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.

Студент гр. 0383 _____ Коротков А.В.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Разработать программу построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы, связав модуль на Ассемблере с файлом на ЯВУ.

Задание.

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. Необходимые датчики псевдослучайных чисел находятся в каталоге RAND_GEN (при его отсутствии получить у преподавателя). Следует привести числа к целому виду с учетом диапазона изменения.

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.

Исходные данные:

1. Длина массива псевдослучайных целых чисел - NumRanDat (<= 16K)

2. Диапазон изменения массива псевдослучайных целых чисел [Xmin, Xmax] (может быть биполярным, например, [-50, 50])

3. Массив псевдослучайных целых чисел {Xi}.

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

5. Массив левых границ интервалов разбиения LGrInt.

В общем случае интервалы разбиения диапазона изменения псевдослучайных чисел могут иметь различную длину, левые границы могут задаваться в произвольном порядке и иметь произвольные значения. Если Xmin < LGrInt(1), то часть данных не будет участвовать в формировании распределения. Каждый интервал, кроме последнего, следует интерпретировать как [LGrInt(i), LGrInt(i+1)). Если у последнего интервала правая граница меньше Xmax, то часть данных не будет участвовать в формировании распределения.

Результаты:

Текстовая таблица, строка которой содержит:

- номер интервала,

- левую границу интервала,

- количество псевдослучайных чисел, попавших в интервал.

Количество строк должно быть равно числу интервалов разбиения.

Таблица должна выводиться на экран и сохраняться в файле.

Вариант работы — 9

Вид распределения — равномерное,

число процедур — 1,

Nint < Dx,

Lg1> Xmin,

ПГпосл > Xmax

Выполнение работы.

Реализовано считывание количества генерируемых чисел, граничных значений генерируемых чисел, количества интервалов разбиения и левых границ интервалов на языке C++. Случайные числа генерируются и заносятся в массив, левые границы интервалов заносятся в отдельный массив, создается результирующий массив, в который в дальнейшем по i-тому индексу будет заносится количество чисел, попавших в i-тый интервал.

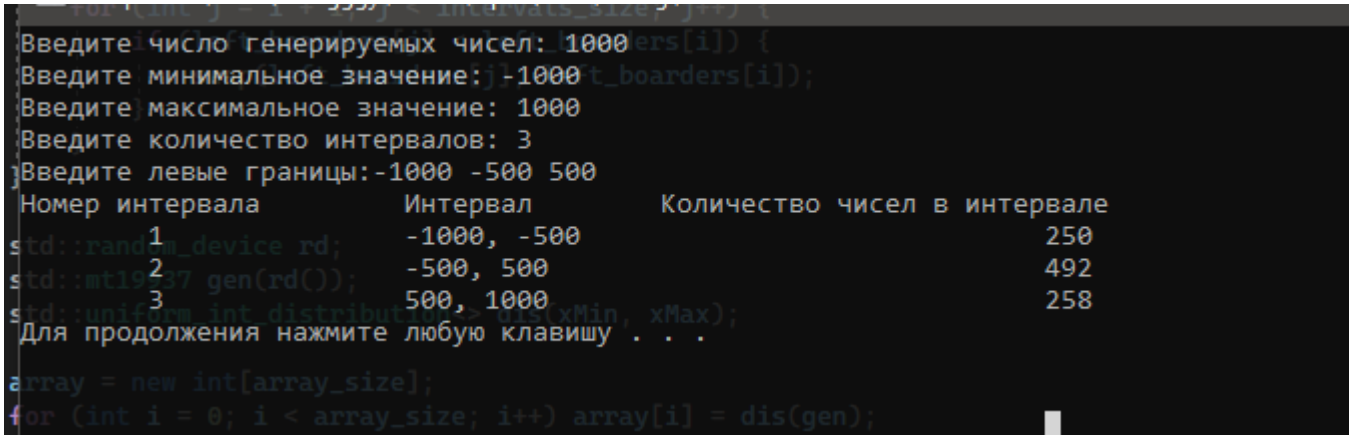
В ассемблерный модуль в процедуру FUNC передаются указатель на массив сгенерированных чисел, его размер, указатель на массив левых границ интервалов и его размер, указатель на результирующий массив. В процедуре совершается цикл по всем элементам массива сгенерированных чисел, для каждого находится интервал, в который оно попадает и в результирующем массиве инкрементируется соответствующий элемент.

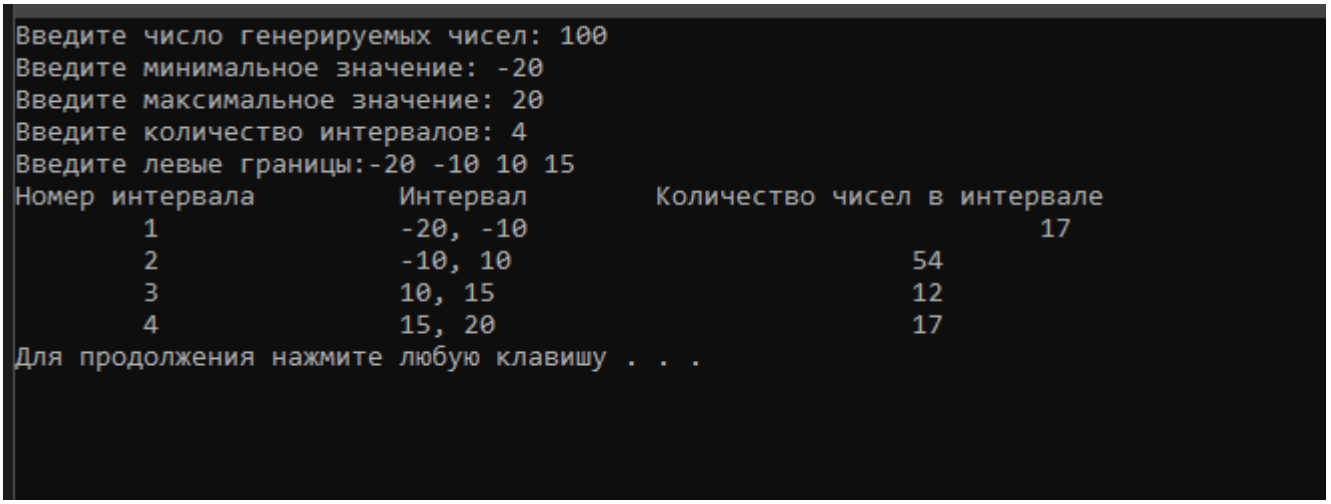
После того, как процедура из ассемблерного модуля завершила работу, на экран и в файл out.txt выводится текстовая таблица, содержащая номера интервалов, их левые границы и количество чисел, попавших в каждый интервал.

Тексты исходного файла программы см. в приложении А.

Тексты диагностических сообщений см. в приложении Б.

Тестирование.





Выводы.

В ходе выполнения данной лабораторной работы была изучена организация связи кода на ассемблере с ЯВУ. Была реализована программа частотного распределения случайных чисел по заданным интервалам на языке C++ с использованием ассемблерного модуля.

ПРИЛОЖЕНИЕ А

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММЫ

Название файла: **lb6.cpp**

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

extern "C" void FUNC(int* array, int array_size, int* left_borders, int
intervals_size, int* result_array);

using namespace std;
int main() {
	setlocale(0, "");

	std::ofstream file("out.txt");
	int xMin, xMax;
	int array_size;
	int intervals_size;
	int* array;
	int* left_boarders;
	int* result_array;

	cout << "Введите число генерируемых чисел: ";
	cin >> array_size;
	cout << "Введите минимальное значение: ";
	cin >> xMin;
	cout << "Введите максимальное значение: ";
	cin >> xMax;
	if (xMax < xMin) {
		cout << "Неверно введены максимальное и минимальное значения";
		return 0;
	}
	cout << "Введите количество интервалов: ";
	cin >> intervals_size;
```

```
if (intervals_size <= 0) {
	cout << "Неверно введено количество интервалов";
	return 0;
}

left_boarders = new int[intervals_size + 1];
cout << "Введите левые границы:";
for (int i = 0; i < intervals_size; i++)
	cin >> left_boarders[i];

for (int i = 0; i < intervals_size - 1; i++) {
	for (int j = i + 1; j < intervals_size; j++) {
		if (left_boarders[j] < left_boarders[i]) {
			swap(left_boarders[j], left_boarders[i]);
		}
	}
}

std::random_device rd;
std::mt19937 gen(rd());
std::uniform_int_distribution<> dis(xMin, xMax);

array = new int[array_size];
for (int i = 0; i < array_size; i++) array[i] = dis(gen);

file << "Сгенерированные числа: ";
for (int i = 0; i < array_size; i++) {
	file << array[i] << ' ';
	cout << array[i] << ' ';
}
file << '\n';
cout << '\n';

result_array = new int[intervals_size];
```

```
	for (int i = 0; i < intervals_size; i++)
		result_array[i] = 0;

	FUNC(array, array_size, left_boarders, intervals_size, result_array);

	cout << "Номер интервала \tИнтервал \tКоличество чисел в интервале" << '\n';
	file << "Номер интервала \tИнтервал \tКоличество чисел в интервале" << '\n';
	left_boarders[intervals_size] = xMax;
	for (int i = 1; i <= intervals_size; i++) {
		cout << "\t" << i << "\t\t" << left_boarders[i-1] << ", " << left_boarders[i] << "\t\t\t\t" << result_array[i-1] << '\n';
		file << "\t" << i << "\t\t" << left_boarders[i-1] << ", " << left_boarders[i] << "\t\t\t\t" << result_array[i-1] << '\n';
	}

	system("pause");
	return 0;
}
```

Название файла: **module.asm**

```
.586
.MODEL FLAT, C
.CODE
FUNC PROC C array:dword, array_size:dword, left_boarders:dword, intervals_size:dword, result_array:dword
; сохранение регистров
push ecx
push esi
push edi
push eax
push ebx

mov ecx, array_size
```

```
mov esi, array
mov edi, left_boarders
mov eax, 0
l1:
     mov ebx, 0
     boarders:
          cmp ebx, intervals_size
          jge boarders_exit
          push eax
          mov eax, [esi+4*eax]
          cmp eax, [edi+4*ebx]
          pop eax
          jl boarders_exit
          inc ebx
          jmp boarders
     boarders_exit:
     dec ebx

     cmp ebx, -1
     je skip
     mov edi, result_array
     push eax
     mov eax, [edi+4*ebx]
     inc eax
     mov [edi+4*ebx], eax
     pop eax
     mov edi, left_boarders
     skip:
     inc eax
loop l1

pop ebx
pop eax
pop edi
```

```
pop esi
pop ecx
ret
FUNC ENDP
END
```