

# Интерфейсы.

## Содержание.

|                                                                      |           |
|----------------------------------------------------------------------|-----------|
| <b>Тема 1. Введение.....</b>                                         | <b>1</b>  |
| 1.1. Структуры интерфейса вычислительных машин.....                  | 5         |
| 1.2. Определения и некоторые компоненты интерфейсов.....             | 7         |
| <b>Тема 2. Классификация интерфейсов.....</b>                        | <b>12</b> |
| 2.1. Классификация по способу соединения компонентов<br>системы..... | 12        |
| 2.2. Классификация по способу передачи информации.....               | 14        |
| 2.3. Классификация по принципу обмена информацией.....               | 15        |
| 2.4. Классификация по режиму обмена информации.....                  | 16        |
| 2.5. Классификация по функциональному назначению.....                | 17        |
| 2.6. Классификация по конструктивному исполнению.....                | 19        |
| 2.7. Классификация по природе сигнала.....                           | 20        |
| 2.8. Классификация по топологии интерфейса.....                      | 21        |
| 2.9. Классификация по природе сигналов информации.....               | 23        |
| 2.10. Классификация видов интерфейсов.....                           | 24        |
| <b>Тема 3. Параллельный интерфейс.....</b>                           | <b>25</b> |
| <b>Тема 4. Последовательный интерфейс.....</b>                       | <b>27</b> |
| 4.1. Способы последовательной передачи.....                          | 27        |
| 4.2. Интерфейс RS-232с.....                                          | 28        |
| 4.3. Интерфейс токовая петля.....                                    | 28        |
| 4.4. Интерфейс IEEE-1394 (Fire Wire, i.LINK).....                    | 29        |
| 4.5. Интерфейс SATA.....                                             | 31        |
| <b>Тема 5. Беспроводный интерфейс.....</b>                           | <b>31</b> |
| 5.1. Инфракрасный интерфейс IRDA.....                                | 32        |
| 5.2. Радиointерфейс Bluetooth.....                                   | 33        |

|                                                                                                                  |           |
|------------------------------------------------------------------------------------------------------------------|-----------|
| <b>Тема 6. Шина USB.....</b>                                                                                     | <b>36</b> |
| 6.1. Общие положения.....                                                                                        | 36        |
| 6.2. Модель передачи данных.....                                                                                 | 38        |
| 6.3. Типы передачи данных.....                                                                                   | 39        |
| <b>Тема 7. Интерфейсы периферийных устройств.....</b>                                                            | <b>40</b> |
| 7.1. Интерфейсы накопителей на магнитных жестких дисках.....                                                     | 40        |
| 7.2. Интерфейсы оптических дисков.....                                                                           | 48        |
| 7.3. Интерфейсы мониторов.....                                                                                   | 50        |
| 7.4. Интерфейсы клавиатуры.....                                                                                  | 54        |
| 7.5. Интерфейсы манипуляторов типа мышь.....                                                                     | 56        |
| <b>Тема 8. Пользовательский интерфейс.....</b>                                                                   | <b>59</b> |
| 8.1. Классификация интерфейсов.....                                                                              | 59        |
| 8.2. Типы пользовательских интерфейсов и этапы разработки... ..                                                  | 66        |
| 8.3. Психофизические особенности человека, связанные с<br>восприятием, запоминанием и обработкой информации..... | 71        |
| 8.4. Пользовательская и программная модели интерфейса.....                                                       | 75        |
| 8.5. Классификации диалогов и общие принципы их разработки... ..                                                 | 78        |
| 8.6. Оценка пользовательских интерфейсов.....                                                                    | 83        |
| 8.7. Человеческие ошибки.....                                                                                    | 91        |
| 8.8. Обучение работе с системой.....                                                                             | 96        |
| 8.9. Элементы управления.....                                                                                    | 102       |

## Тема 1. Введение.

Создание современных средств вычислительной техники связано с задачей объединения в один комплекс различных блоков - устройств хранения и отображения информации, аппаратуры данных и непосредственно ЭВМ. Большое разнообразие информационно-вычислительных систем достигается благодаря компоновке разных устройств посредством интерфейсов.

**Интерфейс** - совокупность унифицированных технических, программных и конструктивных средств, необходимых для реализации взаимодействия различных функциональных элементов в автоматических системах обработки информации при условиях, предписанных стандартом и направленных на обеспечение информационной, электрической и конструктивной совместимости указанных элементов.

Интерфейс - представляет собой совокупность линий и шин, сигналов, электронных схем и алгоритмов, которая предназначена для осуществления обмена информацией между устройствами, т.е. внутрисистемных и межсистемных связей и устройств сопряжения.

От интерфейса зависит технология общения человека с компьютером. Компьютер обменивается информацией с человеком посредством определенных правил, обязательных как для машины, так и для человека. Эти правила в компьютерной литературе называются **пользовательским интерфейсом**.

Интерфейс может быть понятным и непонятным, дружелюбным и нет.

Производительность, область применения и эффективность использования ЭВМ определяется не только возможностями ее процессора и основной памяти, но в очень большой степени составом ее периферийных устройств, техническими данными и способом организации их совместной работы с ядром ЭВМ.

Используются термины "интерфейсная система", "программный интерфейс", "физический интерфейс", "аппаратный интерфейс", и т.д.

Вычислительная машина содержит помимо процессора (процессоров) и основной памяти, образующих ее **ядро**, многочисленные и разнообразные по

выполняемым функциям и принципам действия периферийные устройства, предназначенные для хранения больших объемов информации (внешние запоминающие устройства) и для ввода в ЭВМ и вывода из нее информации.

Передача информации с периферийного устройства в ядро ЭВМ (память или процессор) называется операцией ввода, а передача из ядра ЭВМ в периферийное устройство - операцией вывода.

Поскольку процессоры, памяти и периферийное оборудование выпускаются в готовом виде, именно от выбора интерфейсов и их качества зависят свойства создаваемой системы. Ошибки в выборе интерфейсов приводят к большим потерям средств и времени.

Инженер, создающий систему, изучает генеральную задачу и характеристики отдельных доступных устройств, чтобы подобрать интерфейс и режимы его работы для оптимального решения задачи. Главной областью персональных знаний **инженера-интегратора** являются интерфейсы.

При разработке систем интерфейсов должны быть решены следующие задачи:

- возможность реализации ЭВМ с переменным составом оборудования (переменной конфигурацией), в первую очередь с различным набором периферийных устройств, с тем чтобы пользователь мог выбирать состав оборудования (конфигурацию) машины в соответствии с ее назначением, легко дополнять машину новыми устройствами;
- для эффективного и высокопроизводительного использования оборудования, в ЭВМ должны реализовываться параллельная во времени работа процессора над программой и выполнение периферийными устройствами процедур ввода-вывода;
- необходимо обеспечить автоматическое распознавание и реакцию ядра ЭВМ на многообразие ситуаций, возникающих в периферийном устройстве.

Современное развитие микроэлектроники, а также тенденции и практика построения микропроцессорных систем определили следующие **направления развития интерфейсов**:

1. Дальнейшее повышение уровня унификации интерфейсного оборудования и стандартизации, условий совместимости существующих наиболее распространенных интерфейсов на основе обобщения опыта их широкого использования. Это совершенствование направлено на создание новых стандартных интерфейсов или на повышение уровня стандартизации существующих.
2. Модернизация и расширение функциональных возможностей существующих интерфейсов без нарушения условий совместимости благодаря новейшим достижениям в микроэлектронной технологии и технологии разработки средств передачи информации. Основная цель этого направления - удлинение сроков морального старения стандартных интерфейсов и расширение области их применения.
3. Создание принципиально новых интерфейсов и разработка требований на их унификацию и стандартизацию. Эта тенденция обусловлена в первую очередь разработкой систем с параллельной распределенной обработкой информации на основе качественно новых принципов организации вычислительного процесса, а также интегрированных распределенных систем.

#### **4.1. Структуры интерфейса вычислительных машин.**

Вычислительная машина с одной шиной. В структурах взаимосвязей с одной шиной имеется одна **системная шина**, обеспечивающая обмен информацией между процессором и памятью, а также между УВВ с одной стороны, и процессором либо памятью - с другой (Рис. 1). Для такого подхода характерны простота и низкая стоимость. Однако одношинная организация не в состоянии обеспечить высокую интенсивность и скорость **транзакций**, причем «узким местом» становится именно шина.



Рис. 1. Структура взаимосвязей с одной шиной.

Если к шине подключено большое число устройств, ее пропускная способность падает, поскольку частая **передача прав управления** шиной от одного устройства к другому приводит к ощутимым задержкам. По этой причине во многих ЭВМ предпочтение отдается использованию нескольких шин, образующих определенную иерархию.

Вычислительная машина с двумя видами шин. Хотя контроллеры устройств ввода/вывода (УВВ) могут быть подсоединены непосредственно к системной шине, больший эффект достигается применением одной или нескольких шин ввода/вывода (Рис. 2). УВВ подключаются к шинам ввода/вывода, которые берут на себя основной трафик, не связанный с выходом на процессор или память.

Адаптеры шин обеспечивают буферизацию данных при пересылке между системной шиной и контроллерами УВВ.



Рис. 2. Структура взаимосвязей с двумя видами шин.

Это позволяет ЭВМ поддерживать работу множества устройств ввода/вывода и одновременно «развязать» обмен информацией по тракту процессор - память и обмен информацией с УВВ. Подобная схема существенно снижает нагрузку на скоростную шину «процессор-память» и способствует повышению общей производительности ЭВМ.

Вычислительная машина с тремя видами шин. Для подключения быстродействующих периферийных устройств в систему шин может быть добавлена высокоскоростная **шина расширения** (Рис. 3).



Рис. 3. Структура взаимосвязей с тремя видами шин.

Шины ввода/вывода подключаются к шине расширения, а уже с нее через адаптер к шине «процессор-память». Схема еще более снижает нагрузку на шину «процессор-память». Такую организацию шин называют архитектурой с «пристройкой».

## 1.2. Определения и некоторые компоненты интерфейсов.

**Периферийное устройство (ПУ)** - программно доступные элементы, не попавшие в центральную часть ЭВМ:  
- устройства хранения данных (диски);

- устройства ввода/вывода (мышь, клавиатура, монитор, принтер, сканер);
- коммуникационные устройства (сетевые платы, модем, Bluetooth адаптер).

**Стык** - место соединения устройств передачи сигналов данных, входящих в сети передачи данных (СПД). Это понятие используется для описания функций и средств сопряжения элементов СПД.

**Назначение интерфейсов, стыков и протоколов - унификация внутрисистемных и межсистемных связей, внутрисетевых и межсетевых связей СВТ. Основная функция интерфейсов и стыков - обеспечение информационной, электрической и конструктивной совместимости СВТ.**

**Информационная совместимость** - согласованность действий функциональных элементов в соответствии с совокупностью **логических условий**. Логические условия информационной совместимости определяют функциональную и структурную организацию интерфейса.

**Электрическая совместимость** - согласованность статических и динамических параметров электрических сигналов в системе шин с учетом ограничений на пространственное размещение устройств интерфейса и техническую реализацию приемопередающих элементов.

**Конструктивная совместимость** - согласованность конструктивных элементов интерфейса, предназначенных для обеспечения механического контакта соединений и механической замены схемных элементов, блоков и устройств.

**Линии интерфейсов** – электрические цепи, являющиеся составными физическими связями интерфейса.

**Магистраль** - совокупность всех линий интерфейса - информационных каналов, каналов управления информационными каналами. По

информационной магистрали передаются коды адресов, команд, данных, состояния.

**Шина** - совокупность линий, сгруппированных по функциональному назначению. Операции на шине называются **транзакциями**. Интерфейсная транзакция - законченная операция по пересылке некоторой порции информации. Транзакция может состоять из последовательности фаз (состояний, шагов), в каждой из которых решается какая-то элементарная задача. Основные виды транзакций – транзакции записи и транзакции чтения. Если в обмене участвуют устройства ввода /вывода, то говорят о транзакциях ввода/вывода.

**Протокол** - строго заданная совокупность правил или процедура, определяющая способ выполнения определенного класса функций соответствующими средствами вычислительной техники (СВТ), обеспечивающий выполнение транзакций. **Сложность протокола** зависит от возможного числа участников, их отношений и ряда характеристик обмена, которые должен обеспечить интерфейс.

**Шины адреса** предназначены для выборки в магистрали узлов устройства, ячеек памяти.

**Шины команд** используются для управления операциями на магистрали.

**Шины данных** используются для передачи в основном двоичных кодов (в формате машинных кодов во внутрисистемных интерфейсах, стандартных кодов типа ASCII в интерфейсах измерительных систем).

**Шины состояния** используются для передачи сообщений, описывающих результат выполнения операции на интерфейсе или состояния устройств сопряжения.

**Шина передачи управления** используется для реализации операций

приоритетного занятия магистрали (арбитража ресурсов шины). Состав и конфигурация шины зависят от структуры управления интерфейсом. Различают децентрализованную и централизованную структуры.

**Шина прерывания** применяется в основном в системных интерфейсах ЭВМ и программно - модульных системах управления и измерения для идентификации устройства, запрашивающего сеанс связи. Устройство идентифицируется либо адресом источника прерывания, либо адресом программы обслуживания прерывания, так называемым вектором прерывания.

**Шины управления** режимом работы и специальных управляющих сигналов содержат линии, обеспечивающие работоспособность интерфейса, в том числе приведение устройств в исходное состояние, контроль источников питания, контроль и службу времени и т.п.

**Шинные формирователи**, называемые также приемопередатчиками, шинными драйверами или магистральными вентилями - буферами, включаются между источником информации и шиной. Они усиливают сигналы по мощности при работе на шину, отключают источник информации от шины, когда он не участвует в обмене, формируют при необходимости требуемые уровни сигналов логической 1 или 0. Двухнаправленные ШФ позволяют в зависимости от сигнала управления передавать сигналы в шину или, напротив, принимать их с шины и передавать приемнику данных.

**Буферные регистры.** Буферные регистры служат для подключения к магистрали внешнего устройства. В отличие от шинных формирователей, буферные регистры **способны хранить данные**. Благодаря этому они могут выполнять временную буферизацию данных, что составляет важнейшую функцию портов. Буферные каскады с тремя состояниями на выходах регистра обеспечивают портам возможность отключения от магистрали под действием управляющих сигналов, а также необходимую нагрузочную способность.

Через порты ввода данные от ВУ поступают в магистраль, а через порты вывода данные с магистрали передаются тому или иному модулю. Порты ввода-вывода могут выполнять обе указанные операции.

**Периферийные адаптеры** выполняют сложные операции по передаче данных по шине. Программируемость адаптеров обеспечивает им широкую область применения вследствие изменяемости процедур обмена без изменений в схеме (с помощью команд программы), в том числе и во время работы микропроцессорной системы.

**Программируемые связные адаптеры.** При увеличении расстояний, на которые передаются данные, параллельные связи становятся неприемлемо сложными и дорогими. В этом случае применяют преобразование параллельных данных в последовательные для их передачи по одной сигнальной линии. Адаптеры реализуют асинхронный режим или асинхронно/синхронный.

Адаптер **программируется процессором** для работы с различной аппаратурой, принимает от процессора символы в параллельной форме и преобразует их в последовательную форму для передачи или получает последовательные данные и преобразует их в параллельные символы для процессора. Кроме того, связной адаптер сигнализирует процессору о готовности принять новый символ для передачи или о том, что получил символ для процессора.

Адаптер имеет набор управляющих входных и выходных сигналов для управления, например модемом или для управления другими устройствами.

**Программируемые контроллеры прерываний** – для обслуживания событий, требующих немедленной реакции, что обеспечивается прерыванием выполняемых программ и переходом к обслуживанию запросов прерывания.

В простейшем виде это блоки приоритетного прерывания, решающие несложные задачи обработки нескольких прерываний при **фиксированных приоритетах запросов**. Более сложные задачи решаются программируемыми контроллерами прерываний. Контроллеры обеспечивают различные виды

прерываний. Система прерываний должна выдать команду перехода к той подпрограмме обслуживания, которая соответствует признанному запросу.

### **Контроллеры обеспечивает различные виды прерываний:**

**1. Прерывания с фиксированными приоритетами входов.** Прерывания программы обслуживания по запросу другой подпрограммы с более высоким приоритетом, которая, в свою очередь, также может быть прервана более приоритетной программой и т.д. Прерывания с фиксированными приоритетами реализуются просто, но запросы неравноправные и при интенсивном поступлении запросов с высокими приоритетами запросы с низкими приоритетами могут вообще не получить обслуживания.

**2. Прерывания с круговым (циклическим) приоритетом.** В этом случае у каждого входа тоже есть свой приоритет, но после обслуживания он изменяется в круговом порядке так, что обслуженный вход получает низший приоритет. Такая дисциплина обслуживания характерна для ситуации с источниками, не имеющими преимуществ между собой.

**3. Специальное маскирование.** Происходит блокирование восприятия только одного входа запросов при отсутствии маскирования младших по приоритету. После снятия маски обслуживание запросов становится возможным.

**Контроллеры прямого доступа к памяти** - создание прямого тракта передач данных от внешних устройств к памяти или от памяти к внешним устройствам. При обычном обмене передачи между ВУ и памятью требуют вначале принять данные от источника в процессор, а затем выдать их из процессора приемнику, т.е. реализуются за два командных цикла. При ПДП данные не проходят через процессор, и передача слова производится за один цикл, посредством специальных БИС КПДП.

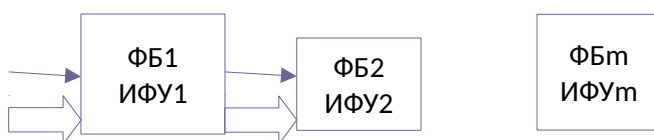
## **Тема 2. Классификация интерфейсов**

В настоящее время не существует достаточно полной объективной классификации интерфейсов. Имеющиеся классификации основываются, как правило, на одном классификационном признаке или же строятся для одного класса интерфейсов. Определенным обобщением этих классификаций является стандарт на классификационные признаки интерфейсов (ГОСТ 26.016-81), включающий четыре признака классификации:

- способ соединения компонентов системы (магистральный, радиальный, цепочечный, смешанный);
- способ передачи информации (параллельный, последовательный, параллельно - последовательный);
- принцип обмена информацией (асинхронный, синхронный);
- режим обмена информацией (симплексный; полудуплексный; дуплексный и мультиплексный режим обмена).

## 2.1. Классификация по способу соединения компонентов системы

**1. Каскадная (Цепочечная).** При цепочечной структуре каждое устройство связано не более чем с двумя другими. Частным случаем цепочечной структуры является кольцевая. Обмен информацией осуществляется непосредственно от блока к блоку, они управляют друг другом, и адресация не требуется.

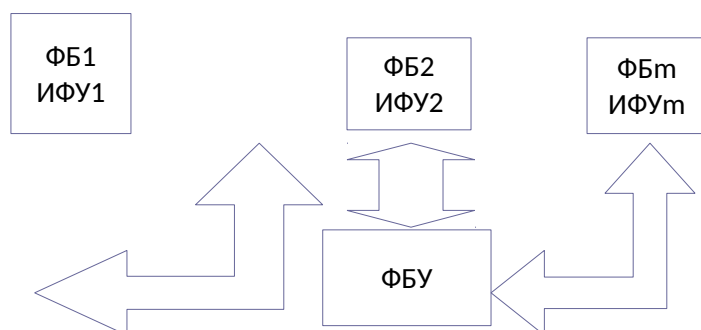


Пример использования каскадной цепочки: измерительный цифровой канал. Сигнал поступает на аналоговый вход АЦП, где преобразуется в цифровой сигнал и в дальнейшем передается от блока к блоку последовательно.

**2. Радиальная (Звезда).** Топология - способ соединения более двух узлов, при котором к каждому периферийному узлу от центрального идет свой *набор интерфейсных линий*. Как правило, центральный узел является и единственным ведущим устройством, так что из интерфейсного протокола исключается задача арбитража. Задача адресации решается исключительно средствами центрального узла, что упрощает протокол. Однако при большом числе сигнальных линий радиальные интерфейсы оказываются слишком дороги, так что радиальная топология преимущественно используется в последовательных интерфейсах. Пример радиальной топологии - новый интерфейс Serial ATA.

В системе с радиальной структурой имеется центральное устройство (контроллер или концентратор), связанный с каждым из абонентов индивидуальной группой однонаправленных линий.

Управление всеми блоками ведется одновременно, что обеспечивает высокое быстродействие.



**2. Магистральная, радиально-магистральная.** При магистральном способе имеются коллективные шины, к которым подключены все устройства системы. Характерно, что сигналы шины доступны всем устройствам, но в каждый момент времени только два устройства могут обмениваться данными последовательно во времени.

Данная структура позволяет подключить к микро-ЭВМ несколько различных устройств, что расширяет функциональные возможности системы. Радиально-магистральная несет в себе характеристики как радиальной, так и магистральной структуры.

## 2.2. Классификация по способу передачи информации

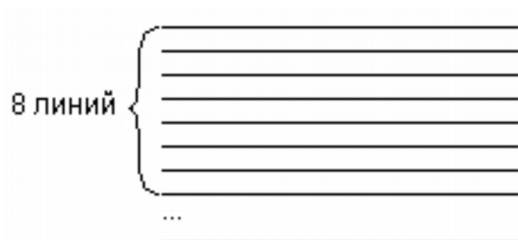
Самый распространенный способ представления данных сигналами - двоичный. Один *двоичный сигнал* за один квант времени передает *один бит* информации. Процессор с ПУ обменивается байтами (8 бит), словами (16 бит), двойными словами (32 бит).

**1. Последовательный интерфейс** - использует лишь 1 сигнальную линию и биты-группы передаются друг за другом по очереди, на каждый из них отводится свой квант времени.

Скорость передачи данных равна числу бит, передаваемых за 1 квант времени, деленному на длительность кванта.

Передача происходит по бит/сек, с использованием одной линии (RS-232C, RS-422, RS-423 HP-IL). Последовательные интерфейсы могут быть выполнены с разомкнутой или замкнутой (петлевой) магистралью, с одноступенчатой или много ступенчатой адресацией. Петлевая структура позволяет повысить надежность системы.

**2. Параллельный интерфейс.** Передача происходит по байт/сек, с использованием нескольких параллельных линий (магистраль GPIB) и все биты группы передаются одновременно за 1 квант времени, т.е. передвигаются по сигнальным линиям параллельно.



. По сравнению с последовательным, приборным, параллельный интерфейс позволяет получать существенно большую скорость обмена информацией, а также передачу информации по байтам.

В параллельных интерфейсах существует явление перекося, т.е. какие-то данные стартуя, в процессе передачи байта от источника к приемнику происходит

неодновременное поступление битов к приемнику, из-за чего ограничивается длина кабеля. На качество передачи так же влияет битность кабеля и частота. Развитие параллельного интерфейса ограничено.

На первый взгляд организация параллельного интерфейса проще и нагляднее, не нужно выстраивать биты в очередь и собирать их из принятой последовательности. Параллельный интерфейс быстрее передает, но очень много проводов в соединительном кабеле.

У последовательного интерфейса разрядность - 1 бит, у параллельного - столько, сколько сигнальных линий данных.

В последовательном интерфейсе явление перекоса отсутствует, так что повышать частоту можно до предела допустимого для приемного устройства.

В последовательном - кабель проще, но узлы сложнее.

**3. Комбинированный интерфейс.** Интерфейс, в котором группы из определенного числа бит (чаще всего из 8 бит) передаются параллельно по соответствующему числу линий, а количество групп (байт) воспринимается приемниками последовательно.

### **2.3. Классификация по принципу обмена информацией**

**1. Синхронный.** При синхронном методе, передача (прием) информации происходит в фиксированный момент времени, определенный синхроимпульсами. К нему привязаны все события интерфейса: передача бит в последовательных и байт (слов) в параллельных интерфейсах. Синхроимпульс, как правило, имеет постоянную (и точно поддерживаемую) частоту, хотя есть ряд синхронных интерфейсов и с произвольными периодами тактового сигнала (например, SPI). Сигнал синхронизации имеется и у передатчика, и у приемника; для передачи сигнала синхронизации используется специальная линия интерфейса.

Характеризуется хорошей помехозащищенностью, простотой организации, не реализует оптимального быстродействия.

**2. Асинхронный.** В асинхронных передачах данных и интерфейсах участники не имеют друг перед другом никаких особых обязательств по времени: инициатор в любой момент может начать транзакцию, а целевое устройство, как правило, может ее приостановить в случае своей неготовности.

Темп инициатора и целевого устройства согласуется с помощью механизмов квитирования или (и) управления потоком. **Квитирование** - это взаимное подтверждение отдельных шагов протокола обоими участниками транзакции. Квитирование широко применяется в параллельных интерфейсах (например, в шинах расширения), для чего используются специальные интерфейсные линии.

**Управление потоком** - это уведомление источника (передатчика) данных о возможностях их приема противоположной стороной: если приемник не успевает обрабатывать приходящие данные, он "просит" передатчик приостановить передачу на определенное время или до особого разрешения.

Асинхронная передача применима для всех устройств, не связанных с реальным временем: принтеров, сканеров, устройств хранения и т.п.

При асинхронном методе увеличивается скорость обмена информацией.

**3. Изохронная передача данных** - это передача с постоянной средней скоростью: за определенный (фиксированный) интервал времени должно быть передано определенное число данных. Это позволяет использовать один интерфейс для подключения множества устройств и организовывать множество одновременных изохронных каналов передачи.

#### **2.4. Классификация по режиму обмена информации**

**В симплексном режиме** лишь один из двух абонентов может инициировать в любой момент времени передачу информации по интерфейсу.

**В дуплексном режиме** для случая связи двух абонентов, каждый абонент может начать передачу информации другому в произвольный момент времени.

**В полудуплексном режиме** любой из двух абонентов может начать передачу информации другому, если линия связи интерфейса при этом оказывается свободной.

**В мультиплексном режиме** в случае связи нескольких абонентов, в каждый момент времени связь может быть осуществлена между парой абонентов в любом, но в единственном направлении от одного из абонентов к другому.

## **2.5. Классификация по функциональному назначению**

По функциональному назначению интерфейсы можно разделить на следующие основные классы:

- машинные (или системные);
- периферийного оборудования;
- мульти микропроцессорных систем;
- распределенных ВС (вычислительных локальных сетей, распределенных систем управления).

**Машинные интерфейсы** предназначены для организации связей между составными компонентами ЭВМ, ВК, ВС, т.е. непосредственно для их построения и связи с внешней средой. Пример – общая шина, шина данных и т.д. В интерфейсах ЭВМ преимущественно используется **магистральный принцип** построения с программным управлением процессом обмена. В настоящее время наблюдается тенденция к унификации машинных интерфейсов для всех ЭВМ.

**Интерфейсы периферийного оборудования** выполняют функции сопряжения процессоров, контроллеров с УВВ, измерительными приборами, исполнительными механизмами, аппаратурой передачи данных и внешними запоминающими устройствами. Интерфейсы периферийного оборудования представляют самый большой класс систем сопряжения, что объясняется широкой номенклатурой и разнообразием периферийного оборудования.

Интерфейсы периферийного оборудования могут быть разделены на группы интерфейсов с радиальной структурой и магистральной структурой.

**Интерфейсы с радиальной структурой - малые интерфейсы**, применяют для сопряжения исполнительных механизмов ввода-вывода с контроллерами. К ним относятся системы сопряжения с параллельной передачей информации, предназначенные для подключения стандартной периферии, системы сопряжения для подключения устройств, размещенных на большом удалении друг от друга.

**Интерфейсы магистральной структуры** используются самостоятельно, так и в качестве системотехнического дополнения, расширяющего возможностей ЭВМ на уровне связи с объектом управления, например сопряжение с программируемыми контроллерами и ЭВМ с измерительными приборами, преобразователями информации, генераторами, датчиками, пультами оператора. К такого рода интерфейсам относятся SCSI, USB.

**Интерфейсы мульти микропроцессорных систем** представляют собой в основном магистральные системы сопряжения, ориентированные на объединение в единый комплекс нескольких процессоров, модулей оперативных запоминающих устройств (ОЗУ), контроллеров ВЗУ, ограниченно размещенных в пространстве, например внутриблочные системы сопряжения. Этот класс интерфейсов отличают высокая пропускная способность и минимальное время доступа процессора к общей ОЗУ.

**Интерфейсы распределенных вычислительных сетей** предназначены для интеграции средств обработки информации, размещенных на значительном расстоянии с бит - последовательной передачей информации магистральной или кольцевой структуры. Этот класс интерфейсов в зависимости от назначения разделяется на группы интерфейсов:

- локальных сетей (с длиной магистрали от десятков метров до нескольких километров);

- распределенных систем управления;

-территориально и географически распределенных сетей ЭВМ (с длиной линии более десяти километров).

## **2.6. Классификация по конструктивному исполнению**

По конструктивному исполнению интерфейс могут быть разделены на четыре категории:

- **межблочные**, обеспечивающие взаимодействие компонентов на уровне прибора, автономного устройства, блока, стойки, шкафа;

- **внутриблочные**, обеспечивающие взаимодействие на уровне плат, субблоков;

- **внутриплатные**, обеспечивающие взаимосвязь между интегральными схемами (СИС, БИС, СБИС) на печатной плате;

- **внутрикорпусные**, обеспечивающие взаимодействие компонентов внутри СБИС.

**Межблочное сопряжение** реализуется на уровне следующих конструктивных средств:

-коаксиального и оптоволоконного кабеля;

-многожильного плоского кабеля (шлейфа);

-многожильного кабеля на основе витой пары проводов.

**Внутри блочное** сопряжение печатных плат, субблоков выполняется печатным способом или накруткой витой парой проводов внутри блока, стойки, шкафа.

**Внутри платное** сопряжение реализуется печатным способом, внутри корпусное - методами микроэлектронной технологии.

## **2.7. Классификация по природе сигнала**

Тип и природа используемого сигнала определяет требования к интерфейсу: дальность связи, скорость передачи данных, надежность, достоверность, безопасность, стоимость, удобство подключения, энергопотребление и другими.

**1. Аналоговые.** Аналоговая информация - звуки, которые мы слышим (в том числе и речь), представляют собой непрерывное изменение давления. Задача передачи такой информации встает, например, при подключении микрофона к компьютеру.

Аналоговыми являются различные виды сигналов, используемых в промышленности для сбора информации с аналоговых датчиков. Создание аналогового интерфейса является трудной задачей. Это объясняется тем, что искажение в ней измерительных сигналов может привести к резкому ухудшению метрологических характеристик системы.

**2. Цифровые.** Цифровая - дискретная информация, отображающая процесс конечным числом значений. Элементарная единица дискретной информации - 1 бит, который может принимать лишь одно из двух логических значений: 0 (истина, "да") или 1 (ложь, "нет"). Одним битом, к примеру, можно отобразить состояние кнопки "мыши": нажата или нет. Пример - оцифрованный звук, представляющий собой последовательность отсчетов мгновенных значений давления, берущихся через равные интервалы времени.

Цифровые системы характеризуются высоким быстродействием, передачи и обработки сигнала. Цифровой интерфейс имеет широкое распространение в промышленности, так как цифровые устройства занимают существенную долю в ИИС, и имеется тенденция к их увеличению.

## **2.8. Классификация по топологии интерфейса**

Между центральной частью компьютера и периферийными устройствами располагается **системный контроллер** к которому подключается шина расширения ввода/вывода. К шине расширения подключаются адаптеры и контроллеры. Адаптер подключает периферийное устройство на уровне

электрических сигналов. Контролер - интеллектуальное устройство и выполняет функции адаптера и ЭВМ, разгружая ЦП.

Если интерфейс объединяет более двух устройств, то протокол должен решать задачу адресации - выбора устройства, которое будет отвечать на данную транзакцию.

Протокол решает задачу автоматической идентификации подключенных устройств. Устройства, претендующие на роль ведущих, должны иметь возможность узнать, какие именно устройства подключены к их интерфейсу, и поддерживать конфигурационные таблицы - списки соответствия адресов и идентификаторов устройств. Решение задач автоматической адресации и идентификации позволяет говорить о возможности автоматического конфигурирования, которое может быть статическим и динамическим.

Системы **со статическим конфигурированием** позволяют подключать/отключать устройства только в нерабочем состоянии, после всех изменений конфигурации они требуют перезагрузки. Системы **с динамическим конфигурированием** позволяют подключать/отключать устройства на ходу, почти не влияя на работу остальных.

Отношения между соединяемыми устройствами могут быть равноправными или подчиненными. В случае **равноправного** интерфейса любое из устройств (по крайней мере, не одно) может претендовать на роль ведущего и стать им для каких-либо транзакций. Из этого вытекает задача **арбитража** - определения, какое устройство станет ведущим для очередной транзакции. В случае подчиненных отношений ведущим может быть только одно из устройств, соединенных интерфейсом.

**Хост-центрический** интерфейс означает, что всеми транзакциями управляет один центральный узел, называемый хостом (host, хозяин). Под хостом для интерфейсов системного уровня подразумевается центр компьютера (ЦП и ОЗУ с ближайшим окружением); для периферийного уровня - компьютер с

интерфейсным адаптером (контроллером). Шины PCI, FireWire, SCSI - примеры ранжированных интерфейсов, USB, ATA - хост-центрического.

### **Топология интерфейса**

**Топология** определяет конфигурацию связей между соединяемыми устройствами: двухточечная, шинная, радиальная, смешанная, цепочечная.

**Двухточечная топология**, или выделенный интерфейс - самая простая связь, в которой участвуют только два устройства. Каждое из устройств "знает", что на противоположном конце интерфейса может быть лишь одно устройство, и задачи адресации не возникает. Двухточечный интерфейс с подчиненным устройством будет иметь самый простой протокол. Пример - LPT-порт компьютера в стандартном варианте, к которому подключено лишь одно устройство (принтер).

**Шинная топология** - объединение нескольких устройств на одной шине. Шина в данном контексте - совокупность сигнальных линий, соединяющих несколько устройств. Все устройства для обмена информацией пользуются одними и теми же линиями шины, они одновременно "слышат" друг друга. Для успешного обмена на время транзакции только одно устройство на шине может быть ведущим (задатчиком, инициатором); если на шине присутствует более одного устройства, претендующего на роль задатчика, то должен быть реализован протокол и механизм арбитража.

Арбитраж может выполняться централизованно специальным выделенным узлом-арбитром. Арбитраж может быть как простым, так и приоритетным, с различными механизмами управления приоритетом узлов. Пример шин с централизованным арбитражом - PCI, ISA; с распределенным арбитражом - SCSI, FireWire.

## **2.9. Классификация по природе сигналов информации**

1. **Электрические сигналы** - это электромагнитные колебания до десятков и сотен МГц, передаваемые по электрическим проводам ("0" и "1" уровень напряжения или присутствие тока).

2. Электромагнитные колебания с частотами в сотни МГц - десятки ГГц пригодны и для **беспроводной радиопередачи сигналов**. Более низкие частоты для эффективного излучения и приема требуют антенн неприемлемо большого размера; более высокие частоты вызывают сложности в реализации приемопередатчиков, и распространение сигнала в этом диапазоне имеет неприятную специфику (например, поглощение туманом и дождем).

Bluetooth - использует радиоволны микроволнового диапазона 2,4 ГГц. В этом диапазоне радиоволны распространяются по прямой (недостатки - отражение сигнала, взаимодействие с различной аппаратурой).

3. **Инфракрасный порт** (например порт IrDA) - направленные излучатели и приемники с углом охвата 15-30° и дальность действия порядка 1 м, так, что для соединения устройств их нужно должным образом приблизить друг к другу и сориентировать. Недостаток - малая зона охвата. Достижимая скорость - 4 Мбит/с, более высокие скорости проблематичны из-за слабости сигнала, получаемого фотоприемником. Инфракрасная связь используется и в беспроводных локальных сетях (IEEE 802.11 DFIR), где ненаправленные приемопередатчики за счет отражений от стен и потолка позволяют обеспечивать зону охвата порядка 10 м, но скорости еще ниже (1-2 Мбит/с).

Используются для беспроводного подключения периферии (принтеров и других устройств) к компьютерам.

4. **Проводная оптическая связь** - световые импульсы инфракрасного диапазона передаются по оптоволокну, стеклянному или пластиковому.

Стеклянное волокно в основном используется в телекоммуникациях, где требуется дальность связи. Достоинство - сочетание Гб/с и километровые расстояния.

Резервом повышения пропускной способности является мультиплексирование (по одной жиле может распространяться множество оптических сигналов с

различными длинами волн, не мешая друг другу). Они не чувствительны к электромагнитным излучениям.

## **2.10. Классификация видов интерфейсов**

**Командный интерфейс.** Командный интерфейс называется так по тому, что в этом виде интерфейса человек подает "команды" компьютеру, а компьютер их выполняет и выдает результат человеку. Командный интерфейс реализован в виде пакетной технологии и технологии командной строки.

**WIMP - интерфейс** (Window - окно, Image - образ, Menu - меню, Pointer - указатель) - диалог с пользователем ведется не с помощью команд, а с помощью графических образов - меню, окон, других элементов. Хотя и в этом интерфейсе подаются команды машине, но это делается "опосредственно", через графические образы. Этот вид интерфейса реализован на двух уровнях технологий: простой графический интерфейс и "чистый" WIMP - интерфейс.

**SILK - интерфейс** (Speech - речь, Image - образ, Language - язык, Knowledge - знание). Этот вид интерфейса наиболее приближен к обычной, человеческой форме общения. В рамках этого интерфейса идет обычный "разговор" человека и компьютера. При этом компьютер находит для себя команды, анализируя человеческую речь и находя в ней ключевые фразы. Результат выполнения команд он также преобразует в понятную человеку форму. Этот вид интерфейса наиболее требователен к аппаратным ресурсам компьютера, и поэтому его применяют в основном для военных целей.

## **Тема 3. Параллельный интерфейс**

**Интерфейс IDE** (Integrated Drive Electronics).

Основной интерфейс, используемый для подключения жесткого диска к современному PC, а также накопителей на магнитной ленте, CD/DVD-ROM, дисководов Zip и др.

Фактически он представляет собой связь между системной платой PC и электроникой жесткого диска, встроенной в накопитель или контроллером.

Наиболее распространенной формой IDE является ATA и Serial ATA.

Основная функция контроллера накопителя, или интерфейса, - передача данных из системы в накопитель и обратно. От типа интерфейса зависит, с какой скоростью будут осуществляться эти операции, что во многом определяет общую производительность компьютера.

Интерфейс IDE ATA является 16-разрядным параллельным интерфейсом, т.е. по кабелю интерфейса одновременно передается 16 бит. Serial ATA (SATA) единовременно передает по кабелю не более одного бита данных, что позволяет значительно уменьшить сечение и длину используемого кабеля за счет повышения частоты передачи данных.

Поскольку в накопителе IDE контроллер встроенный, его можно подключать непосредственно к разъему на плате адаптера или на системной плате. Это существенно упрощает установку жесткого диска, так как не нужно подсоединять отдельные кабели для подачи питания, сигналов управления и т.п. Кроме того, при объединении контроллера и жесткого диска сокращается общее количество элементов в устройстве, уменьшается длина соединительных проводов, а в результате повышается надежность, устойчивость к шумам и быстродействие системы по сравнению с тем, когда автономный контроллер подключается к жесткому диску с помощью длинных кабелей.

Объединяя контроллер (в том числе и входящий в его состав шифратор/дешифратор) с жестким диском, удастся существенно повысить надежность воспроизведения данных по сравнению с системами, в которых используются автономные контроллеры. Происходит это потому, что кодирование данных и их преобразование из цифровой формы в аналоговую (и наоборот)

осуществляется непосредственно в жестком диске при меньшем уровне внешних помех. В результате аналоговые сигналы, временные параметры которых весьма критичны, не передаются по плоским кабелям, где они могли бы “набрать” помех. Совмещение контроллера и жесткого диска в едином блоке позволило повысить тактовую частоту шифратора/дешифратора, плотность размещения данных на носителе и общее быстродействие системы.

На сегодняшний день интерфейс ATA IDE является самым распространенным.

Главное достоинство накопителей IDE - их дешевизна. Поскольку для них не нужен отдельный контроллер, количество кабелей и разъемов, необходимых для подключения жесткого диска, оказывается существенно меньшим, чем в стандартном варианте жесткого диска с автономным контроллером. Кроме того, эти устройства более надежны, поскольку контроллер встроен в жесткий диск и в результате шифратор/дешифратор расположен в непосредственной близости от носителя и поэтому менее чувствителен к внешним шумам и помехам.

Еще одно достоинство накопителей IDE – быстродействие, величина которого зависит от конкретной реализации.

Три основные разновидности интерфейса IDE, рассчитанные на взаимодействие с тремя стандартными шинами:

- параллельный AT Attachment (ATA) IDE (16-разрядная шина ISA);
- XT IDE (8-разрядная шина ISA);
- MCA IDE (16-разрядная шина MCA).

## **Тема 4. Последовательный интерфейс**

### **4.1. Способы последовательной передачи**

Последовательный интерфейс для передачи данных использует одну сигнальную линию, по которой информационные биты передаются друг за другом

последовательно. В ряде последовательных интерфейсов применяется гальваническая развязка внешних (обычно входных) сигналов от схемной земли устройства, что позволяет соединять устройства, находящиеся под разными потенциалами.

Примеры последовательного интерфейса: интерфейсы RS-232C, RS-422A, RS-423A, RS-485, токовая петля, MIDI, а также COM-порт.

При **асинхронной передаче** каждому байту предшествует старт-бит, сигнализирующий приемнику о начале посылки, за которым следуют биты данных и, возможно, бит паритета (четности,). Завершает посылку стоп-бит, гарантирующий паузу между посылками. Старт-бит следующего байта посылается в любой момент после стоп-бита, то есть между передачами возможны паузы произвольной длительности. Старт-бит, имеющий всегда строго определенное значение (логический 0), обеспечивает простой механизм синхронизации приемника по сигналу от передатчика. Подразумевается, что приемник и передатчик работают на одной скорости обмена. Чем выше частота передачи, тем больше влияние искажений фронтов на фазу принимаемого сигнала.

Для асинхронного режима принят ряд стандартных скоростей обмена: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19 200, 38 400, 57 600 и 115 200 бит/с.

Асинхронный обмен в РС реализуется с помощью COM-порта с использованием протокола RS-232C.

**Синхронный** режим передачи предполагает постоянную активность канала связи. Посылка начинается с синхробайта, за которым сразу же следует поток информационных бит. Если у передатчика нет данных для передачи, он заполняет паузу непрерывной посылкой байтов синхронизации. Очевидно, что при передаче больших массивов данных накладные расходы на синхронизацию в данном режиме будут ниже, чем в асинхронном. Однако в синхронном режиме необходима внешняя синхронизация приемника с передатчиком, поскольку даже малое отклонение частот приведет к искажению принимаемых данных.

## 4.2. Интерфейс RS-232c

Интерфейс предназначен для подключения аппаратуры, передающей или принимающей данные, к оконечной аппаратуре каналов данных. В роли аппаратуры передачи данных может выступать компьютер, принтер, плоттер и другое периферийное оборудование. В роли оконечной аппаратуры канала данных обычно выступает модем.

Стандарт описывает управляющие сигналы интерфейса, пересылку данных, электрический интерфейс и типы разъемов. В стандарте предусмотрены асинхронный и синхронный режимы обмена, но СОМ-порты поддерживают только асинхронный режим.

Интерфейс не обеспечивает гальванической развязки устройств. Логической единице соответствует напряжение на входе приемника в диапазоне - 12 - 3 В.

Подключение и отключение интерфейсных кабелей устройств с автономным питанием должно производиться при отключенном питании.

Стандарт RS-232C регламентирует типы применяемых разъемов.

#### **4.3. Интерфейс токовая петля**

В интерфейсе токовая петля электрическим сигналом является не уровень напряжения относительно общего провода, а *ток* в двухпроводной линии, соединяющей приемник и передатчик. Логической единице (состоянию "включено") соответствует протекание тока **20 мА**, а логическому нулю - отсутствие тока. Такое представление сигналов для описанного формата асинхронной отправки позволяет обнаружить обрыв линии - приемник заметит отсутствие стоп-бита.

Токовая петля обычно предполагает гальваническую развязку входных цепей приемника от схемы устройства. При этом источником тока в петле является передатчик (этот вариант называют активным передатчиком).

Токовая петля с гальванической развязкой позволяет передавать сигналы на расстояния до нескольких километров. Преобразовать сигналы *RS-232C* в токовую петлю можно с помощью специальных конвекторов.

#### 4.4. Интерфейс IEEE-1394 (Fire Wire,i.LINK)

IEEE-1394, FireWire ("огненный провод" - зарегистрированный товарный знак интерфейса IEEE-1394 фирмы Apple) и i. LINK - это три названия одного и того же **высокоскоростного цифрового последовательного интерфейса**, который служит для передачи любых видов цифровой информации.

Интерфейс IEEE-1394 - тонкий кабель для последовательной передачи сигналов, позволяет подключение в горячем режиме (не требуется перезагружать компьютер (пользователи могут подключать и отключать любые IEEE-1394-устройства, когда шина находится в активном состоянии - они немедленно автоматически распознаются и включаются в систему).

Расширяемая архитектура позволяет использовать подключенные к шине устройства с пропускной способностью в 100, 200, 400 и 800 Мбит/с.

Гибкая топология соединений (звездой или разветвленным деревом) и равноправие устройств позволяют упростить подключение для обмена данными между устройствами.

Открытая архитектура означает отсутствие необходимости использования специального программного обеспечения.

По интерфейсу IEEE-1394 возможны два типа передачи данных: асинхронный и изохронный. Асинхронная передача реализуется по традиционному компьютерному интерфейсу. Изохронные каналы обеспечивают гарантированную передачу данных с предопределенной скоростью.

Возможно, наиболее важным фактором для использования IEEE-1394 в качестве единого цифрового интерфейса для бытовой электронной техники является то, что он предоставляет возможность напрямую подключать устройства друг к другу. Это позволяет, например, производить перезапись с источника на

приемник без помощи компьютера (например, переписывать видео с видеокамеры на магнитофон) или совместно использовать одно устройство несколькими компьютерами без какой-либо специальной поддержки.

Область применения IEEE-1394 - обеспечить высокоскоростной доступ к устройствам хранения информации, таким как жесткие диски, приводы CD и DVD со скоростями до 800 Мбит/с. Связь между устройствами с интерфейсом IEEE-1394 может включаться и выключаться непосредственно на время их работы (так называемое горячее подключение) без отключения питания и перезагрузки.

Таким образом, аудио - и видеооборудование можно интегрировать с компьютерами и таким образом управлять ими простым соединением устройств друг к другу с помощью одного кабеля. После этого при помощи персонального компьютера, выступающего в качестве контроллера, можно производить операции: записывать с CD-проигрывателя на мини-диск, запоминать цифровые радиопередачи, вводить цифровое видео в персональный компьютер для последующего монтажа и редактирования. Разумеется, при этом сохраняется возможность и прямого обмена данными между аудио - и видеооборудованием без использования компьютера.

С интерфейсом IEEE-1394 в последнее время активно конкурирует USB-интерфейс.

Таким образом, IEEE-1394 остается международным стандартом недорогого интерфейса, который позволяет объединять всевозможные цифровые устройства для развлечений, коммуникации и вычислительную технику в бытовой мультимедийный цифровой комплекс.

#### **4.5. Интерфейс SATA**

SATA ([англ. Serial ATA](#)) - последовательный [интерфейс](#) обмена данными с накопителями информации. SATA является развитием параллельного интерфейса [ATA](#) (IDE), который после появления SATA был переименован в PATA (Parallel ATA).

Данный интерфейс наиболее популярен для современных жестких дисков и оптических приводов домашнего использования, например мультимедиа проигрывателей.

Максимальная пропускная способность до 600 Мбайт/с. Максимальная длина кабеля при этом составляет 1 м. Внутри кабеля Serial ATA находятся 2 пары сигнальных проводов (одна пара на прием, другая - на передачу), отделенных тремя жилами общего провода ("земли").

SATA использует 7-контактный разъём вместо 40-контактного разъёма у PATA. SATA-кабель за счёт своей формы более устойчив к многократному подключению. Стандарт SATA отказался от традиционного для PATA подключения по два устройства на шлейф; каждому устройству полагается отдельный кабель.

## **Тема 5. Беспроводный интерфейс**

Беспроводные интерфейсы позволяют освободить устройства от связывающих их интерфейсных кабелей, что особенно привлекательно для малогабаритной периферии, по размеру и весу соизмеримой с кабелями. В беспроводных интерфейсах используются электромагнитные волны инфракрасного (IrDA) и радиочастотного (Blue Tooth) диапазонов. Кроме этих интерфейсов периферийных устройств существуют и беспроводные способы подключения к локальным сетям.

### **5.1. Инфракрасный интерфейс IRDA.**

Применение излучателей и приемников инфракрасного (ИК) диапазона позволяет осуществлять беспроводную связь между парой устройств, удаленных на расстояние до нескольких метров. Инфракрасная связь - *IR (Infra Red) Connection* - безопасна для здоровья, не создает помех в радиочастотном диапазоне и обеспечивает конфиденциальность передачи. ИК-лучи не проходят через стены, поэтому зона приема ограничивается небольшим, легко

контролируемым пространством. Инфракрасная технология привлекательна для, например связи портативных компьютеров со стационарными компьютерами.

Различают инфракрасные системы низкой (до 115,2 Кбит/с), средней (1,152 Мбит/с) и высокой (4 Мбит/с) скорости. Низкоскоростные системы служат для обмена короткими сообщениями, высокоскоростные - для обмена файлами между компьютерами, подключения к компьютерной сети, вывода на принтер, проекционный аппарат и т.п.

Излучателем для ИК-связи является светодиод, имеющий пик спектральной характеристики мощности 880 нм; светодиод дает конус эффективного излучения с углом около 30°. В качестве приемника используют PIN-диоды, эффективно принимающие ИК-лучи в конусе 15°. Спецификация IrDA определяет требования к мощности передатчика и чувствительности приемника, причем для приемника задается как минимальная, так и максимальная мощность ИК-лучей. Импульсы слишком малой мощности приемник не "увидит", а слишком большая мощность "ослепляет" приемник - принимаемые импульсы сольются в неразличимый сигнал. Кроме полезного сигнала на приемник воздействуют помехи: засветка солнечным освещением и лампами накаливания, дающая постоянную составляющую оптической мощности, и помехи от люминесцентных ламп, дающие переменную (но низкочастотную) составляющую.

Приемопередатчик **IrDA** может быть подключен к компьютеру различными способами; по отношению к системному блоку он может быть как внутренним (размещаемым на лицевой панели), так и внешним, размещаемым в произвольном месте. Размещать приемопередатчик следует с учетом угла «зрения» (30° у передатчика и 15° у приемника) и расстояния до требуемого устройства (до 1 м).

**Внешние ИК-адаптеры** выпускают с интерфейсом RS-232C для подключения к COM-порту или же с шиной USB.

Для прикладного использования **IrDA** кроме физического подключения адаптера и трансивера требуется установка и настройка соответствующих драйверов.

В ОС Windows 9x/ME/2000 контроллер **IrDA** попадает в «Сетевое окружение». Сконфигурированное ПО позволяет:

- устанавливать соединение с локальной сетью (для выхода в Интернет, использования сетевых ресурсов);
- передавать файлы между парой компьютеров;
- выводить данные на печать;
- синхронизировать данные PDA, мобильного телефона и настольного компьютера;
- выгружать отснятые изображения из фотокамеры в компьютер и выполнять ряд других полезных действий, не заботясь ни о каком кабельном хозяйстве.

## **5.2. Радиointерфейс Bluetooth**

Bluetooth (синий зуб) - это фактический стандарт на миниатюрные недорогие средства передачи информации с помощью радиосвязи между мобильными (и настольными) компьютерами, мобильными телефонами и любыми другими портативными устройствами на небольшие расстояния.

Группа лидирующих фирм в областях телекоммуникаций, компьютеров и сетей — Ericsson, IBM, Intel, Microsoft, Motorola, Nokia, Toshiba., образовала Bluetooth Special Interest Group, и вывела данную технологию на рынок.

Каждое устройство ВТ имеет радиопередатчик и приемник, работающие в диапазоне частот 2,4 ГГц. Этот диапазон в большинстве стран отведен для промышленной, научной и медицинской аппаратуры и не требует лицензирования, что обеспечивает повсеместную применимость устройств. Кодирование простое - логической единице соответствует положительная девиация частоты, нулю — отрицательная

Передача ведется с перескоком несущей частоты с одного радиоканала на другой, что помогает в борьбе с интерференцией и замираниями сигнала. Физический канал связи представляется определенной псевдослучайной последовательностью используемых радиоканалов (79 или 23 возможных частот). Группа устройств, разделяющих один канал (то есть "знающих" одну и ту же последовательность перескоков), образует так называемую пикосеть (piconet), в которую может входить от 2 до 8 устройств. В каждой пикосети имеется одно ведущее устройство и до 7 активных ведомых. Кроме того, в зоне охвата ведущего устройства в его же пикосети могут находиться "припаркованные" ведомые устройства: они тоже "знают" последовательность перескоков и синхронизируются (по перескокам) с ведущим устройством, но не могут обмениваться данными до тех пор, пока ведущее устройство не разрешит их активность. Каждое активное ведомое устройство пикосети имеет свой временный номер (1-7); когда ведомое устройство деактивируется (паркуется), оно отдает свой номер для использования другими. При последующей активизации оно уже может получить иной номер (потому-то он и временный). Пикосети могут перекрываться зонами охвата, образуя "разбросанную" сеть (scatternet). При этом в каждой пикосети ведущее устройство только одно, но ведомые устройства могут входить в несколько пикосетей, используя разделение времени. Передачи ведутся пакетами, каждый пакет может занимать от 1 до 5 тайм-слотов

Канал делится на тайм-слоты длительностью 625 мкс. Каждый тайм-слот соответствует одной частоте несущей в последовательности перескоков (1600 перескоков в секунду). Последовательность частот определяется адресом устройства-мастера пикосети. Ведущий и ведомые устройства ведут передачу поочередно: в четных слотах передачу ведет ведущий, а в нечетных — адресованное им ведомое устройство (если ему есть что «сказать»).

. Между ведущим и ведомыми устройствами могут устанавливаться физические связи двух типов: **синхронные и асинхронные.**

Синхронные связи используются для передачи изохронного трафика. Эти связи типа "точка-точка" предварительно устанавливает ведущее устройство с выбранными ведомыми устройствами, и для каждой связи определяется период (в слотах), через который для нее резервируются слоты. Связи получаются симметричные двусторонние. Повторные передачи пакетов в случае ошибок приема не используются. Ведущее устройство может установить до трех связей SCO с одним или разными ведомыми устройствами. Ведомое устройство может иметь до трех связей с одним ведущим устройством или иметь по одной связи SCO с двумя различными ведущими устройствами. По сетевой классификации связи SCO относятся к коммутации цепей.

Асинхронные связи без установления соединения реализуют коммутацию пакетов по схеме "точка-множество точек" между ведущим устройством и всеми ведомыми устройствами пикосети. Ведущее устройство может связываться с любым из ведомых устройств пикосети в слотах, не занятых под SCO, послав ему пакет и потребовав ответа. Ведомое устройство имеет право на передачу, только получив обращенный к нему запрос ведущего устройства (безошибочно декодировав свой адрес). Ведущее устройство может посылать и безадресные широковещательные пакеты для всех ведомых устройств своей пикосети. С каждым из своих ведомых устройств ведущее устройство может установить лишь одну связь ACL.

Защита данных от искажения и контроль достоверности производится несколькими способами. Данные некоторых типов пакетов защищаются CRC-кодом.

Каждый голосовой канал обеспечивает скорость по 64 Кбит/с в обоих направлениях. Асинхронный канал может обеспечивать максимальную скорость 723,2 Кбит/с в асимметричной конфигурации. Для обеспечения безопасности в ВТ применяется аутентификация и шифрование данных на уровне связи (link layer), которые, конечно же, могут дополняться и средствами верхних протокольных уровней.

## Тема 6. Шина USB

### 6.1. Общие положения

USB обеспечивает обмен данными между хост-компьютером и множеством периферийных устройств (ПУ). Согласно спецификации USB, устройства (devices) могут являться хабами, функциями или их комбинацией.

Устройство - хаб (hub) обеспечивает дополнительные точки подключения устройств к шине.

Устройство-функция (function) USB предоставляет системе дополнительные функциональные возможности, например подключение к ISDN, цифровой джойстик, акустические колонки с цифровым интерфейсом и т.п.

*Комбинированное устройство* (compound device), содержащее несколько функций, представляется как хаб с подключенными к нему несколькими устройствами.

Работой всей системы USB управляет *хост-контроллер* (host controller), являющийся программно-аппаратной подсистемой хост-компьютера. Шина позволяет подключать, конфигурировать, использовать и отключать устройства во время работы хоста и самих устройств.

Шина USB является хост-центрической: единственным ведущим устройством, которое управляет обменом, является хост-компьютер, а все присоединенные к ней периферийные устройства - исключительно ведомые. Физическая топология шины USB - многоярусная звезда. Ее вершиной является хост-контроллер, объединенный с корневым хабом. Каждый промежуточный хаб имеет несколько *нисходящих* (downstream) портов для подключения периферийных устройств (или нижележащих хабов) и один *восходящий* (upstream) порт для подключения к корневому хабу или нисходящему порту вышестоящего хаба.

Хаб является ключевым элементом системы в архитектуре USB и выполняет множество функций:

- обеспечивает физическое подключение устройств, формируя и воспринимая сигналы в соответствии со спецификацией шины на каждом из своих портов;
- управляет подачей питающего напряжения на нисходящие порты, причем предусматривается установка ограничения на ток, потребляемый каждым портом;
- отслеживает состояние подключенных к нему устройств, уведомляя хост об изменениях;
- обнаруживает ошибки на шине, выполняет процедуры восстановления и изолирует неисправные сегменты шины;
- обеспечивает связь сегментов шины, работающих на разных скоростях.

## **6.2. Модель передачи данных**

Каждое устройство на шине USB (их может быть до 127) при подключении автоматически получает свой уникальный адрес.

Каждая конечная точка имеет свой номер и описывается следующими параметрами:

- требуемая частота доступа к шине и допустимые задержки обслуживания;
- требуемая полоса пропускания канала;
- требования к обработке ошибок;
- максимальные размеры передаваемых и принимаемых пакетов;
- тип передачи;
- направление передачи

*Каналом* в USB называется модель передачи данных между хост-контроллером и конечной точкой устройства. Имеются два типа каналов: потоки и сообщения. Каналы организуются при конфигурировании устройств USB.

*Поток* (stream) доставляет данные от одного конца канала к другому, он всегда однонаправленный. *Сообщение* (message) имеет формат, определенный спецификацией USB.

### **6.3. Типы передачи данных**

Архитектура USB допускает четыре базовых типа передачи данных.

*Управляющие посылки* используются для конфигурирования устройств во время их подключения и для управления устройствами в процессе работы. Протокол обеспечивает гарантированную доставку данных.

*Передачи массивов данных* - это передачи без каких-либо обязательств по задержке доставки и скорости передачи. Приоритет этих передач самый низкий, они могут приостанавливаться при большой загрузке шины. Доставка гарантированная - при случайной ошибке выполняется повтор. Передачи массивов уместны для обмена данными с принтерами, сканерами, устройствами хранения и т.п.

*Прерывания* - короткие передачи, которые имеют спонтанный характер и должны обслуживаться не медленнее, чем того требует устройство. При случайных ошибках обмена выполняется повтор. Прерывания используются, например, при вводе символов с клавиатуры или для передачи сообщения о перемещении мыши.

*Изохронные передачи* - непрерывные передачи в реальном времени, занимающие предварительно согласованную часть пропускной способности шины с гарантированным временем задержки доставки. В случае обнаружения ошибки изохронные данные не повторяются - недействительные пакеты игнорируются.

Архитектура USB предусматривает внутреннюю буферизацию всех устройств. Шина USB должна обеспечивать обмен с такой скоростью, чтобы задержка в устройстве, вызванная буферизацией, не превышала нескольких миллисекунд.

## **Тема 7. Интерфейсы периферийных устройств**

### **7.1. Интерфейсы накопителей на магнитных жестких дисках.**

Первый жесткий диск имел емкость 5 Мб. Устройство состояло из 50 дисков диаметром 24 дюйма с вращающейся частотой 1200 оборотов в минуту. Среднее время поиска составляло около 1 сек. Современные накопители занимают значительно меньше пространства. Время поиска сократилось на несколько порядков. Но главным изменением параметров было и остается количество информации, записываемой на единицу поверхности, т.е. плотность записи.

Плотность записи определяется двумя параметрами:

- плотность дорожек, которая измеряется в количестве дорожек на дюйм;
- линейной плотностью – количеством битов на заданной длине одной дорожки.

Оба эти параметра зависят от того, сколько пространства на диске занимает 1 бит, чем выше плотность записи, тем больше дорожек помещается на диске и тем больше битов информации можно разместить на одной пластине диска, вследствие этого увеличивается скорость поиска и объем данных и уменьшаются размеры самого накопителя.

#### **7.1.1. Основные физические и логические параметры жестких дисков.**

1. **Диаметр диска.** Наиболее распространены – накопители с диаметрами 2,2; 2,3; 3,14; 5,25 дюйма.
2. **Число поверхностей** – определяет количество физических дисков, нанизанных на вал. Однако наиболее распространены устройства с числом поверхностей от 2х до 5ти. Так как операции выполняются всеми головками синхронно, то при равных остальных условиях более быстрыми являются накопители с большим

числом поверхностей.

3. **Число цилиндров** – этот параметр определяет, сколько дорожек (треков) будет располагаться на одной поверхности. В настоящее время все накопители емкостью более 1 Гб имеют число цилиндров более 1024.

4. **Число секторов** – определяет общее число секторов на всех дорожках всех поверхностей накопителя.

5. **Число секторов на дорожке** – общее число секторов на одной дорожке. Число секторов на внешних и внутренних дорожках неравное.

6. **Частота вращения цилиндров** – определяет сколько времени будет затрачено на последовательное считывание одной дорожки или цилиндра. Для дисков емкостью до 1 Гб она обычно равна 5400 оборотам в минуту, а у более вместительных достигает 7200 об/мин.

7. **Время перехода от одной дорожки к другой** – обычно составляет от 3.5 до 5 миллисекунд, а у самых быстрых моделей может быть от 0.6 до 1 миллисекунды. Этот показатель является одним из определяющих быстродействие накопителя, т.к. именно переход с дорожки на дорожку является самым длительным процессом в серии процессов произвольного чтения/записи на дисковом устройстве.

8. **Время успокоения головок** – время, проходящее с момента окончания позиционирования головок на требуемую дорожку до момента начала операции чтения/записи.

9. **Время установки или время поиска** – время, затрачиваемое устройством на перемещение головок чтения/записи к нужному цилиндру из произвольного положения.

10. **Среднее время установки или поиска** – усредненный результат большого числа операций позиционирования на разные цилиндры, часто называют средним временем позиционирования. Например, для 540-мегабайтных дисков наиболее

типичны величины от 10 до 13, а для дисков свыше гигабайта - от 7 до 10 миллисекунд.

11. **Время ожидания** – время, необходимое для прохода нужного сектора к головке, усредненный показатель – среднее время ожидания, получаемое как среднее от многочисленных тестовых проходов. После успокоения головок на требуемом цилиндре контроллер ищет нужный сектор. Это время у накопителей объемом от 540 мегабайт до 1 гигабайта составляет примерно 5.6, а у дисков свыше гигабайта - 4.2 миллисекунды и менее.

12. **Время доступа** – суммарное время, затрачиваемое на установку головок и ожидание сектора. Причем, наиболее долгим является промежуток времени установки головок.

13. **Среднее время доступа к данным** – время, проходящее с момента получения запроса на операцию чтения/записи от контроллера до физического осуществления операции - результат сложения среднего время поиска и среднего времени ожидания. Среднее время доступа зависит от того, как организовано хранение данных и насколько быстро позиционируются головки чтения записи на требуемую дорожку, обычно, оно составляет от 10 до 18 миллисекунд.

14. **Скорость передачи данных**, называемая также пропускной способностью, определяет скорость, с которой данные считываются или записываются на диск после того, как головки займут необходимое положение. Измеряется в мегабайтах в секунду (MBps) или мегабитах в секунду (Mbps) и является характеристикой контроллера и интерфейса. Различают две разновидности скорости передачи - внешняя и внутренняя.

15. **Внешняя скорость передачи данных** показывает, с какой скоростью данные считываются из буфера, расположенного на накопителе в оперативную память компьютера с интерфейсами EIDE или Fast ATA, обычно от 11.1 до 16.6 мегабайта в секунду, а для накопителей с интерфейсами SCSI-2 - от 10 до 40 мегабайт в

секунду.

16. **Внутренняя скорость передачи данных** - отражает скорость передачи данных между головками и контроллером накопителя и определяет общую скорость передачи данных в тех случаях, когда буфер не используется или не влияет (например, когда загружается большой графический или видеофайл).

17. **Размер кэш-буфера контроллера**. Встроенный в накопитель буфер выполняет функцию упреждающего кэширования и призван сгладить громадную разницу в быстродействии между дисковой и оперативной памятью компьютера. Выпускаются накопители с 128, 256 и 512 килобайтным буфером. Чем больше объем буфера, тем потенциально выше производительность при произвольном "длинном" чтении/записи.

18. **Средняя потребляемая мощность (capacity)**. Современные накопители на ЖД потребляют от 5 до 15 Ватт.

19. **Уровень шума (noise level)**, разумеется, является эргономическим показателем и показателем сбалансированности механической конструкции, т.к. шум приводят к более быстрому их износу.

20. **Среднее время наработки на отказ (MTBF)** – определяет, сколько времени способен проработать накопитель без сбоев.

#### **7.1.2. Физический и логический объем накопителей.**

Носители жестких дисков, в отличие от гибких, имеют постоянное число дорожек и секторов, изменить которое невозможно. Эти числа определяются типом модели и производителем устройства. Поэтому, физический объем жестких дисков определен изначально и зависит от типа интерфейса, метода кодирования данных, используемого физического формата и др.

Одним из способов повышения физической емкости является увеличение емкости сектора. В настоящее время, стандартной емкостью сектора для IBM-

совместимых компьютеров является 512 байт. Многие адаптеры позволяют, в процессе физического форматирования, программным путем, изменять емкость сектора, например, до 1024 байт.

Логический объем зависит от того, как операционная система или программа записывает информацию в сектора.

### 7.1.3. Скоростные характеристики жестких дисков (HDD)

Помимо объема, как правило, представляют интерес и скоростные характеристики накопителей. Из них можно выделить две основных: среднее время доступа и скорость линейной передачи данных.

**Время доступа** – время с момента запроса данных до момента их получения. Временем выполнения программы здесь можно пренебречь. **Время ожидания** равно 5.56 мс при частоте вращения 5400 об/мин и 4.17 мс - при 7200 об/мин.

**Время позиционирования** же складывается из времени необходимого на перемещение головки и времени для успокоения ее колебаний после перемещения.

Для современных дисков **время доступа** составляет порядка 15 мс, а скорость линейной передачи данных – порядка 30 Мбайт/с. Информация читается чаще всего либо покластерно, т.е. по 4 Кбайта, либо максимальными фрагментами, поддерживаемыми BIOS – 64 Кбайта. Кроме того, объем однократно считанного фрагмента информации никогда не превышает размера файла, а средний размер файла, как правило, не превосходит нескольких килобайт. Поэтому определяющий вклад в производительность дисковой системы вносит время доступа.

Революцию во времени доступа следует связывать с переходом на твердотельные накопители.

Наиболее эффективным средством повышения производительности

дисковой системы является **кэширование**, т.е. хранение в оперативной памяти наиболее часто используемых данных с жесткого диска. Ведь для доступа к определенному байту, расположенному на диске, требуется около 15 мс, а к расположенному в оперативной памяти – порядка 0.1 мкс. Увеличение объема буфера еще более ускорит этот процесс, поэтому, во-первых, сами накопители содержат буфер объемом, как правило, от 2 до 8 Мбайт, а во-вторых, кэширование производится на уровне ОС.

#### **7.1.4. Контроллеры жестких дисков**

Собственно контроллер накопителя физически расположен на плате электроники и предназначен для обеспечения операций преобразования и пересылке информации от головок чтения/записи к интерфейсу накопителя. Контроллер жестких дисков представляет собой сложнейшее устройство - микрокомпьютер, со своим процессором, ОЗУ и ПЗУ, схемами и системой ввода/вывода и т.п., размещенных в одном или двух микрочипах.

#### **7.1.5. Характеристики интерфейсов жестких дисков**

***IDE/ATA.*** IDE (Integrated Drive Electronics) - это название типа жестких дисков, имеющих интерфейс ATA (AT Attachment).

Один канал ATA может поддерживать до двух дисков, первый - master и вторичный - slave. Но так как IDE может обращаться только к одному устройству на канале одномоментно, то таким образом снижается производительность системы в целом. Все материнские платы имеют по два интегрированных канала IDE, а некоторые и четыре.

Сегодня на рынке присутствуют три основных стандарта IDE дисков: ATA/33 (33Мб/сек), ATA/66 и ATA/100. В данном случае число показывает максимальную пропускную способность в мегабайтах в секунду.

Главным недостатком IDE по-прежнему остается малая скорость,

ограничение по длине кабеля. Преимущества: неплохая производительность, низкая стоимость, совместимость с большинством существующего оборудования.

***SerialATA.*** Вместо параллельного интерфейса ATA вполне удовлетворяющего требованиям современных жестких дисков, принят новый стандарт Serial ATA, который будет поддерживать все накопители, включая винчестеры, CD, DVD, флоппи-дисководы и другие подобные устройства при подключении их к системным платам.

Вместо передачи данных в параллельном режиме по многожильному кабелю, он использует принцип последовательной передачи данных с пропускной способностью до 6 Гбит/с, при этом сохранится обратная совместимость с Serial ATA первой версии.

Интерфейс предназначен для подключения внутренних устройств, поэтому максимальная длина кабеля относительно невелика — 1 метр, тем не менее, это в 2 раза больше, чем у параллельного ATA интерфейса (45 см). Всего кабель состоит из 7 жил: две дифференциальные пары для передачи и для приема и три общих (заземляющих) провода. Контроллер Serial ATA рассчитан на подключение 2 устройств по схеме point-to-point, то есть каждое из них подключается отдельным кабелем и обмен информацией происходит параллельно. В целом последовательный интерфейс Serial ATA рассчитан на те же сферы применения, что и параллельный ATA (Parallel ATA, PATA) — персональные компьютеры любого уровня и ноутбуки, а так же в серверах начального уровня.

***SCSI.*** SCSI - интерфейс для рабочих станций и серверов. Дороже IDE, имеет гораздо большую пропускную способность, поддерживает большее количество устройств на одном канале, гораздо большую длину кабелей (до 12 метров).

Узкая шина SCSI может нести на себе до 8 устройств, а широкая (wide) до 16.

Существует множество различных вариантов SCSI - Ultra, Ultra2 и Ultra160 SCSI.

Ultra SCSI позволяет передачу 20Мб/сек и имеет 8 адресов. Широкая версия Ultra SCSI до 40Мб/сек. Ultra2 SCSI, а Ultra160 SCSI при 16 устройствах на канале и 160Мб/сек.

Самые быстрые, самые объемистые жесткие диски имеют интерфейс SCSI.

Преимущества: большая производительность, большие объемы, возможность подключения, как внутренних устройств, так и внешних.

Недостатки: дороговизна.

***Fibre Channel (оптоволоконный канал).*** Fibre channel - это интерфейс, в корне отличающийся от SCSI и IDE. Интерфейс предназначен для организации сетей, объединения удаленных друг от друга массивов жестких дисков, и прочих операций требующих высокой пропускной способности в сочетании с большими расстояниями, например для соединения с сетью рабочей группы либо сервером.

Пропускная способность Fibre channel в 100Мбит/сек, а теоретический предел данной технологии лежит где-то в районе 1.06Гбит/сек.

Если SCSI ограничивается всего 12 метрами, то Fibre channel позволяет соединения протяженностью до 10км при использовании оптического кабеля и несколько меньше при использовании относительно недорогих медных соединений.

Преимущества: хорошая масштабируемость, большие расстояния соединений (до 10км), высокая пропускная способность.

Недостаток - высокая стоимость.

***IEEE 1394.*** IEEE 1394, он же FireWire используется для передачи цифрового видео, подключения жестких дисков, сканеров, сетевого оборудования,

цифровых камер, и всего, что требует хорошей пропускной способности. FireWire способен поддерживать до 63 устройств на одном канале 400Мбит/сек. А IEEE 1394b – имеет пропускную способность в 800Мбит/сек на канал.

На базе FireWire можно строить производительные локальные сети. Многие модели компьютеров Apple имеют один или два FireWire порта.

Преимущества - высокая пропускная способность, отсутствие разделения устройств по приоритетам.

Недостаток - контроллеры жестких дисков стоят очень дорого.

## **7.2. Интерфейсы оптических дисков**

CD-ROM диск - это общее название ряда цифровых носителей информации, предназначенных для использования в компьютерных системах в качестве постоянного запоминающего устройства.

С точки зрения физического устройства CD-ROM диск полностью идентичен звуковому CD-DA диску, и отличается лишь логической структурой дорожки (дорожек).

Технологически стандартный диск должен состоять из трех слоев: подложка из поликарбонатного пластика, на которой прессом отштампован рельеф диска, напыленное на нее отражающее покрытие из алюминия.

Информация записана на диск в виде спиральной дорожки, идущей от центра к краю диска, на которой расположены углубления (так называемые питы). Информация кодируется чередованием питов (условно - логической 1) и промежутков между ними (условно - логических 0). Дорожка может быть непрерывной, либо делиться на фрагменты (например сессии в мультисессионных дисках). Число сессий в настоящее время не может превышать 64, и наличие нескольких сессий допустимо не во всех стандартах записи.

### **7.2.1. Способы чтения компакт-дисков**

Существует два способа чтения компакт-дисков.

При использовании первого способа чтения скорость передачи данных все время остается постоянной. Этот способ называется чтением данных при постоянной линейной скорости. Так как данные записываются на диск по спирали с одинаковой плотностью, чем ближе к центру диска считываемые данные, тем должна быть большая скорость вращения диска.

При использовании способа чтения с постоянной угловой скоростью скорость вращения диска остается постоянной, а скорость передачи данных варьируется и достигает минимального значения при считывании дорожек, расположенных ближе всего к центру.

### **7.2.2. Интерфейсы CD-ROM**

PCMCIA - интерфейс, применяемый для компактных внешних CD-ROM, подключаемых к маленьким компьютерам-ноутбукам.

IDE - интерфейс, применяемый обычно для подключения HDD, использующий 40-контактный кабель. На одном канале IDE (то есть на одном кабеле) могут находиться одно или два устройства, в последнем случае одно из устройств является ведущим (Master), а второе - ведомым (Slave).

### **7.2.3. Скоростные характеристики CD-ROM приводов**

В качестве стандартной скорости вращения для CD-ROM принята скорость вращения аудио дисков в обычных плеерах. При такой скорости вращения диска поток данных с CD-ROM диска (для формата ISO9660 с сектором 2048 байт) составляет 150 кб/с.

Драйвы, обеспечивающие больший поток, соответственно называются X-

скоростными. Например, драйв, обеспечивающий чтение CD-ROM с потоком данных в 600 кБ/с, называется 4х-скоростным.

### 7.3. Интерфейсы мониторов

До 50-х годов компьютеры выводили информацию только на печатающие устройства. Достаточно часто компьютеры тех лет оснащались осциллографами, которые использовались не для вывода информации, а для проверки электрических цепей вычислительной машины. И лишь в 1950 г. в Кембриджском университете электронно-лучевая трубка (ЭЛТ, или CRT, Cathode Ray Tube) осциллографа была использована для вывода графической информации.

Все широко распространенные современные мониторы, по схеме формирования изображения, делятся на два типа:

- на основе электронно-лучевой трубки (ЭЛТ, или CRT);
- на основе жидких кристаллов (ЖК-панель, LCD-панель).

Чем больше экран, тем с большим разрешением можно на нем работать. Но при этом непропорционально высоко возрастает его цена и увеличивается требуемое место для монитора на столе

**ЭЛТ-мониторы** очень похожи на телевизоры. У них тот же принцип формирования сигнала – направленный электронный пучок вызывает свечение точек на экране. Этот тип мониторов позволяет создание изображения с максимальной контрастностью, яркостью и цветностью. Наиболее вредными для здоровья являются ЭЛТ-мониторы, прежде всего, за счет рентгеновского излучения, возникающего из-за торможения электронов в трубке, и паразитного ультрафиолетового излучения монитора.

Их недостатки – высокое потребление электроэнергии и вред, наносимый здоровью.

**ЖК-мониторы** формируют изображение за счет того, что определенные точки экрана становятся прозрачными или непрозрачными в зависимости от

приложенного электрического поля. Поскольку жидкокристаллические ячейки сами не светятся, ЖК-мониторам нужна подсветка. ЖК-мониторы имеют малое потребление энергии, изображение на них приятно глазам, отсутствует радиационное излучение монитора. Их недостатки – малая контрастность изображения и малые скорости регенерации (обновления изображения) экрана.

### **7.3.1. Основные параметры и характеристики мониторов**

***Физические*** - размер рабочей области экрана

Размер экрана - это размер по диагонали от одного угла экрана до другого. Для ЭЛТ стандартными являются размеры 14", 15", 17", 19", 21", 23", 24" (" – обозначение дюйма.) Для ЖК-мониторов – 13", 14", 15", 17", 19".

У ЖК-мониторов номинальный размер диагонали экрана равен видимому.

***Частотные характеристики мониторов***

**Частота развертки** монитора показывает, какое предельное число, например горизонтальных строк на экране монитора может прочертить электронный луч за одну секунду. Соответственно, чем выше это значение (а именно оно, как правило, указывается на коробке для монитора) тем выше разрешение может поддерживать монитор при приемлемой частоте кадров.

Это параметр, определяющий, как часто изображение на экране заново перерисовывается. Частота горизонтальной развертки в Гц. В случае с традиционными ЖК мониторами время свечения люминофорных элементов очень мало, поэтому электронный луч должен проходить через каждый элемент люминофорного слоя достаточно часто, чтобы не было заметно мерцания изображения. Если частота такого обхода экрана становится меньше 70 Гц, то инерционности зрительного восприятия будет недостаточно для того, чтобы изображение не мерцало. Мерцание изображения приводит к утомлению глаз,

головным болям и даже к ухудшению зрения. Чем больше экран монитора, тем более заметно мерцание, особенно периферийным (боковым) зрением, так как угол обзора изображения увеличивается.

### ***Оптические характеристики мониторов***

**Шаг точек** - это диагональное расстояние между двумя точками люминофора одного цвета (мм). Например, диагональное расстояние от точки люминофора красного цвета до соседней точки люминофора того же цвета. Чем меньше шаг точки или шаг полосы, тем лучше монитор: изображения выглядят более четкими и резкими, контуры и линии получаются ровными и изящными.

В стандартном дисплее **угол обзора** ограничивается 40 градусами по вертикали и 90 градусами по горизонтали. Контрастность и цвет варьируются при изменении угла, под которым пользователь смотрит на экран. Эта проблема стала приобретать все большую актуальность по мере увеличения размеров ЖК-дисплеев и количества отображаемых ими цветов.

**Мертвые точки.** Они характерны для ЖК-мониторов, что вызвано дефектами транзисторов. На экране такие неработающие пиксели выглядят как случайно разбросанные цветные точки. Поскольку транзистор не работает, то такая точка либо всегда черная, либо всегда светится. Обычно 3-5 неработающих точек считается нормой.

**Максимальное разрешение**, поддерживаемое монитором, является одним из ключевых параметров монитора, его указывает каждый производитель. Разрешение обозначает количество отображаемых элементов на экране (точек) по горизонтали и вертикали, например: 1024x768.

### ***Функциональные характеристики мониторов***

**Способ подключения монитора к компьютеру.** Монитору необходимо подведение видеосигналов, несущих информацию, отображаемую на экране. Цветному монитору требуется три сигнала, кодирующих цвет (RGB), и два сигнала синхронизации (вертикальной и горизонтальной развертки). Для подключения монитора к компьютеру используют сигнальные (аналоговые) кабели различных типов.

Помимо сигнального соединения возможно соединение монитора с компьютером через цифровой интерфейс, позволяющий управлять монитором из компьютера: калибровать его внутренние цепи, настраивать геометрические параметры изображения и т.п. в качестве цифрового интерфейса наиболее часто применяется разъем RC-232C.

### **7.3.2. LCD – мониторы**

Экраны LCD-мониторов, т.е. жидкокристаллических мониторов, сделаны из вещества (цианофенил), которое находится в жидком состоянии, но при этом обладает некоторыми свойствами, присущими кристаллическим телам. Фактически это жидкости, обладающие анизотропией свойств (в частности оптических), связанных с упорядоченностью в ориентации молекул.

Работа ЖКД основана на явлении поляризации светового потока. Известно, что так называемые кристаллы поляроиды способны пропускать только ту составляющую света, вектор электромагнитной индукции которой лежит в плоскости, параллельной оптической плоскости поляроида. Для оставшейся части светового потока поляроид будет непрозрачным. Таким образом поляроид как бы "просеивает" свет, данный эффект называется поляризацией света. Когда были изучены жидкие вещества, длинные молекулы которых чувствительны к электростатическому и электромагнитному полю и способны поляризовать свет, появилась возможность управлять поляризацией. Эти аморфные вещества за их схожесть с кристаллическими веществами по электрооптическим свойствам, а также за способность принимать форму сосуда, называли жидкими кристаллами.

## **7.4. Интерфейсы клавиатуры**

### **Последовательный интерфейс RS-232**

Стандарт RS-232 в общем случае описывает четыре интерфейсные функции:

- определение управляющих сигналов через интерфейс;
- определение формата данных пользователя, передаваемых через интерфейс;
- передачу тактовых сигналов для синхронизации потока данных;
- формирование электрических характеристик интерфейса.

Интерфейс RS-232 является последовательным асинхронным интерфейсом. Последовательная передача означает, что данные передаются по единственной линии. Для синхронизации битам данных предшествует специальный стартовый бит, после битов данных следует бит паритета и один или два стоповых бита. Такая группа битов совместно со стартовым и стоповым битом, а также битом паритета носит название старт-стопного символа.

Для передачи символов по интерфейсу RS-232 наибольшее распространение получил формат, включающий в себя один стартовый бит, один бит паритета и два стоповых бита.

Используемые в интерфейсе RS-232 уровни сигналов отличаются от уровней сигналов, действующих в модеме или компьютере. Логический "0" представляется положительным напряжением в диапазоне от +3 до +15В, а логическая "1" - отрицательным напряжением в диапазоне от - 3 до - 15В. .

Аппаратная реализация интерфейса RS-232 включает последовательный адаптер и собственно механический интерфейс (разъемное соединение).

Асинхронный последовательный порт подключается к внешним устройствам

через специальный разъем на 25 или 9 контактов.

Официальное ограничение по длине для соединительного кабеля по стандарту RS-232 составляет порядка 15 м при скорости передачи около 20 Кбит/с. Однако, на практике это расстояние может быть значительно больше и зависит от скорости передачи данных.

### **Интерфейс AT**

5-контактный DIN-разъем интерфейса AT, предназначен исключительно для подключения клавиатуры. Передача и прием происходят по одному каналу. Тактовый канал предназначен для синхронизации этих функций. Канал Reset (сброс) служит для передачи сигнала на перезагрузку PC, но обычно он не используется.

### **Интерфейс PS/2**

В настоящее время наиболее распространенным способом подключения клавиатуры и мыши является их подключение к маленькому 6-контактному порту PS/2. На материнских платах имеется два разъема, один из которых предназначен для мыши, другой - для клавиатуры.

### **Интерфейс USB**

Последовательная, полудуплексная, двунаправленная с производительностью 12 Мбит/с и шлейфовым подключением устройств. Была предложена компанией Intel и получила в настоящее время широкое распространение.

Предназначен для:

- подключения ПК к телефонной сети (для передачи речи и цифровой информации) в соответствии с требованиями некоторых синхронных терминалов (СТТ);

- подключения периферийных устройств;
- поддержки устройств новых типов.

Шина позволяет подключить к ПК до 127 физических устройств. Каждое физическое устройство может, в свою очередь, состоять из нескольких логических.

Сигнал USB переносится по кабелю с двумя парами проводников, который использует четырехконтактный экранированный разъем. Одна пара проводников не перекручивается и "несет" питание 5 В постоянного напряжения. Питаться устройства могут от шины или от собственного источника.

Пропускная способность 1.5 Мбит/с предназначена для мыши и клавиатуры, которые не в состоянии эффективно воспользоваться всеми 12 Мбит/с максимальной пропускной способности.

### **7.5. Интерфейсы манипуляторов типа мышь**

Все распространенные мыши по принципу действия делятся на "оптико-механические" (или просто "механические"), "оптические" и беспроводные.

#### **Оптико-механические мыши**

В них для преобразования движения мыши в информацию об изменении координат применяется покрытый резиной шар, который передает вращение двум пластмассовым валам, имеющим зубчатые диски на концах. Направление и угол поворота зубчатых дисков с большой точностью считываются инфракрасными оптопарами (т.е. парами "светоизлучатель-фотоприемник") и затем преобразуются встроенной микросхемой в электрический сигнал, идущий к компьютеру.

#### **Оптические мыши**

Для сканирования поверхности используется миниатюрная камера, которая

работает со скоростью 1500 снимков в секунду. Так как камера мало что увидит в темноте, мыши обычно оснащены небольшим красным светодиодом, который подсвечивает поверхность. Световые лучи отражаются от поверхности, попадают на CMOS-сенсор и превращаются в электрический сигнал.

Сигнал с сенсора передается на цифровой сигнальный процессор (DSP), который выполняет его анализ. Процессор мощностью около 18 MIPS (могут применяться процессоры как с большей, так и с меньшей производительностью) выделяет отдельные участки изображения и определяет их перемещение относительно предыдущего снимка. Таким образом, принцип работы заключается в анализе последовательности изображений. Процессор находит, как далеко передвинулась мышь и посылает информацию компьютеру. Компьютер передвигает курсор на экране в соответствии с информацией, полученной от мыши. Благодаря большой частоте опроса, движения курсора выглядят плавными.

Оптическая мышь имеет явные преимущества перед обычной:

- отсутствуют движущиеся части в плоскости соприкосновения с поверхностью, что уменьшает износ и понижает шанс поломки механики, ответственной за передвижение курсора;
- грязь не забивается во внутреннюю плоскость устройства и не мешает работе сенсоров;
- увеличенное разрешение мыши приводит к лучшей работе – особенно это критично в графических приложениях и программах, где требуется точное введение данных при помощи мыши;
- мышь не требует специальной поверхности, коврика;
- теоретически не нужно проводить гигиеническую протирку коврика, шарика и роликов.

Оптические мыши работают практически на всех поверхностях, кроме стеклянных, зеркальных, металлических и выполненных из бархата. Кроме того, производители обычно рекомендуют использовать мышь на ровной поверхности.

Недостатком оптических мышей является неспособность обрабатывать очень быстрые перемещения манипулятора.

### **Беспроводные мыши**

Беспроводные модели используют для передачи информации инфракрасный луч или радиосигнал. В случае использования инфракрасного сигнала необходимо наличие визуального контакта между манипулятором и приемником излучения – точно так же, как для пульта дистанционного управления. Приемники сигналов в беспроводных моделях подключаются опять же к портам PS/2, USB или COM. В самой мышке на том месте, откуда должен выходить провод, устанавливается ИК-передатчик, который и транслирует сигналы мыши через приемник на компьютер.

Есть у беспроводных мышей и еще одно свойство, которое нельзя назвать ни достоинством, ни недостатком – это увеличенная масса. Элементы питания и инфракрасный передатчик тоже что-то весят.

Еще один потенциальный недостаток беспроводных мышей – более низкая частота опроса по сравнению с их проводными собратьями (порядка 50-90 Гц), что может быть недостаточным для быстрых игр.

Мыши различаются по способу подключения к компьютеру. В последнее время получил большое распространение тип порта - USB (универсальная последовательная шина). В результате, сейчас можно встретить мышей с тремя видами разъемов: COM (девятоконтный трапециевидный, самый большой из трех), PS/2 (маленький круглый пятиконтактный) и USB (четыреконтактный маленький плоский прямоугольный).

**USB** мышь – современный стандарт подключения, имеет ряд преимуществ - эта мышь имеет стандартную частоту опроса 125 Hz, тем самым гарантируя

идеально плавное перемещение указателя или персонажа игры, а также - порт USB изначально создавался с расчетом на безопасное подключение устройств во время работы. Практически: вы прямо во время работы подключаете мышь к компьютеру, при этом появляется окошко, в котором показано найденное устройство, и если в системе уже существует драйвер для данной мыши, то он тут же будет запущен.

## **Тема 8. Пользовательский интерфейс**

Информационные системы в зависимости от конкретной области применения могут различаться по своим функциям, архитектуре, реализации, хотя два свойства являются общими для всех информационных систем:

- любая информационная система предназначена для сбора, хранения и обработки информации. Поэтому в основе любой информационной системы лежит среда хранения и доступа к данным с необходимым уровнем надежности хранения, быстродействия и эффективности доступа.

- информационные системы ориентируются на конечного пользователя, поэтому они должны обладать простым, удобным, легко осваиваемым интерфейсом, который должен предоставить все необходимые для его работы функции и не дать ему возможность выполнять какие-либо лишние действия.

### **8.1. Классификация интерфейсов**

- 1) **Командный интерфейс.** Командный интерфейс называется так по тому, что в этом виде интерфейса человек подает "команды" компьютеру, а компьютер их выполняет и выдает результат человеку. Командный интерфейс реализован в виде **пакетной технологии** и **технологии командной строки.**

2) **WIMP - интерфейс** (Window - окно, Image - образ, Menu - меню, Pointer - указатель) - диалог с пользователем ведется не с помощью команд, а с помощью графических образов - меню, окон, других элементов. Хотя и в этом интерфейсе подаются команды машине, но это делается "опосредственно", через графические образы. Этот вид интерфейса реализован на двух уровнях технологий: простой графический интерфейс и "чистый" WIMP - интерфейс.

3) **SILK - интерфейс** (Speech - речь, Image - образ, Language - язык, Knowledge - знание). Этот вид интерфейса наиболее приближен к обычной, человеческой форме общения. В рамках этого интерфейса идет обычный "разговор" человека и компьютера. При этом компьютер находит для себя команды, анализируя человеческую речь и находя в ней ключевые фразы. Результат выполнения команд он также преобразует в понятную человеку форму. Этот вид интерфейса наиболее требователен к аппаратным ресурсам компьютера, и поэтому его применяют в основном для военных целей.

### **Пакетная технология**

На вход компьютера подается последовательность символов, в которых по определенным правилам указывается последовательность запущенных на выполнение программ. После выполнения очередной программы запускается следующая и т.д. Машина по определенным правилам находит для себя команды и данные. В качестве этой последовательности может выступать, например, перфолента, стопка перфокарт, последовательность нажатия клавиш электрической пишущей машинки (типа CONSUL). Машина также выдает свои сообщения на перфоратор, алфавитно-цифровое печатающее устройство (АЦПУ), ленту пишущей машинки.

С появлением алфавитно-цифровых дисплеев началась эра по-настоящему пользовательской технологии - командной строки.

### **Технология командной строки.**

При этой технологии в качестве единственного способа ввода информации в компьютер служит клавиатура, а компьютер выводит информацию человеку с помощью алфавитно-цифрового дисплея (монитора). Эту комбинацию (монитор + клавиатура) стали называть **терминалом, или консолью**.

Команды набираются в **командной строке**. Командная строка представляет собой символ приглашения и мигающий прямоугольник - при нажатии клавиши на месте курсора появляются символы, а сам курсор смещается вправо. Это очень похоже на набор команды на пишущей машинке. Однако, в отличие от нее, буквы отображаются на дисплее, а не на бумаге, и неправильно набранный символ можно стереть. Команда заканчивается нажатием клавиши Enter (или Return.) После этого осуществляется переход в начало следующей строки. Именно с этой позиции компьютер выдает на монитор результаты своей работы. Затем процесс повторяется.

Технология командной строки уже работала на монохромных алфавитно-цифровых дисплеях. Поскольку вводить позволялось только буквы, цифры и знаки препинания, то технические характеристики дисплея были не существенны. В качестве монитора можно было использовать телевизионный приемник и даже трубку осциллографа.

Преобладающим видом файлов при работе с командным интерфейсом стали текстовые файлы - их и только их можно было создать при помощи клавиатуры.

### **Графический интерфейс**

Предпосылкой графического интерфейса явилось уменьшение времени реакции компьютера на команду, увеличение объема оперативной памяти, а также развитие технической базы компьютеров. Аппаратным основанием концепции, конечно же, явилось появление алфавитно-цифровых дисплеев на компьютерах, причем на этих дисплеях уже имелись такие эффекты, как "мерцание" символов, инверсия цвета (смена начертания белых символов на черном фоне обратным, то есть черных символов на белом фоне), подчеркивание символов. Эти эффекты распространились не на весь экран, а только на один или более символов.

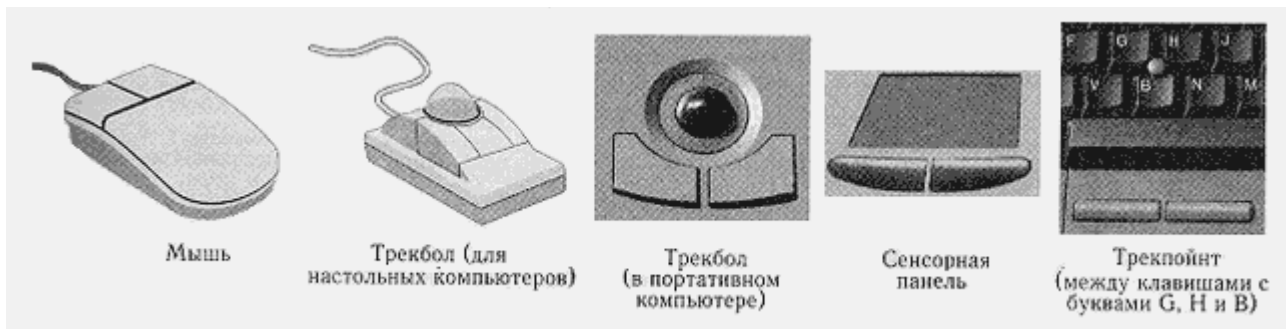
Следующим шагом явилось создание цветного дисплея.

### **Простой графический интерфейс**

На первом этапе графический интерфейс очень походил на технологию командной строки. Отличия от технологии командной строки заключались в следующем:

- при отображении символов допускалось выделение части символов цветом, инверсным изображением, подчеркиванием и мерцанием. Благодаря этому повысилась выразительность изображения;
- в зависимости от конкретной реализации графического интерфейса курсор может представляться не только мерцающим прямоугольником, но и некоторой областью, охватывающей несколько символов и даже часть экрана;
- нажатие клавиши Enter не всегда приводит к выполнению команды и переходу к следующей строке. Реакция на нажатие любой клавиши во многом зависит от того, в какой части экрана находился курсор;
- кроме клавиши Enter, на клавиатуре все чаще стали использоваться "серые" клавиши управления курсором;
- графический интерфейс стал использовать манипуляторы (типа мыши, трекбола

и т.п. - см. рисунок А.4.) Они позволяли быстро выделять нужную часть экрана и перемещать курсор.



Операционная система MS-DOS внедрила этот интерфейс в массы.

Типичным примером использования этого вида интерфейса является файловая оболочка Norton Commander и текстовый редактор Multi-Edit, Лексикон, ChiWriter и текстовый процессор Microsoft Word for Dos.

### **WIMP - интерфейс**

Графический интерфейс WIMP характеризуется следующими особенностями:

- вся работа с программами, файлами и документами происходит в окнах - определенных очерченных рамкой частях экрана;
- все программы, файлы, документы, устройства и другие объекты представляются в виде значков - иконок. При открытии иконки превращаются в окна;
- все действия с объектами осуществляются с помощью меню. Меню становится

основным элементом управления;

- широкое использование элементов управления - манипуляторов для указания на объекты и управление ими.

WIMP требует для своей реализации цветной растровый дисплей с высоким разрешением и манипулятор, а также повышенные требования к производительности компьютера, объему его памяти, пропускной способности шины и т.п. Однако этот вид интерфейса наиболее прост в усвоении и интуитивно понятен. Поэтому сейчас WIMP - интерфейс стал стандартом.

Ярким примером программ с графическим интерфейсом является операционная система Microsoft Windows.

### **Речевая технология**

С середины 90-х годов, после появления недорогих звуковых карт и широкого распространения технологий распознавания речи, появился так называемый "речевая технология" **SILK - интерфейс**. При этой технологии команды подаются голосом путем произнесения специальных зарезервированных слов - команд. Основными такими командами являются:

- "Проснись" - включение голосового интерфейса;
- "Отдыхай" - выключение речевого интерфейса;
- "Открыть" - переход в режим вызова той или иной программы. Имя программы называется в следующем слове;
- "Буду диктовать" - переход из режима команд в режим набора текста голосом;

- "Режим команд" - возврат в режим подачи команд голосом и некоторые другие.

Слова должны выговариваться четко, в одном темпе. Между словами обязательна пауза. Из-за неразвитости алгоритма распознавания речи такие системы требуют индивидуальной предварительной настройки на каждого конкретного пользователя.

"Речевая" технология является простейшей реализацией SILK - интерфейса.

### **Биометрическая технология ("Мимический интерфейс")**

Эта технология возникла в конце 90-х годов XX века. Для управления компьютером используется выражение лица человека, направление его взгляда, размер зрачка и другие признаки. Для идентификации пользователя используется рисунок радужной оболочки его глаз, отпечатки пальцев и другая уникальная информация. Изображения считываются с цифровой видеокамеры, а затем с помощью специальных программ распознавания образов из этого изображения выделяются команды. Эта технология, по-видимому, займет свое место в программных продуктах и приложениях, где важно точно идентифицировать пользователя компьютера.

### **Семантический (Общественный) интерфейс**

Этот вид интерфейса возник в конце 70-х годов XX века, с развитием искусственного интеллекта. Его трудно назвать самостоятельным видом интерфейса - он включает в себя и интерфейс командной строки, и графический, и речевой, и мимический интерфейс. Основная его отличительная черта - это отсутствие команд при общении с компьютером. Запрос формируется на естественном языке, в виде связанного текста и образов. По своей сути это трудно называть интерфейсом - это уже моделирование "общения" человека с компьютером.

В связи с важным военным значением этих разработок (например, для автономного ведения современного боя машинами - роботами, для "семантической" криптографии) эти направления были засекречены.

## **8.2. Типы пользовательских интерфейсов и этапы их разработки**

На ранних этапах развития вычислительной техники пользовательский интерфейс рассматривался как средство общения человека с операционной системой и был достаточно примитивным. В основном он позволял запустить выполнение задания.

По мере совершенствования аппаратных средств появилась возможность создания интерактивного программного обеспечения, использующего специальные пользовательские интерфейсы.

**Пользовательский интерфейс представляет собой совокупность программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером. Основу такого взаимодействия составляют диалоги.** Под диалогом в данном случае понимают регламентированный обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи. Каждый диалог состоит из отдельных процессов ввода-вывода, которые физически обеспечивают связь пользователя и компьютера.

Обмен информацией осуществляется передачей сообщений и управляющих сигналов. Различают входные сообщения, которые генерируются человеком с помощью средств ввода: клавиатуры, манипуляторов, например мыши и т. п., а также выходные сообщения, которые генерируются компьютером в виде текстов,

звуковых сигналов и/или изображений и выводятся пользователю на экран монитора или другие устройства вывода информации.



### Организация взаимодействия компьютера и пользователя.

В основном, пользователь генерирует сообщения следующих типов: запрос информации, запрос помощи, запрос операции или функции, ввод или изменение информации, выбор поля кадра и т. д. В ответ он получает: подсказки или справки, информационные сообщения, не требующие ответа, приказы, требующие действий, сообщения об ошибках, нуждающиеся в ответных действиях, изменение формата кадра и т. д.

Для ввода сообщений используются устройства: клавиатура - текстовый ввод; планшеты - графический ввод; сканеры - графический ввод; манипуляторы, световое перо, сенсорный экран - позиционирование и выбор информации на экране и т. п.

Для вывода сообщений используются устройства: мониторы - вывод оперативной текстовой и графической информации; принтеры - получение «твердой копии» текстовой и графической информации; графопостроители - получение твердой копии графической информации; синтезаторы речи - речевой вывод; звукогенераторы - вывод музыки и т. п.

### Типы интерфейсов

По аналогии с процедурным и объектным подходом к программированию различают процедурно-ориентированный и объектно-ориентированный подходы к разработке интерфейсов.



Типы интерфейсов.

**Процедурно-ориентированные интерфейсы** используют традиционную модель взаимодействия с пользователем, основанную на понятиях «процедура» и «операция». В рамках этой модели программное обеспечение предоставляет пользователю возможность выполнения некоторых действий, для которых пользователь определяет соответствующие данные и следствием выполнения которых является получение желаемых результатов.

**Объектно-ориентированные интерфейсы** используют иную модель взаимодействия с пользователем, ориентированную на манипулирование объектами предметной области, когда пользователю предоставляется возможность напрямую взаимодействовать с каждым объектом и инициировать выполнение операций, в процессе которых взаимодействуют несколько объектов. Задача пользователя формулируется как целенаправленное изменение некоторого объекта. Объект при этом понимается в широком смысле слова, например, модель реальной системы или процесса, база данных, текст и т. п. Пользователю

предоставляется возможность создавать объекты, изменять их параметры и связи с другими объектами, а также инициировать взаимодействие этих объектов.

Элементы таких интерфейсов включены в пользовательский интерфейс Windows, например, пользователь может «взять» файл и «переместить» его в другую папку. Таким образом, он инициирует выполнение операции перемещения файла.

| Процедурно-ориентированные<br>пользовательские интерфейсы                                                                                                                                                                                                 | Объектно-ориентированные<br>пользовательские интерфейсы                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Обеспечивают пользователей функциями, необходимыми для выполнения задач.<br><br>Акцент делается на задачи.<br><br>Пиктограммы представляют приложения, окна или операции.<br><br>Содержание папок и справочников отображается с помощью таблиц и списков. | Обеспечивают пользователям возможность взаимодействия с объектами.<br><br>Акцент делается на входные данные и результаты.<br><br>Пиктограммы представляют объекты.<br><br>Папки и справочники являются визуальными контейнерами объектов. |

Основные отличия пользовательских моделей интерфейсов процедурного и объектно-ориентированного типов.

Различают *процедурно-ориентированные* интерфейсы трех типов: «примитивные», меню и со свободной навигацией.

**Примитивным** называют интерфейс реализует конкретный сценарий работы программного обеспечения, например: ввод данных - решение задачи - вывод результата без отклонения от последовательного процесса.

**Интерфейс-меню** в отличие от примитивного интерфейса позволяет пользователю выбирать необходимые операции из специального списка, выводимого ему программой. Эти интерфейсы предполагают реализацию множества сценариев работы, последовательность действий в которых определяется пользователем.

Различают одноуровневые и иерархические меню. Первые используют для сравнительно простого управления вычислительным процессом, когда вариантов немного (не более 5-7), и они включают операции одного типа, например, Создать, Открыть, Закрыть и т. п. Вторые - при большом количестве вариантов или их очевидных различиях, например, операции с файлами и операции с данными, хранящимися в этих файлах.

**Интерфейсы со свободной навигацией** также называют **графическими пользовательскими интерфейсами** (GUI - Graphic User Interface) или интерфейсами WYSIWYG (What You See Is What You Get - что видишь, то и получишь, т. е., что пользователь видит на экране, то он и получит при печати).

Графические интерфейсы поддерживают концепцию интерактивного взаимодействия с программным обеспечением, осуществляя визуальную обратную связь с пользователем и возможность прямого манипулирования объектами и информацией на экране.

Существенной особенностью интерфейсов данного типа является способность изменяться в процессе взаимодействия с пользователем, предлагая выбор только тех операций, которые имеют смысл в конкретной ситуации.

### 8.3. Психофизические особенности человека, связанные с восприятием, запоминанием и обработкой информации

При проектировании пользовательских интерфейсов необходимо учитывать психофизические особенности человека, связанные с восприятием, запоминанием и обработкой информации.

Исследованием принципов работы мозга человека занимается **КОГНИТИВНАЯ ПСИХОЛОГИЯ**.



Упрощенная информационно-процессуальная модель мозга.

Информация о внешнем мире поступает в наш мозг в огромных количествах. Часть мозга, которую условно можно назвать «**процессором восприятия**», постоянно без участия сознания перерабатывает ее, сравнивая с прошлым опытом, и помещает в **хранилище** уже в виде зрительных, звуковых и прочих образов. Если же наше внимание не было привлечено к информации, то она пропадает, замещаясь следующими порциями.

Любые внезапные или просто значимые для нас изменения в окружении привлекают наше внимание, и тогда интересующая нас информация поступает в **кратковременную память**.

В каждый момент времени фокус внимания может фиксироваться в одной

точке. Поэтому, если возникает необходимость «одновременно» отслеживать несколько ситуаций, то обычно фокус перемещается с одного отслеживаемого элемента на другой. При этом внимание «рассредоточивается», и какие-то детали могут быть упущены. Например, при «прокрутке» текста или рисунка с использованием линейки прокрутки окна Windows приходится одновременно смотреть на текст, чтобы определить, где остановиться, и на ползунок. Поскольку текст важнее, фокус внимания перестает перемещаться на мышь, и она «соскакивает» с ползунка линейки. Следует иметь в виду, что обработка процессором восприятия требует некоторого времени и, если сигнал выдается в течение времени, меньшем времени обработки, то наш мозг его не воспринимает.

Существенно и то, что восприятие во многом основано на **мотивации**. Например, если человек голоден, то он в первую очередь будет замечать все съедобное, а если устал - то, войдя в комнату, он в первую очередь увидит диван или кровать.

Необходимо также учитывать, что в процессе переработки информации мозг сравнивает поступающие данные с предыдущими. При смене кадра мозг на некоторое время блокируется: он «осваивает» новую картинку, выделяя наиболее существенные детали. А значит, если необходима быстрая реакция пользователя, то резко менять картинку не стоит.

Краткосрочная память - самое «узкое» место «системы обработки информации» человека. Ее емкость приблизительно равна  $7 \pm 2$  несвязанных объектов. Краткосрочная память является своего рода оперативной памятью мозга, именно с ней работает процессор познания, но не востребованная информация хранится в ней не более 30 с. Чтобы не забыть какую-нибудь важную для нас информацию, мы обычно повторяем ее «про себя», «обновляя» информацию в краткосрочной памяти. Таким образом, при проектировании интерфейсов следует иметь в виду, что подавляющему большинству людей сложно, например, запомнить и ввести на другом экране число, содержащее более

5 цифр (7 - 2), или некоторое сочетание букв.

Люди вносят в каждую деятельность свое понимание того, как она должна выполняться. Это понимание - модель деятельности - базируется на прошлом опыте человека. Множество таких моделей хранится в долговременной памяти человека.

В долговременную память записываются постоянно повторяемые сведения или информация, связанная с сильными эмоциями. Долговременная память человека - хранилище информации с неограниченной емкостью и временем хранения. Однако доступ к этой информации весьма непрост: по всей вероятности, механизмы извлечения информации из памяти имеют ассоциативный характер. Специальная методика запоминания информации (мнемоника) использует именно это свойство памяти: для запоминания информации ее «привязывают» к тем данным, которые память уже хранит и позволяет легко получить.

Поскольку доступ к долговременной памяти затруднен, целесообразно рассчитывать не на то, что пользователь вспомнит нужную информацию, а на то, что он ее узнает. Именно поэтому интерфейс типа меню так широко используется.

**Особенности восприятия цвета.** Цвет в сознании человека ассоциируется с эмоциональным фоном. Известно, что теплые цвета: красный, оранжевый, желтый человека возбуждают, а холодные: синий, фиолетовый, серый -успокаивают.

Следует иметь в виду, что обилие оттенков привлекает внимание, но быстро утомляет, поэтому не стоит ярко раскрашивать окна, с которыми пользователь будет долго работать, а необходимо предоставить пользователю возможность настройки цветов.

**Особенности восприятия звука.** В интерфейсах звук обычно используют для привлечения внимания, как фон, обеспечивающий некоторое состояние

пользователя, как источник дополнительной информации и т. п. Применяя звук, следует учитывать, что большинство людей очень чувствительны к звуковым сигналам, особенно, если последние указывают на наличие ошибки. Поэтому при создании звукового сопровождения целесообразно предусматривать возможность его отключения.

**Субъективное восприятие времени.** Человеку свойственно субъективное восприятие времени. Считают, что внутреннее время связано со скоростью и количеством воспринимаемой и обрабатываемой информации. Занятый человек обычно времени не замечает. Зато в состоянии ожидания когда мозг оказывается в состоянии информационного вакуума, время тянется бесконечно.

Доказано, что при ожидании более 1-2 сек. пользователь может отвлечься, «потерять мысль», что неблагоприятно сказывается на результатах работы и увеличивает усталость, так как каждый раз после ожидания много сил тратится на включение в работу.

Сократить время ожидания можно, заняв пользователя, но не отвлекая его от работы, например, предоставив ему какую-либо информацию для обдумывания. По возможности, целесообразно выводить пользователю промежуточные результаты: во-первых, он будет занят их обдумыванием, во-вторых, по ним он сможет оценить будущие результаты и отменит операцию, если они его не удовлетворяют.

Известны попытки использования для «развлечения» пользователя анимации, например, в Windows при копировании файлов демонстрируется «ролик» с летающими листочками. Однако следует иметь в виду, что, когда какую-либо анимацию смотришь первый раз, то это интересно, а когда в течение получаса наблюдаешь, как «летают» листочки при получении информации из Интернета, то это начинает раздражать.

Чтобы уменьшить раздражение, возникающее при ожидании, необходимо

соблюдать основное правило: информировать пользователя, что заказанные им операции потребуют некоторого времени выполнения. Обычно для этого используют индикаторы оставшегося времени, анимированные объекты, как в Интернете, и изменение формы курсора мыши на песочные часы. Очень важно точно обозначить момент, когда система готова продолжать работу, например значительным изменением внешнего вида экрана.

В конечном итоге взаимодействие пользователя с интерфейсом будет определяться не только физическими возможностями и особенностями человека по восприятию, обработке и запоминанию информации, представленной в различных формах, а также по выполнению им разнообразных действий, но и пользовательской моделью интерфейса.

#### **8.4. Пользовательская и программная модели интерфейса**

Существуют три совершенно различные модели пользовательского интерфейса: **модель программиста, модель пользователя и программная модель.**

Программист, разрабатывая пользовательский интерфейс, исходит из того, управление какими операциями ему необходимо реализовать в пользовательском интерфейсе, и как это осуществить, не затрачивая существенных ресурсов компьютера, и своих сил и времени. Его интересуют функции не связанные с удобством пользователя характеристики программного обеспечения. Именно поэтому большинство интерфейсов существующих программ вызывают серьезные нарекания пользователей.

**С точки зрения здравого смысла хорошим следует считать интерфейс, при работе с которым пользователь получает именно то, что он ожидал.**

Представление пользователя о функциях интерфейса можно описать в виде **пользовательской модели интерфейса.** Пользовательская модель интерфейса - это совокупность обобщенных представлений конкретного пользователя или

группы пользователей о процессах, происходящих во время работы программы или программной системы.

Эта модель базируется на особенностях опыта конкретных пользователей, который характеризуется:

- уровнем подготовки в предметной области разрабатываемого программного обеспечения;
- уровнем подготовки в области владения компьютером;
- устоявшимися стереотипами работы с компьютером.

Для построения пользовательской модели необходимо изучить перечисленные выше особенности опыта предполагаемых пользователей программного обеспечения. С этой целью используют опросы, тесты и даже фиксируют на пленку последовательность действий, осуществляемых в процессе выполнения некоторых операций.

Приведение в соответствие моделей пользователя и программиста, а также построение на их базе программной модели интерфейса задача не тривиальная. Причем, чем сложнее предметная область, тем сложнее оказывается построить программную модель интерфейса, учитывающую особенности пользовательской модели и не требующую слишком больших затрат как в процессе разработки, так и во время работы.

Часто грамотно построенный интерфейс, который адекватно отображает сущность происходящих процессов, способствует росту квалификации пользователей.

Интуитивные модели выполнения операций в предметной области должны стать основой для разработки интерфейса. Именно нежелание или невозможность следования интуитивным моделям выполнения операций приводит к созданию искусственных надуманных интерфейсов, которые негативно воспринимаются

пользователями.

Ломка стереотипов - процедура болезненная. На это стоит решаться, если некоторое революционное изменение значительно расширяет возможности пользователя или облегчает его работу, например, переход к Windows-интерфейсам существенно упростил работу с компьютером большому числу пользователей – непрофессионалов.

Критерии оценки интерфейса пользователем:

- простота освоения и запоминания операций системы;
- скорость достижения результатов при использовании системы - определяется количеством вводимых или выбираемых мышью команд и настроек;
- субъективная удовлетворенность при эксплуатации системы (удобство работы, утомляемость и т. д.).

Для пользователей-профессионалов, постоянно работающих с одним и тем же пакетом, важными являются второй и третий критерии, а для пользователей-непрофессионалов - первый и третий.

### **8.5. Классификации диалогов и общие принципы их разработки.**

Диалог - это процесс обмена информацией между пользователем и программной системой, осуществляемый через интерактивный терминал и по определенным правилам.

**Типы диалога.** Тип диалога определяет, кто из «собеседников» управляет процессом обмена информацией. Соответственно различают два типа диалога: **управляемые программой и управляемые пользователем.**

**Диалог, управляемый программой,** предусматривает наличие жесткого,

линейного или древовидного, т. е. включающего возможные альтернативные варианты, сценария диалога, заложенного в программное обеспечение. Такой диалог обычно сопровождают большим количеством подсказок, которые уточняют, какую информацию необходимо вводить на каждом шаге.

**Диалог, управляемый пользователем**, подразумевает, что сценарий диалога зависит от пользователя, при этом система обеспечивает возможность реализации различных пользовательских сценариев.

**Формы диалога.** Никакой диалог невозможен, если не существует языка, понятного «собеседникам». Описание языка, на котором ведется диалог, включает определение:

- **синтаксиса** - правил, определяющих допустимые конструкции (слова, предложения) языка или его форму;
- **семантики** - правил, определяющих смысл синтаксически корректных конструкций языка или его содержание.

В зависимости от вида используемых в конкретном случае синтаксиса и семантики различают три формы диалога: **фразовую, директивную, табличную**.

**Фразовая форма** предполагает «общение» с пользователем на естественном языке. Общение может осуществляться в свободном формате, но возможна и фиксация отдельных фраз.

Организация диалога на естественном языке на современном уровне - задача не решенная, так как естественный язык крайне сложен и пока не удастся в достаточной степени формализовать его синтаксис и семантику.

Чаще всего используют диалоги, предполагающие односложные ответы, например:

Программа: Введите свой возраст (полных лет):

В этом случае программа содержит ограниченное описание как синтаксиса, так и семантики используемого ограниченно-естественного языка. Для данного примера достаточно определить синтаксис понятия «целое положительное число» и наложить ограничение на значение числа.

Однако существует некоторый опыт создания интерфейсов с использованием интеллектуальных систем. При обработке фраз в этих случаях оперируют понятием словоформа. Словоформа - отрезок текста между двумя соседними пробелами или знаками препинания. Обработка словоформ вне связи с контекстом называется морфологическим анализом.

После распознавания словоформ осуществляют синтаксический анализ сообщения. Далее выполняют семантический анализ, т. е. определяют смысловые отношения между словоформами.

Таким образом, интерфейс, реализующий фразовую форму диалога, должен: преобразовывать сообщения из естественно-языковой формы в форму внутреннего представления и обратно, выполнять анализ и синтез сообщений пользователя и системы, отслеживать и запоминать пройденную часть диалога.

Основными недостатками фразовой формы при использовании подмножества естественного языка являются:

- большие затраты ресурсов;
- отсутствие гарантии однозначной интерпретации формулировок;
- необходимость ввода длинных грамматически правильных фраз.

Основное достоинство фразовой формы состоит в относительно свободном общении с системой.

**Директивная форма** предполагает использование команд (директив) специально разработанного формального языка.

Команду можно вводить:

- в виде строки текста, специально разработанного формата, например, команды MS DOS, которые вводятся в командной строке;
- нажатием некоторой комбинации клавиш клавиатуры;
- посредством манипулирования мышью, например, «перетаскиванием» пиктограмм;
- комбинацией второго и третьего способов.

Основными достоинствами директивной формы являются:

- сравнительно небольшой объем вводимой информации;
- гибкость - возможности выбора операции в данном случае ограничены только набором допустимых команд;
- ориентация на диалог, управляемый пользователем;
- использование минимальной области экрана или неиспользование ее вообще;
- возможность совмещения с другими формами.

Недостатки директивной формы:

- практическое отсутствие подсказок на экране, что требует запоминания вводимых команд и их синтаксиса;
  - почти полное отсутствие обратной связи о состоянии инициированных процессов;
- необходимость навыков ввода текстовой информации или манипуляций мышью;
- отсутствие возможности настройки пользователем.

Исследования показали, что директивная форма удобна для пользователя-профессионала, быстро запоминающего синтаксис часто используемых команды или комбинации клавиш.

**Табличная форма** предполагает, что пользователь выбирает ответ из предложенных программой. Язык диалога для табличной формы имеет простейший синтаксис и однозначную семантику, что достаточно легко реализовать. Удобна эта форма и для пользователя, однако применение табличной формы возможно, если множество возможных ответов на конкретный вопрос конечно.

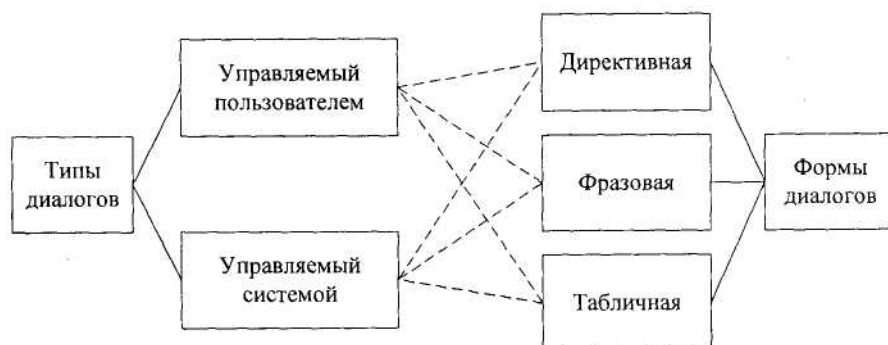
Достоинствами табличной формы являются:

- наличие подсказки, что уменьшает нагрузку на память пользователя;
- сокращение количества ошибок ввода: пользователь не вводит информацию, а указывает на нее;
- сокращение времени обучения пользователя;
- возможность совмещения с другими формами;
- в некоторых случаях возможность настройки пользователем.

К недостаткам данной формы относят:

- необходимость наличия навыков навигации по экрану;
- использование сравнительно большой площади экрана для изображения визуальных компонентов;
- интенсивное использование ресурсов компьютера, связанное с необходимостью постоянного обновления информации на экране.

Следует иметь в виду, что типы и формы диалога выбирают независимо друг от друга: любая форма применима для обоих типов диалогов.



Соответствие типов диалогов и его форм.

Помимо диалогов, происходящих в процессе нормальной работы программного обеспечения и называемых **синхронными**, возникают диалоги, по инициативе системы или пользователя при нарушении сценария нормального процесса. Такие диалоги называют **асинхронными**. Обычно их используют для выдачи экстренных сообщений от системы или пользователя.

**Разработка диалогов.** Процесс проектирования и реализации диалогов можно разделить на следующие стадии:

- определение множества необходимых диалогов, их основных сообщений и возможных сценариев - проектирование абстрактных диалогов;
- определение типа и формы каждого диалога, а также синтаксиса и семантики используемых языков - проектирование конкретных диалогов;
- выбор основных и дополнительных устройств и проектирование процессов ввода-вывода.

В основу диалогов должна закладываться идеология технологического процесса, для автоматизации которого предназначается программный продукт.

## 8.6. Оценка пользовательских интерфейсов

Четыре основных критерия качества любого интерфейса: скорость работы пользователей, количество человеческих ошибок, скорость обучения и субъективное удовлетворение пользователей.

### 8.6.1. Скорость выполнения работы

Длительность выполнения работы пользователем состоит из длительности восприятия исходной информации, длительности интеллектуальной работы (в смысле – пользователь думает, что он должен сделать), длительности физических действий пользователя и длительности реакции системы. Как правило, длительность реакции системы является наименее значимым фактором.

Для определения скорости выполнения задачи нужно только разложить эту задачу на составляющие и, зная продолжительность каждой составляющей, всё сложить и узнать длительность всего процесса. Обычно тот интерфейс лучше, при котором время выполнения задачи меньше.

Критерий скорости работы называется GOMS «Предсказание скорости» (Goals, Operators, Methods, and Selection Rules – цели, операторы, методы и правила их выбора).

Правила GOMS:

- нажатие на клавишу клавиатуры, включая Alt, Ctrl и Shift (K): 0,28сек; -
- нажатие на кнопку мыши (M): 0,1 сек; -
- перемещение курсора мыши (П): 1,1 сек; -
- взятие или бросание мыши (В): 0,4 сек; -
- продолжительность выбора действия (Д): 1,2 сек; -
- время реакции системы (Р): от 0,1 сек до бесконечности. Для базовых операций,

таких как работа с меню, это время можно не засчитывать, поскольку с момента создания метода производительность компьютеров многократно возросла.

### **8.6.2. Длительность интеллектуальной работы**

Взаимодействие пользователя с системой (не только компьютерной) состоит из шести шагов:

1. Формирование цели действий.
2. Определение общей направленности действий.
3. Определение конкретных действий.
4. Выполнение действий.
5. Восприятие нового состояния системы.
6. Интерпретация состояния системы.
7. Оценка результата.

Из этого списка становится видно, что процесс размышления занимает почти все время, в течение которого пользователь работает с компьютером, во всяком случае, шесть из семи этапов полностью заняты умственной деятельностью. Соответственно, повышение скорости этих размышлений приводит к существенному улучшению скорости работы.

К сожалению, существенно повысить скорость собственно мышления пользователей невозможно. Тем не менее, уменьшить влияние факторов, усложняющих (и, соответственно, замедляющих) процесс мышления, вполне возможно.

### **8.6.3. Непосредственное манипулирование**

Перед действием пользователь из общего, еще неконкретного замысла формирует четкую последовательность действий.

Предположим, пользователь хочет выпить чаю. Желание выпить чаю есть цель действий. После осознания этого пользователь строит алгоритм своих действий: Подойти к чайнику и открыть крышку. Если воды в чайнике мало или нет вовсе, перенести чайник к раковине и наполнить его водой, после чего поставить его на плиту. Если воды в чайнике достаточно, сразу поставить его на плиту. Закрывать чайник крышкой. Найти спички. Открыть коробок, вытащить одну спичку, закрыть коробок, зажечь спичку. Спичкой зажечь под чайником газ, установив подачу газа на максимум. Потушить спичку и выкинуть её. Подождать, пока чайник не закипит, в это время найти достаточно чистый стакан и налить в него заварки. По желанию, найти сахарницу и добавить сахар в стакан. Выключить газ. Налить кипятка из чайника в стакан. Употребить жидкость по назначению.

В реальной жизни такую сложную программу пользователь не создает – действие стало автоматическим и создаваемый алгоритм состоит в лучшем случае из элементов высшего порядка (поставить чайник, налить чаю).

Анализируя пример с чаем, можно выделить определенные требования к человеку, выполняющему работу. Он должен знать:

- что он хочет получить на выходе (чай);
- как минимум одну последовательность действий, приводящую к успешному результату (наполнить чайник, поставить его на плиту, дождаться закипания, налить кипятка в стакан с заваркой);
- где ему найти все объекты, участвующие в процедуре;
- как определять годность объектов к использованию (есть ли вода в чайнике);
- как управляться с объектами (как включить газ).

Такая последовательность действий называется непосредственным манипулированием. Смысл этого метода очень прост. **Пользователь не отдает команды системе, а манипулирует объектами.** Когда вы хотите зажечь газ в плите, вы ведь не командуете плите «Зажги газ!», а манипулируете спичками и

плитой.

Чтобы лучше оценить прелесть этого метода, сравним три варианта действий пользователя на примере удаления файлов:

| Выбор команд из меню                                     | Использование горячих клавиш | Использование элемента на панели инструментов | Непосредственное манипулирование |
|----------------------------------------------------------|------------------------------|-----------------------------------------------|----------------------------------|
| Формирование цели действий и общего замысла              |                              |                                               |                                  |
| Определение необходимых действий и их последовательности |                              |                                               |                                  |

|                                                |                                    |                                             |                         |
|------------------------------------------------|------------------------------------|---------------------------------------------|-------------------------|
| Выбор файла                                    |                                    |                                             |                         |
| Поиск меню, ответственного за стирание         | Поиск в памяти команды стирания    | Поиск на экране соответствующей пиктограммы | Поиск корзины           |
| Поиск элемента меню, вызывающее стирание файла | Поиск клавиши Delete на клавиатуре | Нажатие на пиктограмму                      | Перенос файла в корзину |
| Выбор нужного элемента меню                    | Нажатие клавиши Delete             |                                             |                         |

Для ускорения мыслительной работы пользователя необходимо не только сокращать количество действий, но и делать их более простыми. Из таблицы сразу видно, что метод выбора команды из меню плох уже тем, что состоит из большого количества действий.

Количество элементов второго метода, использующего горячую клавишу, также велико, но у него есть определенные плюсы. При достаточной степени автоматизма нет ни необходимости искать клавишу на клавиатуре, ни думать, какую клавишу нажать. Таким образом, для опытных пользователей этот метод очень хорош.

Третий способ, нажатие на кнопку в панели инструментов, состоит из не большого количества элементов, так что формально он хорош, тем более, что подсказка к действию постоянно находится на экране, так что пользователю не приходится копаться в своей памяти (что может быть очень долгим).

Четвертый способ – непосредственное манипулирование и процесс удаления файла состоит из одного *единого действия*, т.е. пользователь выбирает файл, высматривает корзину и перетаскивает туда файл одним движением. Непосредственное манипулирование позволяет серьезно снизить как количество команд в системе и длительность обучения.

Не обязательно стараться реалистично отразить действие, важнее более реалистично отразить объект, над которым это действие совершается. Например, компьютерную панель управления работой осветительных приборов необязательно снабжать точными имитациями выключателей. Главное реалистично отразить на ней план помещения и расположение источников света, равно как и показать прямую связь между этой информацией и собственно выключателями.

#### 8.6.4. Потеря фокуса внимания

Пользователи работают с системой отнюдь не всё время, в течение которого они работают с системой. Дело в том, что пользователи постоянно отвлекаются.

Телефонный звонок, обеденный перерыв и т.п. - в течение работы происходит множество таких отвлечений. Каждый раз, когда пользователь прерывает свою деятельность и начинает думать о том, что ему делать дальше, он отвлекается тоже.

Каждое такое отвлечение сбивает **фокус внимания**, т.е. обработку текущего действия. После каждого такого отвлечения пользователь должен либо вспоминать текущую задачу, либо заново её ставить перед собой. Дело в том, что у человека

есть только один фокус внимания, так что при любом отвлечении (которое есть не что иное, как переключение на другую задачу) старый фокус внимания теряется.

Но при отвлечении новые стимулы заменяют содержимое кратковременной памяти, так что для возвращения к работе от пользователя требуется заново поместить в свою память нужную информацию.

Таким образом, необходимо максимально облегчать возвращение пользователей к работе.

Итак, для продолжения работы пользователь должен знать:

- на каком шаге он остановился;
- какие команды и параметры он уже дал системе;
- что именно он должен сделать на текущем шаге;
- куда было обращено его внимание на момент отвлечения.

Предоставлять пользователю всю эту информацию лучше визуально. Чтобы показать пользователю, на каком шаге он остановился, традиционно используют конструкцию «Страница N из N». К сожалению, эта конструкция не визуальна. Когда читатель держит в руках книгу, он может понять, в какой её части он находится, по толщине левой и правой части разворота.

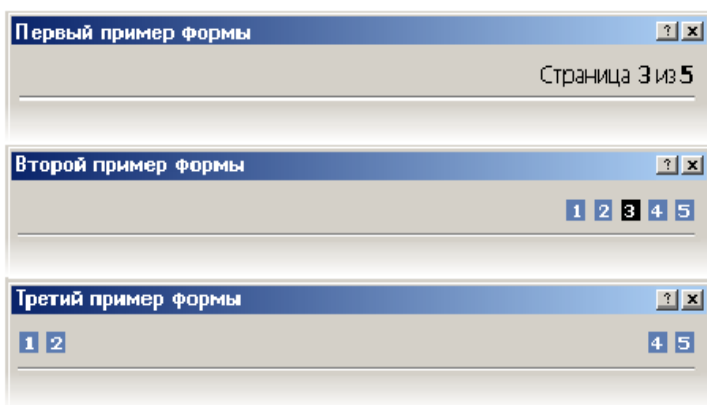


Рис. 1. Три варианта индикации степени заполнения экранной формы. Второй и третий варианты значительно визуальнее. Не используйте в подобных случаях ползунки (см. стр. 76).

Показ пользователю ранее отданных им команд проблематичен т.к. размеры экрана ограничены, но можно максимально облегчить переход к предыдущим экранам.

Можно показывать пользователю, что он должен сделать на текущем шаге процедуры.

И, наконец, четвертый пункт: показ пользователю, куда было обращено его внимание на момент отвлечения. Нужно делать фокус ввода максимально более заметным, например цветовым кодированием активного элемента.

#### **8.6.5. Длительность физических действий**

Длительность физических действий пользователя, прежде всего, зависит от степени автоматизации работы и степени необходимой точности работы. Понятно, что чем больше работы делает компьютер, тем лучше.

**Быстрый или точный.** Любое физическое действие, совершаемое с помощью мускулатуры, может быть или точным или быстрым. Вместе точность и быстрота встречаются исключительно редко, поскольку для этого нужно выработать существенную степень автоматизма. Объясняется это сугубо физиологическими факторами: при резком движении невозможно быстро остановиться, соответственно, чем точнее должно быть движение, тем более плавным и замедленным оно должно быть.

Пользователь, как правило, управляет компьютером двумя способами, а именно мышью и клавиатурой.

Клавиатура не требует особой точности движений – неважно, быстро нажали клавишу или медленно, равно как сильно или слабо и обеспечивает большую скорость работы.

Мышь не является прецизионным инструментом и не предназначена для очень точных манипуляций, а также инерционна – есть разница между медленным её перемещением и быстрым, сильным приложенным усилием и слабым.

Перемещение руки с клавиатуры на мышь и потом обратно занимает почти секунду.

#### **8.6.6. Длительность реакции системы**

Часто пользователи надолго прерывают свою работу. Помимо потери фокуса внимания, о котором уже сказано, это плохо тем, что лишенная руководства система начинает простаивать. Пользователь нередко отвлекается потому, что система делает что-либо длительное. Проиллюстрировать это очень удобно на процессе печати когда оператор отправив документ в печать, начинают бездельничать.

Проблема в том, что сразу после того, как человек отвлекается, системе

зачастую, во что бы то ни стало, начинает требоваться что-либо от человека. Таким образом, человек и система бездельничают.

Это делает всегда верным следующее правило: если процесс предположительно будет длительным, система должна убедиться, что она получила всю информацию от пользователя до начала этого процесса.

Возможно, когда пользователь запускает какой-либо процесс, система показывает ему индикатор степени выполнения. Пользователь экстраполирует эти данные и резонно решает, что у него есть три минуты, чтобы размяться.

Индикаторы степени выполнения обычно рассматриваются программистами не как показатели процента выполнения задачи, но как индикаторы того, что система вообще работает.

## 8.7. Человеческие ошибки

Важным критерием эффективности интерфейса является количество человеческих ошибок. В некоторых случаях одна или две человеческих ошибки погоды не делают, но только тогда, когда эти ошибки легко исправляются. Однако часто минимальная ошибка приводит к совершенно катастрофическим последствиям, например, ошибка операциониста в банке.

Компьютеры (как и все сложные технические системы) вообще не могут быть используемы человеком без совершения ошибок. Компьютеры требуют от человека **точности, логического мышления, способности абстрагироваться** от реального мира. Человек же практически на это не способен. Человек не цифровая система, неспособная на ошибку, но система аналоговая. Именно благодаря этому он плох в логике, зато имеет интуицию, не приспособлен к точности, зато может подстраиваться к ситуации, слабо абстрагируется, зато хорошо разбирается в реальном мире.

Суммируя, можно сказать, совершение ошибок есть естественное занятие

человека. А раз ошибки естественны, значит система, неспособная сама их обнаружить и исправить, порочна.

Под словосочетанием «человеческая ошибка» будем иметь в виду только действие пользователя, не совпадающее с целью действий этого пользователя.

Типы ошибок:

- ошибки, вызванные недостаточным знанием предметной области - легко исправляются обучением пользователей;
- опечатки. Опечатки происходят, когда не все внимание уделяется выполнению текущего действия и когда в мысленный план выполняемого действия вклинивается фрагмент плана из другого действия;
- несчитывание показаний системы, например потому, что у них уже сложилось мнение о текущем состоянии или потому, что они либо забывают считывать показания, либо не знают, что это нужно делать (и как это делать);
- моторные ошибки. Сущностью этих ошибок являются ситуации, когда пользователь знает, что он должен сделать, знает, как этого добиться, но не может выполнить действие нормально из-за того, что физические действия, которые нужно выполнить, выполнить трудно.

Усилия, при борьбе с ошибками нужно направлять на:

- обучение пользователей в процессе работы;
- повышение требований к бдительности;
- повышение разборчивости и заметности индикаторов;
- снижение чувствительности системы к ошибкам.

Для этого есть три основных способа:

- блокировка потенциально опасных действий пользователя до получения подтверждения правильности действия;

- проверка системой всех действий пользователя перед их принятием;
- самостоятельный выбор системой необходимых команд или параметров, при которых, от пользователя требуется только проверка.

Разберем эти три способа подробнее.

#### **8.7.1. Блокировка потенциально опасных действий до получения подтверждения**

Команда удаления файла в любой операционной системе снабжена требованием подтвердить удаление. Эта блокировка приносит пользу только начинающим пользователям, которые проверяют каждый свой шаг.

Для опытных пользователей это диалоговое окно с требованием подтверждения не работает. Во-первых, оно не защищает нужные файлы. Во-вторых, оно без пользы отвлекает пользователя и тратит его время.

Надо требовать подтверждения не после команды пользователя, а до неё. Предположим, чтобы удалить файл, нужно сначала в контекстном меню выбрать команду «Разблокировать», после чего выбрать этот же файл и запустить процесс его удаления (неважно, с клавиатуры или из меню). В этом случае от пользователя действительно требуется подтвердить удаление, поскольку эти два действия напрямую не связаны друг с другом – если в одном из них была допущена ошибка, файл удалить не удастся.

Не делайте опасные для пользователя кнопки кнопками «по умолчанию».

#### **8.7.2. Проверка действий пользователя перед их принятием**

Два способа проверки.

**Во-первых**, это меню. В случаях, когда пользователь выбирает команду из

списка, система может без труда делать так, чтобы в этот список попадали только корректные команды.

**Во-вторых**, если действие запускается непосредственным манипулированием объектами, можно индцировать возможные действия изменением поведения этих объектов. Например, если бы форматирование диска запускалось не нажатием кнопки, а перенесением пиктограммы диска в область форматирования, можно было бы показывать пользователю, как с выбранного диска исчезают все файлы и папки.

Проверкой всех действий пользователя перед их принятием можно также успешно защищать вводимые пользователем данные, в особенности данные численные.

В большинстве ОС есть специальный элемент управления, именуемый крутилкой (spinner). Этот элемент имеет то существенное достоинство, что при использовании мыши значение в этом элементе всегда находится в нужном диапазоне и обладает нужным форматом.

Всегда показывайте границы диапазона во всплывающей подсказке.

Если пользователь ввёл некорректное число с клавиатуры, нужно индцировать возможную ошибку изменением начертания шрифта в обычных программах, заменой цвета фона.

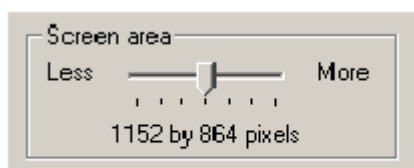


Рис. 3. Ползунок.

В тех же случаях, когда количество возможных значений невелико, лучше

использовать другой элемент управления – ползунок. Он позволяет устанавливать только определенные значения.

### 8.7.3. Самостоятельный выбор команд

Системе, лучше знать, какие именно команды или параметры для неё пригодны. Соответственно, чем меньше действий требуется совершить пользователю, тем меньше вероятность ошибки. Вопрос состоит в том, как системе узнать, что именно нужно пользователю. Система сама может узнать большинство из тех сведений, которые она запрашивает у пользователя. В этом случае источниками этих сведений являются: здравый смысл разработчика системы, предыдущие установленные параметры, наиболее часто устанавливаемые параметры.

### 8.7.4. Уровни ошибок

Ошибки, расставленные по уровням их негативного эффекта:

|                                                                                                                                                                                   |                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Ошибки, исправляемые во время совершения действия, например пользователь перетаскивает файл в корзину и во время перетаскивания замечает, что он пытается стереть не тот файл. | 3. Ошибки, которые исправить можно, но с трудом, например реальное стирание файла, при котором никаких его копий не остается.                                                                                                                                                |
| 2. Ошибки, исправляемые после выполнения действия, например, после ошибочного уничтожения файла его копия переносится из корзины.                                                 | 4. Ошибки, которые на практике невозможно исправить, т.е. ошибки, которые по тем или иным причинам невозможно обнаружить формальной проверкой (т.е. невозможно обнаружить их не случайно). Пример: грамматическая ошибка в тексте, формально удовлетворяющем правилам языка. |

Ошибки, «исправляемые после», снижают производительность работы. Всякий раз, когда пользователь обнаруживает, что он совершает ошибку, ему приходится возвращаться назад на несколько этапов.

Чтобы исправить совершенную ошибку, от пользователя требуется:

- понять, что ошибка совершена;
- понять, как её исправить;
- потратить время на исправление ошибки.

В результате значительный процент времени уходит не на действие (т.е. на продуктивную работу), а на исправление ошибок.

Ошибки, «исправляемые после», *воспринимаются* пользователем как ошибки. Ошибки же, исправляемые во время, как ошибки не воспринимаются, просто потому, что для пользователей это не ошибки вообще. Ошибка же, не воспринимаемая как таковая, пользователей не раздражает, что весьма положительно действует на их субъективное удовлетворение системой.

**Наличие человеческих ошибок, которых нельзя обнаружить и исправить до окончательного совершения действия, всегда свидетельствует о недостаточно хорошем качестве интерфейса.**

Исправить что-либо «во время» можно только тогда, когда во время совершения действия видно, что происходит и как это действие повлияет на изменяемый объект. Соответственно, чтобы дать пользователям исправлять их действия на ходу, этим пользователям надо дать обратную связь.

## **8.8. Обучение работе с системой**

В традиционной науке о человеко-машинном взаимодействии роль обучения операторов чрезвычайно велика.

Начиная с определенного объема функциональности системы, количество пользователей, знающих всю функциональность, неуклонно снижается.

Чем объемней система, тем больше шансов на то, что среднестатистический пользователь знает о ней очень немного (относительно общего объема функциональности).

Плохо это по многим причинам:

- пользователи работают с системой не эффективно, поскольку вместо методов адекватных они используют методы знакомые;
- часто случается, что пользователи, не зная, что имеющийся продукт делает то, что им нужно, ищут (и находят) продукт конкурента;
- затруднительно продавать новые версии продукта: если пользователь не умеет пользоваться и тем.

Есть ещё одно правило: пользователь будет учиться какой-либо функции, только если он знает о её существовании и что её использование даст эффект. Таким образом, чтобы пользователь начал учиться, ему нужно рассказать о функциональности системы. А дальше пользователь сам будет учиться, если, конечно, стимул достаточен.

Человек начнет обучение, если будут обучающие материалы и общая «понятность» системы.

**Понятность системы** включает: ментальную модель, метафору, аффорданс и стандарт.

### **8.8.1. Ментальная модель**

Чтобы успешно пользоваться какой-либо системой, человеку необходимо однозначно понимать, как система работает. При этом необязательно точно понимать сущность происходящих в системе процессов, более того, необязательно правильно их понимать. Это понимание сущности системы называется ментальной моделью.

Утюгом никогда не сможет воспользоваться человек, который не знает, что провод от утюга надо воткнуть в розетку. Но, обладая таким знанием, человек может пользоваться утюгом, не зная, сколько энергии утюг потребляет, что по проводам течёт электричество.

Таким образом, без корректной ментальной модели пользователи неспособны научиться пользоваться системой.

### **8.8.2. Метафора**

Как было сказано, чтобы научиться пользоваться системой, пользователю нужно построить ментальную модель этой системы. Можно воспользоваться применением метафоры, которая позволяет пользователю не создавать новую модель, воспользоваться моделью, которую он ранее построил по другому поводу.

Самым простым примером метафоры в интерфейсе является устройство программ для проигрывания звуков на компьютере. Исторически сложилось, что вся аудиотехника имеет почти одинаковый набор кнопок: несколько кнопок со стрелками (назад/вперед), кнопка с треугольником (воспроизведение), кнопка с двумя дощечками (пауза), кнопка с квадратиком (полная остановка) и красный кружок (запись). Про них нельзя сказать, что они совершенно понятны, но научиться им можно без труда.

При этом обычно жизнь складывается так, что сначала человек учится пользоваться этими кнопками на материальных устройствах, а уж потом начинает

пользоваться компьютером.

Соответственно, при проектировании программы аналогичного назначения разумно скопировать существующую систему маркировки кнопок. Благодаря этому пользователям для использования программы ничему не приходится учиться.

Недостатки метафор:

- не для любой функциональности можно подобрать подходящую метафору;
- подходящая метафора может оказаться бесполезной, если её не знает существенная часть аудитории или её тяжело однозначно передать интерфейсом;
- метафора будет сковывать функциональные возможности, если проектируемая система обладает большим количеством функций, чем копируемый образец;
- метафору можно использовать в документации, не перенося её в интерфейс, достаточно просто написать, что «система во многом напоминает ...» и нужный результат будет достигнут.

Таким образом, метафора – средство для избавления пользователя от обучения.

### **8.8.3. Аффорданс**

В современном значении этого термина аффордансом называется ситуация, при которой объект показывает субъекту способ своего использования своими неотъемлемыми свойствами. Например, надпись «На себя» на двери не является аффордансом, а *облик* двери, который подсказывает человеку, что она открывается на себя, несет в себе аффорданс.

Полезьа аффорданса заключается в том, что он позволяет пользователям обходиться без какого-либо предварительного обучения.

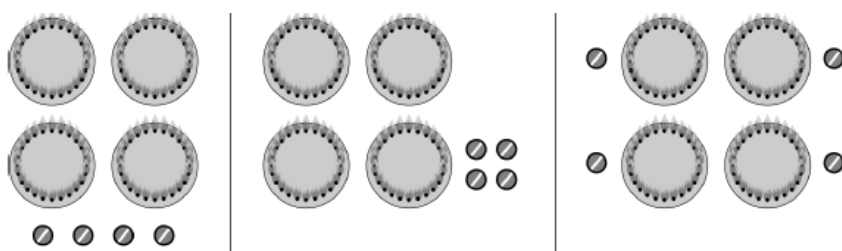


Рис. 6. Отсутствие аффорданса в дизайне кухонной плиты. Слева стандартный вариант, недостаток которого заключается в том, что невозможно умозрительно определить, какой диск управляет какой конфоркой. В центре и справа варианты с аффордансом, не имеющие этой проблемы, при этом работающие по-разному. В центральном примере расположение регуляторов повторяет расположение рабочих объектов (конфорок), благодаря чему неоднозначность исчезает. В правом примере каждому объекту соответствует отдельный регулятор. Эти два способа уничтожения неопределенности являются основными в экранном дизайне.

В целом, создание аффордансов является сложной задачей, стоящей перед графическим дизайнером, работающим над интерфейсом.

#### 8.8.4. Стандарт

Если что-либо нельзя сделать «самопроизвольно» понятным, всегда можно сделать это везде одинаково, чтобы пользователи обучались только один раз.

Например, кран с горячей водой всегда маркируют красным цветом, а кран с холодной – синим. Частично это соответствует свойствам человеческого восприятия (недаром красный цвет мы называем тёплым, а синий – холодным), но в основном здесь работает привычка.

Таким образом, чтобы стандарт заработал, он должен быть популярен. Например, стандарт интерфейса MS Windows популярен почти во всех программах для Windows, именно поэтому его нужно придерживаться.

#### 8.8.5. Обучающие материалы

**Базовая справка** - объясняет пользователю сущность и назначение системы.

**Обзорная справка** - рекламирует пользователю функции системы.

**Справка предметной области** отвечает на вопрос «Как сделать хорошо?».

Поскольку от пользователей зачастую нельзя рассчитывать знания предметной области, необходимо снабжать их этим знанием на ходу.

**Процедурная справка** отвечает на вопрос «Как это сделать?», должна быть максимально доступна, поскольку если пользователь не найдет нужную информацию быстро, он перестанет искать и так и не научится пользоваться функцией.

**Контекстная справка** отвечает на вопросы «Что это делает?» и «Зачем это нужно?».

**Справка состояния** отвечает на вопрос «Что происходит в настоящий момент?». Поскольку она требуется именно что в настоящий момент, она не может быть вынесена из интерфейса.

#### **8.8.6. Эстетика интерфейса**

Все знают, что значительно легче и приятнее пользоваться эстетически привлекательными объектами. Это наблюдение породило весь промышленный дизайн. Принципы многих направлений дизайна вполне применимы к дизайну интерфейсов.

Интерфейс передает информацию своему пользователю. Интерфейс обычно предназначен для длительного использования. Это серьезно сближает его с промышленным дизайном и дизайном среды обитания.

Часто приходится искать компромисс между эстетикой и функцией.

Какие их принципы могут быть использованы в дизайне интерфейса?

- конструируемый предмет должен быть незаметен при его использовании, но он должен приятно ощущаться на бессознательном уровне;
- избегайте ярких цветов и острых углов;
- делайте изображение максимально более легким и воздушным;
- добиваться контраста не сменой насыщенности элементов, но расположением пустот;
- делайте интерфейс максимально насыщенным визуальными закономерностями.

Чем больше закономерностей у разных объектов, тем больше гармонии:

- высота кнопок может быть равна удвоенному значению полей диалогового окна;
- стремитесь не столько к красоте интерфейса, сколько к его элегантности;
- старайтесь использовать модульные сетки, т.е. привязывайте все объекты к линиям (лучше узлам) воображаемой сетки, которую выдерживайте во всем интерфейсе;
- старайтесь привязывать все размеры и координаты (как минимум пропорции диалоговых окон) к золотому сечению (0.618 x 0.382).

Любая красота со временем надоедает и в лучшем случае перестает восприниматься, поэтому в интерфейсах обычно элегантность и гармония гораздо лучше.

## **8.9. Элементы управления**

### **8.9.1. Кнопки**

Кнопкой называется элемент управления, всё взаимодействие пользователя с которым ограничивается одним действием – нажатием.

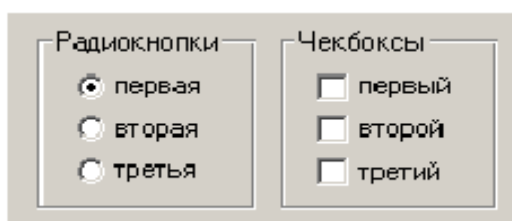
**Командные кнопки** - нажатие на такую кнопку запускает какое-либо явное действие.



Чем больше кнопка, тем легче попасть в нее курсором. Одновременно пользователю должно быть трудно нажать не на ту кнопку. Добиться этого можно либо изменением состояния кнопки при наведении на неё курсором, либо установлением пустого промежутка между кнопками.

Помимо текста, на кнопках можно выводить пиктограммы. Пиктограммы должны передавать пользователям идею действия (т.е. глагол), а действие плохо передается пиктограммами.

### 8.9.2. Чекбоксы и радиокнопки



Они являются кнопками отложенного действия, т.е. их нажатие ни при каких обстоятельствах не должно инициировать какое-либо немедленное действие. С их помощью пользователи вводят параметры, которые скажутся после, когда действие будет запущено иными элементами управления.

Главное различие этих кнопок заключается в том, что группа чекбоксов дают возможность пользователям выбрать любую комбинацию параметров, радиокнопки же позволяют выбрать только один параметр.

У чекбокса есть три состояния (выбранное, не выбранное, смешанное), а у радиокнопки только два, поскольку смешанного состояния у неё быть просто не

может.

Если параметров больше двух, выбор прост: если параметры можно комбинировать, нужно использовать чекбоксы (например, текст может быть одновременно и жирным и курсивным); если же параметры комбинировать нельзя, нужно использовать радиокнопки (например, текст может быть выровнен или по левому, или по правому краю).

И чекбоксы и радиокнопки крайне желательно расставлять по вертикали, поскольку это значительно ускоряет поиск нужного элемента

Внешний вид. Традиционно сложилось так, что чекбоксы выглядят как квадраты, а радиокнопки – как кружки. Нарушать это правило нельзя.

Текст подписей. Каждая подпись должна однозначно показывать эффект от выбора соответствующего элемента. Поскольку как радиокнопки, так и чекбоксы, не вызывают немедленного действия, формулировать подписи к ним лучше всего в форме существительных, хотя возможно использование глаголов.

### 8.9.3.Списки

Списки бывают **пролистываемыми и раскрывающимися**, причем пролистываемые могут обеспечивать как единственный, так и множественный выбор, а раскрывающиеся работают исключительно как радиокнопки.

Ширина списка как минимум должна быть достаточна для того, чтобы пользователь мог определить различия между элементами.

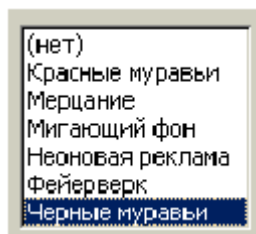
**Раскрывающиеся списки.** Малая высота списка позволяет с большой легкостью визуально отображать команды, собираемые из составляющих.



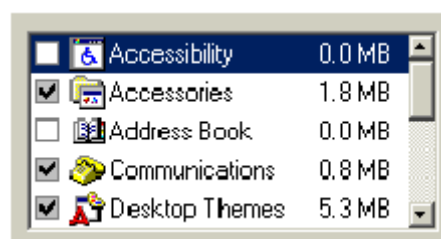
**Пролистываемые списки.** Более сложный вариант списка.

Пролистываемые списки могут позволять пользователям совершать как единственный, так и множественный выбор.

Списки единственного выбора. **Список единственного выбора является промежуточным вариантом между группой радиокнопок и раскрывающимся списком. Он меньше группы радиокнопок с аналогичным числом элементов, но больше раскрывающегося списка.**

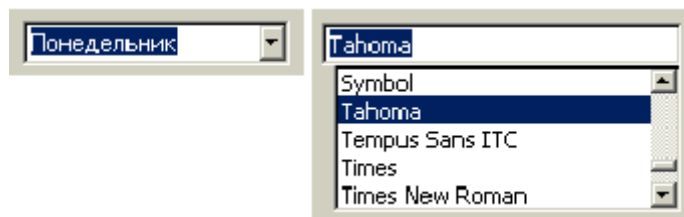


Списки множественного выбора.



#### 8.9.4. Комбобоксы

Комбобоксами (combo box), называются гибриды списка с полем ввода: пользователь может выбрать существующий элемент, либо ввести свой. Комбобоксы бывают двух видов: раскрывающиеся и расширенные.



Раскрывающийся комбобокс с установленным фокусом ввода (слева) и расширенный комбобокс (справа).

Раскрывающийся комбобокс выглядит в точности как раскрывающиеся списки, визуально отличаясь от них только наличием индикатора фокуса ввода. Комбобоксом можно пользоваться, как обычным списком.

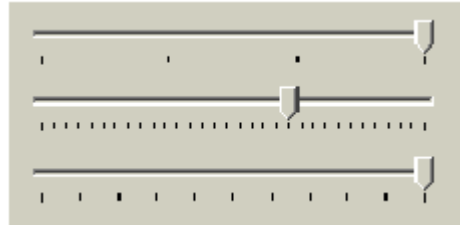
### 8.9.5. Крутилки и ползунки

Крутилка (spinner, little arrow) это поле ввода.



При вводе значения мышью не теряется время из-за переноса руки с клавиатуры, не сбивает фокус внимания, а кроме того система может позволить пользователям вводить только корректные данные и в корректном формате. Это резко уменьшает вероятность человеческой ошибки. Таким образом, использование крутилок для ввода любых численных значений оправдано.

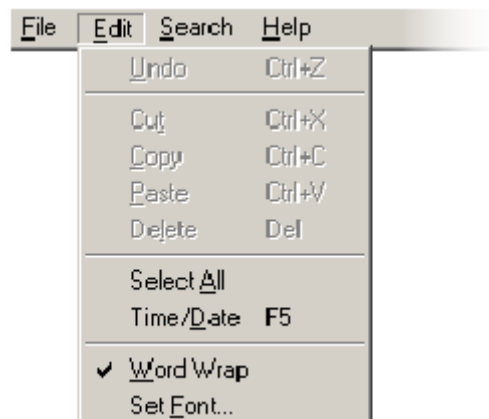
Ползунки позволяют пользователям выбирать значение из списка, не позволяя вводить произвольное значение, особенно, если значения, стоят в хорошо ранжирующемся ряду.



Видно, что количество параметров в ползунке может быть весьма значительным.

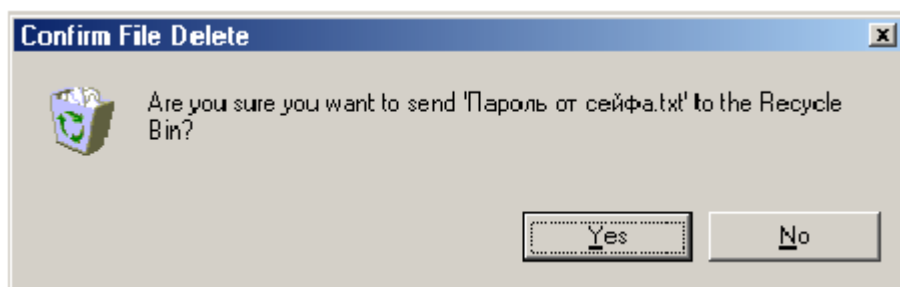
### 8.9.6. Меню

Меню – это метод взаимодействия пользователя с системой, при котором пользователь выбирает из предложенных вариантов, а не предоставляет системе свою команду. Хотя, при упоминании термина меню применительно к интерфейсу, большинство людей немедленно представляют стандартные раскрывающиеся меню.



Стандартное выпадающее меню.

Соответственно, диалоговое окно с несколькими кнопками (и без единого поля ввода) также является меню.



Меню позволяет снизить нагрузку на мозги пользователей, поскольку для выбора команды не надо вспоминать, какая именно команда нужна и как именно её нужно использовать – вся нужная информация уже содержится на экране.

Поскольку меню ограничивает диапазон действий пользователей, появляется возможность в значительной мере изъять из этого диапазона ошибочные действия.

Меню показывает пользователям объем действий, которые они могут совершить благодаря системе, и тем самым обучают пользователей (самым эффективным средством обучения).

Типы меню:

- статические меню, т.е. меню, постоянно присутствующие на экране.

Характерным примером такого типа меню является панель инструментов;

- динамические меню, в которых пользователь должен вызвать меню, чтобы выбрать какой-либо элемент. Примером является обычное контекстное меню.

В некоторых ситуациях эти два типа меню могут сливаться в один.

Контекстные меню позволяют максимально повысить скорость работы пользователей. Преимущество контекстных (всплывающих) меню заключается в том, что они полностью встраиваются в контекст действий пользователей: не нужно переводить взгляд и курсор в другую область экрана, практически не нужно прерывать текущее действие для выбора команды. При этом они не занимают места на экране, что всегда ценно. С другой стороны, из-за того, что они не находятся всё время на экране, они практически неспособны чему-либо научить пользователя.