

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»

Студентка гр. 1381

Туркова Д.Н.

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 распределения 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 $[X_{min}, X_{max}]$, 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 $[X_{\min}, X_{\max}]$).

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).

Вариант работы №30 (для бригад с четным номером)

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 lab6.cpp:

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

using namespace std;
ofstream file;

extern "C" {void first(int* array, int lenArray, int Xmin, int* arrResalt);
            void second(int* array, int Xmin, int Xmax, int* LGrInt, int*
arr_out); }

void print_first(int Xmin, int Xmax, int* arr) {
    std::cout << endl << "Результат 1 модуля: " << std::endl;
    std::cout << "Интервал:\t" << "Значение:\t" << "Кол-во:" << std::endl;
    for (int i = Xmin, j = 0; i <= Xmax; i++, j++) {
        std::cout << "    "<<j + 1<<"\t"          " << i << "\t"          " <<
arr[j] << std::endl;
    }
}

void print_second(int NInt, int NumRanDat, int*& arr, int*& LGrInt, int*&
answer) {
    std::cout << "Результат 2 модуля:" << std::endl;
    file << std::endl;
    std::cout << "Номер\t" << "Левое значение\t" << "Кол-во" << std::endl;
    file << "Номер\t" << "Левое значение\t" << "Кол-во" << std::endl;
    for (int i = 0; i < NInt; i++) {
        std::cout << "    " << i + 1 << "\t"          " << LGrInt[i] << "\t
" << answer[i] << std::endl;
        file << "    " << i + 1 << "\t"          " << LGrInt[i] << "\t          " <<
answer[i] << std::endl;
    }
}

int main()
{
    int NumRanDat, Xmin, Xmax, NInt;
    setlocale(LC_ALL, "Russian");
    cout << "Введите длину массива: ";
    cin >> NumRanDat;
    while (NumRanDat <= 0 || NumRanDat > 1024 * 16)
    {
        cout << "Массив неправильного размера. Введите значение ещё раз:";
        cin >> NumRanDat;
    }

    cout << "Введите диапазон изменения массива псевдослучайных целых чисел."
<< endl;
    cout << "Минимум: ";
    cin >> Xmin;
    cout << "Максимум: ";
    cin >> Xmax;
    if (Xmax < Xmin)
    {
        swap(Xmin, Xmax);
        cout << "Неверный ввод. Значения поменяли местами. Теперь Xmin = "<<
Xmin << ", Xmax = " << Xmax;
```

```

}

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

cout << "Исходный массив: ";
for (int i = 0; i < NumRanDat; i++)
    cout << arr[i] << " ";

int* arrAnswer = new int[rangeLen] {0};
first(arr, NumRanDat, Xmin, arrAnswer);
print_first(Xmin, Xmax, arrAnswer);

cout << endl << "Введите количество интервалов, на которые разбивается
диапазон изменения массива псевдослучайных целых чисел: ";
cin >> NInt;
while (NInt <= 0 || NInt > 24 || NInt > rangeLen)
{
    cout << "Неверный ввод. Попробуй ещё раз: ";
    cin >> NInt;
}

int* LGrInt = new int[NInt + 1]; //массив левых границ для единичных
отрезков
LGrInt[0] = Xmin;
cout << "Введите левые границы интервалов разбиения: " << endl;
for (int i = 1; i < NInt; i++)
{
    std::cin >> LGrInt[i];
    while (LGrInt[i] > Xmax || LGrInt[i] < Xmin) {
        std::cout << "Неправильное значение, попробуйте ещё раз: ";
        std::cin >> LGrInt[i];
    }
}
LGrInt[NInt] = Xmax + 1;

int* resultArr = new int[NInt] {0};
second(arrAnswer, Xmin, Xmax, LGrInt, resultArr);
file.open("answer.txt", std::ios_base::out);
print_second(NInt, NumRanDat, arr, LGrInt, resultArr);
delete[] arr;
delete[] LGrInt;
delete[] resultArr;
delete[] arrAnswer;
file.close();

}

```

Название файла first.asm:

```
.586
.MODEL FLAT, C
.CODE
first PROC C arr:dword, NumRanDat:dword, Xmin:dword, answer_arr:dword
mov eax, arr
mov ebx, answer_arr
mov ecx, NumRanDat
xor edx, edx
xor edi, edi

finding:
    mov edi, [eax + 4 * edx]           ;значение
    sub edi, Xmin                     ;индекс
    inc dword ptr [ebx + 4 * edi]      ;увеличение счетчика числа
    inc edx                           ;следующее значение
    loop finding                      ;уменьшает счётчик длины

    ret
first endp
end
```

Название файла second.asm:

```
.586
.MODEL FLAT, C
.CODE
second PROC C arr:dword, Xmin:dword, Xmax:dword, LGrInt:dword, resultArr:dword
mov eax, Xmin
mov edi, resultArr
mov edx, LGrInt
mov esi, arr
xor ebx, ebx
xor ecx, ecx

finding:
    cmp eax, [edx + 4]                ;Xmin ? LGrInt
    jl adding                         ;1 < 2
    cmp eax, Xmax
    jg finish                         ;1>2
    cmp eax, Xmax
    je func                           ; =
    add edx, 4
    add edi, 4
    jmp finding

func:
    add edi, 4                        ;j++
    mov ebx, [esi + ecx * 4]          ;ebx = arr[k]
    add [edi], ebx                    ;arrResult[j] += arr[k]
    jmp finish

adding:
    mov ebx, [esi + ecx * 4]          ;ebx = arr[k]
    add [edi], ebx                    ;arrResult[j] += arr[k]
    inc ecx                           ;k++
    inc eax                           ;Xmin++
    jmp finding

finish:
    ret
second endp
end
```

Приложение Б. Тестирование.

Табл. №1: Тестирование программы lab6.asm

Длина массива псевдослучайных чисел	Диапазон изменения массива	Массив псевдослучайных чисел	Массив левых границ	Массив 2 модуля
5	[2; 7]	7 7 6 6 7	3 4 6	0 0 5
20	[-3; 1]	-2 -1 1 -3 1 1 0 0 -1 1 -3 -3 -2 -1 -2 -2 -3 -1 -1 -2	-3 -2	4 16
7	[11; 20]	12 18 15 11 20 15 19	11 13 16 18	2 2 0 3