

# **Программная структура систем административного управления**

Лекция 7

Администрирование ИС

# Что понимается под открытыми системами?

Применение подхода **открытых систем** в настоящее время является основной тенденцией в области информационных технологий и средств вычислительной техники, поддерживающих эти технологии.

Для рассмотрения этого вопроса воспользуемся **определениями открытых систем**, которые приведены в руководстве, изданном Французской ассоциацией пользователей UNIX (AFUU) в 1992 году.

*"Открытая система - это система, которая состоит из компонентов, взаимодействующих друг с другом через стандартные интерфейсы".*

Это определение, данное одним из авторов упомянутого руководства Жаном-Мишелем Корну, подчеркивает системный аспект (структуру открытой системы).

*"Исчерпывающий и согласованный набор международных стандартов информационных технологий и профилей функциональных стандартов, которые специфицируют интерфейсы, службы и поддерживающие форматы, чтобы обеспечить интероперабельность и мобильность приложений, данных и персонала".*

Это **определение**, данное специалистами IEEE, подчеркивает аспект среды, которую предоставляет открытая система для ее использования (внешнее описание открытой системы).

# Открытая система

**Открытая система** – система реализующая открытые спецификации на интерфейсы, службы и форматы данных достаточные для того чтобы обеспечить:

1. Возможность переноса (*мобильность*) прикладных систем, разработанных с минимальными изменениями на широкий диапазон систем.
2. Совместную работу (*интероперабельность*) с другими прикладными системами на локальных и удаленных платформах.
3. Взаимодействие с пользователями в стиле облегчающем переход от системы к системе.

# Общие свойства открытых ИС:

- **расширяемость/масштабируемость** - обеспечение возможности добавления новых функций ИС или изменения некоторых уже имеющихся при неизменных остальных функциональных частях ИС;
- **мобильность/переносимость** - обеспечение возможности переноса программ и данных при модернизации или замене аппаратных платформ ИС и возможности работы с ними специалистов, пользующихся ИТ, без их переподготовки при изменениях ИС;
- **взаимодействие** - способность к взаимодействию с другими ИС (технические средства, на которых реализована информационная система, объединяются сетью или сетями различного уровня - от локальной до глобальной);

## Общие свойства открытых ИС:

- **стандартизуемость** - ИС проектируются и разрабатываются на основе согласованных *международных стандартов и предложений*, реализация открытости осуществляется на базе функциональных стандартов (профилей) в области информационных технологий;
- **дружественность к пользователю** - развитые *унифицированные интерфейсы* в процессах взаимодействия в системе "человек-машина" позволяют работать пользователю, не имеющему специальной "компьютерной" подготовки.

# Структура среды информационной системы

Обобщенная структура любой ИС может быть представлена двумя взаимодействующими частями:

1. **функциональная часть**, включающая прикладные программы, которые реализуют функции прикладной области;
2. **среда или системная часть**, обеспечивающая исполнение прикладных программ.

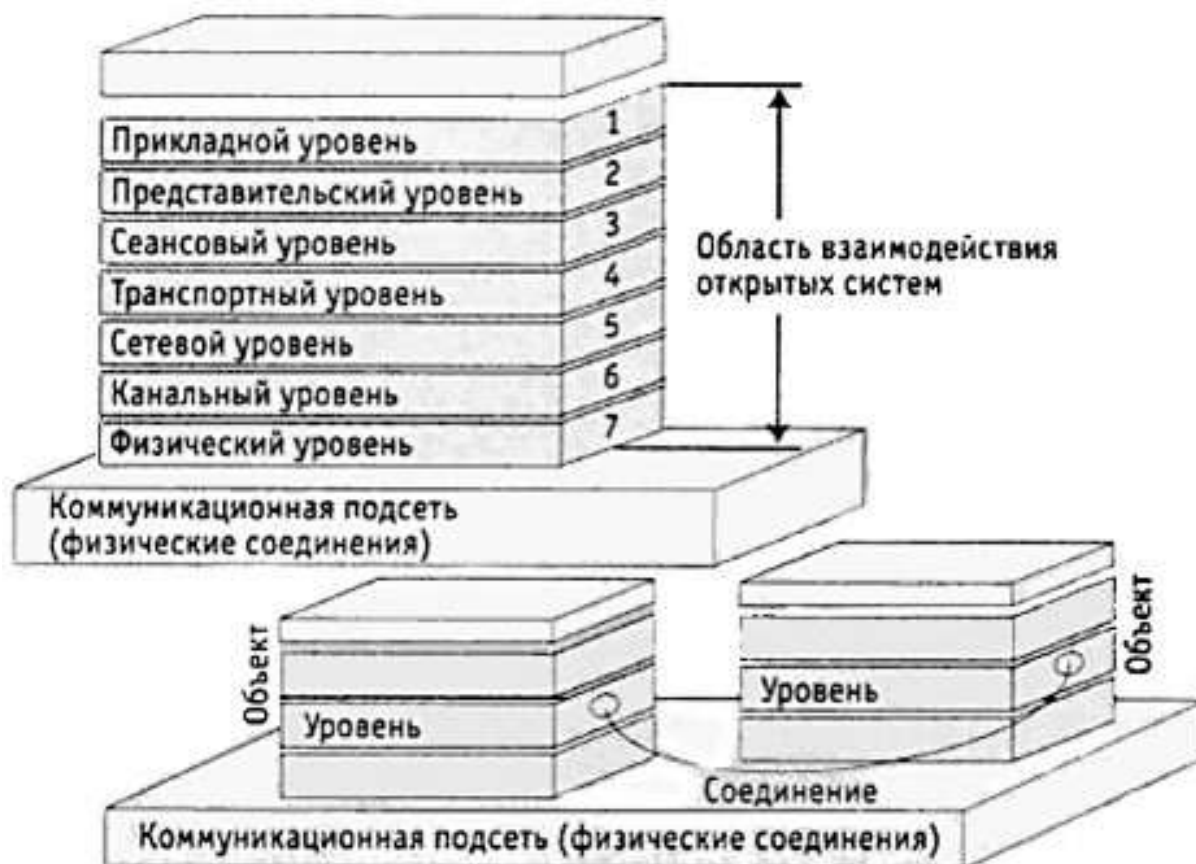
С этим разделением тесно связаны **две группы вопросов стандартизации**:

1. стандарты интерфейсов взаимодействия прикладных программ со средой ИС, *прикладной программный интерфейс (Application Program Interface - API)*;
2. стандарты интерфейсов взаимодействия самой ИС с внешней для нее средой (*External Environment Interface - EEI*).

Эти две группы интерфейсов определяют спецификации внешнего описания среды ИС - **архитектуру**, с точки зрения *конечного пользователя, проектировщика ИС, прикладного программиста*, разрабатывающего функциональные части ИС.

**Спецификации внешних интерфейсов среды ИС** и, как будет видно далее, **спецификации интерфейсов взаимодействия между компонентами самой среды**, - это точные описания всех необходимых функций, служб и форматов определенного интерфейса. Совокупность таких описаний составляет **эталонную модель открытых систем (Reference Open System Model)**.

Эта модель используется более 30 лет и определяется системной сетевой архитектурой (SNA), предложенной IBM в 1974 году. Она основана на **разбиении вычислительной среды на семь уровней**, взаимодействие между которыми описывается соответствующими стандартами и обеспечивает связь уровней *вне зависимости от построения уровня в каждой конкретной реализации.*



# **Эталонная модель среды открытых систем**

**Основным достоинством этой модели** является *детальное описание связей в среде с точки зрения технических устройств и коммуникационных взаимодействий*. Вместе с тем она не принимает в расчет взаимосвязь с учетом мобильности прикладного программного обеспечения.

**Эталонная модель среды открытых систем (OSE/RM)** определяет разделение любой информационной системы **на приложения** (прикладные программы и программные комплексы) **и среду**, в которой эти приложения функционируют.

**Между приложениями и средой** определяются **стандартизированные интерфейсы (API)**, которые являются необходимой частью профилей любой *открытой системы*.

Кроме того, в профилях ИС могут быть определены унифицированные интерфейсы взаимодействия функциональных частей друг с другом и интерфейсы взаимодействия между компонентами среды ИС.



**В стандарте ISO 7498 выделено семь уровней (слоев) информационного взаимодействия, которые отделены друг от друга стандартными интерфейсами:**

- 1. уровень приложения (прикладной уровень)*
- 2. уровень представления*
- 3. сеансовый (уровень сессии)*
- 4. транспортный*
- 5. сетевой*
- 6. канальный*
- 7. физический.*

В соответствии с этим, информационное взаимодействие двух или более систем представляет собой **совокупность информационных взаимодействий** **уровневых подсистем**, причем **каждый слой локальной информационной системы взаимодействует, как правило, с соответствующим слоем удаленной системы.** Взаимодействие осуществляется при помощи соответствующих протоколов связи и интерфейсов. Кроме того, применяя методы инкапсуляции, можно использовать одни и те же программные модули на различных уровнях.

- **Протоколом** является набор алгоритмов (правил) взаимодействия объектов одноименных уровней различных систем.
- **Интерфейс** - это совокупность правил, в соответствии с которыми осуществляется взаимодействие с объектом данного или другого уровня. Стандартный интерфейс в некоторых спецификациях может называться услугой.
- **Инкапсуляция** - это процесс помещения фрагментированных блоков данных одного уровня в блоки данных другого уровня.

Каждый уровень имеет протокольную спецификацию, т.е. набор правил, управляющих взаимодействием равноправных процессов одного и того же уровня, и перечень услуг, которые описывают стандартный интерфейс с расположенным выше уровнем.

Каждый уровень использует услуги расположенного ниже уровня, каждый расположенный ниже предоставляет услуги расположенному выше.

**Уровень 1 - уровень приложения или прикладной уровень (Application Layer).**

Этот уровень связан с **прикладными процессами**.

Протоколы уровня предназначены для **обеспечения доступа к ресурсам сети и программам-приложениям пользователя**.

На данном уровне **определяется интерфейс с коммуникационной частью приложений**.

*В качестве примера протоколов прикладного уровня можно привести протокол Telnet, который обеспечивает доступ пользователя к "хосту" (главному вычислительному устройству, одному из основных элементов в многомашинной системе или любому устройству, подключенному к сети и использующему протоколы TCP/IP) в режиме удаленного терминала.*

Прикладной уровень выполняет задачу обеспечения различных форм взаимодействия прикладных процессов, расположенных в разнообразных системах информационной сети.

## **Функции прикладного уровня:**

- *описание форм и методов взаимодействия прикладных процессов;*
- *выполнение различных видов работ (управление заданиями, передача файлов, управление системой и т. д.);*
- *идентификацию пользователей по их паролям, адресам, электронным подписям;*
- *определение функционирующих абонентов;*
- *объявление о возможности доступа к новым прикладным процессам;*
- *определение достаточности имеющихся ресурсов;*
- *посылку запросов на соединение с другими прикладными процессами;*
- *подачу заявок представителю уровня на необходимые методы описания информации;*
- *выбор процедур планируемого диалога процессов;*
- *управление данными, которыми обмениваются прикладные процессы;*
- *синхронизацию взаимодействия прикладных процессов;*
- *определение качества обслуживания (время доставки блоков данных, допустимой частоты ошибок и т. д.);*
- *соглашение об исправлении ошибок и определении достоверности данных;*
- *согласование ограничений, накладываемых на синтаксис (наборы символов, структура данных).*

Прикладной уровень часто делится на **два подуровня**.

**Верхний** подуровень включает **сетевые службы**.

**Нижний** - содержит **стандартные сервисные элементы**, поддерживающие работу сетевых служб.

## **Уровень 2 - уровень представления (Presentation Layer).**

На этом уровне **информация преобразуется к такому виду, в каком это требуется для выполнения прикладных процессов.**

Уровень представления обеспечивает *кодирование данных, выдаваемых прикладными процессами, и интерпретацию передаваемых данных.*

*Например, выполняются алгоритмы преобразования формата представления данных для печати - ASCII или КОИ-8. Или, если для визуализации данных используется дисплей, то эти данные по заданному алгоритму формируются в виде страницы, которая выводится на экран.*

**Представительный уровень выполняет следующие основные функции:**

- *выбор образа представлений из возможных вариантов;*
- *изменение образа представления в заданный виртуальный образ;*
- *преобразование синтаксиса данных (кодов, символов) в стандартный;*
- *определение формата данных.*

### **Уровень 3 - сеансовый уровень или уровень сессии (Session Layer).**

На данном уровне **устанавливаются, обслуживаются и прекращаются сессии между представительными объектами приложений (прикладными процессами).**

*В качестве примера протокола сеансового уровня можно рассмотреть протокол RPC (Remote Procedure Call).*

Как следует из названия, данный протокол предназначен для **отображения результатов выполнения процедуры на удаленном хосте.**

В процессе выполнения этой процедуры между приложениями **устанавливается сеансовое соединение.** Назначением данного соединения является обслуживание запросов, которые возникают, например, при взаимодействии приложения-сервера с приложением-клиентом.

**Сеансовый уровень** обеспечивает взаимодействие с транспортным уровнем, координирует прием и передачу данных одного сеанса связи, содержит функции управления паролями, подсчета платы за использование ресурсов сети и т.д.

**Этот уровень обеспечивает выполнение следующих функций:**

- *установление и завершение на сеансовом уровне соединения между партнерами;*
- *выполнение нормального и срочного обмена данными между прикладными процессами;*
- *синхронизация работы сеансовых соединений;*
- *извещение прикладных процессов об исключительных ситуациях;*
- *установление в прикладном процессе меток, позволяющих после отказа либо ошибки восстановить его выполнение от ближайшей метки;*
- *прерывание в нужных случаях прикладного процесса и его корректное возобновление;*
- *прекращение сеанса без потери данных;*
- *передачу особых сообщений о ходе проведения сеанса.*

## **Уровень 4 - транспортный уровень (Transport Layer).**

Транспортный уровень *предназначен для управления потоками сообщений и сигналов.*

**Управление потоком** является важной функцией транспортных протоколов, поскольку этот механизм позволяет *надёжно обеспечивать передачу данных по сетям с разнородной структурой*, при этом в описание маршрута включаются все компоненты коммуникационной системы, обеспечивающие передачу данных на всем пути от устройств отправителя до приемных устройств получателя.

**Управление потоком** заключается в *обязательном ожидании передатчиком подтверждения приема обусловленного числа сегментов приемником.* Количество сегментов, которое передатчик может отправить без подтверждения их получения от приемника, называется *окном.*

**Существует два типа протоколов транспортного уровня** - *сегментирующие протоколы и дейтаграммные протоколы.*

**Сегментирующие протоколы** транспортного уровня *разбивают исходное сообщение на блоки данных транспортного уровня - сегменты.* Основной функцией таких протоколов является обеспечение доставки этих сегментов до объекта назначения и восстановление сообщения.

**Дейтаграммные протоколы** не сегментируют сообщение, *они отправляют его одним пакетом вместе с адресной информацией.* Пакет данных, который называется "дейтаграмма" (Datagram), маршрутизируется в сетях с переключением адресов или передается по локальной сети прикладной программе или пользователю.



**На транспортном уровне** может выполняться также согласование сетевых уровней различных несовместимых сетей через специальные шлюзы.

Рассматриваемый уровень определяет адресацию абонентских систем и административных систем.

Главной задачей транспортного уровня является использование виртуальных каналов, проложенных между взаимодействующими абонентскими системами и административными системами, для передачи в пакетах блоков данных.

**Основные функции, выполняемые транспортным уровнем:**

- управление передачей блоков данных и обеспечение их целостности;
- обнаружение ошибок, их частичная ликвидация, сообщение о неисправленных ошибках;
- восстановление передачи после отказов и неисправностей;
- укрупнение либо разукрупнение блоков данных;
- предоставление приоритетов при передаче блоков;
- передача подтверждений о переданных блоках данных;
- ликвидация блоков при тупиковых ситуациях в сети.

Кроме этого, **транспортный уровень может восстанавливать блоки данных, потерянные на нижних уровнях.**

## Уровень 5 - сетевой уровень (Network Layer).

Основной задачей протоколов сетевого уровня является *определение пути, который будет использован для доставки пакетов данных при работе протоколов верхних уровней (маршрутизация)*.

Для того чтобы пакет был доставлен до какого-либо заданного хоста, этому хосту должен быть поставлен в соответствие известный передатчику сетевой адрес.

Группы хостов, объединенные по территориальному принципу, образуют сети.

Для упрощения задачи маршрутизации **сетевой адрес хоста состоит из двух частей: адреса сети и адреса хоста.**

Таким образом, задача маршрутизации распадается на две - **поиск сети и поиск хоста в этой сети.**

**На сетевом уровне могут выполняться следующие функции:**

- *создание сетевых соединений и идентификация их портов;*
- *обнаружение и исправлений ошибок, возникающих при передачи через коммуникационную сеть;*
- *управление потоками пакетов;*
- *организация (упорядочение) последовательностей пакетов;*
- *маршрутизация и коммутация;*
- *сегментация и объединение пакетов;*
- *возврат в исходное состояние;*
- *выбор видов сервиса.*

## **Уровень 6 - канальный уровень или уровень звена данных (Data Link Layer).**

Назначением протоколов канального уровня является **обеспечение передачи данных в среде передачи по физическому носителю.**

*В канале формируется стартовый сигнал передачи данных, организуется начало передачи, производится сама передача, проводится проверка правильности процесса, производится отключение канала при сбоях и восстановление после ликвидации неисправности, формирование сигнала на окончание передачи и перевода канала в ждущий режим.*

**Таким образом, канальный уровень может выполнять следующие функции:**

- организацию (установление, управление, расторжение) канальных соединений и идентификацию их портов;*
- передачу блоков данных;*
- обнаружение и исправление ошибок;*
- управление потоками данных;*
- обеспечение прозрачности логических каналов (передачи по ним данных, закодированных любым способом).*

**На канальном уровне** данные передаются в виде блоков, которые называются **кадрами**.

Тип используемой среды передачи и её топология во многом определяют *вид кадра протокола транспортного уровня*, который должен быть использован.

*При использовании топологии "общая шина" (Common Bus) и "один-ко-многим" (Point-to-Multipoint) средства протокола канального уровня задают физические адреса, с помощью которых будет производиться обмен данными в среде передачи и процедура доступа к этой среде.*

Примерами таких протоколов являются протоколы Ethernet (в соответствующей части) и HDLC.

Протоколы транспортного уровня, которые предназначены для работы в среде типа "один-к-одному" (Point-to-Point), не определяют физических адресов и имеют упрощенную процедуру доступа. Примером протокола такого типа является протокол PPP.

## **Уровень 7 - физический уровень (Physical Layer).**

Протоколы физического уровня обеспечивают *непосредственный доступ к среде передачи данных для протоколов канального и последующих уровней.*

**Данные передаются с помощью протоколов данного уровня в виде последовательностей битов** (для последовательных протоколов) или групп битов (для параллельных протоколов).

На этом уровне *определяются набор сигналов, которыми обмениваются системы, параметры этих сигналов (временные и электрические) и последовательность формирования сигналов при выполнении процедуры передачи данных.*

**Физический уровень выполняет следующие функции:**

- *устанавливает и разъединяет физические соединения;*
- *передает последовательность сигналов;*
- *"прослушивает" в нужных случаях каналы;*
- *выполняет идентификацию каналов;*
- *оповещает о появлении неисправностей и отказов.*

Кроме того, на данном уровне **формулируются требования к электрическим, физическим и механическим характеристикам среды передачи, передающих и соединительных устройств.**

## **Сетезависимые и сетенезависимые уровни.**

Указанные выше функции всех уровней можно отнести к **одной из двух групп**:

1. к функциям, ориентированным на работу с приложениями вне зависимости от устройства сети,
2. к функциям, зависящим от конкретной технической реализации сети.

**Три верхних уровня - прикладной, представительный и сеансовый** ориентированы на приложения и практически **не зависят** от технических особенностей построения сети. На протоколы этих уровней не влияют какие-либо изменения в топологии сети, замена оборудования или переход на другую сетевую технологию.

**Три нижних уровня - физический, канальный и сетевой** являются **сетезависимыми**, то есть протоколы этих уровней тесно связаны с технической реализацией сети и используемым коммуникационным оборудованием.

**Транспортный уровень является промежуточным, он скрывает все детали функционирования нижних уровней от верхних.** Это позволяет разрабатывать приложения, не зависящие от технических средств непосредственной транспортировки сообщений.

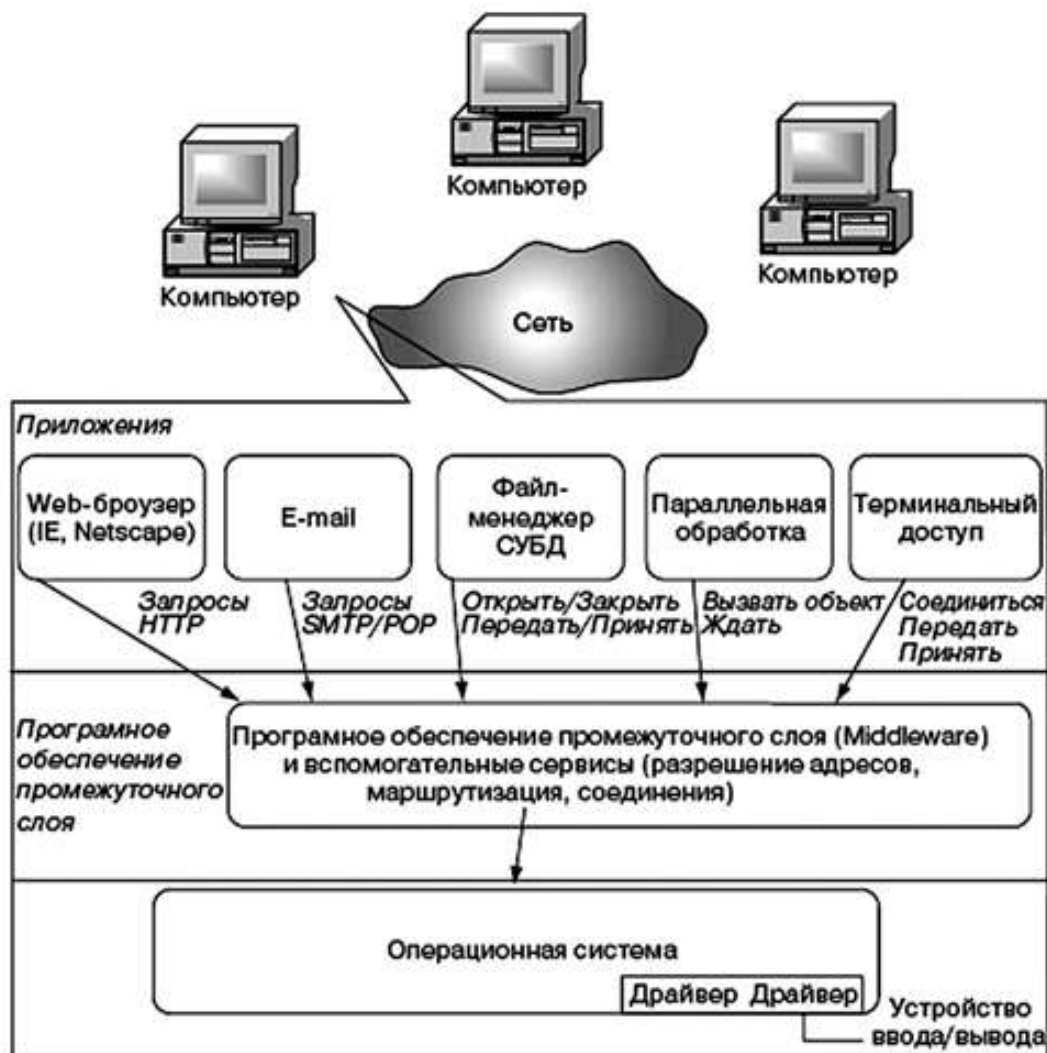
**В некоторых случаях семь уровней модели OSI объединяют в три группы:**

1. группа "утилит и приложений" - 1-й уровень;
2. группа "преобразования информации" - с 2-го по 5-й уровни;
3. группа "физическая" - 6-ый и 7-ой уровни.

# **Взаимодействие открытых систем (ВОС)**

Взаимодействие открытых систем (ВОС) – совокупность принципов взаимодействия, которые организованы в соответствии с международной организацией стандартов.

# Схема практической реализации модели ВОС (OSI), в которой работают различные элементы сети.





Компьютер с установленной на нем сетевой операционной системой взаимодействует с другим компьютером с помощью протоколов всех семи уровней.



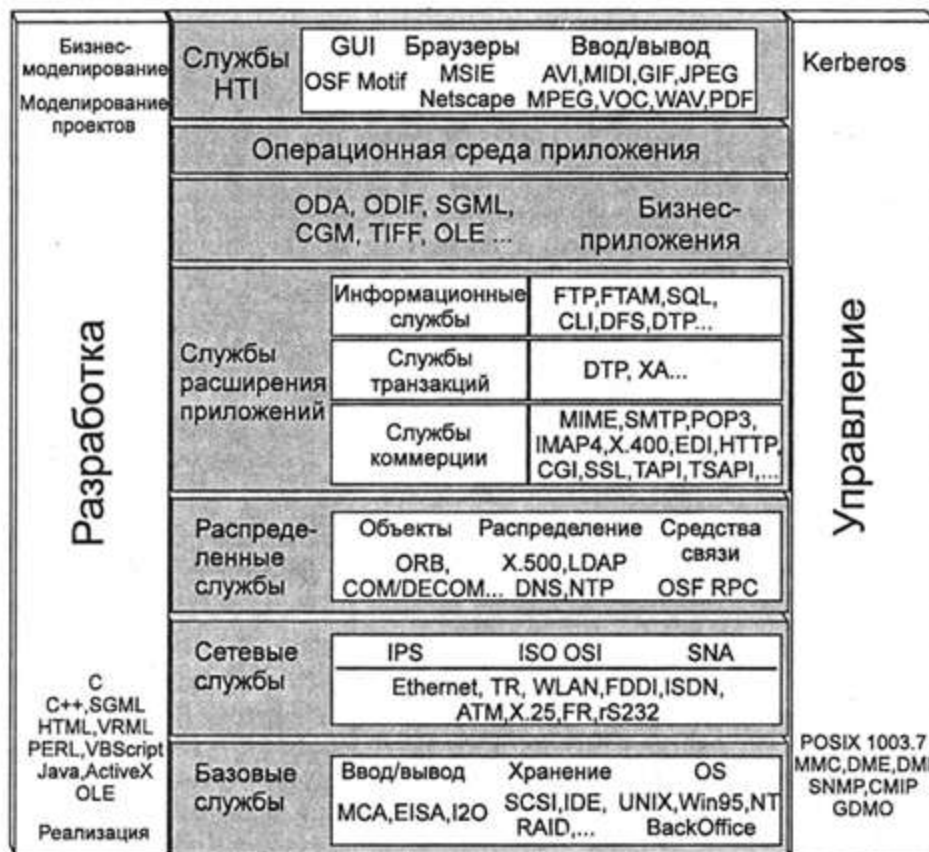
Это взаимодействие компьютеры осуществляют через различные коммуникационные устройства: *концентраторы, модемы, мосты, коммутаторы, маршрутизаторы, мультиплексоры.*

**Коммуникационное устройство в зависимости от типа может работать:**

- только на физическом уровне (повторитель);
- на физическом и канальном уровнях (мост);
- на физическом, канальном и сетевом уровнях, иногда захватывая и транспортный уровень (маршрутизатор).

**Эталонная модель взаимосвязи (взаимодействия) открытых систем (OSI)** описывает и реализует стандартизованную систему взаимодействия в процессах обмена информацией и данными между прикладными программами и системами в вычислительных сетях.

Для реализации конкретных вычислительных, информационных и коммуникационных процессов и процедур на базе *моделей среды открытых систем (OSE)* и *взаимодействия открытых систем (OSI)* создаются разнообразные **прикладные службы**.



**Стандартизация интерфейсов** обеспечивает полную прозрачность взаимодействия вне зависимости от того, каким образом устроены уровни в конкретных реализациях (службах) модели.

## **Управление взаимосвязью открытых систем**

– наблюдение, контроль и координация функции ресурсов, обеспечивающих связь прикладных процессов в среде ВОС

*(тестирование работоспособности и рабочих параметров, аппаратных и программных средств,*

*выявление и регистрация неисправностей,*

*функции сбора статистики,*

*обеспечение ИБЗИ).*

**Процесс (process)** – последовательная смена состояний, явлений, ход развития чего последовательность действий для решения задачи, определяемая программой.

Процессы вычисляются по соответствующим алгоритмам программами или устройствами. *Они реализуются на всех уровнях области взаимодействия.*

**Процессы разделяются на этапы, называемые фазами.**

Любая система создается для выполнения прикладных процессов. Они для информационной сети являются основными.

**Прикладной процесс (application process)** – процесс, выполняющий обработку данных для нужд пользователей или некоторое приложение пользователя, реализованное в прикладной программе. Взаимодействие абонентских ЭВМ в сети следует рассматривать как взаимодействие прикладных процессов конечных пользователей через коммуникационную сеть. Для того чтобы могли взаимодействовать процессы, между ними должна существовать и логическая связь.

*В базовой эталонной модели взаимодействия открытых систем (ВОС – OSI – Open Systems Interconnection) прикладные процессы располагаются над прикладным уровнем и выполняются под управлением операционной системы (ОС). Их запуск осуществляется операцией вызова процедуры.*

## ***Классификация прикладных процессов (ПП):***

- 1. Программные ПП** определяются прикладной программой, либо группой взаимосвязанных ПП.
- 2. Человеко-аппаратные ПП** (Обработка информации осуществляется не программой а человеком).

*В зависимости от целей:*

1. Прикладные процессы пользователей.
2. ПП управления, для поддержания сетей и систем.

При разработке **прикладного процесса** необходимо учесть ряд **следующих особенностей**:

- *синхронизация данных;*
- *тупиковые ситуации;*
- *безопасность данных.*

При выполнении прикладного процесса возникают ситуации, именуемые **«тупиковыми»**. Тупиковую ситуацию можно предупреждать двояко. Если хотя бы один из ресурсов занят, то освободить остальные ресурсы. Можно также однозначно определить резервирование ресурсов.

**Управление взаимодействием прикладных процессов** относится к процедурам передачи данных между системами, которые «открыты» друг другу благодаря совместному использованию ими соответствующих стандартов. Сложность функций области взаимодействия привели к тому, что они в соответствии с базовой эталонной моделью взаимодействия открытых систем поделены на расположенные друг над другом **слои, именуемые уровнями**.

Любой уровень состоит из **объектов**. **Объекты** одного и того же уровня для обеспечения взаимодействия могут связываться друг с другом соединениями (**интерфейсом**). Последние проходят все нижерасположенные уровни и физические средства соединения.

Взаимодействие объектов, расположенных на одном и том же уровне, определяется стандартами, называемыми **«протоколами»**.

Взаимодействие открытых систем стало основной концепцией, заложенной в архитектуру информационных сетей.

**Передача данных** (data communications) – процесс пересылки данных от одного устройства к другому.

**Смежными системами** являются те, которые непосредственно соединены **каналом передачи данных** либо группой параллельно идущих каналов.

Если же между взаимодействующими системами расположен хотя бы один узел коммутации, то говорят о несмежных системах. **В сетях с «селекцией» информации** смежными являются все входящие в них системы.

**В сетях с коммутацией пакетов** для передачи данных между не смежными системами создается **сетевая платформа**, образуемая физическими средствами соединения: физическим уровнем, канальным уровнем и сетевым уровнем.

Передача данных между смежными системами осуществляется **физической платформой**.



# Каналы передачи данных

- *низкоскоростные;*
- *среднескоростные;*
- *высокоскоростные.*

С учетом **возможностей изменения направления передачи** информации различают каналы:

- *симплексные*, обеспечивающие передачу информации только в одном направлении по одному каналу связи;
- *полудуплексные*, позволяющие передавать поочередную информацию в двух направлениях по одному каналу связи;
- *дуплексные*, передающие информацию одновременно в обоих направлениях по двум каналам связи.

В зависимости **от способа передачи данных** различают каналы с *последовательной и параллельной* передачей сигналов.

*При последовательной передаче* двоичные разряды каждого символа передаются последовательно по одним и тем же линиям связи.

*При параллельной передаче* все разряды каждого символа передаются одновременно по отдельным линиям связи.

В системах передачи данных используются **два способа передачи данных**: *синхронный и асинхронный*. *Синхронная передача* – высокоскоростная и почти безошибочная. Она используется для обмена сообщениями между ЭВМ в вычислительных сетях. *При асинхронной передаче* данные передаются в канал связи как последовательность битов, из которой при приеме необходимо выделить для последующей их обработки.

**Управляемый объект** – это элемент составляющий ресурсы обработки и передачи данных (ресурсы ВОС и ресурсы сети), которым можно управлять согласно протоколам управления ВОС с помощью систем управления сетью.

**Ресурсы взаимосвязи открытых систем** – ресурсы обработки данных и обмена данными относящиеся к ВОС.

- **Служба ВОС** – совокупность аппаратных и/или программных средств входящая в состав открытой информационной сети.
- **Среда ВОС** – совокупность функций, которая дает возможность открытым системам осуществлять обмен данными через физические средства соединения между прикладными процессами с целью их взаимодействия в соответствии со стандартами международной организации стандартов.

# *Функции и иерархия управления ресурсами ВОС*

Модель ВОС выполняет координирующие действия по функционированию в сети различных ИС.

- 1. Взаимодействие прикладных процессов*
- 2. Формы предоставления данных*
- 3. Единообразное хранилище данных*
- 4. Управление сетевыми ресурсами*
- 5. Информационная безопасность и защита информации*
- 6. Диагностика программ и тех.средств*

Основные элементы модели взаимодействия открытых систем:

- 1. Уровень*
- 2. Объекты*
- 3. Соединения*
- 4. Физические средства соединения*

**Функциональные профили модели** – подмножества стандартов взаимодействия открытых систем, предназначенные для конкретных нужд пользователей.

При создании и развитии сложных, распределенных, тиражируемых программных и информационных систем требуется гибкое формирование и применение согласованных (гармонизированных) совокупностей базовых и рабочих стандартов, нормативных документов разного уровня, выделение в них требований и рекомендаций, необходимых для реализации заданных функций ИС.

Для унификации и регламентирования такие совокупности базовых стандартов должны адаптироваться и конкретизироваться применительно к определенным классам проектов, процессов функций и компонентов разрабатываемых систем.

В связи с этим выделилось и сформировалось понятие **профиля программной или информационной системы** как основного инструмента функциональной стандартизации.

**Профиль** - это "совокупность нескольких (или подмножество одного) базовых стандартов с четко определенными и гармонизированными подмножествами обязательных и рекомендуемых возможностей, предназначенная для реализации заданной функции или группы функций ИТ/ИС в конкретной функциональной среде. Функциональная характеристика объекта стандартизации является исходной позицией для формирования и применения профиля этого объекта или процесса".

Примерами такой среды могут быть *среда рабочей станции для управления встроенными вычислительными устройствами, распределенная среда передачи и обработки данных, среда офисного документооборота и т.д.*

Профиль включает набор согласованных между собой стандартов, охватывающих взаимодействие аппаратных и программных компонент системы, и определяет спецификации протоколов и интерфейсов, составляющих структуру **открытой системы**.

Для построения профиля структура системы разбивается, по крайней мере, на 3 уровня: **прикладные программы** (комплексы программ), **операционная среда** и **аппаратная платформа**, между которыми определяются правила взаимодействия. На каждом из этих уровней определяются типы взаимодействующих функциональных компонент в соответствии с принятой моделью открытых систем:

- *интерфейсы системы с пользователем и внешней средой;*
- *системы и языки программирования;*
- *управление прохождением задач;*
- *управление данными;*
- *коммуникации;*
- *обеспечение безопасности систем и данных и т.д.*

Основными целями применения профилей при создании и использовании ИС являются:

- *снижение трудоемкости и повышение связности проектов ИС;*
- *обеспечение переносимости прикладного программного обеспечения (ППО);*
- *обеспечение расширяемости ИС по набору прикладных функций и масштабируемости системы в целом и отдельных её частей;*
- *предоставление возможности функциональной интеграции в ИС задач, которые раньше решались раздельно и менее эффективно;*
- *повышение надежности и качества компонентов ИС.*

Выбор стандартов и документов для формирования конкретных профилей ИС зависит от того, какие из этих целей определены приоритетными.

В качестве общей методологической базы построения и применения профилей сложных распределенных ИС предлагается использовать **технический отчет ИСО/МЭК ТО 10000 (ISO/IEC TR 10000-1, ISO/IEC TR 10000-2, ISO/IEC TR 10000-3)**. Части 1 и 2 этого документа введены в России в качестве стандарта ГОСТ Р. Часть 3, определяющую основы и таксономию профилей среды открытых систем, предлагается задействовать при построении и использовании профилей ИС как документ прямого применения.