

## Лабораторная работа asm1 (2 часа)

### Конструирование программного обеспечения

#### Создание программы на языке ассемблера

**Цель работы:** Создание MASM-проекта в Visual Studio, изучение этапов обработки исходного кода, исследование свойств проекта в интегрированной среде разработки Visual Studio.

#### Задание.

1. Используйте материал лекции asm01.
2. Создайте проект-приложение с именем **SE\_Asm01** на языке ассемблера.
3. Разработайте в Visual Studio программу **asm01**, убедитесь в ее работоспособности.

```
.586P                                ; система команд(процессор Pentium)
.MODEL FLAT, STDCALL                 ; модель памяти, соглашение о вызовах
includelib kernel32.lib              ; компоновщику: компоновать с kernel32

ExitProcess PROTO : DWORD             ; прототип функции для завершения процесса Windows
MessageBoxA PROTO : DWORD, : DWORD, : DWORD, : DWORD ; прототип API-функции MessageBoxA

.STACK 4096                           ; выделение стека

.CONST                               ; сегмент констант

.DATA                                ; сегмент данных
MB_OK EQU 0                          ; EQU определяет константу
STR1 DB "Моя первая программа", 0    ; строка, первый элемент данные + нулевой бит
STR2 DB "Привет всем!", 0           ; строка, первый элемент данные + нулевой бит
HW DD ?                             ; двойное слово длиной 4 байта, неинициализировано

.CODE                                ; сегмент кода

main PROC                            ; точка входа main
START :                              ; метка

    PUSH MB_OK
    PUSH OFFSET STR1
    PUSH OFFSET STR2
    PUSH HW
    CALL MessageBoxA                 ; вызов функции

    push - 1                         ; код возврата процесса Windows(параметр ExitProcess)
    call ExitProcess                 ; так завершается любой процесс Windows
main ENDP                            ; конец процедуры

end main                             ; конец модуля main
```

4. Исследуйте свойства проекта **SE\_Asm01**, ответы сохранить в виде отчета в WORD -документе.
5. Какое расширение имеет файл исходного кода на языке ассемблера?
6. Какие типы файлов создаются ассемблером?
7. Найдите в папке проекта созданный объектный модуль. В какой папке он находится?

8. Найдите в папке проекта исполняемый модуль.
9. Сгенерируйте файл листинга программы. Какое расширение имеет файл листинга?
10. Укажите, для какого типа процессора создается программа, какая модель памяти используется в операционной системе Windows.
11. Какое соглашение о вызовах используется в программе?
12. Перечислите имена сегментов, использующихся в программе.
13. Какой участок в программе отмечает директива .CODE?
14. Какие директивы ассемблера определяют начало и конец процедуры?
15. Для чего предназначена директива PROTO?
16. Какие WinAPI-функции вызываются в программе?
17. Назовите используемые метки и их назначение в программе.
18. Какая функция позволяет завершить процесс в Windows? С каким кодом завершается программа Asm01?
19. Модифицируйте исходный текст программы. Выполните ее.

```

.586P                                ; система команд(процессор Pentium)
.MODEL FLAT, STDCALL                 ; модель памяти, соглашение о вызовах
includelib kernel32.lib              ; компановщику: компоновать с kernel32

ExitProcess PROTO : DWORD             ; прототип функции для завершения процесса Windows
MessageBoxA PROTO : DWORD, : DWORD, : DWORD, : DWORD ; прототип API - функции MessageBoxA

.STACK 4096                           ; выделение стека

.CONST                               ; сегмент констант

.DATA                                ; сегмент данных
MB_OK EQU 0                          ; EQU определяет константу
STR1 DB "Моя первая программа", 0    ; строка, первый элемент данные + нулевой бит
STR2 DB "Привет всем!", 0           ; строка, первый элемент данные + нулевой бит
HW DD ?                             ; двойное слово длиной 4 байта, неинициализировано

.CODE                                ; сегмент кода

main PROC                           ; точка входа main
START :                             ; метка

    INVOKE MessageBoxA, HW, OFFSET STR2, OFFSET STR1, MB_OK

    push - 1                         ; код возврата процесса Windows(параметр ExitProcess)
    call ExitProcess                 ; так завершается любой процесс Windows
main ENDP                           ; конец процедуры

end main                             ; конец модуля main

```

20. Какая директива ассемблера позволяет автоматизировать вызов функции и передачу ей параметров?
21. Опишите окно с сообщением, которое выводит на экран функция MessageBox.