

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Введение

Моделирование является одним из наиболее распространенных способов изучения различных процессов и явлений и широко используется в научных исследованиях и инженерной практике. Различают физическое и математическое моделирование. При **физическом** моделировании модель воспроизводит изучаемый процесс с сохранением его физической природы. Под **математическим** моделированием понимают способ исследования различных процессов путем изучения явлений, имеющих различное физическое содержание, но описываемых одинаковыми математическими соотношениями. Например, детерминистические объекты могут быть описаны конечными автоматами, дифференциальными уравнениями, а стохастические объекты, учитывающие случайные факторы - вероятностными автоматами, системами массового обслуживания и марковскими процессами.

Построение математической модели сложной системы в целом часто оказывается практически невозможным из-за сложности процессов ее функционирования. В этих случаях систему декомпозируют на отдельные подсистемы вплоть до элементов, сохраняя связи между подсистемами. Тогда сложную систему можно определить как многоуровневую конструкцию из взаимодействующих элементов, объединяемых в подсистемы различных уровней. В качестве такой системы можно рассматривать автоматизированные системы управления различного назначения, построенные по иерархическому принципу.

Любую сложную систему будем рассматривать как совокупность элементов и подсистем, предназначенную для решения определенного класса задач или же подчиненную единой цели. Если цели и задачи системы определены, то ставится вопрос об оценке качества ее функционирования с помощью **показателей эффективности**. В зависимости от назначения системы показатели эффективности могут быть различными, но чаще всего в качестве основного показателя эффективности выступает **производительность** системы, которая в свою очередь включает различные классы индексов. В таблице 1 приведены основные классы количественных индексов производительности вычислительных систем.

Расчет показателей эффективности сложных систем, т.е. задача анализа производительности, представляет собой весьма сложную задачу, которая требует привлечения специальных математических методов и, как правило, решается с помощью ЭВМ. Показатели эффективности зависят от структуры системы, значений ее параметров, характера воздействия внешней среды, внешних и внутренних случайных факторов, поэтому их можно считать функционалами, заданными на множестве процессов функционирования системы. Такие функционалы широко используются в теории сложных систем и системном анализе.

Таблица 1 - Основные классы количественных индексов производительности вычислительных систем

Класс индекса	Примеры индексов	Общее определение
Продуктивность	Пропускная способность Скорость выработки Максимальная выработка (максимум пропускной способности) Скорость выполнения команд Скорость обработки данных	Объем информации, обрабатываемой системой в единицу времени
Реактивность	Время ответа Время прохождения Время реакции	Время между предъявлением системе входных данных и появлением соответствующей выходной информации
Использование	Коэффициенты использования оборудования (центральный процессор, канал ввода-вывода, устройство ввода-вывода) Коэффициент использования операционной системы Коэффициент использования общего модуля программного обеспечения (например,	Отношение времени использования указанной части системы (или ее использования для заданной цели) в течение заданного интервала времени к длительности этого интервала

	компилятора) Коэффициент использования базы данных	
--	--	--

В связи с тем, что сложные системы функционируют в условиях действия случайных факторов, значения функционалов являются случайными величинами и поэтому в задачах анализа производительности пользуются средними значениями функционалов. Например, **среднее количество** изделий, выпускаемых за смену, **средняя прибыль** (для производственных процессов), **средняя стоимость** перевозки (для транспорта), **среднее время ожидания** в очереди (для систем массового обслуживания) и другие.

Таким же путем можно характеризовать и другие свойства сложных систем как надежность, помехозащищенность, качество управления и другие.

Для того, чтобы получить ответы на вопросы о производительности данной системы, разработчик системы на ранних этапах проектирования (системного проектирования) должен получить информацию об индексах производительности при определенных значениях параметров системы. Эту необходимую для исследования информацию можно получить посредством методов оценки производительности как от самой системы (методы измерения), если она существует, так и от модели системы (методы моделирования).

В настоящее время существует целый арсенал **измерительных** средств, как аппаратных, так и программных и микропрограммных. Под **моделью** системы будем понимать такое ее представление, которое состоит из определенного объема организованной информации о ней и построено с целью ее изучения. Для одной и той же системы может быть построен ряд различных моделей в зависимости от точек зрения и степени детализации системы (расчленения на компоненты).

Место и роль концептуальных (мыслимых) моделей при проектировании сложных систем определим следующим образом. Во-первых, концептуальные (математические) модели играют **фундаментальную** роль в оценке производительности и надежности сложных систем. Во-вторых, математическое моделирование является современным средством оценки **качества** проектных решений по сложным системам, в том числе и уже существующих систем в процессе их эксплуатации.

Концептуальные модели являются основой методов измерения, а также двух классов методов моделирования: **имитационного** и **аналитического**.

Очень распространенное и удобное описание поведения системы основывается на концепциях **состояния** и **перехода между состояниями**. Состояние системы в момент времени определяется как множество значений интересующих нас параметров системы в момент времени. Любое изменение этих значений параметров означает переход системы в другое состояние. Если поведение модели во времени в основном воспроизводит поведение системы и прослеживается эволюция решений уравнений модели на заданном интервале времени с сохранением хронологической последовательности изменения переменных состояния модели и системы, то мы имеем имитационную модель.

В аналитическом моделировании уравнения модели решаются чаще всего путем эквивалентных формульных преобразований, которые не отражают хронологию функционирования самой системы. Однако и здесь существуют численные методы (типа решения задачи Коши для дифференциальных уравнений), которые представляют собой последовательную процедуру, в чемто копирующую эволюцию реальной системы.

Существенным условием применимости любой модели является ее **адекватность** реальной системе и при оценке производительности системы **точность** модели должна быть определена к индексам производительности, выбранным для этой цели. Значения этих индексов, полученные в эксперименте на модели, должны быть достаточно близки к значениям моделируемой системы при тех же входных воздействиях.

На рисунке 1 показана иллюстрация этого определения для простого случая системы обработки данных из N заданий, где в качестве индекса производительности взято общее время $t_{общ}$ обработки N заданий. Модель

считается точной, если $|t_{общ} - t'_{общ}| < \epsilon$, где ϵ - заданная максимальная ошибка, а $t'_{общ}$ - результат моделирования.

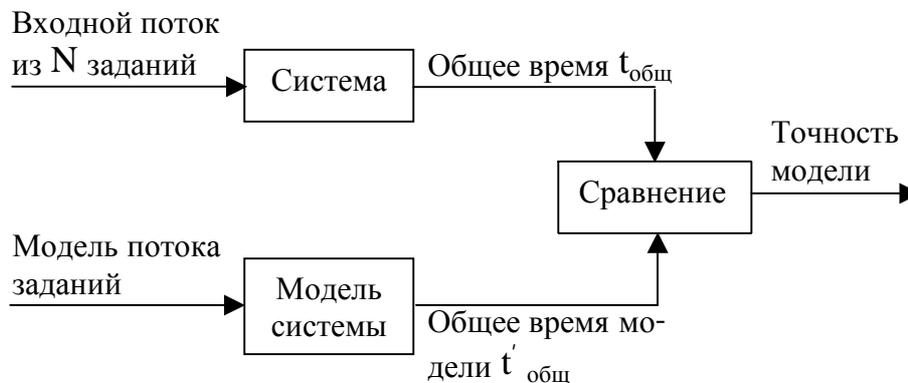


Рисунок 1- Иллюстрация понятия точности модели

При проектировании, когда моделируемая система не существует физически или не доступна для эксперимента, моделируемую систему представляют в виде концептуальной модели в действительности. Тогда точность модели можно оценить по схеме, представленной на рисунке 2.

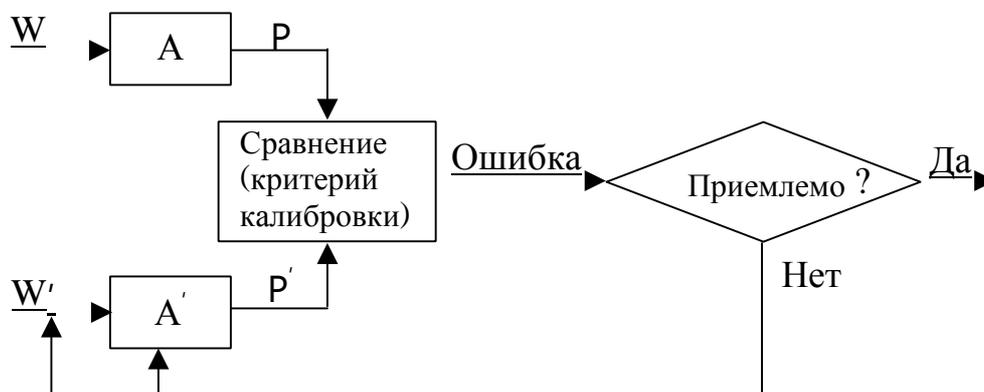


Рисунок 2 - Схема итеративной процедуры калибровки

Система A при рабочей нагрузке W имеет производительность P , где P совокупность индексов производительности (скаляры, средние и дисперсии функционалов). Модель системы A' при нагрузке W' имеет производительность

P' . Сравнение значений одноименных индексов производительности дает меру точности P' и W' . Если точность модели не удовлетворительна, то в модель необходимо внести изменения, а процесс проверки повторить. Эта операция называется **калибровкой** модели. Критерии калибровки и меры точности для вычислительных систем приведены в таблице 2.

Таблица 2-Критерии калибровки модели (индекс производительности: время прохождения задания)

Основание критерия	Ошибка E
Среднее время прохождения задания	$\frac{1}{N} \sum_{j=1}^N (t_j - t_j')$
Время прохождения отдельного задания	$\sum_{j=1}^N \sum_{i=1}^{n_j} (t_{ji} - t_{ji}')$
Продолжительность отдельного шага задания	
Символ	Определение
t_j	Время прохождения задания j в системе t_j'
	Время прохождения задания j в модели t_{ji}
	Время выполнения шага i задания j в системе
t_{ji}	Время выполнения шага i задания j в модели
N	Общее число заданий n_j
	Число шагов в задании j

Существующие методы и модели анализа производительности вычислительных систем представлены следующей структурной схемой (рисунок 3).

Здесь **алгебраические и аппроксимационные** методы образуют класс методов и моделей аналитического **вероятностного моделирования**. Алгебраические методы в теории массового обслуживания ограничены предположением о **пуассоновских** входных потоках и **экспоненциальности** времени обслуживания (когда известны точные результаты для вероятности

состояния сетевой модели в виде произведения), что далеко не всегда имеет место при исследовании реальных процессов.

Следующий момент связан с неоднородностью реальных потоков. **Неоднородность** потоков случайных событий прежде всего обусловлена зависимостью времени обслуживания от параметров входного потока, а также разнородностью используемых в сложных системах управления средств вычислительной техники и разнотипностью классов решаемых задач. Неоднородность потоков в экспоненциальных и сетевых моделях также не может быть учтена. Игнорирование же этих двух важных факторов при использовании экспоненциальных сетей для решения задачи анализа производительности вычислительных систем может внести **существенную погрешность** в результаты моделирования.

Это и послужило основанием для появления аппроксимационных методов. Среди них в пособии выделены методы диффузионной аппроксимации процессов функционирования систем массового обслуживания.

Под методом **имитационного моделирования** будем понимать **способ вычисления статистических** характеристик интересующих нас случайных величин посредством воспроизведения реализаций соответствующего случайного процесса с помощью его математической модели. В области управления экономикой, планировании, исследовании операций, проектировании термин «имитационный эксперимент» означает **способ выбора рационального управления** сложным процессом (рационального плана, рациональной конструкции проектируемого изделия) путем сравнения различных вариантов.

В дальнейшем будем различать аналитическое вероятностное моделирование от имитационного моделирования в том смысле, как различаются **теория вероятностей и математическая статистика**.

В настоящее время существует несколько десятков специализированных имитационных систем моделирования или же проблемно – ориентированных систем, и их число растет.

В данном пособии для решения простейших задач по моделированию вычислительных систем использован язык системного моделирования GPSS.

Материал пособия изложен в той логической последовательности, которая приведена на рисунке 3. При этом в классе приближенных методов и

моделей (аппроксимационный подход) изложены научные и практические результаты Тарасова В.Н. в данной области.

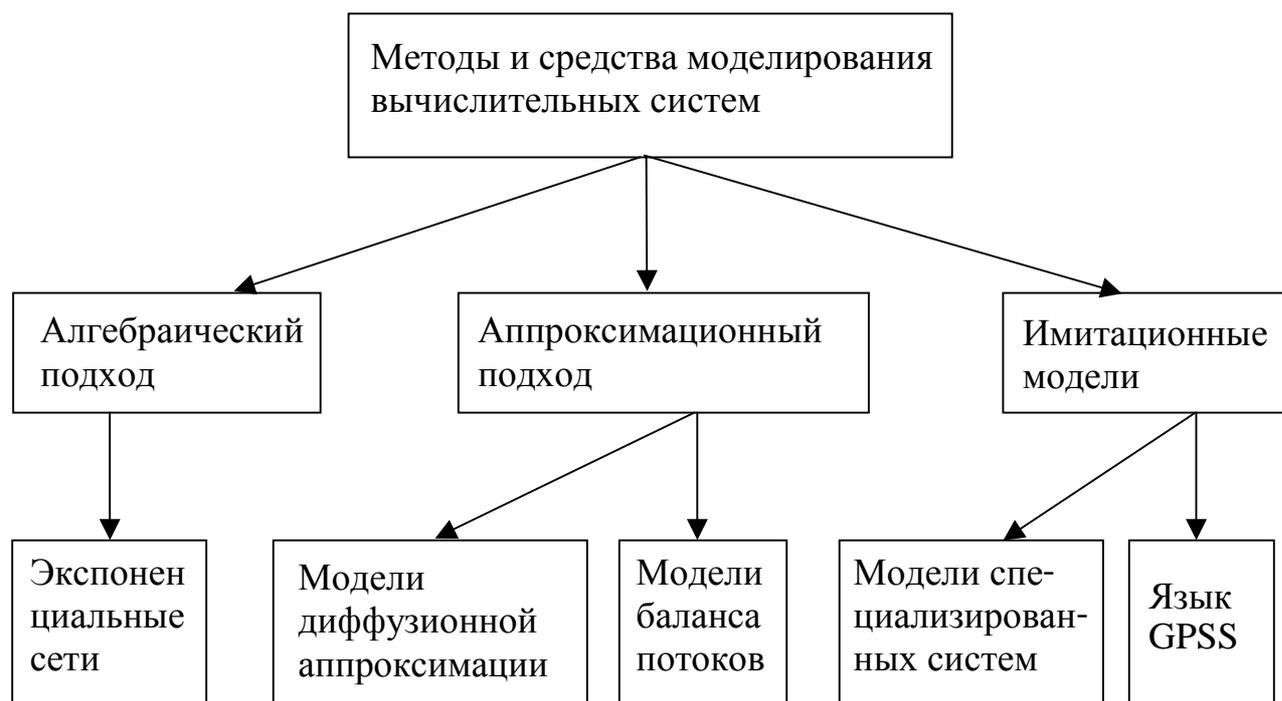


Рисунок 3 - Существующие методы и модели анализа производительности вычислительных систем

1 Задание на лабораторную работу №1

- 1 Изучить теоретический материал к лабораторной работе.
- 2 Сгенерировать временной ряд с заданным законом распределения с объёмом выборки, равным $N=500$ (количество реализации для каждого модельного эксперимента равно 29).
- 3 Проверить качество генерирования, воспользовавшись для определения параметров аналитического выражения законов распределения методом моментов.
- 4 Определить погрешности оценки параметров модели.
- 5 Пункты 2-4 повторить для объёмов выборки $N=1000, 2000, 5000$.

1.1 Содержание отчёта

- 1 Цель работы.

2 Метод и алгоритм моделирования некоррелированных временных рядов для заданного закона распределения.

3 Обратная функция закона распределения вероятностей.

4 Пример реализации некоррелированного временного ряда.

5 Примеры гистограмм для различного объёма выборки - $N=500, 1000, 2000, 5000$, $M=20$.

6 Значения параметров, определенные по методу моментов, и модуль относительной погрешности оценки параметров закона распределения для $N=500, 1000, 2000, 5000$, представленные в табличной форме (количество реализации для каждого модельного эксперимента равно 29). Для определения параметра закона распределения и вычисления погрешности оценки параметра можно воспользоваться пакетом Excel.

7 Графическая зависимость максимальной по модулю относительной погрешности оценки параметров закона распределения от объёма выборки - $N=500, 1000, 2000, 5000$. Для построения графических зависимостей можно воспользоваться пакетом Excel.

8 Выводы по работе.

Пример оформления результатов выполненной лабораторной работы для экспоненциального закона распределения приведен ниже (пункты 4-7 отчёта).

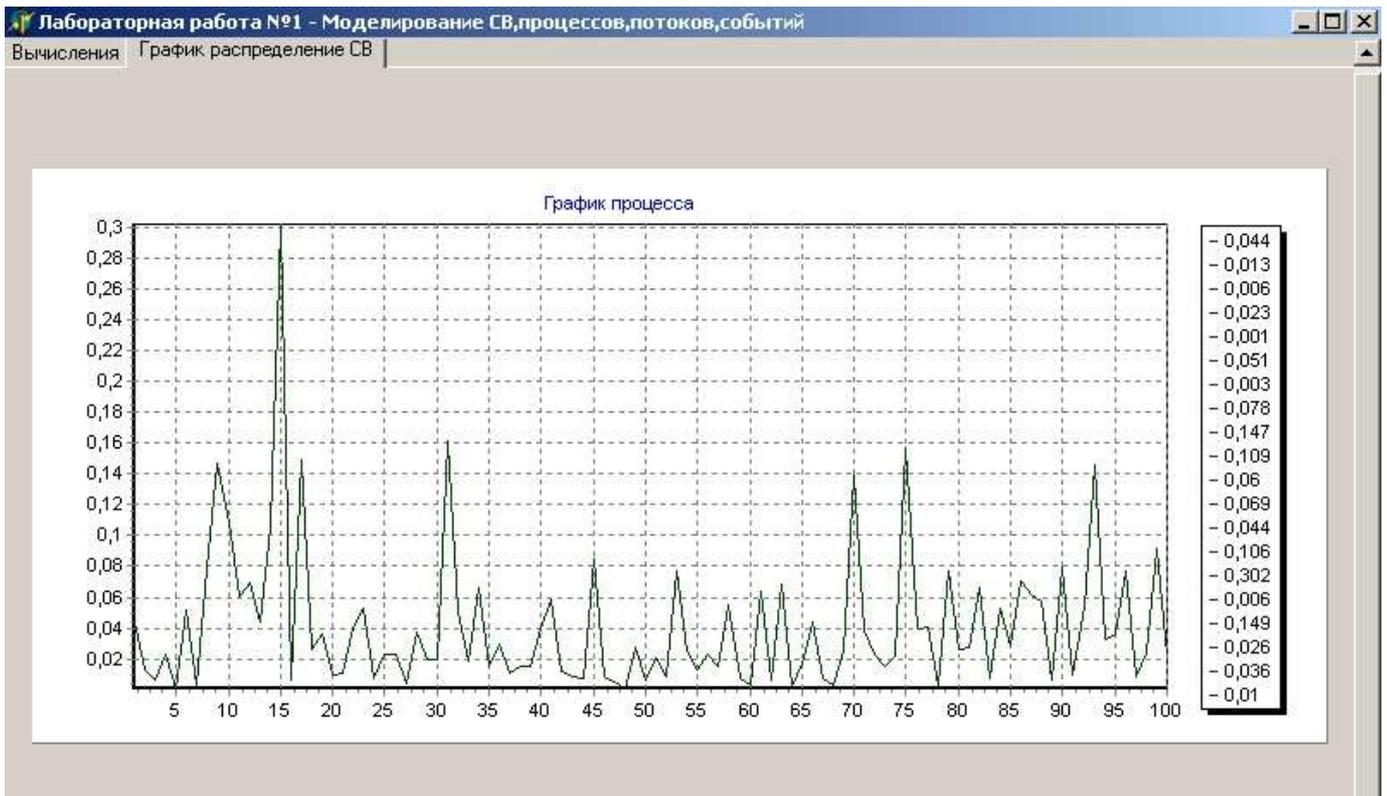


Рисунок 1.1 - Генерирование ПСП с экспоненциальным законом распределения методом инверсного преобразования

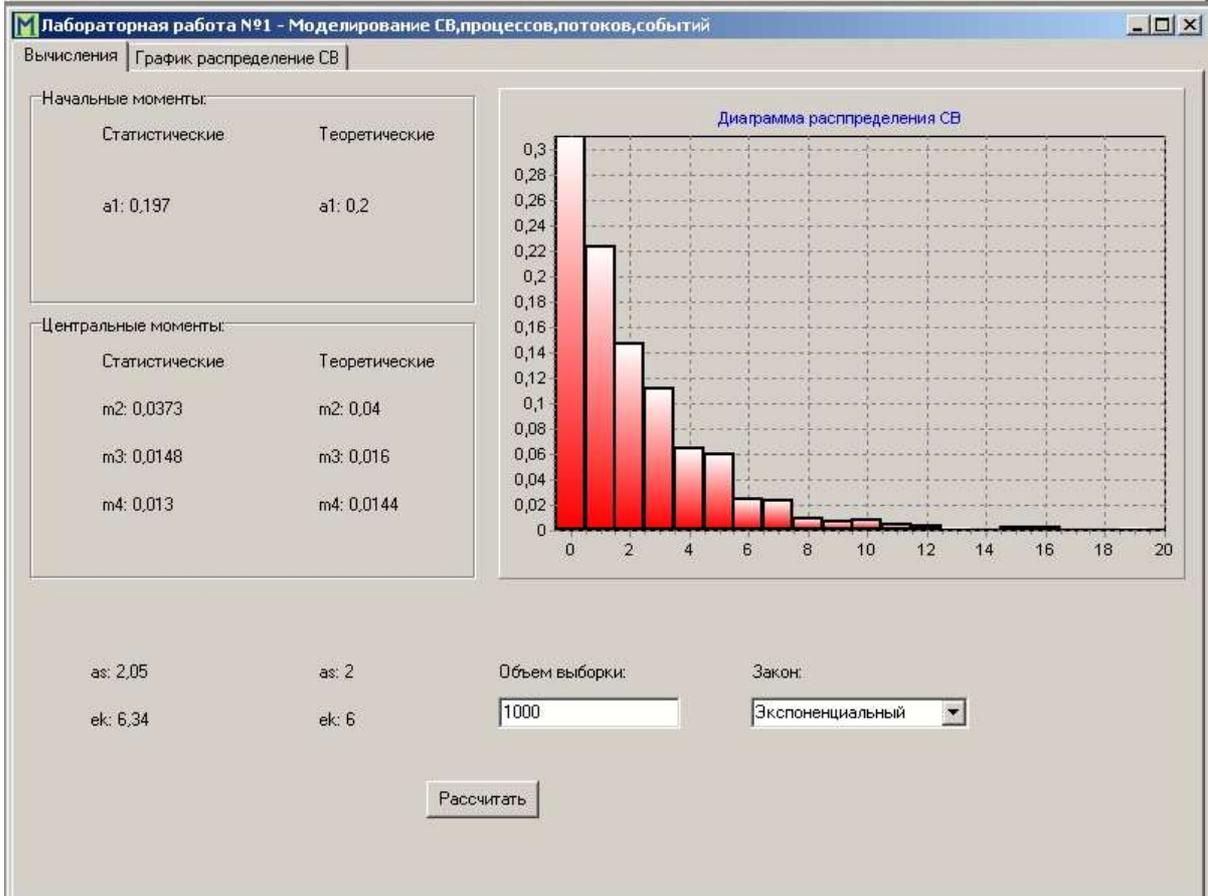
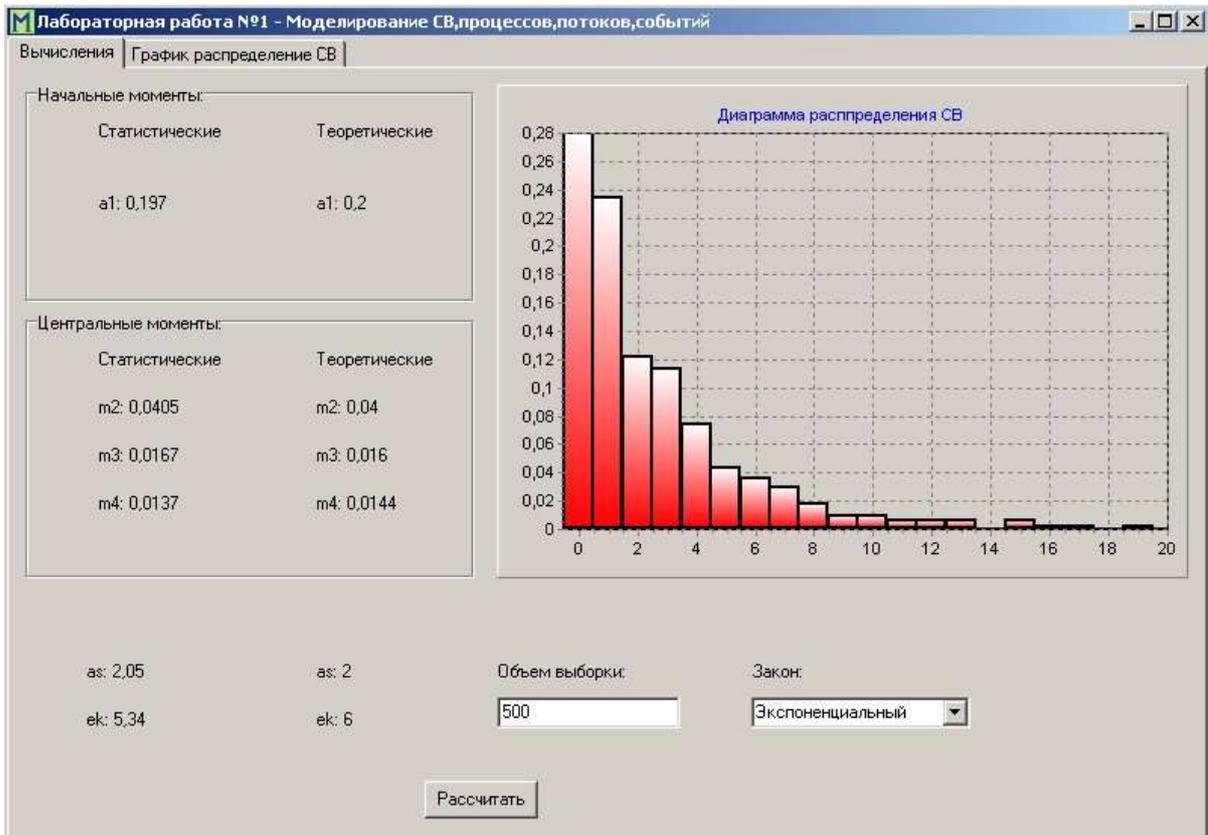


Рисунок 1.2 - Пример генерирования ПСП с экспоненциальным законом распределения

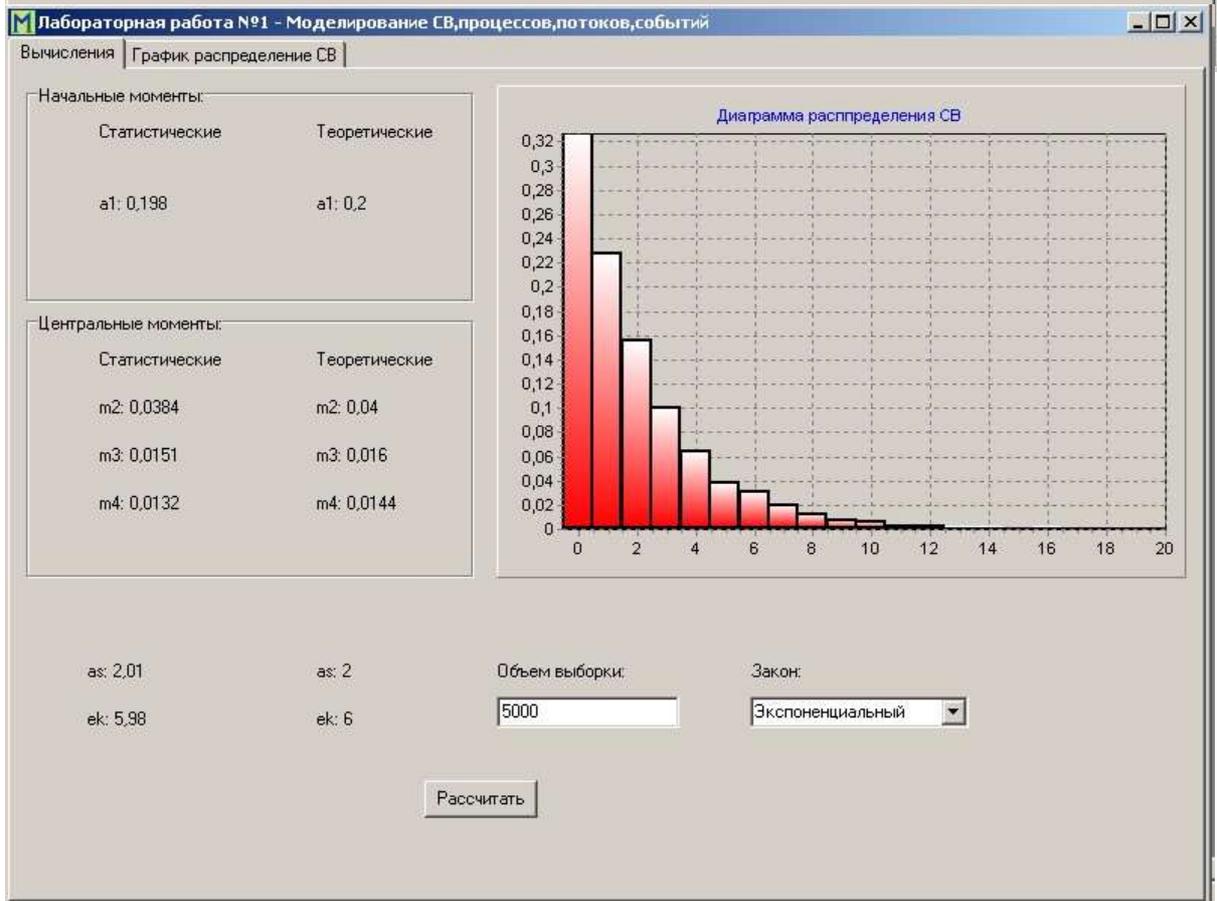
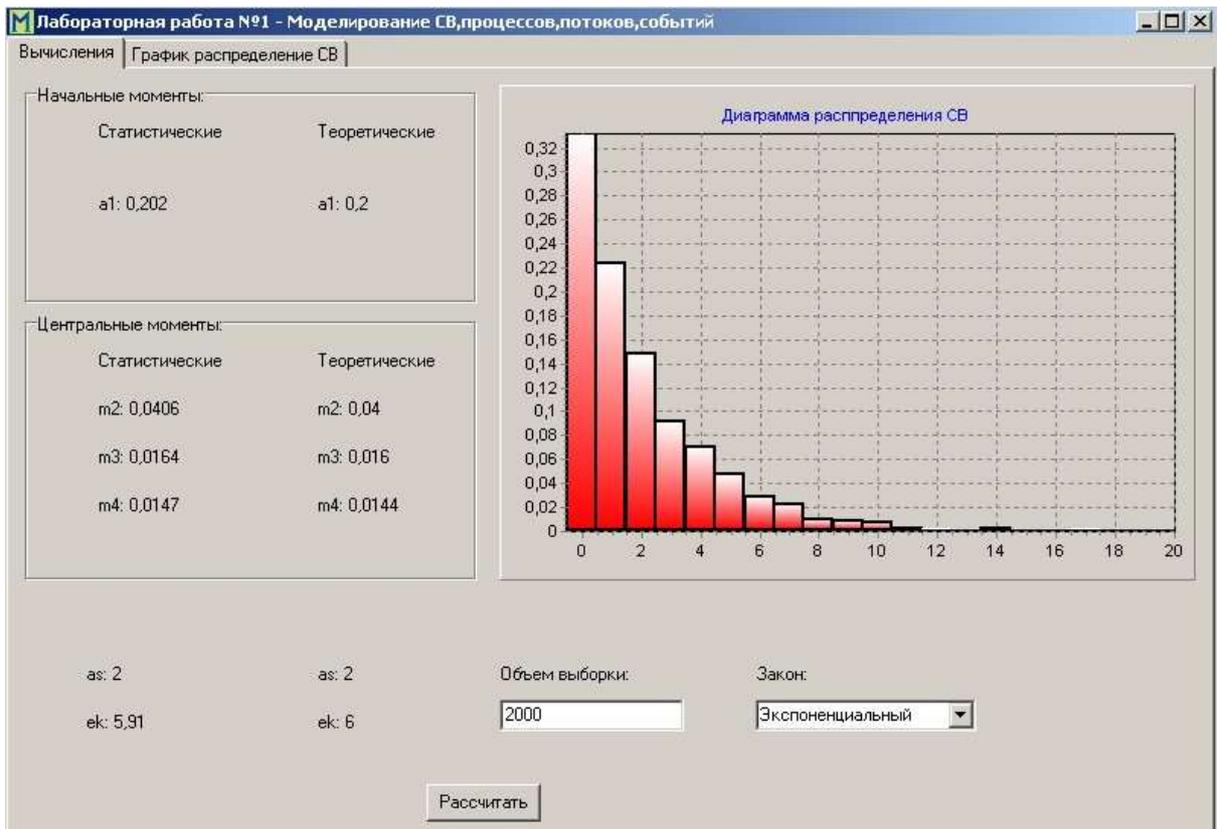


Рисунок 1.3 - Пример генерирования ПСП с экспоненциальным законом распределения

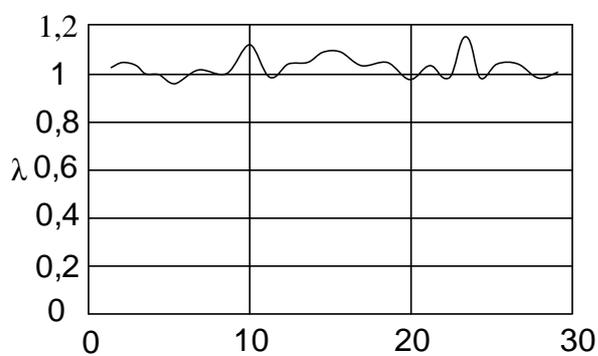
Таблица 1.1 - Значения параметров, определенные по методу моментов, и относительные погрешности оценки параметров закона распределения

N=500				N=1000			
	$\hat{\alpha}_{-1}$	$\hat{\lambda} = 1/\hat{\alpha}_{-1}$	$\delta = \frac{(\hat{\lambda} - \lambda)}{\lambda}$		$\hat{\alpha}_{-1}$	$\hat{\lambda} = 1/\hat{\alpha}_{-1}$	$\delta = \frac{(\hat{\lambda} - \lambda)}{\lambda}$
1	0,97723	1,023301	0,023301	1	0,94009	1,063728	0,063728
2	0,96093	1,040659	0,040659	2	0,99149	1,008583	0,008583
3	0,94707	1,055888	0,055888	3	1,01004	0,99006	-0,00994
4	1,0122	0,987947	-0,01205	4	0,98566	1,014549	0,014549
5	1,01325	0,986923	-0,01308	5	1,02001	0,980383	-0,01962
6	1,06513	0,938853	-0,06115	6	0,92825	1,077296	0,077296
7	1,04756	0,954599	-0,0454	7	1,02934	0,971496	-0,0285
8	0,98242	1,017895	0,017895	8	1,0109	0,989218	-0,01078
9	1,01613	0,984126	-0,01587	9	0,99031	1,009785	0,009785
10	0,90731	1,102159	0,102159	10	1,0059	0,994135	-0,00587
11	1,05346	0,949253	-0,05075	11	0,94411	1,059199	0,059199
12	0,97002	1,030907	0,030907	12	0,99562	1,004399	0,004399
13	0,92659	1,079226	0,079226	13	0,9987	1,001302	0,001302
14	0,94311	1,060322	0,060322	14	0,9672	1,033912	0,033912
15	0,91677	1,090786	0,090786	15	1,1127	0,898715	-0,10129
16	0,91441	1,093601	0,093601	16	0,98334	1,016942	0,016942
17	0,97835	1,022129	0,022129	17	1,01539	0,984843	-0,01516
18	0,98175	1,018589	0,018589	18	1,04018	0,961372	-0,03863
19	0,97255	1,028225	0,028225	19	0,98262	1,017687	0,017687
20	1,05078	0,951674	-0,04833	20	1,0151	0,985125	-0,01488
21	1,0076	0,992457	-0,00754	21	0,99286	1,007191	0,007191
22	0,95833	1,043482	0,043482	22	0,925	1,081081	0,081081
23	1,03565	0,965577	-0,03442	23	1,02148	0,978972	-0,02103
24	0,87468	1,143275	0,143275	24	0,97933	1,021106	0,021106
25	1,06397	0,939876	-0,06012	25	0,99113	1,008949	0,008949
26	0,95993	1,041743	0,041743	26	1,00296	0,997049	-0,00295
27	0,96412	1,037215	0,037215	27	1,00701	0,993039	-0,00696
28	1,04051	0,961067	-0,03893	28	1,01401	0,986184	-0,01382
29	0,99359	1,006451	0,006451	29	0,99266	1,007394	0,007394

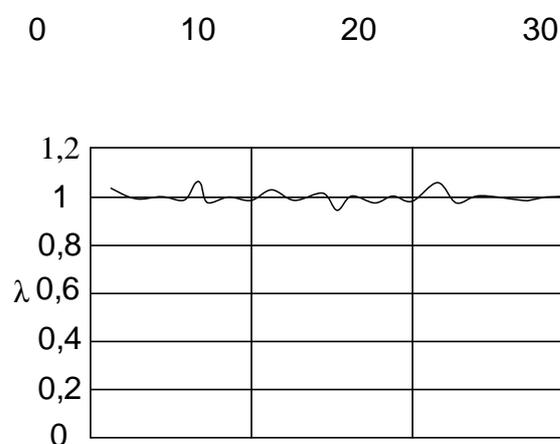
Продолжение таблицы 1.1

N=2000				N=5000			
	α_{-1}	$\lambda_{\wedge} = 1/\alpha_{\wedge 1}$	$\delta = \frac{(\bar{\lambda} - \lambda)}{\lambda}$		α_{-1}	$\lambda_{\wedge} = 1/\alpha_{\wedge 1}$	$\delta = \frac{(\bar{\lambda} - \lambda)}{\lambda}$
1	1,00292	0,997089	-0,00291	1	1,02181	0,978656	-0,02134
2	0,99638	1,003633	0,003633	2	0,98327	1,017015	0,017015
3	1,00708	0,99297	-0,00703	3	0,99148	1,008593	0,008593
4	0,98824	1,0119	0,0119	4	0,98502	1,015208	0,015208
5	1,02102	0,979413	-0,02059	5	1,02117	0,979269	-0,02073
6	0,99564	1,004379	0,004379	6	1,00947	0,990619	-0,00938
7	0,96806	1,032994	0,032994	7	1,00089	0,999111	-0,00089
8	0,98639	1,013798	0,013798	8	1,01286	0,987303	-0,0127
9	1,02966	0,971194	-0,02881	9	0,98589	1,014312	0,014312
10	0,99591	1,004107	0,004107	10	0,98445	1,015796	0,015796
11	0,99639	1,003623	0,003623	11	0,99633	1,003684	0,003684
12	1,02298	0,977536	-0,02246	12	0,99761	1,002396	0,002396
13	0,99853	1,001472	0,001472	13	0,99352	1,006522	0,006522
14	0,99237	1,007689	0,007689	14	1,00774	0,992319	-0,00768
15	0,99152	1,008553	0,008553	15	1,00557	0,994461	-0,00554
16	1,02363	0,976915	-0,02308	16	1,01113	0,988993	-0,01101
17	1,00942	0,990668	-0,00933	17	0,99911	1,000891	0,000891
18	1,00899	0,99109	-0,00891	18	1,00916	0,990923	-0,00908
19	0,98241	1,017905	0,017905	19	0,99684	1,00317	0,00317
20	0,98853	1,011603	0,011603	20	1,01254	0,987615	-0,01238
21	0,9678	1,033271	0,033271	21	1,00414	0,995877	-0,00412
22	1,00999	0,990109	-0,00989	22	0,99648	1,003532	0,003532
23	0,98163	1,018714	0,018714	23	1,01124	0,988885	-0,01112
24	0,95262	1,049737	0,049737	24	1,00915	0,990933	-0,00907
25	0,9806	1,019784	0,019784	25	1,00342	0,996592	-0,00341
26	1,02819	0,972583	-0,02742	26	0,98785	1,012299	0,012299

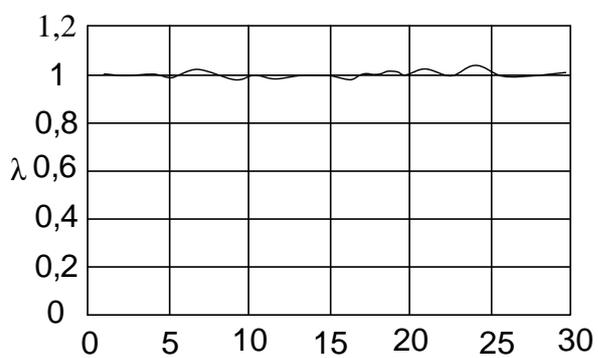
27	1,01243	0,987723	-0,01228	27	0,98125	1,019108	0,019108
28	0,99446	1,005571	0,005571	28	0,99468	1,005348	0,005348
29	0,97052	1,030375	0,030375		1,02684	0,973862	-0,02614



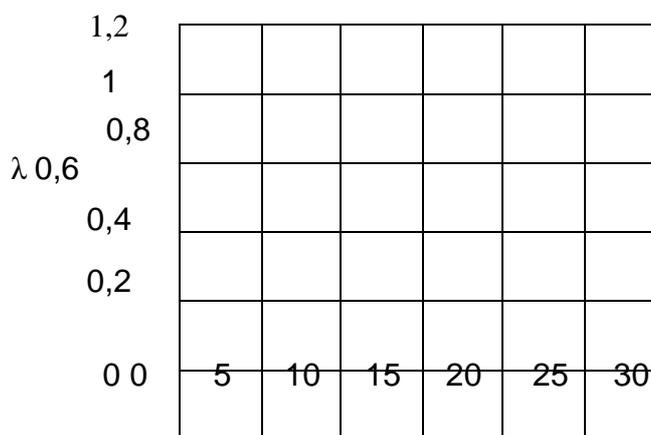
Параметр закона распределения N=500



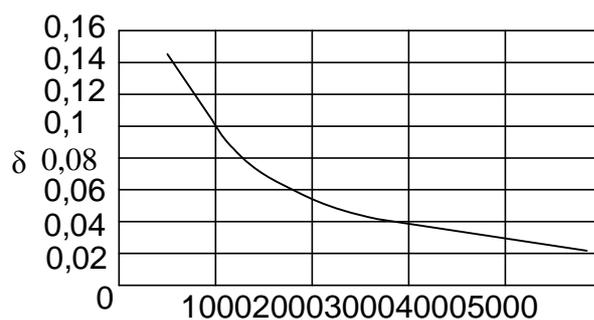
Параметр закона распределения N=5000



Параметр закона распределения N=2000



Параметр закона распределения N=1000



Минимальное значение модуля погрешности

Рисунок 1.4 - Результаты моделирования

Таблица 1.2

Вид распределения	Плотность	Алгоритм	Примечания
Равномерное	$\frac{1}{b-a}, a \leq x \leq b$	$x=a+(b-a)\xi$	-

Гистограмма	$\sum_{i=1}^s p_i f_i(x),$ $\sum_{i=1}^s p_i = 1,$ <p>где $f_i(x)$ – равномер- ное распределение с параметрами a_i и b_i</p>	$\xi_1 \rightarrow i; x = a_i + (b_i - a_i)\xi_2$	Сначала имитируется дискретная величина i заданная рядом распределения p_i
Треугольное	$\alpha (x - a); \leq x \leq \frac{b + a}{2}$ $a \frac{a}{2};$ $-\alpha (x - \frac{b + a}{2}) \leq x \leq b;$ $\frac{4}{(a - b)^2}$	$x = a + \frac{b - a}{2}(o_1 + o_2)$	

Нормальное	$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m_x)^2}{2\sigma^2}}$	<p>$x = m_x + \epsilon\sigma$:</p> <p>1) $e = \prod_{i=1}^N \frac{1}{\sigma} e^{-\frac{o_i^2}{2\sigma^2}}$</p> <p>$o_i = \frac{x - m_x}{\sigma} = \frac{1 - \xi}{\sigma}$</p> <p>а) $N = 6$</p> $e = \prod_{i=1}^6 \frac{1}{\sigma} e^{-\frac{o_i^2}{2\sigma^2}}$ <p>б) $N = 12$</p> $e = \prod_{i=1}^{12} \frac{1}{\sigma} e^{-\frac{o_i^2}{2\sigma^2}}$ $= 1 + 0,992o + 0,045o^2$ <p>2) e</p> <hr/> <p>при $\xi > 0,5$ в противном случае $\xi := 1 - \xi$ и перед ϵ поставить знак минус</p>	<p>Центральная предельная теорема</p> <p>Метод аппроксимации обратной функции</p>
------------	--	--	---

Продолжение таблицы 1.2

Вид распределения	Плотность	Алгоритм	Примечания
Экспоненциальное	$\lambda e^{-\lambda x}, x \geq 0$	1) $x = -\frac{1}{\lambda} \ln o_1$; 2) $x = -\frac{1}{\lambda} \ln e_i$, $i=1,2$; $\varepsilon_1 = -\xi_3 \ln(\xi_1, \xi_2)$; $\varepsilon_2 = -(1-\xi_3) \ln(\xi_1, \xi_2)$;	
Эрланга порядка S	$\frac{(\lambda x)^s e^{-\lambda x}}{s!}, x \geq 0$	$x = -\frac{1}{\lambda} \ln(o_1 \dots o_s)$	Сумма s экспоненциальных величин
Гиперэкспоненциальное	$\sum_{i=1}^s p_i f_i(x)$, $\sum_{i=1}^s p_i = 1$, где $f_i(x)$ – экспоненциальное распределение с параметрами λ_i	$\xi \rightarrow i$; $x = -\frac{1}{\lambda} \ln o_2$	См. примечание к распределению «гистограмма»

2 Задание на лабораторную работу №2 по разделу «системы массового обслуживания»

1. Изучить теоретический материал к лабораторной работе.

2. Написать процедуру (подпрограмму) с именем SMO расчета характеристик системы массового обслуживания (СМО). Входные параметры: λ - интенсивность входного потока; μ - интенсивность обслуживания;

m - число обслуживающих приборов(число каналов); k

– число мест ожидания.

Выходные параметры:

\underline{P} - коэффициент загрузки системы;

\underline{W} - среднее время ожидания в очереди; N_q

- длина очереди;

N - число требований (количество заявок) в системе; $P_0, \dots,$

P_k - вероятности состояний.

3. Варьируя интенсивность входного потока (время обслуживания нормированное $\mu=1$) получить график зависимости основных характеристик от загрузки ρ . Рассмотреть два случая:

а) число мест ожиданий не ограничено;

б) число мест ожидания ограничено.

Для второго случая рассчитать вероятности потери заявок.

Схема алгоритма программы приведена на рисунке 2.1

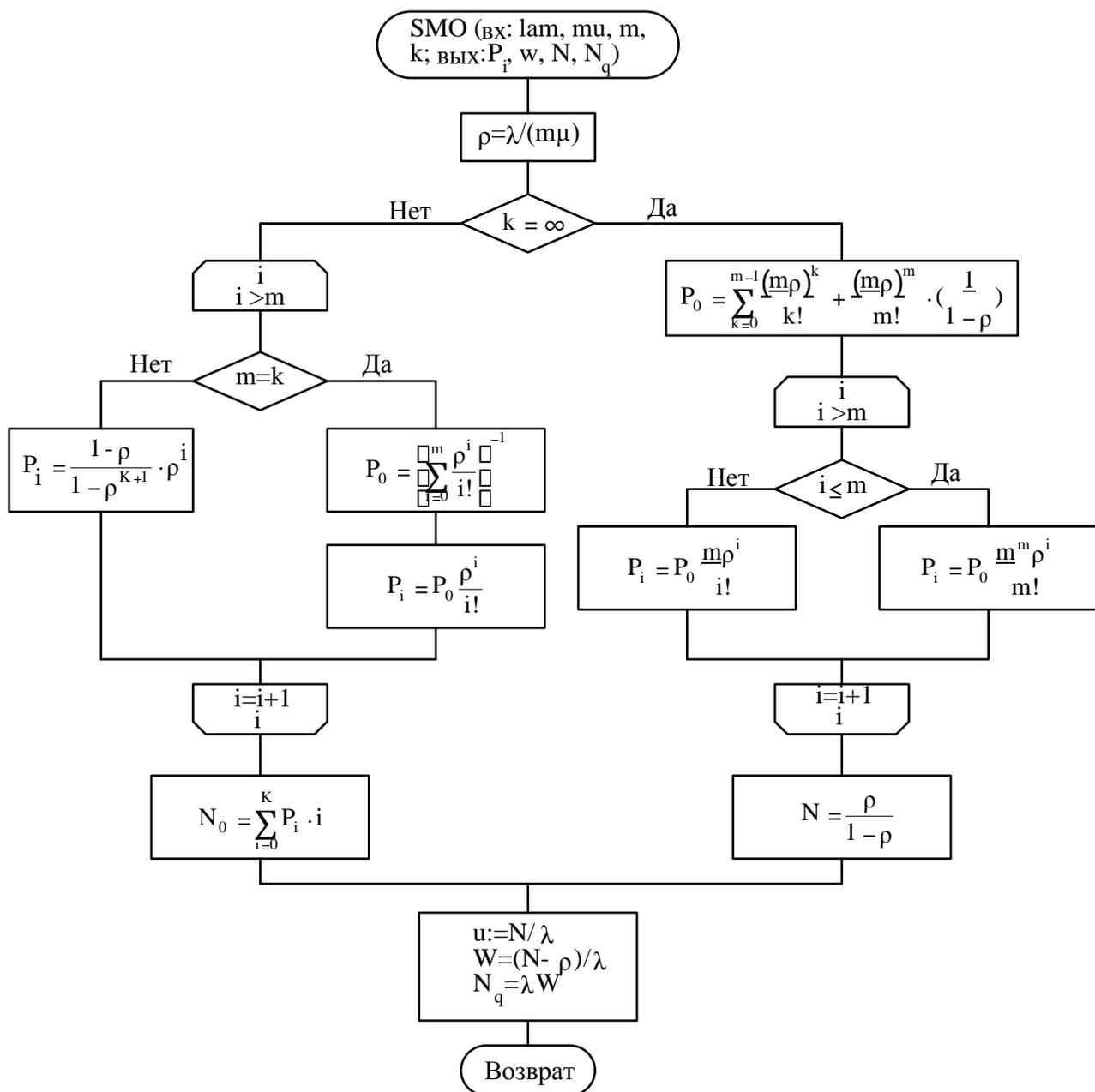


Рисунок 2.1

2.1 Содержание отчета

- 1 Исходный текст программы.
- 2 Результаты работы программы (значения основных характеристик СМО, графики зависимостей).
- 3 Выводы по работе.

Пример оформления результатов выполненной лабораторной работы приведен ниже. Исходный текст программы приведен в приложении Г.

Результат работы программы приведен на рисунках 2.2 и 2.3.

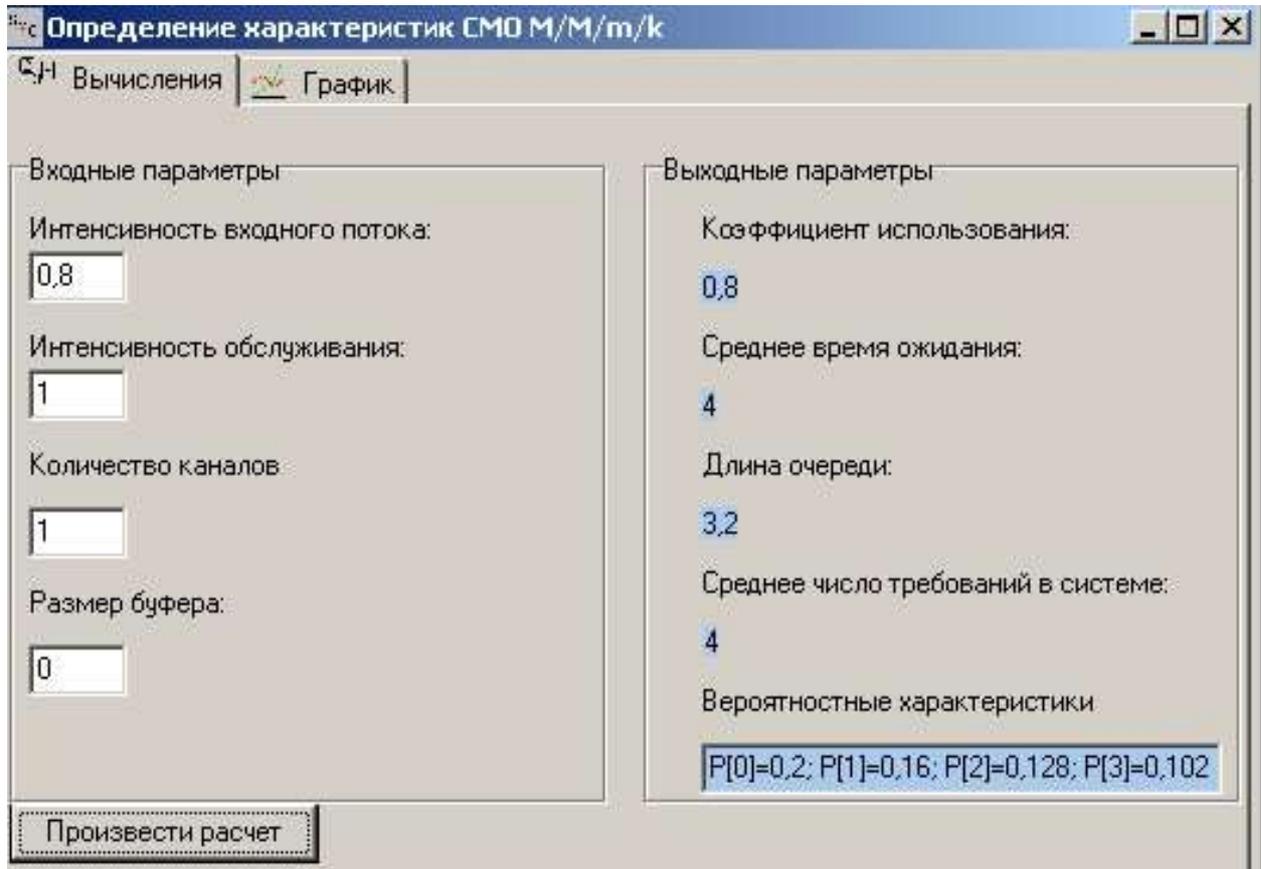


Рисунок 2.2 – СМО М/М/1/∞ с одним каналом и бесконечным буфером.

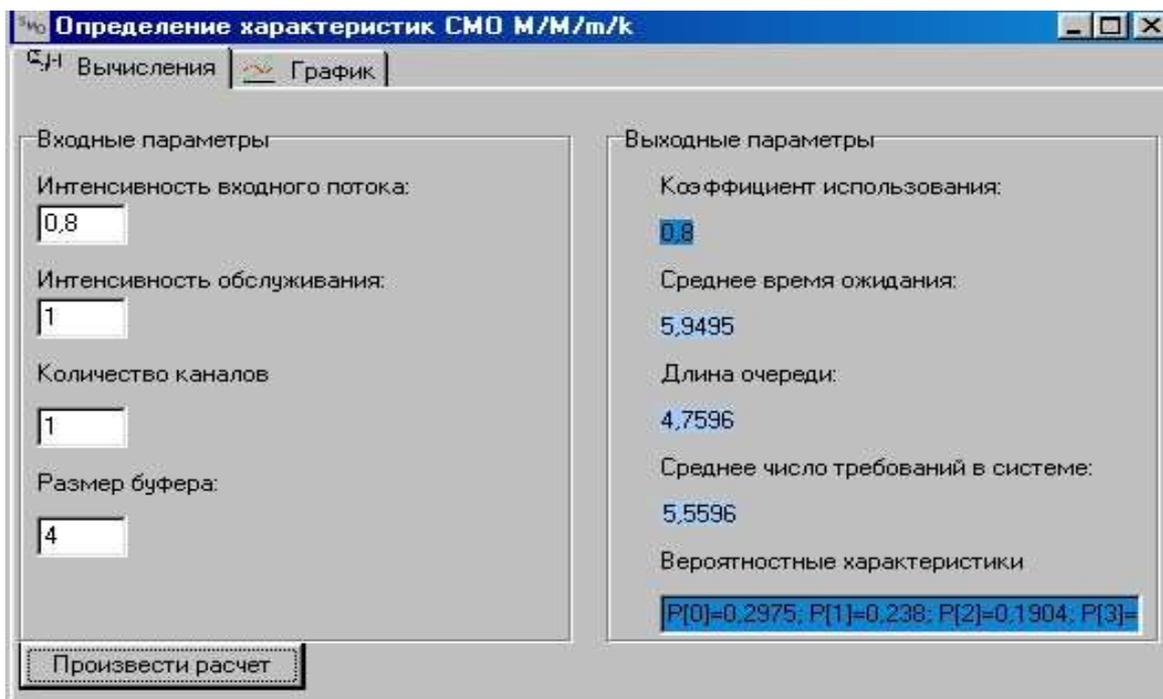


Рисунок 2.3 – СМО М/М/1/4 с буфером $k=4$.

Далее на рисунке 2.4 изображены графики зависимости среднего времени ожидания от коэффициента загрузки системы.

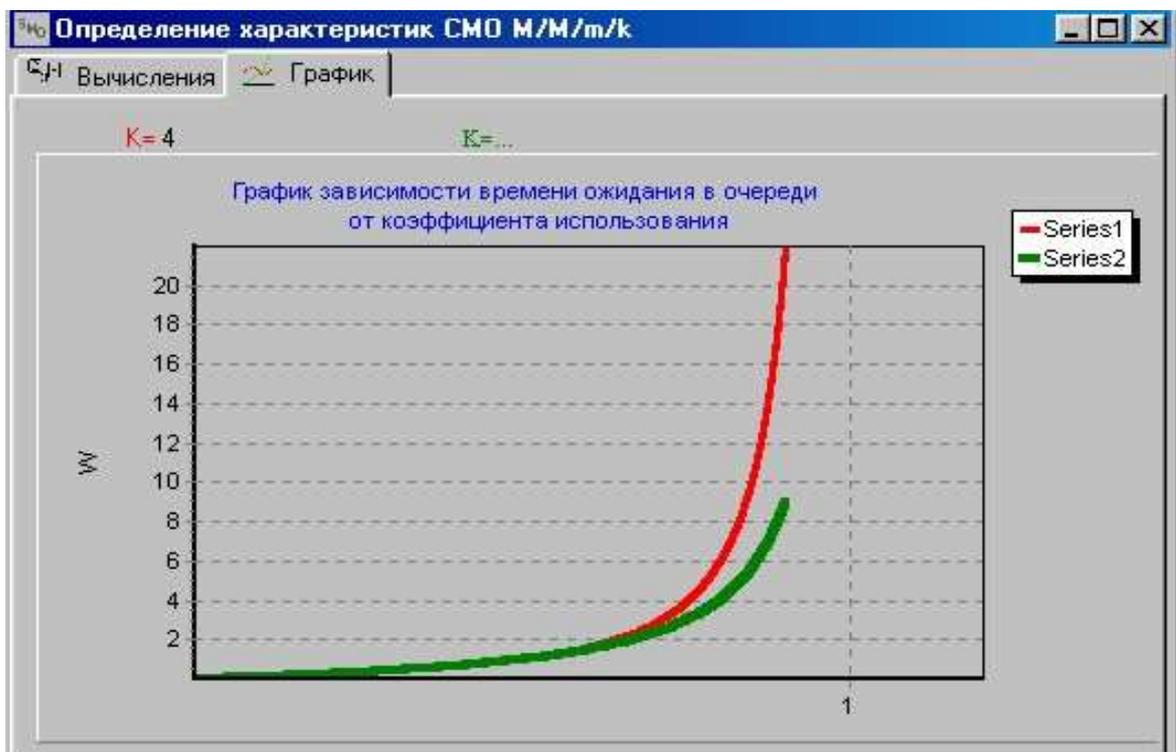


Рисунок 2.4 – Графики зависимости среднего времени ожидания от коэффициента загрузки системы (для заданного k и бесконечной очереди).

3 Задание на лабораторную работу №3 по разделу «Сети массового обслуживания»

1 Изучить теоретический материал к лабораторной работе.

2 Написать головную программу MAIN SSМО расчета характеристик сетевых моделей СМО с использованием процедуры SМО. Входные параметры:

- $r_{i,j}$ - матрица переходных вероятностей;
- N - число требований (количество заявок) в системе.

Кроме этого для каждого узла в сети задаются следующие параметры:

- γ - интенсивность поступления заявок от внешнего источника;
- μ - интенсивность обслуживания;
- m - число обслуживающих приборов (число каналов); - k - размер буфера.

Выходные параметры (узловые):

- λ - интенсивность входного потока узла;
- ρ - коэффициент загрузки системы;
- \bar{W} - среднее время ожидания заявки в узле;
- U - среднее время пребывания в узле;
- N_q - длина очереди для узла;
- P_0, \dots, P_k - вероятности состояний. Выходные параметры (сетевые):
- \bar{W}^c - среднее время ожидания заявки в сети;
- N^c - число требований (количество заявок) в сети;
- \bar{U}^c - среднее время пребывания заявки в сети;
- N_q^c - общая длина очереди.

2 Варьируя интенсивность внешнего источника γ выявить «узкие места» (наиболее загруженные узлы) сетевой модели в предположении, что все узлы представляют собой СМО с бесконечной очередью. Построить графики среднего времени ожидания в сети от интенсивности γ .

3 Один узел сетевой модели представить в виде СМО с конечной очередью и потерями. Потерянные заявки направить в другой узел, изменив исходную матрицу вероятностей передач R . Построить графики зависимостей среднего времени ожидания и вероятности потери заявок от интенсивности γ .

3.1 Содержание отчета

Пример оформления результатов выполненной лабораторной работы приведен ниже.

- 1 Исходный текст программы.
- 2 Результаты работы программы (значения основных характеристик СМО, графики зависимостей).
- 3 Выводы по работе.

В качестве примера рассмотрим открытую сеть массового обслуживания, граф передачи заявок которой представлен на рисунке 3.1.

Эта модель состоит из четырех узлов, причем в i -том узле находится m_i показательных обслуживающих приборов, для каждого из которых среднее время обслуживания равно $1/\mu_i$ секунд. Внешний пуассоновский источник генерирует поток требований, поступающих в i -й узел с интенсивностью γ_i требований в секунду (штриховые входящие линии на рис. 3.1). После обслуживания в i -м узле требование поступает в узел j с вероятностью r_{ij} или покидает

4

сеть с вероятностью $1 - \sum_{j=1}^4 r_{ij}$ (штриховые исходящие линии).

Для приведенного примера вероятности передач заявок заданы матрицей

	S_0	S_1	S_2	S_3	S_4
S_0	0	1/8	0	5/8	1/4
S_1	1/4	0	1/4	0	1/2

$$R = S_2 \begin{bmatrix} 1/3 & 1/3 & 0 & 1/3 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix},$$

$$S_3 \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$S_4 \begin{bmatrix} 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$

а

интенсивности внешнего источника $\gamma=(1,0,5,2)$.

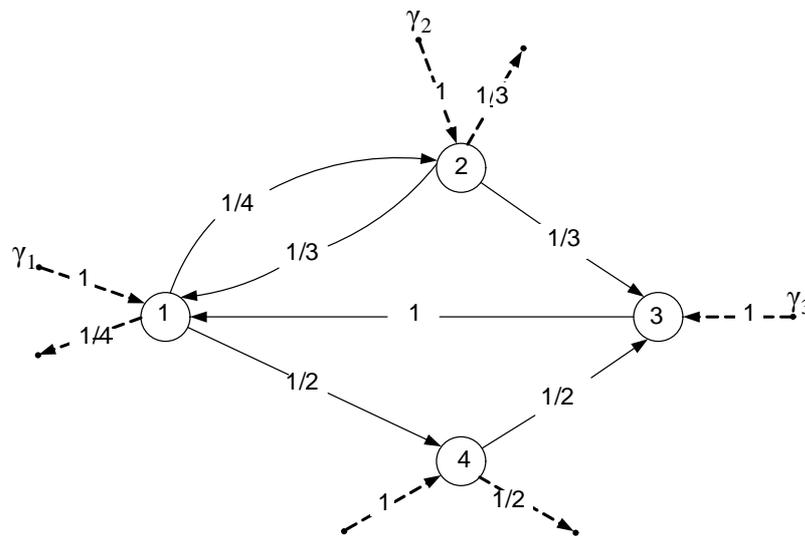


Рисунок 3.1 - Пример открытой сети (у ребер указаны значения $r_{i,j}$)

При расчете характеристик таких сетевых моделей кроме узловых характеристик, важно знать и сетевые характеристики. Для их определения вычисляют т.н. коэффициенты передачи заявок

$$\alpha_i = \lambda_i / \gamma.$$

Тогда характеристики сети в целом определяются следующим образом:

$$\bar{N}_q^c = \sum_{i=1}^n \bar{N}_{qi}; \quad \text{- среднее число заявок, ожидающих обслуживания в сети in}$$

$$\bar{N}^c = \sum_{i=1}^n \bar{N}_i; \quad \text{- среднее число заявок, пребывающих в сети}$$

$$\bar{W}^c = \sum_{i=1}^n \alpha_i \bar{W}_i; \quad \text{- среднее время ожидания заявки в сети}$$

$$\bar{U}^c = \sum_{i=1}^n \alpha_i \bar{U}_i;$$

- среднее время пребывания заявки в сети

$i=1$ где n – число узлов

сетевой модели.

Укрупненная схема алгоритма программы расчета характеристик сетевых моделей SSMO приведена на рисунке 3.2, а исходный текст программы в приложении Д.

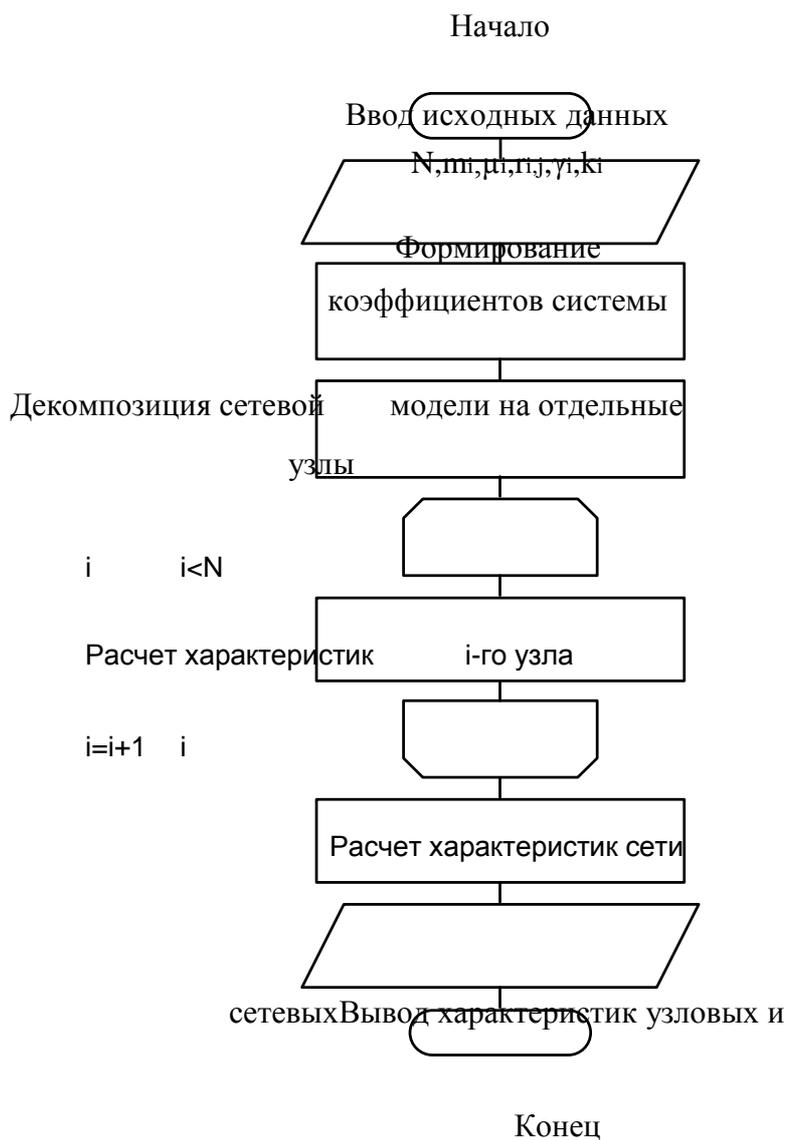
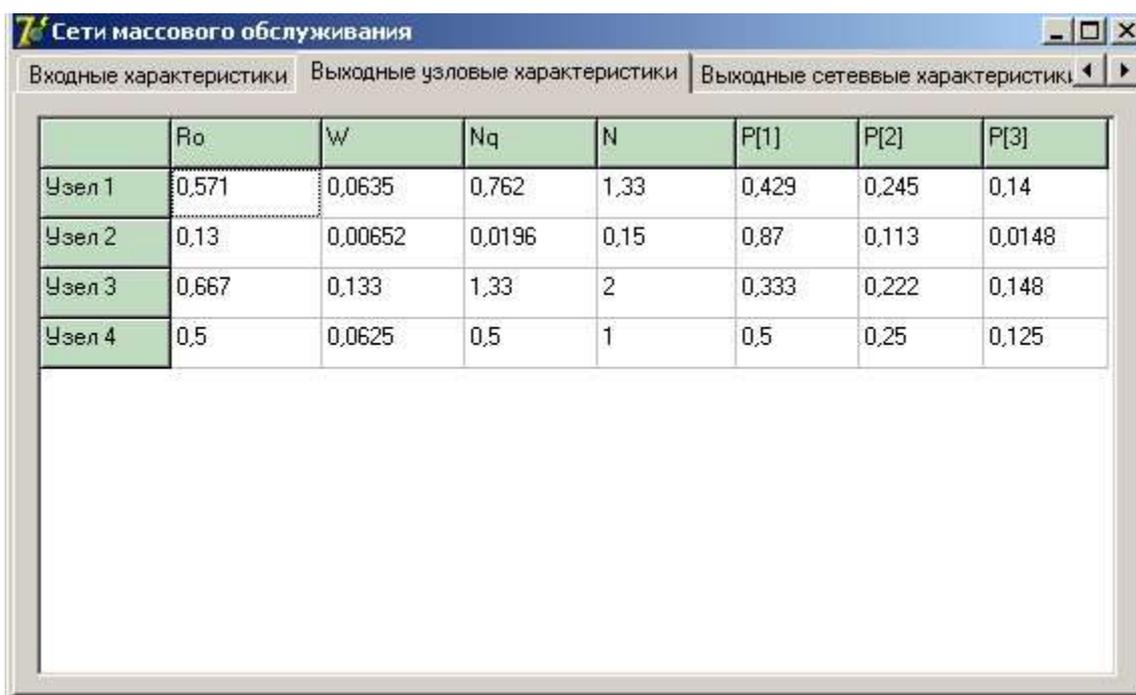


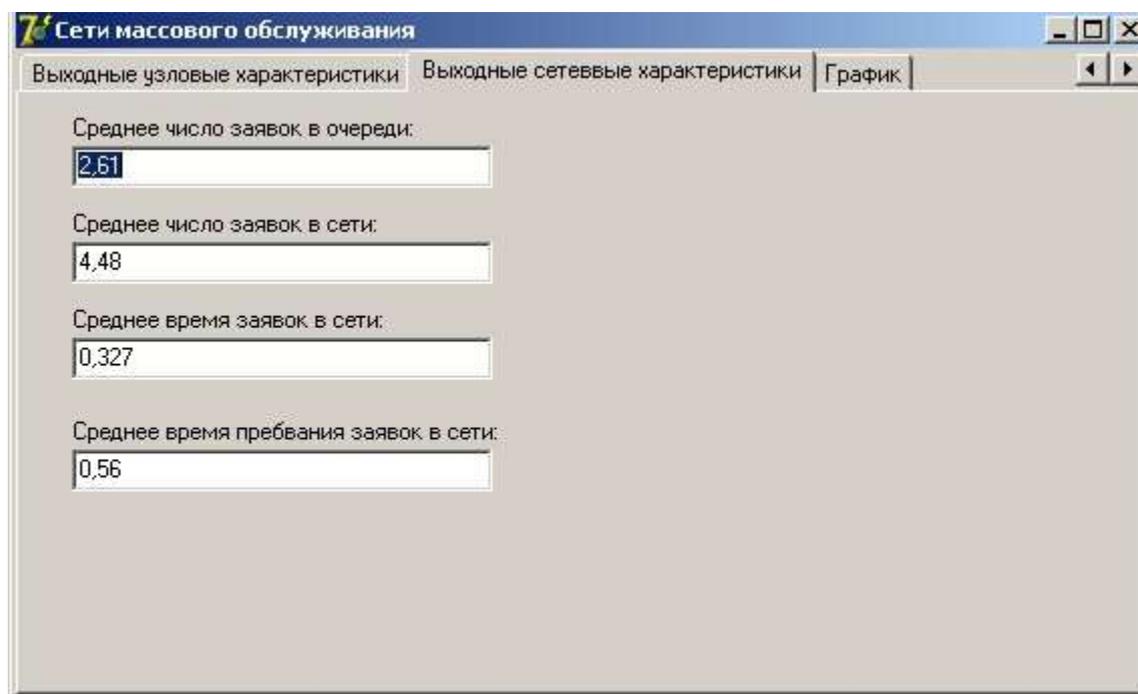
Рисунок 3.2

Далее на рисунках приведены примеры работы программы.



Узел	R_0	W	N_q	N	$P[1]$	$P[2]$	$P[3]$
Узел 1	0,571	0,0635	0,762	1,33	0,429	0,245	0,14
Узел 2	0,13	0,00652	0,0196	0,15	0,87	0,113	0,0148
Узел 3	0,667	0,133	1,33	2	0,333	0,222	0,148
Узел 4	0,5	0,0625	0,5	1	0,5	0,25	0,125

Рисунок 3.3 - Выходные узловые характеристики



Среднее число заявок в очереди:
2,61

Среднее число заявок в сети:
4,48

Среднее время заявок в сети:
0,327

Среднее время пребывания заявок в сети:
0,56

Рисунок 3.4 - Выходные сетевые характеристики



Рисунок 3.5 - График зависимости среднего времени ожидания заявки в сети от интенсивности внешнего потока

4. Интерактивная система вероятностного моделирования вычислительных систем PROBMOD

4.1 Структура программной системы

Программная система PROBMOD написана на алгоритмическом языке Turbo Pascal 7.0 с использованием среды программирования Delphi 6.0. Загрузочный модуль VERMOD.EXE имеет размер 4840 Кбайт.

Программная система легко расширяется и модифицируется. Она имеет удобный графический пользовательский интерфейс. Все кнопки на панели инструментов снабжаются комментариями о назначении при наведении на них курсора мыши. Система состоит из четырнадцати модулей.

Unit main, uTypeNet, unit2, unit3, unit4, unit5 uResGGM - модули организации интерфейса системы.

Unit mytypes – модуль описания используемых типов данных системы.

Unit U_Simg – модуль решения системы линейных алгебраических уравнений методом Гаусса с выбором ведущего элемента.

Unit LoadSave – модуль работы с файлами.

Unit uGG1 – модуль для расчета характеристик СМО типа GI/G/1/∞.

Unit uOdnNet, nUnit1 – модули для организации вычислений характеристик сети с однородным трафиком.

UnituNeOdnNet – модуль организации вычислений характеристик сети с неоднородным трафиком.

Алгоритм работы системы

1. Выбор типа трафика или выход из системы.

- 1 – ВС с однородным трафиком;
- 2 – расчет узла;
- 3 – ВС с неоднородным трафиком; 4 – выход из системы.

Если выбор 1, то

а) Ввод исходных данных: n – количество узлов; P – матрица вероятностей передач заявок от j -го узла к i -му; λ_0 – интенсивность входного потока для узлов; C_{λ_0} – коэффициенты вариации входного потока заявок для узлов; v – вектор быстродействия узлов; τ – вектор трудоемкости обработки заявок узлами; C_{μ} – коэффициенты вариации времени обработки заявок узлами;

NBUF – вектор номеров узлов с ограниченной емкостью; MBUF – соответствующий размер буферов.

б) Решение систем уравнений (3.21),(3.22) из /14/совместно с (3.11) и (3.12) методом последовательных приближений. В качестве первого приближения в уравнениях (3.11) и (3.12) используется замена $p'_0 = p_0$, $\tau'_\lambda = \tau_\lambda$, $D'_\lambda = D_\lambda$, а все узлы с бесконечной очередью. Это приближение в случае экспоненциальной сети не вносит погрешности, а уравнения (3.22) становятся линейными относительно искомым дисперсий.

в) Проведение последующих приближений для уточнения λ_{ivx} , D_{ivx} с заданной точностью методом двумерной диффузионной аппроксимации по формулам (3.12), (3.19)– (3.22) из /14/.

г) Определение характеристик узлов по формулам (4.1),(4.2),(4.5) – (4.7),(3.17) из /14/.

д) Расчет характеристик узлов с ограниченной емкостью и анализ $P_{отк}$. Если $P_{отк} \neq 0$, то переопределение $A1=A0$ и $B1=B0$ и переход к пункту г), для вычисления характеристик узлов с разреженным входным потоком.

е) Определение сетевых характеристик по формулам (4.8)–(4.11) из /14/.

ж) Вывод результатов.

з) Переход к п.1 в случае необходимости варьирования параметров или выхода из системы.

Если выбор 2, то

а) Ввод исходных данных узла λ_0 , C_{λ_0} , V , τ , C_μ , MBUF.

б) Расчет характеристик узла типа GI/G/1/ ∞ с неограниченной емкостью.

в) Расчет характеристик узла типа GI/G/1/ m с ограниченной емкостью.

г) Переход к п.1 в случае необходимости варьирования параметров или выхода из системы.

Если выбор 3, то алгоритм с одномерным трафиком модифицируется следующим образом:

а) Ввод данных: вводятся параметры M – количество классов трафика и для каждого класса вводятся вышеперечисленные параметры p^m , λ^m , C_{λ^m} , τ^m , C_μ^m . В частных случаях матрицы вероятностей передач P^m могут не зависеть от классов заявок.

б) Решение системы уравнений (3.21) для всех классов трафика $l=1, \dots, M$.

в) Приведение неоднородного трафика к однородному по формулам (3.24)–(3.27) из /14/.

Далее пункты б) – з) алгоритма расчета ВС с однородным трафиком по порядку.

На рисунке 4.1 приведена укрупненная схема работы программной системы.

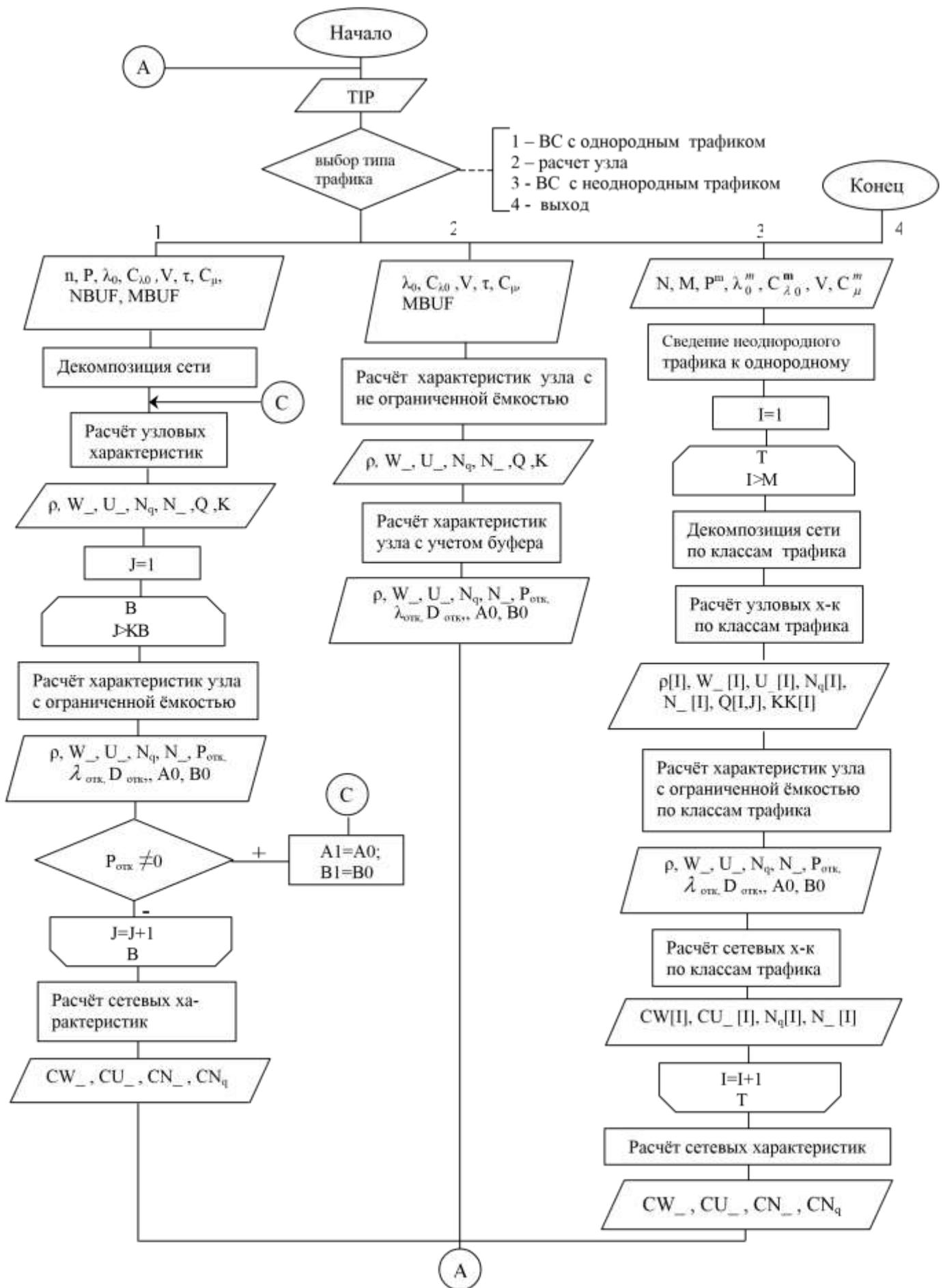


Рисунок 4.1 - Укрупнённая схема работы программной системы Основные процедуры и функции расчета узловых и сетевых характеристик.

Процедура VNGG1 предназначена для расчета характеристик СМО типа GI/G/1/∞.

PROCEDURE VNGG1 (A1, B2, A2, B2: double; Var R0, Tps, Tis, Wt, S0, Dt1: double; Var kk:integer; Var Q :TDinVec);

Таблица 4.1 - Спецификация входных параметров процедуры VNGG1

Имя параметра	Назначение	Тип
A1	Коэффициент сноса $a_1 = \tau_\lambda^{-1}$	double
B1	Коэффициент диффузии процесса x_1	double
A2	Коэффициент сноса $a_2 = \tau_\mu^{-1}$	double
B2	Коэффициент диффузии процесса x_2	double

Таблица 4.2 - Спецификация выходных параметров процедуры VNGG1.

Имя параметра	Назначение	Тип
R0	Коэффициент загрузки	double
Tps	Средний период занятости	double
Tis	Средний период простоя	double
Wt	Среднее время ожидания	double
S0	Средняя длина очереди	double
Dt1	Среднее значение дисперсии времени между заявками в выходном потоке	double
kk	Максимальный индекс	integer
$Q_k(k=1, \dots, kk)$	Вероятности того, что на периоде занятости поступило k заявок	TDinVec

Процедура VNGGM предназначена для расчета характеристик СМО типа GI/G/1/m с конечной очередью и потерями.

PROCEDURE VNGGM(A1, B2, A2, B2: double; Var Q:TDinVec; kk:integer; MBUF:byte; Var POT, ELOT, DOT, A0, B0: double);

Таблица 4.3 - Спецификация входных параметров процедуры VNGGM

Имя параметра	Назначение	Тип
A1	Коэффициент сноса $a_1 = \tau_\lambda^{-1}$	double

Продолжение таблицы 4.3

Имя параметра	Назначение	Тип
B1	Коэффициент диффузии процесса x_1	double
A2	Коэффициент сноса $a_2 = \tau_\mu^{-1}$	double
B2	Коэффициент диффузии процесса x_2	double
$Q_k(k=1, \dots, k_k)$	Вероятности того, что на периоде занятости в СМО поступило k заявок	TDinVec
k_k	Максимальный индекс	integer
MBUF	Максимально допустимое количество заявок (объем буфера)	byte

Таблица 4.4 - Спецификация выходных параметров процедуры VNGGM

Имя параметра	Назначение	Тип
POT	Вероятность потери заявки	double
ELOT	Интенсивность потока отказов	double
DOT	Дисперсия времени между заявками в потоке отказов	double
A0	Пересчитанный для разреженного потока коэффициент сноса	double
B0	Пересчитанный коэффициент диффузии	double

Процедура TAR1 вычисляет значение функции Макдональда $K_1(\cdot)$ в формулах (3.3) и (3.4) из /14/.

PROCEDURE TAR1 (x: double; Var Bk:double);

Таблица 4.5 - Спецификация входных параметров процедуры TAR1

Имя параметра	Назначение	Тип
x	Аргумент функции	double

Таблица 4.6 - Спецификация выходных параметров процедуры TAR1

Имя параметра	Назначение	Тип
Vk	Значение функции Макдональда	double

Функция QXY вычисляет значение переходной функции Q_{ϕ} в формуле (3.3).

FUNCTION QXY (A1, B1, A2, B2, X, Y:double): double ;

Входные параметры A1, B1, A2, B2 описаны в таблице 4.1.

Таблица 4.7 - Спецификация входных параметров функции QXY

Имя параметра	Назначение	Тип
X	Аргумент функции ϕ_{k-1}	double
Y	Аргумент функции ϕ_k	double

Функция QUV вычисляет значение переходной функции Q_{ψ} в формуле (3.4).

FUNCTION QUV (A1, B1, A2, B2, U, V:double): double ;

Входные параметры A1, B1, A2, B2 описаны в таблице 4.1.

Таблица 4.8 - Спецификация входных параметров функции QUV

Имя параметра	Назначение	Тип
U	Аргумент функции ψ_k	double
V	Аргумент функции ϕ_{k-1}	double

4.2 Функциональные возможности системы

Интерактивная система вероятностного моделирования компьютерных сетей на основе метода обобщенной двумерной диффузионной аппроксимации позволяет проводить анализ и проектировать компьютерные сети, учитывая ограничения на ресурсы, неоднородность трафика сети и произвольный закон поступления и обслуживания заявок.

Система может использоваться, когда в процессе проектирования сети отсутствует подробная информация о законах распределения вероятностей

параметров потоков, а имеются данные лишь о средних значениях и дисперсиях этих распределений. Она решает следующие задачи:

- расчет характеристик качества функционирования различных компонент компьютерных сетей, включая оценку вероятностно-временных характеристик узлов коммутации и маршрутизации;
- анализ производительности сетей;
- анализ буферной памяти узлов;
- определение потоков отказов и загрузки линий связи при передаче данных.

Входные данные:

- матрица вероятностей перехода заявок $\{P_{ij}\}$;
- λ_{0i} – интенсивность входного потока в i -й узел;
- $C_{\lambda 0}$ – коэффициент вариации входного потока заявок в i -й узел; - V_i – быстродействие i -ого узла;
- τ_i – трудоёмкость обработки заявки i -м узлом;
- $C_{\mu i}$ – коэффициент вариации времени обработки; - классы узлов;
- MBUF - размер буфера, для узлов с ограниченной емкостью.

В случае неоднородного трафика данные задаются для каждого типа трафика.

Выходные данные для каждого узла сети:

- среднее количество заявок, прошедших за период занятости \bar{Y} ;
- средняя длина периода простоя \bar{I} ;
- среднее время ожидания \bar{W} ; - средняя длина очереди N ; q
- среднее количество заявок \bar{N} ;
- среднее времени между заявками в выходном потоке $\bar{\tau}_{\text{вых}}$;
- дисперсия этого времени $D_{\text{вых}}$;
- вероятности q_k того, что на периоде занятости поступило k заявок; - для узла с ограниченной емкостью:
 - а) интенсивность потока отказов $\lambda_{\text{отк}}$;
 - б) дисперсия $D_{\text{отк}}$ времени между заявками в потоке отказов.

В случае неоднородного трафика характеристики рассчитываются для каждого типа трафика.

Выходные данные для сети:

- среднее время ожидания заявки в сети \bar{W} ; c

- среднее время пребывания заявки в сети U ; c
- общая длина всех очередей в сети \bar{N} ; c_q
- общее количество заявок в сети \bar{N} . c

На рисунке 4.2 приведена функциональная схема системы, на которой представлены основные функциональные возможности системы и ресурсы, необходимые для их реализации.

1 Расчет характеристик сети с однородным трафиком. Ввод информации возможен из файла данных и с клавиатуры. Вывод информации осуществляется на экран и в файл Rez1.TXT.

2 Расчет характеристик сети с неоднородным трафиком. Ввод информации для каждого класса аналогичен пункту 1. Вывод узловых и сетевых характеристик по классам трафика и характеристик сети возможен как в файл Rez2.TXT , так и на экран.

3 Расчет характеристик узла сети с однородным трафиком с конечной очередью и потерями. Вводится информация о номере узла и размере буфера, вероятности q_k , $k = 1, k_k$, индекс k_k , A_1 , B_1 , A_2 , B_2 . Вывод результатов возможен в файл и на экран.

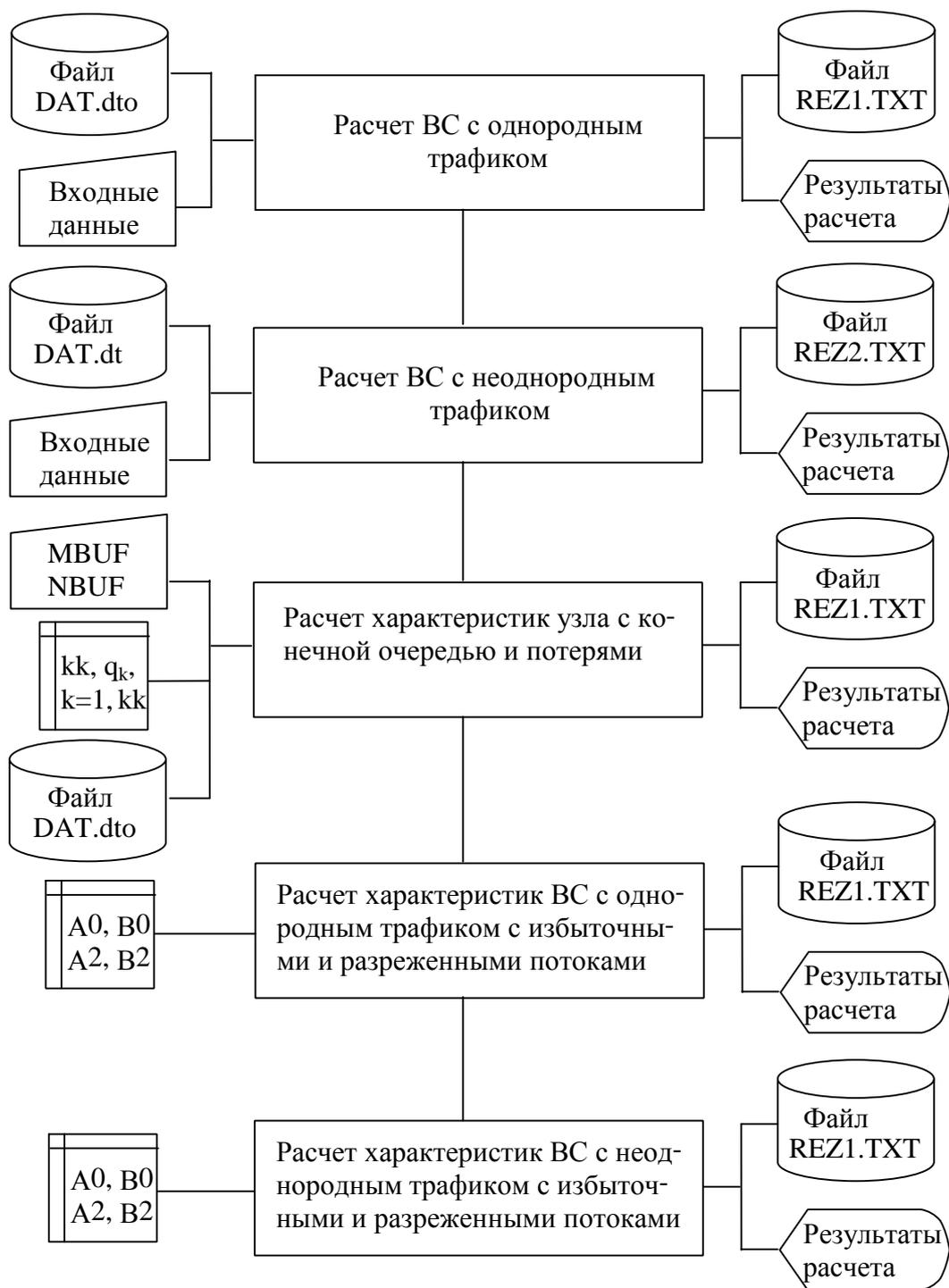


Рисунок 4.2–Функциональная схема программной системы

4 Расчет характеристик сети с однородным трафиком с избыточными и разреженными потоками. Входные данные берутся из оперативной памяти. Результаты расчета записываются в файл или/и отображаются на экране.

5 Расчет характеристик сети с неоднородным трафиком с избыточными и разреженными потоками. Ввод и вывод информации аналогичен пункту 4.

4.3 Инструкция пользователя

Запуск программы.

Запускается исполняемый файл «vermod.exe»

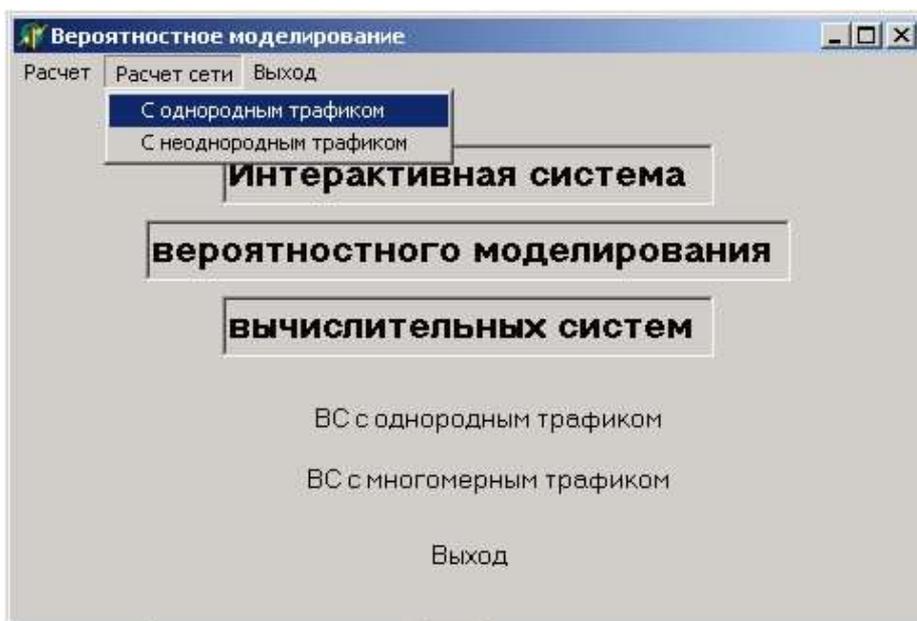


Рисунок 4.3- Экранная форма выбора моделируемой ВС

В появившейся форме (рисунок 4.3) выбирается пункт «Расчет сети», а потом пункт, соответствующий типу моделируемой сети.

1 Моделирование ВС с однородным трафиком

Выбирается пункт «С однородным трафиком».

Появится форма для ввода данных (рисунок 4.4).

1.1 Ввод данных

В поле «Кол. узлов» вводится количество узлов сети.

Для ввода новых данных (или очистки формы от старых) нажимается кнопка  на панели инструментов.

- В форме для ввода классов узлов сети (рисунок 4.5) указывается -
- класс узла 1,2,3;
 - Mbuf- размер буфера для классов узлов 2 и 3.

Рисунок 4.4- Экранная форма ввода исходных данных для ВС с однородным трафиком

Узел	Узел 1	Узел 2	Узел 3	Узел 4	Узел 5	Узел 6	Узел 7
Класс	1	2	2	1	2	1	1
MBuf		3	10		5		

Рисунок 4.5- Экранная форма ввода классов узлов сети и размеров буферов

Входными данными для ВС с однородным трафиком являются:

- $\{P_{ij}\}$ – матрица вероятностей перехода заявок из i -ого узла в j -ый (Рисунок 4.4, таблица «Матрица вероятностей перехода заявок»). Сумма элементов в любой не нулевой строке матрицы всегда должна быть равна 1;
- λ_{0i} – интенсивность входного потока для i -ого узла (Рисунок 4.4, таблица «Инт. поступл.» в группе «Поступление заявок»);
- $C\lambda_{0i}$ – коэффициент вариации входного потока заявок в i -й узел (Рисунок 4.4, таблица «Коэф. вариации» в группе «Поступление заявок»);
- V_i – быстродействие i -ого узла (Рисунок 4.4, таблица «Быстр. узлов» в группе «Обработка заявок»);
- τ_i – трудоёмкость обработки заявки i -м узлом (на форме таблица «Трудоёмкости» в группе «Обработка заявок») (отношение V_i / τ_i определяет μ_i – интенсивность обработки заявки i -м узлом);
- λ_{0i} – интенсивность входного потока (Рисунок 4.4, таблица «Коэф. вариации» в группе «Обработка заявок»).

1.2 Сохранение исходных данных в файле

Введённые исходные данные можно сохранить в файле.

Для сохранения исходных данных нажимается кнопка  на панели инструментов. В появившемся диалоге указывается имя файла с расширением “*.dto”.

1.3 Загрузка исходных данных из файла

Для загрузки исходных данных нажимается кнопка  на панели инструментов. В появившемся диалоге выбирается файл из предложенного списка.

1.4 Расчет характеристик модели

Для расчета характеристик модели нажимается кнопка  на панели инструментов. После расчета выводится экранная форма с рассчитанными характеристиками модели. Рассчитанные характеристики можно сохранить в текстовом файле.

1.5 Просмотр результатов

Для просмотра результатов последнего расчета нажимается кнопка  на панели инструментов.

Результаты расчетов представлены экранными формами на рисунках 4.6 и 4.7.



The screenshot shows a window with the title "Результат расчета характеристик сети с однородным трафиком." It contains two tables. The first table, "Характеристики узлов:", lists data for five nodes. The second table, "Характеристики сети:", shows overall network statistics.

Характеристики узлов:						
	К. пер.	Инт. вх. п.	Кэф. з.	Зан. кан.	Заяв. в. уз.	Дл. оч.
Узел 1	0,007	0,330	0,132	1,000	0,139	0,007
Узел 2	0,019	0,330	0,396	1,000	0,415	0,019
Узел 3	0,120	0,330	0,626	1,000	0,746	0,120
Узел 4	0,015	0,330	0,264	1,000	0,279	0,015
Узел 5	0,162	0,330	0,406	1,000	0,531	0,162

Характеристики сети:			
Производ.	Сред. вр. пре	Сред. вр. ож.	Общ. дл. оч.
2,111	6,879	0,979	0,323

Рисунок 4.6- Экранная форма результатов расчета характеристик модели ВС с однородным трафиком

Нажимается кнопка «Узлы GGM» (рисунок 4.6) для просмотра характеристик узлов с ограниченной емкостью.

Экранная форма «Характеристики узла типа GGM » (рисунок 4.7) позволяет просмотреть следующие характеристики узлов:

- вероятность потери заявки $P_{отк}$;
- интенсивность потока отказов $\lambda_{отк}$;
- дисперсию времени между заявками в потоке отказов $D_{отк}$.

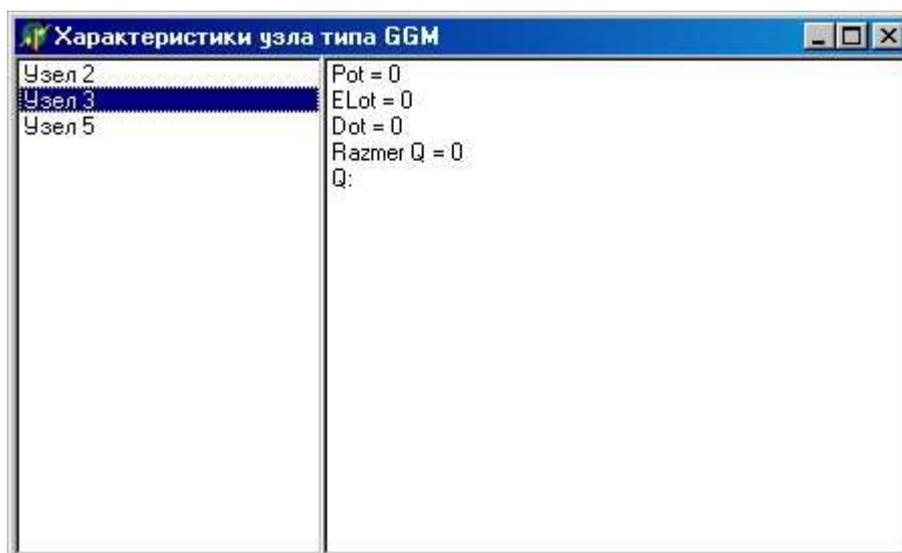
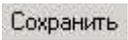


Рисунок 4.7- Экранная форма результатов расчета характеристик узлов с ограниченной емкостью.

1.6 Сохранение результатов расчета

Для сохранения результатов расчета нажимается кнопка .

2 Моделирование ВС с неоднородным трафиком

В экранной форме (рисунок 4.3) выбирается пункт «С неоднородным трафиком».

Появится форма для ввода данных (рисунок 4.8).

2.1 Ввод данных

- N- количество узлов модели;
- M- количество типов заявок, присутствующих в трафике.

Для ввода новых данных (или очистки формы от старых) нажимается кнопка  на панели инструментов.

В форме для ввода классов узлов сети (рисунок 4.5) указывается

- класс узла;
- Mbuf- размер буфера для классов 2 и 3.

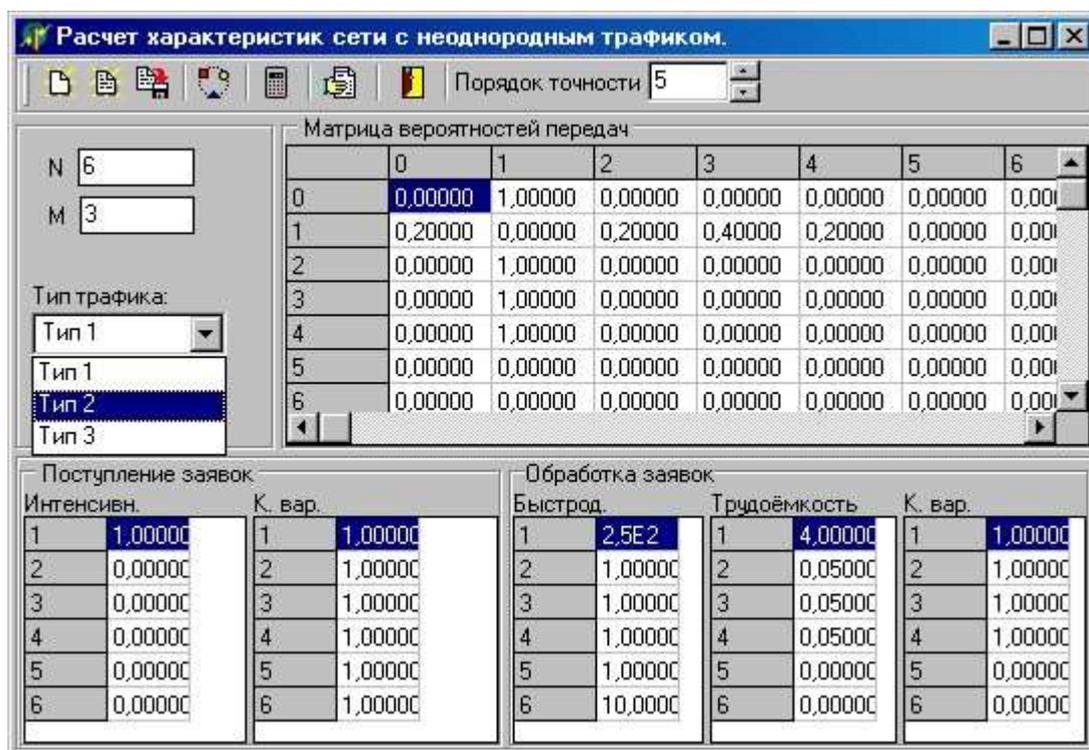


Рисунок 4.8 - Экранная форма ввода исходных данных для ВС с неоднородным трафиком

Входными данными для ВС с неоднородным трафиком являются:

- $\{P_{ij}^m\}$ – матрица вероятностей перехода заявок из i -ого узла в j -ый для потока типа « m » (рисунок 4.8, таблица «Матрица вероятностей передач») (сумма в любой не нулевой строке матрицы всегда должна быть равна 1);
- λ_{0i}^m – интенсивность входного потока типа « m » в i -й узел (рисунок 4.8, таблица «Инт. поступл.» в группе «Поступление заявок»);
- $C_{\lambda 0im}$ – коэффициент вариации входного потока типа « m » в i -й узел (рисунок 4.8, таблица «К. вар.» в группе «Поступление заявок»);
- V_i – быстродействие i -ого узла (рисунок 4.8, таблица «Быстр. узлов» в группе «Обработка заявок»);
- τ_i^m – трудоёмкость обработки заявки потока типа « m » i -м узлом (рисунок 4.8, таблица «Трудоёмкости» в группе «Обработка заявок»);

- $C_{\mu i}^m$ – коэффициент вариации времени обработки заявки потока типа «m» i-м узлом (рисунок 4.8, таблица «К. вар.» в группе «Обработка заявок»).
- Тип потока, характеристики которого вводятся, определяется пунктом меню «Тип трафика».

2.2 Сохранение исходных данных в файле

Введённые исходные данные можно сохранить в файле.

Для сохранения исходных данных нажимается кнопка  на панели инструментов. В появившемся диалоге указывается имя файла с расширением “*.dt”.

2.3 Загрузка исходных данных из файла

Для загрузки исходных данных нажимается кнопка  на панели инструментов. В появившемся окне выбирается файл из предложенного списка.

2.4 Расчет характеристик модели

Для расчета нажимается кнопка  на панели инструментов. После расчета выводится экранная форма с рассчитанными характеристиками модели (рисунок 4.9). Рассчитанные характеристики можно сохранить в текстовом файле.

2.5 Просмотр результатов.

Для просмотра результатов последнего расчета нажимается кнопка  на панели инструментов. В окне просмотра результатов для трафика характеристики отображаются для каждого типа заявок. Чтобы просмотреть результат для заданного типа трафика пользуются ниспадающим меню «Тип трафика».

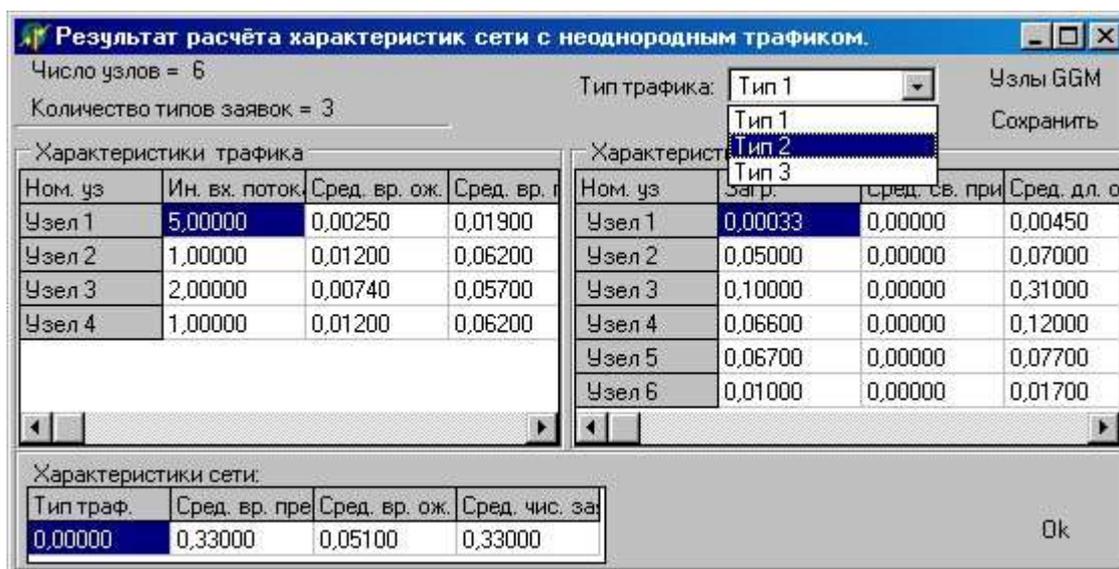


Рисунок 4.9-Экранная форма результатов расчета характеристик модели ВС с неоднородным трафиком

Нажимается кнопка «Узлы GGM» (рисунок 4.9) для просмотра характеристик узлов с ограниченной емкостью.

Экранная форма «Характеристики узла типа GGM » (рисунок 4.7) позволяет просмотреть следующие характеристики узлов:

- вероятность потери заявки $P_{отк}$;
- интенсивность потока отказов $\lambda_{отк}$;
- дисперсию времени между заявками в потоке отказов $D_{отк}$

2.6 Сохранение результатов

Для сохранения результатов расчета нажимается кнопка Сохранить.

3 Расчет характеристик узла

В окне (рисунок 4.3) выбирается пункт « Расчет узла »
 Появится форма для ввода данных (рисунок 4.10).

3.1 Ввод данных

Вводятся следующие характеристики узла:

- λ –интенсивность входного потока;
- C_λ – коэффициент вариации времени поступления заявок;

- τ – трудоёмкость обработки заявки узлом;
- C_μ – коэффициент вариации времени обработки заявок.

Для узла с ограниченной емкостью вводится дополнительно -
Mbuf-размер буфера.

3.2 Расчет характеристик

Для расчета узла с бесконечной емкостью нажимается кнопка «GG1».

Для расчета узла с конечной емкостью нажимается кнопка «GGM».

Результаты расчета отображаются в правой части окна (рисунок 4.10).

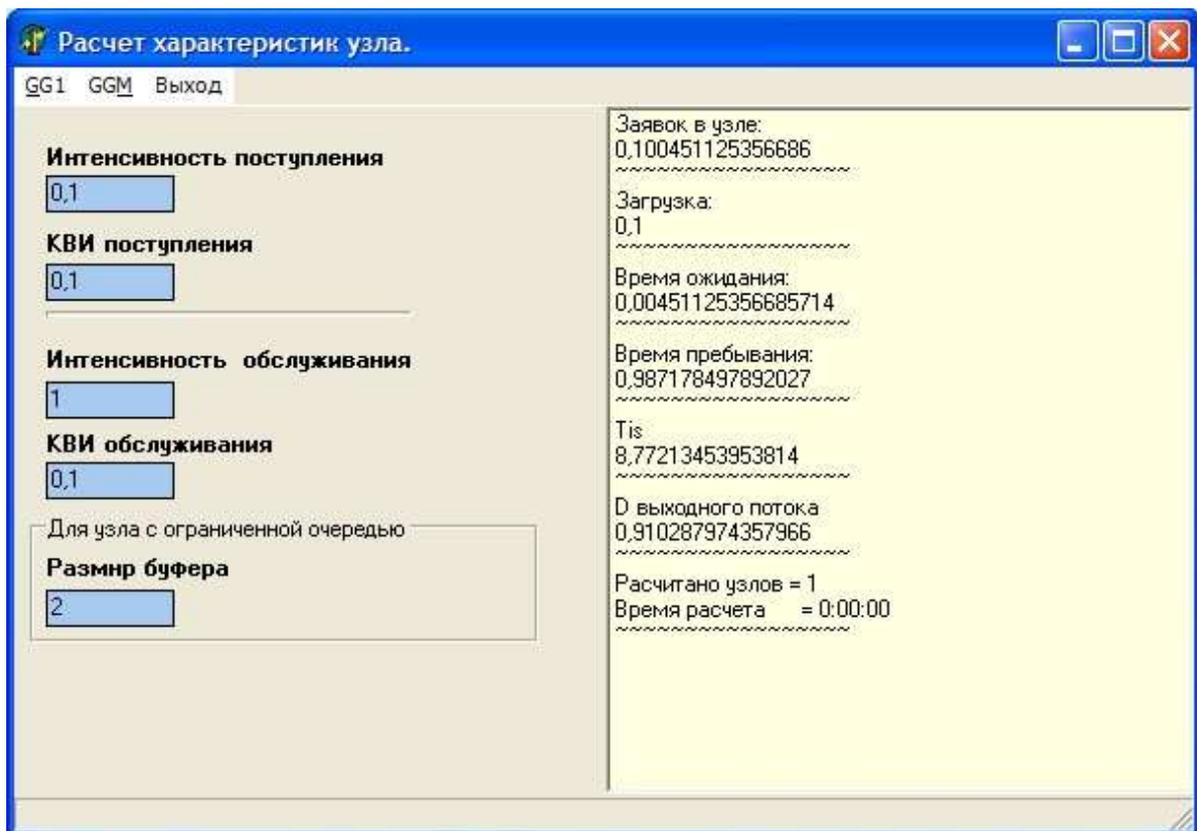


Рисунок 4.10- Экранная форма для ввода данных и расчета характеристик узла.

4.4 Результаты проведенных расчетов и их анализ

Расчеты одиночных узлов сети

Точность метода обобщенной двумерной диффузионной аппроксимации с использованием процедуры VNGG1 исследована для широкого диапазона изменения параметров трафика и закона обслуживания. При этом

коэффициент загрузки ρ варьировался от 0,1 до 0,9, а коэффициенты вариаций распределений длин интервалов между заявками во входном потоке C_λ и времени обслуживания C_μ – от 0 до 5.

В таблице 4.9 приведены значения среднего количества заявок \bar{N} в узле, а для сравнения в этой же таблице приведены значения \bar{N} , полученные имитационным моделированием. Анализ этих данных показывает, что точность метода обобщенной двумерной диффузионной аппроксимации, несомненно, выше точности известных методов одномерной диффузионной аппроксимации. Таким образом, относительная погрешность предлагаемого метода для широкого диапазона изменения параметров примерно равномерна и не превышает 5-7%.

При проведении экспериментов в качестве одного из параметров моделирования задавалось количество циклов занятости, которое в зависимости от загрузки изменялось от 1000 до 20000 /12/.

Таблица 4.9 – Зависимость среднего количества заявок в узле \bar{N} от ρ , C_μ , C_λ

		\bar{N}				
		C_μ				
ρ	C_λ	0,1	0,5	1,0	2,0	5,0
0,1	0,1	0,100	0,101	0,105	0,117	0,211
		0,101	0,101	0,101	0,103	0,214
	0,5	0,111	0,111	0,111	0,117	0,220
		0,101	0,10	0,102	0,113	0,220
	1,0	0,111	0,112	0,115	0,125	0,232
0,096		0,10	0,101	0,119	0,242	
2,0	0,118	0,122	0,133	0,171	0,360	
	0,108	0,129	0,148	0,154	0,315	
5,0	0,421	0,394	0,431	0,586	1,076	
	-	-	0,506	-	-	
0,3	0,1	0,302	0,314	0,340	0,459	1,746
		0,301	0,301	0,315	0,489	1,824
	0,5	0,315	0,320	0,351	0,519	1,824
		0,308	0,317	0,349	0,521	1,855
	1,0	0,334	0,352	0,404	0,605	1,961
		0,336	0,362	0,401	0,606	1,948

Продолжение таблицы

4.9

		$N^{\bar{}}$				
		C_{μ}				
ρ	C_{λ}	0,1	0,5	1,0	2,0	5,0
0,3	2,0	0,518	0,545	0,640	0,924	2,397
		-	-	0,472	0,846	2,307
	5,0	4,522	4,361	4,367	4,560	6,142
		-	-	-	-	-
0,5	0,1	0,506	0,534	0,671	1,344	6,429
		0,500	0,500	0,647	1,376	6,596
	0,5	0,534	0,576	0,737	1,446	6,607
		0,556	0,589	0,738	1,467	6,691
	1,0	0,670	0,741	0,945	1,714	6,934
		0,676	0,751	0,954	1,713	6,959
	2,0	1,748	1,704	1,911	2,764	8,145
		-	1,108	1,531	2,565	7,902
	5,0	11,449	11,421	11,633	12,535	17,945
		-	-	-	-	-
0,7	0,1	0,715	0,844	1,384	3,746	20,561
		0,70	0,773	1,360	3,779	20,912
	0,5	0,844	1,019	1,601	3,990	20,949
		0,907	1,040	1,608	4,040	21,142
	1,0	1,459	1,652	2,285	4,732	21,747
		1,439	1,665	2,285	4,742	21,878
	2,0	4,691	4,817	5,423	7,834	24,864
		3,794	4,034	4,868	7,470	24,556
	5,0	27,959	28,093	28,750	31,460	48,791
		21,754	21,863	22,984	29,049	44,863
0,9	0,1	0,964	1,913	4,848	16,798	100,819
		0,934	1,740	4,718	16,881	101,826
	0,5	1,996	2,925	5,886	17,831	102,213
		1,940	2,801	5,794	17,918	102,899
	1,0	5,145	6,110	9,096	21,062	105,380
		4,974	5,939	8,968	21,072	106,163
	2,0	18,135	19,036	22,112	34,153	118,207
		17,551	18,317	21,317	33,796	117,274
	5,0	111,128	112,004	115,126	127,566	212,596
		107,642	107,616	-	-	-

1-я строка – результаты двумерного диффузионного приближения, 2-я строка – результаты имитационного моделирования.

5 Задание на лабораторную работу №4 с использованием программной системы PROBMOD

- 1 Изучить теоретический материал к лабораторной работе.
- 2 Изучить схему алгоритма, функциональные возможности и инструкцию пользователя системы PROBMOD (п.п. 4.1-4.4).
- 3 Рассчитать основные характеристики отдельного узла сетевой модели типа GI/G/1/∞ для диапазона изменения параметров: загрузки ρ от 0,1 до 0,9; коэффициентов вариаций времени поступления C_λ и времени обслуживания C_μ от 0,1 до 2.
- 4 Построить графики зависимости среднего времени ожидания \bar{W} заявки в очереди от указанных параметров.
- 5 Рассчитать основные характеристики сетевой модели (вариант задается преподавателем). Варьированием интенсивности внешнего источника λ_0 определить узкие места данной модели.
- 6 Рассмотреть пункты 2 и 3 данного задания для узла типа GI/G/1/m с потерями, где m – емкость буфера ($m=2,3,5,10,15$). Построить графики зависимостей вероятности потери сообщения $P_{отк}$ от указанных в пункте 2 параметров.
- 7 Используя полученные в п.5 графики зависимости решить задачу оптимального выбора объемов канальных буферов при заданном качестве обслуживания: $P_{отк} < P_0, W < W_0$.

Пример выполнения задания приведен ниже.

В таблице 5.1 приведены значения среднего времени ожидания заявок \bar{W} в отдельном узле.

Таблица 5.1 – Зависимость среднего времени ожидания заявок \bar{W} в узле от ρ, C_μ, C_λ

\bar{W}
C_μ

Продолжение таблицы

ρ	C_λ	0,1	0,5	1,0	2,0	5,0
0,1	0,1	0,005	0,015	0,046	0,175	1,106
	0,5	0,101	0,104	0,109	0,195	1,205
	1,0	0,112	0,117	0,149	0,252	0,318
	2,0	0,117	0,220	0,334	0,713	2,601
	5,0	2,211	2,939	3,306	4,862	9,764
0,3	0,1	0,007	0,046	0,132	0,531	4,820
	0,5	0,049	0,066	0,169	0,731	5,080
	1,0	0,114	0,173	0,348	1,015	5,538
	2,0	0,726	0,816	1,133	2,080	6,990
	5,0	12,072	13,536	13,556	14,202	19,472

5.1

		\bar{W}				
		C_μ				
ρ	C_λ	0,1	0,5	1,0	2,0	5,0
0,5	0,1	0,011	0,068	0,342	1,689	11,857
	0,5	0,068	0,152	0,475	1,892	12,214
	1,0	0,339	0,482	0,890	2,428	12,867
	2,0	2,406	2,408	2,822	4,529	15,291
	5,0	21,897	21,842	22,265	24,070	34,889
0,7	0,1	0,021	0,205	0,977	4,352	28,373
	0,5	0,206	0,456	1,288	4,699	28,927
	1,0	1,084	1,359	2,264	5,760	30,066
	2,0	5,701	5,881	6,748	10,192	34,520
	5,0	38,942	39,133	40,072	43,943	68,701
0,9	0,1	0,071	1,126	4,386	17,665	111,022
	0,5	1,218	2,250	5,540	18,813	112,570
	1,0	4,716	5,789	9,107	22,402	116,088
	2,0	19,150	20,151	23,569	36,947	130,341
	5,0	122,476	123,449	126,918	140,740	235,218

На рисунке 5.1 приведены графики зависимости задержки сообщений в узле от параметров трафика и закона обслуживания.

Эти результаты доказывают необходимость учета вторых моментов распределений временных параметров трафика и обслуживания в узлах, что не может быть сделано методами теории экспоненциальных сетей или другими

приближенными методами с такой точностью и за приемлемое время. Графики построены по данным таблицы 5.1.

В таблице 5.2 приведены результаты расчетов значений вероятности потери сообщений $P_{отк}$ и среднего времени задержки W в зависимости от объема буфера m и интенсивности входного потока λ (время обслуживания нормированное). При этом варьировалось значение коэффициента вариации времени обслуживания C_{μ} .

а) $C = 1$

б) $C = 1$

Продолжение таблицы

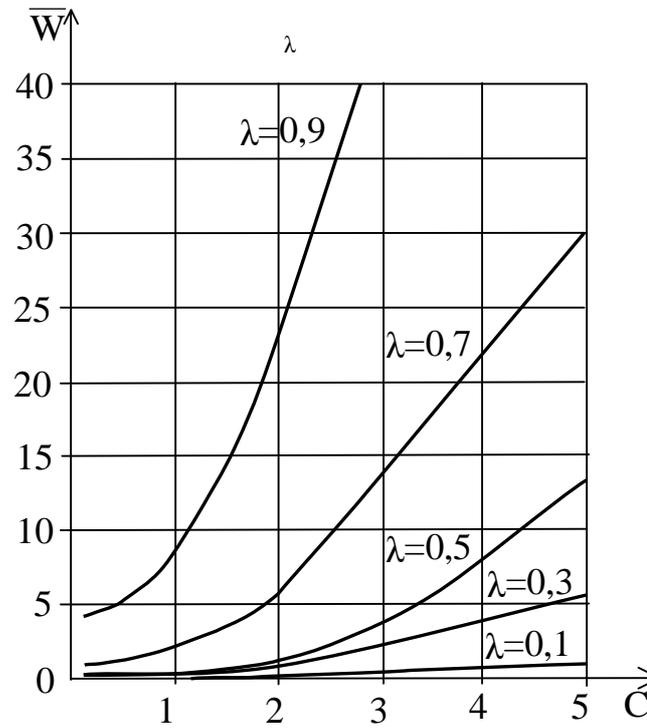
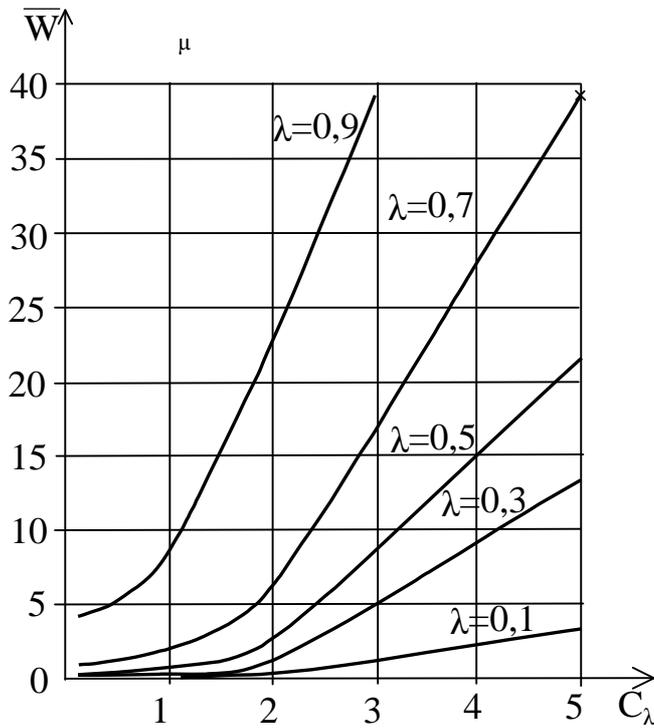


Рисунок 5.1 – Зависимость времени задержки в узле при различных значениях интенсивности λ входного трафика (время обслуживания нормированное):

- а) – от коэффициента вариации входного потока;
- б) – от коэффициента вариации времени обслуживания

Таблица 5.2 -Зависимость вероятности потери сообщений $P_{отк}$ и среднего времени задержки \bar{W} от объема буфера m , интенсивности входного потока λ и коэффициента вариации C_μ

$P_{отк}/\bar{W}$
C_μ

m	ρ	0,1	1	2
2	0,1	0,00004 / 0,1121	0,00254 / 0,1493	0,00924 / 0,2547
	0,3	0,04267 / 0,1245	0,05900 / 0,3639	0,06129 / 0,9891
	0,5	0,18242 / 0,4518	0,15234 / 0,9029	0,11910 / 2,1570
	0,7	0,33984 / 1,5765	0,24822 / 1,8735	0,17279 / 4,0395
	0,9	0,47627 / 2,0861	0,00545 / 3,7100	0,22158 / 7,3703
3	0,1	0,00000 / 0,1120	0,00017 / 0,1488	0,00135 / 0,2527
	0,3	0,00645 / 0,1152	0,01347 / 0,3510	0,01748 / 1,0098
	0,5	0,05170 / 0,3609	0,04824 / 0,8837	0,04074 / 2,3162
	0,7	0,12375 / 1,0511	0,09179 / 1,9791	0,06477 / 4,9237
	0,9	0,19666 / 2,8062	0,13551 / 4,6195	0,08791 / 12,2135

5.2

		$P_{отк}/W$		
		C_{μ}		
m	ρ	0,1	1	2
5	0,1	0,00000 / 0,1120	0,00000 / 0,1487	0,00007 / 0,2524
	0,3	0,00037 / 0,1138	0,00181 / 0,3484	0,00359 / 1,0141
	0,5	0,00988 / 0,3437	0,01186 / 0,8875	0,01169 / 2,356
	0,7	0,03752 / 1,0601	0,02964 / 2,1440	0,02179 / 5,4454
	0,9	0,07430 / 3,3815	0,05075 / 6,4783	0,03249 / 17,1533
8	0,1	0,00000 / 0,1120	0,00000 / 0,1487	0,00000 / 0,2524
	0,3	0,00000 / 0,1137	0,00019 / 0,3481	0,00069 / 1,0151
	0,5	0,00160 / 0,3401	0,00293 / 0,8895	0,00359 / 2,4178
	0,7	0,01158 / 1,0765	0,01056 / 2,2231	0,00826 / 5,6368
	0,9	0,03083 / 3,9521	0,02178 / 7,7246	0,01386 / 19,8232
10	0,1	0,00000 / 0,1120	0,00000 / 0,1487	0,00000 / 0,2524
	0,3	0,00000 / 0,1137	0,00005 / 0,3481	0,00029 / 1,0152
	0,5	0,00056 / 0,3396	0,00141 / 0,8899	0,00198 / 2,4223
	0,7	0,00620 / 1,0796	0,00636 / 2,2391	0,00519 / 5,6818
	0,9	0,01999 / 4,1763	0,01466 / 8,1217	0,00933 / 20,5996
12	0,1	0,00000 / 0,1120	0,00000 / 0,1487	0,00000 / 0,2524
	0,3	0,00000 / 0,1137	0,00000 / 0,3481	0,00013 / 1,0153
	0,5	0,00022 / 0,3395	0,00074 / 0,8901	0,00119 / 2,4246

Продолжение таблицы

	0,7	0,00361 / 1,0812	0,00415 / 2,2478	0,00353 / 5,7065
	0,9	0,01409 / 4,3162	0,01062 / 8,3691	0,00677 / 21,0661
14	0,1	0,00000 / 0,1120	0,00000 / 0,1487	0,00000 / 0,2524
	0,3	0,00000 / 0,1137	0,00000 / 0,3481	0,00000 / 1,0153
	0,5	0,00009 / 0,3394	0,00041 / 0,8902	0,00076 / 2,4258
	0,7	0,00229 / 1,0821	0,00286 / 2,2529	0,00253 / 5,7214
	0,9	0,01078 / 4,4014	0,00808 / 8,5329	0,00516 / 21,3688
16	0,1	0,00000 / 0,1120	0,00000 / 0,1487	0,00000 / 0,2524
	0,3	0,00000 / 0,1137	0,00000 / 0,3481	0,0000 / 1,0153
	0,5	0,00000 / 0,3394	0,00024 / 0,8902	0,00051 / 2,4265
	0,7	0,00162 / 10825	0,00205 / 2,2561	0,00189 / 5,7310
	0,9	0,00888 / 4,4525	0,00640 / 8,6460	0,00408 / 21,5766

На рисунках 5.2-5.5 приведены графики зависимостей вероятности потери сообщения $P_{отк}$ в каналах приема-передачи от интенсивности λ входного потока (при интенсивности обслуживания $\mu=1$) и от объема буферной памяти m , выраженного в единицах от сообщений. Графики построены для диапазона изменения коэффициента вариации времени обслуживания заявок C_{μ} от 0,1 до 2,0 при коэффициенте вариации распределения времен поступления $C_{\lambda} = 1$.

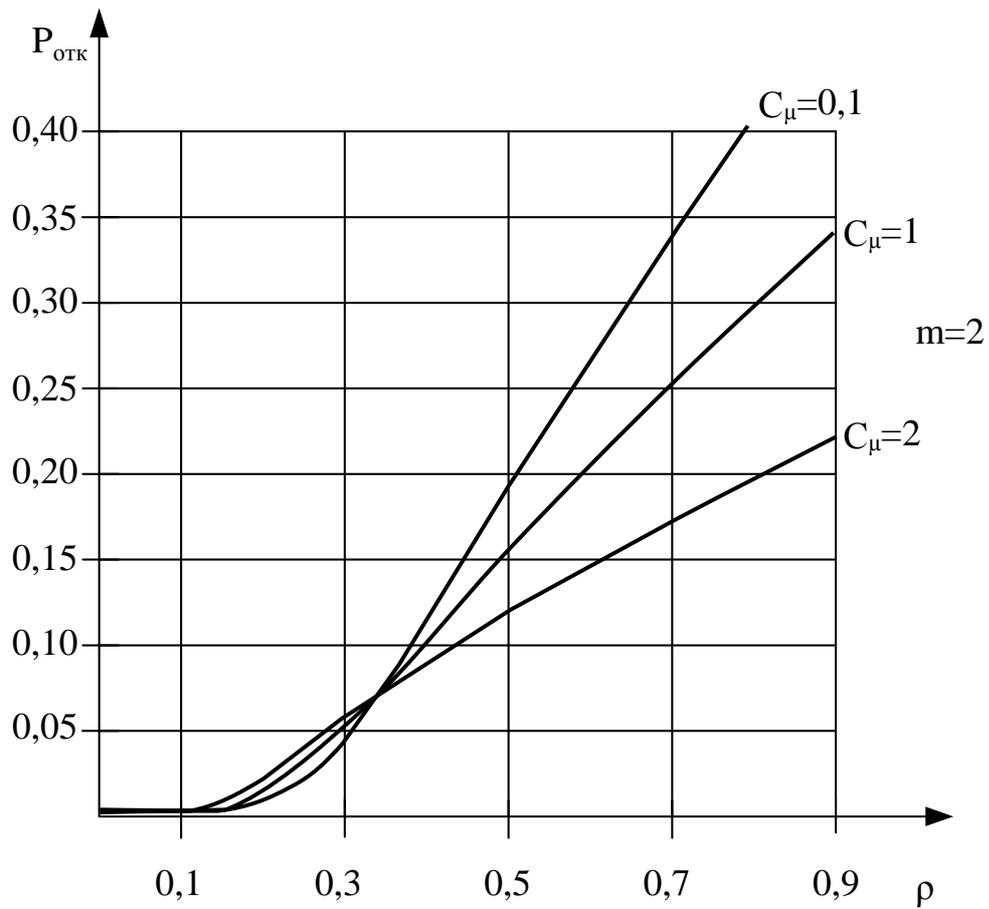


Рисунок 5.2 –Графики зависимости вероятности потери сообщений от загрузки ρ при $m=2$

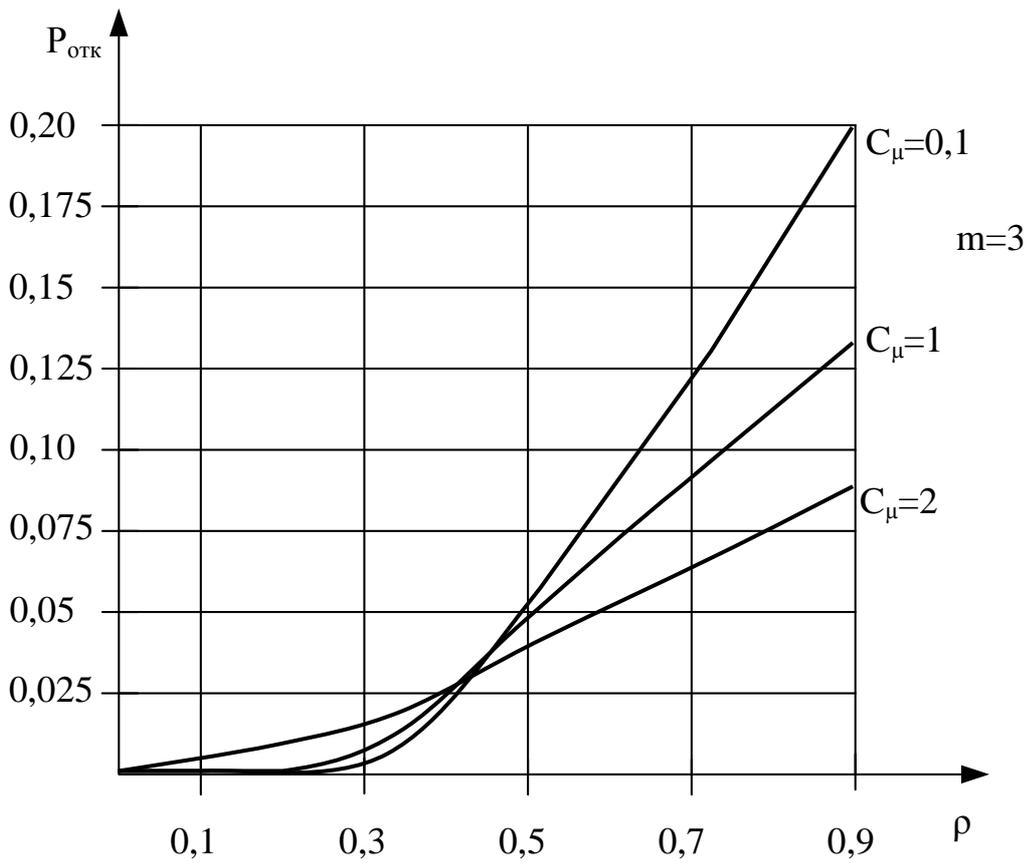


Рисунок 5.3 – Графики зависимости вероятности потери сообщений от загрузки ρ при $m=3$

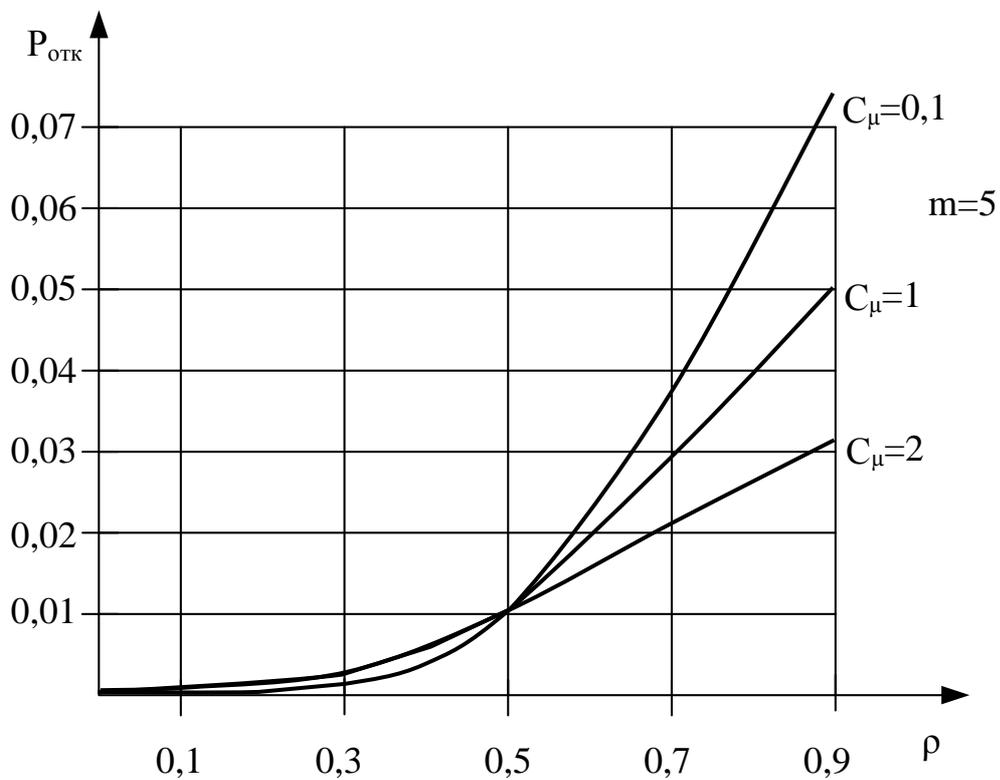


Рисунок 5.4 – Графики зависимости вероятности потери сообщений от загрузки ρ при $m=5$

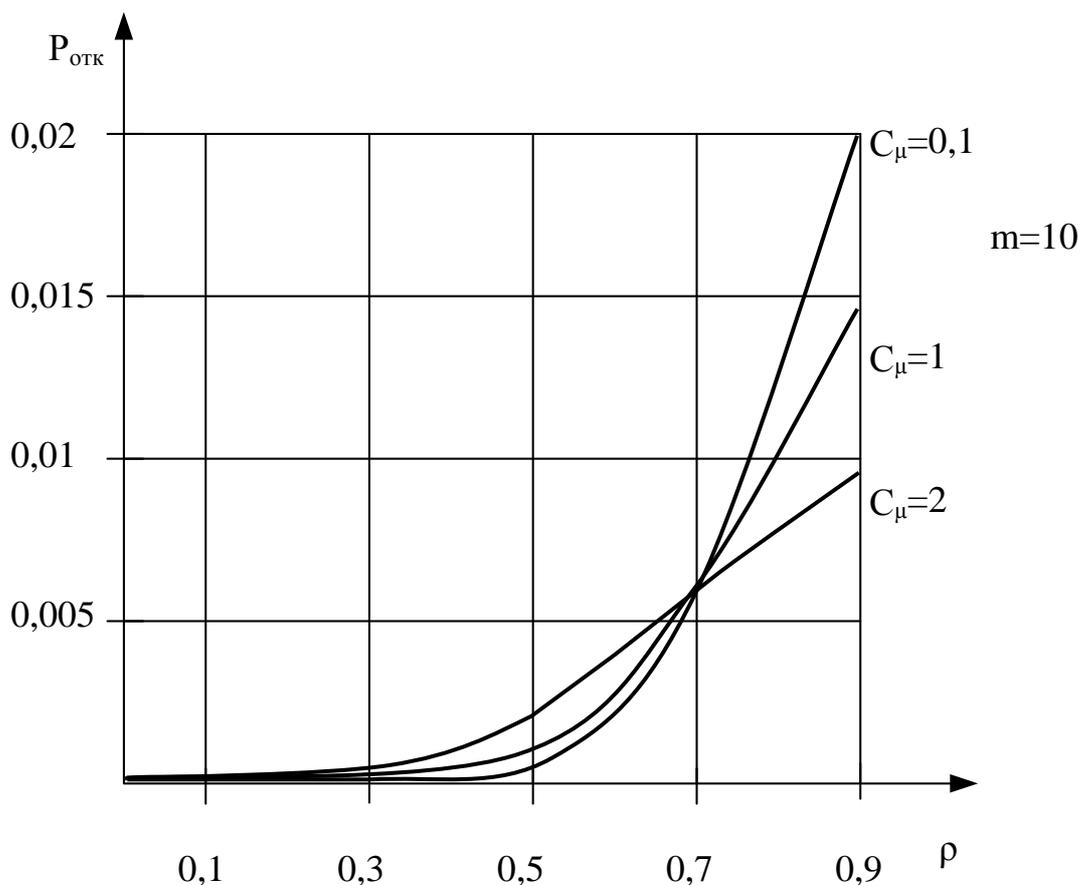


Рисунок 5.5 – Графики зависимости вероятности потери сообщений от загрузки ρ при $m=10$

Аналогичные графики изменения среднего времени задержки сообщения в очереди \bar{W} в узле коммутации в зависимости от тех же параметров, показаны на рисунках 5.6-5.9. Указанные зависимости получены по методике п. 3.1 анализа моделей СМО с ограничениями на длину очереди с потерями при следующих допущениях:

- 1) сеть находится в состоянии статистического равновесия;
- 2) потоки сообщений независимы, т.е. сообщения, переданные по любой линии сети, статистически не зависят от сообщений, появляющихся в других узлах сети.

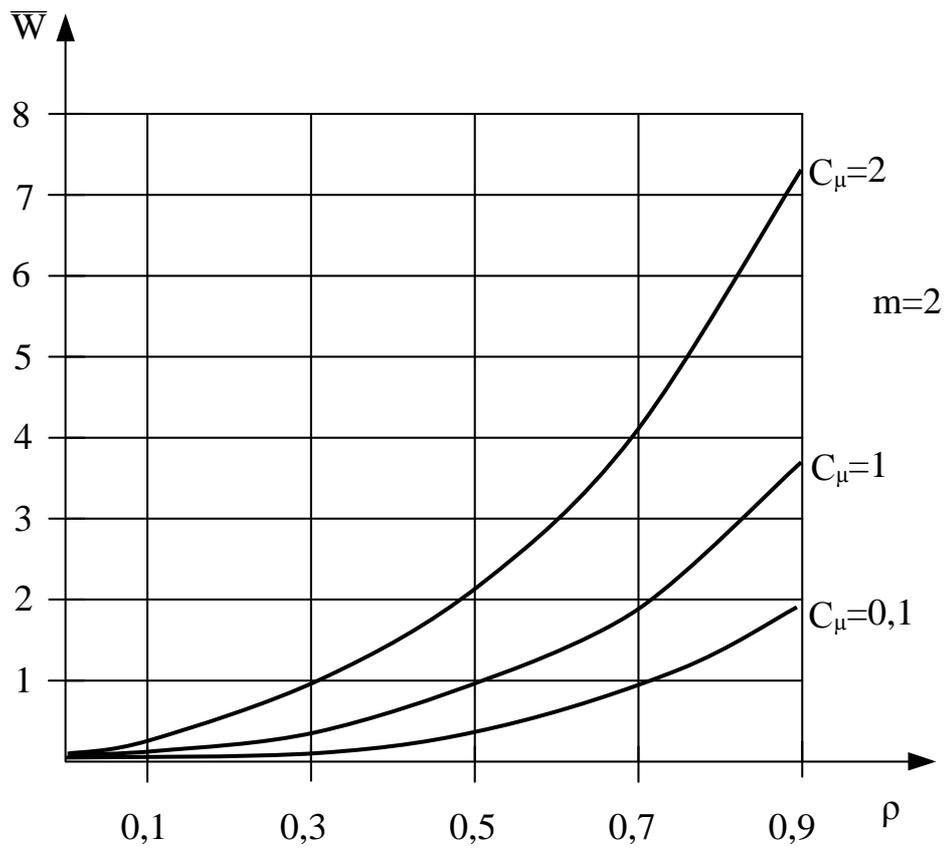
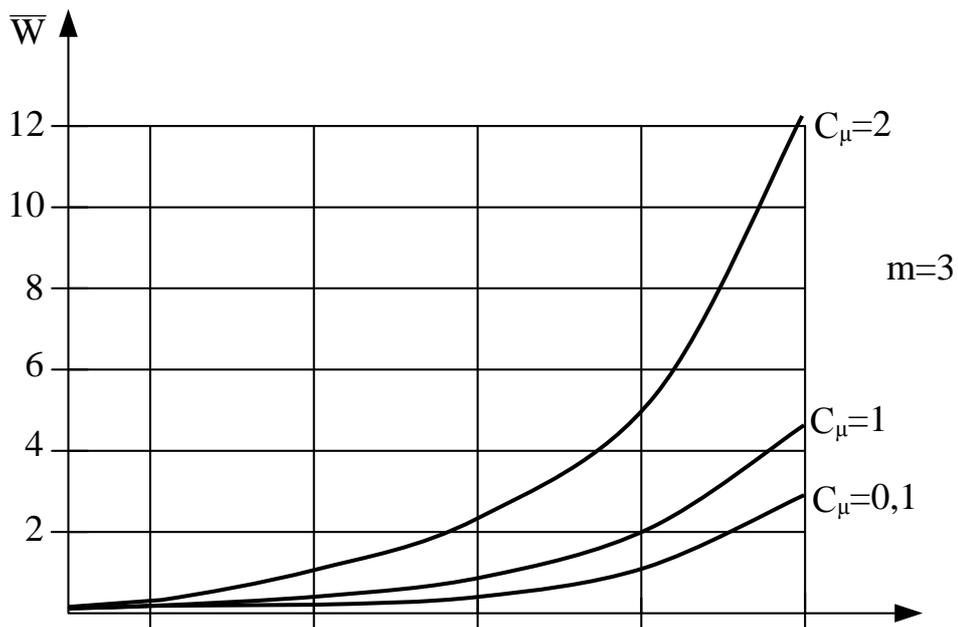
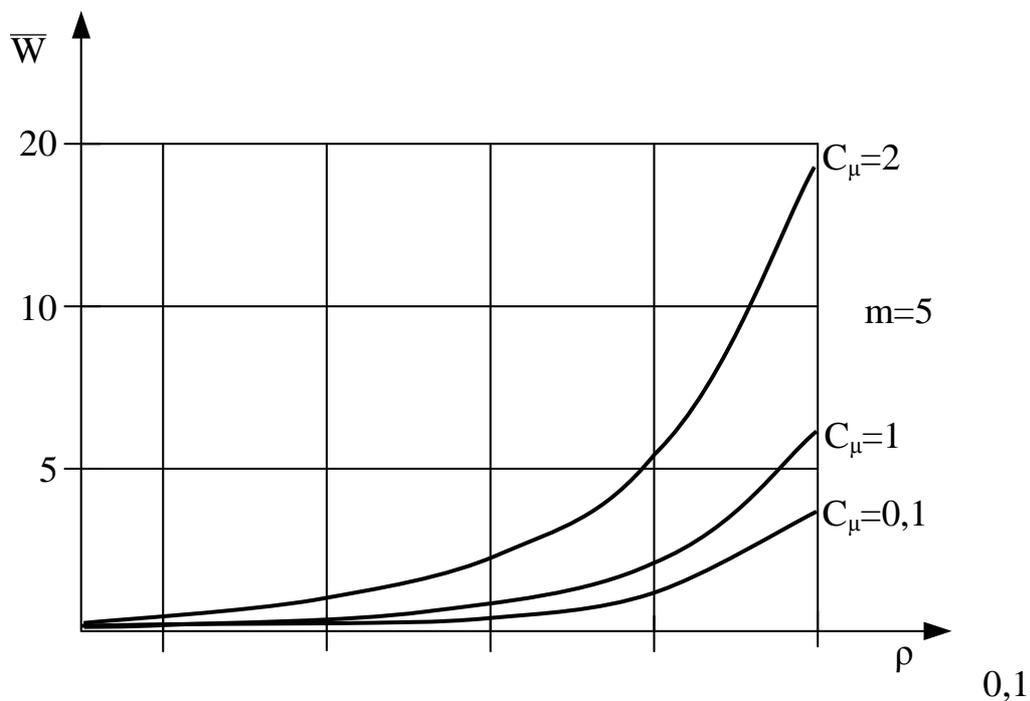


Рисунок 5.6 – Графики зависимости среднего времени ожидания от загрузки ρ при $m=2$



0,1 0,3 0,5 0,7 0,9 ρ

Рисунок 5.7 – Графики зависимости среднего времени ожидания от загрузки ρ при $m=3$



0,3 0,5 0,7 0,9

Рисунок 5.8 – Графики зависимости среднего времени ожидания от загрузки ρ при $m=5$

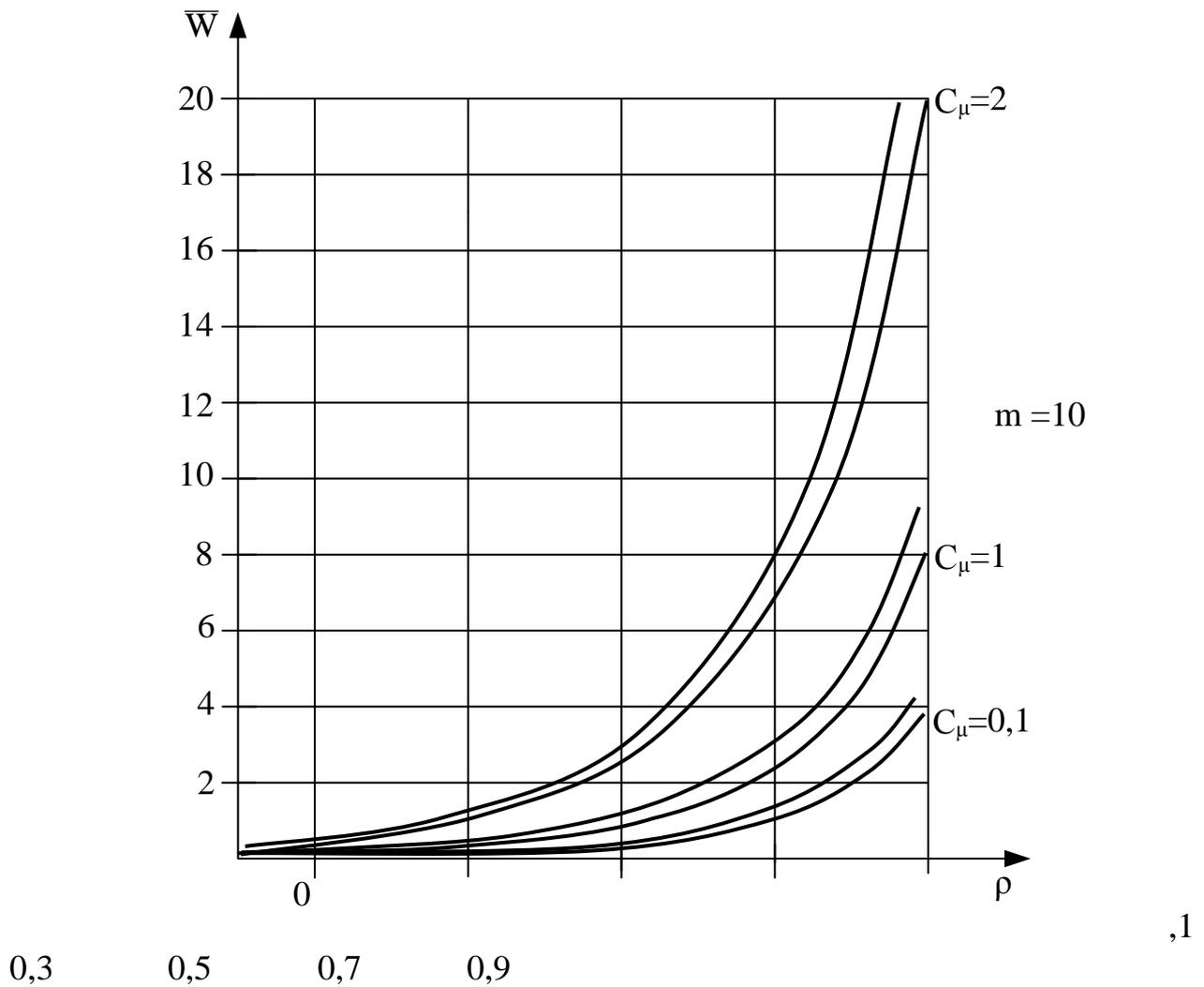


Рисунок 5.9 – Графики зависимости среднего времени ожидания от загрузки ρ при $m=10$

Графики, приведенные на рисунках 5.6-5.9, позволяют рассчитать необходимые объемы памяти для буферных накопителей при ограничениях на вероятность потери и на время задержки сообщения в узле коммутации. При этом значения объемов, выраженные в единицах от сообщений, можно пересчитать в единицы от бит умножением значений объемов на среднюю длину сообщения.

Из вышеприведенных графиков видно, что при увеличении объема буфера среднее время ожидания \bar{W} стремится к времени ожидания СМО в случае с бесконечной очередью (пунктирная линия на рисунке 5.9).

6 Задание на лабораторную работу № 5. Расчет характеристик ВС с неоднородными потоками

Рассмотрим модель вычислительной системы с неоднородными потоками специализированной информационно-справочной службы. Сетевая модель данной системы приведена на рисунке 6.1.

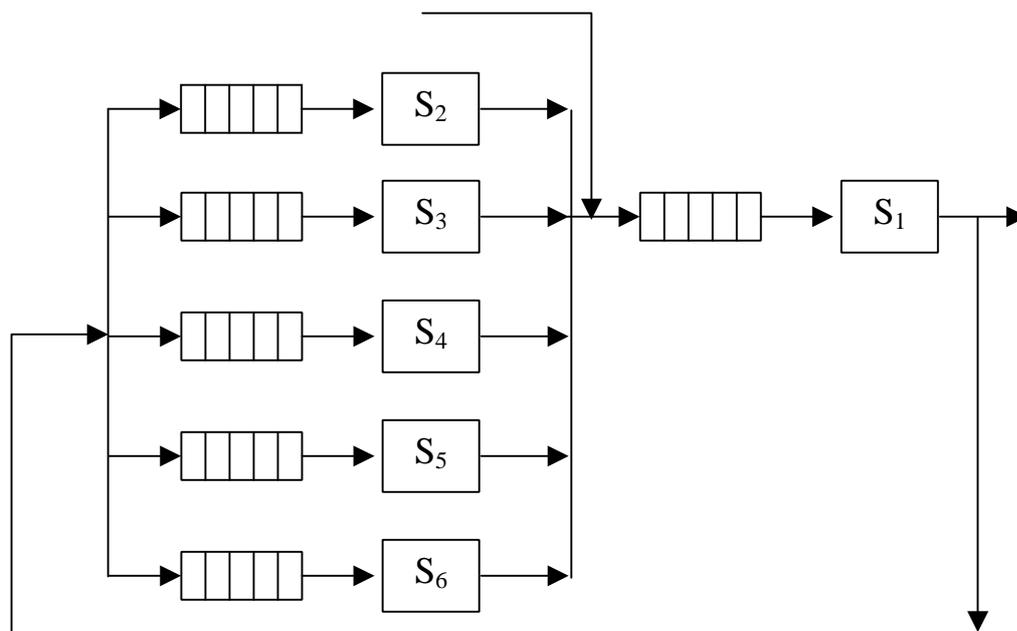


Рисунок 6.1 - Сетевая модель ВС информационно-справочной службы.

Параметры системы указаны в таблице 6.1

Таблица 6.1

№ узла	Класс устройств	Характеристики устройств
1	Процессор 2 (1)	Быстродействие 250 у.е. оп/с (120 у.е. оп/с)
2-4	НМД1-НМД3	Среднее время поиска информации 50 мс
5	НМЛ	Среднее время поиска информации 5с
6	Принтер	Скорость печати 20 строк/с

Вычислительную нагрузку систем образуют 3 класса задач.

1 Обработка запросов, включающая обращения к накопителям на магнитных дисках (НМД1-НМД3) и процессорную обработку. Трудоемкость обработки в процессоре между двумя обращениями к НМД в среднем составляет 4 тыс. операций.

2 Создание итоговых ведомостей (файлов), переносящая всю информацию с НМДЗ на НМЛ. Трудоемкость процессорной обработки составляет 1 тыс. операций.

3 Печать итоговых ведомостей с НМЛ на принтер. Трудоемкость обработки в процессоре составляет 5 тыс. операций.

Интенсивности поступления задач в систему связаны между собой. Задачи класса I поступают в систему с терминалов в случайные моменты времени. Процесс их поступления моделируется простейшим потоком. Задачи класса 2 инициируются в системе при заполнении НМДЗ. Это происходит 1 раз на 1000 задач класса 1. Таким образом, поток задач класса 2 имеет распределение Эрланга 1000 порядка, коэффициент вариации которого близок к нулю. Задачи класса 3 поступают в среднем 7 раз на 1000 задач типа 1, и коэффициент вариации распределения потока задач данного типа приблизительно равен 0,1.

Задание.

1 Предварительно для каждого класса заявок рассчитать трудоемкости их обслуживания в отдельных узлах ВС.

2 Из условий функционирования ВС рассчитать матрицы вероятностей передач заявок для каждого класса.

3 Варьирование совместно интенсивности поступления (с сохранением пропорции между ними) в систему задач всех классов получить зависимости времени пребывания U^c задач класса 1 от интенсивности их поступления λ_1 для двух типов процессоров.

4 Используя полученные зависимости определить оптимальное количество терминалов, которые может обслуживать данная ВС при ограничении времени ответа, равного 2 с.

Результаты проведенных расчетов приведены ниже.

В таблице 6.2 приведены значения параметров трудоемкости заявок по каждому классу.

Таблица 6.2

№ типа заявки	Наименование типа заявки	Интенсивность входного потока	Параметры трудоемкости заявок			
			№ узла	Среднее ко-во обращений	Средняя трудоемкость обращения	Коэффициент вариации

1	Запрос Б	1.000	1	6.000	4.000	1.000
			2	1.000	0.050	1.000
			3	2.000	0.050	1.000
2	Сброс МД	0.100E-02	4	1.000	0.050	1.000
			1	1008.000	1.000	1.000
			4	1000.000	0.050	1.000
3	Печ.вед.	0.100E-01	5	7.000	5.000	1.000
			1	4.000	0.500	1.000
			6	1.000	100.000	1.000
			6	1.000	5.000	1.000

В таблице 6.3 приведены расчетные значения коэффициентов загрузки узлов модели при интенсивности входного потока задач класса 1 равной 5,5 ед/с.

Таблица 6.3

№ узла	Коэффициент загрузки ρ
	Процессор 2
1	0,447
2	0,280
3	0,550
4	0,370
5	0,232
6	0,270

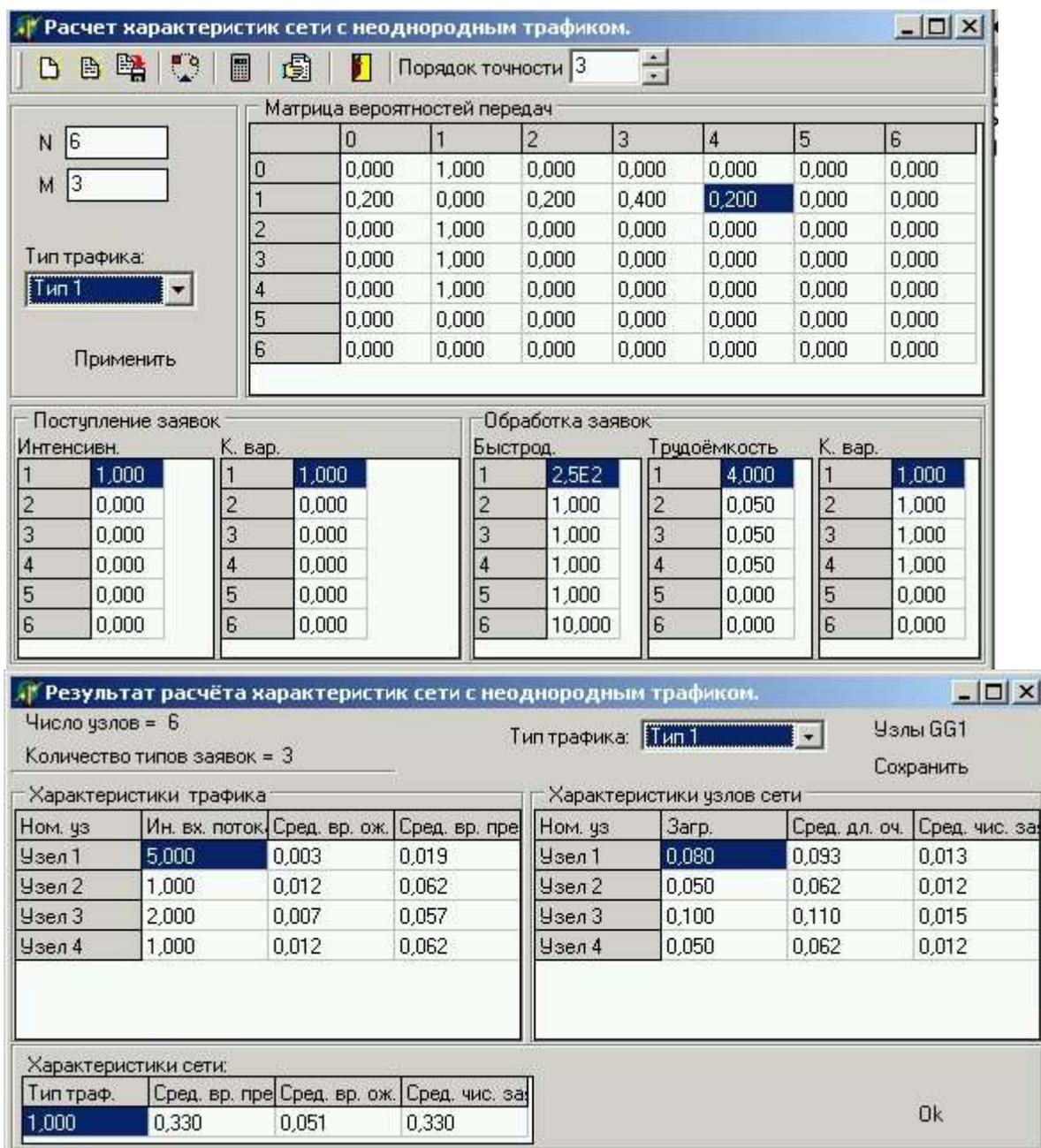


Рисунок 6.2- Входные данные и результаты расчетов для заявок класса 1

Расчет характеристик сети с неоднородным трафиком.

Порядок точности 3

N 6
M 3
Тип трафика: Тип 2
Применить

Матрица вероятностей передач

	0	1	2	3	4	5	6
0	0,000	1,000	0,000	0,000	0,000	0,000	0,000
1	0,003	0,000	0,000	0,000	0,987	0,010	0,000
2	0,000	0,000	0,000	0,000	0,000	0,000	0,000
3	0,000	0,000	0,000	0,000	0,000	0,000	0,000
4	0,000	1,000	0,000	0,000	0,000	0,000	0,000
5	0,000	1,000	0,000	0,000	0,000	0,000	0,000
6	0,000	0,000	0,000	0,000	0,000	0,000	0,000

Поступление заявок

Интенсивн.	К. вар.
1 0,001	1 0,001
2 0,000	2 0,000
3 0,000	3 0,000
4 0,000	4 0,000
5 0,000	5 0,000
6 0,000	6 0,000

Обработка заявок

Быстрод.	Трудоёмкость	К. вар.
1 2,5E2	1 1,000	1 1,000
2 1,000	2 0,000	2 0,000
3 1,000	3 0,000	3 0,000
4 1,000	4 0,050	4 1,000
5 1,000	5 5,000	5 1,000
6 10,000	6 0,000	6 0,000

Результат расчёта характеристик сети с неоднородным трафиком.

Число узлов = 6
Количество типов заявок = 3
Тип трафика: Тип 2
Узлы GG1
Сохранить

Характеристики трафика

Ном. уз	Ин. вх. поток	Сред. вр. ож.	Сред. вр. пре
Узел 1	0,330	0,001	0,005
Узел 4	0,330	0,018	0,068
Узел 5	0,003	1,900	6,900

Характеристики узлов сети

Ном. уз	Загр.	Сред. дл. оч.	Сред. чис. за
Узел 1	0,001	0,002	0,000
Узел 4	0,016	0,022	0,006
Узел 5	0,017	0,023	0,006

Характеристики сети:

Тип траф.	Сред. вр. пре	Сред. вр. ож.	Сред. чис. за
2,000	47,000	13,000	0,047

Ok

Рисунок 6.3 - Входные данные и результаты расчетов для заявок класса 2

Расчет характеристик сети с неоднородным трафиком.

Порядок точности 3

N 6
M 3
Тип трафика: Тип 3
Применить

Матрица вероятностей передач

	0	1	2	3	4	5	6
0	0,000	1,000	0,000	0,000	0,000	0,000	0,000
1	0,330	0,000	0,000	0,000	0,000	0,330	0,330
2	0,000	0,000	0,000	0,000	0,000	0,000	0,000
3	0,000	0,000	0,000	0,000	0,000	0,000	0,000
4	0,000	0,000	0,000	0,000	0,000	0,000	0,000
5	1,000	0,000	0,000	0,000	0,000	0,000	0,000
6	0,000	1,000	0,000	0,000	0,000	0,000	0,000

Поступление заявок

Интенсивн.	К. вар.
1 0,010	1 0,100
2 0,000	2 0,000
3 0,000	3 0,000
4 0,000	4 0,000
5 0,000	5 0,000
6 0,000	6 0,000

Обработка заявок

Быстрод.	Трудоёмкость	К. вар.
1 2,5E2	1 5,000	1 1,000
2 1,000	2 0,000	2 0,000
3 1,000	3 0,000	3 0,000
4 1,000	4 0,000	4 0,000
5 1,000	5 5,000	5 1,000
6 10,000	6 1E2	6 1,000

Результат расчёта характеристик сети с неоднородным трафиком.

Число узлов = 6
Количество типов заявок = 3
Тип трафика: Тип 3
Узлы GG1
Сохранить

Характеристики трафика

Ном. уз	Ин. вх. поток	Сред. вр. ож.	Сред. вр. пре
Узел 1	0,015	0,001	0,021
Узел 5	0,005	1,800	6,800
Узел 6	0,005	2,500	13,000

Характеристики узлов сети

Ном. уз	Загр.	Сред. дл. оч.	Сред. чис. за
Узел 1	0,000	0,000	0,000
Узел 5	0,025	0,034	0,009
Узел 6	0,049	0,062	0,012

Характеристики сети:

Тип траф.	Сред. вр. пре	Сред. вр. ож.	Сред. чис. за
3,000	9,600	2,100	0,096

Ok

Рисунок 6.4- Входные данные и результаты расчетов для заявок класса 3

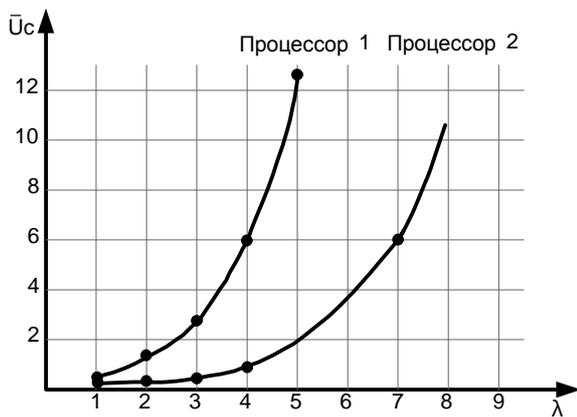


Рисунок 6.5 - Зависимость среднего времени пребывания задач типа 1 в системе от интенсивности их поступления.

Из графиков, приведенных на рисунке 6.5 и таблицы 6.3 видно, что использование процессора 2 позволяет примерно в 2 раза увеличить допустимую при заданном ограничении на время ответа интенсивность поступления запросов. Она составляет 5 ед/с. Отсюда количество терминалов, которое можно подключить к системе при интенсивности запросов от каждого терминала 0,008 ед/с (один запрос за 2 минуты) составляет 600 единиц.

Расчет модели показал, что при использовании процессора 2 все устройства системы загружены равномерно и используются более эффективно, чем при использовании процессора 1.

7 Моделирование вычислительных систем средствами языка GPSS WORLD STUDENT VERSION 4.3.5

7.1 Описание основных блоков языка GPSS

GENERATE - это блок, через который транзакты входят в модель. Не существует ограничений на число различных блоков GENERATE в одной модели. Интервал времени между последовательными появлениями транзактов блока GENERATE называют интервалом поступления.

Формат блока следующий:

GENERATE A,B,C,D,E.

Когда транзакт входит в модель через блок GENERATE (генерировать), интерпретатор планирует время поступления последующего транзакта путем розыгрыша случайного числа в соответствии с распределением интервалов времени поступления с последующим добавлением разыгранного значения к

текущему значению таймера. При достижении этого значения времени следующий транзакт вводится в модель через блок **GENERATE** и т. д.

Информация, необходимая для данного блока, задается операндами А и В. Все возможные виды распределения интервалов времени прибытия в GPSS делят на равномерно распределенные и все другие виды распределения.

Таким образом, специально рассматривают только простейшее из всех нетривиальных распределений, а именно равномерное распределение. Для того, чтобы задать более сложный вид распределения интервалов времени прихода в GPSS необходимо записать так называемое определение функции.

Блок **GENERATE** вместе с операндами А и В изображен на рисунке 7.1.

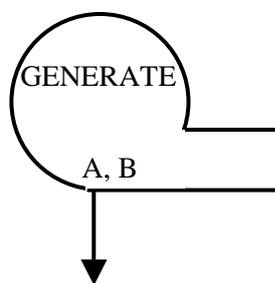


Рисунок 7.1 - Блок **GENERATE**

В поле А задается среднее значение интервала времени между моментами поступления в модель двух последовательных транзактов. Если этот интервал постоянен, то поле В не используется. Если же интервал поступления является случайной величиной, то в поле В указывается модификатор среднего значения, который может быть задан в виде модификатора-интервала или модификатора-функции. Модификатор-интервал используется, когда интервал поступления транзактов является случайной величиной с равномерным законом распределения вероятностей. В этом случае в поле В может быть задан любой стандартный числовой атрибут (СЧА), кроме ссылки на функцию, а диапазон изменения интервала поступления имеет границы $A-B$, $A+B$.

Например, блок

GENERATE 100,40 создает транзакты через случайные интервалы времени, равномерно распределенные на отрезке [60;140].

Модификатор-функция используется, если закон распределения интервала поступления отличен от равномерного. В этом случае в поле В должна быть записана ссылка на функцию, описывающую этот закон, и

случайный интервал поступления определяется, как целая часть произведения поля A (среднего значения) на вычисленное значение функции.

В поле C задается момент поступления в модель первого транзакта. Если это поле пусто или равно 0, то момент появления первого транзакта определяется операндами A и B.

Поле D задает общее число транзактов, которое должно быть создано блоком **GENERATE**. Если это поле пусто, то блок генерирует неограниченное число транзактов до завершения моделирования.

В поле E задается приоритет, присваиваемый генерируемым транзактам. Число уровней приоритетов не ограничено, причем самый низкий приоритет - нулевой. Если поле E пусто, то генерируемые транзакты имеют нулевой приоритет.

Транзакты удаляются из модели, попадая в блок **TERMINATE** (завершить). Блоки **TERMINATE** всегда позволяют выйти всем транзактам, которые пытаются это сделать. В модели может быть любое число блоков **TERMINATE**.

На рисунке 7.2 изображен блок **TERMINATE** с операндом A. Операнд A является указателем уменьшения счетчика завершений. Он задает величину, которая должна вычитаться из специального счетчика, называемого счетчиком завершений, каждый раз, когда транзакт входит в блок **TERMINATE**. Если разработчик не желает задавать операнд A блока **TERMINATE**, то по умолчанию подразумевается значение 0. Вход транзакта в такой блок **TERMINATE** не вызывает уменьшения содержания счетчика завершений.

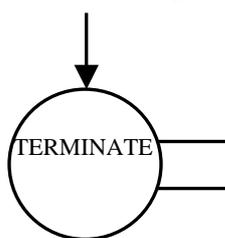


Рисунок 7.2 – Блок **TERMINATE**

Для моделирования ограниченных ресурсов СМО в модели должны присутствовать аппаратные объекты: одноканальные или многоканальные устройства. Одноканальные устройства создаются в текущей модели при использовании блоков **SEIZE** (занять) и **RELEASE** (освободить), имеющих следующий формат:

```
SEIZE  A
RELEASE A.
```

В поле A указывается номер или имя устройства. Если транзакт входит в блок **SEIZE**, то устройство, указанное в поле A, становится занятым и остаётся в этом состоянии до тех пор, пока этот же транзакт не пройдёт соответствующий блок **RELEASE**, освобождая устройство. Если устройство, указанное в поле A блока **SEIZE**, уже занято каким-либо транзактом, то никакой другой транзакт не может войти в этот блок и остаётся в предыдущем блоке. Транзакты, задержанные (заблокированные) перед блоком **SEIZE**, остаются в списке текущих событий и при освобождении устройства обрабатываются с учетом приоритетов и очередности поступления.

Блок **SEIZE** показан на рисунке 7.3.

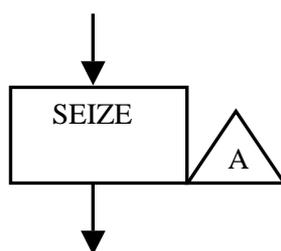


Рисунок 7.3 - Блок SEIZE

Блок **RELEASE** показан на рисунке 7.4.

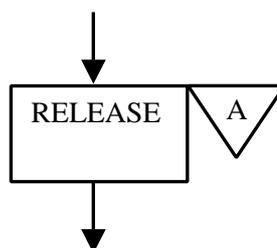


Рисунок 7.4 - Блок RELEASE

В моделях систем массового обслуживания заявки обслуживаются приборами (каналами) СМО в течение некоторого промежутка времени, прежде чем покинуть СМО. Для моделирования такого обслуживания, т.е. для задержки транзактов на определенный отрезок модельного времени, служит блок **ADVANCE** (задержать), имеющий следующий формат:

ADVANCE A,B.

Операнды в полях А и В имеют тот же смысл, что и в соответствующих полях блока **GENERATE**. Транзакты, входящие в блок **ADVANCE**, переводятся из списка текущих событий в список будущих событий, а по истечении вычисленного времени задержки возвращаются назад, в список текущих событий, и их продвижение по блок-схеме продолжается. Если вычисленное время задержки равно 0, то транзакт в тот же момент модельного времени переходит в следующий блок, оставаясь в списке текущих событий. Блок **ADVANCE** показан на рисунке 7.5.

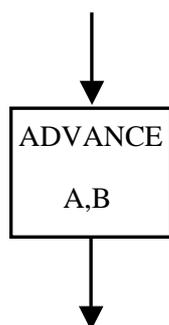


Рисунок 7.5 - Блок ADVANCE

Модели одноканальной и многоканальной СМО с ожиданием разрабатываются обычно для исследования различных характеристик, связанных с ожиданием заявок в очереди: длины очереди, времени ожидания и т.п., а очередь транзактов образуется в списке текущих событий и недоступна исследователю. Для регистрации статистической информации о процессе ожидания транзактов в модели должны присутствовать статистические объекты: очереди или таблицы.

Объекты типа «очередь» создаются в модели путем использования блоков - регистраторов очередей: **QUEUE** (стать в очередь) и **DEPART** (уйти из очереди), имеющих следующий формат:

QUEUE A,B
DEPART A,B.

В поле А указывается номер или имя очереди, а в поле В – число единиц, на которое текущая длина очереди увеличивается при входе транзакта в блок **QUEUE** или уменьшается при входе транзакта в блок **DEPART**. Обычно поле В пусто, и в этом случае его значение по умолчанию принимается равным 1. Для сбора статистики о транзактах, заблокированных перед каким-либо блоком модели, блоки **QUEUE** и **DEPART** помещаются перед и после этого блока соответственно. При прохождении транзактов через блоки **QUEUE** и

DEPART соответствующим образом изменяются следующие стандартные числовые атрибуты очередей: Q - текущая длина очереди; QM - максимальная длина очереди; QA - целая часть средней длины очереди; QC - общее число транзактов, вошедших в очередь; QZ - число транзактов, прошедших через очередь без ожидания (число "нулевых" входов); QT - целая часть среднего времени ожидания с учетом "нулевых" входов; QX - целая часть среднего времени ожидания без учета "нулевых" входов.

Блоки **QUEUE** и **DEPART** показаны на рисунке 7.6.

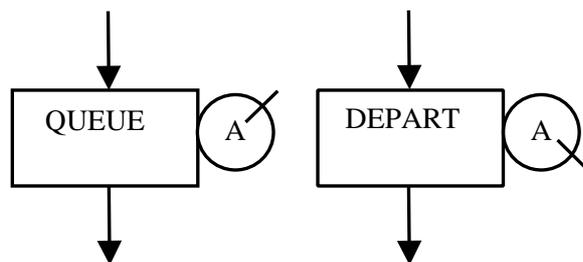


Рисунок 7.6 - Блоки **QUEUE** и **DEPART**

Транзакты, выходящие из любого блока, всегда поступают в следующий блок. В более сложных моделях возникает необходимость направления транзактов к другим блокам в зависимости от некоторых условий. Эту возможность обеспечивают блоки изменения маршрутов транзактов.

Блок **TRANSFER** (передать) служит для передачи входящих в него транзактов в блоки, отличные от следующего. Блок имеет девять режимов работы, из которых рассмотрим здесь лишь три наиболее часто используемых.

В этих трех режимах блок имеет следующий формат:

TRANSFER A,B,C.

Смысл операндов в полях A, B и C зависит от режима работы блока. В режиме безусловной передачи поля A и C пусты, а в поле B указывается имя блока, к которому безусловным образом направляется транзакт, вошедший в блок **TRANSFER**. Например:

TRANSFER , FINAL.

В режиме статистической передачи операнд A определяет вероятность, с которой транзакт направляется в блок, указанный в поле C. С вероятностью 1-A транзакт направляется в блок, указанный в поле B (в следующий, если поле B пусто). Вероятность в поле A может быть задана непосредственно

десятичной дробью, начинающейся с точки. Например, блок

TRANSFER .75,THIS,THAT

с вероятностью 0,75 направляет транзакты в блок с именем **THAT**, а с вероятностью 0,25 - в блок с именем **THIS**.

Если же поле А начинается не с десятичной точки и не содержит одного из ключевых слов - признаков других режимов работы блока, то его значение рассматривается как количество тысячных долей в вероятности передачи. Например, предыдущий блок **TRANSFER** можно записать также в следующем виде:

TRANSFER 750,THIS,THAT.

В режиме логической передачи в поле А записывается ключевое слово **BOTH** (оба). Транзакт, поступающий в блок **TRANSFER**, сначала пытается войти в блок, указанный в поле В (или в следующий блок, если поле В пусто), а если это не удастся, т.е. блок В отказывает транзакту во входе, то в блок, указанный в поле С. Если и эта попытка неудачна, то транзакт задерживается в блоке **TRANSFER** до изменения условий в модели, делающего возможным вход в один из блоков В или С, причем при одновременно возникшей возможности предпочтение отдается блоку В. Например:

TRANSFER BOTH,MET1,MET2.

Блок **TRANSFER** показан на рисунке 7.7.

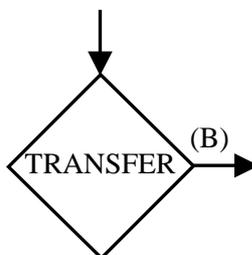


Рисунок 7.7 – Блок TRANSFER

Для занятия

и освобождения каналов обслуживания многоканальных устройств (МКУ) используется пара блоков **ENTER** (войти) и **LEAVE** (покинуть), имеющих следующий формат:

ENTER A,B

LEAVE A,B.

В поле А указывается номер или имя **МКУ**, в поле В число каналов **МКУ**, занимаемых при входе в блок **ENTER** или освобождаемых при входе в блок **LEAVE**. Обычно поле В пусто, и в этом случае по умолчанию занимает или освобождается один канал.

При входе транзакта в блок **ENTER** текущее содержимое **МКУ** увеличивается на число единиц, указанное в поле **B**. Если свободная емкость **МКУ** меньше значения поля **B**, то транзакт не может войти в блок **ENTER** и остается в предыдущем блоке, образуя очередь в списке текущих событий.

При входе транзакта в блок **LEAVE** текущее содержимое **МКУ** уменьшается на число единиц, указанное в поле **B**. Не обязательно освобождается такое же число каналов **МКУ**, какое занималось при входе данного транзакта в блок **ENTER**, однако текущее содержимое **МКУ** не должно становиться отрицательным.

Блоки **ENTER** и **LEAVE** показаны на рисунке 7.8.

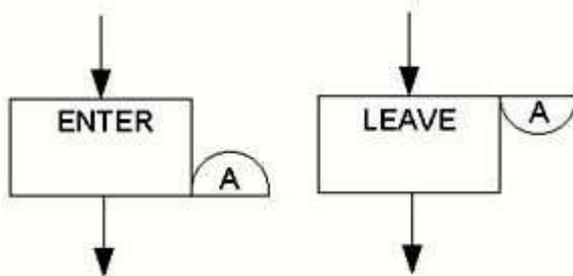


Рисунок 7.8 - Блоки **ENTER** и **LEAVE**

Для присваивания параметрам начальных значений или изменения этих значений служит блок **ASSIGN** (присваивать), имеющий следующий формат: имя **ASSIGN A,B,C**.

В поле **A** указывается номер или имя параметра, в который заносится значение операнда **B**. Если в поле **A** после имени (номера) параметра стоит знак **+** или **-**, то значение операнда **B** добавляется или вычитается из текущего содержимого параметра. В поле **C** может быть указано имя или номер функции-модификатора, действующей аналогично функции-модификатору в поле

В блока **GENERATE**. Например, блок

ASSIGN 5,0

записывает в параметр с номером **5** значение **0**, а блок

ASSIGN COUNT+,1

добавляет **1** к текущему значению параметра с именем **COUNT**. Блок **ASSIGN** приведен на рисунке 7.9.

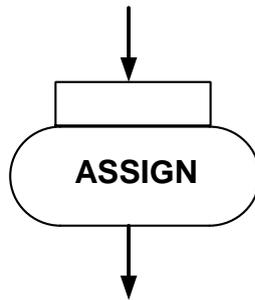


Рисунок 7.9- Блок

ASSIGN

Для записи текущего модельного времени в заданный параметр транзакта служит блок **MARK** (отметить), имеющий следующий формат: имя **MARK A**.

В поле А указывается номер или имя параметра транзакта, в который заносится текущее модельное время при входе этого транзакта в блок **MARK**. Содержимое этого параметра может быть позднее использовано для определения транзитного времени стандартного числового атрибута пребывания транзакта в какой-то части модели с помощью СЧА с названием **MP**.

Например, если на входе участка модели поместить блок **MARK MARKER**, то на выходе этого участка СЧА **MP\$MARKER** будет содержать разность между текущим модельным временем и временем, занесенным в параметр **MARKER** блоком **MARK**.

Если поле А блока **MARK** пусто, то текущее время заносится на место отметки времени входа транзакта в модель, используемой при определении резидентного времени транзакта с помощью СЧА **M1**. Блок **MARK** показан на рисунке 7.10.

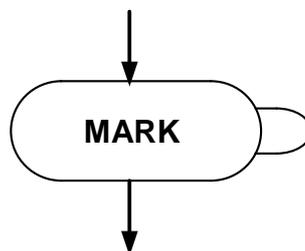


Рисунок 7.10- Блок MARK

Для моделирования захвата (прерывания) одноканального устройства вместо блоков **SEIZE** и **RELEASE** используются соответственно блоки

PREEMPT (захватить) и **RETURN** (вернуть). Блок **PREEMPT** имеет следующий формат:

PREEMPT A,B,C,D,E.

В поле А указывается имя или номер устройства, подлежащего захвату. В поле В кодируется условие захвата. Если это поле пусто, то захват возникает, если обслуживаемый транзакт сам не является захватчиком. Если же в поле В записан операнд PR, то захват возникает, если приоритет транзактазахватчика выше, чем приоритет обслуживаемого транзакта.

Поля С, D и E определяют поведение транзактов, обслуживание которых было прервано. Поле С указывает имя блока, в который будет направлен прерванный транзакт. В поле D может быть указан номер или имя параметра прерванного транзакта, в который записывается время, оставшееся этому транзакту до завершения обслуживания на устройстве. При отсутствии операнда в поле E прерванный транзакт сохраняет право на автоматическое восстановление на устройстве по окончании захвата. Если же в поле E указан операнд RE, то транзакт теряет такое право. Блоки **PREEMPT** и **RETURN** показаны на рисунке 7.11.

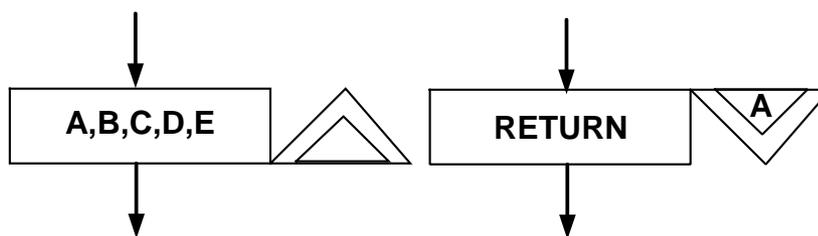


Рисунок 7.11- Блоки **PREEMPT** и **RETURN**

Блок **RETURN** имеет единственный операнд А, содержащий имя или номер устройства, подлежащего освобождению от захвата. Блоки **PREEMPT** и **RETURN** могут быть использованы для моделирования СМО с абсолютными приоритетами. В простейших случаях, при одном уровне захвата, в блоке **PREEMPT** используется единственный операнд А. При этом прерванный транзакт переводится симулятором из списка будущих событий в так называемый список прерываний устройства, а по окончании захвата устройства возвращается в список будущих событий с предварительно вычисленным временем занятия устройства для продолжения обслуживания.

Таблицы в GPSS/PC могут использоваться в общем случае не только для табулирования времени ожидания в очереди, но и для получения выборочных

распределений произвольных СЧА любых объектов модели. Для определения таблиц служит оператор **TABLE** (таблица), формат которого совпадает с форматом оператора **QTABLE**. Отличие состоит лишь в том, что в поле А оператора **TABLE** записывается стандартный числовой атрибут, выборочное распределение которого необходимо получить, а операнды В, С и D определяют разбиение на частотные интервалы диапазона всевозможных значений этого СЧА.

Занесение информации в таблицу, определяемую оператором **TABLE**, уже не может быть выполнено симулятором автоматически, как в случае Qтаблиц. Для этого используется специальный блок **TABULATE** (табулировать), имеющий следующий формат:

TABULATE A.

В поле А указывается номер или имя таблицы, определенной соответствующим оператором **TABLE**.

При входе транзакта в блок **TABULATE** текущее значение табулируемого аргумента таблицы, указанного в поле А оператора **TABLE**, заносится в нее в соответствии с заданным в операторе **TABLE** разбиением области значений аргумента на частотные интервалы. Одновременно корректируются текущие значения СЧА таблицы: счетчик входов в таблицу ТС, среднее время ожидания ТВ и среднеквадратическое отклонение времени ожидания TD. Блок **TABULATE** показан на рисунке 7.12.

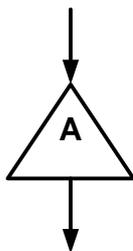


Рисунок 7.12- Блок **TABULATE**

Блок **TEST** (проверить) служит для задержки или изменения маршрутов транзактов в зависимости от соотношения двух СЧА. Он имеет следующий формат:

TEST X A,В,С.

Вспомогательный операнд X содержит условие проверки соотношения между СЧА и может принимать следующие значения: L (меньше); LE (меньше или равно); E (равно); NE (не равно); GE (больше или равно); G (больше). Поле А содержит первый, а поле В - второй из сравниваемых СЧА. Если проверяемое условие А X В выполняется, то блок **TEST** пропускает

транзакт в следующий блок. Если же это условие не выполняется, то транзакт переходит к блоку, указанному в поле С, а если оно пусто, то задерживается перед блоком **TEST**.

Например, блок

TEST LE P\$TIME,C1

не пропускает транзакты, у которых значение параметра с именем **TIME** больше текущего модельного времени. Блок

TEST L Q\$LINE,5,OUT

направляет транзакты в блок с именем **OUT**, если текущая длина очереди **LINE** больше либо равна 5. Блок **TEST** показан на рисунке 7.13.

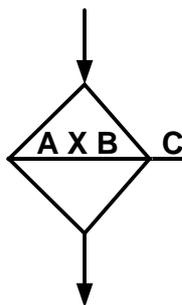


Рисунок 7.13- Блок **TEST**

Для изменения сохраняемых величин в процессе моделирования служит блок **SAVEVALUE** (сохранить величину), имеющий следующий формат:

SAVEVALUE A,B.

В поле **A** указывается номер или имя сохраняемой величины, в которую записывается значение операнда **B**. Если в поле **A** после имени (номера) сохраняемой величины стоит знак **+** или **-**, то значение операнда **B** добавляется или вычитается из текущего содержимого сохраняемой величины. Например:

SAVEVALUE 5,Q\$LINE

SAVEVALUE NREF+,1.

Блок **SAVEVALUE** показан на рисунке 7.14.

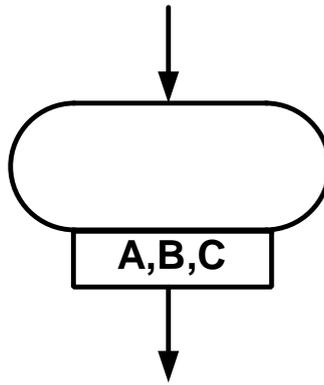


Рисунок 7.14- Блок SAVEVALUE

8 Задание на лабораторную работу №6. Построение модели ВС для определения загрузки устройств и длин очередей к устройствам

Базовая схема для моделирования состоит из терминала пользователя, одного канала передачи данных к ЭВМ и ЭВМ. Заявки на выполнение заданий поступают в интервале $[a,b]$ с равномерным законом распределения. Время обработки заданий равномерно распределено в интервале $[c,d]$. Время обслуживания в канале постоянно и равно пяти. Определить время обработки заявок, загрузку ЭВМ и длину очереди на выполнение заявок.

Порядок выполнения лабораторной работы следующий:

- 1 Написать программу модели.
- 2 Рассчитать загрузку ЭВМ и длину очереди к ЭВМ для заданного варианта задачи.
- 3 Набрать программу в системе моделирования GPSS. Запустить модель.

Получить листинг результатов моделирования.

- 4 Сравнить полученные результаты с теоретическими. **Пример выполнения работы**

Схема модели приведена на рисунке 8.1.

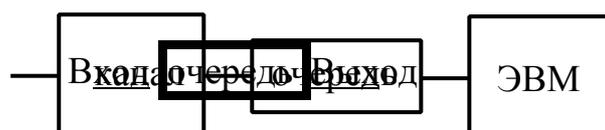


Рисунок 8.1 - Схема модели

Определить длины очередей к каналу и ЭВМ, загрузку устройств и среднее время обработки заданий в соответствии с заданными значениями времен генерации заявок и обработки заявок в устройствах.

Поступление заявок с равномерным распределением на отрезке [0.4,5].
Времена обработки заявок равномерно распределены на отрезке [1,7].

Для решения задачи моделирования необходимо выбрать масштаб времени:
 $t_3 = [(5+0.4)/2 \pm (5-0.4)/2] * 10$, $t_{обр} = (7+1)/2 \pm (7-1)/2 = (4 \pm 3) * 10$. Получаем $t_3 = 40 \pm 30$,
 $t_{обр} = 27 \pm 23$.

Текст программы-модели приведен далее.

```

10 GENERATE 27,23 ;Моделирование заявок
20 QUEUE QCHAN ;Очередь к каналу
30 SEIZE CHAN ;Канал
40 DEPART QCHAN
50 ADVANCE 50
60 RELEASE CHAN
70 QUEUE QEBM ;Очередь к ЭВМ
80 SEIZE EBM ;ЭВМ
90 DEPART QEBM
100 ADVANCE 40,30
110 RELEASE EBM
120 TERMINATE
130 GENERATE 10000 ;Время работы модели
140 TERMINATE 1

```

Теоретические расчеты заданных характеристик системы.

Интенсивность поступления заявок- $\lambda_{вх} = 1/t_3 = 1/27 = 0.037037$. Интенсивность обработки заявок - $\mu = 1/t_{обр} = 1/40 = 0.025$. Загрузка ЭВМ - $\rho = \lambda/\mu = 40/50 = 0.8$. Загрузка, меньшая 1, говорит о том, что к ЭВМ не будет очереди. Очередь будет только к каналу.

Длины очередей следующие:

$$L_{кан} = (\lambda_{вх} - \lambda_{вых}) * T_{мод} / 2 = (1/27 - 1/50) * 10000 / 2 \approx 85. \quad L_{ЭВМ} = (\lambda_{вх} - \lambda_{вых}) * T_{мод} / 2 = (1/50 - 1/40) * 10000 / 2 < 0.$$

Результаты моделирования. Листинг результатов моделирования приведен далее.

```

START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
0.000             10000.000   14      2           0
NAME              VALUE
CHAN              10001.000
EBM                10003.000
QCHAN             10000.000
QEBM              10002.000

LABEL            LOC  BLOCK TYPE  ENTRY COUNT  CURRENT COUNT  RETRY
1  GENERATE      369          0          0
2  QUEUE         369         169         0
3  SEIZE         200          0          0
4  DEPART        200          0          0
5  ADVANCE       200          1          0
6  RELEASE       199          0          0
7  QUEUE         199          0          0
8  SEIZE         199          0          0
9  DEPART        199          0          0
10 ADVANCE       199          1          0
11 RELEASE       198          0          0
12 TERMINATE     198          0          0
13 GENERATE      1           0          0          14
    TERMINATE     1           0          0

FACILITY          ENTRIES  UTIL.  AVE. TIME AVAIL.  OWNER  PEND  INTER  RETRY  DELAY
CHAN              200     0.996   49.786  1       201   0     0     0     169
EBM               199     0.779   39.136  1       200   0     0     0     0
QUEUE            MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME  AVE.(-0) RETRY
QCHAN            169  169    369     1     83.477  2262.247  2268.394  0
QEBM             2    0     199    112    0.139   6.975    15.954   0
FEC XN  PRI      BDT      ASSEM  CURRENT  NEXT  PARAMETER  VALUE
  371   0      10038.909  371    0        1
  201   0      10042.749  201    5        6
  200   0      10048.057   200   10       11
372   0      20000.000  372    0        13

```

Выводы по работе. За время работы модели было сгенерировано 369 транзактов. Только 169 успело обработаться. Основные потерянные заявки (199) - на канале. Длина очереди к каналу примерно соответствует рассчитанной. Загрузка устройств: $UTIL_{\text{кан}} = 0.996$, $UTIL_{\text{ЭВМ}} = 0.779$. На основании моделирования можно рекомендовать в заданной структуре поставить более быстрый канал. Тогда производительность всей системы будет выше. Блок-диаграмма модели приведена на рисунке 8.2.

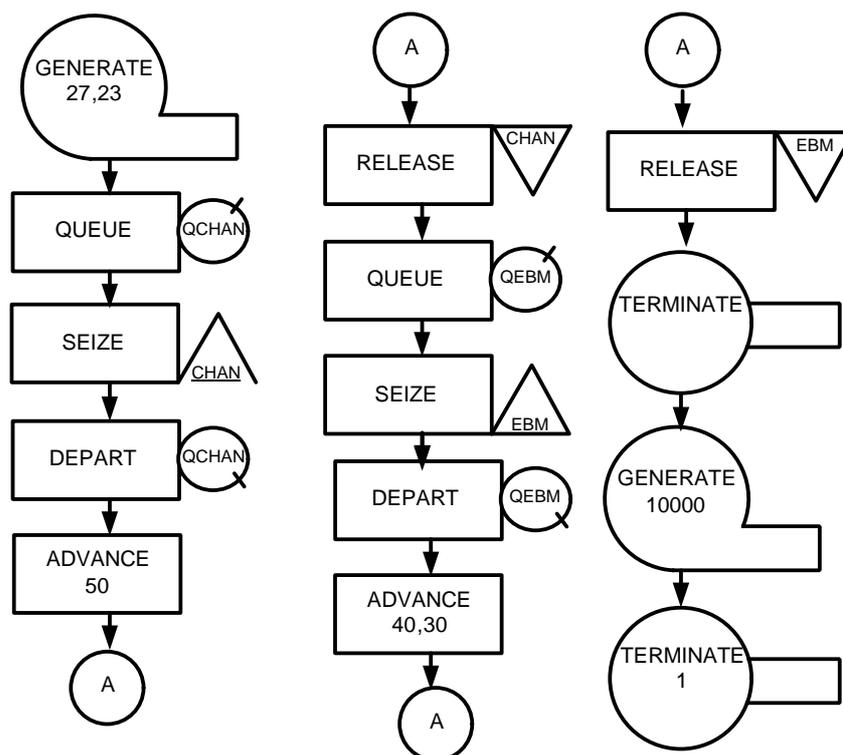


Рисунок 8.2 - Блок-диаграмма модели

9 Задание на лабораторную работу №7.

Исследование механизма формирования цепей текущих и будущих событий

Задача, решаемая в лабораторной работе, позволяет изучить механизм движения транзактов в модели, формирование модельного времени в системе моделирования GPSS и правила формирования цепей текущих и будущих событий.

Задана схема из терминала, канала для передачи сообщений на ЭВМ и ЭВМ.

- 1 Описать модель функционирования системы на языке GPSS/PC.
- 2 Составить таблицы времен случайных значений для генерации транзактов T, передачи сообщений P, обработки заданий на ЭВМ S.
- 3 На основании составленных таблиц рассчитать цепи текущих и будущих событий.
- 4 Промоделировать заданную схему в указанное время моделирования M.
- 5 Сравнить полученные результаты с расчетными.

6 Определить загрузку ЭВМ и среднюю длину очереди теоретически и в результате моделирования.

7 Построить гистограмму поступления запросов на обработку на ЭВМ. **Пример выполненной работы.** Времена генерации транзактов - $T_{ген} = [8 \pm 3]$. Время передачи сообщений в канале - $T_{пер} = [12 \pm 6]$, время обработки сообщений на ЭВМ - $T_{обр} = [13 \pm 4]$, наборы случайных значений времен - $T_{ген} = \{11, 9, 8, 11, 10, 6, 5, 10\}$; $T_{сум} = \{11, 20, 28, 39, 49, 55, 60, 69, 79\}$; $T_{пер} = \{17, 8, 11, 7, 12, 6, 18, 7\}$; $T_{обр} = \{10, 9, 16, 12, 9, 17, 12, 13\}$. Текст программы модели приведен далее.

```
10 GENERATE 8,3 ; Время генерации заявок
20 QUEUE QCHAN
30 SEIZE CHAN ; Канал передачи
40 DEPART QCHAN
50 ADVANCE 12,6 ; Время работы канала
60 RELEASE CHAN
70 QUEUE QEBM
80 SEIZE EBM ; ЭВМ
90 DEPART QEBM
100 ADVANCE 13,4 ; Время обработки заявок
110 RELEASE EBM
120 TERMINATE
130 GENERATE 75 ; Время работы модели 140
TERMINATE 1 ; Задается в варианте задачи.
```

Пояснения к составлению таблиц событий. При составлении таблицы цепей текущих и будущих событий времена генерации транзактов и задержки в устройствах берутся из выбранных случайных значений времен. Номер текущего блока - это тот блок модели, в который транзакт входит в наступившее модельное время. Номер следующего блока определяется тем блоком модели, до которого транзакт двигается без изменения модельного времени.

В цепях будущих событий номер следующего блока - это тот блок, в который попадет транзакт по истечении времени задержки. Пребывание в очереди к устройству выводит транзакт в список очередей, поэтому в результатах моделирования все транзакты, стоящие в очереди не присутствуют в списках текущих и будущих событий. Фаза моделирования определяется очередным временем движения транзакта в модели. В расчетной таблице

получилось 16 фаз моделирования. Из таблицы видно, что за указанное время было сгенерировано 10 транзактов, а обработано только 5 транзактов. Остальные заявки остались необработанными, т. к. очередь к каналу имеет максимальную длину 4, а очередь к ЭВМ - 1.

Таблица 9.1 – Рассчитанные цепи текущих и будущих событий.

Фаза моделирования	Модельное время	Цепь текущих событий	Цепь будущих событий
1	0	Пусто	[1,0,11,0,1] [2,0,75,0,13]
2	11	[1,0,11,1,5]	[1,0, 28, нет, 6] [3,0,20,0,1] [2,0,75,0,13]
3	20	[3,0,20,1,5]-очередь	[1,0,28,нет,6] [2,0,75,0,13] [4,0,28,0,1]
4	28	[1,0,28,6,10] [3,0,28,1,5] [4,0,28,1,5]-очередь	[1,0,38,нет,11] [2,0,75,0,13] [3,0,36,нет,6] [5,0,39,0,1]
5	36	[3,0,36,6,10]-очередь [4,0,36,1,5]	[1,0,38,нет,11] [2,0,75,0,13] [4,0,47,нет,6] [5,0,39,0,1]
6	38	[1,0,38,11,12]* [3,0,38,6,10]	[2,0,75,0,13] [3,0,47,нет,11] [4,0,47,нет,6] [5,0,39, 0, 1]
7	39	[5,0,39,1,5]-очередь	[2,0,75,0,13] [3,0,47,нет,11] [4,0,47,нет,6] [6,0,49,0,1]
8	45	[5,0,45,1,5]-очередь	[2,0,75,0,13] [4,0,47,нет,6] [6,0,49,0,1]
9	47	[3,0,47,11,12,]* [4,0,47,6,10] [5,0,47,1,5]	[2,0,75,0,13] [4, 0,63, нет, 11] [5, 0,54, нет, 6] [6,0,49, 0.1]
10	49	[6, 0,49,1,5] -очередь	[2,0,75,0,13][4,0,63, нет,11] [5,0,54, нет,6] [7,0,55,0,1]
11	54	[5,0,54,6,10] - очередь [6,0,54,1,5]	[2,0,75,0,13] [4,0,63,нет,11] [6,0,76,нет, 6] [7,0,55,0,1]

Продолжение таблицы 9.1

Фаза моделирования	Модельное время	Цепь текущих событий	Цепь будущих событий
12	55	[5,0,55,6,10]-очередь [7,0,55, 1, 5]-очередь	[2,0,75,0,13] [4,0,63,нет,11] [6,0,76,нет,6] [8,0,60,0,1]
13	60	[5,0,60,6,10]-очередь [7,0,60,1,5]-очередь [8,0,60,1,5]- очередь	[2,0,75,0,13] [4,0,63,нет,11] [6,0,76,нет, 6] [9,0,69,0,1]
14	63	[4,0,63,12,12]* [5,0,63,6,10] [7, 0,63, 1,5] - очередь [8,0,63,1,5]- очередь	[2,0,75,0,13] [5,0,75,нет,11] [6,0,76,нет, 6] [9,0,69,0,1]
15	69	[7,0,69,1,5] - очередь [8,0,69,1,5]-очередь [9,0,69,0,1] - очередь	[2,0,75,0,13] [5,0,75,нет,11] [6,0,76,нет,6] [10,0,79,0,1]
16	75	[2,0,75,13,14]* [5,0,75,11,12]* [7,0,75,1,5]-очередь [8,0,75,1,5]-очередь [9,0,75,0,1]-очередь	[6,0,76,нет,6] [10,0,79,0,1] [11,0,150,0,13]

Знаком * отмечены транзакты, которые завершили движение по модели. Обозначение очередь - это указание на пребывание транзакта в очереди к устройству.

Расчеты загрузки ЭВМ и средних длин очередей. Интенсивность поступления заявок - $\lambda_{\text{вх,кан}}=1/8=0.125$, $\lambda_{\text{вх,эвм}}=1/12=0.083$. Интенсивность обработки заявок - $\mu_{\text{эвм}}=1/13=0.076923$, $\mu_{\text{кан}}=1/12=0.083$. Загрузка ЭВМ $\rho_{\text{эвм}}=\lambda/\mu=1.079$, $\rho_{\text{кан}}=0.125/0.076923=1.625 > 1$. Длина очереди к каналу - $L_{\text{средн}} = (\lambda_{\text{вх}}-\lambda_{\text{вых}})/2 * T_{\text{мод}} = 1.575$. Длина очереди к ЭВМ - $L_{\text{средн}}=(\lambda_{\text{вх}}-\lambda_{\text{вых}})/2 * T_{\text{мод}} = 0.225$. Листинг результатов моделирования приведен далее.

GPSS World Simulation Report - lr2.1.1

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	75.000	14	2	0

NAME	VALUE
CHAN	10001.000
EBM	10003.000
QCHAN	10000.000
QEBM	10002.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
-------	-----	------------	-------------	---------------	-------

1	GENERATE		9		0		0			
2	QUEUE		9		4		0			
3	SEIZE		5		0		0			
4	DEPART		5		0		0			
5	ADVANCE		5		1		0			
6	RELEASE		4		0		0			
7	QUEUE		4		0		0			
8	SEIZE		4		0		0			
9	DEPART		4		0		0			
10	ADVANCE		4		1		0			
11	RELEASE		3		0		0			
12	TERMINATE		3		0		0			
13	GENERATE		1		0		0			14
	TERMINATE		1		0		0			
FACILITY										
CHAN		ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
EBM		5	0.866	12.989	1	6	0	0	0	4
QUEUE		4	0.645	12.093	1	5	0	0	0	0
QCHAN		MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY	
QEBM		4	4	9	1	1.323	11.026	12.404	0	
FEC	XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE		
5	0		77.593	5	10	11				
6	0		80.918	6	5	6				
11	0		85.498	11	0	1				
12	0		150.000	12	0	13				

В листинге присутствует список цепей текущих и будущих событий на момент окончания моделирования. Суть различия - во временах генерации транзактов по сравнению с ручным расчетом, например 5-й транзакт имеет время генерации 77, а в ручном расчете - 39. Кроме того, успел уже сгенерироваться в модели 11 транзакт, а в расчетах только 10 транзактов успели возникнуть в течение заданного времени моделирования. Цепи текущих и будущих событий в промежуточные моменты времени моделирования показывают, что рассчитанные цепи совпадают с получаемыми в модели.

Выводы по результатам моделирования: загрузка ЭВМ получилась, как и положено, меньше 1 и равна 0.645. Очередь к ЭВМ практически отсутствует, т. к. очередь существует к каналу и основная часть заявок осталась на входе канала. Очередь к каналу соответствует расчетной. Анализ рассчитанных и полученных в модели цепей текущих и будущих событий показывает, что подготовка была выполнена правильно. (В листинге цепей текущих и будущих событий приведено сравнение содержимого цепей для разных текущих времен моделирования.) Числа обработанных и задержанных транзактов совпадают. Несколько различаются времена генерации транзактов, т. к. выбор значений этих времен случаен в пределах заданных в модели границ.

Выбранные для анализа характеристики устройств и времена поступления заявок не обеспечивают устойчивую работу системы, т. к. накапливается очередь, загрузка ЭВМ значительно меньше 1. Для нормальной работы такой системы необходимо или увеличить пропускную способность канала, или оптимизировать времена поступления заявок.

Блок-диаграмма модели приведена на рисунке 9.1.

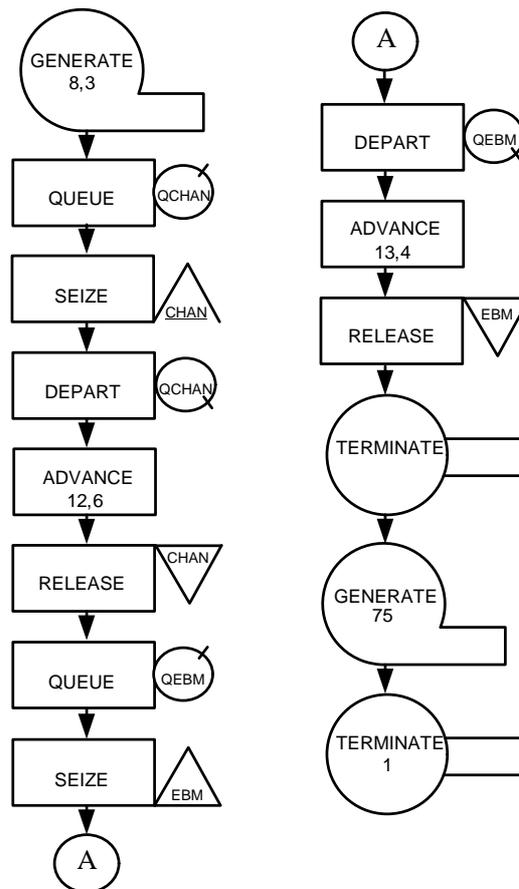


Рисунок 9.1 - Блок-диаграмма модели

10 Задание на лабораторную работу №8. Определение оптимального времени поступления заявок в заданной конфигурации ВС

Базовая схема для моделирования состоит из М терминалов пользователей, канала прямой передачи данных и канала обратной передачи данных. По сигналу пользователя данные могут считываться из некоторой базы данных и отсылаться на хранение в эту базу данных. Рассчитать

производительность ЭВМ, длины очередей к каналам и ЭВМ и определить оптимальное время поступления запросов на обработку для заданных в варианте конфигурации схемы, характеристиках обслуживания ЭВМ и времени работы любого из каналов. Обязательны таблицы времен обработки транзактов на ЭВМ.

Оптимальное время поступления заявок должно обеспечивать наивысшую загрузку ЭВМ и минимальную очередь на обработку заданий, в пределах 2-4 заявок.

Список обозначений исходных данных: Т - терминал пользователя; КБД - канал базы данных; ЭВМ - процессор и память; БД - база данных; К - канал прямой и обратной передачи данных; КП - канал прямой передачи сообщений; КО - канал обратной передачи сообщений; μ - интенсивность обслуживания сообщений с пуассоновским законом распределения времени обработки заявок; P1, P2 - интервал обработки заявок с равномерным законом обслуживания в ЭВМ; K1, K2 - интервал времен работы любого из каналов.

Порядок выполнения лабораторной работы.

- 1 Написать программу модели.
- 2 Промоделировать систему с разными значениями t, x .
- 3 Построить зависимости $\rho = f(t_{вх})$ И $Q = f(t_{вх})$.
- 4 Выполнить ручной расчет загрузки устройств, длин очередей и производительности системы.
- 5 Построить гистограммы времен обработки транзактов для каждого типа заявок.
- 6 Проанализировать полученные результаты моделирования и определить оптимальное время поступления заявок для заданной конфигурации схемы.

Пример выполненной лабораторной работы

Задана схема из двух терминалов, канала передачи сообщений на ЭВМ и от ЭВМ, ЭВМ, канала к БД и БД. Время передачи сообщений по любому из каналов на отрезке [4.9]. Время обработки сообщений на ЭВМ равномерно распределено на отрезке [10.15].

Текст программы-модели приведен далее.

10 SIMULATE

20 PROBA EQU 480 ; установка $t_{вх}$
 25 T_TAB TABLE MP4,100,100,5 ; формат таблицы для табулирования 30
 GENERATE PROBA
 40 ASSIGN 1,1 ;определение номера терминала
 50 TRANSFER ,МЕТКА
 60 GENERATE PROBA
 70 ASSIGN 1,2 ;определение номера терминала
 80 МЕТКА QUEUE QKAN
 90 SEIZE KAN
 100 DEPART QKAN
 110 ADVANCE 65,25
 120 RELEASE KAN
 130 MARK 4 ;отметка времени входа транзакта в очередь к ЭВМ
 140 QUEUE QEBM 150
 SEIZE EBM
 160 DEPART QEBM
 170 ADVANCE 125,25
 180 RELEASE EBM
 190 TABULATE T_TAB ;первая точка табуляции времени пребывания
 200 QUEUE QKBD ;транзакта в очереди к ЭВМ и в самой ЭВМ
 210 SEIZE KBD
 220 DEPART QKBD
 230 ADVANCE 65,25
 240 RELEASE KBD
 250 QUEUE BD 260
 SEIZE BD
 270 DEPART BD
 280 ADVANCE 0
 290 RELEASE BD
 300 QUEUE QBD 310
 SEIZE KBD
 320 DEPART QBD
 330 ADVANCE 65,35
 340 RELEASE KBD
 350 MARK 4 ;отметка вхождения в очередь транзакта при движении от
 360 QUEUE QEBM ;базы данных
 370 SEIZE EBM
 380 DEPART QEBM
 390 ADVANCE 125,25
 400 RELEASE EBM
 410 QUEUE QKAN 420
 SEIZE KAN

430 DEPART QKAN
 440 ADVANCE 65,25
 450 RELEASE KAN
 455 TABULATE T_TAB ; Табуляция времени пребывания транзактов в ЭВМ
 460 TEST E P1,1,TERM2 ; переменные позволят определить количество
 470 SAVEVALUE PER1+,1; обработанных заявок с каждого из терминалов
 480 TRANSFER ,МЕТКА1 ; пользователей
 490 TERM2 SAVEVALUE PER2+,1
 500 МЕТКА1 TERMINATE
 510 GENERATE 10000 ; выбранное время моделирования
 520 TERMINATE 1

Листинг результатов моделирования приведен далее.

GPSS World Simulation Report - lr3.15.1

	START TIME	END TIME	BLOCKS	FACILITIES	STORAGES	
0.000	10000.000	51	4	0		
	NAME		VALUE			
	BD		10008.000			
	EBM		10005.000			
	KAN		10003.000			
	KBD		10007.000			
	МЕТКА		6.000			
	МЕТКА1		49.000			
	PER1		10010.000			
	PER2		10011.000			
	PROBA		480.000			
	QBD		10009.000			
	QEBM		10004.000			
	QKAN		10002.000			
	QKBD		10006.000			
	TERM2		48.000			
	T_TAB		10001.000			
LABEL	LOC	BLOCK	TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
1		GENERATE	20	0	0	
2		ASSIGN	20	0	0	
3		TRANSFER	20	0	0	
4		GENERATE	20	0	0	
5		ASSIGN	20	0	0	
МЕТКА	6	QUEUE		40	0	0
7		SEIZE	40	0	0	
8		DEPART	40	0	0	
9		ADVANCE	40	0	0	
10		RELEASE	40	0	0	
11		MARK	40	0	0	
12		QUEUE	40	1	0	
13		SEIZE	39	0	0	
14		DEPART	39	0	0	
15		ADVANCE	39	0	0	
16		RELEASE	39	0	0	
17		TABULATE	39	0	0	
18		QUEUE	39	0	0	
19		SEIZE	39	0	0	

20	DEPART	39	0	0						
21	ADVANCE	39	1	0						
22	RELEASE	38	0	0						
23	QUEUE	38	0	0						
24	SEIZE	38	0	0						
25	DEPART	38	0	0						
26	ADVANCE	38	0	0						
27	RELEASE	38	0	0						
28	QUEUE	38	0	0						
29	SEIZE	38	0	0						
30	DEPART	38	0	0						
31	ADVANCE	38	0	0						
32	RELEASE	38	0	0						
33	MARK	38	0	0						
34	QUEUE	38	1	0						
35	SEIZE	37	0	0						
36	DEPART	37	0	0						
37	ADVANCE	37	1	0						
38	RELEASE	36	0	0						
39	QUEUE	36	0	0						
40	SEIZE	36	0	0						
41	DEPART	36	0	0						
42	ADVANCE	36	0	0						
43	RELEASE	36	0	0						
44	TABULATE	36	0	0						
45	TEST	36	0	0						
46	SAVEVALUE	18	0	0						
47	TRANSFER	18	0	0						
TERM2		48	SAVEVALUE	18		0		0		
METKA1		49	TERMINATE	36		0		0		
50	GENERATE	1	0	0						
51	TERMINATE	1	0	0						
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY	
KAN	76	0.486	63.975	1	0	0	0	0	0	
EBM	76	0.936	123.188	1	38	0	0	0	2	
KBD	77	0.478	62.085	1	40	0	0	0	0	
BD	38	0.000	0.000	1	0	0	0	0	0	
QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY		
QKAN	2	0	76	39	0.212	27.920	57.350	0		
QEBM	4	2	78	6	1.226	157.238	170.341	0		
QKBD	1	0	39	31	0.017	4.351	21.213	0		
BD	1	0	38	38	0.000	0.000	0.000	0		
QBD	1	0	38	38	0.000	0.000	0.000	0		
TABLE	MEAN	STD.DEV.		RANGE		RETRY	FREQUENCY	CUM.%		
T_TAB	314.767	113.433				0				
			100.000	-	200.000		10	13.33		
			200.000	-	300.000		24	45.33		
			300.000	-	400.000		24	77.33		
			400.000	-	-		17	100.00		
SAVEVALUE		RETRY	VALUE							
PER1		0	18.000							
PER2		0	18.000							
FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE			
40	0	10019.946	40	21	22	1	1.000			
						4	9645.135			
	38	0	10079.160	38	37	38	1	1.000		
						4	9665.276			
42	0	10080.000	42	0	1					
43	0	10080.000	43	0	4					
44	0	20000.000	44	0	50					

Выводы по работе. На основании проведенного моделирования заданной конфигурации ВС и обработки результатов имитации видно, что оптимальным временем для поступления заявок является $t_{вх}$, равное 480 единицам времени. При этом среднем времени поступления заявок длина очереди равна 2 единицам, а загрузка ЭВМ - порядка 0.93.

Табуляция времен обработки заявок на ЭВМ показала, что среднее время обработки при выбранном времени генерации заявок равно 314.7, из которого только 123 единицы времени - собственно обработка заявок, а остальное время расходуется на пребывание в очереди.

Замечание по структуре: быстродействие ЭВМ должно быть выше, чтобы в целом производительность системы была выше.

Блок-диаграмма модели приведена на рисунке 10.1.

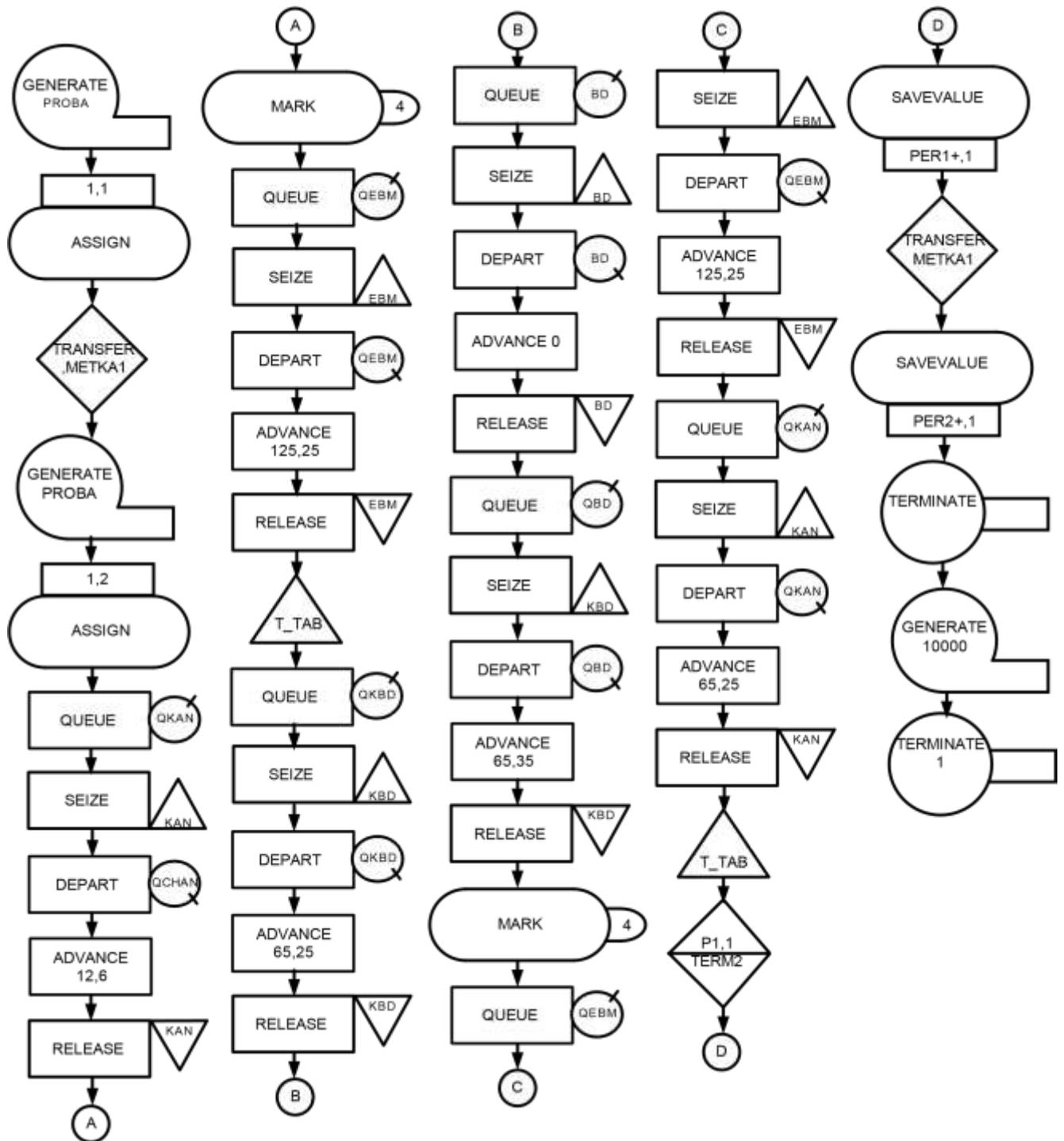


Рисунок 10.1 - Блок-диаграмма модели

11 Задание на лабораторную работу №9. Построение модели ВС заданной конфигурации. Определение основных характеристик ВС

Для каждого варианта задания предлагается некоторая абстрактная конфигурация ВС с различными вариантами обработки команд, организации памяти и количеством процессоров для выполнения команд. При решении

задачи моделирования необходимо описать абстрактными объектами системы моделирования заданные правила работы системы и определить зависимости характеристик системы от различных временных характеристик объектов системы.

Методика решения задачи следующая.

- 1 Написать программу модели для заданного варианта.
- 2 Выполнить ручной расчет основных характеристик ЭВМ.
- 3 Отладить модель.
- 4 С помощью таблиц и отметок времени движения заявок в модели получить заданные характеристики.
- 5 Обработать результаты моделирования. Построить соответствующие зависимости.
- 6 Сделать выводы по результатам моделирования.

Пример выполнения работы. Построить модель процесса функционирования СЭВМ, выполняющей n программ в мультипрограммном режиме. В состав СЭВМ входят процессор ЦП и m селекторных каналов СК. Времена обслуживания запросов процессором и каналами - случайные величины, распределенные для процессора по экспоненциальному закону с интенсивностью $\mu_{цп}$ и для каналов по равномерному закону в диапазоне t_n - t_b . Требуется получить коэффициенты загрузки всех устройств, средние длины очередей запросов к ЦП и СК, гистограммы распределения времен нахождения запросов в очереди и среднее время между последовательными обращениями к ЦП каждой из программ.

Дополнительные пояснения к варианту задачи: запросы к ресурсам обслуживаются на основе относительных приоритетов (приоритет равен условному номеру пославшей его программы и может принимать значения от 1 до n); очередь к каналам отдельная, обращение к каждому из каналов равновероятно. Значения $n = 3$, $m=2$, $t_n= 20$ ms, $t_b= 90$ ms. Исследовать зависимость характеристик для $\mu_{цп}= 0.002, 0.004, 0.008$ ms⁻¹. На рисунке 11.1 приведена моделируемая СЭВМ.

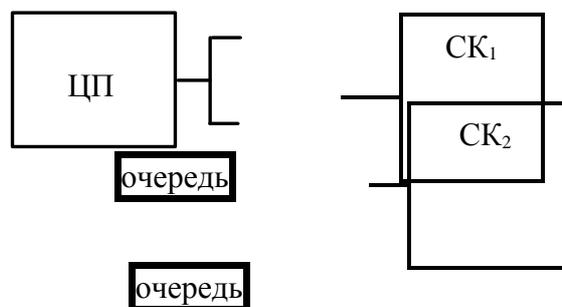


Рисунок 11.1 – Моделируемая СЭВМ

Текст программы-модели приведен далее.

```

10 SIMULATE
20 TCAN1 TABLE MP1,100,500,10
30 TCAN2 TABLE MP1,100,500,10
40 AAA TABLE MP1,100,200,10
;Таблицы для контроля за временем обработки заявок на участках модели
50 XPDIS FUNCTION RN1,C24
;Описание экспоненциальной функции распределения времен распр. заявок
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/.9998,8
60 GENERATE 1,,1,1 70
ASSIGN 5,100
80 TRANSFER ,INCP
90 GENERATE 1,,1,2 100
ASSIGN 5,100
110 TRANSFER ,INCP
120 GENERATE 1,,1,3
130 ASSIGN 5,100
; Раздел описания генерации завершен. В каждом генераторе был задан транзакт
соответствующего приоритета
140 INCP MARK 1
150 QUEUE CP
160 PREEMPT CP,PR ; Процессор обрабатывает приоритет заявок
170 DEPART CP
175 ADVANCE 125,FN$XPDIS 180
RETURN CP
190 TRANSFER .5,INCAN1,INCAN2
; Равновероятный выбор канала для всех заявок
200 INCAN1 QUEUE CAN1

```

```

210 PREEMPT CAN1,PR 220
DEPART CAN1
230 ADVANCE 550,350 240
RETURN CAN1
; завершение работы CAN1
250 TRANSFER ,CIR
260 INCAN2 QUEUE CAN2
270 PREEMPT CAN2,PR 280
DEPART CAN2
290 ADVANCE 550,350 300
RETURN CAN2
; завершение работы CAN2
310 CIR TEST E PR,1,МЕТКА ; Проверка приоритета заявок
320 SAVEVALUE BB1+,1 ; Переменная для подсчета заявок с приоритетом=1
330 TABULATE TCAN1
340 TRANSFER ,CIRCLE
350 МЕТКА TEST E PR,2,МЕТКА1
360 SAVEVALUE BB2+,1 ;Переменная для подсчета заявок с приоритетом=2
370 TABULATE TCAN2
380 TRANSFER ,CIRCLE
390 МЕТКА1 TABULATE AAA ; Табуляция времени обработки заявки
400 SAVEVALUE BB3+,1 ; Подсчет заявок с приоритетом=3
410 CIRCLE LOOP 5,INCP ; Возврат транзактов
420 TERMINATE 1

```

Листинг результатов моделирование приведен далее.

```

GPSS World Simulation Report - lr4.1.1
START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
0.000          65503.987   38       3           0
NAME          VALUE
AAA          10002.000
BB1          10009.000
BB2          10008.000
BB3          10007.000
CAN1         10006.000
CAN2         10005.000
CIR           27.000
CIRCLE       37.000
CP          10004.000
INCAN1       16.000
INCAN2       22.000
INCP         9.000
МЕТКА        31.000
МЕТКА1       35.000
TCAN1       10000.000

```

		TCAN2			10001.000					
		XPDIS			10003.000					
LABEL	LOC	BLOCK	TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY		
1	GENERATE	1		0	0					
2	ASSIGN	1		0	0					
3	TRANSFER	1		0	0					
4	GENERATE	1		0	0					
5	ASSIGN	1		0	0					
6	TRANSFER	1		0	0					
7	GENERATE	1		0	0					
8	ASSIGN	1		0	0					
INCP	9	MARK		179		0		0		
10	QUEUE	179		0	0					
11	PREEMPT	179		0	0					
12	DEPART	179		0	0					
13	ADVANCE	179		0	0					
14	RETURN	179		0	0					
15	TRANSFER	179		0	0					
INCAN1	16	QUEUE		89		0		0		
17	PREEMPT	89		0	0					
18	DEPART	89		0	0					
19	ADVANCE	89		1	0					
20	RETURN	88		0	0					
21	TRANSFER	88		0	0					
INCAN2	22	QUEUE		90		0		0		
23	PREEMPT	90		0	0					
24	DEPART	90		0	0					
25	ADVANCE	90		1	0					
26	RETURN	89		0	0					
CIR	27	TEST		177		0		0		
28	SAVEVALUE	31		0	0					
29	TABULATE	31		0	0					
30	TRANSFER	31		0	0					
METKA	31	TEST		146		0		0		
32	SAVEVALUE	46		0	0					
33	TABULATE	46		0	0					
34	TRANSFER	46		0	0					
METKA1	35	TABULATE		100		0		0		
	36	SAVEVALUE		100		0		0		
CIRCLE	37	LOOP		177		0		0		
	38	TERMINATE		1		0		0		
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY	
CP	179	0.347	127.034	1	0	0	0	0	0	
CAN2	90	0.756	550.196	1	1	0	0	0	0	
CAN1	89	0.720	529.913	1	2	0	0	0	0	
QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY		
CP	2	0	179	154	0.051	18.606	133.216	0		
CAN2	2	0	90	64	0.261	189.880	657.279	0		
CAN1	2	0	89	65	0.276	203.403	754.285	0		
TABLE	MEAN	STD. DEV.	RANGE		RETRY	FREQUENCY	CUM. %			
TCAN1	2091.939	1293.318			0					
			100.000	-	600.000			2	6.45	
			600.000	-	1100.000			3	16.13	
			1100.000	-	1600.000			4	29.03	
			1600.000	-	2100.000			11	64.52	
			2100.000	-	2600.000			4	77.42	
			2600.000	-	3100.000			4	90.32	
			3100.000	-	3600.000			0	90.32	
			3600.000	-	4100.000			1	93.55	
			4100.000	-				2	100.00	
TCAN2	1396.376	794.577				0				

				100.000	-	600.000		9	19.57
				600.000	-	1100.000		10	41.30
				1100.000	-	1600.000		11	65.22
				1600.000	-	2100.000		7	80.43
				2100.000	-	2600.000		5	91.30
				2600.000	-	3100.000		4	100.00
AAA		655.030	247.988					0	
				100.000	-	300.000		3	3.00
				300.000	-	500.000		31	34.00
				500.000	-	700.000		28	62.00
				700.000	-	900.000		17	79.00
900.000	-	1100.000		16	95.00				
				1100.000	-	1300.000		4	99.00
				1300.000	-	1500.000		1	100.00
SAVEVALUE			RETRY	VALUE					
BB3			0	100.000					
BB2			0	46.000					
BB1			0	31.000					
FEC	XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE	
	2	2	65536.432	2	19	20	1	64234.311	
							5	54.000	
	1	1	66074.939	1	25	26	1	64851.103	
							5	69.000	

Из листинга результатов моделирования проанализируем наиболее существенные разделы: состояние устройств, очередей и таблицы, определяющие обработку заявок с разными приоритетами. Количество обработанных заявок каждым устройством отражается в разделе Entries.

Переменные BB1, BB2, BB3 показывают распределение заявок по приоритетам при их обработке: транзакты с приоритетом 3 в заданном количестве 100, с приоритетом 2 - 41, а с приоритетом 1 - 25. Соответственно распределено и время их обработки, что отражено в таблицах TCAN1, TCAN2, AAA. Приведенные результаты получены для интенсивности обработки транзактов равной, 0.008. Все временные параметры модели соответственно увеличены в 10 раз, чтобы значения времен были целыми. Для сопоставления результатов моделирования с реальными характеристиками заданной ВС все времена надо уменьшить в 10 раз.

Измерения длин очередей и загрузки процессора для заданных в работе значений $\mu_{\text{цп}}$ показали, что соотношение по приоритетам при разных интенсивностях не меняется, но распределение длин очередей к процессору и каналам показывает, что быстродействие каналов должно быть больше по сравнению с заданным.

Блок-диаграмма модели приведена на рисунке 11.2.

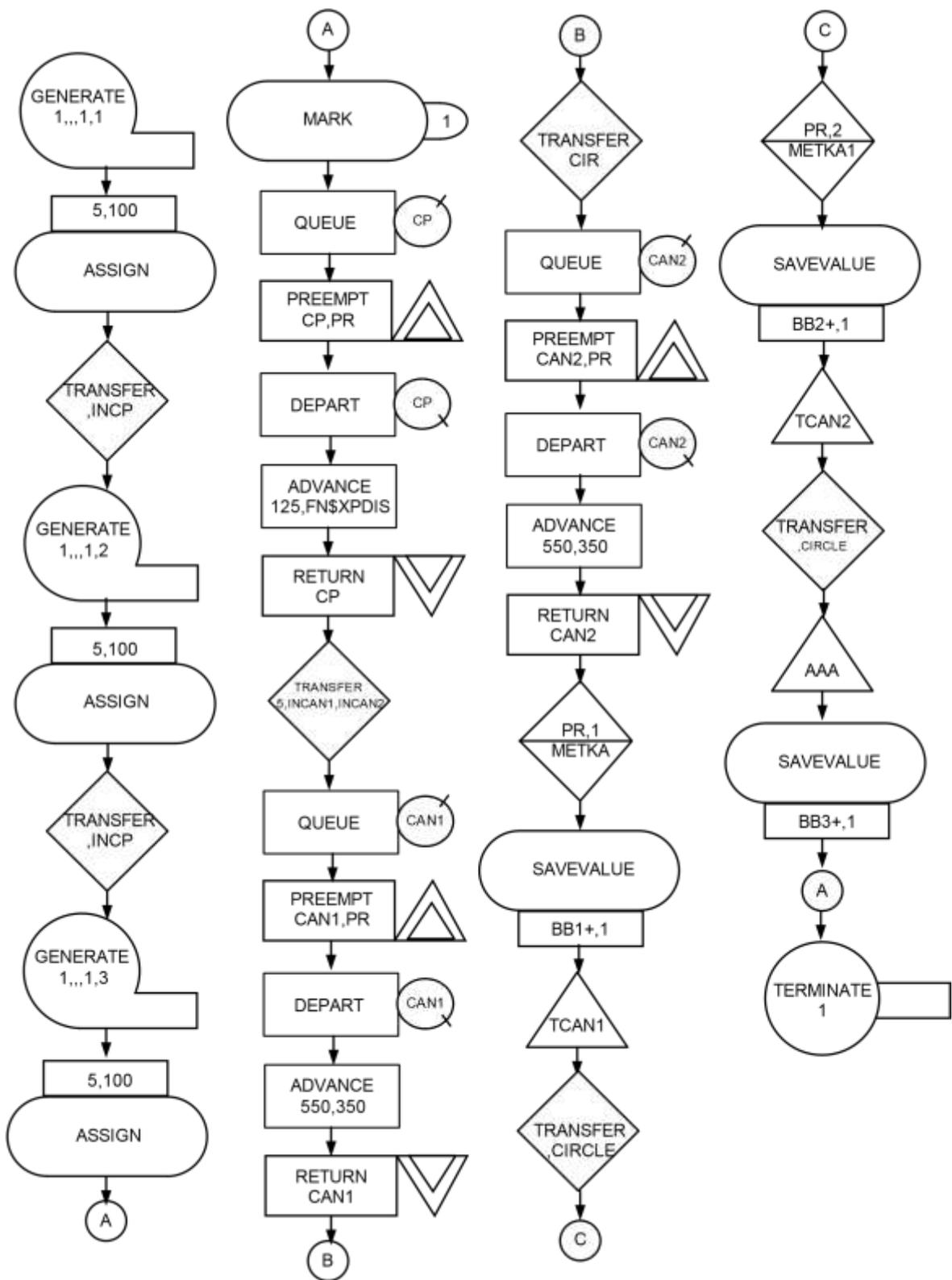


Рисунок 11.2 - Блок-диаграмма модели

12 Задание на лабораторную работу №10. Исследование на имитационной модели процесса передачи данных в информационно-вычислительной сети

Цель работы - изучение особенностей построения моделей информационных процессов и их машинной реализации с использованием языка GPSS, исследование вероятностно-временных характеристик процесса функционирования сети передачи данных методом имитационного моделирования.

Объектом исследования в данной лабораторной работе является СПД, осуществляющая передачу пакетов между узлами коммутации информационно-вычислительной сети (ИВС).

Данные — это факты и(или) понятия, описанные в формализованном виде. В ИВС различают пользовательские (информационные) и управляющие (служебные) данные. Пользовательские данные — это данные, вводимые пользователями в ИВС или получаемые ими из сети. Управляющие данные — это данные, используемые для управления работой ИВС.

Сеть представляет собой совокупность средств передачи и распределения данных. Выделяют магистральную (базовую) и терминальную (абонентскую) части ИВС. Магистральная часть ИВС служит для передачи данных между вычислительными комплексами, ресурсы которых доступны для пользователей сети, и включает узлы коммутации (УК) и соединяющие их каналы связи (КС). Узел коммутации выполняет функции маршрутизации, передачи и коммутации данных и имеет для этого соответствующие аппаратнопрограммные средства. Канал связи представляет собой совокупность технических средств и среды распространения, которая обеспечивает доставку данных в требуемую точку сети.

Терминальная часть ИВС используется для подключения непосредственно либо через концентраторы нагрузки абонентских пунктов и терминалов пользователей. Концентратор — устройство, обеспечивающее сопряжение входных низкоскоростных каналов связи с выходным высокоскоростным каналом связи. Абонентские пункты оборудуются аппаратурой передачи данных и устройствами ввода-вывода, т. е. терминалами, с помощью которых пользователи могут осуществлять доступ к вычислительным ресурсам и базам данных сети. Обычно терминалы группируются и подсоединяются к терминальной сети. В качестве терминалов могут быть использованы как простейшие устройства ввода-вывода

(телетайпы, дисплеи и т. п.), так и персональные (интеллектуальные) терминалы.

В рассматриваемой ИВС реализован режим коммутации пакетов, представляющий такой способ передачи, при котором данные из сообщений пользователей разбиваются на отдельные пакеты. Маршруты передачи пакетов в сети от источника к получателю определяются в каждом УК, куда они поступают. Под сообщениями понимается конечная последовательность символов, имеющая смысловое содержание. Пакет — это блок данных с заголовком, представленный в установленном формате и имеющий ограниченную максимальную длину. Обычно в ИВС используются пакеты постоянной длины, содержащие от 500 до 2000 двоичных знаков (бит). Отметим, что ИВС с коммутацией пакетов обладают высокой эффективностью благодаря возможности быстрой перестройки путей передачи данных (маршрутизации) при возникновении перегрузок и повреждении элементов ИВС. Эффективность различных вариантов построения ИВС и ее фрагментов оценивается средними временами доставки данных пользователям и вероятностями отказа в установлении в данный момент времени требуемого пользователю соединения. Совокупность таких показателей для оценки эффективности процесса функционирования ИВС принято называть ее вероятностно-временными характеристиками.

Для упрощения объекта моделирования (в данном случае из-за необходимости упрощения учебного примера с точки зрения его обозримости и, что особенно важно, уменьшения машинных затрат на его реализацию) рассмотрим фрагмент ИВС, представляющий процесс взаимодействия двух соседних УК сети, которые обозначим УК1 и УК2. Эти узлы соединены между собой дуплексным дискретным каналом связи (ДКС), позволяющим одновременно передавать данные во встречных направлениях, т. е. имеется два автономных однонаправленных ДКС: К1 и К2.

Структурная схема варианта УК представлена на рисунке 12.1, где ВхБН и ВыхБН — входные и выходные буферные накопители соответственно; К — коммутаторы; ЦП — центральный процессор. Данный УК функционирует следующим образом. После поступления пакета из одного из входных КС узла он помещается в ВхБН. Затем ЦП на основании заголовка пакета и хранимой в УК маршрутной таблицы определяет требуемое направление дальнейшей передачи пакета и помещает его в соответствующий ВыхБН для последующей передачи по выходному КС.

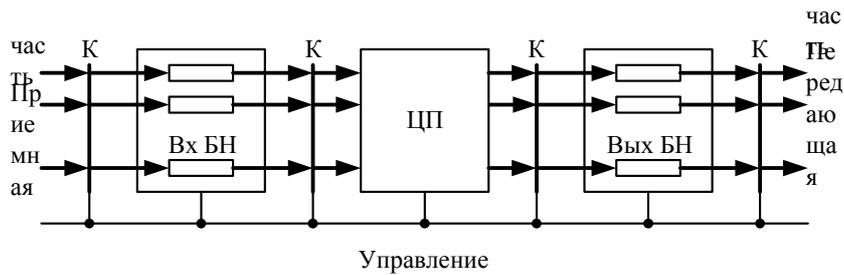


Рисунок 12.1 - Структурная схема варианта УК

Структурная схема варианта ДКС с решающей обратной связью показана на рис. 6.22, где КУ и ДКУ — соответственно кодирующее и декодирующее устройства; УУК — устройство управления каналом; КА — каналообразующая аппаратура. На передающей стороне пакет из ВыхБН узла коммутации попадает в КУ, где производится кодирование, т. е. внесение избыточности, необходимой для обеспечения помехоустойчивой передачи по КС. Согласование с конкретной средой распространения реализуется КА (например, организация коротковолнового радиоканала через спутник — ретранслятор для СПД или оптического канала с использованием световода для локальной СПД). На приемной стороне из КА пакет попадает в ДКУ, которое настроено на обнаружение или исправление ошибок. Все функции управления КУ, ДКУ (в том числе и принятие решений о необходимости повторного переспроса копии пакета с передающего УК) и взаимодействия с центральной частью узла реализуется УУК, которое является либо автономным, либо представляет собой часть процессов, выполняемых ЦП узла.

Процесс функционирования СПД заключается в следующем. Пакеты данных поступают в исследуемый фрагмент по линии связи. Считается, что интервалы между моментами поступления распределены по экспоненциальному закону. После обработки в центральном процессоре они поступают в выходной накопитель. Далее в порядке очереди копия пакета передается по дискретному каналу связи и поступает во входной накопитель второго узла. После обработки в центральном процессоре второго узла пакет данных передается в выходную линию (3 или 4) и формируется подтверждение приема, которое в виде короткого пакета поступает в выходной накопитель для передачи в исходный узел. После приема подтверждения в исходном узле осуществляется уничтожение пакета и подтверждения.

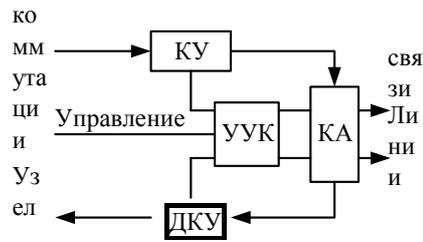


Рисунок 12.2 – Структурная схема варианта ДКС

Задание к работе. Исследуемый фрагмент СПД, представленный в виде композиции *Q-схем*, приведен на рис. 12.3, обозначения которого соответствуют введенным ранее.

Исходные данные для моделирования: средний интервал между пакетами данных — 25 ед. вр; емкости накопителей — 20 ед.памяти; время передачи пакета - данных по ДКС — 20 ед. вр; время передачи подтверждения по ДКС — 1 ед. вр; время обработки пакета в ЦП — 2 ед. вр.

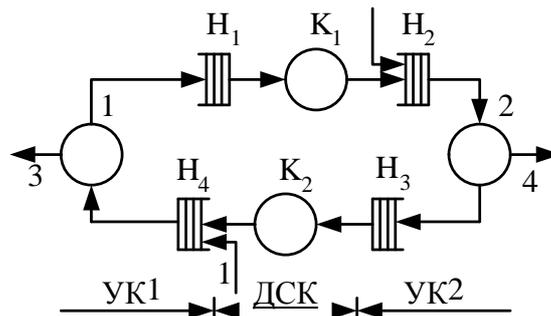


Рисунок 12.3 - Исследуемый фрагмент СПД

Текст исходной программы приведен далее, где CPUJ — обозначение j-го центрального процессора, DCHL — обозначение L-го дискретного канала связи.

```
SIMULATE
0001 EXPON      FUNCTION      RN1,C24
0,0/.100,.104/.200,.222/.300,.355/.400,.509/.500,.690/.600,.915/.700,1.200
.750,1.380/.800,1.600/.840,1.830/.880,2.120/.900,2.300/.920,2.520/.940,2.810/.950,.
990/.960,3.200/.970,3.500/.980,3.900/.990,4.600/.995,5.300/.998,6.200/.999,7/1,8
```

0008	BUF1 STORAGE	20
0009	BUF2 STORAGE	20
0010	BUF3 STORAGE	20
0011	BUF4 STORAGE	20
0008	TAB1 TABLE M1,25,25,15	
0009	MET1 ENTER	BUF4,1
0010	SEIZE CPU1	
0015	ADVANCE	2
0020	LEAVE BUF4,1	
0025	RELEASE	CPU1
0030	ENTER BUF1,1	
0035	TEST E P2,1,MET4	
0040	TEST E P1,20,COP1	
0045	SPLIT 1,NEX1	
0046	MARK	
0050	COP1 MATCH	COP1
0055	LEAVE BUF1,1	
0060	ASSEMBLE	2
0065	TABULATE	TAB1
0070	TERMINATE	
0075	MET4 SEIZE	DCH1
0080	LEAVE BUF1,1	
0085	TRANSFER ,DEX1	0090 NEX1
	SEIZE DCH1	0095 DEX1 ADVANCE
	P1	
0100	RELEASE	DCH1
0105	ASSIGN 1,1	
0110	MET2 ENTER	BUF2,1
0115	SEIZE CPU2	
0120	ADVANCE	2
0125	LEAVE BUF2,1	
0130	RELEASE	CPU2
0135	ENTER BUF3,1	
0140	TEST E P2,2,MET5	
0145	TEST E P1,20,COP2	
0150	SPLIT 1,NEX2	
0155	COP2 MATCH	COP2
0160	LEAVE BUF3,1	
0165	TERMINATE	
0170	MET5 SEIZE	DCH2
0175	LEAVE BUF3,1	

```

0180 TRANSFER      ,DEX2 0185 NEX2
SEIZE      DCH2 0190 DEX2      ADVANCE
P1
0195      RELEASE      DCH2
0200      ASSIGN 1,1
0205      TRANSFER      ,MET1
0210 GEN2      GENERATE      25,FN$EXPON
0215      ASSIGN 1,20
0220      ASSIGN 2,2
0225      TRANSFER      ,MET2
0230 GEN1      GENERATE      25,FN$EXPON
0235      ASSIGN 1,20
0240      ASSIGN 2,1
0245      TRANSFER      ,MET1
0250      GENERATE      1,,10000
0255      TERMINATE      1
0260      START  1

```

При выполнении задания данной лабораторной работы необходимо добавить к тексту исходной GPSS-программы, приведенной выше, операторы, обеспечивающие при моделировании процесса передачи информации в СПД следующее:

- 1) определение функции распределения времени передачи пакетов сообщений между УК;
- 2) получение графиков загрузки выходных накопителей;
- 3) определение вероятностей переполнения накопителей;
- 4) получение соотношений пакетов и подтверждений во входных и выходных накопителях;
- 5) получение графика изменения длины очереди пакетов в выходном накопителе;
- 6) определение функций распределения времени передачи подтверждений между УК;
- 7) определение вероятности передачи пакетов из выходной очереди во входную при переполнении выходной очереди;
- 8) получение функции распределения времени ожидания подтверждения пакетом в выходной очереди.

Блок-диаграмма GPSS-модели процесса передачи приведена на рисунке 12.4.

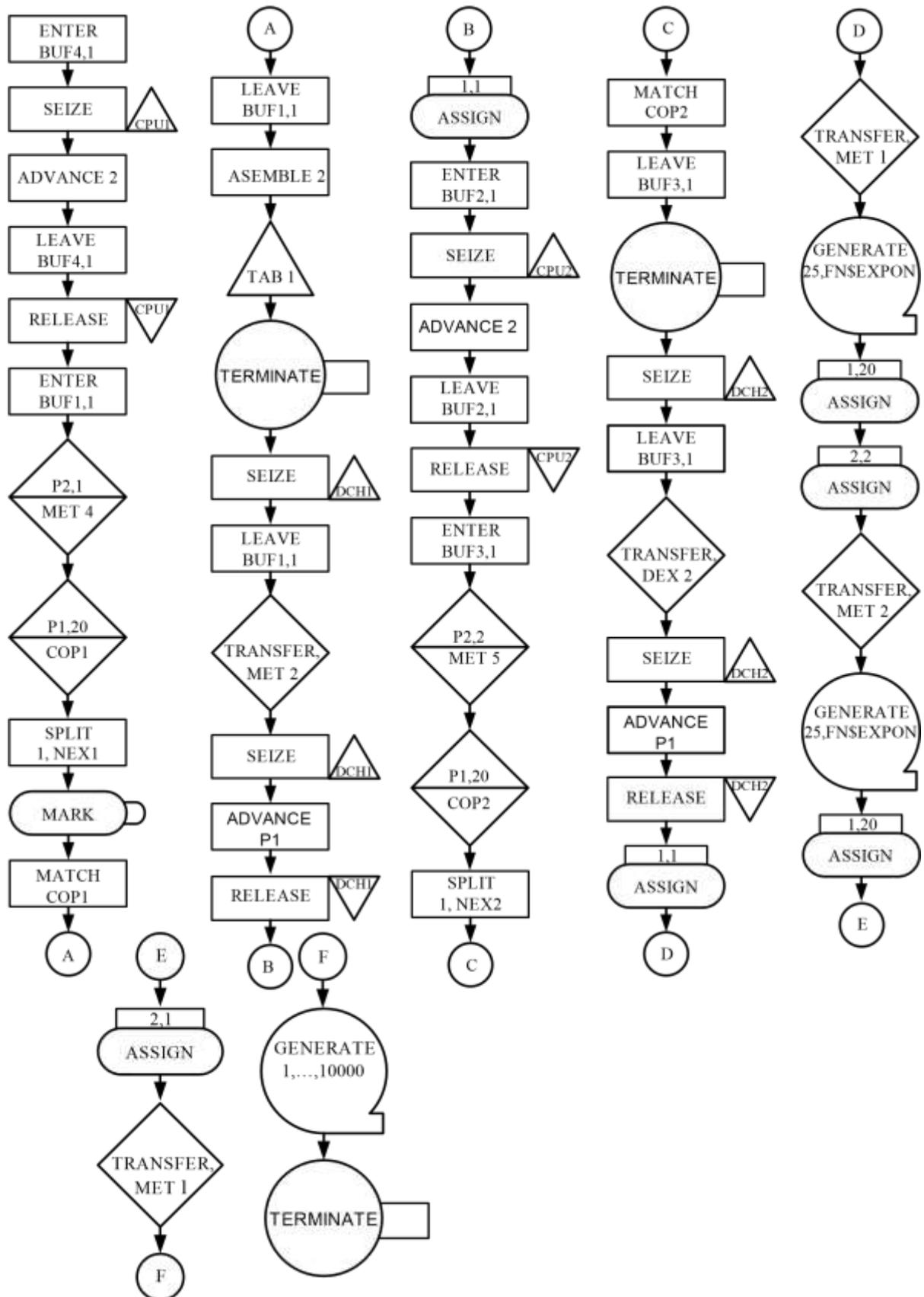


Рисунок 12.4 - Блок-диаграмма GPSS-модели процесса передачи
Порядок выполнения работы следующий.

1. Составить по заданному варианту блок-диаграмму GPSS и программу на языке GPSS.

2. Провести имитационный эксперимент на ПЭВМ с моделью процесса передачи данных для получения исследуемых вероятностно-временных характеристик.

Пример выполнения работы. Пусть необходимо выполнить п. 1 задания. Блок-диаграмма модели процесса передачи информации в СПД, представленной в виде Q-схемы показана на рисунке 12.4, а текст соответствующей GPSS-программы приведен ранее.

Гистограмма распределения времени передачи пакета данных между узлами УК1 и УК2 СПД показана на рисунке 12.5.

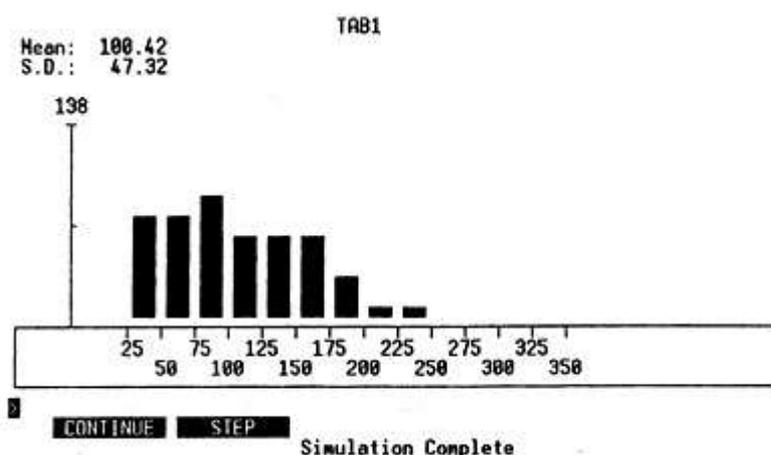


Рисунок 12.5 - Гистограмма распределения времени передачи пакета данных

Отчет о работе должен содержать:

- 1) задание и исходные данные по заданному варианту лабораторной работы;
- 2) краткое описание используемых технических и программных средств;
- 3) обоснование принимаемых решений;
- 4) текст результирующих программ и окончательные результаты их решения, выводы по результатам моделирования.

Листинг результатов моделирования приведен далее.

	START TIME		END TIME	BLOCKS	FACILITIES	STORAGES
0.000	10000.000	52	4	4		
	NAME		VALUE			
	BUF1		10001.000			
	BUF2		10002.000			
	BUF3		10003.000			
	BUF4		10004.000			
	COP1		11.000			
	COP2		32.000			
	CPU1		10006.000			
	CPU2		10008.000			
	DCH1		10007.000			
	DCH2		10009.000			
	DEX1		20.000			
	DEX2		39.000			
	GEN1		47.000			
	GEN2		43.000			
	MET1		1.000			
	MET2		23.000			
	MET4		16.000			
	MET5		35.000			
	NEX1		19.000			
	NEX2		38.000			
	TAB1		10005.000			
	XPDIS		10000.000			

LABEL	LOC	BLOCK	TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY
MET1	1	ENTER		1201		0		0
2		SEIZE	1201	0	0			
3		ADVANCE	1201	0	0			
4		LEAVE	1201	0	0			
5		RELEASE	1201	0	0			
6		ENTER	1201	0	0			
7		TEST	1201	0	0			
8		TEST	810	0	0			
9		SPLIT	406	0	0			
10		MARK	406	0	0			
COP1	11	MATCH		810		2		0
12		LEAVE	808	0	0			
13		ASSEMBLE	808	0	0			
14		TABULATE	404	0	0			
15		TERMINATE	404	0	0			
MET4	16	SEIZE		391		0		0
17		LEAVE	391	0	0			
18		TRANSFER	391	0	0			
NEX1	19	SEIZE		406		0		0
DEX1	20	ADVANCE		797		1		0
21		RELEASE	796	0	0			
22		ASSIGN	796	0	0			
MET2	23	ENTER		1189		1		0
24		SEIZE	1188	0	0			
25		ADVANCE	1188	1	0			
26		LEAVE	1187	0	0			
27		RELEASE	1187	0		0		28
		1187	0	0				ENTER
29		TEST	1187	1	0			
30		TEST	782	0	0			
31		SPLIT	393	0	0			
COP2	32	MATCH		782		4		0
33		LEAVE	778	0	0			
34		TERMINATE	778	0	0			
MET5	35	SEIZE		404		0		0

36	LEAVE		404		0		0				
37	TRANSFER		404		0		0				
NEX2		38	SEIZE			392		0		0	
DEX2		39	ADVANCE			796		1		0	
40	RELEASE		795		0		0				
41	ASSIGN		795		0		0				
42	TRANSFER		795		0		0				
GEN2		43	GENERATE			393		0		0	
44	ASSIGN		393		0		0				
45	ASSIGN		393		0		0				
46	TRANSFER		393		0		0				
GEN1		47	GENERATE			406		0		0	
48	ASSIGN		406		0		0				
49	ASSIGN		406		0		0				
50	TRANSFER		406		0		0				
51	GENERATE		1			0		0		52	
	TERMINATE		1			0		0			
FACILITY											
		ENTRIES	UTIL.	AVE.	TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
	CPU1	1201	0.240	2.000	1		0	0	0	0	0
	DCH1	797	0.849	10.654	1	1601	0	0	0	0	0
	CPU2	1188	0.238	2.000	1	1584	0	0	0	0	1
	DCH2	796	0.824	10.355	1	1597	0	0	0	0	2
STORAGE											
		CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
	BUF1	20	18	0	20	1201	1	5.944	0.297	0	0
	BUF2	20	18	0	5	1189	1	0.311	0.016	0	0
	BUF3	20	15	0	20	1187	1	5.714	0.286	0	0
	BUF4	20	20	0	5	1201	1	0.313	0.016	0	0
TABLE											
		MEAN	STD.DEV.			RANGE		RETRY	FREQUENCY	CUM.%	
	TAB1	109.786	54.912					0			
				25.000	-	50.000			58	14.36	
	50.000	-	75.000					68		31.19	
	75.000	-	100.000					72		49.01	
	100.000	-	125.000	58	63.37						
				125.000	-	150.000			53	76.49	
				150.000	-	175.000			40	86.39	
				175.000	-	200.000			27	93.07	
				200.000	-	225.000			15	96.78	
				225.000	-	250.000			11	99.50	
	250.000	-	275.000	2	100.00						
FEC XN PRI											
		BDT	ASSEM	CURRENT	NEXT	PARAMETER		VALUE			
	1598	0	10000.366	1598	0	43					
	1584	0	10000.368	1577	25	26	1	1.000			
							2		2.000		
	1602	0	10001.000	1602	0	51					
	1597	0	10001.757	1594	39	40	1	20.000			
							2	2.000			
	1601	0	10019.878	1591	20	21	1	20.000			
							2	1.000			
	1600	0	10022.566	1600	0	47					

13 Задание на лабораторную работу №11. Исследование на имитационной модели процесса функционирования локальной вычислительной сети

ЛВС состоит из сервера и 8 рабочих станций. Поток заявок от рабочей станции распределен по закону Пуассона со средним интервалом 15, 12, 17, 20,

10с. Время обработки заявки распределено по нормальному закону с математическим ожиданием 18, 16, 19, 23, 15 с и стандартным отклонением 3, 2, 3, 1,

2 с. Сервер обрабатывает одновременно до 8 заявок. На каждой рабочей станции создается очередь в 1 заявку, новые заявки от пользователей не принимаются рабочими станциями до передачи существующей заявки на сервер. Повторно заявка посылается на сервер через промежуток времени, распределенный по нормальному закону со средним 18 с и стандартным отклонением 3.

Пример выполнения задания. Текст GPSS-программы приведен далее.

```
SERVER STORAGE                8

GENERATE                      15, FN$EXPON
ASSIGN WS,1
TEST L  Q$QW1,1, REFUSE
QUEUE QW1
TRANSFER                        ,PROC

GENERATE                      12, FN$EXPON
ASSIGN WS,2
TEST L  Q$QW2,1, REFUSE
QUEUE QW2
TRANSFER                        ,PROC

GENERATE                      12, FN$EXPON
ASSIGN WS,6
TEST L  Q$QW6,1, REFUSE
QUEUE QW6
TRANSFER                        ,PROC

GENERATE                      17, FN$EXPON
ASSIGN WS,3
TEST L  Q$QW3,1, REFUSE
QUEUE QW3
TRANSFER                        ,PROC

GENERATE                      17, FN$EXPON
ASSIGN WS,7
TEST L  Q$QW7,1, REFUSE
QUEUE QW7
TRANSFER                        ,PROC
```

```

GENERATE                20, FN$EXPON
ASSIGN WS,4
TEST L  Q$QW4,1, REFUSE
  QUEUE QW4
TRANSFER                ,PROC
GENERATE                20, FN$EXPON
ASSIGN WS,8
TEST L  Q$QW8,1, REFUSE
  QUEUE QW8
TRANSFER                ,PROC

GENERATE                10, FN$EXPON
ASSIGN WS,5
TEST L  Q$QW5,1, REFUSE
  QUEUE QW5

PROC  GATE SF SERVER, PROCESS
      ADVANCE            (NORMAL(1,18,3))      ; retry delay
      TRANSFER          ,PROC
PROCESS  ENTER          SERVER

      TEST E  P$WS,1, TEST2
      DEPART QW1
      ADVANCE            (NORMAL(1,18,3))      ; 1
      TRANSFER          ,COMPLETED

TEST2  TEST E  P$WS,2, TEST6
      DEPART QW2
      TRANSFER          ,PROC2

TEST6  TEST E  P$WS,6, TEST3
      DEPART QW6

PROC2  ADVANCE            (NORMAL(1,16,2))      ; 2 и 6
      TRANSFER          ,COMPLETED

TEST3  TEST E  P$WS,3, TEST7
      DEPART QW3
      TRANSFER          ,PROC3

TEST7  TEST E  P$WS,7, TEST4
      DEPART QW7

```

```

PROC3  ADVANCE      (NORMAL(1,19,3))      ; 3 и 7
TRANSFER      ,COMPLETED

TEST4  TEST E  P$WS,4,TEST8
        DEPART QW4
        TRANSFER      ,PROC4

TEST8  TEST E  P$WS,8,PROC5
        DEPART QW8

PROC4  ADVANCE      (NORMAL(1,23,1))      ; 4 и 8
TRANSFER      ,COMPLETED

PROC5  DEPART QW5
        ADVANCE      (NORMAL(1,15,2))

COMPLETED  LEAVE      SERVER
            QUEUE PROCESSED      ; подсчёт обработанных
            TERMINATE

REFUSE QUEUE REFUGEE      ; подсчёт отброшенных
            TERMINATE

            GENERATE      28800      ; таймер
            TERMINATE      1

```

```

EXPON  FUNCTION  RN2,C24
0,0/0.1,0.104/0.2,0.222/0.3,0.355/0.4,0.509 0.5,0.69/0.6,0.915/0.7,1.2/0.75,1.38/.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/.9998,8

```

Оценить загрузку сервера за 8 часов работы. Подсчитать общее количество обработанных и необработанных заявок.

	START TIME	END TIME	BLOCKS	FACILITIES	STORAGES				
	0.000	28800.000	77	0	1				
QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY	
QW7	1	0	1146	761	0.315	7.916	23.562	0	
QW1	1	1	1279	866	0.324	7.299	22.604	0	
QW6	1	0	1444	951	0.409	8.161	23.905	0	
QW3	1	0	1175	792	0.306	7.497	23.001	0	
QW2	1	1	1524	1029	0.392	7.402	22.788	0	
QW5	1	0	1670	1147	0.412	7.097	22.660	0	

QW4	1	1	1044	688	0.286	7.881	23.113	0
REFUGEE	5530	5530	5530	0	2741.028	14275.157	14275.157	0
PROCESSED	10293	10293	10293	0	5125.742	14341.920	14341.920	0
QW8	1	0	1022	701	0.264	7.448	23.714	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
SERVER	8	0	0	8	10301	1	6.484	0.811	0	0

Оценить, при каком минимальном значении среднего интервала поступления заявок от первой станции сервер не справится с нагрузкой. Остальные параметры не изменять.

При среднем времени поступления заявок с $WS1 = 4$ с. число отброшенных заявок примерно равно числу обработанных (~10500).

Оценить, при каком максимальном значении среднего времени обслуживания заявок, поступающих от первой станции, сервер не справится с нагрузкой. Остальные параметры не изменять.

При среднем времени обработки заявок с $WS1 = 78$ с. число отброшенных заявок примерно равно числу обработанных (~7900).

Блок-диаграмма GPSS-модели приведена на рисунке 13.1.

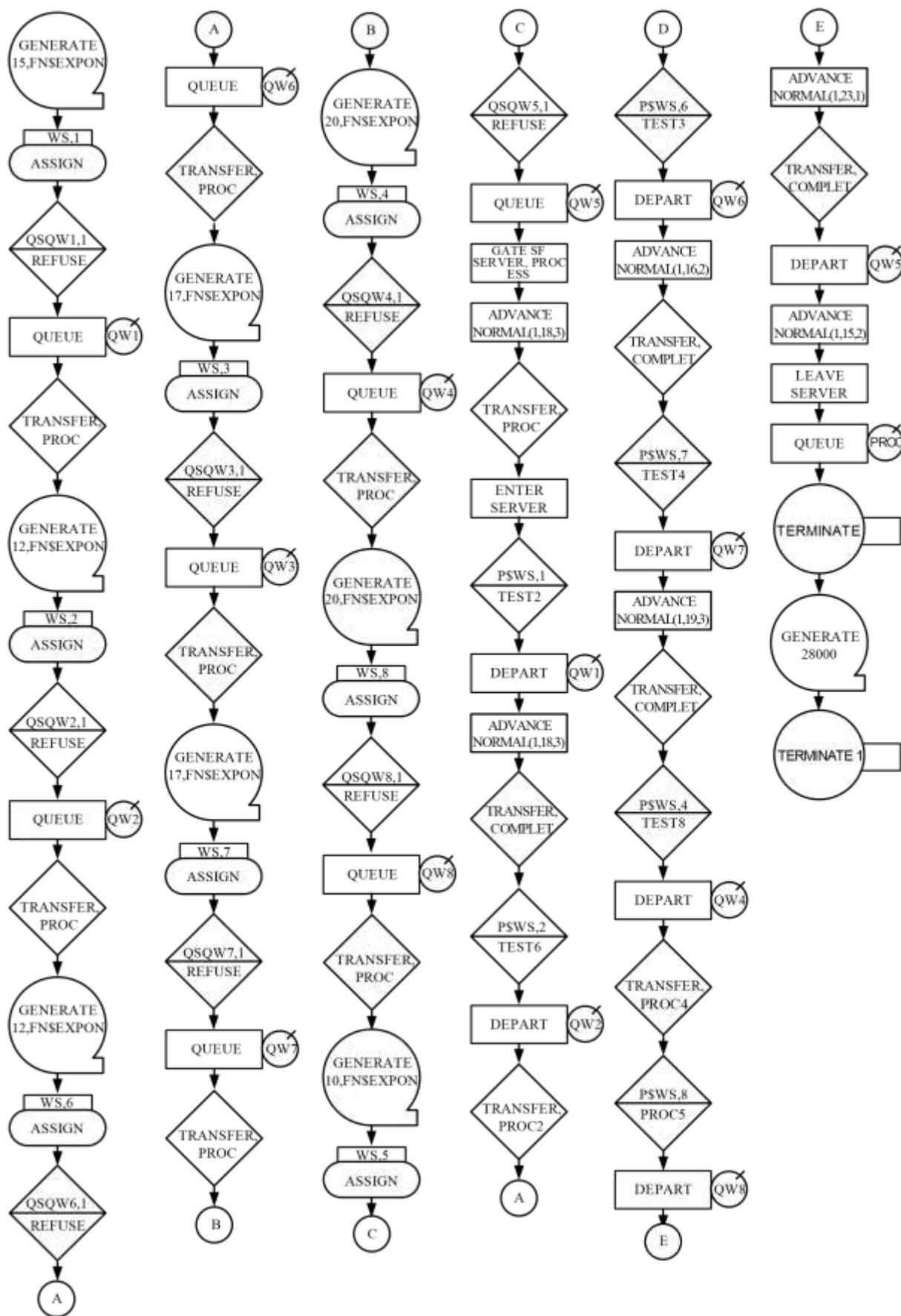


Рисунок 13.1 – Блок-диаграмма GPSS-модели

14 Задание на лабораторную работу №12. Исследование на имитационной модели процесса функционирования локальной вычислительной сети

Локальная вычислительная сеть (ЛВС) состоит из 8 рабочих станций, соединенных в кольцевую структуру. Поток заявок от каждой станции - пуассоновский со средними интервалами соответственно 10, 15, 18 и 20 секунд. По сети циркулирует маркер, поочередно дающий каждой станции право на использование канала для передачи данных. При поступлении маркера рабочая станция проверяет, поступил ли вместе с ним пакет. Если маркер пришел без пакета, то если станция имеет в этот момент свой пакет для передачи, то она передает его вместе с маркером. При отсутствии пакета для передачи маркер передается на следующую станцию. Если маркер пришел с пакетом, то станция анализирует пакет, сравнивая его адрес отправителя со своим адресом. Если эти адреса совпадают, то станция анализирует пакет, прошедший через все кольцо, с хранимым у нее экземпляром того же пакета. Время анализа распределено по равномерному закону со средним значением 10 ± 2 секунды. Вероятность искажения пакета $= 0,1$. Если пакет искажен, станция передает его повторно вместе с маркером. Если не искажен, транслируется маркер без пакета. Время передачи пакета между станциями распределено по нормальному закону с мат. ожиданием 5 секунд и со стандартным отклонением 2 секунды.

Из-за ограничения в 150 блоков, связанного с GPSS World версии Student, удалось реализовать только 7 из 8 станций.

Текст GPSS программы приведен далее.

```
GENERATE          ,,1      ; маркер
ASSIGN DATA,0    ; без пакета
ASSIGN HOST,0     ; без адреса
ASSIGN ERR,0.1    ; вероятность ошибки

; 1 станция
WKS1  TEST E  P$DATA,1,FREETK1 ; маркер с данными?
      TEST E  P$HOST,1,WKS2    ; мои данные?
      ADVANCE          10,2     ; анализ
      TRANSFER      P$ERR,,WKS2 ; повторная передача с вероятностью
ERR
```

```

        ASSIGN DATA,0                ; освобождаем маркер
        ASSIGN HOST,0
        QUEUE SERVICED
TRANSFER      ,WKS2                ; передача пустого маркера FREETK1 GATE
SNE PKTBUF1,WKS2                    ; есть ли данные для передачи
        LEAVE  PKTBUF1                ; забрали из буфера
        ASSIGN DATA,1                ; маркер с данными
        ASSIGN HOST,1                ; с нашим адресом
TRANSFER      ,WKS2                ; ушёл по кольцу

GENERATE станции                    10,FN$EXPON                ; пакеты от первой
GATE SNF буфера на                  PKTBUF1,REFUSE                ; проверка
занятость

ENTER  PKTBUF1                        ; пакет помещается в буфер
TERMINATE

; 2 станция
WKS2  ADVANCE                          (NORMAL(1,5,1))                ; задержка пе-
редачи
        TEST E  P$DATA,1,FREETK2
        TEST E  P$HOST,2,WKS3
        ADVANCE                          10,2
        TRANSFER      P$ERR,,WKS3
ASSIGN DATA,0
ASSIGN HOST,0
        QUEUE SERVICED
TRANSFER      ,WKS3
FREETK2 GATE SNE                      PKTBUF2,WKS3
LEAVE PKTBUF2 ASSIGN DATA,1
        ASSIGN HOST,2
TRANSFER      ,WKS3

GENERATE                          15,FN$EXPON
GATE SNF                          PKTBUF2,REFUSE
ENTER  PKTBUF2
TERMINATE

; 3 станция
WKS3  ADVANCE                          (NORMAL(1,5,1))

```

```

TEST E  P$DATA,1,FREETK3
TEST E  P$HOST,3,WKS4
ADVANCE                10,2
TRANSFER                P$ERR,,WKS4
ASSIGN DATA,0
ASSIGN HOST,0
    QUEUE SERVICED
TRANSFER                ,WKS4
FREETK3 GATE SNE       PKTBUF3,WKS4
LEAVE PKTBUF3  ASSIGN
DATA,1  ASSIGN HOST,3

TRANSFER                ,WKS4

GENERATE                18,FN$EXPON
GATE SNF                PKTBUF3,REFUSE
ENTER  PKTBUF3
TERMINATE

; 4 станция
WKS4  ADVANCE                (NORMAL(1,5,1))
TEST E  P$DATA,1,FREETK4  TEST
E  P$HOST,4,WKS5
ADVANCE                10,2
TRANSFER                P$ERR,,WKS5
ASSIGN DATA,0
ASSIGN HOST,0
    QUEUE SERVICED
TRANSFER                ,WKS5
FREETK4 GATE SNE       PKTBUF4,WKS5
LEAVE PKTBUF4  ASSIGN DATA,1
ASSIGN HOST,4
TRANSFER                ,WKS5

GENERATE                20,FN$EXPON
GATE SNF                PKTBUF4,REFUSE
ENTER  PKTBUF4
TERMINATE

; 5 станция
WKS5  ADVANCE                (NORMAL(1,5,1))

```

```

TEST E P$DATA,1,FREETK5 TEST
E P$HOST,5,WKS6
    ADVANCE 10,2
    TRANSFER P$ERR,,WKS6
    ASSIGN DATA,0
ASSIGN HOST,0
    QUEUE SERVICED
    TRANSFER ,WKS6
FREETK5 GATE SNE PKTBUF5,WKS6
LEAVE PKTBUF5 ASSIGN DATA,1
    ASSIGN HOST,5
    TRANSFER ,WKS6

    GENERATE 10,FN$EXPON
    GATE SNF PKTBUF5,REFUSE
    ENTER PKTBUF5
    TERMINATE

```

; 6 станция

```

WKS6 ADVANCE (NORMAL(1,5,1))
TEST E P$DATA,1,FREETK6 TEST
E P$HOST,6,WKS7
    ADVANCE 10,2
    TRANSFER P$ERR,,WKS7
    ASSIGN DATA,0
ASSIGN HOST,0
    QUEUE SERVICED
    TRANSFER ,WKS7
FREETK6 GATE SNE PKTBUF6,WKS7
LEAVE PKTBUF6 ASSIGN DATA,1
    ASSIGN HOST,6
    TRANSFER ,WKS7

    GENERATE 15,FN$EXPON
    GATE SNF PKTBUF6,REFUSE
    ENTER PKTBUF6
    TERMINATE

```

; 7 станция

```

WKS7 ADVANCE (NORMAL(1,5,1))

```

```

TEST E P$DATA,1,FREETK7 TEST
E P$HOST,7,WKS8
    ADVANCE 10,2
    TRANSFER P$ERR,,WKS8
    ASSIGN DATA,0
ASSIGN HOST,0
    QUEUE SERVICED
    TRANSFER ,WKS8
FREETK7 GATE SNE PKTBUF7,WKS8 LEAVE PKTBUF7 ASSIGN DATA,1
    ASSIGN HOST,7
    TRANSFER ,WKS8

    GENERATE 18,FN$EXPON
    GATE SNF PKTBUF7,REFUSE
    ENTER PKTBUF7
    TERMINATE

WKS8 ADVANCE (NORMAL(1,5,1))
    TRANSFER ,WKS1 ; маркер по кольцу

```

```

REFUSE QUEUE REFUGEE
    TERMINATE

```

; таймер

```

GENERATE 28800
TERMINATE 1

```

```

PKTBUF1 STORAGE 1
PKTBUF2 STORAGE 1
PKTBUF3 STORAGE 1
PKTBUF4 STORAGE 1
PKTBUF5 STORAGE 1
PKTBUF6 STORAGE 1
PKTBUF7 STORAGE 1
PKTBUF8 STORAGE 1

```

```

EXPON FUNCTION RN1,C24

```

```

0,0/0.1,0.104/0.2,0.222/0.3,0.355/0.4,0.509

```

```

0.5,0.69/0.6,0.915/0.7,1.2/0.75,1.38/.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/.9998,8

```

Листинг результатов моделирования приведен далее.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES							
0.000	28800.000	135	0	8							
QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY				
REFUGEE	13995	13995	13995	0	7019.569	14445.416	14445.416	0			
SERVICED	127	127	127	0	63.962	14504.838	14504.838	0			
STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY	
PKTBUF1	1	0	0	1	19	1	0.994	0.994	0	0	
PKTBUF2	1	0	0	1	19	1	0.992	0.992	0	0	
PKTBUF3	1	0	0	1	19	1	0.988	0.988	0	0	
PKTBUF4	1	0	0	1	19	1	0.992	0.992	0	0	
PKTBUF5	1	0	0	1	20	1	0.993	0.993	0	0	
PKTBUF6	1	0	0	1	20	1	0.993	0.993	0	0	
PKTBUF7	1	0	0	1	19	1	0.985	0.985	0	0	
PKTBUF8	1	1	0	0	0	1	0.000	0.000	0	0	

Определить параметры сети при увеличении вероятности искажения пакета в 8 раз ($P\$ERR = 0.8$).

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES							
0.000	28800.000	135	0	8							
QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY				
REFUGEE	13995	13995	13995	0	7019.569	14445.416	14445.416	0			
SERVICED	127	127	127	0	63.962	14504.838	14504.838	0			

Так как программа модели состоит из групп блоков, а каждая из групп необходима для моделирования работы рабочей станции, то на рисунке 14.1 приведена блок-диаграмма моделирования работы одной рабочей станции.

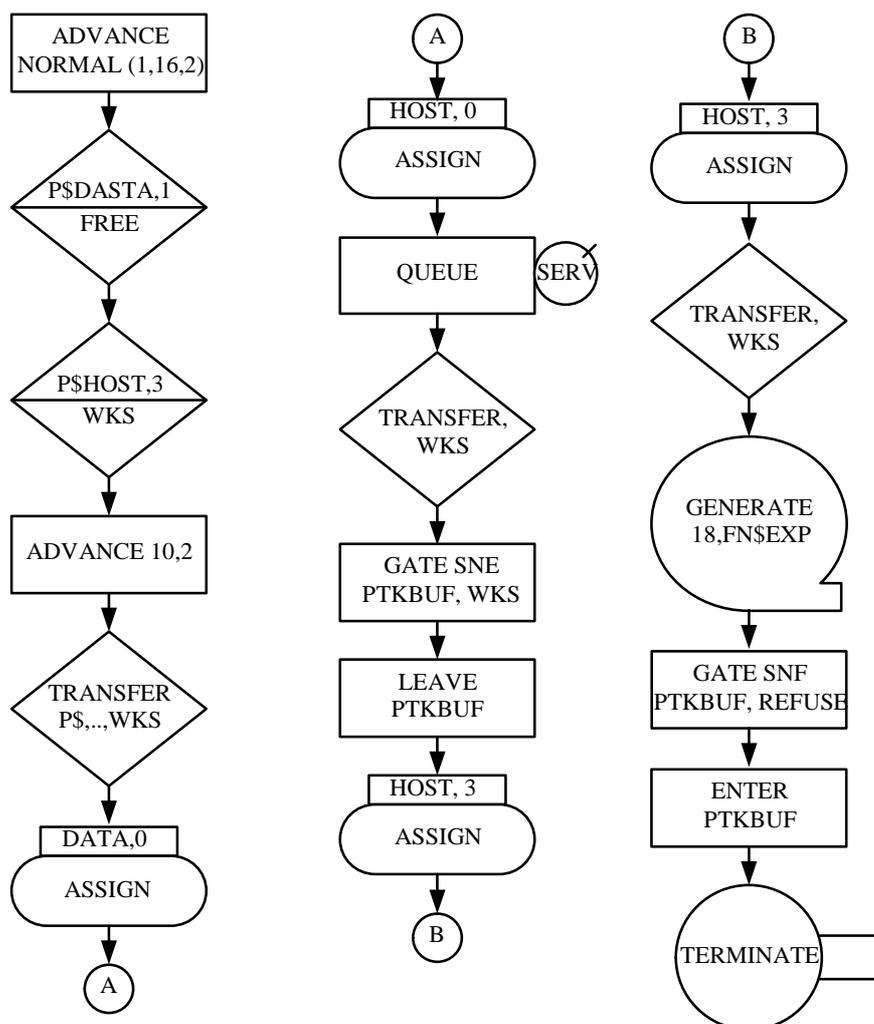


Рисунок 14.1 - Блок-диаграмма моделирования работы рабочей станции

Список использованных источников

- 1 **Авен О.И.** Оценка качества и оптимизация вычислительных систем [Текст] / О.И.Авен, Н.Н. Гурин, Я.А.Коган.- М.: Наука, 1982.-464 с.
- 2 **Бусленко Н.П.** Моделирование сложных систем [Текст]/ Н.П. Бусленко.–М.: Наука, 1978.-399 с.
- 3 **Гнеденко Б.В.** Введение в теорию массового обслуживания [Текст] / Б.В. Гнеденко, И.Н. Коваленко.- М.: Наука, 1987. - 431 с.
- 4 **Ивницкий В.А.** Сети массового обслуживания и их применение в ЭВМ [Текст]/В.А.Ивницкий//Зарубежная радиоэлектроника.-1977.-№7.-С. 33-70.
- 5 **Клейнрок Л.** Вычислительные системы с очередями [Текст]/ Л. Клейнрок; [пер. с англ.]/ Под ред. Б.С. Цыбакова. - М.: Мир, 1979. - 597 с.
- 6 **Клейнрок Л.** Теория массового обслуживания [Текст]/ Л. Клейнрок; [пер. с англ.] / Под ред. В.И.Неймана. – М.: Машиностроение, 1979. - 432 с.
- 7 **Компьютерные системы и технологии [Текст]: лабораторный практикум /Под**

- ред. Л.Д. Забродина. -М.:Диалог-МИФИ, 2001.-336с.
- 8 Основы теории вычислительных систем [Текст] / Под ред. С.А. Майорова. - М.: Высшая школа, 1978. - 408 с.
 - 9 **Павловский Ю.Н.** Имитационные модели и системы [Текст]/Ю.Н. Павловский.– М.: Фазис: ВЦ РАН, 2000. – 134 с.
 - 10 **Прохоров С.А.** Моделирование и анализ случайных процессов [Текст]: лабораторный практикум.-2-е изд., доп. и перераб.- СНЦ РАН, 2002.-277с.
 - 11 **Советов Б.Я.** Моделирование систем [Текст]: практикум; учеб. пособие для вузов / Советов Б.Я., Яковлев С.А.-М.: Высшая школа, 2003.-295с.
 - 12 **Тарасов В.Н.** Вероятностное компьютерное моделирование сложных систем [Текст]/В.Н. Тарасов.–Самара: Самарский научный центр РАН, 2002.-194 с.
 - 13 **Тарасов В.Н.** Организация интерактивной системы вероятностного моделирования стохастических систем [Текст] / В.Н. Тарасов, Н.Ф. Бахарева. // Известия Самарского научного центра РАН.- 2003.- №1 -С. 119 – 126.
 - 14 **Тарасов В.Н.** Компьютерное моделирование вычислительных систем. Теория, алгоритмы, программы [Текст] / В.Н. Тарасов, Н.Ф. Бахарева. - Оренбург.: ИПК ОГУ, 2005.-183 с.
 - 15 **Тихонов В.И.** Марковские процессы. [Текст] / В.И. Тихонов, М.А. Миронов - М.: Сов. радио, 1977.- 488 с.
 - 16 **Феррари Д.** Оценка производительности вычислительных систем [Текст]/ Д.Феррари; [пер.англ.] / Под ред. В.В.Мартынюка. - М.: Мир, 1981. - 576 с.
 - 17 **Шварц М.** Сети ЭВМ. Анализ и проектирование [Текст]/ М.Шварц; [пер. с англ.] / Под ред. В.А. Жожикашвили. - М.: Радио и связь, 1982. - 336 с.
 - 18 **Шеннон Р.** Имитационное моделирование систем-искусство и наука [Текст]/ Р.Шеннон.-М.: Мир, 1978.-272 с.
 - 19 **Шрайбер Т.Дж.** Моделирование на GPSS. [Текст]/ Т.Дж.Шрайбер [пер. с англ.]. –М.: Машиностроение, 1980.-576 с.

Приложение

A

(обязательное)

Таблица А.1-таблица критических точек распределения χ^2

Число степеней свободы k	Уровень значимости α					
	0,01	0,025	0,05	0,95	0,975	0,99
1	6,6	5,0	3,8	0,0039	0,00098	0,00016
2	9,2	7,4	6,0	0,103	0,051	0,020
3	11,3	9,4	7,8	0,352	0,216	0,115
4	13,3	11,1	9,5	0,711	0,484	0,297
5	15,1	12,8	11,1	1,15	0,831	0,554
6	16,8	14,4	12,6	1,64	1,24	0,872
7	18,5	16,0	14,1	2,17	1,69	1,24
8	20,1	17,5	15,5	2,73	2,18	1,05
9	21,7	19,0	16,9	3,33	2,70	2,09
10	23,2	20,5	18,3	3,94	3,25	2,56
11	24,7	21,9	19,7	4,57	3,82	3,05
12	26,2	23,3	21,0	5,23	4,40	3,57
13	27,7	24,7	22,4	5,89	5,01	4,11
14	29,1	26,1	23,7	6,57	5,63	4,66
15	30,6	27,5	25,0	7,26	6,26	5,23
16	32,0	28,8	26,3	7,96	6,91	5,81
17	33,4	30,2	27,6	8,67	7,56	6,41
18	34,8	31,5	28,9	9,39	8,23	7,01
19	36,2	32,9	30,1	10,1	8,91	7,63
20	37,6	34,2	31,4	10,9	9,59	8,26
21	38,9	35,5	32,7	11,6	10,3	8,90
22	40,3	36,8	33,9	12,3	11,0	9,54
23	41,6	38,1	35,2	13,1	11,7	10,2
24	43,0	39,4	36,4	13,8	12,4	10,9
25	44,3	40,6	37,7	14,6	13,1	11,5
26	45,6	41,9	38,9	15,4	13,8	12,2
27	47,0	43,2	40,1	16,2	14,6	12,9
28	48,3	44,5	41,3	16,9	15,3	13,6
29	49,6	45,7	42,6	17,7	16,0	14,3

30

Б

(о б я з а т е л ь н о е)

Таблиц Б.1

таблиц

а значений коэффициента λ к критерию Колмогорова

$\delta \backslash N$	500	1000	2000	5000	8000	10000
0,002	0,0559	0,0790	0,1118	0,1767	0,2236	0,25
0,005	0,1118	0,1581	0,2236	0,3535	0,4472	0,5
0,01	0,2236	0,3162	0,4472	0,7071	0,8944	1
0,02	0,4472	0,6324	0,8944	1,4142	1,7889	2
0,03	0,6708	0,9486	1,3416	2,1213	2,6833	3

Приложение

0,04	0,8972	1,2648	1,7888	2,8284	3,5777	4
0,05	1,1208	1,5810	2,2360	3,5355	4,4721	5

Приложение

В

(рекомендуемое)

Исходный текст программы к заданию №1

```
unit Unit2; interface uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Math; type
  TFrame2 = class(TFrame)
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
  Edit2: TEdit;  Button1:
  TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end; implementation
uses Unit1; {$R *.dfm}
procedure TFrame2.Button1Click(Sender: TObject);var
  x,fx,r:array[1..10000]of real;
  pjk,pj,lm,pm:array[1..30]of real;
  i,ll,lp,n,j,gd,gm,x0:integer;  col:array[1..30]
of integer;
  ep,sig,mm2,mm3,m2,m3,m4,ass,ast,eks,ekt,dx,xs,lamz,dx1,sum,xi2,sum1,max,min,ddx:real;
  va,vb,ers,lam:integer;
  s,tl,slam,spi,tp:string; begin
with Form1 do begin
randomize;
n:=strtoint(edit1.Text);
for i:=1 to n do begin
x[i]:=0;  fx[i]:=0;
r[i]:=0;
      end;
for i:=1 to 30 do begin
pjk[i]:=0; pj[i]:=0;
col[i]:=0
      end;
dx:=0; xs:=0;
```

```

lamz:=0;
dx1:=0;
sum:=0; xi2:=0;
sum1:=0;
max:=0;
min:=0;
ddx:=0;
s:="";
for i:=1 to n do r[i]:=random;
case ComboBox1.ItemIndex of
0: begin
    va:=strtoint(Frame21.Edit1.Text);
    vb:=strtoint(Frame21.Edit2.Text);
    for i:=1 to n do x[i]:=va+(vb-v)*r[i]; {равномерный}
end;
1: begin {нормальный}
    xs:=strtofloat(Frame21.Edit1.Text);
    sig:=strtofloat(Frame21.Edit2.Text); for
i:=1 to n do begin
    for j:=1 to 12 do x[i]:=x[i]+random;
x[i]:=x[i]-6; x[i]:=xs+sig*x[i];
end;
end;
2: for i:=1 to n do x[i]:=ln(r[i])/((-1)*strtoint(Frame21.Edit1.Text)); {экспоненциальный}
3: begin {Эрланга}
    ers:=strtoint(Frame21.Edit2.Text); for
i:=1 to n do
    for j:=1 to ers-1 do r[i]:=r[i]+random;
    for i:=1 to n do x[i]:=ln(r[i])/((-1)*strtoint(Frame21.Edit1.Text));
end;
4: begin {гиперэкспоненциальный}
    ll:=0;
    tl:="";
    for i:=1 to n do lm[i]:=0;
    slam:=Frame21.Edit1.Text;
    spi:=Frame21.Edit2.Text;
    {Ввод вектора} for i:=1
to length(slam) do case
slam[i] of
    ',' ':begin
ll:=ll+1;
    lm[ll]:=strtofloat(tl);
    tl:=""; end;
'0'..'9',',',':':tl:=tl+slam[i];
end; if tl<>" then begin
ll:=ll+1;

```

```

lm[l1]:=strtofloat(tl);
end;    lp:=0;
    tp="";
for i:=1 to length(spi) do case
spi[i] of
    ',' ':begin
lp:=lp+1;
    pm[lp]:=strtofloat(tp);
tp="";    end;
'0'..'9',',',':':tp:=tp+spi[i];
end;
if tp<>" then begin
lp:=lp+1;
    pm[lp]:=strtofloat(tp);
end;
    {_____}
    for i:=1 to n do
for j:=1 to ll do
    x[i]:=x[i]+pm[j]*ln(random)/((-1)*lm[j]);
end; end; xs:=0; m2:=0; m3:=0; m4:=0;
    case ComboBox1.ItemIndex of
0: begin
    LamT.Caption:='a1: '+floattostf((va+vb)/2,ffGeneral,3,3);
    M2T.Caption:='m2: '+ floattostf(sqr(vb-va)/12,ffGeneral,3,3);
    M3T.Caption:='m3: 0';
    M4T.Caption:='m4: '+ floattostf(power((vb-va),4)/80,ffGeneral,3,3);
    Label10.Caption:='as: 0';
Label11.Caption:='ek: -1,2';    end;
1: begin
    LamT.Caption:='a1: '+Frame21.Edit1.Text;
    M2T.Caption:='m2: '+Frame21.Edit2.Text;
    M3T.Caption:='m3: 0';
    M4T.Caption:='m4: '+ floattostf(3*power(sig,4)/80,ffGeneral,3,3);
    Label10.Caption:='as: 0';
Label11.Caption:='ek: 0';    end;
2: begin
    LamT.Caption:='a1: '+ floattostf(1/strtoint(Frame21.Edit1.Text),ffGeneral,3,3);
    M2T.Caption:='m2: '+ floattostf(1/sqr(strtoint(Frame21.Edit1.Text)),ffGeneral,3,3);
    M3T.Caption:='m3: '+ floattostf(2/(power(strtoint(Frame21.Edit1.Text),3)),ffGeneral,3,3);
M4T.Caption:='m4: '+ floattostf(9/(power(strtoint(Frame21.Edit1.Text),4)),ffGeneral,3,3);
Label10.Caption:='as: 2';
    Label11.Caption:='ek: 6';
end; 3: begin
    sig:=strtoint(Frame21.Edit1.Text);
ers:=ers+1;
    LamT.Caption:='a1: '+floattostf(ers/sig,ffGeneral,3,3);
    M2T.Caption:='m2: '+floattostf(ers/sqr(sig),ffGeneral,3,3);

```

```

M3T.Caption:='m3: '+floattstrf(2*ers/power(sig,3),ffGeneral,3,3);
M4T.Caption:='m4: '+floattstrf(3*ers*(ers+2)/power(sig,4),ffGeneral,3,3);
Label10.Caption:='as: '+floattstrf(2/sqrt(ers),ffGeneral,3,3);
Label11.Caption:='ek: '+floattstrf(6/sig,ffGeneral,3,3); end;
4: begin
  for i:=1 to ll do xs:=xs+pm[i]/lm[i];
  LamT.Caption:='a1: '+floattstrf(xs,ffGeneral,3,3);
for i:=1 to ll do mm2:=mm2+pm[i]/sqr(lm[i]);
m2:=2*mm2-sqr(xs);
  M2T.Caption:='m2: '+floattstrf(m2,ffGeneral,3,3);
for i:=1 to ll do mm3:=mm3+pm[i]/power(lm[i],3);
m3:=6*(mm3-xs*mm2)+2*sqr(xs);
  M3T.Caption:='m3: '+floattstrf(m3,ffGeneral,3,3);
for i:=1 to ll do m4:=m4+pm[i]/power(lm[i],4);
m4:=24*(m4-mm3*xs)+3*sqr(xs)*(4*mm2-sqr(xs));
  M4T.Caption:='m4: '+floattstrf(m4,ffGeneral,3,3);
  Label10.Caption:='as: '+floattstrf(m3/power(m2,1.5),ffGeneral,3,3);
Label11.Caption:='ek: '+floattstrf(m4/power(m2,2)-3,ffGeneral,3,3);
end; end; xs:=0; m2:=0; m3:=0; m4:=0;
{ _____nach_mom_____ }
for i:=1 to n do xs:=xs+x[i]; xs:=xs/n;
lamz:=1/xs;
LamS.Caption:='a1: '+floattstrf(xs,ffGeneral,3,3); { _____centr_mom_____ }
for i:=1 to n do m2:=m2+(sqr(x[i]-xs)/n);
sig:=sqrt(abs(m2));
M2S.Caption:='m2: '+ floattstrf(m2,ffGeneral,3,3);
for i:=1 to n do m3:=m3+power((x[i]-xs),3)/n;
M3S.Caption:='m3: '+ floattstrf(m3,ffGeneral,3,3);
ass:=m3;
for i:=1 to n do m4:=m4+power((x[i]-xs),4)/n;
M4S.Caption:='m4: '+ floattstrf(m4,ffGeneral,3,3); eks:=m4;
{ _____spec_____ }
ass:=ass/power(sig,3);
Label15.Caption:='as: '+ floattstrf(ass,ffGeneral,3,3); eks:=eks/power(sig,4)-3;
Label16.Caption:='ek: '+ floattstrf(eks,ffGeneral,3,3);
max:=x[1]; min:=x[1]; for i:=2 to n do begin if
x[i]>max then max:=x[i] else if x[i]<min then
min:=x[i]; end;
ddx:=(max-min)/20;
dx:=min; j:=1;
for i:=1 to n do begin if
x[i]=max then inc(col[20]) else
j:=(trunc(x[i]*100000)div trunc(ddx*100000))+1;
inc(col[j]);
end;
{ while dx<max do begin
for i:=1 to n do

```

```

if (x[i]>=dx)and(x[i]<=dx+ddx) then col[j]:=col[j]+1;
j:=j+1; dx:=dx+ddx; end; dx:=min;
s:="";
memo1.lines.Clear; for
i:=1 to 20 do begin
s:=floattostrf(dx,ffGeneral,3,3)+'-' +floattostrf(dx+ddx,ffGeneral,3,3)+' '+inttostr(col[i]);
memo1.Lines.Add(s); dx:=dx+ddx;
end; } for
i:=1 to 20 do
pjk[i]:=col[i]/n; with
series2 do clear;
for i:=1 to 20 do chart2.Series[0].AddY(pjk[i]);
sum:=0; sum1:=0;
for i:=1 to 20 do sum:=sum+pjk[i]; for
i:=1 to 20 do sum1:=sum1+col[i];
{ label6.Caption:=(floattostr(sum));
label7.Caption:=(floattostr(sum1));}
dx:=0;
i:=1;
while dx<=1 do begin
pj[i]:=(exp((-1)*lamz*dx)-exp((-1)*lamz*(dx+0.05)));
dx:=dx+0.05; i:=i+1; end; xi2:=0;
for i:=1 to 20 do xi2:=xi2+sqr(n*pjk[i]-n*pj[i])/(n*pj[i]);
{ label6.Caption:='Согласно критерию Пирсона:';
if xi2>28.9 then label6.Caption:=label6.Caption+'Гипотеза отвергнута' else
label6.Caption:=label6.Caption+'Нет оснований отвергнуть гипотезу';}
Frame21.Visible:=False;
end;
end;

end.

```

Приложение Г (рекомендуемое)

Исходный текст программы к заданию №2

```
unit Dip1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, TeeProcs, TeEngine, Chart, ComCtrls, Series,
  ImgList, Math; type
  TForm1 = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
  Button1: TButton;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
  Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    GroupBox2: TGroupBox;
    Label5: TLabel;
    LRo: TLabel;
    Label7: TLabel;
    LW: TLabel;
    Label6: TLabel;
    LNq: TLabel;
    Label8: TLabel;
    LNsr: TLabel;
    Label9: TLabel;
    TabSheet2: TTabSheet;
  Chart1: TChart;
    Series1: TLineSeries;
    ImageList1: TImageList;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Series2: TLineSeries;
  LPn: TEdit;
```

```

    procedure Button1Click(Sender: TObject);
    procedure PageControl1Changing(Sender: TObject;
    var AllowChange: Boolean); procedure
    FormActivate(Sender: TObject); private
    { Private declarations }
    public
    { Public declarations }
    end; var
    Form1: TForm1;
    lamm,mmu:real;
    j,i,mm:integer; P:
    array[0..20] of real;
    Ro,W,Nq,Nsr,u,buf:real;
    procedure SMO(lam,mu:real;m,k:integer; var Ro,w,Nq,N:real; var P: array of real); ex-
    ternal 'SMO.dll';
    implementation {$R
    *.dfm}
    function strtoval(st:string):real;
    var rl:real; j:integer; begin
    i:=1; rl:=0;
    while i<=length(st) do
    begin
    if (st[i]<>'.')and(st[i]<>',') then begin
    rl:=rl*10+strtoint(st[i]); i:=i+1
    end
    else begin j:=i;
    i:=i+1;
    while i<=length(st) do begin
    rl:=rl+strtoint(st[i])/power(10,i-j); i:=i+1
    end
    end end; strtoval:=rl
    end;
    procedure TForm1.Button1Click(Sender: TObject);
    var k,m:integer; begin LPn.Text:=""; for i:=1 to 20
    do P[i]:=0;
    if (Edit1.Text<>")and(Edit2.Text<>")and(Edit3.Text<>")and(Edit4.Text<>") then
    begin
    k:=strtoint(Edit4.Text);
    m:=strtoint(Edit3.Text);
    if (k<>0)and(k<>m) then
    begin m:=1;
    Edit3.Text:='1';
    end;
    SMO(strtival(Edit1.Text),strtival(Edit2.Text),m,k,Ro,W,Nq,Nsr,P);
    LRo.Caption:=Currtostr(Ro);
    LW.Caption:=Currtostr(W);

```

```

LNq.Caption:=Currtostr(Nq);
LNsr.Caption:=Currtostr(Nsr);
i:=0;
while i<=20 do begin
if P[i]<>0 then
LPn.Text:=LPn.Text+'P['+inttostr(i)+']='+Currtostr(P[i])+'; ';
i:=i+1
end
end
else ShowMessage('Недостаточно параметров!'); end;

procedure TForm1.PageControl1Changing(Sender: TObject;
var AllowChange: Boolean); begin
if not((Edit1.Text<>")and(Edit2.Text<>")and(Edit3.Text<>")and(Edit4.Text<>")) then
begin
ShowMessage('Недостаточно параметров для построения графика');
end else begin
Chart1.Title.Text.Clear;
Chart1.Title.Text.Add('График зависимости времени ожидания в очереди');
Chart1.Title.Text.Add(' от коэффициента использования ');
Chart1.Series[0].Clear;
Chart1.Series[1].Clear;
label12.Caption:=Edit4.Text;
j:=1;
while Ro<0.9 do begin
SMO(0.01*j,strtval(Edit2.Text),strtoint(Edit3.Text),strtoint(Edit4.Text),Ro,W,Nq,Nsr,P);
Chart1.Series[0].AddXY(Ro,W);
j:=j+1;
end;
j:=1; Ro:=0;
while Ro<0.9 do begin
SMO(0.01*j,strtval(Edit2.Text),strtoint(Edit3.Text),0,Ro,W,Nq,Nsr,P);
Chart1.Series[1].AddXY(Ro,W);
j:=j+1;
end;
end end;

procedure TForm1.FormActivate(Sender: TObject); begin
TabSheet1.Caption:='Вычисления';
TabSheet2.Caption:='График'; end;

end.

```

В программе используется библиотечный модуль, содержащий процедуру SMO.
Текст модуля приведен далее:

```

library SMO;

uses
SysUtils,
Classes;
{$R *.res}

function fuct(n:integer):integer;
var rez,i:integer; begin rez:=1;
for i:=1 to n do rez:=rez*i; fuct:=rez;
end;

function step(x:real;y:integer):real;
var rezult:real; begin
rezult:=exp(y*ln(x));
step:=rezult; end;

procedure MySMO(lam,mu:real;m,k:integer; var Ro,w,Nq,N:real; var P: array of real);
var i:integer;
    u:real; begin
ro:=lam/(m*mu);
{_____M/M/m/..._____}
if k=0 then          begin
p[0]:=0;
for i:=0 to m-1 do begin
p[0]:=p[0]+(step(m*Ro,i)/fuct(i));
p[0]:=p[0]+(step(m*Ro,m)/(fuct(m)*(1-Ro)));
end;
p[0]:=1/p[0];
i:=0; repeat
i:=i+1;
if i<=m then
p[i]:=p[0]*(step(m*Ro,i)/fuct(i)) else
p[i]:=p[0]*((step(Ro,i)*step(m,m))/fuct(m));
until (p[i]<0.0001)or (i>=20);
Nq:=(p[0]*(step(Ro*m,m+1)))/(fuct(m)*m*(1-Ro)*(1-Ro));
W:=Nq/lam;
N:=Nq+Ro*m;
u:=N/lam;
end
{_____M/M/1/K_____}
else          begin if
m=1 then    begin p[0]:=1-
lam*mu)/(1-step(lam/mu,k+1)); for i:=1
to k do p[i]:=p[0]*step(lam/mu,i);

```

```

                end else
if k=m then    begin
p[0]:=0;
  for i:=0 to m do p[0]:=p[0]+(step(lam/mu,i)/fuct(i));
p[0]:=1/p[0]; for i:=1 to k do
p[i]:=p[0]*(step(lam/mu,i)/fuct(i))
                end;
Nq:=(p[0]*(step(Ro*m,m+1)))/(fuct(m)*m*(1-Ro)*(1-Ro));
W:=Nq/lam;
N:=Nq+Ro*m;
u:=N/lam;
                                end;
end;

Exports
MySMO name 'SMO' resident;

begin end.

```

Приложение Д (рекомендуемое)

Исходный текст программы к заданию №3

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, Grids, ImgList, ExtCtrls, StdCtrls, Math, TeEngine,
  Series, TeeProcs, Chart; type
mas=array[1..50] of real;
mass=array[1..50,1..50] of real;
TForm1 = class(TForm)
  PageControl1: TPageControl;
  TabSheet1: TTabSheet;
  TabSheet2: TTabSheet;
  Label1: TLabel;
  Image1: TImage;
  Image2: TImage;
  StringGrid1: TStringGrid;
  StringGrid2: TStringGrid;
  Edit1: TEdit;
  Button1: TButton;
  StringGrid3: TStringGrid;
  Button2: TButton;
  Label2: TLabel;

```

```

    TabSheet3: TTabSheet;
LENq: TLabeledEdit;
LEN: TLabeledEdit;
    LEW: TLabeledEdit;
LEU: TLabeledEdit;
    TabSheet4: TTabSheet;
Chart1: TChart;
    Series1: TLineSeries;
Label3: TLabel;
    procedure SimQ(n:integer; A:mass; var b:mas);
procedure FormCreate(Sender: TObject); procedure
Button1Click(Sender: TObject); procedure
Button2Click(Sender: TObject); procedure
PageControl1Changing(Sender: TObject; var
AllowChange: Boolean); private
    { Private declarations }
public
    { Public declarations }
end; const
e=0.01;
var
    Form1: TForm1;
N:integer;
    m,k:array [1..50] of integer;
mu,gm,la:mas;
    r:array [1..50,1..50] of real;
    A:mass;
    B:mas;
    procedure SMO(lam,mu:real;m,k:integer; var Ro,w,Nq,N,U:real; var P: array of real);
external 'SMO.dll'; implementation {$R *.dfm}
function strtval(st:string):real;
var rl:real; i,j:integer; begin
i:=1; rl:=0;
while i<=length(st) do
begin
if (st[i]<>'.')and(st[i]<>',') then begin
rl:=rl*10+strtoint(st[i]); i:=i+1

end else begin j:=i;
i:=i+1;
while i<=length(st) do begin
rl:=rl+strtoint(st[i])/power(10,i-j); i:=i+1

end end end;
strtval:=rl
end;
{_____SIMQ_____}

```

```

procedure TForm1.SimQ(n:integer; A:mass; var b:mas);
const E=1e-21; var
  i,l,j,k1,ks:integer;
max,u,v: real;
begin for i:=1 to n
do
A[i,n+1]:=B[i];
for i:=1 to n do
begin
max:=abs(A[i,i]);
k1:=i; for l:=i+1
to n do if
abs(A[l,i])>max
then
begin
max:=abs(A[l,i]);
k1:=l;
end;
if (max<E) then begin Ks:=1; exit end else Ks:=0;
if k1<>i then
for j:=i to n+1 do
begin U:=A[i,j];

A[i,j]:=A[k1,j];
A[k1,j]:=U;
end; V:=A[i,i];
for j:=i to n+1 do A[i,j]:=A[i,j]/V;
for l:=i+1 to n do begin
V:=A[l,i];
for j:=i+1 to n+1 do A[l,j]:=A[l,j]-A[i,j]*V;
end; end;
b[n]:=A[n,n+1]; for
i:=n-1 downto 1 do
begin b[i]:=A[i,n+1];
for j:=i+1 to n do b[i]:=b[i]-A[i,j]*b[j];
end;
end;
{_____} procedure
TForm1.FormCreate(Sender: TObject); begin
StringGrid1.Visible:=false;
StringGrid2.Visible:=false;
Label2.Visible:=false; end;
procedure TForm1.Button1Click(Sender: TObject);
var i:integer; begin
if tryStrToInt(Edit1.Text,N) then
begin
StringGrid1.Visible:=true;

```

```

StringGrid1.RowCount:=N+1;
StringGrid1.cells[1,0]:='Gamma';
StringGrid1.cells[2,0]:='Mu';
StringGrid1.cells[3,0]:='m';
StringGrid1.cells[4,0]:='k';
for i:=1 to N do StringGrid1.cells[0,i]:='Узел '+inttostr(i); StringGrid2.Visible:=true;
Label2.Visible:=true; image2.Visible:=false; StringGrid2.RowCount:=N;
StringGrid2.ColCount:=N; end
else ShowMessage('Не задано кол-во узлов!'); end;
procedure TForm1.Button2Click(Sender: TObject); var
g,i,j:integer;
Ro,w,Nq,Nz,U:array[1..50] of real;
P:array [1..50,1..10] of real;
gmi,gmo,wc,Nqc,Nzc,Uc:real;
bad:boolean; begin for i:=1 to N do
begin
gm[i]:=strtoval(StringGrid1.cells[1,i]);
mu[i]:=strtoval(StringGrid1.cells[2,i]);
m[i]:=strtoint(StringGrid1.cells[3,i]);
k[i]:=strtoint(StringGrid1.cells[4,i]);
end;
gmo:=0;
for i:=1 to N do
gmo:=gmo+gm[i]; for i:=0 to N-1
do for j:=0 to N-1 do
R[i+1,j+1]:=strtoval(StringGrid2.cells[j,i]);
for i:=1 to N do for j:=1 to N do if i=j then
A[i,j]:=R[i,j]-1 else A[i,j]:=R[j,i]; for i:=1
to N do la[i]:=-gm[i]; Simq(N,A,la);
for i:=1 to N do if mu[i]<=la[i]
then begin mu[i]:=la[i]+0.1;
StringGrid1.cells[2,i]:=floattostr(mu[i]);
end;
for i:=1 to N do begin
SMO(la[i],mu[i],m[i],k[i], Ro[i],w[i],Nq[i],Nz[i],U[i],P[i]);
StringGrid3.Cells[1,i]:=floattostrf(Ro[i],ffGeneral,3,3);;
StringGrid3.Cells[2,i]:=floattostrf(w[i],ffGeneral,3,3);
StringGrid3.Cells[3,i]:=floattostrf(Nq[i],ffGeneral,3,3);
StringGrid3.Cells[4,i]:=floattostrf(Nz[i],ffGeneral,3,3);
StringGrid3.Cells[5,i]:=floattostrf(P[i,1],ffGeneral,3,3);
StringGrid3.Cells[6,i]:=floattostrf(P[i,2],ffGeneral,3,3);
StringGrid3.Cells[7,i]:=floattostrf(P[i,3],ffGeneral,3,3);
end;
for i:=1 to N do Nqc:=Nqc+Nq[i];
LENq.Text:=floattostrf(Nqc,ffGeneral,3,3); for i:=1 to N do Nzc:=Nzc+Nz[i];
LEN.Text:=floattostrf(Nzc,ffGeneral,3,3); for i:=1 to N do

```

```

wc:=wc+w[i]*(la[i]/gmo); LEW.Text:=floattostrf(wc,ffGeneral,3,3); for i:=1
to N do Uc:=Uc+U[i]*(la[i]/gmo);
LEU.Text:=floattostrf(Uc,ffGeneral,3,3);
{_____График_____}
gmi:=gmo;
Chart1.Series[0].Clear;
bad:=false; repeat for
i:=1 to N do for j:=1
to N do
if i=j then A[i,j]:=R[i,j]-1 else A[i,j]:=R[j,i];
for i:=1 to N do la[i]:=-(gm[i]*gmi)/gmo;
Simq(N,A,la);
for i:=1 to N do if Ro[i]>0.9 {mu[i]<=la[i]} then exit;
for i:=1 to N do SMO(la[i],mu[i],m[i],k[i], Ro[i],w[i],Nq[i],Nz[i],U[i],P[i]);
wc:=0;
for i:=1 to N do wc:=wc+w[i]*(la[i]/gmi);
Chart1.Series[0].AddXY(gmi,wc);
gmi:=gmi+0.1; until bad; end;
procedure TForm1.PageControl1Changing(Sender: TObject;
var AllowChange: Boolean); var i,j:integer; begin
StringGrid3.RowCount:=N+1;
StringGrid3.Cells[1,0]:='Ro';
StringGrid3.Cells[2,0]:='W';
StringGrid3.Cells[3,0]:='Nq';
StringGrid3.Cells[4,0]:='N';
StringGrid3.Cells[5,0]:='P[1]';
StringGrid3.Cells[6,0]:='P[2]';
StringGrid3.Cells[7,0]:='P[3]';
for i:=1 to N do StringGrid3.Cells[0,i]:='Узел '+inttostr(i);
end; end.

```