**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 распределения попаданий псевдослучайных чисел в заданные интервалы.**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент гр.1303 | |  | Иевлев Е.А. |
| Преподаватель |  |  | Ефремов М.А. |

Санкт-Петербург

2022

**Цель работы.**

Рассмотреть способ организации связи Ассемблера с ЯВУ. Разработать

программу построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы.

**Задание.**

**Вариант 9**

На языке С программируется ввод с клавиатуры и контроль исходных данных, а также генерируется массив псевдослучайных целых чисел, изменяющихся в заданном диапазоне и имеющих заданный закон распределения.

Далее должны вызываться 1 или 2 ассемблерных процедуры для формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы. Ассемблерные процедуры должны вызываться как независимо скомпилированные модули. Передача параметров в процедуру должна выполняться через кадр стека.

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.

**Выполнение работы.**

Работа содержит два файла: файл на ЯВУ (main.cpp) и файл с ассемблерной вставкой (module1.asm).

В основном файле main.cpp происходит считывание входных данных с дальнейшей проверкой на корректность ввода и передачей в ассемблерный модуль. Массив псевдослучайных чисел генерируется при помощи стандартной библиотеки ctime и функции srand(). Массив answer[] по длине равен количеству интервалов и заполняется значениями-счётчиками для каждого интервала. Оба массива так же передаются в ассемблерный модуль.

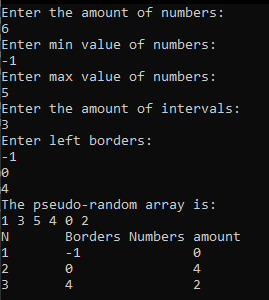
После обработки в ассемблерном модуле файлы передаются обратно в main.cpp и выводятся как в файл, так и в консоль.

Обработка чисел в ассемблерном модуле module1.asm:

1. В регистр ecx помещается смещение до очередного элемента сгенерированного массива чисел.
2. В массиве интервалов через цикл перебираются все границы и сравниваются с ecx
3. В случае попадания числа из ecx в интервал увеличиваем соответствующее значение-счётчик интервала в массиве answer. Иначе повторяем пункт 1 для следующего числа

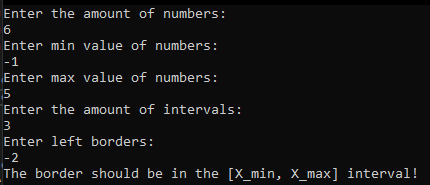
**Тестирование.**

Работа программы с заданными условиями представлена на рисунке 1.

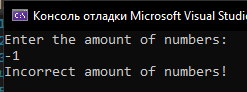


Риc. 1

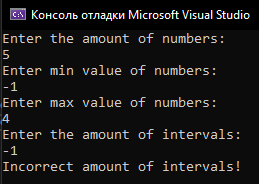
Работа программы при некорректных входных данных:



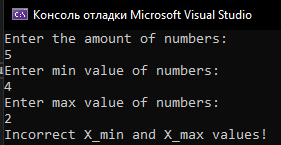
Риc. 2



Риc. 3



Риc. 4



Риc. 5

**Выводы.**

В ходе выполнения работы был рассмотрен способ организации связи между ЯВУ и ассемблером, разработана программа, осуществляющая построение частотного распределения попаданий псевдослучайных чисел с равномерным распределением в заданные интервалы.

**ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД ПРОГРАММ**

Название файла: *main.cpp*

#include <iostream>

#include <fstream>

#include <ctime>

#include <string>

using namespace std;

extern "C" void f(int\* inter, int\* num, int\* answer, int N\_int, int N);

int main() {

srand(static\_cast<unsigned int>(time(NULL)));

int N, N\_int, X\_min, X\_max;

cout << "Enter the amount of numbers:\n";

cin >> N;

if (N <= 0) {

cout << "Incorrect amount of numbers!\n";

return 0;

}

cout << "Enter min value of numbers:\n";

cin >> X\_min;

cout << "Enter max value of numbers:\n";

cin >> X\_max;

if (X\_min >= X\_max) {

cout << "Incorrect X\_min and X\_max values!\n";

return 0;

}

cout << "Enter the amount of intervals:\n";

cin >> N\_int;

if (N\_int <= 0 || N\_int > 24) {

cout << "Incorrect amount of intervals!\n";

return 0;

}

cout << "Enter left borders:" << endl;

auto inter = new int[N\_int + 1];

for (int i = 0; i < N\_int; i++) {

cin >> inter[i];

if (inter[i] < X\_min || inter[i] > X\_max) {

cout << "The border should be in the [X\_min, X\_max] interval!\n";

return 0;

}

}

auto num = new int[N];

int rand\_val = X\_max - X\_min + 1;

for (int i = 0; i < N; i++) {

num[i] = X\_min + rand() % rand\_val;

}

cout << "The pseudo-random array is:\n";

for (int j = 0; j < N; j++) {

cout << num[j] << ' ';

}

cout << endl;

auto answer = new int[N\_int];

for (int i = 0; i < N\_int; i++) {

answer[i] = 0;

}

f(inter, num, answer, N\_int, N);

ofstream file("out.txt");

auto str = "N\tBorders\tNumbers amount";

file << str << endl;

cout << str << endl;

for (int i = 0; i < N\_int; i++) {

auto str\_res = to\_string(i + 1) + "\t" + to\_string(inter[i]) + "\t\t" + to\_string(answer[i]) + "\n";

file << str\_res;

cout << str\_res;

}

return 0;

}

Название файла: *module1.asm*

.MODEL FLAT, C

.CODE

f PROC C inter: dword, num: dword, answer: dword, N\_int: dword, N: dword

push eax

push ebx

push ecx

push edi

push esi

mov eax, 0

mov esi, num

c\_loop:

mov ebx, 0

iter:

cmp ebx, N\_int

jge out\_cur\_iter

mov ecx, [esi + 4\*eax]

mov edi, inter

cmp ecx, [edi + 4\*ebx]

jl out\_cur\_iter

inc ebx

jmp iter

out\_cur\_iter:

dec ebx

mov edi, answer

mov ecx, [edi + 4\*ebx]

inc ecx

mov [edi + 4\*ebx], ecx

next\_number:

inc eax

cmp eax, N

jg exit

jmp c\_loop

exit:

pop edx

pop ecx

pop ebx

pop eax

pop edi

pop esi

ret

f ENDP

END