

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.

Студент гр. 1383

Малых А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Получить навыки реализации связи программы на языке Ассемблера с программой на языке высокого уровня.

Задание.

(Вариант №1)

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 $[Xmin, Xmax]$, значения 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 $[Xmin, Xmax]$).

Результаты:

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)

В зависимости от номера бригады формирование частотного распределения должно производиться по одному из двух вариантов:

Вариант 1.

1. 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 головную программу, написанную на ЯВУ;

Выполнение работы.

Реализована программа на языке Си вызывающая в ходе работы ассемблерный модуль. На языке высокого уровня организован ввод исходных данных, генерация массива случайных чисел и вывод результатов — в файл и в консоль. Ассемблерный модуль `getSpread`, получает на вход массив левых границ `left_borders`, количество интервалов, `Nint`, массив случайных чисел `rand_arr`, его длину `numRanNum` и массив результатов `res`. По метке `loadNum` в `eax` кладётся число за `rand_arr`, увеличивается индексный регистр `rdi`, уменьшается количество непросмотренных чисел, хранящееся в регистре `rdx`. Сохраняется в стеке значение регистра `rcx` — количество интервалов. Далее, начиная с конца просматривается массив `left_borders`, если число в `eax` содержится в промежутке, то нужно инкрементировать значение

соответствующего элемента в массиве `ges`, восстановить из стека значения `gsx` и проверить условие цикла. Всё это выполняется по метке `writeRes`.

Исходный код программы см. в Приложении А

Выводы.

Получены навыки реализации связи программы на языке Ассемблера с программой на языке высокого уровня.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab6.c

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

extern void getSpread(int *rand_arr, int *left_borders, int
NumRanDat, int NInt, int *res);

int *genRandomArray(int length, int x_min, int x_max)
{
    srand(time(0));
    int *arr = (int *)calloc(length, sizeof(int));
    for (int i = 0; i < length; ++i)
    {
        arr[i] = x_min + rand() % (x_max - x_min + 1);
    }
    return arr;
}

int main()
{
    int NumRanDat = 0, x_min = 0, x_max = 0, NInt = 0;
    printf("Enter NumRanDat: ");
    scanf("%d", &NumRanDat);
    printf("Enter x_min, x_range: ");
    scanf("%d %d", &x_min, &x_max);
    printf("Enter NInt: ");
    scanf("%d", &NInt);

    if (NumRanDat < (2 << 14) && x_max > x_min && NInt < 24)
    {

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

```

{
    printf("Enter left borders: ");
    for (int i = 0; i < NInt; i++)
    {
        scanf("%d", LGrInt + i);
        if (LGrInt[i] < x_min || LGrInt[i] > x_max)
        {
            printf("Incorrect value of border!\n");
            free(LGrInt);
            return 0;
        }
    }

    int *rand_arr = genRandomArray(NumRanDat, x_min,
x_max);

    printf("Array of random nums: ");
    for (int i = 0; i < NumRanDat; ++i)
    {
        printf("%d ", *(rand_arr + i));
    }
    putchar('\n');

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

    getSpread(rand_arr, LGrInt, NumRanDat, NInt, res);
    printf("Results: ");
    for (int i = 0; i < NInt; i++)
    {
        printf("%d ", res[i]);
    }
    putchar('\n');
    FILE *file;
    file = fopen("results.txt", "w");
    for (int i = 0; i < NInt; i++)
    {
        fprintf(file, "%d %d %d\n", i + 1, LGrInt[i],
res[i]);

```

```

        }
        fclose(file);
        free(res);
        free(rand_arr);
        free(LGrInt);
    }
}
else
{
    printf("The values you entered are incorrect!\n");
}

return 0;
}

```

Название файла: getSpread.S

```

.global getSpread

getSpread:
    push rax
    and rax, 0
    sub rcx, 1

loadNum:
    mov eax, [rdi] # Берем число из rand_arr
    add rdi, 4
    dec rdx        # Уменьшаем счётчик непрсмотренных чисел в
rand_arr
    push rcx       # Сохраняем Nint, т.к он будет счетчиком.

findInterval:
    cmp eax, [rsi][rcx * 4]
    jge writeRes
    loop findInterval

writeRes:
    incq [r8][rcx * 4]

```

```
        pop rcx                                # восстанавливаем счетчик оставшихся
элементов в left_borders
        cmp rdx, 0
        jne loadNum
    pop rax
    .end
```