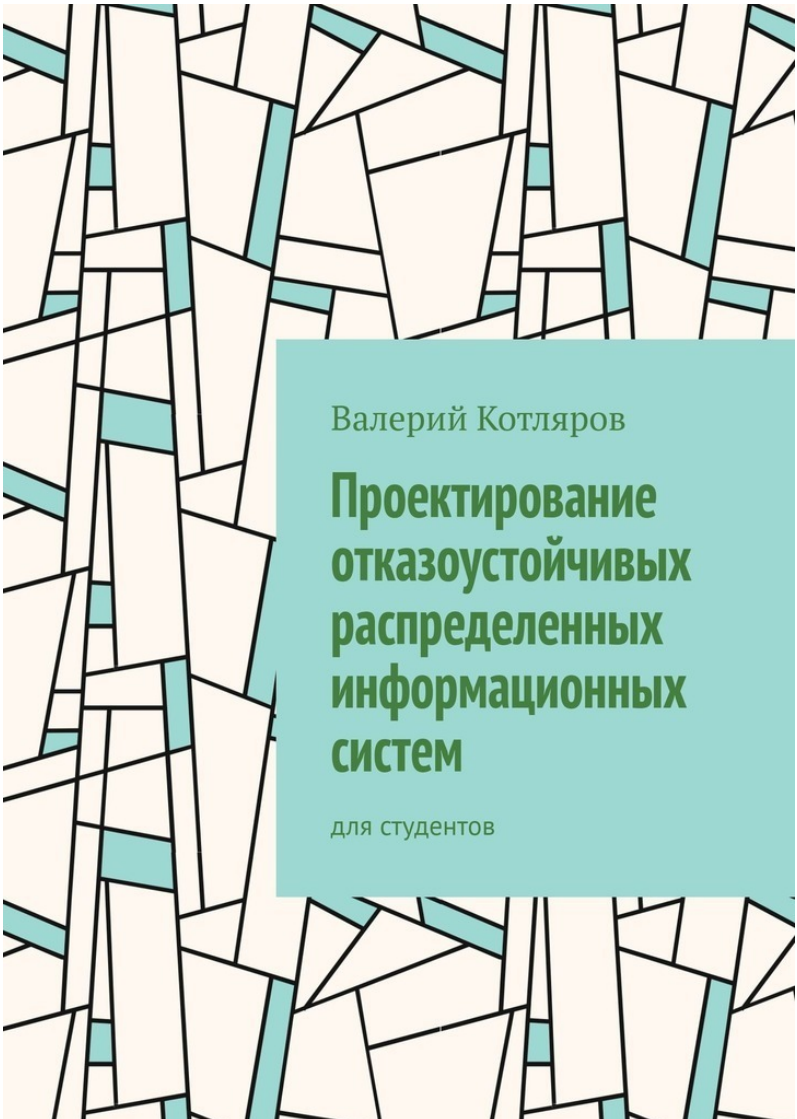


Валерий Котляров

Проектирование отказоустойчивых распределенных информационных систем

для студентов

Валерий Петрович Котляров

Проектирование отказоустойчивых распределенных информационных систем. Для студентов

http://www.litres.ru/pages/biblio_book/?art=42923164
ISBN 9785449650047

Аннотация

Данное пособие, в первую очередь, предназначено для студентов технических специальностей, связанных с информационной безопасностью и информационно-коммуникационными технологиями.

Содержание

Проектирование отказоустойчивых распределенных информационных систем Для студентов

Валерий Петрович Котляров

ISBN 978-5-4496-5004-7
Создано в интеллектуальной издательской системе Ridero

Введение

Целью изучения дисциплины является овладение современными методами и средствами технологии исследования и проектирования, разработки и использования проблемно – ориентированных отказоустойчивых распределенных информационных систем (РИС). Для достижения поставленной цели предусматривается решение следующих основных задач: изучение распределенной обработки информации в автоматизированных информационных системах, архитектуры РИС, технологической базы РИС, распределенных информационных ресурсов и сетей, распределенных баз данных, принципов и технологий управления обменом информацией в РИС, методов и средств доступа к удаленным информационным ресурсам.

1 Обозначение предметной области «распределенные информационные системы» и проблемных вопросов изучения дисциплины

Понятие «персональный компьютер», возникшее в уже далеком 1945 году и обозначающее индивидуальную работу пользователя в отдельно взятой комнате, изолированно от других пользователей, претерпело большие изменения и в реальной обыденной жизни и в виртуальном характере общения пользователя с информационными объектами. Начиная с середины восьмидесятых годов, большие и дорогие майнфреймы уступают место компактным компьютерам с более мощными микропроцессорами. Следующий виток технологического развития обозначается появлением локальных сетей (Local-Area Networks, LAN), позволяющих объединить сотни компьютеров, находящихся в здании, таким образом, что машины в состоянии обмениваться небольшими порциями информации за несколько микросекунд. Большие массивы данных передаются с машины на машину со скоростью от 10 Мбит/с до 10 Гбит/с. Затем появляются глобальные сети (Wide-Area Networks, WAN), позволя-

ющие миллионам машин во всем мире обмениваться информацией со скоростями, варьирующимися от 64 кбит/с (килобит в секунду) до гигабит в секунду.

В результате развития этих технологий сегодня не просто возможно, но и достаточно легко можно собрать компьютерную систему, состоящую из множества компьютеров, соединенных высокоскоростной сетью. Ее можно назвать простейшей распределенной информационной системой (РИС), в отличие от предшествовавших ей централизованных, или однопроцессорных систем, состоявших из одного компьютера, его периферии и, возможно, нескольких удаленных терминалов.

Технологический скачок революционного развития вычислительной техники потребовал концептуальных изменений в использовании средств обработки информации.

Появился новый термин – «распределенная информационная система». Возникает научная задача – термину РИС нужно дать лаконичное и научно обоснованное определение. На сегодняшний день по утверждению известного специалиста в области информатики Э. Таненбаума, не существует общепринятого и в то же время строгого определения распределенной системы. В современной литературе можно выделить следующие научные толкования нашего термина как распределенная автоматизированная система (РАС):

– РАС – это автоматизированная система управления, которая приобрела специфику территориально рассредоточен-

ной автоматизированной системы;

– РАС – это совокупность независимых объектов, которые взаимодействуют с целью решения проблемы, которая не может быть решена одним объектом индивидуально. При таком подходе распределенной является любая вычислительная система, где обработка данных разделена между двумя и более компьютерами;

– РАС – это совокупность независимых компьютеров, представляющаяся пользователям единой объединенной системой.

Такой подход к определению распределенной системы имеет свои недостатки. Например, все используемое в такой распределенной системе программное обеспечение могло бы работать и на одном единственном компьютере, однако с точки зрения приведенного выше определения такая система уже перестанет быть распределенной;

– распределенной является такая вычислительная система, в которой неисправность компьютера, о существовании которого пользователи ранее даже не подозревали, приводит к остановке всей их работы. Значительная часть распределенных вычислительных систем, к сожалению, удовлетворяют такому определению, однако формально оно относится только к системам с уникальной точкой уязвимости.

Первый прототип РИС, имеющий структуру **«клиент – сервер»,** следует рассматривать (рисунок 1.1) как некое типичное приложение, которое в соответствии с современны-

ми представлениями может быть разделено на следующие логические уровни:

– пользовательский интерфейс (ИП);

– логика приложения (ЛП);

– доступ к данным (ДД), работающий с базой данных (БД).

Рисунок 1.1 – Архитектура «клиент – сервер»

Пользователь системы взаимодействует с ней через интерфейс пользователя, база данных хранит данные, описываю-

щие предметную область приложения, а уровень логики приложения реализует все алгоритмы, относящиеся к предметной области.

Поскольку на практике разных пользователей системы обычно интересует доступ к одним и тем же данным, наиболее простым разнесением функций такой системы между несколькими компьютерами будет разделение логических уровней приложения между одной серверной частью приложения, отвечающим за доступ к данным, и находящимися на нескольких компьютерах клиентскими частями, реализующими интерфейс пользователя. Логика приложения может быть отнесена к серверу, клиентам, или разделена между ними. Архитектуру построенных по такому принципу приложений называют клиент-серверной или двухзвенной. На практике подобные системы часто не относят к классу распределенных, но формально они могут считаться простейшими представителями распределенных систем.

Развитием архитектуры клиент-сервер является трехзвенная архитектура, в которой интерфейс пользователя, логика приложения и доступ к данным выделены в самостоятельные составляющие системы, которые могут работать на независимых компьютерах. Запрос пользователя в подобных системах последовательно обрабатывается клиентской частью системы, сервером логики приложения и сервером баз данных. Однако обычно под распределенной системой понимают системы с более сложной архитектурой, чем трехзвенная.

Применительно к приложениям автоматизации деятельности предприятия, распределенными обычно называют системы с логикой приложения, распределенной между несколькими компонентами системы, каждая из которых может выполняться на отдельном компьютере. Например, реализация логики приложения системы розничных продаж (рисунок 1.2) должна использовать запросы к логике приложения третьих фирм, таких как поставщики товаров, системы электронных платежей или банки, предоставляющие потребительские кредиты. Таким образом, в обиходе под распределенной системой часто подразумевают рост многозвенной архитектуры «в ширину», когда запросы пользователя не проходят последовательно от интерфейса пользователя до единственного сервера баз данных.

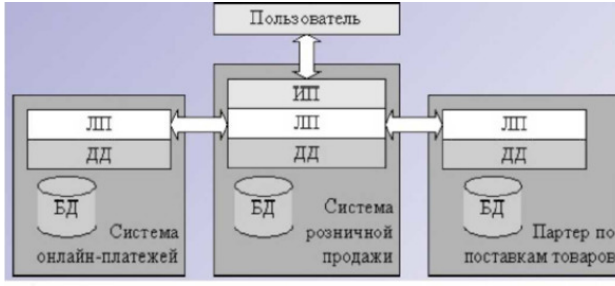


Рисунок 1.2 – Логика приложения системы розничных продаж

В качестве другого примера (рисунок 1.3) распределенной системы можно привести **сети прямого обмена данными между клиентами** (peer-to-peer networks). Если предыдущий пример имел «древовидную» архитектуру, то сети прямого обмена организованы более сложным образом. Подобные системы являются в настоящий момент, вероятно, одними из крупнейших существующих распределенных систем, объединяющие миллионы компьютеров.

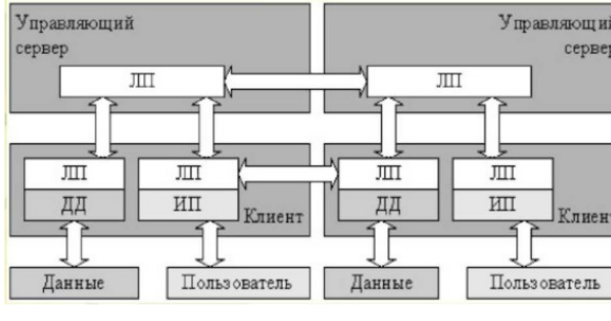


Рисунок 1.3- Сети прямого обмена данными между клиентами

Другая популярная архитектура **– Grid.** Грид вычисления – это форма распределённых вычислений, в которой «супер и виртуальный компьютер» представлен в виде кластера соединенных с помощью сети, слабосвязанных компьютеров, работающих вместе для выполнения огромного количества заданий (операций, работ). Эта технология бы-

ла применена для решения научных, математических задач, требующих для решения значительных вычислительных ресурсов. Грид вычисления используются также и в коммерческой инфраструктуре для решения таких трудоёмких задач как экономическое прогнозирование, сейсмоанализ, разработка и изучение свойств новых лекарств. Если рассмотреть какие же **характеристики** стали присущи к распределенным информационным системам, то можно отметить следующие:

– прозрачность реализации;
– открытость;
– легкая масштабируемость и расширяемость;
– устойчивость к авариям;
– наличие промежуточного уровня.

Первая из характеристик состоит в том, что от пользователей скрыты различия между компьютерами и способы связи между ними. То же самое относится и к внешней организации распределенных систем.

Другой важной характеристикой распределенных систем является способ, при помощи которого пользователи и приложения единообразно работают в распределенных системах, независимо от того, где и когда происходит их взаимодействие.

Распределенные системы должны также относительно легко поддаваться расширению, или масштабированию. Эта характеристика является прямым следствием наличия неза-

висимых компьютеров, но в то же время не указывает, каким образом эти компьютеры на самом деле объединяются в единую систему.

Распределенные системы обычно существуют постоянно, однако некоторые их части могут временно выходить из строя. Пользователи и приложения не должны уведомляться о том, что эти части заменены или починены или что добавлены новые части для поддержки дополнительных пользователей или приложений.

Для того чтобы поддержать представление различных компьютеров и сетей в виде единой системы, организация распределенных систем часто включает в себя дополнительный уровень программного обеспечения, находящийся между верхним уровнем, на котором находятся пользователи и приложения, и нижним уровнем, состоящим из операционных систем.

Соответственно, такая распределенная система обычно называется **системой промежуточного уровня** (middleware).

Использование протокола TCP/IP посредством сокетов предоставляет стандартный, межплатформенный, но низкоуровневый сервис для обмена данными между компонентами. Для выполнения сформулированных выше требований к распределенным системам функции сеансового и представительского уровня должна взять на себя некоторая **промежуточная среда** (middleware), называемая так же промежу-

точным программным обеспечением. Такая среда (рисунок 4) должна помогать разработчикам создавать открытые, масштабируемые и устойчивые распределенные системы.

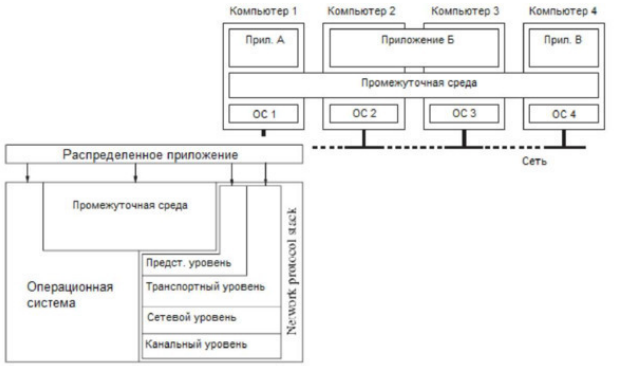


Рисунок 1.4 – Промежуточное программное обеспечение

Для достижения этой цели промежуточная среда должна обеспечить сервисы для взаимодействия компонент распределенной системы. К таким сервисам относятся:

– обеспечение единого и независимого от операционной системы механизма использования одними программными компонентами сервисов других компонент;

– обеспечение безопасности распределенной системы: аутентификация и авторизация всех пользователей сервисов компоненты и защита передаваемой между компонентами информации от искажения и чтения третьими сторонами;

– обеспечение целостности данных: управление транзакциями, распределенными между удаленными компонентами системами;

– балансировка нагрузки на серверы с программными компонентами;

– обнаружение удаленных компонент.

Чтобы сделать разработку и интеграцию распределенных приложений как можно более простой, основная часть программного обеспечения промежуточного уровня базируется на некоторой модели, или парадигме, определяющей распределение и связь. Относительно простой моделью является **представление всех наблюдаемых объектов в виде файлов,** построенной по принципу распределенной файловой системы (distributed file system). Во многих случаях это программное обеспечение всего на один шаг ушло от сетевых операционных систем в том смысле, что прозрачность распределения поддерживается только для стандартных файлов (то есть файлов, предназначенных только для хранения данных). Процессы, например, часто должны запускаться исключительно на определенных машинах. Программное обеспечение промежуточного уровня, основанное на модели распределенной файловой системы, оказалось достаточно легко масштабируемым, что способствовало его популярности.

Другая важная ранняя модель программного обеспечения промежуточного уровня основана на **удаленных вызовах**

процедур (Remote Procedure Calls, RPC). В этой модели акцент делается на сокрытии сетевого обмена за счет того, что процессу разрешается вызывать процедуры, реализация которых находится на удаленной машине. При вызове такой процедуры параметры прозрачно передаются на удаленную машину, где, собственно, и выполняется процедура, после чего результат выполнения возвращается в точку вызова процедуры. За исключением, вероятно, некоторой потери производительности, все это выглядит как локальное исполнение вызванной процедуры: вызывающий процесс не уведомляется об имевшем место факте сетевого обмена.

По мере того как все более входит в моду ориентированность на объекты, становится ясно, что если вызов процедуры проходит через границы отдельных машин, он может быть представлен в виде прозрачного обращения к объекту, находящемуся на удаленной машине. Это привело к появлению разнообразных систем промежуточного уровня, реализующих представление о распределенных объектах (distributed objects). Идея распределенных объектов состоит в том, что каждый объект реализует интерфейс, который скрывает все внутренние детали объекта от его пользователя. Интерфейс содержит методы, реализуемые объектом, не больше и не меньше. Все, что видит процесс, – это интерфейс. Когда процесс вызывает метод, реализация интерфейса на машине с процессом просто преобразует вызов метода в сообщение, пересылаемое объекту. Объект выполня-

ет запрашиваемый метод и отправляет назад результаты. Затем реализация интерфейса преобразует ответное сообщение в возвращаемое значение, которое передается вызвавшему процессу. Microsoft DCOM (Distributed COM – распределённая COM) основана на технологии DCE/RPC (разновидности RPC). DCOM позволяет COM-компонентам взаимодействовать друг с другом по сети. Технология DCOM обеспечивает базовые установки безопасности позволяя задавать, кто и из каких машин может создавать экземпляры объекта и вызывать его методы; OMG CORBA (Common Object Request Broker Architecture – общая архитектура брокера объектных запросов) – это технологический стандарт, продвигаемый консорциумом OMG, задачей которого является осуществить интеграцию изолированных систем, дать возможность программам, написанным на разных языках, работающим на разных узлах сети, взаимодействовать друг с другом так же просто, как если бы они находились в адресном пространстве одного процесса; Java RMI (Remote Method Invocation) – программный интерфейс вызова удаленных методов в языке Java.

Как модели могут упростить использование сетевых систем, вероятно, наилучшим образом видно на примере World Wide Web. Успех среды Web в основном определяется тем, что она построена на базе потрясающе простой, но высокоэффективной модели распределенных документов (distributed documents). В модели, принятой в Web, инфор-

мация организована **в виде документов,** каждый из которых размещен на машине, расположение которой абсолютно прозрачно. Документы содержат ссылки, связывающие текущий документ с другими. Если следовать по ссылке, то документ, с которым связана эта ссылка, будет извлечен из места его хранения и выведен на экран пользователя. Концепция документа не ограничивается исключительно текстовой информацией. Например, в Web поддерживаются аудио- и видеодокументы, а также различные виды документов на основе интерактивной графики.

Итак, еще раз перечислим и кратко охарактеризуем модели ПУ:

1 Распределенная файловая система обозначение – «Distributed File System»; достоинство – Относительно простая модель;

цель: обеспечить прозрачный доступ удаленных пользователей к файловой системе;

пример: NFS.

2 Удаленный вызов процедур

обозначение – «Remote Procedure Call (RPC)»;

цель: обеспечение прозрачности удаленного исполнения кода;

особенности функционирования:

– реализация процедуры находится на сервере;

– клиент передает параметры процедуры;

– сервер исполняет процедуру и возвращает результат

– некоторая потеря производительности;

– весь сетевой обмен скрыт от процесса.

3 Распределенные объекты

обозначение – «Distributed Objects:»; особенности функционирования:

– каждый объект реализует интерфейс;

– интерфейс содержит методы, реализуемые объектом;

– процесс видит только интерфейс;

– наиболее популярные технологии распределенных объектов в настоящее время:

– Microsoft DCOM;

– OMG CORBA

– Java RMI.

4 Распределенные документы

обозначение «Distributed Documents»;

– реализация: World Wide Web

– цель: Прозрачность размещения документов; особенности функционирования:

– ссылки связывают документы;

– содержимое не ограничено текстовой информацией.

Кратко сформулируем задачи промежуточного уровня:

– обеспечение интероперабельности;

– обеспечение безопасности;

– обеспечение целостности данных;

– балансировка нагрузки;

– обнаружение удаленных компонент.

Чтобы достигнуть цели своего существования – улучшения выполнения запросов пользователя – распределенная информационная система должна удовлетворять некоторым необходимым требованиям.

Можно сформулировать следующий **набор требований,** которым в наилучшем случае должна удовлетворять РИС.

Открытость. Все протоколы взаимодействия компонент внутри распределенной системы в идеальном случае должны быть основаны на общедоступных стандартах. Это позволяет использовать для создания компонент различные средства разработки и операционные системы. Каждая компонента должна иметь точную и полную спецификацию своих сервисов. В этом случае компоненты распределенной системы могут быть созданы независимыми разработчиками. При нарушении этого требования может исчезнуть возможность создания распределенной системы, охватывающей несколько независимых организаций.

Масштабируемость. Масштабируемость вычислительных систем имеет несколько аспектов. Наиболее важный из них – возможность добавления в распределенную систему новых компьютеров для увеличения производительности системы, что связано с понятием балансировки нагрузки (load balancing) на серверы системы. К масштабированию относятся так же вопросы эффективного распределения ресурсов сервера, обслуживающего запросы клиентов.

Поддержание логической целостности данных. Запрос пользователя в распределенной системе должен либо корректно выполняться целиком, либо не выполняться вообще. Ситуация, когда часть компонент системы корректно обработали поступивший запрос, а часть – нет, является наихудшей.

Устойчивость. Под устойчивостью понимается возможность дублирования несколькими компьютерами одних и тех же функций или же возможность автоматического распределения функций внутри системы в случае выхода из строя одного из компьютеров. В идеальном случае это означает полное отсутствие уникальной точки сбоя, то есть выход из строя одного любого компьютера не приводит к невозможности обслужить запрос пользователя.

Безопасность. Каждый компонент, образующий распределенную систему, должен быть уверен, что его функции используются авторизированными на это компонентами или пользователями. Данные, передаваемые между компонентами, должны быть защищены как от искажения, так и от просмотра третьими сторонами.

Эффективность. В узком смысле применительно к распределенным системам под эффективностью будет пониматься минимизация накладных расходов, связанных с распределенным характером системы. Поскольку эффективность в данном узком смысле может противоречить безопасности, открытости и надежности системы, следует отметить,

что требование эффективности в данном контексте является наименее приоритетным. Например, на поддержку логической целостности данных в распределенной системе могут тратиться значительные ресурсы времени и памяти, однако система с недостоверными данными вряд ли нужна пользователям.

Классическим примером системы, в значительной мере отвечающей всем представленным выше требованиям, является система преобразования символьных имен в сетевые IP-адреса (DNS). Система имен – организованная иерархически распределенная система, с дублированием всех функций между двумя и более серверами.

Повышение отношения производительности к затратам. Любая задача может быть разделена между между различными компьютерами в распределенной системе. Такая конфигурация обеспечивает лучшее соотношение производительности к стоимости системы. Это особенно актуально для конфигурации «сеть рабочих станций» (NOW).

Масштабируемость. Компьютеры, как правило, подключены к глобальной компьютерной сети, поэтому установка новых компьютеров непосредственно не создает узких мест в компьютерной сети.

Модульность и дополнительная расширяемость. Гетерогенные единицы могут быть добавлены в систему без снижения производительности, так как используется промежуточный уровень взаимодействия. Аналогично, существу-

ющие единицы могут быть легко заменены новыми.

Итак, можно считать, что предметная область обозначена и выделены проблемные вопросы, которые решаются специалистами информационных технологий. Исследования в области распределенных систем достаточно сложные, поэтому за выдающуюся статью по распределенным вычислениям ежегодно вручается «Премия Дейкстры» (http://www.podc.org/dijkstra/).

На данном этапе изучения дисциплины можно запомнить определение:

РИС – это совокупность автономных компьютеров, взаимодействующих через компьютерную сеть и промежуточную среду, которая позволяет компьютерам координировать свою деятельность и предоставлять доступ к ресурсам системы так, что пользователям система представляется единой и целостной. **Распределенная система обладает следующими свойствами:**

– отсутствие общей физической шины;

– отсутствие общей памяти;

– географическое распределение;

– автономность и гетерогенность.

Чтобы подробно разобраться со всеми проблемами, обозначенными в теории и практике РИС, ну проштудировать многотомное собрание сочинений. В данном учебном пособии обозначим рамки и направления изучения распределенных информационных систем. Объектом изучения обозна-

чим корпоративную распределенную информационную систему организации, предприятия (рисунок 1.5).

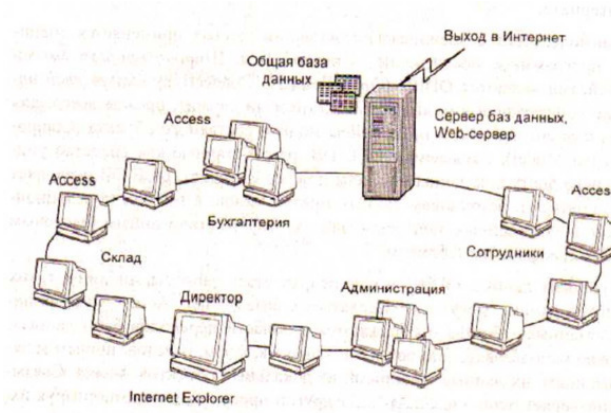
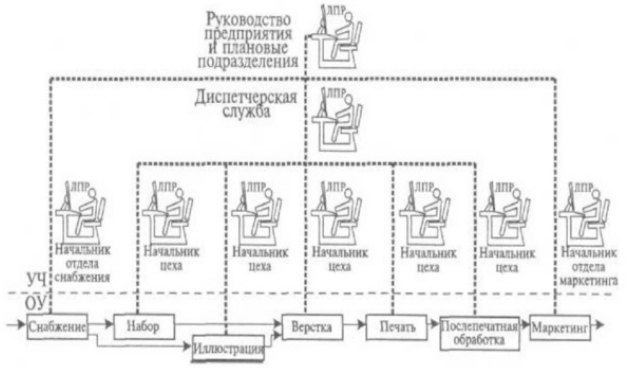


Рисунок 1.5 – Корпоративная сеть с SQL – и WEB – серверами

Типовой пример РИС такой сети можно представить схемой компьютерного интегрированного полиграфического производства (рисунок 1.6).



УЧ – управляющая часть; ОУ – объект управления

Рисунок 1.6 -Типовой пример распределенной информационной системы

Лекционный материал учебного пособия можно сгруппировать по трем разделам. Первый раздел предназначен для изучения основных составляющих автоматизированной информационной системы (АИС). Проведем маленький эксперимент. Ответьте на два простых вопроса: что такое данные? и что такое информация? Большинство отвечающих относятся к одному варианту ответа: данные – это информация; информация – это данные. Круг замыкается! Простые и обыденные понятия, но мы с ними свыклись и не вдаемся в суть определений. Поэтому в первом разделе учебного пособия уделено основное внимание разъяснению терминов, определений объектов, составляющих информационную систему

(ИС), концепции построения баз данных как основы ИС.

Конец ознакомительного фрагмента.