

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 распределение попаданий псевдослучайных
целых чисел в заданные интервалы

Студент гр. 9382

Сорокумов С. В.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2020

Цель работы.

Научиться связывать язык Ассемблера и язык высокого уровня C++ так, чтобы функции ассемблерного модуля вызывались из программы на C++.

Задание.

Реализовать программу формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы реализуется в виде одного ассемблерного модуля (процедуры), сразу получающего требуемое распределение и возвращающего его в головную программу, написанную на ЯВУ.

Ход работы.

На языке высокого уровня программируется ввод с клавиатуры и контроль исходных данных, а также генерируется массив псевдослучайных целых чисел, изменяющихся в заданном диапазоне и имеющих равномерное распределение.

Далее должны вызываться ассемблерная процедура для формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы.

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 ($\leq 16K$)
2. Диапазон изменения массива псевдослучайных целых чисел
[Xmin, Xmax] (м.б. биполярный, например, [-100, 100])
3. Массив псевдослучайных целых чисел.

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

5. Массив левых границ интервалов разбиения LGrInt
(должны принадлежать интервалу [Xmin,Xmax])

В общем случае интервалы разбиения диапазона изменения псевдослучайных чисел могут иметь различную длину.

Тестирование.

Размер массива	Диапазон [X_min, X_max]	Массив	Левые границы	Количество значений
15	[2; 20]	6 4 7 18 10 14 14 13 4 7 13 20 17 5 6	1) 2 2) 5 3) 8 4) 12	
10	[-8, 2]	-1 -1 -6 -7 0 -3 -2 -3 1 -5	1) -8 2) -5 3) -2	
5	[3, 15]	5 12 6 4 15	1) 3 2) 5	
8	[-9, 20]	-8 6 -5 -7 11 -8 -5 8	1) -9 2) -1 3) 6 4) 12	

Вывод.

В ходе выполнения данной лабораторной работы была написана программа на языке Ассемблера, которая строит частотное распределение попаданий псевдослучайных чисел в заданные интервалы.

ПРИЛОЖЕНИЕ А

Исходный код программы source.cpp

```
#include<cstdlib>
#include<iostream>
#include"windows.h"
#include"fstream"
#include<ctime>
#include<iomanip>

using namespace std;

void generation(int *, int, int, int);
int compare(const void *, const void *);

extern"C"
{
    void poisk(int NumRanDat, int *arr, int *LG, int *kol, int NInt);
}

void out1(ofstream&out)
{
    cout << "Результат работы программы:" << endl;
    out << "Результат работы программ:" << endl;
    cout << setw(15) << "Номер интервала ";
    cout << setw(20) << "Границы ";
    cout << setw(35) << "Количество чисел в интервале" << endl;
    out << setw(15) << "Номер интервала ";
    out << setw(20) << "Границы ";
    out << setw(35) << "Количество чисел в интервал" << endl;
}

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    ofstream fout("output.txt");
    int NumRanDat, Xmin, Xmax, NInt, i, j;
    int *kol; //массив для подсчета количества повторений на интервале
    int *arr; //массив псевдослучайных чисел
```

```

int *RG;           //массив правых границ
int *LG;           //массив левых границ

cout << "Введите длину массива
псевдослучайных чисел (<16000)" << endl;
cin >> NumRanDat;
while ((NumRanDat >= 16000) || (NumRanDat <= 0)) {
//проверка корректности
    cout << "Неверная длина массива.
Повторите ввод" << endl;
    cin >> NumRanDat;           //считывание
количества чисел
}
arr = new int[NumRanDat];       //выделение памяти
под массив псевдослучайных чисел
cout << "Введите начало диапазона
изменений Xmin" << endl;
cin >> Xmin;                   //считывание
минимального значения диапазона
cout << "Введите конец диапазона
изменений Xmax" << endl;
cin >> Xmax;                   //считывание
максимального значения диапазона
while (Xmin > Xmax) { //проверка соотношения
между максимумом и минимумом
    cout << "Неверное соотношение между
максимумом и минимумом диапазона" << endl;
    cin >> Xmin;
    cin >> Xmax;
}
generation(arr, NumRanDat, Xmin, Xmax); //вызов
функции для генерации массива
    cout << "Массив псевдослучайных
чисел:" << endl;
    fout << "Массив псевдослучайных чисел:" <<
endl;
    for (i = 0; i < NumRanDat; i++) {
        cout << arr[i] << " ";
        fout << arr[i] << " ";
    }
    cout << endl;
    cout << "Введите количество интервалов
(<24)" << endl;

```

```

    cin >> NInt;                //считывание
количества интервалов
    while ((NInt < 0) || (NInt > 24) || (NInt >= (Xmax - Xmin)))
        //проверка
    {
        cout << "Неверное значение. Повторите
ввод" << endl;
        cin >> NInt;
    }
    kol = new int[NInt];        //выделение памяти
под массив повторений
    RG = new int[NInt];        //выделение памяти
под массив правых границ
    LG = new int[NInt];        //выделение памяти
под массив левых границ
    cout << "Введите левые границы
интервалов" << endl;
    if (NInt != 0)
    {
        LG[0] = Xmin;        //крайняя левая
граница-минимальное значение
псевдослучайных чисел
        i = 0, j = 1;
        for (; i < NInt - 1; i++, j++)
        {
            cin >> LG[j];        //считывание
левых границ
            while ((LG[j] < Xmin) || (LG[j] > Xmax))
                //проверка корректности введенных

            {
                cout << "Неверное значение.
Введите другое значение " << endl;
                cin >> LG[j];
            }
            RG[i] = LG[j] - 1;        //заполняем
массив правых границ

        }
        RG[NInt - 1] = Xmax;        //крайняя правая
граница-максимальное значение

        qsort(LG, NInt, sizeof(int), compare);
        //сортировка массива левых границ по

```

```

        qsort(RG, NInt, sizeof(int), compare);
//сортировка массива правых границ по

        for (int i = 0; i < NInt; i++)
        {
            kol[i] = 0; //обнуление
массива частот
        }
        poisk(NumRanDat, arr, RG, kol, NInt); //вызов
функции для заполнения массива
        out1(fout); //вызов функции для вывод
сообщений на экран и записи их в файл
        for (int i = 0; i < NInt; i++)
        {
            cout << setw(10) << i + 1 << setw(18) << "[" << LG[i]
<< "," << RG[i] << "]" << setw(25) << kol[i] << endl;
            fout << setw(10) << i + 1 << setw(18) << "[" << LG[i]
<< "," << RG[i] << "]" << setw(25) << kol[i] << endl;

        }
    }
    else
    {
        out1(fout);
        cout << setw(10) << 1 << setw(18) << "[" << Xmin << "," <<
Xmax << "]" << setw(25) << NumRanDat << endl;
        fout << setw(10) << 1 << setw(18) << "[" << Xmin << "," <<
Xmax << "]" << setw(25) << NumRanDat << endl;
    }
}

void generation(int *array, int size, int min, int max)
{ //функция для генерации массива

    int i = 0;
    srand(time(0));
    for (i = 0; i < size; i++) {
        array[i] = rand() % (max - min + 1) + min;
    }
}

```

```

int compare(const void *x1, const void *x2) { //функция для
сравнения двух чисел, необходимая

    return(*(int*)x1 - *(int*)x2);
}

```

ПРИЛОЖЕНИЕ Б

Код lab6.asm

```

.386
.model flat, C
.code

poisk proc public, NumRanDat:dword, arr:ptr dword, RG:ptr dword,
kol:ptr dword, NInt:dword
    mov ecx, NumRanDat                ;помещение в
регистр ecx значения количества
псевдослучайных чисел в массиве
    mov esi,0                        ;помещение в регистр esi
значение 0, данный регистр бу-дет
использоваться в качестве счетчика

Chislo:                            ;метка для получения числа из
массива псевдослучайных чисел
    mov eax,arr                      ;помещение в регистр eax
начала массива псевдослучайных чисел
    mov eax,[eax+esi*4]              ;обращение к ячейке
массива псевдослучайных чи-сел, с
использованием счетчика esi
    mov edx,-1                      ;помещение в регистр edx
знаение -1

Interval:
    inc edx                          ;увеличение на 1 значения
регистра, обнуление регистра edx

```



```

        mov ebx,RG          ;помещение в регистр ebx
начала массива границ
        mov ebx,[ebx+edx*4]  ;обращение к ячейке
массива границ с индексом edx
        cmp eax,ebx          ;сравнение
псевдослучайного числа с границей
        jg Interval         ;переход если число
больше

        mov ebx,kol          ;помещение в регистр ebx
начала массива повторений
        mov eax,[ebx+edx*4]  ;помещение в
регистр значения ячейки с индексом edx
        inc eax              ;увеличение на 1 значения
        mov [ebx+edx*4],eax  ;помещение значения
в ячейку с индексом edx
        inc esi              ;увеличение счетчика
        Loop Chislo

        ret
poisk endp
end

```