



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

к лабораторной работе №4

По курсу: «Моделирование»

Тема: «Обслуживающий аппарат»

Студентка ИУ7-75Б
Оберган Т.М
Вариант 14 (2)

Преподаватель
Рудаков И.В.

Москва, 2020 г.

Оглавление

Задание	3
Распределения.....	3
Равномерное распределение:.....	3
Нормальное распределение:	4
Формализация задачи.....	5
Пошаговый подход.....	5
Событийная модель.....	5
Результаты работы	6
Без повторов, 1000 заявок.....	6
10% повторов, 1000 заявок.....	6
10% повторов, 10000 заявок.....	7
50% повторов, 10000 заявок.....	7
100% повторов, 10000 заявок.....	8
Равномерный закон ОА, 0% повторов, 10000 заявок	8
Равномерный закон ОА, 100% повторов, 10000 заявок	8

Задание

Необходимо промоделировать систему, состоящую из генератора, памяти, и обслуживающего аппарата. Генератор подает сообщения, распределенные по равномерному закону, они приходят в память и выбираются на обработку по закону из ЛР2. Количество заявок конечно и задано. Предусмотреть случай, когда обработанная заявка возвращается обратно в очередь. Необходимо определить оптимальную длину очереди, при которой не будет потерянных сообщений. Реализовать двумя способами: используя пошаговый и событийный подходы.

Распределения

Равномерное распределение:

Равномерное распределение — распределение случайной величины, принимающей значения, принадлежащие некоторому промежутку конечной длины, характеризующееся тем, что плотность вероятности на этом промежутке всюду постоянна.

Равномерное распределение обозначают $X \sim R(a, b)$, где $a, b \in \mathbb{R}$.

Функция распределения равномерной непрерывной случайной величины:

$$F(x) = \begin{cases} 0 & \text{при } x \leq a \\ \frac{x - a}{b - a} & \text{при } a \leq x \leq b \\ 1 & \text{при } x > b \end{cases}$$

Плотность распределения равномерной непрерывной случайной величины:

$$f(x) = \begin{cases} \frac{1}{b - a} & \text{при } a \leq x \leq b \\ 0 & \text{иначе} \end{cases}$$

Нормальное распределение:

Нормальное распределение - распределение вероятностей, которое в одномерном случае задаётся функцией плотности вероятности, совпадающей с функцией Гаусса:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

где параметр μ — математическое ожидание (среднее значение), медиана и мода распределения, а параметр σ - среднеквадратическое отклонение (σ^2 - дисперсия) распределения.

Функция распределения:

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx$$

Обозначают нормальное распределение $X \sim N(\mu, \sigma^2)$.

Стандартным нормальным распределением называется нормальное распределение с математическим ожиданием $\mu = 0$ и стандартным отклонением $\sigma = 1$.

Математическое ожидание μ характеризует положение «центра тяжести» вероятностной массы нормального распределения. Получается, что график плотности распределения случайной величины, имеющей нормальное распределение, симметричен относительно $x = \mu$. Дисперсия σ характеризует разброс значений случайной величины относительно «центра тяжести».

Формализация задачи

Пошаговый подход

Закljučается в последовательном анализе состояний всех блоков системы в момент $t + \Delta t$. Новое состояние определяется в соответствии с их алгоритмическим описанием с учетом действия случайных факторов. В результате этого анализа принимается решение о том, какие системные события должны имитироваться на данный момент времени. Основной недостаток: значительные затраты и опасность пропуска события при больших Δt .

Событийная модель

Состояния отдельных устройств изменяются в дискретные моменты времени. При использовании событийного принципа, состояния всех блоков системы анализируются лишь в момент возникновения какого либо события. Момент наступления следующего события, определяется минимальным значением из списка событий.

Результаты работы

Без повторов, 1000 заявок

Входные данные:

```
a, b = 1, 10
generator = EvenDistribution(a, b)

mu, sigma = 4, 0.2 # диапазон +- [3;5]
processor = NormalDistribution(mu, sigma)

total_tasks = 1000
repeat_percentage = 0
step = 0.01
```

Выходные данные:

Максимальная длина очереди в event_model: 4

Максимальная длина очереди в step_model: 4

10% повторов, 1000 заявок

Входные данные:

```
a, b = 1, 10
generator = EvenDistribution(a, b)

mu, sigma = 4, 0.2 # диапазон +- [3;5]
processor = NormalDistribution(mu, sigma)

total_tasks = 1000
repeat_percentage = 10
step = 0.01
```

Выходные данные:

Максимальная длина очереди в event_model: 6

Максимальная длина очереди в step_model: 7

10% повторов, 10000 заявок

Входные данные:

```
a, b = 1, 10
generator = EvenDistribution(a, b)

mu, sigma = 4, 0.2 # диапазон +- [3;5]
processor = NormalDistribution(mu, sigma)

total_tasks = 10000
repeat_percentage = 10
step = 0.01
```

Выходные данные:

Максимальная длина очереди в event_model: 7

Максимальная длина очереди в step_model: 7

50% повторов, 10000 заявок

Входные данные:

```
a, b = 1, 10
generator = EvenDistribution(a, b)

mu, sigma = 4, 0.2 # диапазон +- [3;5]
processor = NormalDistribution(mu, sigma)

total_tasks = 10000
repeat_percentage = 50
step = 0.01
```

Выходные данные:

Максимальная длина очереди в event_model: 2271

Максимальная длина очереди в step_model: 2217

100% повторов, 10000 заявок

Входные данные:

```
a, b = 1, 10
generator = EvenDistribution(a, b)

mu, sigma = 4, 0.2 # диапазон +- [3;5]
processor = NormalDistribution(mu, sigma)

total_tasks = 10000
repeat_percentage = 100
step = 0.01
```

Выходные данные:

Максимальная длина очереди в event_model: 7223

Максимальная длина очереди в step_model: 7365

Равномерный закон ОА, 0% повторов, 10000 заявок

Входные данные:

```
generator = EvenDistribution(5, 6)
processor = EvenDistribution(3, 6)

total_tasks = 10000
repeat_percentage = 0
step = 0.01
```

Выходные данные:

Максимальная длина очереди в event_model: 1

Максимальная длина очереди в step_model: 1

Равномерный закон ОА, 100% повторов, 10000 заявок

Входные данные:

```
generator = EvenDistribution(5, 6)
processor = EvenDistribution(3, 6)

total_tasks = 10000
repeat_percentage = 100
step = 0.01
```

Выходные данные:

Максимальная длина очереди в event_model: 8173

Максимальная длина очереди в step_model: 8199