

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

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

Студент гр. 9382

Демин В.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Научиться организовывать связь Ассемблера с ЯВУ на примере программы построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы.

Задание.

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.

Исходные данные.

1. Длина массива псевдослучайных целых чисел - NumRanDat ($\leq 16K$, $K=1024$)

2. Диапазон изменения массива псевдослучайных целых чисел $[X_{\min}, X_{\max}]$, значения могут быть биполярные;

14

3. Количество интервалов, на которые разбивается диапазон изменения массива псевдослучайных целых чисел - NInt (≤ 24)

4. Массив левых границ интервалов разбиения LGrInt (должны принадлежать интервалу $[X_{\min}, X_{\max}]$).

Результаты:

1. Текстовый файл, строка которого содержит:

- номер интервала,
- левую границу интервала,
- количество псевдослучайных чисел, попавших в интервал.

Количество строк равно числу интервалов разбиения.

2. График, отражающий распределение чисел по интервалам.

(необязательный результат)

В зависимости от номера бригады формирование частотного распределения должно производиться по одному из двух вариантов:

1. Для бригад с нечетным номером: подпрограмма формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы реализуется в виде одного ассемблерного модуля, сразу формирующего требуемое распределение и возвращающего его в

головную программу, написанную на ЯВУ;

Выполнение работы.

На языке C++ происходит подготовка данных к обработке в написанном модуле ассемблера. Метод ASM_FUN(LGrInt, arr, res, NInt, NumRanDat) который считает количество чисел входящих в заданные интервалы.

Выводы.

В процессе выполнения работы был написан модуль на языке ассемблера, который используется в программе написанное на языке C++.

Тестирование.

№	Входные данные	Выходные данные
1	67 51 44 8 34 53 88 16 17 17 54 72 91 79 25	0 (0 ,25): 4 1 (25 ,75): 4 2 (75 ,100): 3
2	1 5 9 2 10 2 10 7 0 8	0 (0 ,10): 7
3	9 -60 30 -100 87 91 -93 - 44 -74 52 76 -86 -8 66 39 56 -87 -56 49 -51 83 51 - 75 72 31 87 92 -66 -22 54 65 -85 61 53 -27 -34 -94 - 76 81 -72 -76 -78 -53 -60 - 27 -81 -47 77 -74 66 59 2 50 -37 -45 -13 -53 -51 0 - 91 22 -37 31 -55 -24 96 - 46 -23 32 -30 -2 -95 -33 29 11 50 -91 -21 71 -80 -4 27 4 91 1 -36 -58 -23 -56 -50 -98 87 -14 -98 -63 -79 -69 14 0 -22	0 (-100 ,-80): 53 1 (-80 ,-60): 48 2 (-60 ,-40): 50 3 (-40 ,-20): 52 4 (-20 ,0): 0 5 (0 ,20): 5 6 (20 ,40): 7 7 (40 ,60): 9 8 (60 ,100): 15

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: MAIN.CPP

```
//

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

using namespace std;

extern "C" {
    void ASM_FUN(int* LGrIn, int* array, int* res, int NInt, int
NumRanDat);
}

int main()
{
    int NumRanDat = 0; //Длина массива числе
    int Xmin = 0; //левая граница
    int Xmax = 0; //правая границв
    int NInt = 0; //количество интервалов
    int* LGrInt = nullptr; // массив левых границ интервалов
    int* arr = nullptr; // массив целых чисел
    int* res = nullptr; // результирующий массив

    cout << "Enter the length of the array\n";
    cin >> NumRanDat;
    while (1) {
        if (NumRanDat > 0 && NumRanDat <= 16 * 1024) {
            break;
        }
        cout << "Wrong: try again\n";
        cout << "Enter the length of the array\n";
        cin >> NumRanDat;
    }
    arr = new int[NumRanDat];
    cout << "Enter the bottom border\n";
    cin >> Xmin;
    cout << "Enter the upper border\n";
    cin >> Xmax;

    cout << "Enter number of intervals\n";
    cin >> NInt;

    while (1) {
        if (NInt > 0 && NInt <= 24) {
            break;
        }
        cout << "Wrong: try again\n";
    }
}
```

```

        cout << "Enter number of intervals\n";
        cin >> NInt;
    }
    LGrInt = new int[NInt];
    res = new int[NInt];
    cout << "Enter " << NInt - 2 << " intervals(!=Xmin&&!=Xmax)\n";
    for (int i = 1; i < NInt - 1; i++)
    {
        cin >> LGrInt[i];
        while (1) {
            if (LGrInt[i] >= Xmin && LGrInt[i] <= Xmax) {
                break;
            }
            cout << "Wrong: try again\n";
            cin >> LGrInt[i];
        }
    }
    LGrInt[0] = Xmin;
    LGrInt[NInt - 1] = Xmax;

    //дискретная равномерное распределение
    std::random_device rand_dev;
    std::mt19937 generator(rand_dev());
    std::uniform_int_distribution<int> distr(Xmin, Xmax);
    for (int i = 0; i < NumRanDat; i++)
    {
        arr[i] = distr(generator);
    }
    for (int i = 0; i < NInt; i++)
    {
        res[i] = 0;
    }

    //вызов ассемблерной функции
    ASM_FUN(LGrInt, arr, res, NInt, NumRanDat);

    std::ofstream file;
    file.open("out.txt");

    for (int i = 0; i < NumRanDat; i++)
    {
        cout << arr[i] << " ";
    }

    cout << "\n";

    for (int i = 0; i < NInt-1; i++)
    {
        cout<<"№ " << i << " (" << LGrInt[i] << " ," <<
LGrInt[i+1]<<"): " << res[i] << "\n";
    }
    for (int i = 0; i < NInt-1; i++)
    {
        file << i << " " << LGrInt[i] << " " << res[i]<<"\n";
    }
}

```

```

    return 0;
}

```

Название файла: fun1.asm

```

.586
.model flat, C
.code
ASM_FUN    proc    C    LGrIn:dword,    array:dword,    res:dword,
NInt:dword,NumRanDat:dword
    push eax
    push ecx
    push ebx
    push edx
    sub eax,eax
    sub ecx,ecx

for_first:
    ;от 0 до NInt-1
    mov edx,array
    for_second:
        ;от 0 до NumRanDat

        ;проверка условия if

        mov ebx,LGrIn
        mov ebx,[ebx+eax*4]
        cmp ebx,[edx+ecx*4]
        jl first_if
        jmp break_if

    first_if:
        mov ebx,LGrIn
        mov ebx,[ebx+eax*4+4]
        cmp ebx,[edx+ecx*4]
        ja second_if
        jmp break_if

    second_if:
        mov edx,res
        mov ebx,[edx+eax*4]
        inc ebx
        mov [edx+eax*4],ebx
        mov edx,array
    break_if:

    inc ecx
    cmp ecx,NumRanDat
    jl for_second
mov ecx,NInt
sub ecx,1
inc eax
cmp eax,ecx
jl for_first

```

```

        ;for (int i = 0; i < NInt - 1; ++i) {+
;   for (int j = 0; j < NumRanDat; ++j) {      +
;       if (array[j] > LGrIn[i] && array[j] < LGrIn[i + 1])
{
        ;           res[i]++;
;           }
;       }
;   }
pop edx
pop ebx
pop ecx
pop eax

ret
ASM_FUN endp
end

```