**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

по дисциплине «Архитектура ЭВМ»

на тему: «Выводы строковых массивов на языке Ассемблер»

Выполнил: студент гр. ИТП-21

Ковтун А.В.

Принял: преподаватель

Савельев В. А.

Гомель 2022

**Цель:** знакомство с Макро Ассемблером *MASM*32. Создание минимального *Windows*-приложения на ассемблере. Изучение вызова функции *API win32* из ассемблерных приложений.

**Задание 1.** Набрать текст приведеной выше программы, произвести ассемблирование и линковку. проверить работоспособность приложения.

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

**Текст программы с использованием push:**

; Определяет набор инструкций, которые могут использоваться в программме

.386

; .model flat (Директова ассемблера, определяющая сегментную модель памяти приложения как плоскость)

; stdcall - говорит MASM32 о порядке передачи параметров (справа налево)

.model flat, stdcall

; Говорит MASM сделать метки чуствительными к регистру (ExitProcess != exitprocess)

option casemap:none

; Подключение Библиотек

; Добавляет содержимое файла в данную директиву

include C:\masm32\include\windows.inc

include C:\masm32\include\kernel32.inc

; Сообщает ассемблеру какие библиотеки использует программа

includelib C:\masm32\lib\kernel32.lib

; .data - Секция инициализированных данных

.data

string db "Hello world", 0Ah, 0h

sConsoleTitle db "My first project", 0

; .data? - Секция неинициализированных данных

.data?

stdout dd ?

cWritten dd ?

; Директива определяющая секцию программы

.code

start:

; Определяет загаловок консоли, который указан в переменной sConsoleTitle

push offset sConsoleTitle

call SetConsoleTitle

; Получам дискриптор

push STD\_OUTPUT\_HANDLE

call GetStdHandle

mov stdout, eax

mov cWritten, ebx

; вывод строки в консоль

push NULL

push offset cWritten ; Заносит в стек указатель на переменную

push sizeof string ; Получаем число символов в строке string

push offset string ; Получаем адрес переменной string

push stdout ; Заносит дискриптер в стек

call WriteConsole ; Записывает строку на экранный буфер консоли

; Вызывается бесконечная задержка

push INFINITE

call Sleep

; Завершение процесса

push NULL ; Задание параметров функции ExitProcess через стек

call ExitProcess ; вызов API-функции завершения процесса

end start

**Текст программы с использованием invoke:**

.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

.data

string db "Hello world", 0Ah, 0h

sConsoleTitle db "Kovtun Andrei lab2", 0

.data?

stdout dd ?

cWritten dd ?

.code

start:

invoke SetConsoleTitle, offset sConsoleTitle

invoke GetStdHandle, STD\_OUTPUT\_HANDLE

mov stdout, eax

mov cWritten, ebx

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

invoke Sleep, INFINITE

invoke ExitProcess, NULL

end start

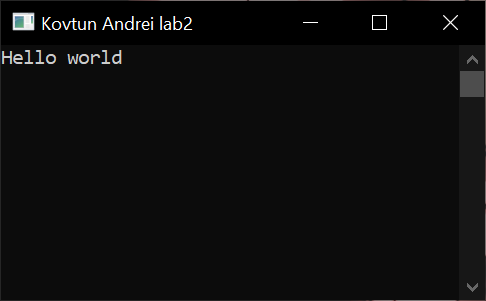


Рисунок 1 – Результат выполнения программы

**Вывод**: в ходе выполнения лабораторной работы были изучены основные языка программирования ассемблер. Основные команды этого языка и способы ассемблирование файлов.