**Введение**

Целью моей курсовой работы является моделирование системы коммутации данных, определение загрузки устройств, вероятность отказа в обслуживании.

Из постановки задачи видно, что система обработки данных, работу которой необходимо смоделировать является системой массового обслуживания. Поэтому для формализации функционирования этой системы будем использовать непрерывно – стохастическую модель (Q – схему).

Q – схема, описывающая процесс функционирования системы массового обслуживания любой сложности, однозначно задается в виде

Q = <W, U, H, Z, R, A>,

где W – поток входящих заявок, U – поток обслуживания, H – подмножество собственных параметров, Z – подмножество состояний элементов Q – схемы, R – оператор сопряжения элементов структуры, A – оператор алгоритмов обслуживания заявок. В качестве элементов Q – схемы рассматриваются элементы трех типов: И – источник, Н – накопитель, К – канал обслуживания заявок.

Существуют два принципа построения моделирующих алгоритмов, использующих непрерывно – стохастическую модель:

а) Принцип Δt. При построении моделирующего алгоритма по “принципу Δt”, т.е. алгоритма с детерминированным шагом, необходимо для построения адекватной модели определить минимальный интервал времени между соседними событиями Δt=min начало ui конец, где ui – поток обслуживания и поток заявок;

б) Принцип δz. В моделирующих алгоритмах, построенных “по принципу δz”, т.е. в алгоритмах со случайным шагом, элементы Q – схемы просматриваются при моделировании только в моменты особых состояний (в моменты появления заявок из И или изменения состояний К). При этом длительность шага Δt=var и зависит как от особенностей самой системы S, так и от воздействий внешней среды E. Моделирующие алгоритмы со случайным шагом могут быть реализованы синхронным и асинхронным способами. При синхронном способе один из элементов Q – схемы выбирается в качестве ведущего и по нему синхронизируется весь процесс моделирования. При асинхронном способе построения моделирующего алгоритма ведущий элемент не используется, а очередному шагу моделирования (просмотру Q – схемы) может соответствовать любое особое состояние всего множества элементов И, Н и К. При этом просмотр элементов Q – схемы организован так, что при каждом особом состоянии либо просматриваются циклически все элементы, либо спорадически – только те, которые могут изменить своё состояние (просмотр с прогнозированием) [1].

Для рассматриваемой задачи моделирования я буду использовать алгоритм моделирования с детерминированным шагом моделирования. Его использование упрощает моделирование процесса обмена информации в системе передачи данных.

**1 Постановка задачи**

В узел коммутации, состоящий из входного буфера, процессора, двух исходящих буферов, двух выходных линий. Сообщения поступают из двух направлений. Сообщение из первого направления поступает на входящий буфер, обрабатывается в процессоре, буферизируется в выходном буфере первой линии и передается по выходной линии. Сообщение со второй линии обрабатывается аналогично, но передается по второй выходной линии. Применяемый метод контроля потоков требует одновременного присутствия в системе не более 3-ех сообщений на каждом направлении. Сообщения поступают через интервалы 15±7 мс, время обработки в процессоре 7 мс, время передачи по выходным линиям 15±5 мс.

Смоделировать процесс коммутации в течении 10 секунд. Определить загрузку устройств, вероятность отказа в обслуживании, изменение в функции распределения времени передачи при снятии ограничений контроля потоков.

**2 Структурная схема процесса функционирования**

Структурная схема процесса функционирования системы передачи данных представлена на рисунке 1.



Рисунок 1- Структурная схема процесса функционирования

**3 Структурная схема модели в символике Q-схем**



Рисунок 2 - Структурная схема модели в символике Q-схем

На рисунке 2 использованы следующие обозначения:

И – источник, К - канал, Н – накопитель;

Н1– входной буфер;

Н2– буфер линии 1;

Н3– буфер линии 2;

К1– процессор;

К2 – выходная линия 1;

К3 – выходная линия 2.

Источники И1,И2 имитируют приходящие сообщения. Если накопитель Н1, заполнен, а с источника И1(И2) поступает очередная заявка, то открывается клапан 4(7) и задача считается потерянной. Если процессор К1 свободен, то открывается клапан 2 и обрабатывается очередное сообщение из накопителя Н1. В зависимости от первоначального направления обработанное в процессоре сообщение поступает в буфер первой (клапан3) или второй (клапан4) линии. Хранящиеся в накопителях Н3 и Н4 сообщения по очереди пересылаются каналами К2 и К3.

**4 Переменные и уравнения имитационной модели**

Экзогенные переменные:

gen\_time = 8, lng\_time = 14 – интервал времени через который приходят задачи (157 с, [8,22]);

Эндогенные переменные:

proc\_speed = 7 – время обработки 1 сообщения ;

time\_line = 10, lng\_linr = 10 – интервал времени передачи по каналам [10,20]);

line\_control = 3 – ограничение сообщений одного направления.

Вспомогательные переменные:

source[i].ntime – время появления на источнике i следуещей задачи;

CPU.ntime – время окончания обработки текущей задачи;

ln[i].ntime – время появления на канале i следуещей задачи;

Переменные состояния:

count\_err – количество потеряных задач;

CPU.type – номер направления текущей задачи;

source[i].type– номер направления;

count\_task – количество принятых задач;

count\_line1 – количество сообщений первого направления;

count\_line2 – количество сообщений второго направления;

Дополнительные переменные используемые в программе:

t – текущее время моделирования;

End\_time – время моделирования.

Параметры модели:

L – емкость буферных регистров, L= 100 [сообщений].

Уравнения модели:

Potk= count\_err/(count\_task + count\_err) – вероятность отказа в обработке задачи

Кзагр = (count\_task\*proc\_speed)/End\_time;

**5 Обобщенная схема моделирующего алгоритма**

Для формализации процесса функционирования системы использован алгоритм с детерминированным шагом. Детерминированный моделирующий алгоритм является наиболее простым и удобным. Значение ∆t выбрано равным 1с. Обобщенная схема моделирующего алгоритма процесса функционирования системы приведена на рисунке 3.

1

Ввод исходных данных

Установка начальных условий моделирования

2

Да

3

Обработка результатов моделирования

Проверка времени окнчания моделирования

8

Нет

Вывод результатов моделирования

9

Передача сообщения процессору

4

7

5

Буферизация обработанных сообщений

Переход к следующему моменту времени

Передача сообщений на выходные линии

6

Рисунок 3 – Обобщенная схема моделирующего алгоритма

**6 Особенности программирования**

6.1 Функции программы

Программа реализована на языке С++ и выполняет моделирование процесса коммутации информации в течение 10 секунд.

Листинг программы приведен в приложении Б.

Программа подсчитывает количество обслуженных заявок, загруженность процессора, вероятность отказа в обслуживании, количество переданных по первой и второй линии сообщений.

6.2 Входные данные

gen\_time – начальное значение интервала времени, через которое на терминал поступают задачи;

lng\_time – длина интервала времени, через которое на терминал поступают задачи;

line\_time – начальное значение интервала времени, через которое линия освобождается;

line\_long – длина интервала времени, через которое линия освобождается;

proc\_speed – время обработки данных;

End\_time – время моделирования;

line\_control – ограничение количества сообщений одного направления;

6.3 Выходные данные

count\_task – заявок обслужено;

count\_err – заявок потеряно;

count\_line1 – сообщений отправлено первой линией;

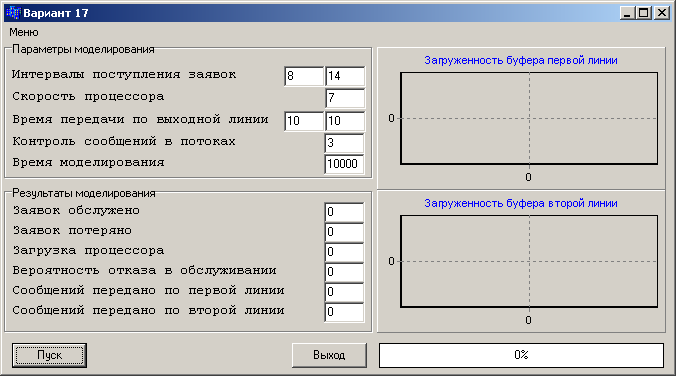
count\_line2 – сообщений отправлено второй линией;

Perr – вероятность отказа в обслуживании.

6.4 Организация интерфейса

При запуске программы появляется окно, содержащее параметры моделирования, а также выходные данные и графики. Для моделирования процесса обработки информации необходимо установить требуемые входные параметры системы. Моделирование начинается нажатием кнопки «Пуск». Ход моделирования отражается на шкале в процентах. По окончанию моделирования результаты можно просмотреть в соответствующих полях и графиках.

Меню содержит два пункта «Задание» и «Q-схема», которые активизируют окна с заданием и Q-схемой алгоритма.

****

**7 Результаты моделирования**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Заявок обсужено | Заявок потеряно | Загрузка процессора | P отказа | Заявок 1 линии | Заявок  2 линии |
| 1334 | 0 | 93,38 | 0 | 670 | 663 |
| 1335 | 0 | 93,45 | 0 | 659 | 675 |
| 1324 | 0 | 92,68 | 0 | 656 | 667 |
| 1327 | 0 | 92,89 | 0 | 657 | 669 |
| 1340 | 0 | 93,8 | 0 | 673 | 666 |
| 1343 | 0 | 94,01 | 0 | 667 | 675 |
| 1337 | 0 | 93,59 | 0 | 662 | 674 |
| 1328 | 0 | 92,96 | 0 | 672 | 655 |
| 1321 | 0 | 92,47 | 0 | 662 | 658 |
| 1329 | 0 | 93,03 | 0 | 663 | 665 |
| Средние значения | | | | | |
| 1331,8 | 0 | 93,226 | 0 | 664,1 | 666,7 |