

**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. 1381 |  | Демчук П. Д.  |
| Преподаватель      |  | Ефремов М. А. |

Санкт-Петербург

2022

## **Цель работы.**

Реализация программы построения частотного распределение попаданий псевдослучайных целых чисел в заданные интервалы, организовать связь яву и Ассемблера.

## **Задание.**

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;
3. Количество интервалов, на которые разбивается диапазон изменения массива псевдослучайных целых чисел - NInt ( $\leq 24$  )

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

Для бригад с нечетным номером: подпрограмма формирования распределения количества попаданий псевдослучайных целых чисел в заданные интервалы реализуется в виде одного ассемблерного модуля, сразу формирующего требуемое распределение и возвращающего его в главную программу, написанную на ЯВУ;

### **Выполнение работы.**

На языке высокого уровня (C) происходит считывание с консоли длины массива, диапазон изменения массива, количество интервалов и левые границы интервалов (считается что границы строго больше XMin и строго меньше XMax, следовательно их количество на 1 меньше количества интервалов). Генерируется массив псевдослучайных чисел.

Код на Ассемблере формирует распределение количества попаданий псевдослучайных целых чисел в заданные интервалы:

next\_elem - в ebx помещается элемент массива

next\_border - элемент сравнивается с левыми границами интервала: если он меньше левой границы, то принадлежит предыдущему интервалу. eax (счетчик левых границ) сравнивается с edx (количество интервалов минус 2). Если значения в этих регистрах равны, значит достигнута последняя левая граница, то есть элемент массива принадлежит последнему интервалу.

Перебор элементов массива происходит с помощью команды loop (в ecx находится количество элементов в массиве).

### **Выводы.**

В ходе лабораторной работы была написана программа построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы.

## ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include <stdio.h>

#include "stdlib.h"

#include "time.h"

int cmp(const void *a, const void *b){
    return *(int*)a - *(int*)b;
}

void func (int*, int*, int, int, int*);

int main () {
    srand(time(NULL));

    int NumRanDat, Xmin, Xmax, NInt;

    printf("Введите длину массива, диапазон изменения массива и
    количество интервалов:\n");

    scanf("%d %d %d %d", &NumRanDat, &Xmin, &Xmax, &NInt);

    if (NumRanDat < 0 || NumRanDat > 16384 || Xmin >= Xmax ||
    NInt < 0 || NInt > 24){

        printf("Неверно введенные данные.\n");

        return 0;
    }

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

    printf("Введите левые границы интервалов интервалов:\n");
```

```

for (int i = 0; i < NInt - 1; i++){
    scanf("%d", &LGrInt[i]);
    if (LGrInt[i] < Xmin || LGrInt[i] > Xmax){
        printf("Неверно введенные данные.\n");
        return 0;
    }
}

qsort(LGrInt, NInt-1, sizeof(int), cmp);

int* arr = calloc(NumRanDat, sizeof(int));
printf("Массив псевдослучайных чисел:\n");
for (int i = 0; i < NumRanDat; i++){
    arr[i] = rand() % (Xmax - Xmin) + Xmin;
    printf("%d ", arr[i]);
}
printf("\n");

int* res = calloc(NInt, sizeof(int));
for (int i = 0; i < NInt; i++){
    res[i] = 0;
}

func(arr, LGrInt, NInt, NumRanDat, res);

FILE* file = fopen("result.txt", "w");

printf("Результат:\n");

```

```

printf("%d\t%d\t%d\n", 1, Xmin, res[0]);
fprintf(file, "%d\t%d\t%d\n", 1, Xmin, res[0]);
for (int i = 2; i < NInt + 1; i++){
    printf("%d\t%d\t%d\n", i, LGrInt[i-2], res[i-1]);
    fprintf(file, "%d\t%d\t%d\n", i, LGrInt[i-2], res[i-1]);
}

fclose(file);
free(LGrInt);
free(arr);
free(res);

return 0;
}

```

**Название файла: func.s**

```

.global func

# rdi arr
# rsi LGrInt
# rdx NInt
# rcx NumRanDat
# r8 res

func:
    sub edx, 2
start:
    mov eax, 0
next_elem:

```

```

        mov ebx, [rdi][rcx*4]
next_border:
        cmp ebx, [rsi][rax*4]
        jl write
        cmp eax, edx
        je write_last_interval
        inc eax
        jmp next_border
write:
        incq [r8][rax*4]
        jmp end
write_last_interval:
        incq [r8][rax*4+4]
end:
        loop start

        ret

```