

# Администрирование информационных систем

Основы сетевого администрирования

# Сетевые операционные системы

---

- Сетевая операционная системы – комплекс программных модулей, предназначенный для повышения эффективности аппаратных ресурсов компьютера путем рационального управления его ресурсами и разделения ресурсов между множеством выполняемых в сети процессов.

# Компьютерные сети

- Под *компьютерной сетью* понимается совокупность компьютеров, связанных коммуникационной системой и снабженных необходимым программным обеспечением, позволяющим пользователям и приложениям получать доступ к ресурсам компьютеров.



# Сетевые и распределенные операционные системы

---

- *Сетевая ОС* обеспечивает пользователю некоторую виртуальную вычислительную систему, упрощающую работу с ней. Данная система не полностью скрывает распределенный характер своего прототипа, т.е. представляет собой *виртуальную сеть*.
- *Распределенная ОС* обеспечивает высокую степень прозрачности сетевых ресурсов, т.е. распределенная ОС предоставляет пользователю и приложениям сетевые ресурсы в виде единой централизованной виртуальной машины. Распределенная ОС автоматически распределяет процессы по различным компьютерам для обработки.

# Функциональные компоненты сетевой ОС

- Основные компоненты сетевой ОС:
  - Средства управления локальными ресурсами компьютера реализует все функции ОС автономного компьютера (управление процессами, оперативной памятью, управление внешней памятью, пользователями и т.п.)
  - Сетевые средства, разделяемые на три компонента:
    - Серверная часть ОС – средства предоставления локальных ресурсов и сервисов в общее пользование
    - Клиентская часть ОС – средства запроса на доступ к удаленным ресурсам и сервисам
    - Транспортные средства ОС, совместно с коммуникационной системой обеспечивающие передачу сообщений между компьютерами

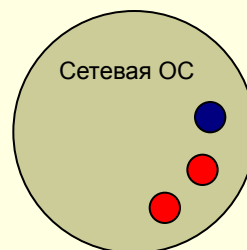
# Сетевые службы и сервисы

---

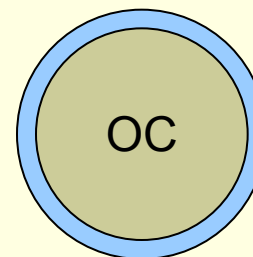
- *Сетевой службой* называется совокупность серверной и клиентской частей ОС, предоставляющих доступ к конкретному типу ресурса компьютера через сеть.
- *Сервис* – интерфейс между потребителем услуг (пользователем или приложением) и поставщиком услуг (службой)

# Подходы к построению сетевых операционных систем

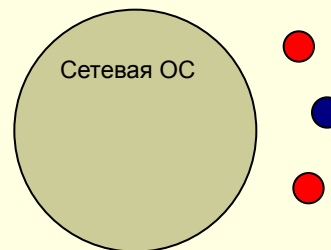
- Сетевые службы глубоко интегрированы в операционную систему (например, Windows NT)
- Сетевые службы объединены в виде некоторого набора программных модулей – *оболочки* (например, LAN Server, NetWare for UNIX)
- Сетевые службы разрабатываются и поставляются в виде отдельных программных модулей (NDS для различных ОС)



Встроенные сетевые службы



Сетевая оболочка



Сетевые службы – отдельные приложения

# Типы сетевых ОС

---

- В зависимости от распределения функций между компьютерами, они могут выступать в роли *выделенного сервера* или *клиентского узла*
- Сеть может быть построена по следующим схемам:
- На основе компьютеров, совмещающих функции клиента и сервера – *одноранговая сеть*
- На основе клиентов и серверов – *сеть с выделенными серверами*
- Сеть, включающая узлы разных типов – *гибридная сеть*.



# Модели сетевых служб и распределенных приложений

---

- Выделяют три основных параметра организации работы приложений в сети:
  - Способ разделения приложения на части, выполняющиеся на разных компьютерах сети;
  - Выделение специализированных серверов в сети, на которых выполняются некоторые общие для всех приложений функции;
  - Способ взаимодействия между частями приложений, работающих на разных компьютерах.

# Способы разделения приложений на части

---

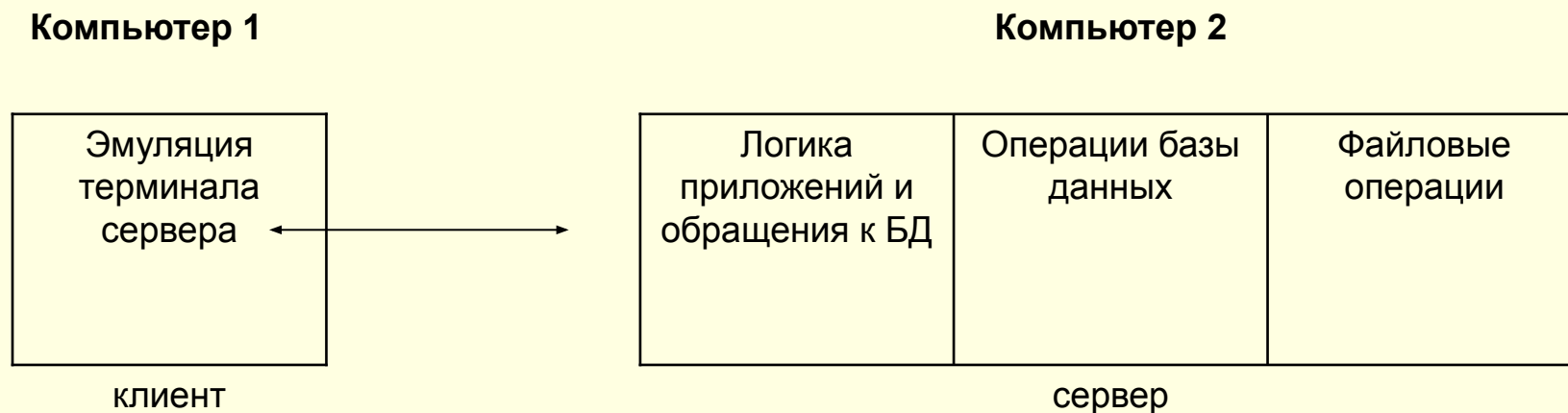
- Приложения условно можно разделить на следующие функциональные части:
  - Средства представления данных на экране;
  - Логика представления данных на экране (описывает правила и сценарии взаимодействия пользователя с приложениями);
  - Прикладная логика (правила для принятия решений, вычислительные процедуры и т.п.);
  - Логика данных – операции с данными, хранящимися в некоторой базе;
  - Внутренние операции БД – действия СУБД, вызываемые в ответ на выполнение запросов логики данных;
  - Файловые операции – стандартные операции над файлами и файловой системой.

# Двухзвенные схемы

---

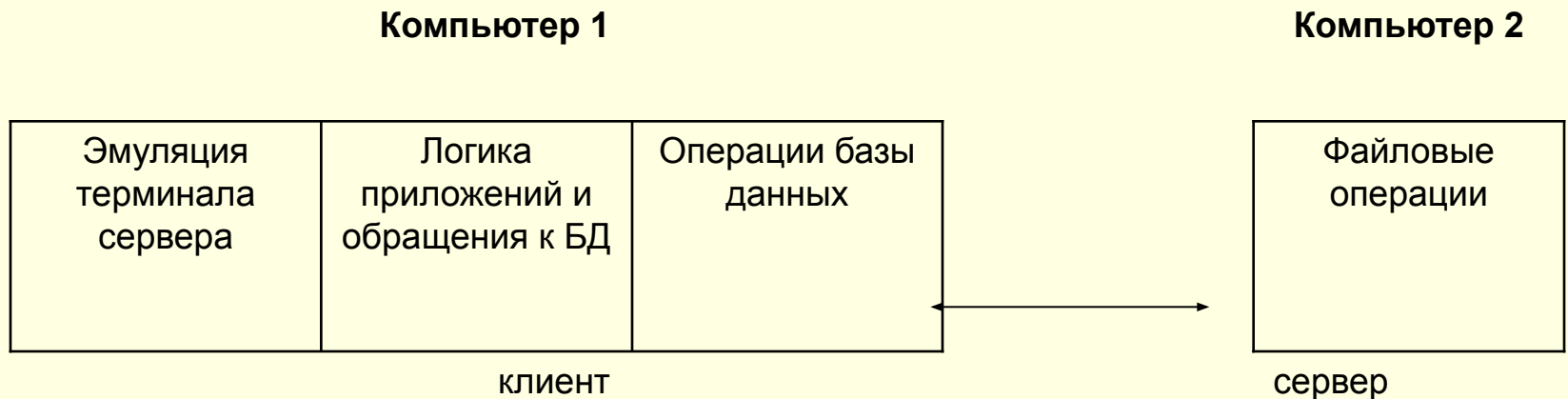
- Двухзвенные схемы описывают разделение функций приложения между двумя компьютерами:
  - Централизованная обработка данных;
  - Схема «файл-сервер»
  - Схема «клиент-сервер»

# Централизованная обработка данных



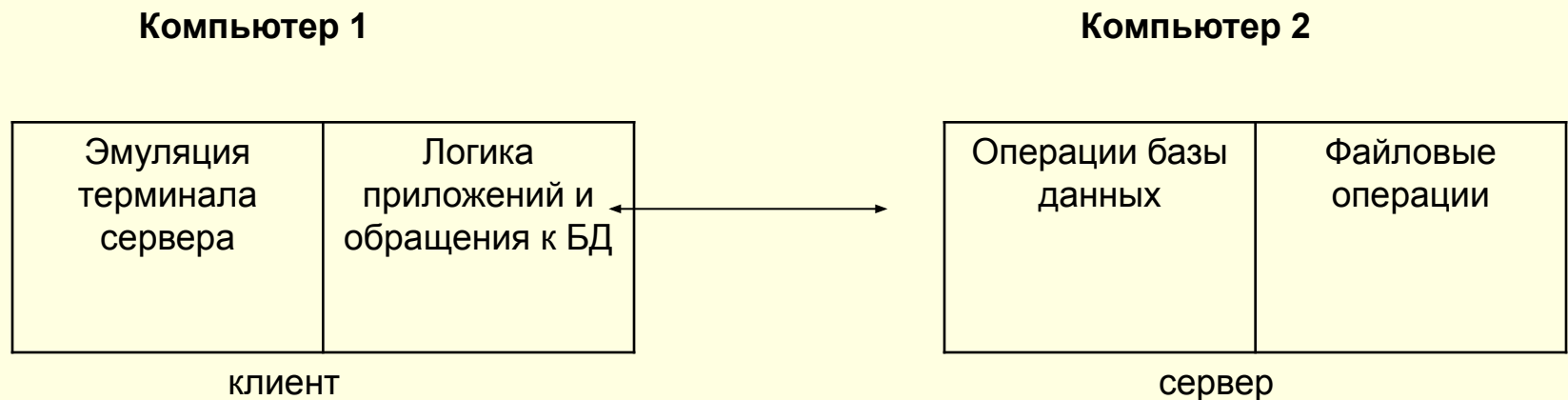
- Достоинства схемы:
  - Ресурсы клиентского компьютера используются в незначительной степени, загружаются только графические средства ввода-вывода;
  - Простота организации программы;
- Недостатки схемы:
  - Недостаточная масштабируемость;
  - Отсутствие отказоустойчивости.

# Схема «файл-сервер»



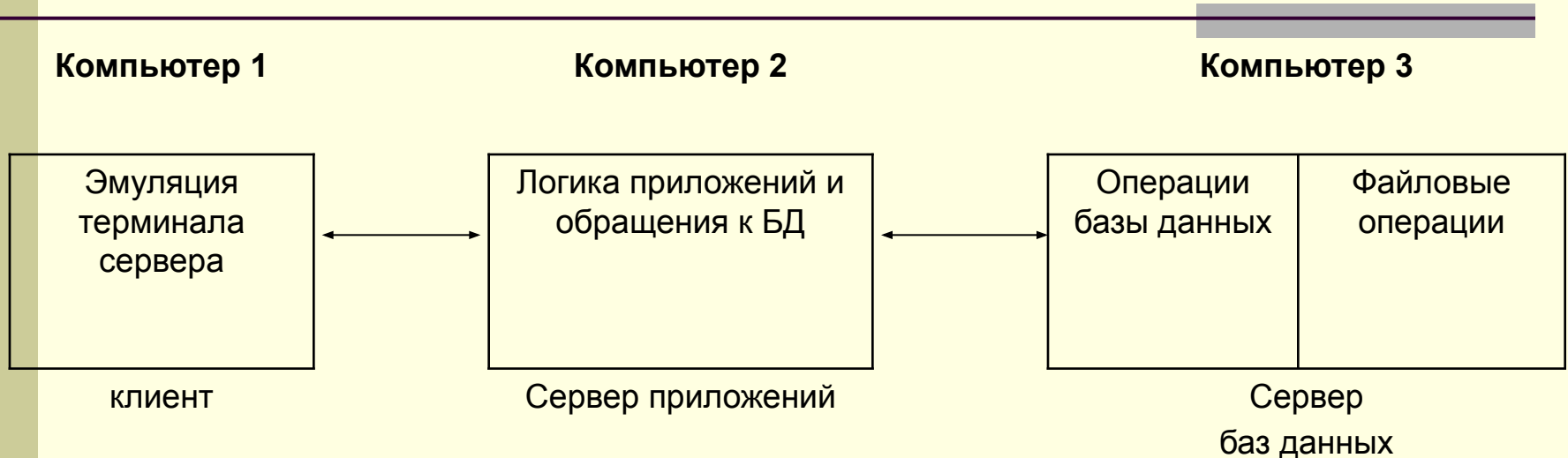
- **Достоинства схемы:**
  - Данная схема обладает хорошей масштабируемостью, поскольку дополнительные пользователи и приложения добавляют лишь незначительную нагрузку на центральный узел – файловый сервер.
- **Недостатки схемы:**
  - Во многих случаях возрастает нагрузка, что приводит к увеличению времени реакции на приложения;
  - Клиентский компьютер должен обладать высокой вычислительной мощностью, чтобы справляться с представлением данных, логикой приложений, логикой данных и поддержкой операции БД

# Схема «клиент-сервер»



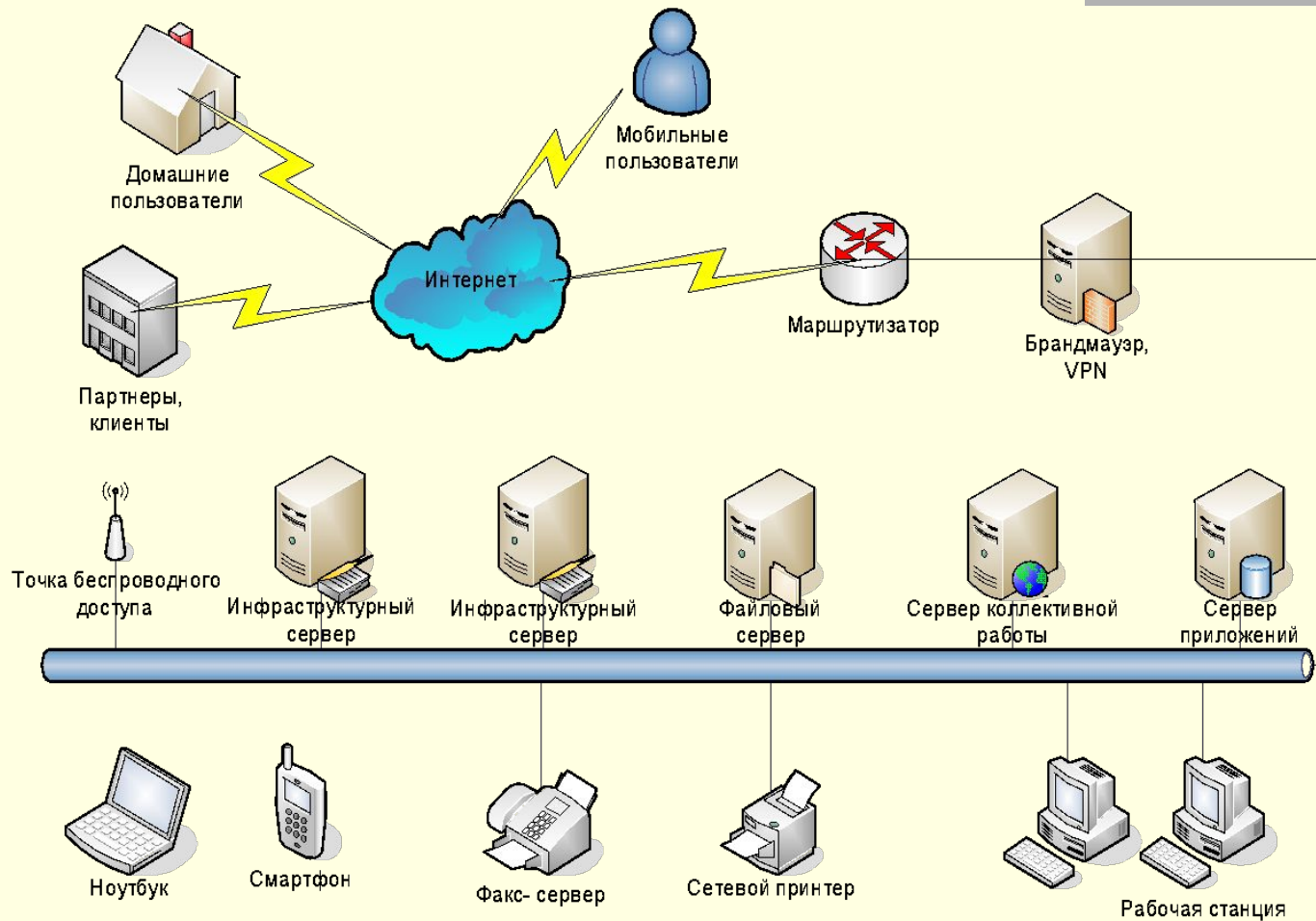
- Достоинства схемы:
  - Данная схема более равномерно распределяет функции между клиентской и серверной частями системы;
- Клиентский компьютер выполняет функции, специфические для данного приложения;
- Сервер – функции, реализация которых не зависит от специфики приложения, и данные функции могут быть оформлены в виде сетевых служб.

# Трехзвенные схемы



- Централизованная реализация логики приложения решает проблему недостаточной вычислительной мощности клиентских компьютеров для сложных приложений, упрощает администрирование и поддержку системы;
- Упрощается разработка крупных приложений, поскольку четко разделены платформы и инструменты для реализации интерфейса и прикладной логики.

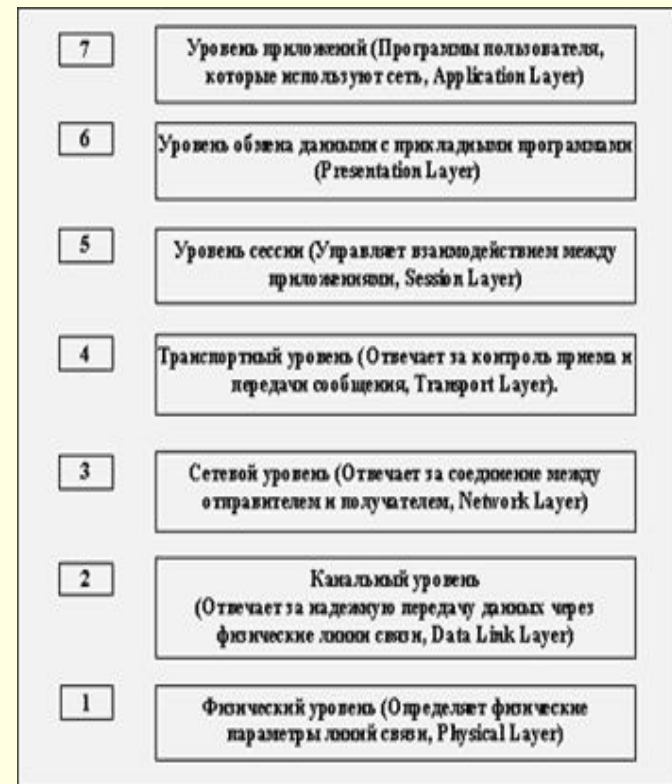
# Типовая сетевая инфраструктура современного предприятия





# Основы межсетевого обмена в сетях TCP/IP

- При рассмотрении процедур межсетевого взаимодействия всегда опираются на стандарты, разработанные *International Standard Organization (ISO)*.
- Данные стандарты называются "Семиуровневой модели сетевого обмена" или в английском варианте "*Open System Interconnection Reference Model*" (*OSI Ref. Model*).
- В модели OSI обмен информацией может быть представлен в виде стека, представленного на рисунке.
- В рамках данной модели определяется все аспекты соединения – от стандарта физического соединения сетей до протоколов обмена прикладного программного обеспечения.



# Уровни в модели OSI

- *Физический уровень* модели определяет характеристики физической сети передачи данных, которая используется для межсетевого обмена. Это такие параметры, как: напряжение в сети, сила тока, число контактов на разъемах и т.п. Типичными стандартами этого уровня являются, например RS232C, V35, IEEE 802.3 и т.п.
- *Канальный уровень* включает протоколы, определяющие соединение, например, SLIP (Strial Line Internet Protocol), PPP (Point to Point Protocol), NDIS, пакетный протокол, ODI и т.п. Речь идет о протоколе взаимодействия между драйверами устройств и устройствами, с одной стороны, а с другой стороны, между операционной системой и драйверами устройства. Драйвер – представляет собой конвертор данных из одного формата в другой, но при этом он может иметь и свой внутренний формат данных.
- К *сетевому (межсетевому) уровню* относятся протоколы, которые отвечают за отправку и получение данных, или, другими словами, за соединение отправителя и получателя. К данному уровню в TCP/IP относят протокол IP (Internet Protocol). На данном уровне определяется отправитель и получатель, именно здесь находится необходимая информация для доставки пакета по сети.

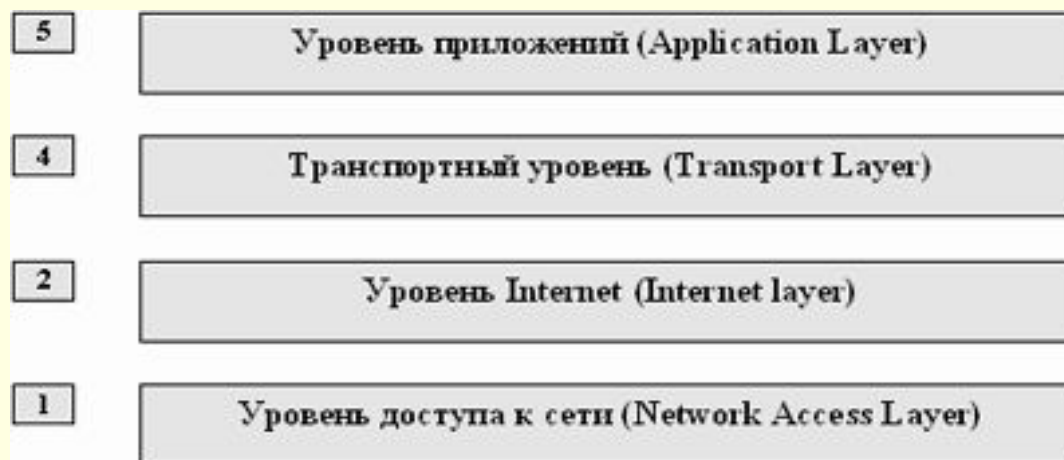
# Уровни в модели OSI

---

- *Транспортный уровень* отвечает за надежность доставки данных, и здесь, проверяя контрольные суммы, принимается решение о сборке сообщения в одно целое.
- *Уровень сессии* определяет стандарты взаимодействия между собой прикладного программного обеспечения. Это может быть некоторый промежуточный стандарт данных или правила обработки информации. Условно к этому уровню можно отнести механизм портов протоколов TCP и UDP и Berkeley Sockets.
- *Уровень обмена данными с прикладными программами (Presentation Layer)* необходим для преобразования данных из промежуточного формата сессии в формат данных приложения.
- *Уровень прикладных программ или приложений* определяет протоколы обмена данными этих прикладных программ.

# Структура стека протоколов TCP/IP

- Стек протоколов TCP/IP отличается от стека в модели OSI.
- Структура стека может быть представлена в виде четырех уровней.



# Сопоставление структуры семейства протоколов TCP/IP с моделью OSI

- Схема модулей, реализующих протоколы семейства TCP/IP в узле сети

