|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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**

|  |  |
| --- | --- |
| **по лабораторной работе №** | **5** |

**Название:**

«Определение вероятности отказа»

**Дисциплина:** Моделирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-76Б |  |  | А. А. Петрова |
|  | (Группа) |  |  | (И.О. Фамилия) |
|  |  |  |  |  | |  |
| Преподаватель |  |  |  | И. В. Рудаков |
|  |  |  |  | (И.О. Фамилия) |

2022 г.

**Задание**

В информационный центр приходят клиенты через интервалы времени 10 ± 2 минуты. Если все три имеющихся оператора заняты, клиенту отказывают в обслуживании. Операторы имеют разную производительность и могут обеспечивать обслуживание среднего запроса пользователя за 20 ± 5, 40 ± 10 и 40 ± 20 минут. Клиенты стремятся занять свободного оператора с максимальной производительностью. Полученные запросы сдаются в приемный накопитель, откуда они выбираются для обработки. На первый компьютер – запросы от первого и второго оператора, на второй – от третьего оператора. Время обработки на первом и втором компьютере равны соответственно 15 и 30 минутам. Смоделировать процесс обработки 300 запросов и определить вероятность отказа. За единицу системного времени выбрать 0,01 минуты. Построить структурную схему модели и модель в терминах СМО.

**Математическая формализация**

На рисунке ниже представлена структурная схема моделируемой системы.

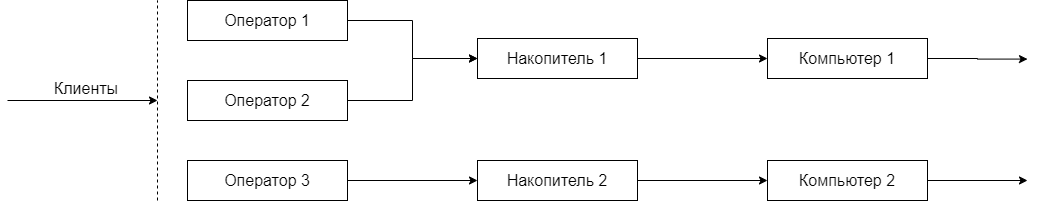


Рисунок 1: структурная схема модели

В процессе взаимодействия клиента с информационным центром возможны два режима:

1. Режим нормального обслуживания (клиент выбирает одного из свободных операторов, отдавая предпочтение тому, у которого меньше номер).
2. Режим отказа в обслуживании клиента (все операторы заняты).

Эндогенные переменные: время обработки задания i-ым оператором и время решения задания на j-ом компьютере.

Экзогенные переменные: число обслуженных клиентов N0 и число клиентов, получивших отказ, N1.

Вероятность отказа:

Ниже представлена схема модели в терминах СМО.

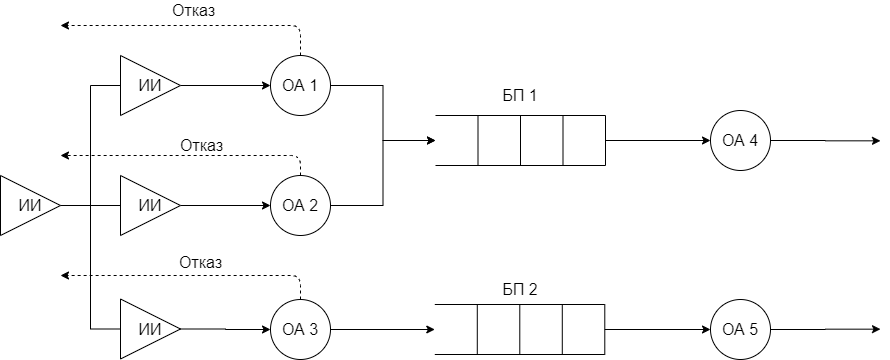


Рисунок 2: модель в терминах СМО

Где ИИ – это источник информации (заявка), БП – буферная память (приемный накопитель), а ОА – обслуживающий аппарат (операторы и компьютеры).

**Реализация**

В листингах ниже представлена реализация генератора заявок, оператора и компьютера.

Листинг 1: генератор заявок

|  |
| --- |
| **class** **TimeProcessor:**  **def** \_\_init\_\_**(**self**,** time\_distribution**):**  self**.**time\_distribution **=** time\_distribution  self**.**remaining\_time **=** 0  **def** update\_time**(**self**):**  **if** self**.**remaining\_time **>** 0**:**  self**.**remaining\_time **-=** TIME\_DELTA  **if** self**.**remaining\_time **<=** 1e-5**:**  self**.**remaining\_time **=** self**.**time\_distribution**.**get\_time**()**  **return** Request**()**  **return** **None** |

Листинг 2: оператор

|  |
| --- |
| **class** **Operator:**  **def** \_\_init\_\_**(**self**,** recipient**,** time\_distribution**):**  self**.**time\_distribution **=** time\_distribution  self**.**recipient **=** recipient  self**.**remaining\_time **=** 0  self**.**is\_busy **=** **False**  self**.**processing\_request **=** **None**  **def** update\_time**(**self**):**  self**.**remaining\_time **-=** TIME\_DELTA  **if** self**.**is\_busy **and** self**.**remaining\_time **<=** 1e-5**:**  self**.**finish\_process\_request**()**  **def** start\_process\_new\_request**(**self**,** request**):**  self**.**is\_busy **=** **True**  self**.**processing\_request **=** request  self**.**remaining\_time **=** self**.**time\_distribution**.**get\_time**()**  **def** finish\_process\_request**(**self**):**  self**.**recipient**.**append**(**self**.**processing\_request**)**  self**.**is\_busy **=** **False**  self**.**processing\_request **=** **None** |

Листинг 3: компьютер

|  |
| --- |
| **class** **Processor:**  **def** \_\_init\_\_**(**self**,** requests\_storage**,** time\_distribution**):**  self**.**requests\_storage **=** requests\_storage  self**.**time\_distribution **=** time\_distribution  self**.**is\_busy **=** **False**  self**.**processing\_request **=** **None**  self**.**remaining\_time **=** 0  **def** update\_time**(**self**):**  **if** self**.**remaining\_time **!=** 0**:**  self**.**remaining\_time **-=** TIME\_DELTA  **if** self**.**is\_busy **and** self**.**remaining\_time **<=** 1e-5**:**  self**.**is\_busy **=** **False**  self**.**processing\_request **=** **None**  **return** FINISH\_PROCESS\_REQUEST  **if** **not** self**.**is\_busy **and** **len(**self**.**requests\_storage**)** **!=** 0**:**  self**.**processing\_request **=** self**.**requests\_storage**.**pop**(**0**)**  self**.**remaining\_time **=** self**.**time\_distribution**.**get\_time**()**  self**.**is\_busy **=** **True** |

**Результаты работы**

Ниже представлены результаты работы программы для 300 заявок.

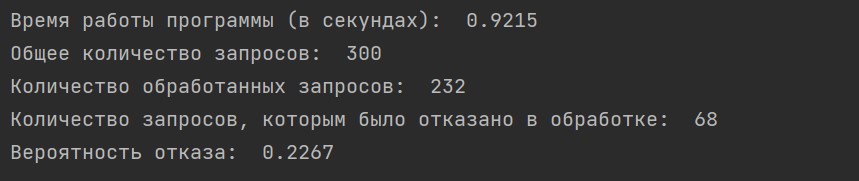
****

Рисунок 3: работа программы при 300 заявках