|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №12**

**«Программирование графики»**

**ДИСЦИПЛИНА: «Машинно-зависимые языки программирования»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-32Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Амеличева К.А. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2021

**Цель:** практическое овладение навыками разработки программного кода на языке Ассемблер. Изучение основных принципов представление данных на мониторе компьютера в графическом виде.

**Задачи:** разработка программы, с использованием графических примитивов.

**Вариант 2**

**Листинг:**

.486

model use16 small

.stack 100h

.data

VIDEOR DB ?

old\_data dd 0

exit\_code db 0

workCode db 0

currentX dd -200

x dd 0.0

y dw 0

two dd 2.0

three dd 3.0

threeFifths dd 0.6

hundred dd 100.0

fsttmp dw 0

.code

mFirstFunc macro

finit

fld x

fabs

fld1

fadd ST(1), ST(0)

fld x

fld x

fmul ST(0), ST(1)

fadd ST(0), ST(1)

fadd ST(0), ST(2)

fld three

fdivr ST(0), ST(3)

fxch ST(1)

fyl2x

fmul ST(2), ST(0)

fnstcw fsttmp

or [fsttmp], 0C00h

fldcw fsttmp

frndint

fsub ST(2), ST(0)

fxch ST(2)

f2xm1

fxch ST(2)

fld1

fadd ST(0), ST(3)

fscale

fdivr ST(0), ST(4)

fld hundred

fmul ST(0), ST(1)

fnstcw fsttmp

and [fsttmp], 0F3FFh

fldcw fsttmp

frndint

fist y

endm

mSecondFunc macro

finit

fld x

fld x

fmul ST(0), ST(1)

fld1

fadd ST(1), ST(0)

fxch ST(1)

fldln2

fxch ST(1)

fyl2x

fxch ST(2)

fcos

fmul ST(1), ST(0)

fmul ST(1), ST(0)

fmul ST(1), ST(0)

fmul ST(1), ST(0)

fld1

fadd ST(0), ST(2)

fld x

fld two

fmul ST(5), ST(0)

fadd ST(0), ST(1)

fdivr ST(0), ST(2)

fadd ST(0), ST(5)

fld hundred

fmul ST(0), ST(1)

fnstcw fsttmp

and [fsttmp], 0F3FFh

fldcw fsttmp

frndint

fist y

endm

mThirdFunc macro

fld1

fld x

fadd ST(0), ST(1)

fld threeFifths

fxch ST(1)

fyl2x

fld1

fmul ST(0), ST(1)

fnstcw fsttmp

or [fsttmp], 0C00h

fldcw fsttmp

frndint

fsub ST(1), ST(0)

fxch ST(1)

f2xm1

fxch ST(1)

fld1

fadd ST(0), ST(2)

fscale

fld hundred

fmul ST(0), ST(1)

fnstcw fsttmp

and [fsttmp], 0F3FFh

fldcw fsttmp

frndint

fist y

endm

mMovX macro num

finit

fld hundred

fidivr num

fst x

endm

mDrawMinus macro x, y

mov cx, x

mov dx, y

int 10h

dec cx

int 10h

dec cx

int 10h

endm

mDrawZero macro x, y

mov cx, x

mov dx, y

int 10h

inc cx

int 10h

inc cx

int 10h

inc dx

int 10h

inc dx

int 10h

inc dx

int 10h

inc dx

int 10h

dec cx

int 10h

dec cx

int 10h

dec dx

int 10h

dec dx

int 10h

dec dx

int 10h

endm

mDrawOne macro x, y

mov cx, x

mov dx, y

int 10h

inc dx

int 10h

inc dx

int 10h

inc dx

int 10h

inc dx

int 10h

endm

mDrawTwo macro x, y

mov cx, x

mov dx, y

int 10h

inc cx

int 10h

inc cx

int 10h

inc dx

int 10h

inc dx

int 10h

dec cx

int 10h

dec cx

int 10h

inc dx

int 10h

inc dx

int 10h

inc cx

int 10h

inc cx

int 10h

endm

mDrawFive macro x, y

mov cx, x

mov dx, y

int 10h

dec cx

int 10h

dec cx

int 10h

inc dx

int 10h

inc dx

int 10h

inc cx

int 10h

inc cx

int 10h

inc dx

int 10h

inc dx

int 10h

dec cx

int 10h

dec cx

int 10h

endm

Start:

jmp PROG\_START

HANDLE:

cmp workCode, 1

jne HANDLE\_END

cmp currentX, -99

jl FIRST\_FUNC

cmp currentX, 0

jl SECOND\_FUNC

cmp currentX, 101

jl THIRD\_FUNC

jmp END\_PROG

FIRST\_FUNC:

mMovX currentX

mFirstFunc

mov dx, 260

sub dx, y

mov cx, word ptr currentX

add cx, 210

mov ax, 0C01h

int 10h

inc currentX

jmp HANDLE\_END

SECOND\_FUNC:

mMovX currentX

mSecondFunc

mov dx, 260

sub dx, y

mov cx, word ptr currentX

add cx, 210

mov ax, 0C04h

int 10h

inc currentX

jmp HANDLE\_END

THIRD\_FUNC:

mMovX currentX

mThirdFunc

mov dx, 260

sub dx, y

mov cx, word ptr currentX

add cx, 210

mov ax, 0C02h

int 10h

inc currentX

jmp HANDLE\_END

END\_PROG:

mov exit\_code, 1

HANDLE\_END:

jmp cs:old\_data

iret

PROG\_START:

mov ax, @data

mov ds, ax

MOV AH, 0Fh

INT 10h

MOV VIDEOR, AL

MOV AX, 0013h

INT 10h

PUSH 0A000h

POP ES

mov cx, 64000

FILL:

dec cx

mov di, cx

MOV byte ptr ES:[DI], 15

cmp cx, 0

jne FILL

mov ah, 0Ch

xor al, al

xor bx, bx

mov cx, 10

mov dx, 190

OX:

int 10h

inc cx

cmp cx, 311

jne OX

mov cx, 210

mov dx, 10

OY:

int 10h

inc dx

cmp dx, 189

jne OY

mov cx, 10

mov dx, 189

OX\_STREAKS:

int 10h

add dx, 2

int 10h

mov dx, 189

add cx, 100

cmp cx, 410

jne OX\_STREAKS

mov cx, 209

mov dx, 10

int 10h

add cx, 2

int 10h

mov cx, 209

mov dx, 60

int 10h

add cx, 2

int 10h

mov cx, 209

mov dx, 160

int 10h

add cx, 2

int 10h

mDrawMinus 9, 195

mDrawTwo 11, 193

mDrawMinus 110, 195

mDrawOne 112, 193

mDrawZero 209, 193

mDrawOne 310, 193

mDrawOne 201, 158

mDrawTwo 205, 58

mDrawTwo 199, 8

mDrawFive 207, 8

mov cx, 203

mov dx, 12

int 10h

xor ax, ax

mov es, ax

mov ah, 35h

mov al, 1Ch

int 21h

mov word ptr old\_data, bx

mov word ptr old\_data + 2, es

xor ax, ax

mov es, ax

push ds

push dx

mov dx, offset HANDLE

mov ax, seg HANDLE

mov ds, ax

mov ah, 25h

mov al, 1Ch

int 21h

pop dx

pop ds

mov workCode, 1

CYCLE:

mov cl, exit\_code

cmp cl, 1

jne CYCLE

push ds

mov dx, word ptr old\_data

mov ax, word ptr old\_data + 2

mov ds, ax

mov ah, 25h

mov al, 1Ch

int 21h

pop ds

XOR AX, AX

INT 16h

MOV AH, 0

MOV AL, VIDEOR

INT 10h

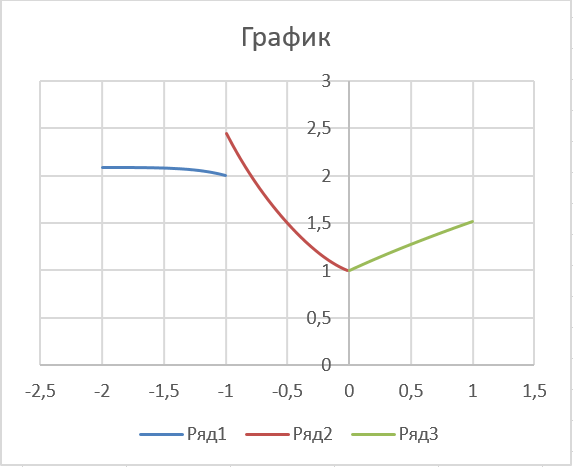
mov ax, 4c00h

int 21h

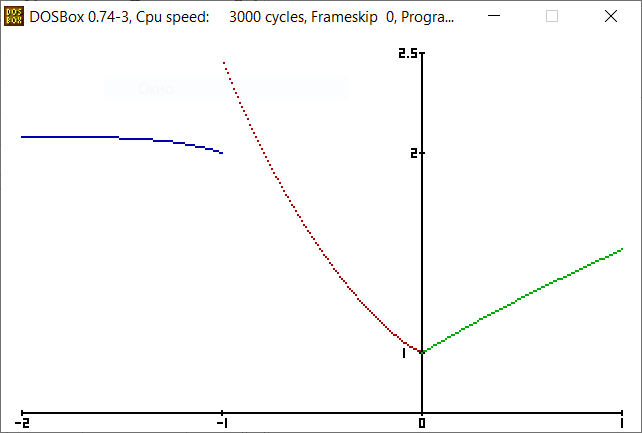
end Start

end

**Результаты работы:**



**Рисунок 1.** Проверка графика



**Рисунок 2.** Результат

**Вывод:** в ходе выполнения лабораторной работы были получены навыки работы с таймером и графикой в Assembler, создания анимации.