

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 псевдослу-
чайных целых чисел в заданные интервалы.

Студент гр. 7383

Бергалиев М.А.

Преподаватель

Кирияничков В.А.

Санкт-Петербург

2018

Цель работы.

Освоить организацию связи Ассемблера с ЯВУ.

Постановка задачи.

На языке высокого уровня программируется ввод с клавиатуры и контроль исходных данных, а также генерируется массив псевдослучайных целых чисел, изменяющихся в заданном диапазоне и имеющих равномерное или гауссовское распределение. 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 ($\leq 16K$)
2. Диапазон изменения массива псевдослучайных целых чисел $[Xmin, Xmax]$ (м.б. биполярный, например, $[-100, 100]$)
3. Массив псевдослучайных целых чисел.
4. Количество интервалов, на которые разбивается диапазон изменения массива псевдослучайных целых чисел - NInt (≤ 24)
5. Массив левых границ интервалов разбиения LgrInt (должны принадлежать интервалу $[Xmin, Xmax]$)

В общем случае интервалы разбиения диапазона изменения псевдослучайных чисел могут иметь различную длину.

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.

Программа формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы реализуется в виде двух ассемблерных модулей (процедур), первый из которых формирует распределение исходных чисел по интервалам единичной длины и возвращает его в вызывающую программу на ЯВУ как промежуточный результат. Также это распределение выводится на экран. 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 текстовой таблицы.

Замечание: На ЯВУ следует реализовать только ввод (возможно с контролем), вывод и генерацию псевдослучайных целых чисел. Всю остальную функциональность следует программировать на ассемблере.

Код программы.

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

extern "C" {
    void section(int xmin, int NInt, int* LGrInt, int* simpInt, int*
res);
    void simply_section(int NumRanDat, int xmin, int* pseudo, int*
res);
}
using namespace std;
int main()
{
    setlocale(LC_ALL, "rus");
```

```

        unsigned seed =
std::chrono::system_clock::now().time_since_epoch().count();
        std::minstd_rand0 generator(seed);
        int NumRanDat, xmin, xmax, NInt;
        cout << "Введите длину массива псевдослучайных чисел: ";
        cin >> NumRanDat;
        if (NumRanDat <= 0) {
            cout << "Некорректная длина массива" << endl;
            return 0;
        }
        cout << "Введите диапазон изменения массива псевдослучайных целых
чисел: ";
        cin >> xmin >> xmax;
        if (xmin >= xmax) {
            cout << "Неправильно заданный интервал" << endl;
            return 0;
        }
        cout << "Введите количество интервалов, на которое будет
разбиваться массив случайных чисел: ";
        cin >> NInt;
        if (NInt <= 0) {
            cout << "Некорректное количество интервалов" << endl;
            return 0;
        }
        int* LGrInt = new int[NInt];
        cout << "Введите последовательно левые границы интервалов(кроме
первой левой):" << endl;
        for (int i = 0; i < NInt - 1; ++i)
            cin >> LGrInt[i];
        LGrInt[NInt - 1] = xmax;
        if (LGrInt[0] > xmax || LGrInt[0] <= xmin || LGrInt[NInt-2] > xmax
|| LGrInt[NInt - 2] <= xmin) {
            cout << "Некорректные границы интервалов" << endl;
            return 0;
        }
        for (int i = 1; i < NInt; ++i)
            if (LGrInt[i] <= LGrInt[i - 1]) {
                cout << "Некорректные границы интервалов" << endl;
                return 0;
            }
        int* pseudo = new int[NumRanDat];
        for (int i = 0; i < NumRanDat; ++i)
            pseudo[i] = generator() % (xmax - xmin + 1) + xmin;
        int* res1 = new int[xmax - xmin + 1];
        for (int i = 0; i < xmax - xmin + 1; ++i)

```

```

        res1[i] = 0;
    cout << "Псевдослучайные числа:" << endl;
    for (int i = 0; i < NumRanDat; ++i)
        cout << pseudo[i] << " ";
    simply_section(NumRanDat, xmin, pseudo, res1);
    cout << endl;
    cout << "Количество чисел в единичных интервалах:" << endl;
    for (int i = 0; i < xmax - xmin + 1; ++i)
        cout << res1[i] << " ";
    cout << endl;
    int* res2 = new int[NInt];
    for (int i = 0; i < NInt; ++i)
        res2[i] = 0;
    section(xmin, NInt, LGrInt, res1, res2);
    cout << "Псевдослучайные числа на заданных отрезках:" << endl;
    cout << "№      left      count" << endl;
    cout << "1          " << xmin << "          " << res2[0] << endl;
    for (int i = 1; i < NInt; ++i)
        cout << i+1 << "          " << LGrInt[i-1] << "          " << res2[i]
<< endl;
    return 0;
}

```

asm.asm:

```

.386
.model flat, C
.CODE
section proc public, xmin : dword, NInt : dword, LGrInt : ptr dword,
simpInt : ptr dword, res : ptr dword
mov eax, simpInt
    mov ebx, LGrInt
    mov esi, res
    cycle1:
        mov edx, [ebx]
        cycle2:
            cmp edx, xmin
            jle end_cycle2
            inc xmin
            mov edi, [eax]
            add [esi], edi
            add eax, 4
        loopnz cycle2
    end_cycle2:
    add ebx, 4
    add esi, 4
    dec NInt

```

```

        cmp NInt, 0
        loopnz cycle1
        sub esi, 4
        mov edi, [eax]
        add [esi], edi

ret
section endp

simply_section proc public, NumRanDat:dword, xmin:dword, pseudo:ptr
dword, res: ptr dword
        mov eax, pseudo
        mov ebx, res
        cycle :
        mov edi, [eax]
            sub edi, xmin
            mov edx, [ebx+4*edi]
            inc edx
            mov [ebx+4*edi], edx
            add eax, 4
            dec NumRanDat
            cmp NumRanDat, 0
            loopnz cycle

ret
simply_section endp
end

```

Тестирование.

Входные данные	Результат
NumRanDat=15 Xmin=1 Xmax=16 NInt=4 LGrInt[i]: 4 7 9	Псевдослучайные числа: 3 13 3 13 3 10 4 1 7 12 5 6 7 7 11 Количество чисел в единичных интервалах: 1 0 3 1 1 1 3 0 0 1 1 1 2 0 0 0 Псевдослучайные числа на заданных отрезках: № left count 1 1 1 2 4 6 3 7 3 4 9 5

NumRanDat=14 Xmin=0 Xmax=24 NInt=8 LGrInt[i]: 2 3 4 5 6 8 9	Псевдослучайные числа: 16 8 2 18 14 1 17 22 7 16 16 3 6 4 Количество чисел в единичных интервалах: 0 1 1 1 1 0 1 1 1 0 0 0 0 1 0 3 1 1 0 0 0 1 0 0 Псевдослучайные числа на заданных отрезках: № left count 1 0 1 2 2 1 3 3 1 4 4 1 5 5 0 6 6 2 7 8 1 8 9 7
NumRanDat=30 Xmin=-10 Xmax=30 NInt=8 LGrInt[i]: -7 -4 -3 0 3 7 16	Псевдослучайные числа: 20 -3 28 -2 1 0 26 28 -1 -3 19 14 24 -6 26 -8 22 -6 5 29 8 3 4 14 29 29 18 -9 0 22 Количество чисел в единичных интервалах: 0 1 1 0 2 0 0 2 1 1 2 1 0 1 1 1 0 0 1 0 0 0 0 0 2 0 0 0 1 1 1 0 2 0 1 0 2 0 2 3 0 Псевдослучайные числа на заданных отрезках: № left count 1 -10 2 2 -7 2 3 -4 0 4 -3 4 5 0 3 6 3 3 7 7 3 8 16 13
NumRanDat=9 Xmin=-6 Xmax=30 NInt=10 LGrInt[i]: -3 0 3 6 9 12 15 18 21	Псевдослучайные числа: -2 20 10 23 23 3 10 28 13 Количество чисел в единичных интервалах: 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 2 0 0 1 0 0 0 0 0 0 1 0 0 2 0 0 0 0 1 0 0 Псевдослучайные числа на заданных отрезках: № left count 1 -6 0 2 -3 1 3 0 0 4 3 1 5 6 0 6 9 2 7 12 1 8 15 0 9 18 1 10 21 3

Выводы.

В ходе данной лабораторной работы была изучена связь Языка Assembler с языком высокого уровня C++ посредством реализации отдельного ассемблерного модуля.