Введение в информационные технологии

Лепустин А.В.
старший преподаватель
каф. ВТ ИК

Структура курса

- Лекции – 8шт (16 часов)
- Лаб. работы – 6шт x 6 = 36 баллов
- Реферат – 15 баллов
- Самостоятельная работа – 9 баллов
- Экзамен
  - Письменная часть – 30 баллов
  - Устная часть – 10 баллов
- Всего за курс – 100 баллов

- Материалы: персональная страница преподавателя http://portal.tpu.ru:7777/SHARED/k/KIM/
- http://habrahabr.ruhttp://habrahabr.ru http://ixbt.com
- http://google.ruhttp://google.ru http://eetimes.com

Оформление ЛР

- цель работы
- постановка задачи
- схема алгоритма (в соответствии с ГОСТ 19.701-90)
- листинг программы (с комментариями основных действий)
- результаты работы программы и ручного тестирования
- выводы по работе

Правила

- Начисление баллов за ЛР, реферат:
  - **Сдача в срок** – в соответствии с качеством исполнения / защиты
  - **Сдача не в срок** – 60% от баллов, начисленных в соответствии с качеством исполнения / защиты
  - Дополнительные баллы (до 10 баллов) – за выступление на лекции с докладом по доп. темам

Экзамен

- Студент допускается к экзамену, если выполняются **все** следующие условия:
  - защищены все лабораторные работы
  - подготовлен и защищен реферат
  - реферат отправлен лектору (через ЛК)
  - набрано 33 и более баллов
- Во время экзамена нельзя:
  - Книги/лекции/шпаргалки/«парашюты»/пр.
  - Телефоны/калькуляторы/пр. гаджеты
- Экзамен проводится:
  - по темам лекций, рефератов
  - в письменной форме (решение задач)
  - в устной форме (ответы на вопросы)
    - «Расскажите всё, что знаете про…»
    - «Чем … отличается от …»
    - «Сравните … и …, что лучше и почему?»

А если не набрано 33 балла?

- Других способов набора баллов в рейтинг-плане нет

- PS: такого пока не было, но Вы можете быть первым! ☺

Общие сведения

Общие сведения

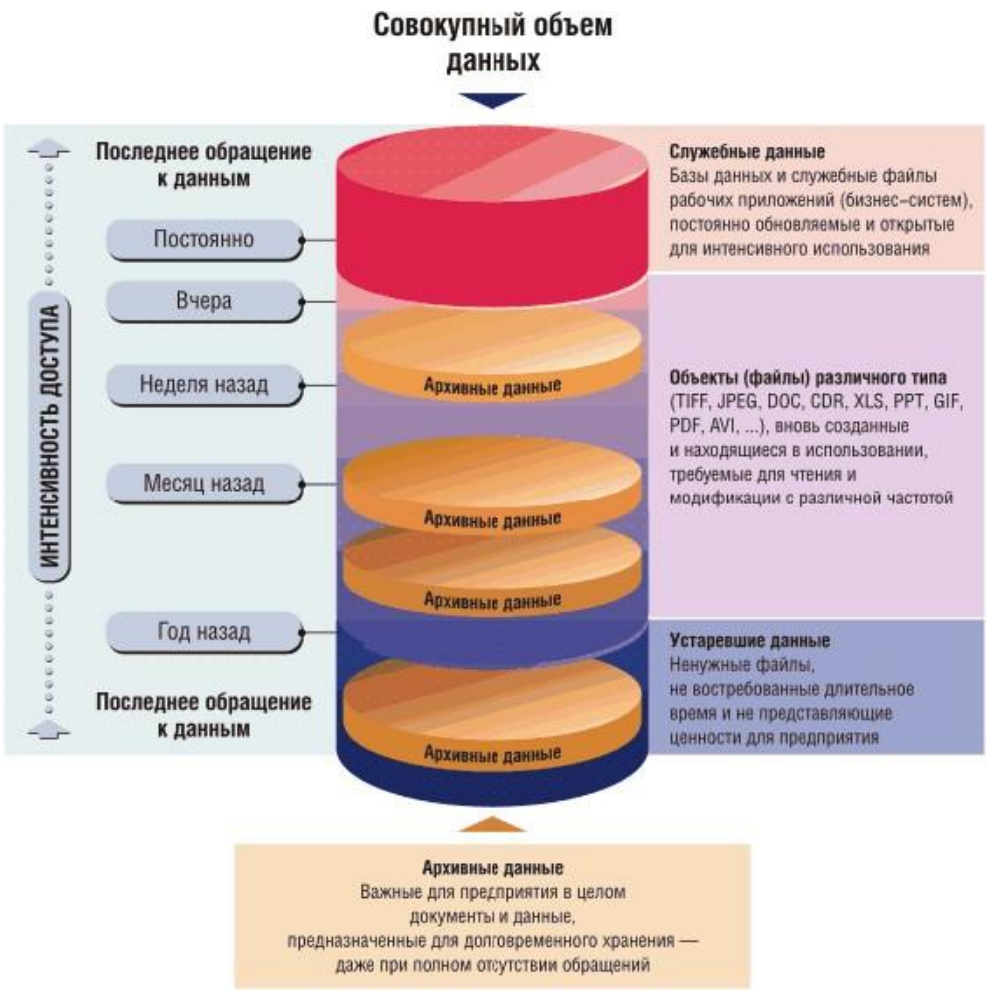
- **Информационные технологии** (ИТ, от англ. information technology, IT) — широкий класс дисциплин и областей деятельности, относящихся к технологиям управления и обработки данных, в том числе, с применением вычислительной техники.
- *Информационные технологии = компьютерные технологии?*
- ИТ имеют дело с использованием компьютеров и программного обеспечения для хранения, преобразования, защиты, обработки, передачи и получения информации.
- Специалистов по компьютерной технике и программированию часто называют ИТ-специалистами.

Общие сведения

- ЮНЕСКО: ИТ — это комплекс взаимосвязанных научных, технологических, инженерных дисциплин, изучающих методы эффективной организации труда людей, занятых обработкой и хранением информации; вычислительную технику и методы организации и взаимодействия с людьми и производственным оборудованием, их практические приложения, а также связанные со всем этим социальные, экономические и культурные проблемы.

- Сами ИТ требуют сложной подготовки, больших первоначальных затрат и наукоемкой техники. Их введение должно начинаться с создания математического обеспечения, формирования информационных потоков в системах подготовки специалистов.

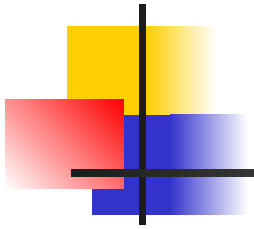
Общие сведения



- Основные черты современных ИТ:
  - компьютерная обработка информации по заданным алгоритмам
  - хранение больших объёмов информации на машинных носителях
  - передача информации на любые расстояния в ограниченное время.

Общие сведения

- Дисциплина информационных технологий:
  - В широком понимании ИТ охватывает все области передачи, хранения и восприятия информации (не только компьютерные технологии).



Информа-ционные системы

Информационные системы

- **В широком смысле** информационная система есть совокупность технического, программного и организационного обеспечения, а также персонала, предназначенная для того, чтобы своевременно обеспечивать надлежащих людей надлежащей информацией.

Информационные системы

- Федеральный закон Российской Федерации от 27 июля 2006 г. N 149-ФЗ «Об информации, информационных технологиях и о защите информации»:
  - Информационная система – совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств»
- Включать ли персонал в ИС?
  - в ФЗ нет уточнений
  - мнения специалистов расходятся

Информационные системы

- **В узком смысле** информационной системой называют только подмножество компонентов ИС в широком смысле, включающее базы данных, СУБД и специализированные прикладные программы

Информационные системы

- Основная задача ИС:
  - удовлетворение конкретных информационных потребностей в рамках конкретной предметной области.
- Современные ИС де-факто немыслимы без использования баз данных и СУБД, поэтому термин «информационная система» на практике сливается по смыслу с термином «система баз данных».

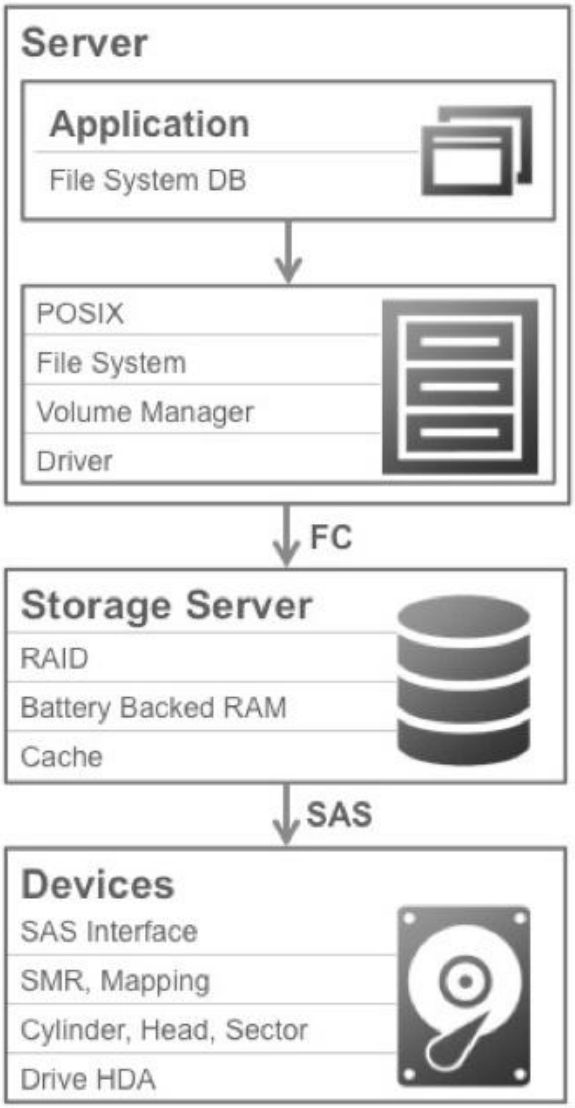
Информационные системы

- ИС по степени распределённости различают:
  - **настольные** (desktop), или локальные ИС, в которых все компоненты (БД, СУБД, клиентские приложения) работают на одном компьютере
  - **распределённые** (distributed) ИС, в которых компоненты распределены по нескольким компьютерам.

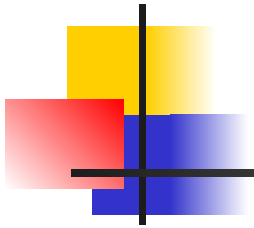
Информационные системы

- Распределённые ИС:
  - **файл-серверные** ИС (ИС с архитектурой «файл-сервер») - база данных находится на файловом сервере, а СУБД и клиентские приложения находятся на рабочих станциях
  - **клиент-серверные** ИС (ИС с архитектурой «клиент-сервер») - база данных и СУБД находятся на сервере, а на рабочих станциях находятся клиентские приложения

Информационные системы



- клиент-серверные ИС:
  - В двухзвенных (two-tier) ИС всего два типа «звеньев»: **сервер баз данных**, на котором находятся БД и СУБД, и **рабочие станции**, на которых находятся клиентские приложения (КП). Клиентские приложения обращаются к СУБД напрямую.
    - Бизнес-логика может быть размещена либо в БД, либо на КП
  - В многозвенных (multi-tier) ИС добавляются промежуточные «звенья»: **серверы приложений** (СП, application servers). Пользовательские клиентские приложения не обращаются к СУБД напрямую, они взаимодействуют с промежуточными звеньями.
    - Бизнес-логика может быть размещена в БД, на СП, в КП. Размещение логики в БД или на СП позволяет реализовать «тонкий клиент» (особенно актуально при реализации мультиплатформенности)



Базы данных

Базы данных

- Базой данных является представленная в объективной форме **совокупность самостоятельных материалов** (статей, расчетов, нормативных актов, судебных решений и иных подобных материалов), **систематизированных** таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью **электронной вычислительной машины** (Гражданский кодекс РФ, ст. 1260).

Базы данных

- **База данных** — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных. (ISO/IEC TR 10032:2003 Information technology — Reference model of data management)

- **База данных** — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причём такое собрание данных, которое поддерживает одну или более областей применения (ISO/IEC 2382-1:1993. Information technology — Vocabulary — Part 1: Fundamental terms)

Базы данных

- База данных — организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера **совокупность данных**, характеризующая актуальное состояние некоторой **предметной области** и используемая для удовлетворения **информационных потребностей** пользователей (Когаловский М. Р. Энциклопедия технологий баз данных)

- База данных — некоторый набор **перманентных** (постоянно хранимых) **данных**, используемых прикладными программными системами какого-либо предприятия (Дейт К. Дж. Введение в системы баз данных)

- База данных — совместно используемый набор **логически связанных данных** (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации (Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика)

Базы данных

- Отличительные признаки:
  - База данных хранится и обрабатывается в **вычислительной системе**. Таким образом, любые внекомпьютерные хранилища информации (архивы, библиотеки, картотеки и т. п.) базами данных не являются.
  - Данные в базе данных **логически структурированы** (систематизированы) с целью обеспечения возможности их эффективного поиска и обработки в вычислительной системе.
  - Структурированность подразумевает **явное выделение составных частей** (элементов), **связей** между ними, а также **типизацию** элементов и связей, при которой с типом элемента (связи) соотносится определённая семантика и допустимые операции (оценивается не физическое хранение, а уровень модели)
  - База данных включает **метаданные**, описывающие логическую структуру БД в формальном виде (в соответствии с некоторой метамоделью).

Базы данных

- Совокупность данных – БД или нет? Определяется общепринятой практикой

- Не называют базами данных файловые архивы, Интернет-порталы или электронные таблицы, несмотря на то, что они в некоторой степени обладают признаками БД. Принято считать, что эта степень в большинстве случаев недостаточна (хотя могут быть исключения).

Базы данных

- Классификация БД по модели данных:
  - Иерархические
  - Сетевые
  - Реляционные
  - Объектные
  - Объектно-ориентированные
  - Объектно-реляционные

Базы данных

- Классификация БД по технологии хранения:
  - БД в третичной памяти (tertiary databases): магнитные ленты и оптические диски, кэш и оперативные данные – на HDD, загрузка данных – спецпроцедура
  - БД во вторичной памяти (традиционные): хранение на HDD, кэш – в ОП
  - БД в оперативной памяти (in-memory databases): вся БД в ОП

Базы данных

- Классификация БД по степени распределённости:
  - Централизованные (сосредоточенные)
  - Распределённые

Базы данных

- Отдельно:
  - пространственные (spatial)
  - временные или темпоральные (temporal)
  - пространственно-временные (spatial-temporal)

Базы данных

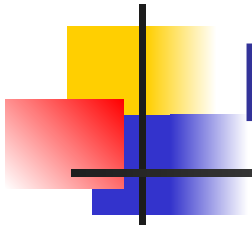
- БД и СУБД

- Многие специалисты указывают на распространённую ошибку, состоящую в некорректном использовании термина база данных вместо термина система управления базами данных. Эти понятия, следовательно, необходимо различать.

Базы данных

- СУБД – специализированная программа (чаще комплекс программ), предназначенная для организации и ведения базы данных.

- Для создания и управления информационной системой СУБД необходима в той же степени, как для разработки программы на алгоритмическом языке необходим транслятор



Базы данных

- Функции СУБД
  - управление данными во внешней памяти (на дисках)
  - управление данными в оперативной памяти с использованием дискового кэша
  - журнализация изменений, резервное копирование и восстановление базы данных после сбоев
  - поддержка языков БД (язык определения данных, язык манипулирования данными).

Базы данных

- Компоненты СУБД:
  - **ядро**, которое отвечает за управление данными во внешней и оперативной памяти, и журнализацию
  - **процессор языка базы данных**, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода
  - **подсистему поддержки времени исполнения**, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД
  - **сервисные программы** (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

Базы данных

- Классификация СУБД по модели данных:
  - Иерархические
  - Сетевые
  - Реляционные
  - Объектно-ориентированные

Базы данных

- Классификация СУБД по степени распределённости:
  - **локальные** СУБД (все части локальной СУБД размещаются на одном компьютере)
  - **распределённые** СУБД (части СУБД могут размещаться на двух и более компьютерах).

Базы данных

- Классификация СУБД по способу доступа к БД:
  - **Файл-серверные**. Файлы данных располагаются централизованно на файл-сервере. СУБД располагается на каждом клиентском компьютере (рабочей станции). Доступ СУБД к данным осуществляется через локальную сеть. Синхронизация чтений и обновлений осуществляется посредством файловых блокировок.
    - Преимущество: низкая нагрузка на ЦП сервера.
    - Недостатки:
      - потенциально высокая загрузка локальной сети;
      - затруднённость централизованного управления;
      - затруднённость обеспечения таких важных характеристик как высокая надёжность, высокая доступность и высокая безопасность.
    - Примеры: Microsoft Access, Paradox, dBase, FoxPro, Visual FoxPro
    - В настоящее время практически не используются

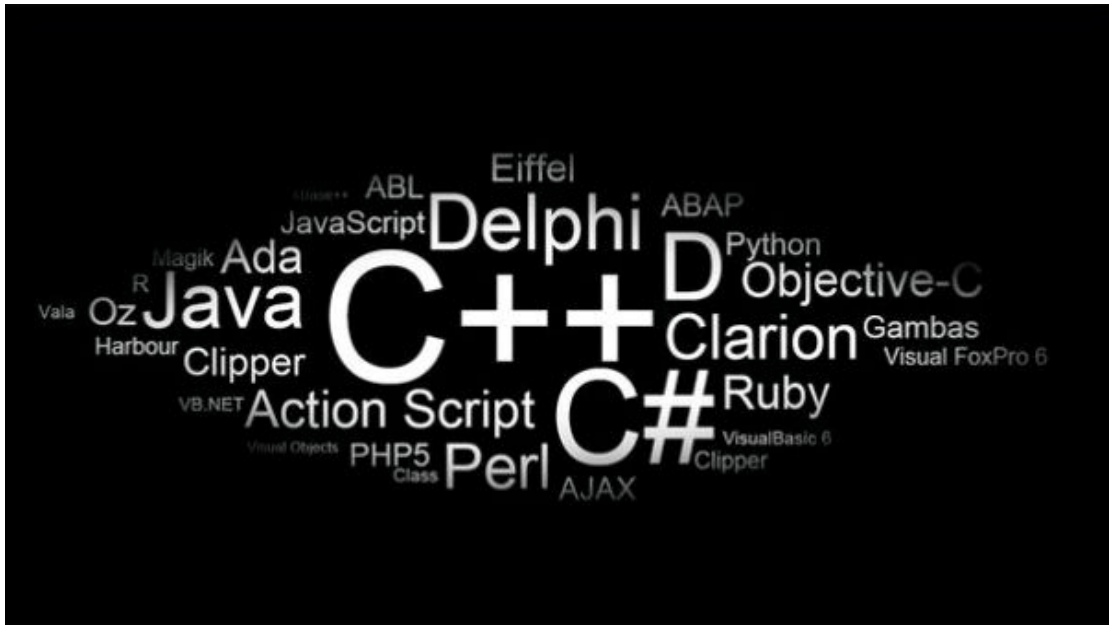
Базы данных

- Классификация СУБД по способу доступа к БД:
  - **Клиент-серверные.** СУБД располагается на сервере вместе с БД и осуществляет доступ к БД непосредственно, в монопольном режиме. Все клиентские запросы на обработку данных обрабатываются клиент-серверной СУБД централизованно.
    - Недостаток: повышенные требования к серверу
    - Достоинства:
      - потенциально более низкая загрузка локальной сети;
      - удобство централизованного управления;
      - удобство обеспечения таких важных характеристик как высокая надёжность, высокая доступность и высокая безопасность.
    - Примеры: Oracle, MS SQL Server, Firebird, MySQL, Interbase, IBM DB2, Sybase, PostgreSQL, ЛИНТЕР, MDBS.

Базы данных

- Классификация СУБД по способу доступа к БД:
  - **Встраиваемая СУБД**. Библиотека, которая позволяет унифицированным образом хранить большие объёмы данных на локальной машине. Доступ к данным может происходить через SQL либо через особые функции СУБД. Встраиваемые СУБД быстрее обычных клиент-серверных и не требуют установки сервера, поэтому востребованы в локальном ПО, которое имеет дело с большими объёмами данных (например, геоинформационные системы).
    - Примеры: OpenEdge, SQLite, BerkeleyDB, один из вариантов Firebird, MySQL, Sav Zigzag, Microsoft SQL Server Compact, ЛИНТЕР.

Стадии разработки ПО и ПД



ИС. Стадии разработки ПО и ПД

- Жизненный цикл информационной системы – это процесс ее построения и развития.
- Жизненный цикл информационной системы — период времени, который начинается с момента принятия решения о необходимости создания информационной системы и заканчивается в момент ее полного изъятия из эксплуатации

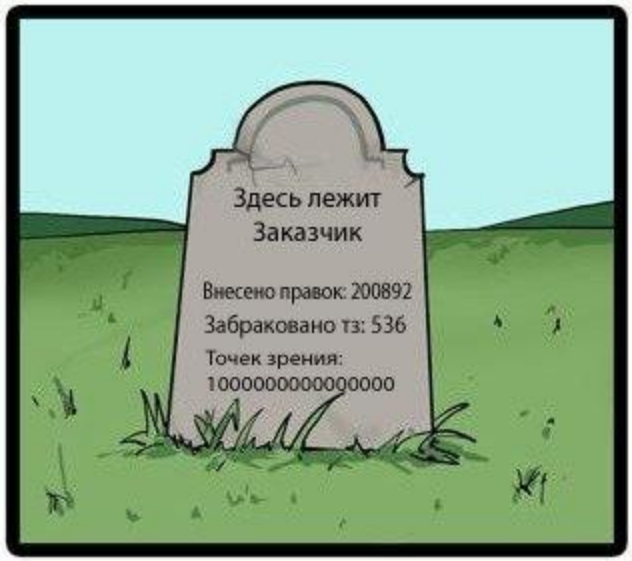
ИС. Стадии разработки ПО и ПД

ИС. Стадии разработки ПО и ПД

- Регламентируются ГОСТами:
  - ГОСТ 19.102-77 Стадии разработки
  - ГОСТ 34.601-90 Автоматизированные системы. Стадии создания
  - ГОСТ Р ИСО/МЭК 12207-2010 Процессы жизненного цикла программных средств

ИС. Стадии разработки ПО и ПД

- ГОСТ 19.102-77
  - 1. Техническое задание
  - 2. Эскизный проект
  - 3. Технический проект
  - 4. Рабочий проект
  - 5. Внедрение
- ГОСТ 34.601-90
  - 1. Формирование требований к АС
  - 2. Разработка концепции АС
  - 3. Техническое задание
  - 4. Эскизный проект
  - 5. Технический проект
  - 6. Рабочая документация
  - 7. Ввод в действие
  - 8. Сопровождение АС



ИС. Стадии разработки ПО и ПД

ГОСТ 19.102-77	ГОСТ 34.601-90
	1. Формирование требований к АС
	2. Разработка концепции АС
1. Техническое задание	3. Техническое задание
2. Эскизный проект	4. Эскизный проект
3. Технический проект	5. Технический проект
4. Рабочий проект	6. Рабочая документация
5. Внедрение	7. Ввод в действие
	8. Сопровождение АС

ИС. Стадии разработки ПО и ПД

- ГОСТ 19.102-77

Стадии разработки	**Этапы работ**	**Содержание работ**
1. Техническое задание	Обоснование необходимости разработки программы	Постановка задачи Сбор исходных материалов Выбор и обоснование критериев эффективности и качества разрабатываемой программы. Обоснование необходимости проведения научно-исследовательских работ.
	Научно-исследовательские работы	Определение структуры входных и выходных данных. Предварительный выбор методов решения задач. Обоснование целесообразности применения ранее разработанных программ. Определение требований к техническим средствам. Обоснование принципиальной возможности решения поставленной задачи
	Разработка и утверждение технического задания	Определение требований к программе. Разработка технико-экономического обоснования разработки программы. Определение стадий, этапов и сроков разработки программы и документации на неё. Выбор языков программирования. Определение необходимости проведения научно-исследовательских

ИС. Стадии разработки ПО и ПД

- ГОСТ 19.102-77

Стадии разработки	**Этапы работ**	**Содержание работ**
2. Эскизный проект	Разработка эскизного проекта	Предварительная разработка структуры входных и выходных данных. Уточнение методов решения задачи. Разработка общего описания алгоритма решения задачи Разработка технико-экономического обоснования.
	Утверждение эскизного проекта	Разработка пояснительной записки. Согласование и утверждение эскизного проекта.

ИС. Стадии разработки ПО и ПД

- ГОСТ 19.102-77

Стадии разработки	**Этапы работ**	**Содержание работ**
3. Технический проект	Разработка технического проекта	Уточнение структуры входных и выходных данных. Разработка алгоритма решения задачи. Определение формы представления входных и выходных данных. Определение семантики и синтаксиса языка. Разработка структуры программы. Окончательное определение конфигурации технических средств.
	Утверждение технического проекта	Разработка плана мероприятий по разработке и внедрению программ. Разработка пояснительной записки. Согласование и утверждение технического проекта.

ИС. Стадии разработки ПО и ПД

- ГОСТ 19.102-77

Стадии разработки	**Этапы работ**	**Содержание работ**
4. Рабочий проект	Разработка программы	Программирование и отладка программы.
	Разработка программной документации	Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77 **Виды программ и программных документов.**
	Испытания программы	Разработка, согласование и утверждение порядка и методики испытаний. Проведение предварительных государственных, межведомственных, приёмо-сдаточных и других видов испытаний. Корректировка программы и программной

ИС. Стадии разработки ПО и ПД

- ГОСТ 19.102-77

Стадии разработки	**Этапы работ**	**Содержание работ**
5. Внедрение	Подготовка и передача программы.	Подготовка и передача программы и программной документации для сопровождения и (или) изготовления. Оформление и утверждение акта о передаче программы на сопровождение и (или) изготовление. Передача программы в фонд алгоритмов и программ.

ИС. Стадии разработки ПО и ПД

- ГОСТ 34.601-90

Стадии разработки	Этапы работ	Содержание работ
1. Формирование требований к АС	1.1. Обследование объекта и обоснование необходимости создания АС	- сбор данных об объекте автоматизации и осуществляемых видах деятельности; - оценку качества функционирования объекта и осуществляемых видов деятельности, выявление проблем, решение которых возможно средствами автоматизации; - оценку (технико-экономической, социальной и т. п.) целесообразности создания АС.
	1.2. Формирование требований пользователя к АС	- подготовку исходных данных для формирования требований к АС (характеристика объекта автоматизации, описание требований к системе, ограничения допустимых затрат на разработку, ввод в действие и эксплуатацию, эффект, ожидаемый от системы, условия создания и функционирования системы); - формулировку и оформление требований пользователя к АС.
	1.3. Оформление отчета о выполненной работе и заявки на разработку АС (тактико-технического задания)	Проводят оформление отчета о выполненных работах на данной стадии и оформление заявки на разработку АС (тактико-технического задания) или другого заменяющего ее документа с аналогичным содержанием.

ИС. Стадии разработки ПО и ПД

- ГОСТ 34.601-90

Стадии разработки	Этапы работ	Содержание работ
2. Разработка концепции АС	2.1. Изучение объекта	Организация-разработчик проводит детальное изучение объекта автоматизации и необходимые научно-исследовательские работы (НИР), связанные с поиском путей и оценкой возможности реализации требований пользователя, оформляют и утверждают отчеты о НИР.
	2.2. Проведение необходимых научно-исследовательских работ	
	2.3. Разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователя	Проводят разработку альтернативных вариантов концепции создаваемой АС и планов их реализации; оценку необходимых ресурсов на их реализацию и обеспечение функционирования; оценку преимуществ и недостатков каждого варианта; сопоставление требований пользователя и характеристик предлагаемой системы и выбор оптимального варианта; определение порядка оценки качества и условий приемки системы; оценку эффектов, получаемых от системы.
	2.4. Оформление отчета о выполненной работе	Подготавливают и оформляют отчет, содержащий описание выполненных работ на стадии, описание и обоснование предлагаемого варианта концепции системы.

ИС. Стадии разработки ПО и ПД

- ГОСТ 34.601-90

Стадии разработки	**Этапы работ**	**Содержание работ**
3. Техническое задание	3.1. Разработка и утверждение технического задания на создание АС	Проводят разработку, оформление, согласование и утверждение технического задания на АС и, при необходимости, технических заданий на части АС.
4. Эскизный проект	4.1. Разработка предварительных проектных решений по системе и ее частям	Определяют: функции АС; функции подсистем, их цели и эффекты; состав комплексов задач и отдельных задач; концепции информационной базы, ее укрупненную структуру; функции системы управления базой данных; состав вычислительной системы; функции и параметры основных программных средств.
	4.2. Разработка документации на АС и ее части	Проводят разработку, оформление, согласование и утверждение документации в объеме, необходимом для описания полной совокупности принятых проектных решений и достаточном для дальнейшего выполнения работ по созданию АС. Виды документов - по **ГОСТ 34.201**.

ИС. Стадии разработки ПО и ПД

- ГОСТ 34.601-90

Стадии разработки	**Этапы работ**	**Содержание работ**
5. Технический проект	5.1. Разработка проектных решений по системе и ее частям	Обеспечивают разработку общих решений по системе и ее частям, функционально-алгоритмической структуре системы, по функциям персонала и организационной структуре, по структуре технических средств, по алгоритмам решений задач и применяемым языкам, по организации и ведению информационной базы, системе классификации и кодирования информации, по программному обеспечению.
	5.2. Разработка документации на АС и ее части	Проводят разработку, оформление, согласование и утверждение документации в объеме, необходимом для описания полной совокупности принятых проектных решений и достаточном для дальнейшего выполнения работ по созданию АС. Виды документов - по **ГОСТ 34.201 Виды, комплектность и обозначения документов при создании автоматизированных систем**.

ИС. Стадии разработки ПО и ПД

- ГОСТ 34.601-90

Стадии разработки	Этапы работ	Содержание работ
5. Технический проект	5.3. Разработка и оформление документации на поставку изделий для комплектования АС и (или) технических требований (технических заданий) на их разработку	Проводят подготовку и оформление документации на поставку изделий для комплектования АС; определение технических требований и составление ТЗ на разработку изделий, не изготавливаемых серийно.
	5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации	Осуществляют разработку, оформление, согласование и утверждение заданий на проектирование в смежных частях проекта объекта автоматизации для проведения строительных, электротехнических, санитарно-технических и других подготовительных работ, связанных с созданием АС.

ИС. Стадии разработки ПО и ПД

- ГОСТ 34.601-90

Стадии разработки	Этапы работ	Содержание работ
6. Рабочая документация	6.1. Разработка рабочей документации на систему и ее части	Осуществляют разработку рабочей документации, содержащей все необходимые и достаточные сведения для обеспечения выполнения работ по вводу АС в действие и ее эксплуатации, а также для поддерживания уровня эксплуатационных характеристик (качества) системы в соответствии с принятыми проектными решениями, ее оформление, согласование и утверждение. Виды документов - по **ГОСТ 34.201 Виды, комплектность и обозначения документов при создании автоматизированных систем**.
	6.2. Разработка или адаптация программ	Проводят разработку программ и программных средств системы, выбор, адаптацию и (или) привязку приобретаемых программных средств, разработку программной документации в соответствии с **ГОСТ 19.101 Виды программ и программных документов.**

ИС. Стадии разработки ПО и ПД

- ГОСТ 34.601-90

Стадии	Этапы работ	Содержание работ
7. Ввод в действие	7.1. Подготовка объекта автоматизации к вводу АС в действие	Проводят работы по организационной подготовке объекта автоматизации к вводу АС в действие, в т. ч.: реализацию проектных решений по организационной структуре АС; обеспечение подразделений объекта управления инструктивно-методическими материалами; внедрение классификаторов информации.
	7.2. Подготовка персонала	Проводят обучение персонала и проверку его способности обеспечить функционирование АС.
	7.3. Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями)	Обеспечивают получение комплектующих изделий серийного и единичного производства, материалов и монтажных изделий. Проводят входной контроль их качества.
	7.4. Строительно-монтажные работы	Проводят: выполнение работ по строительству специализированных зданий (помещений) для размещения технических средств и персонала АС; сооружение кабельных каналов; выполнение работ по монтажу технических средств и линий связи; испытание смонтированных технических средств; сдачу технических средств для проведения пусконаладочных работ.

ИС. Стадии разработки ПО и ПД

- ГОСТ 34.601-90

Стадии	Этапы работ	Содержание работ
7. Ввод в действие	7.5. Пусконаладочные работы	Проводят автономную наладку технических и программных средств, загрузку информации в базу данных и проверку системы ее ведения; комплексную наладку всех средств системы.
	7.6. Проведение предварительных испытаний	Осуществляют: - испытания АС на работоспособность и соответствие техническому заданию в соответствии с программой и методикой предварительных испытаний; - устранение неисправностей и внесение изменений в документацию на АС, в т. ч. эксплуатационную в соответствии с протоколом испытаний; - оформление акта о приемке АС в опытную эксплуатацию.
	7.7. Проведение опытной эксплуатации	Проводят опытную эксплуатацию АС; анализ результатов опытной эксплуатации АС; доработку (при необходимости) программного обеспечения АС; дополнительную наладку (при необходимости) технических средств АС; оформление акта о завершении опытной эксплуатации.
	7.8. Проведение приемочных испытаний	проводят: - испытания на соответствие техническому заданию согласно программе и методике приемочных испытаний; - анализ результатов испытаний АС и устранение недостатков, выявленных при испытаниях; - оформление акта о приемке АС в постоянную эксплуатацию.

ИС. Стадии разработки ПО и ПД

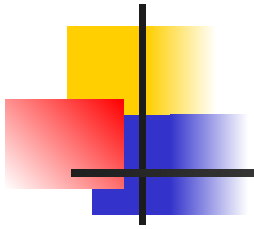
- ГОСТ 34.601-90

Стадии разработки	**Этапы работ**	**Содержание работ**
8. Сопровождение АС	8.1. Выполнение работ в соответствии с гарантийными обязательствами	Осуществляют работы по устранению недостатков, выявленных при эксплуатации АС в течение установленных гарантийных сроков, внесению необходимых изменений в документацию на АС.
	8.2. Послегарантийное обслуживание	Осуществляют работы по: - анализу функционирования системы; - выявлению отклонений фактических эксплуатационных характеристик АС от проектных значений; - установлению причин этих отклонений; - устранению выявленных недостатков и обеспечению стабильности эксплуатационных характеристик АС; - внесению необходимых изменений в документацию на АС.

Методологии разработки ПО

- (Agile) Гибкая методология разработки – Scrum, XP, …
  - работающий продукт в приоритете перед исчерпывающей документации
  - сотрудничество с заказчиком в приоритете перед условиями контракта
  - - рефакторинг!
  - - противоречия!
- (RAD) Быстрая разработка приложений
- (RUP) Методология от Rational Software
- …

ИС. Стадии разработки ПО и ПД



Схемы алгоритмов

Схемы алгоритмов

- **ГОСТ 19.701-90** Единая система программной документации. СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ ДАННЫХ И СИСТЕМ

Схемы алгоритмов

- 1.1. Схемы алгоритмов, программ, данных и систем (далее – схемы) состоят из имеющих заданное значение **символов**, краткого **пояснительного текста** и соединяющих **линий**.
- 1.2. Схемы могут использоваться на **различных уровнях детализации**, причем число уровней зависит от размеров и сложности задачи обработки данных. Уровень детализации должен быть таким, чтобы различные части и взаимосвязь между ними были **понятны в целом**.

- 1.4. В стандарте используются следующие понятия:
  - 1) **основной символ** - символ, используемый в тех случаях, когда точный тип (вид) процесса или носителя данных неизвестен или отсутствует необходимость в описании фактического носителя данных;
  - 2) **специфический символ** - символ, используемый в тех случаях, когда известен точный тип (вид) процесса или носителя данных или когда необходимо описать фактический носитель данных;
  - 3) **схема** - графическое представление определения, анализа или метода решения задачи, в, котором используются символы для отображения операций, данных, потока, оборудования и т.д.

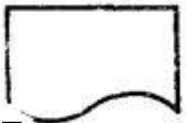
Схемы алгоритмов

- 2.2. Схема программы
  - 2.2.1. Схемы программ отображают последовательность операций в программе.
  - 2.2.2. Схема программы состоит из:
    - 1) символов процесса, указывающих фактические операции обработки данных (включая символы, определяющие путь, которого следует придерживаться с учетом логических условий);
    - 2) линейных символов, указывающих поток управления;
    - 3) специальных символов, используемых для облегчения написания и чтения схемы.

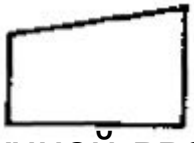
Схемы алгоритмов

- Основные символы

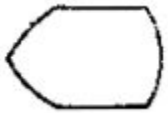
Данные (носитель не определен)



Документ (данные в удобочитаемой форме)



Ручной ввод

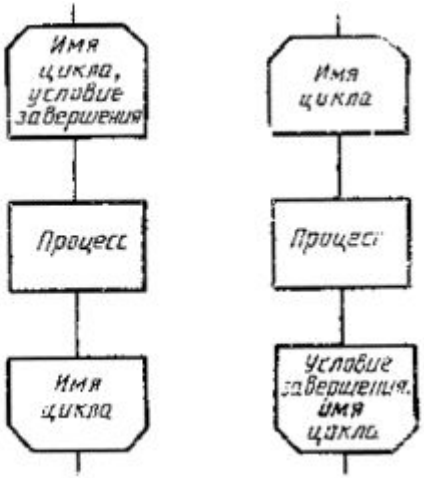


Дисплей

Бумажная лента

Процесс

Терминатор

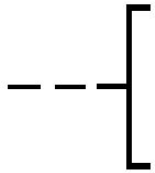


Цикл

Предопределенный процесс

Решение

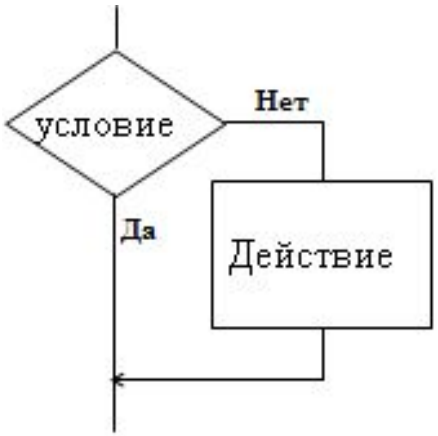
Соединитель

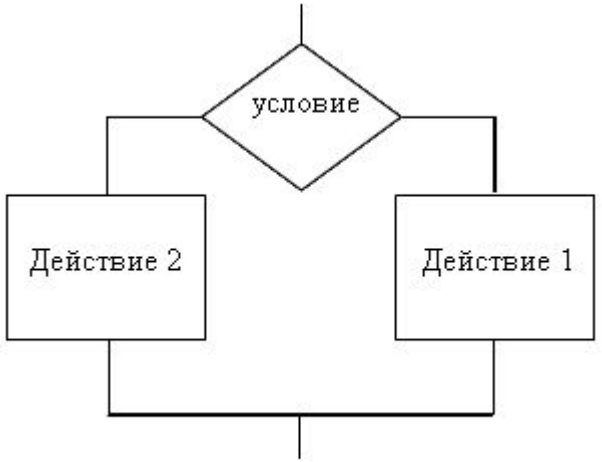


Комментарий

Схемы алгоритмов

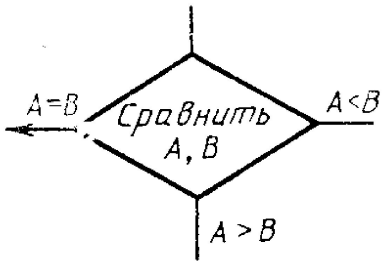
- Оператор «решение»

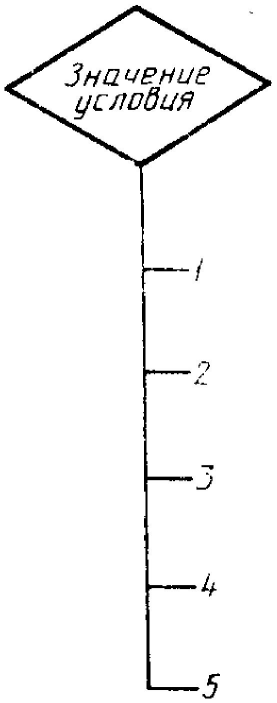




Схемы алгоритмов

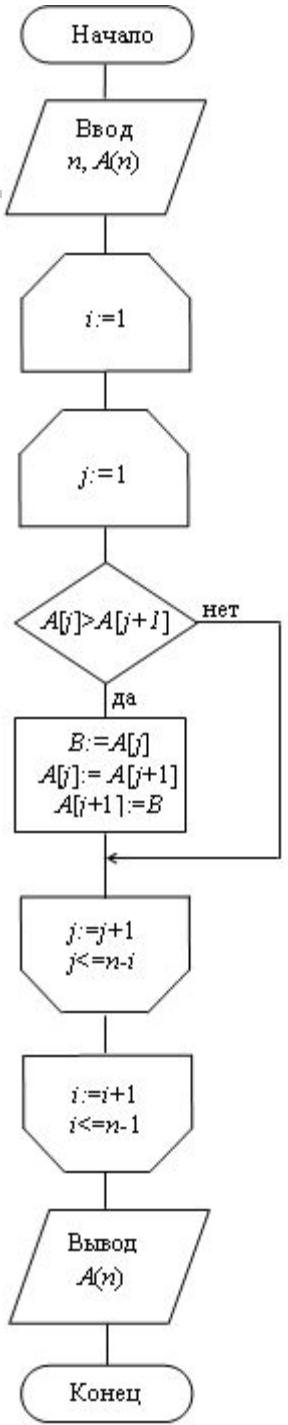
- Специальные условные обозначения
  - Каждый выход из символа должен сопровождаться соответствующими значениями условий, чтобы показать логический путь, который он представляет, с тем, чтобы эти условия и соответствующие ссылки были идентифицированы.





Схемы алгоритмов

```
{
    int n, a[100];
    cin>>n;
    for (int i=0; i<n; i++)
        cin>>a[i];
    for (int i=0; i<n; i++)
        for (int j=0; j<n; j++)
            if (a[j]>a[j+1])
            {
                int b=a[j];
                a[j]=a[j+1];
                a[j+1]=b;
            }
    for (int i=0; i<n; i++)
        cout<<a[i];
}
```



Схемы алгоритмов

- Ещё раз: Уровень детализации должен быть таким, чтобы различные части и взаимосвязь между ними были **понятны в целом**.

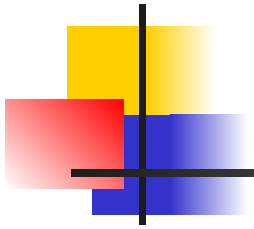
(Мартин Голдинг)

Пишите код так, как будто сопровождать его будет склонный к насилию психопат, который знает, где вы живете.

Стив Макконнелл. «Совершенный код»

В 1998 году читатели журнала «Software Development» признали Стива одним из трех наиболее влиятельных людей в отрасли разработки ПО наряду с Биллом Гейтсом и Линусом Торвальдсом.

Схемы алгоритмов



Массивы и списки

Массивы и списки

- Массив (**индексный массив**) – набор однотипных компонентов (элементов), расположенных в памяти непосредственно друг за другом, доступ к которым осуществляется по индексу (индексам). (*Вирт Н. Алгоритмы и структуры данных*).

- Размерность массива – количество индексов, необходимое для однозначного доступа к элементу массива.

Массивы и списки

- Массив – структура с произвольным доступом
  - A – начало массива
  - L – размер данных (элемента массива)
  - A[k] => A + L*k

Массивы и списки

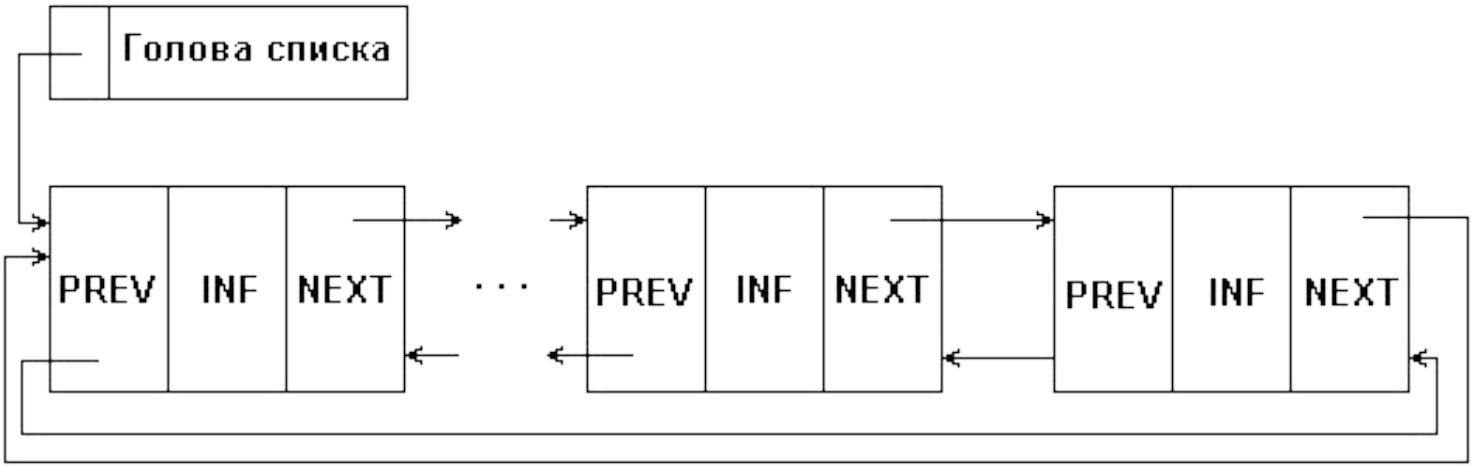
- Достоинства массивов:
  - лёгкость вычисления адреса элемента по его индексу
  - одинаковое время доступа ко всем элементам
  - малый размер элементов: они состоят только из информационного поля
- Недостатки массивов:
  - для статического массива — отсутствие динамики, невозможность удаления или добавления элемента без сдвига других
  - для динамического и/или гетерогенного массива — более низкое (по сравнению с обычным статическим) быстродействие и дополнительные накладные расходы на поддержку динамических свойств и/или гетерогенности.
  - при работе с массивом в стиле C (с указателями) и при отсутствии дополнительных средств контроля — угроза выхода за границы массива и повреждения данных

Массивы и списки

- Динамические массивы – массивы с возможностью изменения размера
  - 1. Выделить память нового размера
  - 2. Скопировать старые данные в новую область
  - 3. Объявить новую память «старым» массивом
  - 4. Освободить старую память
- Гетерогенные массивы – массивы с возможностью хранения разнотипных данных (реализовано не во всех ЯП)

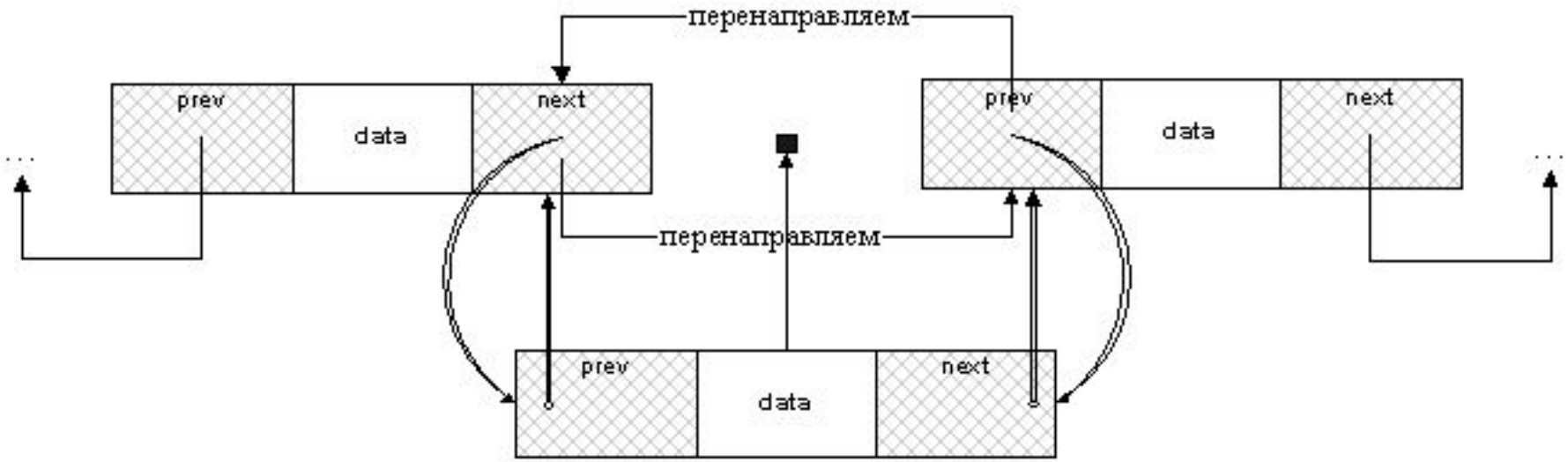
Массивы и списки

- Список – структура с последовательным доступом



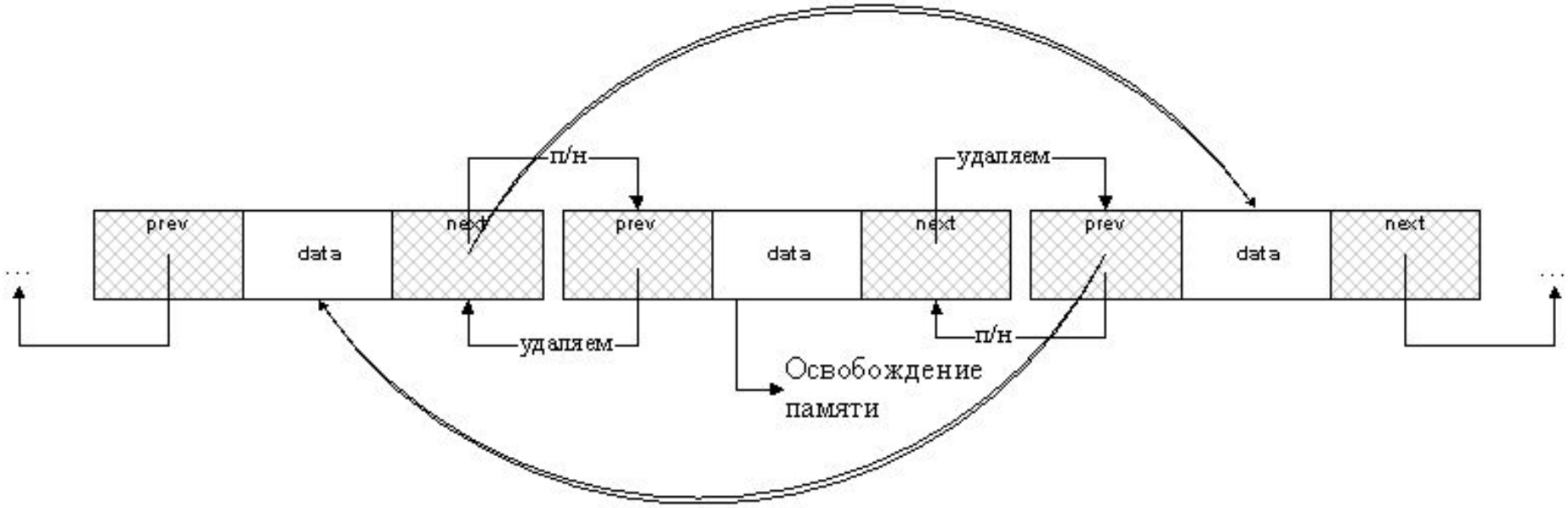
Массивы и списки

- Добавление элемента в середину списка



Массивы и списки

- Удаление элемента из середины списка



Массивы и списки

- Ассоциативный массив (**словарь**) — абстрактный тип данных, позволяющий хранить пары вида **(ключ, значение)** и поддерживающий операции insert, find, remove
- C++:

```
string name, phone;
map< string, string > book;
cin >> name >> phone;
book[ name ] = phone;
```

Массивы и списки

- Возвращаясь к динамическим спискам… Каким образом должен возрастать размер буфера?
- Начальные условия:
  - Изначальный размер – 1 байт
  - Буфер растёт по 1 байту до тех пор, пока не достигнет размера 1 МиБ.
- Каков суммарный объём памяти был задействован?

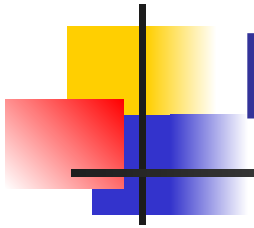
1 + 2 + 3 + … + 1,048,575 + 1,048,576 =
549,756,338,176 байт = ***512 ГБайт***

Массивы и списки

- Экспоненциальный рост:
  - Коэф. = 1.5

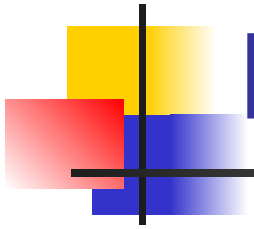
    1 + 2 + 3 + 5 + 8 + 12 + 18 + 27 + … +
    466608 + 699912 + 1049868 =
    3 149 587 байт = ***3 МБайт***
  - Коэф. = 2

    1 + 2 + 4 + 8 + 16 + 32 + … +
    262144 + 524288 + 1048576 =
    2 097 151 байт = ***2 МБайт***



Массивы и списки

- Проблема линейного роста – в большом количестве выделяемой памяти
- Общая проблема роста – кусочно разбросанные остающиеся области памяти



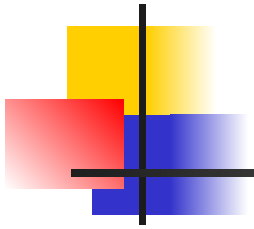
Массивы и списки

99 маленьких багов в коде,

99 маленьких багов в коде,

Один нашли, пофиксили,

127 маленьких багов в коде…

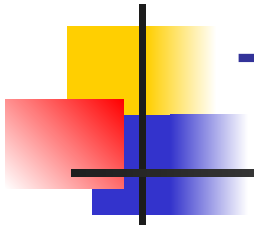


Тестирование и отладка программы

или

Базовые принципы работы начинающих пре-альфа-программистов

Тестирование и отладка программ

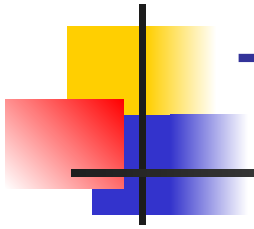


Тестирование и отладка программ

Аксиома 1

- **Тестирование проводится для того, чтобы найти ошибки, а не показать работоспособность программы**

- Хорош тот тест, для которого высока вероятность обнаружить ошибку, а не тот, который демонстрирует правильную работу программы
- Тестирование может доказать, что дефекты в программном обеспечении существуют, но если дефектов не найдено, это не дает гарантии, что их нет.

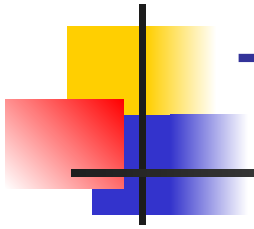


Тестирование и отладка программ

Аксиома 2

- **Наилучшее решение проблемы надежности – не допускать ошибок в программе**

- Роль тестирования – определить местонахождение немногочисленных ошибок, оставшихся в хорошо спроектированной программе.
- Попытки с помощью тестирования достичь надежности плохо спроектированной программы безнадежны.



Тестирование и отладка программ

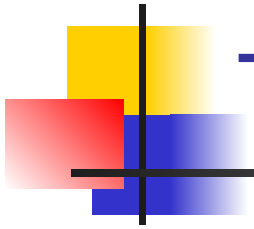
Аксиома 3

- **Совершенное тестирование невозможно**

- Сколько входных данных нужно перебрать для программы (x, y, z – integer)

$$z = x + y$$

чтобы быть уверенным, что она работает правильно?

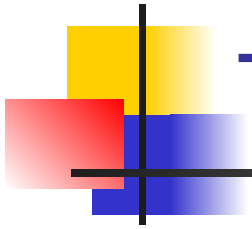


Тестирование и отладка программ

Хорошая привычка

- **Тестирование программы должен производить не автор**

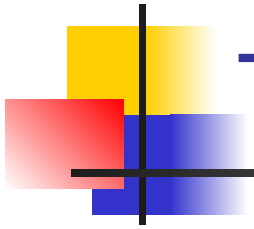
- Простейшие тесты на начальном этапе – автор, далее – человек, не знакомый с задачей
- У автора глаза «зашорены»



Тестирование и отладка программ

Хорошая привычка

- **Подготовка исходных данных и результатов ДО запуска программы**
- Эффект «подгонки» результатов

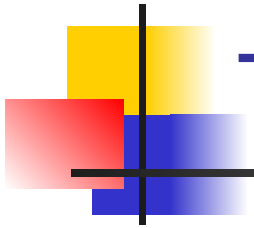


Тестирование и отладка программ

Хорошая привычка

- **Подготовка тестов для правильных и для неправильных данных**

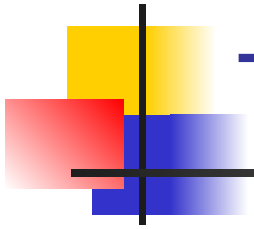
- Программа должна работать всегда!
- Сообщения ОС об ошибках программы – недопустимы



Тестирование и отладка программ

Хорошая привычка

- **Не изменять программу для облегчения тестирования**
- А вдруг уберёте ошибку?

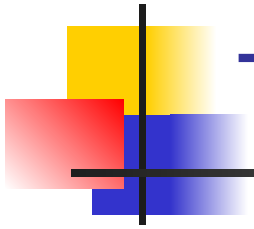


Тестирование и отладка программ

Хорошая привычка

- **Заблаговременное тестирование**

- 1 тестирование (в конце) – 50 ошибок
- 20 тестирований (в процессе) – по 2 ошибки

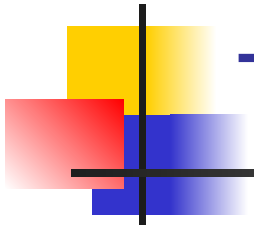


Тестирование и отладка программ

Хорошая привычка

- **Регрессионное тестирование**

- Накопление ошибок
- При доработке программы возможен «возврат ошибок»

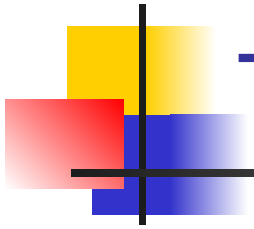


Тестирование и отладка программ

Хорошая привычка

- **Парадокс пестицида**

- Если один и тот же тестовый модуль многократно применять к той же системе, он в конечном счете перестанет находить ошибки.
- Тестовый модуль должен постоянно и систематически корректироваться, а новые тесты должны охватывать все составляющие программного обеспечения



Тестирование и отладка программ

Хорошая привычка

- **Случайное тестирование**

- Много случайных данных иногда позволяют найти ошибки, которые не охватываются «логичными» тестами

Тестирование и отладка программ

Как это на практике?

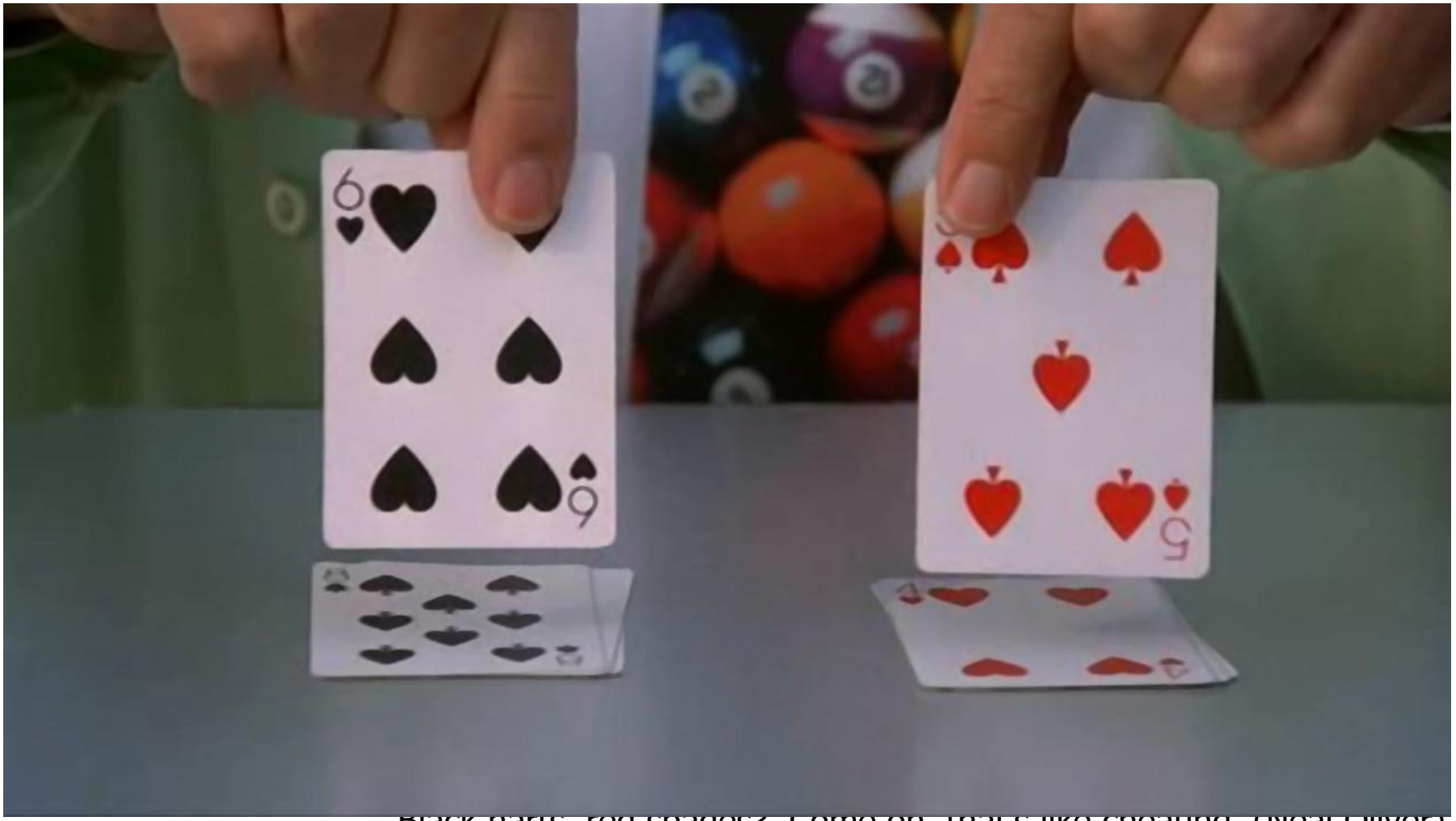
- Тестирование «один из группы»
  - Положительные, отрицательные, нулевые, различные пары…
- Тестирование граничных условий
  - 2я лр – какое последнее слагаемое?
- Массивы
  - все, ни одного, разные
  - выход за границы массива
- Циклы
  - Ни разу, один раз, максимум, промежуточное количество
- Тестирование ветвей кода
- Черный и белый ящик (+серый ящик)
- Тестирование особых случаев («13й этаж»)
- Случайное тестирование
- Регрессионное тестирование

Тестирование и отладка программ

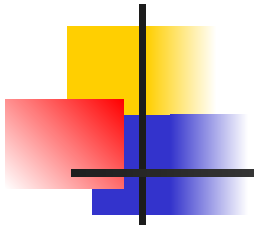
- Ситуации «за гранью добра и зла»

```
-- этот код работает! (SQL)
IF 1 = 0
BEGIN
    SET FMTONLY OFF
END
```

- Но это уже совсем другая история...



Black harts, red spades?. Come on, that's like cheating. (Neal Oliver)



Простейшие сортировки

Простейшие сортировки

- Сортировка пузырьком (простыми обменами)

```
for (int i = 0; i < N - 1; i++)
    for (int j = 0; j < N - 1; j++)
     if (a[j] > a[j + 1])
          swap(a[j], a[j + 1]);

for (int i = 0; i < N - 1; i++)
    for (int j = 0; j < N - i - 1; j++)
     if (a[j] > a[j + 1])
          swap(a[j], a[j + 1]);
```

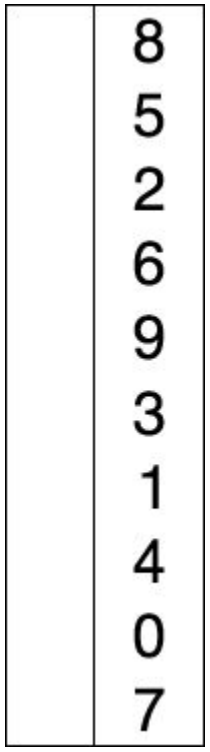
Простейшие сортировки

- Шейкерная сортировка
  - модификация сортировки пузырьком:
    - движение слева направо
    - движение справа налево

- Сортировка «расчёской»
  - достаточно большое расстояние между сравниваемыми элементами
  - фактор уменьшения $\frac{1}{1-\frac{1}{e^{\varphi}}} \approx 1{,}247330950103979$

Простейшие сортировки

- Сортировка выбором
  - находим номер минимального значения в текущем списке
  - производим обмен этого значения со значением первой не отсортированной позиции (обмен не нужен, если минимальный элемент уже находится на данной позиции)
  - теперь сортируем хвост списка, исключив из рассмотрения уже отсортированные элементы



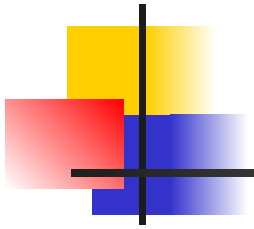
Простейшие сортировки

- Сортировка вставками
  - выбираем текущий элемент
  - находим для него место в отсортированной части, сдвигая элементы вправо
  - вставляем на новое место
  - переходим к следующему элементу

6 5 3 1 8 7 2 4

Простейшие сортировки

https://habrahabr.ru/company/wunderfund/blog/277143/



Системы счисления

Системы счисления

- Непозиционные
  - Единичная
  - Алфавитные
  - Древнеегипетская
  - Римская
- Позиционные
  - Двоичная
  - Десятичная
  - Восьмеричная
  - …

Системы счисления

В непозиционных системах счисления значение (величина) числа определяется как сумма или разность цифр в числе.

- **MCM**LXXX**VIII** = 1000+(1000-100)+(50+10+10+10)+5+1+1+1 = 1988

- **Недостатки непозиционных систем счисления**
  - Существует постоянная потребность введения новых знаков для записи больших чисел.
  - Невозможно представлять дробные и отрицательные числа.
  - Сложно выполнять арифметические операции, т.к. не существует алгоритмов их выполнения

Системы счисления

- В **позиционных** системах счисления значение цифры зависит от ее места (позиции) в числе, а в непозиционных не зависит.
- В позиционной системе счисления один и тот же числовой символ приобретает различные значения (имеет различный вес) в зависимости от позиции.
- Каждая позиция соответствует определенной степени основания системы счисления. **Основание** определяет, во сколько раз отличаются значения одинаковых цифр, стоящих в соседних позициях
- **Достоинства позиционных систем счисления**
  - Простота выполнения арифметических операций
  - Ограниченное количество символов (цифр) для записи любых чисел
- **Алфавит СС** – набор цифр, доступных для использования в данной СС, например, **7: 0..6**
- **Основание СС** – мощность алфавита СС

Системы счисления

- Перевод чисел в 10ю СС:
  - Пронумеровать разряды справа налево, начиная с 0
  - Вычислить вес каждого разряда, возведя основание в степень номера разряда
  - Для каждого разряда найти произведение цифры в нём на его вес
  - Найти сумму произведений

$$\underset{1000\ \ 100\ \ 10\ \ 1}{\overset{3\ \ 2\ \ 1\ \ 0}{2863}}_{10} = 2\cdot 1000 + 8\cdot 100 + 6\cdot 10 + 3\cdot 1 = 2863_{10}$$

$$\underset{343\ \ 49\ \ 7\ \ 1}{\overset{3\ \ 2\ \ 1\ \ 0}{2463}}_{7} = 2\cdot 343 + 4\cdot 49 + 6\cdot 7 + 3\cdot 1 = 927_{10}$$

Системы счисления

$$\underset{196}{5}\overset{2}{\underset{14}{B}}\overset{1}{\underset{1}{7}}\overset{0}{}_{14} = 5\cdot 196 + 11\cdot 14 + 7\cdot 1 = 1141_{10}$$

$$\underset{144}{7}\overset{2}{\underset{14}{B}}\overset{1}{\underset{1}{1}}\overset{0}{}_{12} = 7\cdot 144 + 11\cdot 12 + 1\cdot 1 = 1141_{10}$$

$$\underset{729}{1}\overset{3}{\underset{81}{5}}\overset{2}{\underset{9}{0}}\overset{1}{\underset{1}{7}}\overset{0}{}_{9} = 1\cdot 729 + 5\cdot 81 + 0\cdot 9 + 7\cdot 1 = 1141_{10}$$

$$1507_9 = 1141_{10} = 7B1_{12} = 5B7_{14} = 511_{15}$$

Системы счисления

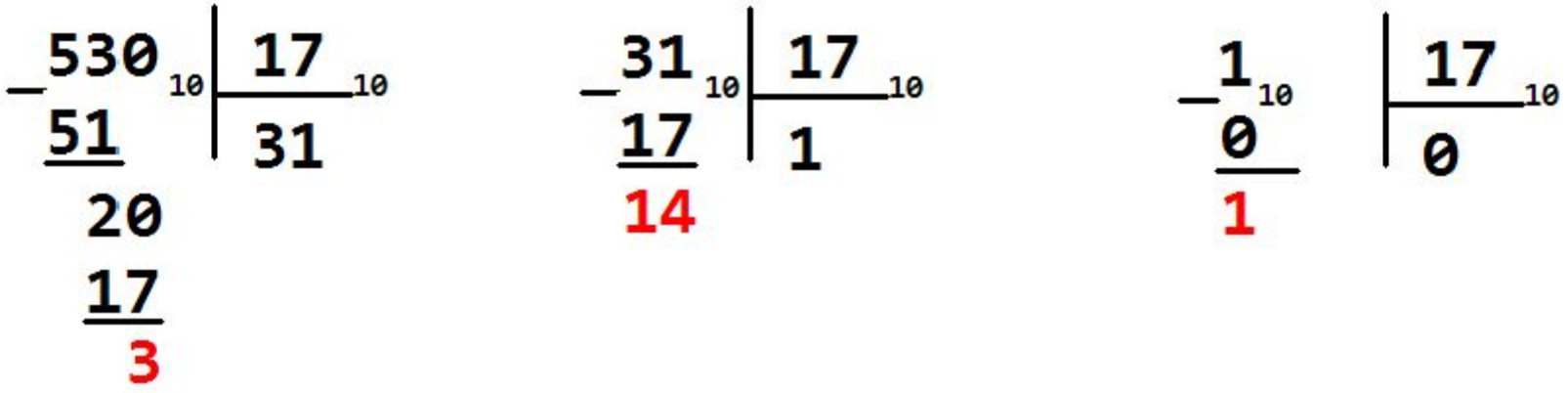
- Перевод из 10й СС:
  - Деление *исходного числа* нацело с остатком на основание целевой СС
  - Деление *полученного частного* нацело с остатком на основание целевой СС
  - Деление продолжается до получения в частном значения 0
  - Составление из остатков (в обратном порядке) числа в целевой СС

Системы счисления

$$
\begin{array}{r|l}
530_{10} & 12_{10} \\
\underline{48} & 44 \\
50 & \\
\underline{48} & \\
2 &
\end{array}
\qquad
\begin{array}{r|l}
44_{10} & 12_{10} \\
\underline{36} & 3 \\
8 &
\end{array}
\qquad
\begin{array}{r|l}
3_{10} & 12_{10} \\
\underline{0} & 0 \\
3 &
\end{array}
$$

- Ответ: 382_{12}

Системы счисления



- Ответ: $1E3_{17}$

Системы счисления

- Прямой перевод из одной СС в другую (X->Y)
  - Возможен только в случае, если $X=Y^n$ или $X^n=Y$
- $10\ 111\ 010\ 011_2=2723_8$ (n=3)

010	111	010	011
2	7	2	3

- $487032_9=11\ 22\ 21\ 00\ 10\ 02_3$ (n=2)

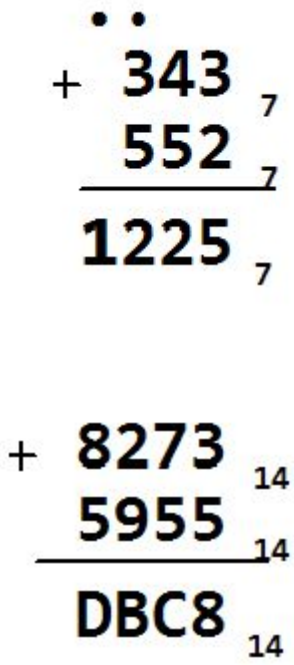
4	8	7	0	3	2
11	22	21	00	10	02

Системы счисления

- Двойной прямой перевод из одной СС в другую (X->Y->Z)
  - Возможен только в случае, если:
    - и $X=Y^n$ или $X^n=Y$
    - и $Y=Z^n$ или $Z^n=Y$
- В остальных случаях перевод X->10->Z

Системы счисления

- Арифметические операции в различных СС
  - При сложении (умножении) необходимо учитывать, получается ли в результате цифра или число:
    - 3+2 = 5 – это цифра в 7-ричной СС
    - 4+5 = 9 – это число в 7-ричной СС
    - 9:7 = 1 (остаток 2)
      - 2 – остаток, пишется в текущий разряд
      - 1 – частное, переносится в старший разряд



Системы счисления

- Арифметические операции в различных СС
  - При вычитании необходимо учитывать, что при займе «1» в более старшем разряде в младший попадает значение, совпадающее с основанием СС

$$\begin{array}{r} 59462_{14} \\ -\ 23723_{14} \\ \hline 35B3D_{14} \end{array}$$

5	9–*1*	4*+14*	6–*1*	2*+14*
2	3	7	2	3
3	5	11	3	13

Системы счисления

- Уравновешенная троичная СС
  - «Знак числа» отсутствует

Положительные десятичные числа	Положительные троичные УВ числа	Отрицательные троичные УВ числа	Отрицательные десятичные числа
1	1	$\bar{1}$	–1
2	1 $\bar{1}$	$\bar{1}$ 1	–2
3	1 0	$\bar{1}$ 0	–3
4	1 1	$\bar{1}$ $\bar{1}$	–4
5	1 $\bar{1}$ $\bar{1}$	$\bar{1}$ 1 1	–5
6	1 $\bar{1}$ 0	$\bar{1}$ 1 0	–6
7	1 $\bar{1}$ 1	$\bar{1}$ 1 $\bar{1}$	–7
8	1 0 $\bar{1}$	$\bar{1}$ 0 1	–8
9	1 0 0	$\bar{1}$ 0 0	–9
10	1 0 1	$\bar{1}$ 0 $\bar{1}$	–10
11	1 1 $\bar{1}$	$\bar{1}$ $\bar{1}$ 1	–11
12	1 1 0	$\bar{1}$ $\bar{1}$ 0	–12
13	1 1 1	$\bar{1}$ $\bar{1}$ $\bar{1}$	–13
14	1 $\bar{1}$ $\bar{1}$ $\bar{1}$	$\bar{1}$ 1 1 1	–14

Системы счисления

- Благодаря тому что основание 3 нечётно, в троичной системе возможно симметричное относительно нуля расположение цифр: −1, 0, 1. Свойства:
  - Естественность представления отрицательных чисел;
  - Отсутствие проблемы округления: обнуление ненужных младших разрядов *округляет* — приближает число к ближайшему «грубому».
  - Для изменения знака представляемого числа нужно изменить ненулевые цифры на симметричные.
  - При суммировании большого количества чисел значение для переноса в следующий разряд растёт с увеличением количества слагаемых не линейно, а пропорционально квадратному корню числа слагаемых.
  - По затратам количества знаков на представление чисел она равна троичной несимметричной системе.

Уникальная Сетунь на основе троичного кода
http://habrahabr.ru/company/ua-hosting/blog/273929/

Системы счисления

- Примеры выполнения операций в уравновешенной троичной СС

$$
\begin{array}{rcr}
 & 38_{10} & = \quad 111\bar{1} \\
+ & 24_{10} & = \quad 10\bar{1}0 \\
\hline
 & 62_{10} & = 1\bar{1}10\bar{1}
\end{array}
\qquad
\begin{array}{rcr}
 & \dot{3}\dot{8}_{10} & = \dot{1}\dot{1}1\bar{1} \\
+ & -24_{10} & = \bar{1}010 \\
\hline
 & 14_{10} & = 1\bar{1}\bar{1}\bar{1}
\end{array}
$$

Системы счисления

- Фибоначчиева система счисления
  - Алфавит – цифры 0 и 1
  - Базис (веса разрядов) – последовательность чисел Фибоначчи: 1, 2, 3, 5, 8, 13, 21, 34, …
  - Преимущество кодов Фибоначчи для практики – в их «естественной» избыточности, которая может быть использована для целей контроля числовых преобразований.
  - Избыточность проявляет себя в свойстве множественности представлений одного и того же числа.

Системы счисления

- Разные представления:
  - операция свертки 011 → 100
  - операция развертки 100 → 011
- $32_{10} = 21*1 + 13*0 + 8*1 + 5*0 + 3*1 + 2*0 + 1*0$
  - 1010100_{fib} - ***минимальная форма***, в которой рядом не встречаются две единицы
  - 1010011_{fib}
  - 1001111_{fib}
  - 0111111_{fib} – ***максимальная (развернутая) форма***, в которой рядом не встречаются два нуля

Системы счисления

- Примеры выполнения операций в ФСС:

$$
\begin{array}{r}
10001000 \\
+\ 01001000 \\
\hline
1100\mathbf{2}000 \\
1100\mathbf{1}\mathbf{1}10 \\
11010010
\end{array}
$$

Тук
Тук
Тук тук
Тук тук тук
Тук тук тук тук тук

— Кто там?
— Фибоначчи

Системы счисления

- Вещественная часть числа

$$\begin{array}{cccccccc} 2 & 1 & 0 & -1 & -2 & -3 & & \\ 2 & 0 & 1 . & 0 & 1 & 3 & & 4 \\ 16 & 4 & 1 & \frac{1}{4} & \frac{1}{16} & \frac{1}{64} & & \end{array}$$

$$2 \cdot 16 + 0 \cdot 4 + 1 \cdot 1 + 0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{16} + 3 \cdot \frac{1}{64} = 33.109375_{10}$$

Системы счисления

- Вещественная часть числа

$$\begin{array}{ccccccc} 2 & 1 & 0 & -1 & -2 & & \\ 3 & A & 5 & . & 1 & 1 & _{12} \\ 144 & 12 & 1 & & \frac{1}{12} & \frac{1}{144} & \end{array}$$

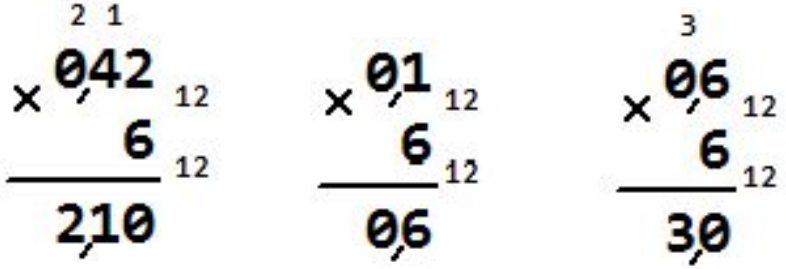
$$3 \cdot 144 + 10 \cdot 12 + 5 \cdot 1 + 1 \cdot \frac{1}{12} + 1 \cdot \frac{1}{144} = 557.0902(7)_{10}$$

- Результат – бесконечная периодическая дробь
- Округление для дальнейших действий **недопустимо**

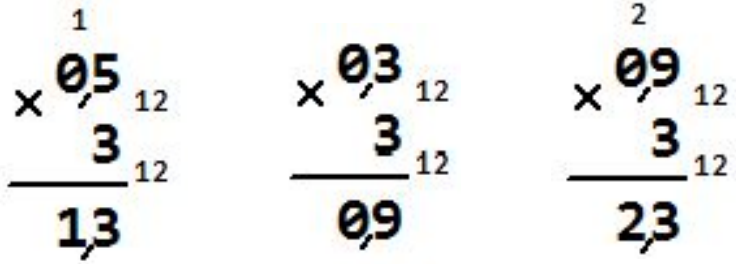
Системы счисления

Общий алгоритм перевода

- $0.42_{12} = 0.203_{6}$



- $0.5_{12} = 0.1(02)_{3}$



Системы счисления

- Перевод X->Y при Y = k*X, где k - целое

$$0.31_7 = \frac{3}{7} + \frac{1}{7 \cdot 7} = \frac{3 \cdot 2}{7 \cdot 2} + \frac{2 \cdot 1 \cdot 2}{(2 \cdot 7) \cdot (7 \cdot 2)} = \frac{6}{14} + \frac{4}{14 \cdot 14} = 0.64_{14}$$

$$0.43_5 = \frac{4}{5} + \frac{3}{5 \cdot 5} = \frac{4 \cdot 3}{5 \cdot 3} + \frac{3 \cdot 3 \cdot 3}{(3 \cdot 5) \cdot (5 \cdot 3)} =$$

$$= \frac{12}{15} + \frac{27}{15 \cdot 15} = \frac{12}{15} + \frac{15+12}{15 \cdot 15} = \frac{12}{15} + \frac{15}{15 \cdot 15} + \frac{12}{15 \cdot 15} =$$

$$= \frac{12}{15} + \frac{1}{15} + \frac{12}{15 \cdot 15} = \frac{13}{15} + \frac{12}{15 \cdot 15} = 0.\mathrm{DC}_{15}$$

Системы счисления

- Прямой перевод из одной СС в другую (X->Y)
  - Возможен только в случае, если $X=Y^n$ или $X^n=Y$
- $0.101\ 110\ 100\ 110_2 = 0.5646_8$ (n=3)

101	110	100	110
5	6	4	6

- $0.487032_9 = 0.11\ 22\ 21\ 00\ 10\ 02_3$ (n=2)

4	8	7	0	3	2
11	22	21	00	10	02

Единицы измерения информации

http://www.absoluteastronomy.com/topics/Binary_prefix

Информатика – единственная наука, в которой **объём** называется **весом** и измеряется в **метрах**

Автор неизвестен

Единицы измерения информации

- 1 бит (1 б) – неделимая единица
- 1 байт (1 Б) = 8 битов

Всё просто?

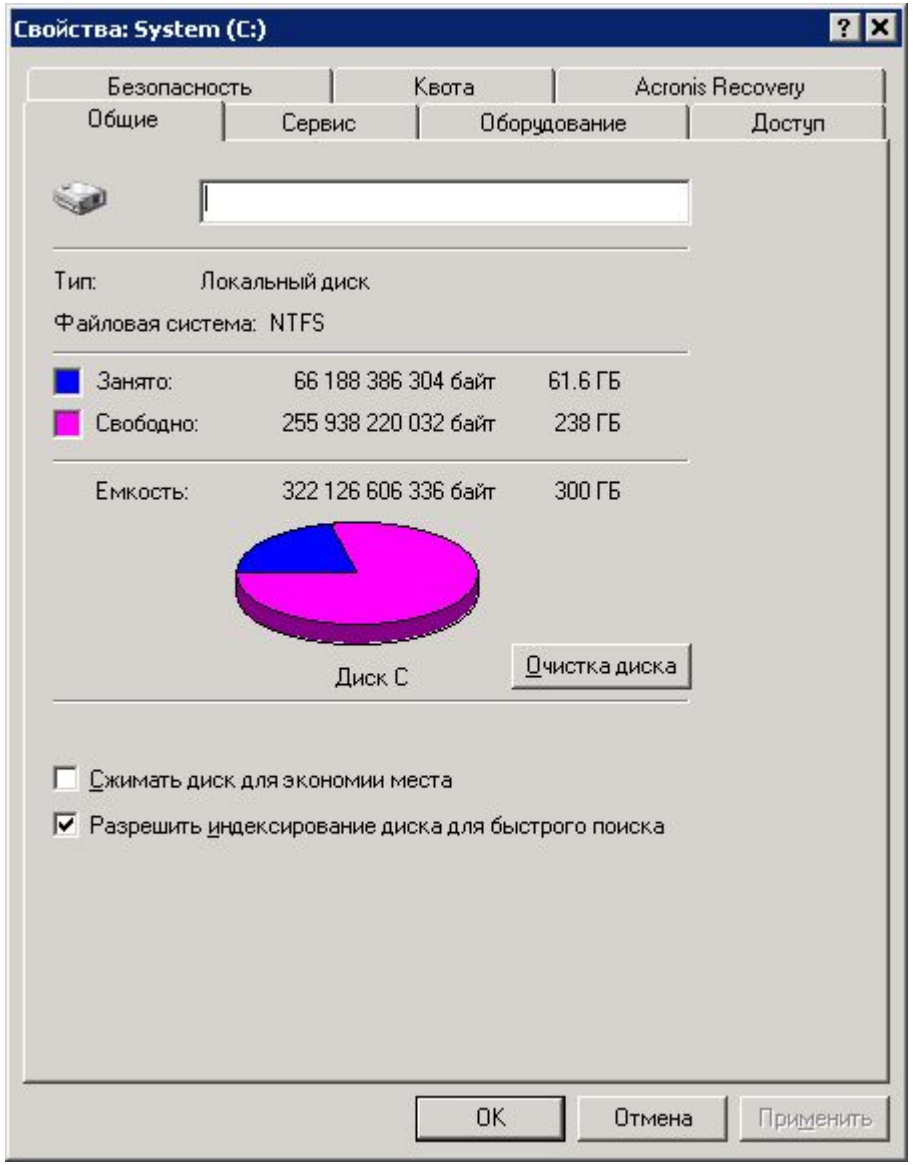
- 1 Кбайт (1 КБ) = 1024 Б
- 1 Мбайт (1 МБ) = 1024 КБ
- …

Всё просто?

Единицы измерения информации

- 66 188 386 304 Б
- 64 637 096 КБ
- 63 122.16 МБ
- 61.642 ГБ

Всё просто?

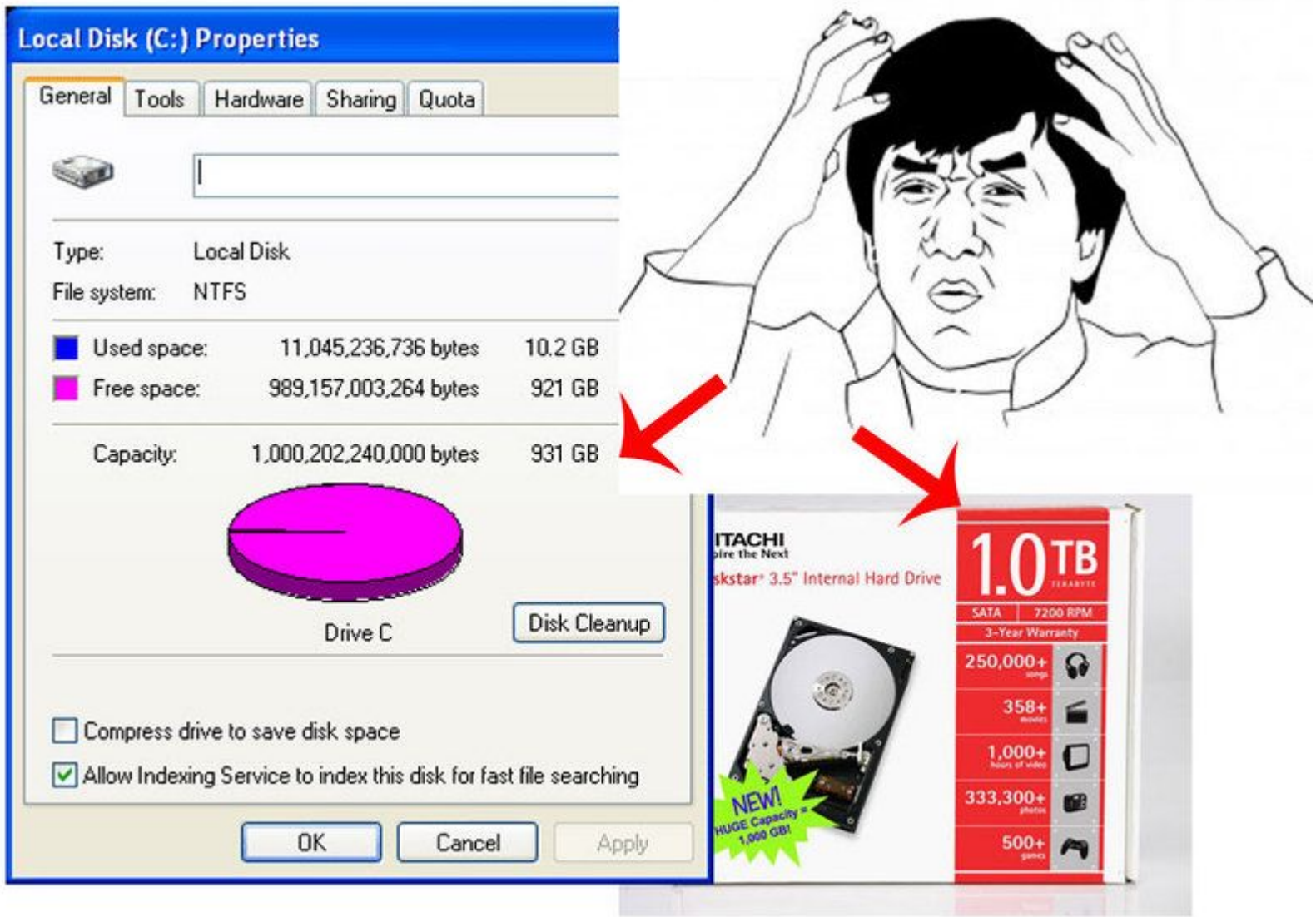


Единицы измерения информации

Говорил или не говорил – теперь уже не важно

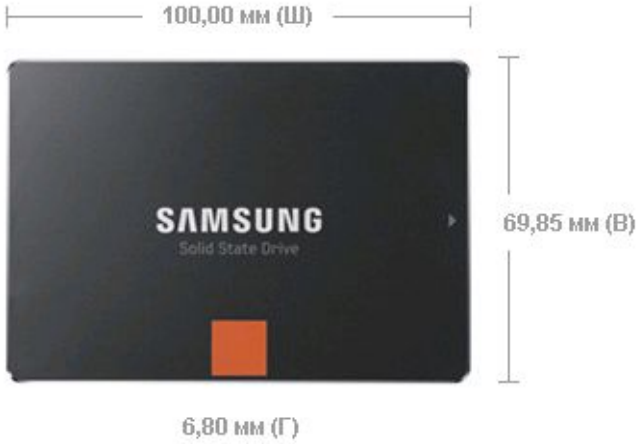
http://imranontech.com/2007/02/20/did-bill-gates-say-the-640k-line/

Единицы измерения информации



Единицы измерения информации

- Оперативная память (проводники!):
  - 512 MB = 512 * 1024 * 1024 байтов
- Жесткие диски:



Применение
* Для клиентских ПК и ПК предприятий категории 1 (напр., серверов), требуется использование технологии over-provisioning (OP, Технология Over-Provisioning резервирует область твердотельного накопителя и делает ее недоступной пользователю) на уровне минимум 6,7%, что устанавливается с помощью ПО Magician.

Емкость
512 ГБ (1 ГБ=1 миллиард байт IDEMA)
*Фактически используемая емкость может быть меньше (в результате форматирования, использования операционной системой, приложениями и др.)

Единицы измерения информации

Flash drives

USB flash drives, flash-based memory cards like CompactFlash or Secure Digital, and flash-based SSDs use SI prefixes;

for example, a "256 MB" flash card provides at least 256 million bytes , not 256×1024×1024 .

These devices usually physically contain the binary capacities, but some space is reserved for internal functions of the flash drive. In other words, there are physically 256×1024×1024 bytes of storage on a typical "256MB" flash drive, but some space is needed for functions like wear leveling. In the case of a "256MB" flash drive, the manufacturer can allocate approximately 12MB to internal functions, and still provide 256 million usable bytes.

Единицы измерения информации

- DVD:
  - 4.7 GB = 4.7 * 1000 * 1000 * 1000
- CD:
  - 700 MB = 700 * 1024 * 1024
- Floppy:
  - 1.44 MB = 1.44 * 1000 * 1024

Oh! That's the biggest whopper of all.
(mr. Cody)

Единицы измерения информации

- IEC 60027-2 (1999):
- Киби, Меби, …?

During its 12th Meeting in 1996, the Comité Consultatif des Unités (the Consultative Committee for Units, CCU) discussed the widespread misuse of prefixes of the International System of Units (SI) to denote powers of two [1]. For example, the prefix kilo (symbol k) is often used in information and computer technology to denote the factor $2^{10} = 1024 \neq 10^3 = 1000$. The discussion at the CCU led to the inclusion in the 7th edition of the SI brochure [2] of a statement that the SI prefixes, which denote exact powers of ten, should not be used to denote exact powers of two. The Comité International des Poids et Mesures (CIPM) also adopted a Recommendation [3] in which the International Electrotechnical Commission (IEC) was encouraged to develop prefixes for binary multiples – the IEC and its Technical Committee IEC/TC 25, Quantities, units, and their letter symbols, being responsible for standardization of quantities and units in the field of information science and technology.

IEC/TC 25 has now completed this task and the IEC has published an International Standard, IEC 60027-2, for the new prefixes for binary multiples [4].

These prefixes are as follows:

Factor	Name	Symbol	Origin	Derived from
2^{10}	kibi	Ki	kilobinary $(2^{10})^1$	kilo $(10^3)^1$
2^{20}	mebi	Mi	megabinary $(2^{10})^2$	mega $(10^3)^2$
2^{30}	gibi	Gi	gigabinary $(2^{10})^3$	giga $(10^3)^3$
2^{40}	tebi	Ti	terabinary $(2^{10})^4$	tera $(10^3)^4$
2^{50}	pebi	Pi	petabinary $(2^{10})^5$	peta $(10^3)^5$
2^{60}	exbi	Ei	exabinary $(2^{10})^6$	exa $(10^3)^6$

Examples
one kibibit: 1 Kibit = 2^{10} bit
one kilobit: 1 kbit = 10^3 bit
one mebibyte: 1 MiB = 2^{20} B
one megabyte: 1 MB = 10^6 B

Of course the new prefixes can also be used outside information and computer technology. It should be noted, however, that these prefixes are not part of the SI.

References

1. *BIPM Com. Cons. Unités*, 1996, **12**, 68 p.
2. *Le Système International d'Unités (SI)*, 7th ed., Sèvres, Bureau International des Poids et Mesures, 1998.
3. Recommendation 2 (CI-1996), *BIPM Proc.-Verb. Com. Int. Poids et Mesures*, 1996, **64**, 146; Quinn T. J., *Metrologia*, 1997, **34**, 189.
4. International Standard IEC 60027-2, Letter symbols to be used in electrical technology – Part 2: Telecommunications and electronics, Amendment 2, Geneva, International Electrotechnical Commission, 1999.

Единицы измерения информации

- ГОСТ 8.417-2002:

ПРИЛОЖЕНИЕ А
(справочное)

Единицы количества информации

Таблица А.1

Наименование величины	Единица				Примечание
	Наименование	Обозначение		Значение	
		международное	русское		
Количество информации[1]	бит[2] байт[2], [3]	bit B (byte)	бит Б (байт)	1 1 Б = 8 бит	Единица информации в двоичной системе счисления (двоичная единица информации)

[1] Термин «количество информации» используют в устройствах цифровой обработки и передачи информации, например в цифровой вычислительной технике (компьютерах), для записи объема запоминающих устройств, количества памяти, используемой компьютерной программой.

[2] В соответствии с международным стандартом МЭК 60027-2 единицы «бит» и «байт» применяют с приставками СИ (таблица 8 и раздел 7) [7].

[3] Исторически сложилась такая ситуация, что с наименованием «байт» некорректно (вместо $1000 = 10^3$ принято $1024 = 2^{10}$) использовали (и используют) приставки СИ: 1 Кбайт = 1024 байт, 1 Мбайт = 1024 Кбайт, 1 Гбайт = 1024 Мбайт и т. д. При этом обозначение Кбайт начинают с прописной буквы в отличие от строчной буквы «к» для обозначения множителя 10^3.

- 1 кБ = 1000 Б,
- 1 КБ = 1024 Б (неофициально)

Всё просто?

Единицы измерения информации

- Постановление Правительства РФ №879 от 31.10.2009 «Об утверждении положения о единицах величин, допускаемых к применению в РФ» (с изм. от 15.08.2015):

6. Наименование и обозначение единицы количества информации "байт" (1 байт = 8 бит) применяются с двоичными приставками "Кило", "Мега", "Гига", которые соответствуют множителям 2^{10}, 2^{20} и 2^{30} (1 Кбайт = 1024 байт, 1 Мбайт = 1024 Кбайт, 1 Гбайт = 1024 Мбайт). Данные приставки пишутся с большой буквы. Допускается применение международного обозначения единицы информации с приставками "K" "M" "G", рекомендованного Международным стандартом Международной электротехнической комиссии МЭК 60027-2 (KB, MB, GB, Kbyte, Mbyte, Gbyte).

Единицы измерения информации

Property	Default base	Default measures
RAM	1024	MiB, GiB
Хранилища данных (HDD, SSD, Cloud, USB-Sticks, DVD, BD)	1000	GB, TB
Хранилища данных (CD)	1024	MiB
Размеры файлов	1024	KiB, MiB, GiB
Такт процессора, частота э/м волн	1000	MHz, GHz
Число точек	1000	MPixel
Скорость передачи данных	1000	Mbit/s, Gbit/s

Единицы измерения информации

СЕЙЧАС ОЧЕНЬ МНОГО НЕРАЗБЕРИХИ ИЗ-ЗА 1024 vs 1000, КБАЙТ vs КБИТ, И ЗАГЛАВНЫХ БУКВ В КАЖДОМ.

И ВОТ, НАКОНЕЦ, ПОЯВИЛСЯ ОДИН ОБЩИЙ ДЛЯ ВСЕХ СТАНДАРТ:

СИМВОЛ	НАЗВАНИЕ	РАЗМЕР	ОПИСАНИЕ
кБ	КИЛОБАЙТ	1024 БАЙТА ИЛИ 1000 БАЙТ	1000 БАЙТ В ВИСОКОСНЫЙ ГОД И 1024 В ОСТАЛЬНЫЕ
КБ	СТАНДАРТ КЕЛЛИ-БУТЛА	1012 БАЙТ	КОМПРОМИСС МЕЖДУ 1000 И 1024 БАЙТАМИ
КiБ	МНИМЫЙ КИЛОБАЙТ	$1024\sqrt{-1}$ БАЙТ	ИСПОЛЬЗУЕТСЯ В КВАНТОВЫХ КОМПЬЮТЕРАХ
кб	КИЛОБАЙТ INTEL	1023,936528 БАЙТ	РАССЧИТАНО НА PENTIUM F.P.U.
Кб	КИЛОБАЙТ НА НЖМД	СЕЙЧАС ЭТО 908 БАЙТ	СОКРАЩАЕТСЯ НА Ч БАЙТА КАЖДЫЙ ГОД. МАРКЕТИНГ...
КБА	БАЛЬШОЙ КИЛОБАЙТ	1152 БАЙТА	9 БИТ В БАЙТЕ, ВЕДЬ ВЫ НАШ ЛУЧШИЙ ПОКУПАТЕЛЬ

Представление целых чисел в памяти ЭВМ

Представление целых чисел

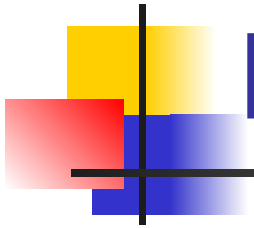
- Под каждое число выделяется область памяти определённого размера
- Целые числа:
  - Знаковые (все биты – информационные), хранение только неотрицательных чисел
  - Беззнаковые (старший бит – знаковый, остальные – информационные), имеется возможность хранения отрицательных значений
- Переполнение – ситуация, при которой результат операции требует больше памяти, чем выделено
- Факт переполнения означает, что полученный результат неверен с **математической** точки зрения

Представление целых чисел

- Беззнаковые числа (n=3)
  - $000_2 = 0_{10}$
  - $001_2 = 1_{10}$
  - $010_2 = 2_{10}$
  - $011_2 = 3_{10}$
  - $100_2 = 4_{10}$
  - $101_2 = 5_{10}$
  - $110_2 = 6_{10}$
  - $111_2 = 7_{10}$
  - $1000_2 = 0_{10}$

$2^3 = 8$ различных значений

Представление целых чисел

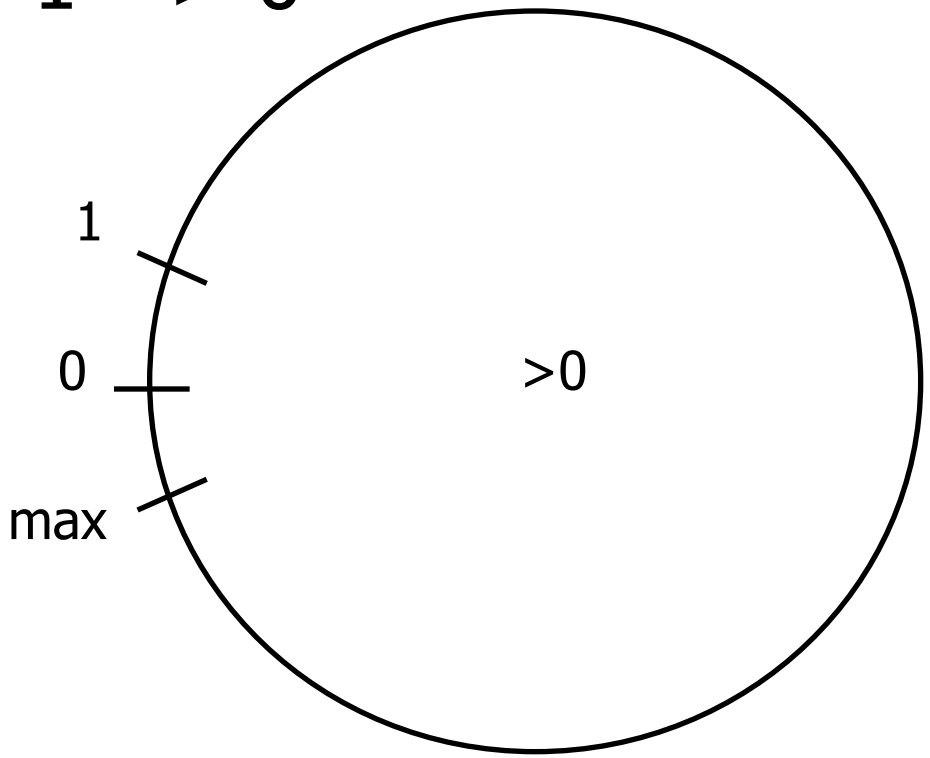


- Беззнаковые числа (n=3)
  - **111** + 1 = $\underline{1}\mathbf{000}_{\mathbf{2}} = 0_{10}$
- Признак возникновения переполнения – наличие старшей единицы, которая в дальнейшем отбрасывается
- При низкоуровневом программировании (например, на ASM) имеется возможность отследить факт возникновения переполнения
- Нет числовой прямой, есть ***числовое кольцо***

Представление целых чисел

0 – 1 => max

max + 1 => 0



Представление целых чисел

- Знаковые числа
  - **Прямой код** (ПК) числа – код, полученный простым переводом числа из $10^{й}$ в $2^{ю}$ СС
  - **Обратный код** (ОК) числа – код, полученный инвертированием всех разрядов ПК
  - **Дополнительный код** (ДК) числа – ОК, к которому выполнили арифметическое +1

Представление целых чисел

- ДК позволяет заменить операцию вычитания операцией сложения (числа A и B – положительные):

$$a - b = a + (-b) = a + c$$

- Целое положительное число C ведёт себя так же, как отрицательное число (-b)
- Это возможно из-за того, что числовой набор представляет не прямую, а кольцо
- ДК унифицирует алгоритмы выполнения операций знаковых и беззнаковых чисел в ЭВМ (упрощение архитектуры)

Представление целых чисел

- Считается, что в ДК переводятся только отрицательные числа
- Представления неотрицательных чисел в ПК и ДК совпадают
- Алгоритмы перевода ПК->ДК и ДК->ПК совпадают:
  - Инвертирование
  - +1

Представление целых чисел

- N=5 (жирным – знаковый разряд)
- ПК->ДК
  - ПК числа +12: $\mathbf{0}1100_2$
  - ОК числа -12: $\mathbf{1}0011_2$
  - ДК числа -12: $\mathbf{1}0100_2$
- ДК->ПК
  - ДК числа -12: $\mathbf{1}0100_2$
  - Инверсия ДК: $\mathbf{0}1011_2$ (это не ОК!)
  - ПК числа +12: $\mathbf{0}1100_2$

Представление целых чисел

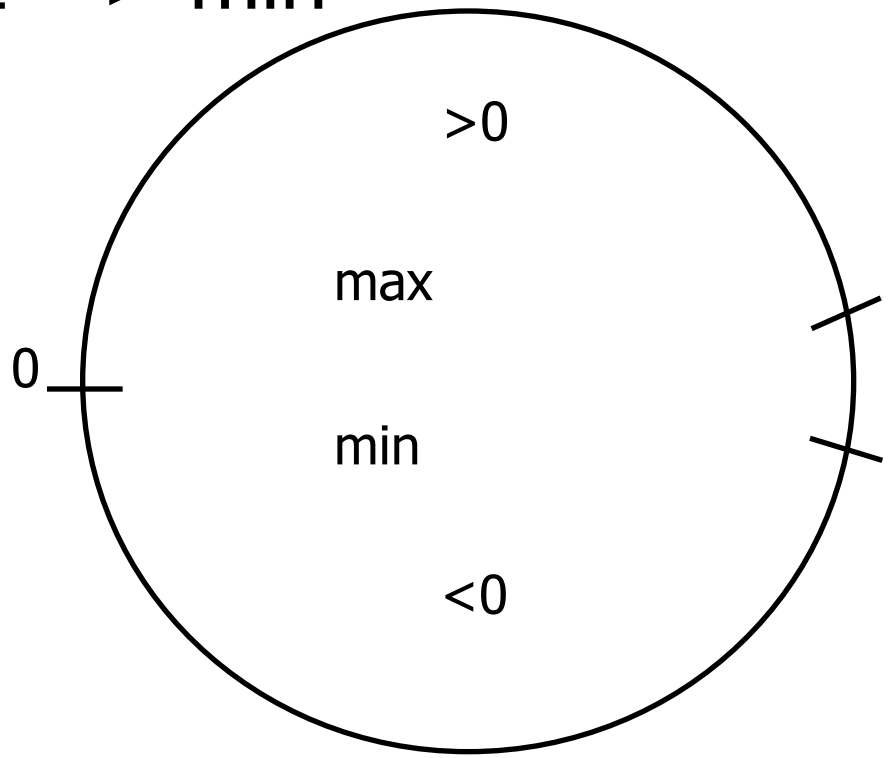
- Знаковые числа (n=3)
  - $000_2 = 0_{10}$
  - $001_2 = 1_{10}$
  - $010_2 = 2_{10}$
  - $011_2 = 3_{10}$
  - $100_2 = -4_{10}$ *(переполнение!)*
  - $101_2 = -3_{10}$
  - $110_2 = -2_{10}$
  - $111_2 = -1_{10}$
  - $1000_2 = 0_{10}$

$2^3 = 8$ различных значений

Представление целых чисел

min – 1 => max

max + 1 => min



Представление целых чисел

- (n=5) Пример выполнения операции 12-5:
- 12: ПК = **0**1100$_2$
- 5:
  - ПК = **0**0101$_2$ (+5)
  - ОК = **1**1010$_2$ (-5)
  - ДК = **1**1011$_2$ (-5)
- 12_{10}-5_{10} = **0**1100$_2$+**1**1011$_2$= **(1)0**111$_2$ = 7_{10}
- Ноль в знаковом разряде – признак ПК

0	1	1	0	0
1	1	0	1	1
(1) 0	0	1	1	1

Представление целых чисел

- Признак переполнения – наличие нечётного суммарного количества «единиц» в четырёх знаковых разрядах операндов и результата (включая теряющийся разряд – при наличии)
- -6: ПК=**0**0110$_2$, ОК=**1**1001$_2$, ДК=**1**1010$_2$
- -11: ПК=**0**1011$_2$, ОК=**1**0100$_2$, ДК=**1**0101$_2$
- Переполнение произошло, результат неверен

-6	**1**	1	0	1	0
-11	**1**	0	1	0	1
-17 -> +15	**(1) 0**	1	1	1	1

Представление целых чисел

- Единица в знаковом разряде – признак ДК
- (n=5) Пример выполнения операции 8+10:
- 8: ПК=**0**1000_2
- 10:ПК=**0**1010_2
- Переполнение произошло, результат неверен
- ДК=**1**0010_2, x=**0**1101_2, ПК=**0**1110_2=14_{10}

8	**0**	1	0	0	0
10	**0**	1	0	1	0
18 -> -14	**(0) 1**	0	0	1	0

Представление целых чисел

- Допускается запись в память числа без знака, а чтение со знаком (и наоборот), например:
- BC++ 3.1:

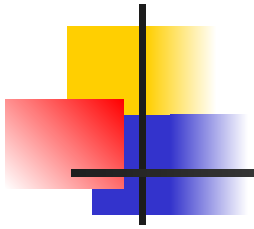
```
unsigned int k = -200;
short int p = 40000;
cout<<k<<p;
```

- MS VS 2008 (C#)

```
short x = 20000, y = 20000;
short z = (short)(x + y);
MessageBox.Show(z.ToString());
```

Представление целых чисел

- Диапазоны хранимых значений:
  - беззнаковые – $[0; 2^n-1]$
  - Знаковые – $[-2^{n-1}; 2^{n-1}-1]$
- Стандартные типы в ЭВМ
  - **8 битов** (unsigned char, byte / char, shortint) – [0; 255], [-128; 127]
  - **16 битов** (unsigned short int, word / short int, integer) – [0; 65535], [-32768; 32767]
  - **32 бита** (unsigned long int, cardinal / long int, longint) – [0; 4.2млрд], [-2.1млрд; 2.1млрд]
  - **64 бита** (int64) – $[0; 2^{64}-1]$, $[-2^{63}; 2^{63}-1]$



Кодирование символьной информации

Кодирование символьной информации

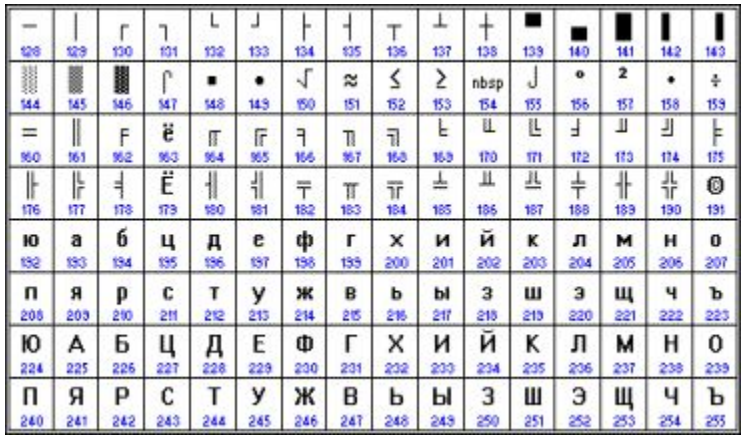
- Таблица ASCII – American Standard Code for Information Interchange
  - 1 символ = 8 битов
  - 2^8=256 символов:
    - 0..127 – базовая часть
    - 128..255 – расширенная часть

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

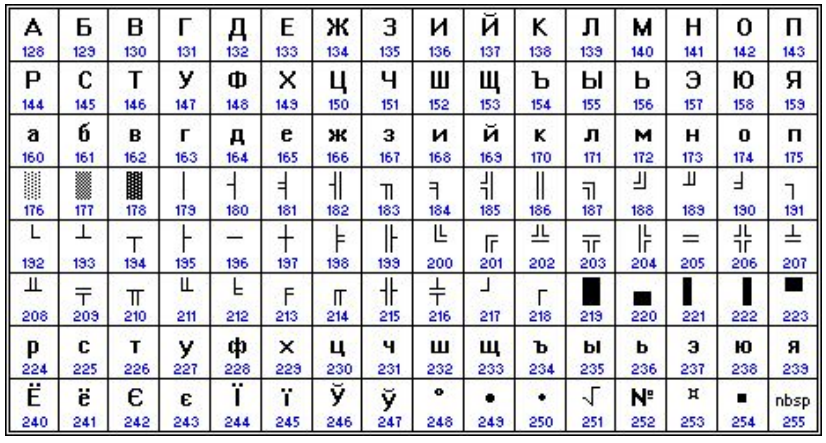
Кодирование символьной информации

КОИ8-Р

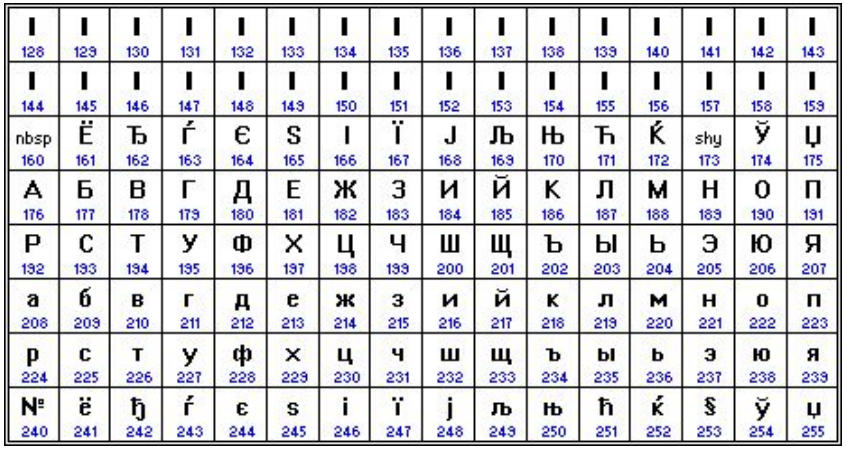


СР-1251

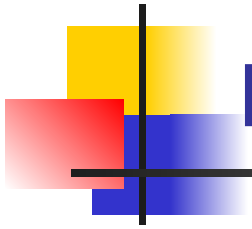
СР866



ISO



Пример: ╧юяЁюёшЄх фтр фюяюыэшЄхы№э√ї срыыр є ыхъЄюЁр ш эшъюьє юс ¤Єюь эх Ёрёёърч√трщЄх

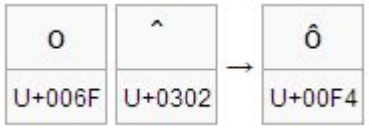


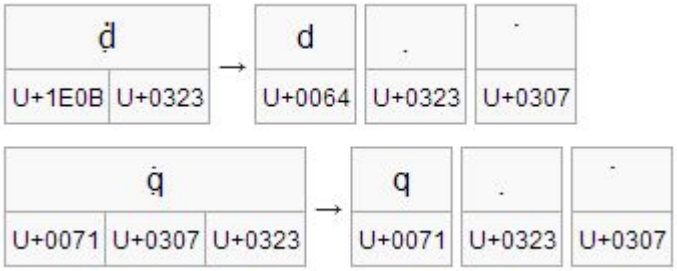
Кодирование символьной информации

- Unicode – стандарт 1991 года
  - 1 символ = 16 бит
  - 2^{16} = 65536 символов
    - 0..127 совпадает с ASCII (для совместимости)
  - Кодирует символы почти всех письменных языков
  - Избавляет от необходимости переключать кодовые страницы
  - Поддерживает написание LTR и RTL
  - Поддерживает кодирование little-endian и big-endian – определяется в начале файла после маркера последовательности байтов

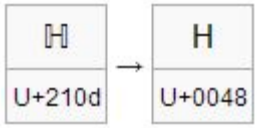
Кодирование символьной информации

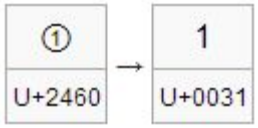
- Не поддерживает вертикальное письмо (поддержку должны обеспечивать текстовые редакторы)
- Поддерживает формы нормализации (композиции и декомпозиции)

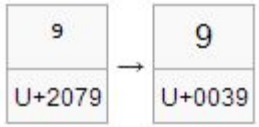


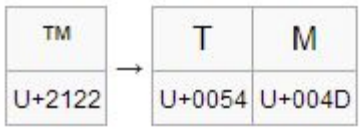


- Поддерживает таблицы совместимой декомпозиции









Представление чисел с ПЗ в памяти ЭВМ

Представление чисел с ПЗ

- Любое вещественное число представимо в системе счисления N в виде:

$$K = \pm M \cdot N^{\pm p}$$

- M – мантисса
- p – порядок
  - X – характеристика (смещённый порядок)

166

Представление чисел с ПЗ

- Нормализация:
  - Справа – после запятой стоит **не ноль**

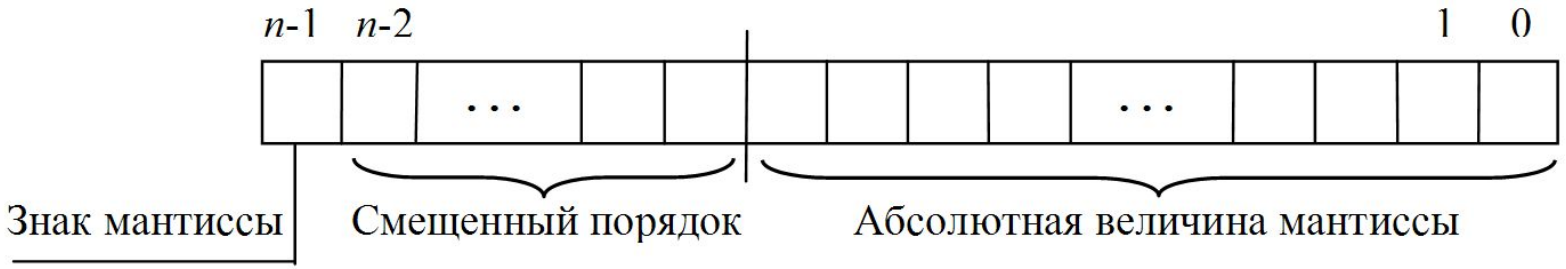
$$372{,}95_{10} = 0{,}37295 \cdot 10^{3}$$

$$0{,}011011_{2} = 0{,}11011 \cdot 2^{-1}$$

$$0{,}1_{2} = 0{,}1 \cdot 2^{0}$$

  - Слева. Согласно стандарту IEEE 754 – мантисса принимает значения $\mathbf{1 \le m \le N}$
    - Недостаток: невозможно закодировать «0», поэтому представление предусматривает специальный признак нулевого значения

Представление чисел с ПЗ



Общая структура вещественных чисел

$$X = 2^{b-1} + k + p$$

(*k* – *по*правочный коэффициент IBM)

Тип	Р-р памяти, бит = *b*+M	Знач. цифр	Диапазон
Real (k = +1)	**48=7+40**	**11–12**	
Single (k= – 1)	**32=8+24**	**7–8**	$1.4\cdot10^{-45}..3.4\cdot10^{38}$
Double (k= – 1)	**64=11+53**	**15–16**	$4.9\cdot10^{-324}..1.8\cdot10^{308}$
Extended (k= – 1)	**80=15+64**	**19–20**	$3.1\cdot10^{-4944}_{168}..1.2\cdot10^{4932}_{168}$

Представление чисел с ПЗ

- Алгоритм формирования двоичного представления вещественного числа:
  - Число представляется в двоичном коде.
  - Двоичное число нормализуется. При этом для чисел, больших единицы, плавающая точка переносится влево, определяя положительный порядок. Для чисел, меньших единицы, точка переносится вправо, определяя отрицательный порядок.
  - С учетом типа вещественного числа по таблице определяется характеристика.
  - В отведенное в памяти поле, в соответствии с типом числа, записываются мантисса, характеристика и знак числа
  - первый бит мантиссы (для нормализованного числа всегда 1) для чисел типа real, single, double не хранится (является скрытым). В числах типа extended все разряды мантиссы хранятся в памяти ЭВМ.

Представление чисел с ПЗ

- Пример: $-15{,}375_{10}$
  - Двоичная СС: $1111{,}011_2$
  - Нормализованное число: $1{,}111011_2 \cdot 2^3$
  - M = $1,\mathbf{111011}_2$
  - m = $\mathbf{111011}0\ldots0_2$
  - Порядок p = 3; X = $2^7 - 1 + 3 = 130_{10} = 10000010_2$
  - Знак s = 1
  - Результат: 00 00 76 C1 или 0x76C1 (*single*)

3								2								1								0							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*

3								2								1								0							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C				1				7				6				0				0				0				0			

170

Представление чисел с ПЗ

- Пример: $0{,}1875_{10}$
  - Двоичная СС: $0{,}0011_2$
  - Нормализованное число: $1{,}1_2 \cdot 2^{-3}$
  - M = $1,\mathbf{1}_2$
  - m = $\mathbf{1}0\ldots0_2$
  - Порядок p = –3; X = $2^7 - 1 - 3 = 124_{10} = 01111100_2$
  - Знак s = 0
  - Результат: 00 00 40 3E или 0x403E (*single*)

3								2								1								0							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*	*m*

3								2								1								0							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**0**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3				E				4				0				0				0				0				0			

171

Представление чисел с ПЗ

- Пример: 0.1_{10}
- Двоичная СС:
  - $0,0(0011)_2$
  - M=1,**10011001100110011001100**110
  - m= **10011001100110011001101**0
    - 0.10000001490116_{10}
    - результирующее число чуть больше 0.1_{10}
    - при выводе числа на экран может быть как 0.1, так и не 0.1 (зависит от компилятора среды программирования и точности вывода числа по умолчанию)

Неочевидные особенности вещественных чисел

Неочевидные особенности вещ. чисел

- Сетка чисел, которые способна отобразить арифметика с плавающей запятой, неравномерна:
  - более густая для чисел с малыми порядками
  - более редкая для чисел с большими порядками
- ***Машинной эпсилон*** называется наименьшее положительное число ε такое, что $1 \oplus \varepsilon \neq 1$
  - $\oplus$ – машинное сложение

Неочевидные особенности вещ. чисел

```
var R:Single;
begin
   R:=0.1;
   if R=0.1
    then Label1.Caption:='Равно'
    else Label1.Caption:='Не равно'
end;
```

Неочевидные особенности вещ. чисел

```
var R:Single;
      I:Integer;
begin
   R:=1;
   for I:=1 to 10 do
    R:=R-0.1;
   Label1.Caption:=FloatToStr(R)
end;
```

Неочевидные особенности вещ. чисел

$$S = 10^{-9} + 10^{-9} + \ldots 10^{-9}$$
$$10^{9}$$

```
var s,p: single;
      i: longint;
begin
   s:=0; p:=1e-9;
   for i:=1 to 1000000000 do
    s:=s+p;
   writeln(s);
end.
```

Неочевидные особенности вещ. чисел

Результат: $0{,}03125 = 0{,}00001_2 = 1{,}0_2 \cdot 2^{-5}$

При типе double результат равен
0,9999999925399328 80

- Сложение 0,03125 и $1 \cdot 10^{-9}$
- Выравнивание порядков:
  - $1.000000 \cdot 10^{-9}$ $3{,}125000 \cdot 10^{-2}$
  - $0.100000 \cdot 10^{-8}$ $3{,}125000 \cdot 10^{-2}$
  - $0.010000 \cdot 10^{-7}$ $3{,}125000 \cdot 10^{-2}$
  - $0.001000 \cdot 10^{-6}$ $3{,}125000 \cdot 10^{-2}$
  - $0.000100 \cdot 10^{-5}$ $3{,}125000 \cdot 10^{-2}$
  - $0.000010 \cdot 10^{-4}$ $3{,}125000 \cdot 10^{-2}$
  - $0.000001 \cdot 10^{-3}$ $3{,}125000 \cdot 10^{-2}$
  - $0.000000 \cdot 10^{-2}$ $3{,}125000 \cdot 10^{-2}$

Неочевидные особенности вещ. чисел

```
var s,p: single;
      i: longint;
begin
   s:=1; p:=1e-9;
   for i:=1 to 1000000000 do
    s:=s+p;
   s:=s+1;
   writeln(s);
end.
```

$$S = 1 + 10^{-9} + 10^{-9} + \ldots + 10^{-9}$$

10^9

От перемены мест слагаемых сумма меняется!

В первом случае результат = 1

Во втором случае результат = 1,03125

Неочевидные особенности вещ. чисел

```
for (double i=0; i<=2; i+=0.1)
    cout<<i<<endl;
```

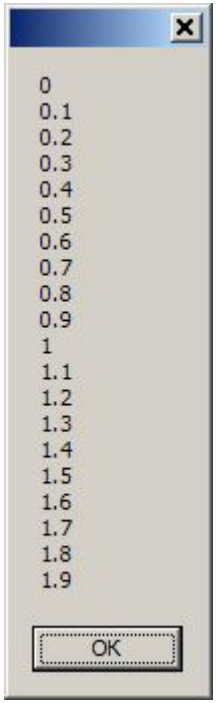
```
for (double i=0; i<3; i+=0.3)
    cout<<i<<endl;
```

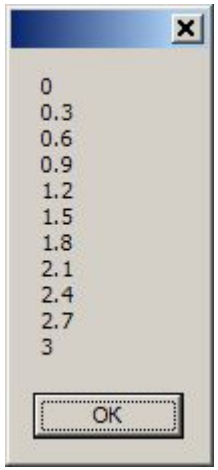
Выведутся ли 2 и 3 на экран?

Неочевидные особенности вещ. чисел

MS VS 2008 (C#)

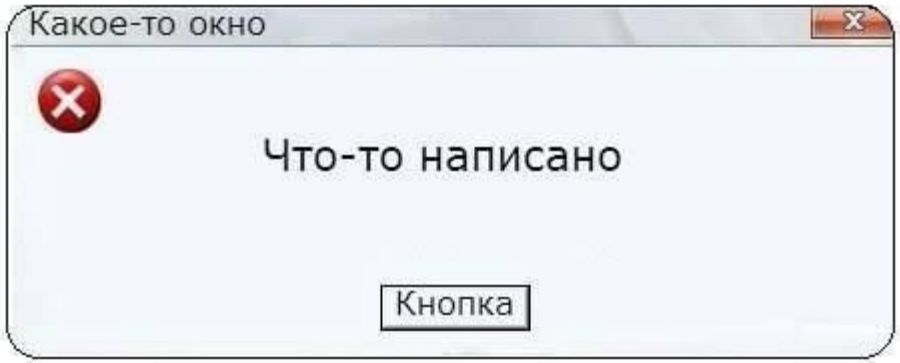
```
string s = "";
for (double k = 0; k <= 2; k += 0.1)
    s += k.ToString() + "\n";
MessageBox.Show(s);
```

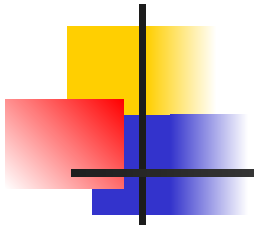





```
string s = "";
for (double k = 0; k < 3; k += 0.3)
    s += k.ToString() + "\n";
MessageBox.Show(s);
```

Ошибка глазами простого смертного:

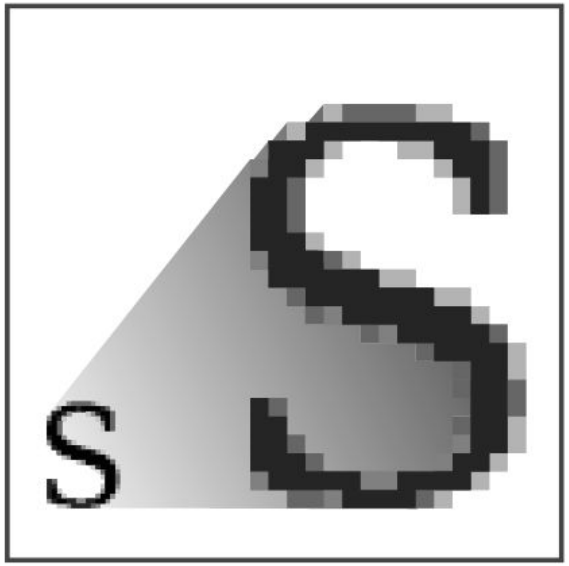




Кодирование графической информации

Кодирование графической информации

- Графика:
  - Растровая – изображение формируется из сетки цветных точек (как правило, прямоугольных)
  - Векторная – изображение формируется из элементарных геометрических объектов, таких как: точки, линии, сплайны и многоугольники. Объекты ВГ являются графическими изображениями математических функций.

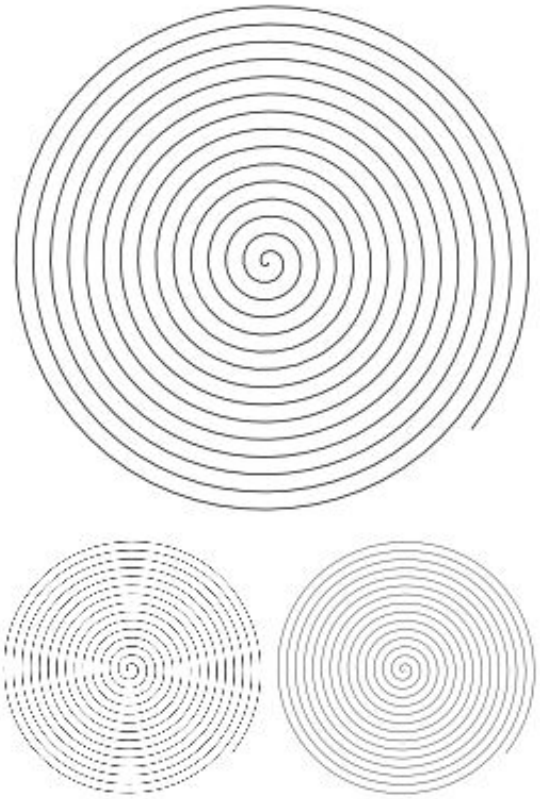


Кодирование графической информации

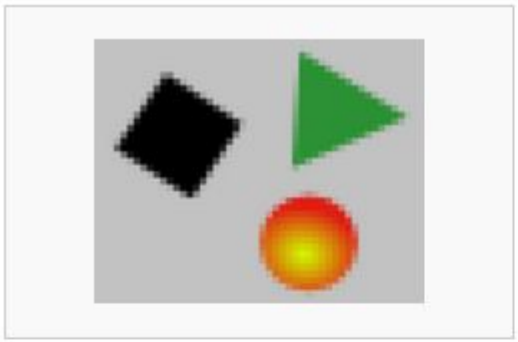
- Сильные стороны растровой графики:
  - Любой рисунок – одинаковый объём (при неизменных параметрах)
  - Высокая скорость обработки (за исключением масштабирования)
  - Естественно для большинства цифровых устройств ввода-вывода (мониторы, сканеры, принтеры, фотоаппараты)
- Слабые стороны растровой графики:
  - Большой размер файла даже у простых рисунков
  - Невозможность идеального масштабирования

Кодирование графической информации

- Масштабирование растра:



Эффект муара
Оригинал
Уменьшение в 2 раза без фильтрации
Уменьшение в 2 раза с фильтрацией.



Изображение, увеличенное в 4 раза
без фильтра

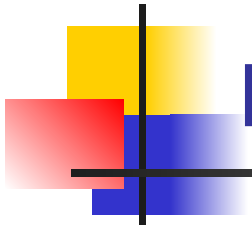
Изображение, увеличенное в 4 раза
с билинейной интерполяцией

Изображение, увеличенное в 4 раза
с фильтром Гаусса

Изображение, увеличенное в 4 раза
фильтром Ланцоша

Кодирование графической информации

- Сильные стороны векторной графики:
  - Размер файла не зависит от размера, но зависит от количества объектов
  - Идеальное масштабирование
  - Операции с объектами легко выполняются и не ухудшают качества рисунка
  - Каждый атрибут объекта меняется независимо от других (например, толщина линий не изменяется при изменении размера объекта)
  - Наличие z-координаты

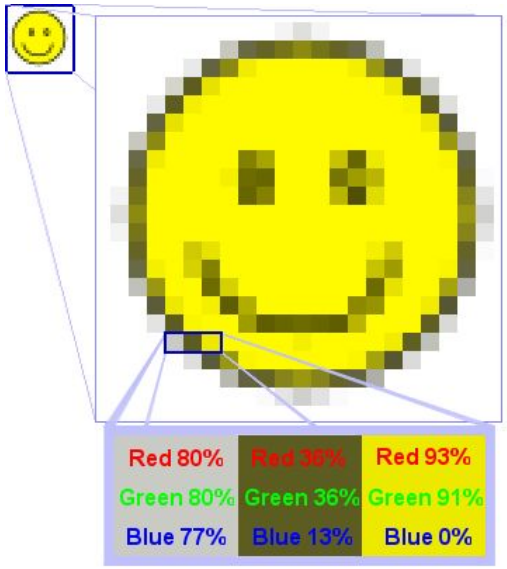


Кодирование графической информации

- Принципиальные проблемы с векторной графикой:
  - Не все изображения представимы в векторном виде
  - Растеризация проста, векторизация требует значительных затрат
  - Труднопримени́м для малых разрешений

Кодирование графической информации

- Характеристики растрового изображения:
  - количество точек
    - длина × ширина: 1920x1080
    - Общее количество точек: 2 Мп
  - Количество цветов (N) или глубина цвета (n): $N=2^n$
    - 256 цветов
    - 8 битов
  - Цветовое пространство: RGB, CMYK, YUV, …
  - Разрешение (справочная величина) – связывает размер изображения в цифровом виде и размер изображения на бумаге (dpi – **d**ots **p**er **i**nch)



Кодирование графической информации

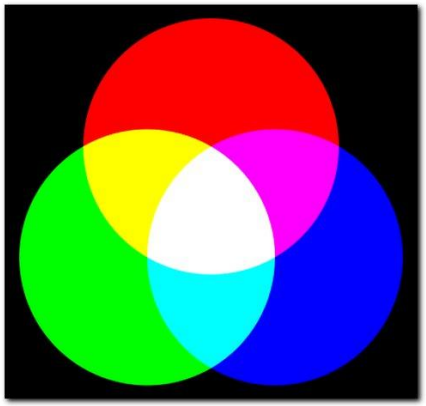
- Какого размера может получиться изображение формата HD?
  - При печати 600 dpi:
    - 1920/600 = 3.2” = 8.128 см
    - 1080/600 = 1.8” = 4.572 см
  - При печати 300 dpi:
    - 1920/300 = 6.4” = 16.256 см
    - 1080/300 = 3.6” = 9.144 см
- Фотоаппарат 24 Мп: 6000 x 4000, печать 600 dpi:
  - 6000/600 = 10” = 25.4 см
  - 4000/600 = 6,7” = 16.9 см

http://www.popmech.ru/technologies/9819-milliard-pikseley-gigapiksel/#full

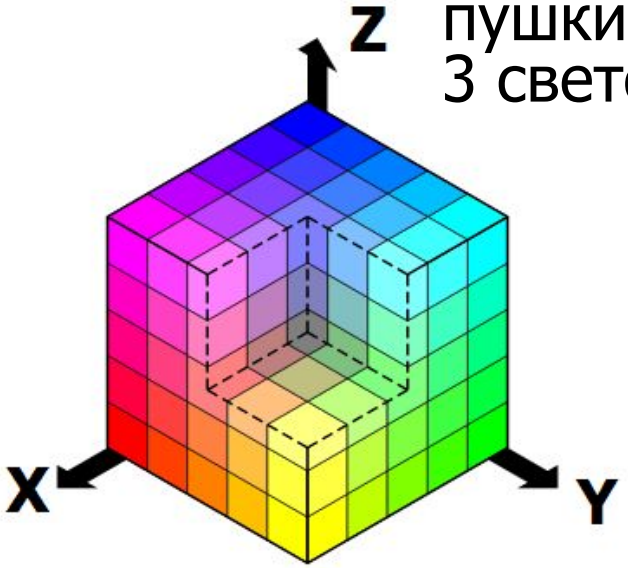
Кодирование графической информации

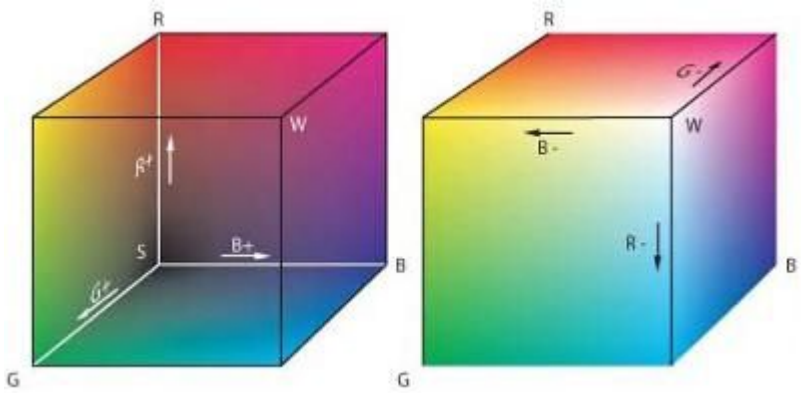
- Хотим сделать фотообои 2x1 м:
  - Разрешение не менее 300 dpi
  - (200 см = 78.74″) * 300 = 23 622 точки
  - (100 см = 39.37″) * 300 = 11 811 точек
  - 23622 * 11811 = 266 Мп
- Хотим сделать рекламный баннер 6x3 м:
  - Разрешение не менее 200 dpi
  - (600 см = 236.2″) * 200 = 47 244 точки
  - (300 см = 118,1″) * 200 = 23 622 точки
  - 47244 * 23622 = 1 Гп

Кодирование графической информации

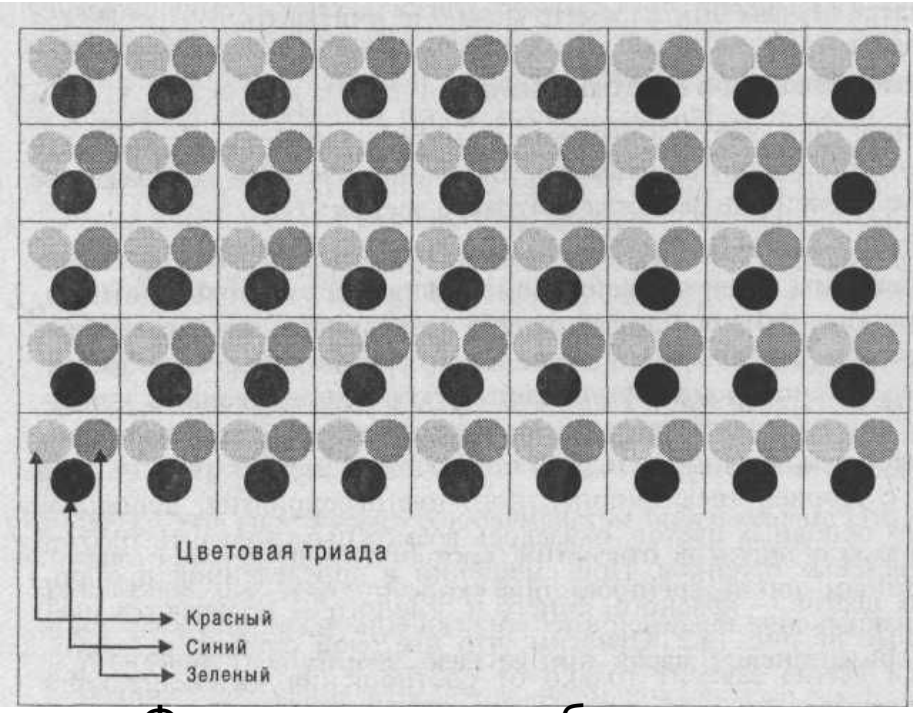


- Цветовое пространство RGB
  - Аддитивная цветовая модель
  - Добавление цветов к чёрному:
    - Красный (R), зелёный (G), синий (B)
  - Интенсивность: 0..255
    - (0,0,0) – чёрный
    - (0,0,255) - синий
  - Используется в мониторах, проекторах (3 электронные пушки / 3 ЖК / 3 светодиода / 3 светофильтра…)

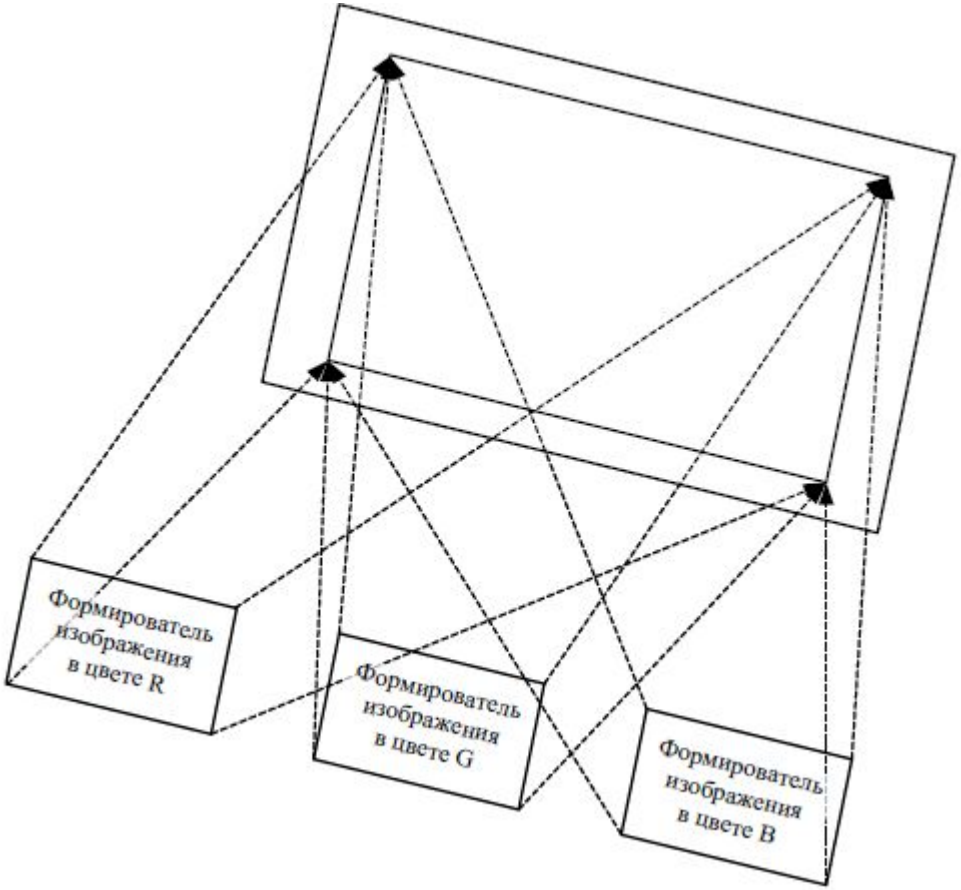




Кодирование графической информации



Формирование изображения на мониторе



Формирование изображения проектором на экране

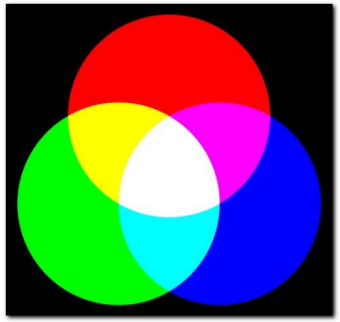
Кодирование графической информации

- При кодировании 1 бит/канал:

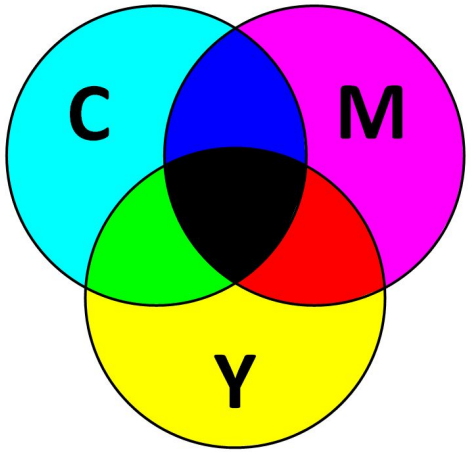
R	G	B	Цвет
1	1	1	белый
1	1	0	желтый
1	0	1	пурпурный
1	0	0	красный
0	1	1	голубой
0	1	0	зеленый
0	0	1	синий
0	0	0	черный

- При кодировании 8 бит/канал:
  - 2^8=256 градаций/канал
  - 256^3=16.7 млн цветов (TrueColor)

Кодирование графической информации

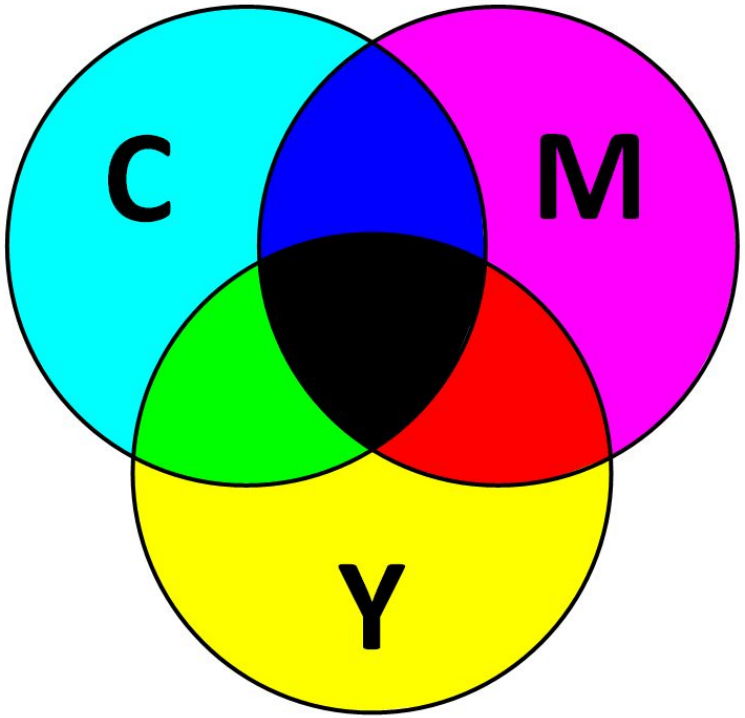


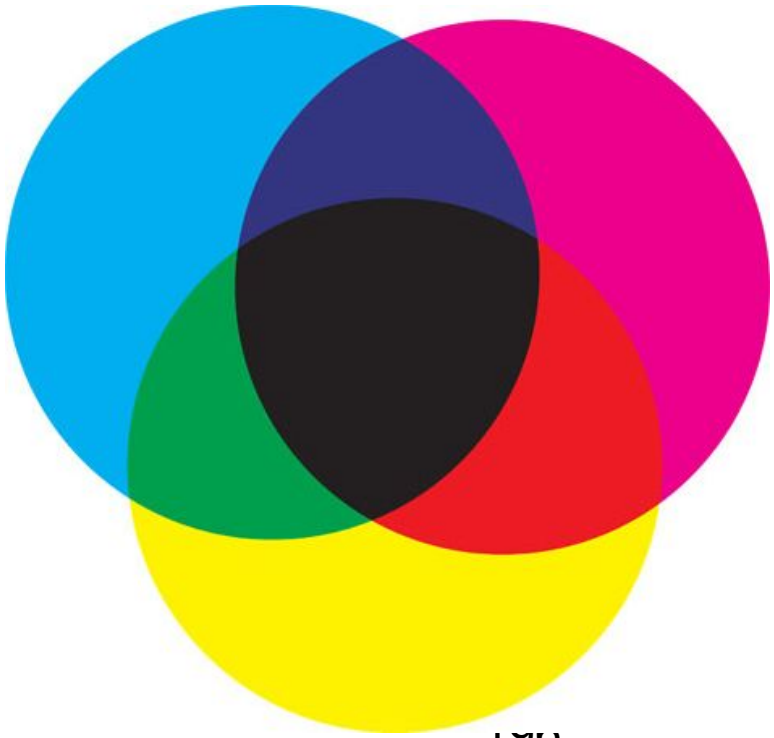
- Цветовое пространство CMY
  - Субтрактивная цветовая модель
  - Вычитание *первичных* цветов из белого с получением цветов:
    - Голубой? (C), пурпурный? (M), жёлтый (Y)
  - Интенсивность: 0..100
  - Используется при печати
  - Отдельные модели CMYK для каждого типа печати на каждом типе бумаги

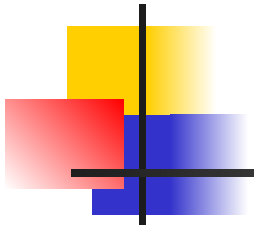


Кодирование графической информации

- CMYK:
  - Различие идеального и реального красителей
  - Ограничение по сумме красок зачастую меньше 300% (проблема чрезмерного смачивания бумаги)
  - Дешевизна четвёртого красителя



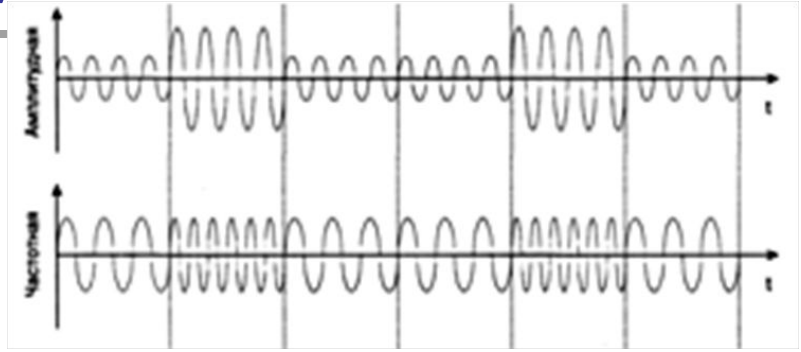




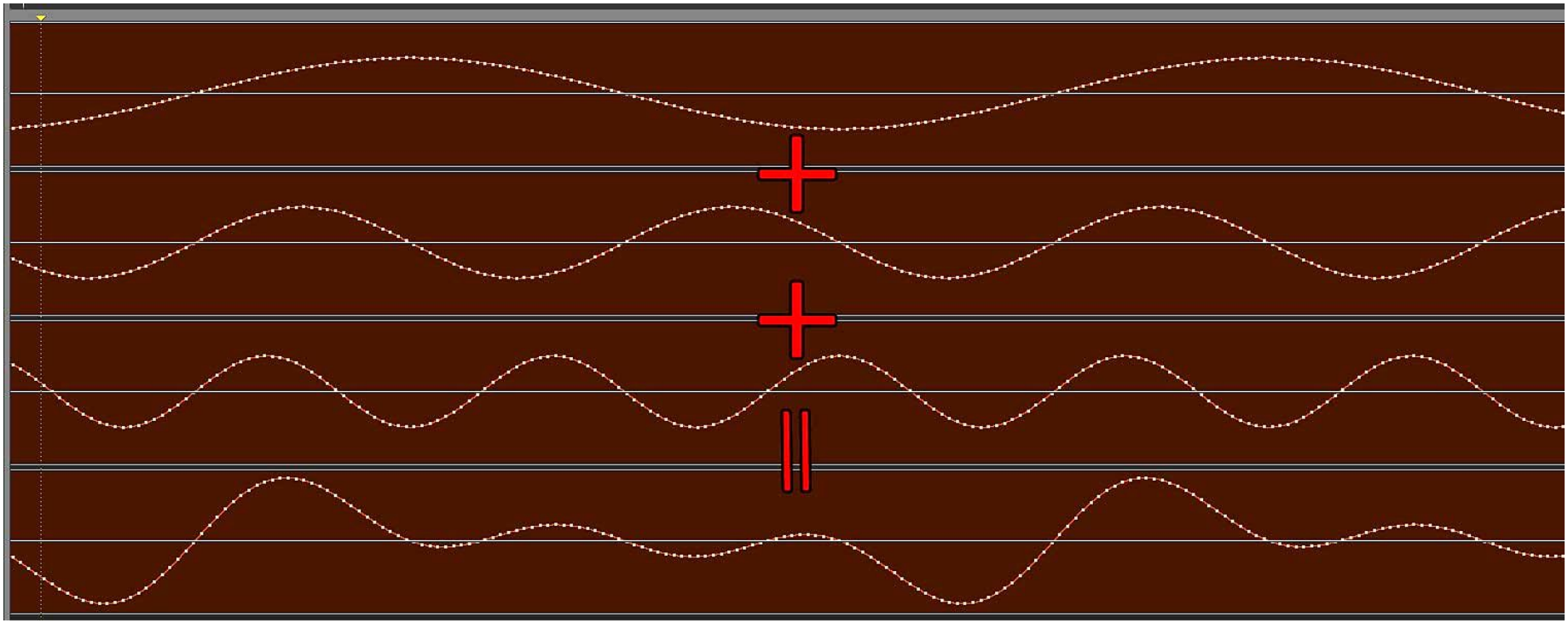
Кодирование аудио

Кодирование аудио

- Амп/частота:



- Сложение частот:



Кодирование аудио

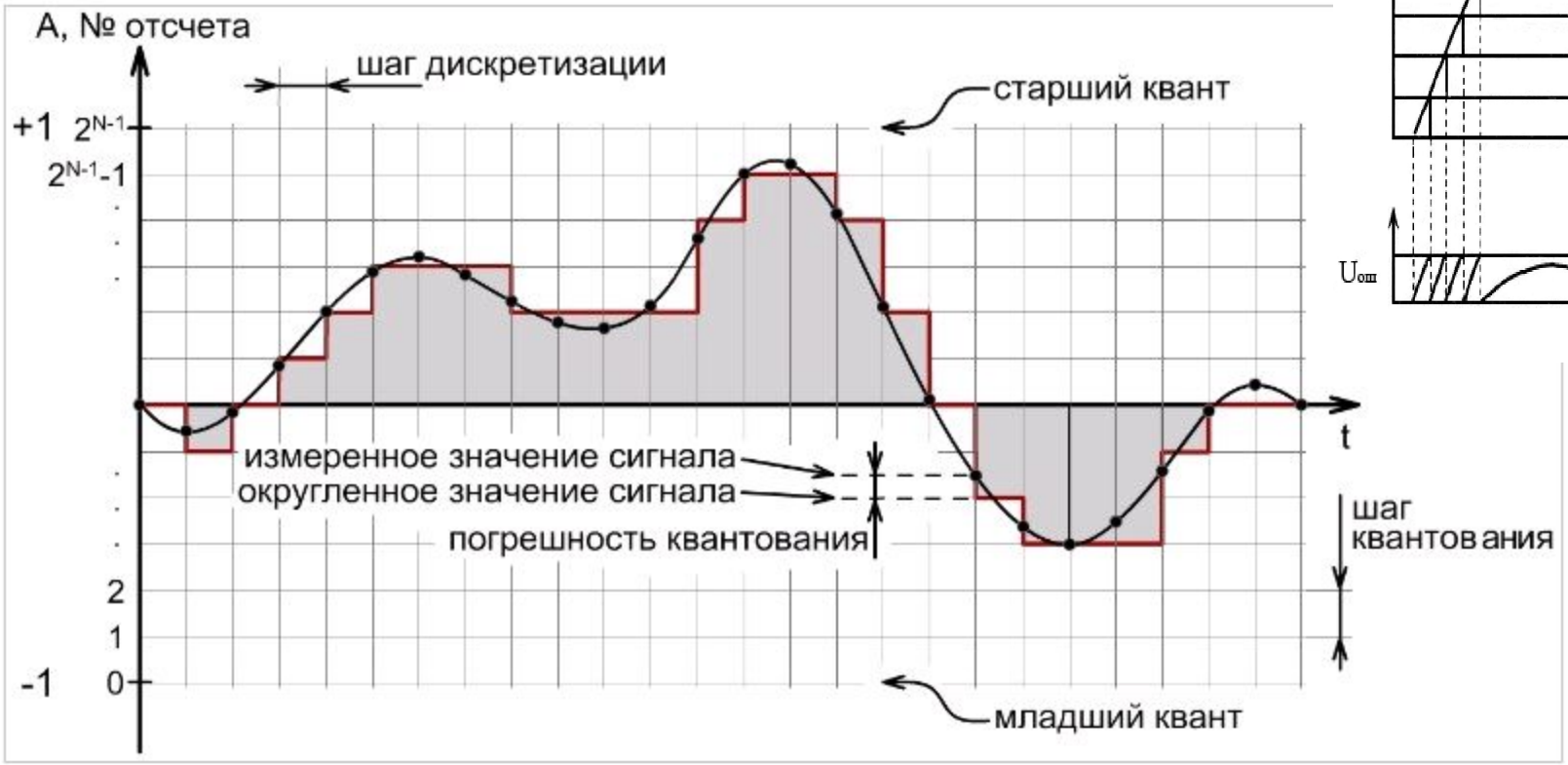
- ***Дискретизация по времени*** – процесс получения исходных значений сигнала с определенным временным шагом *(шагом дискретизации)*.
- ***Квантование по амплитуде*** – процесс замены реальных значений амплитуды сигнала значениями, приближенными с некоторой точностью.

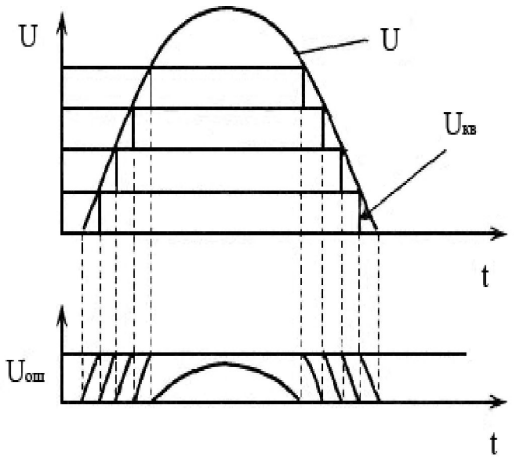
Кодирование аудио

- ***Частота дискретизации*** – количество замеров величины сигнала, осуществляемых в одну секунду
  - Чем меньше шаг дискретизации, тем выше частота дискретизации и тем более точное представление о сигнале нами будет получено.

- Теорема Котельникова (теорема Шеннона):
  - Аналоговый сигнал с ограниченным спектром может быть точно описан дискретной последовательностью значений его амплитуды, если эти значения берутся с частотой, как минимум вдвое превышающей наивысшую частоту спектра сигнала.

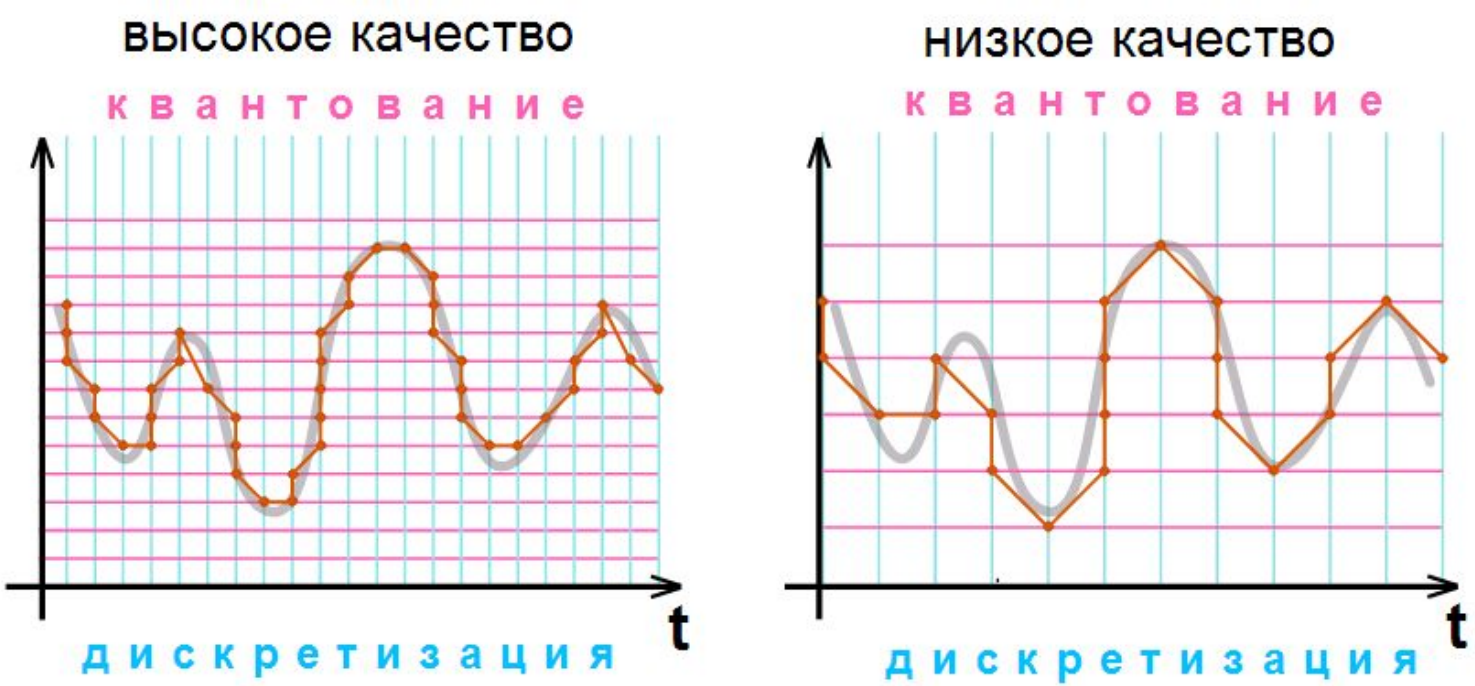
Кодирование аудио

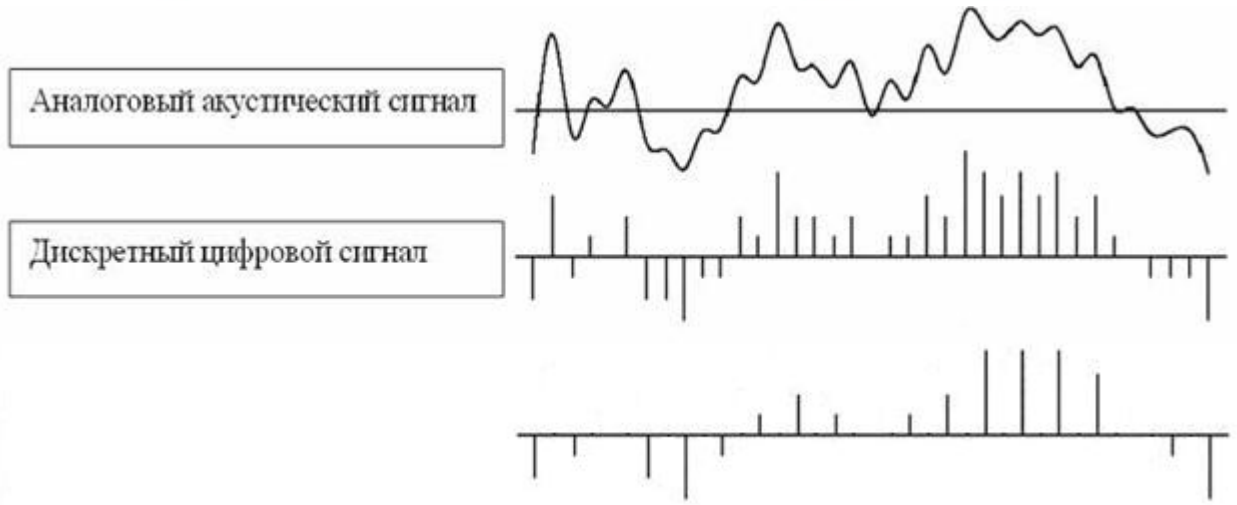




- Число N – *разрядность квантования*
- В результате округления значений амплитуды числа – *отсчеты* или *семплы*
- Принимается, что погрешности квантования, являющиеся результатом квантования с разрядностью 16 бит, остаются для слушателя почти незаметными
- PCM, ИКМ – Pulse Code Modulation, импульсно-кодовая модуляция
- DPCM, ДИКМ – дифференциальная ИКМ (кодирование только разности сигналов)
- ADPCM – адаптивная ДИКМ (изменяемый шаг квантования)

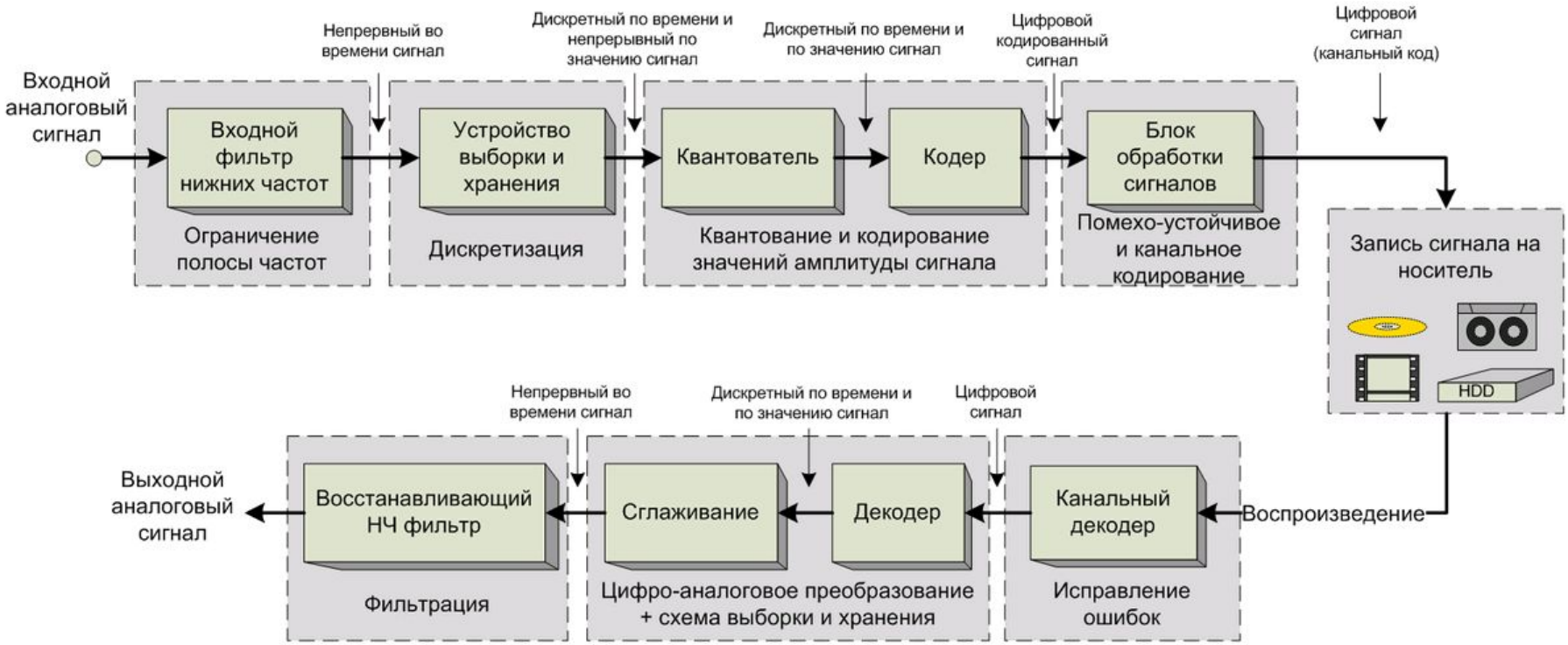
Кодирование аудио





Кодирование аудио

- Общая схема преобразования аналоговых и цифровых сигналов



Кодирование аудио

- АЦП (аналого-цифровое преобразование):
  - Ограничение полосы частот. Производится при помощи фильтра нижних частот для подавления спектральных компонент, частота которых превышает половину частоты дискретизации.
  - Дискретизация во времени. Эта задача решается путём использования специальной схемы на входе АЦП — устройства выборки-хранения.
  - Квантование по уровню. Представляет собой замену величины отсчета сигнала ближайшим значением из набора фиксированных величин — уровней квантования.
  - Кодирование или оцифровка. В результате значение каждого квантованного отсчета представляется в виде числа, соответствующего порядковому номеру уровня квантования.

Кодирование аудио

- ЦАП (цифро-аналоговое преобразование):
  - Декодер ЦАП преобразует последовательность чисел в дискретный квантованный сигнал
  - Путем сглаживания во временной области из дискретных отсчетов вырабатывается непрерывный во времени сигнал
  - Окончательное восстановление сигнала производится путем подавления побочных спектров в аналоговом фильтре нижних частот

Кодирование аудио

- Сравнение аудиоформатов

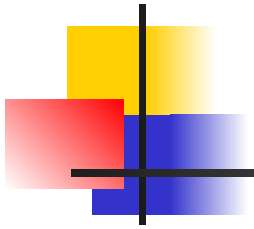
	Бит	Частота дискретизации, кГц	Число каналов	Величина потока данных с диска, кбит/с	Величина сжатия / упаковки
MP3	до 16	До 48	2	128 (12..320)	~11:1 / с потерями (зависит от потока)
CD	16	44,1	2	1411,2	1:1, без потерь
DolbyDigital 5.1	16..24	48	6	до 640	~12:1 с потерями
DTS	20...24	48 96	до 8	до 1536	~7:1 с потерями
DVD-audio	24	44,1 48 88,2 96	6	6912	2:1, без потерь
DVD-audio	24	176,4 192	2	4608	2:1, без потерь
AAC	до 64	до 96	до 48	до 529	с потерями
Ogg Vorbis	до 32	до 192	до 255	до 1000	с потерями
FLAC	до 32 *(24)*	до 655.35	до 8	не регл.	1.4:1 — 4:1 без потерь

Кодирование аудио

- Оценить объем стереоаудиофайла длительностью звучания 1 секунда при высоком качестве звука (16 битов, 48 кГц).
  - 16 битов • 48 000 Гц • 2 канала • 1 секунда = 1 536 000 битов = 192 000 байтов = 187,5 КБ.

Кодирование аудио

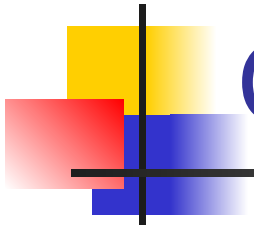
- (DTS-HD Master Audio) Звуковая дорожка 7.1-канального фильма длительностью 1.5 часа (24 бита, 96кГц):
  - 24 бита • 96 000 Гц • 8 каналов • 5400 секунд = 99 532 800 000 битов = 12 441 600 000 байтов = 11,59 ГБ



Сжатие данных

Сжатие данных

- **Сжатие данных без потерь** – метод сжатия данных, при использовании которого закодированные данные однозначно могут быть восстановлены с точностью до бита.
  - Для каждого из типов цифровой информации, как правило, существуют свои оптимальные алгоритмы сжатия без потерь.
- **Сжатие данных с потерями** — метод сжатия (компрессии) данных, при использовании которого распакованные данные отличаются от исходных, но степень отличия не является существенной с точки зрения их дальнейшего использования *(сжатие с точностью до чувствительности органов чувств человека)*.
  - Часто применяется для сжатия статических изображений, аудио- и видеоданных, особенно в потоковой передаче данных и цифровой телефонии



Сжатие данных

- Формирование префиксного кода:
  - Префиксный код – код со словом переменной длины, имеющий свойство: если в код входит слово *a*, то для любой непустой строки *b* слова *ab* в коде не существует.
- Хотя префиксный код состоит из слов разной длины, эти слова можно записывать без разделительного символа.
- Пример непрефиксного:
  - Слова: 0, 10, 11 и 100
  - Прочтение1: 0 10 0 11 0 11 10
  - Прочтение2: 0 100 11 0 11 10
- Пример префиксного:
  - Слова: 0, 10 и 11
  - Единственное прочтение: 0 10 0 11 0 11 10

Сжатие данных

- Пример:
  - Алфавит 4 символа
  - Сообщение 50 символов
- Стандартное кодирование:
  - N=4, n=2
  - K*n = 50*2 = 100 битов

А	Б	В	Г
00	01	10	11

- Код Хаффмана (частный случай энтропийного кодирования): построение кодов на основе информации о вероятности появления символов в сообщении
  - 28*1 + (10+6+6)*3 = 94 бита (-6%)

28	10	6	6
А	Б	В	Г

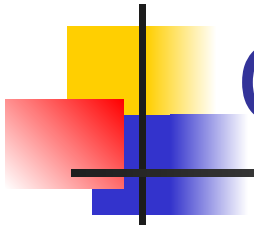
А	Б	В	Г
0	100	101	110

Сжатие данных

- Графические форматы, хранящие информацию без потерь:
  - BitMaP image (BMP)
  - Tagged Image File Format (TIFF)
  - Graphics Interchange Format (GIF)
  - Portable Network Graphic (PNG)
  - Photoshop Document (PSD)
  - RAW

Сжатие данных

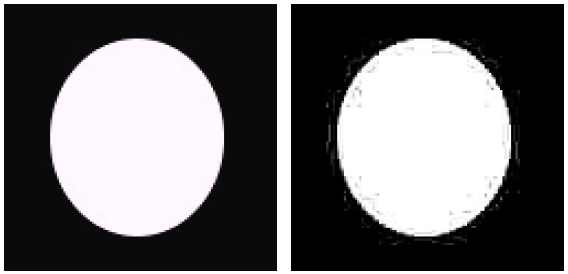
- Сжатие с потерями:
  - Существенно превосходят по степени сжатия
  - Используется для сжатия изображений, видео, аудио. Не применяется, например, для текстовой информации
  - Распакованный файл отличается при побитном сравнении, но *почти* неразличим органами чувств человека
  - При распаковке объем файла восстанавливается, качество – нет
  - При повторном сжатии (при редактировании) качество снова снижается



Сжатие данных

- Графические форматы, хранящие информацию с потерями:
  - Tagged Image File Format (TIFF)
  - Joint Photographic Experts Group (JPEG)

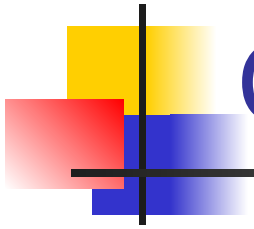
Сжатие данных



Эффект Гиббса

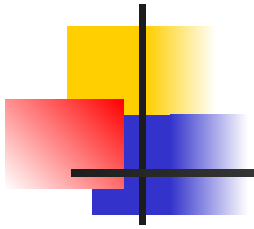
Эффект блочности в JPEG

HD в сравнении с SD



Сжатие данных

- Самостоятельно ознакомиться с информацией о форматах графических файлов:
  - BMP
  - JPEG / JPEG 2000 / JPEG-LS
  - GIF
  - PNG
  - WMF
  - Raw
  - TIFF



Целостность передачи информации

Целостность передачи информации

- Рекомендации по стандартизации Р 50.1.053-2005 и Р 50.1.056-2005:
  - **Целостность информации** — состояние информации, при котором отсутствует любое ее изменение либо изменение осуществляется только преднамеренно субъектами, имеющими на него право.
  - **Целостность ресурсов информационной системы** — состояние ресурсов информационной системы, при котором их изменение осуществляется только преднамеренно субъектами, имеющими на него право, при этом сохраняются их состав, содержание и организация взаимодействия.

Целостность передачи информации

Общая схема передачи информации

http://habrahabr.ru/company/mosigra/blog/274373/

Целостность передачи информации

- Борьба с помехами:
  - обнаружение ошибок в блоках данных и автоматический запрос повторной передачи повреждённых блоков (например, TCP)
  - обнаружение ошибок в блоках данных и отбрасывание повреждённых блоков (при отсутствии времени на повторную передачу и наличии возможности потери части данных, например, UDP)
  - исправление ошибок

Целостность передачи информации

- Корректирующие коды – коды, служащие для обнаружения или исправления ошибок, возникающих при передаче информации под влиянием помех, а также при её хранении.

- Источник информации для кодов – избыточные данные, добавляемые в сообщение (контрольное число)

Целостность передачи информации

- Контрольная сумма — некоторое значение, рассчитанное по набору данных путём применения определённого алгоритма и используемое для проверки целостности данных при их передаче или хранении.
  - Различие контрольных сумм означает различие входных данных
  - **НО!** Различие входных данных не обязательно влечёт различие контрольных сумм

Целостность передачи информации

- Пример простого контрольного числа:
- Исходное сообщение: 16353
- Контрольное число: Σ%10:
  - (1+6+3+5+3)%10 = 18%10 = 8
- Сообщение к отправке: 16353**8**

- Способность отследить ошибку: 2635**3**8
  - (2+6+3+5+3)%10 = 19%10 = 9 ≠ 8
- Нечувствительность к компенсирующим ошибкам: 16263**8**
- Коллизии: 16335**8**, 13365**8**, 11114**8**

Целостность передачи информации

- *Коды обнаружения ошибок* способны лишь определить факт возникновения ошибки
- *Коды, исправляющие ошибки*, способны исправить не более N ошибок в сообщении, однако они способны обнаружить факт возникновения большего количества ошибок
- *Блочные коды* делят информацию на фрагменты постоянной длины и обрабатывают каждый из них в отдельности
- *Свёрточные коды* работают с данными как с непрерывным потоком.

Целостность передачи информации

- Критерии «хорошего» блочного кода:
  - способность исправлять как можно большее число ошибок,
  - как можно меньшая избыточность,
  - простота кодирования и декодирования (→ линейность кодирования).

- Критерии 1 и 2 противоречат друг другу, решение – разработка индивидуальных кодов под конкретные задачи

Целостность передачи информации

- Линейные блочные коды преобразует фрагмент длиной **K** бит в кодовые слова длиной **N** бит.

- Расстояние Хемминга (для сообщения с контрольной информацией):
  - `110001010010110`
  - `110010110011110`
  - `******##**********`
  - d = 4

- Минимальное расстояние Хемминга определяет *корректирующую способность* (гарантированное количество исправляемых ошибок):

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$

http://habrahabr.ru/post/140611/

Целостность передачи информации

- Код Хемминга – самоконтролирующийся и самокорректирующийся линейный код общего вида

$$(i_1 \quad i_2 \quad i_3 \quad i_4)\begin{pmatrix}1&0&0&0&1&0&1\\0&1&0&0&1&1&1\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{pmatrix}=(i_1 \quad i_2 \quad i_3 \quad i_4 \quad r_1 \quad r_2 \quad r_3)$$

$$(i_1 \quad i_2 \quad i_3 \quad i_4 \quad r_1 \quad r_2 \quad r_3)\begin{pmatrix}1&0&1\\1&1&1\\1&1&0\\0&1&1\\1&0&0\\0&1&0\\0&0&1\end{pmatrix}=(S_1 \quad S_2 \quad S_3)$$

Синдром	Конфигурация ошибок	Ошибка в символе
000	нет ошибок	
001	0000001	r_3
010	0000010	r_2
011	0001000	i_4
100	0000100	r_1
101	1000000	i_1
110	0010000	i_3
111	0100000	i_2

- Для удобства обнаружения ошибок контрольные значения располагают в позициях с номерами целых степеней двойки

Целостность передачи информации

- Линейные циклические коды – линейные коды, в которых каждая циклическая перестановка кодового слова также является кодовым словом
- ЛЦК проще декодируются, чем линейные общего вида

$$v(x) = \sum_{i=0,\dots} v_i x^i$$

- Слова ЛЦК проще представлять в виде многочлена. Циклические сдвиги в этом случае – умножение на x по модулю x^n-1
- Примеры:
  - CRC (только обнаружение),
  - БЧХ (возможность построения кода с d_{min} не меньше заданного),
  - RS (работа с группами битов)

Целостность передачи информации

- Хеширование – преобразование по определённому алгоритму входного массива данных произвольной длины в выходную битовую строку фиксированной длины.
- Преобразования – хеш-функции или функции свёртки
- Результаты преобразований – хеш, хеш-код или сводка сообщения (*message digest*).

Целостность передачи информации

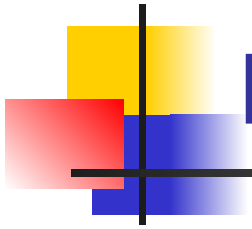
- Применение:
  - построение ассоциативных массивов
  - поиск дубликатов в сериях наборов данных
  - построение *?*уникальных*?* *(коллизии!)* идентификаторов для наборов данных
  - контрольное суммирование с целью обнаружения случайных или намеренных ошибок при хранении или передаче
  - хранение паролей в системах защиты

Целостность передачи информации

- Хеш-код короче исходных данных, поэтому одному хеш-коду может соответствовать несколько исходных данных – **коллизии**

- Характеристики хеш-функций:
  - разрядность
  - вероятность возникновения коллизии
  - Вычислительная сложность
  - криптостойкость

Целостность передачи информации



- Хеш-таблица – ассоциативный массив (ключ, значение)
- Например, алгоритм вычисления ключа: Σ%10
- 0 – АГД
- 1 –
- 2 – ВГД
- 3 –
- 4 –
- 5 –
- 6 – АБВ
- …
- ? – ВВВ
- ? – ГГГ

- Доступ к элементу – за время O(1)

Целостность передачи информации

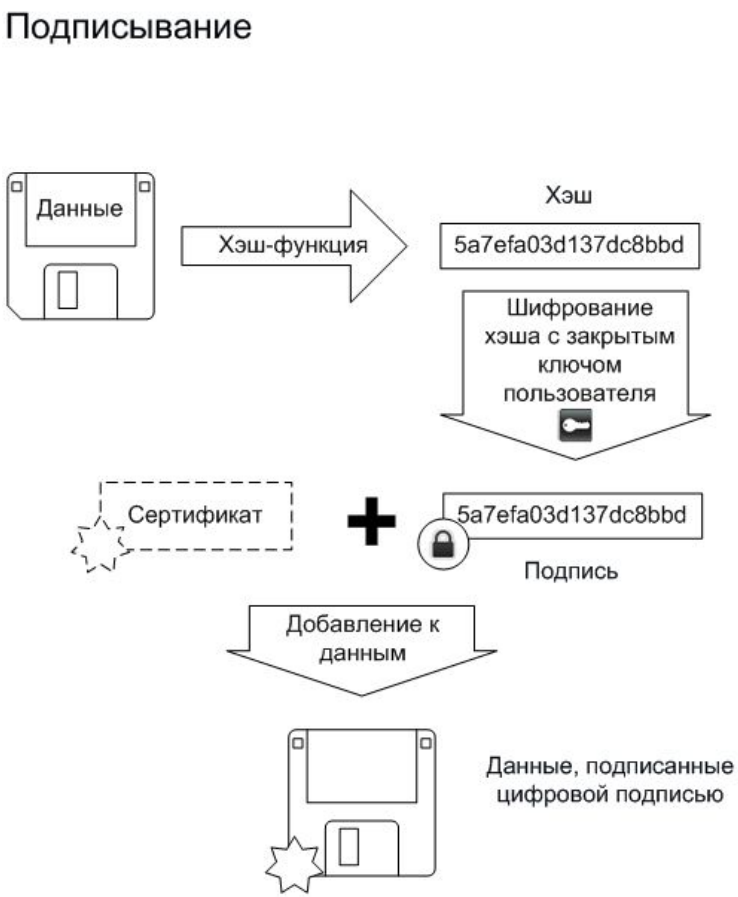
- Соль (модификатор) – строка случайных данных, которая подается на вход хеш-функции вместе с исходными данными.
- Удлинняет строку пароля, осложняет восстановление группы исходных паролей за один проход полного перебора или с помощью предварительно построенных радужных таблиц.
- **Не защищает** от полного перебора каждого пароля в отдельности.
- должна быть уникальной для каждого пароля
- **Не является секретной**
- Важная задача соли – в случае, если два пользователя поставили вдруг одинаковые пароли, сделать разными их хеши. Это скрывает факт совпадения паролей

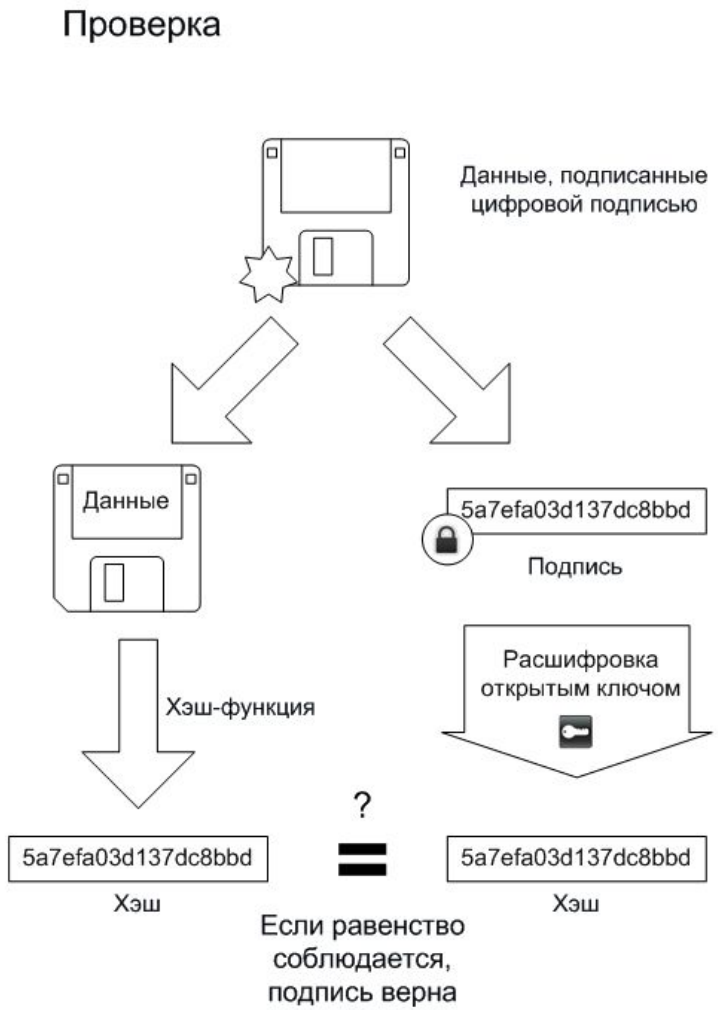
Salt	Pwd
06EAAEA6-C7D0-433C-B3E0-1ED313BB974E	0x591A93751F4BC7AEA5C2CEB71A28B756
B2715299-94B7-4D9E-A755-26161AF6B0A0	0x7757A761E73500EB707834CD8A09C7DA
9C287216-5DE0-4D11-A982-CAC4722B3C09	0x0577CFBAF8D3E6C2EB286954708C93CF
3216D4F3-ECC1-49D0-81AC-A17C49D2D8DE	0x1F5AEF0D9716F30C159761F0887D8960
B5763AEA-9D1C-4056-9475-667251E9C4BA	0xD59639458C508D0F055FF36F6043CA2A
26FA6487-F23B-4883-9293-F111C2CCD6C8	0x39CC1C1F466464AB04E086EE6D6B7A5C

Целостность передачи информации

- Электронная подпись (ЭП) — реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий установить отсутствие искажения информации в электронном документе с момента формирования подписи и проверить принадлежность подписи владельцу сертификата ключа подписи.
  - Контроль целостности передаваемого документа
  - Защиту от изменений (подделки) документа
  - Доказательное подтверждение авторства документа
  - Невозможность отказа от авторства

Целостность передачи информации





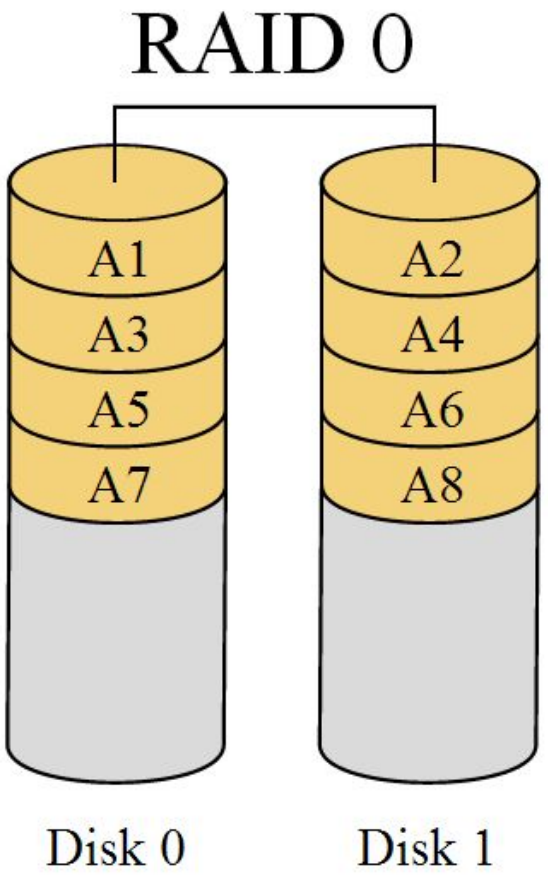
Надежность хранения информации

Надежность хранения информации

- RAID – redundant array of independent disks – избыточный массив независимых дисков
- Массив из нескольких дисков, управляемых контроллером, связанных между собой скоростными каналами передачи данных и воспринимаемых внешней системой как единое целое.

Надежность хранения информации

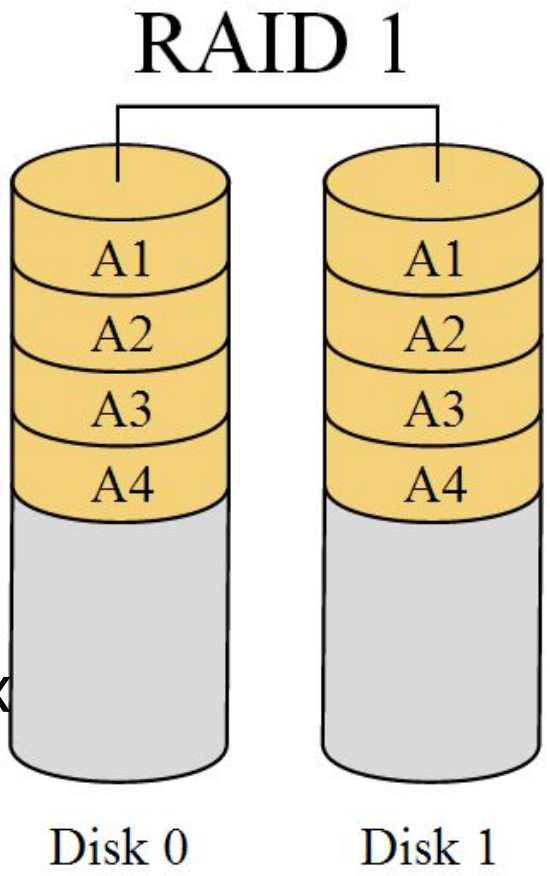
- RAID-0 (stripping)
- 2+ дисков
- Резервирование отсутствует.
- Информация разбивается на блоки данных (A_k) фиксированной длины и записывается на оба/несколько дисков одновременно.



- Один раздел большого объёма
- Позволяет использовать диски разного размера
- Выход из строя одного диска – потеря всей информации

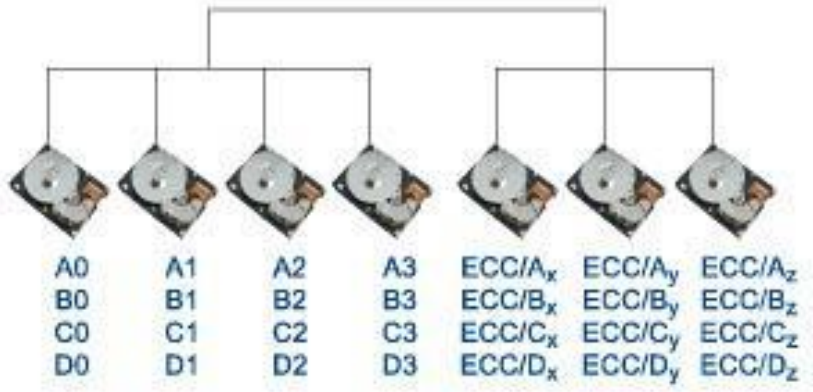
Надежность хранения информации

- RAID-1 (mirroring)
- 2 диска
- Резервирование – зеркалированием
- Скорость:
  - записи – как на любой из дисков
  - чтения – x2 (параллелизм)
- Цена: x2
- Простота восстановления данных (копирование)
- На практике используется в комбинации с RAID-0



Надежность хранения информации

- RAID-2
- 3+ (7+) дисков
- Использует кодирование Хемминга
- 2^n-1 дисков:
  - n дисков для контрольных сумм
  - 2^n-n-1 дисков с данными



Надежность хранения информации

- RAID-2
- Достоинства:
  - достаточно простая реализация
  - коррекция ошибок "на лету"
  - очень высокая скорость передачи данных
  - при увеличении количества дисков накладные расходы уменьшаются
- Недостатки:
  - низкая скорость обработки запросов
  - высокая стоимость
  - большая избыточность
  - сложность увеличения размера массива
- Не получил коммерческого применения
  - Плохо справляется с высокой нагрузкой
  - Предназначен для исправления ошибок «на лету»
  - Пользователей устраивает восстановление данных

Надежность хранения информации

- RAID-2

Количество дисков с данными	Количество дисков коррекции	Перерасход дисков %	Всего дисков
0	1	100,0000	1
1	2	66,6667	3
4	3	42,8571	7
11	4	26,6667	15
26	5	16,1290	31
57	6	9,5238	63
120	7	5,5118	127
247	8	3,1373	255
502	9	1,7613	511
1013	10	0,9775	1023
2036	11	0,5374	2047
4083	12	0,2930	4095
8178	13	0,1587	8191
16369	14	0,0855	16383
32752	15	0,0458	32767
65519	16	0,0244	65535
131054	17	0,0130	131071
262125	18	0,0069	262143
524268	19	0,0036	524287
1048555	20	0,0019	1048575

Надежность хранения информации

- RAID-3, RAID-4
- 3+ дисков
- Нет исправления ошибок «на лету»
- Меньшая избыточность, чем у RAID-2
- Данные хранятся на разных дисках:
  - побитно/побайтно (RAID-3)
  - поблочно (RAID-4)

Надежность хранения информации

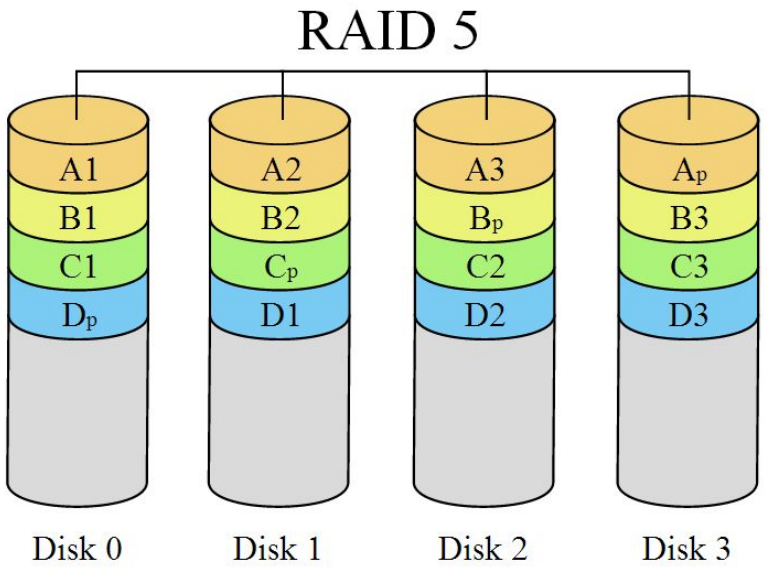
- Достоинства:
  - высокая надёжность хранения данных (допускается потеря не более 1 диска)
  - отказ диска мало влияет на скорость работы массива
  - высокая скорость чтения, приемлемая скорость записи
  - высокий коэффициент использования дискового пространства
- Недостатки:
  - все диски массива должны работать синхронно, что не даёт возможности обрабатывать одновременно более одного запроса
  - сложность реализации
  - сложное восстановление данных (для RAID-4)

Надежность хранения информации

- Самый большой недостаток уровней RAID от 2-го до 4-го – наличие отдельного (физического) диска, хранящего информацию о четности.
- Операции считывания не требуют обращения к этому диску, и, как следствие, скорость их выполнения достаточно высока, но при каждой операции записи на нем изменяется информация, поэтому схемы RAID 2-4 не позволяют проводить параллельные операции записи.

Надежность хранения информации

- RAID-5
- 3+ дисков
- Полезный объём – n-1 диск
- Контрольная сумма не привязана к конкретному диску
- XOR:
  - a⊕b=c
  - a⊕c=b

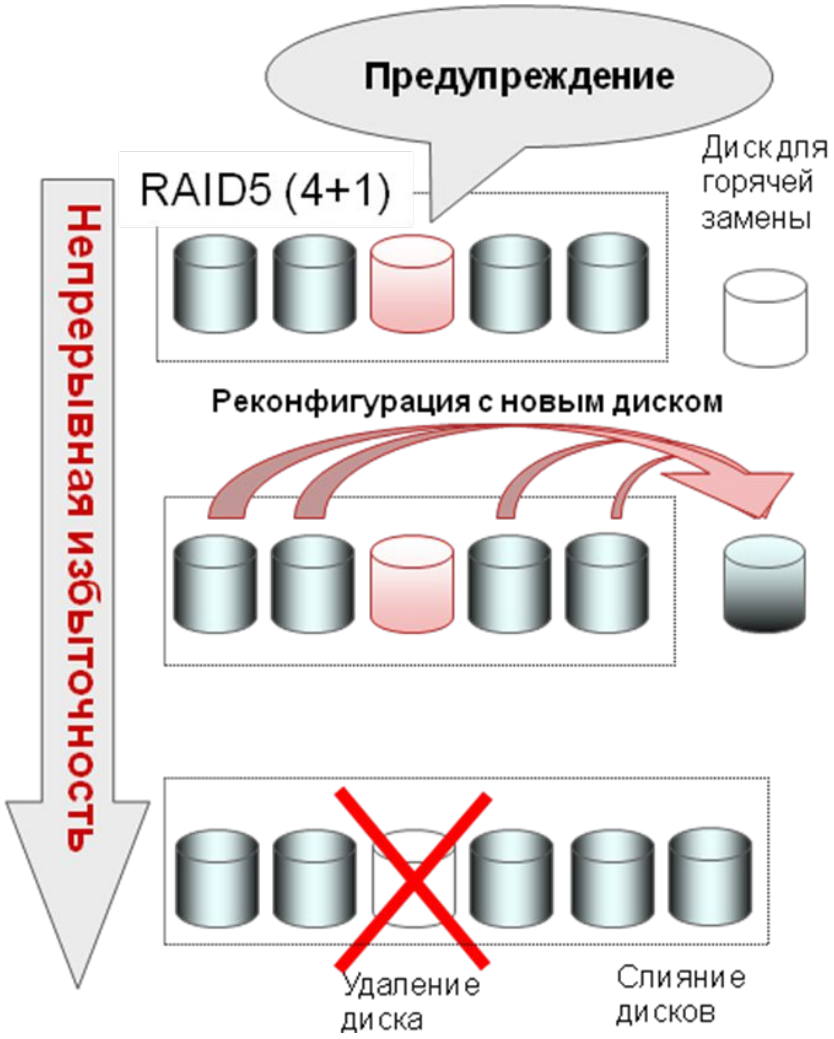


Надежность хранения информации

- Достоинства:
  - высокая надёжность хранения данных (допускается потеря не более 1 диска)
  - высокая скорость чтения
  - лучшая, чем у RAID-4 скорость записи
- Недостатки:
  - ограничения производительности записи из-за необходимости вычислять, пересчитывать и обновлять блоки чётности
  - При выходе из строя одного диска резко падает скорость записи

Надежность хранения информации

RAID-5 используется, как правило, с контроллерами, поддерживающими «диски горячей замены» (hot-spare disks)

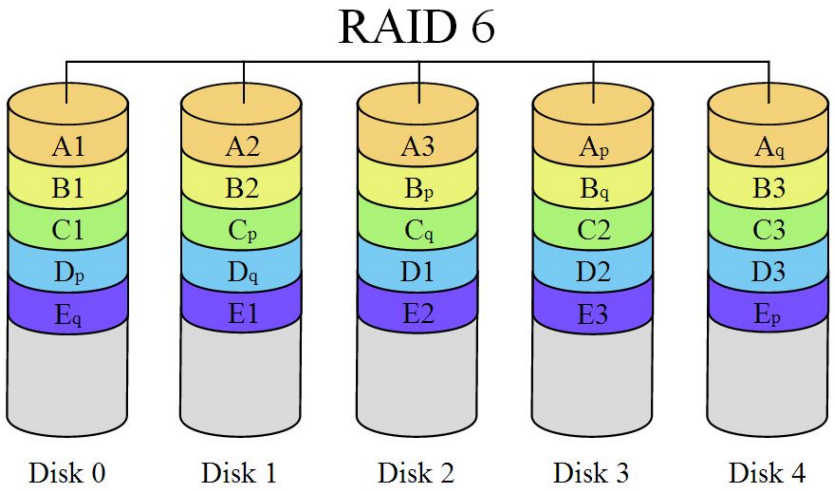


Надежность хранения информации

- RAID-6
- 4+ дисков
- Полезный объём – n-2 диска
- Контрольные суммы вычисляются различными алгоритмами

$$\mathbf{P} = \bigoplus_i D_i = \mathbf{D}_0 \oplus \mathbf{D}_1 \oplus \mathbf{D}_2 \oplus \ldots \oplus \mathbf{D}_{n-1}$$

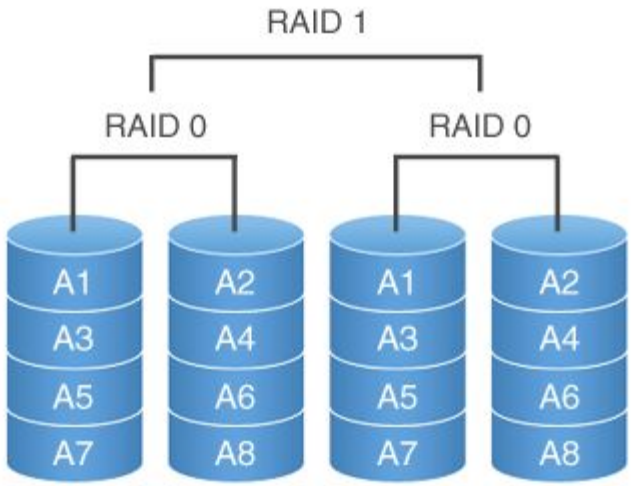
$$\mathbf{Q} = \bigoplus_i g^i D_i = g^0\mathbf{D}_0 \oplus g^1\mathbf{D}_1 \oplus g^2\mathbf{D}_2 \oplus \ldots \oplus g^{n-1}\mathbf{D}_{n-1}$$



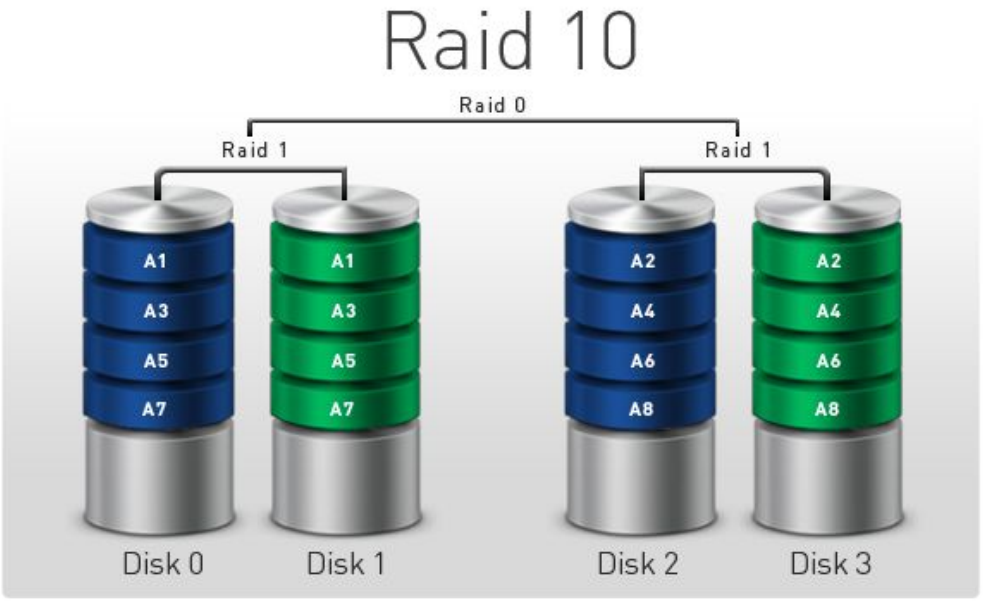
Надежность хранения информации

- Достоинства:
  - высокая надёжность хранения данных (допускается потеря не более 2 дисков)
  - высокая скорость чтения
- Недостатки:
  - Скорость записи ниже на 10-15%, чем у RAID-5 из-за двух контрольных сумм
  - При выходе из строя одного диска резко падает скорость записи
  - Очень сложная реализация
  - Сложное восстановление данных

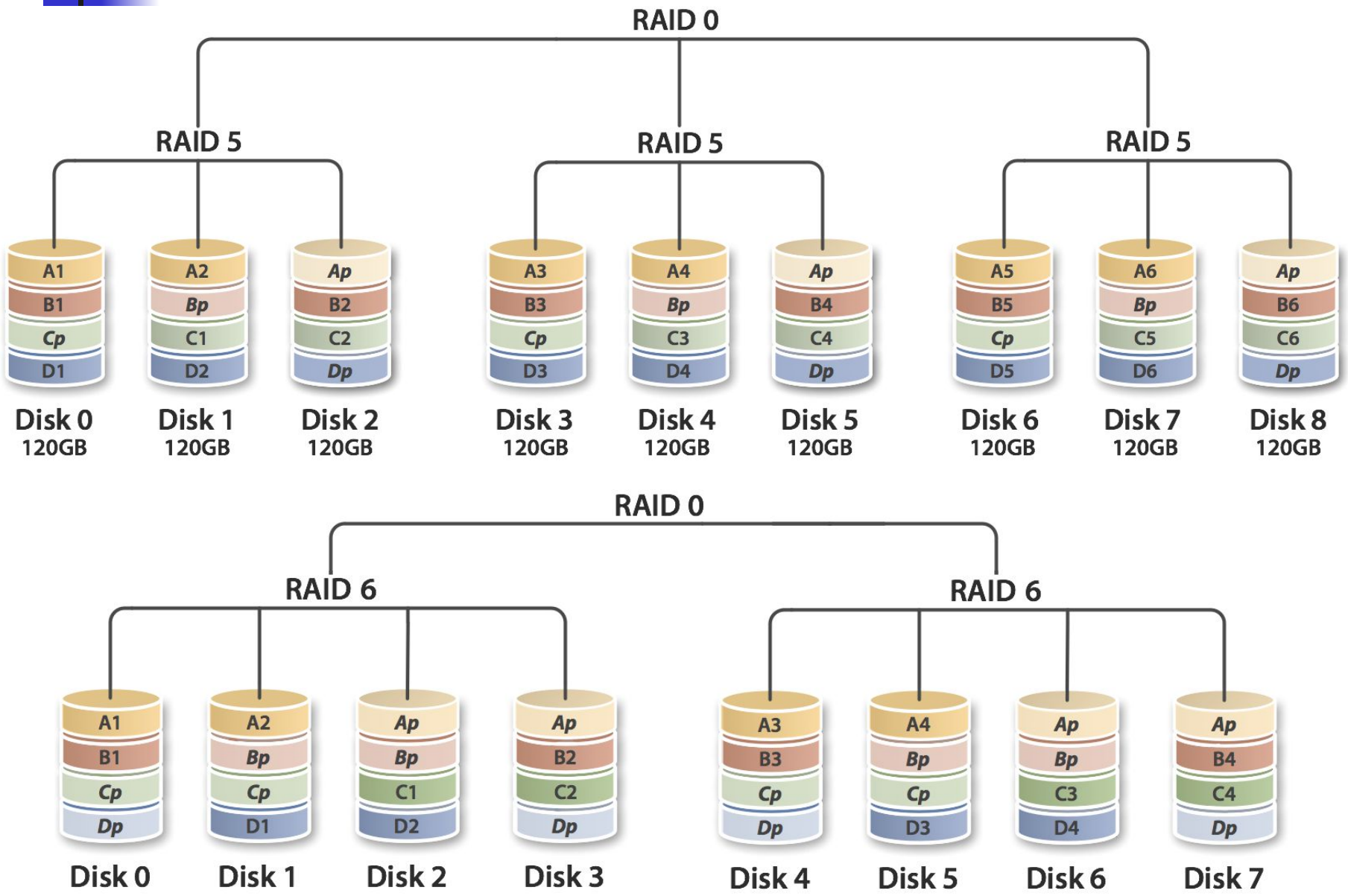
Надежность хранения информации



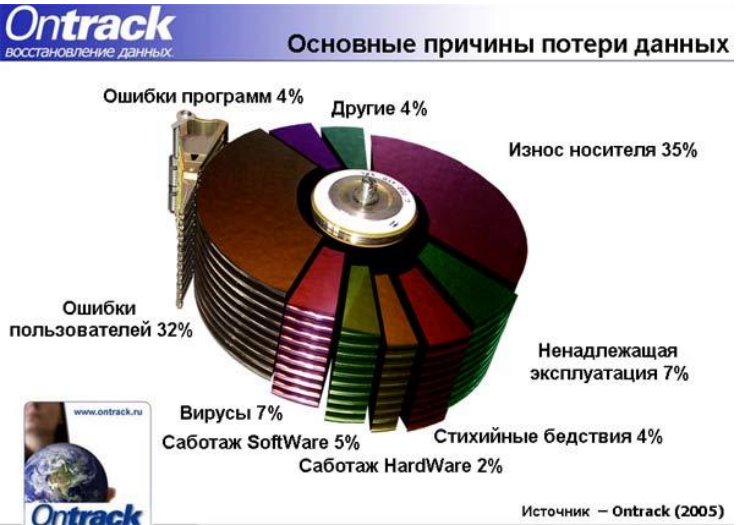
RAID 0+1

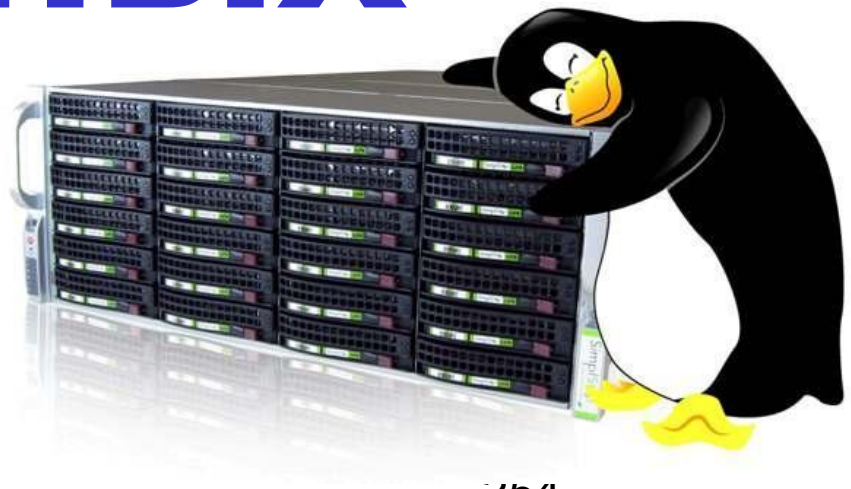


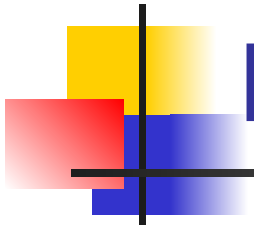
Надежность хранения информации



Резервное копирование данных







Резервное копирование данных

В бизнес требованиях никогда не написано «хранить файлы», а даже когда написано…

То, что называется системой резервного копирования – нет, это не система резервного копирования, это система аварийного восстановления, никому не нужно хранить файлы, всем нужно читать файлы – это важно.

Даниил Подольский,

конференция HighLoad++ Junior

Резервное копирование данных

- Резервное копирование (backup copy) – процесс создания копии данных на носителе. Созданная копия используется для восстановления данных в оригинальном или новом месте их расположения в случае их повреждения или разрушения.
- Фактически, резервная копия – это отражение данных в определённый момент времени
- Чем чаще меняются данные, тем чаще следует выполнять их резервное копирование

Резервное копирование данных

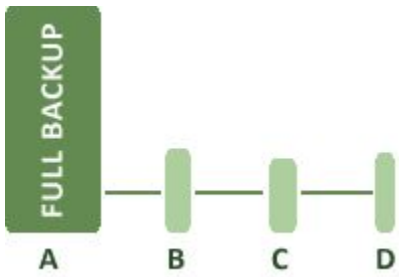
- Ключевые параметры:
  - RPO – Recovery Point Objective
  - RTO – Recovery Time Objective
- RPO определяет момент времени, на который будут восстановлены данные
- RTO определяет скорость восстановления данных (затраченное время)

Резервное копирование данных

- Полная копия (Full Backup)
  - Сохраняется весь набор информации
  - Проверка факта изменения не производится
  - Занимают наибольшее (по сравнению с другими типами копий) место на носителе
  - Наиболее быстро восстанавливается
  - Должны выполняться регулярно, но при этом должен соблюдаться баланс частоты/объёма (нагрузка на диски/сеть/т.д.) в зависимости от требований организации

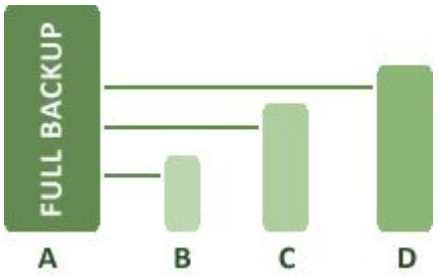
Резервное копирование данных

- Добавочная копия (incremental backup)
  - Сохраняется измененный объём данных (с момента FB или IB)
  - Занимает место, равное суммарному объёму изменённых данных
  - Требует значительного времени на восстановление



Резервное копирование данных

- Разностная копия (differential backup)
  - Сохраняется измененных объем данных (с момента FB)
  - Каждый DB содержит всю информацию, изменённую со времени FB (в общем случае каждый DB – больше, чем аналогичный IB).
  - Восстанавливается быстрее, чем IB



Резервное копирование данных

- Носители резервных копий:
  - Жесткий (магнитный) диск (дисковое хранилище)
  - Магнитная лента
  - Оптические диски
  - Сеть (другое ДХ / облако / и т.д.)
  - Твердотельные накопители

Шифрование данных

Шифрование данных

- **Кодирование** информации – процесс преобразования сигнала из формы, удобной для непосредственного использования информации, в форму, удобную для передачи, хранения или автоматической переработки.

- **Шифрование** информации – обратимое преобразование информации в целях сокрытия от неавторизованных лиц. При этом авторизованные пользователи имеют доступ к исходной информации.

- Цель шифрования: конфиденциальность передаваемой информации

- Алгоритм шифрования использует **ключ**

Шифрование данных

- Состояния безопасности информации:
  - Конфиденциальность
  - Целостность
  - Идентифицируемость

- Шифр – какая-либо система преобразования текста с ключом для обеспечения секретности передаваемой информации

Шифрование данных

- Шифрование (E, D – функции):
  - $E_{k1}(M) = C$
  - $D_{k2}(C) = M$
- **Симметричный** шифр использует один ключ для шифрования и дешифрования (k1=k2)
- **Асимметричный** шифр использует два различных ключа (k1≠k2).
- **Блочный** шифр шифрует сразу целый блок текста, выдавая шифротекст после получения всей информации.
- **Поточный** шифр шифрует информацию и выдает шифротекст по мере поступления

Шифрование данных

- ***Криптографическая стойкость*** –свойство криптографического шифра противостоять криптоанализу, то есть анализу, направленному на изучение шифра с целью его дешифрования
- Самый простой способ – полный перебор

$Log_{10}N$	**Операций**	**Прим.**
50	$1.4*10^{10}$	Суперкомпьютеры
100	$2.3*10^{15}$	Предел современных технологий
200	$1.2*10^{23}$	За пределами возм. современных технологий
400	$2.7*10^{34}$	Требует существенных изменений в технологии
800	$1.3*10^{51}$	Нераскрываем

Шифрование данных

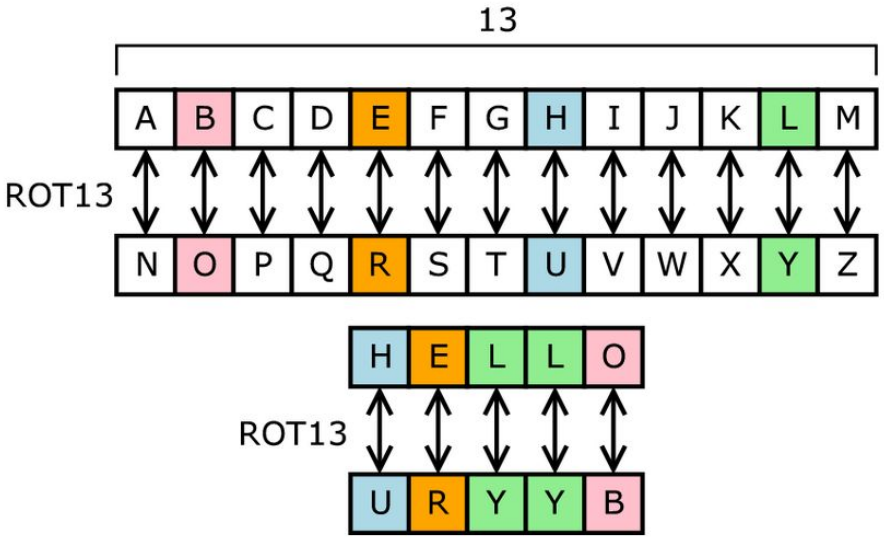
- **Абсолютно стойкие системы**
  - Ключ генерируется для каждого сообщения (каждый ключ используется один раз).
  - Ключ статистически надёжен (то есть вероятности появления каждого из возможных символов равны, символы в ключевой последовательности независимы и случайны).
  - Длина ключа равна или больше длины сообщения.
- **Достаточно стойкие системы**
  - вычислительная сложность полного перебора для данной системы
  - известные на данный момент слабости (уязвимости) системы и их влияние на вычислительную сложность.
- **Современные представления о длине ключа:**
  - 768 бит — для частных лиц;
  - 1024 бит — для коммерческой информации;
  - 2048 бит — для особо секретной информации.

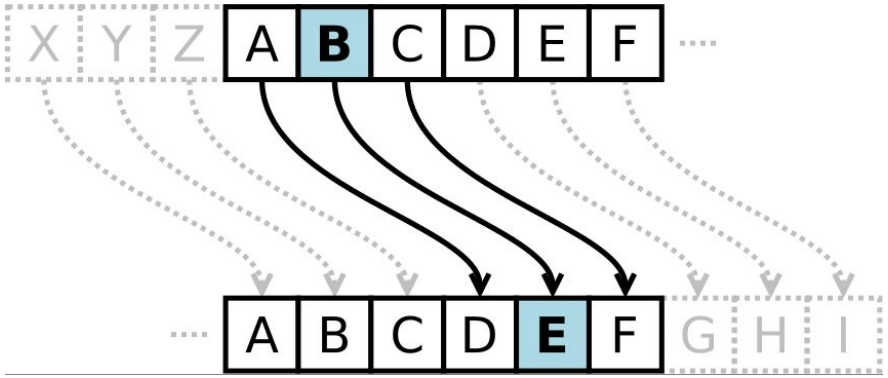
Шифрование данных

- Симметричные алгоритмы:
  - Алгоритм и ключ выбирается заранее и известен обеим сторонам.
  - Сохранение ключа в секретности – важная задача для установления и поддержки защищённого канала связи.
  - Проблема начальной передачи ключа (синхронизации ключей)
  - сложность управления ключами в большой сети: квадратичное возрастание числа пар ключей (генерация, передача, хранение, уничтожение).
    - 10 абонентов – 45 ключей
    - 100 абонентов – 4950
    - 1000 абонентов – 499500
  - Комбинированная (гибридная) криптографическая схема:
    - асимметричное шифрование – передача сеансового ключа
    - сеансового ключ симметричного шифрования обеспечивает обмен данными.

Шифрование данных

- Шифр простой замены – сопоставление каждой букве исходного сообщения единственной буквы шифротекста
  - Проблема – сохранение частоты встречаемости символов





Шифрование данных

- Омофоническая замена – шифр подстановки, при котором каждый символ открытого текста заменяется на один из нескольких символов шифралфавита, причём количество заменяющих символов для одной буквы пропорционально частоте этой буквы.
  - А → 1, 2, 3, 4, 5
  - Б → 6, 7, 8
  - Ъ → 9

Шифрование данных

16.	3 и	2 р	13 д
5 з	10 е	11 г	8 ю
9 С	6 ж	7 а	12 о
4 е	15 я	14 н	1 П

- **Магические квадраты**
  - Вписывание букв по номерам квадратов
- **Книжный шифр**
  - Обе стороны используют книгу одного издания и выпуска.
  - Шифр:
    - номер страницы
    - номер строки
    - номер буквы
- **Одноразовый блокнот**
  - Ключ – одноразовый блокнот
  - Операция – XOR
  - Стойкость – абсолютная

Шифрование данных

- Асимметричные алгоритмы:
  - Два ключа: открытый и закрытый,
  - **Открытый ключ** передаётся по открытому каналу и используется для шифрования сообщения и для проверки ЭЦП.
  - Для расшифровки сообщения и для генерации ЭЦП используется **закрытый ключ**.

Шифрование данных

- Несколько открытых ключей:
  - Сторона1 шифрует сообщение так, чтобы только сторона2 могла прочитать его
  - Сторона2 шифрует сообщение так, чтобы только сторона1 могла прочитать его

https://intsystem.org/security/asymmetric-encryption-how-it-work/

Удачи на экзамене