

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 примерепрограммы
построения частотного
распределение попаданий псевдослучайных целых чисел в заданные
интервалы.

Студент гр. 1383

Валиев Р.Р.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

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

Задание.

На языке высокого уровня (Pascal или C) генерируется массив псевдослучайных целых чисел, изменяющихся в заданном диапазоне и имеющих равномерное распределение.

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)

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 главную программу, написанную на ЯВУ.

Выполнение работы.

На языке C++ реализовано считывание начальных данных. Левые границы заносятся в массив `LgrInt`, а генерируемые числа добавляются в массив `arr`. Отдельно создается массив, который будет хранить результат работы.

В ассемблерный модуль в процедуру `BORDER` передаются указатель на массив сгенерированных чисел и его размер, указатель на массив левых границ интервалов и его размер, указатель на массив, хранящих результат работы. В процедуре для каждого числа находится интервал, в который он попадает, и результат записывается в результирующий массив.

После этого результат работы выводится в консоль и записывается в файл `result.txt`.

Тестирование.

На рисунке 1,2 и 3 отображается работа программы:

```
Количество псевдослучайных чисел: 6
От: 7
До: 69
Количество интервалов: 3
Левые границы интервалов: 9 17 45
Номер интервала  Левая граница интервала  Количество псевдослучайных чисел, попавших в интервал
1                  9                        1
2                  17                       4
3                  45                       1
25 41 38 53 21 13
```

Рисунок 1 – Первый тест

```

Количество псевдослучайных чисел: 6
От: 1
До: 10
Количество интервалов: 3
Левые границы интервалов: 2 6 8
Номер интервала  Левая граница интервала  Количество псевдослучайных чисел, попавших в интервал
1                  2                        1
2                  6                        1
3                  8                        3
8 9 8 4 7 1

```

Рисунок 2 – Второй тест

```

Количество псевдослучайных чисел: 5
От: -5
До: 10
Количество интервалов: 3
Левые границы интервалов: -4 0 3
Номер интервала  Левая граница интервала  Количество псевдослучайных чисел, попавших в интервал
1                  -4                        2
2                   0                        1
3                   3                        2
-2 5 1 3 -3

```

Рисунок 3 – Третий тест

Выводы.

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

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab6.cpp

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

```
extern "C" void BORDER(int* array, int array_size, int* LGrInt, int NInt, int*
result_array);
```

```
int main() {
    system("chcp 1251 > nul");
    setlocale(LC_CTYPE, "rus");
    std::ofstream file("result.txt");
    int NumRanDat;
    std::cout << "Количество псевдослучайных чисел: ";
    std::cin >> NumRanDat;
    int Xmax, Xmin;
    std::cout << "От: ";
    std::cin >> Xmin;
    std::cout << "До: ";
    std::cin >> Xmax;
    if (Xmax < Xmin) {
        std::cout << "Ошибка\n";
        return 0;
    }

    int NInt;
    std::cout << "Количество интервалов: ";
    std::cin >> NInt;
    if (NInt <= 0) {
        std::cout << "Ошибка\n";
        return 0;
    }

    int* LGrInt = new int[NInt];
    std::cout << "Левые границы интервалов: ";
    for (int i = 0; i < NInt; i++) {
        std::cin >> LGrInt[i];
    }

    std::random_device rand;
```

```

std::mt19937 gen(rand());
std::uniform_int_distribution<> dis(Xmin, Xmax);
int* arr = new int[NumRanDat];
for (int i = 0; i < NumRanDat; i++) {
    arr[i] = dis(gen);
}

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

file << '\n';
int* result_arr = new int[NInt];
for (int i = 0; i < NInt; i++) {
    result_arr[i] = 0;
}

BORDER(arr, NumRanDat, LGrInt, NInt, result_arr);
std::cout << "Номер интервала Левая граница интервала Количество
псевдослучайных чисел, попавших в интервал\n";
file << "Номер интервала Левая граница интервала Количество
псевдослучайных чисел, попавших в интервал\n";
for (int i = 0; i < NInt; i++) {
    std::cout << i + 1 << "\t\t" << LGrInt[i] << "\t\t\t" << result_arr[i] <<
'\n';
    file << i + 1 << "\t\t" << LGrInt[i] << "\t\t\t" << result_arr[i] << '\n';
}

std::cout << '\n';
for (int i = 0; i < NumRanDat; i++) {
    std::cout << arr[i] << ' ';
}

delete[] LGrInt;
delete[] arr;
delete[] result_arr;
}

```

Название файла: lab6a.asm

.586p

.MODEL FLAT, C

.CODE

BORDER PROC C USES EDI ESI, array:dword, len:dword, LGrInt:dword,
NInt:dword, answer:dword

```
push eax
push ebx
push ecx
push edi
push esi
```

```
mov ecx, len
mov esi, array
mov edi, LGrInt
mov eax, 0
```

cycle:

```
mov ebx, 0
    search:
        cmp ebx, NInt
        jge outside

        push eax
        mov eax, [esi + 4 * eax]
        cmp eax, [edi + 4 * ebx]
        pop eax
        jl outside
        inc ebx
        jmp search
```

outside:

```
dec ebx

cmp ebx, -1
je number
mov edi, answer
push eax
mov eax, [edi + 4 * ebx]
inc eax
mov [edi + 4 * ebx], eax
pop eax
mov edi, LGrInt
```

number:

```
inc eax
```

loop cycle

pop esi

pop edi

pop ecx

pop ebx

pop eax

ret

BORDER ENDP

END