

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

Студент гр. 9383

Корсунов А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Научиться связывать программу на ЯВУ с ассемблерными модулями.
Написать программу на основе изученного.

Текст задания.

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: 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) В зависимости от номера бригады формирование частотного распределения должно производиться по одному из двух вариантов:

2. Для бригад с четным номером: подпрограмма формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы реализуется в виде двух ассемблерных модулей, первый из которых формирует распределение исходных чисел по интервалам единичной длины и возвращает его в вызывающую программу на ЯВУ как промежуточный результат. Это распределение должно выводиться в текстовом виде для контроля. Затем вызывается второй ассемблерный модуль, который по этому промежуточному распределению формирует окончательное распределение псевдослучайных целых чисел по интервалам произвольной длины (с заданными границами). Это распределение возвращается в головную программу и выдается как основной результат в виде текстового файла и, возможно, графика.

Ход работы:

Для реализации поставленной задачи была написана программа, состоящая из трех файлов:

`main.cpp` — в функции `main` происходит генерация псевдослучайных целых чисел, их обработка с помощью вызова ассемблерных модулей и вывод результатов в консоль и текстовый файл;

`mod1.asm` — первый ассемблерный модуль, в котором происходит заполнение массива `array_mod1` вхождений каждого сгенерированного числа по интервалам единичной длины;

`mod2.asm` — второй ассемблерный модуль, в котором происходит заполнение массива `array_mod2` на количество вхождений каждого элемента в интервалы, определенные пользователем;

Тестирование:

```
Введите длину массива псевдослучайных целых чисел
(Длина массива не может быть больше 16*1024 и меньше 0)
10

Введите диапазон изменения массива (минимальное и максимальное значения
(если максимальное значение будет меньше минимального, то значения поменяются местами)
-5 5

Введите количество интервалов, на которые разбивается диапазон изменения массива
Количество интервалов должно быть больше 0, но меньше 24
3

Введите левые границы:
-2 1 3

Псевдослучайные целые числа: 3 4 4 -4 2 0 0 5 -4 -5

Распределение по единичным интервалам:
-5 -4 -3 -2 -1 0 1 2 3 4 5
1 2 0 0 0 2 0 1 1 2 1
```

Номер интервала	Левая граница интервала	Количество чисел в интервале
0	-5	3
1	-2	2
2	1	1
3	3	4

Рисунок 1 - Пример работы программы

```
Введите длину массива псевдослучайных целых чисел
(Длина массива не может быть больше 16*1024 и меньше 0)
5

Введите диапазон изменения массива (минимальное и максимальное значения
(если максимальное значение будет меньше минимального, то значения поменяются местами)
-6 4

Введите количество интервалов, на которые разбивается диапазон изменения массива
Количество интервалов должно быть больше 0, но меньше 24
2

Введите левые границы:
-1 3

Псевдослучайные целые числа: 2 3 3 -5 1

Распределение по единичным интервалам:
-6 -5 -4 -3 -2 -1 0 1 2 3 4
0 1 0 0 0 0 0 1 1 2 0
```

Номер интервала	Левая граница интервала	Количество чисел в интервале
0	-6	1
1	-1	2
2	3	2

C:\Users\Eldorado\source\repos\main\Debug\lab6.exe (процесс 9356) завершил работу с кодом 0.

Рисунок 2 — Пример работы программы

```

Введите длину массива псевдослучайных целых чисел
(Длина массива не может быть больше 16*1024 и меньше 0)
21

Введите диапазон изменения массива (минимальное и максимальное значения
(если максимальное значение будет меньше минимального, то значения поменяются местами)
0 3

Введите количество интервалов, на которые разбивается диапазон изменения массива
Количество интервалов должно быть больше 0, но меньше 24
3

Введите левые границы:
1 2 3

Псевдослучайные целые числа: 1 3 2 0 1 0 2 2 2 0 1 1 1 3 1 3 3 2 3 0 3

Распределение по единичным интервалам:
0 1 2 3
4 6 5 6

```

Номер интервала	Левая граница интервала	Количество чисел в интервале
0	0	4
1	1	6
2	2	5
3	3	6

Рисунок 3 — Пример работы программы

Вывод:

Произошло ознакомление со связыванием программы на ЯВУ с ассемблерными модулями. Написана программа на основе изученного.

Приложение А

```
main.cpp:
#include <iostream>
#include <random>
#include <fstream>

using namespace std;

extern "C"
{
    void mod1(int* array_mod1, int* NumArray, int NumRanDat, int Xmin);
    void mod2(int* array_mod1, int NumRanDat, int* LGrInt, int NInt, int Xmin,
int Xmax, int* array_mod2);
}

int main()
{
    setlocale(0, "");
    int NumRanDat;
    int Xmin, Xmax;
    int    NInt;
    ofstream open_file;
    open_file.open("opened_file.txt");
    cout << "Введите длину массива псевдослучайных целых чисел\n";
    cout << "(Длина массива не может быть больше 16*1024 и меньше 0)\n";
    cin >> NumRanDat;
    if (NumRanDat > 16 * 1024 || NumRanDat < 0)
    {
        while (NumRanDat > 16 * 1024 || NumRanDat < 0)
        {
```

```

        cout << "Вы ввели неправильную длину массива, попробуйте
еще раз\n";

        cin >> NumRanDat;

    }

}

cout << "\nВведите диапазон изменения массива (минимальное и
максимальное значения\n";

    cout << "(если максимальное значение будет меньше минимального, то
значения поменяются местами)\n";

    cin >> Xmin;
    cin >> Xmax;
    if (Xmax < Xmin)
    {
        swap(Xmax, Xmin);
    }

    cout << "\nВведите количество интервалов, на которые разбивается
диапазон изменения массива\n";

    cout << "Количество интервалов должно быть больше 0, но меньше 24\n";
    cin >> NInt;
    if (NInt < 0 || NInt > 24)
    {
        while (NInt > 24 || NInt < 0)
        {
            cout << "Вы ввели неправильное количество интервалов,
попробуйте еще раз\n";
            cin >> NInt;
        }
    }

    int* NumArray = new int[NumRanDat];
    int* LGrInt = new int[NInt];

```

```

int* array_mod1 = new int[abs(Xmax - Xmin) + 1];
int* array_mod2 = new int[NInt + 1];
cout << "\nВведите левые границы:\n";
for (int i = 0; i < NInt; i++)
{
    cin >> LGrInt[i];
    if (LGrInt[i] < Xmin || LGrInt[i] > Xmax)
    {
        while (LGrInt[i] < Xmin || LGrInt[i] > Xmax)
        {
            cout << "Вы ввели неправильную левую границу,
попробуйте еще раз\n";
            cin >> LGrInt[i];
        }
    }
}

for (int i = 0; i < NumRanDat; i++)
{
    NumArray[i] = rand() % (Xmax - Xmin + 1) + Xmin;
}
for (int i = 0; i < abs(Xmax - Xmin) + 1; i++)
{
    array_mod1[i] = 0;
}
mod1(array_mod1, NumArray, NumRanDat, Xmin);
cout << "\n";
for (int i = 0; i < NInt + 1; i++)
{
    array_mod2[i] = 0;
}

```



```

}
mod2(array_mod1, NumRanDat, LGrInt, NInt, Xmin, Xmax, array_mod2);
cout << "\nПсевдослучайные целые числа: ";
open_file << "\nПсевдослучайные целые числа: ";
for (int i = 0; i < NumRanDat; i++)
{
    cout << NumArray[i] << " ";
    open_file << NumArray[i] << " ";
}
cout << "\n";
open_file << "\n";
cout << "\nРаспределение по единичным интервалам: ";
open_file << "\nРаспределение по единичным интервалам: ";
cout << "\n";
open_file << "\n";
for (int i = 0; i < abs(Xmax - Xmin) + 1; i++)
{
    cout << Xmin + i << " ";
    open_file << Xmin + i << " ";
}
cout << "\n";
open_file << "\n";
for (int i = 0; i < abs(Xmax - Xmin) + 1; i++)
{
    cout << array_mod1[i] << " ";
    open_file << array_mod1[i] << " ";
}
cout << "\n";
open_file << "\n";

```

```

        cout << " Номер интервала \t Левая граница интервала \t Количесство чисел
в интервале\n";

        open_file << "Номер интервала \t Левая граница интервала \t Количество
чисел в интервале\n";

        cout << " " << 0 << " \t\t\t " << Xmin << " \t\t\t " << array_mod2[0] <<
"\n";

        open_file << " " << 0 << " \t\t\t " << Xmin << " \t\t\t " << array_mod2[0] <<
"\n";

        for (int i = 1; i < NInt+1; i++)
        {
            cout << " " << i << " \t\t\t " << LGrInt[i - 1] + Xmin << " \t\t\t " <<
array_mod2[i] << "\n";

            open_file << " " << i << " \t\t\t " << LGrInt[i - 1] + Xmin << " \t\t\t "
<< array_mod2[i] << "\n";

        }

        delete[] array_mod1;
        delete[] array_mod2;
        delete[] LGrInt;
        delete[] NumArray;

        return 0;
    }

```

Приложение В

Файл mod1.asm:

```
.586p
.MODEL FLAT, C
.CODE
PUBLIC C mod1
mod1 PROC C array_mod1: DWORD, NumArray: DWORD, NumRanDat:
DWORD, Xmin: DWORD

    push esi
    push edi

    mov edi, NumArray ;массив чисел
    mov ecx, NumRanDat ;длина этого массива
    mov esi, array_mod1 ;массив на выход

    for_NumRanDat:
        mov eax, [edi] ;кладем элемент массива чисел
        sub eax, Xmin ;получаем индекс выходного массива
        mov ebx, [esi + 4*eax] ;кладем значение по этому индексу из выходного
массива
        inc ebx ;++
        mov [esi + 4*eax], ebx ;кладем обратно
        add edi, 4 ;следующий элемент
        loop for_NumRanDat

    pop edi
    pop esi

    ret
```

```
mod1 ENDP  
END
```

Файл mod2.asm:

```
.586p  
.MODEL FLAT, C  
.CODE  
PUBLIC C mod2  
mod2 PROC C array_mod1: DWORD, NumRanDat: DWORD, LGrInt: DWORD,  
NInt: DWORD, Xmin: DWORD, Xmax: DWORD, array_mod2:DWORD
```

```
mov esi, LGrInt ; массив левых границ  
mov ecx, NInt ; количество границ  
mov edx, Xmin ; минимальное значение
```

```
for_LGrInt: ; получаем массив индексов для элементов массива интервалов  
mov eax, [esi]  
sub eax, xmin  
mov [esi], eax  
add esi, 4  
loop for_LGrInt
```

```
mov esi, LGrInt ;  
mov ecx, NInt  
mov ebx, 0  
mov edi, array_mod1  
mov eax, [esi] ; элемент массива левых интервалов
```

```
for2i:  
push ecx
```

mov ecx, eax ; число элементов массива левых интервалов

push esi

mov esi, array_mod2

for2j:

mov eax, [edi] ; элемент массива array_mod1

add [esi + ebx*4], eax

add edi, 4

loop for2j

pop esi

mov eax, [esi]

add esi, 4

sub eax, [esi]

neg eax

inc ebx ;для обращения к следующему элементу

pop ecx

loop for2i

mov esi, array_mod2

mov ecx, NInt

mov eax, 0

for3: ;для посчета последнего интервала

add eax, [esi] ;

add esi, 4

loop for3

sub eax, NumRanDat ;

```
neg eax      ;
```

```
mov esi, array_mod2
```

```
add [esi + 4 * ebx], eax ;
```

```
ret
```

```
mod2 ENDP
```

```
END
```