Учреждение образования Республики Беларусь

«Гомельский государственный технический университет им. П. О. Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Отчет по лабораторной работе №3

по дисциплине «Архитектура ЭВМ»

по теме «Вывод строковых массивов на языке ассемблера»

Выполнил:

студент группы ИТП-21

Половцев М. С.

Проверил:

преподаватель

Савельев В. А.

Гомель, 2022

**Цель:** знакомство с Макро Ассемблером MASM32, создание минимального Windows–приложения на ассемблере, изучение вызова функций API win32 из ассемблерных приложений.

**Ход работы**

Задание:

1) Набрать текст программы из лабораторной работы 3, произвести ассемблирование и линовку, проверить работоспособность приложения.

2) Переписать приведенный выше текст программы с использованием макрокоманды **invoke**.

К основным командам языка Ассемблера относятся push (помещение в стек), call (вызов функции), mov (перемещение значения).

Листинг:

;--------------------------------------------------

; Программа на masm32 для консоли (вывод сообщения)

;--------------------------------------------------

.386

.model flat, stdcall

option casemap:none

include C:\masm32\include\windows.inc

include C:\masm32\include\kernel32.inc

includelib C:\masm32\lib\kernel32.lib

; Сегмент данных

;--------------------------------------------------

.data

string db "Hello World!", 0Ah, 0h

sConsoleTitle db "My first project", 0

.data?

stdout dd ?

cWritten dd ?

; Сегмент кода

;--------------------------------------------------

.code

start:

; заголовок консоли

push offset sConsoleTitle

call SetConsoleTitle

; получаем дескриптор

push STD\_OUTPUT\_HANDLE

call GetStdHandle

mov stdout, eax

mov cWritten, ebx

; выводим в консоль строку

push NULL

push offset cWritten

push sizeof string

push offset string

push stdout

call WriteConsole

; задержка

push INFINITE

call Sleep

; завершаем процесс

push NULL

call ExitProcess

end start

На рисунке 1 представлен результат работы программы №1:

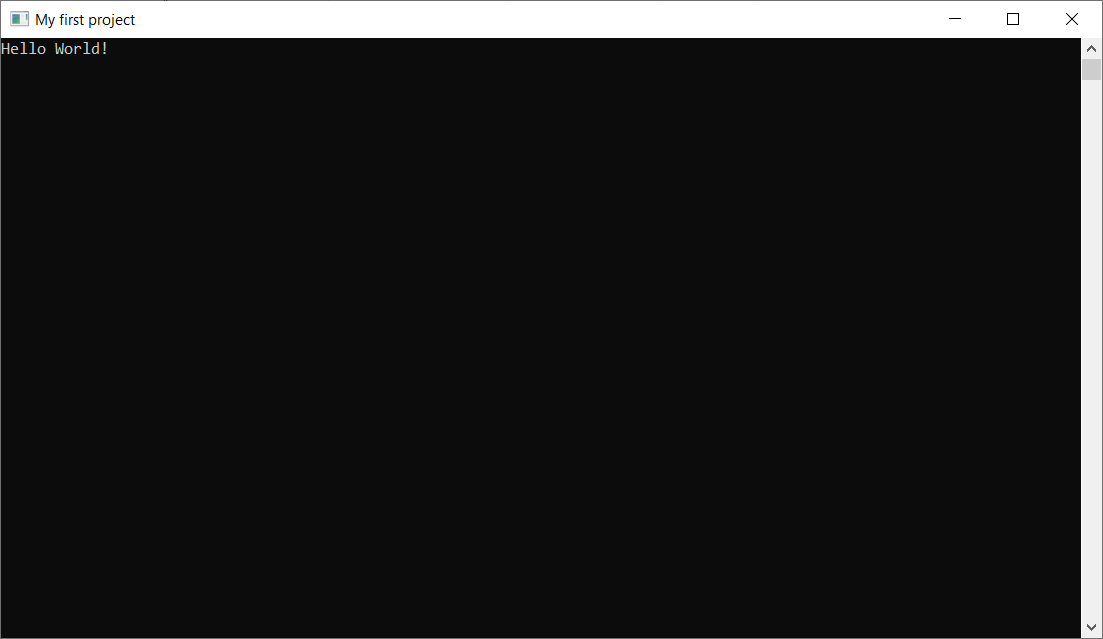


Рисунок 1 – Результат работы программы №1

Для реализации второго задания будем использовать команду invoke, которая позволяет записывать вызовы в более удобном виде.

Листинг программы:

;--------------------------------------------------

; Программа на masm32 для консоли (вывод сообщения)

;--------------------------------------------------

.386

.model flat, stdcall

option casemap:none

include C:\masm32\include\windows.inc

include C:\masm32\include\kernel32.inc

includelib C:\masm32\lib\kernel32.lib

; Сегмент данных

;--------------------------------------------------

.data

string db "Hello World!", 0Ah, 0h

sConsoleTitle db "My first project", 0

.data?

stdout dd ?

cWritten dd ?

; Сегмент кода

;--------------------------------------------------

.code

start:

; заголовок консоли

invoke SetConsoleTitle, addr sConsoleTitle

; получаем дескриптор

invoke GetStdHandle, STD\_OUTPUT\_HANDLE

mov stdout, eax

mov cWritten, ebx

; выводим в консоль строку

invoke WriteConsole, stdout, addr string, sizeof string, addr cWritten, NULL

call WriteConsole

; задержка

invoke Sleep, INFINITE

; завершаем процесс

invoke ExitProcess, NULL

end start

Результат работы программы №2 представлен на рисунке 2:

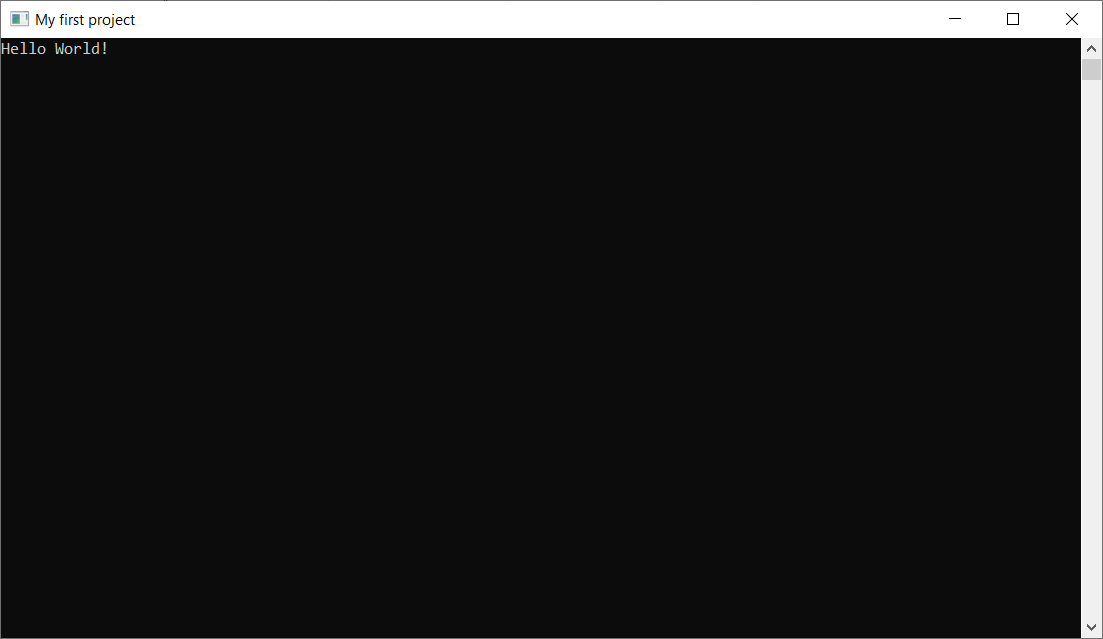


Рисунок 2 – Результат работы программы №2

Для запуска программ были созданы 2 bat-файла следующего содержания:

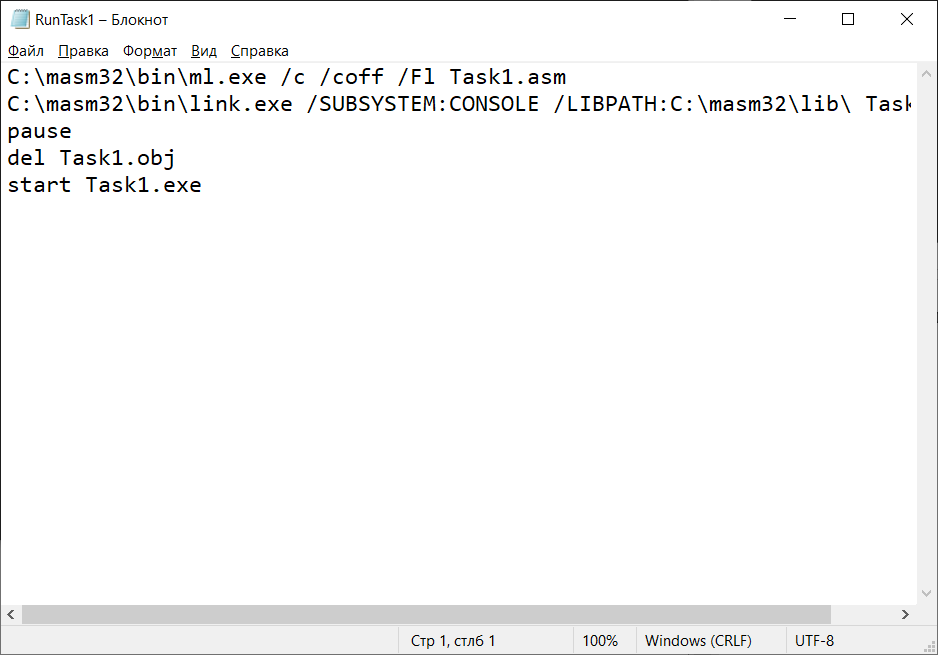


Рисунок 3 – Содержимое bat-файла для запуска программы

Вывод: для работы на языке Ассемблера часто используются следующие команды push (помещение в стек), call (вызов функции), mov (перемещение значения) и функции GetStdHandle, WriteConsole, ExitProcess, Sleep. Оператор offset возвращает адрес переменной в точку вызова. Также в программе используются метки, устанавливающие границы кода.