МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Вятский государственный университет»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

ЛАБОРАТОРНАЯ РАБОТА №4

«Организация связи программы на языке высокого уровня и программы на Assembler»

Отчёт по лабораторной работе дисциплины

«Технологии программирования»

Выполнил студент группы ИВТб-21\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К. А./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Долженкова М. Л./

Киров 2022

**Цель работы:** изучить основы написания программ на низкоуровневом языке Assembler. Также изучить основы написания программных модулей и их подключение к программам.

**Задание:** Реализовать одну из операций лабораторной разрыты 3 на языке Assembler. Оформить ее в виде отдельного модуля и подключить к проекту приложения. Для работы с параметрами использовать содержимое стека. Проанализировать особенности вызова функций и возврата результатов работы.

**Программный модуль** "scal.asm"**:**

.MODEL FLAT, C

.DATA

.CODE

;УМНОЖЕНИЕ

multy proc mas: dword, n: dword, num: dword

push esi

push edx

push eax

mov esi, dword ptr [ebp + 8]

mov eax, dword ptr [ebp + 16]

mul eax

mov ecx, eax

mov eax, 0

l1:

mov eax, [esi]

mov edx, dword ptr [ebp + 12]

imul edx

mov [esi], eax

add esi, 4

dec ecx

cmp ecx, 0

jne l1

pop eax

pop edx

pop esi

ret

multy endp

;ДЕЛЕНИЕ

divide proc mas: dword, n: dword, num: dword

push esi

push edx

push eax

mov esi, dword ptr [ebp + 8]

mov eax, dword ptr [ebp + 16]

mul eax

mov ecx, eax

mov eax, 0

mov ebx, dword ptr [ebp + 12]

l1:

mov eax, [esi]

bt eax, 31

jc ll1

jnc ll2

ll2:

mov edx, 0

jmp ll3

ll1:

mov edx, 4294967295

ll3:

idiv ebx

mov [esi], eax

add esi, 4

dec ecx

cmp ecx, 0

jne l1

pop eax

pop edx

pop esi

ret

divide endp

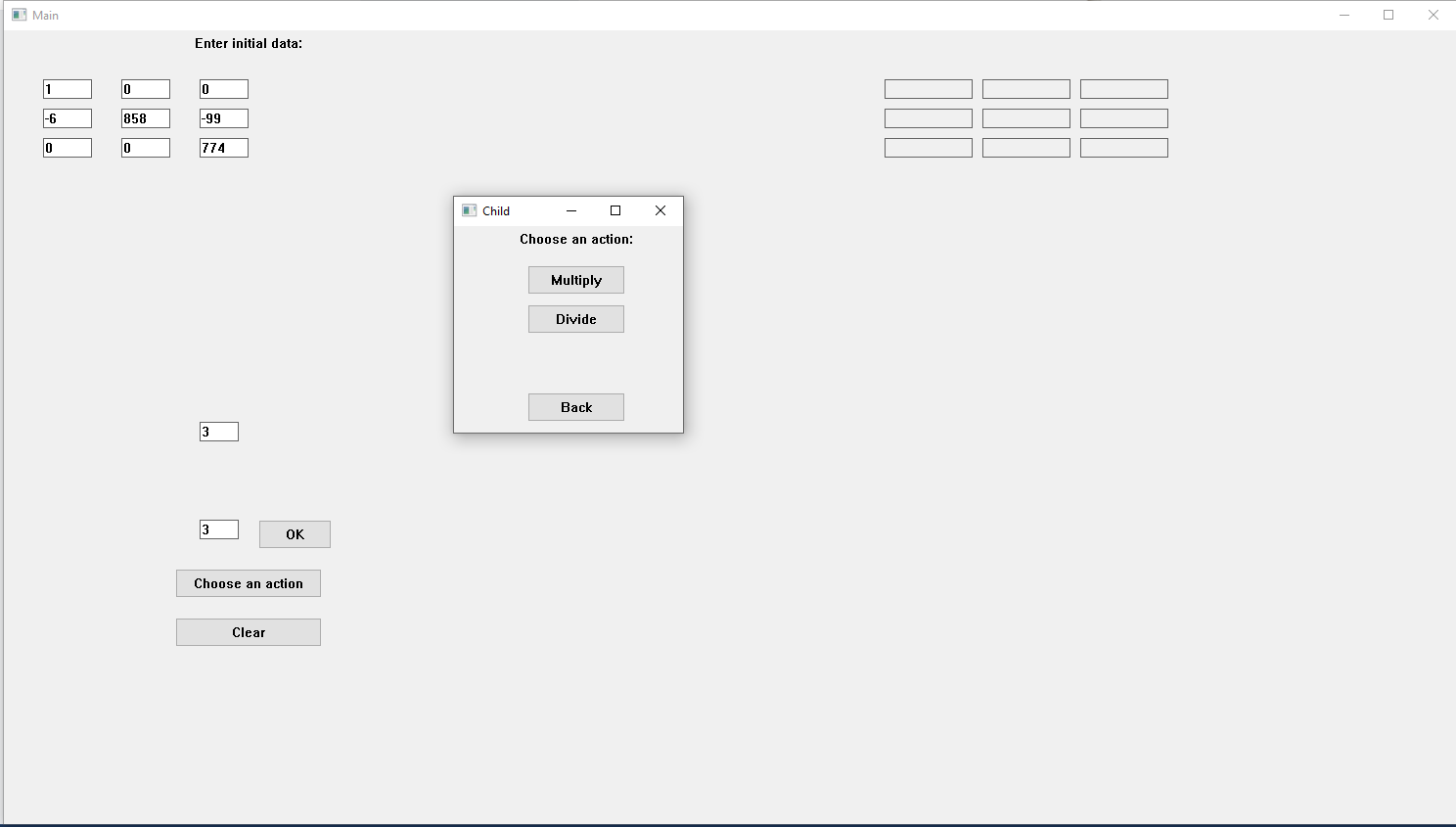
end

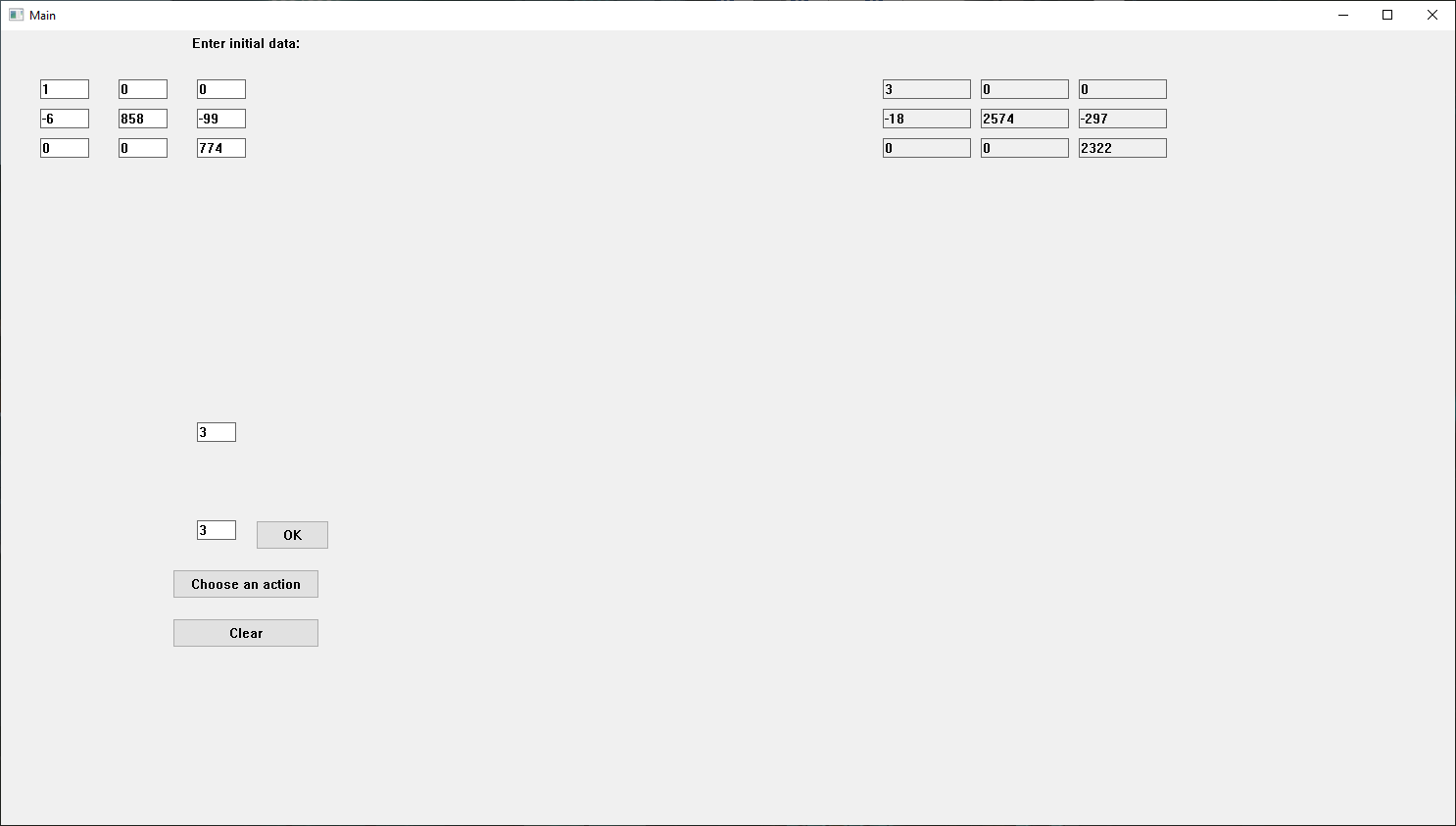
**Фрейм стека**

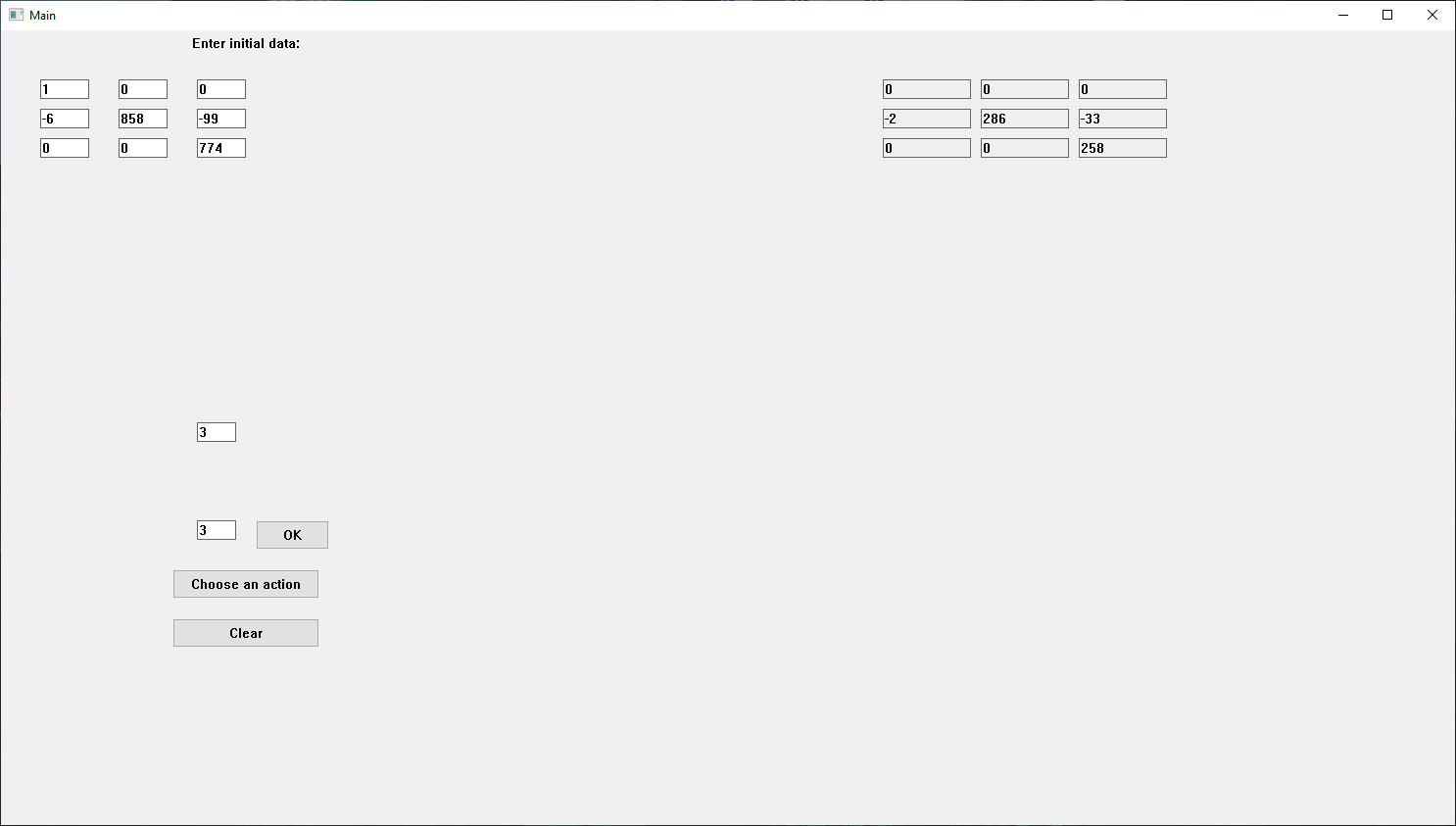
|  |  |
| --- | --- |
| \*mas | Ebp+8 |
| n | Ebp+12 |
| num | Ebp+16 |

**Экранные формы программы:**









**Вывод:**

Таким образом, в результате выполнения данной лабораторной работы был написан программный модуль на Assembler, который умножает или делит матрицу на скаляр. Данный модуль был подключен к программе из лабораторной работы №3.

Для написания программного модуля на языке Assembler был использован встроенный в Visual Studio компилятор MASM, на его диалекте структура программы выглядит следующим образом:

.MODEL FLAT, C - модель памяти 32bit, передача параметров на синтаксис С

.STACK - При использовании с [. MODEL](https://docs.microsoft.com/ru-ru/cpp/assembler/masm/dot-model?view=vs-2019) определяет сегмент стека. Необязательный размер определяет число байтов для стека. Директива автоматически закрывает оператор .STACK.

.DATA - При использовании с . MODEL запускает ближайший сегмент данных для инициализированных данных.

.CODE - Cекция с кодом

end

Для подключения программного модуля, нужно подключить в сборку в Visual Studio компилятор MASM, создать .asm файл и сохранить в проекте. Также нужно зайти в свойства .asm файла, свойства конфигурации, общие и выбрать тип элемента “Microsoft Macro Assembler”. Затем необходимо составить базовую структуру с созданием процедуры, в данном случае multy PROC и divide PROC. Затем в коде C++ необходимо подключить прототип импортируемой функции, в данном случае:

extern "C" void multy(int\*mas, int n, int number);

extern "C" void divide(int\*mas, int n, int number);

Из всего вышеперечисленного можно сделать вывод, что были изучены основы написания программ на низкоуровневом языке Assembler, а также основы написания программных модулей и их подключение к программам.