

Лабораторная работа 14. Модели обработки заказов

14.1. Модель оформления заказов клиентов одним оператором

14.1.1. 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.

14.1.2. 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:

```
;operator
GENERATE 15,4
QUEUE operator_q
SEIZE operator_q
DEPART operator_q
ADVANCE 10,2
RELEASE operator
TERMINATE 0
```

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:

```
;timer
GENERATE 480
TERMINATE 1
START 1
```

14.1.3.1 - REPORT										
START TIME			END TIME		BLOCKS	FACILITIES		STORAGES		
0.000			480.000		9	1		0		
NAME					VALUE					
OPERATOR					10001.000					
OPERATOR_Q					10000.000					
LABEL	LOC	BLOCK	TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY		
	1	GENERATE		32		0	0	0		
	2	QUEUE		32		0	0	0		
	3	SEIZE		32		0	0	0		
	4	DEPART		32		0	0	0		
	5	ADVANCE		32		1	0	0		
	6	RELEASE		31		0	0	0		
	7	TERMINATE		31		0	0	0		
	8	GENERATE		1		0	0	0		
	9	TERMINATE		1		0	0	0		
FACILITY	ENTRIES	UTIL.	AVE.	TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
OPERATOR	32	0.639		9.589	1	33	0	0	0	0
QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.	CONT.	AVE.	TIME	AVE. (-0)	RETRY
OPERATOR_Q	1	0	32	31	0.001		0.021		0.671	0
FEC	XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE		
33	0		489.786	33	5	6				
34	0		496.081	34	0	1				
35	0		960.000	35	0	8				

Рис. 14.1. Отчёт по модели оформления заказов в интернет-магазине

После запуска симуляции получаем отчёт (рис. 14.1).

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 учёта «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.

14.2. 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: A задается элемент данных, чья частотное распределение будет заноситься в таблицу (может быть именем, выражением в скобках или системным числовым атрибутом (СЧА)); B задается верхний предел первого частотного интервала; C задает ширину частотного интервала — разницу между верхней и нижней границей каждого частотного класса; D задаёт число частотных интервалов.

Код программы будет следующим:

```
Waittime  QTABLE      operator_q,0,2,15
          GENERATE    3.34,1.7
          TEST        LE Q$operator_q,1,Fin
          SAVEVALUE    Custnum+,1
          ASSIGN       Custnum,X$Custnum
          QUEUE        operator_q
          SEIZE        operator
          DEPART       operator_q
```

```

ADVANCE    6.66,1.7
RELEASE    operator
Fin        TERMINATE 1

```

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.

Упражнение. Проанализируйте отчёт и гистограмму по результатам моделирования.

14.3. Модель обслуживания двух типов заказов от клиентов в интернет-магазине

14.3.1. Постановка задачи

В интернет-магазин к одному оператору поступают два типа заявок от клиентов — обычный заказ и заказ с оформлением 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.

14.3.2. 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 оформления обычных заказов:

```

; order
GENERATE    15,4
QUEUE       operator_q
SEIZE       operator_q
DEPART      operator_q

```

```
ADVANCE 10,2
RELEASE operator
TERMINATE 0
```

Сегмент моделирования оформления заказов с дополнительным пакетом услуг.
; order and service package

```
GENERATE 30,8
QUEUE operator_q
SEIZE operator_q
DEPART operator_q
ADVANCE 5,2
ADVANCE 10,2
RELEASE operator
TERMINATE 0
```

Сегмент моделирования таймера:

```
;timer
GENERATE 480
TERMINATE 1
START 1
```

После запуска симуляции получаем отчёт (рис. 14.2).

14_2.5.1 - REPORT

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	480.000	17	1	0
NAME		VALUE		
OPERATOR		10001.000		
OPERATOR_Q		10000.000		

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	32		0	0
	2	QUEUE	32		4	0
	3	SEIZE	28		0	0
	4	DEPART	28		0	0
	5	ADVANCE	28		1	0
	6	RELEASE	27		0	0
	7	TERMINATE	27		0	0
	8	GENERATE	15		0	0
	9	QUEUE	15		3	0
	10	SEIZE	12		0	0
	11	DEPART	12		0	0
	12	ADVANCE	12		0	0
	13	ADVANCE	12		0	0
	14	RELEASE	12		0	0
	15	TERMINATE	12		0	0
	16	GENERATE	1		0	0
	17	TERMINATE	1		0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
OPERATOR	40	0.947	11.365	1	42	0	0	0	7

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
OPERATOR_Q	8	7	47	2	3.355	34.261	35.784	0

FEC	XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
42	0		487.825	42	5	6		
50	0		493.164	50	0	1		
49	0		499.562	49	0	8		
51	0		960.000	51	0	16		

Рис. 14.2. Отчёт по модели оформления заказов двух типов

Задание: проанализируйте полученный отчёт.

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.

14.4. Модель оформления заказов несколькими операторами

14.4.1. 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.

14.4.2. 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
```

После запуска симуляции получаем отчёт (рис. 14.3).

14.4.3. Задание

- 1) Проанализируйте полученный отчёт.
- 2) 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`).
- 3) Проанализируйте отчёт изменённой модели.

14_3.13.1 - REPORT

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	480.000	9	0	1

NAME	VALUE
OPERATOR	10000.000
OPERATOR_Q	10001.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	93	0	0
	2	QUEUE	93	0	0
	3	ENTER	93	0	0
	4	DEPART	93	0	0
	5	ADVANCE	93	2	0
	6	LEAVE	91	0	0
	7	TERMINATE	91	0	0
	8	GENERATE	1	0	0
	9	TERMINATE	1	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
OPERATOR_Q	1	0	93	93	0.000	0.000	0.000	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
OPERATOR	4	2	0	4	93	1	1.926	0.482	0	0

FEC	XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
95	0		480.457	95	0	1		
93	0		482.805	93	5	6		
94	0		483.473	94	5	6		
96	0		960.000	96	0	8		

Рис. 14.3. Отчёт по модели оформления заказов несколькими операторами