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 построения частотного
распределение попаданий псевдослучайных целых чисел в заданные интервалы.

Студентка гр. 9383 _____ Пономаренко С. А.

Преподаватель _____ Ефремов М. А.

Санкт-Петербург
2020

Цель работы.

Изучить способ связи Ассемблера с ЯВУ (Си++), написать программу построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы и скомпилировать файл.

Задание.

Вариант 15

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 целыхчисел - NumRanDat (<= 16K, K=1024)

2. Диапазон изменения массива псевдослучайных целых чисел [Xmin, Xmax] , значения могут быть биполярные;

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

4. Массив левых границ интервалов разбиения LGrInt (должны принадлежать интервалу [Xmin, Xmax]).

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).

Ход работы.

Файл lab6.cpp

Создаются переменные, выделяется память под массивы. Пользователь вводит все необходимые входные данные. Генерируются псевдослучайные числа в массиве array. Вызывается ассемблерный модуль, в который передаются массив рандомных чисел, заполенный нулями результирующий массив, в котором будут храниться количества вхождений в каждый интервал, массив левых границ, количество интервалов, количество элементов массива рандомных

чисел. После выполнения ассемблерного модуля результаты выводятся на экран и записываются в файл.

Файл lab6.asm

В регистры заносятся все переданные в модуль параметры. Функция 1 рассматривает элемент массива рандомных чисел. Функция 2 (внутренний цикл) сравнивает элемент массива рандомных чисел с элементом массива левых границ. Если первый больше или равен второму, переход к функции 3. Если нет, увеличивается индекс массива массива левых границ. Создается цикл с помощью команды loop. Функция 3 увеличивает на 1 счетчик количества элементов, попавших в интервал, определяя необходимое смещение. Создается цикл с помощью команды loop.

Тестирование.

Входные данные:	Исходные данные:
NumRanDat = 10 Xmin = -10 Xmax = 10 Nint = 2 LGrInt = {-10, 0} array = {3, 5, -10, 2, 9, 9, 0, -3, -4, 8}	Left border: 0 ___ Amount of numbers: 7 Left border: -10 ___ Amount of numbers: 3
NumRanDat = 9 Xmin = 0 Xmax = 10 Nint = 3 LGrInt = {0, 3, 6} array = {9, 4, 5, 2, 8, 4, 1, 9, 10}	Left border: 6 ___ Amount of numbers: 4 Left border: 3 ___ Amount of numbers: 3 Left border: 0 ___ Amount of numbers: 2

Вывод.

Изучили способ связи Ассемблера с ЯВУ (Си++), написали программу построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы и скомпилировали файл.

Приложение А.

Код написанной программы.

Файл lab6.cpp

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

extern "C"
{
    void ASM_MOD(short* arr, unsigned short* res, short* LGrInt, unsigned short NInt, unsigned short NumRanDat);
}

int main()
{
    srand(time(0));
    unsigned short NumRanDat;
    short Xmin;
    short Xmax;
    unsigned short NInt;
    std::cout << "Input length of the array: ";
    std::cin >> NumRanDat;
    if (NumRanDat > 16 * 1024 || NumRanDat <= 0)
    {
        std::cout << "Invalid length!\n";
        return 0;
    }
    std::cout << "Input min of values: ";
    std::cin >> Xmin;
```

```
            std::cout << "Input max of values: ";
            std::cin >> Xmax;
            if (Xmin > Xmax)
            {
               std::cout << "Minimum > maximum!\n";
               return 0;
            }
            std::cout << "Input number of interval: ";
            std::cin >> NInt;
            if (NInt > 24)
            {
               std::cout << "Invalid number!\n";
               return 0;
            }
            short* LGrInt = new short[NInt];
            std::cout << "Input elements of left borders:\n";
            for (int i = 0; i < NInt; i++)
            {
               std::cin >> LGrInt[i];
               if (LGrInt[i] < Xmin || LGrInt[i] > Xmax) std::cout << "Invalid
border!\n";
            }
            for (int i = 0; i < NInt - 1; i++)
            {
               for (int j = 0; j < NInt - 1 - i; j++)
               {
                  if (LGrInt[j] < LGrInt[j + 1])
                  {
                     short temp = LGrInt[j];
                     LGrInt[j] = LGrInt[j + 1];
```

```
                LGrInt[j + 1] = temp;
            }
        }
    }
    short* arr = new short[NumRanDat];
    std::cout << "Random array: ";
    for (int i = 0; i < NumRanDat; i++)
    {
        arr[i] = Xmin + rand() % (Xmin + Xmax);
        std::cout << arr[i] << ' ';
    }
    unsigned short* res = new unsigned short[NInt];
    for (int i = 0; i < NInt; i++) res[i] = 0;
    ASM_MOD(arr, res, LGrInt, NInt, NumRanDat);
    std::ofstream out;
    out.open("file.txt");
    for (int i = 0; i < NInt; i++)
    {
        std::cout << "Left border: " << LGrInt[i] << " ___ " << "Amount of
numbers: " << res[i] << '\n';
        out << "Left border: " << LGrInt[i] << " ___ " << "Amount of
numbers: " << res[i] << '\n';
    }
    delete[] arr;
    delete[] res;
    delete[] LGrInt;
    out.close();
    return 0;
}
```

Файл lab6.asm

```
.686
.MODEL FLAT, C
.DATA
elem DW 0
.CODE

ASM_MOD PROC C
   mov eax, [esp+4] ; arr
   mov ebx, [esp+8] ; res
   mov edx, [esp+12] ; LGrInt
   mov edi, [esp+16] ; NInt
   mov ecx, [esp+20] ; NumRanDat
   mov ecx, 0 ; and ecx, 0000FFFFH (обнуляем верхние биты счетчиков с помощью команды умножения)
   mov edi, 0 ; and edi, 0000FFFFH

f_1:
   push bx ; ebx
   mov bx, [eax]
   mov elem, bx
   pop bx

   mov esi, 0 ; индекс результирующего массива
   push ecx
   mov ecx, edi

f_2: ; внутренний цикл
   push edx
   mov dx, [edx+esi]
   cmp elem, dx
```

```
        jge f_3
        add esi, 2
        pop edx
        loop f_2 ; пока cx не будет равен 0, то есть пока не закончится
массив чисел

    f_3: ; внешний цикл. увеличивает кол-во попавших в диапазон чисел
        mov dx, [ebx+esi]
        inc dx
        mov [ebx+esi], dx
        pop edx
        pop ecx
        add eax, 2
        loop f_1

    ret

    ASM_MOD ENDP
    END
```