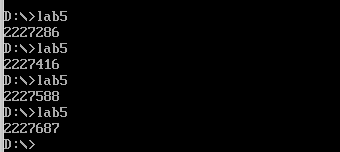
Выполнение работы

В Main загружается в DS адрес начала сегмента данных, сохраняется сегмент и смещение текущего прерывания с помощью функции 35 прерывания 21H. Функция 25H прерывания 21H устанавливает вектор прерывания 16H, после чего вызывается созданное прерывание. После всех процедур необходимо восстановить старый вектор прерывания.

SUBR\_INT – процедура прерывания. Все изменяемые переменные сохраняются в стек. В регистр AH записывается функция чтения часов 00h, вызывается прерывание 1ah, которое записывает время в регистры cx и dx, для которых вызывается функция Timer. После регистры восстанавливаются и происходит возврат из прерывания. Функция Timer конвертирует время в символы. В регистр CX записывается ноль, данный регистр будет хранить количество символов. В регистр BX записывается основание системы счисления. Пока AX не равен нулю делим на BX, преобразовываем остатки от деления и складываем в стек. Увеличиваем значение регистра CX на один. В регистр AH записываем значения 02H для вывода на экран всех символов из стека.

**Тестирование**

****

****

* 1. Выводы

В ходе выполнения лабораторной работы были получены навыки написания программы с собственным прерыванием на языке Ассемблер.

Приложение А  
Исходный код программы

Название файла: lab5.asm

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

STACK SEGMENT STACK

DW 1024 DUP(?)

STACK ENDS

DATA SEGMENT

KEEP\_CS DW 0

KEEP\_IP DW 0

DATA ENDS

CODE SEGMENT

Timer PROC

push dx

push cx

xor cx, cx

mov bx, 10

proc1:

xor dx, dx

div bx

push dx

inc cx

test ax, ax

jnz proc1

mov ah, 2

proc2:

pop dx

add dl, '0'

int 21h

loop proc2

pop cx

pop dx

ret

Timer endp

SUBR\_INT PROC FAR

JMP start

save\_sp DW 0000h

save\_ss DW 0000h

INT\_STACK DB 40 DUP(0)

start:

mov save\_sp, sp

mov save\_ss, ss

mov sp, SEG INT\_STACK

mov ss, sp

mov sp, offset start

push ax

push cx

push dx

mov ah, 00h

int 1AH

mov ax, cx

call Timer

mov ax, dx

call Timer

pop dx

pop cx

pop ax

mov ss, save\_ss

mov sp, save\_sp

mov al, 20h

out 20h, al

iret

SUBR\_INT ENDP

Main PROC FAR

push DS

sub AX,AX

push AX

mov AX,DATA

mov DS,AX

MOV AH, 35H

MOV AL, 16H

INT 21H

MOV KEEP\_IP, BX

MOV KEEP\_CS, ES

try:

mov ah, 0

int 16h

cmp ah, 39h

jnz try

PUSH DS

MOV DX, OFFSET SUBR\_INT

MOV AX, SEG SUBR\_INT

MOV DS, AX

MOV AH, 25H

MOV AL, 16H

INT 21H

POP DS

int 16H

CLI

PUSH DS

MOV DX, KEEP\_IP

MOV AX, KEEP\_CS

MOV DS, AX

MOV AH, 25H

MOV AL, 16H

INT 21H

POP DS

STI

MOV AH, 4Ch

INT 21h

Main ENDP

CODE ENDS

END Main

1. Приложение Б  
   Листинг файл исходной программы

lab5.lst:

Microsoft (R) Macro Assembler Version 5.10 11/24/22 22:24:2

Page 1-1

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

0000 STACK SEGMENT STACK

0000 0400[ DW 1024 DUP(?)

????

]

0800 STACK ENDS

0000 DATA SEGMENT

0000 0000 KEEP\_CS DW 0

0002 0000 KEEP\_IP DW 0

0004 DATA ENDS

0000 CODE SEGMENT

0000 Timer PROC

0000 52 push dx

0001 51 push cx

0002 33 C9 xor cx, cx

0004 BB 000A mov bx, 10

0007 proc1:

0007 33 D2 xor dx, dx

0009 F7 F3 div bx

000B 52 push dx

000C 41 inc cx

000D 85 C0 test ax, ax

000F 75 F6 jnz proc1

0011 B4 02 mov ah, 2

0013 proc2:

0013 5A pop dx

0014 80 C2 30 add dl, '0'

0017 CD 21 int 21h

0019 E2 F8 loop proc2

001B 59 pop cx

001C 5A pop dx

001D C3 ret

001E Timer endp

001E SUBR\_INT PROC FAR

001E EB 2D 90 JMP start

0021 0000 save\_sp DW 0000h

0023 0000 save\_ss DW 0000h

0025 0028[ INT\_STACK DB 40 DUP(0)

00

Microsoft (R) Macro Assembler Version 5.10 11/24/22 22:24:2

Page 1-2

]

004D start:

004D 2E: 89 26 0021 R mov save\_sp, sp

0052 2E: 8C 16 0023 R mov save\_ss, ss

0057 BC ---- R mov sp, SEG INT\_STACK

005A 8E D4 mov ss, sp

005C BC 004D R mov sp, offset start

005F 50 push ax

0060 51 push cx

0061 52 push dx

0062 B4 00 mov ah, 00h

0064 CD 1A int 1AH

0066 8B C1 mov ax, cx

0068 E8 0000 R call Timer

006B 8B C2 mov ax, dx

006D E8 0000 R call Timer

0070 5A pop dx

0071 59 pop cx

0072 58 pop ax

0073 2E: 8E 16 0023 R mov ss, save\_ss

0078 2E: 8B 26 0021 R mov sp, save\_sp

007D B0 20 mov al, 20h

007F E6 20 out 20h, al

0081 CF iret

0082 SUBR\_INT ENDP

0082 Main PROC FAR

0082 1E push DS

0083 2B C0 sub AX,AX

0085 50 push AX

0086 B8 ---- R mov AX,DATA

0089 8E D8 mov DS,AX

008B B4 35 MOV AH, 35H

008D B0 16 MOV AL, 16H

008F CD 21 INT 21H

0091 89 1E 0002 R MOV KEEP\_IP, BX

0095 8C 06 0000 R MOV KEEP\_CS, ES

0099 try:

0099 B4 00 mov ah, 0

009B CD 16 int 16h

009D 80 FC 39 cmp ah, 39h

00A0 75 F7 jnz try

Microsoft (R) Macro Assembler Version 5.10 11/24/22 22:24:2

Page 1-3

00A2 1E PUSH DS

00A3 BA 001E R MOV DX, OFFSET SUBR\_INT

00A6 B8 ---- R MOV AX, SEG SUBR\_INT

00A9 8E D8 MOV DS, AX

00AB B4 25 MOV AH, 25H

00AD B0 16 MOV AL, 16H

00AF CD 21 INT 21H

00B1 1F POP DS

00B2 CD 16 int 16H

00B4 FA CLI

00B5 1E PUSH DS

00B6 8B 16 0002 R MOV DX, KEEP\_IP

00BA A1 0000 R MOV AX, KEEP\_CS

00BD 8E D8 MOV DS, AX

00BF B4 25 MOV AH, 25H

00C1 B0 16 MOV AL, 16H

00C3 CD 21 INT 21H

00C5 1F POP DS

00C6 FB STI

00C7 B4 4C MOV AH, 4Ch

00C9 CD 21 INT 21h

00CB Main ENDP

00CB CODE ENDS

END Main

Microsoft (R) Macro Assembler Version 5.10 11/24/22 22:24:2

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

CODE . . . . . . . . . . . . . . 00CB PARA NONE

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

STACK . . . . . . . . . . . . . 0800 PARA STACK

Symbols:

N a m e Type Value Attr

INT\_STACK . . . . . . . . . . . L BYTE 0025 CODE Length = 0028

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

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

MAIN . . . . . . . . . . . . . . F PROC 0082 CODE Length = 0049

PROC1 . . . . . . . . . . . . . L NEAR 0007 CODE

PROC2 . . . . . . . . . . . . . L NEAR 0013 CODE

SAVE\_SP . . . . . . . . . . . . L WORD 0021 CODE

SAVE\_SS . . . . . . . . . . . . L WORD 0023 CODE

START . . . . . . . . . . . . . L NEAR 004D CODE

SUBR\_INT . . . . . . . . . . . . F PROC 001E CODE Length = 0064

TIMER . . . . . . . . . . . . . N PROC 0000 CODE Length = 001E

TRY . . . . . . . . . . . . . . L NEAR 0099 CODE

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

@FILENAME . . . . . . . . . . . TEXT lab5

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

131 Source Lines

131 Total Lines

20 Symbols

48018 + 461289 Bytes symbol space free

0 Warning Errors

0 Severe Errors