МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ЕВФРОСИНИИ ПОЛОЦКОЙ»

Факультет информационных технологий Кафедра технологий программирования

**Отчет по выполнению лабораторной работы №4**

**по курсу «Конструирование программного обеспечения»**

«Использование подпрограмм в языке низкого уровня»

Выполнил: студент группы 22-ИТ-зд

Шастовская М.С.

Проверил:

Скуковская А.А.

Полоцк, 2024

**Цель:** Целью работы приобретение навыков написания подпрограмм (процедур) при программировании на языке ассемблера.

**Задание 1.** Вывод на экран дисплея результата работы программы при помощи

процедуры.

1. Для выполнения данной практической работы необходимо восстановить и

отладить программу – Подсчет количества чисел, равных 0, в массиве из 10 чисел.

2. Получить объектный файл программы приложения А.

3. Отладив программу подсчета количества чисел, равных 0, сделать

изменения в ней:

3.1. Директивой EXTRN MAS10:proc — объявить имя внешней процедуры;

3.2. Вставить команду загрузки результата в регистр CL;

3.3. Вставить команду CALL MAS10 — вызов подпрограммы.

4. Оттранслировать новый вариант программы, получить второй объектный

файл.

5. При помощи команды TLINK скомпоновать объектные файлы программы

приложения А и модифицированной программы подсчета чисел, равных 0, в

исполняемую программу.

6. Запустить полученный загрузочный модуль. При правильной работе на

экран будет выведен результат – сколько нулей в массиве.

**Решение**

Листинг 1.1 – исходный код фала main.asm

option casemap:none

.model flat, stdcall

includelib kernel32

extern GetStdHandle@4: near

extern WriteFile@20: near

extern ExitProcess@4: near

extern MAS10@0: near

stack segment para 'STACK'

DB 64 DUP('STACK')

stack ends

cseg segment para 'CODE'

\_start:

call MAS10@0

push 0

call ExitProcess@4

cseg ends

end \_start

Листинг 1.2 – исходный код фала mas10.asm

option casemap:none

.model flat, stdcall

includelib kernel32.lib

extern GetStdHandle@4: near

extern WriteFile@20: near

extern ExitProcess@4: near

public MAS10

stack segment para 'STACK'

DB 64 DUP('STACK')

stack ends

dseg segment para 'DATA'

array DWORD 1, 0, 5, 0, 0, 8, 0, 3, 0, 7

array\_size = 10

message db "Number of zeros: "

result db '5', 0

buffer db 16 DUP(?)

consoleHandler DWORD 0

dseg ends

cseg segment para 'CODE'

MAS10 proc

push -11

call GetStdHandle@4

mov consoleHandler, eax

mov ecx, array\_size

mov eax, 0

mov esi, OFFSET array

count\_zeros:

cmp dword ptr [esi], 0

je increment\_count

add esi, 4

loop count\_zeros

;mov result, eax

push eax

push offset buffer

push 0

push 0

push 17

push offset message

push consoleHandler

call WriteFile@20

push 0

push 0

push 4

push offset result

push consoleHandler

call WriteFile@20

push 0

call ExitProcess@4

increment\_count:

inc eax

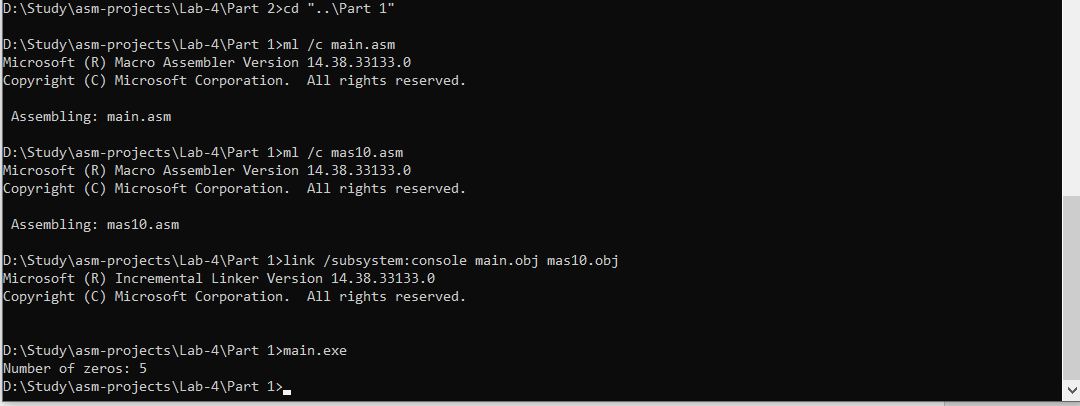
add esi, 4

jmp count\_zeros

MAS10 endp

cseg ends

end



**Задание 2.** Выделение из программы блока обработки массива в отдельную

процедуру.

1. Для выполнения данной части практической работы необходимо

воспользоваться отлаженной программой подсчёта в массиве чисел, равных 0

(исходная программа первой части).

2. Модифицировать исходную программу, создав две программы – главную

и подпрограмму. Подпрограмма должна содержать блок обработки массива (поиск

элементов, равных 0). Передачу параметров осуществить через стек.

3. Далее осуществить действия, аналогичные пп. 4-6 предыдущего задания.

**Решение**

Листинг 2.1 – исходный код фала main.asm

option casemap:none

.model flat, stdcall

includelib kernel32

extern GetStdHandle@4: near

extern WriteFile@20: near

extern ExitProcess@4: near

extern MAS10@0: near

stack segment para 'STACK'

DB 64 DUP('STACK')

stack ends

cseg segment para 'CODE'

\_start:

call MAS10@0

push 0

call ExitProcess@4

cseg ends

end \_start

Листинг 2.2 – исходный код фала mas10.asm

option casemap:none

.model flat, stdcall

includelib kernel32.lib

extern GetStdHandle@4: near

extern WriteFile@20: near

extern ExitProcess@4: near

public MAS10

stack segment para 'STACK'

DB 64 DUP('STACK')

stack ends

dseg segment para 'DATA'

array DWORD 1, 0, 5, 0, 0, 8, 0, 3, 0, 7

array\_size = 10

message db "Number of zeros: "

result db '5', 0

buffer db 16 DUP(?)

consoleHandler DWORD 0

dseg ends

cseg segment para 'CODE'

subMAS10S proc

mov ecx, array\_size

mov eax, 0

mov esi, OFFSET array

count\_zeros:

cmp dword ptr [esi], 0

add esi, 4

loop count\_zeros

ret increment\_count:

inc eax add esi, 4

jmp count\_zeros

subMAS10S endp

MAS10 proc

push -11

call GetStdHandle@4

mov consoleHandler, eax

call subMAS10S

mov result, eax

push eax

push offset buffer

push 0

push 0

push 17

push offset message

push consoleHandler

call WriteFile@20

push 0

push 0

push 4

push offset result

push consoleHandler

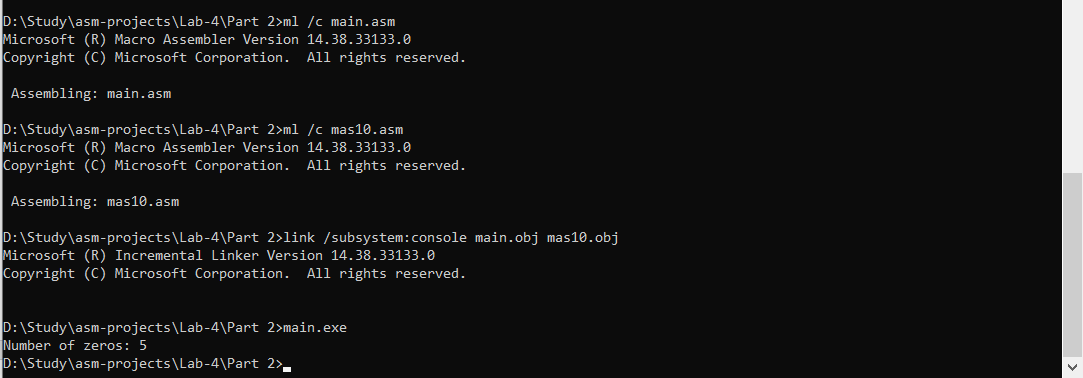
call WriteFile@20

ret

MAS10 endp

cseg ends

end



**Вывод:** Изучила механизм и инструменты для работы со строками на языке низкого уровня.