РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

ОТЧЕТ

по лабораторной работе № 14

# дисциплина: Моделирование информационных процессов

Студент: Тейшейра Боа Морте Селмилтон

Группа: НкНбд-01-20

**МОСКВА**

2023 г.

**Цель работа**

Цель лабораторной работы

• Используя теоретические методы анализа сетей Петри, провести анализ сети с

помощью построения дерева достижимости. Определить, является ли сеть

безопасной, ограниченной, сохраняющей, имеются ли тупики.

• Промоделируйте сеть с помощью CPN Tools.

• Вычислить пространство состояний. Сформировать отчет о пространстве

состояний и проанализировать его. Построить граф пространства состояний.

# Выполнение работы

Модель оформления заказов клиентов одним оператором

1. Постановка задачи

В интернет-магазине заказы принимает один оператор.

Интервалы поступления заказов распределены равномерно с интервалом 15 ± 4 мин. Время оформления заказа также распределено равномерно на интервале 10 ± 2 мин. Обработка поступивших заказов происходит в порядке очереди (FIFO). Требуется разработать модель обработки заказов в течение 8 часов.

1. Построение модели

Порядок блоков в модели соответствует порядку фаз обработки заказа в реальной системе:

1) клиент оставляет заявку на заказ в интернет-магазине;

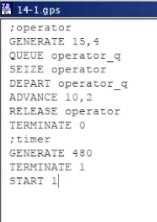
2) если необходимо, заявка от клиента ожидает в очереди освобождения оператора для оформления заказа;

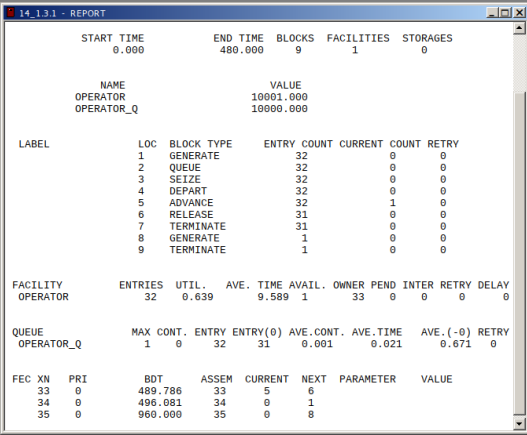
3) заявка от клиента принимается оператором для оформления заказа; 4) оператор оформляет заказ;

5) клиент получает подтверждение об оформлении заказа (покидает систему).

Модель будет состоять из двух частей: моделирование обработки заказов в интернет-магазине и задание времени моделирования. Для задания равномерного распределения поступления заказов используем блок GENERATE, для задания равномерного времени обслуживания (задержки в системе) — ADVANCE. Для моделирования ожидания заявок клиентов в очереди используем блоки QUEUE и DEPART, в которых в качестве имени очереди укажем operator\_q Для моделирования поступления заявок для оформления заказов к оператору используем блоки SEIZE и RELEASE с параметром operator — имени «устройства обслуживания». Таким образом, имеем:

Требуется, чтобы модельное время было 8 часов. Соответственно, параметр блока GENERATE — 480 (8 часов по 60 минут, всего 480 минут). Работа программы начинается с оператора START с начальным значением счётчика завершений, равным 1; заканчивается — оператором TERMINATE с параметром 1, что задаёт ординарность потока в модели. Таким образом, имеем:





Результаты работы модели:

– модельное время в начале моделирования: START TIME=0.0;

– абсолютное время или момент, когда счетчик завершений принял значение 0: END TIME=480.0;

– количество блоков, использованных в текущей модели, к моменту завершения моделирования: BLOCKS=9;

– количество одноканальных устройств, использованных в модели к моменту завершения моделирования: FACILITIES=1;

– количество многоканальных устройств, использованных в текущей модели к моменту завершения моделирования: STORAGES=0.

Имена, используемые в программе модели: operator, operator\_q. Далее идёт информация о блоках текущей модели, в частности,

ENTRY COUNT — количество транзактов, вошедших в блок с начала процедуры моделирования. Затем идёт информация об одноканальном устройстве FACILITY (оператор, оформляющий заказ), откуда видим, что к оператору попало 33 заказа от клиентов (значение поля OWNER=33), но одну заявку оператор не успел принять в обработку до окончания рабочего времени (значение поля ENTRIES=32). Полезность работы оператора составила 0, 639. При этом среднее время занятости оператора составило 9, 589 мин.

Далее информация об очереди:

– QUEUE=operator\_q — имя объекта типа «очередь»;

– MAX=1 — в очереди находилось не более одной ожидающей заявки от клиента;

– CONT=0 — на момент завершения моделирования очередь была пуста;

– ENTRIES=32 — общее число заявок от клиентов, прошедших через очередь в течение периода моделирования;

– ENTRIES(O)=31 — число заявок от клиентов, попавших к оператору без ожидания в очереди;

– AVE.CONT=0, 001 заявок от клиентов в среднем были в очереди;

– AVE.TIME=0.021 минут в среднем заявки от клиентов провели в очереди (с учётом всех входов в очередь);

– AVE.(–0)=0, 671 минут в среднем заявки от клиентов провели в очереди (без учета «нулевых» входов в очередь). В конце отчёта идёт информация о будущих событиях:

– XN=33 — порядковый номер заявки от клиента, ожидающей поступления для оформления заказа у оператора;

– PRI=0 — все клиенты (из заявки) равноправны;

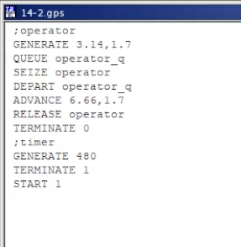
– BDT=489, 786 — время назначенного события, связанного с данным транзактом;

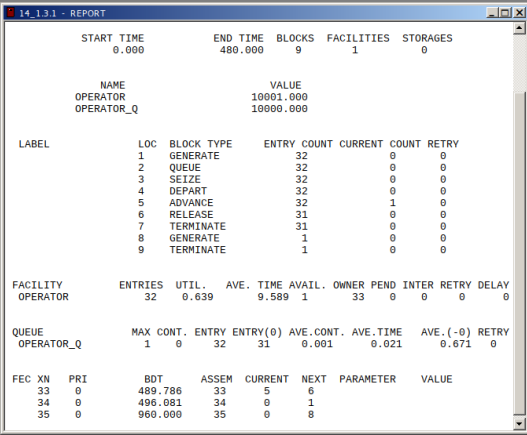
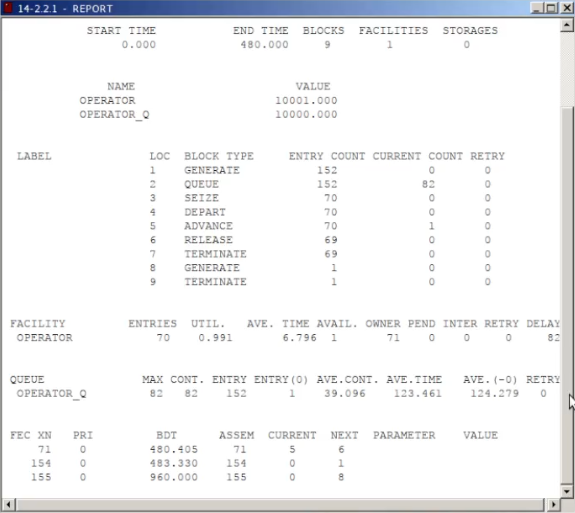
– ASSEM=33 — номер семейства транзактов;

– CURRENT=5 — номер блока, в котором находится транзакт;

– NEXT=6 — номер блока, в который должен войти транзакт.

**Упражнение**. Скорректируйте модель в соответствии с изменениями входных данных: интервалы поступления заказов распределены равномерно с интервалом 3.14 ± 1.7 мин; время оформления заказа также распределено равномерно на интервале 6.66 ± 1.7 мин. Проанализируйте отчёт, сравнив результаты с результатами предыдущего моделирования.





|  |  |
| --- | --- |
| Lab14-1 | Lab14-2 |
| – MAX=1  – CONT=0  – ENTRIES=32  – ENTRIES(O)=31  – AVE.CONT=0, 001  – AVE.TIME=0.021  – AVE.(–0)=0, 671 минут  – XN=33  – PRI=0  – BDT=489, 786  – ASSEM=33  – CURRENT=5  – NEXT=6 | – MAX=82  – CONT=82  – ENTRIES=152  – ENTRIES(O)=1  – AVE.CONT=39.096  – AVE.TIME=123.461  – AVE.(–0)=124.279 минут  – XN=71  – PRI=0  – BDT=480.405  – ASSEM=71  – CURRENT=5  – NEXT=6 |

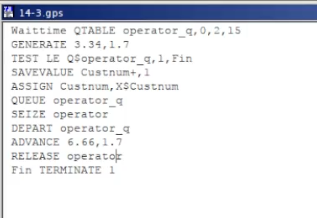
Построение гистограммы распределения заявок в очереди Предположим требуется построить гистограмму распределения заявок, ожидающих обработки в очереди в примере из предыдущего упражнения. Для построения гистограммы необходимо сформировать таблицу значений заявок в очереди, записываемых в неё с определённой частотой. Команда описания такой таблицы QTABLE имеет следующий формат:

Name QTABLE A,B,C,D Здесь Name — метка, определяющая имя таблицы.

Далее должны быть заданы операнды:

А задается элемент данных, чьё частотное распределение будет заноситься в таблицу (может быть именем, выражением в скобках или системным числовым атрибутом (СЧА));

B задается верхний предел первого частотного интервала; С задает ширину частотного интервала — разницу между верхней и нижней границей каждого частотного класса; D задаёт число частотных интервалов. Код программы будет следующим:



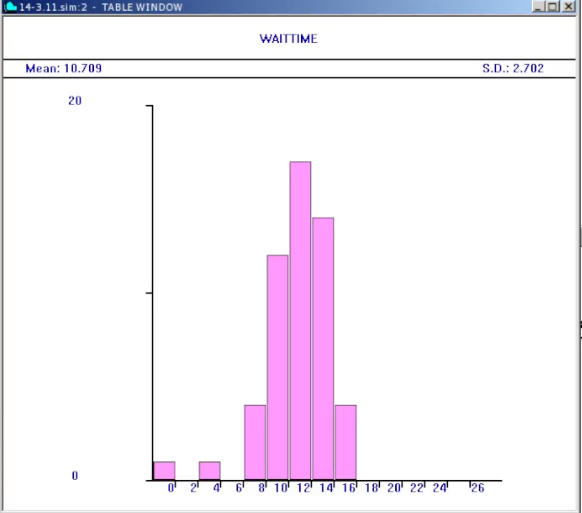
Здесь Waittime — метка оператора таблицы очередей QTABLE, в данном случае название таблицы очереди заявок на заказы. Строка с оператором TEST по смыслу аналогично действиям оператора IF и означает, что если в очереди 0 или 1 заявка, то осуществляется переход к следующему оператору, в данном случае к оператору SAVEVALUE, в противном случае (в очереди более одной заявки) происходит переход к оператору с меткой Fin, то есть заявка удаляется из системы, не попадая на обслуживание. Строка с оператором SAVEVALUE с помощью операнда Custnum подсчитывает число заявок на заказ, попавших в очередь. Далее оператору ASSIGN присваивается значение СЧА оператора Custnum. Для запуска разработанного кода на счёт и построения гистограммы необходимо последовательно выполнить:

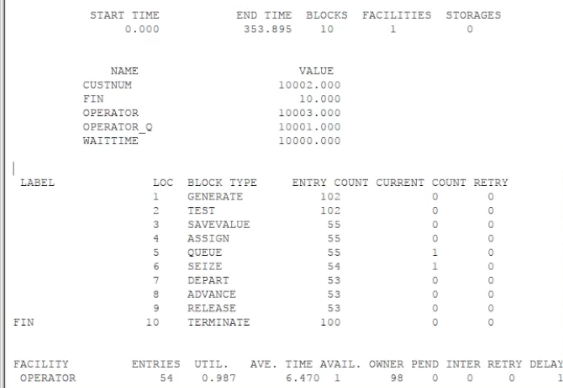
– Command →Create Simulation;

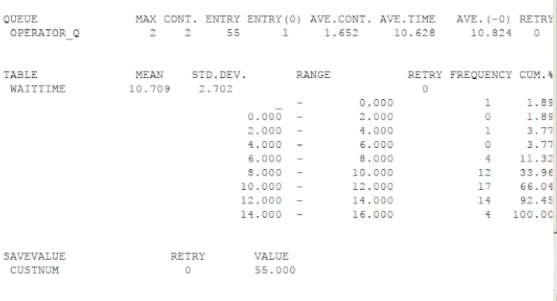
– Command →START, в появившемся окне задать, например, START 100;

– Window →Simulation Window →Table Window.

У нас по оси X это количество заявок по оси Y это сколько раз данное количество заявок было в очереди.





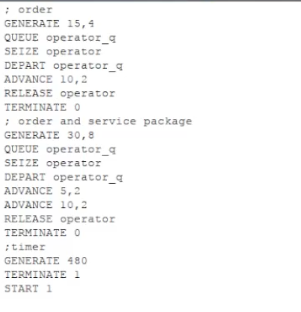


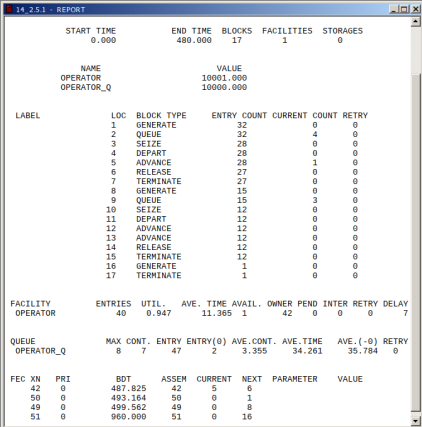
Модель обслуживания двух типов заказов от клиентов в интернет-магазине Постановка задачи

В интернет-магазин к одному оператору поступают два типа заявок от клиентов — обычный заказ и заказ с оформление дополнительного пакета услуг. Заявки первого типа поступают каждые 15 ± 4 мин. Заявки второго типа — каждые 30 ± 8 мин. Оператор обрабатывает заявки по принципу FIFO («первым пришел — первым обслужился»). Время, затраченное на оформление обычного заказа, составляет 10 ± 2 мин, а на оформление дополнительного пакета услуг — 5 ± 2 мин. Требуется разработать модель обработки заказов в течение 8 часов, обеспечив сбор данных об очереди заявок от клиентов.

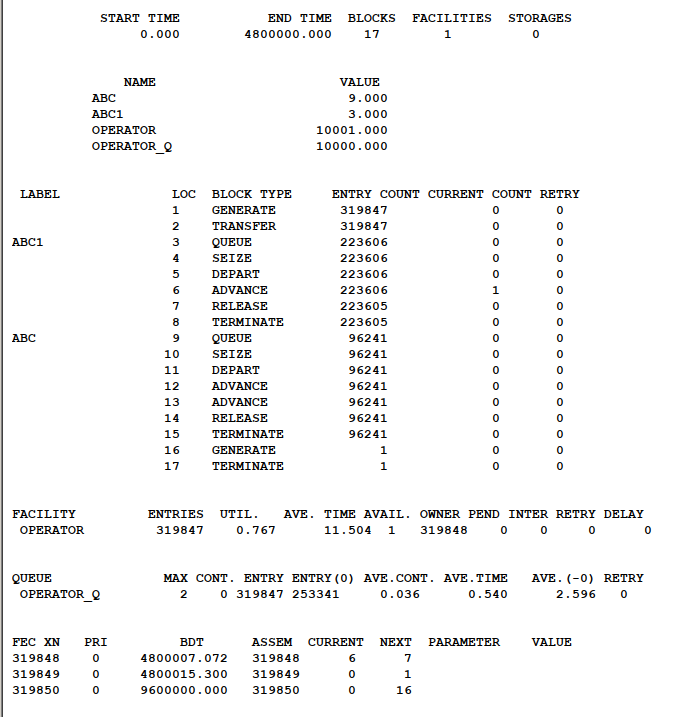
Построение модели

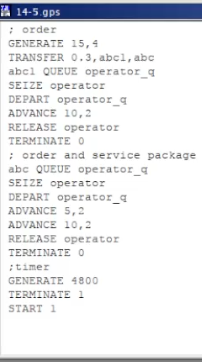
Необходимо реализовать отличие в оформлении обычных заказов и заказов с дополнительным пакетом услуг. Такую систему можно промоделировать с помощью двух сегментов. Один из них моделирует оформление обычных заказов, а второй — заказов с дополнительным пакетом услуг. В каждом из сегментов пара QUEUE–DEPART должна описывать одну и ту же очередь, а пара блоков SEIZE–RELEASE должна описывать в каждом из двух сегментов одно и то же устройство и моделировать работу оператора. Сегмент моделирования оформление обычных заказов:





**Упражнение**. Скорректируйте модель так, чтобы учитывалось условие, что число заказов с дополнительным пакетом услуг составляет 30% от общего числа заказов. Используйте оператор TRANSFER. Проанализируйте отчёт.





Модель оформления заказов несколькими операторами

Постановка задачи

В интернет-магазине заказы принимают 4 оператора. Интервалы поступления заказов распределены равномерно с интервалом 5 ± 2 мин. Время оформления заказа каждым оператором также распределено равномерно на интервале 10 ± 2 мин. Обработка поступивших заказов происходит в порядке очереди (FIFO). Требуется определить характеристики очереди заявок на оформление заказов при условии, что заявка может обрабатываться одним из 4-х операторов в течение восьмичасового рабочего дня.

Построение модели Сегмент моделирования обслуживания клиентов:

operator STORAGE 4

GENERATE 5,2

QUEUE operator\_q

ENTER operator,1

DEPART operator\_q

ADVANCE 10,2

LEAVE operator,1

TERMINATE 0

Сегмент моделирования таймера:

;timer

GENERATE 480

TERMINATE 1

START 1

Задание

1. Проанализируйте полученный отчёт.

– количество блоков, использованных в текущей модели, к моменту завершения моделирования: BLOCKS=9;

– количество одноканальных устройств, использованных в модели к моменту завершения моделирования: FACILITIES=1;

– количество многоканальных устройств, использованных в текущей модели к моменту завершения моделирования: STORAGES=0.

Имена, используемые в программе модели: operator, operator\_q. Далее идёт информация о блоках текущей модели, в частности,

Далее информация об очереди:

– QUEUE=operator\_q — имя объекта типа «очередь»;

– MAX=1 — в очереди находилось не более одной ожидающей заявки от клиента;

– CONT=0 — на момент завершения моделирования очередь была пуста;

– ENTRIES=93 — общее число заявок от клиентов, прошедших через очередь в течение периода моделирования;

– ENTRIES(O)=93 — число заявок от клиентов, попавших к оператору без ожидания в очереди;

– AVE.CONT=0, 000 заявок от клиентов в среднем были в очереди;

– AVE.TIME=0.000 минут в среднем заявки от клиентов провели в очереди (с учётом всех входов в очередь);

– AVE.(–0)=0,000 минут в среднем заявки от клиентов провели в очереди (без учета «нулевых» входов в очередь). В конце отчёта идёт информация о будущих событиях:

– XN=95 — порядковый номер заявки от клиента, ожидающей поступления для оформления заказа у оператора;

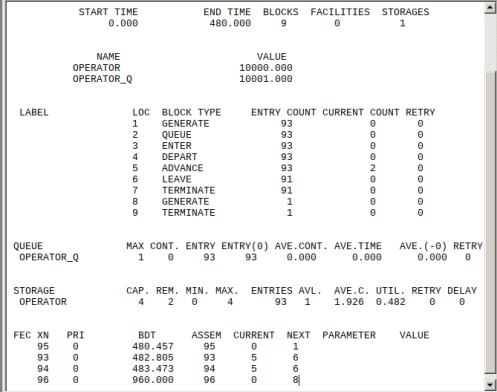
– PRI=0 — все клиенты (из заявки) равноправны;

– BDT=480, 457 — время назначенного события, связанного с данным транзактом;

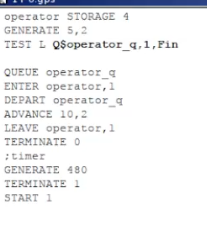
– ASSEM=95 — номер семейства транзактов;

– CURRENT=0 — номер блока, в котором находится транзакт;

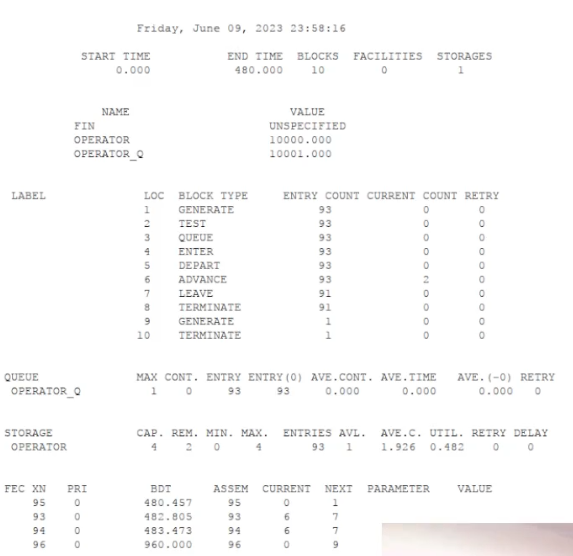
– NEXT=1 — номер блока, в который должен войти транзакт.



1. Измените модель: требуется учесть в ней возможные отказы клиентов от заказа — когда при подаче заявки на заказ клиент видит в очереди более двух других заявок, он отказывается от подачи заявки, то есть отказывается от обслуживания (используйте блок TEST и стандартный числовой атрибут Qj текущей длины очереди j).



1. Проанализируйте отчёт изменённой модели



# Заключение

Выводы по сделанной лабораторной работе.