

**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**

Студентка гр. 1381

\_\_\_\_\_

Манцева Т.К.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Изучить организацию связи Ассемблера с ЯВУ на примере программы построения частотного распределения попаданий псевдослучайных целых чисел в заданные интервалы.

### **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  $[X_{\min}, X_{\max}]$ ).

### **Выполнение работы.**

Для выполнения лабораторной работы написан код на Си, при выполнении которого считываются значения диапазона псевдослучайных чисел, количество интервалов и их левые границы, генерируются псевдослучайные числа, вызывается далее описываемый код на Ассемблере для нахождения количества чисел, попавших в тот или иной интервал и, наконец, записывается результат. Функция, определяющая количество чисел, попавших в интервалы, принимает указатели на первые элементы(или массивы) массивов со случайными числами и границами, длины этих массивов, указатель на первый элемент массива с результатами. Все эти значения хранятся в регистрах `rdi`, `rsi`, `rdx`, `rcx`, `r8` соответственно. Во время выполнения метки `main_loop` осуществляется проверка на вхождение в интервал текущего числа, начиная с конца массива `randarr`(счетчик-уменьшающийся с каждой итерацией `rcx`). Для этого выполняется метка-внутренний цикл `internal_loop`, в котором значение текущего числа сохраняется в регистр `ebx`, сравнивается со значениями границ по индексам `eax`, `eax+1`. Если оно больше или равно `borders[eax]` и строго меньше `borders[eax+1]` или `eax` равно уменьшенной на единицу длине массива границ, то выполняется метка `write`, в которой увеличивается значение массива с результатами, соответствующему текущей левой границе. В иных случаях значение `eax` увеличивается на единицу, выполняется сравнение со следующими границами. Разработанный код представлен в приложении А.

### **Выводы.**

Была изучена организация связи Ассемблера с ЯВУ.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb6.s

```
.globl distribution
.type distribution, @function

distribution:

main_loop:
    mov eax, 0
    internal_loop:
        cmp eax, edx
        jge exit
        mov ebx, [rdi][rcx*4-4]
        cmp ebx, [rsi][rax*4]
        jl exit
        push rdx
        dec edx
        cmp eax, edx
        pop rdx
        je write
        cmp ebx, [rsi][rax*4 +4]
        jl write
        inc eax
        jmp internal_loop
    write:
        incq [r8 + 4*rax]
    exit:
loop main_loop

ret
```

Название файла: lab6.c

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

extern void distribution(int *randarr, int *borders, int borders_len,
int randarr_len, int *result);

int main()
{
    int randarr_len;
    int min;
    int max;
    int borders_len;
    printf("Enter array length\n");
    scanf("%d", &randarr_len);
    if (0 >= randarr_len)
    {
        printf("Wrong length\n");
        return 0;
    }
    printf("Enter min and max\n");
    scanf("%d %d", &min, &max);
    if (min >= max)
    {
        printf("Wrong range\n");
        return 0;
    }
    printf("Enter number of intervals\n");
    scanf("%d", &borders_len);
    if (0 >= borders_len || borders_len > 24)
    {
        printf("Wrong number\n");

        return 0;
    }
    int *borders = calloc(borders_len, sizeof(int));
    printf("Enter borders in ascending order\n");
    for (int i = 0; i < borders_len; i++)
    {
        scanf("%d", &borders[i]);
        if (borders[i]<min | borders[i]> max)
        {
            printf("Wrong border\n");
            free(borders);
            return 0;
        }
    }

    int *result = calloc(borders_len, sizeof(int));
    srand(time(NULL));
```

```

int *randarr = calloc(randarr_len, sizeof(int));
printf("Random numbers:\n");
for (int i = 0; i < randarr_len; i++)
{
    randarr[i] = min + rand() % (max - min + 1);
    printf("%d ", randarr[i]);
}

distribution(randarr, borders, borders_len, randarr_len, result);

printf("\n");
FILE *file;
file = fopen("result.txt", "w");
for (int i = 0; i < borders_len; i++)
{
    if (i < borders_len - 1)
    {
        printf("%d numbers in [%d,%d) interval\n", result[i],
borders[i], borders[i + 1]);
    }
    else
    {
        printf("%d numbers in [%d,%d] interval\n", result[i],
borders[i], max);
    }
    fprintf(file, "interval №%d, border - %d, result - %d\n", i +
1, borders[i], result[i]);
}
fclose(file);
free(result);
free(borders);
free(randarr);

return 0;
}

```

Название файла: Makefile

```
all: lab6.o lb6.o
    gcc -g3 lab6.o lb6.o -Wall -masm=intel
lab6.o: lab6.c
    gcc -c lab6.c
lb6.o: lb6.s
    as lb6.s -msyntax=intel -mnaked-reg -mmnemonic=intel --gstabs -o
lb6.o
```