

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. 1381

Возмитель В.Е.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022 г.

Цель работы

Получить практические навыки программирования на языке Ассемблера. Разработать программу на ЯВУ с использованием языка Ассемблера.

Задание/краткие сведения

Организация связи Ассемблера с ЯВУ на примере программы построения частотного распределение попаданий псевдослучайных целых чисел в заданные интервалы. 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 $[Xmin, Xmax]$, значения могут быть биполярные;
3. Количество интервалов, на которые разбивается диапазон изменения массива псевдослучайных целых чисел - NInt (≤ 24)

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

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).

Вариант работа №4.

Вывод

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

В результате выполнения лабораторной работы были получены практические навыки программирования на языке Ассемблер.

Приложение А. Код программы (LB6.cpp, first.asm, second.asm)

Название файла LB6.cpp:

```
#include <iostream>
#include <ctime>
#include <windows.h>
#include <locale>
#include <algorithm>

using namespace std;

extern "C" {
    void first(int*, int, int*, int);
    void second(int*, int*, int*, int, int);
}

void OutPut1(int* arr, int* L_arr, int NInt)
{
    cout << "\n№ Отрезок Количество чисел" << '\n';
    for (int j = 0; j < NInt; j++)
        cout << j + 1 << " " << L_arr[j] << " \t " << arr[j]
        << '\n';
}

void OutPut2(int* arr, int* L_arr, int NInt)
```

```

{
    cout << "№    Интервал    Количество чисел" << '\n';
    for (int j = 0; j < NInt; j++)
        cout << j + 1 << "    [" << L_arr[j] << ", " << L_arr[j +
1] - 1 << "]\t\t" << arr[j] << '\n';
}

int main()
{
    srand(time(0));
    setlocale(0, "");
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);
    int NumRanDat, Xmin, Xmax, NInt;

    cout << "Введите длину массива: ";
    cin >> NumRanDat;
    if (NumRanDat > 16 * 1024 || NumRanDat < 1)
    {
        cout << "Неверный ввод"; return 0;
    }

    cout << "Введите диапазон чисел\n";
    cout << "Мин: ";
    cin >> Xmin;
    cout << "Макс: ";
    cin >> Xmax;
    if (Xmax < Xmin)
    {
        cout << "Неверный ввод"; return 0;
    }

    int n = Xmax - Xmin + 1; //длина диапазона
    int* arr = new int[NumRanDat]; //массив случайных чисел
    for (int i = 0; i < NumRanDat; i++)
        arr[i] = Xmin + rand() % n;

    cout << endl << "Исходный массив:\n";
    for (int i = 0; i < NumRanDat; i++)
        cout << i + 1 << " значение: " << arr[i] << '\n';

    int* result = new int[n]; //массив полученного распределения
    единичных отрезков
    int* L = new int[n + 1]; //массив левых границ для единичных
    отрезков
    for (int i = 0; i < n; i++)
    {
        result[i] = 0;
        L[i] = Xmin + i;
    }
}

```

```

first(arr, Xmin, result, NumRanDat);
OutPut1(result, L, n);

cout << endl << "Введите количество интервалов разбиения: ";
cin >> NInt;
if (NInt > 24 || NInt < 1 || NInt > n)
{
    cout << "Неверное количество интервалов"; return 0;
}

int* L_array = new int[NInt + 1]; // массив левых границ
L_array[0] = Xmin;
cout << "1 граница: " << Xmin << '\n';
for (int j = 1; j < NInt; j++) {
    cout << j + 1 << " граница: ";
    cin >> L_array[j];
    if ((L_array[j] < Xmin) || (L_array[j] > Xmax))
    {
        cout << "Неверная граница";
        return 0;
    }
}
cout << '\n';

int* res = new int[NInt]; // конечный массив
for (int i = 0; i < NInt; i++)
    res[i] = 0;

L_array[NInt] = Xmax + 1;
second(result, res, L_array, Xmin, Xmax);
OutPut2(res, L_array, NInt);
return 0;

```

Название файла first.asm:

```

.586
.MODEL FLAT, C
.CODE
first PROC C arr:dword, Xmin:dword, result_array:dword,
length_:dword

mov eax, arr
mov ebx, result_array
mov ecx, length_
xor edx, edx
xor edi, edi

cycle:
    mov edi, [eax + 4 * edx] ;значение
    sub edi, Xmin
    inc dword ptr [ebx + 4 * edi] ;увеличение счетчика числа

```

```

        inc edx ;следующее значение
        loop CYCLE
finish:
        ret
        first endp
        end

```

Название файла second.asm:

```

.586
.MODEL FLAT, C
.CODE
second PROC C arr:dword, result:dword, L_array:dword, Xmin:dword,
Xmax:dword

;arr - массив количества всех цифр на единичном отрезке
;result - конечный массива
;L_array - массив левых границ
;Xmin - начальное значение отрезка
;Xmax - конечное значение отрезка

mov eax, Xmin ; значение на отрезке
xor ebx, ebx
xor ecx, ecx
mov edi, result
mov edx, L_array
mov esi, arr

cycle:
        стр eax, [edx + 4] ;если точка вышла за границы первого
отрезка
        jge adding ;переход на новый отрезок

mov ebx, [esi + ecx * 4] ;количество цифр
add[edi], ebx ;добавление к отрезку
inc ecx ;счетчик для массива количества цифр
inc eax ;следующая цифра

        стр eax, Xmax
        jle cycle
        jmp finish

adding:
        add edi, 4 ;следующая ячейка конечного массива
        add edx, 4 ;следующая граница
        jmp cycle

finish:
        ret
        second endp
        end

```

Приложение Б. Тестирование.

Табл. №1: Тестирование программы lab5.asm

Длина массива псевдослучайных чисел	Диапазон изменения массива	Массив псевдослучайных чисел	Массив левых границ	Результирующи й массив
20	[-10; 20]	1 значение: 8 2 значение: 10 3 значение: 15 4 значение: 15 5 значение: 9 6 значение: 12 7 значение: -10 8 значение: -4 9 значение: -2 10 значение: 13 11 значение: 4 12 значение: 1 13 значение: -8 14 значение: -7 15 значение: -1 16 значение: 15 17 значение: 6 18 значение: 18 19 значение: 6 20 значение: 17	1 граница: -10 2 граница: 3 3 граница: 5 4 граница: 8 5 граница: 10	[-10; 2] : 9 [3; 4] : 1 [5; 7] : 3 [8; 9] : 0 [10; 20] : 7
10	[-15; -5]	1 значение: -7 2 значение: -8 3 значение: -9 4 значение: -14 5 значение: -6 6 значение: -7 7 значение: -5 8 значение: -14 9 значение: -8 10 значение: -5	1 граница: -15 2 граница: -9	[-15;-10] : 5 [-9;-5] : 5
5	[-100; 100]	1 значение: -13 2 значение: 6 3 значение: -2 4 значение: 14 5 значение: 40	1 граница: -100 2 граница: 0	[-100;-1] : 1 [0;100] : 4