



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 №5

По курсу: «Моделирование»

Тема: «Определение вероятности отказа»

Студентка ИУ7-75Б
Оберган Т.М
Вариант 14

Преподаватель
Рудаков И.В.

Москва, 2020 г.

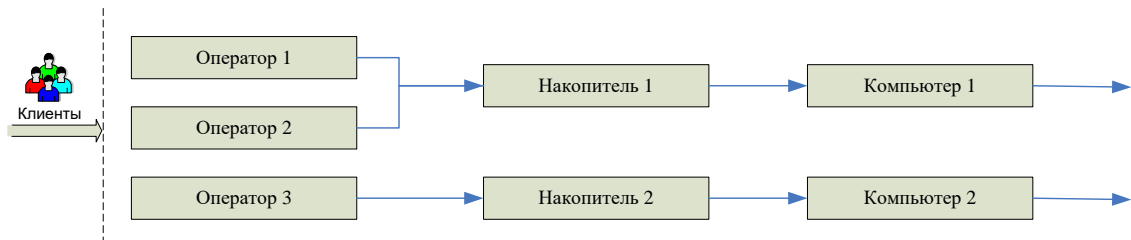
Оглавление

Задание	3
Аналитическая часть	4
Переменные и уравнения имитационной модели	4
Листинг	5
Результаты работы	7
300 заявок	7
1000 заявок	7
3000 заявок	7
Вывод	7

Задание

В информационный центр приходят клиенты через интервал 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 ; 40 ± 20 . Клиенты стремятся занять свободного оператора с максимальной производительностью. Полученные запросы сдаются в накопитель. Откуда выбираются на обработку. На первый компьютер запросы от 1 и 2-ого операторов, на второй – запросы от 3-его. Время обработки запросов первым и 2-м компьютером равны соответственно 15 и 30 мин. Промоделировать процесс обработки 300 запросов.

Для выполнения поставленного задания необходимо создать концептуальную модель в терминах СМО, определить эндогенные и экзогенные переменные и уравнения модели. За единицу системного времени выбрать 0,01 минуты.



Аналитическая часть

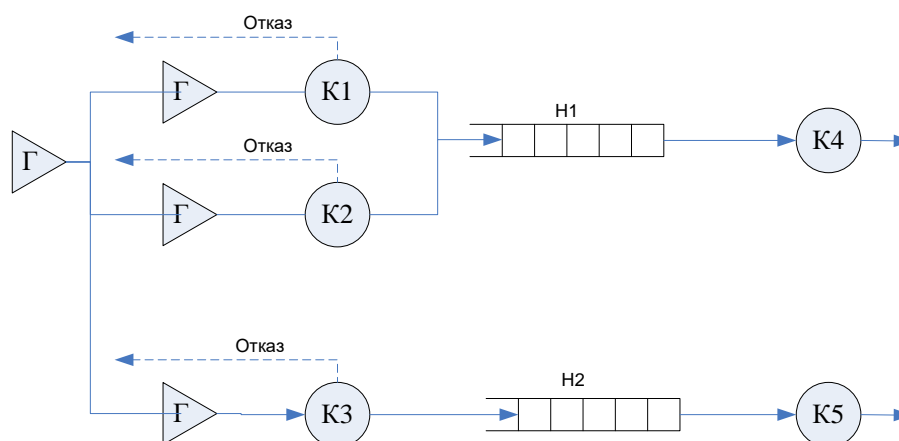
В процессе взаимодействия клиентов с информационным центром возможно:

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

Переменные и уравнения имитационной модели

Эндогенные переменные: время обработки задания i -ым оператором, время решения этого задания j -ым компьютером.

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



$$P_{отк} = \frac{C_{отк}}{C_{отк} + C_{обсл}}$$

Листинг

Листинг 1 – функция обработки инкремента шага по времени

```
19 def one_step(generator, operators, processors, request_info, generate_new=True):
20     # Обновление генератора
21     if generate_new:
22         request = generator.upd_time(unit_of_time)
23         if request:
24             request_info['generated'] += 1
25             i_operator = pick_operator(operators)
26             if i_operator == -1: # все операторы заняты
27                 request_info['lost'] += 1
28             else:
29                 operators[i_operator].accept_request(request)
30
31     # Обновление операторов
32     for cur_operator in operators:
33         cur_operator.upd_time(unit_of_time)
34
35     # Обновление компьютеров
36     for cur_processor in processors:
37         res = cur_processor.upd_time(unit_of_time)
38         if res == 'req fin': # заявка была обработана
39             request_info['processed'] += 1
```

Листинг 2 – циклы, обеспечивающие пошаговую работу системы

```
44 def modeling(generator, operators, processors, total_incoming_requests):
45     request_info = {'generated': 0, 'lost': 0, 'processed': 0}
46
47     # Пока не сгенерируется нужное число заявок
48     while request_info['generated'] < total_incoming_requests:
49         one_step(generator, operators, processors, request_info)
50
51     # Пока все сгенерированные заявки не пройдут систему
52     while request_info['lost'] + request_info['processed'] < total_incoming_requests:
53         one_step(generator, operators, processors, request_info, False)
54
55     return request_info
```

Листинг 3 – задание входных параметров и получение результата

```
58 def main():
59     client_generator = Generator(EvenDistribution(8, 12))
60
61     first_queue = []
62     second_queue = []
63
64     operators = [
65         Operator(first_queue, EvenDistribution(15, 25)), # самый производительный
66         Operator(first_queue, EvenDistribution(30, 50)),
67         Operator(second_queue, EvenDistribution(20, 60)) # наименее производительный
68     ]
69
70     processors = [
71         Processor(first_queue, EvenDistribution(15, 15)), # ровно 15 минут
72         Processor(second_queue, EvenDistribution(30, 30)) # ровно 30 минут
73     ]
74
75     total_requests = 300
76
77     t_start = time()
78     res = modeling(client_generator, operators, processors, total_requests)
79
80     print('time seconds', time() - t_start)
81     for key in res.keys():
82         print(key, res[key])
83
84     print('lost', res['lost'] / total_requests)
```

Результаты работы

300 заявок

```
time seconds 0.5375847816467285
generated 300
lost 66
processed 234
lost 0.22
```

1000 заявок

```
time seconds 1.8780181407928467
generated 1000
lost 212
processed 788
lost 0.212
```

3000 заявок

```
time seconds 5.603071212768555
generated 3000
lost 649
processed 2351
lost 0.21633333333333332
```

Вывод

При 300 заявках, процент потерянных заявок в данной системе равен 22%.