Приоритетное обслуживание

- В сетях связи для ЭВМ характерной является передача сообщений с различными приоритетами. Коротким сообщением, содержащим подтверждения, часто назначают более высокий приоритет, чем информационным сообщениям. По сети могут передаваться сообщения двух и более категорий срочности. Например, некоторые пользователи, передающие в среднем сообщения более короткие, чем у других абонентов, получают приоритет для ускорения общей доставки сообщений. В связи с этим представляет интерес исследование системы *M/G/1* с несколькими классами сообщений, обладающих разными приоритетами.

- Для упрощения этой задачи основное внимание будет уделено определению среднего времени ожидания, а не времени задержки. Как видно из выражения (4), среднее время задержки всегда можно получить, добавив среднее время передачи сообщения к среднему времени ожидания. Будем предполагать, что сообщения разных классов обладают относительными приоритетами. При этом сообщение с более высоким приоритетом располагается в очереди перед сообщениями с более низким приоритетом, но уже начавшееся обслуживание сообщений с более низким приоритетом не прерывается.
- В рассматриваемой системе обслуживания предполагается, что классы сообщений, обозначаемые индексом $p = 1, 2, 3, ..., r,$ пронумерованы в порядке уменьшения приоритета. Рассмотрим обслуживание (начало передачи) с момента времени t_1 с целью получения общего соотношения для среднего времени $M(T_{ож})$ ожидания сообщения с приоритетом p.

- Для этого разберем, из каких компонентов складывается $T_{ож}$. Очевидно, что сюда входят: время T_0 необходимое для завершения текущего обслуживания; времена $T_к$ необходимые для обслуживания $m_к$ сообщений с приоритетами $к$ = 1, 2, ...,p-1, уже ожидающих обслуживания в очереди к моменту поступления рассматриваемого сообщения, и времена $T_к^{1}$ $к$=1,2, ..., p – 1 необходимые для обслуживания сообщений с более высоким приоритетом, которые могут поступить за интервал ожидания и будут обслужены раньше данного сообщения. Суммируя средние значения всех этих случайных величин, получим

$$M(T_{ожс})_p = M(T_0) + \sum_{k=1}^{p} M(T_k) + \sum_{k=1}^{p-1} M(T_k^1)$$

- Для оценки $M(T_к)$ допустим, что среднее число ожидающих сообщений с приоритетом *к* составляет $M(m_к)$. Если каждое из них требует для обслуживания в среднем $1/\mu_к$ единиц времени, то
- $M(T_к) = M(m_к)/\mu_к$. (22)
- Но $M(m_к)$ представляет собой разность двух величин - среднего числа сообщений, ожидающих и обслуживаемых в системе $M(n_к)$, и среднего числа обслуживаемых сообщений. Число последних составляет
- $\rho_к = \lambda_к/\mu_к$, где $\lambda_к$ - интенсивность потока сообщений *к-й* категории. Из теоремы Литтла следует, что

- $M(n_к) = M(m_к) + \rho_k$
- Следовательно
- $M(T_к) = \rho_k M(T_{ож})_k$

- По аналогии
- $M(T_к^{1}) = \rho_k M(T_{ож})_p$

Можно показать, что время ожидания для сообщений с приоритетом p можно найти по формуле

$$M(T_{ож})_p = \frac{M(T_0)}{(1-\sigma_p)(1-\sigma_{p-1})}$$

Где

$$\sigma_p = \sum_{k=1}^{p} \rho^k \qquad (23)$$

- Определим теперь величину времени $M(T_0)$, необходимого для завершения текущего обслуживания. Рассмотрим сначала систему обслуживания *MIGI1* с одним классом требований. Сравнивая выражения (5) и (23), получим в этом случае
- $M(T_0) = \lambda M(\tau^2)/2.$
- С целью проверки предположим, что распределение длин сообщений экспоненциальное. Тогда легко показать, что $M(T_0) = \rho/\mu.$
- Указанная величина может рассматриваться как произведение вероятности занятости системы обслуживания ρ на среднюю длину сообщения 1/μ.
- В более общем случае, для системы обслуживания с несколькими классами требований, получим

$$M(T_0) = \frac{1}{2}\sum_{k=1}^{r}\lambda_k M(\tau_k^2), \lambda = \sum_{k=1}^{r}\lambda_k$$

Система обслуживания *MIMINIm*

- Пусть на СМО *MIMINIm* с числом обслуживающих приборов N и числом мест для ожидания *m* поступает поток заявок с интенсивностью λ, которые обслуживаются каждым прибором с интенсивностью μ.
- Пусть также время ожидания в очереди распределено по экспоненциальному закону с параметром (интенсивностью) v.
- Определим вероятность обслуживания требований, вероятность ожидания требованием начала обслуживания и среднее время ожидания. (Задача Бухмана)
- В рассматриваемой СМО существуют следующие рабочие состояния:

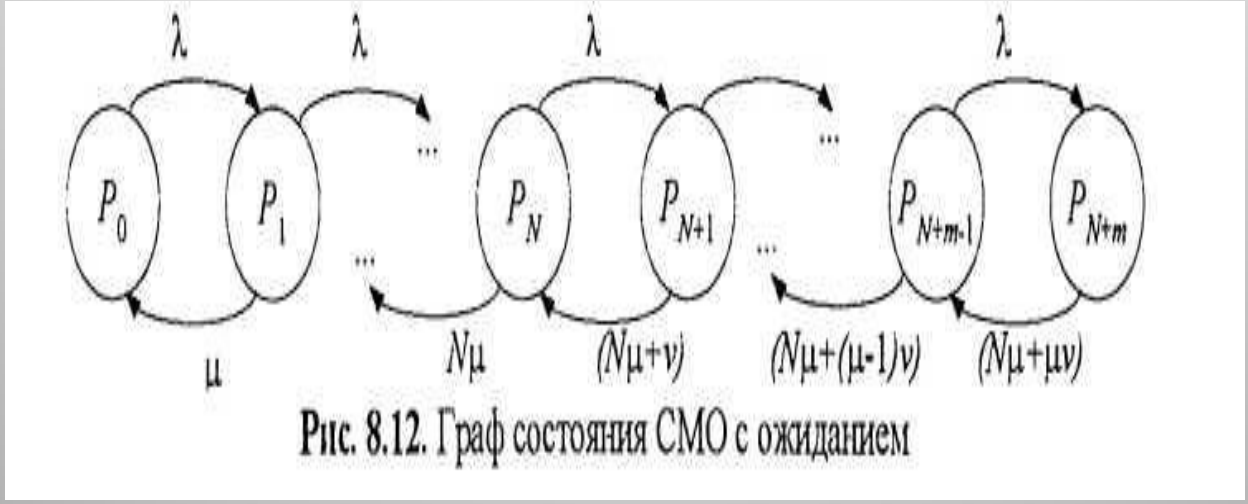

Рис. 8.12. Граф состояния СМО с ожиданием

- система обслуживает *s* требований с интенсивностью *sμ,* если *0<=s<N;*
- система ставит требование в очередь, если число требований больше числа обслуживающих приборов, но меньше числа мест ожидания *N<= s <m,* при этом интенсивность поступления требований из очереди равна *(s - N)v*
- система отказывает требованиям в обслуживании, если *s > (N+ m).*
- Под состоянием сети будем понимать значение числа требований, находящихся на обслуживании (в системе распределения ресурса и в очереди) в момент времени *t.* Обозначим через s = O, ..., *S* номер состояния СМО (число требований в ней), где *S= N+m.*
- Для аппроксимации вероятностно-временного механизма перехода СМО из одного состояния в другое используем аппарат марковских цепей.
- Решение задачи было найдено Эрлангом

- Среднее время ожидания начала обслуживания

- $T_{ож} = P(t_{ож}>0)/(\mu N-\lambda)$

- Средняя длина очереди вычисляется по формуле Литтла.

- Математический аппарат ТМО охватывает широкий класс СМО с простейшими, примитивными и рекуррентными потоками и может быть использован для анализа и синтеза СМО с отказами, с ожиданием и ненадежными единицами ресурса. Трудность аналитического разрешения уравнений состояния для СМО большой размерности делает целесообразным применение для их исследования методов имитационного моделирования и численных методов расчета на ЭВМ. Особо следует отметить важность постановки и решения оптимизационных задач для СМО. В качестве целевых функций критериев при этом целесообразно использовать полученные вероятностно-временные характеристики (ВВХ), а оптимизируемыми переменными могут стать интенсивности входящего потока требований, число мест для ожидания, число обслуживающих приборов, дисциплина обслуживания, алгоритм предоставления ресурса.

Системы массового обслуживания с отказами

- Рассмотрим задачу построения модели и анализа вероятностно-временных характеристик СМО с отказами на примере многоканальных систем. Пусть на вход СМО, содержащей N обслуживающих приборов, действует простейший поток требований (поток, обладающий свойствами стационарности, ординарности и отсутствием последействия) с интенсивностью λ, а обслуживание требований каждым прибором СМО осуществляется с интенсивностью μ. При занятости всех N приборов вновь пришедшая заявка получает отказ и покидает СМО. Данная задача была рассмотрена Эрлангом, и им впервые определены выражения для вероятности отказов (обслуживания) в СМО данного класса

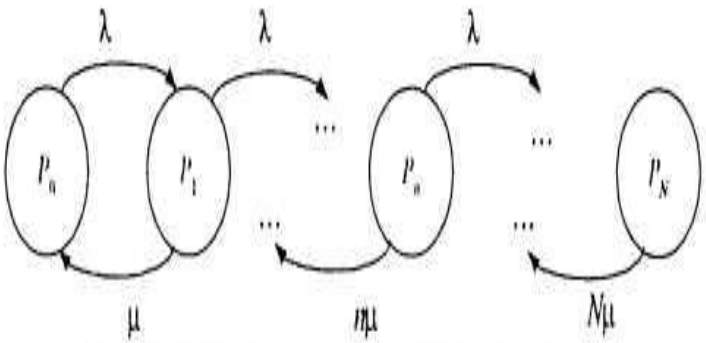

Рис. 8.13. Граф состояния СМО *M/M/N* с отказами

- Как видно из рисунка, СМО может находиться в одном из следующих состояний:
  - все приборы свободны, заявок на входе СМО нет;
  - один прибор занят, обслуживается одна заявка;
  - *n* приборов занято, обслуживается *n* требований;
  - все *N* каналов заняты, обслуживается *N* требований, а вновь пришедшая заявка теряется.
- Динамика вероятностей состояний СМО может быть описана системой дифференциальных уравнений, составленных по следующему мнемоническому правилу:
- производная $dP_n(t)/dt$ вероятности пребывания системы в состоянии *n* равна алгебраической сумме членов, число которых равно числу стрелок на графе состояния, соединяющих состояние *n* с другими состояниями;
- если стрелка направлена в состояние n, то член берется со знаком «минус»;
- если стрелка направлена из состояния *n,* то член берется со знаком «плюс»;
- каждый член суммы равен произведению вероятности того состояния, из которого направлена стрелка, и интенсивности потока событий, переводящего систему по данной стрелке.

- Данная система уравнений описывает вероятностный механизм смены состояний марковского процесса гибели и размножения:

$$\frac{dP_0(t)}{dt} = -\lambda P_0(t) + \mu P_1(t)$$

.....

- $$\frac{dP_n(t)}{dt} = -(\lambda + n\mu)P_n(t) + \lambda P_{n-1}(t) + (n+1)\mu P_{n+1}(t) \qquad (33)$$

.....

$$\frac{dP_N(t)}{dt} = N\mu P_N(t) + \lambda P_{N-1}(t)$$

- где $P_n(t)$ – вероятность принятия СМО состояния n=1,2,… N.

- Для установившегося режима работы СМО справедливы условия $dp_n(t)/dt = 0$, $\lambda/\mu <= 1$, $n = 1, \ldots, N$.
- В этом случае финальные вероятности, найденные из решения системы уравнений (33), имеют вид

$$P_1 = P_0 \frac{\lambda}{\mu}; \ldots; P_n = P_0 \frac{\rho^n}{n!}; \ldots; P_N = 1 - \sum_{n=0}^{N-1} P_n.$$

- Здесь введено понятие *загрузки прибора* (единицы ресурса) $\rho = \lambda/(n\mu)$.
- Из условия нормировки $\sum P_n = 1$ нетрудно получить выражение для вероятности сохранения незанятого состояния СМО:

$$P_0 = \left(\sum_{n=0}^{N} \frac{\rho^n}{n!} \right)^{-1}$$

- На основе этого выражения может быть определена вероятность отказа, т. е. вероятность того, что все *N* приборов заняты:

$$P_{отк} = \frac{\rho^n}{N!}\left(\sum_{n=0}^{N}\frac{\rho^n}{n!}\right)^{-1}$$

- а также вероятность обслуживания поступающих требований (1-я формула Эрланга)

- $P_{обсл} = 1 - P_{отк}$

Основные сведение о языке GPSS

- Язык имитационного моделирования GPSS (General Purpose System Simulator) разработан в 1961 г. фирмой IBM вслед за разработкой компилятора языка Фортран. Представляет собой фортран-ориентированную версию языка ИМ. Первые реализации GPSS строились в виде препроцессора, т.е. исходным текстом программ, анализирующих предложения GPSS, были тексты на Фортране. Существует много версий GPSS, являющегося наиболее распространенным ЯИМ данного класса. В настоящее время разработаны полные версии GPSS для ПЭВМ. С 1968 г. этот язык входит в математическое обеспечение машин фирмы IBM и является одним из наиболее популярных языков ИМ.

- GPSS составлен из объектов и операций (логических правил). Объекты делятся на семь классов.
- *Динамические объекты* (ДО) - элементы потока обслуживания заявки или «транзакты». Они создаются и уничтожаются. С каждым транзактом может быть связано некоторое число «параметров»
- *Аппаратно-ориентированные объекты* (АО) соответствуют элементам оборудования, которые управляют ДО. К ним относятся накопители, устройства, логические переключатели.
- *Статистические объекты* (СО) включают очереди и таблицы.
- *Запоминающие объекты* (30) состоят из ячеек и матриц ячеек.
- *Группирующие объекты* (ГО) - группы и списки.
- *Вычислительные объекты* (ВО) состоят из арифметических и булевых переменных, а также функций.
- *Операционные объекты* (00) — блоки, формирующие логику системы, давая транзактам указания, куда идти дальше.

- Чтобы смоделировать систему, необходимо составить ее описание в терминах GPSS. Затем симулятор генерирует транзакты, продвигает их через заданные блоки и выполняет действия, соответствующие блокам. Продвижение создает блок GENERATE. Каждое продвижение транзакта является событием, которое должно произойти в определенный момент времени. Симулятор регистрирует время наступления каждого события, после чего производит обработку событий в правильной хронологической последовательности.
- Если транзакты заблокированы, то симулятор сможет продвинуть их только тогда, когда изменятся блокирующие правила.
- Симулятор моделирует часы, показания которых в любой момент времени называют абсолютным временем. Относительное время показывает текущее время в модели. При помощи специальной операции относительное время может устанавливаться в нуль, и последующий счет времени будет производиться от этой точки. Специальной операцией оба значения времени могут устанавливаться в нуль. Относительное и абсолютное время в модели отображаются целыми числами. Симулятор рассчитывает схему по принципу ближайшего события. Центральной задачей симулятора является просмотр и проверка всех возможных событий.

- Программа на GPSS создается в текстовом редакторе в определенном формате. Формат ввода содержит 3 различные поля: *метка, операция* и *переменные.* Поле переменных содержит подполя, которые обозначены *A, B, C, D, ..., H.* Последующие поля отделяются от предыдущих запятыми. Пропущенное значение в поле переменных выделяется запятыми (кроме конца поля). Каждый из объектов требует определенного числа ячеек ОЗУ, в которых во время моделирования хранятся атрибуты объекта (АТО).
- Те из АТО, к которым может обращаться программист, называются *стандартными числовыми атрибутами* (СЧА), имеющими одно- или двухбуквенные мнемонические обозначения. Мнемонические обозначения указывают на тип СЧА, а целочисленное значение - на конкретное значение СЧА.

Динамические объекты GPSS. Транзактно-ориентированные блоки

- В системах массового обслуживания транзакт - это динамический объект, соответствующий заявке на обслуживание в СМО. Язык GPSS располагает средствами для порождения (генерации) транзактов, последовательного продвижения их от объекта к объекту, их задержки на время, соответствующее длительности активности, уничтожения (удаление из системы) транзактов.
- **Оператор GENERATE.** Первоначальный ввод транзактов в модель всегда осуществляется специальным блоком GENERATE. Моменты порождения транзактов и ввода их в модель могут образовывать как детерминированный, так И случайный поток событий. Если тип потока событий будет детерминированным, то интервалы между моментами ввода транзактов в модель будут отстоять друг от друга на равные временные промежутки.

- Значения интервалов в единицах модельного времени задает целая константа в поле *A.* Следует иметь в виду, что модельное время в GPSS - целое без знака (0. 1, 2, ...). Следовательно, все параметры закона распределения случайных интервалов между соседними событиями в потоке, имеющие смысл времени, должны быть приведены к целому формату с помощью масштаба времени.
- Если *A* = const, *B* = const, то оператор GENERATE описывает равномерный закон распределения длины интервала между соседними событиями в потоке.
- Опишем назначение остальных параметров, используемых в блоке GENERATE:
- *B* - может быть отличен от const и рассматривается как модификатор, в этом случае длина интервала определяется как *AB;*
- C-задержка начала генерации;
- *D* — число генерируемых транзактов (емкость источника);
- E- приоритет транзактов. Целое без знака: 0, 1,2, ...
- Операнды могут быть опущены, тогда по умолчанию *A = B=C = E=*D = 0.

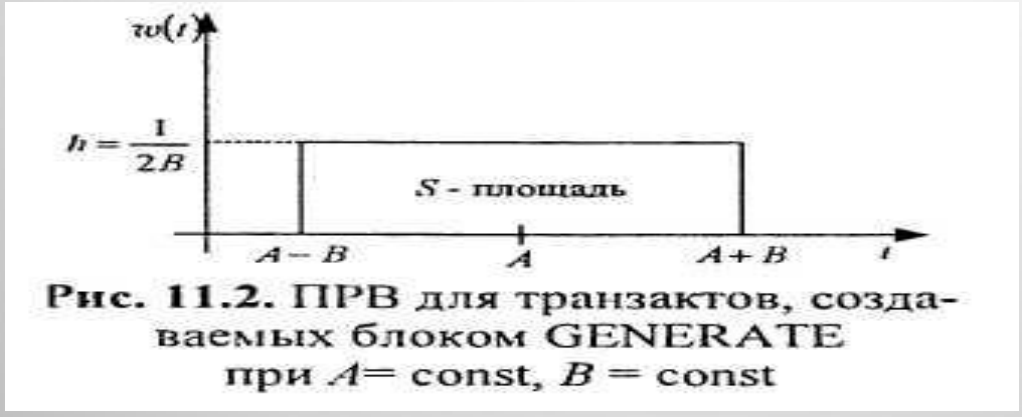

Рис. 11.2. ПРВ для транзактов, создаваемых блоком GENERATE при A= const, B = const

- S=1/λ – математическое ожидание,
- A>=B, S=2Bh, h=1/(2B)

- **Оператор ADVANCE.** В процессе движения транзактов по модели в определенных точках может возникать необходимость задержки транзактов на детерминированное или на случайное время. Чаще всего задержка транзакта связана с имитацией обслуживания (обработки).
- Задержка транзактов осуществляется блоком ADVANCE, имеющем два поля - *A* и *B.* Если поле *B* пусто, а в поле *A* указана целая константа, то транзакт, войдя в блок ADVANCE, остается в нем в течение интервала модельного времени, длительность которого определяется полем *A.*
- Если в поле *B* указывается целая константа, не превышающая константы в поле *A,* то осуществляется случайная задержка транзакта с равномерным распределением на интервале *(A + B, A - B)*

- **Оператор TERMINATE.** Начав свой путь на выходе блока GENERATE и пройдя то число операционных блоков GPSS-модели, которое при создавшейся случайной ситуации предусмотрено логикой модели, транзакт выводится из модели. Вывод транзакта из модели сопровождается уничтожением в памяти ЭВМ всех записей, характеризовавших состояния транзакта во время его продвижения по модели. Уничтожение транзакта производит блок TERMINATE.
- Блок TERMINATE имеет одно поле *A,* в котором записывается целая константа (или же дается ссылка на СЧА). В момент входа транзакта в блок TERMINATE следует вывод его из модели, при этом из специального счетчика вычитается указанная в поле *A* константа. В момент модельного времени, когда значение счетчика станет равным 0 (или меньше 0), моделирование прекращается, и на печать выдаются результаты в виде таблиц с соответствующими комментариями, статистики, накопленные в процессе моделирования.
- При пустом поле *A* блока TERMINATE его значение считается равным 0, из счетчика ничего не вычитается.

- **Аппаратно-ориентированные блоки**
- **А**ппаратно-ориентированные блоки (операторы) описывают действия по занятию и освобождению ресурсов (каналов обслуживания) с образованием очередей к занятым ресурсам.
- **Операторы SEIZE и RELEASE.** В начале моделирования все одноканальные приборы обслуживания считаются свободными (их статус считается равным *NU*- от английского NOT USE). Занятие устройства происходит в момент прохода транзактом блока SEIZE, в поле *A* которого указывается символическое имя (или порядковый номер прибора). Особенностью блока SEIZE является его способность задерживать транзакты, если в момент подхода транзакта к блоку SEIZE прибор с указанным именем занят другим транзактом (находится в состоянии *U*— от английского USE).

- Если в течение некоторого интервала модельного времени несколько транзактов пытаются войти в блок SEIZE, то организуется очередь транзактов, ждущих разрешения на вход в блок SEIZE. Это эквивалентно образованию на входе устройства очереди с неограниченным числом мест. В реальной системе моделирования длина очереди ограничена ресурсами, выделяемыми системой моделирования для организации очередей.
- Чтобы не было бесконечного возрастания длины очереди, необходимо обеспечить выполнение условия, при котором существует установившийся режим в системе с чистым ожиданием: $\rho < 1$, $\rho = \lambda/\mu$.
- По умолчанию принимается дисциплина обслуживания очереди FIFO.
- Освобождение прибора (перевод прибора из состояния *U* в состояние NU) происходит в момент прохода транзактом блока с именем RELEASE. В поле *A* этого блока должно указываться то же имя, что и в блоке SEIZE. Попытка входа транзаката в блок RELEASE, ранее не прошедшего блок SEIZE с тем же именем в поле *A,* что и в блоке RELEASE, приводит к прекращению моделирования из-за нарушения логики моделирования.

- *Пример.* Требуется построить имитационную модель одноканальной СМО с чистым ожиданием (рис. 11.3). Реальным объектом моделирования является, например, однопроцессорная ЭВМ, обрабатывающая в оперативном режиме запросы пользователя для случая, когда область памяти, выделенная для буферизации запросов, настолько велика, что влиянием ее размера на характеристики системы можно пренебречь.

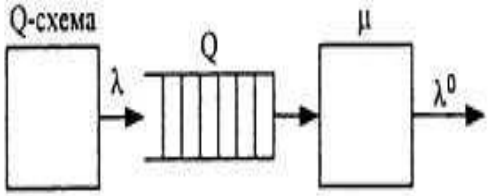

Рис. 11.3. Имитационная модель одноканальной СМО с чистым ожиданием

- Интенсивность поступления транзактов (запросов) в очередь λ=20 с -1, интенсивность обслуживания μ = 40 с$^{-1}$, коэффициент использования
- ρ = 0,5 < 1. Общее число транзактов, которое необходимо смоделировать, равно 1000.
- Будем считать, что входной поток и поток обслуживания имеют равномерно распределенные длины интервалов с 20%-м отклонением от средних длин (не простейшие потоки).
- Распределение интервалов входящего потока и потока обслуживания показано на рис 11.4.
-

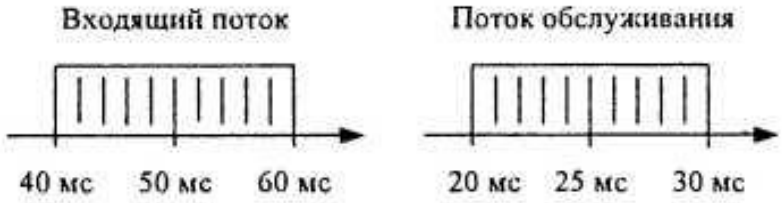

Рис. 11.4. ПРВ входящего потока и потока обслуживания

- Масштаб (единица модельного времени) моделирования $\Delta t = 1$ мс. На рис. 11.5 приведена блок-диаграмма ИМ.

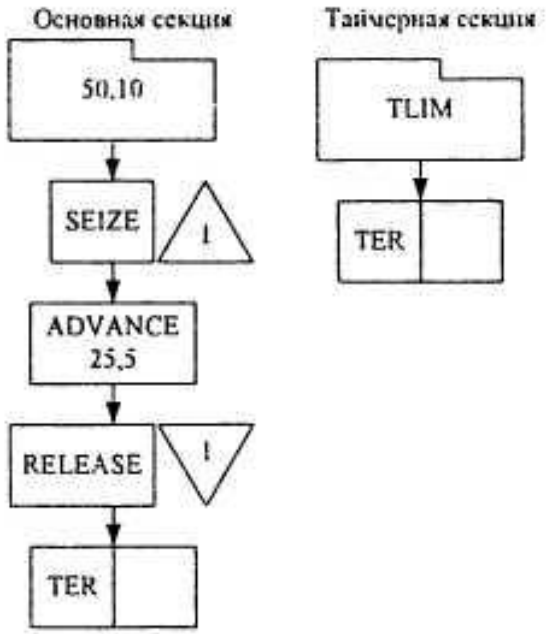

Рис. 11.5. Блок-диаграмма ИМ

- Общее время моделирования (в единицах модельного времени) запишется как TLIM = $N/(\lambda \Delta t) = 1000/(20*10^{-3}) = 50*10^3$, где N- общее число транзактов.
- На основании блок-диаграммы ИМ запишем программу на языке GPSS (листинг 11.1).
- Листинг 11.1.
- GENERATE 50,10
- SEIZE 1
- ADVANCE 25,5
- RELEASE 1
- TERMINATE
- ;Time Section
- GENERATE 50000
- TERMINATE 1

Многоканальное обслуживание

- Для моделирования многоканального обслуживания в GPSS используются специальные объекты, называемые накопителями.
- Моделирование параллельно работающих каналов обслуживания в GPSS осуществляется с помощью накопителей (STORAGE). Накопители (многоканальные устройства), в отличие от устройства (канала обслуживания), позволяют моделировать сложный ресурс, который может выделяться частями, причем отдельными частями накопителя (каналами) может одновременно обслуживаться несколько транзактов. Накопители характеризуются емкостью (CAPASITY), задаваемой целым положительным числом. Емкость накопителей описывается оператором STORAGE, в поле *A* которого указывается имя (порядковый номер) накопителя, а в поле *B* - целая константа, определяющая емкость. Например, запись TERM STORAGE 24 означает, что накопитель с именем TERM имеет емкость, равную 24.
- Емкость накопителя можно интерпретировать как число мест для размещения транзактов, хотя на самом деле транзакты не размещаются в накопителе.

- Для фиксации входа транзакта в память применяется блок ENTER , в поле *A* которого указывается имя или номер памяти, а в поле *B* - число единиц памяти, занимаемых в ней транзактом. Поле *B* может быть опущено, в этом случае считается, что занимается одна единица памяти.
- Если в момент подхода транзакта к блоку ENTER все места в накопителе заняты (статус накопителя в этот момент равен *SF* - от английского STORAGE FULL), или же число свободных мест меньше константы в поле *B* блока ENTER, то транзакт не пропускается блоком ENTER. При этом организуется очередь транзактов на вход блока аналогично тому, как организуется очередь к блоку SEIZE.
- Если в памяти нет достаточного числа свободных единиц, запрашиваемых блоком ENTER, то транзакт задерживается на входе этого блока до тех пор, пока в памяти не будет освобождено необходимое число единиц памяти. Причем в то время, пока этот транзакт ждет входа в блок ENTER, другой транзакт, пришедший позже, может войти в блок ENTER, если для него достаточно свободных единиц памяти.
- Отметим, что в начальный момент времени все накопители свободны и их статус считается равным *SE* - от английского STORAGE EMPTY.

- Освобождение мест в накопителе происходит в момент прохода транзактом блока LEAVE; поля этого блока имеют тот же смысл, что и поля блока ENTER. Транзакт, входящий в блок LEAVE, обязательно должен перед этим пройти блок ENTER, в противном случае будет зафиксирована ошибка в процессе моделирования. Значения же полей B блоков ENTER и LEAVE не должны обязательно совпадать.
- Логика работы блоков ENTER и LEAVE позволяет моделировать обслуживание в многоканальных СМО. В этом случае занятие одного места в накопителе можно интерпретировать как занятие одного канала обслуживания. Задержка канала в течение некоторого времени соответствует обслуживанию в многоканальной СМО и моделируется блоком ADVANCE аналогично случаю одноканального обслуживания. Блок ADVANCE помещается между блоками ENTER и LEAVE.