

Министерство образования и науки Украины
Севастопольский национальный технический университет

**ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК НЕПРЕРЫВНО-
И ДИСКРЕТНО - СТОХАСТИЧЕСКИХ СИСТЕМ С
ИСПОЛЬЗОВАНИЕМ ИМИТАЦИОННОГО
МОДЕЛИРОВАНИЯ НА БАЗЕ GPSS И MVS**

Методические указания
к лабораторным работам №1-3
по дисциплине
“МОДЕЛИРОВАНИЕ СИСТЕМ”
для студентов специальности 7.080401 –
"Информационные управляющие системы и технологии"
всех форм обучения
Часть 1

Севастополь
2006



УДК 004.92

Исследование характеристик непрерывно- и дискретно- стохастических систем с использованием имитационного моделирования на базе GPSS и MVS: Методические указания к лабораторным работам №1-3 по дисциплине “Моделирование систем”, для студентов специальности 7.080401 – “Информационные управляющие системы и технологии” всех форм обучения, Часть 1/Сост. А.А.Ерошко, О.Л. Тимофеева. – Севастополь: Изд-во СевНТУ, 2006.– 44 с.

Цель методических указаний: выработка у студентов практических навыков по разработке и исследованию характеристик систем с использованием имитационного моделирования на базе GPSS и MVS.

Методические указания рассмотрены и утверждены на заседании кафедры Информационных Систем. Протокол № 8 от 15 февраля 2006 г.

Рецензент: доцент кафедры информационных систем СевНТУ, канд.техн.наук Кудрявченко И.В.

Допущено учебно-методическим центром в качестве методических указаний.

СОДЕРЖАНИЕ

Введение	4
Лабораторная работа № 1. Имитационное моделирование одноканальной системы массового обслуживания.	5
Лабораторная работа № 2. Изучение типовых конструкций программ имитационного моделирования на языке GPSS.	11
Лабораторная работа № 3. Изучение технологии имитационного моделирования.	20
Библиографический список	27
Приложение А. Система имитационного моделирования GPSS	28
Приложение Б. Базовые блоки системы имитационного моделирования GPSS	35
Приложение В. Текст базовой программы модели мультипрограммной вычислительной системы	43

ВВЕДЕНИЕ

Сложные функции моделирующего алгоритма могут быть реализованы средствами универсальных языков программирования (Паскаль, Си), что предоставляет неограниченные возможности в разработке, отладке и использовании модели. Однако подобная гибкость приобретается ценой больших усилий, затрачиваемых на разработку и программирование весьма сложных моделирующих алгоритмов, оперирующих со списковыми структурами данных. Альтернативой этому является использование специализированных языков имитационного моделирования. Программы имитационных моделей на специализированных языках моделирования близки к описаниям моделируемых систем на естественном языке. Одним из наиболее эффективных и распространенных языков моделирования сложных дискретных систем, процесс функционирования, которых можно представить в виде множества состояний и правил перехода из одного состояния в другое, является язык GPSS (General Purpose Simulation System), разработанный в 1974г, фирмой IBM. В настоящее время широко используется система GPSS World, предоставляющая пользователю набор интерфейсных средств для разработки, отладки программной модели и анализа результатов моделирования.

Примерами объектов, определяемых в дискретной пространственно-временной области являются:

1. сети ЭВМ;
2. транспортные объекты;
3. предприятия, склады;
4. системы передачи сообщений.

Формальные модели таких объектов:

1. системы массового обслуживания (СМО);
2. автоматы;
3. стохастические сети.

СМО представляет собой класс математических моделей для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания заявок: на обработку информации ЭВМ от удаленных терминалов, сбоев вычислительных систем, индексирования документов в банках данных и т.д. Характерным является случайное появление заявок на обслуживание и завершение обслуживания в случайные моменты времени, т.е. стохастический характер процесса их обслуживания [2].

В состав GPSS входят следующие типы объектов: заявки (сообщения), блоки, списки, устройства, памяти, логические ключи, очереди, таблицы, функции, переменные. Любую модель на языке GPSS можно представить в виде комбинации компонентов, взятых из числа указанных объектов.

На каждую из лабораторных работ отводится 4 часа аудиторных занятий и 4 часа самостоятельной работы.

ЛАБОРАТОРНАЯ РАБОТА №1.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ОДНОКАНАЛЬНОЙ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ

1. Цель работы:

Исследование характеристик одноканальной системы массового обслуживания, используя аналитический и имитационный методы моделирования. Изучение особенностей работы и получение практических навыков постановки, отладки и получения результатов в системе GPSS-World.

2. Основные положения

Рассмотрим модель системы, состоящей из одного канала обслуживания запросов, поступающих на вход системы (одноканальная СМО), используя аналитический и имитационный подход.

2.1. Математическая модель одноканальной СМО

При ряде упрощающих предположений (допущение об отсутствии последствия) для оценки вероятностно-временных характеристик процесса обслуживания можно использовать Марковские модели[4]. Самая известная модель – это СМО типа М/М/1, где М – Марковские процессы распределения времени поступления и обслуживания, 1 – число обслуживающих устройств. При этом предполагается, что поток входных заявок является простейшим.

Простейший поток обладает следующими свойствами:

стационарностью – вероятность характеристик потока не зависит от времени;

отсутствием последствия – заявки поступают не зависимо друг от друга, длина интервала времени до момента поступления следующей заявки не зависит от того, поступила в начальный момент заявка или нет;

ординарностью – в каждый момент времени в систему может поступить не более одной заявки.

Для простейшего потока число заявок, поступающих в систему за промежуток времени t , описывается распределением Пуассона.

$$P(t) = \frac{(\lambda t)^n}{n!} \cdot e^{-\lambda t}, \quad (1)$$

вероятность того, что за время t в систему поступит точно n заявок; λ – интенсивность потока заявок.

Для того, чтобы при моделировании задать пуассоновский поток заявок в систему, достаточно задать экспоненциальное распределение интервалов времени поступления для соседних заявок.

Длительность обслуживания экспоненциально распределена с плотностью

$$f(t) = \mu \cdot e^{-\mu t}, \quad (2)$$

μ – интенсивность обслуживания – количество заявок, которое может быть обслужено в единицу времени.

Соответственно

$T_{\lambda} = 1/\lambda$ – среднее время между поступлением заявок;

$T_{\mu} = 1/\mu$ – среднее время обслуживания заявок.

Для стационарного режима ($\lambda < \mu$) справедливо рекуррентное соотношение для расчета вероятностей

$$(1 + \rho)P_n = P_{n+1} + \rho \cdot P_{n-1}, \quad n \geq 1, \quad P_1 = \rho \cdot P_0, \quad (3)$$

где $\rho = \frac{\lambda}{\mu}$ – загрузка системы (характеристика качества функционирования системы),

Так как $P_0 = 1 - \rho$ – вероятность того, что заявки в системе отсутствуют, то

$$P_n = \rho^n \cdot (1 - \rho). \quad (4)$$

вероятность того, что в системе находится n – заявок.

Среднее и дисперсия числа заявок в системе определяются как:

$$L = \sum_{n=0}^{\infty} n \cdot P_n = (1 - \rho) \cdot \sum_{n=0}^{\infty} n \cdot \rho^n = \frac{\rho}{1 - \rho}, \quad (5)$$

$$\sigma_L^2 = \sum_{n=0}^{\infty} (n - L)^2 \cdot P_n = \frac{\rho}{(1 - \rho)^2}. \quad (6)$$

Среднее и дисперсия числа заявок, находящихся в очереди к прибору, соответственно равны:

$$L_Q = \sum_{n=1}^{\infty} (n - 1) \cdot P_n = \frac{\rho^2}{1 - \rho}, \quad (7)$$

$$\sigma_{L_Q}^2 = \sum_{n=1}^{\infty} (n - L)^2 \cdot P_n - L^2 = \frac{\rho^2 \cdot (1 + \rho - \rho^2)}{(1 - \rho)^2}. \quad (8)$$

Вероятность того, что время пребывания заявки в системе T не превосходит t при условии, что длина очереди $L_Q = n$ определяется из условия

$$P(T \leq t | L_Q = n) = \int_0^t \frac{\mu^{n+1} \cdot \tau^n \cdot e^{-\mu \cdot \tau}}{\Gamma(n+1)} d\tau, \quad (9)$$

где $\Gamma(n)$ – гамма-функция.

По формуле полной вероятности можно определить вероятность того, что время пребывания заявки в системе T не превосходит t

$$Q(t) = P(T \leq t) = \sum_{n=0}^{\infty} P(T \leq t | L_Q = n) \cdot \rho(1 - \rho) = 1 - e^{-\mu \cdot (1 - \rho) \cdot t}. \quad (10)$$

Среднее и дисперсия времени пребывания заявки в системе рассчитывается по формулам

$$\bar{T} = \int_0^{\infty} t dQ(t) = \frac{1}{\mu \cdot (1 - \rho)}, \quad (11)$$

$$\sigma_{\bar{T}}^2 = \int_0^{\infty} (t - \bar{T})^2 dQ(t) = \frac{1}{[\mu(1 - \rho)]^2}. \quad (12)$$

Возможности оценки характеристик стохастических систем при помощи аналитических моделей ограничены по сравнению с имитационными моделями, использующими специально разработанные языки моделирования (GPSS, SYMULINK, UML). Эти ограничения вызваны:

- различием в обслуживании заявок разных типов, например связанных с наличием приоритетов;
- невозможностью представления в аналитических моделях обслуживания заявок одновременно несколькими приборами;
- трудностями получения зависимостей в явной аналитической форме при большом числе систем в сети; сети, содержащие более трех систем, уже не поддаются аналитическому исследованию[4,5].

2.2. Имитационная модель

Имитационная модель представляет собой систему, отображающую структуру и функционирование исходного объекта в виде алгоритма, связывающего входные и выходные переменные, принятые в качестве характеристик исследуемого объекта. При ее реализации на ЭВМ производится накопление статистических данных по тем атрибутам модели, характеристики которых являются предметом исследований. По окончании моделирования накопленная статистика обрабатывается, и результаты моделирования получаются в виде выборочных распределений исследуемых величин [1,3,5].

Программа моделирования на языке GPSS одноканальной СМО может быть представлена в виде следующих сегментов:

- 1) декларативная часть – описание функций распределения;
- 2) собственно модель – набор блоков, соответствующий одноканальной СМО;
- 3) таймер – набор блоков, определяющих время работы модели.

Соответствие между объектами одноканальной СМО и операторами языка GPSS приведено на Рисунке 1. Назначение блоков программы подробно изложено в Приложении А и источниках [3,5].

Минимальное время моделирования, можно определить из следующей зависимости

$$\Delta t = (T_A + T_M). \quad (13)$$

Для того, чтобы получить автоматический сбор статистических данных, описывающих вынужденное ожидание заявки на входе в прибор необходимо воспользоваться парой взаимодополняющих блоков: QUEUE и DEPART.

Если в модели используются объекты типа «очередь», то в файле стандартной статистики будет представлена информация об этих объектах.

Описание стандартной статистики и назначение блоков регистрации очереди изложено в Приложение А и источниках [3,5].

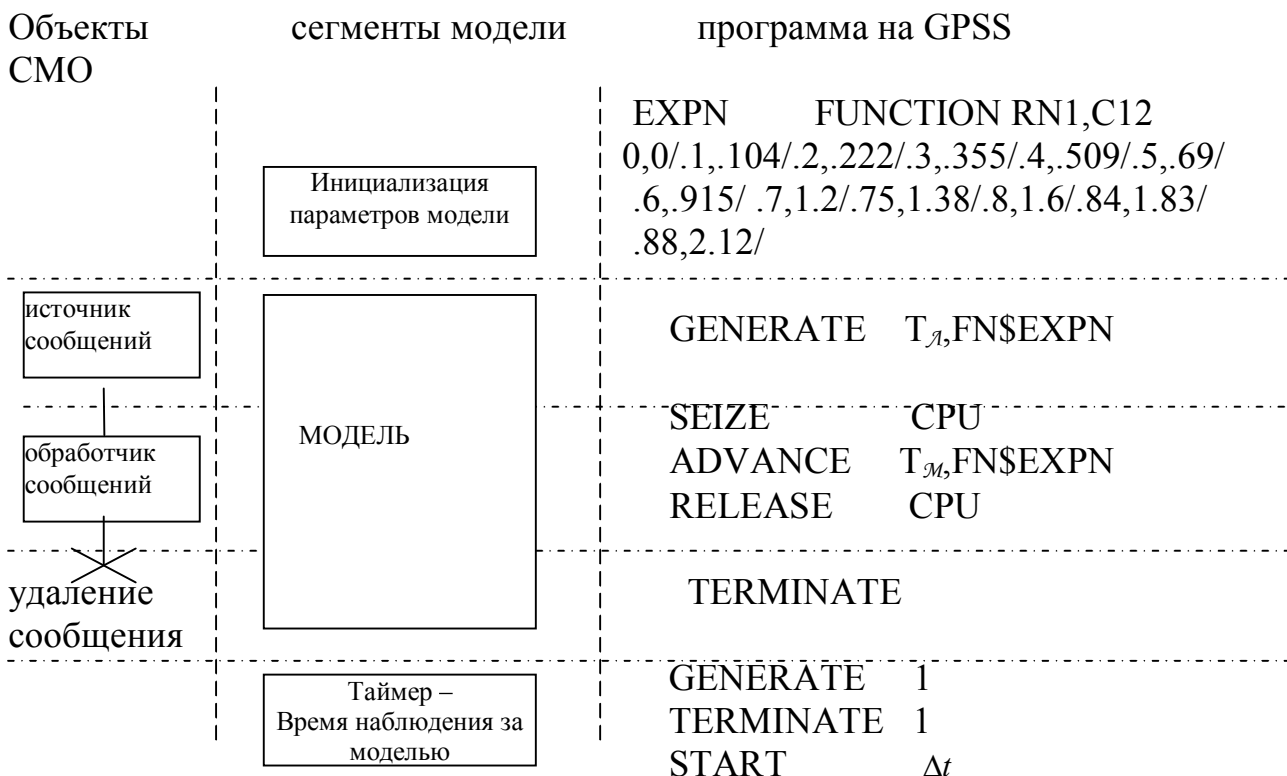


Рисунок 1 – Соответствие между объектами СМО и программой на языке GPSS

Полная последовательность блоков описания модели, с учетом блоков сбора статистики времени нахождения сообщения в очереди при обслуживании устройством CPU, приведена в Таблице 1.

Таблица 1 – Блоки описания модели с учетом блоков сбора статистики

№ блока	Имя блока	Параметры блока	Комментарии
1	GENERATE	$T_{л}, FN$EXPN$	
2	QUEUE	QCPU	поставить заявку в очередь к прибору
3	SEIZE	CPU	
4	DEPART	QCPU	прибор захвачен, покинуть очередь
5	ADVANCE	$T_{м}, FN$EXPN$	
6	RELEASE	CPU	
7	TERMINATE		

2.3. Анализ результатов имитационного моделирования

После прогона модели, будет получен файл результатов, содержащий статистику об использовании устройства, состоянии очереди QCPU на входе устройства. Оценкой среднего времени пребывания заявки в системе по имитационной модели может служить величина

$$\bar{t} = t_Q + t_n, \quad (14)$$

где

t_Q – среднее время пребывания заявки в очереди;

t_n – среднее время обслуживания заявки прибором (Приложение А).

Сопоставляя результаты моделирования с расчетами по аналитическим зависимостям :

- среднее время пребывания заявки в системе $\bar{t}$ и $\bar{T}$;
- значение загрузки системы ρ и коэффициент использования устройства Util;
- наличие очереди и ее характеристики (L_Q и Ave.count)

можно, на примере простейшей СМО, качественно оценить пригодность использования аппарата имитационного моделирования.

3. Ход работы

1. Оценить аналитическими методами вероятность нахождения в системе n заявок P_n (3) для $n = 0, 1, 2, \dots, 10$, среднее число и дисперсию числа заявок в системе и в очереди (5,6,7,8).
2. Построить графики функции распределения времени пребывания заявки в системе $Q(t)$ (9) для $t = 0, \Delta t, 2 * \Delta t, \dots, 10 * \Delta t$.
3. Оценить среднее и дисперсию времени пребывания заявки в системе (11,12).
4. Запрограммировать модель одноканальной СМО, в соответствии с требованиями языка моделирования (Приложение А или [3,5]). Подставить в нее исходные данные (для источника и обслуживающего прибора), согласно варианта. Получить файл статистики и сохранить его для дальнейшего анализа.
5. Повторить п.4, введя в программу операторы снятия статистики об ожидании в очереди при обслуживании устройством CPU . Сопоставить полученные файлы результатов. Определить $\bar{t}$.
6. Повторить п.4 для значений $t = \Delta t, 5 * \Delta t, \dots, 50 * \Delta t$. Определить $\bar{t}$. Построить график зависимости $\bar{t}$ и Util-коэффициента, использования прибора.
7. Сравнить результаты моделирования с расчетами по аналитическим зависимостям. Сделать выводы.
8. Оформить отчет по работе.

4. Содержание отчета

1. Цель работы.
2. Результаты расчетов (таблицы, графики) по аналитическим зависимостям.
3. Текст программы.
4. Графики зависимости длины очереди, Util и среднего времени пребывания заявки в системе от времени моделирования.
5. Выводы по работе.

5. Варианты заданий

- λ – последняя цифра дня даты рождения.
 μ – последние две цифры № зачетной книжки.

6. Контрольные вопросы

1. К каким классам задач, применим аппарат имитационного моделирования?
2. Свойства простейшего потока событий?
3. Граф состояний одноканальной СМО с очередью?
4. Система GPSS назначение, тип исследуемых моделей?
5. Назначение блока GENERATE языка GPSS?
6. Описание устройств обслуживания и очередей в языке GPSS?
7. Генерация случайных чисел с заданным законом распределения в языке GPSS?

ЛАБОРАТОРНАЯ РАБОТА №2. ИЗУЧЕНИЕ ТИПОВЫХ КОНСТРУКЦИЙ ПРОГРАММ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ НА ЯЗЫКЕ GPSS

1. Цель работы:

Изучение базовых блоков языка GPSS и получение практических навыков программирования имитационных моделей (ИМ).

2. Основные положения

2.1. Внутренняя организация GPSS

2.1.1. Модельное время

Для обеспечения правильной временной последовательности событий в модели, организованы часы, хранящие значения текущего момента времени в модели. Время в GPSS организовано так, что часы меняют свое значение только для того, чтобы указать время наступления ближайшего события.

Например: текущее значение часов модели равно 2, а очередное событие должно наступить в момент времени 7 (например: генерация нового сообщения), следовательно значение часов увеличивается сразу на 5 единиц. Основной условной единицей времени в модели можно выбрать любую единицу (секунда, минута, час, год), которая позволит получить необходимую точность моделирования.

2.1.2. Заявки (транзакты, сообщения)

Заявки, или сообщения являются абстрактными подвижными элементами, которые моделируют объекты реального мира: сообщения, программы, сбои. Перемещаясь между блоками модели, заявки вызывают (и испытывают) различные действия (задержка в некоторых точках модели, изменение маршрутов и направления движения, расщепление заявок на несколько копий).

2.1.3. Списки событий

Списки относятся к элементам внутренней организации системы GPSS. Они представляют собой структуры данных, в которых размещается полная информация о заявках. С помощью списков обеспечивается внутренняя логика работы моделирующей системы. Всего имеется 5 типов списков:

1. текущих событий;
2. будущих событий;
3. прерываний;
4. синхронизируемых;
5. пользователя.

и сообщение в любой момент времени может находиться в одном из списков.

В **списке текущих событий** находятся сообщения, соответствующие событиям, время наступления которых меньше (сообщения, которые должны были начать двигаться в некоторый момент в прошлом, но были заблокированы) или равно текущему времени. В списке текущих событий сообщения расположены в порядке убывания приоритета (т.е. сообщения с более высокими

приоритетами расположены ближе к началу списка); внутри класса сообщений с одинаковыми приоритетами сообщения располагаются в порядке поступления их в список. Каждое сообщение в списке текущих событий может находиться либо в активном состоянии (т.е. просматриваться в данный момент условного времени), либо в состоянии задержки. Если сообщение находится в активном состоянии, то процедура, осуществляющая просмотр, пытается продвигать эти сообщения к следующим блокам. Если движение сообщения блокируется каким-либо оборудованием, то вход в следующий блок выполнить невозможно, и сообщение переводится в состояние задержки и помещено в список задержки "оборудование занято" для данного оборудования.

Список будущих событий содержит сообщения, соответствующие событиям, время наступления которых больше текущего времени, т.е. событиям, которые должны произойти в будущем. Соответствующие этим событиям сообщения находятся в блоках, задающих время наступления события явным образом, например, в блоках ADVANCE и GENERATE.

Список прерываний содержит те отдельные сообщения, обслуживание которых соответствующими устройствами было прервано, а также те, которые вызвали прерывание.

Список синхронизируемых сообщений содержит сообщения, находящиеся в данный момент в состоянии сравнения.

Список пользователя содержит сообщения, удаленные пользователем из списка текущих событий и помещенные в список пользователя как временно неактивные.

2.1.4. Продвижение сообщений

Если сообщение в списке текущих событий находится в активном состоянии, то интерпретатор пытается его продвинуть, т.е. занять следующий блок по ходу движения сообщения.

Если в выполняемом блоке не задана явным образом задержка сообщения, интерпретатор сразу же пытается продвинуть сообщение через следующий блок. Этот процесс продолжается до тех пор, пока сообщение не получает отказ при попытке войти в блок, не встречает блок ADVANCE с заданной явно задержкой во времени или не уничтожается блоком TERMINATE или ASSEMBLE.

Если движение сообщения блокируется из-за состояния оборудования, то индикатор просмотра для этого сообщения устанавливается в "1", это означает, что сообщение перешло в состояние задержки. Сообщение помещается в соответствующий список задержки, и затем интерпретатор определяет, произошло ли изменение состояния оборудования.

Если движение сообщения прекращено из-за заданной для него задержки во времени, оно помещается в список будущих событий, и затем интерпретатор также проверяет, не изменилось ли состояние оборудования.

Если при обслуживании текущего активного сообщения произошло изменение состояния оборудования, просмотр начинается сначала, и снова обслуживаются все сообщения списка текущих событий, находящиеся в активном состоянии. Если изменение состояния оборудования не происходило,

интерпретатор снова обращается к списку текущих событий и проверяет, остались ли в нем сообщения, которые нужно обработать.

Описанная выше последовательность событий происходит для каждого нового момента условного времени до тех пор, пока счетчик завершения моделирования не станет равным 0 (описание блока START Приложение Б.); по выполнении этого условия моделирование прекращается.

2.2. Стандартные числовые атрибуты

Все объекты GPSS обладают свойствами – числовыми характеристиками, называемыми стандартными числовыми атрибутами (СЧА).

СЧА заявок:

PR1 – приоритет;

M1 – время, которое прошло с момента входа заявки в модель;

P\$j – параметр с номером j;

X\$j – номер транзакта.

MB\$j – флаг синхронизации:

1 – если сообщение в блоке j принадлежит тому же семейству, что и текущее сообщение;

0 – в противном случае.

Параметры заявок P\$j – это свойства заявки, определяемые пользователем. Параметры заявки являются локальными переменными, которые доступны только данной заявке. В процессе перемещения заявки по модели, ее параметры могут задаваться и модифицироваться в соответствии с логикой работы модели.

В GPSS существует ряд блоков, с помощью которых можно изменить значение параметров заявок. Блок ASSIGN модифицирует значение параметра заявки, входящего в этот блок. Для организации в модели циклов используется блок LOOP, с помощью которого можно предусмотреть многократное прохождение заявкой заданной цепочки блоков. Изменение приоритета заявки осуществляет блок PRIORITY.

2.3. Типовые конструкции программ

Рассмотрим алгоритмы функционирования типовых узлов моделей систем в терминах GPSS:

Задача 1. Очередь к прибору с ограниченным количеством мест в очереди:

Схема функционирования данной задачи приведена на Рисунке 1.

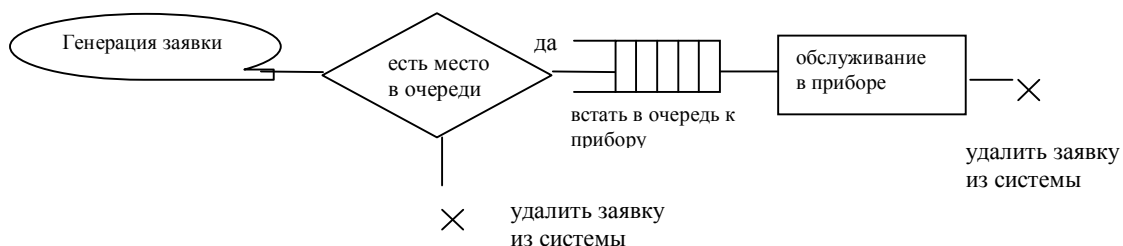


Рисунок 1 – Схема функционирования модели системы с потерями

Необходимо отметить, что очередь с ограниченным числом мест можно реализовать в GPSS, используя накопитель (память).

Задача 2. Параллельная обработка заявки двумя приборами.

Схема функционирования данной задачи приведена на Рисунке 2.



Рисунок 2 – Схема функционирования модели системы с параллельной обработкой заявки двумя приборами

Из приведенной схемы видно для того, чтобы проимитировать параллельную обработку заявки двумя приборами необходимо:

- создать копию заявки, поступающей на обслуживание;
- заявка направляется на первый прибор, где обслуживается заданное время;
- указанная копия направляется на второй прибор, где обслуживается заданное время;
- после обработки объединить заявку со своей копией.

Задача 3. Два потока заявок, поступающих к одному прибору.

Схема функционирования данной задачи приведена на Рисунке 3.

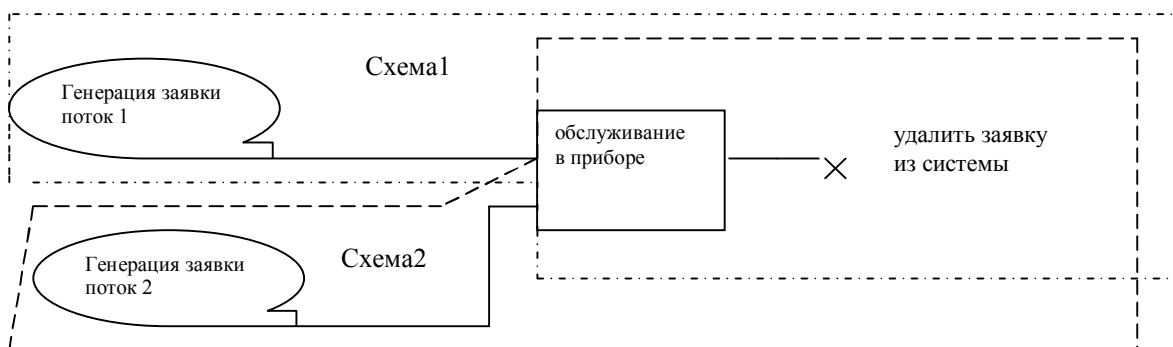


Рисунок 3 – Схема функционирования модели системы с двумя потоками к одному обслуживающему прибору

Имитацию системы с двумя потоками можно разложить на две независимые схемы:

Схема 1 – Поток 1 поступает к прибору и обслуживается в приборе.

Схема 2 – Поток 2 поступает к прибору и обслуживается в приборе.

Соответственно программа на GPSS может быть представлена двумя независимыми наборами операторов, в каждый из которых входит набор блоков, описывающих работу устройства обслуживания (SEIZE-ADVANCE-RELEASE).

Захват прибора заявками из первого или из второго потока будет зависеть от времени появления заявки в системе. Если второй поток прерывает первый, то соответственно первый набор будет включать группу SEIZE-ADVANCE-RELEASE, второй PREEMPT-ADVANCE-RETURN, обеспечивающую прерывание работы прибора и возврат его в исходное состояние.

Задача 4. Поток заявок, поступающих к прибору, проходит несколько циклов обслуживания.

Схема функционирования данной задачи приведена на Рисунке 4.

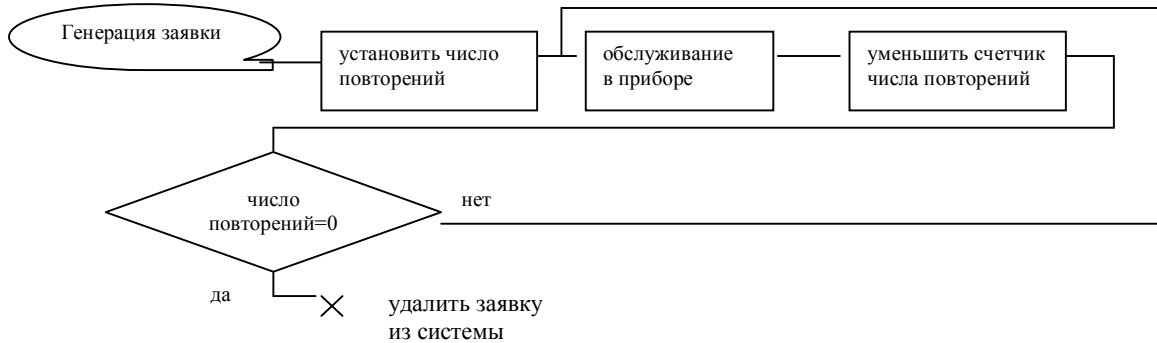


Рисунок 4 – Схема функционирования модели системы с заданным количеством циклов обработки заявки в устройстве

Итак, для того, чтобы обеспечить полное обслуживание заявки, она должна пройти несколько циклов обработки, т.е. многократно занимать и покидать устройство. Так например: заявка1 должна пройти четыре цикла обработки, после второго цикла заявка1 в очередной раз покидает устройство, но в очередь к устройству уже поставлена заявка2, следовательно устройство будет занято заявкой2, которая пойдет на свой первый цикл обслуживания в устройстве, после первого цикла обслуживания заявка2 покидает устройство и устройство занимает заявка1. Пояснение данного примера приведено на диаграмме – рисунок 5. Принцип функционирования системы GPSS описан в пунктах 2.1.3. и 2.1.4.



t1 – t16 –отметки времени наступления событий

Рисунок 5 – Диаграмма функционирования модели системы с заданным количеством циклов обработки заявки в устройстве

Установка счетчика числа повторений (циклов), для каждой заявки производится блоком ASSIGN, с помощью которого можно произвести установку параметра (собственной переменной) заявки в заданное значение.

Задача 5. Поток заявок распределяется на два прибора, четные заявки обрабатываются на первом приборе, нечетные на втором.

Схема функционирования данной задачи приведена на Рисунке 6.

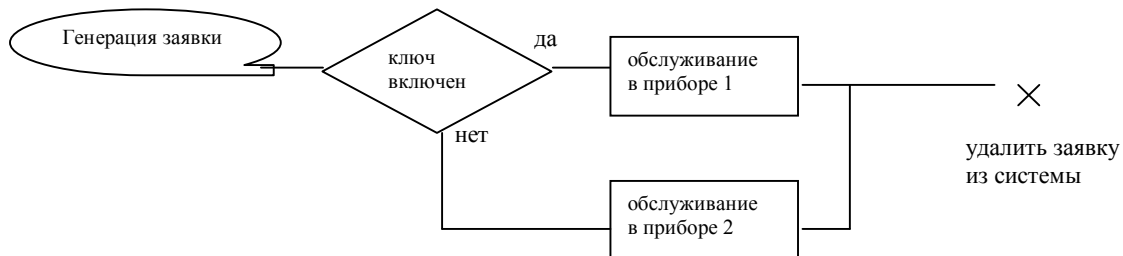


Рисунок 6 – Схема функционирования модели системы с перенаправлением потока на два прибора

В модели можно использовать только предварительно проинициализированные ключи (Приложение Б).

3. Ход работы

1. Запрограммировать имитационные модели, согласно варианта - Таблица 1.
2. Предъявить преподавателю действующие программы.
3. Обработать полученный статистический материал, вывести графические зависимости $\bar{t}$ и Util-коэффициента, использования прибора.
4. Оформить отчет по работе.

4. Содержание отчета

1. Цель работы.
2. Текст программ.
3. Графики зависимости: длины очереди, Util и среднего времени пребывания заявки в системе от времени моделирования.
4. Выводы по работе.

5. Варианты заданий

Варианты заданий приведены в Таблице 1.

Таблица 1 – Варианты заданий

Вариант	Номера задач	Значения параметров
1	1	X1=100,X2=50,X3=70,X4=20 X23=40,X24=15,C=10
	4	
	9	
2	10	X5=200,X6=50,X7=100,X8=30,X9=50,X10=10,X11=15, X12=7
	8	X17=300,X18=50,X19=70,X20=20,X21=30,X22=15

Продолжение таблицы 1

Вариант	Номера задач	Значения параметров
3	1	$X_1=200, X_2=60, X_3=100, X_4=45$
	5	$X_{13}=70, X_{14}=25, X_{15}=45, X_{16}=20$
	11	$P_1=0,4, P_2=0,6$
4	1	$X_1=100, X_2=50, X_3=70, X_4=20$
	7	$S_1=10$
	9	$X_{23}=33, X_{24}=13, C=10$
5	1	$X_1=120, X_2=45, X_3=40, X_4=15$
	2	
	5	$X_{13}=70, X_{14}=20, X_{15}=35, X_{16}=15$
6	1	$X_1=12, X_2=5, X_3=4, X_4=1$
	4	
	8	$X_{17}=10, X_{18}=5, X_{19}=7, X_{20}=2, X_{21}=3, X_{22}=2$
7	2	$X_5=20, X_6=5, X_7=10, X_8=3, X_9=5, X_{10}=1, X_{11}=20, X_{12}=7$
	6	$X_{17}=30, X_{18}=5, X_{19}=7, X_{20}=2, X_{21}=3, X_{22}=2$
8	1	$X_1=220, X_2=30, X_3=44, X_4=22$
	10	
	9	$X_{23}=34, X_{24}=22, C=7$
9	3	$X_5=440, X_6=66, X_7=88, X_8=23, X_9=50, X_{10}=20, X_{11}=15, X_{12}=5$
	8	$X_{17}=220, X_{18}=50, X_{19}=50, X_{20}=20, X_{21}=60, X_{22}=15$
10	1	$X_1=400, X_2=50, X_3=120, X_4=20$
	10	$X_{13}=50, X_{14}=20, X_{15}=40, X_{16}=20$
	6	$P_1=0,8, P_2=0,2$
11	1	$X_1=100, X_2=50, X_3=70, X_4=20$
	11	$S_1=10$
	9	$X_{23}=33, X_{24}=13, C=10$
12	1	$X_1=100, X_2=30, X_3=30, X_4=20$
	4	
	5	$X_{13}=50, X_{14}=20, X_{15}=40, X_{16}=10$
13	1	$X_1=15, X_2=8, X_3=8, X_4=4$
	7	
	8	$X_{17}=15, X_{18}=5, X_{19}=8, X_{20}=6, X_{21}=4, X_{22}=1$
14	3	$X_5=15, X_6=4, X_7=15, X_8=2, X_9=4, X_{10}=2$
	6	$X_{11}=15, X_{12}=5, X_{17}=20, X_{18}=5, X_{19}=5, X_{20}=2, X_{21}=4$
	8	$X_{22}=2$

Согласно варианта из Таблицы 2 необходимо выбрать номера задач, имитационные модели, которых необходимо запрограммировать.

Таблица 2 – Словесное описание типовых конструкций программ

Номер задачи	Описание типовых конструкций программы	Примечание
1	Процесс прохождения заявок, поступление которых подчиняется равномерному закону с интервалом $X1\pm X2$ единицы времени, а обработка – равномерному закону со средним временем $X3\pm X4$ единицы. Требуется получить статистические данные об очереди заявок, ожидающих обслуживания в приборе	Воспользуйтесь оператором QUEUE и DEPART
2	На обработку поступает два потока заявок : первый – с интервалом $X5\pm X6$ единиц, второй $X7\pm X8$ единицы. Время обработки заявок первого потока $X9\pm X10$ единицы, второго $X11\pm X12$ единицы. а) потоки имеют равные приоритеты обработки; б) приоритет 1-го потока выше приоритета 2-го потока	Воспользуйтесь блоками ASSIGN, TRANSFER (в безусловном режиме).
3	На обработку поступает два потока заявок : первый – с интервалом $X5\pm X6$ единиц, второй $X7\pm X8$ единицы. Второй поток прерывает обработку заявок первого. Время обработки заявок первого потока $X9\pm X10$ единицы, второго $X11\pm X12$ единицы.	Воспользуйтесь блоком PREEMPT, RETURN
4	В условиях задачи 1 необходимо произвести параллельную обработку заявки двумя приборами.	Блоки SPLIT, ASSEMBLE и TRANSFER (в безусловном режиме).
5	В условиях задачи 1 сообщения могут обрабатываться на одном из двух приборов : на первом – со временем $X13\pm X14$ единицы, на втором $X15\pm X16$ единицы. Причем предпочтительнее обработка на первом приборе.	Воспользуйтесь блоком TRANSFER в режиме BOTH.
6	В условиях задачи 5 заявки поступают к блоку THIS с вероятностью $P1$ и к блоку THAT с вероятностью $P2$.	Блок TRANSFER в вероятностном режиме работы.
7	В условиях задачи 1 заявки поступают на обработку с ограниченным числом мест в очереди, равным $S1$. Если очередь заполнена, то заявки покидают систему.	Блоки STORAGE, GATE (SNF), ENTER, LEAVE

Продолжение таблицы 2

Номер задачи	Описание типовых конструкций программы	Примечание
8	На обработку поступает поток заявок с интервалом $X_{17} \pm X_{18}$ единиц времени. Нечетные заявки обрабатываются на первом приборе со временем $X_{19} \pm X_{20}$ единицы, четные заявки – на втором приборе со временем $X_{21} \pm X_{22}$ единицы.	Воспользуйтесь блоками LOGIC (I), GATE (LS), INITIAL.
9	В условиях задачи 1 каждая заявка проходит С-циклов обработки на приборе со временем $X_{23} \pm X_{24}$	Воспользуйтесь блоками ASSIGN, LOOP.
10	На обработку поступает два потока заявок : первый – с интервалом $X_5 \pm X_6$ единиц, второй $X_7 \pm X_8$ единицы. Оба потока поступают на обработку с ограниченным числом мест в очереди, равным S_1 . Если очередь заполнена, то заявки покидают систему. Время обработки заявок для каждого потока $X_9 \pm X_{10}$. Оценить потери для каждого потока.	Блоки STORAGE, GATE (SNF), ENTER, LEAVE
11	В условиях задачи 1 необходимо произвести параллельную обработку заявки тремя приборами.	Блоки SPLIT, ASSEMBLE и TRANSFER (в безусловном режиме).

Описание требуемых для выполнения лабораторной работы блоков GPSS приведено в Приложении Б.

6. Контрольные вопросы

1. Внутренняя организация GPSS – списки событий?
2. Как осуществляется продвижение сообщения в системе?
3. Назначение базовых блоков языка GPSS – организация памяти, повторений обработки, ключей?
4. Какие блоки, изменяют порядок прохождения блоков сообщениями?
5. Как производится обработка сообщений, принадлежащих одному семейству?
6. Какие объекты, можно моделировать, используя язык GPSS?

ЛАБОРАТОРНАЯ РАБОТА №3.

ИЗУЧЕНИЕ ТЕХНОЛОГИИ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

1. Цель работы:

Исследование технологии имитационного моделирования на примере имитационной модели мультипрограммной вычислительной системы (ВС).

2. Основные положения

2.1. Моделирование вычислительных систем

Ресурсы вычислительной системы – это различные устройства (процессор, дисплей, клавиатура, накопители на магнитных дисках и т.д.), а также память, файлы, программные модули. В случае занятости нужного ресурса, который требует программа, к нему создается очередь. Таким образом, если программы пользователей, которые выполняются компьютером используют одинаковые ресурсы в некоторые промежутки времени, то выполнение таких программ будет существенно задерживаться.

Компьютерная система обычно работает в мультипрограммном или многозадачном режиме. В любом режиме работы (мультипрограммном или многозадачном) компьютер с одним процессором в каждый момент времени выполняет только одну программу (одну команду). Если используется мультипроцессорная система с несколькими одинаковыми процессорами, то становится возможным выполнение нескольких программ на разных процессорах. Большей частью такая система моделируется как многоканальная СМО в отличие от СМО с одним устройством обслуживания, которая моделирует работу однопроцессорной системы.

При выполнении операций ввода-вывода для записи или считывания данных с магнитных носителей информации, эти операции большей частью осуществляются через буфер ввода-вывода. Обмен данными между памятью и внешними устройствами осуществляется через контроллеры ввода-вывода, но скорость их работы на порядок больше скорости электромеханических устройств. При моделировании можно считать, что эти операции выполняют сами устройства. Поиск информации на магнитных дисках связан с перемещением головок на нужный сектор и с непосредственным выполнением операции записи или считывания данных, которые зависят от скорости вращения диска. Поэтому для моделирования времени поиска информации используют равномерное распределение, для которого минимальное и максимальное время берут из технических характеристик устройств.

Таким образом, работу вычислительной системы можно представить как работу сети СМО.

При моделировании обычно исследуется время пребывания программы пользователя в системе или число программ выполненных системой за единицу времени. Типичная программа пользователя может описываться: временем поступления в систему, емкостью, занимаемой, памяти, количеством запросов к устройствам ввода-вывода и распределением их по времени, количеству данных, которые вводятся и выводятся на тех или иных устройствах.

2.2. Постановка задачи и содержание работы

2.2.1. Описание функционирования вычислительной системы

В состав мультипрограммной ВС входят (рисунок 1):

- а. центральный процессор;
- б. оперативная память;
- в. селекторный канал с запоминающими устройствами на магнитных дисках.

Для обеспечения мультипрограммирования оперативная память выделяется заданиям фиксированными разделами.

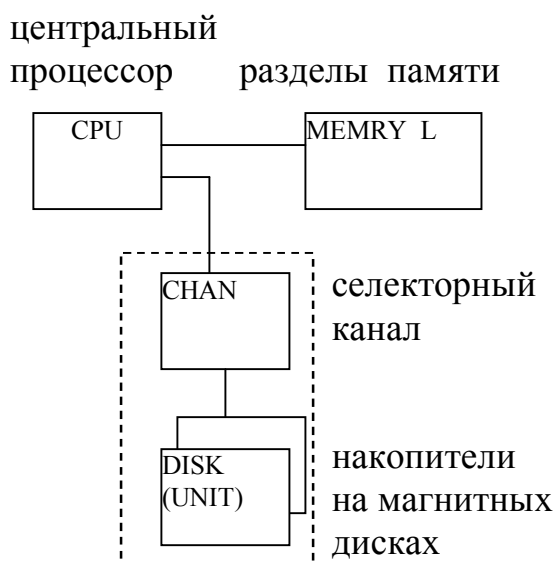


Рисунок 1 – Структура моделируемой системы

Система обрабатывает поток заданий. Процесс выполнения задания (рисунок 2) состоит из следующих этапов:

- а. запуска задания;
- б. цикла выполнения задания;
- в. завершения обработки.

На этапе запуска задания, поступающее задание помещается в очередь системы (блок 1). Если есть доступный раздел памяти, то он выделяется заданию (блок 2). При отсутствии доступного раздела, задание ждет освобождения памяти.

На этапе цикла выполнения задания - фазы обработки в процессоре повторяются (блок 2.1) и выполняются операции ввода-вывода (блок 2.2). Фаза ввода-вывода в свою очередь разбивается на операции установки головок на магнитном диске (блок 2.2.1) и передачи данных через селекторный канал (блок 2.2.2) в оперативную память.

На этапе завершения обработки (блок 3) освобождается память и выводится задание из системы.

Входные характеристики заданий: объем обработки в процессоре, число операций ввода-вывода, интенсивность поступления заданий, средний объем памяти программы, распределение обращений к дискам.

Входные характеристики системы: объем доступной для программ памяти, характеристики дисков (количество, время установки головок, время передачи данных), число каналов.

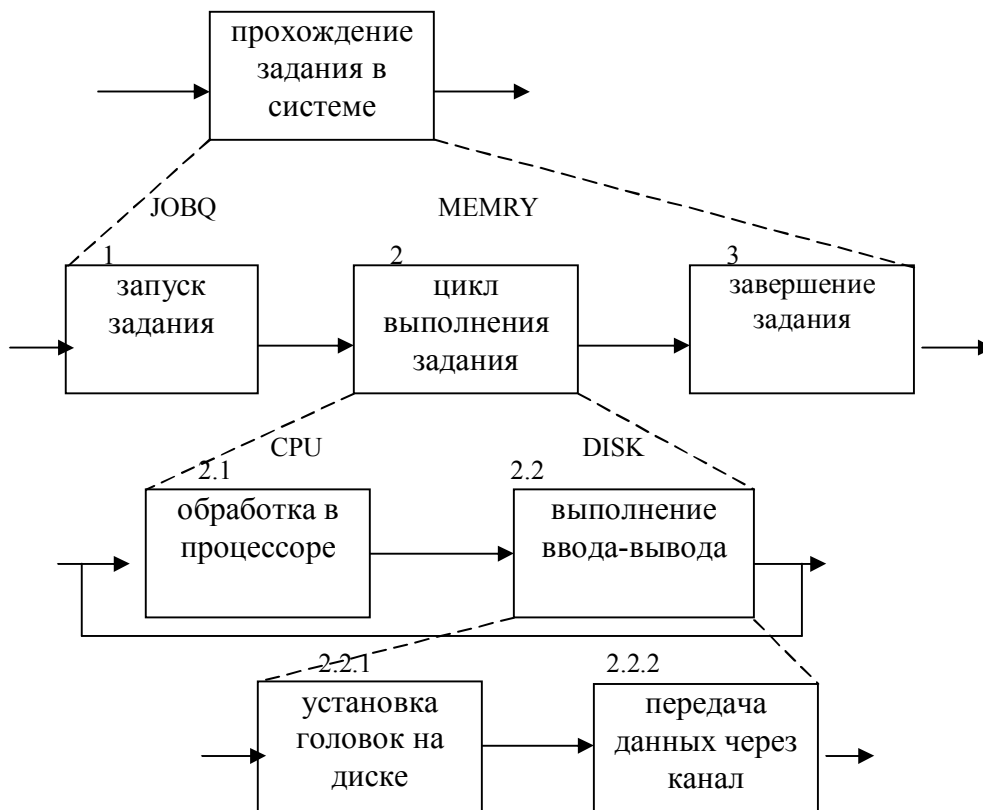


Рисунок 2 – Структура моделируемого процесса

Изменяя характеристики заданий или конфигурацию системы, можно исследовать следующие выходные характеристики:

- загрузку устройств,
- среднее время: выполнения заданий, операций ввода-вывода,
- время прохождения задания,
- уровень мультипрограммирования.

2.2.2. Описание программы имитационной модели

При построении программы модели был сделан ряд допущений, позволяющих упростить модель. Например, все задания считаются статистически однородными, обслуживание запросов к процессору, дискам и каналам производится в порядке очередности поступления, поток заданий предполагается пуассоновским, а распределение времени обслуживания - равномерным или детерминированным. Не учитывается зависимость доступа к диску от размещения информации и периодичности вращения носителя. Число операций ввода-вывода считается фиксированным, а распределение обращений к накопителям - равномерным. Время передачи данных принимается равным времени оборота диска.

Параметрами модели являются :

- 1) емкость многоканального устройства, равная числу каналов;
- 2) емкость многоканального устройства, равная числу дисков;
- 3) распределение номеров дисков, определяемое распределением обращений к дискам;
- 4) емкость многоканального устройства, соответствующая числу разделов памяти и зависящая от объема памяти и среднего размера программы;
- 5) среднее время обслуживания в устройстве, зависящее от объема вычислений, быстродействия процессора и числа операций ввода-вывода;
- 6) среднее время обслуживания в устройстве, равное времени передачи данных;
- 7) средний интервал поступления заданий;
- 8) масштаб времени в модели.

Текст базовой программы моделирования мультипрограммной вычислительной системы приведен в Приложении В.

2.2.3. Порядок проведения исследования

Необходимо провести исследование модели Приведенной в приложении В, изменяя интенсивность входного потока заданий в соответствии со значениями, приведенными в таблице 1.

Таблица 1 – Значение параметров проведения исследования модели

Вариант прогона	Значение интенсивности заданий/с	Примечание
1	0.1	Необходимо помнить, что в приведенной модели все значения даны в миллисекундах.
2	0.2	
3	1.0	
4	1.5	
5	2.0	
6	5.0	

2.2.4. Анализ результатов моделирования

При анализе результатов работы системы необходимо проанализировать полное время обслуживания заявки, которое фиксируется в таблице JTIME – параметр M1 (описание блоков организации GPSS – таблиц приведено в Приложении Б). Стандартный числовой атрибут M1, измеряет время, которое прошло с момента входа заявки в модель и до момента ее фиксации (табуляции) в таблице. Значение M1 для каждой заявки изменяется в процессе моделирования. Оценивается как гистограмма распределения времени обслуживания заявок, так и усредненное значение времени обслуживания для каждого прогона модели.

Исследовать изменение уровня мультипрограммирования при изменении интенсивности входного потока.

3. **Ход работы**

1. Получение и анализ варианта задания Таблица 2.
2. Прогон базового варианта модели (Приложение В), в соответствии с п.2.2.3.
3. Сбор статистических данных по каждому варианту прогона.
4. Расчет новых параметров модели, в соответствии с вариантом задания .
5. Изменение текста программы и отладка программ модели.
6. Постановка задания на моделирование, в соответствии с п.2.2.3.
7. Сбор статистических данных по каждому варианту прогона.
8. Анализ результатов моделирования – сопоставление базового и измененного варианта.
9. Оформление отчета по работе.

4. **Содержание отчета**

1. Входные характеристики системы в базовом и измененном вариантах.
2. Расчет параметров моделирования.
3. Распечатка результатов трансляции и выполнения модели (или показать на экране).
4. Таблица входных характеристик системы.
5. Оценка результатов с точки зрения администрации системы и ее пользователей.
6. Графики зависимостей выходных характеристик системы от интенсивности потока заданий.
7. Характеристики программы модели (объем памяти, время трансляции и моделирования, среднее время исполнения блока).

5. **Варианты заданий**

Каждое задание предполагает исследование характеристик измененной системы и их сравнение с характеристиками исходной системы. Изменению подвергаются входные характеристики заданий и программ или состав и структура ВС.

Таблица 2 – Варианты заданий

№ варианта	Описание задания
1	В системе планируется добавить дополнительный селекторный канал. А также произвести оптимизацию программы по объему занимаемой памяти до 30%.
2	В результате оптимизации программы во время выполнения удалось сократить объем вычислений в процессоре на 40% и уменьшить количество обращений к дискам на 20%. Однако в 2,5 раза увеличился общий объем программы за счет роста массивов данных.

Продолжение таблицы 2

№ варианта	Описание задания
3	В системе произведена замена процессора и двух дисков на более быстродействующие. Быстродействие ЦП возросло на 50%, новые диски имеют время оборота 20мс и установку головок 0...40. Обеспечить обслуживание 60% обращений новыми дисками.
4	При оптимизации программы по объему занимаемой памяти для ее сокращения вдвое потребовалось увеличить объем вычислений на 50% и количество обращений к дискам – на 25 %.
5	На ВЦ установлена новая ЭВМ. Процессор имеет в 2,5 раза большее быстродействие. Объем памяти, доступной для программ, увеличился на 60%. Подсистема ВЗУ имеет два диска большой емкости с временем оборота 17 мс и временем установки 0...50 мс. Ожидается, что 65% всех обращений будут адресованы к одному диску.
6	Организация программы в виде структуры с перекрытиями (оверлеями) позволила уменьшить объем раздела памяти на 37%. Но это привело к увеличению вдвое числа обращений к одному из дисков.
7	На ВЦ поступает два потока заданий. Второй поток оптимизирован по объему занимаемой памяти, но для ее сокращения вдвое потребовалось увеличить для второго потока объем вычислений на 50% и количество обращений к дискам – на 25 %.
8	В системе планируется замена дисков на более быстродействующие с временем доступа 5...65 мс, но не обеспечивающие предварительную установку головок на дисках. Ожидается снижение числа обращений на 37,5% при увеличении объема раздела памяти на 25%.
9	В вычислительной системе расширяется подсистема ВЗУ за счет добавления двух дисков с устройствами управления и одного селекторного канала. Новые диски имеют время оборота 17 мс и установки 0...50мс. Ожидается, что они обслужат 50% обращений. Конфигурация ВЗУ обеспечивает два пути доступа к каналам.
10	Использование сблокированного формата записей на диске позволило снизить в 5 раз количество обращений программы к одному из дисков, но при этом за счет роста буферов ввода/вывода объем программы увеличился на 25%.
11	На ВЦ поступает два потока заданий. Для обслуживания заданий второго потока требуется объем вычислений в процессоре на 40% меньший в сравнение с исходным потоком и меньшее количество обращений к дискам на 20%. Однако задания второго потока имеют в 2 раза больший общий объем программы за счет больших массивов данных.

Продолжение таблицы 2

№ варианта	Описание задания
12	В системе произведена замена процессора и основной памяти. Быстродействие ЦП возросло на 150%. При увеличении степени мультипрограммирования в 3 раза ожидается 100-процентный прирост обращений к дискам.
13	На ВЦ поступает два потока заданий. Второй поток использует сблокированный формат записей на диске, что позволило снизить в 5 раз количество обращений программы к одному из дисков, но при этом за счет роста буферов ввода/вывода объем программы увеличился на 25%.
14	В системе планируется добавить дополнительный селекторный канал и произвести увеличение основной памяти в два раза. А также произвести замену дисков на более быстродействующие с временем доступа 5...45 мс.

6. Контрольные вопросы

1. Какие допущения приняты при построении модели?
2. Перечислите входные характеристики заданий и системы?
3. Перечислите выходные характеристики системы?
4. Как связаны параметры модели с характеристиками заданий и системы?
5. В чем заключается противоречие требований к эффективности системы администрации ВС и ее пользователей?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Советов Б.Я. Моделирование систем/ Б.Я.Советов, С.А.Яковлев – М.: Высш.шк.,1999. – 271с.
2. Кузин Л.Т. Основы кибернетики/ Л.Т.Кузин – Л.: Энергия, 1978. – 564с.
3. Шрайбер Т.Дж. Моделирование на GPSS/ Т.Дж. Шрайбер – М.: Мир,1980.– 592с.
4. Основы теории вычислительных систем/ С.А.Майоров, Г.И. Новиков, Т.И.Алиев, и др; Под.ред. С.А.Майорова – М.: Высш.шк.,1978.–408с
5. Томашевский В.Имитационное моделирование в среде GPSS/ В.Томашевский, Е.Жданова – М.: Бестселер,2003.–412с.
6. Разработка САПР: В 10 кн. Кн. 10. Лабораторный практикум на базе учебно-исследовательской САПР: Практ. пособие / А.В.Петров, В.М.Черненко, В.В.Тимофеев, и др; Под.ред. А.В.Петрова – М.: Высш. шк., 1991. – 160 с.

- entry count – число заявок, входивших в блок за все время моделирования;
- current count - текущее число заявок, находящихся в блоке в момент окончания моделирования;
- retry - число заявок, ожидающих специальных условий, зависящих от состояния данного блока.

На приведенном фрагменте отображена информация по трем блокам: поток сообщений составил 2006 заявок; на момент окончания моделирования 8 заявок находятся в очереди к прибору, ожидая обработки; 4 заявки в приборе находятся в режиме специальных условий (обработка была прервана).

– Устройства

FACILITY	ENTRIES	UTIL.	AVE._TIME	AVAILABLE	OWNER	PEND	INTER	RETRY	DELAY
CPU	1998	0.752	15.00	0	0	0	0	0	0

- facility - имя или номер устройства;
- entries - число обслуживаний, или число входов в устройство;
- util. - коэффициент использования устройства - часть периода моделирования, в течение которого устройство было занято;
- ave.time - среднее время обслуживания одной заявки;
- available- состояние готовности устройства: 1 -готово, 0 - не готово;
- owner - номер последнего обслуженной заявки.

В системе используется одноканальное устройство с именем CPU , число входов в устройство 1998, коэффициент использования устройства – 0.752 единицы, среднее время обслуживания одной заявки – 15.00 единиц времени, на момент окончания моделирования устройство используется,

– Очереди

QUEUE	MAX	CONT.	ENTRIES	ENTRIES(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
M	10	8	2006	1575	0.04	0.82	3.83	0

- queue – имя или номер очереди;
- max – максимальная длина очереди;
- cont. – текущее содержимое очереди в конце периода моделирования;
- entries – число входов в очередь;
- entries(0)– число нулевых входов - число заявок, входивших в очередь с нулевым временем пребывания в ней;
- ave.count – средняя длина очереди;
- ave.time – среднее время пребывания заявок в очереди;
- ave.(-0) – среднее время, проведенное в очереди без учета "нулевых" входов в очередь;

! Если entries = entries (0) – следовательно, очередь в процессе работы не формировалась.

Приводится статистика по очереди с именем М: число входов в очередь – 2006 заявок, 431 заявка за время моделирования в очереди не задерживалась, max длина очереди (число одновременно задерживаемых на входе в устройство заявок) составила – 10 единиц, на момент окончания моделирования длина очереди – 8 заявок, среднее время пребывания заявки в очереди – 0.82 единиц времени.

A.1.5. Базовые блоки языка GPSS

A.1.5.1. Блок GENERATE

Блок GENERATE является источником потока сообщений в модели. В данном блоке производится подготовка заявок и запуск их в модель через интервалы времени, заданные пользователем.

GENERATE [$\langle A \rangle$],[$\langle B \rangle$],[$\langle C \rangle$],[$\langle D \rangle$],[$\langle E \rangle$]

Операнд А – среднее значение интервала времени между моментами генерации сообщений;

Операнд В – задает модификатор среднего значения

Может быть два типа модификаторов:

модификатор – интервал – задает равномерный закон распределения;

GENERATE 25, 10 – генерируются заявки с интервалом прихода от 15 до 35 единиц времени (т.е. 25 ± 10).

- модификатор - функция.

GENERATE 25, FN\$EXPN - генерируются заявки, интервалы времени между ними определяются произведением числа 25 на значение функции FN\$EXPN ($25 * FN$EXPN$)

Операнд С – задает начальную задержку, т.е. время появления первой заявки;

Операнд D – задает общее число генерируемых заявок – максимальное число сообщений, которое будет создано в блоке GENERATE. Если операнд D пуст, то блок генерирует неограниченное число сообщений;

Операнд E – определяет уровень приоритета каждого сообщения.

GENERATE 25, 10, ,400, 7 – генерируются заявки с уровнем приоритета 7, интервал времени между ними определяется равномерным законом 25 ± 10 , максимально будет создано 400 сообщений.

! Нельзя использовать в качестве значения операнда параметры сообщения.

! Все СЧА, описанные в блоке GENERATE, должны быть уже определены. В модели блоку GENERATE должны предшествовать операторы описания (инициализации) INITIAL, FUNCTION и VARIABLE.

! При использовании блока GENERATE необходимо помнить, что сообщение не должно входить в блок GENERATE. Если сообщение пытается это сделать, возникает ошибка выполнения.

A.1.5.2. Блок TERMINATE

Блок TERMINATE удаляет из модели входящие сообщения.

TERMINATE [<A>]

Операнд А – задает число единиц, на которое этот блок изменяет содержимое счетчика завершений (операнд А карты START), определяющего момент окончания моделирования. Если операнд А пуст, то сообщение уничтожается, а содержимое счетчика не изменяется.

A.1.5.3. Блок ADVANCE

Блок ADVANCE моделирует работу устройства (совместно с блоками занятия и освобождения устройств), выполняя задержку продвижения заявки на заданный период времени.

ADVANCE <A>,[]

Операнд А задает среднее время пребывания сообщения в блоке ADVANCE. Операнд В указывает способ модификации среднего значения, заданного в поле А.

Может быть два типа модификаторов:

- модификатор-интервал – ADVANCE 5,1 – задержка выбирается из интервала [4,6];
- модификатор-функция – ADVANCE 100,FN\$EXPN – задержка формируется, как произведение 100 на значение функции EXPN.

Т.е действие модификаторов аналогично действию модификаторов операнда В блока GENERATE.

A.1.5.4. Блоки описывающие устройства

Устройства используются при моделировании систем для имитации работы оборудования единичной емкости, например, процессора, каналов передачи данных, устройства ввода-вывода, линии связи и т.д. Такое оборудование в любой момент времени может обрабатывать только одно сообщение.

Блок SEIZE

Свободный блок SEIZE позволяет вошедшему в него сообщению занять указанное устройство.

SEIZE <A>

Операнд А – задает номер или имя занимаемого устройства.

Когда устройство занято, сообщение задерживается на входе в блок SEIZE. В этом случае оно помещается в список задержки, связанный с этим устройством. Сообщение, которое заняло устройство может быть прервано на этом устройстве другим сообщением.

Блок RELEASE

Блок RELEASE предназначен для освобождения устройства тем сообщением, которым оно было занято.

RELEASE <A>

Операнд А – задает номер или имя занимаемого устройства.

A.1.5.5. Блоки описывающие объекты типа "очередь"

В GPSS объекты типа "очередь" вводятся для сбора статистических данных. Статистика об очередях собирается в моменты входа сообщения в блок QUEUE (вход в очередь) или в блок DEPART (выход из очереди); задержку сообщений в очереди вызывает не очередность сообщений, а условия, при которых сообщение входит в очередь (например, состояния объектов типа "устройство"). Только объекты типа "очереди" обеспечивают пользователя данными относительно задержки.

! Использование очередей никоим образом не влияет на внутренние операции GPSS.

Блок QUEUE

Блок QUEUE увеличивает длину очереди.

QUEUE <A>[,]

Операнд А – задается номер или имя очереди;

Операнд В – определяет число единиц, на которое увеличивается текущая длина очереди. Если операнд В пуст, то прибавляется единица.

Одно и то же сообщение может одновременно увеличить длину нескольких очередей, т.е. сообщение может войти в несколько блоков QUEUE перед тем, как войти в соответствующие блоки DEPART.

QUEUE 3 – Увеличивает длину очереди номер 3 на единицу.

QUEUE QUE, 4 – Увеличивает длину очереди с именем QUE на 4 единицы.

Блок DEPART

Блок DEPART служит для уменьшения длины очереди.

DEPART <A>[,]

Операнд А – задает номер или имя очереди;

Операнд В – задает число единиц, на которое уменьшается длина очереди. Это число не должно превышать текущую длину очереди. Если операнд В пуст, длина очереди уменьшается на единицу.

DEPART 5 – Уменьшает длину очереди номер 5 на единицу.

DEPART QUE,2 – Уменьшает длину очереди с именем QUE на 2 единицы.

A.1.6. Операторы контроля управления

A.1.6.1. Оператор START

Оператор START используется для инициирования начала моделирования.

START <A>

Операнд А – задает значение счетчика завершений, определяющее момент окончания прогона модели (в процессе счета сообщения будут входить в блоки TERMINATE; операнд А блока TERMINATE может задавать число единиц, вычитаемое из счетчика завершений, заданного в операнде А оператора START).

Процесс моделирования или выполнение программы идет до тех пор, пока счетчик завершения моделирования не достигнет нуля.

! Обычно для управления временем моделирования в программу вводится отдельный сегмент вида

```
GENERATE nn
TERMINATE 1
START kn
```

Тогда время моделирования будет определяться произведением указанных параметров

$kn * nn$

Данный сегмент получил название – системных часов.

А.1.6.2. Способы задания и описания функций

Используя функции, пользователь может производить вычисление непрерывных или дискретных функциональных зависимостей. Функции широко применяются, например, для задания случайного интервала времени между генерацией двух сообщений.

Оператор описания функций имеет следующий формат:

<NAME> FUNCTION <A>,

Метка NAME определяет имя функции (задается с начала строки, длина имени не менее четырех символов).

Операнд А – содержит аргумент (независимую переменную) функции. Если в качестве аргумента функции используется генератор случайных чисел RN_j , то значения аргумента будут числа с плавающей точкой в диапазоне $[0,1]$;

Операнд В – определяет тип и количество пар точек функции;

тип функции: С – непрерывная

D – дискретная.

За операндом В следует наборы пар чисел для задания координат $X[i]$ и $Y[i]$ функции.

! Не допускается использование комментариев между оператором описания функции и операторами задания координат.

Правила при написании операторов, задающих значение координат:

- запись должна начинаться с позиции 1;
- значение координат $X[i]$ и $Y[i]$ одной точки функции разделяются запятой;
- наборы координат разделяются знаком /
- каждое последующее значение $X[i]$ должно быть больше предыдущего;
- каждая функция должна иметь, по крайней мере, две описанные точки;
- длина строки не должна превышать 80 символов.

Например:

```
EXPN FUNCTION RN1,C8
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2
```

Описана функция с именем EXPN, функция непрерывная, задается 8 точек. Генерируется число в диапазоне $[0,1]$ – значение координаты $X[i]$ (например: $X=0.2$), в соответствии полученному числу ставится значение координаты $Y[i]$ ($Y=0.222$), которое и будет являться значением функции EXPN ($EXPN=0.222$).

Следовательно, если функции используется в операторе

```
GENERATE 25,FN$EXPN
```

то следующая заявка будет сгенерирована через $25 * 0.222 = 5.55$ условных единиц времени

ПРИЛОЖЕНИЕ Б

(обязательное)

Базовые блоки системы имитационного моделирования GPSS

Б.1.1. Формирование файла отчета выходной статистики.

Система автоматически подготавливает стандартный вывод данных по используемым в модели элементам.

– Памяти

storage – имя или номер объекта типа "многоканальное устройство" – память;

cap. – емкость данной памяти;

remain – число единиц свободной емкости памяти в конце периода моделирования;

util. – коэффициент использования памяти - период моделирования, в течение которого память использовалась;

entries – число входов в память;

min – минимальное количество используемой емкости памяти за период моделирования;

max – максимальное количество используемой емкости памяти за период моделирования.

– Включенные логические ключи

logicswitch – имя или номер объекта типа логический ключ;

value – значение логического ключа в конце моделирования.

1 – "установлен" или "истина",

0 – "сброшен" или "ложь".

– таблицы

Для накопления выборочных значений случайных величин и статистической обработки этих выборок используются GPSS – таблицы. В модели может быть несколько таблиц.

Если в модели используются блоки TABLE или QTABLE в файле стандартной статистики будет представлена информация о таблицах.

table – определяет имя или номер объекта типа "таблица";

mean – среднее взвешенное значение табулируемого аргумента. Значение преобразовывается в формат двойной точности при выводе в файл статистики;

std.dev – определяет взвешенное среднеквадратичное отклонение.

$S.D. = \sqrt{SOS/(COUNT-1) - (SUM/(COUNT)(COUNT-1))}$,

где SOS – сумма квадратов значений аргументов;

COUNT – число входов в таблицу;

SUM – квадрат суммы значений аргументов таблицы.

retry – количество сообщений, ожидающих выполнения специальных условий, зависящих от состояния объекта типа "таблица";

range – определяет верхний и нижний пределы частотных классов. При попадании табулируемого аргумента в интервал, который имеет значение большее нижней границы частотного класса или меньшее

или равное верхней границе, изменяется значение частоты (FREQUENCY);

frequency – суммарная величина, которая формируется при попадании табулируемого аргумента в указанные границы;

sum.% – определяет величину частоты в процентах к общему количеству значений табулируемого аргумента.

Б.1.2. Базовые блоки языка GPSS

Б.1.2.1. Блоки описывающие устройства.

Организация режима прерывания устройств.

Блок PREEMPT

Блок PREEMPT переводит устройство в прерванное состояние.

PREEMPT <A>,[]

Операнд А – задает номер или имя устройства.

Операнд В – задает приоритетный режим (PR) или режим прерывания.

При работе в этом режиме сообщение, занимающее уже устройство, может быть прервано только сообщением, приоритет которого выше приоритета данного сообщения.

Блок RETURN.

Блок RETURN предназначен для освобождения ранее захваченного устройства.

RETURN <A>

Операнд А – задает номер или имя устройства, с которого снимается прерывание. Прерывание может быть снято в блоке RETURN только тем сообщением, которым оно было генерировано.

Б.1.2.2. Многоканальные устройства.

Оператор описания многоканального устройства (оператор контроля управления)

<NAME> STORAGE <A>

NAME – имя или номер многоканального устройства, указывается в поле метки.

Оператор STORAGE определяет емкость многоканального устройства в текущей модели.

Операнд А – определяет объем многоканального устройства (положительное, целое).

MEMRY STORAGE 7 – задается многоканальное устройство с именем MEMRY, емкостью 7 единиц.

Блок используется в разделе инициализации программы.

Блок ENTER

Блок ENTER позволяет вошедшему сообщению занимать, указанное число единиц объема многоканального устройства (памяти).

ENTER <A>,[]

Операнд А – задает номер или имя памяти.

Операнд В – содержится число занимаемых единиц многоканального устройства. Если поле В пусто, то предполагается что занимает одна единица.

ENTER MEMORY,2 – вошедшее сообщение занимает 2 единицы памяти MEMORY.

Блок LEAVE

Блок LEAVE освобождает определенное число единиц многоканального устройства.

LEAVE <A>[,]

Операнд А – задает номер или имя памяти.

Операнд В – число освобождаемых единиц памяти. Если это поле пусто, предполагается 1.

Занятый объем памяти уменьшается на число освобождаемых единиц. Оставшаяся емкость памяти увеличивается на ту же величину. Недоступность многоканального устройства не влияет на действия в блоке LEAVE.

!Число освобождаемых единиц не должно превышать текущее содержимое многоканального устройства.

!Нельзя освободить более единиц памяти, чем захватил.

LEAVE MEMORY,2 – удаляет сообщение из памяти MEMORY, освобождая 2 единицы памяти.

Б.1.2.3. Логические ключи

Инициализация логического ключа (оператор контроля управления)

Инициализация логического ключа производится оператором описания объектов GPSS – INITIAL

INITIAL <A>

Операнд А – задает номер – LS<положительное целое>

INITIAL LS\$1 – ключ 1 установлен в состояние включено.

INITIAL LS\$1,LS\$7 – ключи 1, 7 установлены в состояние включено

Одним оператором INITIAL можно инициализировать до 6 ключей.

Оператор используется в разделе инициализации программы.

Блок LOGIC

Блок LOGIC устанавливает положение ключа, которое может быть проверено любым сообщением в любой части модели.

LOGIC <X> <A>

Операнд А – задает имя или номер логического переключателя, значение которого должно быть изменено.

Внутренний операнд X – устанавливает состояние логического ключа:

- S – логический ключ "включен";
- R – логический ключ "выключен";
- I – логический ключ инвертируется.

LOGIC I 1 – инвертирует значение ключа 1.

Б.1.2.4. Блоки, изменяющие порядок прохождения блоков сообщениями

Блок TRANSFER

Блок TRANSFER является основным средством, позволяющим направить сообщение к любому блоку модели.

TRANSFER [<A>],[],[<C>],[<D>]

Операнд А задает режим выбора следующего блока, к которому должно перейти сообщение:

– безусловный (пробел)– сообщение передается в блок, определяемый операндом В;

TRANSFER ,MET1 – безусловная передача сообщения в блок с меткой MET1

– статистический (.) – указывается десятичное число, указывающее вероятность перехода в блок С, дополнение до единицы десятичного числа – вероятность перехода в блок В;

TRANSFER .0370,MET1,MET2 – с вероятностью 0.630 в блок с меткой MET1, с вероятностью 0,370 в блок с меткой MET2

– BOTH – сообщение последовательно пытается войти в блок В, затем в блок С, до тех пор, пока один из них не станет доступным;

TRANSFER BOTH,MET1,MET2 – сообщение последовательно пытается войти в блок с меткой MET1, затем в блок с меткой MET2.

– ALL – сообщение последовательно пытается перейти в блоки, определяемые значениями В+ D, В+2D,...,С;

– параметр (Р) – значение аргумента поля В интерпретируется как номер j параметра входящего сообщения; номер следующего блока для данного сообщения определяется суммой значения этого параметра со значением аргумента поля С.

Операнд В – следующий блок (номер блока или метка). Если поле В пусто, то переход к номеру блока, следующего за блоком TRANSFER.

Операнд С – следующий блок (номер блока или метка).

Операнд D – значение индекса, используемое в режиме ALL.

Блок GATE

Блок GATE проверяет состояние устройств, памяти, логических ключей. Данный блок может задержать сообщение на входе, если не задан альтернативный выход.

GATE <X> <A>,[]

Операнд А – содержит имя или номер объекта, для которого проводится проверка.

Операнд В – содержит метку или номер следующего блока для входящего сообщения, когда логический оператор имеет значение "ложь".

Внутренний операнд X – определяет проверяемое условие в виде стандартного логического атрибута:

1) логические атрибуты, связанные с устройством:

- NU – свободно; U – занято
- NI – не прервано; I – обслуживает прерывание;
- FV – доступно; FNV – не доступно;

GATE NU CPU, 20 – если устройство с именем CPU не свободно, то переход к блоку номер 20

2) логические операторы, связанные с многоканальными устройствами:

- SE – пусто; SNE – не пусто;
- SF – заполнено; SNF – не заполнено;
- SV – доступно; SNV – не доступно;

3) логические операторы, связанные с логическими ключами:

- LS – включен; LR – выключен;

4) логические операторы, связанные с сообщениями:

- M – сообщение находится в состоянии синхронизации с сообщением, находящемся в блоке, заданном в операнде A.
- NM – в блоке, заданном в операнде A, нет ни одного сообщения, принадлежащего к тому же семейству, что и проверяемое сообщение.

GATE M 11, NEXT – проверка устройства под номером 11; если в устройстве содержится сообщение, принадлежащее к тому же семейству, что и проверяемое сообщение, то переход к блоку номер 11, нет переход на метку NEXT.

GATE M 11 – проверка устройства под номером 11; если в устройстве содержится сообщение, принадлежащее к тому же семейству, что и проверяемое сообщение, то переход к блоку номер 11, нет ожидание синхронизации.

Блок LOOP

Блок LOOP используется для управления числом повторений определенной последовательности блоков в модели.

LOOP <A>,

Операнд A – задает параметр текущего сообщения, значение которого определяет число повторений (количество циклов);

Операнд B – номер блока или метка.

Значение параметра операнда A уменьшается каждый раз на 1, при входе сообщения в блок LOOP. Если значение параметра не равно нулю, сообщение переходит в блок операнда B. В противном случае (достижение параметром сообщения нуля) сообщение переходит в следующий блок.

Установка значений параметров сообщения осуществляется в блоке ASSIGN.

Блок TEST

В блоке TEST выполняется проверка числовых выражений.

TEST X <A>, [, <C>]

Внутренний операнд X – определяет проверяемое условие в виде стандартного логического атрибута:

G	–A больше B;	NE	–A не равно B;
GE	–A больше или равно B;	LE	–A меньше или равно B;
E	–A равно B;	L	–A меньше B;

Операнд A – стандартный числовой атрибут.

Операнд B – стандартный числовой атрибут.

Операнд C – содержит метку или номер следующего блока для входящего сообщения, когда логический оператор имеет значение "ложь".

Операнды A и B – имена СЧА, которые сравниваются. Внутренний оператор X указывает способ сравнения этих двух СЧА друг с другом.

Б.1.2.5 Блоки, ориентированные на сообщения – Блок ASSIGN

Блок ASSIGN изменяет текущее значение параметра сообщения на заданное значение.

ASSIGN <A>,[,<C>]

Операнд A – задает номер параметра, которому присваивается значение.

Операнд B – новое значение параметра;

Операнд C – модификатор-функция (действие аналогично GENERATE).

Если значение параметра, ранее определенное, нужно увеличить или уменьшить, то справа за значением операнда A ставится знак сложения или вычитания.

ASSIGN 2,8 – присвоение второму параметру сообщения значения 8;

ASSIGN 2+,5 – добавляет значение 5 ко значению второго параметра сообщения (8+5=13 – текущее значение второго параметра сообщения);

Б.1.2.6. Блоки для обработки сообщений, принадлежащих одному семейству

Блок SPLIT

Блок SPLIT выполняет функцию копирования входящего в него сообщения, которое называется исходным или порождающим.

SPLIT <A>,[,<C>]

Операнд A – задает число создаваемых копий.

Операнд B – задает номер блока или метку, на которую переходят копии исходного сообщения.

Операнд C – номер параметра сообщения, используемого для хранения порядкового номера копии;

SPLIT 1,METK,2 – создается 1 копия, вошедшего в блок сообщения, копия будет направлена на метку METK, порядковый номер копии будет храниться во 2-м параметре сообщения.

Сообщения, принадлежащие к одному семейству, объединяются интерпретатором в список. Когда сообщение уничтожается, интерпретатор автоматически исключает его из членов соответствующего семейства. Таким

образом, семейство существует до тех пор, пока из модели не удаляется последний из его членов.

В модели одновременно может существовать произвольное число семейств.

Блок ASSEMBLE

Блок ASSEMBLE объединяет заданное число сообщений, принадлежащих к одному семейству.

ASSEMBLE <A>

Операнд A – задает число сообщений, участвующих в сборке.

Первое сообщение семейства достигнув блока ASSEMBLE, задерживается в нем до тех пор, пока остальные члены семейства (размноженные ранее копии) не поступят в этот блок; семейство собрано – копии удаляются, движение продолжает одно сообщение. В одном и том же блоке ASSEMBLE возможна одновременная сборка сообщений нескольких семейств.

Б.1.3 Операторы контроля управления

Б.1.3.1 Оператор SIMULATE

Оператор SIMULATE используется для установки реального времени моделирования.

SIMULATE n

где n – целое число, равное абсолютному времени моделирования в минутах.

Б.1.3.2 Оператор EQU

Оператор предназначен для присвоения числовых значений именам, используемым в модели.

<NAME> EQU <A>

Метка NAME определяет имя объекта.

Операнд A – содержит число в диапазоне от -32766 до +32766.

Оператор используется в разделе инициализации программы.

Б.1.3.3. Таблица – инструмент сбора статистической информации.

Оператор описания таблицы.

Оператор определяет аргумент, а также число и ширину частотного интервала.

<NAME> TABLE <A>,,<C>,<D>

Метка NAME определяет имя таблицы.

Операнд A – задает аргумент таблицы – элемент данных, чье частотное распределение будет табулироваться.

Операнд B – задает верхний предел первого интервала.

Операнд C – ширина частотного интервала – разница между верхней и нижней границей каждого частотного класса.

Операнд D – число частотных интервалов

TIME TABLE M1,1000,500,20 Определена таблица с именем TIME, в таблице фиксируется величина M1 – время пребывания в модели сообщения, обрабатываемого программой в данный момент. Первый интервал для табулируемого значения будет равен от 0 до 1000, шаг – 500, таких интервалов – 20. То есть будет сохраняться суммарная величина времени пребывания заявки в системе, которая формируется при попадании табулируемого аргумента в указанные границы по интервалам от 0 до 11000.

Блок TABULATE.

Блок TABULATE заносит значение заданного аргумента в таблицу. Способ табуляции зависит от режима работы таблицы, который определяется оператором таблицы TABLE.

TABULATE <A>[,]

Операнд A – задает номер или имя таблицы, в которую заносится значение аргумента;

Операнд B – задает сколько раз значение должно быть занесено в таблицу. Если операнд B пуст, эта величина полагается равной единице.

TABULATE TIME – заносит текущее значение аргумента M1 в таблицу с именем TIME.

ПРИЛОЖЕНИЕ В

(обязательное)

Текст базовой программы модели мультипрограммной вычислительной системы

```

10 * ----- РАЗДЕЛ ИНИЦИАЛИЗАЦИИ -----
20 * ----- ОПРЕДЕЛЕНИЕ ОБЪЕКТОВ МОДЕЛИ -----
30 CPU      EQU      5
40 MEMRY STORAGE  10
50 CHAN     STORAGE  1
60 JTIME    TABLE   M1,1000,500,20
70 DISK     STORAGE  4
80 EXPN     FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.75,1.38/.8,1.6/.84,1.83/
.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9/.99,4.6/.995,5.3/
.998,6.2/.999,7.0/.9997,8.0/
90 UNIT     FUNCTION RN8,D4
.25,1/.5,2/.75,3/1,4
100 * -----ОПИСАНИЕ МОДЕЛИРУЕМОГО ПРОЦЕССА -----
110 * -----1.ЗАПУСК ЗАДАНИЯ НА ВЫПОЛНЕНИЕ -----
120         GENERATE 10000,FN$EXPN
130         QUEUE   JOBQ
140         ENTER   MEMRY
150         DEPART  JOBQ

160 * -----2.ЦИКЛ ВЫПОЛНЕНИЯ ЗАДАНИЯ -----
170         ASSIGN  1,20

180 * -----2.1.ОБРАБОТКА В ПРОЦЕССОРЕ -----
190 CYCLE SEIZE   CPU
200         ADVANCE 3,1
210         RELEASE CPU

220 * -----2.2.ВЫПОЛНЕНИЕ ВВОДА-ВЫВОДА -----
230         ASSIGN  2,FN$UNIT

240 * ----- ИСКЛЮЧЕНИЕ КОНФЛИКТА РЕСУРСОВ -----
242         TRANSFER ,MTK2
244 MTK1 LEAVE CHAN
246 MTK2 GATE NU P2

250 * -----2.2.1. УСТАНОВКА ГОЛОВОК НА ДИСКЕ -----
260         ENTER   CHAN
262         GATE NU P2,MTK1
270         SEIZE   P2

```

280 LEAVE CHAN
 290 ENTER DISK
 300 ADVANCE 45,45

310 * -----2.2.2. ПЕРЕДАЧА ДАННЫХ ЧЕРЕЗ КАНАЛ -----

320 ENTER CHAN
 330 ADVANCE 25
 340 LEAVE CHAN
 350 RELEASE P2
 360 LEAVE DISK
 370 LOOP 1,CYCLE

380 * -----3. ЗАВЕРШЕНИЕ ЗАДАНИЯ -----

390 LEAVE MEMRY
 400 TABULATE JTIME
 410 TERMINATE 1

420 * ----- ВЫПОЛНЕНИЕ ПРОГОНА МОДЕЛИ -----

430 START 100

Заказ № _____ от « _____ » _____ 200____. Тираж _____ экз.
 Изд-во СевНТУ