

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ РАСПРЕДЕЛЕНИЯ И ИСПРАВЛЕНИЯ ОШИБОК В ЭЛЕМЕНТАХ ПАМЯТИ

А.В. Орлов

(БГТУ, г. Минск)

Компьютерная модель памяти предназначена для рассмотрения алгоритмов распределения отказов в элементах памяти, оптимизации нахождения ошибок, расчета и определения эффективности исправляющих корректирующих кодов.

Анализ методом имитационного моделирования заключается в определении происходящих в системе памяти событий и их привязки к моментам дискретного времени и элементам системы.

Время, в котором функционирует моделируемый объект, называется *модельным*. Дискретизация модельного времени производится путем разбиения заданного для имитации интервала $[0, T_k]$ на участки. Любое событие, происшедшее на участке с границами t_i и t_{i+1} , считается происшедшим в момент времени t_{i+1} .

Математическая модель, готовая к имитации прохождения заявок, представляет собой программу, включающую ряд операторов. Операторы, относящиеся к группе основных, вызывают действия над заявками. Это операторы генерирования заявок, направления заявок к определенным узлам (микросхема памяти, строка, столбец, ячейка), постановки в очередь на обслуживание, занятия узла, освобождения и др. Исполнение такой программы по отношению к конкретной заявке выявляет ее маршрут и времена прохождения этапов маршрута. Однако поведение заявки в системе не является независимым, оно обуславливается событиями, в которых фигурируют также другие заявки. Поэтому процесс имитации на ЭВМ должен отображать хронологию событий в последовательности, имеющей место в реальной работе микросхем памяти. В связи с этим исполнение программы имитации по отношению к конкретной заявке, например заявке A , должно прерываться, когда заявка задерживается в очереди или в обслуживающем аппарате, и программа должна переходить к имитации более ранних событий, чем события, связанные с дальнейшим продвижением по маршруту заявки A .

Для соблюдения при имитации правильной хронологии событий все события, которые в данный момент модельного времени уже можно предвидеть,

упорядочиваются по времени наступления и оформляются в виде списка. Элементами списка являются имена (ссылки) заявок, с которыми связаны рассматриваемые события. В списке событий можно выделить его начало – *список текущих событий*, т. е. событий, которые должны произойти в текущий момент времени t_i . Остальная часть списка называется *списком будущих событий*.

После того как программа прерывает имитацию прохождения заявкой A своего маршрута, она переходит к обработке очередного события из списка текущих событий. Пусть это событие связано с заявкой B . Программа исполняется, начиная с того оператора, который ранее определил задержку заявки B , и доходит до такого оператора, который означает новую задержку заявки B . Если эта задержка происходит в связи с обслуживанием в некотором устройстве, то при этом определяется, длительность обслуживания, и, следовательно, момент времени $t_i + \tau$, в который, обслуживание закончится. Далее ссылка на заявку B включается в список будущих событий на место, определяемое значением $t_i + \tau$. После этого программа выбирает очередной элемент из списка текущих событий и имитирует продвижение новой заявки. Сведения о том месте, в котором была задержана заявка и с которого, следовательно, нужно возобновлять продвижение, фиксируются в специальном массиве параметров заявок.

После того как все события текущего момента времени t_i исчерпаны, модельное время увеличивается до значения времени t_{i+1} ближайшего события, список событий в момент времени t_{i+1} становится списком текущих событий. Процесс имитации продолжается по тому же алгоритму. Окончание процесса имитации связывается с исчерпанием всего списка событий или с прохождением всего заданного интервала времени T_k .

Следовательно, имитационное моделирование осуществляется путем многократного просмотра операторов имитационной модели и выполнения предписываемых ими действий. Просмотр операторов относится только к тем моментам времени, в которые произошли какие-либо события. При просмотре происходят обращения к математическим моделям тех элементов памяти, с которыми связаны происходящие события. Такое моделирование, избирательное по времени и месту событий, называется *событийным*. Использование принципа событийности позволяет выполнять только те действия, которые действительно необходимы для получения результата, и, следовательно, приводит к сокращению затрат машинного времени.

Программная реализация алгоритмов событийного моделирования представляет собой пакет подпрограмм, реализующий математические модели элементов, программы, осуществляющие процедуры выработки случайных чисел, статистической обработки результатов моделирования и др. Управляющая программа пакета содержит обращения к подпрограммам устройств, источников, памятей, с ее помощью реализуются ввод исходных данных, упорядочение списков событий, изменение модельного времени, продвижение заявок.

В базе данных (БД), с которой работает программа имитации, имеются следующие основные массивы данных:

Массив очередей и списков. Массив очередей упорядочен по устройствам, массив списков — по моментам наступления событий. Элементами этих массивов являются ссылки на заявки. Поэтому при обработке списков по этим элементам могут быть найдены адреса, по которым хранятся сведения о самих заявках.

Массив заявок. Здесь хранятся сведения о каждой заявке: имя заявки, ее тип, значения приоритетов, время очередного события, связанного с заявкой, текущее место нахождения заявки. Доступ к этому массиву осуществляется по ссылкам заявок, содержащимся в массивах очередей и списков.

Массив параметров элементов. В нем содержатся значения таких параметров, как числовые характеристики законов распределения интервалов обслуживания, данные о дисциплинах обслуживания, накопленные суммы, характеризующие на текущий момент время занятого состояния каждого устройства, количество обслуженных заявок, длины очередей, средние времена ожидания в очередях и т. п.

Процесс исполнения программы имитации и ее взаимодействия с БД показан на рис. 1. Каждый этап вычислений, начинается с выборки очередного элемента из списка текущих событий. Этот элемент представляет собой имя заявки (ее адрес в массиве заявок). Обозначим имя обрабатываемой заявки A . По извлеченным из массива заявок сведениям определяется имя освобождающегося OA и номер оператора в программе имитации, с которого будет прослеживаться продвижение заявки A . Сначала программа имитации проверяет наличие очереди на входе освободившегося OA и, если очередь имеется, OA начинает обслуживать некоторую заявку B из очереди в соответствии с заданной дисциплиной обслуживания. При этом происходит обращение к подпрограмме рассматриваемого OA , определяется время обслуживания и тем са-

мым становится предвидимым новое событие – окончание обслуживания заявки *B* в этом *OA*. Ссылка на это событие помещается в список будущих событий, и производятся изменения сведений о заявке *B* в массиве заявок. Далее имитируется продвижение в системе заявки *A*. Это продвижение прерывается, когда в соответствии с заданной программой производится запрос на обслуживание заявки *A* в некотором *OA*. При этом происходит обращение к массиву очередей по имени этого *OA* и либо поступление заявки *A* на обслуживание, либо ее постановка в очередь. Изменения, произошедшие в состояниях очередей, обслуживающих аппаратов и заявок, отражаются в изменении соответствующих данных в массиве параметров.

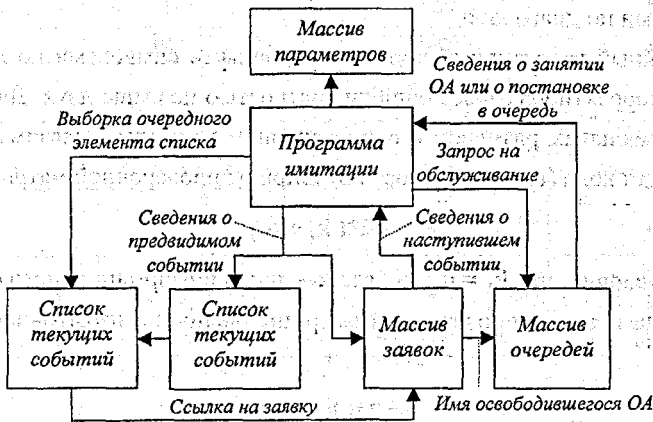


Рис. 1: Взаимодействие программы имитации с массивами БД

Литература.

1. Норенков И.П., Маничев В.Б. Системы автоматического проектирования электронной и вычислительной аппаратуры. – М.: Высшая школа, 1983, 272 с.
2. Северцев Н.А. Надежность сложных систем в эксплуатации и отработке. – М.: Высшая школа, 1989, 432 с.

ОПТИМИЗАЦИЯ ПАРАМЕТРОВ ПРОВЕРОЧНЫХ МАТРИЦ ДЛЯ ЛИНЕЙНОГО И ТРЕХМЕРНОГО ИТЕРАТИВНОГО КОДОВ С ПРОВЕРОЧНЫМИ СИМВОЛАМИ ПО ДИАГОНАЛИ

Д.М. Романенко
(БГТУ, г. Минск)

Наблюдаемое в последние годы резкое увеличение информационных потоков и связанное с этим ужесточение требований к целостности обрабаты-