

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. 1383

Федорова О.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Разработать программу на ЯВП с модульной функцией на языке ассемблер, которая будет высчитывать количество псевдослучайных чисел, попавших в заданные левые границы.

Задание.

На языке высокого уровня (Pascal или C) генерируется массив псевдослучайных целых чисел, изменяющихся в заданном диапазоне и имеющих равномерное распределение.

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.

Выполнение работы.

Разработана программа на языке C, производящая считывание всех необходимых данных и их запись в массивы. NumRanDat — размер массива случайных чисел, Xmin — максимальный размер числа, Xmax - минимальный, Nint — количество левых границ, LgrInt — массив границ. После считывания данных, происходит генерирование случайных чисел (функция rand(), для соблюдения границ происходит обработка $Xmin + \text{modul}(\text{rand()} \% (Xmax - Xmin))$, где modul — функция взятия модуля числа, написанная в программе)

После вызывается функция module(arr, LGrInt, res, NumRanDat, Nint), в которой вся обработка происходит на языке ассемблер. В нем данные автоматически помещаются в регистры, а именно:

rdi будет хранить смещение до массива случайных чисел arr

rsi — смещение до массива с границами

rdx — смещение до массива результата res

ecx - размер массива случайных чисел

r8d — размер массива с границами

В цикле производится обход всех элементов массива arr, при переходе на следующий элемент в регистр eax помещается 0, после чего происходит сравнение с каждой из границ, после каждого сравнение происходит либо переход на следующий элемент(если число не меньше), либо увеличение ячейки массива res с индексом eax и последующее увеличение eax, затем сравнение с r8d и, если не меньше, то переход на следующий символ массива arr.

Для построения графиков в коде программы потребуется закомментировать стоку с выводом случайных чисел и при сборке указать make txt.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	5	2 8 1 8 6	Если элемент равен границе — то он тоже считается
	1 10	0 1 1	
	3	1 3 2	
	1 3 6	2 6 3	
2.	7	2 8 1 8 6 8 2	

	1	10			0	1	1	
	3	1	4	7	1	4	3	
					2	7	4	

Выводы.

Изучена обработка данных в языке высшего порядка с использованием модульной функции на языке ассемблер.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

modules.s

.global module

module:

next_elem:

 mov ebx, [rdi][(rcx - 1)*4]

 mov eax, 0

next_border:

 cmp ebx, [rsi][rax*4]

 jle write

 inc eax

 cmp eax, r8d

 jle next_border

 jmp end

write:

 incq [rdx][rax*4]

 inc eax

 cmp eax, r8d

 jl next_border

end:

 loop next_elem

ret

;
 # rdi -- int* arr

;
 # rsi -- int* LGrInt

;
 # rdx -- int* res

;
 # ecx -- int NumRanDat # r8d -- int NInt

main.c

#include <stdio.h>

#include <stdlib.h>

/*

1. Длина массива псевдослучайных целых чисел - NumRandat ($\leq 16K$, $K=1024$)

2. Диапазон изменения массива псевдослучайных целых чисел $[Xmin, Xmax]$, значения

могут быть биполярные;

13

3. Количество интервалов, на которые разбивается диапазон

изменения

массива

псевдослучайных целых чисел - NInt (≤ 24)

4. Массив левых границ интервалов разбиения LGrInt (должны принадлежать интервалу

$[Xmin, Xmax]$).

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)

*/

int modul(int a) {

```

return a > 0 ? a : -a;
}

```

```

void module(int*, int*, int*, int, int);

```

```

int * sort(int * a, int size) {
    for(int i = 0; i < size - 1; i++) {
        for(int j = i + 1; j < size; j++) {
            if(a[i] > a[j]) {
                int t = a[i];
                a[i] = a[j];
                a[j] = t;
            }
        }
    }
}

```

```

int main () {

```

```

    int NumRanDat, Xmin, Xmax, NInt;

```

```

    //          длина массива, мин, макс, количество интервалов
    scanf("%d %d %d %d", &NumRanDat, &Xmin, &Xmax, &NInt);

```

```

    int* LGrInt = calloc(NInt, sizeof(int));

```

```

    //Введ левых границ интервалов

```

```

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

```

```

        scanf("%d", &LGrInt[i]);
    }
    //сортировка пузырьком)))
    sort(LGrInt, NInt);

    int* arr = calloc(NumRanDat, sizeof(int));

    //Заполнение массива рандомом чисел
    for (int i = 0; i < NumRanDat; i++){
        arr[i] = Xmin + modul(rand() % (Xmax - Xmin));
        printf("%d ", arr[i]);
    }
    printf("\n");

    int* res = calloc(NInt, sizeof(int));
    //result
    module(arr, LGrInt, res, NumRanDat, NInt);

    for (int i = 0; i < NInt ; i++){
        printf("%d\t%d\t%d\n", i, LGrInt[i], res[i]);
    }
    free(LGrInt);
    free(arr);
    free(res);
    return 0;
}

```


