

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №6
по дисциплине «ОЭВМ»
Тема: Организация связи Ассемблера с ЯВУ на примере программы
построения частотного распределение попаданий псевдослучайных целых
чисел в заданные интервалы.

Студент гр. 9383

Крейсманн К.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы

Изучение связи Ассемблера с ЯВУ на примере программы построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы.

Задача

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; 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:

Текстовый файл, строка которого содержит: - номер интервала, - левую границу интервала, - количество псевдослучайных чисел, попавших в интервал. Количество строк равно числу интервалов разбиения.

Выполнение работы

1. Происходит инициализация всех данных в C++.
2. Создаем массив псевдослучайных чисел.
3. Сортируем массив левых границ по убыванию.
4. Вызываем ассемблерный модуль, в который передаем все данные.
5. В ассемблерном модуле проходимся по каждому элементу из массива псевдослучайных чисел, и сравниваем его с левыми границами. Если элемент больше либо равен левой границы, он попадает в интервал и соответствующий элемент в массиве result увеличивается на 1.
6. В модуле C++ записываем результат в файл "res.txt".

Тестирование:

№	Исходные данные	Результат
1	NumRanDat = 30 Xmin = 1 Xmax = 20 NInt = 5 Левые границы: 1, 5, 7, 10, 19	1: Левая граница 19; Количество чисел, попавших в интервал: 1 2: Левая граница 10; Количество чисел, попавших в интервал: 16 3: Левая граница 7; Количество чисел, попавших в интервал: 7 4: Левая граница 5; Количество чисел, попавших в интервал: 2 5: Левая граница 1; Количество чисел, попавших в интервал: 4
2	NumRanDat = 100 Xmin = -10 Xmax = 10 NInt = 5 Левые границы: -10,-5,0,5,9	1: Левая граница 9; Количество чисел, попавших в интервал: 3 2: Левая граница 5; Количество чисел, попавших в интервал: 21 3: Левая граница 0; Количество чисел, попавших в интервал: 24 4: Левая граница -5; Количество чисел, попавших в интервал: 30 5: Левая граница -10; Количество чисел, попавших в интервал: 22

3	NumRanDat = 100 Xmin = 0 Xmax = 100 NInt = 10 Левые границы: 0,10,20,30,40,50,60,70,80,90	1: Левая граница 90; Количество чисел, попавших в интервал: 12 2: Левая граница 80; Количество чисел, попавших в интервал: 10 3: Левая граница 70; Количество чисел, попавших в интервал: 5 4: Левая граница 60; Количество чисел, попавших в интервал: 11 5: Левая граница 50; Количество чисел, попавших в интервал: 16 6: Левая граница 40; Количество чисел, попавших в интервал: 11 7: Левая граница 30; Количество чисел, попавших в интервал: 6 8: Левая граница 20; Количество чисел, попавших в интервал: 10 9: Левая граница 10; Количество чисел, попавших в интервал: 9 10: Левая граница 0; Количество чисел, попавших в интервал: 10
4	NumRanDat = 16000 Xmin = 0 Xmax = 400 NInt = 24 Левые границы: 0,10,25,53,54,55,56,90,101,105,201,204,213,244,295,300,303,313,341,342,343,344,360,399	1: Левая граница 399; Количество чисел, попавших в интервал: 38 2: Левая граница 360; Количество чисел, попавших в интервал: 1440 3: Левая граница 344; Количество чисел, попавших в интервал: 651 4: Левая граница 343; Количество чисел, попавших в интервал: 43 5: Левая граница 342; Количество чисел, попавших в интервал: 47 6: Левая граница 341; Количество чисел, попавших в интервал: 34 7: Левая граница 313; Количество чисел, попавших в интервал: 1124 8: Левая граница 303; Количество чисел, попавших в интервал: 398 9: Левая граница 300; Количество чисел, попавших в интервал: 105 10: Левая граница 295; Количество чисел, попавших в интервал: 193 11: Левая граница 244; Количество чисел, попавших в интервал: 2098

		<p>12: Левая граница 213; Количество чисел, попавших в интервал: 1327</p> <p>13: Левая граница 204; Количество чисел, попавших в интервал: 362</p> <p>14: Левая граница 201; Количество чисел, попавших в интервал: 123</p> <p>15: Левая граница 105; Количество чисел, попавших в интервал: 3767</p> <p>16: Левая граница 101; Количество чисел, попавших в интервал: 175</p> <p>17: Левая граница 90; Количество чисел, попавших в интервал: 409</p> <p>18: Левая граница 56; Количество чисел, попавших в интервал: 1418</p> <p>19: Левая граница 55; Количество чисел, попавших в интервал: 27</p> <p>20: Левая граница 54; Количество чисел, попавших в интервал: 38</p> <p>21: Левая граница 53; Количество чисел, попавших в интервал: 37</p> <p>22: Левая граница 25; Количество чисел, попавших в интервал: 1129</p> <p>23: Левая граница 10; Количество чисел, попавших в интервал: 615</p> <p>24: Левая граница 0; Количество чисел, попавших в интервал: 402</p>
5	<p>NumRanDat = 20</p> <p>Xmin = 0</p> <p>Xmax = 20</p> <p>NInt = 20</p> <p>Левые границы:</p> <p>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19</p>	<p>1: Левая граница 19; Количество чисел, попавших в интервал: 0</p> <p>2: Левая граница 18; Количество чисел, попавших в интервал: 1</p> <p>3: Левая граница 17; Количество чисел, попавших в интервал: 1</p> <p>4: Левая граница 16; Количество чисел, попавших в интервал: 2</p> <p>5: Левая граница 15; Количество чисел, попавших в интервал: 0</p> <p>6: Левая граница 14; Количество чисел, попавших в интервал: 0</p> <p>7: Левая граница 13; Количество чисел, попавших в интервал: 0</p> <p>8: Левая граница 12; Количество чисел, попавших в интервал: 3</p>

		<p>9: Левая граница 11; Количество чисел, попавших в интервал: 1</p> <p>10: Левая граница 10; Количество чисел, попавших в интервал: 1</p> <p>11: Левая граница 9; Количество чисел, попавших в интервал: 0</p> <p>12: Левая граница 8; Количество чисел, попавших в интервал: 0</p> <p>13: Левая граница 7; Количество чисел, попавших в интервал: 1</p> <p>14: Левая граница 6; Количество чисел, попавших в интервал: 3</p> <p>15: Левая граница 5; Количество чисел, попавших в интервал: 0</p> <p>16: Левая граница 4; Количество чисел, попавших в интервал: 1</p> <p>17: Левая граница 3; Количество чисел, попавших в интервал: 2</p> <p>18: Левая граница 2; Количество чисел, попавших в интервал: 1</p> <p>19: Левая граница 1; Количество чисел, попавших в интервал: 1</p> <p>20: Левая граница 0; Количество чисел, попавших в интервал: 2</p>
--	--	--

Выводы.

Получены навыки использования связи Ассемблера с ЯВУ на примере программы построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы

Приложение А

Исходный код программы

Файл main.cpp

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

extern "C"
{
    void funcMasm(short* result, short* LGrInt, short *randArray, short NInt, short NumRanDat);
}

int main()
{
    setlocale(LC_ALL, "rus");
    srand(time(0));

    short NumRanDat; //длина массива псевдослучайных чисел
    short Xmin, Xmax; //Границы диапазона
    short NInt; //количество интервалов
    short* LGrInt; //массив левых границ интервалов разбиения
    short* randArray; //массив псевдослучайных чисел
    short* result; //массив с результатом
    std::ofstream file("res.txt"); //файл с результатом

    std::cout << "Длина массива псевдослучайных чисел (<=16000):";
    std::cin >> NumRanDat;

    std::cout << "\nXmin u Xmax:";
    std::cin >> Xmin >> Xmax;
```

```
std::cout << "\nКоличество интервалов (<=24):";
```

```
std::cin >> NInt;
```

```
std::cout << "Левые границы интервалов:(>=Xmin = " << Xmin  
<< ", <Xmax = " << Xmax << "):" << "\n";
```

```
LGrInt = new short[NInt];
```

```
for (int i = 0; i < NInt; i++)
```

```
{
```

```
std::cout << i + 1 << ":";
```

```
std::cin >> LGrInt[i];
```

```
std::cout << '\n';
```

```
}
```

```
for (int i = 0; i < NInt; i++)
```

```
{
```

```
for (int j = 0; j < NInt - i - 1; j++)
```

```
{
```

```
if (LGrInt[j] < LGrInt[j + 1])
```

```
{
```

```
int temp = LGrInt[j];
```

```
LGrInt[j] = LGrInt[j + 1];
```

```
LGrInt[j + 1] = temp;
```

```
}
```

```
}
```

```
}
```

```
randArray = new short[NumRandat];
```

```
//генерация псевдослучайных чисел
```

```
for (int i = 0; i < NumRandat; i++)
```

```
{
```

```
randArray[i] = Xmin + rand()%(Xmax - Xmin);
```

```
}
```



```

    result = new short[NInt];
    for (int i = 0; i < NInt; i++)
    {
        result[i] = 0;
    }

    funcMasm(result, LGrInt, randArray, NInt, NumRanDat);

    for (int i = 0; i < NInt; i++)
    {
        file << i+1 << ":" << " Левая граница " << LGrInt[i] << "; Количество чисел, попавших в
интервал: " << result[i] << '\n';
    }

    /*for (int i = 0; i < NumRanDat; i++)
    {
        std::cout << randArray[i] << ' ';
    }
    std::cout << '\n';
    for (int i = 0; i < NInt; i++)
    {
        std::cout << result[i] << ' ';
    }
    std::cout << '\n';*/

    delete[] randArray;
    delete[] LGrInt;
    delete[] result;
    file.close();

    return 0;
}

```

Файл `assem.asm`

`.Model flat,c`

`.data`

`tempI dw 0 ;для элементов массива randArray`

`.code`

`funcMasm proc C`

`mov eax,[esp+4] ; result в eax`

`mov ebx,[esp+8] ; LGrInt в ebx`

`mov edx,[esp+12]; randArray в edx`

`mov di, [esp+16] ; NInt в di`

`mov cx, [esp+20] ; NumRanDat в cx`

`and ecx,0000ffffh`

`and edi,0000ffffh`

`metka1:`

`push ax`

`mov ax,[edx]`

`mov tempI,ax ;в tempI - текущее число из randArray`

`pop ax`

`mov esi,0 ; индекс для массивов LGrInt и result`

`push ecx ; сохраняем счетчик для внешнего цикла`

`mov ecx,edi ; в ecx - счетчик для внутреннего цикла`

`metka2:`

`push ebx`

`mov bx,[ebx+esi] ;в bx - текущий элемент из LGrInt`

`cmp tempI,bx ;сравняем очередную левую границу с очередным случайным числом`

`jge metka3 ;если число больше границы, то оно входит в интервал`

`add esi,2 ;прибавляем к индексу массива LGrInt 2`

`pop ebx`

`loop metka2 ;продвигаемся дальше по внутреннему циклу`

`metka3:`

```

mov bx, [eax+esi] ;получаем элемент из result
inc bx            ;увеличиваем его на 1
mov [eax+esi],bx ;зааносим обратно в result
pop ebx
pop ecx           ;в ecx счетчик для внешнего цикла
add edx,2         ;прибавляем к индексу массива randArray 2
loop metka1

ret
funcMasm endp
end

```