

Александр Александрович Олейников

Компьютерные и телекоммуникационные сети

Лекция 1.1.2.Связь компьютера с периферийными устройствами

Астрахань, 2018

Связь компьютера с периферийными устройствами

- Для организации связи между компьютером и периферийным устройством (ПУ) в обоих этих устройствах предусмотрены внешние физические интерфейсы.
- Интерфейс - в широком смысле - формально определенная логическая и физическая границы между взаимодействующими независимыми объектами. Интерфейс задает параметры, процедуры и характеристики взаимодействия объектов
- **Физический интерфейс** (называемый также **портом**) - определяется набором электрических связей и характеристиками сигналов.

- Обычно он представляет собой разъем с набором контактов, каждый из которых имеет определенное назначение, например, это может быть группа контактов для передачи данных, контакт синхронизации данных и т. п. Пара разъемов соединяется кабелем, который состоит из набора проводов, каждый из которых соединяет соответствующие контакты (рис. 2.1).

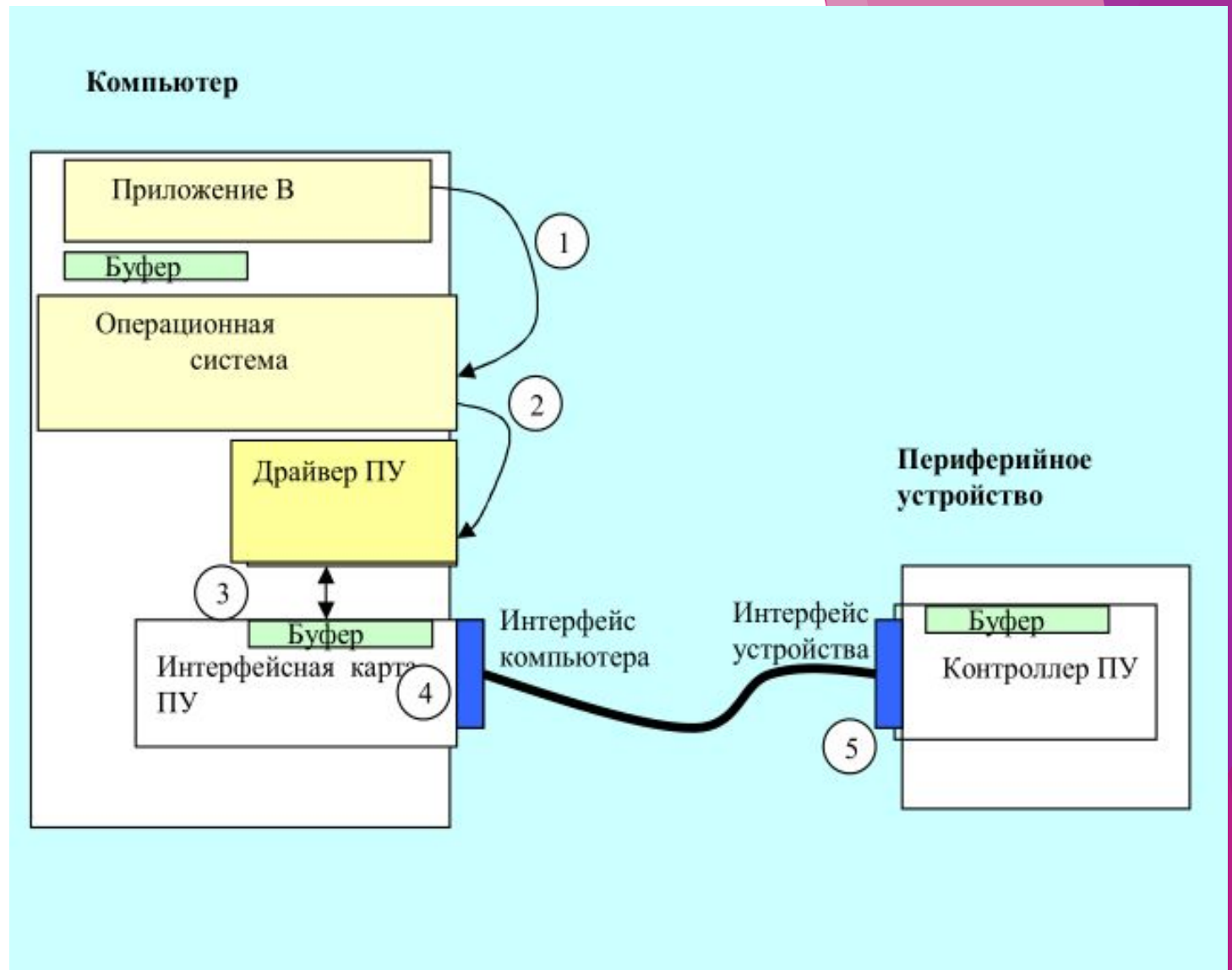


Рис.2.1. Взаимодействие компьютера с периферийным устройством. 1- запрос приложения к ОС, 2 – вызов драйвера, 3 – загрузка в буфер интерфейсной карты команды или данных, 4 – побитная передача информации в линию связи, 5 - прием битов и размещение их в буфере

- ▶ **Логический интерфейс** — это набор информационных сообщений определенного формата, которыми обмениваются два устройства или две программы, а также набор правил, определяющих логику обмена этими сообщениями.
- ▶ **Примерами стандартных интерфейсов**, используемых в компьютерах, являются параллельный (передающий данные байтами) интерфейс Centronics, предназначенный, как правило, для подключения принтеров, и последовательный интерфейс (передающий данные битами) RS-232C (известный также как COM-порт), который имеет более универсальное назначение — он поддерживается не только принтерами, но и графопостроителями, манипуляторами типа «мышь» и многими другими устройствами. Существуют также специализированные интерфейсы, которые предназначены для подключения уникальных периферийных устройств, например сложной физической экспериментальной установки. **В настоящее время широкое распространение нашел интерфейс USB (Universal Serial Bus - универсальная последовательная шина).**

- ▶ В компьютере операции интерфейса реализуются совокупностью аппаратных и программных средств: интерфейсной платой (аппаратное устройство) и специальной программой, управляющей этим контроллером, которую называют драйвером соответствующего периферийного устройства.
- ▶ В ПУ интерфейс чаще всего полностью реализуется аппаратным устройством - **контроллером**, хотя встречаются и программно-управляемые контроллеры для управления современными принтерами, обладающими более сложной логикой.

- ▶ Периферийные устройства могут принимать от компьютера как данные, например байты информации, которую нужно распечатать на бумаге, так и команды управления, в ответ на которые контроллер ПУ может выполнять специальные действия (перевести головку диска на требуемую дорожку, вытолкнуть лист бумаги из принтера и т. д.). Контроллер принтера, например, поддерживает некоторый набор достаточно простых команд, таких как «Печать символа», «Перевод строки», «Возврат каретки» и т. п., которые он получает от компьютера по интерфейсу и обрабатывает, управляя электромеханическими частями принтера.

Итак, рассмотрим порядок действий, в результате которых приложение распечатывает данные на принтере.

- ▶ Приложение обращается с запросом на выполнение операции ввода-вывода к операционной системе. В запросе указываются адрес данных в оперативной памяти, идентифицирующая информация о периферийном устройстве и операция, которую надо выполнить.
- ▶ Получив запрос, операционная система запускает драйвер принтера. Дальнейшие действия по выполнению операции ввода-вывода со стороны компьютера реализуются интерфейсной картой, работающей под управлением драйвера.

- ▶ Драйвер принтера оперирует командами, понятными контроллеру принтера, то есть командами «Печать символа», «Перевод строки», «Возврат каретки». Драйвер в определенной последовательности помещает коды этих команд в регистр интерфейсной карты, которая побайтно передает их по линиям связи контроллеру периферийного устройства. Для одного и того же контроллера можно разработать различные драйверы, которые с помощью одного набора команд будут реализовывать разные алгоритмы управления ПУ.
- ▶ Интерфейсная карта выполняет низкоуровневую работу, она не вдастся в смысл данных и команд, передаваемых ей драйвером, считая их однородным потоком битов. После получения от драйвера очередного байта интерфейсная карта просто последовательно передает биты в линию связи, представляя каждый бит электрическим сигналом.

Чтобы контроллеру ПУ стало понятно, что начинается передача байта, перед передачей первого бита интерфейсная карта формирует **стартовый сигнал** специфической формы, а после передачи последнего информационного бита - **стоповый сигнал**. Эти сигналы *синхронизируют* передачу байта. Контроллер, опознав стартовый бит, начинает принимать информационные биты, формируя из них байт и своем приемном буфере.

Помимо информационных битов карта может передавать бит контроля четности для определения достоверности обмена. При корректно выполненной передаче в регистре контроллера устанавливается соответствующий признак.

- Получив очередной байт, контроллер интерпретирует его и запускает заданную операцию принтера. Закончив работу по печати всех символов документа, контроллер принтера сообщает об этом драйверу компьютера. Драйвер передает операционной системе сообщение о выполнении запроса, а та, в свою очередь, сигнализирует об этом событии приложению.

Простейший случай взаимодействия двух компьютеров

- ▶ Вернемся к исходному вопросу: как пользователю, работающему с некоторым приложением на компьютере А. распечатать текст на принтере компьютера В (рис. 2.2).

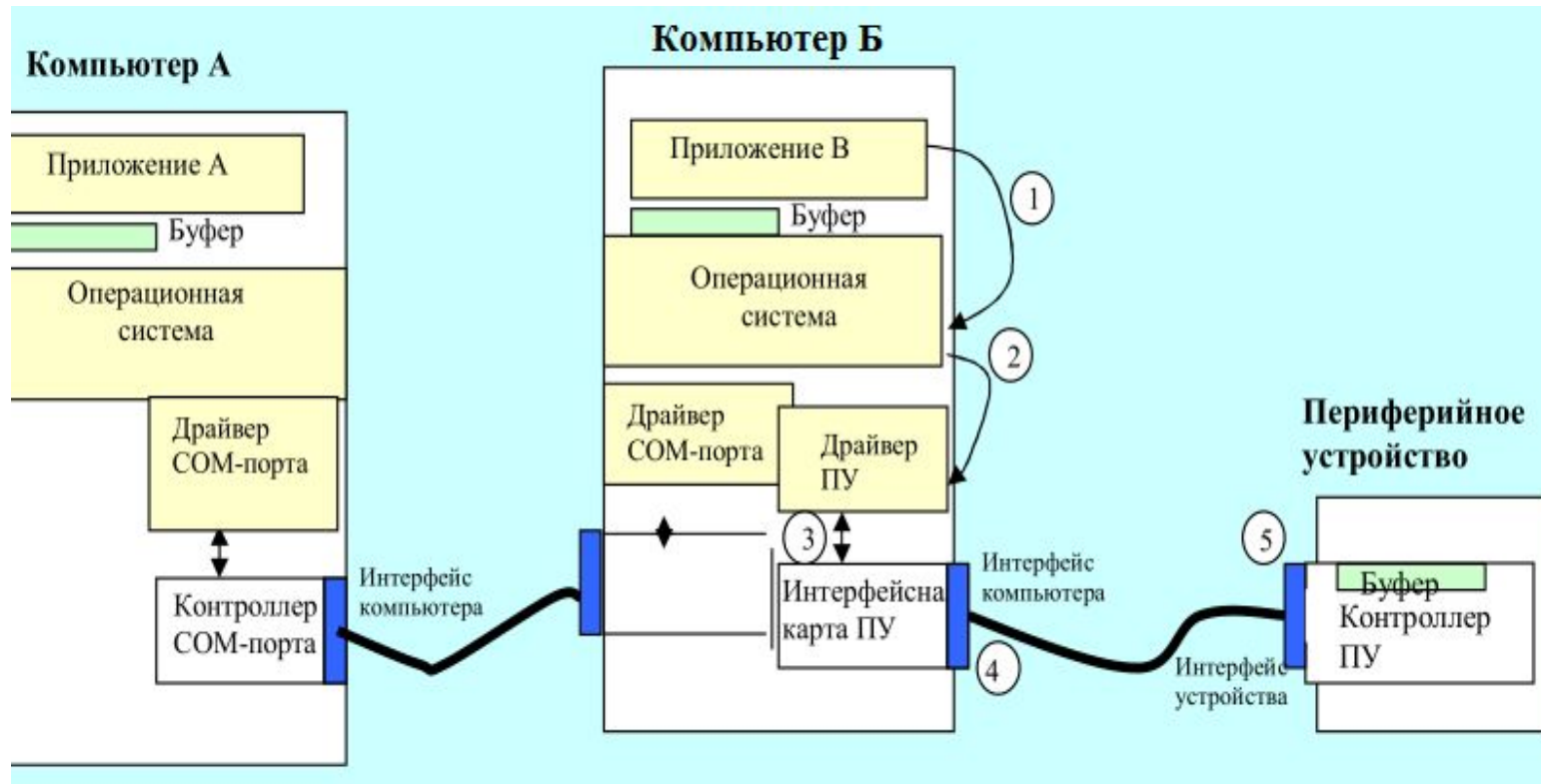


Рис. 2.2. Совместное использование периферийного устройства

- ▶ Приложение А не может получить непосредственный доступ к ресурсам компьютера В — его дискам, файлам, принтеру. Оно может только «попросить» об этом другую программу, выполняемую на том компьютере, которому принадлежат эти ресурсы. Эти «просьбы» выражаются в виде сообщений, передаваемых по каналам связи между компьютерами. Сообщения могут содержать как команды на выполнение некоторых действий («открыть файл»), так и собственно информационные данные (содержимое некоторого файла).
- ▶ **Сообщение - блок данных стандартного формата. Формат сообщения: заголовок, данные.**

- ▶ Механизмы взаимодействия компьютеров в сети многое позаимствовали у схемы взаимодействия компьютера с периферийными устройствами. В самом простом случае связь компьютеров может быть реализована с помощью тех же самых средств, которые используются для связи компьютера с периферией. Пусть для определенности связь между компьютерами будет осуществляться через последовательный интерфейс — СОМ-порт. С каждой стороны контроллер СОМ-порта работает под управлением драйвера СОМ-порта. Вместе они обеспечивают передачу по кабелю между компьютерами одного байта информации.
- ▶ *В локальных сетях подобные функции передачи данных в линию связи выполняются сетевыми интерфейсными картами (Network Interface Card, NIC), называемыми также сетевыми адаптерами, и их драйверами.*

- ▶ Итак, механизм обмена байтами между двумя компьютерами определен. Однако этого еще недостаточно для решения поставленной задачи — распечатки текста на «чужом» принтере. В частности, необходимо, чтобы компьютер В «понял», какую операцию он должен выполнить с передаваемыми данными, на каком из имеющихся в его распоряжении устройств, в каком виде должен быть распечатан текст и т. п. Обо всем этом должны договориться приложения А и В путем обмена сообщениями.
- ▶ Чтобы приложения могли «понимать» получаемую друг от друга информацию, программисты, разрабатывавшие приложения А и В, должны *строго оговорить* форматы сообщений, которыми будут обмениваться приложения, и их семантику. Например, они могут договориться о том, что любое выполнение удаленной операции печати начинается с передачи сообщения, запрашивающего информацию о готовности приложения В; что в следующем сообщении идут идентификаторы компьютера и пользователя, сделавшего запрос; что признаком срочного завершения печати является определенная кодовая комбинация и т. п. ***Тем самым определяется протокол взаимодействия приложений.***

- ▶ **Протокол** — это набор информационных сообщений определенного формата, которыми обмениваются два устройства или две программы, а также набор правил, определяющих логику обмена этими сообщениями.

Рассмотрим взаимодействие всех элементов этой небольшой сети, которые позволят приложению на компьютере А распечатать текст на принтере компьютера В:

- ▶ Приложение А формирует сообщение-запрос для приложения В на печать текста и помещает его в свой буфер. Чтобы передать данный запрос компьютеру В, приложение А обращается к локальной ОС, которая запускает драйвер COM-порта компьютера и сообщает ему адрес буфера, где хранится запрос. Затем, по ранее описанной схеме, драйвер и контроллер COM-порта компьютера А, взаимодействуя с драйвером и контроллером COM-порта компьютера В, передают сообщение байт за байтом в компьютер В.

- ▶ Драйвер СОМ-порта компьютера В постоянно находится в режиме ожидания прихода информации из внешнего мира. В некоторых случаях драйвер вызывается асинхронно, по прерываниям от контроллера. Получив очередной байт и убедившись в его корректности, драйвер помещает его в буфер приложения В.
- ▶ Приложение В принимает сообщение, интерпретирует его и формирует запрос к локальной ОС на выполнение тех или иных действий с принтером. В ходе печати могут возникнуть ситуации, о которых необходимо сообщить приложению А. В этом случае используется симметричная схема: теперь запрос на передачу сообщения поступает от приложения В к локальной ОС компьютера В. Драйверы и контроллеры СОМ-портов обоих компьютеров организуют побайтную передачу сообщения, которое затем помещается в буфер приложения А.

- ▶ Потребность в доступе к удаленным файлам может возникать у пользователей многих других приложений: текстового редактора, графического редактора, системы управления базой данных (СУБД). Очевидно, нерационально включать рассмотренные универсальные функции по организации ввода-вывода в состав каждого приложения. Более эффективно решают задачу пара специализированных программных модулей.
- ▶ **Клиент** - системный программный модуль, предназначенный для формирования сообщений-запросов к удаленной машине от разных приложений, а затем приема результатов и передачи их соответствующим приложениям.

- ▶ Очень удобной и полезной функцией клиентской программы является способность отличить запрос к удаленному ресурсу от запроса к локальному ресурсу. Если клиентская программа умеет это делать, то приложения не должны заботиться о том, с каким принтером они работают (локальным или удаленным), клиентская программа сама распознает и перенаправляет (redirect) запрос к удаленной машине. Отсюда и название, часто используемое для клиентского модуля, — **редиректор**.
- ▶ Клиент и сервер выполняют системные функции по обслуживанию запросов всех приложений компьютера А на удаленный доступ к ресурсу (принтеру, файлам, факсу) компьютера В. Чтобы приложения компьютера В могли пользоваться ресурсами компьютера А. описанную схему нужно симметрично дополнить клиентом для компьютера В и сервером для компьютера А.

Схема взаимодействия клиента и сервера с приложениями и локальной операционной системой приведена на рис. 2.3.

Взаимодействие между компьютерами сети происходит за счет передачи сообщений через сетевые адаптеры и каналы связи. С помощью этих сообщений один компьютер обычно запрашивает доступ к локальным ресурсам другого компьютера. Такими ресурсами могут быть как данные, хранящиеся на диске, так и разнообразные периферийные устройства - принтеры, плоттеры, факс-аппараты и т.д.

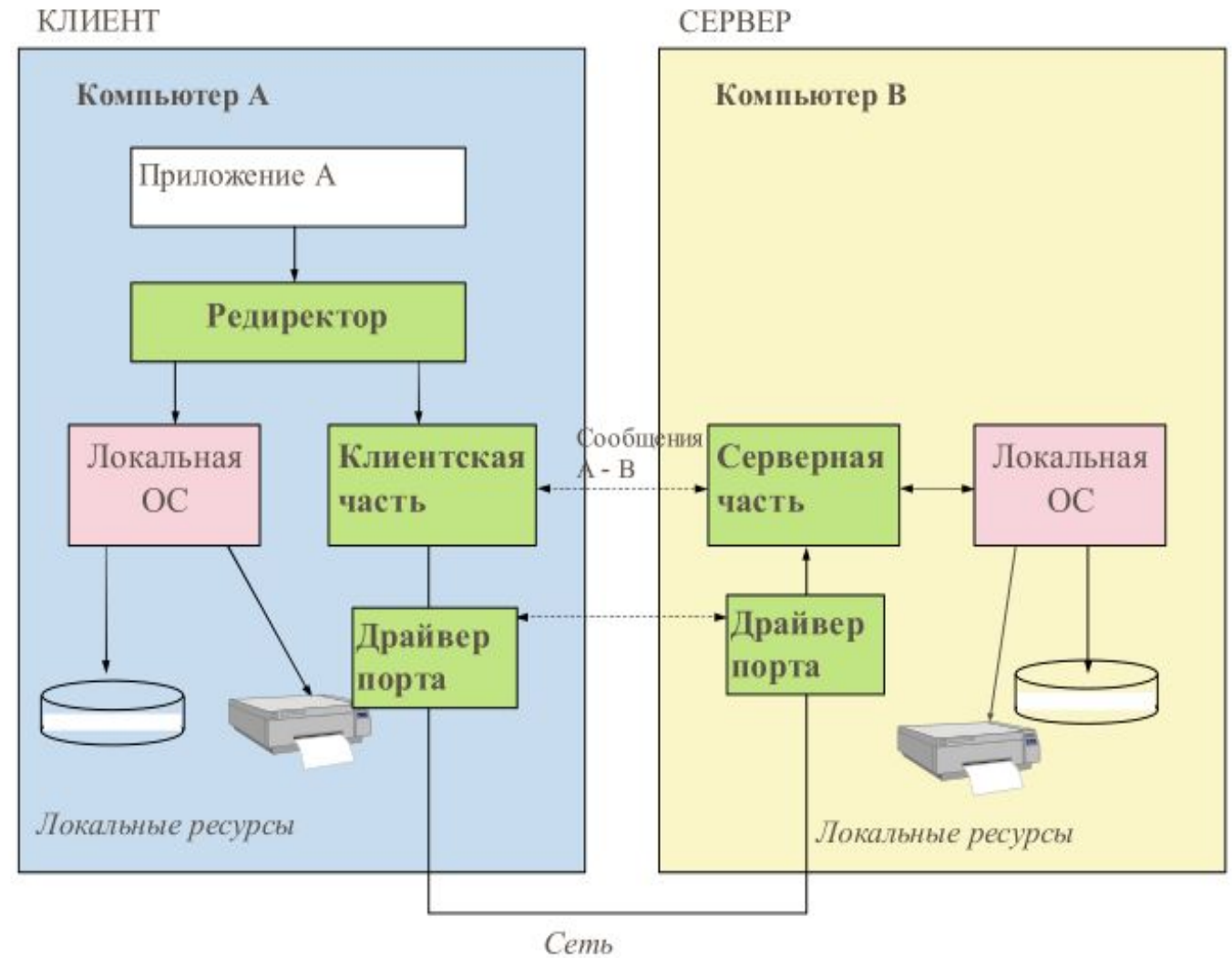


Рис. 2.3. Взаимодействие программных компонентов при связи двух компьютеров

Разделение локальных ресурсов каждого компьютера между всеми пользователями сети - основная цель создания вычислительной сети.

Несмотря на то, что мы рассмотрели очень простую схему связи только двух компьютеров, функции программ, обеспечивающих удаленный доступ к принтеру, во многом совпадают с функциями сетевой операционной системы, работающей в сети с более сложными аппаратными связями компьютеров.

К основным функциональным компонентам сетевой ОС относятся средства управления локальными ресурсами и сетевые средства.

- ▶ *Сетевые средства можно разделить на три компонента:*
 - ▶ *Средства предоставления локальных ресурсов и услуг в общее пользование - серверная часть ОС.*
 - ▶ *Средства запроса доступа к удаленным ресурсам - клиентская часть ОС.*
 - ▶ *Транспортные средства ОС, которые с коммуникационной системой обеспечивают передачу сообщений между компьютерами сети.*

В связи с этим, локальные сети можно подразделить в зависимости от используемой сетевой операционной системы на:

- ▶ Серверные сети
- ▶ Одноранговые сети
- ▶ Комбинированные сети.

Термины «клиент» и «сервер» используются для обозначения не только программных модулей, но и компьютеров, подключенных к сети. Если компьютер преимущественно предоставляет свои ресурсы другим компьютерам сети, то он называется **сервером**, а если он их потребляет — **клиентом**. Иногда один и тот же компьютер может одновременно играть роли и сервера, и клиента.

Сетевые службы и приложения

- ▶ Предоставление пользователям совместного доступа к определенному типу ресурсов, например, к файлам, называют также предоставлением сервиса.
- ▶ Обычно сетевая операционная система поддерживает несколько видов сетевых сервисов для своих пользователей – файловый сервис, сервис печати, сервис электронной почты, сервис удаленного доступа и т. п. Программы, реализующие сетевые сервисы, относятся к классу распределенных программ.
- ▶ **Распределенная программа** - это программа, которая состоит из нескольких взаимодействующих частей (в приведенном на рис. 2.4 примере - из двух), причем каждая часть, как правило, выполняется на отдельном компьютере сети.

- **Сетевые службы** - это *системные* распределенные программы, реализующие сетевые сервисы. Они представляют собой пару «клиент-сервер» и являются неотъемлемыми компонентами сетевой ОС.



Рис. 2.4. Взаимодействие частей распределенного приложения

- ▶ Однако в сети могут выполняться и распределенные пользовательские приложения. Распределенное приложение также состоит из нескольких частей, каждая из которых выполняет какую-то определенную законченную работу по решению прикладной задачи. Например, одна часть приложения, выполняющаяся на компьютере пользователя, может поддерживать специализированный графический интерфейс, вторая — работать на мощном выделенном компьютере и заниматься статистической обработкой введенных пользователем данных, третья - заносить полученные результаты в базу данных на компьютере с установленной стандартной СУБД.

Распределенные приложения в полной мере используют потенциальные возможности распределенной обработки, предоставляемые вычислительной сетью, и поэтому часто называются *сетевыми приложениями*.

- ▶ Не всякое приложение, выполняемое в сети, является распределенным. Значительная часть истории локальных сетей связана как раз с использованием таких нераспределенных приложений. Рассмотрим, например, как происходила работа пользователя с известной в свое время СУБД dBase. Файлы базы данных, с которыми работали все пользователи сети, располагались на файловом сервере. Сама же СУБД хранилась на каждом клиентском компьютере в пиле единого программного модуля. Программа dBase была рассчитана только на обработку данных, расположенных на том же компьютере, что и сама программа. Пользователь запускал dBase на своем компьютере, и программа искала данные на локальном диске, совершенно не принимая во внимание существование сети. Чтобы обрабатывать с помощью dBase данные, расположенные на удаленном компьютере, пользователь обращался к услугам файловой службы, которая доставляла данные с сервера на клиентский компьютер и создавала для СУБД эффект их локального хранения.

Большинство приложений, используемых в компьютерных сетях в середине 80-х годов, являлись обычными нераспределенными приложениями. И это понятно — они были написаны для автономных компьютеров, а потом просто были перенесены в сетевую среду. Создание же распределенных приложений, хотя и сулило много преимуществ (снижение сетевого трафика, специализация компьютеров), оказалось делом совсем не простым. Нужно было решать множество дополнительных проблем:

- ▶ на сколько частей разбить приложение,
- ▶ какие функции возложить на каждую часть,
- ▶ как организовать взаимодействие этих частей, чтобы в случае сбоев и отказов оставшиеся части корректно завершали работу и т. д.

Поэтому до сих пор только небольшая часть приложений являются распределенными, хотя очевидно, что именно за этим классом приложений будущее, так как они в полной мере могут использовать потенциальные возможности сетей, по распараллеливанию вычислений.

Физическая передача данных по линиям связи

Даже при рассмотрении простейшей сети, состоящей всего из двух машин, можно выявить многие проблемы, связанные с физической передачей сигналов по линиям связи.

► Кодирование

В вычислительной технике для представления данных используется двоичный код. Внутри компьютера единицам и нулям данных соответствуют дискретные электрические сигналы.

Существуют различные способы кодирования двоичных цифр, например **потенциальный способ**, при котором единице соответствует один уровень напряжения, а нулю — другой, или **импульсный способ**, когда для представления цифр используются импульсы различной полярности, либо часть импульса - перепад (фронт).

- ▶ Аналогичные подходы применимы для кодирования данных и при передаче их между двумя компьютерами по линиям связи. Однако эти линии связи отличаются по своим характеристикам от линий внутри компьютера. **Главное отличие внешних линий связи от внутренних состоит в их гораздо большей протяженности, а также в том, что они проходят вне экранированного корпуса по пространствам, зачастую подверженным воздействию сильных электромагнитных помех.** Все это приводит к существенно большим искажениям прямоугольных импульсов (например. «заваливанию» фронтов), чем внутри компьютера. Поэтому для надежного распознавания импульсов на приемном конце линии связи при передаче данных внутри и вне компьютера не всегда можно использовать одни и те же скорости и способы кодирования. Например, медленное нарастание фронта импульса из-за высокой емкостной нагрузки линии требует, чтобы импульсы передавались с меньшей скоростью (чтобы передний и задний фронты соседних импульсов не перекрывались, и импульс успел «дорасти» до требуемого уровня).

- ▶ В вычислительных сетях применяют как потенциальное, так и импульсное кодирование дискретных данных, а также специфический способ представления данных, который никогда не используется внутри компьютера, — **модуляцию** (рис. 2.5). При модуляции дискретная информация представляется синусоидальным сигналом той частоты, которую хорошо перелает имеющаяся линия связи.

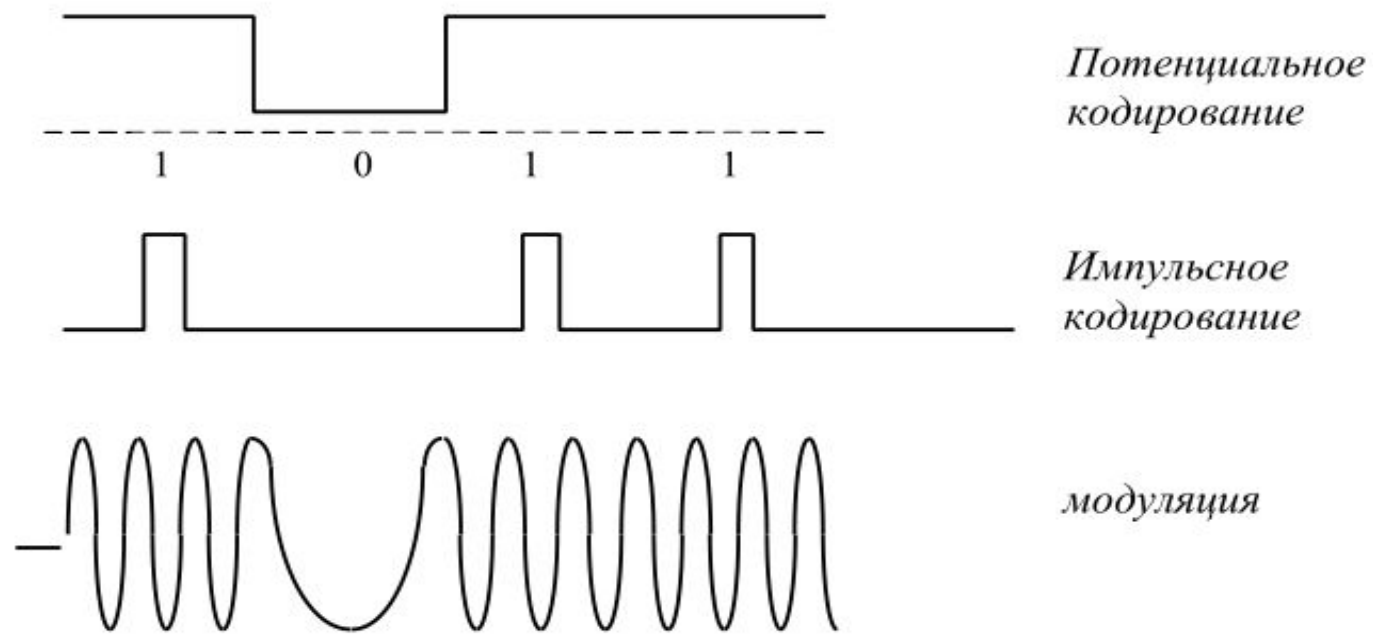


Рис. 2.5. Примеры представления дискретной информации

Модуляция является таким способом физического кодирования при котором информация кодируется изменением амплитуды, частоты или фазы синусоидального сигнала несущей частоты. (Рис. 2.6а)

- ▶ При **амплитудной модуляции** для логической единицы выбирается один уровень амплитуды синусоиды несущей частоты, а для логического нуля - другой. Этот способ редко используется в чистом виде на практике из-за низкой помехоустойчивости, но часто применяется в сочетании с другим видом модуляции - фазовой модуляцией.
- ▶ При **частотной модуляции** значения 0 и 1 исходных данных передаются синусоидами с различной частотой - f_0 и f_1 . Этот способ модуляции не требует сложных схем в модемах и обычно применяется в низкоскоростных модемах, работающих на скоростях 300 или 1200 бит/с.
- ▶ При **фазовой модуляции** значениям данных 0 и 1 соответствуют сигналы одинаковой частоты, но с различной фазой, например 0 и 180 градусов или 0, 90, 180 и 270 градусов.

► **Потенциальное, или импульсное, кодирование** применяется на каналах *высокого качества*, а **модуляция на основе синусоидальных сигналов** предпочтительнее в том случае, когда канал вносит **сильные искажения в передаваемые сигналы**. Например, модуляция используется в глобальных сетях при передаче данных через аналоговые телефонные каналы связи, которые были разработаны для передачи голоса в аналоговой форме и поэтому плохо подходят для непосредственной передачи импульсов.

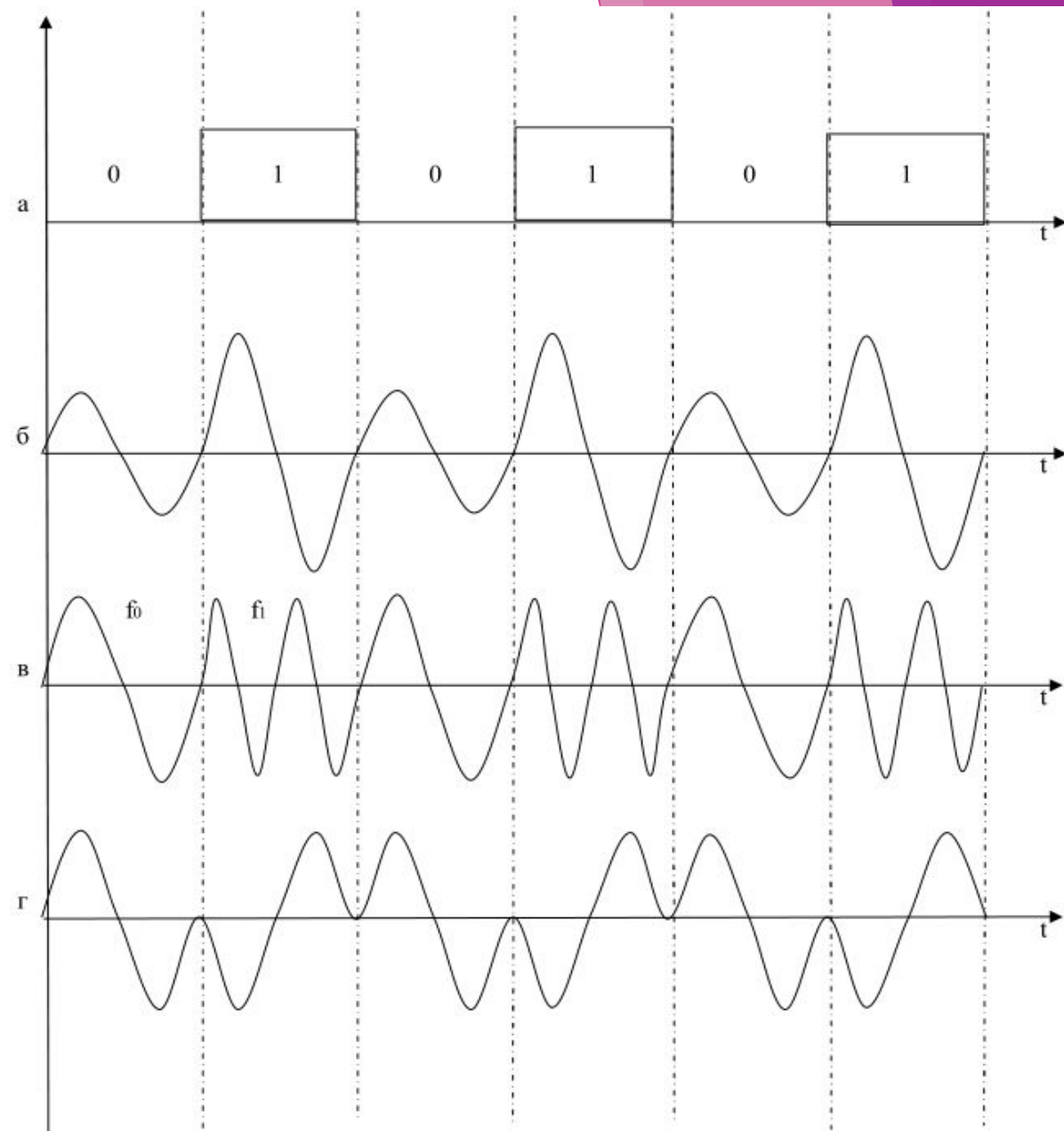


Рис. 2.6. Различные типы модуляций

- ▶ На способ передачи сигналов влияет и *количество проводов* в линиях связи между компьютерами. Для снижения стоимости линий связи в сетях обычно стремятся к сокращению количества проводов и из-за этого используют не параллельную передачу всех битов одного байта или даже нескольких байтов, как это делается внутри компьютера, а последовательную побитную передачу, требующую всего одной пары проводов.
- ▶ Еще одной проблемой, которую нужно решать при передаче сигналов, является проблема взаимной **синхронизации** передатчика одного компьютера с приемником другого.

Синхронизация передатчика и приемника

- ▶ Синхронизация передатчика и приемника нужна для того, чтобы приемник точно знал, в какой момент времени необходимо считывать новую информацию с линии связи. Эта проблема в сетях решается сложнее, чем при обмене данными между близко расположенными устройствами, например между блоками внутри компьютера или же между компьютером и принтером.
- ▶ При связи компьютера с периферийным устройством синхронизация обеспечивается путем периодической синхронизации заранее обусловленными кодами или импульсами характерной формы, отличающейся от формы импульсов данных.
- ▶ На небольших расстояниях (внутри компьютера) хорошо работает схема, основанная на отдельной тактирующей линии связи (рис. 2.66), так что информация снимается с линии данных только в момент прихода тактового импульса. В сетях использование этой схемы вызывает трудности из-за неоднородности характеристик проводников в кабелях.

- ▶ На больших расстояниях неравномерность скорости распространения сигнала может привести к тому, что тактовый импульс придет настолько позже или раньше соответствующего сигнала данных, что бит данных будет пропущен или считан повторно. Другой причиной, по которой в сетях отказываются от использования тактирующих импульсов, является экономия проводников в дорогостоящих кабелях.
- ▶ Поэтому **в сетях применяются так называемые самосинхронизирующиеся коды, сигналы которых несут для передатчика указания о том, в какой момент времени нужно осуществлять распознавание очередного бита (или нескольких бит, если код ориентирован более чем на два состояния сигнала). Любой резкий перепад сигнала - так называемый фронт - может служить хорошим указанием для синхронизации приемника с передатчиком. (манчестерский код)**

- ▶ При использовании синусоид в качестве несущего сигнала результирующий код обладает свойством самосинхронизации, так как изменение амплитуды несущей частоты дает возможность приемнику определить момент появления входного кода.
- ▶ Несмотря на предпринимаемые меры (выбор соответствующей скорости обмена данными, линий связи с определенными характеристиками, способа синхронизации приемника и передатчика), существует вероятность искажения некоторых битов передаваемых данных. **Для повышения надежности передачи данных между компьютерами часто используется стандартный прием — подсчет контрольной суммы и передача ее по линиям связи после каждого байта или после некоторого блока байтов.** Часто в протокол обмена данными включается как обязательный элемент сигнал-квитанция, который подтверждает правильность приема данных и посылается от получателя отправителю.

Характеристики физических каналов

Существует большое количество характеристик, связанных с передачей трафика через физические каналы. С теми из них, которые будут необходимы нам уже в ближайшее время, мы познакомимся сейчас.

- **Предложенная нагрузка** - это поток данных, поступающий от пользователя на вход сети. Предложенную нагрузку можно характеризовать скоростью поступления данных в сеть — в битах в секунду (или килобитах, мегабитах и т. д.).
- **Скорость передачи данных** (information rate или throughput, оба английских термина используются равноправно) — это *фактическая* скорость потока данных, прошедшего через сеть. Эта скорость может быть меньше, чем скорость предложенной нагрузки, так как данные в сети могут искажаться или теряться.

- **Емкость канала связи (capacity), называемая также пропускной способностью, представляет собой максимально возможную скорость передачи информации по каналу.**

Спецификой этой характеристики является то, что она отражает не только параметры **физической среды передачи**, но и особенности **выбранного способа передачи** дискретной информации по этой среде. Например, емкость канала связи в сети Ethernet на оптическом волокне равна 10 Мбит/с. Эта скорость является предельно возможной для сочетания технологии Ethernet и оптического волокна. Однако для того же самого оптического волокна можно разработать и другую технологию передачи данных, отличающуюся способом кодирования данных, тактовой частотой и другими параметрами, которая будет иметь другую емкость. Так, технология Fast Ethernet обеспечивает передачу данных по тому же оптическому волокну с максимальной скоростью 100 Мбит/с, а технология Gigabit Ethernet - 1000 Мбит/с. Передатчик коммуникационного устройства должен работать со скоростью, равной пропускной способности канала. Эта скорость иногда называется битовой скоростью передатчика (bit rate of transmitter).

Еще одна группа характеристик канала связи связана с возможностью передачи информации по каналу в одну или обе стороны.

При взаимодействии двух компьютеров обычно требуется передавать информацию в обоих направлениях, от компьютера А к компьютеру В и обратно. Даже в том случае, когда пользователю кажется, что он только получает информацию (например, загружает файл из Интернета) или передает (отправляет электронное письмо), обмен информацией идет в двух направлениях. Просто существует основной поток данных, которые интересуют пользователя, и вспомогательный поток противоположного направления, который образуют квитанции о получении этих данных.

Физические каналы связи делятся на несколько типов в зависимости от того, могут они передавать информацию в обоих направлениях или нет.

- ▶ **Дуплексный канал** обеспечивает одновременную передачу информации в обоих направлениях. Дуплексный канал может состоять из двух физических сред, каждая из которых используется для передачи информации только в одном направлении. Возможен вариант, когда одна среда служит для одновременной передачи встречных потоков, в этом случае применяют дополнительные методы выделения каждого потока из суммарного сигнала.
- ▶ **Полудуплексный канал** также обеспечивает передачу информации в обоих направлениях, но не одновременно, а по очереди. То есть в течение определенного периода времени информация передается в одном направлении, а в течении следующего периода — в обратном.
- ▶ **Симплексный канал** позволяет передавать информацию только в одном направлении. Часто дуплексный канал состоит из двух симплексных каналов.

Вопросы и задания

- ▶ 1. Какая информация передается по каналу, связывающему внешние интерфейсы компьютера и периферийного устройства?
- ▶ 2. Какие компоненты включает интерфейс устройства?
- ▶ 3. Какие задачи решает ОС при обмене с периферийным устройством?
- ▶ 4. Какие функции возлагаются на драйвер периферийного устройства?