***Методика построения программной модели ВС.***

Для разработки программной модели исходная система должна быть представлена как *стохастическая система массового обслуживания*. Это можно объяснить следующим: информация от внешней среды поступает в случайные моменты времени, длительность обработки различных типов информации может быть в общем случае различна. Т.о. внешняя среда является *генератором сообщений*. А комплекс вычислительных устройств (ВС) – *обслуживающими устройствами*.

**Обобщенная структурная схема ВС.**



**ИИ** – источники информации – выдают на вход буферной памяти (БП) независимые друг от друга сообщения. Закон появления сообщений – произвольный, но задан на перед.

В **БП** (буферной памяти) сообщения записываются «в навал» и выбираются по одному в обслуживающий аппарат (ОА) по принципу FIFO/LIFO. Длительность обработки одного сообщения в **ОА** в общем случае так же может быть случайной, но закон обработки сообщений должен быть задан. Т.к. быстродействие ОА ограничено то на входе системы в БП возможно сложение данных ожидающих обработки.

**А** – абоненты.

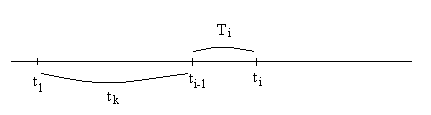
Программная модель из этой системы создается следующим образом:



Должна быть обязательно программа сбора статистики (**ПБССт –** программный блок сбора статистики). Причем статистику программа должна собирать по каждому из объектов модели. Так же должна быть программа, которая позволит "оживить" систему – это **программа синхронизации** (**блок синхронизации**), которая покажет когда и в какое время будут активизированы те или иные фрагменты модели.

**Моделирование работы источника информации (ИИ).**

Поток сообщений обычно имитируется моментами времени, отображающими появление очередного сообщения в потоке.





где Ti – интервал времени между появлением i-го и (i-1)-го сообщения.

Программа – имитатор выработки таких интервалов:

* + 1. Обратиться к генератору равномерно распределенных случайных величин на [a,b]
    2. Ti – по заданному закону
    3. К текущему времени + Ti

// процедурка равномерного распределения псевдослучайных чисел на итервале [a,b]

// U - равном. распр. на [0, 1]

// x = a + (b - a)U

double get\_time (int i)

{

double S = 0;

srand(seek);

if ( i > 1 ) S += get\_time(i - 1);

S += a + (b - a)get\_u();

return S;

}

или

double get\_time (int i)

{

double S = 0;

srand(seek);

for (int i = 0; i < .. ; i++)

S += a + (b - a)rand();

return S;

}

Выражения для вычисления времени с разлчным распределением:

|  |  |
| --- | --- |
| **Вид распределения** | **Выражение** |
| равномерное на [a,b] |  |
| Нормальное | , n ~= 12 |
| Экспоненциальное |  |
| Эрланга |  |

**Моделирование работы Обслуживающего Аппарата.**

Программа-имитатор работы ОА представляет собой комплекс, вырабатывающий случайные отрезки времени, соответствующие длительностям обслуживания требований. Например, если требования от источника обрабатываются в ОА по нормальному закону с параметрами *Mx* и *σx*, то длительность обработки *i*-ого требования:



**Схема алгоритма имитатора.**



Ri – случайное число с равномерным законом распределения

ТОБР – время обработки очередного сообщения

T – время освобождения ОА

XM – Мат ожидание для заданного закона обратки

DX – СКО (средне квадратичное отклонение) для заданного закона обратоки



**Моделирование работы абонентов.**

Абонент может рассматриваться как Обслуживающий Аппарат, поток информации, который поступает от процессора.

Для имитации работы абонентов необходимо составить программу выработки длительности обслуживания требования. Кроме того, абонент сам может быть источником заявок на те или иные ресурсы вычислительной системы. Эти заявки могут моделироваться с помощью генератора сообщений, распределенными по заданному закону. Таким образом, абонент либо имитируется как ОА, либо как генератор.

**Моделирование работы буферной памяти.**

**Память** - относится к электромеханическому устройству, включающее в себя: среду для запоминания, устройство управления, (информация находится по адресу) база + смещение + [индекс].

**Свойства памяти**: предназначена для хранения, чтения и записи информации.

В блок статистики: ошибки записи, ошибки чтения.

Блок буферной памяти должен производить запись и считывание чисел, выдавать сигналы переполнения и отсутствия данных в любой момент времени располагать сведениями о количестве требований (заявок) в блоке. Сама запоминающая среда в простейшем случае имитируется одномерным массивом, размер которого определяет ёмкость памяти. Каждый элемент этого массива может быть либо свободен и в этом случае мы считаем, что он равен 0, либо «занят», в этом случае в качестве эквивалента требования ему присваивается значение времени появления требования.

**Структурная схема модели программной памяти**:



Алгоритм реализации работы буферной памяти:



|  |  |  |  |
| --- | --- | --- | --- |
| *P* | массив сообщений | *LM* | объем буферной памяти |
| *WYB* | признак обращения к буф. памяти  = 1 – режим выборки сообщений  = 0 – режим записи | *NPOS* | номер последнего сообщения, поступившего в память |
| *NP* | число сообщений в памяти | *NPER* | номер первого сообщения в памяти |
| *POLN* | признак переполнения памяти  = 1 – нет свободных ячеек | *PUST* | признак отсутствия сообщений  = 1 – в памяти нет сообщений |
| *NPOS* | = NPOS + 1, если NPOS < LM  = NPOS – LM + 1, иначе | *NPER* | = NPER – 1, если NPER < 1  = NPER – LM + 1, иначе |
| *X* | ячейка для сообщения |  |  |

**Разработка программы для сбора статистики.**

Задача блока статистики заключается в накоплении численных значений необходимых для вычисления статистических оценок, заданных параметров работы моделируемой системы. При моделировании простейшей модели СМО, как правило, оценивают ***среднее время ожидания в очереди***. Для каждого сообщения *время ожидания в очереди* равно разности между моментами времени когда оно было выбрано на обработку обслуживающим аппаратом и моментом времени когда оно пришло в систему от источника информации.

Суммируя количество сообщений в блоке памяти через небольшие промежутки времени и разделив полученную сумму на число суммирований, получим *среднее значение длины очереди*.

*Коэффициент загрузки* обслуживающего аппарата (ОА) определяется как отношение времени работы ОА, к общему времени моделирования.

Чтобы определить *вероятность потери сообщений* в системе, нужно разделить кол-во потерянных сообщений на сумму потерянных и обработанных сообщений в системе.