**Федеральное государственное образовательное бюджетное учреждение  
высшего образования**

**«Финансовый университет при Правительстве Российской Федерации»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

|  |  |
| --- | --- |
|  | |
|  |  |

**ОТЧЕТ**

**по практическому занятию**

**Машинно-ориентированное программирование для решения задач защиты информации**

**Студент:** Матвейчук Анастасия Сергеевна

**Специальность: 10.02.05 Обеспечение информационной безопасности автоматизированных систем**

**Группа: *3ОИБАС-718***

Оценка:\_\_\_\_\_\_\_\_\_\_\_\_

**Москва**

**2020 г.**

**Содержание**

[Введение 3](#_Toc62042075)

[Практическая часть 4](#_Toc62042076)

[Практическая часть 2 11](#_Toc62042077)

# Введение

**Цель работы:** вывести на экран свою ФИО и «Привет, Мир!», написав программу. Сделать её резидентной, также написать программу, которая её вызывает.

# Практическая часть

1. Устанавливаем FASM и DosBox

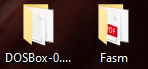


Рис. 1 – Установка FASM и DosBox

Дальше запишем команды для запуска программы Hello World! через FASM и COM файла программы через DOSBox, в блокнот и сохраняем как пакетный файл(.bat):

Код:

@echo off

echo %cd%

cd C:\Users\13\Desktop\Fasm

echo %cd%

C:\Users\13\Desktop\Fasm\Fasm\FASM.exe

"C:\Program Files (x86)\DOSBox-0.74-3\DOSBox.exe" \_000\_.COM

Pause

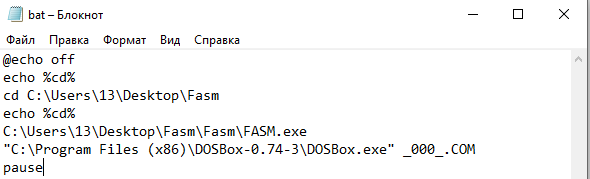


Рис. 1.1 – Код, сохраненный в .bat

1. Нужно написать программу, которая будет выводить «Hello, World!»

Код программы:

use16

org 100h

jmp start

hello db 'Hello, world!$'

start:

mov dx,hello

mov ah,9

int 21h

mov dl,0ah

mov ah,2

int 21h

mov ah,01h

int 21h

mov ax,4C00h

int 21h

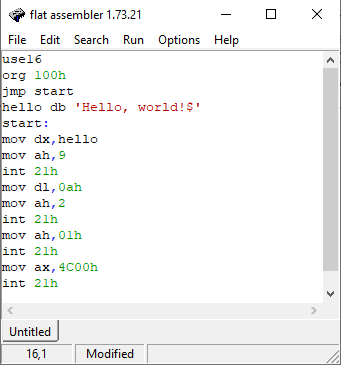


Рис. 2 – Код программы

1. Создаем код программы, который будет выводить ФИО и номер группы

Код программы:

; fasm code...

; Резидент. Перехватывает F1

;--------------------------—

org 100h

push setup

ret ;настройки..

new\_9: db 60h

mov ah,2

mov dl,0ah

int 21h

mov dl,77

int 21h

mov dl,97

int 21h

mov dl,116

int 21h

mov dl,118

int 21h

mov dl,101

int 21h

mov dl,121

int 21h

mov dl,99

int 21h

mov dl,104

int 21h

mov dl,117

int 21h

mov dl,107

int 21h

mov dl,32

int 21h

mov dl,65

int 21h

mov dl,110

int 21h

mov dl,97

int 21h

mov dl,115

int 21h

mov dl,116

int 21h

mov dl,97

int 21h

mov dl,115

int 21h

mov dl,105

int 21h

mov dl,121

int 21h

mov dl,97

int 21h

mov dl,32

int 21h

mov dl,83

int 21h

mov dl,101

int 21h

mov dl,114

int 21h

mov dl,103

int 21h

mov dl,101

int 21h

mov dl,101

int 21h

mov dl,118

int 21h

mov dl,110

int 21h

mov dl,97

int 21h

mov dl,32

int 21h

mov dl,51

int 21h

mov dl,79

int 21h

mov dl,73

int 21h

mov dl,66

int 21h

mov dl,65

int 21h

mov dl,83

int 21h

mov dl,45

int 21h

mov dl,55

int 21h

mov dl,49

int 21h

mov dl,56

;cut —--------------8<---------------------—

;....

;cut —--------------8<---------------------—

old\_9: dw 0,0

msg db 'Damned Help!!!'

len = $-msg

;======= Перехват вектора(9) ==========================

setup:

mov ax,3509h ;запомним оригинал

int 21h ;

mov [old\_9],bx ;

mov [old\_9+2],es ;

mov ax,2509h ;ставим свой обработчик

mov dx,new\_9 ;

int 21h ;

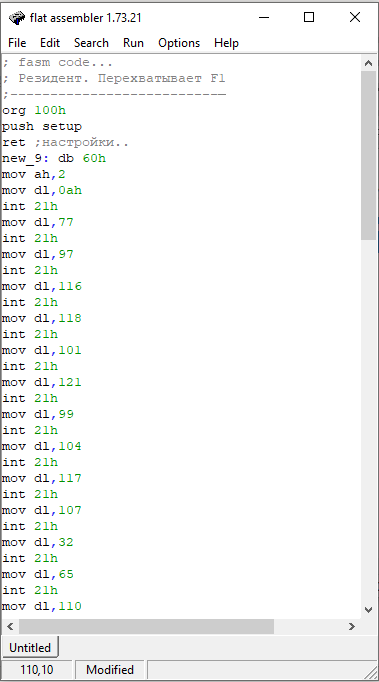


Рис. 3 – Код программы

1. Запускаем DosBox и выводим «Hello, World»:

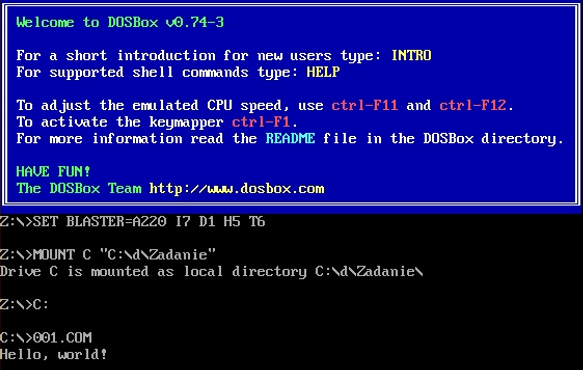


Рис. 4 – Вывод «Hello, World»

1. Также запускаем следующий код и выводим ФИО и номер группы



Рис. 5 – Вывод ФИО и номера группы

# Практическая часть 2

**Задание 1.**

Реализовать на языке ассемблера вычисления выражения, записанного на языке Паскаль: begin if (x >=5) and (x <=7) then x: = 1 else x: = 2; end.

Реализуем программу на языке С++ с ассемблерной вставкой, которая проверяет след. условия: if (x >=5) and (x <=7) then x: = 1 else x: = 2.

Решение:

Создаем файл Matveychuk.ASM и Matveychuk.Prog, затем открываем их через NotePad++.

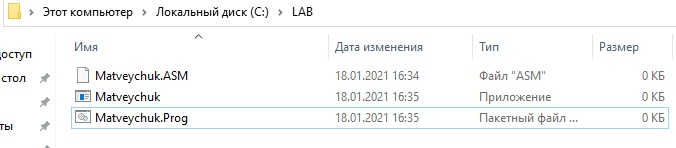


Рис. 6

В файле Matveychuk.ASM в первой строчке указываем путь к Fasm.exe и путь к нашему файлу Matveychuk.ASM

Во второй строчке указываем путь, где необходимо создать EXE файл для запуска программы.

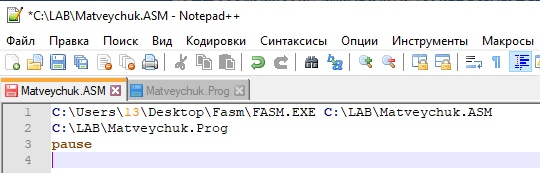


Рис. 7

В папку с FASM закидываем скачанный файл OPENGL.INC и прописываем пути.

В первой строчке пишем путь к WIN64A.INC. Во второй строчке пишем путь до OPENGL.INC.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 8

Переходим к написанию кода, для определения значения переменной, больше или меньше 5 и 7.

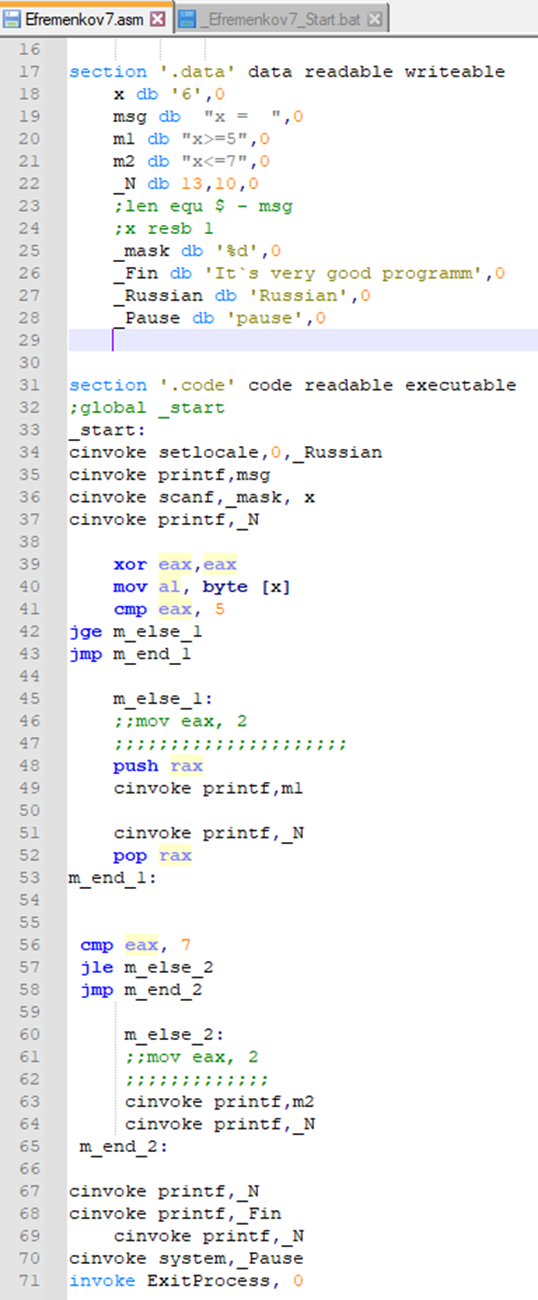


Рис. 9

Дальше запускаем её и получаем результат:

Изображение выглядит как текст

Автоматически созданное описание

Рис. 10

КОД ПРОГРАММЫ:

format PE64 console

entry \_start

include 'C:\FASM\INCLUDE\WIN64A.INC'

include 'C:\FASM\INCLUDE\OPENGL.INC'

section '.idata' import data readable

library kernel,'kernel32.dll',\

msvcrt,'msvcrt.dll'

import kernel,\

ExitProcess,'ExitProcess'

import msvcrt,\

setlocale,'setlocale',\

printf,'printf',\

scanf,'scanf',\

system,'system'

section '.data' data readable writeable

x db '6',0

msg db "x = ",0

m1 db "x>=5",0

m2 db "x<=7",0

\_N db 13,10,0

;len equ $ - msg

;x resb 1

\_mask db '%d',0

\_Fin db The program is working,0

\_Russian db 'Russian',0

\_Pause db 'pause',0

section '.code' code readable executable

;global \_start

\_start:

cinvoke setlocale,0,\_Russian

cinvoke printf,msg

cinvoke scanf,\_mask, x

cinvoke printf,\_N

xor eax,eax

mov al, byte [x]

cmp eax, 5

jge m\_else\_1

jmp m\_end\_1

m\_else\_1:

;;mov eax, 2

;;;;;;;;;;;;;;;;;;;;;

push rax

cinvoke printf,m1

cinvoke printf,\_N

pop rax

m\_end\_1:

cmp eax, 7

jle m\_else\_2

jmp m\_end\_2

m\_else\_2:

;;mov eax, 2

;;;;;;;;;;;;;

cinvoke printf,m2

cinvoke printf,\_N

m\_end\_2:

cinvoke printf,\_N

cinvoke printf,\_Fin

cinvoke printf,\_N

cinvoke system,\_Pause

invoke ExitProcess, 0

**Задание 2:**

Реализовать на языке ассемблера вычисления выражения, записанного на языке Паскаль: begin if (x <=5) and (x >=7) then x: = 1 else x: = 2; end.

Реализуем программу на языке С++ с ассемблерной вставкой, которая проверяет след. условия: if (x <=5) and (x >=7) then x: = 1 else x: = 2.

Решение:

Создаем файл Matveychuk.ASM и Matveychuk.Prog, затем открываем их через NotePad++.

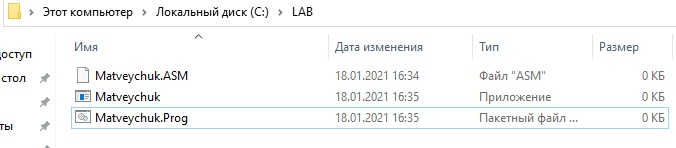


Рис. 11

В файле Matveychuk.ASM в первой строчке указываем путь к Fasm.exe и путь к нашему файлу Matveychuk.ASM

Во второй строчке указываем путь, где необходимо создать EXE файл для запуска программы.

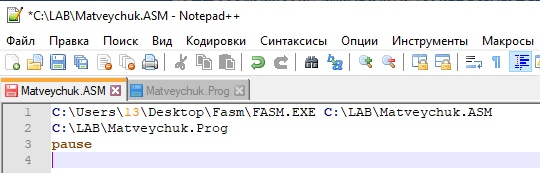


Рис. 12

В папку с FASM закидываем скачанный файл OPENGL.INC и прописываем пути.

В первой строчке пишем путь к WIN64A.INC. Во второй строчке пишем путь до OPENGL.INC.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 13

Переходим к написанию кода, для определения значения переменной, больше или меньше 5 и 7.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 14

Дальше запускаем её и получаем результат:

Изображение выглядит как текст

Автоматически созданное описание

Рис. 10

КОД ПРОГРАММЫ:

format PE64 console

entry \_start

include 'C:\FASM\INCLUDE\WIN64A.INC'

include 'C:\FASM\INCLUDE\OPENGL.INC'

section '.idata' import data readable

library kernel,'kernel32.dll',\

msvcrt,'msvcrt.dll'

import kernel,\

ExitProcess,'ExitProcess'

import msvcrt,\

setlocale,'setlocale',\

printf,'printf',\

scanf,'scanf',\

system,'system'

section '.data' data readable writeable

x db '6',0

msg db "x = ",0

m1 db "x<=5",0

m2 db "x>=7",0

\_N db 13,10,0

;len equ $ - msg

;x resb 1

\_mask db '%d',0

\_Fin db The program is not working,0

\_Russian db 'Russian',0

\_Pause db 'pause',0

section '.code' code readable executable

;global \_start

\_start:

cinvoke setlocale,0,\_Russian

cinvoke printf,msg

cinvoke scanf,\_mask, x

cinvoke printf,\_N

xor eax,eax

mov al, byte [x]

cmp eax, 5

jge m\_else\_1

jmp m\_end\_1

m\_else\_1:

;;mov eax, 2

;;;;;;;;;;;;;;;;;;;;;

push rax

cinvoke printf,m1

cinvoke printf,\_N

pop rax

m\_end\_1:

cmp eax, 7

jle m\_else\_2

jmp m\_end\_2

m\_else\_2:

;;mov eax, 2

;;;;;;;;;;;;;

cinvoke printf,m2

cinvoke printf,\_N

m\_end\_2:

cinvoke printf,\_N

cinvoke printf,\_Fin

cinvoke printf,\_N

cinvoke system,\_Pause

invoke ExitProcess, 0