ФГБОУ ВО Государственное бюджетное образовательное учреждение

«Уфимский университет науки и технологий»

Институт информатики, математики и робототехники

Кафедра вычислительной математики и кибернетики

Лабораторная работа

по дисциплине «Компьютерное моделирование»

**ИМИТАЦИЯ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ С ДВУМЯ УСТРОЙСТВАМИ СОЕДИНЕННЫМИ ПАРАЛЛЕЛЬНО**

Выполнил: студенты группы …

…

Проверил преподаватель:

…

УФА 2023

**Постановка задачи:** Предприятие производит некоторую однородную продукцию и имеет собственный магазин по ее продаже с 2 продавцами, а так же склад для ее хранения. Часть запасов продукции со склада поступает в собственный магазин, другая часть в торговые организации. Продукция со склада или магазина доставляется клиентам автомобильными транспортными средствами одинаковой грузоподъемностью

**Требуется:** Проанализировать работу магазина (задача СМО для двух устройств(два продавца), которые соединины параллельны); учесть время работы, количество посетителей в день/час (для интенсивности); изучить предметную область для корректного подбора параметров; посчитать различные оценки интенсивности (среднее число времени в очереди, среднее число времени в устройстве и т.д.); определить наилучший маршрут доставки продукции клиентам

1. Постановка задачи
2. Алгоритм
3. Пример вычисления оценок и гистограмма
4. Верификация программы
5. Сама программа

**Постановка задачи:**

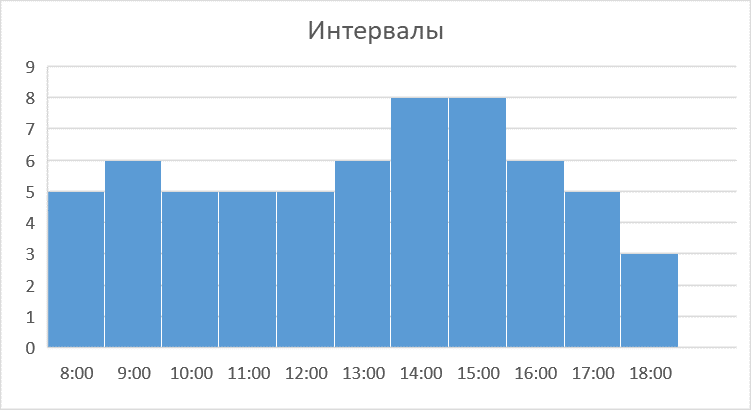
Моделирование системы массового обслуживания будет осуществлено для Уфимского хлебобулочного завода №7. Данный завод производит однородную продукцию – хлебобулочные изделия. У завода имеется свой склад и магазин. В среднем в сутки завод вырабатывает 220,8 продукции. 30% от продукции поставляется оптовым агентам, остальную часть вырабатывает в магазине.

Магазин при заводе имеет следующий график:

Время работы магазина – ежедневно с 8:00 до 19:00 (11 часов)

* Устройство обслуживает клиентов в том порядке, в каком они прибывают (FIFO)
* Имеется фиксированное время , после которого не допускается прибытие новых клиентов в систему, но те клиенты, которые ко времени уже были в системе, должны быть обслужены.
* Время обслуживания клиента – случайная переменная (независимая от всех других времен обслуживания и от процесса прибытия клиентов), подчиненная закону распределения .

| время | среднее кол-во |
| --- | --- |
| 08:00-09:00 | 5 |
| 09:00-10:00 | 6 |
| 10:00-11:00 | 5 |
| 11:00-12:00 | 5 |
| 12:00-13:00 | 5 |
| 13:00-14:00 | 6 |
| 14:00-15:00 | 8 |
| 15:00-16:00 | 8 |
| 16:00-17:00 | 6 |
| 17:00-18:00 | 5 |
| 18:00-19:00 | 3 |



**Функция интенсивности**

*Входные данные*: t – время в системе.   
*Выходные данные:* Значение функции интенсивности в момент времени t

Выбор основывался на распределении среднего количества посетителей в течении всего дня по часам. С помощью гистограммы представленной выше производится выбор функции интенсивности по времени в системе.

8 человек в час

**Переменные**

– переменная времени

***Количественные переменные:***

– число прибывших клиентов к моменту времени

- число уходов клиентов к моменту времени

***Переменная состояния:***

**-** число клиентов в системе к моменту времени

***События***: пребывания и уходы клиентов.

***Список событий***:

- время следующего прибытия клиента (после );

- время завершения работы устройства после обслуживания клиента

(, если никакой клиент в данный момент не обслужен).

***Случайные переменные***:

— время поступления клиента ;

— время между поступлениями клиентов *i -* 1 и ();

— время, потраченное устройством на обслуживание клиента *i*

(без учета времени задержки клиента в очереди);

— время задержки клиента *i* в очереди;

— время ухода клиента *i* по завершении обслуживания;

— время возникновения события j любого типа.

**Выходные переменные:**

- время прибытия клиента

- время ухода клиента

- время после , когда последний клиент уходит

**-** количество времени, в течение которого клиент проводит в системе

- оценка среднего времени, в течение которого клиент проводит в системе

- наблюдаемое значение (время после , когда ушел последний клиент)   
 - среднее время, в течение которого клиент проводит, – оценка ожидаемой средней задержки клиентов в очереди

- число клиентов в системе в период времени [0, T]

**Используемые оценки:**

**Среднее время клиента в системе:**

,

где - это количество времени, в течение которого клиент проводит в системе; - общее число клиентов, которые пришли до времени .

**Средняя задержка клиентов в очереди:**

,

где - время задержки клиента в очереди, - число клиентов, которые прибыли в период времени

**Среднее число клиентов в очереди:**

где - число клиентов, которые побывали в очереди в период времени

**Ожидаемое число клиентов в очереди:**

,

где - число клиентов в системе в период времени .

**Среднее время обслуживания клиентов:**

,

где - время обслуживания клиента.

**Оценка занятости устройств:**

,

где - время обслуживания клиентов в период времени

**Используемые функции для генерации случайных значений.**

**Функция генерации неоднородного пуассоновского процесса***Входные данные:* s **–** время в системе L – интенсивность появления событий   
*Выходные данные:* t – время следующего события в системе.

Алгоритм:

1. Присвоить t = s
2. Генерируем случайную переменную U1 = Uniform[0, 1]
3. Присвоить
4. Генерируем случайную переменную U2 = Uniform[0, 1], независимую от U1.
5. Если , то:
   1. Ts = t
   2. Конец алгоритма
6. Иначе повторить с шага 2.

**Функция генерации времени обслуживания по экспоненциальному закону***Входные данные:* L – интенсивность появления событий   
*Выходные данные:* Y – значение показателей случайной величины

Алгоритм:

1. Генерируем случайное число U = Uniform[0, 1]
2. Возвращаем величину

**СХЕМА ИМИТАЦИОННОГО АЛГОРИТМА**

Задать - время начала работы системы

Задать - время конца работы системы

Создать пустые списки AI, DI1, DI2, VI1, VI2, WI, NA, ND, SI, Ai, Di1, Di2, Vi1, Vi2, Wi, Na, Nd, Si.

Создать переменные Na, Nd, n, c1, c2, t и присвоить им значения 0.

Устройствам 1 и 2 присвоить состояния “Не занят”.

Создать пустой список клиентов.

**Повторить цикл i раз:**

Сгенерировать (генерация неоднородного пуассоновского процесса)

**Пока < T выполнять:**

**Если < t1 или < t2:**

Записать время прихода клиента в список Ai

Записать клиента в список клиентов с состоянием 0 (только пришел)

(время следующего прибытия клиента)

Сгенерировать время прихода следующего клиента (генерация неоднородного пуассоновского процесса )

=

**Если устройства 1 и 2 свободны:**

Сгенерировать время обслуживания

- время завершения работы устройства 1

n = n + 1 (появился новый клиент в системе)

Устройству 1 присвоить состояние “Занят”

Записать время обслуживания клиента в список Vi1

Записать время ухода клиента в список Di1

Записать время нахождения в системе клиента - в список Si

Записать время ожидания клиента в список Wi (Wi = 0)

Присвоить клиенту в состоянии 0 - состояние 1 (Обслуживается первым устройством)

**Если же устройство 1 занято, а 2 свободно:**

Сгенерировать время обслуживания

- время завершения работы устройства 2

n = n + 1 (появился новый клиент в системе)

Устройству 2 присвоить состояние “Занят”

Записать время обслуживания клиента в список Vi2

Записать время ухода клиента в список Di2

Записать время нахождения в системе клиента - в список Si

Записать время ожидания клиента в список Wi (Wi = 0)

Присвоить клиенту в состоянии 0 - состояние 2 (Обслуживается вторым устройством)

**Если же устройство 1 свободно, а 2 занято:**

Сгенерировать время обслуживания

- время завершения работы устройства 1

n = n + 1 (появился новый клиент в системе)

Устройству 1 присвоить состояние “Занят”

Записать время обслуживания клиента в список Vi1

Записать время ухода клиента в список Di1

Записать время нахождения в системе клиента - в список Si

Записать время ожидания клиента в список Wi (Wi = 0)

Присвоить клиенту в состоянии 0 - состояние 1 (Обслуживается первым устройством)

**Если же устройство 1 и устройство 2 заняты:**

n = n + 1 (появился новый клиент в системе)

Записать в список Wi время прихода клиента

Присвоить клиенту в состоянии 0 - состояние 3 (Ожидает в очереди)

**Если < и :**

Подсчитать количество обслуженных клиентов устройства 1 и в целом: c1 = c1 + 1; Nd = Nd + 1

Устройству 1 присвоить состояние “Не занят”

**Если n = 1 или n = 2:**

n = n - 1 (клиент вышел из системы)

**=**

Клиенту в состоянии 1 присвоить состояние 4 (Обслужен)

**Если же n > 2:**

n = n - 1 (клиент вышел из системы)

Клиенту в состоянии 1 присвоить состояние 4 (Обслужен)

Сгенерировать время обслуживания

- время завершения работы устройства 1

**Если ни одно из значений в списке Di1 не превышает время конца рабочего дня (T):**

Устройству 1 присвоить состояние “Занят”

Записать время обслуживания клиента в список Vi1

Записать время ухода клиента в список Di1

В списке Wi изменить значение времени ожидания для клиента с состоянием 3 вычтя из него время (обслуживания предыдущего клиента)

Записать время нахождения в системе клиента в список Si как - + Wi(для клиента с состоянием 3)

Присвоить клиенту с состоянием 3 - состояние 1

**Если < и :**

Подсчитать количество обслуженных клиентов устройства 2 и в целом: c2 = c2 + 1; Nd = Nd + 1

Устройству 2 присвоить состояние “Не занят”

**Если n = 1 или n = 2:**

n = n - 1 (клиент вышел из системы)

**=**

Клиенту в состоянии 2 присвоить состояние 4 (Обслужен)

**Если же n > 2:**

n = n - 1 (клиент вышел из системы)

Клиенту в состоянии 2 присвоить состояние 4 (Обслужен)

Сгенерировать время обслуживания

- время завершения работы устройства 2

**Если ни одно из значений в списке Di2 не превышает время конца рабочего дня (T):**

Устройству 2 присвоить состояние “Занят”

Записать время обслуживания клиента в список Vi2

Записать время ухода клиента в список Di2

В списке Wi изменить значение времени ожидания для клиента с состоянием 3 вычтя из него время (обслуживания предыдущего клиента)

Записать время нахождения в системе клиента в список Si как - + Wi(для клиента с состоянием 3)

Присвоить клиенту с состоянием 3 - состояние 2

**Пока устройство 1 занято и устройство 2 занято:**

**Если устройство 1 занято и в системе есть клиент с состоянием 1:**

Подсчитать количество обслуженных клиентов устройства 1 и в целом: c1 = c1 + 1; Nd = Nd + 1

Устройству 1 присвоить состояние “Не занят”

**Если n = 1 или n = 2:**

n = n - 1 (клиент вышел из системы)

**=**

Клиенту в состоянии 1 присвоить состояние 4 (Обслужен)

**Если же n > 2:**

n = n - 1 (клиент вышел из системы)

Клиенту в состоянии 1 присвоить состояние 4 (Обслужен)

**Если ни одно из значений в списке Di1 не превышает время конца рабочего дня (T):**

Устройству 1 присвоить состояние “Занят”

Сгенерировать время обслуживания

- время завершения работы устройства 1

Записать время обслуживания клиента в список Vi1

Записать время ухода клиента в список Di1

В списке Wi изменить значение времени ожидания для клиента с состоянием 3 вычтя из него время (обслуживания предыдущего клиента)

Записать время нахождения в системе клиента в список Si как - + Wi(для клиента с состоянием 3)

Присвоить клиенту с состоянием 3 - состояние 1

**Если устройство 2 занято и в системе есть клиент с состоянием 2:**

Подсчитать количество обслуженных клиентов устройства 2 и в целом: c2 = c2 + 1; Nd = Nd + 1

Устройству 2 присвоить состояние “Не занят”

**Если n = 1 или n = 2:**

n = n - 1 (клиент вышел из системы)

**=**

Клиенту в состоянии 2 присвоить состояние 4 (Обслужен)

**Если же n > 2:**

n = n - 1 (клиент вышел из системы)

Клиенту в состоянии 2 присвоить состояние 4 (Обслужен)

**Если ни одно из значений в списке Di2 не превышает время конца рабочего дня (T):**

Устройству 2 присвоить состояние “Занят”

Сгенерировать время обслуживания

- время завершения работы устройства 2

Записать время обслуживания клиента в список Vi2

Записать время ухода клиента в список Di2

В списке Wi изменить значение времени ожидания для клиента с состоянием 3 вычтя из него время (обслуживания предыдущего клиента)

Записать время нахождения в системе клиента в список Si как - + Wi(для клиента с состоянием 3)

Присвоить клиенту с состоянием 3 - состояние 2

**Если в списке клиентов нет клиентов, находящихся в состоянии 1, нет клиентов, находящихся в состоянии 2, а также число клиентов в системе n > 0:**

**Пока в списке клиентов есть клиенты, находящиеся в состоянии 3 (ожидают):**

В списке Wi изменить значение времени ожидания для клиента с состоянием 3 вычтя из него время T (конца рабочего дня)

Записать время нахождения в системе клиента в список Si как Wi(для клиента с состоянием 3)

Присвоить клиенту с состоянием 3 - состояние 4 (вышел из системы)

Записать в списки AI, DI1, DI2, VI1, VI2, WI, NA, ND, SI значения Ai, Di1, Di2, Vi1, Vi2, Wi, Na, Nd, Si соответственно.

Списки Ai, Di1, Di2, Vi1, Vi2, Wi, Si очистить

Na = Nd = n = c1 = c2 = 0

Присвоить устройствам 1 и 2 состояние “Не занят”

Список клиентов очистить

Для вычисления среднего времени клиента в системе значения Si для каждого дня просуммировать и разделить на их количество. После получения значения для каждого дня, также просуммировать их и разделить на количество дней.

Для вычисления средней задержки клиентов в очереди значения Wi для каждого дня просуммировать и разделить на их количество минус 1. После получения значения для каждого дня, также просуммировать их и разделить на количество дней.

Для вычисления среднего числа клиентов в очереди найти количество ненулевых значений Wi для каждого дня просуммировать и разделить на их количество. После получения значения для каждого дня, также просуммировать их и разделить на количество дней.

Для вычисления ожидаемого числа клиентов в очереди значения Si для каждого дня просуммировать и разделить на время работы системы. После получения значения для каждого дня, также просуммировать их и разделить на количество дней.

Для вычисления оценки занятости устройств значения Vi1 и Vi2 для каждого дня просуммировать и разделить на время работы системы. После получения значения для каждого дня и устройства, также просуммировать их и разделить на количество дней, а затем сложить значения по каждому устройству.

Для вычисления среднего времени обслуживания устройств значения Vi1 и Vi2 для каждого дня просуммировать и разделить на их количество. После получения значения для каждого дня и устройства, также просуммировать их и разделить на количество дней, а затем найти среднее значение из двух устройств.

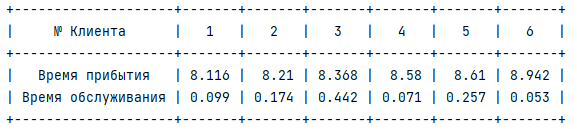
В результате работы алгоритма получим следующие данные:

* Приходы клиентов: массив A, содержащий время поступления каждого клиента.
* Уходы клиентов: массивы DI1 и DI2, содержащие время ухода клиентов с устройств 1 и 2 соответственно.
* Время обслуживания клиентов: массивы VI1 и VI2, содержащие время обслуживания клиентов на устройствах 1 и 2 соответственно.
* Время нахождения в очереди: массив WI, содержащий время ожидания клиентов в очереди на обслуживание.
* Время нахождения в системе: массив SI, содержащий время нахождения клиентов в системе.

**Верификация программы (…):**

Период моделирования T = 1 час (с 8:00 до 9:00), СМО имеет 2 устройства k = 1,2 соединенных параллельно, программа генерирует данные (см. Табл. 1):

Данные моделирования, k = 1,2 Таблица 1





По результатам программы, среднее время, проведенное этими шестью клиентами в системе равно 0,18967

Рассчитаем вручную процесс системы массового обслуживания на этих данных (см. Табл. 2):

Распределение времени при имитации СМО с устройствами k = 1,2 Таблица 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № клиента | Время прибытия | Время начала обслуживания | Время ухода | Время в системе |
| 1 | 8.116 | 8.116 k = 1 | 8.116 + 0.099 = 8.215  k = 1 свободно | 0.099 |
| 2 | 8.210 | 8.210 < 8.215 № 2 пошел на k = 2 (FIFO) | 8.210 + 0.174 = 8.384  k = 2 свободно | 0.174 |
| 3 | 8.368 | 8.215 < 8.384 № 3 пошел на k = 1 (FIFO) | 8.368 + 0.442 = 8.810  k = 1 свободно | 0.442 |
| 4 | 8.580 | 8.384 < 8.810 № 4 пошел на k = 2 (FIFO) | 8.580 + 0.071 = 8.650  k = 2 свободно | 0.071 |
| 5 | 8.610 | 8.650 < 8.810 № 5 пошел на k = 2 (FIFO) | 8.650 + 0.257 = 8.907  k = 2 свободно | 8.650 - 8.610 = 0.04 (Время ожидания обслуживания)  0.257 + 0.04 = 0.297 |
| 6 | 8.942 | 8.810 < 8.907 № 6 пошел на k = 1 (FIFO) | 8.942 + 0.053 = 8.995  k = 1 свободно | 0.053 |

Рассчитаем среднее время, проведенное в системе шестью клиентами:

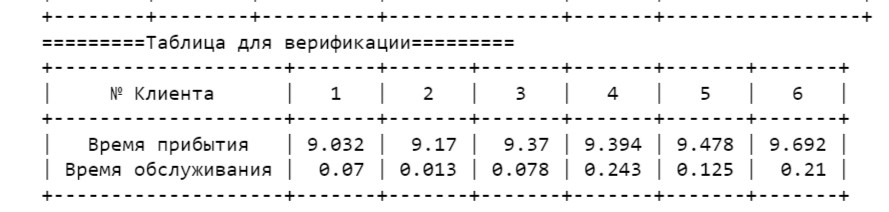
Вычислим погрешность:

Расхождения значений возникли из-за погрешности округления. Погрешность не превышает уровень точности 0.001, что говорит об успешной верификации программы.

**Верификация программы (…)**

Период моделирования T = 1 час (с 9:00 до 10:00), СМО имеет 2 устройства k = 1,2 соединенных параллельно, программа генерирует данные (см. Табл. 3):

Данные моделирования, k = 1,2 Таблица 3





Рассчитаем вручную процесс системы массового обслуживания на этих данных (см. Табл. 4):

Распределение времени при имитации СМО с устройствами k = 1,2 Таблица 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № клиента | Время прибытия | Время начала обслуживания | Время ухода | Время в системе |
| 1 | 9.037 | 9.037 k = 1 | 9.037 + 0.018 = 9.055  k = 1 свободно | 0.018 |
| 2 | 9.375 | 9.375 > 9.055 № 2 пошел на k = 1 (FIFO) | 9.375 + 0.129 = 9.504  k = 1 свободно | 0.129 |
| 3 | 9.574 | 9.574 > 9.504 № 3 пошел на k = 1 (FIFO) | 9.574 + 0.227 = 9.801  k = 1 свободно | 0.227 |
| 4 | 9.581 | 9.581 < 9.801 № 4 пошел на k = 2 (FIFO) | 9.581 + 0.311 = 9.892  k = 2 свободно | 0.311 |
| 5 | 9.848 | 9.848 < 9.892 № 5 пошел на k = 1 (FIFO) | 9.848 + 0.025 = 9.873  k = 1 свободно | 0.025 |
| 6 | 9.849 | 9.874 < 9.892 № 6 пошел на k = 1 (FIFO) | 9.692 + 0.087 = 9.902  k = 1 свободно | 9.874–9.849 = 0.025  (Время ожидания) 0.025 + 0.087 = 0.112 |
| 7 | 9.853 | 9.892 < 9.902 № 7 пошел на k = 2 (FIFO) | 9.892 + 0.038 = 9.902  k = 2 свободно | 9.892–9.853 = 0.039  (Время ожидания)  0.039 + 0.038 = 0.077 |

Рассчитаем среднее время проведенное в системе шестью клиентами:

Вычислим погрешность:

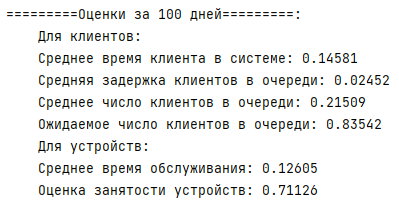
Расхождения значений возникли из-за погрешности округления. Погрешность не превышает уровень точности 0.001, что говорит об успешной верификации программы.

**ВЫВОД**

В ходе выполнения лабораторной работы была построена модель системы массового обслуживания с двумя устройствами, соединенными параллельно, очередь в которые регулировалась методом FIFO.

Для модели была проведена верификация, основанная на сравнении значений, которые выводит программа, и вручную рассчитанных значений. Представленная программа успешно прошла верификацию, что свидетельствует о правильном порядке моделирования.

В ходе моделирования системы СМО для периода в 100 дней были получены следующие значения:



В ходе моделирования системы СМО для периода в 100 000 дней были получены следующие значения:

