Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчет по лабораторной работе № 5

на тему

«Построение и исследование имитационной модели непрерывно – стохастической СМО»

Вариант 1

Выполнили: ст. группы 950505

Киреев Ю. В.

Денисов В. А.

Проверила: Герман Ю. О.

Минск 2022

1. **Цель работы**

Изучить методы имитационного моделирования поведения непрерывно-стохастической СМО.

1. **Краткие теоретические сведения**

Общий механизм выполнения лабораторной работы. Нужно

1. Сгенерировать последовательность равномерно распределенных вероятностей от 0 до 1. Можно использовать готовые таблицы.

2. По этим вероятностям и заданному закону F(t) времен поступления заявок в систему сгенерировать последовательность времен t1, t2, …, tz в соответствии с законом распределения F(t).

3. По этим вероятностям и заданному закону G(t) времен обслуживания заявок в системе сгенерировать последовательность времен τ1, τ2, …, τz в соответствии с законом распределения G (t).

4. На временных осях отложить соответствующие времена, как показано ниже:

t=6 t=8 t–время поступления

t=7.2 t=11.7 t– время поступления

Заметим, что в интервале от 7.2 до 8 система простаивает – нет заявок, ничего не обслуживается. Кроме того, возможно появление очереди на обслуживание.

1. **Задание**
2. Произвести имитационное моделирование для системы с одним прибором. Интенсивность поступления заявок λ = 0.1 сек−1, интенсивность обслуживания заявок μ = 2 сек-1. Закон распределения вероятностей времен поступления обозначен как F(t), времени обслуживания – как G(t).
3. По результатам моделирования найти – среднее время обслуживания, среднее время пребывания заявки в системе, среднее число заявок в системе, процент загрузки обслуживающего прибора (канала).

F(t) = 1−e−λt

G(t) = 1−e−μt

1. **Ход работы**

Итак, нужно показать, как генерировать случайные числа с показательным законом распределения. Воспользуемся линейным конгруэнтным методом, описанным в предыдущей лабораторной работе:

*X(i) = (a \* X(i-1) + c) mod m*

Возьмем первую вероятность 0,45. Тогда имеем

F(t) = 1−e-λt = 0.45

Отсюда

e-λt = 0.55

Интенсивность нам должна быть задана заранее. Пусть λ = 0.1 (сек-1).

Получаем

e-0.1t = 0.55

Логарифмируем

−0.1t = ln 0.55 = -0.597

t = 5.97 ≈ 6

Итак, первая заявка придет через 6 сек. Аналогично разыгрываем момент поступления второй заявки. Берем вторую вероятность – 0.16. Решаем уравнение

1−e-λt = 0.16

Отсюда

e-λt = 0.84

e-0.1t = 0.84

−0.1t = ln 0.84 = -0.17

t = 1.7

Вторую заявку ожидаем через 1.7 сек. По аналогии рассчитываем моменты поступления остальных заявок на обслуживание. Эти моменты времени имеют экспоненциальное распределение. Вообще, распределение может быть любым – главное, чтобы была задана соответствующая функция распределения F(t).

Теперь обратимся к процессу обслуживания. Нам нужно заранее знать функцию распределения времени обслуживания. Пусть в нашем примере это будет распределение Вейбулла. Формула такова

G (t) = 1− e−µt

Здесь как и ранее: μ − интенсивность потока обслуживания заявок, t − время, k − коэффициент формы, например, возьмем k=1. Пусть μ = 2 (сек-1).

Мы будем использовать ту же последовательность равномерно распределенных случайных чисел, представляющих вероятности. Итак, первая вероятность равна 0.45.

Имеем уравнение

0.45 = 1− e−2t

Отсюда

0.55 = e−2t

ln (0.55) = -2t

-0.597 /(-2) = t

t = 0.298 ≈ 0.3

Итак, первая заявка будет обслужена за время, равное 0.3 сек. Аналогичные вычисления проделаем со второй заявкой.

0.16 = 1− e−2t

ln (0.84) = -2t

-0.174 /(-2) = t

t = 0.087

Вторая заявка будет обслужена приблизительно за 0.087 сек. И т.д.

Выполним 100 таких итераций. Код программы:

import java.util.Arrays;

public class Main {

private static final int ITERATIONS = 100;

private static double getRandomNum(double num) {

int a = 131, c = 1021, m = 100;

num = ((a \* num) + c) % m;

return num / 100;

}

public static void main(String[] args) {

calculateModeling();

}

private static void calculateModeling() {

double[] arrivalTime = new double[ITERATIONS];

double[] processingTime = new double[ITERATIONS];

double tempNum, prevNum = 37;

for (int i = 0; i < ITERATIONS; i++) {

tempNum = getRandomNum(prevNum);

arrivalTime[i] = (Math.log(1 - tempNum)) / (-0.1);

processingTime[i] = (Math.log(1 - tempNum)) / (-2));

prevNum = tempNum;

}

System.out.print("Время прихода заявок: ");

Arrays.stream(arrivalTime).forEach(el-> System.out.printf("%.1f; ", el));

System.out.println("\nВремя обработки заявок: ");

Arrays.stream(processingTime).forEach(el-> System.out.printf("%.1f; ", el));

System.out.printf("\nСреднее время обслуживания: %.1f", Arrays.stream(processingTime).average().getAsDouble());

System.out.printf("\nСреднее время прихода заявок: %.1f", Arrays.stream(arrivalTime).average().getAsDouble());

}

}

Результат выполнения программы:

Время прихода заявок: 11,4; 1,1; 4,2; 10,7; 0,7; 3,6; 9,4; 0,1; 2,5; 6,9; 19,8; 4,2; 10,6; 0,7; 3,6; 9,2; 67,6; 7,3; 22,0; 4,7; 12,1; 1,4; 4,8; 12,2; 1,4; 4,9; 12,6; 1,6; 5,2; 13,4; 1,9; 5,8; 15,5; 2,8; 7,5; 23,0; 4,9; 12,7; 1,7; 5,3; 13,7; 2,1; 6,1; 16,5; 3,1; 8,2; 29,1; 6,0; 16,0; 2,9; 7,9; 25,6; 5,4; 14,2; 2,3; 6,5; 18,1; 3,6; 9,4; 0,1; 2,5; 6,9; 20,0; 4,2; 10,7; 0,7; 3,6; 9,4; 0,1; 2,5; 6,9; 19,8; 4,1; 10,6; 0,7; 3,5; 9,1; 50,5; 7,2; 21,2; 4,5; 11,6; 1,1; 4,3; 11,1; 0,9; 3,9; 10,1; 0,5; 3,1; 8,3; 29,3; 6,0; 16,1; 3,0; 7,9; 26,1; 5,5; 14,5; 2,4;

Время обработки заявок: 0,6; 0,1; 0,2; 0,5; 0,0; 0,2; 0,5; 0,0; 0,1; 0,3; 1,0; 0,2; 0,5; 0,0; 0,2; 0,5; 3,4; 0,4; 1,1; 0,2; 0,6; 0,1; 0,2; 0,6; 0,1; 0,2; 0,6; 0,1; 0,3; 0,7; 0,1; 0,3; 0,8; 0,1; 0,4; 1,2; 0,2; 0,6; 0,1; 0,3; 0,7; 0,1; 0,3; 0,8; 0,2; 0,4; 1,5; 0,3; 0,8; 0,1; 0,4; 1,3; 0,3; 0,7; 0,1; 0,3; 0,9; 0,2; 0,5; 0,0; 0,1; 0,3; 1,0; 0,2; 0,5; 0,0; 0,2; 0,5; 0,0; 0,1; 0,3; 1,0; 0,2; 0,5; 0,0; 0,2; 0,5; 2,5; 0,4; 1,1; 0,2; 0,6; 0,1; 0,2; 0,6; 0,0; 0,2; 0,5; 0,0; 0,2; 0,4; 1,5; 0,3; 0,8; 0,1; 0,4; 1,3; 0,3; 0,7; 0,1;

Среднее время обслуживания: 0,5

Среднее время прихода заявок: 9,1

Таким образом, среднее время обслуживания составляет 0.5 сек.

На временных осях отложим соответствующие времена:

11.4 12.5 16.7 27.4 28.1 31.7 41.1 41.2 43.7 50.6 t

12.0 12.6 16.9 27.9 28.1 31.9 41.6 43.8 50.9 t

Заметим, что в интервалах система простаивает – нет заявок, ничего не обслуживается.

Найдем среднее время пребывания заявки в системе по формуле

u = b / (1 - λb),где b – среднее время обслуживания заявок

u = 0.5 / (1 – 0.5 \* 0.1) = 0.526

Далее найдем среднее число заявок в системе по формуле

m = λu = 0.1 \* 0.526 = 0.0526

Рассчитаем процент загрузки обслуживающего прибора. Это отношение интенсивности потока заявок к суммарной интенсивности обслуживания. Процент загрузки позволяет определить, будет ли система справляться с задачами или из-за перегрузки будет неработоспособной.

k = (λ / 1\*μ) \* 100% = 0.1 / 2 \* 100% = 5%

1. **Вывод**

В результате выполнения лабораторной работы были изучены методы имитационного моделирования поведения непрерывно-стохастической СМО.