Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное 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 (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

по дисциплине «Моделирование»

«Дискретно-событийное моделирование системы обслуживания заявок»

Группа ИУ7-71Б

Студент _____ Лукьяненко В.А.
подпись, дата *фамилия, и.о.*

Преподаватель _____ Рудаков И. В.
подпись, дата *фамилия, и.о.*

Оценка _____

2025г

Состояние системы меняется только в моменты наступления событий. В имитационной схеме используется упорядоченный список событий, содержащий записи вида

$$e = (\ t,\ \text{type},\ \text{actor}\ ),$$

где t - момент времени, в который произойдет событие; type - тип события; actor - объект модели, для которого это событие относится.

Случайные временные интервалы

Основой модели являются случайные величины, задающие моменты поступления заявок и длительности обслуживания. Наиболее часто применяются равномерные распределения

$$X \sim U(a,b), \qquad f_X(x) = \frac{1}{b-a}, \quad x \in [a,b],$$

которые используются для генерации:

- интервалов между поступлениями заявок;
- времени обслуживания операторов.

Если X - случайное время обслуживания или задержки, тогда следующий момент события определяется формулой

$$t_{\text{next}} = t_{\text{current}} + X.$$

Генерация заявок

Генератор формирует последовательность поступлений требований. Пусть X_i - случайный интервал между i-й и $(i+1)$-й заявкой. Тогда времена поступлений представлены рекуррентной формулой

$$T_{i+1} = T_i + X_i.$$

Каждое поступление порождает событие типа `GEN`.

Работа операторов

Оператор является обслуживающим устройством с одним рабочим местом. Время обработки заявки оператором - случайная величина Y с заданным распределением. Если оператор свободен, он начинает обслуживание немедленно:

$$t_{\text{finish}} = t_{\text{now}} + Y.$$

По достижении момента t_{finish} происходит событие `OP_FINISH`. Если оператор занят в момент поступления заявки и система не предусматривает ожидания на данном этапе, заявка отклоняется.

Обслуживающие процессоры

После операторов заявки передаются на вычислительные устройства, каждое из которых обрабатывает одну заявку в течение фиксированного или случайного времени Z:

$$t_{\text{finish}} = t_{\text{now}} + Z.$$

Событие окончания обработки - `PC_FINISH`. Если очередь непуста, процессор немедленно начинает обработку следующей заявки.

Алгоритм выбора следующего события

Пусть $E = \{e_1, e_2, \ldots, e_n\}$ - список событий, упорядоченный по возрастанию времени:

$$t(e_1) \leq t(e_2) \leq \cdots \leq t(e_n).$$

Основной цикл моделирования:

1. Извлекается событие e_1 с минимальным временем.
2. Устанавливается текущее время $t_{\text{now}} = t(e_1)$.
3. Обрабатывается логика данного события.
4. Генерируются новые события, формируются соответствующие t_{new}.

5. Новые события вставляются в список E с сохранением порядка.

Вероятность отказа

Заявка считается отказанной, если в момент её поступления все операторы заняты и система не допускает ожидания на этой стадии. Пусть R - число отказов, N - число обработанных заявок. Тогда оценка вероятности отказа равна

$$P_{\text{отк}} = \frac{R}{N + R}.$$

Данная величина служит показателем пропускной способности системы и её способности справляться с заданной нагрузкой.

Формирование нагрузки в системе

Нагрузка на операторы определяется математическим ожиданием случайного времени обслуживания и интенсивностью поступления заявок:

$$\rho = \lambda \cdot \mathbb{E}[Y],$$

где λ - интенсивность входного потока (обратная величина к $\mathbb{E}[X]$). При $\rho \geq 1$ система гарантированно перегружается, что приводит к росту отказов и увеличению длины очередей.

Итоговое описание процесса

Таким образом, модель представляет собой последовательность событий:

$$\texttt{GEN} \rightarrow \texttt{OP\_FINISH} \rightarrow \texttt{PC\_FINISH}.$$

Каждое событие порождает следующее, формируя стохастическую цепочку обработки требований.