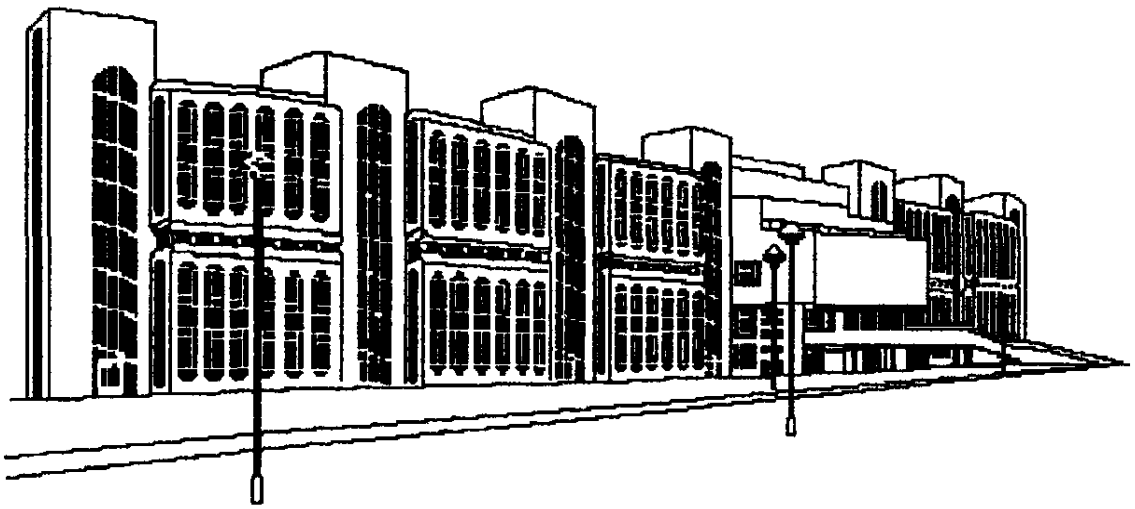
**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ГРАЖДАНСКОЙ АВИАЦИИ**

Л.Е.Рудельсон

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ ВОЗДУШНЫМ ДВИЖЕНИЕМ

Часть I

СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ



Москва – 2007

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ

Кафедра вычислительных машин, комплексов, систем и сетей
Л.Е.Рудельсон

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ ВОЗДУШНЫМ ДВИЖЕНИЕМ

Часть I

СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Книга 2

ОПЕРАЦИОННЫЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ
МАТЕМАТИЧЕСКИЕ МОДЕЛИ

Утверждено Редакционно-издательским советом МГТУ ГА в качестве учебного пособия

Москва – 2007

УДК
ББК ……..
Р..

Печатается по решению редакционно-издательского совета
Московского государственного технического университета ГА
Рецензенты: д-р технических наук, профессор В.Л. Кузнецов;
д-р технических наук, профессор В.С. Семенихин (МЦСТ)

Рудельсон Л.Е.

Р.. Программное обеспечение автоматизированных систем управления воздушным движением. Часть I. Системное программное обеспечение. Книга 2. Операционные системы реального времени. Математические модели: Учебное пособие. – М.: МГТУ ГА, 2007. - с 96.

ISBN 5-86311-…….

Рассматриваются проблемы адаптации операционных систем (ОС) к работе в автоматизированных системах управления воздушным движением. МГТУ ГА готовит инженеров, которые будут дорабатывать фирменные ОС для нужд гражданской авиации, подгоняя их характеристики под требования безопасного, экономичного и регулярного управления воздушным движением. Анализируются типичные ситуации, с которыми сталкиваются специалисты при развертывании ПО АС УВД на объектах: управление файловой системой в реальном времени, мультипроцессорная обработка событий и приоритетная диспетчеризация вычислительного процесса. Приведены технические решения и дано их математическое обоснование.

Данное учебное пособие издается в соответствии с учебным планом для студентов специальности 23.01.01 дневного обучения.

Рассмотрено и одобрено на заседаниях кафедры ВМКСС 22.05.07 и методического совета __.__.07.

Р $\frac{2404000000 - \ldots}{\text{Ц } 33\ (07)}$

ББК ……
св. тем. план 2007
поз.

РУДЕЛЬСОН Лев Ефимович
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
АВТОМАТИЗИРОВАННЫХ СИСТЕМ
УПРАВЛЕНИЯ ВОЗДУШНЫМ ДВИЖЕНИЕМ
Часть I. Системное программное обеспечение
Книга 2. Операционные системы реального времени. Математические модели
Учебное пособие

Редактор …….

ЛР №020580 от 05.09.01 г.		Подписано в печать
Печать офсетная	Формат 60x84/16	6,0 уч.-изд. л.
усл.печ.л.	Заказ №	Тираж экз.

Московский Государственный Технический Университет ГА
125993 Москва, Кронштадтский бульвар, д.20
Редакционно–издательский отдел
125493 Москва, ул. Пулковская, д.6а
ISBN……..

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

Данное пособие входит в первую часть серии «Программное обеспечение автоматизированных систем управления воздушным движением» (ПО АСУ ВД), подготовленной кафедрой Вычислительных машин, комплексов, систем и сетей для организации учебного процесса в рамках одноименной дисциплины. В соответствии с традиционным делением ПО на системную и функциональную составляющие, первая часть «Системное программное обеспечение» объединяет в своем составе следующие книги:

- Книга 1. Информационная база автоматизированных систем организации воздушного движения.
- Книга 2. Операционные системы реального времени. Математические модели.
- Книга 3. Управление периферией и связью в АС УВД.

Вторая часть «Функциональное программное обеспечение» посвящена основным задачам, ради решения которых развертываются системы УВД – аэронавигации и управлению потоками воздушного движения. Она освещает вопросы проектирования полной конфигурации системы и содержит книги:

- Книга 4. Модель использования воздушного пространства. Обработка плановой информации.
- Книга 5. Обработка радиолокационной информации.
- Книга 6. Обработка данных автоматического зависимого наблюдения.
- Книга 7. Обработка метеорологической информации.
- Книга 8. Программная поддержка интегрированной технологии УВД.

В данном пособии рассматриваются проблемы адаптации операционных систем (ОС) к работе в автоматизированных системах управления воздушным движением. Московский государственный технический университет гражданской авиации готовит инженеров, которые будут дорабатывать фирменные ОС для нужд воздушного транспорта, подгоняя их характеристики под требования безопасного, экономичного и регулярного управления воздушным движением. Исходя из сказанного, в книге анализируются типичные ситуации, с которыми сталкиваются специалисты при развертывании ПО АС УВД на объектах: управление файловой системой в реальном времени, мультипроцессорная обработка событий и приоритетная диспетчеризация вычислительного процесса. Приведены технические решения и дано их математическое обоснование.

1. ВВЕДЕНИЕ

1.1. Основные определения. Операционная система (ОС) это комплекс взаимосвязанных системных программ, назначение которого – организовать взаимодействие пользователя с компьютером и выполнение приложений [1–4]. ОС выступает в роли связующего звена между аппаратурой и выполняемыми программами, а также между программно-аппаратным комплексом и пользователем. Как следствие, она в значительной степени определяет облик всей вычислительной системы в целом. Несмотря на это, даже активные пользователи зачастую испытывают затруднения при попытке дать ее определение. Частично это связано с тем, что ОС выполняет две по существу мало связанные функции: обеспечение программисту разнообразных удобств посредством предоставления ему так называемой расширенной машины и повышение эффективности использования компьютера путем рационального управления его ресурсами.

Работа на уровне машинного языка затруднительна, особенно это касается ввода-вывода. Например, для организации чтения блока данных программист может использовать 16 различных команд, каждая из которых требует 13 параметров, таких как номер блока на диске, номер сектора на дорожке и т. п. По завершении операции возвращаются 23 значения, отражающих наличие и типы ошибок, которые, нужно анализировать. Даже если не входить в проблемы ввода-вывода, ясно, что нашлось бы не много желающих непосредственно заниматься программированием этих операций. Вопросы, подобные таким, использовать ли при записи усовершенствованную частотную модуляцию или в каком состоянии сейчас находится двигатель механизма перемещения считывающих головок, не должны волновать пользователя. Программа, которая скрывает от программиста все реалии аппаратуры и предоставляет возможность простого, удобного просмотра указанных файлов, чтения или записи – это и есть ОС. Она ограждает программистов от аппаратуры и предоставляет простой интерфейс, берет на себя все малоприятные обязанности, связанные с обработкой прерываний, управлением таймерами и оперативной памятью, а также другие низкоуровневые проблемы. Абстрактная, воображаемая машина, с которой теперь имеет дело пользователь, проще и удобнее в обращении, чем реальная аппаратура, лежащая в основе этой виртуальной, или расширенной, машины.

В функции операционной системы входят:

- осуществление диалога с пользователем;
- ввод-вывод и управление данными;
- планирование и организация процесса обработки программ;
- распределение ресурсов (памяти, процессора, внешних устройств);
- запуск программ на выполнение;
- всевозможные вспомогательные операции обслуживания;
- передача информации между различными внутренними устройствами;

- программная поддержка работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей, принтера, специальной аппаратуры).

ОС можно назвать программным продолжением устройства управления компьютера. Она скрывает от пользователя сложные ненужные подробности взаимодействия с аппаратурой, образуя прослойку между ними. В результате люди освобождаются от трудоемкой работы по организации взаимодействия процессоров, памяти, таймеров, дисков, накопителей на лентах, сетевых коммуникаций, принтеров и других устройств. Функцией ОС является их распределение между процессами, конкурирующими за обладание этими ресурсами. ОС должна управлять ими таким образом, чтобы обеспечить максимальную эффективность функционирования. Критерием эффективности может быть, например, пропускная способность или реактивность системы. Управление ресурсами включает решение двух общих, не зависящих от типа ресурса задач:

- планирование ресурса – т. е. определение: кому, когда, а для делимых ресурсов – в каком количестве необходимо выделить данный ресурс;
- отслеживание состояния ресурса – т. е. поддержание оперативной информации о том, занят или не занят ресурс, а для делимых ресурсов – какое количество ресурса уже распределено, а какое свободно.

Для решения этих общих задач используются различные алгоритмы, которые характеризуют облик ОС в целом, включая показатели производительности, области применения и пользовательского интерфейса. Например, алгоритм управления процессором фактически определяет, является ли ОС системой разделения времени, или пакетной обработки, или реального времени.

1.2. Требования к операционным системам реального времени. Понятия «реальное время», «работа в реальном масштабе времени», «ОС реального времени» известны всем, но зачастую толкуются они по-разному, и спектр этих интерпретаций очень широк. Многие путают такие термины, как «реальное время» и «быстродействие». Другие полагают, что применение ОС реального времени (ОСРВ) автоматически разрешит все проблемы создания надежной предсказуемой системы. Иногда, наоборот, считают, что это занятие для теоретиков, а любую задачу реального времени можно решить, используя популярные ОС общего назначения – достаточно быть просто хорошим программистом и знать архитектуру компьютера.

Принципиальное различие в том, что ОС общего назначения ориентированы на оптимальное распределение ресурсов компьютера между пользователями и задачами (системы разделения времени). В ОСРВ подобная задача отходит на второй план, все подчинено главной цели: вовремя реагировать на события, происходящие в сложном объекте. Реакция на изменения его состояния должна обеспечивать своевременное прохождение информации, выработку решений, эффективное воздействие на ход управляемого процесса. Различают ОС с высокой (жестко регламентированные системы) и с низкой реактивностью, т. е. скоростью реакции на изменение состояния объекта.

В «жестких» ОСРВ неспособность обеспечить реакцию в заданное время ведет к отказам и невозможности выполнения поставленной задачи. Они рассматриваются как системы с детерминированным временем, в которых время реакции должно быть минимальным. Таковы ОС в системах управления скоротечными боевыми операциями и в автоматизированных системах управления воздушным движением (АС УВД). К системам мягкого реального времени относят ОСРВ, не попадающие под определение «жесткие», например, системы управления городским коммунальным хозяйством, события в которых происходят в темпе смены сезонов. Другой пример – вычислительная сеть. Если система не успела обработать очередной пакет, это приведет к таймауту на передающей стороне и повторной посылке. Данные при этом не теряются, но производительность сети снижается. «Мягкие» ОСРВ могут не успевать решать задачу, но это не приводит к отказу системы в целом. Для реального времени необходимо введение некоторого директивного срока, называемого «время жизни», до истечения которого задача должна обязательно (для «мягких» ОСРВ – желательно) выполниться. Этот срок используется планировщиком задач как для назначения приоритета задачи при запуске, так и при выборе задачи на выполнение.

Применение ОСРВ всегда связано с разнообразной аппаратурой, с объектом, с происходящими в нем событиями. Этот аппаратно-программный комплекс включает в себя датчики, регистрирующие события на объекте, средства ввода-вывода, преобразующие показания датчиков в цифровой код для обработки и, наконец, компьютер с программой, реагирующей на изменения. ОСРВ принципиально ориентирована на обработку внешних событий. Именно это порождает коренные отличия (по сравнению с ОС общего назначения) в структуре программы, в функциях ядра, в схеме ввода-вывода. Она может быть похожа по интерфейсу пользователя на ОС общего назначения (к этому стремятся производители), однако устроена она совершенно иначе.

Добавим, что ОСРВ в большинстве случаев рассчитаны на ограниченный класс изделий. Если ОС общего назначения обычно воспринимается пользователями (не разработчиками) как готовый набор приложений, то ОСРВ служит лишь инструментом для создания аппаратно-программного комплекса конкретной АСУ. Как следствие, их потребителями становятся разработчики систем управления и сбора данных. На всех этапах проекта программист с различной степенью достоверности, однако, *всегда* знает, какие события могут произойти на объекте, знает критические сроки обслуживания каждого из этих событий. Всегда ожидается, что ОСРВ будет в предсказуемые времена обслуживать непредсказуемый поток внешних событий. Она должна успевать реагировать на:

- любое событие в течение критического для него «времени жизни», определяемого замыслом системы, всякое опоздание считается ошибкой;
- каждое из нескольких внешних событий в течение интервалов времени, критических для этих событий.

Рассмотрим признаки операционных систем реального времени.

1.3. Свойства операционных систем реального времени. Четкое *разграничение систем* разработки (средства создания и отладки приложения) и систем исполнения (ОС и компьютер, на котором реализуется конечный продукт) – одно из коренных внешних отличий ОСРВ от систем общего назначения. Средства исполнения – это совокупность инструментов (ядро, драйверы, исполняемые модули), обеспечивающих работу приложения реального времени. Аппаратные средства как неотъемлемая часть АСУ реального времени должны быть адекватны решаемой задаче, и ведущие ОСРВ перекрывают ряд популярных архитектур, чтобы удовлетворять их самые разные требования.

Средства разработки современных ОСРВ поддерживают современные пакеты программирования и так называемые резидентные средства разработки, исполняемые в собственной среде. Они также содержат средства удаленной отладки, средства профилирования (измерение времени выполнения отдельных участков кода), средства эмуляции целевого процессора, специальные средства отладки взаимодействующих задач, а иногда и средства моделирования.

Время реакции системы на прерывание – параметр, оценивающий выполнение цепочки действий от возникновения запроса на прерывание и до выполнения первой инструкции обработчика. Это время нужно уметь определять в худшей для системы ситуации, т. е. в предположении, что процессор загружен, что в это время могут происходить другие прерывания, что могут выполняться какие-то действия, блокирующие прерывания.

Размер системы исполнения, т.е. объем минимально необходимого для работы приложения набора (ядро, системные модули, драйверы) остается важным параметром, и производители ОСРВ стремятся к тому, чтобы размеры ядра и обслуживающих модулей системы были невелики. *Возможность исполнения из постоянной памяти* позволяет создавать компактные надежные встроенные ОСРВ с ограниченным энергопотреблением, без внешних накопителей.

Система приоритетов и алгоритмы диспетчеризации. В ОС общего назначения используются, как правило, различные модификации алгоритма круговой диспетчеризации, основанные на понятии непрерывного кванта времени, предоставляемого процессу. Планировщик по истечении каждого кванта времени просматривает очередь активных процессов и принимает решение, кому передать управление, основываясь на численных значениях приоритетов, присвоенных процессам. Приоритеты могут быть фиксированными или меняться со временем, это зависит от алгоритмов планирования, но рано или поздно обслуживаются все процессы.

В чистом виде алгоритмы круговой диспетчеризации не применимы для ОСРВ. Основной недостаток – непрерывный квант времени, в течение которого процессором владеет только один процесс. Нужно иметь возможность сменять процесс еще до истечения установленного кванта. Известные дисциплины планирования – динамические, приоритетные, монотонные,

адаптивные – всегда преследуют одну цель – предоставить инструмент, позволяющий в любой момент времени исполнять именно тот процесс, который необходим.

Механизмы межзадачного взаимодействия. Для ОСРВ характерна развитость механизмов обмена данными и синхронизации процессов. К ним относятся семафоры, мьютексы (взаимное исключение), события, сигналы, средства работы с разделяемой памятью, каналы данных, очереди сообщений. Их аналоги существуют в ОС общего назначения, но в ОСРВ время исполнения системных вызовов почти не зависит от состояния системы, и всегда есть, по меньшей мере, один быстрый механизм передачи данных от процесса к процессу.

Средства для работы с таймерами необходимы системам с жестким временным регламентом. Эти средства, как правило, позволяют:

- генерировать прерывания по истечении временных интервалов;
- измерять и задавать различные промежутки времени (от 1 мкс);
- создавать разовые и циклические «будильники».

Файловая система – это часть любой ОС, назначение которой состоит в том, чтобы обеспечить удобный интерфейс для работы с данными, хранящимися на диске, и совместное использование файлов процессами.

В широком смысле понятие «файловая система» включает:

- совокупность всех файлов на диске;
- наборы структур данных, используемых для управления файлами, такие, например, как каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске;
- комплекс системных программных средств, реализующих управление файлами, в частности: создание, уничтожение, чтение, запись, именование, поиск и другие операции над файлами.

Разработчики новых ОС стремятся обеспечить пользователя возможностью работать сразу с несколькими файловыми системами. В новом понимании файловая система состоит из многих составляющих, в число которых входят и файловые системы в традиционном понимании. Для ОСРВ задача сопровождения файлов усложняется тем, что ради выполнения основной задачи системы доступ высокоприоритетных функциональных программ к файлам не должен блокироваться даже в процессе их копирования.

Данная книга написана не для разработчиков ОС или системных программистов, такая литература легко доступна, а для будущих инженеров, которым предстоит дорабатывать фирменные образцы для нужд авиационных систем. Основное содержание составляют задачи, с которыми сталкиваются специалисты в области ПО АС УВД при развертывании центров управления полетами: управление файлами, параллельный счет, диспетчеризация вычислений и т.д. Показано, какая практическая потребность породила ту или другую функцию ОСРВ, какой инженерный замысел заложен в решение и какими математическими средствами даются количественные оценки технической идеи.

2. Управление файлами

2.1. Файловые системы

2.1.1. Терминология. Файл – это основной элемент хранения данных в компьютере, известное любому пользователю понятие, воспринимаемое как определенный объект (предмет), у которого есть назначение, есть начало и длина, который отличается от остальных файлов именем и расположением. Как любой другой предмет, файл можно создать, переместить и уничтожить, однако без внешнего вмешательства он будет сохраняться неизменным. Файлы предназначены для хранения данных любого типа – текстовых, графических, звуковых, исполняемых программ и многого другого. Компьютерная информация хранится на жестких дисках, компакт-дисках (CD), флешках и дискетах. Носители, организация записи и чтения на них различны. Соответственно, различаются и способы хранения информации.

Для ясного представления задачи хранения файлов необходимо знать принципы построения файловых систем. Файловая система (ФС) является важной частью любой ОС, которая отвечает за организацию хранения и доступа к информации на любых носителях. Рассмотрим в качестве примера файловые системы для наиболее распространенных из них, для магнитных дисков. Как известно, информация на жестком диске хранится в секторах (обычно 512 байт) и само устройство может лишь выполнять команды считать или записать информацию в определенный сектор на диске. В отличие от этого файловая система позволяет пользователю оперировать с более удобным для него понятием – файл. ФС берет на себя организацию взаимодействия программ с файлами, расположенными на дисках. Для идентификации файлов используются имена. Современные ФС предоставляют пользователям возможность давать файлам достаточно содержательные мнемонические названия. Напомним основные определения:

- Файл – именованная область диска, предназначенная для выполнения функции хранения данных.
- Файловая система – общий, заранее описанный набор правил расположения и организации данных на носителе.
- Имя файла – одно из его свойств, однозначно указывающее на объект.
- Кластер – конгломерат дискретных дисковых пространств, используемых в комплексе для хранения данных.
- Сектор – основная логическая единица дискового пространства.
- Блок – основная физическая единица дискового пространства.

ФС с точки зрения пользователя – это упорядоченное пространство, в котором размещаются файлы. Ее наличие позволяет определить не только *как* называется файл, но и *где* он находится. Различать файлы лишь по имени было бы неэффективно: про каждый файл приходилось бы помнить, как он называется и при этом заботиться, чтобы имена не повторялись. Более того, необходим механизм, позволяющий работать с группами тематически связанных между собой файлов (например, компонентов одной и той же про-

граммы или глав книги). Иначе говоря, файлы нужно *систематизировать*.

Развитие ФС привело к изменению понятия файл: от первоначального толкования как упорядоченной последовательности логических записей до понятия объекта, определенного набором атрибутов (имя файла, его псевдоним, время создания и собственно данные). Под каталогом в ФС понимается, с одной стороны, группа файлов, объединенных пользователем исходя из замысла системы, с другой стороны, каталог – это файл, содержащий системную информацию о группе составляющих его файлов. ФС обычно строится как иерархическая структура, ярусы которой представляют собой каталоги, содержащие информацию о файлах, и каталоги более низкого уровня.

Базовой единицей жесткого диска является раздел**,** создаваемый во время разметки. Каждый раздел содержит один том, обслуживаемый какой-либо ФС и имеющий таблицу оглавления файлов – корневой каталог. Некоторые ОС поддерживают создание томов, охватывающих несколько разделов. Жесткий диск может содержать до четырех основных разделов. Это ограничение связано с характером организации данных на жестких дисках. Многие операционные системы позволяют создавать так называемый расширенный раздел, который может разбиваться на несколько логических дисков.

В первом физическом секторе жесткого диска располагается головная запись загрузки и таблица разделов (табл. 2.1). Головная запись загрузки – первая часть данных на жестком диске. Она зарезервирована для программы начальной загрузки BIOS, которая при загрузке с жесткого диска считывает в оперативную память первый физический сектор на активном разделе диска, называемый загрузочным сектором. Каждая запись в таблице разделов содержит начальную позицию и размер раздела на жестком диске, а также информацию о том, первый сектор какого раздела содержит загрузочный сектор.

Таблица 2.1

Размер (байт)	Описание
446	Загрузочная запись
16	Запись 1 раздела
16	Запись 2 раздела
16	Запись 3 раздела
16	Запись 4 раздела
2	Сигнатура 055AAh

Различие между ФС заключается, в основном, в способах распределения пространства между файлами на диске и организации на диске служебных областей. Современные ОС обеспечивают пользователя возможностью работать одновременно с несколькими ФС. В этом случае ФС рассматривается как часть подсистемы ввода-вывода. В большинстве ОС реализуется механизм переключения файловых систем (ПФС), позволяющий поддерживать различные типы ФС. В соответствии с этим подходом информация о файловых системах и файлах разбивается на две части – зависимую от ФС и не зависимую. ПФС обеспечивает интерфейс между ядром и ФС, транслируя запросы ядра в операции, зависящие от типа ФС. При этом ядро имеет представление только о независимой части ФС. ПФС преобразует запросы к файлам в формат, воспринимаемый следующим уровнем – уровнем драйверов файловых систем. Для выполнения своих функций драйверы ФС обращаются

к драйверам конкретных устройств хранения информации.

Клиент-серверные приложения предъявляют повышенные требования к производительности ФС. Современные файловые системы должны обеспечивать эффективный доступ к файлам, поддержку носителей достаточно большого объема, защиту от несанкционированного доступа и сохранение целостности данных. Под целостностью данных подразумевается способность ФС обеспечивать отсутствие ошибок и нарушений согласованности в данных, а также восстанавливать поврежденные данные.

2.1.2. Восстановление данных. Развитие файловых систем определялось двумя факторами – появлением новых стандартов на носители информации и ростом требований к характеристикам ФС со стороны прикладных программ (разграничение уровней доступа, поддержка длинных имен файлов). Сначала первостепенное значение имело увеличение скорости доступа к данным и минимизация объема хранимой служебной информации. Впоследствии, с появлением более быстрых жестких дисков и увеличением их объемов, на первый план вышло требование надежности хранения информации, которое привело к необходимости избыточного хранения данных.

Эволюция файловой системы связана с развитием технологий реляционных баз данных. ФС использовала последние достижения, разработанные для систем управления базами данных (СУБД): механизмы транзакций, защиты данных, систему самовосстановления в результате сбоя. Пройден путь от простой системы, взявшей на себя функции управления файлами, до системы, представляющей собой полноценную СУБД, обладающую встроенным механизмом протоколирования и восстановления данных.

На сегодняшний день люди оказались в такой зависимости от информационных технологий, что обозначился рубеж, за которым носители стали дешевле хранимых на них сведений. Особенно это касается больших территориальных систем реального времени, к классу которых принадлежит АС УВД. Однако и средняя стоимость ноутбуков руководителей большинства серьезных компаний достигает десятков миллионов долларов. Ценность определяется именно содержимым. В связи с этим в любой современной ФС остро встает вопрос информационной безопасности и сохранности данных.

Известно [4], что физическим отражением данных является их носитель. Все носители являются физическими объектами и, следовательно, им свойственно утрачивать изначальные качества. Особенно важен такой критический параметр, как отказоустойчивость, которая характеризуется средним временем до выхода технического устройства из строя. Ни один прибор, в том числе носители, не может работать бесконечно, с учетом той разницы, что с их отказом теряется хранимое на них дорогостоящее содержимое.

Меры защиты данных, при всем их разнообразии, укладываются в два схожих подхода: дублирование и резервирование информации. Дублирование проводится по событиям, а резервирование заключается в периодическом сохранении критических данных на независимый носитель. Обоим методам присущи достоинства и недостатки. Дублирование эффективнее для полноты

сохраняемых данных. Создание дубля в реальном времени подразумевает учет всех изменений, зафиксированных на момент отказа. Однако все изменения, сделанные человеком, в том числе и ошибочные, алгоритм принимает за верные и записывает «не задумываясь». Метод хорошо защищает от технологических ошибок, но беззащитен от человеческого фактора. Вторая группа методов – резервирование – хорошо зарекомендовала себя при частых ошибках обслуживающего персонала, так как дает системе время на распознавание человеческого фактора, однако из-за этой задержки теряются все корректировки данных, произведенные в интервале времени между копированием данных и отказом основного носителя.

Аспекты восстановления данных делятся на физический и логический. Физическое восстановление данных требуется при критических изменениях в механике и электронике носителя информации. Логическое восстановление обычно применяется при программных ошибках (искажение данных, неверное удаление файла, ошибки чтения-записи и т.д.). Физическое восстановление требует выводить отказавшее устройство из системы, т.е. ее реконфигурации. Логическое восстановление при некоторых типичных проблемах можно выполнять в реальном времени, в процессе работы программного обеспечения (ПО) автоматизированных систем управления воздушным движением (АС УВД). Рассмотрим пример нейтрализации потерь данных с возможностью последующего логического восстановления.

2.1.3. Поддержание информационной целостности. Схема компьютерной обработки полетной информации в современных АС УВД заключает в себе риск потери целостности данных. Согласно замыслу, решение общей задачи – поддержания интегрированной технологии работы диспетчерского персонала – делится на два направления. Во-первых, это сопровождение обновляемых данных о среде, в которой развивается процесс УВД – о состоянии атмосферы, пропускной способности элементов структуры воздушного пространства (ВП), техническом состоянии радиометрических средств и аппаратуры связи. Во-вторых, это сопровождение данных об объектах управления – воздушных судах (ВС), совершающих полеты. Соответственно, структура ПО построена на принципах разделения функций его элементов. На комплекс программ (КП) контроля периферийных источников возложено тестирование и управление локаторами, пеленгаторами, линиями передачи данных. КП обработки метеорологической информации обобщает информацию о погоде в зоне ответственности системы. Специальные программы рассылают по рабочим местам диспетчеров данные о режимных ограничениях. В результате, о каждом аэродроме, о каждом источнике измеренной и плановой информации, о ситуации в секторах – в базе данных (БД) системы создаются автономно обновляемые и дополняющие друг друга описания. Частично они совпадают друг с другом (например, координаты аэродрома, его наименование, код взлетно-посадочной полосы – ВПП), частично различаются (состояние облачности, температура воздуха, коэффициент сцепления ВПП для метеорологической подсистемы или нарезка секторов подхода и круга,

описание коридоров связи с трассами – для плановой).

Аналогично, каждый объект управления – ВС сопровождается в раздельных описаниях КП обработки плановой и измеренной (радиолокационной и спутниковой) информации. В соответствии с задачами этих подсистем, каждая из них использует различные математические схемы расчета траектории одного и того же ВС. «Плановики» работают с точностью плановых сообщений, в которых время указывается в часах и минутах, расстояния преобразуются из координат и наименований аэродромов, районов УВД и трасс, а высота задается с точностью до эшелона. Они не учитывают при расчетах ни маневров разворота в точках излома трасс, ни других нюансов траектории полета. В противоположность этому, КП обработки радиолокационной информации в процессе прокладки траектории отслеживает параметры движения ВС с точностью радарных измерений и использует мощный математический аппарат теории оптимальных статистических решений. Наконец, КП обработки данных автоматического зависимого наблюдения (АЗН) получает с бортов ВС результаты радионавигационных измерений в совокупности с расчетами отклонений фактического пути от заданного. Доклады АЗН содержат и расчеты маневров ВС для устранения этих отклонений.

Как следствие, описания каждого элемента ВП и каждого ВС, сопровождаемые различными КП ПО АС УВД, не являются идентичными. С течением времени рассогласования становятся все существеннее, и риск потери целостности данных нарастает. Заметную роль в этой тенденции играют вводы диспетчеров, корректирующие информацию в БД. Каждый акт приема-передачи управления, смены позывного ВС, уточнения параметров движения, последовательно выполняемые различными диспетчерами, фиксируются в каждой копии как в компьютерах рабочих мест, так и в БД сервера системы. Моменты фиксации разнесены во времени и подчинены установленной в локальной сети дисциплине обмена и доступа. В таких условиях становятся вероятными различия не только в данных взаимодействующих функциональных КП, но и в их копиях на сервере и на рабочих местах.

Традиционный путь преодоления нарушений целостности основан на организационных мероприятиях [4]. Неоднозначность копий приводит к выдаче на экран противоречивой информации, к ошибкам в расчетах и другим последствиям, совпадающим по характеру проявлений с результатами недостаточной отладки ПО. Так они и классифицируются. Соответственно, нейтрализация рассогласования информационных копий элементов системы и объектов управления производится, как правило, перезапуском ПО. Все копии формируются заново и становятся непротиворечивыми. Однако при этом теряется часть информации, которая вследствие недостаточной синхронизации привела к нарушению целостности.

2.2. Нейтрализация рассогласований полетной информации

2.2.1. Постановка задачи. Рассмотрим альтернативную схему согла-

сования данных, сопровождаемых различными КП и рассредоточенных в копиях рабочих мест и сервера вычислительной сети. Логика ее работы состоит в том, чтобы в процессе нарастания отклонений не дожидаться момента потери целостности, приводящего к информационному отказу, но своевременно обнаруживать неизбежно возникающие отклонения и оперативно разрешать их. Для выполнения этой функции необходимо определить критерии обнаружения рассогласования данных и построить стратегию их применения, гарантирующую минимальные затраты вычислительных ресурсов при заданной вероятности поддержания достоверности информации.

Классифицируем причины рассогласования данных о ВС в описаниях различных КП системы управления воздушным движением:

- различие математических схем прокладки траектории ВС;
- нарушение синхронизации обновления копий в разных узлах сети;
- непредсказуемые искажения информации (сбои ПО и аппаратуры).

Радикальным средством устранения причин первой группы явилось бы объединение описаний различных КП в одно, сопровождаемое совместно. Однако устранение этой избыточности лишь смещает проблему в другую сферу. Опыт показывает, что в процессе работы пользователи вступают друг с другом в информационные конфликты как на общем поле памяти, так и по нарушению отношений предшествования. В результате объединения описаний ВС на первый план выдвигаются причины второй группы. В любом случае применение единой расчетной схемы предъявляет требования к выравниванию погрешности входной информации в процессе УВД. Сведения из планов полетов (например, курс движения в каждой точке траектории) должны использоваться на этапе экстраполяции радиолокационной траектории или расчета профиля полета на борту ВС, оснащенного аппаратурой АЗН. Пространственное положение и скорость движения наиболее точно вычисляются на бортах и должны служить эталоном для других элементов ПО. Например, КП обработки планов полетов может снизить по ним погрешность своей информации до необходимого для задач УВД уровня. К ряду систем уже выдвинуто требование автоматической корректировки планов по результатам радиолокационных измерений и спутниковой навигации.

Причины второй группы обусловлены стохастическим характером изменения информации о движущихся ВС. Корректировки копий, хранящихся на сервере, рассылаются по рабочим местам диспетчеров секторов, управляющих этими ВС. Отправленные сервером кодограммы достигают адресатов не одновременно. На этот нерегулярный поток накладывается встречный, складывающийся из реакций ПО на вводы функций с рабочих мест диспетчеров. С известной вероятностью встречные сообщения могут содержать противоречивую информацию о навигационных и других параметрах ВС. Такие ситуации должны анализироваться ПО, и результирующие изменения данных всех участников процесса взаимодействия должны фиксироваться одинаково. Для достижения однозначности организуются дисциплины квитирования обмена и выставления пауз выравнивания копий.

Нейтрализация причин третьей группы, в силу непредсказуемости искажения данных в результате их воздействия, основывается на статистическом наблюдении. Обнаружить нарушение информационной целостности можно таким простым способом, как последовательными вычислениями контрольных сумм значений всех полей записи об объекте и сопоставлениями со значениями, вычисленными в момент предыдущего включения процедуры. Эти же контрольные суммы используются для выравнивания содержимого копий, хранящихся на других рабочих местах сети. При каждом санкционированном изменении записи суммы пересчитываются заново, и рассогласование данных обнаруживается только в случаях, если изменения вызваны нарушениями вычислительного процесса.

2.2.2. Формализация задачи

2.2.2.1. Алгоритмическое описание. Сформулируем задачу поддержания информационной целостности ПО АС УВД. Должна быть разработана трехступенчатая схема нейтрализации рассогласования информационных копий управляемых объектов, обеспечивающая следующие качества:

- целостность (единство) описания навигационных параметров ВС в копиях взаимодействующих КП с помощью ранжирования данных по приоритетности источника (первая ступень);
- синхронизацию санкционированных изменений данных в сопровождаемых копиях каждого описания ВС (в различных узлах вычислительной сети и в различных КП) путем квитирования обмена (вторая ступень);
- защиту данных от несанкционированного доступа и от непредсказуемых искажений вследствие аппаратных и программных сбоев в процессе обработки информации (третья ступень).

Упрощенная схема процедуры поддержания целостности представлена на рис. 2.1. Первая ступень включается функциональными КП по событиям завершения обработки кодограмм, поступивших от источников информации о движении ВС (плановые сообщения, доклады АЗН, данные пеленгаторов и радаров), а также с рабочих мест. Рассчитанные разными КП текущие параметры движения сопоставляются и при необходимости выравниваются по данным приоритетного КП. Наименьшей погрешностью отличаются данные АЗН, наибольшей – телеграммы плановой подсистемы. Изменения фиксируются в кодограммах сетевого обмена и рассылаются для согласования копий. По каждому факту обмена выставляется пауза ожидания квитанции, подтверждающей прием посланной кодограммы абонентом.

Вторая ступень тактируется прерываниями от таймера системы. При каждом подключении анализируется, поступила ли квитанция о завершении обмена. В случае успешного исхода выставленная пауза аннулируется. Если же по истечении паузы квитанция так и не поступила отправителю, процедура повторяет рассылку.

Третья ступень также включается периодически. По каждой записи о

ВС, содержащейся в каждом узле сети, формируются кодограммы, содержащие ключевые параметры (контрольные суммы полей записи) и значение момента времени последнего обновления соответствующей копии. Сформированные кодограммы рассылаются по сети и фиксируются в файле сопоставления. При обнаружении расхождения значений ключевых параметров, вычисленных в разных узлах сети, производится выравнивание информации. Приоритетной считается запись с более поздним временем обновления.

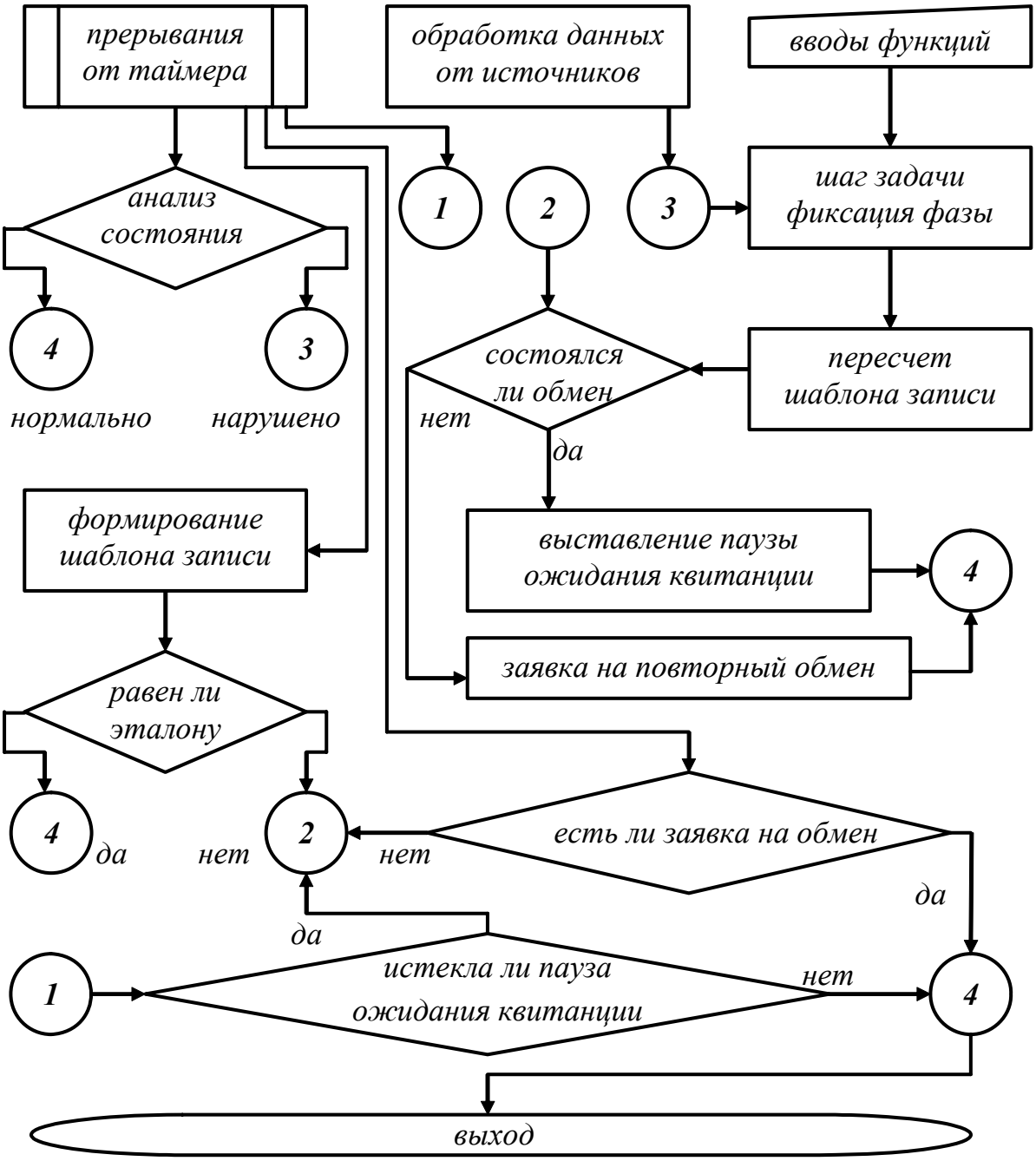


Рис. 2.1. Упрощенная блок-схема процедуры поддержания информационной целостности

2.2.2.2. ВЕКТОРНЫЙ ПРОЦЕСС СОГЛАСОВАНИЯ КОПИЙ. Формализуем поставленную задачу отыскания оптимальной последовательности вмешательств в вычислительный процесс для нейтрализации рассогласования опи-

саний объектов в разных КП. Модель анализируется в векторном информационном пространстве БД. Каждая запись интерпретируется как вектор $\overline{X}$, координаты которого $x_1, x_2,\ldots, x_j,\ldots, x_J$, представляют собой численные значения длины пути доступа к записи по каждому из входящих в нее атрибутов, $j=\overline{1,J}$ – порядковый номер атрибута, J – количество атрибутов записей. Таких векторов, отображающих достижимость данных, насчитывается N по числу записей, хранящихся в системе. Эффективность управления характеризуется в каждый момент t длительностью доступа (длиной или временем прохождения пути поиска) к запрошенным записям. Количество J путей доступа к любой записи, равное количеству атрибутов, определяет размерность вектора $\overline{X}_{NJ}(t)$ состояния системы: $\overline{X}_{NJ}(t)=\left\{x_{N1}(t),...,x_{Nj}(t),...,x_{NJ}(t)\right\}$. Устанавливая предельно допустимую длину L_0 для любого пути доступа, или задавая последовательность $\{L_{0j}\}$ для каждого пути, можно в качестве оценки состояния системы принять: $\max\limits_j \dfrac{\overline{X}_{Nj}(t)}{L_{0j}} \le 1$.

Если в некоторый дискретный момент t_i значение хотя бы одной координаты $\overline{X}_{Nj}(t_i)$ превысит соответствующий ей порог L_{0j}, то процесс управления данными требует оперативного вмешательства, восстанавливающего нарушенную меру состояния. При этом вмешательство (регулирование) можно осуществлять либо по событию ввода новой записи и установления путей доступа к составляющим ее атрибутам, либо посредством организации периодических процедур анализа текущих значений L_{0j}. Первый способ экономичнее по затратам ресурса времени, второй более универсален и может применяться на всех трех ступенях поддержания целостности данных. Каждый акт регулирования состоит в перераспределении запросов очередей в ступенях процедуры, позволяющем не допускать превышения $\overline{X}_{Nj}(t)$ над L_{0j}. Как правило, преобразование фрагментов очередей выполняется тем быстрее, чем короче существующие пути доступа. Если наряду с последовательностью пороговых значений $\{L_{0j}\}$ задать последовательность критических уровней L_{cj}, $L_{cj} < L_{0j}$, то профилактическое регулирование очереди следует производить уже при нарушении условия $\overline{X}_{Nj}(t) \le L_{cj}$.

Обозначим затраты времени на оперативное вмешательство как T_0, на профилактическое – T_p ($T_0 > T_p$), на вычисление длины пути доступа – T_0. Пусть за время t наблюдения было выполнено q_o оперативных и q_p профилактических корректировок, а количество измерений длины $\overline{X}_{Nj}$ пути доступа составило q_m (для стратегии периодического анализа очередей $q_m = t/\Delta t$, Δt – период измерений длины). Тогда суммарные затраты времени T_Σ на динамическую оптимизацию управления данными составят $T_\Sigma = q_o T_o + q_p T_p + q_m T_m$. Если в дискретный момент времени $q_m\ \Delta t$ хотя бы для одного j координата $\overline{X}_{Nj}\left(q_m \cdot \Delta t\right) \ge L_{cj}$, но для всех j соблюдается $\overline{X}_{Nj}\left(q_m \cdot \Delta t\right) \le L_{oj}$, то производится профилактическое регулирование j-го фрагмента записи. Решение состоит

в выборе либо шага $\Delta\hat{t}$ вычисления параметров $\overline{X}_{Nj}$ и L_{cj}, либо разности пороговых значений $\Delta\hat{L} = \hat{L}_{oj} - \hat{L}_{cj}$, при которых средние (по интервалу наблюдения) затраты T_Σ времени: $\hat{T}_\Sigma = \hat{T}_\Sigma\left(\Delta\hat{t}, \hat{L}_{c1}, \ldots, \hat{L}_{cj}\right) = \lim\limits_{t\to\infty} \frac{T_o q_o(t) + T_p q_p(t) + T_m q_m(t)}{t}$ будут минимальными, т.е. $\hat{T}_\Sigma = \min\limits_{\Delta t, Lcj}\left\{T_\Sigma\left(\Delta t, L_{cj}\right)\right\}$.

2.2.2.3. Допущения и ограничения. Для отыскания оптимальной стратегии выбора T_Σ необходимо ввести ряд допущений на параметры процессов наблюдения и измерений, процедур профилактического и оперативного регулирования, и показать их справедливость в приложении к исследуемой модели. Отправным пунктом формализации становится аппроксимация дискретного описания области определения оценочной функции T_Σ. Нужно сопоставить каждой координате $\{x_j(t_i)\}$ вектора $\overline{X}_J$ квазинепрерывный случайный процесс $\xi_j(t)$, сглаживающий измеренные значения по линейной схеме с учетом последовательного применения управляющих воздействий. Пусть $x_j^0(t_0) = x_j^0$ есть значение j-й координаты до первой регулировки. Пусть первое вмешательство (оперативное или профилактическое) произведено в момент $q_1 \cdot \Delta t$. Обозначим через $x_j^1(\tau)$ значение j-й координаты, достигнутое в результате первой корректировки для всех $\tau \le q_2 \cdot \Delta t$, где $q_2 \cdot \Delta t$ есть момент второго регулирования (вообще, $q_i \cdot \Delta t$ – момент i-го регулирования). Определим:

$$\xi_j(\tau) = \begin{cases} x_j^0(\tau), \quad 0 < \tau \le q_1 \cdot \Delta t, \\ x_j^0(q_1\Delta t) + x_j^1(\tau) - x_j(t_0), \quad (q_1 \cdot \Delta t) < \tau \le (q_2 \cdot \Delta t), \\ x_j^0(q_1\Delta t) + x_j^1(q_2 \cdot \Delta t) - x_j(t_0) + x_j^2(\tau) - x_j(t_1), \quad (q_2 \cdot \Delta t) < \tau \le (q_3 \cdot \Delta t), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \end{cases}$$

Первое ограничение на предлагаемую модель случайного процесса изменения векторов $\{x_j(t)\}$, $\{\xi_j(\tau)\}$ вводится для обоснования допустимости использования оценочной функции, усредняемой по времени при $t \to \infty$, к выбору стратегии T_Σ (усреднение по множеству S компонент затрат ресурсов $\{T_\Sigma\}$). Пусть векторный процесс $\bar{\xi}_j(\tau) = \{\xi_1(\tau), \ldots, \xi_j(\tau), \ldots, \xi_J(\tau)\}$, $\tau \ge 0$, имеет стационарные приращения, представимые эргодическими составляющими (процессами). В приложении к измерениям пути доступа к элементам записи правомерность ограничения следует из анализа сформулированной модели.

Второе ограничение говорит о свойстве ординарности процесса образования и удаления записей. Вероятность нарушения одновременно (при $\Delta t \to 0$) более чем одной меры $x_j(t) \le L_{cj}$ состояния пренебрежимо мала:

$$\underset{\Delta t \to \infty}{P}\{x_j(t) \le L_{cj}, x_k(t) \le L_{ck}, x_j(t+\Delta t) > L_{cj}, x_k(t+\Delta t) > L_{ck}\} \to 0, \quad j \ne k;\ j,k = \overline{1,J}\ .$$

Заметим, что для описания модели поддержания целостности в векторном информационном пространстве данное ограничение несущественно, так как не затрагивает свойства эргодичности приращений процессов $\xi_j(t)$. Однако без его введения возникают затруднения при расчете значений парамет-

ров $q_o(t)$, $q_p(t)$ и $q_m(t)$ для всех непериодических компонент затрат ресурсов времени на сопровождение данных.

Третье ограничение вводится как обоснование достаточности перехода от дискретной совокупности измеренных координат вектора $\overline{X}_{Nj}$ к квазинепрерывному описанию процесса наблюдений. Допустим, что на каждом интервале наблюдений $\{k\cdot\Delta t,(k+1)\cdot\Delta t\}$ каждый процесс $\xi_j(t)$ с исчезающе малой ошибкой аппроксимируется сплайном третьего порядка, таким, когда к рассмотрению привлекаются лишь соседние наблюдения: $(k-1)\cdot\Delta t,(k+1)\cdot\Delta t$. Тогда, если результаты измерений *s*-й компоненты ресурса времени (вычисление длины пути доступа, сопоставление ключевых параметров) представимы как совокупность $DX_s=\left\{\xi_{js}(k\cdot\Delta t);k=0,1,...;j=\overline{1,J};s=\overline{1,S}\right\}$, то после сглаживания сплайнами она преобразуется к виду: $D\Xi_s=\left\{\xi_{js}(\tau);j=\overline{1,J};s=\overline{1,S}\right\}$, где $\xi_{js}(\tau)$ – значения процесса $\xi_j(\tau)$ для *s*-й компоненты затрат ресурсов системы.

Четвертое ограничение обосновывает правомерность автономного рассмотрения *S* компонент ресурсов времени, затрачиваемого на измерение координат $x_j(t)$ процесса управления данными. Пусть как измеренные DX_s, так и сглаженные $D\Xi_s$ совокупности, независимо от *s*, распределены одинаково и при любом $s\neq k;k,s=\overline{1,S}$ не коррелированны друг с другом. В приложении к вычислению длины путей доступа и сопоставлению ключевых параметров справедливость ограничения очевидна: первое отображает процесс образования и удаления записей, второе – воздействие потока сбоев и отказов.

Пятое ограничение – отсутствие последействия – говорит о том, что регулирование координат процесса не сказывается на природе их дальнейшего изменения. Правомерность допущения нетрудно показать, так как интенсивности поступления записей и потока искажений информации определяются внешними причинами. Однако введение этих допущений, в силу их очевидности, позволяет строго утверждать, что наблюдаемые фрагменты $x_j^i(q_i\cdot\Delta t)$ и результаты их аппроксимации процессами $\xi_j(\tau)$ имеют идентичные распределения. Это означает, что оптимальность выбранной стратегии становится инвариантной относительно количества произведенных вмешательств.

При сделанных допущениях результаты $\hat{T}_\Sigma$ регулирования, полученные при оптимальном (без учета регулировок) значении $\Delta\hat{t}$ (или $\Delta\hat{L}_c$), можно пересчитывать в T_Σ^*, полученные при новых параметрах Δt^* ($\Delta L_c{}^*$) и при новых критических уровнях $L_{cj}{}^*$, $j=\overline{1,J}$. Отыскиваем: $T_{\Sigma s}^*(t)=T_o q_o^*(t)+T_p q_p^*(t)+T_m q_m^*(t)$ (в случае регулярных компонент $q_m^*=t_s/\Delta t^*$, где t_s – время наблюдения за *s*-й компонентой затрат ресурсов системы). В силу эргодичности приращений векторного процесса найденные значения можно усреднить по *S* компонентам $\left(s=\overline{1,S}\right)$: $T_\Sigma^*(t)=\frac{1}{S}\sum_{s=1}^{S}T_{\Sigma s}^*(t)$. Если $T_\Sigma^*(t)<\hat{T}_\Sigma(t)=\frac{1}{S}\sum_{s=1}^{S}\hat{T}_{\Sigma s}(t)$, то новая стратегия $\left(\Delta t^*,L_{c1}^*,...,L_{cj}^*\right)$ лучше исходной по критерию суммарных затрат времени на

динамическое управление данными, независимо от количества вмешательств.

2.2.3. КОЭФФИЦИЕНТЫ ОПТИМАЛЬНОЙ СТРАТЕГИИ. Полученный результат позволяет не только найти оптимальную последовательность вмешательств, но и реализовать ее в виде несложных алгоритмических процедур. Необходимые расчетные соотношения ограничены следующим списком.

Затраты T_o на оперативное вмешательство, т.е. на полное преобразование всех очередей обмена:

$$T_o = \sum_{j=1}^{J} T_{oj} = \sum_{j=1}^{J} t_w \cdot W_{oj} = t_{w\cdot} \sum_{j=1}^{J} W_{oj} = t_{w\cdot} \sum_{j=1}^{J} (2L + 3N) = 3JNt_{w\cdot} + 2t_{w\cdot} \sum_{j=1}^{J} 10^{\Delta p_j},$$

где t_w – время исполнения операции; W – количество операций, необходимых для реализации j-й ступени поддержания целостности; Δp_j – десятичный диапазон представления (разрядность) j-го атрибута; N – количество записей, J – атрибутов БД. Оценка является верхней. Необходимое количество L дискретов индекса поиска элементов очереди обычно бывает меньше $10^{\Delta p}$. Например, для представления суточного интервала изменения атрибута достаточно двадцати четырех часовых дискретов, и $(L=24) < 10^2$, хотя $\Delta p = 2$.

Затраты T_p времени на профилактическое регулирование, т.е. на преобразование одного j-го фрагмента записи: $T_p = t_w \left(2 \cdot 10^{\Delta p_j} + 3N \right)$.

Затраты T_m времени на вычисление длины пути доступа к очереди при введенных допущениях, позволяющих рассматривать процесс образования и удаления записей как эрланговский: $T_m = \sum_{j=1}^{J} \left(1 + \vartheta^2\right) \cdot \lambda \cdot T \cdot t_w = \left(1 + \vartheta^2\right) \cdot \rho \cdot t_w \cdot J$, где T есть среднее время пребывания записи в системе, λ – интенсивность потока вводов записей в систему, $\rho = \lambda T$ – загрузка системы сохраняемыми записями, ϑ- коэффициент вариации, равный отношению среднеквадратического отклонения σ времени «жизни» записи к его среднему значению.

Вероятность q_p профилактического вмешательства при введенных допущениях определяется произведением ρ интенсивности λ поступления записей в систему на среднее значение T времени их пребывания в ней и установленным критическим уровнем L_{cj} пути доступа к очереди записей, усредненным по всем J атрибутам: $q_p = \frac{1}{J} \sum_{j=1}^{J} \frac{\rho^{L_{cj}} (1 - \rho)}{1 - \rho^{L_{cj}+1}}$. Аналогично, *вероятность q_o оперативного вмешательства* при введенных допущениях определяется произведением ρ интенсивности λ поступления записей в систему на среднее значение T времени их пребывания в ней и установленным пороговым уровнем L_{oj} вычисленной длины пути доступа к очереди, усредненным по всем J атрибутам: $q_o = \frac{1}{J} \sum_{j=1}^{J} \frac{\rho^{L_{oj}} (1 - \rho)}{1 - \rho^{L_{oj}+1}}$. Наконец, *частота q_m вычислений* длины пути доступа по каждому атрибуту совпадает с интенсивностью λ потока поступ-

ления записей. В случае дисциплины периодических проверок $q_m = t/\Delta\hat{t}$, где t есть интервал наблюдения, $\Delta\hat{t}$ – выбранное значение шага вычислений.

2.3. Сопровождение файлов записей в системе реального времени

2.3.1. Постановка задачи. Ведение динамических (изменчивых) файлов, к классу которых принадлежат постоянно обновляемые файлы полетной информации, представляет собой одну из трудоемких задач управления БД. Стохастический процесс работы системы включает операции удаления устаревших записей; обновление сегментов сохраняемых записей, приводящее к изменению их длины; ввод новых записей. Перечисленные события порождают корректировку содержимого индексов и списков поиска. В файлах записей постоянной длины задача упрощается тем, что на место удаленной записи можно вставить вновь вводимую, или одновременно с удалением устаревшей записи на освободившееся место перенести самую последнюю (*N*-ю, *N* – количество записей в файле). При этом все затруднения, связанные с задачей сопровождения данных, ограничиваются поддержанием адекватности содержимого записей его отображению в списках и в индексах, а также рациональной организацией областей переполнения. В файлах записей переменной длины проблема усугубляется появлением неконтролируемых пустот в теле файла при удалении записей. Освободившееся место в общем случае нельзя заполнить новой записью, так как их размеры чаще всего не совпадают по величине. Традиционной схемой [2] ведения изменчивого файла запи-

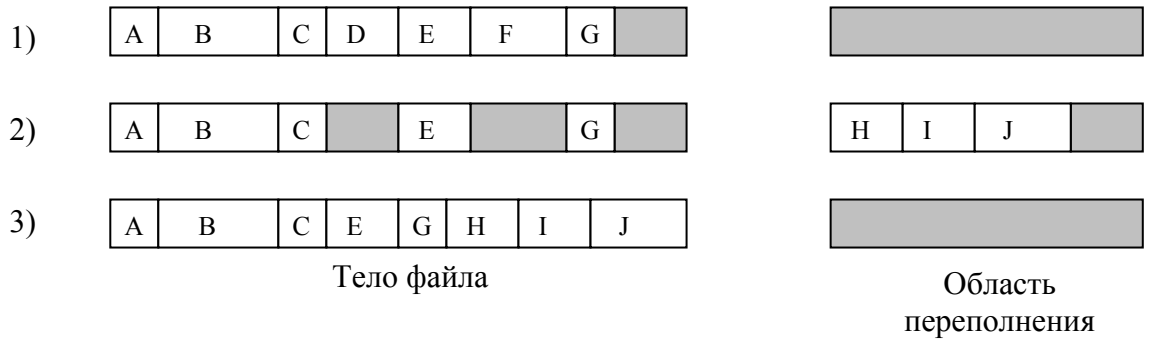


Рис. 2.2. Традиционная схема ведения изменчивого файла записей переменной длины

(1 – начальное размещение, 2 – размещение после ввода трех новых и удаления двух устаревших записей, 3 – размещение после уплотнения)

сей переменной длины является периодическое (или по мере заполнения) сжатие свободных полей в пространстве файла (рис. 2.2).

Все записи последовательно переносятся вплотную к предыдущим, начиная с первой. Списки и индексы поиска формируются заново. На время уплотнения доступ к файлу блокируется. Основные усилия специалистов, проектирующих алгоритмы управления БД, или пользующихся системой, не обеспеченной средствами такой поддержки, направлены на рациональную организацию распределения сегментов записей в областях переполнения.

Весьма болезненно поддаются процедурам ведения сцепленные мультисписки поиска, требующие громоздких операций корректировки адресов и взаимных ссылок, осложняющихся еще более в случае двунаправленных цепей и колец. На возрастающий объем организационных операций накладывается низкая надежность сцепленных структур, разрушающихся при искажении или потере одного из звеньев цепочки указателей. Известные схемы ведения файлов записей переменной длины основаны на подходе к задаче как к целенаправленному противостоянию неизбежным издержкам, как к борьбе против потерь ресурсов памяти и производительности, сопутствующих работе систем управления БД и снижающих их эффективность. Следствием такого рассмотрения становится необходимость уплотнения, объединяющего пустоты в теле файла, в обобщенное поле памяти, выносимое в конец, за границу значащего содержимого, что позволяет размещать в нем вновь поступающие записи.

Обсудим противоположный подход к задаче, отвергающий стратегию эпизодических восстановлений файла. Цикл непрерывной работы ПО АС УВД не допускает блокировок доступа к данным. Взамен предлагается использовать складывающуюся в процессе работы ситуацию динамического равновесия между подвижными границами перемежающихся свободных и заполненных областей изменчивого файла. Пустоты файла будем рассматривать как записи специфического типа – фиктивные. Такой подход позволяет исключить мероприятия по склеиванию пустот файла в единое поле. Появляется возможность ориентироваться на их сосуществование с действительными записями как равноправных с ними. Эти фиктивные записи должны сопровождаться системой на общих основаниях, т.е. сортироваться и отображаться в индексах и списках для удобства поиска. Намеченный подход освобождает от необходимости переформирования файлов, индексов и списков в широком классе задач организации данных.

Для пояснений обратимся к примеру БД полетной информации, содержащей сведения о ВС, совершающих полеты. Как и во всяком описании объектов, в одних графах (сегментах) записей имеются данные различного объема, в других – пустоты (отсутствие информации), в результате чего записи имеют неодинаковую длину. При посадке соответствующие записи удаляются, при вылетах образуются новые. Со временем содержимое записей корректируется диспетчерами. При этом изменение, например, высоты или скорости влечет за собой замену лишь количественного значения атрибута, не приводящую к изменению длины записи, в то время как направление на обходной маршрут, отказы оборудования и другие происшествия связаны с увеличением или уменьшением этой длины. Тогда обновленную запись невозможно оставлять на старом месте и ее приходится перемещать.

2.3.2. Формализация задачи. Исходя из сказанного, задачу ведения изменчивого файла записей переменной длины сформулируем следующим образом. Пусть мы имеем БД, контролирующую N записей, каждая из которых содержит J атрибутов, описывающих сопровождаемые объекты. Факти-

ческое количество $j_n \le J$ атрибутов, принадлежащих каждой n-й записи ($n=\overline{1,N}$), есть случайная величина с известным математическим ожиданием (либо с рассчитанным эмпирически средним значением). Выяснение характера распределения величины j_n в рамках данного изложения не представляет интереса, хотя трудно найти убедительные аргументы против ее аппроксимации гауссовым законом. Плотность распределения важна при формировании входного индекса, дискреты которого порождают цепи указателей атрибутов разной величины; в данном случае, напротив, каждому типу записей сопоставлен собственный дискрет входного индекса. Для каждой конкретной задачи известны из практики средние величины интенсивности ввода новых записей (частота вылетов) и удаления устаревших (посадки) с учетом нештатных ситуаций, интерпретируемых как последовательно исполняемые операции удаления первоначальной и ввода обновленной записи.

Последняя интерпретация порождает обсуждаемый ниже принцип ведения неоднородных изменчивых (динамических) файлов записей. Если корректировка записи не привела к изменению ее длины l (например, в графе «скорость полета» вместо числа 900 записывается 910), то обновленная запись может быть возвращена на свое место, а не фиксироваться вслед за последней N-й записью БД. Более того, она может быть помещена в любом свободном поле (пустоте) длиной l в теле файла, если информация о таких пустотах сопровождается системой управления БД. И, наконец, если подобного рода сопровождение осуществляется, то и при изменении величины l длины корректируемой записи до $m \ne l$ нет необходимости выносить обновленную запись на ($N+1$)-е место в файле. Нужно найти свободную область (пустоту) размером m в теле файла и разместить в ней скорректированную запись. Точно так же при вводе новой записи произвольной длины m, она должна размещаться не на ($N+1$)-м месте или в области переполнения, а в свободном поле длиной m в теле файла. Информация о наличии свободных полей при этом также корректируется, т.е. из списка пустот вычеркивается заполняемое свободное место. При удалении устаревшей записи в этот список вносится адрес освобождаемого поля памяти.

В изложенной постановке удается отказаться от принципа ведения изменчивых файлов записей переменной длины на основе резервирования в пространстве БД областей переполнения и периодическим (или по мере заполнения) уплотнением файла. Она дает возможность рассматривать пустоты в теле файла как специфический тип записей, образуемых при удалении действительных, равноправных с ними и сопровождаемых обычными схемами организации данных. Для построения такой схемы необходимо определить вероятностные характеристики модели ее функционирования.

Найдем количественную оценку объема памяти, необходимого для реализации процедуры сопровождения фиктивных записей (свободных полей) в изменчивом файле. Входной поток записей, вводимых в БД, удовлетворяет условию стационарности, что следует из рассмотрения приведенного примера. Менее очевидно соблюдение в такой системе условий ординарно-

сти потока и отсутствия последействия. Каждую операцию обновления данных в записях удобно интерпретировать как последовательные удаление и ввод, что на первый взгляд противоречит условию ординарности, если не учитывать, что это – разные события, принадлежащие двум самостоятельным потокам. В целом, как и в классическом примере анализа работы телефонных станций, где соблюдение всех трех условий постулируется с еще большей натяжкой, входной поток записей, вводимых в систему, можно аппроксимировать простейшим (пуассоновским). По аналогии с распределением длительности телефонных переговоров, время пребывания записи в БД аппроксимируется показательным распределением. Тогда процесс функционирования системы, состоящий из поступления в нее записей, пребывания и удаления, можно описать марковской цепью сообщающихся соседних состояний. Если задать при этом в качестве ограничения допустимую вероятность события, при котором очередная запись, поступая в систему, не находит в ней места для размещения, то схему ведения изменчивого файла нетрудно вписать в рамки модели Эрланга для системы, насчитывающей n сообщающихся соседних состояний. В общепринятых обозначениях имеем параметры модели: $\lambda[сек^{-1}]$ – интенсивность входного потока; $\mu[сек^{-1}]$ – частота, с которой записи удаляются из системы; n – количество сообщающихся состояний, соответствующих действительным записям; $n = \overline{0,N}$; N – количество записей, которое можно разместить в файле; $\rho = \lambda / \mu$ – загрузка системы. Нетрудно получить выражение для вычисления вероятности P_n любого из сообщающихся состояний системы через величины загрузки и вероятности P_0 отсутствия записей в файле (см. параграф 3.2.2): $P_n = \rho^n P_0$, где P_n – вероятность состояния, при котором в БД сопровождается ровно n записей. Учитывая, что величина n пробегает все значения от 0 до N и сумма вероятностей всех состояний равна единице $\sum_{n=0}^{N} P_n = 1$, определяем вероятность отсутствия каких бы то ни было записей в файле из $P_0 \sum_{n=0}^{N} \rho^n = 1$: $\sum_{n=0}^{N} \rho^n = \frac{(1-\rho^{N+1})}{(1-\rho)}$; $P_0 = \frac{(1-\rho)}{(1-\rho^{N+1})}$.

Отсюда вероятность P_n состояния, при котором заняты все N мест для записей, и новую информацию негде размещать, выразится как

$$P_N = \rho^N \frac{(1-\rho)}{(1-\rho^{N+1})}.$$

При заданном пороговом значении P_n усредненный объем (количество N записей) файла вычисляется элементарными преобразованиями:

$$\rho^N = P_N/(1 - \rho + \rho P_N);$$

$$N \ln\rho = \ln P_N - \ln(1 - \rho + \rho P_N);$$

$$N = [\ln P_N - \ln (1 - \rho + \rho P_N)] / \ln \rho.$$

Для перехода от средних значений к вероятностным мерам воспользуемся известной зависимостью наиболее вероятного значения L количества занятых мест для записей от усредненной длины M в эрланговских системах:

$L = (1+\upsilon^2)M$, где υ – коэффициент вариации, равный отношению среднеквадратического отклонения случайной величины к ее математическому ожиданию. Тогда выражение для расчета количества N записей усредненной длины, которое должен размещать файл для обеспечения заданного уровня вероятности P_N отсутствия свободного места, определится как

$$]N[= (1+\upsilon^2)[\ln P_N - \ln (1 - \rho + \rho P_N)]/ \ln \rho,$$

где символ][обозначает ближайшее большее целое вычисленной величины наиболее вероятного количества мест, необходимого для нормальной работы процедуры ведения изменчивого файла записей переменной длины.

2.3.3. Алгоритм ведения изменчивого файла записей переменной длины основан на двух допущениях:

- количество типов (значений l длины) записей в файле ограничено и равно r;
- интенсивность λ поступлений записей не превосходит частоты μ их удалений с учетом возможных перераспределений записей из одного типа в другой при их обновлении.

Первое допущение позволяет интерпретировать поля переменной длины, освобождающиеся при удалении устаревших записей, как специфические фиктивные записи, характеризуемые единственным атрибутом – своей длиной. Ограничение r на количество типов дает возможность пронумеровать их, присвоив каждому свой номер i, $i=\overline{1,r}$, значение которого совпадает со значением атрибута данной фиктивной записи. Тогда ко всей совокупности фиктивных записей изменчивого файла применимы обычные способы их организации, и задачей построения алгоритма становится выбор наиболее компактной схемы сортировки и поиска. В качестве такой схемы в рамках данного исследования естественно использовать одну из модификаций метода непосредственной расстановки, рассмотренного выше.

Второе допущение, уже учтенное при формализации модели ($\lambda < \mu$, иначе $\rho \geq 1$ и ряд $\sum_{n=0}^{N} \rho^n$ расходящийся), говорит, что обновление записи допускает изменение ее длины.

В анализировавшемся примере с файлом полетных данных длина каждой записи о ВС, как правило, с течением времени увеличивается. Записи переходят из типа в тип, пока не совершается посадка или выход из зоны действия системы. Процесс убытия в среднем компенсируется вылетами новых бортов. Положенный в основу алгоритма метод непосредственной расстановки сцепленных указателей адресов записей, содержащих равновеликие атрибуты, в приложении к задаче ведения изменчивых файлов выступает эффективным инструментом управления данными как с точки зрения минимизации необходимых ресурсов памяти и производительности, так и в отношении технологичности пользования.

Область данных алгоритма составляют совокупность свободных полей файла, образованных после удаления устаревших записей, и индекс входа в цепи указателей их адресов, встроенных в пустующие фиктивные записи.

Дискреты индекса, пронумерованные от единицы до *r*, соответствуют номерам типов записей, для управления которыми используется БД. В каждом дискрете фиксируется единственная величина: адрес свободного поля файла, запись из которого была удалена последней среди удаленных записей данного типа. Если свободных полей для записей *i*-го типа нет, то в соответствующем дискрете устанавливается некорректный код. Такая ситуация характерна для начальной фазы работы системы, пока ни одна запись еще не удалялась. Вновь вводимые записи располагаются при этом (некорректный код в индексе) вслед за последней записью файла.

В стационарном режиме в БД сопровождаются записи *r* типов, часть из которых фиктивные (удаленные). В дискретах входного индекса зафиксированы адреса свободных полей, или фиктивных записей соответствующего типа. Внутри фиктивных записей хранятся сцепленные со входным индексом указатели следующих фиктивных записей того же типа. Указатель, размещенный в последней записи цепи данного типа либо некорректный, либо закольцованный, либо содержит другую избыточную информацию для самоконтроля или других целей. В качестве некорректного кода при действительной (или при относительной) адресации может использоваться любое нечетное число, так как адресуемые поля памяти выровнены на границу двух байтов. При косвенной (символической) адресации обращение к образующим элементам производится в соответствии с их порядковыми номерами, половина среди которых – нечетные, и некорректным кодом адреса может служить величина, превышающая объем памяти системы.

2.3.4. Пример реализации. Пусть БД сопровождает фиктивные записи пяти типов: длиной 64, 128, 192, 256 и 320 байтов. В изображенной на рис. 2.3 схеме изменчивый файл содержит действительные записи, размещенные в нем в порядке поступления, и фиктивные записи, т.е. свободные поля, оставшиеся после удаления устаревших записей. Информация об устаревших записях сосредоточена в пяти (по количеству типов записей) дискретах входного индекса, каждый из которых порождает цепь указателей адресов следующих фиктивных записей того же типа, встроенную внутрь свободных полей. На рис. 2.3 стрелками указана одна из таких цепей, сформированная для фиктивных записей четвертого типа (длиной 256 байтов); нетрудно построить аналогичные цепи для других типов записей. В четвертом дискрете входного индекса, номер которого совпадает с номером типа фиктивной записи, указан адрес (смещение 13312 от начала файла) свободного пространства длиной 256 байтов. Внутрь этого поля встроен указатель адреса продолжения цепи, направляющий к следующему (в инвертированном порядке удаления из системы) звену цепи, т.е. к фиктивной записи длиной 256 байтов, расположенной со смещением 5760 от начала файла. В свою очередь, и в ней зафиксирован указатель, адресующий к 256-байтному полю, свободному от записей и размещенному со смещением 9856. Следующая фиктивная запись находится по смещению 1600, далее 4416, 3044 и 5184. Этот адрес указывает на смещение последней фиктивной записи четвертого типа, имеющейся в

файле; значение встроенного в нее указателя некорректно.

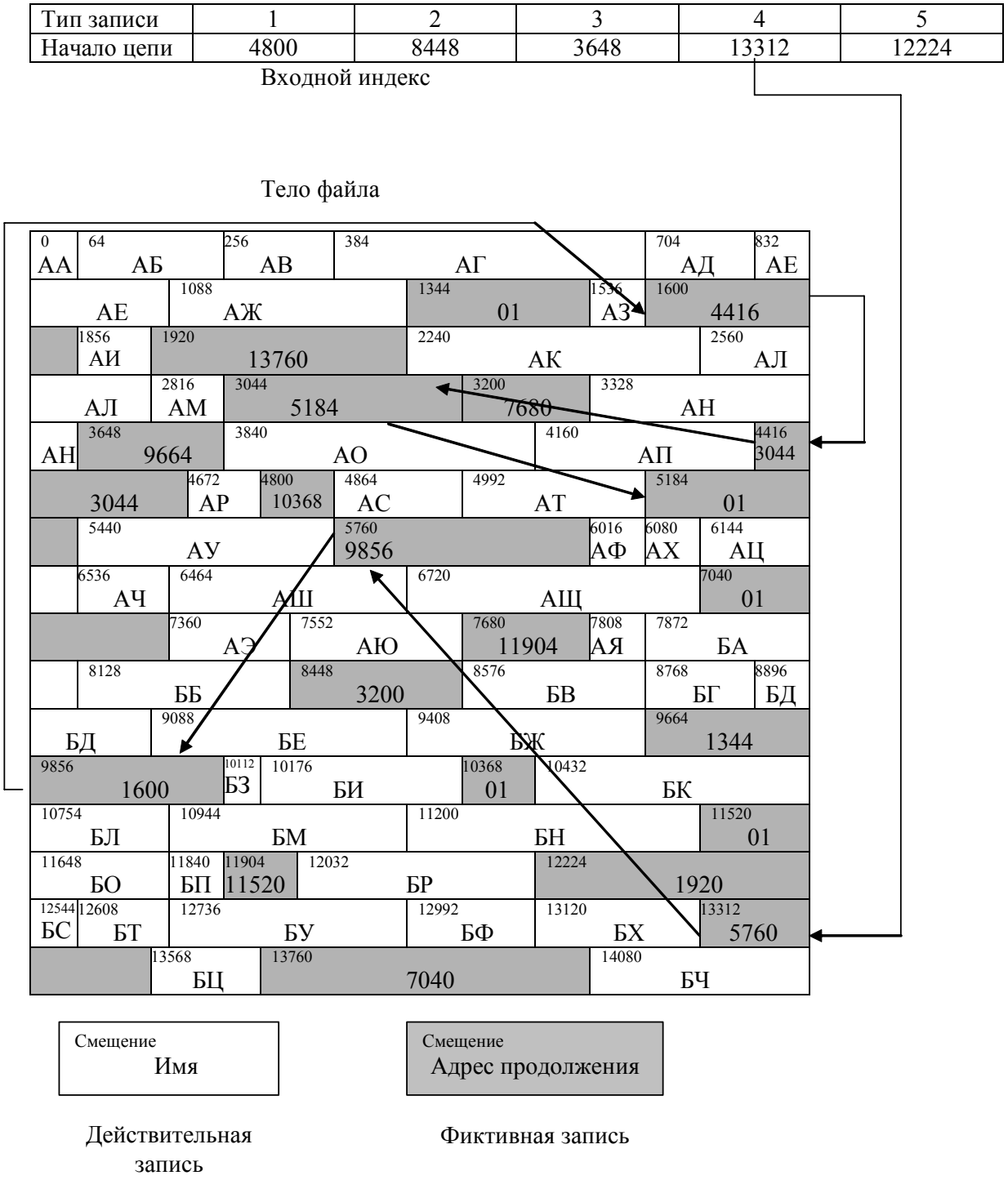


Рис. 2.3 Сцепление фиктивных записей

Процесс формирования цепи происходит по традиционной схеме сцепления и изображен на рис. 2.4, поясняющем ввод новой записи длиной 256 байтов, и на рис. 2.5, иллюстрирующем удаление такой же записи. При вводе исполняется обращение к четвертому дискрету (по типу записи) входного индекса. Содержимое дискрета указывает адрес свободного 256-байтного поля файла, необходимого для размещения поступающей записи. Встроенный в

это занимаемое поле указатель, хотя бы и некорректный, переписывается в четвертый порождающий дискрет входного индекса, вытесняя из него адрес заполняемой области файла.

Сопоставляя рис. 2.4 и 2.5, нетрудно видеть изменение содержимого четвертого дискрета входного индекса в процессе работы алгоритма. Старое значение, хранившееся до ввода новой записи, вытесняется. Длинная пунктирная стрелка указывает свободную область памяти, в которую вводится новая запись с именем БШ. Короткая пунктирная стрелка направлена к следующему по схеме рис. 2.4 звену цепи фиктивных записей. После ввода новой записи это звено становится начальным, и его адрес, ранее встроенный в заполняемое поле, переписывается из него в четвертый дискрет индекса.

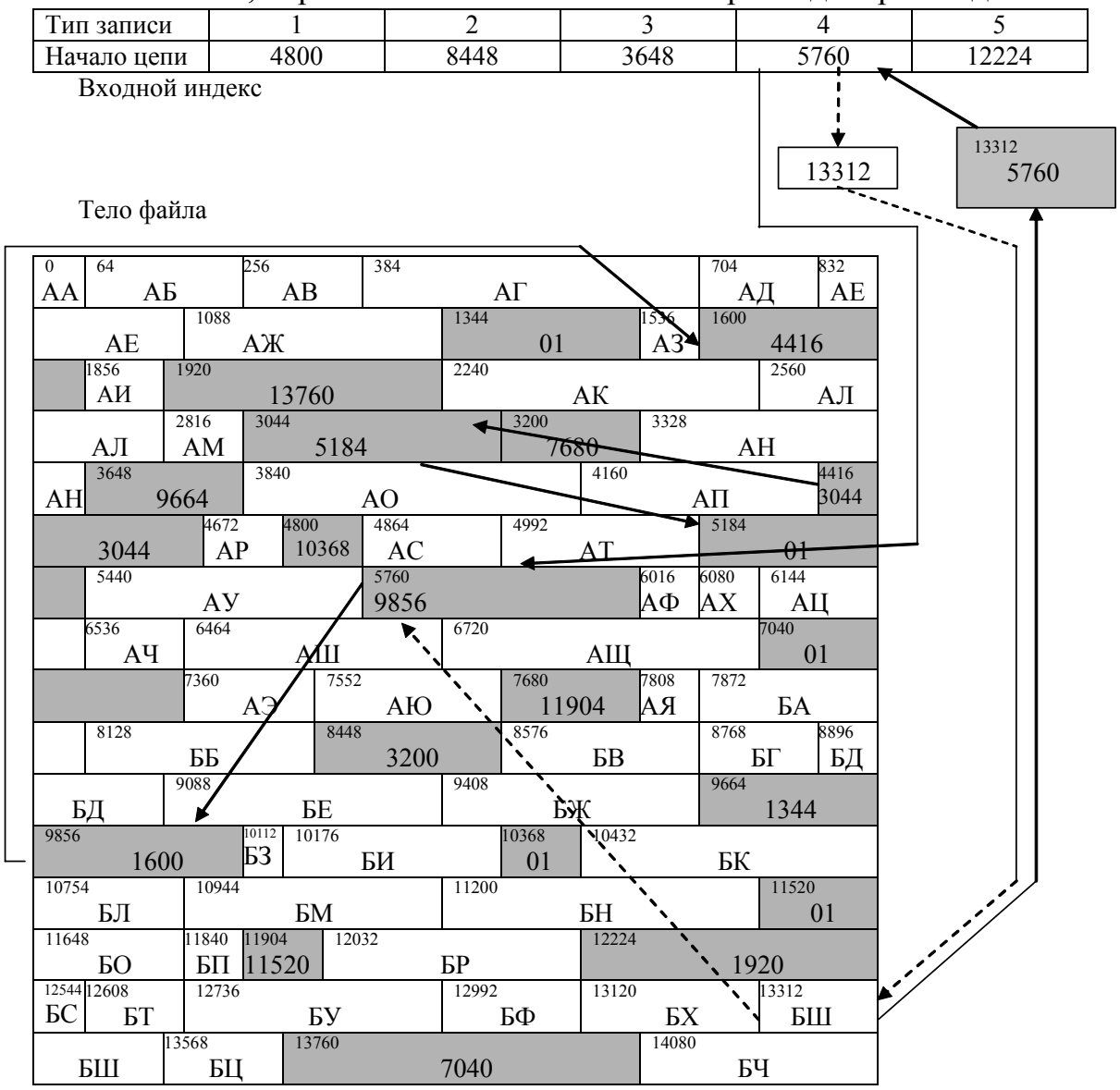


Рис. 2.4. Размещение новой записи

После удаления очередной устаревшей записи освобождаемое поле обнуляется, и в него из дискрета входного индекса, номер которого равен номеру типа удаляемой записи, переписывается содержимое этого дискрета, хотя бы и некорректное. В «освобожденный» таким образом дискрет зано-

сится адрес фиктивной записи файла, образуемой вновь вследствие удаления действительной. Рис. 2.5 поясняет процесс присоединения к цепи новой фиктивной записи. В отличие от состояния файла на рис. 2.4, здесь освобождается область, равная по величине 256 байтам и расположенная со смещением от начала 9408. Величина этого смещения заносится в четвертый дискрет входного индекса. Вытесненное прежнее содержимое, т.е. указатель адреса, хотя бы и некорректный, встраивается внутрь вновь образованного свободного поля. Пунктирная стрелка указывает направление поиска по цепи рис. 2.4, а сплошная присоединяет к ней указатель на новую фиктивную запись.

Схема реализованного в алгоритме метода для сопровождения измен-

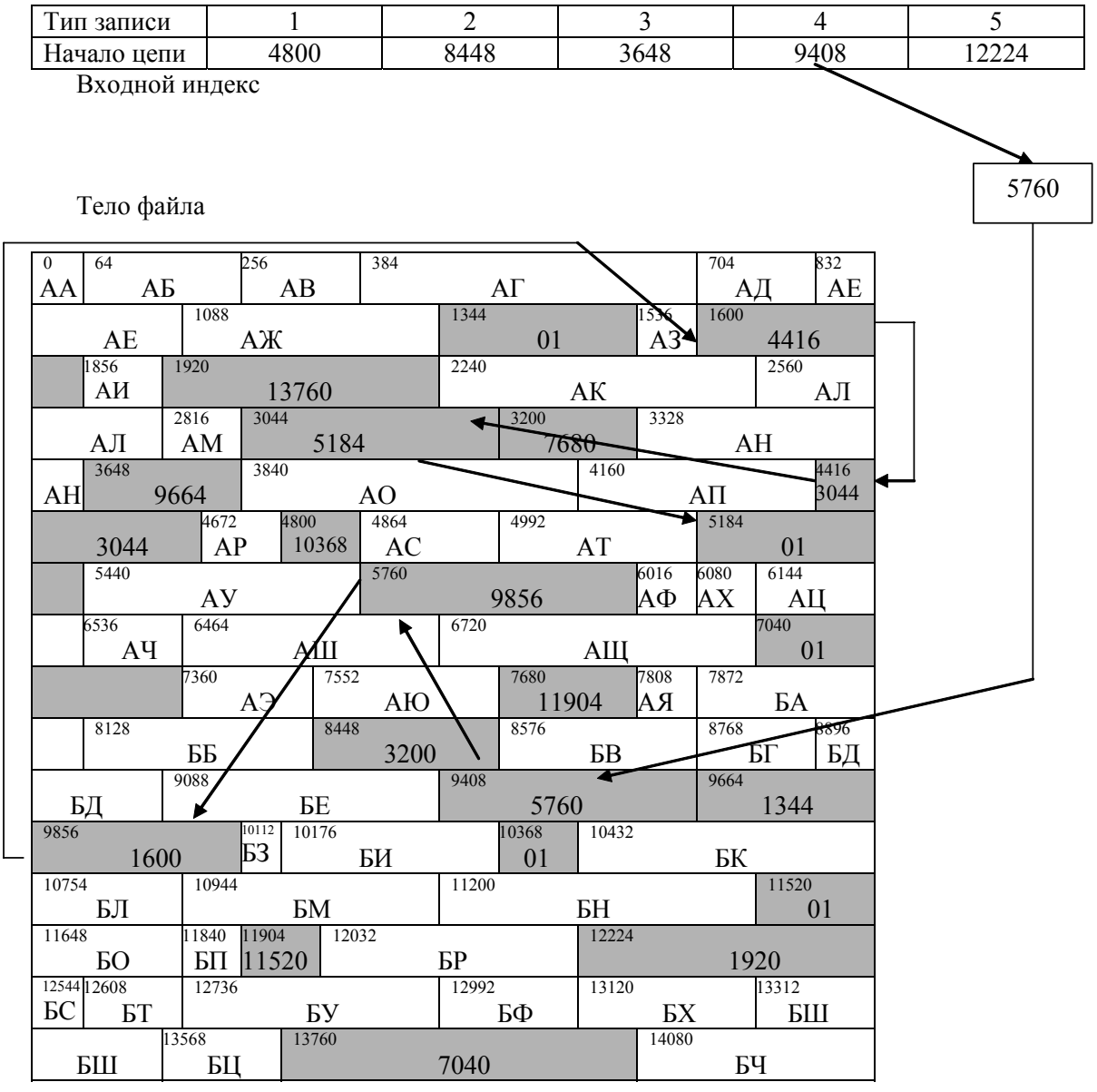


Рис. 2.5. Удаление устаревшей записи

чивых файлов записей переменной длины разветвляется, как следует из сказанного, на две технологические последовательности, включаемые либо при удалении, либо при вводе очередной записи БД соответственно. В первом случае она состоит из следующих операций:

- обнуление поля файла, содержащего удаляемую запись;

• пересылка из дискрета входного индекса, соответствующего типу удаляемой записи, его содержимого, хотя бы и некорректного, в адрес начала удаляемой действительной (т.е. образуемой фиктивной) записи;

• фиксация адреса удаляемой записи в освободившемся дискрете входного индекса.

Последовательность операций при вводе новой записи:

• обращение к дискрету входного индекса, номер которого равен номеру типа вводимой записи;

• присоединение новой записи к нижней границе файла – выполняется, если содержимое выбранного дискрета некорректно, т.е. свободных полей (фиктивных записей) данного типа нет в системе, и новая запись размещается вплотную вслед за *N*-ой;

• присоединение новой записи внутри файла – выполняется в противном случае, если содержимое выбранного дискрета корректное и можно произвести обращение к фиктивной записи, адрес которой хранится в анализируемом разряде входного индекса;

• перенос из выбранного свободного поля упакованного в нем адреса, хотя бы и некорректного, следующего свободного поля той же длины в дискрет входного индекса, из которого произведено обращение в тело файла;

• ввод новой записи в выбранное свободное поле файла.

Метод ведения изменчивых файлов записей переменной длины прост в реализации и сводит к минимуму затраты на поддержание работоспособности БД за счет исключения избыточных по смыслу задачи областей переполнения и блокировок доступа к записям в периоды сжатия файлов. Его модификации можно использовать для так называемого фонового уплотнения, исполняемого периодически включаемыми программами сопровождения файлов БД. Основная задача таких программ – контроль и самовосстановление при обнаружении искажений информации. Наряду с этой функцией на них можно дополнительно возложить перенос ограниченного количества последних записей файла в соответствующие по размеру пустоты с целью «склеивания» свободного пространства в непрерывный участок. Необходимость в такой организации возникает в БД, обслуживающих нерегулярные потоки быстро изменяющихся записей, допускающие резкие колебания интенсивности поступления записей в систему. К этому классу относятся КП АС УВД, сопровождающие наборы полетных данных, которые представляют собой динамические файлы записей переменной длины.

Вопросы для самопроверки

1. Дайте определения файловой системы и файла. Какая практическая потребность вызвала к жизни функцию управления файлами (п. 2.1.1)?

2. В чем состоит особенность проблемы сохранения и восстановления данных на физических носителях? Каковы способы ее решения (п. 2.1.2)?

3. Как вы понимаете задачу поддержания информационной целостно-

сти АС УВД и в чем ее отличие от ссылочной целостности БД (п. 2.1.3)?

4. Подготовьте критический анализ предложенной в п. 2.2 схемы нейтрализации рассогласований полетной информации в реальном времени. С какой целью в ней построены три ступени восстановления данных (контроль целостности, синхронизация изменений, защита от искажений)?

5. На каких допущениях построена модель векторного процесса управления восстановлением данных в реальном времени (п. 2.2.2)?

6. Сформулируйте проблему ведения динамических файлов (п. 2.3.1).

7. Предложите алгоритм ведения динамических файлов, использующий принципы, отличные от предложенных в п. 2.3.3 сцепленных структур.

3. Управление сетевыми ресурсами

3.1. Особенности задачи управления ресурсами

3.1.1. Основные определения. От эффективности алгоритмов управления локальными ресурсами компьютеров во многом зависит качество работы сетевой ОС в целом. Реализация функций управления процессорами, памятью, внешними устройствами каждого компьютера составляет основу традиционной классификации по следующим признакам [2].

Поддержка многозадачности. По числу одновременно выполняемых задач ОС делятся на однозадачные и многозадачные. Первые предоставляют пользователю виртуальную машину, делая более простым и удобным процесс взаимодействия пользователя с компьютером. Однозадачные ОС включают средства управления периферийными устройствами, управления файлами, общения «человек-машина». Многозадачные ОС, кроме того, управляют разделением совместно используемых ресурсов сети.

Поддержка многопользовательского режима. По числу одновременно работающих пользователей ОС делятся на однопользовательские и многопользовательские. Главным отличием многопользовательских систем является наличие средств защиты информации каждого пользователя от несанкционированного доступа других пользователей. Очевидно, что не всякая многозадачная система является многопользовательской, и не всякая однопользовательская ОС является однозадачной.

Вытесняющая и невытесняющая многозадачность. Важнейшим разделяемым ресурсом является процессорное время. Способ распределения процессорного времени между несколькими одновременно действующими в системе процессами (или нитями) во многом определяет специфику ОС. Основным различием между вытесняющим и невытесняющим вариантами является степень централизации планирования процессов. В первом случае механизм планирования процессов целиком сосредоточен в ОС, а во втором – распределен между системой и прикладными программами. При невытесняющей многозадачности активный процесс выполняется до тех пор, пока он сам, по собственной инициативе, не отдаст управление ОС для того, чтобы та выбрала из очереди другой готовый к выполнению процесс. При вытесняю-

щей многозадачности решение о переключении процессора с одного процесса на другой принимается ОС, а не самим активным процессом. Важным качеством системы становится возможность распараллеливания вычислений в рамках одной задачи. *Многонитевая* ОС разделяет процессорное время не между задачами, а между их отдельными ветвями (нитями).

Многопроцессорная обработка. Другим важным свойством ОС является наличие или отсутствие средств поддержки многопроцессорной обработки или *мультипроцессирование*. Мультипроцессирование приводит к усложнению всех алгоритмов управления ресурсами. В АС УВД поддержка многопроцессорной обработки данных становится общепринятой.

ОС, работающие с многопроцессорной архитектурой, классифицируют по способу организации вычислительного процесса на асимметричные и симметричные. Первые целиком выполняются только на одном из процессоров, распределяя прикладные задачи по остальным процессорам. Вторые полностью децентрализованы и используют все процессоры, разделяя их между системными и прикладными задачами. Помимо процессоров, важное влияние на характеристики ОС в целом, на возможности ее использования в АС УВД оказывают особенности других подсистем управления локальными ресурсами – подсистем управления памятью, файлами, устройствами ввода-вывода.

На протяжении существования процесса его выполнение может быть многократно прервано и продолжено. Для возобновления процесса, необходимо восстановить режим работы процессора и состояние его операционной среды, которое фиксируется в регистрах и программном счетчике, указателями на открытые файлы, информацией о незавершенных операциях ввода-вывода, кодами ошибок выполняемых данным процессом системных вызовов и т.д. Эта информация называется *контекстом процесса*.

Специфика ОС проявляется и в том, каким образом она реализует сетевые функции: распознавание и перенаправление в сеть запросов к удаленным ресурсам, передача сообщений по сети, выполнение удаленных запросов. При реализации сетевых функций возникает комплекс задач, связанных с распределенным характером хранения и обработки данных в сети, т.е. *мониторинг* состояния всех доступных ресурсов и серверов, адресация взаимодействующих процессов, обеспечение прозрачности доступа, тиражирование данных, согласование копий, поддержка безопасности данных.

3.1.2. Параллельная обработка информации. В ОСРВ заложен параллелизм, возможность одновременной обработки нескольких событий, поэтому все они являются многозадачными (многопроцессными, многонитиевыми). Для того чтобы уметь оценивать накладные расходы системы при обработке параллельных событий, необходимо знать время, которое система затрачивает на передачу управления от процесса к процессу (от задачи к задаче, от нити к нити), т. е. время переключения контекста. При вытесняющей многозадачности механизм планирования задач целиком сосредоточен в ОС, и программист пишет свое приложение, не заботясь о том, что оно будет вы-

полняться параллельно с другими задачами. При этом операционная система выполняет следующие функции: определяет момент снятия с выполнения активной задачи, запоминает ее контекст, выбирает из очереди готовых задач следующую и запускает ее на выполнение, загружая ее контекст.

При невытесняющей многозадачности механизм планирования распределен между ОС и прикладными программами. Прикладная программа, получив управление, сама определяет момент завершения своей очередной итерации и возвращает управление ОС с помощью какого-либо системного вызова, а ОС формирует очереди задач и выбирает в соответствии с некоторым алгоритмом (например, с учетом приоритетов) следующую задачу на выполнение. Такой механизм создает проблемы как для пользователей, так и для разработчиков.

Для пользователей это означает, что управление системой теряется на произвольный период времени, который определяется приложением (а не пользователем). Если приложение тратит слишком много времени на выполнение какой-либо работы, например, на обработку плана полета, система не может переключиться с этой задачи на другую задачу, например, на обработку радиолокационных измерений, а задача планирования продолжалась бы в фоновом режиме. Такая ситуация нежелательна, так как диспетчер за пультом не должен ждать, когда машина завершит не самые важные задачи.

По этой причине разработчики приложений для невытесняющей операционной среды, возлагая на себя функции планировщика, должны создавать приложения так, чтобы они выполняли свои задачи небольшими порциями (квантами). Например, программа обнаружения конфликтных ситуаций может обработать данные о взаимном положении части ВС и вернуть управление системе. После выполнения более срочных задач система возвратит ей управление, чтобы она продолжила работу. Подобный метод разделения времени между задачами существенно затрудняет разработку приложений и предъявляет повышенные требования к квалификации программиста. Программист должен обеспечить «дружественное» отношение своей программы к другим выполняемым одновременно с ней программам, достаточно часто отдавая им управление. Крайним проявлением «недружественности» является зависание, которое приводит к информационному отказу системы. При вытесняющей многозадачности такие ситуации, как правило, исключены, так как центральный планирующий механизм снимет зависшую задачу.

Для этих целей современные ОСРВ предлагают использовать механизм *многонитевой обработки*. При этом понятие «процесс» в известной степени меняет смысл. Мультипрограммирование реализуется на уровне нитей, и задача, оформленная в виде нескольких нитей в рамках одного процесса, может быть выполнена быстрее за счет параллельного выполнения ее отдельных частей. Например, если суточный план использования воздушного пространства был разработан с учетом возможностей многонитевой обработки, то диспетчер планирования может запросить пересчет списка по своему сектору и одновременно продолжать вводить в систему новые планы. Особенно эф-

фективно можно использовать многонитевость для выполнения распределенных приложений, например, многонитевый сервер может параллельно выполнять запросы сразу нескольких диспетчеров.

Нити, относящиеся к одному процессу, не настолько изолированы друг от друга, как процессы в традиционной многозадачной системе, между ними легко организовать тесное взаимодействие. В отличие от процессов, которые принадлежат разным, вообще говоря, конкурирующим приложениям, все нити одного процесса всегда принадлежат одному приложению, поэтому программист может заранее продумать работу множества нитей процесса таким образом, чтобы они могли взаимодействовать, а не бороться за ресурсы.

Нередко бывает желательно иметь несколько нитей, разделяющих единое адресное пространство, но выполняющихся квазипараллельно, благодаря чему нити становятся подобными процессам (за исключением разделяемого адресного пространства). Нити иногда называют облегченными процессами или мини-процессами. Каждая нить выполняется строго последовательно и имеет свой собственный программный счетчик и стек. Нити, как и процессы, могут, например, порождать нити-потомки, могут переходить из состояния в состояние. Подобно традиционным процессам (т. е. процессам, состоящим из одной нити), нити могут находиться в состояниях: выполнение, ожидание и готовность. Пока одна нить заблокирована, другая нить того же процесса может выполняться. Нити разделяют процессор так, как это делают процессы в соответствии с различными вариантами планирования.

Однако нити в рамках одного процесса не настолько независимы, как отдельные процессы. Все такие нити имеют одно и то же адресное пространство и разделяют одни и те же глобальные переменные. Поскольку каждая нить имеет доступ к каждому виртуальному адресу, она может использовать стек другой нити. Между нитями нет полной защиты, это не нужно. Все нити одного процесса решают общую задачу одного процесса, и аппарат нитей используется для более быстрого решения задачи путем ее распараллеливания. Программисту важно иметь удобные средства организации взаимодействия различных частей одной задачи. Итак, нити имеют собственные: программный счетчик, стек, регистры, нити-потомки, состояние. Нити разделяют: адресное пространство, глобальные переменные, открытые файлы, таймеры, семафоры, статистическую информацию.

Многонитевая обработка повышает эффективность системы по сравнению с многозадачной обработкой. Например, в многозадачной среде можно одновременно работать с плановыми списками и редактором плановых сообщений. Однако если запрашивается пересчет списка по сектору, суточный план блокируется до тех пор, пока эта операция не завершится, что может потребовать значительного времени. В многонитевой среде в случае, если план разработан с учетом возможностей многонитевой обработки, предоставляемых программисту, этой проблемы не возникает, и все диспетчеры всегда имеют доступ к плану использования воздушного пространства.

Концепция многозадачности (псевдопараллелизм) является существен-

ной для ОСРВ с одним процессором, приложения которой должны быть способны обрабатывать многочисленные внешние события, происходящие практически одновременно. Концепция процесса, пришедшая из мира UNIX, плохо реализуется в многозадачной системе, поскольку процесс имеет тяжелый контекст. Возникает понятие потока, который понимается как подпроцесс, или легковесный процесс (*light-weight process*). Потоки существуют в одном контексте процесса, поэтому переключение между ними происходит очень быстро, а вопросы безопасности не принимаются во внимание. Потоки являются легковесными, потому что их регистровый контекст меньше, т.е. их управляющие блоки намного компактнее. Уменьшаются накладные расходы, вызванные сохранением и восстановлением управляющих блоков прерываемых потоков. Объем управляющих блоков зависит от конфигурации памяти. Если потоки выполняются в разных адресных пространствах, система должна поддерживать отображение памяти для каждого набора потоков.

В ОСРВ процесс распадается на задачи или потоки. В любом случае каждый процесс рассматривается как приложение. Между приложениями не должно быть избыточного взаимодействий, и в большинстве случаев они имеют различную природу – жесткого реального времени, мягкого реального времени, не реального времени. Функции, позволяющие осуществлять те или иные виды межпроцессорного взаимодействия, выполняют программные *менеджеры* семафоров, сообщений, событий, сигналов. Рассмотрим их работу на примере системы *Real-Time Executive for Multiprocessor Systems* (RTEMS). Это некоммерческая ОСРВ, созданная по заказу министерства обороны США для использования в системах управления ракетными комплексами. Система разработана для многопроцессорных комплексов и рассчитана на платформы MS-Windows и UNIX (GNU/Linux, FreeBSD, Solaris, MacOS X).

3.1.3. Очерк системы RTEMS. Ядро этой ОСРВ обеспечивает базовую функциональность, в его возможности входят: мультизадачная обработка; планирование, управляемое событиями; планирование с монотонной скоростью; взаимодействие задач и синхронизация; приоритетное наследование; управление ответным прерыванием; распределение динамической памяти; конфигурирование системы для уполномоченных пользователей; переносимость на многие целевые платформы.

Ядро отвечает за управление основной памятью компьютера и виртуальной памятью выполняемых процессов, за управление процессорами и планирование распределения процессорных ресурсов между совместно выполняемыми процессами, за управление внешними устройствами и, наконец, за обеспечение базовых средств синхронизации и взаимодействия процессов. При этом ядро использует соответствующие менеджеры. Привязка ОСРВ к аппаратуре производится с помощью специальной библиотеки подпрограмм BSP (*board support package*) и специализированных подпрограмм для различных архитектур. В состав BSP входят программа инициализации аппаратуры и драйверы устройств. Поддержка мультипроцессорных систем позволяет использовать ее для управления как однородными, так и неоднородными

системами. Ядро автоматически учитывает различия в архитектуре используемых процессоров, выполняя в случае необходимости перестановку байтов и другие процедуры. Это позволяет осуществлять переход на другое семейство процессоров без значительных изменений системы.

ОСРВ RTEMS можно рассматривать как набор компонентов, обеспечивающих ряд базовых сервисных функций для программ пользователя. Программный интерфейс приложения состоит из директив, распределенных по логическим наборам соответствующих менеджеров. Функции, используемые несколькими менеджерами, такие как распределение процессорного времени, диспетчеризация и управление объектами, реализованы в ядре. Ядро содержит также небольшой набор процедур, зависящих от типа используемого процессора: доступ к физической памяти, инициализация контроллера прерываний и периферийных устройств, специфичных для данного процессорного ядра, и т.д.

Предусмотрены следующие виды межпроцессорного взаимодействия:

- обмен данными между задачами;
- обмен между задачами и программами обработки прерываний;
- синхронизация между задачами;
- синхронизация задач и программ обработки прерываний.

Менеджеры семафоров, сообщений, событий, сигналов предназначены исключительно для осуществления межпроцессорного взаимодействия.

Менеджер семафоров. RTEMS поддерживает стандартные двоичные семафоры со счетчиками, обеспечивающие синхронизацию и монопольный доступ к ресурсам.

Менеджер событий. Служит для синхронизации выполнения задач. Флаг события используется задачей для того, чтобы информировать другую задачу о возникновении определенного события. Каждой задаче соответствуют 32 флага событий. Совокупность одного или более флагов называется набором событий. Одна задача может послать другой задаче набор событий, а также выяснить состояние набора событий соответствующей функцией.

Менеджер сообщений. Служит для обмена между задачами сообщениями переменной длины. Сообщения передаются через очереди типа FIFO (*first in – first out*, «первый пришел, первым обслужен»). Имеется возможность посылки срочного сообщения. Для каждой очереди задается максимальная длина сообщения. Сообщения могут использоваться для синхронизации задач. Задача может ожидать прихода определенного сообщения или проверять наличие сообщения в очереди.

Менеджер сигналов. Используется для асинхронного взаимодействия между задачами. Задача может включать в себя процедуру обработки асинхронного сигнала, которой передается управление при получении сигнала. Флаг сигнала используется задачей для того, чтобы проинформировать другую задачу о возникновении нештатной ситуации. Каждой задаче соответствуют 32 флага сигналов. Совокупность одного или более флагов называется набором сигналов.

Менеджер задач. Обеспечивает полный набор функций для создания, удаления и управления задачами. По замыслу необходима наименьшая последовательность команд, которая может самостоятельно конкурировать за использование системных ресурсов. Каждой задаче соответствует блок контроля TCB (*Task Control Block*). Это структура, которая содержит всю информацию, относящуюся к выполнению задачи. В процессе инициализации RTEMS выделяет TCB для каждой задачи в системе. Элементы TCB изменяются в соответствии с системными вызовами, которые выполняются приложением в ответ на внешние запросы. Блок TCB – это единственная внутренняя структура данных, доступная приложению через дополнительные процедуры. При переключении задач в TCB сохраняется контекст задачи. При возвращении управления задаче ее контекст восстанавливается. При перезапуске задачи исходный контекст восстанавливается в соответствии со стартовым контекстом, хранящемся в TCB. Задача может находиться в одном из пяти состояний: выполнение; готовность к выполнению (управление может быть передано задаче); остановка (задача заблокирована); ждущий режим (созданная, но не запущенная задача); отсутствие (задача не создана или удалена).

3.2. Модель вычислительного процесса в центре управления полетами

3.2.1. Оценка эффективности параллельной обработки.

Основным критерием качества работы ОСРВ служат показатели ее способности своевременно реагировать на все события, составляющие управляемый процесс, в темпе их наступления. События классифицируются по уровню важности для выполнения функций АС УВД и ранжируются по приоритетам исходя из замысла проекта: обеспечения безопасности, экономичности и регулярности полетов. Для каждого события устанавливаются допустимые значения вероятности того, что информация о нем будет потеряна, а также допустимые значения среднего времени ожидания его обработки и потерь времени на организацию параллельных вычислений.

ПО АС УВД должно выполнять функции обработки измеренных данных о движении ВС, поддержки принятия диспетчерских решений и выработки управляющих воздействий в жестком режиме реального времени, обработки плановой информации и метеорологических прогнозов – в мягком реальном времени, вести ряд работ в фоновом режиме. Для выяснения предельных возможностей систем параллельного счета отвлечемся от потерь ресурсов на взаимодействие процессоров. Рассмотрим математическую модель совместной работы произвольного количества компьютеров. Задачами такого рода традиционно занимается теория очередей (теория массового обслуживания – ТМО). Напомним ее основные положения.

ТМО – это раздел математической теории случайных процессов, занимающийся изучением моделей реального обслуживания с учетом случайного характера спроса и предоставления услуг. Другими словами, это математическая дисциплина, изучающая системы, предназначенные для обслуживания

массового потока требований случайного характера (случайными могут быть как моменты появления требований, так и затраты времени на их обслуживание). Типичным примером объектов исследования являются автоматические телефонные станции, индустрия сервиса, компьютерные сети, на которые случайным образом поступают «заявки» – вызовы абонентов, приходы клиентов. «Обслуживание» состоит в соединении абонентов, поддержании связи во время разговора и т. д. Целью развиваемых методов является отыскание разумной организации обслуживания, обеспечивающей заданное качество. С этой точки зрения ТМО рассматривают как ветвь исследования операций.

ТМО широко использует аппарат теории вероятностей и математической статистики. Ее задачи, сформулированные математически, обычно сводятся к изучению случайных процессов специального типа. Исходя из заданных вероятностных характеристик поступающего потока заявок и продолжительности их обслуживания, в зависимости от схемы системы (например, от наличия отказов или очередей), ТМО определяет соответствующие характеристики качества обслуживания. Это вероятность отказа, среднее время ожидания начала обслуживания, среднее время простоя компьютеров и т. д. В простых моделях их можно рассчитать аналитическими методами, в более сложных случаях приходится прибегать к статистическому моделированию соответствующих случайных процессов.

Пример. Пусть компьютерная сеть имеет n одинаково доступных для заявок компьютеров (каналов обслуживания в терминах ТМО). Заявки поступают в случайные моменты времени. Если при поступлении очередной заявки все n каналов сети оказываются занятыми, то поступившая заявка получает отказ и теряется. По аналогии с телефонной сетью: когда абонент занят, мы получаем отказ и слышим частые гудки. В противном случае немедленно начинается разговор по одному из свободных каналов, длящийся, вообще говоря, неопределенное время, которое мы аппроксимируем подходящим распределением случайных величин. Одной из характеристик эффективности работы такой сети является доля заявок, получающих отказ, то есть предел p при $T\to\infty$ (если он существует) отношения n_T/N_T числа n_T заявок, потерянных в течение времени T, к общему числу N_T заявок, поступивших за это время. Этот предел называют вероятностью отказа в обслуживании.

Другим показателем качества служит относительное время занятости, т. е. предел p^* при $T\to\infty$ (если он существует) отношения t_T/T, где t_T – суммарное время, в течение которого за период T все n каналов сети одновременно заняты. Этот предел называют вероятностью занятости. Обозначим $X(t)$ – число каналов, занятых в момент t. Можно показать, что если: моменты поступления заявок образуют пуассоновский поток однородных событий, длительности обслуживания независимы (между собой и от моментов поступления заявок) и одинаково распределены, то случайный процесс $X(t)$, $t \geq 0$, обладает распределением Эрланга с плотностью

$$p(x) = \frac{(n\mu)^n}{\Gamma(n)} x^{n-1} e^{-n\mu x}, \quad x > 0,$$

где целое $n \geq 1$ и действительное $\mu > 0$ – параметры. При $n = 1$ наблюдается совпадение с показательным распределением с параметром μ.

3.2.2. Формализация задачи. В общепринятых терминах задача формулируется следующим образом. Пусть мы имеем систему массового обслуживания (СМО), насчитывающую в общем случае n одинаковых каналов (обслуживающих аппаратов). В данном контексте каналом является компьютер, исполняющий обслуживающую операцию обработки заявки. Входящий поток – простейший с интенсивностью поступления заявок в систему, равной λ. Время обслуживания – экспоненциальное с показателем μ. Любой из n каналов (любой компьютер сети) может обслужить любую заявку. Каждая заявка, поступая в систему, принимается к обслуживанию одним из свободных каналов. Если все каналы заняты, то заявка ожидает обслуживания в общем буферном накопителе (БН) объемом r мест для ожидания. Заявка, поступившая в систему и заставшая занятыми все n каналов и r мест для ожидания, получает отказ в обслуживании и теряется.

Требуется получить расчетные формулы, устанавливающие зависимость количества r мест для ожидания в функции известных λ, μ, n и наперед заданной допустимой вероятности π потери заявки.

Рассматриваемый однородный поток заявок обладает тремя свойствами: ординарность, стационарность, отсутствие последействия. Исследование процесса поступления и обработки заявок проведем на модели СМО без учета корреляции между заявками. Такие модели анализируются в большинстве руководств по исследованию операций, теории массового обслуживания [6] и теории вероятностей. На рис. 3.1 представлен граф переходов и сообщающихся состояний системы.

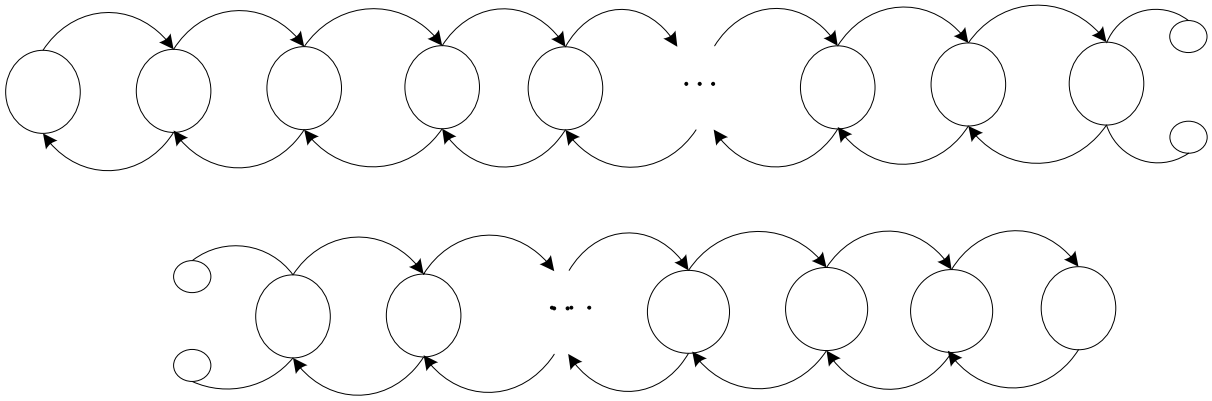


Рис. 3.1. Граф переходов и состояний для системы с однородным потоком записей

Левое граничное состояние P_0 соответствует отсутствию заявок на вычислительные работы. С интенсивностью λ осуществляются правонаправленные переходы $P_0 \rightarrow P_1$ – в системе одна принятая к обслуживанию заявка, $P_1 \rightarrow P_2$ – в системе обслуживаются две заявки и так далее. В силу ординарности потока на графе отсутствуют взаимные переходы, минующие соседние состояния. После прихода n-й заявки дальнейшие правонаправленные пере-

ходы отображают процесс образования очереди заявок, ожидающих обслуживания. Правое граничное состояние соответствует случаю, когда заняты все n каналов и r мест для ожидания. Левонаправленные переходы осуществляются в результате окончания обслуживания очередных заявок.

Вероятность (любого за исключением граничных) i-го состояния системы, $i = 1,\ldots, n+r-1$, в силу допущений об ординарности и отсутствия последействия, описывается известным [6] уравнением (3.1):

$$P_i(t+\Delta t) = P_i(t)\left[(1-\lambda\cdot\Delta t)(1-\mu\cdot\Delta t)\right] + P_{i-1}(t)\cdot\lambda\cdot\Delta t + P_{i+1}(t)\cdot\mu\cdot\Delta t \qquad (3.1)$$

где: t – произвольный момент времени на числовой оси;

Δt – рассматриваемый промежуток времени;

λ – интенсивность входного потока;

μ – пропускная способность каждой из n ЭВМ.

Запишем аналогичное выражение для левого граничного состояния P_0:

$$P_0(t+\Delta t) = P_0(t)(1-\lambda\cdot\Delta t) + P_1(t)\cdot\mu\cdot\Delta t. \qquad (3.2)$$

Преобразуем выражение (3.1):

$$P_i(t+\Delta t) = P_i(t)\left(1-\mu\cdot\Delta t-\lambda\cdot\Delta t+\lambda\cdot\mu\cdot\Delta t^2\right) + P_{i-1}(t)\cdot\lambda\cdot\Delta t + P_{i+1}(t)\cdot\mu\cdot\Delta t.$$

Сомножитель Δt^2 – есть величина второго порядка малости, поэтому пренебрегаем им и получаем выражение (3.3):

$$P_i(t+\Delta t) = P_i(t) - P_i(t)\cdot\mu\cdot\Delta t - P_i(t)\cdot\lambda\cdot\Delta t + P_{i-1}(t)\cdot\lambda\cdot\Delta t + P_{i+1}(t)\cdot\mu\cdot\Delta t. \qquad (3.3)$$

Разделим на Δt уравнения (3.1) и (3.2):

$$\left[P_i(t+\Delta t) - P_i(t)\right]/\Delta t = -(\lambda+\mu)\cdot P_i(t) + \lambda\cdot P_{i-1}(t) + \mu\cdot P_{i+1}(t), \qquad (3.4)$$

$$\left[P_0(t+\Delta t) - P_0(t)\right]/\Delta t = -\lambda\cdot P_0(t) + \mu\cdot P_1(t). \qquad (3.5)$$

Пусть $\Delta t \to 0$, тогда, при переходе к пределу, получим в левой части (3.4 и 3.5), в силу стационарности потока, производные постоянных по времени (стационарных) величин, откуда:

$$(\lambda+\mu)\cdot P_i = \lambda\cdot P_{i-1} + \mu\cdot P_{i+1}, \qquad (3.6)$$

$$\lambda\cdot P_0 = \mu\cdot P_1. \qquad (3.7)$$

Разделим обе части уравнений (3.6) и (3.7) на μ:

$$(\lambda/\mu + \mu/\mu)\cdot P_i = \lambda/\mu\cdot P_{i-1} + \mu/\mu\cdot P_{i+1}, \qquad (3.8)$$

$$\lambda/\mu\cdot P_0 = \mu/\mu\cdot P_1. \qquad (3.9)$$

Обозначим загрузку системы через $\rho = \lambda/\mu$ и преобразуем выражения (3.8) и (3.9):

$$(\rho+1)\cdot P_i = \rho\cdot P_{i-1} + P_{i+1} \qquad (3.10)$$

$$\rho\cdot P_0 = P_1 \qquad (3.11)$$

Продолжим преобразования. Если в (3.10) положить $i = 1$, то:

$$\rho\cdot P_1 + P_1 = \rho\cdot P_0 + P_2$$

Подставим P_1, используя уравнение (3.11):

$$\rho^2\cdot P_0 + \rho\cdot P_0 = \rho\cdot P_0 + P_2 \qquad \to \qquad P_2 = \rho^2\cdot P_0$$

Запишем полученный результат в общем виде:

$$P_i = \rho^i\cdot P_0, \qquad (3.12)$$

где i – порядковый номер состояния системы.

Выражение (3.12) получено для СМО с одним каналом, т. е. при $n = 1$.

Теперь рассмотрим вариант с системой в n каналов (компьютеров).

$$P_1 = n \cdot \rho \cdot P_0. \quad (3.13)$$

$$P_2 = n/2 \cdot \rho \cdot P_1. \quad (3.14)$$

Подставим (3.13) в (3.14):

$$P_2 = n^2/2 \cdot \rho^2 \cdot P_0. \quad (3.15)$$

Опираясь на (3.13) и (3.15), запишем по индукции для произвольного s-го состояния:

$$P_s = n^{s-1}/(s-1)! \cdot \rho^{s-1} \cdot P_0,$$

где s – индекс текущего состояния, т.е. количество обслуживаемых в произвольный момент времени заявок; $s = 1,...,n$; если в СМО более n заявок, то:

$$P_k = n^{n-1}/(n-1)! \cdot \rho^k \cdot P_0,$$

где: k – текущее число заявок во всей системе в ситуации, когда заняты все компьютеры и образовалась очередь ожидающих заявок; $k = n+1,...,n+r$.

Учитывая, что сумма вероятностей всех изображенных на графе (рис. 3.1) состояний системы равна единице, фиксируем:

$$P_0 + P_1 + ... + P_n + P_{n+1} + ... + P_{n+k} + ... + P_{n+r} = \sum_{s=0}^{n} P_s + \sum_{k=n+r}^{n+r} P_k = 1.$$

В левой части сосредоточены вероятности всех состояний системы, которые можно выразить через искомую вероятность P_0 простоя системы. Это известный математический ряд – геометрическая прогрессия, ряд сходящийся, если знаменатель меньше единицы, сумма его известна. Выражая P_k через P_0, выносим P_0 за скобки, снижаем показатель степени на единицу:

$$P_0\left[1 + n\rho + n^2\rho^2/2 + ... + n^{n-1}\rho^{n-1}/(n-1)! + n^{n-1}/(n-1)! \sum_{k=n+1}^{n+r} \rho^k\right] = 1.$$

После элементарных преобразований:

$$P_0\left[\frac{n^{n-1}}{(n-1)!}\left(1 + \rho + \rho^2 + ... + \rho^{n-1} + \sum_{k=n+1}^{r} \rho^k\right) - \sum_{k=0}^{n}\left(\frac{n^{n-1} - k^{n-1}}{n-1}\rho^k\right)\right] = 1, \text{ откуда:}$$

$$P_0 = \frac{(1-\rho)}{\sum_{k=0}^{n} \frac{n^{k-1}(n-k)}{k!}\rho^k - \frac{n^{n-1}}{(n-1)!}\rho^{n+r+1}}.$$

Априорно допустимая вероятность π потери записи в файле есть вероятность P_{n+r} правого граничного состояния графа (рис. 3.1):

$$\pi = P_{n+r} = \frac{\frac{n^{n-1}}{(n-1)!}\rho^{n+r}(1-\rho)}{\sum_{k=0}^{n} \frac{n^{k-1}(n-k)}{k!}\rho^k - \frac{n^{n-1}}{(n-1)!}\rho^{n+r+1}}. \quad (3.16)$$

Вынесем искомое r в левую часть уравнения. Умножим π на знаменатель (3.16): $\pi \sum_{k=0}^{n} \frac{n^{k-1}(n-k)}{k!}\rho^k - \pi \frac{n^{n-1}}{(n-1)!}\rho^{n+r+1} = \frac{n^{n-1}}{(n-1)!}\rho^{n+r}(1-\rho).$

Продолжим преобразования:

$$\frac{n^{n-1}}{(n-1)!}\rho^{n+r}(\rho\cdot\pi+1-\rho)=\pi\sum_{k=0}^{n}\frac{n^{k-1}(n-k)}{k!}\rho^{k}.$$

$$\rho^{n+r}=\left(\pi\sum_{k=0}^{n}\frac{n^{k-1}(n-k)}{k!}\rho^{k}\right)/\left[\frac{n^{n-1}}{(n-1)!}(\rho\cdot\pi+1-\rho)\right].$$

$$(n+r)\cdot\ln\rho=\ln\left(\pi\sum_{k=0}^{n}\frac{n^{k-1}(n-k)}{k!}\rho^{k}\right)-\ln\left[\frac{n^{n-1}}{(n-1)!}(\rho\cdot\pi+1-\rho)\right]. \qquad (3.17)$$

Из выражения (3.17) получаем формулу для расчета r:

$$)r(=\frac{\left\{\ln\left(\pi\sum_{k=0}^{n}\frac{n^{k-1}(n-k)}{k!}\rho^{k}\right)-\ln\left[\frac{n^{n-1}}{(n-1)!}(\rho\cdot\pi+1-\rho)\right]\right\}}{\ln\rho}-n. \qquad (3.18)$$

где символ $)r($ обозначает выражение $r=\max\{0,]r[\}$, т.е. максимальное неотрицательное большее целое от вычисленного значения r. Требование неотрицательности выдвигается вследствие того, что при больших значениях допустимой вероятности π потери записи или при малых загрузках системы ρ соотношение параметров системы может оказаться таким, что заданный порог π удовлетворяется даже в абстрактном случае отрицательного количества r мест для ожидания, что физически бессмысленно.

Величина r в (3.18) означает объем БН, который с наперед заданной вероятностью π гарантирует обслуживание всех заявок. Записи о планах полетов, об отождествленных с ними измеренных параметрах движения ВС, размещаемые в системе, связаны отношениями предшествования и общими атрибутами. В процессе планирования и непосредственного УВД они корректируются – как со стороны расчетных программ, так и при вводах функций диспетчеров. Обращения к записям могут происходить одновременно, и в таких случаях конфликтующие запросы ставятся ОСРВ в очередь ожидания обслуживания. Простои приводят к потерям производительности, что должно учитываться при анализе модели как снижение пропускной способности канала, или как снижение величины μ параметра обслуживания.

3.3. Учет связности заявок

3.3.1. Постановка задачи. Организация совместной работы компьютеров приводит к потерям производительности на взаимодействие, которые можно оценить статистически, используя последовательности значений $\{Q_i\}$ корреляции выполняемых программных функций по общим данным и отношениям предшествования. Параметр Q_i введен для учета связей по управлению и данным и назван корреляцией между заявками. Понятие определяется как некоторая вероятность Q_i (i = 1,…,n) обслуживания очередной заявки i-м свободным компьютером при том условии, что любые (i – 1) других компьютеров системы заняты. Это – интегральная характеристика состояния, указывающая, с какой вероятностью смогут, начиная с текущего момента, быть занятыми i компьютеров системы, причем в силу стационарности входного потока этот момент инвариантен относительно сдвига по оси времени. Для на-

глядности используем следующую аналогию. КП обработки радиолокационной информации (РЛИ) АС УВД периодически, с темпом обзора антенны радиолокационной станции (РЛС), решает следующие задачи обработки РЛИ:

- сбор и обработка сообщений первичной и вторичной радиолокации;
- вычислительные процессы обнаружения, захвата, сопровождения ВС;
- организация фаз ассоциации, фильтрации, экстраполяции траекторий;
- первичная, вторичная и третичная обработка РЛИ;
- обработка данных радиопеленгаторов;
- радиолокационное сопровождение в сложной информационной обстановке (пропуски целей, ложные отметки, помехи, маневры объектов);
- анализ качества прокладки траекторий.

На каждом обзоре должен выполняться весь список задач, однако они связаны отношениями предшествования. Фаза экстраполяции не может начаться раньше фильтрации и ассоциации, которые, в свою очередь, должны дождаться результатов обработки координатных измерений, выполняемых после окончания сбора сообщений. Помимо функциональных связей, данные коррелированны между собой вследствие их общего использования взаимодействующими КП обработки планов полетов, докладов подсистемы автоматического зависимого наблюдения, функций ввода диспетчерского персонала. Другими словами, обязательные для исполнения процессы не могут начаться на свободном (незанятом) компьютере сети до тех пор, пока не создадутся необходимые условия. Количественно такие потери номинальной производительности сети учитываются показателем Q_i.

В данном разделе исследуются области изменения перечисленных параметров модели, в которых применение компьютерной сети становится предпочтительнее, чем одной вычислительной машины эквивалентной производительности. Сначала проследим общие закономерности изменения требований к объему памяти, отводимой для размещения заявок, без учета влияния корреляции, а затем распространим результат на более общий случай.

3.3.2. Модель без учета корреляции заявок. Для исследования тенденции изменения объема БН, необходимого для сопровождения r записей, в зависимости от количества n компьютеров сети, введем функцию β_n относительного выигрыша, доставляемого обслуживанием поступающих заявок на n-канальной системе в сравнении с одноканальной. На первом шаге рассмотрения ограничимся $n = 2$. При этом вероятность потери записи вследствие недостаточного объема r БН для одного канала ($n = 1$) составляет величину $\pi_1 = \dfrac{\rho^{r+1}(1-\rho)}{1-\rho^{r+2}}$, для двух каналов ($n = 2$) $\pi_2 = \dfrac{2\rho^{r+2}(1-\rho)}{1+\rho-2\rho^{r+3}}$. Тогда $\beta_2 = \dfrac{\pi_1 - \pi_2}{\pi_1}$, или $\beta_2 = \dfrac{1-\rho}{1+\rho-2\rho^{r+3}} = P_{02}$, где P_{02} есть вероятность простоя двухканальной системы; вообще, относительный выигрыш, как показывает анализ полученных формул, совпадает по значению с вероятностью простоя системы: $\beta_n = P_{0n}$.

В предельном случае, при $r \to \infty$ (неограниченный объем файла) функ-

ция относительно выигрыша не зависит от числа n каналов вычислительной системы: $\beta_{r\to\infty} = (1 - \rho) / (1 + \rho)$. Другими словами, для правильно рассчитанной по загрузке ($\lambda < \mu$, $\rho < 1$) системы безразлично число обслуживающих аппаратов, так как все записи находят место в файле. При $\rho \to 1$ выигрыш $\beta_{r\to\infty}$ отсутствует вообще. Со снижением загрузки наблюдается прирост значения β_n, достигающий максимума при $\rho \to 0$, однако он носит чисто символический характер, объясняемый упрощениями анализируемой модели. Фактически утверждается, что с ростом n больше заявок одновременно обслуживаются в системе, т. е. больше процессов разнесены по локальным компьютерам сети и меньший (относительно одноканальной системы) объем БН с общим доступом занят очередью на обслуживание.

Графики зависимостей $\beta_2 = f(\rho)$ для некоторых значений r представлены на рис. 3.2. Недостатки модели проявляются здесь наиболее ощутимо. Для распределенной вычислительной сети, не имеющей БН с общим доступом, в противоположном предельном случае, при $r = 0$, даже в условиях полной загрузки выигрыш составляет двадцать процентов ($\beta_2 = 0{,}2$). Дальнейшее наращивание количества n каналов еще выше поднимает минимум β_n для вычислительной сети без БН общего доступа. Однако использование даже небольшого общего БН ($r = 5$) резко снижает символический эффект вычислительной сети относительно одного канала, а при $r \geq 10$ преимущества параллельной обработки по рассматриваемому критерию практически исчерпываются. На графике видно, что в области нормальной загрузки вычислительной системы значение показателя β стремится к нулю.

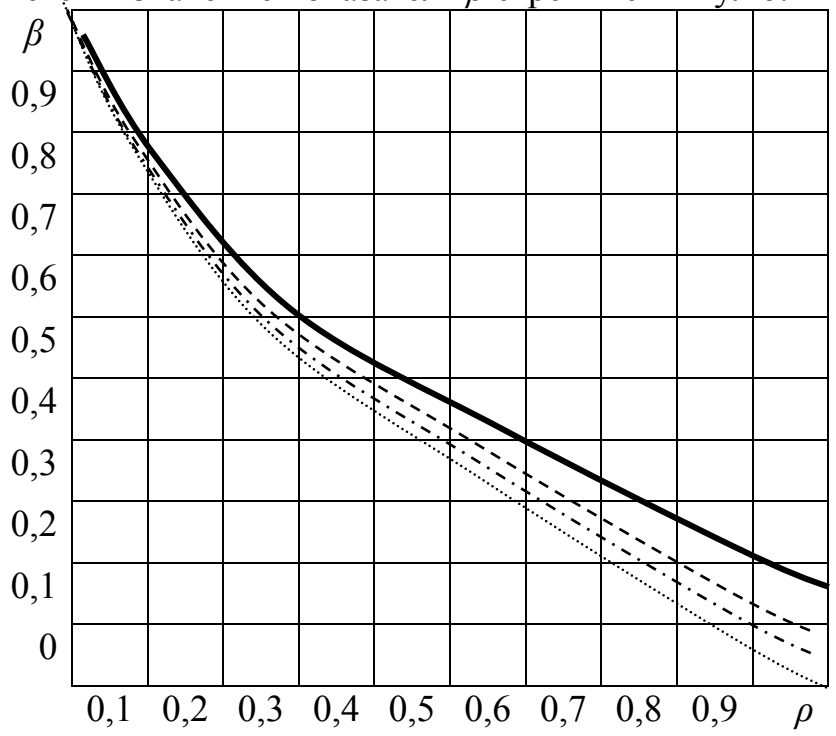


Рис. 3.2. График зависимости относительного выигрыша β от загрузки ρ системы.

Условные обозначения:

Анализ семейства кривых $\beta_2 = f(\rho)$ позволяет обойти недостатки упрощенной модели. Увеличение количества n каналов системы интерпретирует-

$$a_{n+r} = \frac{\pi_r^{(n)}}{\pi_{n+r-1}^{(1)}} = \frac{\dfrac{n^{n-1}}{(n-1)!}\rho^{n+r}\left(1-\rho^{n+r+1}\right)}{\displaystyle\sum_{k=0}^{n}\frac{n^{k-1}(n-k)}{k!}\rho^k - \frac{n^{n-1}}{(n-1)!}\rho^{n+r+1}}.$$

ся в ней как тривиальное приращение числа мест для ожидания. Считается, что заявки распространяются по локальным файлам распределенной вычислительной сети, и доступ к информации соседа возможен лишь как запрос данных, реализуемый через отдельную заявку. Для формализации модели БН с общим доступом достаточно стабилизировать в ней общее количество $n + r$ мест в системе. Тогда увеличение количества n каналов будет компенсироваться равным ему сокращением объема r БН. В результате отношение α_{n+r} вероятности $\pi_r^{(n)}$ потери заявки в n-канальной системе с объемом r БН общего доступа к вероятности $\pi_{r+n-1}^{(1)}$ потери заявки в эквивалентной по пропускной способности и количеству размещаемых записей одноканальной системе составит (α_{n+r} есть аналог β_n при фиксированном значении суммы $n + r$)

Нетрудно видеть, что во всем диапазоне изменения загрузки ρ системы $0 \le \rho \le 1$ при любых $n > 1$ отношение $\alpha_{n+r} > 1$ и возрастает с увеличением n ($n + r$ = Const). Это значит, что потери записей в системе, содержащей n каналов и общий БН объемом на r заявок, выше, чем в одноканальной системе с БН длиной $r + n + 1$. В поставленной задаче величина π задается априорно в качестве ограничительного параметра. Следовательно, для удовлетворения условий допустимого уровня потери записей, объем БН общего доступа в многоканальной системе необходимо резервировать более вместительным, чем в одноканальной системе. Широко известна и обратная задача: при заданных значениях интенсивности входного потока λ, параметра обслуживания μ, количества приборов n и мест ожидания r определить вероятность π потери заявки, также решенная полученными формулами.

3.3.3. Модель с учетом корреляции. Для сопоставления многоканальных систем при различных Q_i необходимо нормировать их суммарным значением: $\sum_{i=1}^{n} Q_i$. На первом шаге рассмотрения ограничимся сравнением СМО с одним и двумя приборами (компьютерами), в дальнейшем это ограничение будет снято. Напомним, что нерегулярность входного потока проявляется в наличии периодов спада и возрастания текущих значений частоты поступления заявок (рис. 3.3). График, изображенный на рис. 3.4, дает представление о соотношении величин вероятностей π потери заявки для предельного случая $r = 0$ отсутствия файла общего доступа, т.е. при распределенной вычислительной сети. В гипотетической ситуации, соответствующей области малых загрузок ($\rho <= 0.3$), в которой вычислительные системы обычно не эксплуатируются, использование двух каналов дает выигрыш даже при полном запрете приема на обслуживание заявки вторым прибором, если работает первый ($Q_2 = 0$). Соот-

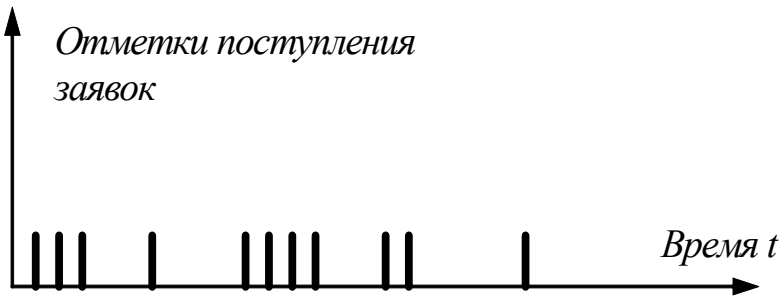


Рис. 3.3. Ситуации нерегулярности потока

ветственно, суммарная загрузка СМО достигает лишь половины показателя для случая отсутствия корреляции. Сказывается малая занятость системы в среднем. Вследствие нерегулярности входного потока типична следующая ситуация. «Сгущения» моментов поступлений расположены на оси времени таким образом, что использование второго канала в качестве БН полезнее для системы, пусть даже и вдвое менее производительной, чем один быстродействующий канал, не успевающий «захватить» (вследствие отсутствия буфера) очередную заявку и простаивающий затем в периоды относительного «разрежения» входного потока.

С увеличением загрузки при $Q_2 = 0$ потери в двухканальной системе резко возрастают и при $\rho \to 1$ достигают максимума, значительно превосходя соответствующий показатель для одноканальной системы. Вероятность потери записи для случая $Q_2 = 1$ всегда ниже, чем при использовании одного прибора. Напомним, что функция относительного выигрыша β_2 (рис. 3.2) даже при $\rho \to 1$ достигает величины 0,2. Эта тенденция отчетливо прослеживается на графике рис. 3.5. Кривая, соединяющая наивысшие значения π_c для предельных случаев $Q_2 = 0$ и $Q_2 = 1$, есть годограф функции $\pi_c = f(Q_2)$ при стремлении значения загрузки системы ρ к единице $\rho \to 1$. При этом суммарная загрузка двухканальной СМО поддерживается на уровне $\rho(1 + Q_2)$ и не достигает удвоенной величины при $Q_2 < 1$.

Совместное решение уравнений кривых π и π_c позволяет получить набор критических значений Q_2, при которых использование двух каналов ста-

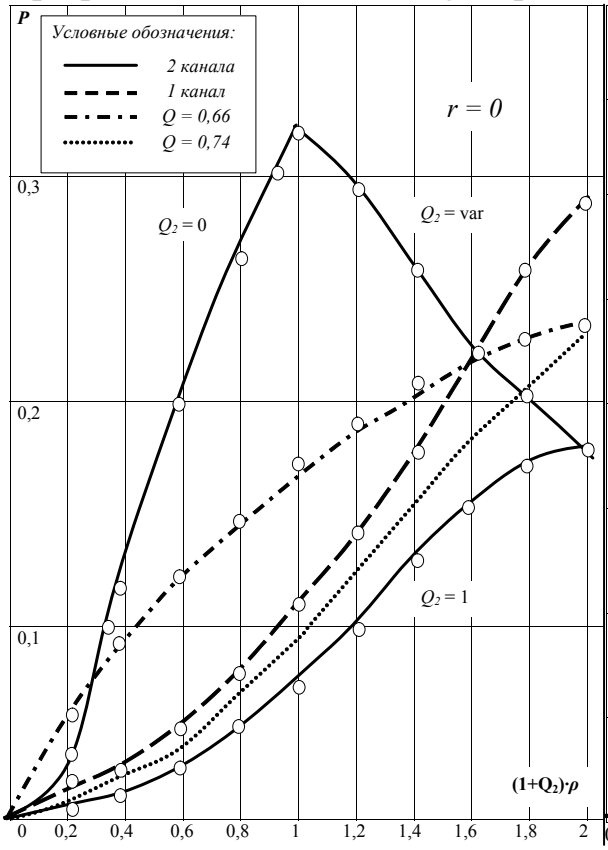


Рис. 3.4. Сравнительные оценки для вероятностей потерь заявок в СМО с одним и с двумя каналами при отсутствии БН (r = 0)

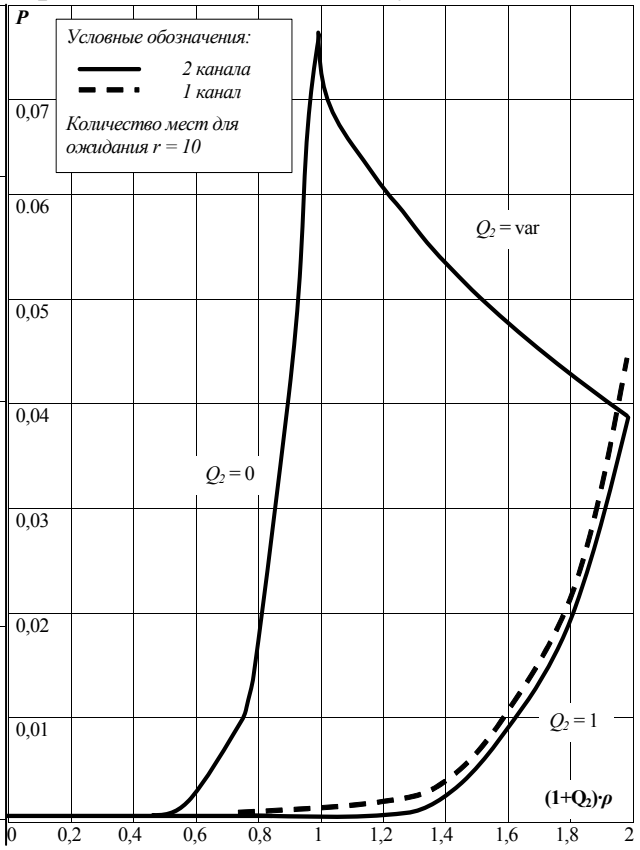


Рис. 3.5. Сравнительные оценки для вероятностей потерь заявок в СМО с одним и с двумя каналами при объеме БН r = 10

новится предпочтительнее, чем одного. При отсутствии в системе БН общего доступа (объем $r = 0$) и полной загрузке ($\rho \to 1$) таким значением является $Q_{кр} = 0{,}74$. С уменьшением загруженности системы связанность заявок, при которой два прибора все еще предпочтительнее, может достигать более высоких значений, например, при $\rho = 0{,}9$ показатель $Q_2 = 0{,}66$. Преимущества «многоканальности» весьма условны, так как введение даже небольшого по объему БН, без которого система становится менее эффективной, сводит их к минимуму, что наглядно демонстрирует график, представленный на рис. 3.5.

Для получения сравнительных оценок эффективности в системах различной конфигурации с учетом корреляции между записями, рассмотрим отношение α_c вероятности π_c потери заявки в многоканальной системе к вероятности π потери заявки в системе с одним каналом. Оно вводится аналогично показателю α_{n+r} относительного выигрыша, исследованному выше в модели обслуживания без учета корреляции заявок, и наследует характер соотношения параметров. Отметим, что при анализе связности заявок по данным или по управлению, аналогом количества n каналов при значащих Q_i служит $\sum_{i=1}^{n} Q_i = 1 + Q_2 + \ldots + Q_n$, аналогом k – $\sum_{i=1}^{k} Q_i$, аналогом значения $k!$ – $\prod_{l=1}^{k} \sum_{i=1}^{l} Q_i$, а загрузка системы при занятости каналов $\rho = \dfrac{\lambda}{\mu \sum_{i=1}^{k} Q_i}$. Искомое соотношение:

$$\alpha_c = \frac{\pi_c}{\pi} = \frac{\dfrac{\left(\sum_{i=1}^{n} Q_i\right)^{n-1}}{\prod_{l=1}^{n-1} \sum_{i=1}^{l} Q_i} \cdot \rho^{n-1} \cdot \left(1 - \rho^{r+n+1}\right)}{\sum_{k=0}^{n} \dfrac{\left(\sum_{i=1}^{n} Q_i\right)^{k-1} \left[\left(\sum_{i=1}^{n} Q_i\right) - \sum_{i=1}^{k} Q_i\right]}{\prod_{l=1}^{k} \sum_{i=1}^{l} Q_i} \cdot \rho^{k} - \dfrac{\left(\sum_{i=1}^{n} Q_i\right)^{n-1}}{\prod_{l=1}^{n-1} \sum_{i=1}^{l} Q_i} \cdot \rho^{n-1} \cdot \left(1 - \rho^{r+n+1}\right)}.$$

В зависимости от значений вероятностей $\{Q_i\}$, количества n каналов СМО, загрузки ρ и объема r БН отношение α_c пробегает всю правую полуось, начиная от единицы.

Организация совместной работы компьютеров в АС УВД приводит к потерям производительности на взаимодействие, которые можно оценить статистически, используя последовательности значений $\{Q_i\}$ корреляции выполняемых программных функций по общим данным и отношениям предшествования. Финальные вероятности переходов процесса по разветвляющимся дугам исходного графа могут с достаточной достоверностью определяться априорно, исходя из заданных характеристик проекта. В такой постановке задача расчета последовательности вероятностей Q_i становится методически столь же реальной, как вычисление значений λ и μ. Эти величины характеризуются известным распределением моментов наступления событий, интенсивностью их потоков, подтвержденными опытом эксплуатации. Исследо-

ванные модели поступления и обслуживания заявок в вычислительных системах позволяют наметить оценки работы СМО, основанные на вероятностях потери заявок на обслуживание. Полученные результаты помогают понять физическую природу образования очередей. Однако в силу ряда упрощений, без которых трудно было бы дать направление анализа, их можно рекомендовать лишь для ориентировочных расчетов объемов БН. Для более полного учета факторов, влияющих на вычислительный процесс, необходимо рассмотреть модели с неоднородным входным потоком.

На основании аналогичных изложенным при выводе выражения (3.18) рассуждений нетрудно получить формулу для расчета объема r БН с учетом связанности заявок на вычислительные работы:

$$)r(=\frac{\ln\left\{\pi\sum_{k=0}^{n}\frac{\left(\sum_{i=1}^{n}Q_i\right)^{k-1}\left[\left(\sum_{i=1}^{n}Q_i\right)-\sum_{i=1}^{k}Q_i\right]}{\prod_{l=1}^{k}\sum_{i=1}^{l}Q_i}\cdot\rho^k\right\}-\ln\left[\frac{\left(\sum_{i=1}^{n}Q_i\right)^{n-1}}{\prod_{l=1}^{n-1}\sum_{i=1}^{l}Q_i}\cdot(1-\rho+\pi\cdot\rho)\right]}{\ln\rho}-\sum_{i=1}^{n}Q_i\,,$$

где символ) (, охватывающий в левой части обозначение объема r необходимой памяти, означает ближайшее большее неотрицательное целое, а все обозначения в правой части соответствуют параметрам системы:

π – допустимая вероятность переполнения БН, определяемая замыслом системы (установленная техническим заданием);

$\rho = \lambda/\mu$ – загрузка системы, равная отношению интенсивности λ потока заявок (сообщений) о ВС и об изменении состояния элементов структуры воздушного пространства (ВП) к параметру μ их обслуживания;

n – количество компьютеров вычислительной сети центра УВД.

Остается открытым вопрос расчета значений $\{Q_i\}$ корреляции выполняемых программных функций по общим данным и отношениям предшествования, решению которого посвящается следующий параграф.

3.4. Оценка связности задач в центре управления

3.4.1. Постановка задачи. Любое программное изделие еще на этапе проектирования принято представлять графом, вершины которого соответствуют выполняемым функциям, а дуги – допустимым переходам между ними (рис. 2.1 параграфа 2.2.2). В целях однозначного понимания замысла разработчика, изобразительные элементы алгоритмических схем стандартизованы, установлены правила начертания блоков операторов, разветвлений, циклов и других элементов. Прослеживается аналогия со структурными схемами технических систем разного уровня сложности и обобщения, наглядно демонстрирующими логику их работы. В данном изложении важно то обстоятельство, что финальные вероятности переходов процесса по разветвляющимся дугам исходного графа могут с достаточной достоверностью определяться априорно, исходя из заданных характеристик проекта. В такой постановке за-

дача расчета последовательности вероятностей Q_i становится методически столь же реальной, как вычисление значений λ и μ. Эти величины характеризуются известным распределением моментов наступления событий, интенсивностью их потоков, подтвержденными опытом эксплуатации. Затруднения вызывает лишь громоздкость вычисления вероятностей прохождения процесса по десяткам тысяч возможных путей на графе сложной программы.

Важная особенность задачи состоит в том, что для каждой отдельной вершины или дуги графа можно легко сформулировать условие их принадлежности искомой ветви программного процесса и указать вероятность выбора дальнейшего пути в узле разветвления. Выполнение такого условия равносильно соблюдению исходных ограничений. В рассматриваемом случае это выбор множества программных функций (вершин графа), связанных между собой отношениями предшествования или общими данными. Сложнее подобрать подходящее отображение сформированного множества в компьютерной памяти, удобное для дальнейшей обработки. Нужно фиксировать найденные пути, рассчитать их длины, вероятности прохождения по ним, другие статистические характеристики. Традиционные матричные способы представления графов громоздки и неудобны для такого рода вычислительных задач. Известны более компактные инструменты работы с графами, основанные на аппарате булевой алгебры. Вместо двумерной матрицы размерности m, где m – число вершин графа, используются m функций алгебры логики (ФАЛ) вида $f(y_1,...,y_m)$. Функция принимает единичное значение – отображает возможный путь на графе – если определена на двоичном наборе $\{y_i\}$, $i=\overline{1,m}$, в котором единичные переменные соответствуют вершинам, включенным в этот путь. Тогда все операции по нахождению допустимых путей сводятся к построению и минимизации ФАЛ, дизъюнктивная нормальная форма (ДНФ) которой содержит все возможные решения. В результате выбор оптимального решения упрощается, нужно лишь указать способ перехода от ДНФ к задаче линейного программирования с булевыми переменными и получить оценки ее сложности.

Сопоставим вершинам $x_1,...,x_m$ графа G (X, Γ) двоичные переменные $y_1,...,y_m$. Пусть x_i включается в допустимое множество $R \subseteq X$, если и только если $y_i = 1$ в некотором наборе значений переменных $y_1,...,y_m$. Тогда такой набор однозначно определяет искомое подмножество вершин графа. Отметим, что если в нем фиксированы не все m значений переменных, то он определяет целостную совокупность подмножеств вершин с указанными свойствами, т.е. учитываются все разветвления графа, отвечающие условиям корреляции.

Удобство такого подхода состоит в том, что становится возможным вместо связей между вершинами графа – программными функциями – налагающих ограничения на искомые множества связанных вершин или дуг, рассматривать более простые зависимости между значениями двоичных переменных y_i. Следовательно, упомянутое неформальное условие принадлежности вершины определенному пути, сформулированное в терминах теории графов, можно записать строго в виде логической функции двоичных пере-

менных: $f(y_1,\ldots,y_m) = 1$ в том и только в том случае, когда для каждой вершины x_i выполнено заданное условие ее принадлежности искомому пути на графе программы.

Переход от многозначности графовых моделей к двоичным функциям сводит задачу расчета параметра Q_i к отысканию всех наборов значений двоичных переменных $y_1,\ldots,y_m$, на которых $f(y_1,\ldots,y_m)$ принимает единичное значение. Описанием условий исходной задачи (нахождения вершин, обладающих свойством принадлежности искомой ветви программы) становится конъюнкция функций f_i по всем вершинам графа программы:

$$F(y_1,\ldots y_m) = \bigwedge_{i=1}^{m} f_i(y_1,\ldots y_m),$$

если и только если для множества $\{x_i \,/\, y_i^{\circ} = 1\}$ выполнено условие $F(y_1^{\circ},\ldots,y_m^{\circ}) = 1$. Таким образом, задача размерности m поиска связанных частей сложного программного комплекса сводится к простому (линейному) перечислению всех наборов значений двоичных переменных $y_1^{\circ},\ldots,y_m^{\circ}$, на которых конъюнкция $F(y_1,\ldots,y_m)$ обращается в единицу.

Для наглядности представим себе произвольный граф программы, все ребра которого взвешены вероятностями переходов по ним от вершины к вершине. На данном шаге анализа мы пытаемся выделить существующие пути, не учитывая вероятностей их прохождения, и заменяем все ненулевые значения единицами. Как только искомые пути найдены, значения вероятностей восстанавливаются, и вычисляется значение корреляции Q_i для каждой задачи, входящей в граф полной программы.

Пусть F представлена в минимальной ДНФ. Тогда каждая ее импликанта f описывает семейство множеств (решений), и вся совокупность искомых решений оказывается объединением таких семейств. Каждое множество семейства наглядно прослеживается на графе последовательностью вершин, соединенных ненулевыми ребрами, составляющих возможный путь исполнения программы. В семейство множеств-решений, описываемое импликантой $K(y_1,\ldots,y_m) = \bigwedge_{i\in I_K} y_i^{\circ}$, входит всякое множество $R \subseteq X$, определяемое таким набором $y_1^{\circ},\ldots,y_m^{\circ}$, что $K(y_1^{\circ},\ldots,y_m^{\circ}) = 1$, т.е. $\bigwedge_{i\in I_K}(y_i^{\circ} \vee \overline{y_i}^{\circ}) = 1$. Иначе говоря, каждое множество таких решений обладает двумя свойствами:

- если импликанта содержит y_i, то вершина x_i входит в R;
- если импликанта содержит $\overline{y_i}$, то вершина x_i не входит в R.

Отметим, что по условиям задачи достаточно рассматривать только одно множества из каждого семейства, а именно – предельное множество, содержащее максимум элементов по каждому допустимому пути на графе сложной программы.

3.4.2. Анализ достижимости вершин графа сложной программы. Для образования последовательности значений $\{Q_i\}$ достаточно определить множество вершин графа, достижимых из любой заданной вершины, либо решить обратную задачу: найти множество вершин, из которых данная вер-

шина достижима. Схожие операции связаны в теории множеств с ее основным топологическим понятием – замыканием, методы отыскания которого требуют большого объема вычислений. Частным случаем является проверка наличия контура, проходящего через две данные вершины. Решение состоит в построении для графа $G(X, \Gamma)$ соответствующего транзитивного замыкания (пути из i-й вершины в j-ю, если он имеется). Обычно для этого используют матрицу смежности графа $G(X, \Gamma)$, формируемую с помощью громоздкой процедуры возведения в $(m-1)$-ю степень матрицы смежности исходного графа. Применение ФАЛ существенно упрощает анализ достижимости.

Сопоставим каждой вершине графа $G(X, \Gamma)$ двоичную переменную и определим логическую функцию F_Γ следующим образом:

$$F_\Gamma(y_1,\ldots,y_m)=\bigwedge_{i=1}^{m}\bigwedge_{x_j\in\Gamma_{x_i}}\left(\left(\overline{y}_i\vee y_j\right)\right)=\bigwedge_{i=1}^{m}\left(\overline{y}_i\vee\bigwedge_{x_j\in\Gamma_{x_i}}y_j\right). \qquad (3.19)$$

Назовем y_i индексом вершины x_i. Пусть индекс вершины k есть единица, $y_k = 1$. Рассмотрим условия, при которых $F_\Gamma(y_1,\ldots,y_{k-1},1,y_{k+1},\ldots y_m) = 1$. Согласно (3.19), индексы всех вершин из подмножества $X_1 = \{x_i/x_i \in \Gamma x_k\}$ должны быть равны единице. В свою очередь, индексы всех вершин из подмножества $X_2 = \{x_i/x_i \in \Gamma X_1\}$ также должны быть равны 1. Проведем эти рассуждения для последующих подмножеств, порождаемых таким же образом, пока не окажется $X_r \supseteq X_{r+1}$. В результате получим, что индексы всех вершин, достижимых из x_k, и только они, должны обязательно иметь значение 1, чтобы функция F_Γ обращалась в единицу при $y_k = 1$.

Если F_Γ приведена к ДНФ, то весьма удобно отыскивать такие переменные, тем самым находя достижимые вершины. Преобразуем ДНФ F_Γ к виду $D_0 \vee \overline{y}_i D_1$. Так как при $y_i = 1$ $\overline{y}_i D_1 = 0$, то для $F_\Gamma(y_1,\ldots,y_{i-1},1,y_{i+1},\ldots y_m) = 1$ необходимо $D_0 = 1$. Если во всех импликантах D_0 какая-то переменная y_j содержится без отрицания, то $D_0 = 1$ только при $y_j = 1$. Нетрудно видеть, что любая другая переменная может быть равна нулю при $D_0 = 1$. Таким образом, поиск сводится к выделению переменных, входящих без отрицания во все импликанты, не содержащие $\overline{y}_i$ (но содержащие, быть может, y_1).

По аналогии можно показать, что при $y_j = 0$ рассматриваемая функция $F_\Gamma(y_1,\ldots,y_{i-1}, 0, y_{i+1},\ldots y_m) = 1$ только в том случае, когда придано нулевое значение всем таким переменным y_j, что из вершины x_j достижима вершина x_i. В этом случае поиск состоит в нахождении по ДНФ F_Γ тех переменных, которые входят с отрицанием во все импликанты, не содержащие y_j (без отрицания), но содержащие, быть может, $\overline{y}_i$. Наличие контура, проходящего через x_i и x_j, однозначно определяется взаимной достижимостью этих вершин.

Для наглядности рассмотрим частичный граф КП обработки плановой информации в АС УВД (рис. 3.6). Вершинам и дугам соответствуют основные программные функции и связи между ними. Дуги взвешены вероятностями условных переходов между функциями при различных сочетаниях исходных данных и стадиях прохождения задачи. Замысел формирования последовательности значений $\{Q_i\}$ корреляции программных функций состоит в том, чтобы взвесить ребра графа вероятностями их прохождения в процессе реальной работы, основываясь на опыте эксплуатации прототипов, накопленном экспертами, и после статистической обработки результатов опроса использовать их для расчета характеристик обслуживания заявок.

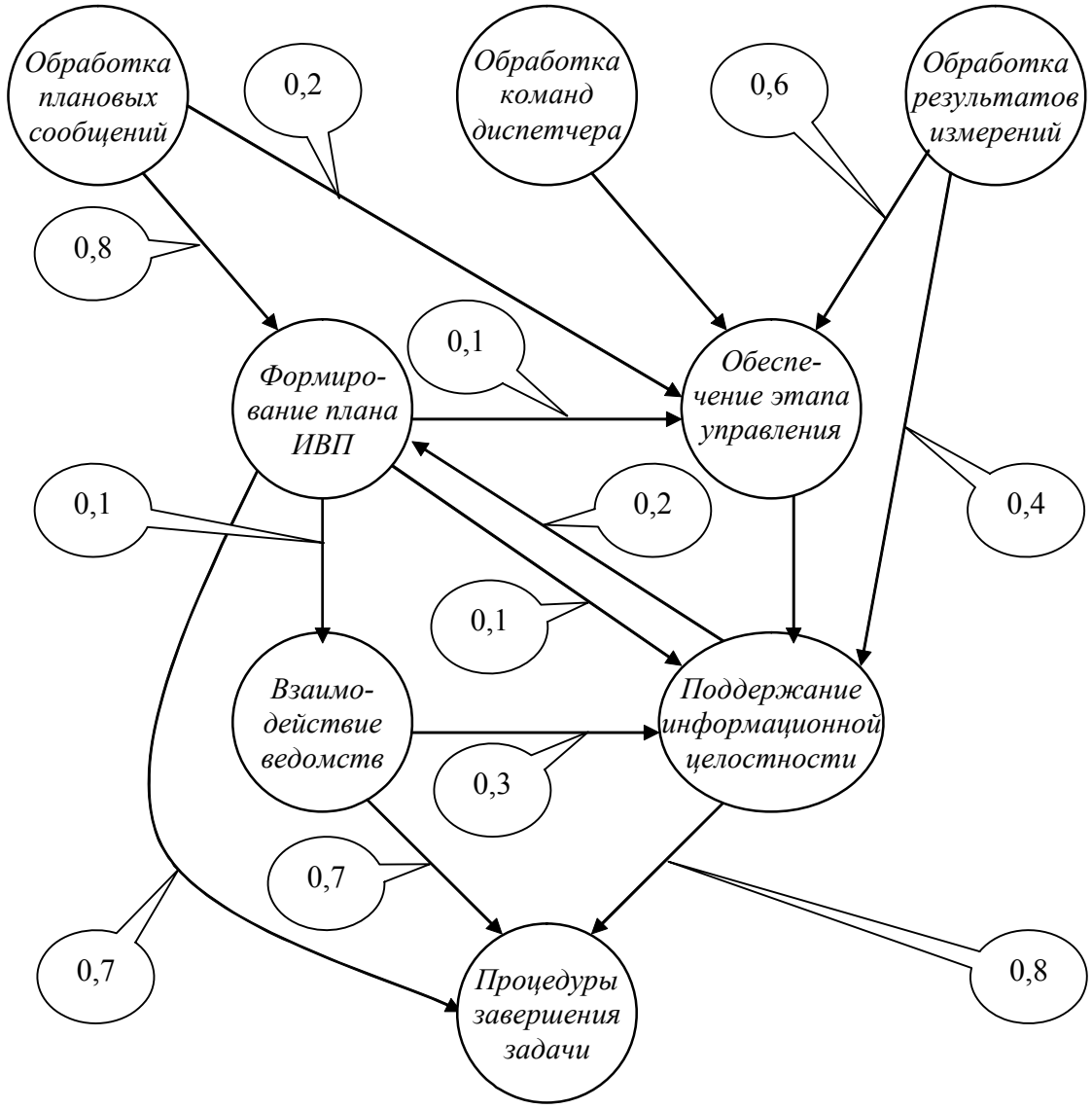


Рис. 3.6. Граф связей основных функций КП планирования полетов

На упрощенной схеме, представленной на рис. 3.6, достижимость каждой вершины легко проследить визуально. Однако при детализации каждая вершина отображается разветвленным графом высокой размерности с циклами, и возможности анализа общей картины человеком исчерпываются взвешиванием дуг вероятностями условных переходов. Значения вероятно-

стей предоставлены экспертами в области проектирования ПО. В табл. 3.1 приведен фрагмент результатов анкетирования группы квалифицированных специалистов, в различные годы принимавших участие в разработке программных комплексов АС УВД. Бросается в глаза малый разброс указанных ими величин вероятностей условных переходов между ветвями КП. Аналогичная ситуация отмечена и авторами, использовавшими аппарат булевой алгебры для оценки полноты и качества отладки программного обеспечения управляющих систем реального времени. Уверенность разработчиков и сходство предложенных ими количественных мер частотных характеристик подтверждаются результатами расчетов значений Q_i с использованием данных, полученных в процессе обработки экспертных оценок. Таблица 3.1

Эксперты	Программные комплексы по схеме рис. 3.6 и оценки вероятностей переходов			
	План ИВП	Этап ОВД	Взаимодействие	Целостность
1. Абезгауз Я.И.	0,7	0,7	0,15	0,75
2. Бененсон З.М.	0,8	0,75	0,2	0,8
3. Бубнов Г.П.	0,75	0,7	0,1	0,9
4. Володин С.В.	0,8	0,6	0,1	0,85
5. Гайдаенко В.С.	0,7	0,65	0,18	0,75
6. Еремеев Г.А.	0,75	0,75	0,2	0,8
7. Жуков В.М.	0,7	0,67	0,15	0,7
8. Зайцева Ж.Н.	0,8	0,85	0,15	0,8
9. Каширская Н.А.	0,8	0,8	0,2	0,8
10. Лепин Б.М.	0,7	0,6	0,1	0,75
11. Лившиц А.Л.	0,75	0,8	0,2	0,85
12. Матюхин Н.Я.	0,85	0,8	0,15	0,8
13. Нечуятов А.А.	0,8	0,85	0,15	0,8
14. Новиков П.П.	0,75	0,7	0,2	0,75
15. Ольдерогге Г.Б.	0,75	0,8	0,2	0,8
16. Поплавский Р.П.	0,67	0,7	0,15	0,7
17. Рыбенков В.И.	0,75	0,7	0,2	0,8
18. Сухачев М.П.	0,75	0,7	0,2	0,8
19. Тимонов В.М.	0,7	0,67	0,15	0,75
20. Турков В.Е.	0,8	0,75	0,2	0,7
21. Фетисов А.Ф.	0,75	0,8	0,15	0,8
22. Халявин А.М.	0,7	0,75	0,2	0,8
23. Цепляев Ю.Ф.	0,75	0,75	0,15	0,7
24. Чикунов Н.В.	0,75	0,8	0,2	0,8
25. Шнейдер Б.Н.	0,7	0,7	0,2	0,75
26. Шульгин В.А.	0,8	0,75	0,15	0,8
27. Юдин Д.Б.	0,7	0,7	0,2	0,75
28. Янукьян Л.А.	0,85	0,75	0,1	0,85

Вычисление условных вероятностей перехода от исходной вершины

очередного допустимого пути к его конечной вершине, которые интерпретируются как последовательности искомых значений $\{Q_i\}$, выполняется программно. Терминология и расчетные процедуры общеизвестны. Для приведенного на рис. 3.6 примера с учетом мнений двадцати восьми экспертов, частично представленных в табл. 3.1, результаты сведены в табл. 3.2. В предположении нормального распределения ошибок экспертов (согласно, например, [7]) в качестве вычисляемой оценки $\tilde{m}_x$ математического ожидания для вероятности исполнения задач программного комплекса, имеем: $\tilde{m}_x = \dfrac{\sum_{i=1}^{n} x_i}{n}$, где $x_1, x_2, \ldots x_n$ – указанные экспертами значения вероятностей условных переходов, $i = \overline{1,n}$ – индекс суммирования, n – количество экспертов, участвующих в опросе. Истинное значение математического ожидания заключено в пределах доверительного интервала, определяемого по известной (например, [7]) формуле $P\left(\tilde{m} - t_\beta \sigma_{\tilde{m}} < m < \tilde{m} + t_\beta \sigma_{\tilde{m}}\right) = \beta$, где β – доверительная вероятность попадания в интервал, $\sigma_{\tilde{m}} = \dfrac{\tilde{\sigma}_x}{\sqrt{n}}$ – среднеквадратическое отклонение результатов обработки экспертных оценок, $\tilde{\sigma}_x = \sqrt{\dfrac{n}{n-1}\left(\dfrac{\sum_{i=1}^{n} x_i^{\,2}}{n} - \tilde{m}_x^{\,2}\right)}$ – среднеквадратическое отклонение величин, указанных экспертами. Табличное значение $t_\beta = 1{,}643$ соответствует доверительной вероятности $\beta = 0{,}9$. Остальные обозначения приведены выше.

Таблица 3.2

Характеристика ($\tilde{m}_x, \sigma_{\tilde{m}}, I_\beta$)	План ИВП	Этап ОВД	Взаимодействие	Целостность
Вероятность исполнения $\tilde{m}_x$	0,8	0,75	0,2	0,8
Среднеквадратическое отклонение $\sigma_{\tilde{m}}$	0,03	0,05	0,04	0,033
Доверительный интервал оценок I_β	0,751-0,849	0,668-0,832	0,134 - 0,266	0,757 - 0,854

Организация совместной работы компьютеров приводит к потерям производительности на взаимодействие, которые можно оценить статистически, используя последовательности значений $\{Q_i\}$ корреляции выполняемых программных функций по общим данным и отношениям предшествования. Расчет соответствующих величин Q_i не связан с трудностями концептуального характера, однако требует использования громоздких в вычислительном отношении процедур преобразования матриц большой размерности. Для упрощения процедуры в данном разделе предлагаются правила перехода от представления сложной программы как графа к ее отображению в виде

функций алгебры логики. В результате – громоздкие операции над матрицами замещаются линейно зависящими от количества n вершин графа обращениями к значащим (ненулевым) двоичным функциям, определенным на множестве функциональных компонент программного комплекса. Эффективность излагаемого подхода проверена на задачах оценки полноты и качества отладки программного обеспечения управляющих систем, работающих в реальном масштабе времени, а также на рассматриваемом здесь принципе расчета коэффициентов связности алгоритмов по управлению и данным.

Вопросы для самопроверки

1. Какая практическая потребность вызвала к жизни функции ОС по управлению сетевыми ресурсами и параллельных вычислений (п. 3.1.1)?
2. В чем состоит отличие понятий *многозадачная* ОС и *многопользовательская* ОС? *Мультипроцессорная* и *мультипрограммная* ОС? Что такое *нити, потоки, процессы* (п. 3.1.1)?
3. Какими критериями оценивается эффективность ОСРВ (п. 3.2.1)?
4. Какому состоянию системы на графе рис. 3.1 (п. 3.2.2) соответствует вероятность отказа в обслуживании? Вероятность потери заявки? Простоя?
5. Как изменяется величина вероятности потери заявки при переходе от одного компьютера к компьютерной сети (п. 3.3.2)?
6. В чем проявляется явление связности (корреляции) вычислительных задач? Как оно влияет на показатели качества работы системы (п. 3.3.3)?
7. Каким образом подготавливаются исходные данные для оценки эффективности вычислительного процесса: λ, μ, ρ, n, Q_i (п. 3.2. – 3.4)?

4. Управление вычислительным процессом

4.1. Диспетчеризация и планирование вычислений

4.1.1. Системы приоритетов и алгоритмы диспетчеризации. Инструментами управления поведением системы являются *приоритеты* процессов (задач) и *алгоритмы планирования* (диспетчеризации) ОСРВ. Выше (параграф 1.3) упоминалось, что в многозадачных ОС общего назначения используются, как правило, различные модификации *алгоритма круговой диспетчеризации*, основанные на понятии непрерывного кванта времени, предоставляемого процессу для работы. Планировщик по истечении каждого кванта просматривает очередь активных процессов и принимает решение, какому из них передать управление, основываясь на приоритетах процессов (присвоенных им численных значениях). Приоритеты могут быть фиксированными, могут меняться со временем, это зависит от принятых алгоритмов планирования, но рано или поздно процессорное время получают все процессы.

Алгоритмы круговой (циклической) диспетчеризации неприменимы в чистом виде в ОСРВ. Основной недостаток – непрерывность кванта времени, в течение которого процессором владеет только один процесс. Планировщи-

кам же операционных систем реального времени должна предоставляться возможность сменить процесс до истечения кванта, если в этом возникла необходимость. Один из возможных алгоритмов планирования при этом «приоритетный с вытеснением». Сфера ОСРВ отличается богатым разнообразием алгоритмов планирования: динамические, приоритетные, монотонные, адаптивные, цель которых одна – предоставить инструмент, позволяющий в каждый момент времени исполнять именно тот процесс, который необходим.

Для наглядности обратимся к упомянутой ранее (п. 1.2) проблеме «времени жизни» заявки, в связи с которой главной задачей в ОСРВ становится планирование, которое обеспечило бы предсказуемое поведение системы при любых обстоятельствах. Процесс с директивным сроком окончания должен стартовать и выполняться так, чтобы он уложился в назначенное время. Если это невозможно, процесс должен быть отклонен. В этом направлении развиваются два подхода – *статические и динамические алгоритмы планирования*. Первый используется для формального доказательства условий предсказуемости системы. Для его реализации необходимо планирование на основе приоритетов, прерывающих обслуживание. Приоритет заранее назначается каждому процессу. Процессы должны удовлетворять следующим условиям:

- процесс должен быть завершен за время его периода;
- процессы не зависят друг от друга;
- процессам требуется одинаковое время на каждом интервале;
- у непериодических процессов нет жестких сроков;
- прерывание процесса происходит за ограниченное время.

Процессы выполняются в соответствии с приоритетами. При планировании предпочтение отдается задачам с самыми короткими периодами выполнения. В динамических алгоритмах высший приоритет присваивается процессу, у которого осталось наименьшее время выполнения, что создает преимущества перед статической дисциплиной при больших загрузках.

Приоритетное прерывание обслуживания является неотъемлемой составляющей ОСРВ, так как должны существовать гарантии, что событие с высоким приоритетом будет обработано раньше события более низкого приоритета. Как следствие, ОСРВ нуждается не только в механизме планирования на основе приоритетов, прерывающих обслуживание, но также и в механизме управления прерываниями. Нужно уметь запрещать прерывания, когда должен быть выполнен критический код, который нельзя прерывать. Длительность обработки прерываний должна сводиться к минимуму. Различают абсолютные и относительные приоритеты. *Абсолютный приоритет* означает, что поступление прерывания немедленно прекращает процесс обслуживания менее приоритетной заявки, в то время как *дисциплина с относительным приоритетом* разрешает закончить ее обслуживание и лишь после этого приступить к работе над поступившей заявкой высокого приоритета. *Смешанная дисциплина* допускает сосуществование прерываний обоих типов.

ОСРВ должна обладать развитой системой приоритетов. Во-первых,

это требуется, потому что сама она может рассматриваться как набор серверных приложений, подразделяющихся на потоки, и несколько высоких уровней приоритетов должно быть назначено системным процессам и потокам. Во-вторых, в сложных приложениях необходимо все потоки реального времени помещать на разные приоритетные уровни, а все другие потоки – на один уровень (ниже, чем любые потоки реального времени). При этом потоки не реального времени можно планировать циклически.

При планировании с приоритетами необходимо решить две проблемы:

- обеспечить выполнение процесса с наивысшим приоритетом;
- не допустить инверсии приоритетов, когда высокоприоритетные задачи ожидают ресурсы, захваченные задачами с низкими приоритетами.

Для борьбы с инверсией приоритетов в ОСРВ используется механизм наследования приоритетов, однако при этом приходится отказываться от статического планирования, поскольку приоритеты становятся динамическими.

При управлении прерываниями обычно различают две процедуры:

- обработка прерывания – программа низкого уровня в ядре с ограниченными системными вызовами;
- поток обработки прерывания – поток уровня приложения, который управляет прерыванием, с доступом ко всем системным вызовам.

Обработка прерывания реализуется производителем аппаратуры, а драйверы устройств выполняют управление прерываниями с помощью потоков, которые действуют как любые другие потоки и используют ту же самую систему приоритетов. Используются аппаратные и программные средства, обеспечивающие временное прекращение выполнения последовательности команд для перехода к выполнению другой последовательности команд или для возвращения к ранее прерванной программе. Система прерываний позволяет процессору изменять свое состояние, если при выполнении программы возникла ошибка или вычисления по данной программе окончены, если для ввода или вывода подготовлены массивы данных и необходимо обратиться к соответствующим устройствам, если пользователю или управляемому объекту необходимо немедленно скорректировать данные, которые могут изменить ход вычислительного процесса. Во всех этих случаях отсутствие системы прерываний приводит к потере полезного машинного времени или невыполнению функций, возложенных на АС УВД.

Различают прерывания от схемы контроля компьютера, от устройств ввода-вывода информации, прерывания при обращении к ОСРВ, программные и внешние. Прерывания от системы контроля обеспечивают нахождение неисправности при сбоях и отказах. Прерывания от устройств ввода-вывода дают возможность ответить на запросы этих устройств о своевременном обмене информацией. Прерывания при обращении к ОС осуществляются с помощью специальных привилегированных команд, например, командой перехода к мультипрограммной работе. Программные прерывания вызываются неправильным заданием или использованием команд и данных (например, нарушена защита памяти, переполнена разрядная сетка и т.д.). Внешние пре-

рывания осуществляются с пульта оператора, с абонентских пунктов пользователей через линии связи, от объектов, работающих в реальном масштабе времени, от датчиков абсолютного и относительного времени.

В процессе выполнения программ могут появиться сигналы запроса от нескольких источников прерываний. Порядок нескольких прерываний определяется либо последовательностью их поступления, либо приоритетом, либо и тем и другим вместе. Приоритетный принцип состоит в задании порядка прерываний по значимости (например, прерывание от схем контроля имеет высший приоритет, так как продолжение вычислений становится нецелесообразным, пока причина отказа не будет устранена). Аппаратные средства обнаруживают сигналы запросов на прерывания, запоминают управляющую информацию, причины и коды прерываний и восстанавливают контекст после прерывания. Программные средства собирают управляющую информацию, определяют источник и анализируют причины прерываний, организуют обработку кодов прерываний. Имеется возможность управления прерываниями путем запрещения (маскирования) или разрешения их обработки.

Для организации прерываний используются различные службы времени. Операционная система отслеживает текущее время, в соответствии с ним запускает задачи и потоки и приостанавливает их на определенные интервалы. В службах времени ОСРВ используются часы реального времени. Обычно используются высокоточные аппаратные часы. Для каждого процесса и потока определяются часы процессорного времени. На базе этих часов создаются таймеры, которые измеряют расход времени процессом или потоком, позволяя динамически выявлять программные ошибки или ошибки вычисления максимально возможного времени выполнения. В высоконадежных, критичных ко времени системах важно выявление ситуаций, при которых задача превышает максимально возможное время своего выполнения, так как при этом работа системы может выйти за рамки допустимого времени отклика. Часы времени выполнения позволяют выявить возникновение перерасхода времени и активизировать соответствующие действия по обработке ошибок.

4.1.2. Организация сбора и обработки плановых сообщений. Рассмотрим метод анализа систем обслуживания с относительным приоритетом, отличающийся тем, что с целью получения расчетных формул применяются физические ограничения процесса образования очередей заявок. Это позволяет формализовать взаимное влияние парциальных входных потоков описанием их динамического равновесия и, как следствие, избежать ветвления графа переходов и состояний системы, сводя ее модель к простой композиции марковских цепей для каждой составляющей суммарного потока заявок. В качестве примера исследуем процессы приоритетного сбора и приоритетной обработки заявок на использование воздушного пространства в централизованной службе планирования полетов (ЦСПП), создаваемой для единой системы организации воздушного движения (ЕС ОрВД) России. В основе концепции централизации лежит опыт Западной Европы и США. Ее положения использованы в новой редакции «Табеля сообщений о движении воз-

душных судов в РФ» и в аэронавигационных справочниках.

Действующая в стране схема подачи и обработки планов полетов (флайт-планов – ФПЛ) и сообщений по их обновлению характеризуется рядом недостатков. Авиакомпания разрабатывает план полета, подает его в аэродромный диспетчерский пункт (АДП), определяет адреса рассылки в органы обслуживания воздушного движения (ОВД) по маршруту. АДП нужна полная информация об аэронавигационной инфраструктуре на всю глубину полета, что практически неосуществимо для всех АДП мира. Доля вылетов из иностранных аэропортов без отправления ФПЛ в ЕС ОрВД достигает 30%.

После подачи ФПЛ в АДП командир ВС производит вылет даже в том случае, если ФПЛ составлен с ошибками, не дошел до нужных адресатов, не учитывает текущую обстановку. Возникающие проблемы решаются органами ОВД уже во время полета. Очевидны следующие недостатки такой схемы использования воздушного пространства (ИВП):

1) В ЕС ОрВД не поступает значительная доля сообщений о движении ВС, что затрудняет реализацию разрешительного порядка ИВП России.

2) Вследствие ошибок адресации сообщения поступают в органы ОВД, не используемые подаваемым планом полета, где также подлежат обработке.

3) В органы ОВД, затрагиваемые полетом, нередко поступают телеграммы неудовлетворительного качества, требующие ручной обработки.

4) Отсутствует механизм воздействия центров ЕС ОрВД на составителей для достижения корректности, полноты и своевременности планов.

Считается, что ЦСПП создает следующие преимущества:

- централизация позволяет подавать плановые сообщения в единственный адрес, что практически исключает ошибки адресации, а все остальные исправляются однократно перед рассылкой;
- устанавливается обратная связь: если план не содержит ошибок, то он принимается; при наличии незначительных ошибок ЦСПП редактирует их и прилагает исправленный план в ответное сообщение; податель имеет право не согласиться с изменениями и представить новый план; если ошибки окажутся критическими, подателя уведомляют о причинах отклонения плана;
- централизация гарантирует, что после принятия плана все сообщения о движении ВС будут в установленные сроки направлены в необходимые адреса по маршруту.

Обеспечиваются следующие преимущества перед ныне существующей распределенной системой обработки планов полетов:

- своевременное получение всеми заинтересованными службами сообщений о движении ВС для планирования и контроля ИВП;
- соблюдение разрешительного порядка ИВП РФ, при котором полеты выполняются только при наличии подтверждения контролирующих органов;
- повышение качества, целостности и непротиворечивости плановой информации, поступающей в органы ОВД;
- снятие нагрузки с персонала ОВД по обработке телеграмм;
- повышение дисциплины подателей сообщений о движении ВС;

• создание основы для оптимизации сводного плана ИВП в результате сосредоточения всей необходимой информации о движении ВС в ЦСПП;

• улучшение ситуации для авиакомпаний (и АДП) за счет предоставления им данных об условиях обеспечения полетов и снятия с них задачи рассылки сообщений о движении ВС в воздушном пространстве (ВП) России.

Реализация концепции ЦСПП выдвигает следующие проблемы.

1. Концентрация функций обработки планов делает всю систему ОВД зависимой от работоспособности центра. Как следствие, к ЦСПП предъявляются повышенные требования по надежности. В частности, предполагается сохранить как резерв существующую схему сбора плановой информации, что приводит к необходимости поддерживать две системы одного назначения.

2. Централизация в масштабах страны требует, чтобы сообщения сначала направлялись в центр ЕС ОрВД, оттуда – в органы ОВД, затрагиваемые маршрутом. Соответственно увеличивается нагрузка на средства связи. Возникает вопрос о пропускной способности центра. Требуется разработка нового программного обеспечения и повышение ответственности пользователей ВП, в том числе иностранных, за выдерживание правильной последовательности сообщений о движении ВС, их своевременность и качество.

Исследуем характеристики обслуживания сообщений, поступающих в ЦСПП (вероятность потери сообщений, время ожидания и т. д.). Будем анализировать поток сообщений, который образуется при внедрении этой службы, оценим требования к аппаратным средствам обслуживания, определим области изменения параметров системы, при которых внедрение ЦСПП окажется оправданным. Основная задача следующего раздела складывается из решения таких вопросов как:

• анализ структуры потока сообщений, поступающих в ЦСПП;

• разработка метода исследования характеристик процесса обработки, наиболее полно учитывающего выявленную структуру потока сообщений;

• определение зависимости характеристик обслуживания от параметров системы, нахождение областей изменения этих параметров, при которых внедрение ЦСПП становится целесообразным;

• оценка достоверности найденных зависимостей.

4.1.3. Приоритетное обслуживание с общей очередью. В гражданской авиации (ГА) России телеграфные сообщения в зависимости от содержания и допустимого времени обработки подразделяются на следующие приоритетные категории срочности:

СС – телеграммы о чрезвычайных происшествиях в полете;

ДД – сообщения о чрезвычайных происшествиях на земле;

ФФ – для немедленной передачи экипажу ВС и о планах полетов;

ГГ – о посадках, задержках, отменах, возвратах рейсов;

ЙЙ – сообщения службы аэронавигационной информации;

КК – об административной и эксплуатационной деятельности ГА;

ЛЛ – телеграммы, которые не могут быть направлены авиапочтой.

Оценим характеристики приоритетного обслуживания в такой системе.

4.1.3.1. Постановка задачи. Традиционным инструментом анализа пропускной способности сетей связи является математический аппарат теории очередей [6]. Напомним, что его использование правомерно, если исследуемая система отвечает ряду ограничений, накладываемых как на ее структуру, так и на параметры входного потока и дисциплину обслуживания. В общепринятых терминах ЦСПП представляет собой многоканальную СМО с ограниченной очередью и относительным приоритетом. Входной поток телеграфных сообщений, согласно многочисленным экспериментам, подчиняется пуассоновскому распределению, обслуживание с учетом ручного исправления ошибок в телеграммах – экспоненциальное. Такая постановка вписывается в рамки модели Эрланга, однако не учитывает существенное ограничение. В авиационной сети циркулируют телеграммы различной приоритетности, определяемой характеристикой «серия срочности». Входной поток не является однородным, а граф переходов и состояний СМО нельзя отобразить классической цепью Маркова, как это удалось в разделе 3.2 (п. 3.2.2), со связями только между соседними сообщающимися состояниями, что затрудняет создание модели в целом.

В ряде случаев анализ систем с приоритетами удается свести к известным математическим схемам. Результаты в виде итерационных процедур и рекуррентных соотношений позволяют выразить вероятность любого состояния СМО через предшествующие состояния. Стационарные распределения описываются системой алгебраических уравнений конечного порядка. Используя специфическую структуру матрицы этой системы, можно получать искомые распределения. Однако алгоритмические и расчетные схемы настолько громоздки, что их применение ограничено самыми простыми случаями. Количество уравнений связано с числом входных потоков показательной зависимостью.

Развиваются эвристические модели, основанные на умении своих создателей выделить доминирующие закономерности исследуемого процесса и отвлечься от второстепенных связей и отношений. С этой целью вводятся дополнительные ограничения, сужающие область взаимного влияния различных системных факторов и позволяющие упростить их анализ.

Рассмотрим с этих позиций одну из самых распространенных дисциплин приоритетного обслуживания с приемом поступающих заявок на ИВП в общий БН объемом на r мест для ожидания (рис. 4.1). Основные закономерности проследим на двухприоритетной одноканальной модели, а затем распространим полученный результат на общий случай (произвольное количество m входных потоков и n каналов обслуживания).

Пусть на вход СМО поступают два пуассоновских потока заявок с интенсивностями λ_1 и λ_2 соответственно. Заявки первого типа обслуживаются с относительным приоритетом. Это означает, что если в момент поступления такой заявки уже производится обработка менее приоритетной заявки, то прерывания последней не происходит и она удовлетворяется. Лишь после этого единственный канал системы занимает заявка более высокого уровня

приоритетности. Выбор каждой следующей заявки из БН на обслуживание осуществляется по известному правилу. Сначала на обработку назначаются заявки высшего приоритета (ЗВП), и лишь при полном освобождении системы от них обслуживаются заявки низшего приоритета (ЗНП). Дисциплина приема в БН также основана на предпочтении ЗВП. В случае отсутствия в нем свободных мест поступающая ЗВП вытесняет из накопленной очереди ЗНП, последняя получает отказ в обслуживании и теряется. Отказ в приеме ЗВП возможен только в случае заполнения ими всего объема r БН.

Времена обслуживания распределены экспоненциально с параметрами

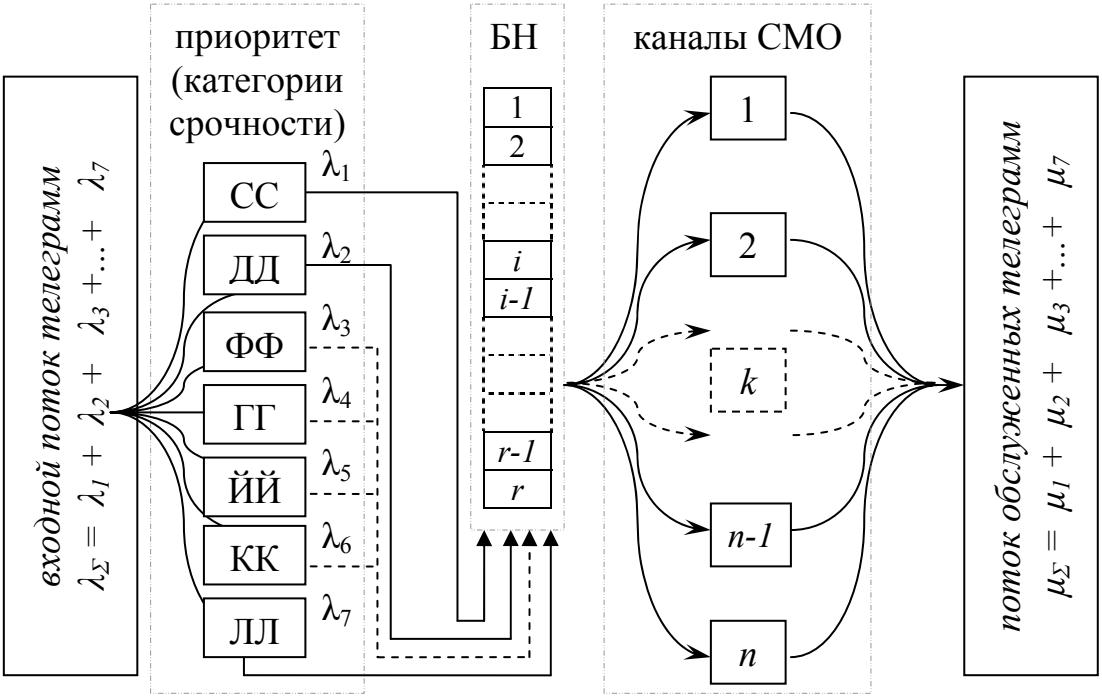


Рис. 4.1. Система обслуживания с относительными приоритетами и приоритетной записью в общий буферный накопитель.

μ_1 и μ_2 соответственно. Суммарная загрузка системы не превосходит единицы: $\rho_\Sigma = \rho_1 + \rho_2 < 1$; $\rho_1 = \lambda_1/\mu_1$; $\rho_2 = \lambda_2/\mu_2$. Заметим, что на характеристики обслуживания ЗВП второй поток воздействует лишь созданием занятости канала, т.е. при назначении на обработку принадлежащих ему неприоритетных заявок. В этих случаях канал как бы исключается из контура СМО, переходя в состояние простоя для заявок первого типа. В любое другое время в распоряжение ЗВП предоставлен весь ресурс системы. Следовательно, существенным показателем, характеризующим процесс обслуживания, становится соотношение γ значений T_i среднего времени обслуживания заявок разных потоков: $\gamma = T_2/T_1 = \mu_1/\mu_2$. Исследуем сформулированную модель.

4.1.3.2. Формализация задачи. Ключевым событием в исследуемой модели становится прием на обслуживание ЗНП в условиях отсутствия ЗВП в БН. Такое событие происходит с конечной вычисляемой ниже вероятностью. За время обслуживания одной ЗНП со средним значением $T_2 = 1/\mu_2$, в БН образуется очередь ЗВП, ожидающих освобождения канала. Пусть в сеансе обслуживания одной ЗНП длина L_1 накапливающейся очереди ЗВП не

превосходит объема r БН. Тогда все поступившие приоритетные заявки могут быть размещены в нем хотя бы и за счет вытеснения неприоритетных, и в стационарном режиме вероятность π_1 потери таких заявок определяется лишь создаваемой этим потоком загрузкой ρ_1 и полным объемом r БН

$$\pi_1 = \frac{\rho_1^{r+1}(1-\rho_1)}{1-\rho_1^{r+2}}, \text{ если } L_1 \le r.$$

Определение длины L_1 очереди ЗВП составляет ядро развиваемого метода. Речь идет о вероятностных мерах оценки функционирования системы, вследствие чего используется мода, или наиболее вероятное значение случайной величины L. Последнее, как показано в [6], для эрланговских моделей очень просто зависит от среднего значения и от коэффициента вариации ϑ, равного отношению среднеквадратического отклонения σ к среднему времени обслуживания T. Выражение для длины L_1 очереди ЗВП, образующейся в общем БН объемом r за время T_2 обслуживания одной ЗНП, выглядит как

$$L_1 = (1+\vartheta^2)\lambda_1 T_2 = (1+\vartheta^2)\rho_1 \gamma \le r. \qquad (4.1)$$

При нарушении условия (4.1) помимо потерь, обусловленных классическим выражением, при каждом взятии на обслуживание ЗНП возникают дополнительные потери ЗВП, которые нетрудно рассчитать. За время T_2 в СМО с наибольшей вероятностью поступают $(1+\vartheta^2)\rho_1 \gamma$ ЗВП, из которых в БН смогут разместиться лишь r. Следовательно, в каждом случае назначения одной ЗНП на обслуживание будут с наибольшей вероятностью потеряны $(1+\vartheta^2)\rho_1 \gamma - r$ ЗВП, что составит относительную долю ξ их потерь, равную

$$\xi = \frac{(1+\vartheta^2)\rho_1\gamma - r}{(1+\vartheta^2)\rho_1\gamma} = 1 - \frac{r}{(1+\vartheta^2)\rho_1\gamma}.$$

Вероятность события, при котором теряются ξ ЗВП, есть вероятность обслуживания ЗНП. Она пропорциональна загрузке ρ_2 системы заявками второго типа и вероятности P_2 того, что хотя бы одна такая заявка будет обслужена системой в условиях приоритетного приема в общий БН. Последняя легко определяется из физического смысла модели и ее особенностей: при невыполнении (4.1) весь БН предоставлен в распоряжение потока ЗВП. Заявки второго типа, даже если они ожидают в очереди, вытесняются из системы и, следовательно, их обслуживание осуществляется в области справа от точки $(1+\vartheta^2)\rho_1 \gamma = r$ по оси γ по правилам для СМО без БН ($r = 0$). Вероятность их потерь в такой модели: $\pi_2 = \frac{\rho_\Sigma(1-\rho_\Sigma)}{1-\rho_\Sigma^2} = \frac{\rho_\Sigma}{1+\rho_\Sigma}$, если $(1+\vartheta^2)\rho_1 \gamma > r$.

Суммирование загрузки по обоим потокам $\rho_\Sigma = \rho_1 + \rho_2$ подчеркивает тот факт, что обслуживание ЗНП производится лишь при отсутствии ЗВП. Вероятность P_2 обслуживания ЗНП в условиях приоритетной записи справа от точки $(1+\vartheta^2)\rho_1 \gamma = r$ по оси γ вычисляется как дополнение π_2 до единицы

$$P_2 = 1 - \pi_2 = 1 - \frac{\rho_\Sigma}{1+\rho_\Sigma} = \frac{1}{1+\rho_\Sigma}, \text{ если } (1+\vartheta^2)\rho_1 \gamma > r.$$

Тогда формула для оценки вероятности π_1 потери ЗВП при произвольных соотношениях L_1 и r

$$\pi_1 = \frac{\rho_1^{r+1}(1-\rho_1)}{1-\rho_1^{r+2}} + \delta\frac{\rho_\Sigma}{1+\rho_\Sigma}\left[1-\frac{r}{(1+\vartheta^2)\rho_1\gamma}\right],$$

где $\delta = \begin{cases} 0, & \text{если } (1+\vartheta^2)\rho_1\gamma \le r, \\ 1 & \text{в противном случае} \end{cases}$ – аналог символа Кронекера.

Выше была оценена вероятность π_2 потери ЗНП, которая при нарушении условия (4.1) определяется суммарной загрузкой системы и отсутствием мест для ожидания в БН, заполненном заявками первого типа. При выполнении (4.1) условия обслуживания ЗНП улучшаются вследствие появления в БН свободных от приоритетных заявок мест для ожидания. Наиболее вероятная длина r' свободного от ЗВП участка БН равна $r' = r - L_1 = r - (1+\vartheta^2)\rho_1\gamma$. Составная формула для вычисления π_2 принимает вид

$$\pi_2 = (1-\delta)\frac{\rho_\Sigma^{r-(1+\vartheta^2)\rho_1\gamma}(1-\rho_\Sigma)}{1-\rho_\Sigma^{r+1-(1+\vartheta^2)\rho_1\gamma}} + \delta\frac{\rho_\Sigma}{1+\rho_\Sigma}.$$

Показатели степени при ρ_Σ не включают количество каналов СМО (в данном случае $n = 1$) как места нахождения в системе, так как в стационарном режиме они предпочтительно заняты ЗВП. Зависимость вероятностей π_i потери заявок различных типов от ρ_i, r и γ приобретает вид семейства составных кривых с изломами в точке $L_1 = (1+\vartheta^2)\rho_1\gamma = r$. На рис. 4.2 представлены графики $\pi_{1,2} = f(\gamma)$, построенные для СМО с параметрами $\rho_1 = \rho_2 = 0.45$ и очередью $r = 10$. Сплошной линией изображены расчетные кривые, пунктирной линией – результаты статистического моделирования. Удаление по оси γ критической точки от начала отсчета определяется как $\gamma_{кр} = r/(1+\vartheta^2)\rho_1$ и для приведенного примера составляет около $\gamma_{кр} \approx 11$.

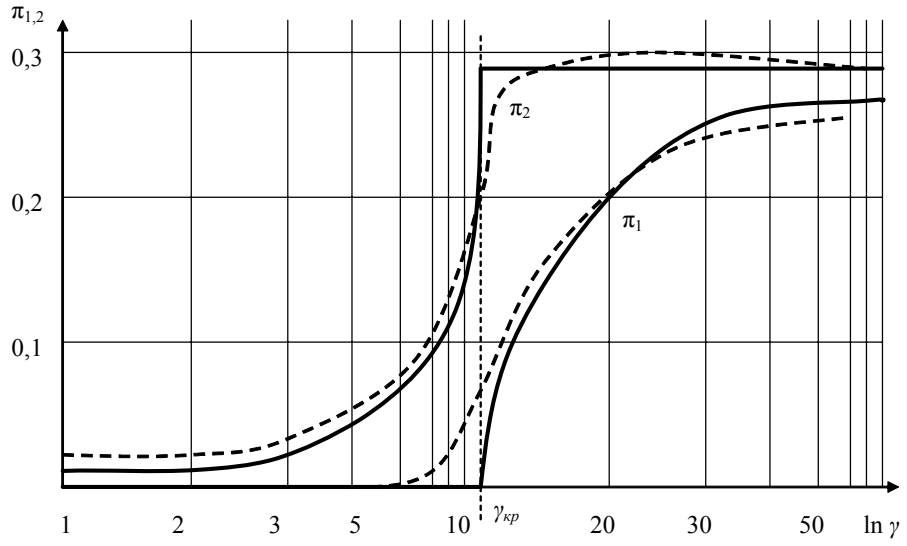


Рис. 4.2. График зависимости вероятностей потери заявок от соотношения средних значений времени их обслуживания в системе с двумя приоритетными входными потоками

Результаты обобщения представляются тремя аспектами (режимами) работы. «Щадящий» режим обслуживания заявок k-го уровня приоритетности объединяет все состояния СМО, в которых наиболее вероятная длина $L_{k\Sigma}$ очереди всех приоритетных потоков от высшего (первого) до k-го не превосходит количества r мест для ожидания.

В «критическом» режиме длина очереди заявок более высокого при-

оритета, чем *k*-й, может быть размещена в БН, однако заявки самого *k*-го потока в стационарном режиме частично вытесняются более приоритетными заявками. Наконец, в режиме «перегрузки» в сеансе обслуживания одной ЗНП в СМО накапливается очередь заявок более высокого приоритета, чем *k*-й, длина которой превосходит объем *r* БН. Рисунок 4.3 содержит семейство кривых $\pi_i = f(\gamma)$, $i = \{\overline{1,m}\}$, описывающих СМО с параметрами: $n = 1$; $m = 5$; $\rho_i = 0.45$ и $r = 10$. Расчетные кривые представлены сплошными линиями, результаты моделирования – пунктирными.

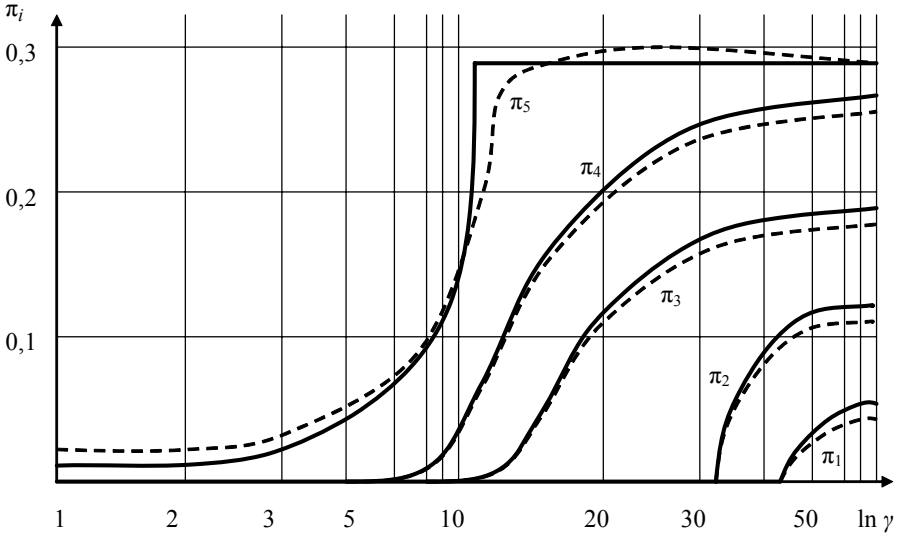


Рис. 4.3.

Семейство кривых $\pi_i = f(\gamma)$ для СМО с общим буферным накопителем и параметрами: $n = 1$; $m = 5$; $\rho_i = 0.45$ и $r = 10$

В «щадящем» режиме за период хранения одной записи *k*-го уровня приоритетности в БН упаковывается очередь поступающих заявок на полеты, начиная от высшего приоритета до *k*-го включительно. Наиболее вероятная длина накапливающейся очереди не превосходит объема *r* БН. Вероятность π_k потери заявки в функции параметров системы для *k*-го потока

$$\pi_k = \frac{\left(\sum_{i=1}^{k}\rho_i\right)^{r-\left(1+\vartheta^2\right)\gamma_k\sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right)}\left(1-\sum_{i=1}^{k}\rho_i\right)}{1-\left(\sum_{i=1}^{k}\rho_i\right)^{r-\left(1+\vartheta^2\right)\gamma_k\sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right)+1}}, \text{ если } \left(1+\vartheta^2\right)\gamma_k\sum_{i=1}^{k}\left(\frac{\rho_i}{\gamma_i}\right)\le r, \qquad (4.2)$$

где $\gamma_i = \mu_1/\mu_i$ – соотношение параметров обслуживания или обратных им величин среднего времени T_i хранения заявки в БН $\gamma_i = T_i/T_1$. В «критическом» режиме очередь входящих высокоприоритетных заявок, включая поток *k*-го типа, даже с учетом вытеснения ими записей низкого приоритета, превышает количество *r* мест для их хранения, однако без учета *k*-го потока может быть размещена (*k* – индекс потока). Тогда

$$\pi_k = \frac{\left(\sum_{i=1}^{k}\rho_i\right)^{r-\left(1+\upsilon^2\right)\gamma_k\sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right)}\left(1-\sum_{i=1}^{k}\rho_i\right)}{1-\left(\sum_{i=1}^{k}\rho_i\right)^{r-\left(1+\upsilon^2\right)\gamma_k\sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right)+1}} + \sum_{j=k}^{m}\delta_{jk}\,\rho_j\,\frac{1}{1+\sum_{i=1}^{k}\rho_i}\left\{1-\frac{\gamma_k\left[r-\left(1+\vartheta^2\right)\gamma_j\sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right)\right]}{\left(1+\vartheta^2\right)\gamma_j\,\rho_k}\right\}, \qquad (4.3)$$

если $\left(1+\vartheta^2\right)\gamma_j \sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right) \le r < \left(1+\vartheta^2\right)\gamma_j \sum_{i=1}^{k}\left(\frac{\rho_i}{\gamma_i}\right)$,

где δ_{jk} – аналогичный символу Кронекера переключатель; $j,k=\overline{1,m}$, m – количество входных потоков.

В режиме «перегрузки» очередь заявок, имеющих даже более высокий, чем k-й, приоритет, превосходит количество r мест для ожидания

$$\pi_k = \sum_{i=1}^{k}\rho_i \Big/ \left(1+\sum_{i=1}^{k}\rho_i\right), \text{ если } \left(1+\vartheta^2\right)\gamma_j \sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right) > r\,. \qquad (4.4)$$

4.2. Прием заявок в раздельные секции буферного накопителя. Широкое распространение данной дисциплины объясняется упорядоченностью заявок на входе СМО. Используя допущение о правомерности замены случайной величины длины очереди ее наиболее вероятным значением L, рассмотрим модель процесса образования и удаления записей в системе. Специфика обслуживания состоит в том, что вероятность потери заявки произвольного k-го потока, $k=\overline{1,m}$, определяется не динамически изменяющимся остатком БН, свободным от записей i-го приоритета, $i=\overline{1,k-1}$, а фиксированным размером собственной секции файла. Тогда резкие изменения поведения семейства кривых на графике $\pi_i = f(\gamma)$, т.е. точки

$$\gamma_{кр} = r/(1+\vartheta^2)\sum_{i=1}^{k}\rho_i$$

излома, определяются таким соотношением параметров, при которых за сеанс обслуживания одной заявки j-го приоритета, $j=\overline{k,m}$, в k-й секции разделенного БН образуется очередь, наиболее вероятная длина L_k которой превышает количество r_k имеющихся в ней мест для ожидания (рис. 4.4).

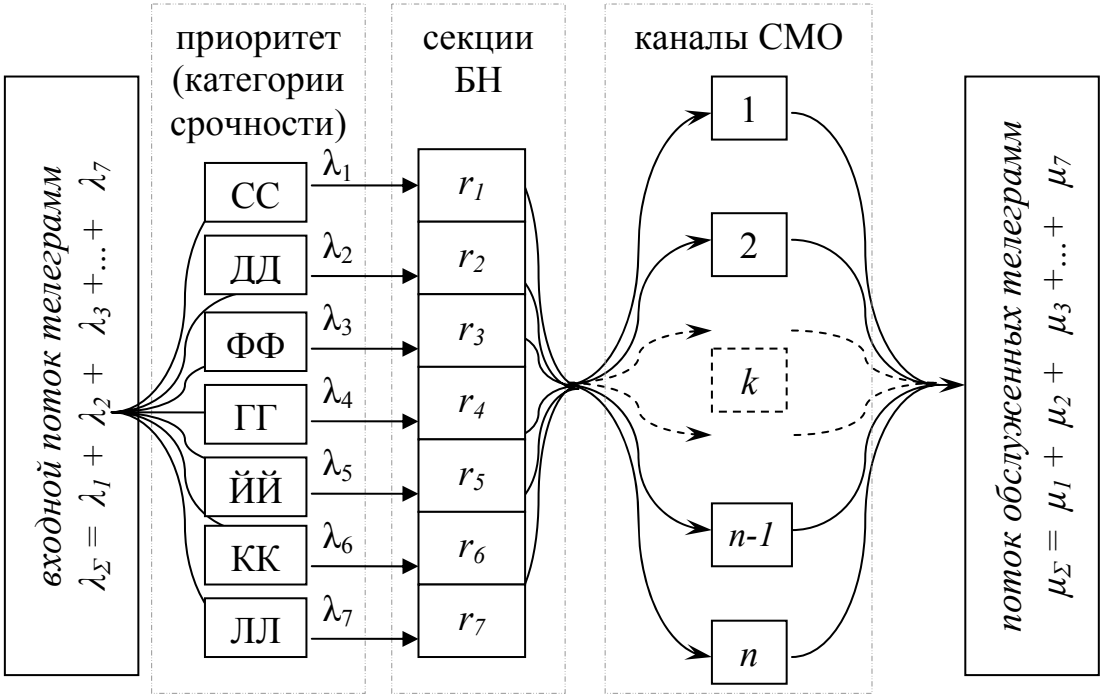


Рис. 4.4. Система обслуживания с относительными приоритетами и приемом заявок в раздельные секции буферного накопителя.

4.2.1. Модель с двумя входными потоками. Принципиальные закономерности работы СМО с секционированным БН проследим на одноканальной двухприоритетной модели. Пусть для каждого входящего потока выделены собственные секции объемами r_1 для ЗВП и r_2 для ЗНП. Потоки – пуассоновские с интенсивностями λ_1 и λ_2 соответственно. Времена обслуживания – экспоненциальные с параметрами μ_1 и μ_2, причем $\gamma = \mu_1 / \mu_2$, а суммарная загрузка $\rho_\Sigma = \rho_1 + \rho_2 < 1$. В силу последнего условия $\rho_1 < 1$ и, следовательно, вероятность освобождения системы от ЗВП, при котором на обслуживание назначается ЗНП, конечна. За время ее обработки, среднее значение T_2 которого составляет $T_2 = 1/\mu_2$, в буферной секции первого потока образуется очередь, наиболее вероятная длина которой $Q_1 = (1 + \vartheta^2)\rho_1\gamma$. Введение обозначения Q_i длины очереди в отличие от символа L_k в предыдущем разделе продиктовано стремлением формально подчеркнуть, что при переходе от потока к потоку она не суммируется, как это было в модели с общим БН, а определяется отдельно по каждому потоку. Если

$$Q_1 = (1+\vartheta^2)\rho_1\gamma \le r_1, \qquad (4.5)$$

т.е. накапливающаяся очередь может быть размещена в соответствующей буферной зоне, то согласно введенному выше допущению, на характеристики обслуживания ЗВП второй поток не оказывает существенного влияния. Следовательно, при выполнении неравенства (4.5) вероятность π_1 потери заявки первого типа может быть приближенно оценена как

$$\pi_1 = \frac{\rho_1^{r_1+1}(1-\rho_1)}{1-\rho_1^{r_1+2}}, \quad \text{если } (1+\vartheta^2)\rho_1\gamma \le r_1.$$

Действие формулы предполагается справедливым в области слева от точки излома $\gamma_{кр} = r_1 / (1 + \vartheta^2)\rho_1$. Для полного исследования поведения функции $\pi_1 = f(\gamma)$ справа от $\gamma_{кр}$ рассчитаем по аналогии с материалами предыдущего параграфа долю ξ_1 потерь ЗВП за время обслуживания одной ЗНП

$$\xi_1 = 1 - r_1 / (1 + \vartheta^2)\rho_1\gamma,$$

а также вероятность события, при котором теряются ξ_1 ЗВП, пропорциональную загрузке ρ_2 системы потоком ЗНП и вероятности P_2 обслуживания ЗНП в условиях приоритетной выборки ЗВП, являющейся дополнением до единицы вероятности π_2 потери ЗНП. На этом аналогия с моделью СМО с общим БН

$$q_1 = \begin{cases} (1+\vartheta^2)\rho_1\gamma \le r_1, \text{ если условие (3.1) выполняется,} \\ r_1 \text{ в противном случае.} \end{cases}$$

исчерпывается. Вероятность π_2 потери ЗНП зависит от соотношения длины Q_2, образующейся во второй секции очереди с количеством имеющихся в ней мест для ожидания, а не определяется отсутствием очереди ЗНП справа от $\gamma_{кр}$. Накопление ЗНП происходит во время занятости СМО обслуживанием ЗВП. Рассчитаем наиболее вероятное значение длительности этого времени. Длину q_1 очереди ЗВП можно оценить выражением

Очередь q_1 образуется в единичном сеансе обслуживания ЗНП. Наиболее вероятное значение времени полного освобождения системы от q_1 ЗВП можно представить как $T_q = (1 + \vartheta^2)q_1T_1$. За этот период в секции БН,

отведенной для первого потока, могут поступить и быть принятыми еще q_1' заявок, причем $q_1' = (1 + \vartheta^2)\lambda_1 T_q = (1 + \vartheta^2) q_1 \lambda_1 T_1 = (1 + \vartheta^2)\, q_1 \rho_1$.

Обратим внимание на неизменность степени множителя $(1 + \vartheta^2)$. Возрастает степень вычисляемого значения, а переход к наиболее вероятной величине всякий раз происходит заново. Продолжая анализ, нетрудно заметить, что и за время обслуживания очереди длиной q_1' в системе накапливается новая порция q_1'' ожидающих обработки ЗВП с наиболее вероятной длиной

$$q_1'' = (1 + \vartheta^2) q_1' \lambda_1 T_1 = (1 + \vartheta^2)\, q_1 \rho_1^2.$$

Вообще за период занятости системы ЗВП с учетом убывающего потока приращений q_1', q_1'',..., $q_1^{(l)}$ будет обслужена очередь, наиболее вероятное значение q_1^* длины которой складывается из частичных сумм $q_1^{(i)}$

$$q_1^* = (1 + \vartheta^2)\,(1 + \rho_1 + \rho_1^2 + \ldots + \rho_1^i + \ldots + \rho_1^l)\, q_1.$$

Последовательность $\{\rho_1^i\}$ в скобках представляет собой геометрическую прогрессию со знаменателем $\rho_1 < 1$, а величина q_1 в исследуемой области (справа от точки излома $\gamma_{кр}$) равна r_1. Следовательно

$$q_1^* = \frac{(1+\vartheta^2)(1-\rho_1^{l+1})}{1-\rho_1}.$$

Показатель степени при ρ_1^l определяется из тех соображений, что минимальная длина очереди, способной загрузить систему, не превосходит единицы: $(1+\vartheta^2) r_1 \rho_1^l \le 1$; $]l[= -\ln[(1+\vartheta^2) r_1] / \ln \rho_1$. Тогда наиболее вероятное значение длительности T периода занятости системы обслуживанием очереди q_1^* ЗВП с учетом убывающих приращений ее длины составляет

$$T = (1+\vartheta^2)\, q_1^*\, T_1 = (1+\vartheta^2)(1-\rho_1^{l+1})\, r_1 T_1 / (1-\rho_1).$$

За это время в секции БН объемом r_2 мест для ожидания, отведенной ЗНП, накопится очередь, наиболее вероятная длина Q_2 которой

$$Q_2 = (1+\vartheta^2)\,\lambda_2\, T = (1+\vartheta^2)(1-\rho_1^{l+1})\, r_1 T_1 \lambda_2 / (1-\rho_1) = (1+\vartheta^2)(1-\rho_1^{l+1})\, r_1 \rho_2 / \gamma (1-\rho_1).$$

Очевидно, что если $Q_2 > r_2$, то за время освобождения СМО от очереди q_1^* ЗВП вторая секция БН будет переполняться, что повлечет за собой дополнительные потери, доля ξ_2 которых равна

$$\xi_2 = 1 - \frac{r_2}{r_1} \frac{\gamma (1-\rho_1)}{(1+\vartheta^2)(1-\rho_1^{l+1})\, \rho_2}.$$

В противном случае, при $Q_2 \le r_2$, потери ЗНП определяются только объемом r_2 собственной секции БН и суммарной загрузкой системы ρ_Σ, так как их обслуживание возможно лишь после освобождения СМО от ЗВП

$$\pi_2 = \frac{\rho_\Sigma^{r_2}(1-\rho_\Sigma)}{1-\rho_\Sigma^{r_2+1}}, \quad \text{если } \frac{(1+\vartheta^2)(1-\rho_1^{l+1})\, r_1 \rho_2}{\gamma (1-\rho_1)} \le r_2.$$

При этом вероятность P_2 обслуживания ЗНП составляет

$$P_2 = 1 - \pi_2 = 1 - \frac{\rho_\Sigma^{r_2}(1-\rho_\Sigma)}{1-\rho_\Sigma^{r_2+1}} = \frac{1-\rho_\Sigma^{r_2}}{1-\rho_\Sigma^{r_2+1}}, \quad \text{если } \frac{(1+\vartheta^2)(1-\rho_1^{l+1})\, r_1 \rho_2}{\gamma (1-\rho_1)} \le r_2.$$

В случае $Q_2 > r_2$, когда каждый сеанс обслуживания ЗНП приводит к переполнению обеих секций, формула для вероятности P_2 ее обслуживания учитывает вызываемые этим сеансом дополнительные потери. Она рассчитывается как пересечение вероятностей нахождения в БН хотя бы одного

места для ожидания и доли ξ_2 потерь ЗНП вследствие его переполнения при каждом обслуживании ЗНП

$$P_2 = (1-\pi_2)(1-\xi_2) = \frac{1-\rho_\Sigma^{r_2}}{1-\rho_\Sigma^{r_2+1}} \frac{r_2}{r_1} \frac{\gamma(1-\rho_1)}{(1+\vartheta^2)(1-\rho_1^{l+1}) r_1 \rho_2} > r_2 .$$

Получив все вспомогательные величины, определяющие характеристики обслуживания ЗВП, т.е. долю ξ_1 потерь вследствие переполнения первой секции в сеансе обслуживания одной ЗНП и вероятности событий, при которых могут происходить потери ξ_1 ЗВП, можно записать

$$\pi_1 = \begin{cases} \dfrac{\rho_1^{r_1+1}(1-\rho_1)}{1-\rho_1^{r_1+2}}, \quad \text{если } (1+\vartheta^2)\rho_1\gamma \le r_1; \\ \dfrac{\rho_1^{r_1+1}(1-\rho_1)}{1-\rho_1^{r_1+2}} + \rho_2\left[1-\dfrac{r_1}{(1+\vartheta^2)\rho_1\gamma}\right]\dfrac{1-\rho_\Sigma^{r_2}}{1-\rho_\Sigma^{r_2+1}}, \\ \text{если } (1+\vartheta^2)\rho_1\gamma > r_1 \text{ и } \dfrac{(1+\vartheta^2)(1-\rho_1^{l+1}) r_1\rho_2}{\gamma(1-\rho_1)} \le r_2. \\ \dfrac{\rho_1^{r_1+1}(1-\rho_1)}{1-\rho_1^{r_1+2}} + \left[1-\dfrac{r_1}{(1+\vartheta^2)\rho_1\gamma}\right]\dfrac{1-\rho_\Sigma^{r_2}}{1-\rho_\Sigma^{r_2+1}}\dfrac{r_2}{r_1}\dfrac{\gamma(1-\rho_1)}{(1+\vartheta^2)(1-\rho_1^{l+1}) r_1\rho_2}, \\ \text{если } (1+\vartheta^2)\rho_1\gamma > r_1 \text{ и } \dfrac{(1+\vartheta^2)(1-\rho_1^{l+1}) r_1\rho_2}{\gamma(1-\rho_1)} > r_2. \end{cases}$$

Отметим характерное отличие полученного выражения от формулы для дисциплины с общим БН. Вероятность потери ЗВП в зависимости от соотношения параметров теперь имеет либо одну, либо две точки излома: обязательную при $\gamma_{кр} = r_1/(1+\vartheta^2)\rho_1$, когда переполняется собственная зона и значение π_1 резко возрастает, и другую, когда переполняется секция ЗНП и нарастание π_1 замедляется – в точке $\gamma_{кр2} = (1+\vartheta^2)(1-\rho_1^{l+1}) r_1 \rho_2 / r_2 (1-\rho_1)$.

Приближенное выражение для вероятности потери ЗНП фактически уже получено при анализе возможных аспектов обслуживания ЗВП

$$\pi_2 = \frac{\rho_\Sigma^{r_2}(1-\rho_\Sigma)}{1-\rho_\Sigma^{r_2+1}} + \delta \frac{1-\rho_\Sigma^{r_2}}{1-\rho_\Sigma^{r_2+1}}\left[1-\frac{r_2}{q_1}\frac{\gamma(1-\rho_1)}{(1+\vartheta^2)(1-\rho_1^{l+1})\rho_2}\right], \text{ где}$$

$$\delta = \begin{cases} 0, \text{ если } (1+\vartheta^2)(1-\rho_1^{l+1}) r_1\rho_2 / \gamma(1-\rho_1) \le r_2, \\ 1 \text{ в противном случае,} \end{cases} \qquad q_1 = \begin{cases} (1+\vartheta^2)\rho_1\gamma, \text{если } (1+\vartheta^2)\rho_1\gamma \le r_1, \\ r_1 \text{ в противном случае.} \end{cases}$$

Приведенные выражения подчеркивают функциональную зависимость вероятностей π_i потери заявки от соотношения γ параметров обслуживания различных потоков в системах с относительными приоритетами. Данная зависимость обнаруживалась в ряде работ с помощью метода статистического моделирования, однако преподносилась обычно как аксиома, не нуждающаяся в объяснении. Вместе с тем обосновать ее можно достаточно простыми рассуждениями. С возрастанием γ при неизменности других параметров снижается интенсивность потока ЗНП и как следствие – вероятность образования в единицу времени очереди, превышающей размер второй секции БН (длина Q_2 очереди обратно пропорциональна γ). При этом период занятости системы ЗВП фиксируется сверху объемом r_1 выделенной им секции БН. В

этих условиях вероятность потери ЗНП уменьшается. Вследствие этого с ростом γ повышается частота обслуживания таких заявок и увеличивается вероятность события, при котором теряются ξ_1 ЗВП, что и приводит к резкому возрастанию вероятности π_1 их потери. Скорость возрастания π_1 зависит от условий обслуживания ЗНП; если в процессе освобождения системы от ЗВП вторая секция БН не переполняется, то вероятность π_2 потерь ЗНП незначительна и π_1 возрастает резко; в противном случае рост π_1 ограничивается фиксированным объемом r_2 буферной зоны потока ЗНП. При этом начинает играть роль соотношение r_2/r_1 объемов секций БН, аналитически выведенная выше и установленная эмпирически в известных работах.

Данные расчетов по формулам подтверждаются результатами статистического моделирования. На рис. 4.5*а* представлены графики для системы с параметрами: $n = 1$; $\rho_1 = \rho_2 = 0.45$; $m = 2$; $r_1 = 2$; $r_2 = 8$; на рис. 4.5*б* – при тех же n, m и ρ_i, но с одинаковыми размерами секций БН $r_1 = r_2 = 5$; на рис. 4.5*в* – при сохранении n, m и ρ_i, но $r_1 = 8$ и $r_2 = 2$. Нетрудно видеть, как с увеличением соотношения γ интенсивностей обслуживания снижается эффективность приоритетных систем обсуждаемого класса. Вероятности потери заявок различных потоков не только выравниваются, как это было в модели с общим БН, но обнаруживается недопустимая тенденция: ЗВП теряются чаще ЗНП.

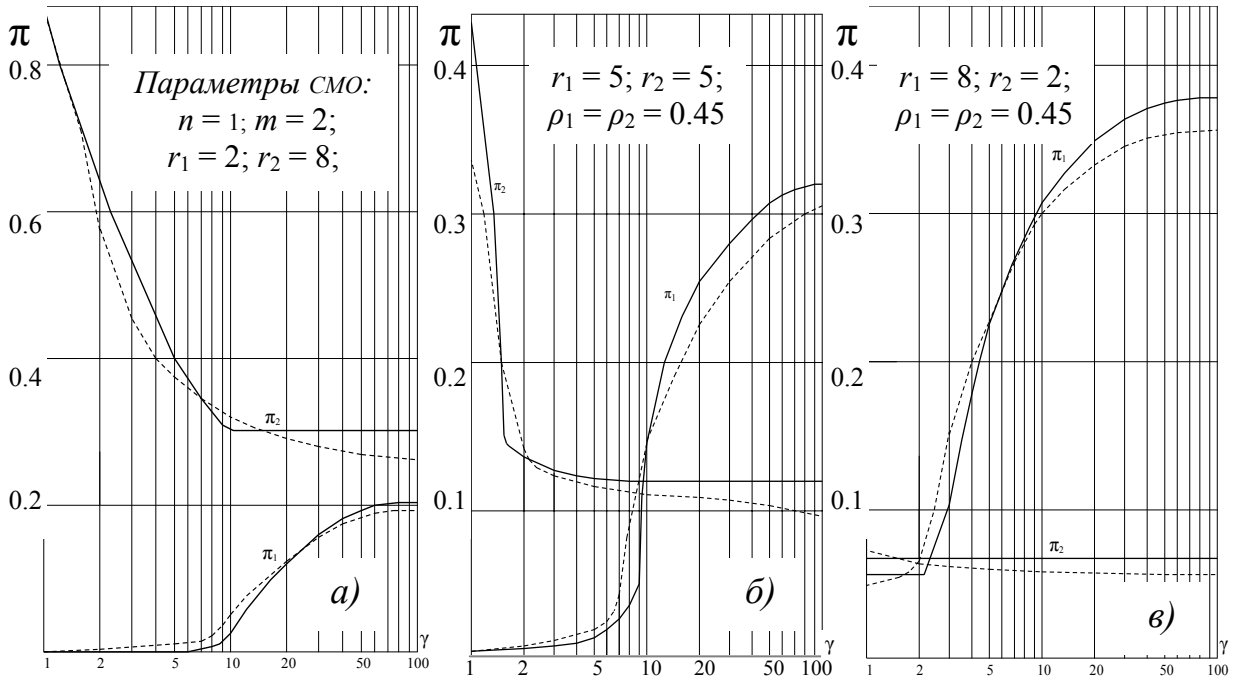


Рис. 4.5. Зависимости вероятностей потери заявки в системе с приоритетным обслуживанием и приемом в раздельные секции БН

Выведенная ниже формула для произвольного числа потоков показывает богатую палитру возможных деформаций отношений предпочтения. Так, для трехприоритетной системы в наихудших условиях как по критерию π, так и с точки зрения ожидания в очереди оказываются заявки второго потока.

4.2.2. Произвольное количество входящих потоков. Распространим

полученный результат на произвольное количество входных потоков. Пусть мы имеем одноканальную (n = 1) СМО, обслуживающую с относительным приоритетом m пуассоновских входных потоков с интенсивностями λ_i каждый ($i = \overline{1,m}$). Для наглядности сначала будем считать, что назначение приоритетов выполнено в соответствии с возрастанием величин γ_i. В дальнейшем это ограничение снимается за счет введения последовательности символов δ_{jk}. Времена обслуживания заявок всех потоков независимы и распределены экспоненциально с параметрами μ_i, причем $\gamma_i = \mu_1 / \mu_i$. Каждому i-му приоритетному потоку выделена собственная зона БН (секция файла) объемом на r_i мест для ожидания. Рассчитаем вероятность π_k потери заявки произвольного k-го потока в функции параметров системы.

Условие обязательного излома кривой π_k, определяемого процессом заполнения k-й секции БН, принимает вид $Q_k = (1+\vartheta^2)\,\lambda_k\,T$, где T – время освобождения СМО от заявок более высоких, чем k-й, приоритетов. По аналогии с двухприоритетной моделью нетрудно установить, что общее количество φ изломов кривой $\pi_k = f(\gamma)$ может колебаться от одного до $\varphi_{\max} = m - k + 1$, т.е. зависит не только от условий обслуживания заявок низших, чем k-й, приоритетов (с индексами $j > k$), но и от места k-го потока в шкале приоритетов.

Для определения величины времени T рассмотрим процесс образования очереди $L_k - 1$ заявок более высоких, чем k, приоритетов, исходя из соотношения $L_{k-1} = \sum_{i=1}^{k-1} Q_i$, где $Q_i = \begin{cases} (1+\vartheta^2)\,\gamma_k \left(\dfrac{\rho_i}{\gamma_i}\right), \text{ если } Q_i \le r_i, \\ r_i \text{ в противном случае.} \end{cases}$

Наиболее вероятное значение T_{Qi} занятости системы обслуживанием очереди Q_i заявок составит $T_{Qi} = (1+\vartheta^2)\,Q_i\,T_i$, где T_i – среднее время обслуживания заявки i-го типа, и для очереди $L_{k\text{-}1}$ можно приближенно записать

$$T_L = \sum_{i=1}^{k-1} T_{Qi} = (1+\vartheta^2)\,\gamma_k \sum_{i=1}^{k-1} \left(T_i\,\rho_i/\gamma_i\right).$$

В течение этого времени в системе образуется приращение $\Delta L'$ очереди заявок потоков высших приоритетов с индексами $i < k$. В силу стационарности и условия $\sum_{i=1}^{m} \rho_i < 1$ интенсивности λ_i поступлений таких заявок меньше параметров μ_i их обслуживания и суммарная очередь $L_{k\text{-}1}$ вместе с приращениями $\Delta L'$, $\Delta L''$, …, ΔL^{l_i} со временем укорачивается

$$\Delta L' = (1+\vartheta^2)\,\gamma_k \sum_{i=1}^{k-1} \left(\frac{\rho_i}{\gamma_i}\right)^2, \quad \Delta L'' = (1+\vartheta^2)\,\gamma_k \sum_{i=1}^{k-1} \left(\frac{\rho_i}{\gamma_i}\right)^3 \ldots$$

Вообще при расчете $L_{k\text{-}1}$ можно рассмотреть l_i монотонно убывающих приращений суммарной очереди заявок i-х уровней приоритетности, более высоких, чем k-й, ($i < k$). Соответственно процедура вычисления $L_{k\text{-}1}$ заканчивается на l_i-м шаге, когда (как и в модели с общим БН) получаем приращение очереди, не превышающее единицы. Тогда считаем, что система полностью освободилась от ЗВП, принадлежащих потокам с индексами $i < k$, и может приступить к обслуживанию заявок k-го типа.

В силу того, что поступление заявок разных типов пропорционально интенсивностям соответствующих входящих потоков, для выполнения последнего неравенства достаточно потребовать $(1+\vartheta^2)\gamma_k(\rho_i/\gamma_i)^{l_i} \le \lambda_i \Big/ \sum_{j=1}^{k-1}\lambda_j$, откуда $\Delta L^{l_i} = (1+\vartheta^2)\gamma_k \sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right)^{l_i} \le 1$, а $]l_i[= \ln\lambda_i - \ln\sum_{j=1}^{k-1}\lambda_j \Big/ \ln\left[(1+\vartheta^2)\gamma_k\frac{\rho_i}{\gamma_i}\right]$.

Тогда наиболее вероятная длина L очереди всех приоритетных заявок с индексами $i < k$, образующейся в системе за время обслуживания одной заявки j-то типа, $j = \overline{k,m}$, с учетом l_i приращений ее длины при освобождении от $L_{k\text{-}1}$ ЗВП, составит

$$L = (1+\vartheta^2)\gamma_k \sum_{i=1}^{k-1}\left[1-\left(\frac{\rho_i}{\gamma_i}\right)^{l_i+1}\right] \Big/ \left(1-\frac{\rho_i}{\gamma_i}\right),$$

а период T занятости системы обслуживанием накопленной очереди L заявок

$$T = (1+\vartheta^2)\gamma_k \sum_{i=1}^{k-1}T_i\left[1-\left(\frac{\rho_i}{\gamma_i}\right)^{l_i+1}\right] \Big/ \left(1-\frac{\rho_i}{\gamma_i}\right).$$

За это время в k-й секции разделенного БН образуется очередь заявок k-го типа, наиболее вероятная длина Q_k которой равна

$$Q_k = (1+\vartheta^2)\rho_k \sum_{i=1}^{k-1}\gamma_i\left[1-\left(\frac{\rho_i}{\gamma_i}\right)^{l_i+1}\right] \Big/ \left(1-\frac{\rho_i}{\gamma_i}\right).$$

Очередь Q_k определяет условие обязательного излома (по переполнению собственной зоны БН) кривой $\pi_k = f(\gamma)$

$$Q_k = (1+\vartheta^2)\rho_k \sum_{i=1}^{k-1}\gamma_i\left[1-\left(\frac{\rho_i}{\gamma_i}\right)^{l_i+1}\right] \Big/ \left(1-\frac{\rho_i}{\gamma_i}\right) \le r_k. \qquad (4.6)$$

При выполнении неравенства (4.6) потери заявок k-го типа определяются суммарной загрузкой системы первыми k потоками и объемами r_k собственных секций БН

$$\pi_k = \left(\sum_{i=1}^{k}\rho_i\right)^{r_k}\left(1-\sum_{i=1}^{k}\rho_i\right) \Big/ 1-\left(\sum_{i=1}^{k}\rho_i\right)^{r_k+1},$$

если $Q_k \le r_k$. Справа от точки излома потери возрастают за счет переполнения k-й секции БН в сеансе обслуживания одной заявки j-го типа, $j = \overline{k,m}$, причем доля ξ_k потерь при этом составляет

$$\xi_k = \frac{Q_k - r_k}{Q_k} = 1 - r_k \Big/ (1+\vartheta^2)\rho_k \sum_{i=1}^{k-1}\gamma_i\left[1-\left(\frac{\rho_i}{\gamma_i}\right)^{l_i+1}\right] \Big/ \left(1-\frac{\rho_i}{\gamma_i}\right).$$

Вероятность события, при котором за время T_j обслуживания одной заявки j-го типа ($j > k$) теряются ξ_k заявок k-го типа, пропорциональна загрузке ρ_j системы потоком j-й приоритетности и вероятности обслуживания заявки k-го типа при условии возможного прореживания j-го потока

$$P_k = (1-\pi_k)(1-\xi_k) = \frac{1-\left(\sum_{i=1}^{k}\rho_i\right)^{r_k}}{1-\left(\sum_{i=1}^{k}\rho_i\right)^{r_k+1}}\, r_k \Big/ (1+\vartheta^2)\,\rho_k \sum_{i=1}^{k-1}\gamma_i \left[1-\left(\frac{\rho_i}{\gamma_i}\right)^{l_i+1}\right] \Big/ \left(1-\frac{\rho_i}{\gamma_i}\right).$$

Прореживание j-го потока имеет место при нарушении сходного с (4.6) условия $Q_j \le r_j$; если же оно выполняется, то $P_k = 1 - \pi_k$. Окончательная форма записи приближенного выражения для оценки вероятности π_k потери заявки произвольного k-го потока принимает вид

$$\pi_k = \frac{\left(\sum_{i=1}^{k}\rho_i\right)^{r_k}\left(1-\sum_{i=1}^{k}\rho_i\right)}{1-\left(\sum_{i=1}^{k}\rho_i\right)^{r_k+1}}, \quad \text{если для всех } j=\overline{k+1,m} \text{ справедливо } (1+\vartheta^2)\gamma_j\frac{\rho_k}{\gamma_k} \le r_k;$$

$$\pi_k = \frac{\left(\sum_{i=1}^{k}\rho_i\right)^{r_k}\left(1-\sum_{i=1}^{k}\rho_i\right)}{1-\left(\sum_{i=1}^{k}\rho_i\right)^{r_k+1}} + \sum_{j=1}^{m}\delta_{jk}\,\rho_j \frac{1-\left(\sum_{i=1}^{k}\rho_i\right)^{r_k}}{1-\left(\sum_{i=1}^{k}\rho_i\right)^{r_k+1}}\left[1-\frac{r_k\cdot\gamma_k}{(1+\vartheta^2)\cdot\gamma_j\cdot\rho_k}\right],$$

если хотя бы для одного j, $j=\overline{k+1,m}$, $(1+\vartheta^2)\gamma_j\frac{\rho_k}{\gamma_k} > r_k$ и очередь

$$Q_k = (1+\vartheta^2)\,\rho_k \sum_{i=1}^{k-1}\gamma_i\left[1-\left(\frac{\rho_i}{\gamma_i}\right)^{l_i+1}\right] \Big/ \left(1-\frac{\rho_i}{\gamma_i}\right) \le r_k; \qquad (4.7)$$

$$\pi_k = \frac{\left(\sum_{i=1}^{k}\rho_i\right)^{r_k}\left(1-\sum_{i=1}^{k}\rho_i\right)}{1-\left(\sum_{i=1}^{k}\rho_i\right)^{r_k+1}} + \sum_{j=1}^{m}\delta_{jk}\,\rho_j \frac{1-\left(\sum_{i=1}^{k}\rho_i\right)^{r_k}}{1-\left(\sum_{i=1}^{k}\rho_i\right)^{r_k+1}}\left[1-\frac{r_k\,\gamma_k}{(1+\vartheta^2)\,\gamma_j\,\rho_k}\right]\times$$

$$\times\, r_k \Big/ \left\{(1+\vartheta^2)\,\rho_k\sum_{i=1}^{k-1}\gamma_i\left[1-\left(\frac{\rho_i}{\gamma_i}\right)^{l_i+1}\right] \Big/ \left(1-\frac{\rho_i}{\gamma_i}\right)\right\}, \quad \text{где} \quad \delta_{jk} = \begin{cases} 0, \text{ если } (1+\vartheta^2)\gamma_j\frac{\rho_k}{\gamma_k} \le r_k, \\ 1 \text{ в противном случае,} \end{cases}$$

если хотя бы для одного j, $j=\overline{k+1,m}$, $\left(1+\vartheta^2\right)\gamma_j\frac{\rho_k}{\gamma_k} > r_k$ и $Q_k > r_k$.

Выражения (4.2) – (4.4) и (4.7) дают возможность приближенно рассчитывать вероятности потери заявок в системах с относительными приоритетами при известных значениях ρ_i, γ_i, r_i по каждому входящему потоку. Результаты вычислений по формулам подтверждаются статистическим моделированием. Тем не менее, необходимо еще раз подчеркнуть, что допущение о правомерности замены случайной величины длины очереди ее наиболее вероятным значением, на котором основан их вывод, и, следовательно, методы расчета объемов БН, вносит в результаты дополнительную погрешность. Для системы с параметрами $n = 1$; $m = 2$; $r = 10$; $\rho_1 = \rho_2 = 0.45$ ошибка относительно результатов моделирования достигает 15% в окрестности точки излома при $\gamma = r / (1 + \vartheta^2)\,\rho_1$ и менее 10% – в остальном диапазоне изменения γ.

С увеличением n и m погрешность относительно метода статистиче-

ских испытаний снижается. На этапе проектирования, располагая лишь приблизительными исходными данными, трудно ожидать точное решение задачи. Важно определить области и тенденции устойчивого поведения показателей качества в функции параметров системы, вывести ее в режим, в котором для любого произвольного *k*-го потока соблюдаются условия (4.1) и (4.5), чтобы сохранить назначенные в соответствии с замыслом отношения предпочтения. При любых сочетаниях параметров необходим инструмент, позволяющий выяснить характер взаимного влияния элементов системы и дающий возможность управлять им, т.е. показывающий, какие характеристики системы следует изменить для повышения эффективности ее работы. Именно на такую «косвенную оптимизацию» ориентированы полученные формулы.

4.2.3. ПРИМЕР РАСЧЕТА ШКАЛЫ ПРИОРИТЕТОВ. Оценим параметры существующей схемы сбора сообщений по ОВД. На рис. 4.6 – 4.8 представлены графики поступления телеграмм разных приоритетов в главный центр ЕС ОрВД в течение года. В табл. 4.1 сведены данные о количестве и пропускной способности рабочих мест для редактирования текста сообщений. Результаты статистической обработки экспериментальных данных, изображенных на графиках, иллюстрирует табл. 4.2. Показано, что поток телеграмм от службы аэронавигационной информации (Notice to airmen – NOTAM) аппроксимируется распределением Пуассона. Для рассматриваемых 10 групп (i = 0, … 9) при двух параметрах распределения (m_i, Σ) вычисляем среднее значение частотного интервала $\bar{k} = \dfrac{\sum k_i m_i}{m} = 3{,}63\left[мин^{-1}\right]$ с дисперсией

$$S^2 = \frac{1}{m-1}\left(\sum k_i^2 m_i - m\bar{k}^2\right) = 3{,}79\left[мин^{-1}\right].$$

Согласно [7], критерий Пирсона $P(\chi_t^2 = 20.09 > \chi_0^2 = 3.19) = 0.01$, что с доверительной вероятностью $\beta = 0.9$ соответствует закону Пуассона. В нумерации рис. 4.1 и 4.4 данного раздела индексируем приоритетность входных потоков от единицы до семи в соответствии с категориями срочности сообщений. Интенсивности λ_i потоков, параметры μ_i их обслуживания, последовательность показателей γ_i и создаваемая загрузка ρ_i, полученные по результатам обработки, размещены в четырех верхних строках табл. 4.3. Условия обслуживания, вычисленные согласно (4.1) $L_k = \left(1+\vartheta^2\right)\gamma_k \sum_{i=1}^{k}\left(\dfrac{\rho_i}{\gamma_i}\right) \le r,$ представлены в ее пятой строке.

Для сравнения дисциплин с общим БН и с раздельными секциями по критерию вероятности π_i потери заявки объемы секций выбраны так, чтобы:

- неравенство (4.1) удовлетворялось;
- суммарный объем секций был равен объему общего БН.

Рассчитаны и сведены в таблицу 4.3 минимальные объемы r_i секций для дисциплины с раздельным БН (4.6). Для дисциплины с обобществлением буферной памяти размеры остатка общего БН, свободного от ЗВП, также представлены в табл. 4.2. Для обеих дисциплин вычислены вероятности по-

тери заявок. Обслуживание с общим БН гарантирует монотонное возрастание показателей π_i по мере убывания приоритетности заявок. При разделении секций эта картина не сохраняется, причем наблюдаются достаточно высокие потери заявок среднего уровня приоритетности. Рассмотренная модель, как и другие средства теории массового обслуживания, не дает целенаправленной процедуры улучшения показателей обслуживания. Разработчику приходится

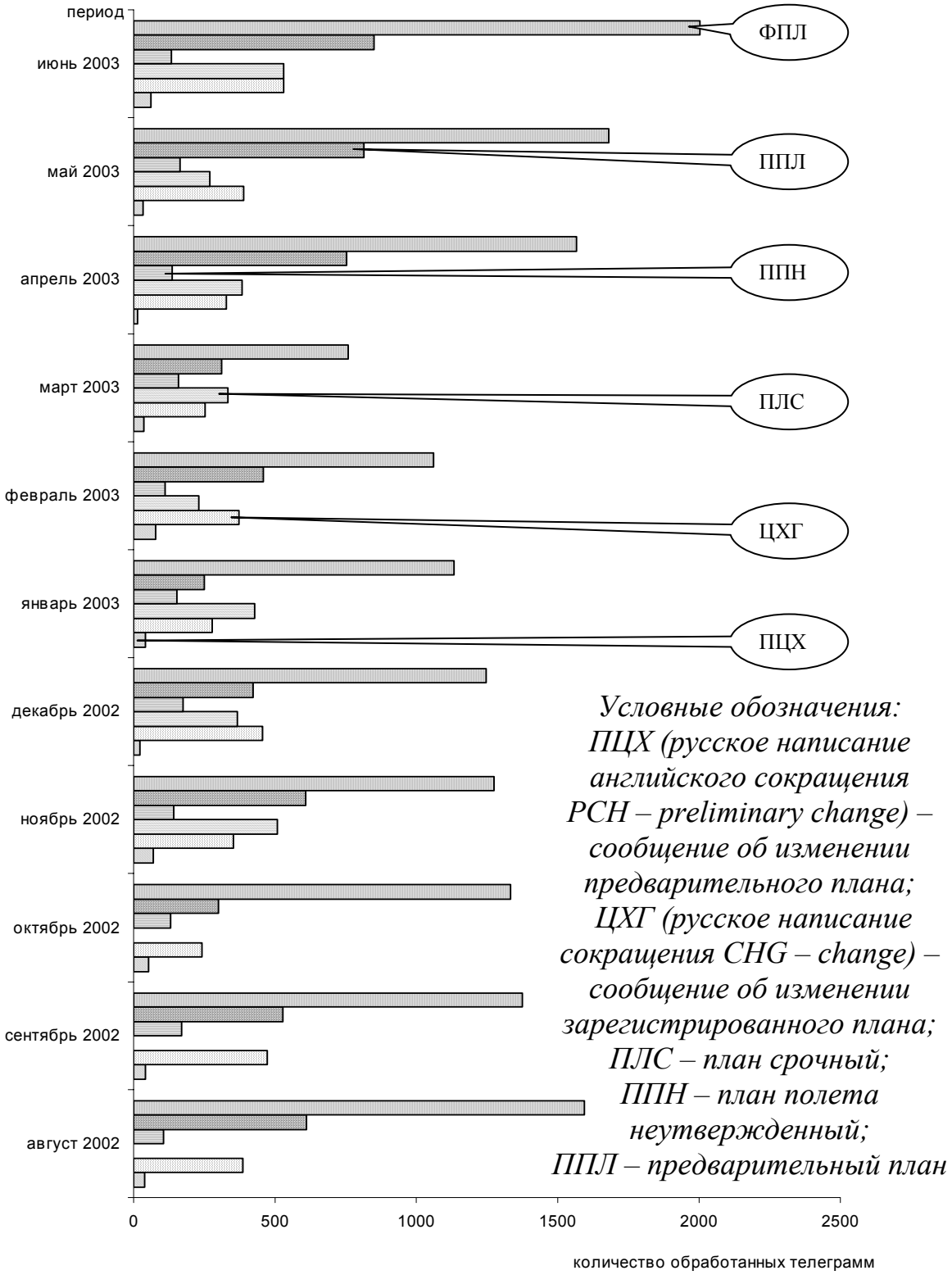


Рис. 4.6. Диаграмма распределения по типам телеграмм, обработанных ПО АС УВД за период с 01.08.2002 г. по 30.06.2003 г.

варьировать параметры СМО, исходя из неформальных соображений, чтобы оценить новую конфигурацию и сопоставить полученные результаты.

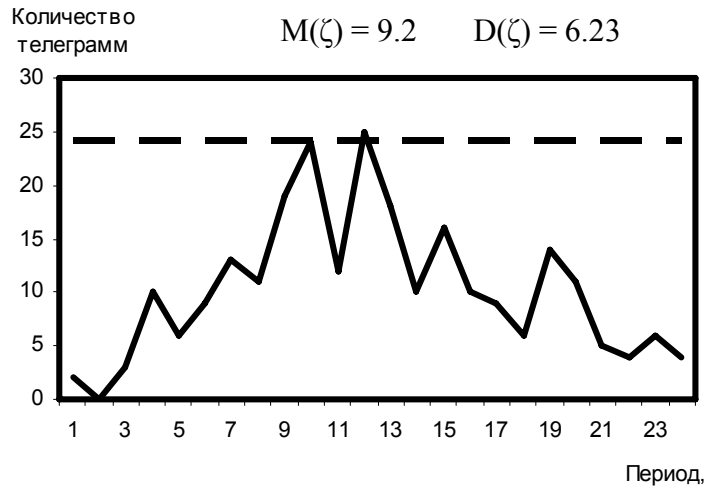


Рис. 4.7. Суммарный график средней почасовой загрузки диспетчера текущего планирования

Условные обозначения:

статистика сбора телеграмм

порог загрузки

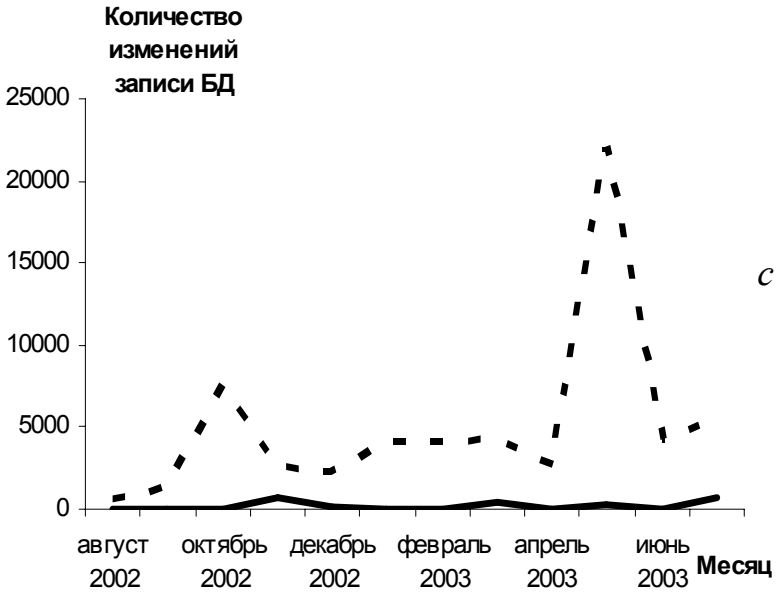


Рис 4.8. График поступления аэронавигационной информации в систему за период с 01.08.2002 г. по 01.07.2003 г.

Условные обозначения:

аэродромы

трассы

Таблица 4.1

Пропускная способность рабочих мест для редактирования текста сообщений

Наименование потока телеграфных сообщений	Количество рабочих мест	Пропускная способность (сообщений в час)
Планирование	4	12
Сообщения NOTAM	1	34
Информация о трассах	4	10
Информация об аэропортах	2	29
Поток аэронавигационных данных	2	10
Информация о ВС	2	27
Поток справочной информации	2	2

Таблица 4.2

Граница k интервала	Частота попаданий, m_i	$P_i = \lambda^i e^{-\lambda}/i! = 3.63^i e^{-3.63}/i!$	$m_i^0 = mP_i$	$m_i - m_i^0$	$(m_i - m_i^0)^2$	$\frac{(m_i - m_i^0)^2}{m_i^0}$
0	3	0.0267	1.92	1.08	1.17	0.6083
1	6	0.0966	6.96	-0.96	0.92	0.1319
2	12	0.1752	12.61	-0.61	0.37	0.0296
3	16	0.2117	15.24	-0.76	0.58	0.0380
4	12	0.1918	13.85	-1.81	3.28	0.2373
5	13	0.1391	10.01	2.99	8.93	0.8915
6	4	0.0840	6.05	-2.05	4.20	0.6941
7	3	0.0435	3.13	-0.13	0.02	0.0056
8	2	0.0197	1.42	0.58	0.34	0.2374
9	1	0.0079	0.57	0.43	0.18	0.3208
Σ	72	—	—	—	—	3.19

Таблица 4.3

Параметры входных потоков	Потоки по категориям срочности (приоритетам обслуживания)						
	СС	ДД	ФФ	ГГ	ЙЙ	КК	ЛЛ
Интенсивность потока, λ_i	0.3	0.5	3	5	1	3	1
Параметр обслуживания, μ_i	36	34	10	29	10	27	12
$\gamma_i = \mu_1 / \mu_i$	1	1.06	3.6	1.24	3.6	1.33	3
Загрузка, ρ_i	0.083	0.014	0.3	0.172	0.1	0.111	0.083
Очередь, L_i	0.017	0.047	0.758	0.605	1.957	0.946	0.768
Общий буферный накопитель объемом $r = 8$ мест для ожидания							
Остаток БН, r_i'	8	7	7	7	6	4	3
Вероятность потери, π_i	$\sim 10^{-7}$	$\sim 10^{-6}$	$\sim 10^{-4}$	$\sim 10^{-3}$	0.012	0.079	0.119
Раздельные секции буферного накопителя, $\Sigma r_i = 8$							
Объем БН, r_i (i-я секция)	1	1	1	1	2	1	1
Вероятность потери, π_i	0.077	0.014	0.41	0.004	0.286	0.099	0.077

4.3. Приоритетное обслуживание на компьютерной сети

4.3.1. Статическое разделение заявок. Под статическим разделением понимается дисциплина распараллеливания, при которой за каждым каналом СМО (или группой каналов) жестко закрепляются заявки одного потока (или ограниченной группы), и заявки других потоков объявляются недоступными в процессе работы. Если таких «закрепленных» заявок нет, то канал (или

группа каналов) простаивает в ожидании даже при перегрузке других каналов системы. Например, вводы диспетчера УВД обрабатываются на его персональном компьютере, а вводы других диспетчеров – на их компьютерах.

4.3.1.1. Модель с раздельными секциями БН, $m = n$. Обобщение модели процесса поступления и обслуживания заявок на произвольное количество n обслуживающих каналов начнем с самой простой модели. Пусть мы имеем вычислительную честь, содержащую n каналов и обслуживающую $m = n$ простейших потоков записей интенсивностью λ_i ($i = \overline{1,m}$) каждый, которым выделены собственные секции БН объемами по r_i мест для ожидания. В случае занятости канала и отсутствия свободных мест в БН заявка i-го потока, приходя в систему, получает отказ в обслуживании и теряется. Возможно, что в каком-либо другом, k-м БН, $k = \overline{1,n}, k \neq i$, есть при этом незанятые места, что есть простаивающие каналы системы. Времена обслуживания распределены экспоненциально с параметрами μ_i. Загрузка i-го канала $\rho_i = \lambda_i / \mu_i < 1$.

Очевидно, что рассматриваемая модель тривиальна и легко распадается на n одноканальных СМО, работающих автономно и независимо друг от друга. Вероятность π_i потери заявки i-го уровня приоритетности равна

$$\pi_i = \frac{\rho_i^{r_i+1}(1-\rho_i)}{1-\rho_i^{r_i+2}}$$

Суммарные нормированные потери по всем n каналам исследуемой модели достигают:

$$\pi_\Sigma = \frac{1}{n}\sum_{i=1}^{n}\pi_i = \frac{1}{n}\sum_{i=1}^{n}\frac{\rho_i^{r_i+1}(1-\rho_i)}{1-\rho_i^{r_i+2}}$$

Рассмотренная простая модель позволяет обнаружить одно характерное свойство. По-видимому, именно элементарность постановки делает его очевидным. Вероятность потери заявки не зависит ни от приоритетности потока, которому она принадлежит, ни от соотношения $\gamma_i = \mu_1/\mu_i$ параметров обслуживания, столь сильно влияющих на эффективность одноканальных систем. Величина критерия определятся лишь размером БН, предоставленного в распоряжение i-го потока, и создаваемой им загрузкой ρ_i. При дальнейших исследованиях более сложных моделей необходимо анализировать, сохраняется ли тенденция снижения эффективности приоритетного обслуживания при переходе от одноканальной к многоканальной СМО, или отмеченное выравнивание вероятностей потерь заявок остается качеством лишь полностью разобщенной системы.

Первым шагом обобществления ресурсов модели является использование файла с общим доступом. Пусть СМО содержит, как и прежде, n однотипных каналов, обслуживающих с относительным приоритетом n пуассоновских входных потоков с интенсивностями λ_i каждый ($i = \overline{1,n}$); времена обслуживания распределены экспоненциально с параметрами μ_i. Прием заявок осуществляется по принципу приоритетного вытеснения из общего БН объемом r мест для ожидания. Объединение буферной памяти создает известную

упорядоченность на входе системы за счет возможного отказа в обслуживании неприоритетным заявкам. Вследствие этого возникает корреляция между потоками, реализующая заданные отношения предшествования. Другим содержательным отличием от предыдущей модели является динамическое распределение ресурса БН. С одной стороны, становится маловероятной ситуация, при которой заявка произвольного *i*-го потока, поступая в систему, застает занятым весь объем *r* общего БН. С другой стороны заметим, что эта малая вероятность реализуется целиком за счет неприоритетных требований, что приводит к увеличению вероятности простоя соответствующих каналов.

4.3.1.2. МОДЕЛЬ С ОБЩИМ БН. Первоначальное рассмотрение ограничим моделью двухприоритетной двухканальной СМО, на которой проследим основные закономерности обслуживания. Специфика функционирования такой системы предоставляет в распоряжение ЗВП один канал и весь БН объемом на *r* мест для ожидания. Вероятность π_1 их потери может быть оценена как

$$\pi_1 = \frac{\rho_1^{r_1+1}(1-\rho_1)}{1-\rho_1^{r_1+2}}$$

Очевидно, что и здесь отсутствует зависимость $\pi = f(\gamma)$, так как первый канал по условиям задачи никогда не бывает занят ЗНП. Процесс обслуживания потока ЗВП определяется лишь его собственными характеристиками. Наиболее вероятную длину L_1 очереди ЗВП, накапливающейся в общем БН за время обслуживания одной из них, вновь оценим по известному приближению $L_1 = (1+\upsilon^2)\lambda T_1 = (1+\upsilon^2)\rho_1$. Этот параметр позволяет рассчитать величину остатка r' БН, которым по правилам приоритетного вытеснения может пользоваться ЗНП, в виде $r' = r - (1+\upsilon^2)\rho_1$. Тогда

$$\pi_2 = \frac{\rho_2^{r+1-(1+\vartheta^2)\rho_1}(1-\rho_2)}{1-\rho_2^{r+2-(1+\vartheta^2)\rho_1}}, \text{ если } r \geq (1+\upsilon^2)\rho_1.$$

В случае невыполнения условия $r \geq (1+\upsilon^2)\rho_1$ режим обслуживания ЗНП ухудшается. По существу, неприоритетный поток полностью лишается возможности образования очереди ожидания, а принадлежащие ему заявки могут занимать выделенный им канал лишь в моменты его простоя. Преобразуя известную формулу для одноканальной СМО, не имеющей БН ($r = 0$), можно записать $\pi_2 = \rho_2 / (1+\rho_2)$. Заметим, что подобная ситуация весьма маловероятна. В предположении пуассоновского потока на входе и экспоненциального времени обслуживания ($\upsilon = 1$) даже при $\rho \to 1$, наиболее вероятная длина очереди ЗВП не достигает двух заявок, то есть $L_1 < 2$. На рис. 4.9 представлены графики зависимостей $\pi_i = f(\rho)$ для обсуждавшейся модели вычислительной сети.

Нетрудно видеть, что для равноценных по аппаратным затратам систем с параметрами $n = 2$; $r_1 = r_2 = r/2 = 5$ использование общего БН при жестком (статическом) разделении потоков по обслуживающим каналам становится предпочтительнее по критерию минимизации вероятностей потери заявки. Этот эффект можно объяснить динамическим использованием буферной па-

мяти, при котором относительный сдвиг по времени «сгущений» и «разрежений» нерегулярных входных потоков несколько компенсируется подвижными границами зон общего БН. Сказывается, хотя не в полной мере, и восстановление отношений предпочтительности на входе за счет возможности вытеснения из БН неприоритетных заявок. Обобщая модель на случай n каналов и $m = n$ входящих потоков, определим наиболее вероятную длину $L_{k\text{-}1}$ очереди всех заявок, расположенных в шкале приоритетов выше произвольного k-го потока, в виде $L_{k-1} = (1+\vartheta^2)\sum_{i=1}^{k-1}\rho_i$.

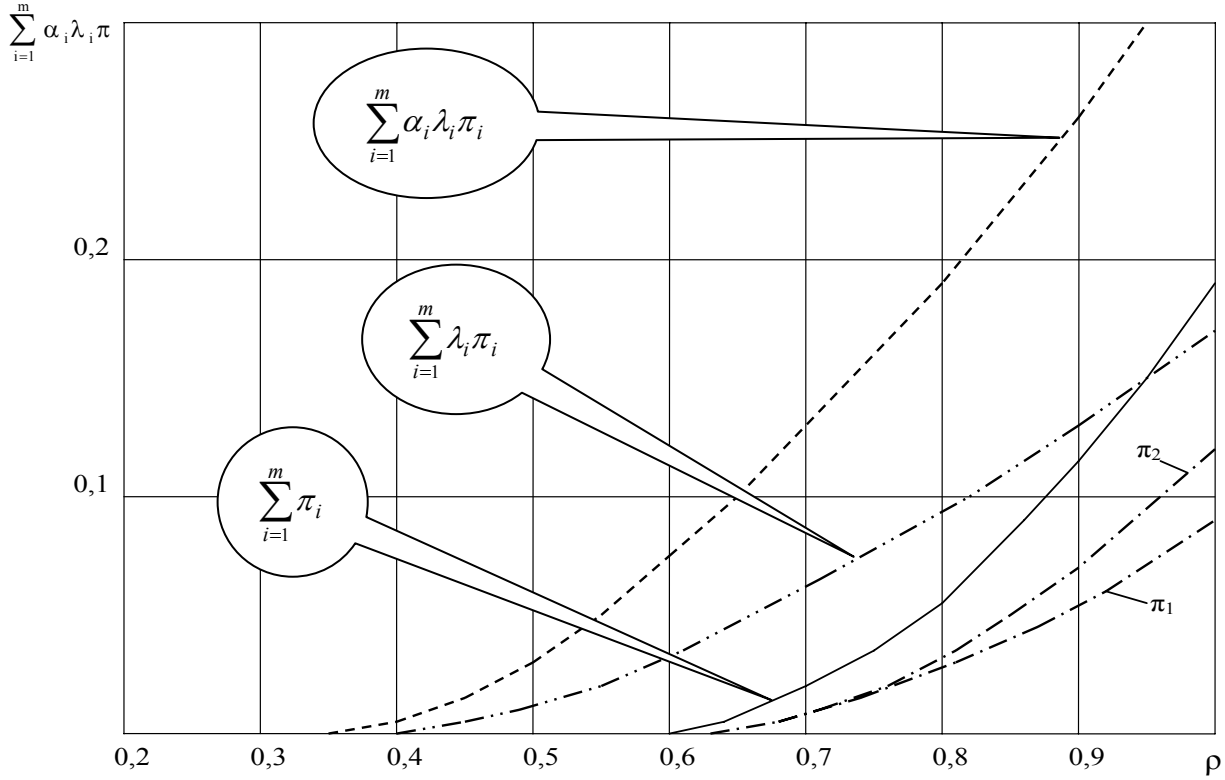


Рис. 4.9. Графики зависимостей $\pi_i = f(\rho)$ для статического разделения

Тогда приближенная формула для расчета вероятности потери заявки k-го типа запишется как:

$$\pi_k = \begin{cases} \dfrac{\rho_k^{\,r+1-(1+\vartheta^2)\sum_{i=1}^{k-1}\rho_i}(1-\rho_k)}{1-\rho_k^{\,r+2-(1+\vartheta^2)\sum_{i=1}^{k-1}\rho_i}}, & \text{если } r \ge (1+\vartheta^2)\sum_{i=1}^{k-1}\rho_i. \\ \dfrac{\rho_k}{1+\rho_k} \text{ в противном случае.} \end{cases}$$

Приведенное выражение не учитывают исследованное в предыдущей главе влияние корреляции между заявками. Считается, что входные потоки независимы. Однако и в случае, если обслуживание заявки i-го типа разрешается при занятости $(i - 1)$-го канала лишь с вероятностью Q_i, вид обеих формул можно оставить прежним, но загрузку ρ_i рассчитывать как $\rho_i = \lambda_i / Q_i\mu_i$. Очевидно, что вероятность потери заявки при этом возрастает.

Дальнейшая детализация постановки для задачи исследования статической дисциплины разделения входных потоков заявок по каналам СМО приводит к следующим моделям обслуживания.

4.3.1.3. Вариации статической дисциплины. Пусть имеем n-канальную систему, которая обслуживает $m < n$ пуассоновских входных потоков заявок с интенсивностями λ_i соответственно ($i = \overline{1,m}$); времена обслуживания распределены экспоненциально с параметрами μ_i. Прием заявок осуществляется по принципу записи в раздельные зоны БН с объемами r_i каждая, причем число таких зон равно количеству m входных потоков. При организации работы подобной системы n процессоров будут разделены на m групп, каждая из которых получит в свое распоряжение один из входных потоков с приданным ему собственным БН, и n_i каналов, $i = \overline{1,m}$, $\sum_{i=1}^{m} n_i = n$. Будем считать, что корреляция между заявками либо отсутствует, либо учтена в параметре загрузки ρ_i. Исследуемая модель представляется композицией из m многоканальных СМО, на каждую из которых поступает одномерный i-й поток заявок, причем на входе его воспринимает БН объемом r_i мест для ожидания. Вероятность потери заявки произвольно выбранного i-го потока оценивается формулой, в которую подставляются конкретные значения r_i и n_i. Уровень приоритетности не влияет на величину π_i. Включение в СМО большего, чем m, количества буферных зон бессмысленно, так как фактически это будет означать простое наращивание одного или нескольких (всех) БН.

Более реален случай, при котором число d разделенных приемных зон меньше, чем количество m входящих потоков, т. е. $d < m$. При этом некоторые группы из n_i каналов будут обобществлять один физический объем памяти, в котором в соответствии с наперед заданной приоритетной шкалой осуществляется размещение заявок нескольких потоков с вытеснением менее приоритетных. Правила приема создают известную упорядоченность на входе, ставя в благоприятные условия заявки высоких приоритетов. Пусть в СМО поступают d групп входных потоков. Каждая группа с индексом i ($i = \overline{1,d}$) насчитывает S_i пуассоновских потоков с интенсивностями λ_j ($j = \overline{1,S}$), пронумерованных в соответствии со шкалой приоритетов, сумма $\sum_{i=1}^{d} S_i = m$. Заявки j-го типа, принадлежащие i-й группе входных потоков, поступают на обслуживание S_i разными группами каналов и, в случае занятости j-й группы каналов, направляются в очередь для ожидания в i-й обобществленный БН. Для произвольно выбранного j-го потока эффективный остаток БН, свободный от заявок более высоких приоритетов, может быть приближенно оценен как $]r'[= r_i - (1+\vartheta^2)\sum_{l=1}^{j-1} \rho_l$. Тогда для расчета вероятности потери заявки j-го потока при положительном ближайшем большем целом вычисленной величины r' будет справедлива формула (3.16), в которую в качестве объема БН следует подставлять $]r'[$, а в противном случае приравнивать r нулю. Количество n каналов СМО в формуле (3.16) определяется исходя из того, сколько их на-

значено на обслуживание заявок *j*-го типа по условиям задачи.

Рассмотрим наиболее распространенную реализацию статического распараллеливания. Пусть система обслуживает *m* приоритетных потоков на *n* каналах, причем $m > n$. Количество разделенных секций БН равно числу *m* входных потоков. В это случае *m* потоков заявок разделяются на *n* групп, каждая из которых назначается на отдельный канал. Внутри *i*-й группы ($i = \overline{1,n}$) осуществляется приоритетной обслуживание l_i потоков ($l_i < m$, $\sum_{i=1}^{n} l_i = m$). Вероятность потери заявки в такой системе оценивается с помощью выражения (4.7) для одноканальной СМО, работающей на l_i разделенных БН. Учет корреляции между заявками производится путем вычисления загрузки системы ρ_k анализируемым *k*-м потоком по формуле $\rho_k = \lambda_k / Q_k\mu_k$. Наконец, если количество секций БН равно количеству каналов, т. е. каждый из них обслуживает l_i потоков с приоритетным приемом в *i*-й БН, то для определения вероятности потери заявки следует пользоваться выражениями (4.2) – (4.4).

Обобщим модель статического распараллеливания *m* потоков на $n = m$ каналов с приоритетным приемом заявок в общий БН для произвольного соотношения $m \neq n$. Пусть $n < m$, что соответствует наиболее реальной ситуации. Ее можно представить суперпозицией *n* обслуживающих каналов, каждый из которых обрабатывает заявки *j*-й группы пуассоновских потоков, ($j = \overline{1,n}$), накапливаемых в динамически изменяющейся зоне общего БН, свободной от ЗВП. Размер r_j этой зоны можно приближенно рассчитать как

$$r_j = r - (1+\vartheta^2) \times \left(\sum_{s=k_j+1}^{l_j} \delta_{ks} \cdot \gamma_{l_j} \cdot \sum_{i=1}^{s} \frac{\rho_i}{\gamma_i} + \sum_{p=l_j+1}^{m} \delta_{kp} \cdot \gamma_m \cdot \sum_{s=1}^{p-1} \sum_{i=1}^{l_s} \frac{\rho_i}{\gamma_i} \right),$$

где k_j – индекс произвольного входящего потока, включенного в *j*-ю группу, $k_j = \overline{1,l_j}; \sum_{j=1}^{m} l_j = m;$ δ_{kp} – переключатель, аналогичный символу Кронекера и определенный выше, принимающий единичное значение, позволяющее учитывать переполнение общего БН заявками высоких приоритетов в сеансах обслуживания заявок низких приоритетов, и равный нулю при отсутствии событий такого переполнения. Далее, $\gamma_i = \mu_1 / \mu_i$ – соотношение параметров обслуживания заявок высшего и *i*-го приоритетов; двойная сумма до $p - 1$ учитывает заполнение БН заявками потоков, принадлежащих группам более высокой, чем *j*, приоритетности, в сеансах обслуживания заявок потоков, принадлежащих группам низшей приоритетности; сумма до *s* – то же относительно *k*-го потока заявок внутри *j*-й группы, в которой производится обслуживание с относительными приоритетами l_j самостоятельных потоков.

Рассматриваемая модель вписывается в рамки ограничений, оговоренных перед выводом приближенной формулы (4.7). При ее использовании вместо *r* следует подставлять r_j. При $(1+\vartheta^2) \times \sum_{p=l_j+1}^{m} \delta_{kp} \cdot \gamma_m \cdot \sum_{s=i}^{p-1} \sum_{i=1}^{l_s} \frac{\rho_i}{\gamma_i} > r_j$ потери по

любому k_j-му входящему потоку, принадлежащему j-й группе, рассчитываются как для системы с $r = 0$. Это означает, что вероятность обслуживания заявки произвольного k-го потока, независимо от уровня его приоритетности, есть вероятность P_{Ok_j} застать j-й обслуживающий канал свободным, а вероятность потери заявки есть ее дополнение до единицы:

$$\pi_{k_j} = 1 - P_{Ok_j} = \frac{\sum_{s=1}^{j-1}\sum_{i=1}^{l_s}\rho_i + \sum_{i=1}^{k_j}\rho_i}{1 + \sum_{s=1}^{j-1}\sum_{i=1}^{l_s}\rho_i + \sum_{i=1}^{k_j}\rho_i}.$$

Наконец, при $n > m$ модель представляет собой совокупность нескольких многоканальных СМО, обслуживающих каждая один k-й поток, $k = \overline{1,m}$, использующих каждая собственную k-ю динамическую секцию общего БН, объем которой вычисляется как

$$)r_k (= r - (1 + \vartheta^2) \cdot \sum_{j=k+1}^{m} \delta_{jk} \cdot \gamma_j \cdot \sum_{i=1}^{k-1} \frac{\rho_i}{\gamma_i}.$$

Ближайшее большее неотрицательное целое вычисленной величины подставляется в выражения (4.2) – (4.4) для оценки вероятности потери заявки по k-му потоку в такой системе.

4.3.2. Динамическое разделение заявок. В отличие от статической, динамическая дисциплина разделения записей допускает взятие на обслуживание любым освободившимся каналом СМО заявки любого входного потока, руководствуясь при выборе лишь шкалой приоритетности, а не жестким назначением типа потока. Основные закономерности реализации отношений предпочтения между заявками в такой модели проследим на двухприоритетной двухканальной СМО ($n = m = 2$). Распределение обоих потоков пуассоновское, интенсивности поступления заявок λ_1 и λ_2 соответственно, обслуживание экспоненциальное, его параметры μ_1 и μ_2. Заявки принимаются в общий БН объемом r мест для ожидания по принципу приоритетного вытеснения. Загрузка ρ_Σ системы обоим потоками не превосходит единицы, обслуживание выполняется с относительным приоритетом.

4.3.2.1. Модель с общим БН. Основываясь на допущении о правомерности замены случайной величины длины очереди, образующейся в БН, ее наиболее вероятным значением $L_1 = (1 + v^2)\rho_1\gamma$, оценим вероятности π_1 и π_2 потери ЗВП и ЗНП в такой системе. В момент ее освобождения от всех ЗВП, на обслуживание, среднее время которого $T_2 = 1/\mu_2$, назначается ЗНП. Если за время T_2 в БН образуется очередь ЗВП, наиболее вероятная длина которой $L_1 \le r$, т.е. выполняется условие (4.1), то вероятность потери ЗВП практически не зависит от второго потока и может быть подсчитана по формуле для $n = 2$

$$\pi_1 = \frac{2\rho_1^{r+2}(1-\rho_1)}{1+\rho_1-2\rho_1^{r+3}}, \text{ если } (1 + \vartheta^2)\rho_1\gamma \le r.$$

При невыполнении условия (4.1) в сеансе обслуживания одной ЗНП БН

переполняется очередью L_1, и часть ее, равная $L_1 - r$, теряется, причем доля потерь ЗВП, как и в случае одноканальной СМО, составляет

$$\xi_1 = 1 - \frac{r}{(1+\vartheta^2)\rho_1\gamma}, \text{ если } (1+\vartheta^2)\rho_1\gamma \le r.$$

Вероятность события, при котором теряются ЗВП, как и в модели с одним каналом, пропорциональна ρ_2 и приобретающей более громоздкий вид вероятности P_2 обслуживания ЗНП в условиях отсутствия для них свободного места в БН ($r = 0$)

$$P_2 = \frac{1+\rho_\Sigma - 2\cdot\rho_\Sigma^2}{1+\rho_\Sigma - 2\cdot\rho_\Sigma^3}, \text{ если } (1+\vartheta^2)\rho_1\gamma > r.$$

Составная формула для оценки вероятности π_1 потери ЗНП приобретает вид:

$$\pi_1 = \frac{2\rho_1^{r+2}(1-\rho_1)}{1+\rho_1-2\rho_1^{r+3}} + \delta\cdot\rho_2\cdot\frac{1+\rho_\Sigma-2\cdot\rho_\Sigma^2}{1+\rho_\Sigma-2\cdot\rho_\Sigma^2}\cdot\left[1-\frac{r}{(1+\vartheta^2)\rho_1\gamma}\right], \text{ где } \delta = \begin{cases} 0, \text{ если } (1+\vartheta^2)\rho_1\gamma \le r, \\ 1 \text{ в противном случае.} \end{cases}$$

Учитывая, что при выполнении условия (4.1), т.е. слева от точки излома $\gamma_{кр} = r / (1 + \vartheta^2)\rho_1$, зона общего БН, свободная от ЗВП, возрастает до величины $r - (1 + \vartheta^2)\rho_1\gamma$, можно записать для оценки вероятности π_2 потери ЗНП

$$\pi_2 = \frac{2\rho_\Sigma^{2+(1-\delta)\left[r-(1+\vartheta^2)\rho_1\gamma\right]}(1-\rho_\Sigma)}{1+\rho_\Sigma-2\cdot\rho_\Sigma^{3+(1-\delta)\left[r-(1+\vartheta^2)\rho_1\gamma\right]}}$$

Отметим, что справа от точки излома кривой вероятности π_2 потери ЗНП в зависимости от γ абсолютные значения π_2 снижаются относительно обслуживания одним каналом, так как

$$\pi_2 = \frac{2\rho_\Sigma^2(1-\rho_\Sigma)}{1+\rho_\Sigma-2\cdot\rho_\Sigma^3} < \frac{\rho_\Sigma}{1+\rho_\Sigma}, \quad \rho_\Sigma \ne 0.$$

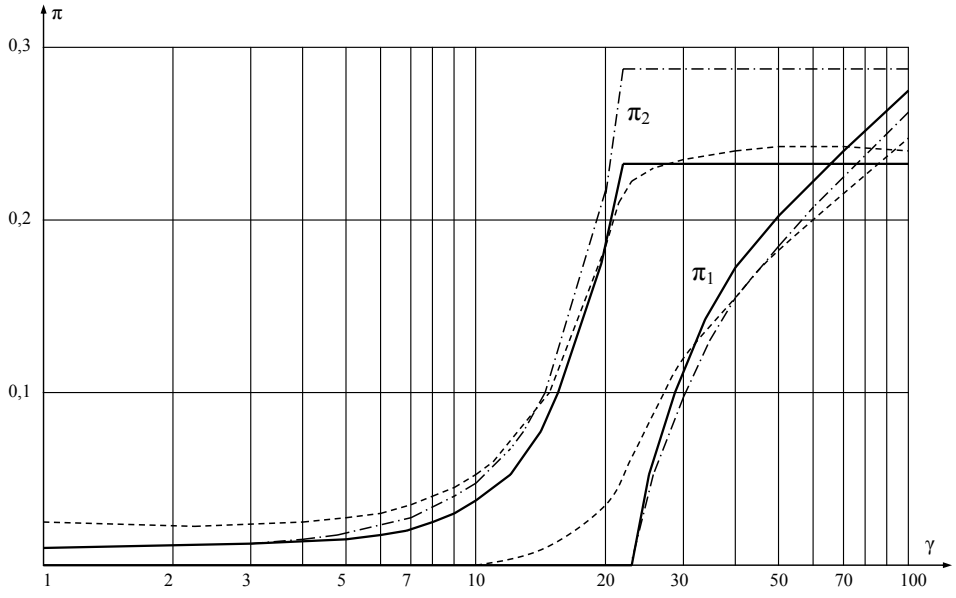


Рис. 4.10. Графики зависимостей $\pi_i = f(\gamma)$ для дисциплины динамического разделения с приемом заявок в общий БН

В параграфе 3.3.3 было показано, что относительный выигрыш β по критерию π при r=0 и $\rho \to 1$ от перехода к двухканальной СМО достигает 20% (β = 0,2). Для сопоставления результатов слева от точки излома, рассмотрим графики $\pi_i = f(\gamma)$, представленные на рис. 4.10. Параметры исследованных систем: n = {1, 2}; $\rho_1 = \rho_2$ = 0,45; ϑ = 0; r = 10. Приведены результаты как расчета по приближенным формулам, так и эксперимента с помощью статистического моделирования на ЭВМ. В срав-

нении с обслуживанием одним прибором, динамическое распределение дает относительное возрастание вероятности π_1 потери ЗВП и снижение вероятности π_2 потери ЗНП. Слева от точки излома снижаются потери по обоим потокам, так как в двухканальной системе на одно место для ожидания больше, но для ЗНП это приобретение существеннее, так как они обладают абсолютно меньшим объемом БН. Справа от точки излома потери ЗНП снижаются на 20%, следовательно, возрастает вероятность события, приводящего к потере ξ ЗВП, и происходит рост π_1 относительно одноканального варианта.

Таким образом, как и при статическом разделении записей, снижается эффективность приоритетного обслуживания, т.е. происходит выравнивание вероятностей потерь заявок, принадлежащих потокам различной приоритетности. Более того, существует некоторое критическое значение $\gamma_{кр}$, равное для анализировавшихся СМО $\gamma_{кр} \approx 65$, начиная с которого значение потери ЗВП при обслуживании двумя каналами превосходят потери ЗНП. Заметим, что критическое значение $\gamma_{кр}$, при котором происходит пересечение кривых π_1 и π_2 на графике, построенном по результатам статистического моделирования, лежит несколько правее, чем это предсказано расчетными формулами: $\gamma_{кр} \approx 85$, а также обратим внимание на то обстоятельство, что подобные соотношения длительностей времени обслуживания практически не встречаются. Важна обнаруженная тенденция, а не ее частное проявление.

Для распространения полученных результатов на случай m приоритетных потоков, обслуживаемых n каналами с относительным приоритетами и приоритетной записью заявок в общий БН, рассмотрим три фазы процесса. Пусть различаются соотношения объема r файла и значений длины L_k, $L_{k\text{-}1}$ накопленных в нем очередей. Сначала будем считать, что все $Q_i = 1$, т.е. явление корреляции отсутствует, и что назначение приоритетов произведено в соответствии с неубыванием величин $\gamma_i = \mu_1 / \mu_i$. В дальнейшем оба ограничения снимаются. Первая возможная фаза обусловлена неравенством

$$L_k = (1 + \vartheta^2) \cdot \sum_{i=1}^{k} T_m \lambda_i = (1 + \vartheta^2) \cdot \gamma_m \cdot \sum_{i=1}^{k} \left(\frac{\rho_i}{\gamma_i} \right) \le r, \quad i = \overline{1,k},$$

утверждающим, что за время T_m обслуживания любой заявки, принадлежащей потоку, расположенному в приоритетной шкале не выше k-го включительно, т.е. требующей более, чем другие, продолжительной обработки, вероятная длина L_k очереди заявок всех высших приоритетных потоков, включая k-й, не превосходят объема r общего БН. Согласно ранее введенному допущению, вероятность π_k потери заявки произвольно выбранного k-го приоритета определяются при этом суммарной загрузкой, создаваемой первыми k потоками, и остатком общего БН, свободным от заявок $k - i$ высших приоритетов. Для этого случая, помеченного верхним индексом первой фазы (I) при π, справедлива приближенная формула

$$\pi_k^{(I)} = \frac{\dfrac{n^{n-1}}{(n-1)!}\left(\sum_{i=1}^{k}\rho_i\right)^{r+n-(1+\vartheta^2)\gamma_m\cdot\sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right)} \times \left(1-\sum_{i=1}^{k}\rho_i\right)}{\sum_{h=0}^{n}\dfrac{n^{h-1}(n-h)}{h!}\left(\sum_{i=1}^{k}\rho_i\right)^{h} - \dfrac{n^{n-1}}{(n-1)!}\left(\sum_{i=1}^{k}\rho_i\right)^{r+n-(1+\vartheta^2)\gamma_m\cdot\sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right)+1}}.$$

Вторая фаза функционирования модели ограничена двойным неравенством

$$(1+\vartheta^2)\cdot\gamma_m\cdot\sum_{i=1}^{k}\left(\frac{\rho_i}{\gamma_i}\right) > r \ge (1+\vartheta^2)\cdot\gamma_m\cdot\sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right),$$

утверждающим, что в единичном сеансе обслуживания заявки *m*-го типа, самом продолжительном по времени исполнения, наиболее вероятная величина L_k длины образующейся в системе общей очереди заявок высших приоритетов, включая *k*-й, превосходит количество *r* мест для ожидания. Однако суммарная очередь L_{k-1} заявок с индексами $i < k$, т.е. поставленных в шкале приоритетов выше анализируемого *k*-го уровня, может быть размещена в общем БН. В этом случае часть поступающих заявок *k*-го типа будет принята в систему, а другая часть – получит отказ в обслуживании и окажется потерянной, причем доля ξ потерь, как и в одноканальной СМО, составит

$$\xi_{km} = 1 - \frac{\gamma_k\left[r-(1+\vartheta^2)\gamma_m\cdot\sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right)\right]}{(1+\vartheta^2)\cdot\rho_k\cdot\gamma_m}.$$

Вероятность P_j обслуживания заявок *j*-го потока, $j=\overline{k+1,m}$, при условии их приоритетного вытеснения из общего БН более приоритетными заявками определяется выражением

$$P_j = \frac{\sum_{h=0}^{n}\dfrac{n^{h-1}(n-h)}{h!}\left(\sum_{i=1}^{j}\rho_i\right)^{h} - \dfrac{n^{n-1}}{(n-1)!}\left(\sum_{i=1}^{j}\rho_i\right)^{r+n-(1+\vartheta^2)\gamma_m\cdot\sum_{i-1}^{j-1}\left(\frac{\rho_i}{\gamma_i}\right)}}{\sum_{h=0}^{n}\dfrac{n^{h-1}(n-h)}{h!}\left(\sum_{i=1}^{j}\rho_i\right)^{h} - \dfrac{n^{n-1}}{(n-1)!}\left(\sum_{i=1}^{j}\rho_i\right)^{r+n-(1+\vartheta^2)\gamma_m\cdot\sum_{i-1}^{j-1}\left(\frac{\rho_i}{\gamma_i}\right)+1}}.$$

Вероятность π_k потери заявки при ограничениях, накладываемых второй фазой функционирования модели (помечена верхним индексом II), приближенно оценивается выражением

$$\pi_k^{(II)} = \pi_k^{(I)} + \sum_{j=k+1}^{m}\delta_{jk}\cdot\rho_j\cdot P_j\cdot\xi_{kj}, \quad \text{где} \quad \delta = \begin{cases} 0, \text{если } L_k \le r, \text{т.е.} (1+\theta^2)\cdot\gamma_m\cdot\sum_{i=1}^{k}\left(\frac{\rho_i}{\gamma_i}\right), \\ 1 \text{ в противном случае.} \end{cases}$$

Третья возможная фаза функционирования характеризуется условием $(1+\vartheta^2)\cdot\gamma_j\cdot\sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right) > r$, значащим, что за сеанс обслуживания любой заявки *j*-го приоритета ($j=\overline{k+1,m}$) в общем БН образуется очередь ЗВП с индексами потоков $i=\overline{1,k-1}$, занимающая все *r* мест для ожидания, и все заявки *k*-го типа, принятые ранее, вытесняются, а приходящие вновь не записываются. В этом случае вероятность $\pi_k = 1 - P_k$ подсчитываются, исходя из суммарной загруз-

ки системы заявками k первых потоков и при отсутствии БН ($r = 0$):

$$\pi_k^{(III)} = \frac{\dfrac{n^{n-1}}{(n-1)!}\left(\sum_{i=1}^{j}\rho_i\right)^{n-(1+\vartheta^2)\gamma_m\cdot\sum_{i=1}^{j-1}\left(\frac{\rho_i}{\gamma_i}\right)} \times\left(1-\sum_{i=1}^{j}\rho_i\right)}{\sum_{h=0}^{n}\dfrac{n^{h-1}(n-h)}{h!}\left(\sum_{i=1}^{j}\rho_i\right)^{h} - \dfrac{n^{n-1}}{(n-1)!}\left(\sum_{i=1}^{j}\rho_i\right)^{n-(1+\vartheta^2)\gamma_m\cdot\sum_{i=1}^{j-1}\left(\frac{\rho_i}{\gamma_i}\right)+1}}.$$

Для учета корреляции между заявками следует в выражения для P_j и π_j подставлять значения ρ и n, рассчитанные по формулам параграфа 3.3.3. Для перехода к определению необходимого объема r БН общего доступа следует, если это возможно, пользоваться верхней частью составного выражения, отображающей первую фазу функционирования модели, обеспечивающую самые благоприятные условия для размещения заявок всех потоков. В противном случае (когда соотношения параметров системы ρ, γ и заданная вероятность π потери заявки не позволяют обеспечить приемлемые размеры БН, и r превосходит ограничения по памяти) следует либо снизить требования к π, либо отдавать себе отчет в последствиях вывода системы в режим эпизодических переполнений буфера с потерей ξ записей при каждом таком событии.

4.3.2.2. Модель с раздельными секциями БН. Рассмотрим еще одну модель СМО, реализующей динамическую дисциплину распараллеливания. Обслуживание производится с относительным приоритетом и приемом записей в раздельные секции БН. Ограничимся моделью, в которой $n = m = 2$; интенсивности входящих пуассоновских потоков λ_1, λ_2; параметры экспоненциального обслуживания μ_1 и μ_2, $\gamma = \mu_1 / \mu_2$; $\rho_i = \lambda_i / \mu_i$, $i = 1, 2$; $\rho_\Sigma = \rho_1 + \rho_2 < 1$; объемы секций БН – r_1, r_2. Условие обязательного излома кривой $\pi_1 = f(\gamma)$ определяется соотношением параметров модели, при котором наиболее вероятная длина L_1 очереди ЗВП, образующейся в первой секции разделенного БН, не превосходят ее объема r_1. Слева от точки излома вероятность π_1 потери ЗВП не зависит от характеристик обслуживания второго потока и приближенно выражается через ее собственные параметры как

$$\pi_1 = \frac{2\cdot\rho_1^{r_1+2}(1-\rho_1)}{1+\rho_1-2\cdot\rho_1^{r+3}}, \quad \text{если } (1+\vartheta^2)\rho_1\gamma \le r_1.$$

В распоряжение ЗНП также предоставлена отдельная секция БН с фиксированным объемом r_2 и, следовательно:

$$\pi_2 = \frac{2\cdot\rho_1^{r_2+2}(1-\rho_\Sigma)}{1+\rho_\Sigma-2\cdot\rho_\Sigma^{r_2+3}}, \quad \text{если } L_2 \le r_2,$$

т.е. если наиболее вероятная длина L_2 очереди ЗНП, образующийся во второй зоне в процессе полного освобождения СМО от всех ЗВП, не превосходит r_2. Характеристики этого процесса аналогичны рассмотренным выше при исследовании одноканальной модели. Очередь L_2 ЗНП, накапливающаяся за время полного освобождения системы от L_1 ЗВП с учетом убывающих приращений $\{\Delta L^i\}$, составляет

$$L_2 = \frac{(1+\vartheta^2)(1-\rho_1^{h+1})r_1\rho_2}{\gamma(1-\rho_1)}.$$

Доли ξ_1 и ξ_2 потерь заявок по обоим потокам, возникающих при переполнении принадлежащих им секций БН, равны

$$\xi_1 = 1 - \frac{r_1}{(1+\vartheta^2)\rho_1\gamma}, \quad \text{если } (1+\vartheta^2)\rho_1\gamma > r_1; \quad \xi_2 = 1 - \frac{r_2}{r_1}\cdot\frac{\gamma\cdot(1-\rho_1)}{(1+\vartheta^2)(1-\rho_1^{h+1})\rho_2}, \quad \text{если } L_2 > r_2.$$

Вероятность P_2 обслуживания двух ЗНП одновременно, в процессе которого переполняется первая секция БН, определяется как

$$P_2 = \frac{1+\rho_\Sigma - 2\rho_\Sigma^{r_2+2}}{1+\rho_\Sigma - 2\rho_\Sigma^{r_2+3}}, \quad \text{если } \frac{(1+\vartheta^2)(1-\rho_1^{h+1})r_1\rho_2}{\gamma\cdot(1-\rho_1)} \le r_2,$$

$$P_2 = \frac{r_2}{r_1}\cdot\frac{\gamma\cdot(1-\rho_1)}{(1+\vartheta^2)(1-\rho_1^{h+1})\rho_2}\cdot\frac{1+\rho_\Sigma - 2\rho_\Sigma^{r_2+2}}{1+\rho_\Sigma - 2\rho_\Sigma^{r_2+3}} \text{ в противном случае.}$$

Приближенная формула для вычисления π_1 приобретает вид:

$$\pi_1 = \begin{cases} \dfrac{2\cdot\rho_1^{r_1+2}(1-\rho_1)}{1+\rho_1-2\cdot\rho_1^{r+3}}, \quad \text{если } (1+\vartheta^2)\rho_1\gamma \le r_1; \\ \dfrac{2\cdot\rho_1^{r_1+2}(1-\rho_1)}{1+\rho_1-2\cdot\rho_1^{r+3}} + \rho_2\cdot\dfrac{1+\rho_\Sigma-2\rho_\Sigma^{r_2+2}}{1+\rho_\Sigma-2\rho_\Sigma^{r_2+3}}\cdot\left[1-\dfrac{r_1}{(1+\vartheta^2)\rho_1\gamma}\right], \\ \text{если } (1+\vartheta^2)\rho_1\gamma > r_1 \text{ и } \dfrac{(1+\vartheta^2)(1-\rho_1^{h+1})r_1\rho_2}{\gamma\cdot(1-\rho_1)} \le r_2, \\ \dfrac{2\cdot\rho_1^{r_1+2}(1-\rho_1)}{1+\rho_1-2\cdot\rho_1^{r+3}} + \dfrac{r_2}{r_1}\cdot\dfrac{\gamma\cdot(1-\rho_1)}{(1+\vartheta^2)(1-\rho_1^{h+1})\rho_2}\cdot\left[1-\dfrac{r_1}{(1+\vartheta^2)\rho_1\gamma}\right]\cdot\dfrac{1+\rho_\Sigma-2\rho_\Sigma^{r_2+2}}{1+\rho_\Sigma-2\rho_\Sigma^{r_2+3}}, \\ \text{если } (1+\theta^2)\rho_1\gamma > r_1 \text{ и } \dfrac{(1+\vartheta^2)(1-\rho_1^{h+1})r_1\rho_2}{\gamma\cdot(1-\rho_1)} > r_2. \end{cases}$$

Вероятности π_2 потери ЗНП соответствует кривая с одним изломом

$$\pi_2 = \frac{2\cdot\rho_1^{r_2+2}(1-\rho_\Sigma)}{1+\rho_\Sigma-2\cdot\rho_\Sigma^{r_2+3}} + \delta\cdot\left[1-\frac{r_2}{r_1}\cdot\frac{\gamma\cdot(1-\rho_1)}{(1+\vartheta^2)(1-\rho_1^{h+1})\rho_2}\right]\times\frac{1+\rho_\Sigma-2\rho_\Sigma^{r_2+2}}{1+\rho_\Sigma-2\rho_\Sigma^{r_2+3}},$$

где $\delta = \begin{cases} 0, \text{ если } \dfrac{(1+\vartheta^2)(1-\rho_1^{h+1})q_1\rho_2}{\gamma(1-\rho_1)} \le r_2, \\ 1 \text{ в противном случае.} \end{cases}$ $\quad q_1 = \begin{cases} (1+\vartheta^2)\rho_1\gamma, \text{ если } (1+\vartheta^2)\rho_1\gamma \le r_1, \\ r_1 \text{ в противном случае.} \end{cases}$

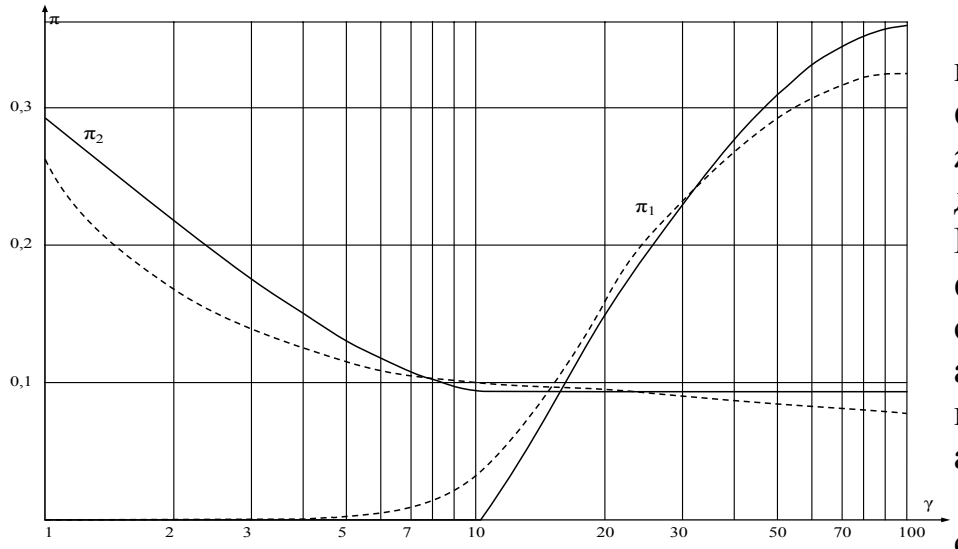


Рис. 4.11. Графики $\pi_i = f(\gamma)$ для дисциплины динамического разделения с приемом заявок в раздельные секции БН

На рис. 4.11 представлены графики зависимостей $\pi_i = f(\gamma)$ для исследованной модели. Рассмотрение графиков позволяет сделать выводы, аналогичные тем, к которым привел анализ дисциплины распараллеливания с общим БН. Слева от обязательных

точек излома, определяемых соотношениями $L_1 \le r_1$, $L_2 \le r_2$, вероятности потери заявки по сравнению с одноканальным вариантом по абсолютной величине снижаются по обоим потокам. Справа от точки излома $L_2 = r_2$ вероятность потери ЗНП снижается относительно одноканальной модели благодаря увеличению P_2 вследствие наличия в СМО двух приборов. По той же причине справа от точки излома $L_1 = r_1$ возрастают потери заявок первого потока.

Распространяя действие полученных формул на случай m приоритетных потоков и n каналов, с помощью рассуждений, подобных приведенным выше, можно получить выражения для оценки вероятностей потерь заявок по произвольно выбранному k-му приоритетному потоку. Как и в случае с общим БН, оно отображает три фазы функционирования модели, однако не в аспекте анализа длины динамически изменяющегося свободного от ЗВП остатка БН, а по степени заполнения собственных секций:

- секция не переполняется;
- переполняется, но все соседние секции не переполняются;
- своя и часть соседних секций переполняются.

Для первой фазы (помечена верхним индексом I) имеем:

$$\pi_k^{(I)} = \frac{\dfrac{n^{n-1}}{(n-1)!}\left(\sum\limits_{i=1}^{k}\rho_i\right)^{r_k+n} \times \left(1-\sum\limits_{i=1}^{k}\rho_i\right)}{\sum\limits_{h=0}^{n}\dfrac{n^{h-1}(n-h)}{h!}\left(\sum\limits_{i=1}^{k}\rho_i\right)^{h} - \dfrac{n^{n-1}}{(n-1)!}\left(\sum\limits_{i=1}^{k}\rho_i\right)^{r_k+n+1}}, \text{ если } L_k \le r_k.$$

Для второй фазы (помечена верхним индексом II):

$$\pi_k^{(II)} = \frac{\dfrac{n^{n-1}}{(n-1)!}\left(\sum\limits_{i=1}^{k}\rho_i\right)^{r_k+n} \times \left(1-\sum\limits_{i=1}^{k}\rho_i\right)}{\sum\limits_{h=0}^{n}\dfrac{n^{h-1}(n-h)}{h!}\left(\sum\limits_{i=1}^{k}\rho_i\right)^{h} - \dfrac{n^{n-1}}{(n-1)!}\left(\sum\limits_{i=1}^{k}\rho_i\right)^{r_k+n+1}} + \sum_{j=k+1}^{m}\delta_{jk}\cdot\rho_j\cdot\left[1-\frac{\gamma_k r_k}{(1+\vartheta^2)\rho_k\gamma_j}\right]\times$$

$$\times\frac{\sum\limits_{h=0}^{n}\dfrac{n^{h-1}(n-h)}{h!}\left(\sum\limits_{i=1}^{j}\rho_i\right)^{h} - \dfrac{n^{n-1}}{(n-1)!}\left(\sum\limits_{i=1}^{j}\rho_i\right)^{r_k+n}}{\sum\limits_{h=0}^{n}\dfrac{n^{h-1}(n-h)}{h!}\left(\sum\limits_{i=1}^{j}\rho_i\right)^{h} - \dfrac{n^{n-1}}{(n-1)!}\left(\sum\limits_{i=1}^{j}\rho_i\right)^{r_k+n+1}}, \quad \text{если } L_k > r_k \text{ и все } L_j \le r_j.$$

Для третьей фазы, помеченной верхним индексом III:

$$\pi_k^{(III)} = \frac{\dfrac{n^{n-1}}{(n-1)!}\left(\sum\limits_{i=1}^{k}\rho_i\right)^{r_k+n} \times \left(1-\sum\limits_{i=1}^{k}\rho_i\right)}{\sum\limits_{h=0}^{n}\dfrac{n^{h-1}(n-h)}{h!}\left(\sum\limits_{i=1}^{k}\rho_i\right)^{h} - \dfrac{n^{n-1}}{(n-1)!}\left(\sum\limits_{i=1}^{k}\rho_i\right)^{r_k+n+1}} + \sum_{j=k+1}^{m}\delta_{jk}\cdot\frac{r_j}{L_k}\cdot\left[1-\frac{\gamma_k r_k}{(1+\vartheta^2)\rho_k\gamma_j}\right]\times$$

$$\times\frac{\sum\limits_{h=0}^{n}\dfrac{n^{h-1}(n-h)}{h!}\left(\sum\limits_{i=1}^{j}\rho_i\right)^{h} - \dfrac{n^{n-1}}{(n-1)!}\left(\sum\limits_{i=1}^{j}\rho_i\right)^{r_k+n}}{\sum\limits_{h=0}^{n}\dfrac{n^{h-1}(n-h)}{h!}\left(\sum\limits_{i=1}^{j}\rho_i\right)^{h} - \dfrac{n^{n-1}}{(n-1)!}\left(\sum\limits_{i=1}^{j}\rho_i\right)^{r_k+n+1}},$$

если $L_k > r_k$ и хотя бы для одного $j = \overline{k+1, m}$ $L_j > r_j$,

где $L_k = (1+\vartheta^2)\cdot\rho_k \cdot \sum_{i=1}^{k-1}\gamma_i \dfrac{1-\left(\dfrac{\rho_i}{\gamma_i}\right)^{h_i+1}}{1-\dfrac{\rho_i}{\gamma_i}} \cdot h_i = \dfrac{\ln\lambda_i - \ln\sum_{s=1}^{k-1}\lambda_s}{\ln\left[(1+\vartheta^2)\gamma_i\dfrac{\rho_i}{\gamma_i}\right]}$, $\delta_{jk} = \begin{cases} 0, \text{если } L_k \le r_k, \\ 1 \text{ в противном случае.} \end{cases}$

Для учета влияния связности между записями следует в выражения π_k и P_j подставлять $\rho = \lambda / Q\mu$.

4.3.2.3. Сравнительные оценки динамической и статической дисциплин распараллеливания. Для сопоставления эффективности статической и динамической дисциплин на рис. 4.12 представлены графики зависимостей $\pi = f(\gamma)$ для случаев статического распараллеливания с приемом записей в раздельные секции БН и для динамического разделения с приемом в БН с общим доступом. Потери в системе с жестким разделением (подстрочный индекс *стат* на рисунке) не зависят от соотношения γ параметров обслуживания, а при динамическом (индекс *дин*) – нарастают с увеличением аргумента. При $\gamma_{кр} = r / (1+\vartheta^2)\rho_1$ суммарный нормированный штраф $\alpha\pi$ за потерю заявки для динамического распараллеливания достигает уровня, доставляемого при статической дисциплине, и с дальнейшим ростом γ превосходит его.

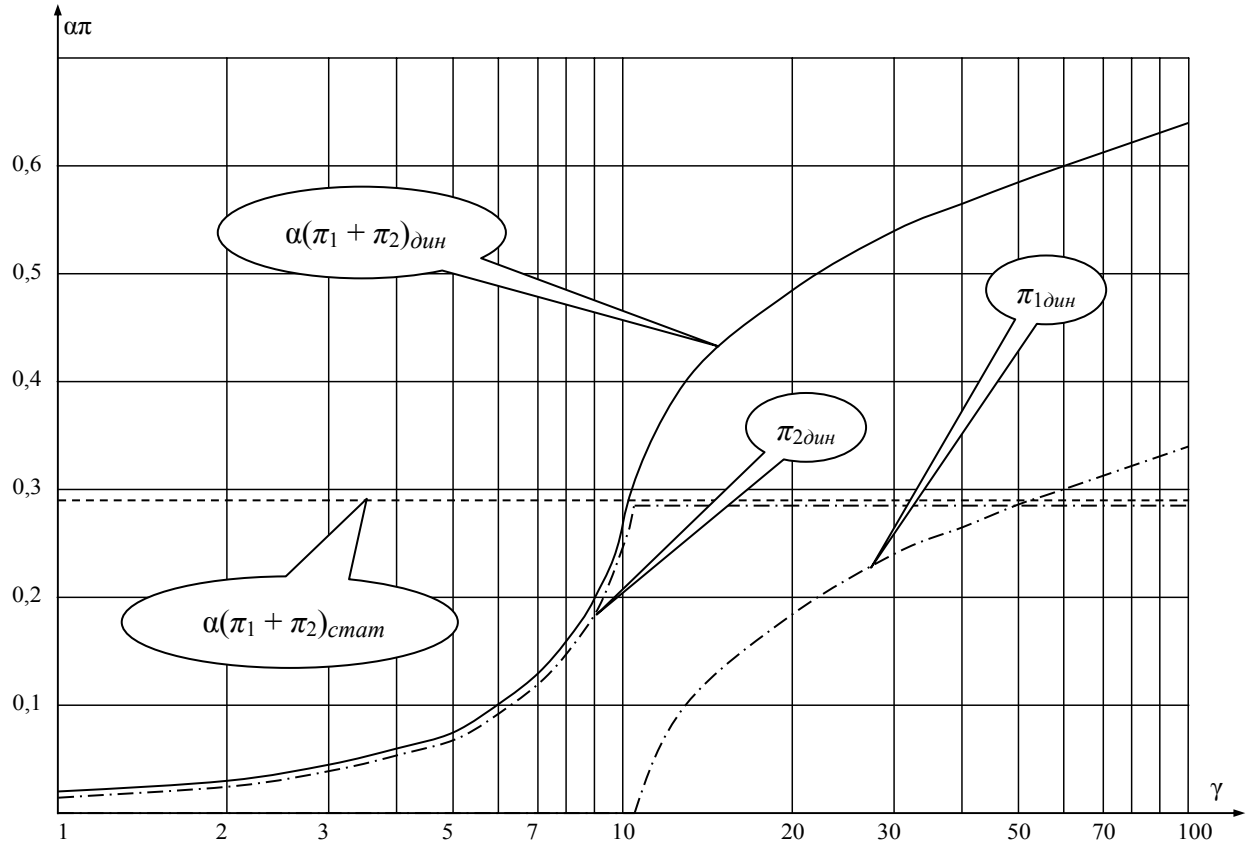


Рис. 4.12. Графики зависимостей $\pi = f(\gamma)$ для статического распараллеливания с приемом записей в раздельные секции БН и для динамического разделения с приемом в БН с общим доступом

Полученные результаты, помимо основной направленности на оценку необходимых объемов буферных зон, могут быть обобщены как исследова-

ние частных дисциплин диспетчеризации вычислительного процесса в системах с приоритетами. Такое обобщение возможно, во-первых, вследствие наличия аппарата, подкрепленного экспериментальной проверкой методом статистического моделирования, позволяющего рассчитать проект в виде объемов БН для заявок на обслуживание при заданных параметрах входных потоков. Во-вторых, расчеты по формулам, полученным выше, дают возможность найти области устойчивого поведения критерия при колебаниях этих параметров.

4.3.3. Характеристики времени обслуживания заявок

4.3.3.1. Расчеты времени обслуживания производятся с целью уменьшения нагрузок на обслуживающие приборы, уменьшения длины очередей, снижения затрат на обслуживание, увеличения пропускной способности системы. Основные показатели СМО: длина очереди, время нахождения требования в системе, доля времени, в течение которого прибор бывает свободен. Для ОСРВ главной задачей является гарантия обработки всех происходящих событий в установленные по замыслу АС УВД директивные сроки. Оценки вероятностей потери заявок при различных дисциплинах получены выше. В данном разделе рассмотрены показатели времени ожидания обслуживания.

Рассмотрим модель с пуассоновским входящим потоком и экспоненциальным распределением времени обслуживания. Напомним, что распределение Пуассона есть распределение вероятностей случайных величин x_i, принимающих целые неотрицательные значения $k = 0,1,2,\ldots, n$ с вероятностями $P(x = k) = \frac{\lambda^k e^{-\lambda}}{k!}$, где $\lambda > 0$ – интенсивность входного потока.

Математическое ожидание, дисперсия и моменты более высоких порядков равны λ. Сумма независимых случайных величин X_i, имеющих распределение Пуассона с параметрами λ_i, подчиняется также распределению Пуассона с параметрами $\sum\lambda_i$.

Время обслуживания (как и время между событиями в системе), когда поток обслуживания (или событий) обладает свойствами стационарности, ординарности и отсутствия последействия, распределено по экспоненциальному закону $g(t) = \mu e^{-\mu t}$, где μ – параметр обслуживания, величина, обратная среднему времени обслуживания одной заявки: $\mu = 1/T$. Пусть требование поступает в систему и дожидается обслуживания без каких-либо приоритетов. Тогда расчетные формулы для такой системы имеют вид:

- вероятность P_0 того, что обслуживающий прибор свободен $P_0 = 1 - \rho$;
- среднее число $E(n)$ требований в системе (находящихся в очереди и на обслуживании) $E(n) = \rho/(1 - \rho)$;
- среднее время $E(t)$ ожидания обслуживания $E(t) = \rho/[\mu(1 - \rho)]$;
- средняя длина $E(n_o)$ очереди, ожидающей обслуживания:

$$E(n_o) = \rho^2/(1 - \rho);$$

• среднее время $E(t_c)$, проведенное заявкой в системе $E(t_c) = 1/[\mu(1-\rho)]$.

Пример 1. Требования поступают на обслуживающее устройство случайно, причем средний промежуток времени между поступлениями требований равен 1,0 мин, среднее время обслуживания – 0,8 мин. Определить: среднее число требований в системе; среднее время ожидания обслуживания; среднюю длину очереди, ожидающей обслуживания; среднее время; проведенное требованием в системе; вероятность отсутствия требований в системе, если она состоит из одного прибора и имеет пуассоновский входящий поток и экспоненциальное время обслуживания.

Решение. Так как средний промежуток времени между поступлениями заявок известен $T = 1$ мин, то их среднее число в течение 1 мин. $\lambda = 1$. Поскольку среднее время обслуживания $T = 0{,}8$ мин, то среднее число заявок, обслуживаемых в 1 мин, $\mu = 1/T$; $\mu = 1/0{,}8 = 1{,}25$. Тогда вероятность простоя системы $P_0 = 1 - \rho$; $P_0 = 1 - 0{,}8 = 0{,}2$, т. е. 20 % рабочего времени система простаивает. Среднее число заявок в СМО (в очереди плюс одна обслуживается) $E(n) = \rho/(1 - \rho)$; $E(n) = 0{,}8/(1 - 0{,}8) = 4$. Среднее время ожидания в очереди $E(t) = \rho/\mu(1-\rho)$; $E(t) = 0{,}8/(1{,}25 \cdot 0{,}2) = 3{,}2$ мин. Средняя длина очереди, ожидающей обслуживания, $E(n_0) = \rho^2/(1-\rho)$; $E(n) = 0{,}8^2/(1 - 0{,}8) = 3{,}2$. т. е., как правило, немногим больше трех заявок ожидают в очереди. Среднее время, проведенное в СМО, сначала ожидания в очереди, а потом и собственно обслуживания, $E(t_c) = 1/\mu(1-\rho)$; $E(t_c) = 1/[1{,}25 \cdot (1 - 0{,}8)] = 4$ мин.

Пример 2. При тех же условиях задачи рассматривается ситуация, когда добавлен еще один канал обслуживания. Как изменятся первые три основных показателя?

Решение. Вероятность P_0 простоя системы $P_0 = (2 - \rho)/(2 + \rho)$, откуда $P_0 = (2 - 0{,}8)/(2 + 0{,}8) = 0{,}43$, т. е. 43% времени каналы простаивают. Среднее число заявок в системе $E(n) = 2\rho/(4 - \rho^2)$; $E(n) = 2 \cdot 0{,}8/(4 - 0{,}8^2) = 0{,}48$, т. е. очереди практически нет. Среднее время ожидания обслуживания $E(t) = \rho^2/[\mu(4-\rho^2)]$; $E(t) = 0{,}8^2/1{,}25(4 - 0{,}8^2) = 0{,}15$ мин.

4.3.3.2. Время ожидания в приоритетной СМО с общим БН. В параграфе 4.1.3.2 рассмотрены фазы процесса обслуживания заявок k-го уровня приоритетности, объединяющего все состояния СМО, в которых наиболее вероятная длина $L_{k\Sigma}$ очереди всех приоритетных потоков от высшего (первого) до $(k-1)$-го не превосходит количества r мест для ожидания. В «щадящем» и «критическом» режимах в БН накапливаются $\left(1+\vartheta^2\right)\gamma_k \sum_{i=1}^{k}\left(\frac{\rho_i}{\gamma_i}\right) \le r$ ожидающих обслуживания заявок. Если среднее время обслуживания i-й заявки равно T_i, тогда среднее значение $\overline{T_k}$ времени ожидания обслуживания заявками k-го приоритета есть $\overline{T_k} = \sum_{i=1}^{k-1} \rho_i T_i$, а наиболее вероятное значение суммарного времени $T_{k\Sigma}$ освобождения СМО от $L_{k\Sigma}$ ЗВП – $T_{k\Sigma} = \left(1+\vartheta^2\right)\sum_{i=1}^{k-1} \rho_i T_i$.

В режиме перегрузки, когда за время обслуживания одной заявки k-го

приоритета в БН накапливается очередь ЗВП, длина которой превышает его объем r, среднее время ожидания обслуживания уменьшается за счет прореживания потока ЗВП и рассчитывается по следующей схеме. На первом шаге вычисляется длина очереди ЗВП с индексами i потоков, меньшими k, $j < k$, которая может быть размещена в БН, т.е. не превосходит r. Для этих заявок рассчитывается время освобождения СМО от них, затем определяется время обслуживания $[r - (1+\vartheta^2)\sum_{i=1}^{j-1}\rho_i\,]T_j$ заявок j-го уровня приоритетности. Сумма полученных величин $T_{k\Sigma} = (1+\vartheta^2)\sum_{i=1}^{j-1}\rho_i T_i + \left[r - (1+\vartheta^2)\sum_{i=1}^{j-1}\rho_i \right] T_j$ и есть искомое время ожидания обслуживания для заявок k-го типа в режиме перегрузки.

4.3.3.3. Время ожидания в приоритетной СМО с раздельным БН. Соответствующие выражения уже получены в параграфе 4.2.2 данного раздела. Для определения величины времени T рассмотрен процесс образования очереди $L_k - 1$ заявок более высоких, чем k, приоритетов, исходя из соотношения

$$L_{k-1} = \sum_{i=1}^{k-1} Q_i \text{, где } \quad Q_i = \begin{cases} (1+\vartheta^2)\,\gamma_k \left(\dfrac{\rho_i}{\gamma_i}\right), \text{ если } Q_i \le r_i, \\ r_i \text{ в противном случае.} \end{cases}$$

Наиболее вероятное значение T_{Qi} занятости системы обслуживанием очереди Q_i заявок составит $T_{Qi} = (1+\vartheta^2)\, Q_i\, T_i$, где T_i – среднее время обслуживания заявки i-го типа, и для очереди $L_{k\text{-}1}$ можно приближенно записать

$$T_L = \sum_{i=1}^{k-1} T_{Qi} = (1+\vartheta^2)\,\gamma_k \sum_{i=1}^{k-1} (T_i\,\rho_i/\gamma_i).$$

В течение этого времени в системе образуется приращение $\Delta L'$ очереди заявок потоков высших приоритетов с индексами $i < k$. В силу стационарности и условия $\sum_{i=1}^{m}\rho_i < 1$, интенсивности λ_i поступлений таких заявок меньше параметров μ_i их обслуживания и суммарная очередь $L_{k\text{-}1}$ вместе с приращениями $\Delta L'$, $\Delta L''$, …, ΔL^{l_i} со временем укорачивается

$$\Delta L' = (1+\vartheta^2)\,\gamma_k \sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right)^2, \ \Delta L'' = (1+\vartheta^2)\,\gamma_k \sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right)^3 \ \ldots$$

Вообще при расчете $L_{k\text{-}1}$ можно рассмотреть l_i монотонно убывающих приращений суммарной очереди заявок i-х уровней приоритетности, более высоких, чем k-й, ($i < k$). Процедура вычисления $L_{k\text{-}1}$ заканчивается на l_i-м шаге, когда получаем приращение очереди, не превышающее единицы. Тогда считаем, что СМО полностью освободилась от ЗВП, принадлежащих потокам с индексами $i < k$, и может приступить к обслуживанию заявок k-го типа.

В силу того, что поступление заявок разных типов пропорционально интенсивностям соответствующих входящих потоков, для выполнения последнего неравенства достаточно потребовать $(1+\vartheta^2)\,\gamma_k\,(\rho_i/\gamma_i)^{l_i} \le \lambda_i \Big/ \sum_{j=1}^{k-1}\lambda_j$, от-

куда $\Delta L^{l_i} = (1+\vartheta^2)\gamma_k \sum_{i=1}^{k-1}\left(\frac{\rho_i}{\gamma_i}\right)^{l_i} \le 1$, а $]l_i[= \ln\lambda_i - \ln\sum_{j=1}^{k-1}\lambda_j \Big/ \ln\left[(1+\vartheta^2)\gamma_k \frac{\rho_i}{\gamma_i}\right]$.

Тогда наиболее вероятная длина L очереди всех приоритетных заявок с индексами $i < k$, образующейся в системе за время обслуживания одной заявки j-то типа, $j = \overline{k,m}$, с учетом l_i приращений ее длины при освобождении от $L_{k\text{-}1}$ ЗВП, составит

$$L = (1+\vartheta^2)\gamma_k \sum_{i=1}^{k-1}\left[1-\left(\frac{\rho_i}{\gamma_i}\right)^{l_i+1}\right] \Big/ \left(1-\frac{\rho_i}{\gamma_i}\right),$$

а период T занятости системы обслуживанием накопленной очереди L заявок

$$T = (1+\vartheta^2)\gamma_k \sum_{i=1}^{k-1} T_i \left[1-\left(\frac{\rho_i}{\gamma_i}\right)^{l_i+1}\right] \Big/ \left(1-\frac{\rho_i}{\gamma_i}\right).$$

Расчетные значения сопоставляются с директивными сроками («временем жизни»), в течение которых заявки должны быть обслужены. Если результаты расчетов не укладываются в заданные сроки, необходимо изменить условия обслуживания таким образом, чтобы они выполнялись.

Вопросы для самопроверки

1. Какая практическая потребность вызвала к жизни функцию ОС по диспетчеризации и планированию вычислений (п. 4.1.1)?
2. По каким причинам методы анализа СМО малопригодны для оценки систем с неоднородным входящим потоком (п. 4.1.2)?
3. Сформулируйте задачу анализа дисциплины облуживания с относительным приоритетом и приоритетным приемом заявок в общий БН (п. 4.1.3).
4. Укажите отличия модели приоритетного облуживания с общим БН от модели с раздельными секциями (п. 4.2).
5. Как изменяется эффективность приоритетного обслуживания при переходе от работы на одном компьютере к компьютерной сети (п. 4.3.1)?
6. В чем состоят различия между дисциплинами статического и динамического разделения задач на сети компьютеров (п. 4.3.2)?
7. Какими временными оценками характеризуется эффективность организации вычислительного процесса (п. 4.3.3)?

5. Заключение

Данная книга подготовлена для выпускников МГТУ ГА, которым предстоит работа на авиационных предприятиях. Компьютеры проникли во все сферы современной жизни, обеспечивать их полноценное использование необходимо и в Федеральном агентстве воздушного транспорта, и в проектных или в научно-исследовательских институтах отрасли, и в авиаотрядах.

Операционные системы представляют собой основу любого программного обеспечения. Особое место занимает разработка автоматизированных систем организации воздушного движения, в которых действует особый класс ОС – системы реального времени, которым посвящена эта брошюра.

Книга написана не для создателей новых операционных систем, такая литература широко известна, как и разнообразные издания, посвященные конкретным образцам фирменных продуктов. Задача, поставленная автором при подготовке рукописи, состоит в изложении типовых ситуаций, с которыми сталкиваются инженеры, использующие хорошо зарекомендовавшие себя изделия для организации вычислительного процесса в АС УВД. Специфика авиационных систем, управления работой ПО, средствами отображения, подсистемами сбора и рассылки информации такова, что объективно возникает необходимость дорабатывать фирменные образцы, подгоняя их к реалиям предметной области. В книге рассмотрены проблемы:

- управления файловой системой в реальном времени, без прекращения (без блокировок) доступа к хранимой информации;
- организации параллельных вычислений на компьютерной сети;
- диспетчеризации работ в системах с приоритетами.

Каждый раздел начинается с обсуждения основных понятий и терминов, а также разъяснения причин появления соответствующей функции ОС. Далее следует словесное разъяснение технического решения, положенного в основу программной доработки фирменной ОС, которое подкрепляется математическим обоснованием технической идеи. Материал рассчитан на студентов пятого курса специальности 23.01.01.

ЛИТЕРАТУРА

1. **Таненбаум Э.С., Вудхалл А.** Операционные системы. Разработка и реализация. - СПб: Питер, 2007
2. **Олифер В.Г., Олифер Н.А.** Сетевые операционные системы. - СПб: Питер, 2007
3. **Бурдонов И.Б., Косачев А.С., Пономаренко В.Н.** Операционные системы реального времени. - М.: Институт системного программирования РАН, 2006
4. **Жданов А.А.** Операционные системы реального времени. // "PC Week", N 8, 1999
5. **В.А. Ребров, Л.Е. Рудельсон, М.А. Черникова.** Модель сбора и обработки заявок на полеты в задаче планирования авиарейсов. // Известия РАН, Теория и системы управления, 2007, № 3
6. **Вентцель Е.С.** Теория вероятностей. - М.: Высшая школа, 1999
7. **Молоканов Г.Ф.** Точность и надежность навигации летательных аппаратов. - М.: Машиностроение, 1967

На начало документа *К исходному документу*