

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №6
по дисциплине «Организация ЭВМ и систем»
Тема: Организация связи Ассемблера с ЯВУ на примере программы
построения частотного распределение попаданий псевдослучайных
целых чисел в заданные интервалы

Студент гр. 9383

Лапина А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Научиться организовывать связь Ассемблера и ЯВУ. Написать программу построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы.

Текст задания.

На языке высокого уровня (Pascal или C) генерируется массив псевдослучайных целых чисел, изменяющихся в заданном диапазоне и имеющих равномерное распределение. Необходимые датчики псевдослучайных чисел находятся в каталоге Tasks\RAND_GEN (при его отсутствии программу датчика получить у преподавателя).

Далее должен вызываться ассемблерный модуль(модули) для формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы. В общем случае интервалы разбиения диапазона изменения псевдослучайных чисел могут иметь различную длину.

Результирующий массив частотного распределения чисел по интервалам, сформированный на ассемблерном уровне, возвращается в программу, реализованную на ЯВУ, и затем сохраняется в файле и выводится на экран средствами ЯВУ.

Подпрограмма формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы реализуется в виде двух ассемблерных модулей, первый из которых формирует распределение исходных чисел по интервалам единичной длины и возвращает его в вызывающую программу на ЯВУ как промежуточный результат. Это распределение должно выводиться в текстовом виде для контроля. Затем вызывается второй ассемблерный модуль, который по этому промежуточному распределению формирует окончательное распределение псевдослучайных целых чисел по интервалам произвольной длины (с заданными границами). Это распределение

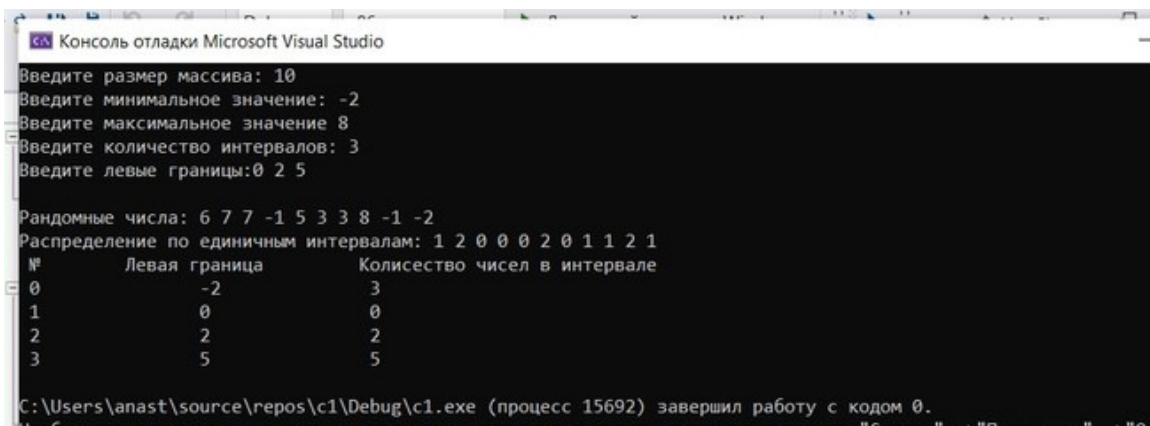
возвращается в главную программу и выдается как основной результат в виде текстового файла и, возможно, графика.

Ход работы.

- 1) Создаем файл main.cpp, в котором происходит считывание данных, вызов функций, из файлов m1.asm, и m2.asm, а также вывод обработанных данных. с помощью спецификатора extern организована связь ассемблерных модулей и основного кода.
- 2) В файле m1.asm создаем модуль, в котором записываем распределение повторений числа по единичным отрезкам.
- 3) В файле m2.asm создаем модуль, в котором формируем числа по интервалам, с помощью вышесформированного в 1 модуле массива.

Тестирование.

1)

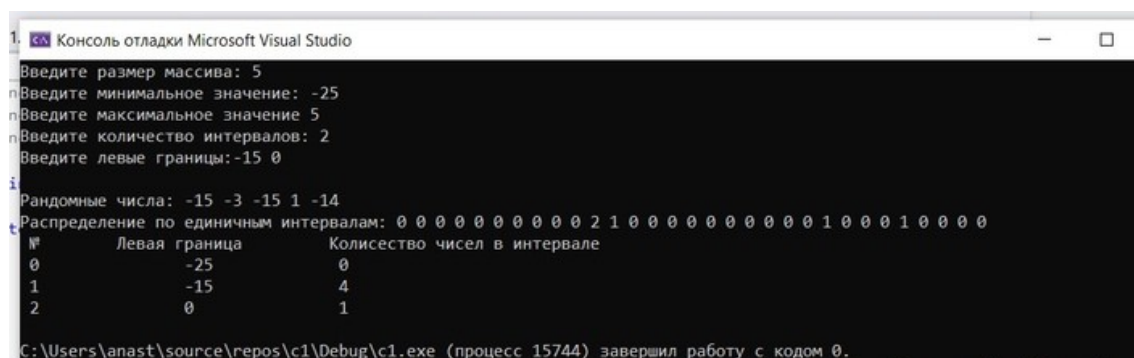


```
Консоль отладки Microsoft Visual Studio
Введите размер массива: 10
Введите минимальное значение: -2
Введите максимальное значение 8
Введите количество интервалов: 3
Введите левые границы:0 2 5

Рандомные числа: 6 7 7 -1 5 3 3 8 -1 -2
Распределение по единичным интервалам: 1 2 0 0 0 2 0 1 1 2 1
№      Левая граница      Количество чисел в интервале
0      -2                  3
1      0                   0
2      2                   2
3      5                   5

C:\Users\anast\source\repos\c1\Debug\c1.exe (процесс 15692) завершил работу с кодом 0.
```

2)



```
Консоль отладки Microsoft Visual Studio
Введите размер массива: 5
Введите минимальное значение: -25
Введите максимальное значение 5
Введите количество интервалов: 2
Введите левые границы:-15 0

Рандомные числа: -15 -3 -15 1 -14
Распределение по единичным интервалам: 0 0 0 0 0 0 0 0 2 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0
№      Левая граница      Количество чисел в интервале
0      -25                0
1      -15                4
2      0                  1

C:\Users\anast\source\repos\c1\Debug\c1.exe (процесс 15744) завершил работу с кодом 0.
```

3)

```
Консоль отладки Microsoft Visual Studio
Введите размер массива: 7
Введите минимальное значение: -5
Введите максимальное значение: 1
Введите количество интервалов: 3
Введите левые границы: -3 -1 0

Рандомные числа: 1 -4 1 0 -2 -3 0
Распределение по единичным интервалам: 0 1 1 1 0 2 2
№      Левая граница      Количество чисел в интервале
0       -5                1
1       -3                2
2       -1                0
3        0                4

C:\Users\anast\source\repos\c1\Debug\c1.exe (процесс 1784) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка"
```

Выводы.

Получены знания об организации связи Ассемблера и ЯВУ. Написана программа построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы.

Содержимое файлов main.cpp, m1.asm, m2.asm представлено в приложении А.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>
#include <random>
#include <fstream>

using namespace std;

extern "C"
{
    void M1(int* resArray, int* arr, int size, int xMin);
    void M2(int* resArray, int resSize, int* leftBorders,
            int intervalsCount, int xMin, int xMax, int* finalArray);
}

int main()
{
    setlocale(0, "");
    int arraySize;
    cout << "Введите размер массива: ";
    cin >> arraySize;

    int xMin, xMax;
    cout << "Введите минимальное значение: ";
    cin >> xMin;
    cout << "Введите максимальное значение ";
    cin >> xMax;
    if (xMin > xMax)
    {
        cout << "Максимальное значение должно быть больше минимального!\n";
        return 0;
    }
    int intervals;
    cout << "Введите количество интервалов: ";
    cin >> intervals;
    int* leftBorders = new int[intervals];
    cout << "Введите левые границы:";
    for (int i = 0; i < intervals; i++)
    {
        cin >> leftBorders[i];
    }
    int* array = new int[arraySize];

    for (int i = 0; i < arraySize; i++)
    {
        array[i] = xMin + rand() % (xMax - xMin + 1);
    }
    int* res1 = new int[abs(xMax - xMin) + 1];
    for (int i = 0; i < abs(xMax - xMin) + 1; i++)
```

```

        {
            res1[i] = 0;
        }
M1(res1, array, arraySize, xMin);
cout << endl;
int* finalArray = new int[intervals + 1];
for (int i = 0; i < intervals + 1; i++) {
    finalArray[i] = 0;
}
M2(res1, arraySize, leftBoarders, intervals, xMin, xMax, finalAr-
ray);
ofstream textFile;
textFile.open("out.txt");
cout << "Рандомные числа: ";
textFile << "Рандомные числа: ";
for (int i = 0; i < arraySize; i++)
{
    cout << array[i] << " ";
    textFile << array[i] << " ";
}
cout << endl;
textFile << endl;
cout << "Распределение по единичным интервалам: ";
textFile << "Распределение по единичным интервалам: ";
for (int i = 0; i < abs(xMax - xMin) + 1; i++)

{
    cout << res1[i] << " ";
    textFile << res1[i] << " ";
}
cout << endl;
textFile << endl;
cout << " № \t " << "Левая граница \t" << "Количество чисел в ин-
тервале\n";
textFile << "Номер интервала " << "Левая граница интервала " <<
"Кол - во чисел в интервале\n";
cout << " " << 0 << " \t \t " << xMin << " \t\t " << finalArray[0]
<< endl;
textFile << " " << 0 << " \t\t\t\t " << xMin << " \t\t\t\t " << fi-
nalArray[0] << endl;
for (int i = 1; i < intervals + 1; i++)
{
    cout << " " << i << " \t \t " << leftBoarders[i - 1] + xMin <<
" \t\t " << finalArray[i] << endl;
    textFile << " " << i << " \t\t\t\t " << leftBoarders[i - 1] +
xMin << " \t\t\t\t " << finalArray[i] << endl;
}
}

```

Название файла: M1.asm

```

.586p
.MODEL FLAT, C
.CODE
PUBLIC C M1

```

```

M1 PROC C RESARRAY:DWORD, ARRAY:DWORD, SIZEARR:DWORD, XMIN:DWORD
PUSH ESI
PUSH EDI;сохранение регистров

PUSH EBP
MOV EDI, RESARRAY
MOV ESI, ARRAY
MOV EAX, XMIN
MOV ECX, SIZEARR
for1:                ;метка для цикла
MOV EBX, [ESI]        ;берем число из массива
SUB EBX, EAX          ;вычитаем минимальное
MOV EBP, [EDI+4*EBX]   ;обращаемся к соответствующей ячейке результиру-
ющего массива
INC EBP               ;увеличиваем его на 1
MOV [EDI+4*EBX], EBP   ;и кладем назад
ADD ESI, 4            ;для перехода к следующему элементу
LOOP for1             ;цикл уменьшает ECX пока не 0
POP EBP
POP EDI
POP ESI
ret
M1 ENDP
END

```

Название файла: M2.asm

```

.586p
.MODEL FLAT, C
.CODE
PUBLIC C M2
M2 PROC C RESARRAY:DWORD, SIZEARRAY:DWORD, BOARDERS:DWORD,
BOARDERSIZE:DWORD, XMIN:DWORD, XMAX:DWORD, FINALARRAY:DWORD

MOV ESI, BOARDERS      ;массив границ
MOV ECX, BOARDERSIZE   ;количество границ
MOV EAX, 0             ;индекс результирующего массива
MOV EDX, XMIN          ;минимальное значение

for1:                  ;вычитаем из каждого и интервала минимальное
MOV EAX, [ESI]
SUB EAX, XMIN
MOV [ESI], EAX
ADD ESI, 4
loop for1

MOV EDI, RESARRAY      ;массив
MOV ECX, BOARDERSIZE   ;количество интервалов
MOV ESI, BOARDERS      ;интервалы
MOV EBX, 0
MOV EAX, [ESI]

for2i:
PUSH ECX

```

```

MOV ECX, EAX ;кладем границы и выполняем цикл до нее
PUSH ESI
MOV ESI, FINALARRAY
    for2j:
        MOV EAX, [EDI]
        ADD [ESI + EBX*4], EAX
        ADD EDI, 4
    loop for2j
POP ESI
MOV EAX, [ESI]
ADD ESI, 4
SUB EAX, [ESI]
NEG EAX

INC EBX ;для обращения к следующему элементу FINALARRAY
POP ECX
loop for2i

MOV ESI, FINALARRAY
MOV ECX, BOARDERSIZE
MOV EAX, 0

    for3: ;для посчета последнего инрвала
        ADD EAX, [ESI] ;считаем сумму подсчитанных интегралов
        ADD ESI, 4
    loop for3

SUB EAX, SIZEARRAY ;вычитаем размер массива
NEG EAX ;меняем знак

MOV ESI, FINALARRAY

ADD [ESI + 4 * EBX], EAX ;записываем последний элемент

ret
M2 ENDP
END

```