

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 распределения попаданий псевдослучайных
целых чисел в заданные интервалы

Студент гр. 1381

Кагарманов Д. И.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Цель работы.

Научиться связывать язык Ассемблера и язык высокого уровня так, чтобы функции ассемблерного модуля вызывались из программы на ЯВУ. Написать программу построения частного распределения попаданий псевдослучайных целых чисел в заданные интервалы.

Задание.

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 распределение.

Далее должны вызываться две ассемблерные 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 – *Legth*
2. Диапазон изменения массива псевдослучайных целых чисел $[X_{mmiii}, X_{mmmm}]$, 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 $[X_{mmiii}, X_{mmmm}]$).

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 головную программу, написанную на ЯВУ.

Выполнение работы.

Программа написана с использованием языка программирования C. В файле *main.c* происходит считывание с консоли длины массива, диапазон массива, количество интервалов и левые границы интервалов (считается, что границы строго больше XMin и строго меньше XMax, следовательно их количество на единицу 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. `ecx` (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).

Исходный код программы представлен в приложении А.

Выводы.

Были изучены принципы работы Ассемблера с ЯВУ, а также разработана программа, которая строит частотное распределение попаданий псевдослучайных целых чисел в заданные интервалы.

ПРИЛОЖЕНИЕ А

КОД ПРОГРАММЫ

Название файла: main.c

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

int rand_between(int const from, int const to) {
    if (to == from)
        return to;
    if (to < from)
        return rand_between(to, from);
    return from + rand() % (to-from+1);
}

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

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

int main () {
    srand(time(NULL));
    int NumRanDat, Xmin, Xmax, NInt;

    printf("\t---Введите длину массива, минимум и максимум диапазона и
количество интервалов---:\n");
    scanf("%d %d %d %d", &NumRanDat, &Xmin, &Xmax, &NInt);

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

    printf("\t---Введите левые границы интервалов---(их %d)\n", NInt-1);

    for (int i = 0; i < NInt - 1; i++){
        scanf("%d", &LGrInt[i]);
    }

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

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

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

    module(arr, LGrInt, NInt, NumRanDat, res);
    FILE* file = fopen("answer.txt", "w");
```

```

printf("\t---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]);
}
free(LGrInt);
free(arr);
free(res);
fclose(file);

return 0;
}

```

Название файла: module.s

```
.global module
```

```

module:
    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

```