Simulation of open type QS. RNG
Shemakhin E. (Russian Federation)
Моделирование СМО открытого типа. ГСЧ
Шемахин Е. Ю. (Российская Федерация)

Шемахин Евгений Юрьевич / Shemakhin Evgeny – аспирант,
кафедра интеллектуальных систем и управления информационными ресурсами,
Казанский национальный исследовательский технологический университет, г. Казань

Аннотация: *рассматриваются вопросы генерации последовательности случайных чисел и тестирования полученной псевдослучайной последовательности, используемой в алгоритме, моделирующем многоканальную открытую систему массового обслуживания.*
Abstract: *the problems of generating a sequence of random numbers and test the resulting pseudo-random-sequence used in the algorithm, simulating an open multi-channel queuing system.*

Ключевые слова: *система массового обслуживания, характеристики системы, моделирование, генерация случайных чисел, тестирование псевдослучайной последовательности.*
Keywords: *queuing system, characteristics of the system, modeling, generation of random numbers, randomness tests.*

Алгоритм [3], реализующий модель многоканальной открытой СМО с ограничениями на каждом шаге, получает две случайные величины: время обслуживания пришедшего в данный момент требования и время до появления следующего требования в системе. При наличии ограничений в системе также будут введены соответствующие им случайные величины. Данные промежутки времени распределены по экспоненциальному закону [2, с. 25]: $F(t)=1-e^{-\lambda\cdot t}$, где λ – интенсивность потока событий. Таким образом, промежуток времени конкретного события, может быть получен из выражения:

$$t_i=-\frac{1}{\lambda}\cdot\ln(1-x_i), \qquad (1)$$

где x_i – случайная величина, обладающая равномерным распределением на интервале $(0;1)$. На практике этот интервал целесообразно взять равным $(0{,}0001;0{,}9999)$, ограничивая последовательность получаемых от генератора случайных величин. При сужении интервала уменьшается точность вычислений, а при расширении сходимость динамической величины не может быть гарантирована. Очевидно, получить при помощи программного обеспечения действительно случайную величину невозможно, можно лишь аппроксимировать некоторые ее свойства. Полученные последовательности случайных величин должны удовлетворять следующим критериям: обладать равномерным распределением на $(0{,}0001;0{,}9999)$, мат. Ожиданием, равным $0{,}5$, и достаточно большим периодом. Если полученная последовательность удовлетворяет этим критериям, то для дальнейшего исследования необходимо произвести ряд тестов: частотный побитовый тест, частотный блочный тест, тест на последовательность одинаковых битов, тест на равномерность распределения по квантилям. Первые три теста входят в пакет статистических тестов «DIEHARD» [6] и «NIST» [4], самых строгих на данный момент тестов, предназначенных для проверки последовательностей случайных величин. Для оценки данными статистическими методами последовательности действительных чисел, последние должны быть преобразованы в последовательность битов согласно следующему правилу: величины, большие $0{,}5$, соответствуют единице; величины, меньшие $0{,}5$, соответствуют нулю; величины, равные $0{,}5$, не рассматриваются. Такое преобразование не искажает результаты тестов. Ниже приведены результаты тестирования двух наиболее удачных генераторов псевдослучайных чисел.

1. Криптографический генератор со смещением бит в последовательности.

Данный генератор является результатом успешной попытки расширения встроенного в среду Visual Studio 2010 криптогенератора [5] и лишен его главного недостатка – «сгущения» элементов создаваемой последовательности возле середины интервала. Получение последовательности случайных чисел происходит следующим образом: на этапе инициализации при помощи криптогенератора создается последовательность битов заданной длины, затем при получении каждого из последующих чисел с помощью второго криптогенератора создается последовательность случайной длины (не превышающей число бит в первоначальной последовательности), содержащая индексы последовательности предыдущего числа, биты которого будут заменены на противоположные, полученное двоичное число переводится в целое, а затем в действительное. Главным недостатком данного генератора является время генерации, превышающее время стандартного криптогенератора более чем в два раза, что компенсируется качеством создаваемой последовательности. Изменяемым параметром генератора является количество разрядов n

генерируемого числа. Гистограмма частот (рис. 1) и характеристики (табл. 1) криптогенератора со смещением представлены ниже.

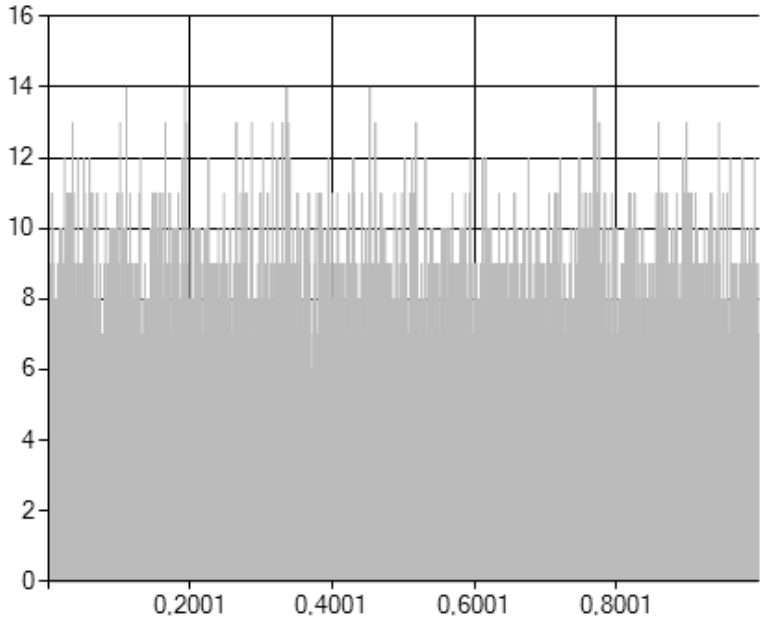

Рис. 1. Криптогенератор со смещением, $N = 50000, n = 64$

Таблица 1. Криптогенератор со смещением, $N = 5 \cdot 10^9, n = 64$

Время генерации	$\sim 1925 мс$
Минимальное значение величины	$\sim 0{,}0005416$
Максимальное значение величины	$\sim 0{,}999458$
Период генератора	$> 5 \cdot 10^9$
Наименьшая частота величины	~ 1
Наибольшая частота величины	~ 14
Частотный побитовый тест, относительное отклонение (%)	$\sim 0{,}00084039$
Частотный блочный тест (5 блоков), максимальное относительное отклонение (%)	$\sim 0{,}040559$
Последовательность одинаковых битов, максимальная длина	~ 26
средняя длина	$\sim 2{,}000333415$
минимальная длина	1
Распределение величин по 10 квантилям (%)	9,96;10,01;...;10,01;9,96
Отклонение от среднего относительное (%)	$\sim 0{,}007532$

2. Встроенный генератор среды Visual Studio 2010.

Среда VS 2010 обладает генератором случайных чисел, основанном на субтрактивном алгоритме Дональда Е. Кнута [1, с. 46]. Случайное число может быть получено из следующего соотношения: $x_{n+1} = (x_{n-24} + x_{n-55}) \bmod (m)$. Очевидно, $n > 55$, а m желательно выбрать таким образом, чтобы оно было наибольшим простым числом. При инициализации генератора имеется возможность задать начальное значение «seed». Достоинством этого генератора является высокая скорость работы, превосходящая другие генераторы, а также высокое качество получаемой последовательности чисел, уступающее лишь криптогенератору со смещением бит, описанному выше. Изменяемым параметром данного генератора является количество разрядов n получаемого числа. Гистограмма частот (рис. 2) и характеристики (табл. 2) встроенного ГСЧ среды VS 2010 представлены ниже.

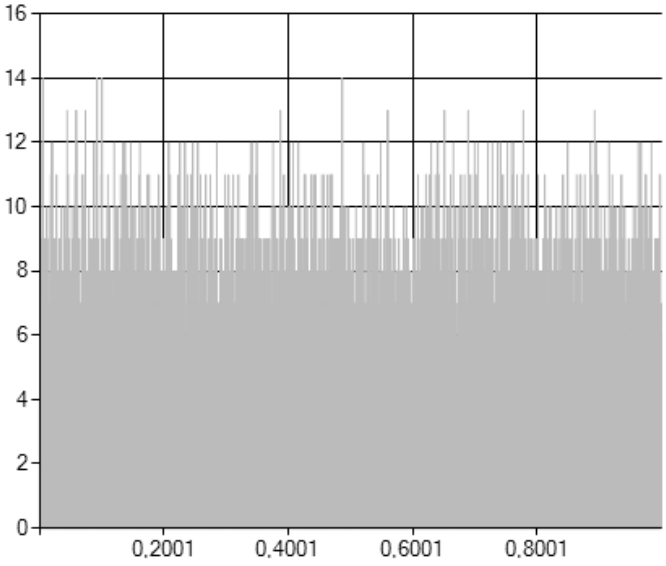

Рис. 2. Встроенный ГСЧ среды VS 2010, $N = 50000, n = 6$

Таблица 2. Встроенный ГСЧ среды VS 2010, $N = 5 \cdot 10^9, n = 6$

Время генерации	$\sim 591 мс$
Минимальное значение величины	$\sim 0{,}00012$
Максимальное значение величины	$\sim 0{,}999899$
Период генератора	$> 5 \cdot 10^9$
Наименьшая частота величины	~ 1
Наибольшая частота величины	~ 14
Частотный побитовый тест, относительное отклонение (%)	$\sim 0{,}0003398$
Частотный блочный тест максимальное относительное отклонение	$\sim 0{,}055$
Последовательность одинаковых битов, максимальная длина	~ 28
средняя длина	$\sim 1{,}999508$
минимальная длина	1
Распределение величин по 10 квантилям (%)	10;...;10
Отклонение от среднего относительное (%)	$\sim 0{,}003297$

На основе полученных данных можно сделать следующий вывод: встроенный генератор, основанный на субтрактивном методе Д. Кнута, имеет смысл использовать при необходимости произвести большое количество прогонок. Модифицированный криптогенератор среды VS 2010 имеет смысл использовать при необходимости произвести небольшое количество прогонок с достаточно большим числом требований в системе, а также для получения динамических значений тех характеристик, аналитические формулы которых не известны.

Литература

1. *Кнут Д.* Искусство программирования. Том 2. – Вильямс, 2002. – 788 с.
2. *Кирпичников А.П.* Методы прикладной теории массового обслуживания. – Казань: Изд-во Казанского университета, 2011. – 200 с.
3. *Шемахин Е.Ю., Кирпичников А.П.* Моделирование многоканальных открытых систем массового обслуживания с ограничениями в среде Visual Studio 2010 // Вестник Казанского технологического университета. – 2015. Т. 18, № 3.
4. A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications. NIST, 2010. 131 p.

5. Microsoft Developer Network [Электронный ресурс]. Руководство по программированию на C#, RNGCryptoServiceProvider. Режим доступа: https://social.msdn.microsoft.com/Search/ru-RU (дата обращения 30.09.2014 г.).
6. G. Marsaglia, The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness. National Science Foundation, 1995.