

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»

**Тема: Организация связи Ассемблера с ЯВУ на примере программы
построения частотного распределение попаданий псевдослучайных
целых чисел в заданные интервалы.**

Студент гр. 0382

Тихонов С.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

На языке С программируется ввод с клавиатуры и контроль исходных данных, а также генерируется массив псевдослучайных целых чисел, изменяющихся в заданном диапазоне и имеющих заданный закон распределения. Необходимые датчики псевдослучайных чисел находятся в каталоге RAND_GEN (при его отсутствии получить у преподавателя).

Следует привести числа к целому виду с учетом диапазона изменения.

Далее должны вызываться 1 или 2 ассемблерных 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 - NumRanDat ($\leq 16K$)
2. Диапазон изменения массива псевдослучайных целых чисел
[Xmin, Xmax] (м.б. биполярный, например, [-100, 100])
3. Массив псевдослучайных целых чисел $\{X_i\}$.
4. Количество интервалов, на которые разбивается диапазон
изменения массива псевдослучайных целых чисел - NInt (≤ 24)
5. Массив левых границ интервалов разбиения LGrInt .

В общем случае интервалы разбиения диапазона изменения псевдослучайных чисел могут иметь различную длину, левые границы могут задаваться в произвольном порядке и иметь произвольные значения. Если $X_{min} < LGrInt(1)$, то часть данных не будет участвовать в формировании распределения. Каждый интервал, кроме последнего, следует интерпретировать как $[LGrInt(i), LGrInt(i+1))$. Если у последнего интервала правая граница меньше Xmax, то часть данных не будет участвовать в формировании распределения.

Результаты:

Текстовая таблица, строка которой содержит:

- номер интервала,
- левую границу интервала,
- количество псевдослучайных чисел, попавших в интервал.

Количество строк должно быть равно числу интервалов разбиения.

Таблица должна выводиться на экран и сохраняться в файле.

Вариант 10

Распределение — нормально. Число ассемблерных процедур 1. $N_{int} < D_x$,
Первая левая граница $> X_{min}$, Правая граница последнего интервала $> X_{max}$.

Порядок выполнения работы.

В функции `main` происходит считывание данных с консоли, проверка введённых значений на корректность. Проверяем длину интервала. Созданием нормального распределения происходит при помощи функции стандартной библиотеки `normal_distribution`. Полученные левые границы сортируем. Проверяем меньше ли минимальное значение самого левого интервала. В случае не корректности введённых данных пользователю выдаётся сообщение об ошибке и программа завершается с кодом 0.

Для подсчёт чисел входящих в соответствующий интервал написана функция на ассемблере(`Source.asm`). В нём реализована необходимая функция, принимающая на вход массив чисел, длину массива, массив границ, длину массива границ и массив куда надо записать результат работы функции — количество чисел входящих в интервал. С помощью цикла `l1` просматриваются все элементы массива, а с помощью меток `boarders` и `boarder_exit` просматриваются все границы. После выхода из цикла проверяется был ли найден подходящий интервал. В случае, если необходимый интервал был найден, значение в `result_array` увеличивается на 1.

Результат работы программы записывается в файл и выводится на экран.

Вывод.

В ходе выполнения данной лабораторной работы была изучена

организация связи ассемблера с ЯВУ. Была реализована программа частотного распределения случайных чисел по заданным интервалам на языке C++ с использованием ассемблерного модуля.

ТЕСТИРОВАНИЕ

Тест 1.

```
Введите количество чисел: 10
Введите максимальное число: 5
Введите минимальное число: -5
Введите количество интервалов: 2
Введите левые границы: -1 1
Сгенерированные числа: 2 1 -2 0 1 -3 -2 -1 -1 1
Номер интервала      Левая граница интервала      Количество чисел в интервале
1                    -1                    3
2                     1                    4
Для продолжения нажмите любую клавишу . . .
```

Работает верно

Тест 2.

```
Введите количество чисел: 30
Введите максимальное число: 15
Введите минимальное число: -15
Введите количество интервалов: 5
Введите левые границы: -10 -8 -5 0 1
Сгенерированные числа: -1 6 -6 -7 7 1 -3 -8 -6 -12 3 -7 -2 9 0 8 -5 7 3 3 -7 3 -4 -10 -6 7 -3 5 -5 13
Номер интервала      Левая граница интервала      Количество чисел в интервале
1                    -10                    1
2                     -8                    7
3                     -5                    7
4                      0                    1
5                      1                   13
Для продолжения нажмите любую клавишу . . .
```

Работает верно