**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: **Разработка собственного прерывания.**

.

|  |  |  |
| --- | --- | --- |
| Студентка гр. 9382 |  | Балаева М.О. |
| Преподаватель |  | Ефремов М,А. |

Санкт-Петербург

2020

**Цель работы.**

2С.Написать прерывание int 60h, прерывание пользователя - должно генерироваться в программе. Приостановить вывод на экран (вставить цикл задержки). Под стек отвести не менее 1Кб**.**

**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.

**Тестирование.**

При запуске программы на экране пользователя выводится сообщение "Первая строка выводится без задержки". Далее, согласно условию задачи, происходит задержка на 2 секунды и последующий вывод строки "А вторая через 2 секунды". Результаты работы программы представлены на рис.1.



Рис.1

**Выводы.**

При выполнении программы были получены результаты, совпадающие с требованиями условия задачи. В ходе выполнения данной лабораторной работы был изучены способы создания, обработки и вызова прерываний в языке Assembler. Получены сведения о реализации сравнения, меток и перехода по ним, ветвления в программе на языке Ассемблера.

**Приложение А. Код программы**

**Lab5.asm**

AStack SEGMENT STACK

DW 1500 DUP(?)

AStack ENDS

DATA SEGMENT

keepcs DW 0; для хранения сегмента

keepip DW 0; и смещения прерывания

firstString LABEL BYTE

DB 'Первая строка выводится без задержки',13,10,'$'

secondString LABEL BYTE

DB 'А вторая через 2 секунды',13,10,'$'

DATA ENDS

CODE SEGMENT

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

Main PROC FAR

mov ah,35h

mov al,60h

int 21h

mov keepip,bx

mov keepcs,es

push ds

mov dx, OFFSET delayInt; смещение для процедуры в DX

mov ax, SEG delayInt; сегмент процедуры

mov ds,ax

mov ah,25h; функция установки вектора

mov al,60h; номер вектора

int 21h

pop ds

mov ax,data

mov ds,ax

mov dx,OFFSET firstString; вывод первой строки

mov ah,09

int 21h

int 60h

mov dx,OFFSET secondString; вывод второй строки

mov ah,09

int 21h

cli

push ds

mov dx,keepip

mov ax,keepcs

mov ds,ax

mov ah,25h

mov al,60h

int 21h

pop ds

sti

mov ah,4ch

int 21h

ret

Main ENDP

delayInt PROC FAR

push ax

push ds

mov di,32

mov ah,0

int 1Ah

mov bx,dx

Delay:

mov ah,0

int 1Ah

sub dx,bx

cmp di,dx

ja Delay

pop dx

pop ax

mov al,20h

out 20h,al

iret

delayInt ENDP

CODE ENDS

END Main

**Приложение Б. Файл листинга**

0000 AStack SEGMENT STACK

0000 05DC[ DW 1500 DUP(?)

????

]

0BB8 AStack ENDS

0000 DATA SEGMENT

0000 0000 keepcs DW 0; для хранения сегмента

0002 0000 keepip DW 0; и смещения прерывания

0004 firstString LABEL BYTE

0004 8F A5 E0 A2 A0 EF DB 'Первая строка выводится без задержк

и',13,10,'$'

20 E1 E2 E0 AE AA

A0 20 A2 EB A2 AE

A4 A8 E2 E1 EF 20

A1 A5 A7 20 A7 A0

A4 A5 E0 A6 AA A8

0D 0A 24

002B secondString LABEL BYTE

002B 80 20 A2 E2 AE E0 DB 'А вторая через 2 секунды',13,10,'$'

A0 EF 20 E7 A5 E0

A5 A7 20 32 20 E1

A5 AA E3 AD A4 EB

0D 0A 24

0046 DATA ENDS

0000 CODE SEGMENT

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

0000 Main PROC FAR

0000 B4 35 mov ah,35h

0002 B0 60 mov al,60h

0004 CD 21 int 21h

0006 89 1E 0002 R mov keepip,bx

000A 8C 06 0000 R mov keepcs,es

000E 1E push ds

000F BA 004B R mov dx, OFFSET delayInt; смещение для п

роцедуры в DX

0012 B8 ---- R mov ax, SEG delayInt; сегмент процедуры

0015 8E D8 mov ds,ax

0017 B4 25 mov ah,25h; функция установки вектора

0019 B0 60 mov al,60h; номер вектора

001B CD 21 int 21h

001D 1F pop ds

001E B8 ---- R mov ax,data

0021 8E D8 mov ds,ax

0023 BA 0004 R mov dx,OFFSET firstString; вывод первой

строки

0026 B4 09 mov ah,09

0028 CD 21 int 21h

002A CD 60 int 60h

002C BA 002B R mov dx,OFFSET secondString; вывод второ

й строки

002F B4 09 mov ah,09

0031 CD 21 int 21h

0033 FA cli

0034 1E push ds

0035 8B 16 0002 R mov dx,keepip

0039 A1 0000 R mov ax,keepcs

003C 8E D8 mov ds,ax

003E B4 25 mov ah,25h

0040 B0 60 mov al,60h

0042 CD 21 int 21h

0044 1F pop ds

0045 FB sti

0046 B4 4C mov ah,4ch

0048 CD 21 int 21h

004A CB ret

004B Main ENDP

004B delayInt PROC FAR

004B 50 push ax

004C 1E push ds

004D BF 0020 mov di,32

0050 B4 00 mov ah,0

0052 CD 1A int 1Ah

0054 8B DA mov bx,dx

0056 Delay:

0056 B4 00 mov ah,0

0058 CD 1A int 1Ah

005A 2B D3 sub dx,bx

005C 3B FA cmp di,dx

005E 77 F6 ja Delay

0060 5A pop dx

0061 58 pop ax

0062 B0 20 mov al,20h

0064 E6 20 out 20h,al

0066 CF iret

0067 delayInt ENDP

0067 CODE ENDS

END Main

Segments and Groups:

N a m e Length Align Combine Class

ASTACK . . . . . . . . . . . . . 0BB8 PARA STACK

CODE . . . . . . . . . . . . . . 0067 PARA NONE

DATA . . . . . . . . . . . . . . 0046 PARA NONE

Symbols:

N a m e Type Value Attr

DELAY . . . . . . . . . . . . . L NEAR 0056 CODE

DELAYINT . . . . . . . . . . . . F PROC 004B CODE Length = 001C

FIRSTSTRING . . . . . . . . . . L BYTE 0004 DATA

KEEPCS . . . . . . . . . . . . . L WORD 0000 DATA

KEEPIP . . . . . . . . . . . . . L WORD 0002 DATA

MAIN . . . . . . . . . . . . . . F PROC 0000 CODE Length = 004B

SECONDSTRING . . . . . . . . . . L BYTE 002B DATA

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

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

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

83 Source Lines

83 Total Lines

15 Symbols

48016 + 461291 Bytes symbol space free

0 Warning Errors

0 Severe Errors

**Приложение В. Карта памяти**

Start Stop Length Name Class

00000H 00BB7H 00BB8H ASTACK

00BC0H 00C05H 00046H DATA

00C10H 00C76H 00067H CODE

Program entry point at 00C1:0000