1)

format PE console 4.0 ; Консольное приложение.

entry start ; Точка входа

include 'C:\Users\79169\Desktop\FASM\INCLUDE\win32a.inc' ; Делаем стандартное включение описателей.

;--- секция кода ---

section '.text' code readable executable

start:

invoke GetStdHandle, STD\_OUTPUT\_HANDLE

mov [stdout], eax

invoke GetStdHandle, STD\_INPUT\_HANDLE

mov [stdin], eax

invoke WriteConsole,[stdout],cMsg,13,NULL,NULL

invoke ReadConsole,[stdin],lpBuffer,1,lpCharsRead,NULL

exit:

invoke ExitProcess, 0

;--- секция данных ---

section '.data' data readable writeable

cMsg db 'Hello, world!' ; Текстовая строка.

lpBuffer db 10 dup (0)

lpCharsRead dd ? ; Количество фактически считанных символов

stdin dd ?

stdout dd ?

;--- секция (таблица) импорта ---

section '.idata' import data readable writeable

library kernel32,'KERNEL32.DLL'

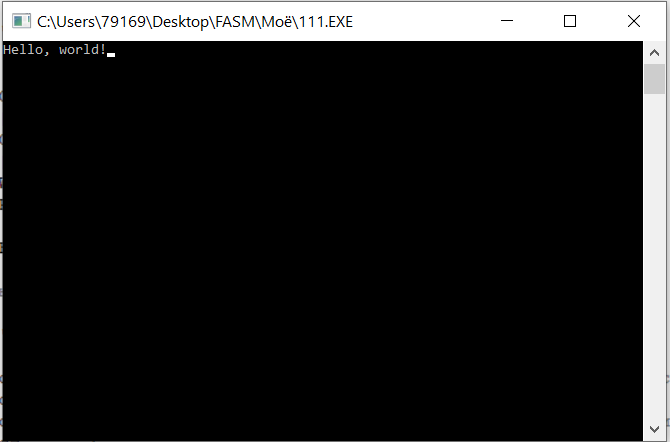
import kernel32,\

GetStdHandle,'GetStdHandle',\

WriteConsole,'WriteConsoleA',\

ReadConsole,'ReadConsoleA',\

ExitProcess,'ExitProcess'



2)

test eax,eax

jz .err

mov [fbuf],eax

; Âîçâðàùàåì óêàçàòåëü ÷òåíèÿ-çàïèñè â íà÷àëî ôàéëà

cinvoke fseek,[fp],0,SEEK\_SET

test eax,eax

jnz .err

; ×èòàåì âåñü ôàéë â äèíàìè÷åñêóþ ïàìÿòü

cinvoke fread,[fbuf],1,[flength],[fp]

cmp eax,[flength]

jne .err

; Çàïèñûâàåì 0 çà êîíöîì äàííûõ äëÿ îáîçíà÷åíèÿ êîíöà ñòðîêè

mov eax,[fbuf]

add eax,[flength]

mov byte [eax],0

; Âûâîäèì ôàéë íà êîíñîëü

cinvoke puts,[fbuf]

cinvoke printf,'Äëèíà ôàéëà = %u%s',[flength],<13,10,0>

; Çàêðûâàåì ôàéë

cinvoke fclose,[fp]

test eax,eax

jnz .err

; Îñâîáîæäàåì äèíàìè÷åñêóþ ïàìÿòü

cinvoke free,[fbuf]

.finish:

cinvoke puts,'Press any key...'

cinvoke \_getch

invoke ExitProcess,0

.err:

cinvoke puts,errmsg

jmp .finish

section '.idata' import data readable writeable

library kernel,'kernel32.dll',\

msvcrt,'msvcrt.dll',\

user32,'user32.dll'

import kernel,\

ExitProcess,'ExitProcess',\

GetCommandLineA,'GetCommandLineA',\

AllocConsole,'AllocConsole',\

FreeConsole,'FreeConsole',\

SetConsoleOutputCP,'SetConsoleOutputCP',\

SetConsoleCP,'SetConsoleCP'

import msvcrt,\

\_\_getmainargs,'\_\_getmainargs',\

fopen,'fopen',\

fseek,'fseek',\

ftell,'ftell',\

malloc,'malloc',\

free,'free',\

fread,'fread',\

fclose,'fclose',\

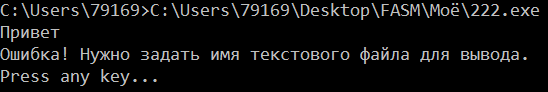
printf,'printf',\

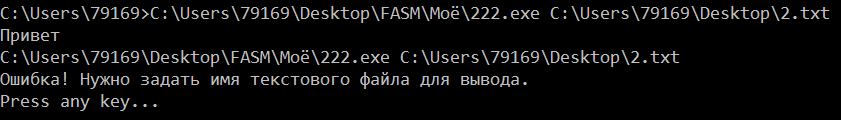
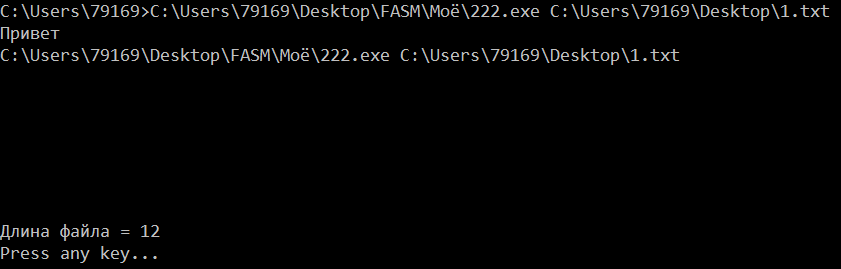
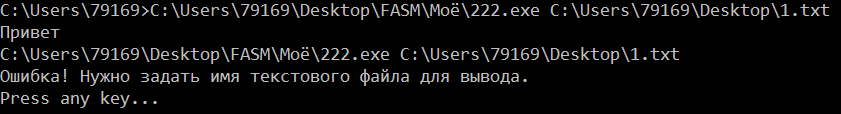
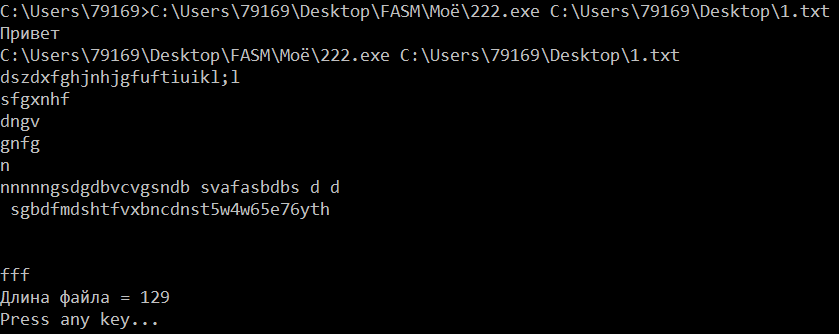
\_getch,'\_getch',\

puts,'puts'

import user32,\

CharToOemA,'CharToOemA'





3)

format PE console 4.0

include 'win32ax.inc'

entry start

;--- секция кода ---

section '.code' code readable executable

start:

invoke FreeConsole ; Освободить существующую консоль

invoke AllocConsole ; Назначить новую консоль

invoke GetStdHandle, STD\_OUTPUT\_HANDLE ; Получить дескриптор стандартного потока вывода

mov [stdout], eax ; Сохранить его

invoke GetStdHandle, STD\_INPUT\_HANDLE ; Получить дескриптор стандартного потока ввода

mov [stdin], eax ; Сохранить его

invoke GetConsoleScreenBufferInfo,[stdout],lpConsoleScreenBufferInfo

mov [ConsoleWindow.Left], 0

mov [ConsoleWindow.Right], 0

mov ax, word [lpConsoleScreenBufferInfo.srWindow.Right]

shr ax,2

mov word [ConsoleWindow.Right], ax

mov ax, word [lpConsoleScreenBufferInfo.srWindow.Bottom]

shr ax,2

mov word [ConsoleWindow.Bottom], ax

invoke SetConsoleWindowInfo,[stdout],TRUE,ConsoleWindow

invoke Sleep, 10000

invoke FreeConsole

exit:

invoke ExitProcess, 0

;--- секция данных ---

section '.data' data readable writeable

stdout dd ?

stdin dd ?

struct COORD

x dw ?

y dw ?

ends

struct SMALL\_RECT

Left dw ?

Top dw ?

Right dw ?

Bottom dw ?

ends

struct CONSOLE\_SCREEN\_BUFFER\_INFO

dwSize COORD ; размеры буфера в строках и столбцах символов

dwCursorPosition COORD ; координаты (позиция) курсора в буфере

wAttributes dw ? ; атрибуты символов

srWindow SMALL\_RECT ; координаты верхнего левого и нижнего правого углов буфера

dwMaximumWindowSize COORD ; максимальные размеры консольного окна

ends

lpConsoleScreenBufferInfo CONSOLE\_SCREEN\_BUFFER\_INFO

ConsoleWindow SMALL\_RECT

dwSize COORD

;--- секция импорта ---

section '.idata' import data readable writeable

library kernel32,'KERNEL32.DLL'

import kernel32,\

AllocConsole,'AllocConsole',\

GetConsoleScreenBufferInfo,'GetConsoleScreenBufferInfo',\

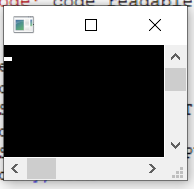
SetConsoleWindowInfo,'SetConsoleWindowInfo',\

FreeConsole,'FreeConsole',\

GetStdHandle,'GetStdHandle',\

Sleep,'Sleep',\

ExitProcess,'ExitProcess'



4)

format PE console 4.0 ; Консольное приложение.

entry start ; Точка входа

include 'win32a.inc' ; Делаем стандартное включение описателей.

include 'ENCODING\WIN1251.INC' ; Подключить кодировку WIN1251

;--- секция кода ---

section '.text' code readable executable

start:

invoke SetConsoleOutputCP,1251

invoke SetConsoleCP,1251

invoke GetStdHandle, STD\_OUTPUT\_HANDLE

mov [stdout], eax

invoke GetStdHandle, STD\_INPUT\_HANDLE

mov [stdin], eax

invoke WriteConsole,[stdout],cMsg,12,NULL,NULL

invoke ReadConsole,[stdin],lpBuffer,1,lpCharsRead,NULL

exit:

invoke ExitProcess, 0

;--- секция данных ---

section '.data' data readable writeable

cMsg db 'Привет, мир!' ; Текстовая строка.

lpBuffer db 10 dup (0)

lpCharsRead dd ? ; Количество фактически считанных символов

stdin dd ?

stdout dd ?

;--- секция (таблица) импорта ---

section '.idata' import data readable writeable

library kernel32,'KERNEL32.DLL'

import kernel32,\

GetStdHandle,'GetStdHandle',\

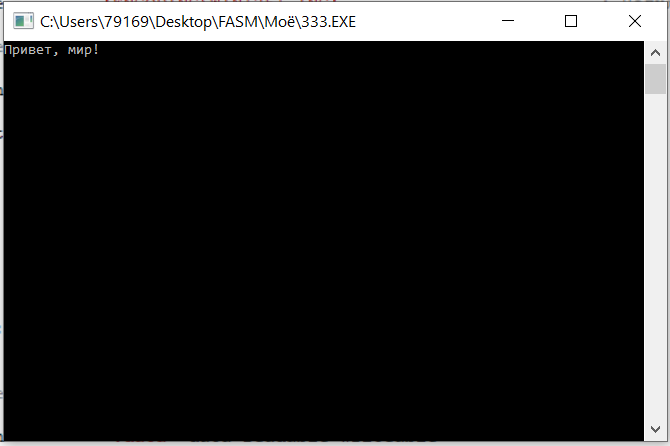
SetConsoleOutputCP,'SetConsoleOutputCP',\

SetConsoleCP,'SetConsoleCP',\

WriteConsole,'WriteConsoleA',\

ReadConsole,'ReadConsoleA',\

ExitProcess,'ExitProcess'



5)

format PE console 4.0

include 'win32a.inc'

start: cinvoke printf, req, 41h ; ????? ?? ?????

cinvoke scanf, tpt, A ; ???? A

cinvoke printf, req, 42h ; ????? ?? ?????

cinvoke scanf, tpt, B ; ???? B

mov eax,[A]

add eax,[B]

cinvoke printf, tpo, eax

invoke sleep, 5000 ; 5 sec. delay

; ?????

gtfo: invoke exit, 0

req db 'Enter %c:',0

tpo db 'A + B = '

tpt db '%d',0

A dd ?

B dd ?

; import data in the same section

data import

library msvcrt,'MSVCRT.DLL',\

kernel32,'KERNEL32.DLL'

import kernel32,\

sleep,'Sleep'

import msvcrt,\

puts,'puts',\

scanf,'scanf',\

printf,'printf',\

exit,'exit'

end data

****