

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

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. 9383

Арутюнян С.Н.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Содержание

1. Цель работы.....	3
2. Задание. Бригада с нечетным номером.....	3
3. Тестирование.....	4
4. Текст программы main.cpp.....	5
5. Текст программы func.asm.....	7
Выводы.....	9

1. Цель работы

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.

2. Задание. Бригада с нечетным номером

Необходимо реализовать подпрограмму формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы реализуется в виде одного ассемблерного модуля, сразу формирующего требуемое распределение и возвращающего его в головную программу, написанную на ЯВУ.

3. Тестирование

```
Количество чисел: 10
Левая граница чисел: -10
Правая граница чисел: 10
Количество интервалов: 3
Левая граница #1: -5
Левая граница #2: 5
Левая граница #3: 7
Массив случайных чисел:
-9 -8 -7 -5 -4 2 8 9 10 10
Граница [-5, 5]: 3
Граница (5, 7]: 0
Граница (7, 10]: 4
```

Рисунок 1 - Пример работы программы

```
Количество чисел: 15
Левая граница чисел: 1
Правая граница чисел: 60
Количество интервалов: 5
Левая граница #1: 2
Левая граница #2: 3
Левая граница #3: 4
Левая граница #4: 5
Левая граница #5: 50
Массив случайных чисел:
7 12 13 18 19 21 23 31 31 37 44 48 51 53 57
Граница [2, 3]: 0
Граница (3, 4]: 0
Граница (4, 5]: 0
Граница (5, 50]: 12
Граница (50, 60]: 3
```

Рисунок 2 - Пример работы программы

4. Текст программы main.cpp

```
#include <iostream>
#include <string>
#include <algorithm>
#include <random>
#include <time.h>
#include <locale.h>

using namespace std;

extern "C" void func(int* result, int* LGrInt, int* randoms, int
NumRanDat);

template <typename T>
void ReadWithWelcome(T& var, const std::string& message) {
    std::cout << message; std::cin >> var;
}

void main() {
    setlocale(LC_ALL, "RUS");
    srand(time(nullptr));

    int NumRanDat;
    ReadWithWelcome(NumRanDat, "Количество чисел: ");

    int Xmin, Xmax;
    ReadWithWelcome(Xmin, "Левая граница чисел: ");
    ReadWithWelcome(Xmax, "Правая граница чисел: ");

    int NInt;
    ReadWithWelcome(NInt, "Количество интервалов: ");

    auto LGrInt = new int[NInt + 1];
    for (int i = 0; i < NInt; ++i) {
        ReadWithWelcome(LGrInt[i], "Левая граница #" +
```

```

std::to_string(i + 1) + ": ");
    }
    LGrInt[NInt] = Xmax;
    std::sort(LGrInt, LGrInt + NInt);

    auto randomness = new int[NumRandat];
    std::for_each(
        randomness, randomness + NumRandat,
        [Xmin, Xmax](int& element) {
            element = Xmin + rand() % (Xmax - Xmin + 1);
        }
    );
    std::sort(randomness, randomness + NumRandat);

    auto result = new int[NInt];
    std::fill(result, result + NInt, 0);

    std::cout << "Массив рандомных чисел:" << std::endl;
    for (int i = 0; i < NumRandat; ++i) {
        std::cout << randomness[i] << " ";
    } std::cout << std::endl;

    func(result, LGrInt, randomness, NumRandat);

    for (int i = 0; i < NInt; ++i) {
        if (i == 0)
            std::cout << "Граница [";
        else
            std::cout << "Граница (";

        std::cout << std::to_string(LGrInt[i]) << ", "
            << std::to_string(LGrInt[i + 1]) << "]: "
            << result[i] << std::endl;
    }
}

```

```
}
```

5. Текст программы func.asm

```
.model flat,c
```

```
.code
```

```
func proc C
```

```
    mov edi, [esp + 4]      ; result  
    mov ebx, [esp + 8]      ; LGrInt  
    mov esi, [esp + 12]     ; randoms  
    mov ecx, [esp + 16]     ; NumRanDat
```

```
random_array_loop:
```

```
    ; достаем очередное число из randoms
```

```
    mov eax, [esi]
```

```
    add esi, 4
```

```
    cmp eax, [ebx]
```

```
    ; если eax меньше текущей левой границы, то текущее число не
```

```
подходит
```

```
    jl loop_end
```

```
continue:
```

```
    ; здесь мы точно знаем, что eax >= [ebx]
```

```
    cmp eax, [ebx+4]
```

```
    ; если eax > [ebx+1], то нигде больше не понадобится [ebx] и [edi]
```

```
    jg eax_is_more_than_ebx_plus_4
```

```
    ; если eax <= [ebx+4], то меняем results
```

```
    mov eax, [edi]
```

```
    add eax, 1
```

```
    mov [edi], eax
```

```
    jmp loop_end
```

```
eax_is_more_than_ebx_plus_4:
```

```
    add ebx, 4
    add edi, 4
    jmp continue ; нужно вернуться и продолжать работать с текущим
числом в eax
```

```
loop_end:
    dec ecx
    cmp ecx, 0
    jne random_array_loop
```

```
ret
```

```
func endp
end
```


Выводы

В процессе выполнения лабораторной работы была организована связь Ассемблера с ЯВУ на примере программы построения частотного распределение попаданий псевдослучайных целых чисел в заданные интервалы.