

**Ф.А. ГАЛИМЯНОВ, А.Ф. ГАЛИМЯНОВ**

**ВВЕДЕНИЕ В ИНФОРМАТИКУ**

**Учебное пособие**

**Казань – 2018**

**Издательство Казанского университета**

**УДК 004.032.26**

**ББК 73**

**Г15**

*Печатается по постановлению  
Редакционно-издательского совета  
Института вычислительной математики и информационных технологий  
Казанского (Приволжского) федерального университета;  
(протокол №1 от 18 октября 2018 г.)*

**Научный редактор**

кандидат педагогических наук, доцент Ч.Б. Миннегалиева

**Рецензенты:**

кандидат физико-математических наук,

доцент кафедры теории функций и приближений КФУ **Ю.Р. Агачев;**

кандидат технических наук,

с.н.с. научно-исследовательского института АН РТ «Прикладная семиотика»

**А.Р. Гатиятуллин**

**Галимянов Ф.А.**

**Г15 Введение в информатику: учеб. пособие /**

Ф.А. Галимянов, А.Ф. Галимянов. – Казань: Изд-во Казан. ун-та, 2018. –  
145 с.

Учебное пособие посвящено изложению основ информатики. Является отчасти повторением материалов из школьного курса. Изложены основы кодирования, технического и программного обеспечения компьютеров.

Адресовано, в первую очередь, студентам-бакалаврам 1 курса различных направлений, в программе которых есть курсы, связанные с компьютерными технологиями.

**УДК 004.032.26**

**ББК 73**

© Галимянов Ф.А., Галимянов А.Ф., 2018

© Издательство Казанского университета, 2018

# Оглавление

|  |    |
|--|----|
| 1. ОБЩИЕ СВЕДЕНИЯ.....   | 6  |
| 1.1. Понятие информации, ее виды и свойства .....  | 6  |
| 1.2. Измерение информации в технике .....  | 8  |
| 1.3. Измерение информации в теории информации (информация как снятая<br>неопределенность)..... | 9  |
| 2. КОДИРОВАНИЕ ЧИСЛОВОЙ ИНФОРМАЦИИ.....  | 9  |
| 2.1. Представление целых чисел .....   | 11 |
| 2.2. Представление целых положительных чисел.....  | 12 |
| 2.3. Представление целых отрицательных чисел.....  | 14 |
| 2.4. Перечисление чисел в целочисленной компьютерной арифметике .....                          | 17 |
| 2.5. Особенности реализации арифметических операций в конечном числе разрядов.....             | 19 |
| 3. ПРЕДСТАВЛЕНИЕ ВЕЩЕСТВЕННЫХ ЧИСЕЛ.....   | 20 |
| 3.1. Нормальная и нормализованная форма .....  | 22 |
| 3.2. Типы чисел с плавающей точкой (по IEEE 754) .....   | 22 |
| 3.3. Диапазон значений чисел с плавающей запятой.....  | 25 |
| 3.4. Особые значения чисел с плавающей точкой.....   | 26 |
| 3.5. Денормализованные числа .....   | 28 |
| 3.6. Действия с числами с плавающей запятой.....   | 31 |
| 3.7. Алгоритм получения представления вещественного числа в памяти ЭВМ .....                   | 33 |
| 4. КОДИРОВАНИЕ СИМВОЛОВ.....   | 35 |
| 4.1. Юникод. UTF-8.....  | 38 |
| 5. КОДИРОВАНИЕ ЗВУКОВОЙ ИНФОРМАЦИИ.....  | 38 |
| 6. КОДИРОВАНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ .....  | 42 |
| 6.1. Восприятие цвета человеком и животными.....   | 42 |
| 6.2. Цветовые модели RGB и CMYK.....   | 44 |
| 6.3. Цветовой куб.....   | 46 |
| 7. СЖАТИЕ (АРХИВАЦИЯ) РАЗЛИЧНЫХ ВИДОВ ИНФОРМАЦИИ.....  | 49 |
| 7.1. Способы обратимого сжатия информации .....  | 51 |
| 8. ЗАКОНОДАТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ О ЗАЩИТЕ КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ.....                 | 55 |
| 8.1. Уголовный Кодекс РФ о преступлениях в сфере компьютерной информации.....                  | 55 |
| 8.2. Требования к организации рабочих мест пользователей ПК.....                               | 59 |
| 9. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ .....                                       | 62 |
| 9.1. Процессор.....  | 62 |
| 9.2. Чипсет .....  | 65 |
| 9.3. Материнская плата .....   | 65 |

|  |     |
|--|-----|
| <b>9.4. Оперативная память</b> .....                                     | 70  |
| <b>9.5. Устройства хранения информации</b> .....                         | 72  |
| 10. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ .....  | 80  |
| <b>10.1. Рыночная классификация ПО</b> .....                             | 81  |
| <b>10.2. Операционные системы</b> .....                                  | 82  |
| <b>10.3. Файловые системы</b> .....                                      | 86  |
| 11. ОС MICROSOFT WINDOWS.....  | 94  |
| <b>11.1. Интерфейс Windows 10</b> .....                                  | 97  |
| 12. ЗАЩИТА ОТ ВИРУСОВ .....  | 99  |
| 13. АНТИВИРУСНЫЕ ПРОГРАММЫ.....  | 104 |
| 14. СОЗДАНИЕ И ПРОСМОТР СПЕЦИАЛЬНЫХ ФОРМАТОВ ДОКУМЕНТОВ .....            | 107 |
| 15. РАБОТА С ИНФОРМАЦИЕЙ В ИНТЕРНЕТЕ .....                               | 108 |
| <b>15.1. История Интернета</b> .....                                     | 109 |
| <b>15.2. Структура Интернета</b> .....                                   | 112 |
| <b>15.3. Адресация в Интернете</b> .....                                 | 117 |
| <b>15.4. Поиск информации в Интернете</b> .....                          | 122 |
| <b>15.5. Обеспечение конфиденциальности информации в Интернете</b> ..... | 124 |
| 16. ОСНОВЫ HTML. РАЗРАБОТКА WEB-САЙТА.....                               | 126 |
| <b>16.1. Web-сайты и Web-страницы</b> .....                              | 126 |
| <b>16.2. Вставка изображений</b> .....                                   | 128 |
| <b>16.3. Гиперссылки</b> .....   | 129 |
| <b>16.4. Панель навигации</b> .....                                      | 129 |
| <b>16.5. Ссылка на электронный адрес</b> .....                           | 130 |
| <b>16.6. Формы</b> .....   | 131 |
| 16.7. Переключатели.....   | 132 |
| <b>16.8. Флажки</b> .....  | 132 |
| <b>16.9. Поля списков</b> .....  | 133 |
| <b>16.10. Текстовая область</b> .....                                    | 133 |
| <b>16.11. Отправка данных из формы</b> .....                             | 133 |
| 17. ЗНАКОМСТВО С XML .....   | 134 |
| <b>17.1. Предназначение XML</b> .....                                    | 135 |
| <b>17.2. Создание XML-документов</b> .....                               | 137 |
| <b>17.3. Отображение XML-документов</b> .....                            | 138 |
| <b>17.4. SGML, HTML и XML</b> .....                                      | 139 |
| <b>17.5. Заменит ли XML HTML?</b> .....                                  | 140 |

|  |            |
|--|------------|
| <b>17.6. Стандартные XML-приложения.....</b>                         | <b>141</b> |
| <b>17.7. XML-приложения, повышающие качество XML-документов.....</b> | <b>142</b> |
| <b>17.8. Реальное использование XML .....</b>                        | <b>142</b> |
| <b>СПИСОК ЛИТЕРАТУРЫ .....</b>                                       | <b>145</b> |

# 1. ОБЩИЕ СВЕДЕНИЯ

## 1.1. Понятие информации, ее виды и свойства

В Федеральном законе Российской Федерации «**Об информации, информатизации и защите информации**» ([http://www.bnti.ru/dbtexts/analmat/06\\_2003/licenz/1.htm](http://www.bnti.ru/dbtexts/analmat/06_2003/licenz/1.htm)) дается следующее определение этого термина: «информация – сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления».

Толковый словарь русского языка Ожегова приводит 2 определения слова **информация**:

- 1) Сведения об окружающем мире и протекающих в нем процессах, воспринимаемые человеком или специальным устройством.
- 2) Сообщения, осведомляющие о положении дел, о состоянии чего-нибудь. (Научно-техническая и газетная информации, средства массовой информации – печать, радио, телевидение, кино).

В **информатике** наиболее часто используется следующее определение этого термина:

**Информация** – это осознанные сведения об окружающем мире, которые являются объектом хранения, преобразования, передачи и использования. *Сведения* – это знания, выраженные в сигналах, сообщениях, известиях, уведомлениях и т.д.

Информация и ее свойства являются объектом исследования целого ряда научных дисциплин, таких как **теория информации** (математическая теория систем передачи информации), **кибернетика** (наука о связи и управлении в машинах и животных, а также в обществе и человеческих существах), **семиотика** (наука о знаках и знаковых системах), **теория массовой коммуникации** (исследование средств массовой информации и их влияния на общество), **информатика, соционика** (теория информационного метаболизма индивидуальной и социальной психики), **информодинамика** (наука об

открытых информационных системах), **информациология** (наука о получении, сохранении и передаче информации для различных множеств объектов) и т.д.

Основные **виды информации** по ее форме представления, способам ее кодирования и хранения, что имеет наибольшее значение для информатики, это:

– **графическая** или **изобразительная** – первый вид, для которого был реализован способ хранения информации об окружающем мире в виде наскальных рисунков, а позднее в виде картин, фотографий, схем, чертежей на бумаге, холсте, мраморе и др. материалах, изображающих картины реального мира;

– **звуковая** – мир вокруг нас полон звуков и задача их хранения и тиражирования была решена с изобретением звукозаписывающих устройств в 1877 г. (см., например, историю звукозаписи на сайте – <http://radiomuseum.ur.ru/index9.html>); ее разновидностью является **музыкальная** информация – для этого вида был изобретен способ кодирования с использованием специальных символов, что делает возможным хранение ее аналогично графической информации;

– **текстовая** – способ кодирования речи человека специальными символами – буквами, причем разные народы имеют разные языки и используют различные наборы букв для отображения речи; особенно большое значение этот способ приобрел после изобретения бумаги и книгопечатания;

– **числовая** – количественная мера объектов и их свойств в окружающем мире; особенно большое значение приобрела с развитием торговли, экономики и денежного обмена; аналогично текстовой информации для ее отображения используется метод кодирования специальными символами – цифрами, причем системы кодирования (счисления) могут быть разными;

- **видеоинформация** – способ сохранения движущихся картин окружающего мира, появившийся с изобретением кино.

Существуют также виды информации, для которых до сих пор не изобретено способов их кодирования и хранения – это **тактильная** ин-

формация, передаваемая ощущениями, **органолептическая**, передаваемая запахами и вкусами и другие виды, для которых современная наука даже не нашла признанных всеми терминов определения (например, экстрасенсорная информация)

Создателем *общей теории информации* и основоположником цифровой связи считается Клод Шеннон (Claude Shannon). Всемирную известность ему принес фундаментальный труд 1948 года – «Математическая теория связи» (A Mathematical Theory of Communication), в котором впервые обосновывается возможность применения двоичного кода для передачи информации

С точки зрения информатики наиболее важными представляются следующие *общие качественные свойства*: объективность, достоверность, полнота, точность, актуальность, полезность, ценность, своевременность, понятность, доступность, краткость и пр.

Социальная (общественная) информация обладает еще и дополнительными свойствами:

имеет семантический (смысловой) характер, т. е. понятийный, так как именно в понятиях обобщаются наиболее существенные признаки предметов, процессов и явлений окружающего мира.

имеет языковую природу (кроме некоторых видов эстетической информации, например изобразительного искусства). Одно и то же содержание может быть выражено на разных естественных (разговорных) языках, записано в виде математических формул и т. д.

## **1.2. Измерение информации в технике**

В технике (теория кодирования и передачи сообщений) под количеством информации понимают количество кодируемых, передаваемых или хранимых символов. При этом используют простой способ определения количества информации, который может быть назван объемным. Он основан на подсчете числа символов в сообщении, т. е. связано с его длиной и не учитывает содержания.



В вычислительной технике применяются две стандартные единицы измерения: бит (binary digit) и байт (byte).

**Бит** – минимальная единица измерения информации – величина, которая может принимать одно из двух значений (в математическом представлении 0 или 1). **Байт** – единица количества информации в системе СИ. Байт – восьмибитный двоичный код, с помощью которого наиболее часто представляют один символ текста. **Информационный объем сообщения** (информационная емкость сообщения) – количество информации в сообщении, измеренное в битах, байтах или производных единицах (Кбайтах, Мбайтах и т. д.).

### 1.3. Измерение информации в теории информации (информация как снятая неопределенность)

В теории информации количеством информации называют числовую характеристику сигнала, не зависящую от его формы и содержания и характеризующую неопределенность, которая исчезает после получения сообщения в виде данного сигнала – в этом случае количество информации зависит от вероятности получения сообщения о том или ином событии. Оценка количества информации основывается на законах теории информации. Клод Шеннон ввел понятие информационной энтропии – меры хаотичности информации, определяющей неопределенность появления какого-либо символа первичного алфавита. При отсутствии информационных потерь численно равна количеству информации на символ передаваемого сообщения.

**Информационная энтропия** для независимых случайных событий  $x$  с  $n$  возможными состояниями (от 1 до  $n$ ) рассчитывается по формуле Шеннона ( $p(i)$  – вероятность  $i$ -го события).

$$H(x) = - \sum_{i=1}^n p(i) \log_2 p(i)$$

## 2. КОДИРОВАНИЕ ЧИСЛОВОЙ ИНФОРМАЦИИ

С конца XX века, века компьютеризации, человечество ежедневно пользуется двоичной системой счисления, так как вся информация,

обрабатываемая современными компьютерами, хранится в них в двоичном виде.

Каждый регистр арифметического устройства компьютера, каждая ячейка памяти представляет собой физическую систему, состоящую из некоторого числа однородных элементов, обладающих двумя устойчивыми состояниями, одно из которых соответствует нулю, а другое — единице. Каждый такой элемент служит для изображения одного из разрядов двоичного числа. Именно поэтому каждый элемент ячейки называют *разрядом*.

Ячейка из  $k$  разрядов

|                  |  |  |       |  |  |  |               |
|------------------|--|--|-------|--|--|--|---------------|
| К-1-ый<br>разряд |  |  | ..... |  |  |  | 0-й<br>разряд |
|------------------|--|--|-------|--|--|--|---------------|

Вы знаете, что вычислительная техника возникла как средство автоматизации вычислений, именно поэтому первые компьютеры назывались ЭВМ — электронно-вычислительные машины. Сегодня компьютеры обрабатывают различные виды информации: числовую, текстовую, звуковую, графическую. Однако современный компьютер может хранить и обрабатывать только дискретную информацию. Поэтому любой вид информации, подлежащий компьютерной обработке, тем или иным способом должен быть закодирован с помощью конечной последовательности целых чисел, которая затем переводится в двоичный вид для хранения в компьютере.

Мы рассмотрим, каким образом решается проблема преобразования исходной информации в компьютерное представление для каждого вида информации. Будет показано, насколько точно компьютерное представление отражает исходную информацию, причем слово “точно” здесь понимается не только в математическом смысле — более реалистично переданы оттенки цвета на мониторе, воспроизводимая музыка наиболее приближена к естественному звучанию музыкальных инструментов или голосу человека и т.д. Задача перевода естественной информации в компьютерную называется

задачей *дискретизации*, или *квантования*. Эту задачу необходимо решать для всех видов информации. Способы дискретизации для разных видов информации различны, но подходы к решению этой задачи построены на одинаковых принципах.

### 2.1. Представление целых чисел

Любое целое число можно рассматривать как вещественное, но с нулевой дробной частью, т.е. можно было бы ограничиться представлением в компьютере вещественных чисел и реализацией арифметических действий над ними. Однако для эффективного использования памяти, повышения скорости выполнения вычислений и введения операции деления нацело с остатком целые числа представляются специально для них предназначенными способами.

Введение специальных способов представления целых чисел оправдано тем, что достаточно часто в задачах, решаемых с помощью компьютера, многие действия сводятся к операциям над целыми числами. Например, в задачах экономического характера данными служат количества акций, сотрудников, деталей, транспортных средств и т.д., по своему смыслу являющиеся целыми числами. Целые числа используются и для обозначения даты и времени, и для нумерации различных объектов: элементов массивов, записей в базах данных, машинных адресов и т.п.

Для компьютерного представления целых чисел обычно используются несколько различных способов представления, отличающихся друг от друга количеством разрядов и наличием или отсутствием знакового разряда. *Беззнаковое* представление можно использовать только для неотрицательных целых чисел, отрицательные числа можно представлять только в *знаковом* виде.

При беззнаковом представлении все разряды ячейки отводятся под само число. При представлении со знаком самый старший (левый) разряд отводится под знак числа, остальные разряды — под собственно число. Если число положительное, то в знаковый разряд помещается 0, если число

отрицательное — 1. Очевидно, в ячейках одного и того же размера без знака можно представить больший диапазон целых чисел, чем со знаком. Например, в одном байте (8 разрядов) можно записать положительные числа от 0 до 255, а со знаком — только до 127. Поэтому, если известно заранее, что некоторая числовая величина всегда является неотрицательной, то выгоднее рассматривать ее как беззнаковую.

Говорят, что целые числа в компьютере хранятся в формате с *фиксированной запятой*.

## 2.2. Представление целых положительных чисел

Для получения компьютерного представления беззнакового целого числа в  $k$ -разрядной ячейке памяти достаточно перевести его в двоичную систему счисления и дополнить полученный результат слева нулями до  $k$  разрядов. Понятно, что существует ограничение на значения чисел, которые мы можем записать в  $k$ -разрядную ячейку.

Максимальное значение целого неотрицательного числа достигается в случае, когда во всех разрядах ячейки хранятся единицы. Для  $k$ -разрядного представления оно будет равно

$2^k - 1$ . Минимальное число соответствует  $k$  нулям, хранящимся в  $k$  разрядах ячейки, и оно всегда равно нулю. Ниже приведены максимальные значения для беззнаковых целых  $k$ -разрядных чисел:

| Количество разрядов | Максимальное значение                 |
|---------------------|---------------------------------------|
| 8                   | 255 ( $2^8 - 1$ )                     |
| 16                  | 65535 ( $2^{16} - 1$ )                |
| 32                  | 4294967295 ( $2^{32} - 1$ )           |
| 64                  | 18446744073709551615 ( $2^{64} - 1$ ) |

При знаковом представлении целых чисел возникают такие понятия, как *прямой*, *обратный* и *дополнительный коды*.

**Определение 1.** Представление числа в привычной для человека форме “знак”—“величина”, при которой старший разряд ячейки отводится

под знак, остальные  $k - 1$  разрядов — под цифры числа в двоичной системе счисления, называется **прямым кодом**.

Положительные целые числа представляются в компьютере с помощью прямого кода. Прямой код отрицательного целого числа отличается от прямого кода соответствующего положительного числа содержимым знакового разряда. Например, прямые коды двоичных чисел  $11001_2$  и  $-11001_2$  для восьмиразрядной ячейки равны  $00011001$  и  $10011001$  соответственно. Но вместо прямого кода для представления отрицательных целых чисел в компьютере используется дополнительный код .

Отметим, что максимальное значение положительного числа, которое можно записать в знаковом представлении в  $k$ -разрядах, равно  $A=2^{k-1} - 1$ , что практически в два раза меньше максимального значения беззнакового представления в тех же  $k$  разрядах.

*Задание.* Подсчитайте максимальное значение положительного числа в восьмиразрядном и шестнадцатиразрядном знаковом способе представления чисел.

*Решение.* Максимальное значение положительного числа в 8 битах равно  $127 (2^7 - 1)$ , в 16 битах —  $32767 = (2^{15} - 1)$ .

**Пример 1.** Число  $53 = 110101_2$  в восьмиразрядном представлении имеет вид

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|

Это же число 53 в 16 разрядах будет записано следующим образом:

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

**Пример 2.** Для числа  $200 = 11001000_2$  представление в восьми разрядах со знаком невозможно, так как максимальное допустимое число в таком представлении равно 127, а в беззнаковом восьмиразрядном представлении оно имеет вид

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

### 2.3. Представление целых отрицательных чисел

Для представления в компьютере целых отрицательных чисел используют *дополнительный код*  $y$  который позволяет заменить арифметическую операцию вычитания операцией сложения, что существенно увеличивает скорость вычислений. Прежде чем вводить определение дополнительного кода, сделаем следующее важное замечание:

***В  $k$ -разрядной целочисленной компьютерной арифметике  $2^k = 0$ .***

Объяснить это можно тем, что двоичная запись числа  $2^k$  состоит из одной единицы и  $k$  нулей, а в ячейке из  $k$  разрядов может уместиться только  $k$  цифр, в данном случае только  $k$  нулей. В таком случае говорят, что значащая единица вышла за пределы разрядной сетки.

**Определение 2.**  $k$ -разрядный дополнительный код отрицательного числа  $m$  — это запись в  $k$  разрядах положительного числа  $2^k - |m|$ , где  $|m|$  — модуль отрицательного числа  $m$ ,  $|m| < 2^{k-1}$ .

Разберемся, что и до чего данный код дополняет. Дополнительный код отрицательного числа  $m$  — это дополнение модуля этого числа до  $2^k$  (т.е. до нуля в  $k$ -разрядной арифметике):  $(2^k - m) + |m| - 2^k = 0$ .

**Алгоритм получения дополнительного  $k$ -разрядного кода отрицательного числа:**

- модуль числа представить прямым кодом в  $k$  двоичных разрядах;
- значения всех разрядов *инвертировать* (все нули заменить на единицы, а единицы на нули), получив, таким образом,  $k$ -разрядный *обратный код* исходного числа;
- к полученному обратному коду, трактуемому как  $k$ -разрядное неотрицательное двоичное число, прибавить единицу.

Обратный код является дополнением до числа  $2^k - 1$ , состоящего из  $k$  двоичных единиц. Поэтому прибавление единицы к инвертированному числу позволяет получить его искомым дополнительным кодом.

**Пример 3.** Получим дополнительный код числа  $-52$  для 8- и 16-разрядной ячеек.

Для 8-разрядной ячейки:

0011 0100 — число  $| -52 | = 52$  в прямом коде;

1100 1011 — число  $-52$  в обратном коде;

1100 1100 — число  $-52$  в дополнительном коде

для 16-разрядной ячейки:

0000 0000 0011 0100 — прямой код числа  $| -52 |$ ;

**1111 1111 1100 1011** — обратный код числа  $-52$ ;

1111 1111 1100 1100 — дополнительный код числа  $-52$ .

Какое минимальное отрицательное число можно записать в  $k$  разрядах?

Описанный выше алгоритм получения дополнительного кода для отрицательного числа знаковый бит в левом разряде образует автоматически при  $|m| < 2^{k-1}$ . Если же  $2^{k-1} < |m| < 2^k$ , то попытка реализации алгоритма приведет к тому, что в левом разряде будет находиться цифра 0, соответствующая компьютерному представлению положительных чисел.

Представим значение  $2^k - |m|$  в следующем виде:

$$2^k - |m| = 2^{k-1} + (2^{k-1} - |m|).$$

Здесь первое слагаемое  $2^{k-1}$  соответствует единице в левом знаковом разряде. Т.е. при представлении отрицательного числа  $m$  дополнительным кодом в самом левом (знаковом) разряде записан знак отрицательного числа (единица), а в остальных разрядах записывается число  $2^{k-1} - |m|$ . Следовательно, минимальное отрицательное (максимальное по модулю) число  $m$ , которое можно представить в  $k$  разрядах, равно  $-2^{k-1}$ .

*Задание.* Постройте дополнительный восьмиразрядный код для чисел  $-128$  (минимальное представимое число в 8 разрядах),  $-127$  (число, обратное максимально представимому положительному числу) и  $0$ .

**Решение**

| Число              | — 128     | -127      | -0        |
|--------------------|-----------|-----------|-----------|
| Прямой код модуля  | 1000 0000 | 0111 1111 | 0000 0000 |
| Обратный код       | 0111 1111 | 1000 0000 | 1111 1111 |
| Дополнительный код | 1000 0000 | 1000 0001 | 0000 0000 |

Отметим, что для числа  $-128$  прямой код совпадает с дополнительным, а дополнительный код числа  $-0$  совпадает с обычным нулем. При преобразовании обратного кода для числа  $-0$  в его дополнительный код правила обычной двоичной арифметики нарушаются, а именно,

$$1111\ 1111_2 + 1 = 1\ 0000\ 0000_2 = 2^k = 0.$$

Восстановить модуль исходного десятичного отрицательного числа по его дополнительному коду можно двумя способами:

провести обратную цепочку преобразований: вычесть единицу из дополнительного кода, инвертировать полученный код и перевести полученное двоичное представление числа в десятичное;

или же по приведенному выше алгоритму построить дополнительный код для имеющегося дополнительного кода искомого числа и представить результат в десятичной системе счисления.

**Пример 4.** Получим десятичное значение числа по его дополнительному коду  $10010111_2$ .

*Способ 1.*

из дополнительного кода вычтем 1:

$$10010111 - 1 = 10010110 \text{ (получили обратный код);}$$

инвертируем полученный код:  $01101001$  (получили модуль отрицательного числа);

переведем полученное двоичное значение в десятичную систему счисления:

$$01101001_2 = 2^6 + 2^5 + 2^3 + 1 = 64 + 32 + 8 + 1 = 105.$$

*Ответ:*  $-105$ .

*Способ 2.*



инвертируем имеющийся дополнительный код: 01101000;

прибавим к результату 1:  $01101000 + 1 = 01101001$  (получили модуль отрицательного числа);

переведем полученное двоичное значение в десятичную систему счисления:

$$01101001_2 = 2^6 + 2^5 + 2^3 + 1 = 64 + 32 + 8 + 1 = 105.$$

*Ответ:* —105.

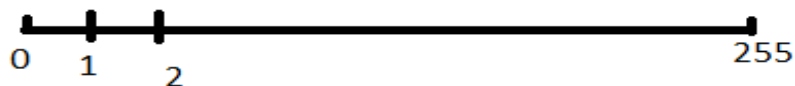
Целые числа со знаком, представимые в  $k$  разрядах, принадлежат диапазону

$[-2^{k-1}, 2^{k-1} - 1]$ , который не является симметричным относительно 0. Это следует учитывать при программировании. Если, например, изменить нить знак у наибольшего по модулю отрицательного числа, то полученный результат окажется уже не представимым в том же числе разрядов.

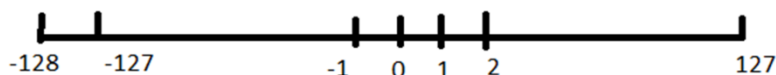
Ниже приведены значения границ диапазонов для знаковых представлений в ячейках с различной разрядностью

#### 2.4. Перечисление чисел в целочисленной компьютерной арифметике

Мы выяснили, что в  $k$ -разрядном знаковом или беззнаковом представлении целых чисел количество представимых чисел ограничено и зависит от  $k$ . Значения представимых чисел можно перечислить по возрастанию. Например, для восьмиразрядного беззнакового представления допустимые значения располагаются на отрезке следующим образом:

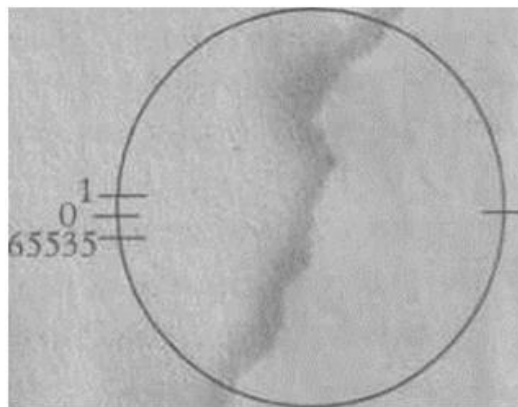
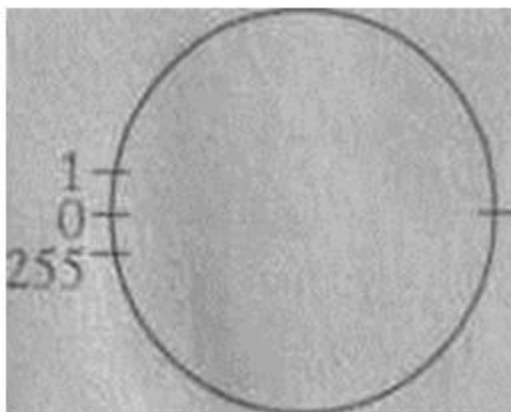


Для восьмиразрядного знакового представления:



Однако в  $k$ -разрядной компьютерной арифметике эти отрезки можно замкнуть в кольцо и перечислять значения соответствующего представления по кругу.

Все целые числа любого  $k$ -разрядного представления можно зафиксировать на кольце, состоящем из расположенных по порядку констант, причем рядом с максимальным значением в том или ином представлении будет находиться минимальное, например: 8 -разрядное беззнаковое.



Действительно, результатом прибавления единицы в арифметике с фиксированным количеством разрядов является следующее по часовой стрелке за исходным значение в кольце, а результатом вычитания единицы — следующее значение по ходу против часовой стрелки. Данный факт очевиден, кроме случаев перехода от максимально допустимого числа к минимальному при прибавлении единицы и от минимального к максимальному при вычитании.

Покажем это. Рассмотрим вначале беззнаковые представления. Максимально допустимое число в  $k$ -разрядном беззнаковом представлении равно  $2^k - 1$  и состоит из  $k$  единиц в двоичном представлении. При прибавлении к такому числу единицы мы получаем  $2^k$ , что, в свою очередь, соответствует нулю в  $k$ -разрядной арифметике. Вычитание же из нуля единицы является обратной операцией: вычитание единицы можно рассматривать и как прибавление к нулю минус единицы, по правилам же  $k$ -разрядной арифметики минус единица — это дополнение единицы до  $2^k$ , то есть число, которое при сложении с единицей даст  $2^k$ . Очевидно, что таковым является число  $2^k - 1$ , состоящее из  $k$  двоичных единиц, значит, в

беззнаковом представлении  $0 + (-1) = 2^k - 1$ , что и равно максимально представимому числу.

В знаковом  $k$ -разрядном представлении максимально допустимым является число  $2^{k-1} - 1$  ( $k-1$  двоичная единица). При увеличении такого числа на 1 мы получаем  $2^{k-1}$  или единицу в знаковом бите и нули в остальных, что соответствует отрицательному числу с максимально возможным модулем. Вычитание единицы из такого числа можно проводить по правилам обычной двоичной арифметики:  $10...0-1 = 01...1 = 2^k - 1$ .

Достоинствами рассмотренных форматов представления целых чисел являются простота и наглядность их представления, а также простота реализации алгоритмов арифметических операций (например, вычитание, благодаря использованию дополнительного кода для представления отрицательных чисел, сводится к сложению)

## 2.5. Особенности реализации арифметических операций в конечном числе разрядов

Целочисленная арифметика в ограниченном числе разрядов несколько отличается от обычной. При выполнении арифметических действий в целочисленной  $k$ -разрядной арифметике возможно возникновение следующих ситуаций, незнание которых может привести к неверному результату при выполнении верных алгоритмов:

старшие (левые) цифры результата, выходящие за отведенное количество разрядов, оказываются утерянными;

при сложении или умножении двух положительных чисел, имеющих знаковое представление, можно получить отрицательное число, если в результате сложения в левом знаковом бите окажется единица.

**Пример 5.** Выполним сложение  $100_{10}$  и  $51_{10}$  в знаковом восьмиразрядном представлении.

В этом компьютерном представлении числа имеют следующий вид:  $100_{10} = 0110\ 0100_2$  и  $51_{10} = 0011\ 0011_2$ . При сложении этих чисел получим

1001 0111<sub>2</sub>. Самая левая единица (знаковый разряд) указывает на то, что в 8 разрядах записано отрицательное число.

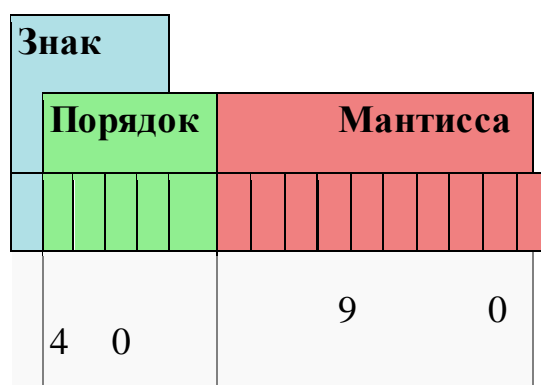
Так как все отрицательные числа в машине представляются дополнительным кодом, то для восстановления десятичного значения этого отрицательного числа надо воспользоваться алгоритмом получения исходного числа по его дополнительному коду (см. пример 4). В результате получим число —105.

Таким образом, в восьмиразрядной знаковой арифметике  $100 + 51 = -105$ , то есть при сложении двух положительных чисел, используя восьмиразрядное знаковое представление, мы получили отрицательное число!

### 3. ПРЕДСТАВЛЕНИЕ ВЕЩЕСТВЕННЫХ ЧИСЕЛ

Вещественные числа обычно представляются в виде чисел с плавающей запятой. Числа с плавающей запятой — один из возможных способов представления действительных чисел, который является компромиссом между точностью и диапазоном принимаемых значений, его можно считать аналогом экспоненциальной записи чисел, но только в памяти компьютера.

Число с плавающей запятой состоит из набора отдельных двоичных разрядов, условно разделенных на так называемые **знак** (англ. *sign*), **порядок** (англ. *exponent*) и **мантиссу** (англ. *mantis*). В наиболее распространённом формате (стандарт IEEE 754) число с плавающей запятой представляется в виде набора битов, часть из которых кодирует собой мантиссу числа, другая часть — показатель степени, и ещё один бит используется для указания знака числа (0 — если число положительное, 1 — если число отрицательное). При этом порядок записывается как целое число в коде со сдвигом, а мантисса — в нормализованном виде, своей дробной частью в двоичной системе счисления. Вот пример такого числа из 16 двоичных разрядов:



Знак — один бит, указывающий знак всего числа с плавающей точкой. Порядок и мантисса — целые числа, которые вместе со знаком дают представление числа с плавающей запятой в следующем виде:

$(-1)^S \times M \times B^E$ , где  $S$  — знак,  $B$  — основание,  $E$  — порядок, а  $M$  — мантисса. Десятичное число, записываемое как  $ReE$ , где  $R$  — число в полуинтервале  $[1; 10)$ ,  $E$  — степень, в которой стоит множитель 10; в нормализованной форме модуль  $R$  будет являться мантиссой, а  $E$  — порядком, а  $S$  будет равно 1 тогда и только тогда, когда  $R$  принимает отрицательное значение. Например, в числе  $-2435e9$

- $S = 1$
- $B = 10$
- $M = 2435$
- $E = 9$

Порядок также иногда называют **экспонентой** или просто **показателем степени**.

При этом лишь некоторые из вещественных чисел могут быть представлены в памяти компьютера точным значением, в то время как остальные числа представляются приближёнными значениями.

Более простым вариантом представления вещественных чисел является вариант с фиксированной точкой, когда целая и вещественная части хранятся отдельно. Например, на целую часть отводится всегда  $X$  бит и на дробную отводится всегда  $Y$  бит. Такой способ в архитектурах процессоров не присутствует. Отдаётся предпочтение числам с плавающей запятой, как компромиссу между диапазоном допустимых значений и точностью.

### 3.1. Нормальная и нормализованная форма

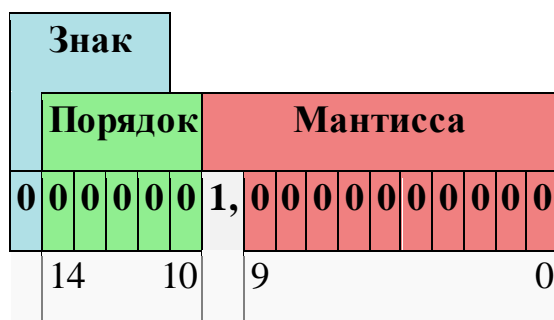
**Нормальной формой** (англ. *normal form*) числа с плавающей запятой называется такая форма, в которой мантисса (без учёта знака) в десятичной системе находится на полуинтервале  $[0; 1)$ . Такая форма записи имеет недостаток: некоторые числа записываются неоднозначно (например, 0,0001 можно записать в разных формах —  $0,0001 \cdot 10^0 = 0,0001 \cdot 10^0 = 0,001 \cdot 10^{-1} = 0,01 \cdot 10^{-2}$ , поэтому распространена также другая форма записи — **нормализованная** (англ. *normalized*), в которой мантисса десятичного числа принимает значения от 1 (включительно) до 10 (не включительно), а мантисса двоичного числа принимает значения от 1 (включительно) до 2 (не включительно). То есть в мантиссе слева от запятой до применения порядка находится ровно один знак. В такой форме любое число (кроме 0) записывается единственным образом. Ноль же представить таким образом невозможно, поэтому стандарт предусматривает специальную последовательность битов для задания числа 0 (а заодно и некоторых других полезных чисел, таких как  $-\infty$  и  $+\infty$ ). Так как старший двоичный разряд (целая часть) мантиссы вещественного числа в нормализованном виде всегда равен «1», то его можно не записывать, сэкономив таким образом один бит, что и используется в стандарте IEEE 754. В позиционных системах счисления с основанием большим, чем 2 (в троичной, четверичной и др.), этого замечательного свойства нет (ведь целая часть там может быть не только единицей).

### 3.2. Типы чисел с плавающей точкой (по IEEE 754)

Число половинной точности (*Binary16, Half precision*)

**Число половинной точности** — компьютерный формат представления чисел, занимающий в памяти половину машинного слова (в случае 32-битного компьютера — 16 бит или 2 байта). В силу невысокой точности этот формат представления чисел с плавающей запятой обычно используется в видеокάρтах,

где небольшой размер и высокая скорость работы важнее точности вычислений.



Порядок записан со сдвигом  $-15$ . То есть чтобы получить актуально значение порядка нужно вычесть из него сдвиг. Сдвиг можно получить по формуле  $2^{b-1} - 1$ , где  $b$  — число бит, отведенное на хранение порядка (в случае числа половинной точности  $b = 5$ ).

### Ограничения точности

- Целые от нуля до 2048 передаются как есть.
- Целые от 2049 до 4096 округляются к ближайшему чётному целому.
- Целые от 4097 до 8192 округляются до ближайшего целого, делящегося нацело на четыре.
- Целые от 8193 до 16384 округляются до ближайшего целого, делящегося на восемь.
- Целые от 16385 до 32768 округляются до ближайшего целого, делящегося на шестнадцать.
- Целые от 32769 до 65535 округляются до ближайшего целого, делящегося на тридцать два.

### Число одинарной точности (*Binary32, Single precision, float*)

**Число одинарной точности** — компьютерный формат представления чисел, занимающий в памяти одно машинное слово (в случае 32-битного компьютера — 32 бита или 4 байта). Используется для работы с вещественными числами везде, где не нужна очень высокая точность.







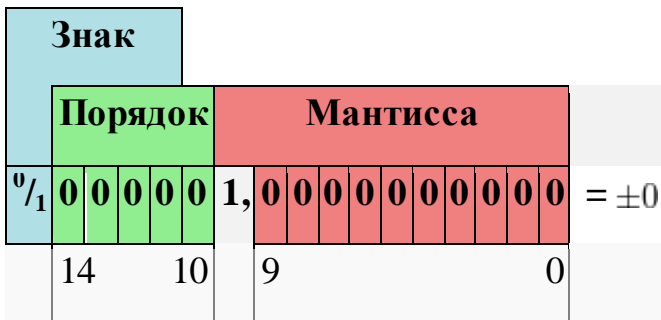
чисел. К ним относятся ноль, значения NaN (Not a Number, "не число", получается как результат операций типа деления нуля на ноль) и  $\pm\infty$ .  
 Данная таблица только лишь примерно указывает границы допустимых значений, без учета возрастающей погрешности с ростом абсолютного значения и существования [денормализованных чисел](#).

| Название в IEEE 754 | Название типа переменной в Си   | Диапазон значений                                 | Бит в мантиссе | Бит на переменную |
|---------------------|---|---|----------------|-------------------|
| Half precision      | -   | $6,10 \times 10^{-5}$ .. $65504$                  | 11             | 16                |
| Single precision    | float   | $3,4 \times 10^{-38}$ .. $3,4 \times 10^{38}$     | 23             | 32                |
| Double precision    | double  | $1,7 \times 10^{-308}$ .. $1,7 \times 10^{308}$   | 53             | 64                |
| Extended precision  | На некоторых архитектурах (например в сопроцессоре Intel) long double | $3,4 \times 10^{-4932}$ .. $3,4 \times 10^{4932}$ | 65             | 80                |

### 3.4. Особые значения чисел с плавающей точкой

#### Ноль (со знаком)

Как уже было оговорено выше, в нормализованной форме числа с плавающей точкой невозможно представить ноль. Поэтому для его представления зарезервированы специальные значения мантиссы и порядка — число считается нулём, если все его биты, кроме знакового, равны нулю. При этом в зависимости от значения бита знака ноль может быть как положительным, так и отрицательным.



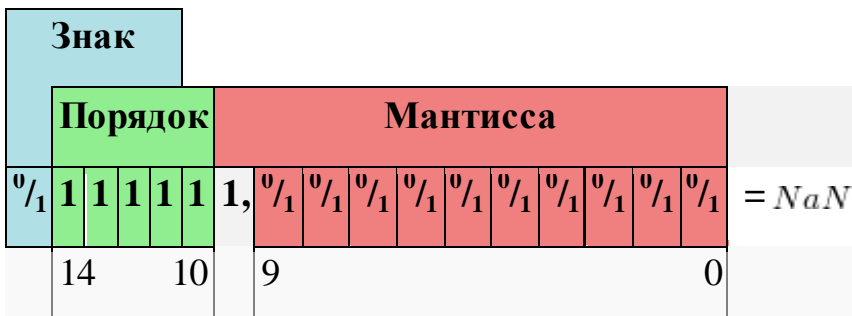
### Арифметика нуля со знаком

Арифметика отрицательного нуля аналогична таковой для любого отрицательного числа и понятна интуитивно. Вот несколько примеров:

- $\frac{-0}{|x|} = -0$  (если  $x \neq 0$ )
- $(-0) \cdot (-0) = +0$
- $|x| \cdot (-0) = -0$
- $x + (\pm 0) = x$
- $(-0) + (-0) = -0$
- $(+0) + (+0) = +0$
- $\frac{-0}{-\infty} = +0$
- $\frac{|x|}{-0} = -\infty$  (если  $x \neq 0$ )

### Неопределенность (NaN)

NaN — это аббревиатура от фразы "not a number". NaN является результатом арифметических операций, если во время их выполнения произошла ошибка (примеры см. ниже). В IEEE 754 NaN представлен как число, в котором все двоичные разряды порядка — единицы, а мантисса не нулевая.



Любая операция с NaN возвращает NaN. При желании в мантиссу можно записывать информацию, которую программа сможет интерпретировать. Стандартом это не оговорено и мантисса чаще всего игнорируется.

### Как можно получить NaN?

- $\infty + (-\infty) = NaN$

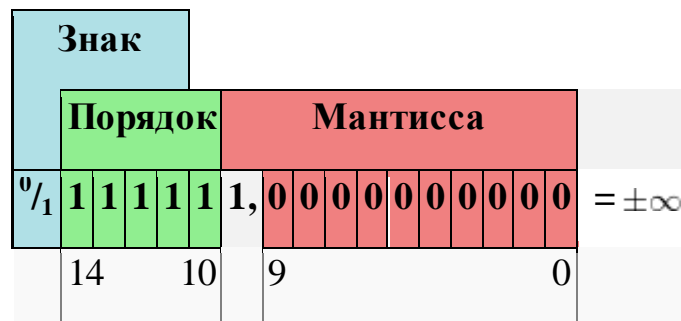
- $0 \times \infty = NaN$
- $\frac{\pm 0}{\pm 0} = \frac{\pm \infty}{\pm \infty} = NaN$
- $\sqrt{x} = NaN$ , где  $x < 0$

Есть и другие способы получения NaN

По определению  $NaN \neq NaN$ , поэтому, для проверки значения переменной нужно просто сравнить ее с собой.

### Бесконечности

В число с плавающей запятой можно записать значение  $+\infty$  или  $-\infty$ . Как и нули со знаком, бесконечности позволяют получить хотя бы близкий к правильному результат вычисления в случае переполнения. Согласно стандарту IEEE 754 число с плавающей запятой считается равным бесконечности, если все двоичные разряды его порядка — единицы, а мантисса равна нулю. Знак бесконечности определяется знаковым битом числа.



Получить бесконечность можно при переполнении и при делении ненулевого

$$\frac{x}{0} = \begin{cases} +\infty, & x > 0; \\ NaN, & x = 0; \\ -\infty, & x < 0. \end{cases}$$

### 3.5. Денормализованные числа

**Денормализованные числа** (англ. *denormalized/subnormal numbers*) - это способ увеличить количество представимых числом с плавающей запятой значений около нуля, дабы повысить точность вычислений. Каждое значение денормализованного числа меньше самого маленького

**нормализованного** ("обычного") значения числа с плавающей запятой.

Согласно стандарту, если порядок равен своему минимальному значению (все его биты — нули, а истинное значение порядка равно его сдвигу) и все биты

мантиссы равны нулю, то это  $\pm 0$ . Если же мантисса не равна нулю, то это число с порядком, на единицу большим минимального (все биты порядка, кроме младшего — нули) и данной мантиссой, **целая часть которой считается равной нулю, а не единице.**

То есть число с плавающей запятой, при учете вышесказанного, можно задать следующим образом:

- $(-1)^s \times 1, M \times 2^E$ , если  $E_{min} \leq E \leq E_{max}$  (нормализованное число)
- $(-1)^s \times 0, M \times 2^{E_{min}}$ , если  $E = E_{min} - 1$  (денормализованное число)

Где  $s$  — бит знака,  $M$  — последовательность битов мантиссы,  $E$  — значение порядка (с учетом сдвига),  $E_{min}$  — минимальное значение порядка, используемое для записи чисел ( $1$  — сдвиг),  $E_{min} - 1$  — минимальное значение порядка, которое он в принципе может принять (все биты нули,  $0$  — сдвиг).

Хоть денормализованные числа и позволяют бороться с погрешностями и обрабатывать очень маленькие значения, за эти возможности приходится дорого платить. Ввиду сложности денормализованные числа крайне редко реализуют на аппаратном уровне - вместо этого используются программные реализации, работающие значительно медленнее.

В современных процессорах обработка денормализованных чисел происходит в десятки раз медленнее, чем обработка нормализованных чисел. Ниже приведена часть таблицы из статьи Isaac Dooley, Laxmikant Kale "Quantifying the Interference Caused by Subnormal Floating-Point Values"<sup>[1]</sup>

| Производитель | Процессор   | Замедление (разы) |
|---------------|-------------|-------------------|
| IBM           | PowerPC 970 | 2,4               |
| AMD           | Athlon      | 6,0               |
| Intel         | Pentium 3   | 15,8              |
| AMD           | Athlon 64   | 21,4              |
| AMD           | Opteron64   | 23,8              |
| Intel         | Core Duo    | 44,2              |

|       |               |       |
|-------|---------------|-------|
| Intel | P4 Xeon       | 97,9  |
| Intel | Pentium 4     | 131,0 |
| Intel | Itanium 2     | 183,2 |
| Sun   | UltraSPARC IV | 520,0 |

В таблице приведены наилучшие результаты тестирования среди всех использованных компиляторов (gcc, icc, xlc) со всеми доступными флагами оптимизации. Исследователи утверждают, что различие среднего случая с худшим незначительно.

Поскольку в стандартных форматах (одинарной и двойной точности) денормализованные числа получаются действительно *очень* маленькими и практически никак не влияют на результат некоторых вычислений (при этом заметно замедляя их скорость), то иногда они просто игнорируются. При этом используются два простых механизма, получивших название *Flush-to-zero (FTZ)* и *Denormals-are-zero (DAZ)*. Первый механизм заставляет операции возвращать ноль, как только становится ясно, что результат будет денормализованным. Второй механизм заставляет операции рассматривать поступающие на вход денормализованные числа как нули.

Ярким примером подобного "отсечения" денормализованных чисел могут послужить видеокарты, в которых резкое падение скорости вычислений в сотню раз недопустимо. Так же, например, в областях, связанных с обработкой звука, нет нужды в очень маленьких числах, поскольку они представляют столь тихий звук, что его не способно воспринять человеческое ухо.

В версии стандарта IEEE 754-2008 денормализованные числа (*denormal* или *denormalized numbers*) были переименованы в *subnormal numbers*, то есть в числа, меньшие "нормальных". Поэтому их иногда еще называют "субнормальными".

### 3.6. Действия с числами с плавающей запятой

#### Умножение и деление

Самыми простыми для восприятия арифметическими операциями над числами с плавающей запятой являются умножение и деление. Для того, чтобы умножить два вещественных числа в нормализованной форме необходимо перемножить их мантиссы, сложить порядки, округлить и нормализовать полученное число.

Соответственно, чтобы произвести деление нужно разделить мантиссу делимого на мантиссу делителя и вычесть из порядка делимого порядок делителя. Затем точно так же округлить мантиссу результата и привести его к нормализованной форме.

#### Сложение и вычитание

Идея метода сложения и вычитания чисел с плавающей точкой заключается в приведении их к одному порядку. Сначала выбирается оптимальный порядок, затем мантиссы обоих чисел представляются в соответствии с новым порядком, затем над ними производится сложение/вычитание, мантисса результата округляется и, если нужно, результат приводится к нормализованной форме.

Пример:

Выполним сложение чисел с плавающей точкой и смещенным порядком в 32-х разрядном формате –  $-269\ 7/32$  и  $405,875$ .

Переведем  $-269\ 7/32$  в машинный вид. Для этого сначала переведем его в двоичную систему счисления.

$$-269\ 7/32 = -269,21875$$

$$-269,21875_{10} = -100001101,00111_2$$

Нормализуем полученное двоичное число по правилам машинной арифметики.

$$-100001101,00111 = -1,0000110100111 \times 101000$$

Найдем смещенный порядок. Так как в условии говорится о 32-разрядном представлении, то смещение порядка равно  $12710$ .

$$E = 810 + 12710 = 10002 + 1111111_2 = 10000111_2$$

Число отрицательное, следовательно, в бите знака будет стоять единица.

Итак, первое число в машинном 32-разрядном представлении с плавающей точкой будет иметь вид:

**110000111**00001101001110000000000 (жирным шрифтом выделен порядок числа, длина мантиссы — 23 бита).

Переведем второе число в машинный вид, совершая те же действия.

$405,87510 = 110010101,111000000000011010..._2 = 1,10010101111000000000011010... \times 10^{1000}$

В качестве мантиссы будут сохранены первые 23 бита после запятой т.е.

10010101111000000000011.

Очевидно, что порядок со смещением у второго числа будет таким же, как и у первого.

Второе число положительное, следовательно, бит знака будет содержать ноль.

Итак в машинном 32-разрядном представлении второе число будет иметь вид:

0**10000111**10010101111000000000011

Далее в арифметических операциях будет использоваться число  $110010101,111_2 = 405,87510$ , а не  $110010101,111000000000011_2 = 405,8751010$  видимо для упрощения(хотя это не совсем корректно).

Порядки у слагаемых равны, поэтому пропускаем шаг выравнивания порядков и проводим вычитание мантиссы по правилам двоичной арифметики. В компьютере этим занимается арифметический сопроцессор, встроенный в центральный процессор машины.

$1,1001010111100_2 - 1,0000110100111_2 = 0,1000100010101_2$

Приводим полученный результат к машинному виду. Для этого мы должны внести поправку в порядок — уменьшить его на единицу.

Знак результата — положительный, следовательно, бит знака содержит ноль.

0**10000110**0001000101010000000000

Проверим правильность наших вычислений. Переведем результат в десятичное представление.



Найдем реальный порядок результата, вычтя из него значение смещения 12710.

$$E = 100001102 - 11111112 = 13410 - 12710 = 710 = 1112$$

Следовательно, число результата будет иметь вид:

$$A = 1,000100010101 \times 10111 = 10001000,101012 = 136,6562510$$

Результат наших вычислений верен, так как  $405,875 - 269,21875 = 136,65625$ .

### 3.7. Алгоритм получения представления вещественного числа в памяти

#### ЭВМ

Покажем преобразование действительного числа для представления его в памяти ЭВМ на примере величины типа Double.

Как видно из таблицы, величина этого типа занимает в памяти 8 байт. На рисунке ниже показано, как здесь представлены поля мантиссы и порядка (нумерация битов осуществляется справа налево):

| Знак | Смещённый порядок | Мантисса |
|------|-------------------|----------|
| 63   | 62..52            | 51..0    |

Можно заметить, что старший бит, отведенный под мантиссу, имеет номер 51, т.е. мантисса занимает младшие 52 бита. Черта указывает здесь на положение двоичной запятой. Перед запятой должен стоять бит целой части мантиссы, но поскольку она всегда равна 1, здесь данный бит не требуется и соответствующий разряд отсутствует в памяти (но он подразумевается).

Значение порядка хранится здесь не как целое число, представленное в дополнительном коде. Для упрощения вычислений и сравнения действительных чисел значение порядка в ЭВМ хранится в виде **смещенного числа**, т.е. к настоящему значению порядка перед записью его в память прибавляется смещение. Смещение выбирается так, чтобы минимальному значению порядка соответствовал нуль. Например, для типа Double порядок занимает 11 бит и имеет диапазон от  $2^{-1023}$  до  $2^{1023}$ , поэтому смещение



|        |        |       |
|--------|--------|-------|
|        | 0      | 000   |
| 6<br>3 | 62..52 | 51..0 |

Прежде всего замечаем, что это код положительного числа, поскольку в разряде с номером 63 записан нуль. Получим порядок этого

числа:  $0111111110_{(2)} = 1022_{(10)}$ ;  $1022 - 1023 = -1$ .

Число имеет вид  $1,1100011 \times 2^{-1}$  или  $0,11100011$ .

Переводом в десятичную систему счисления получаем  $0,88671875$ .

#### 4. КОДИРОВАНИЕ СИМВОЛОВ

На основании одной ячейки информационной ёмкостью 1 бит можно закодировать только 2 различных состояния. Для того чтобы каждый символ, который можно ввести с клавиатуры в латинском регистре, получил свой уникальный двоичный код, требуется 7 бит. На основании последовательности из 7 бит, в соответствии с формулой Хартли, может быть получено  $N=2^7=128$  различных комбинаций из нулей и единиц, т.е. двоичных кодов. Поставив в соответствие каждому символу его двоичный код, мы получим кодировочную таблицу. Человек оперирует символами, компьютер – их двоичными кодами.

Для латинской раскладки клавиатуры такая кодировочная таблица одна на весь мир, поэтому текст, набранный с использованием латинской раскладки, будет адекватно отображен на любом компьютере. Эта таблица носит название ASCII (*American Standard Code of Information Interchange*) по-английски произносится [эски], по-русски произносится [аски]. Ниже приводится вся таблица ASCII, коды в которой указаны в десятичном виде. По ней можно определить, что когда вы вводите с клавиатуры, скажем, символ “\*”, компьютер его воспринимает как код  $42_{(10)}$ , в свою очередь  $42_{(10)}=101010_{(2)}$  – это и есть двоичный код символа “\*”. Коды с 0 по 31 в этой таблице не задействованы.

| код | символ | код | символ | код | символ | код | символ | код | символ | код | символ |
|-----|--------|-----|--------|-----|--------|-----|--------|-----|--------|-----|--------|
| 32  | Пробел | 48  | .      | 64  | @      | 80  | P      | 96  | '      | 112 | p      |
| 33  | !      | 49  | 0      | 65  | A      | 81  | Q      | 97  | a      | 113 | q      |
| 34  | "      | 50  | 1      | 66  | B      | 82  | R      | 98  | b      | 114 | r      |
| 35  | #      | 51  | 2      | 67  | C      | 83  | S      | 99  | c      | 115 | s      |
| 36  | \$     | 52  | 3      | 68  | D      | 84  | T      | 100 | d      | 116 | t      |
| 37  | %      | 53  | 4      | 69  | E      | 85  | U      | 101 | e      | 117 | u      |
| 38  | &      | 54  | 5      | 70  | F      | 86  | V      | 102 | f      | 118 | v      |
| 39  | '      | 55  | 6      | 71  | G      | 87  | W      | 103 | g      | 119 | w      |
| 40  | (      | 56  | 7      | 72  | H      | 88  | X      | 104 | h      | 120 | x      |
| 41  | )      | 57  | 8      | 73  | I      | 89  | Y      | 105 | i      | 121 | y      |
| 42  | *      | 58  | 9      | 74  | J      | 90  | Z      | 106 | j      | 122 | z      |
| 43  | +      | 59  | :      | 75  | K      | 91  | [      | 107 | k      | 123 | {      |
| 44  | ,      | 60  | ;      | 76  | L      | 92  | \      | 108 | l      | 124 |        |
| 45  | -      | 61  | <      | 77  | M      | 93  | ]      | 109 | m      | 125 | }      |
| 46  | .      | 62  | >      | 78  | N      | 94  | ^      | 110 | n      | 126 | ~      |
| 47  | /      | 63  | ?      | 79  | O      | 95  | _      | 111 | o      | 127 | DEL    |

Чтобы хранить также и коды национальных символов каждой страны (в нашем случае – символов кириллицы) требуется добавить еще 1 бит, что увеличит количество уникальных комбинаций из нулей и единиц вдвое, т.е. в нашем распоряжении дополнительно появится 128 свободных кодов (со 128-го по 255-й), в соответствие которым можно поставить символы русского алфавита.

Таким образом, отведя под хранение информации о коде каждого символа 8 бит, мы получим  $N=2^8=256$  уникальных двоичных кодов, что достаточно, чтобы закодировать все символы, которые можно ввести с клавиатуры.

*1 байт = 2<sup>3</sup> бит = 8 бит.*

*На основании одного байта можно получить 2<sup>8</sup>=256 уникальных двоичных кодов.*

В современных кодировочных таблицах под хранение информации о коде каждого символа отводится 1 байт.

*1 символ = 1 байт.*

*В байтах измеряется объем данных (V) при их хранении и передаче по каналам связи. Например, текст “Добрый день!” занимает объем равный 12 байтам.*

Биты в байте нумеруются с конца с 0-го по 7-й. Минимальная комбинация на основании одного байта – восемь нулей, максимальная – восемь единиц.  $11111111_{(2)}=2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0=128+64+32+16+8+4+2+1=255_{(10)}$

Возвращаясь к кодировочным таблицам, заметим, что на сегодняшний день в использовании не одна, а несколько кодировочных таблиц, включающих коды кириллицы, – это стандарты, выработанные в разные годы и различными учреждениями. В этих таблицах различен порядок, в котором расположены друг за другом символы кириллического алфавита, поэтому одному и тому же коду соответствуют разные символы. По этой причине, мы иногда сталкиваемся с текстами, которые состоят из кириллицы, но в бессмысленной для нас последовательности.

Например, текст “Компьютерные вирусы”, введенный в кодировке *Windows-1251* в кодировке *КОИ-8* будет отображен так: “лПНРШАФЕТОШЕ ЧЙТХУЩ”.

| Код | Windows-1251 | КОИ-8 | ISO | Под национальные кодировки отведены коды с 128-го по 255-й. |
|-----|--------------|-------|-----|---|
| 192 | А            | ю     | Р   |   |
| 193 | Б            | я     | С   |   |
| 194 | В            | ѐ     | Т   |   |

Эта проблема разрешима - на каждом компьютере найдутся все основные кодировочные таблицы, и если текст выглядит неадекватно, нужно попробовать перекодировать его, просто указав использовать другую кодировочную таблицу. Но наличие такой проблемы, конечно, вносит неудобства.

Используя 8-битную кодировочную таблицу мы не сможем адекватно увидеть на мониторе и тексты, созданные на тех языках, где используются символы, отличные от латинских и кириллических, например символы с умляутами в немецком языке.

#### **4.1. Юникод. UTF-8**

Теоретически давно существует решение проблем связанных с кодировкой. Оно называется *Unicode (Юникод)*. *Unicode* – это кодировочная таблица, в которой для кодирования каждого символа используется 2 байта, т.е. 16 бит. На основании такой таблицы может быть закодировано  $N=2^{16}=65\ 536$  символов.

Юникод включает практически все современные письменности, в том числе: арабскую, армянскую, бенгальскую, бирманскую, греческую, грузинскую, деванагари, иврит, кириллицу, коптскую, кхмерскую, латинскую, тамильскую, хангыль, хань (Китай, Япония, Корея), чероки, эфиопскую, японскую (катакана, хирагана, кандзи) и другие.

С академической целью добавлены многие исторические письменности, в том числе: древнегреческая, египетские иероглифы, клинопись, письменность майя, этрусский алфавит.

В Юникоде представлен широкий набор математических и музыкальных символов, а также пиктограмм.

На практике распространено представление Юникода UTF-8 (Unicode Transformation Format). UTF-8 обеспечивает наилучшую совместимость с системами, использующими 8-битные символы. Текст, состоящий только из символов с номером меньше 128, при записи в UTF-8 превращается в обычный текст ASCII. Остальные символы Юникода изображаются последовательностями длиной от 2 до 4 байтов. В целом, так как самые распространенные в мире символы – символы латинского алфавита - в UTF-8 по-прежнему занимают 1 байт, такое кодирование экономичнее, чем чистый Юникод.

#### **5. КОДИРОВАНИЕ ЗВУКОВОЙ ИНФОРМАЦИИ**

С начала 90-х годов персональные компьютеры получили возможность работать со звуковой информацией. Каждый компьютер, имеющий звуковую плату, микрофон и колонки, может записывать, сохранять и воспроизводить звуковую информацию. С помощью специальных программных средств

(редакторов аудиофайлов) открываются широкие возможности по созданию, редактированию и прослушиванию звуковых файлов. Создаются программы распознавания речи и появляется возможность управления компьютером при помощи голоса.

У всех источников звука имеются колеблющиеся части, которые приводят в колебательное движение частицы окружающей среды (воздуха) → распространяющаяся звуковая волна вызывает колебательное движение барабанной перепонки уха человека, которое воспринимается мозгом как звук → не все источники колебаний являются источниками звука → звук – механические колебания в частотном диапазоне от 16 Гц до 22000 Гц

Упругие волны в воздухе с частотой от 16 до 20000 Гц вызывают у человека звуковые ощущения. Волны с частотой меньше 16 Гц называют *инфразвуковыми*, а с частотой больше 20 000 Гц - *ультразвуковыми*.

Звук представляет собой звуковую волну с непрерывно меняющейся амплитудой и частотой.

Чем больше амплитуда сигнала, тем он громче для человека, чем больше частота сигнала, тем выше тон.

Для измерения громкости звука применяется специальная единица "децибел" (дБл)

| Звук   | Громкость в децибелах |
|--|-----------------------|
| Нижний предел чувствительности человеческого уха | 0                     |
| Шорох листьев                                    | 10                    |
| Разговор   | 60                    |
| Гудок автомобиля                                 | 90                    |
| Реактивный двигатель                             | 120                   |
| Болевой порог                                    | 140                   |

## Что такое децибел?

Одна из самых распространенных единиц измерения, применяемая в аудиотехнике называется "децибел" и обозначается "дБ". В децибелах выражают уровень и мощность звуковых сигналов.

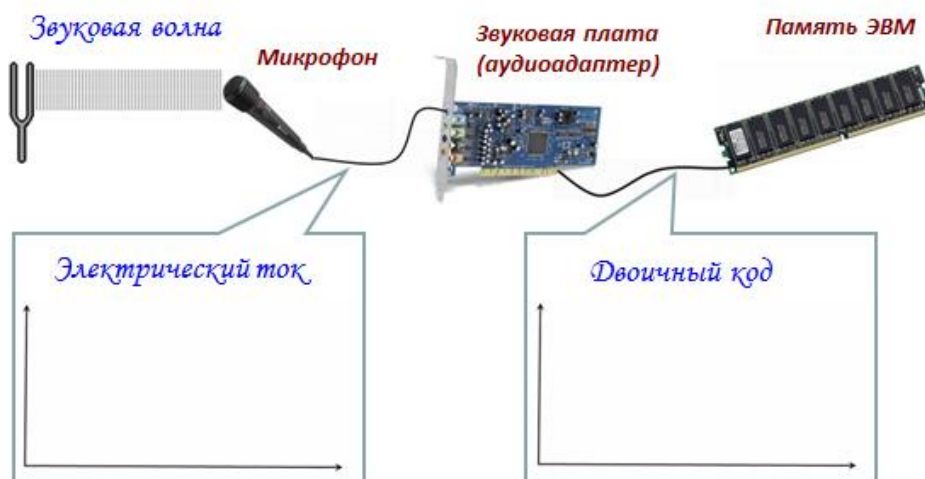
Приставка "деци" применяется для обозначения дольных единиц, равных 1/10 от исходных. Соответственно, децибел -- это 1/10 Бела (единица измерения, названная в честь Александра Грэма Белла). Белл определяется, как логарифм отношения электрических, акустических или других мощностей:

$$\text{Бел} = \log (P1/P0), \text{ дБ} = 10\log (P1/P0).$$

Чувствительность слуха человека к громкости звуковых сигналов носит логарифмический характер, поэтому их мощность, выраженная в децибелах, точнее отражает наше восприятие звуков.

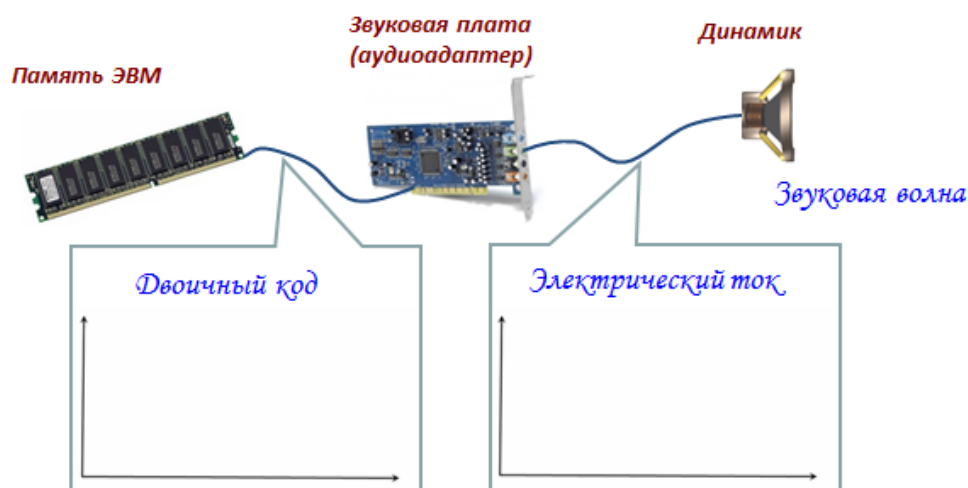
Для того чтобы компьютер мог обрабатывать звук, непрерывный звуковой сигнал должен быть превращен в последовательность электрических импульсов (двоичных нулей и единиц).

## Схема преобразования звуковой волны в двоичный код





## Схема воспроизведения звука, сохранённого в памяти ЭВМ



Непрерывная звуковая волна разбивается на отдельные маленькие временные участки, причем для каждого такого участка устанавливается определенная величина амплитуды.

Таким образом, непрерывная зависимость амплитуды сигнала от времени  $A(t)$  заменяется на дискретную последовательность уровней громкости. На графике это выглядит как замена гладкой кривой на последовательность “ступенек”.

Процесс преобразования непрерывного аналогового сигнала в дискретный (прерывистый) называется временной дискретизацией.

Количество измерений уровня звукового сигнала за 1 секунду называют частотой дискретизации.

Количество информации, которое необходимо для кодирования дискретных уровней громкости цифрового звука называют глубиной кодирования звука.

Задачи для самопроверки.

1. Оцените информационный объем моноаудиофайла длительностью звучания 20 с, если "глубина" кодирования и частота дискретизации звукового сигнала равны соответственно 8 бит и 8 кГц.

2. Рассчитайте время звучания моноаудиофайла, если при 16-битном кодировании и частоте дискретизации 32 кГц его объем равен 700 Кбайт.

## 6. КОДИРОВАНИЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

Графическая информация, как и информация любого другого типа, хранятся в памяти компьютера в виде двоичных кодов. Изображение, состоящее из отдельных точек, каждая из которых имеет свой цвет, называется *растровым изображением*. Минимальный элемент такого изображения в полиграфии называется *растр*, а при отображении графики на мониторе минимальный элемент изображения называют *пиксель (pix)*

Если пиксель изображения может быть раскрашен только в один из 2х цветов, допустим, либо в черный (0), либо в белый (1), то для хранения информации о цвете пикселя достаточно 1 бита памяти ( $\log_2(2)=1$  бит). Соответственно, объем, занимаемый в памяти компьютера всем изображением, будет равен числу пикселей в этом изображении

Если под хранение информации о цвете пикселя выделить 2 бита, то число цветов, допустимых для раскраски каждого пикселя, увеличится до 4х ( $N=2^2=4$ ), а объем файла изображения в битах будет вдвое больше, чем количество составляющих его пикселей

При печати на не цветном принтере обычно допускает 256 градаций серого цвета (от черного (0) до белого (255)) для раскраски каждой точки изображения. Под хранение информации о цвете точки в этом случае отводится 1 байт, т.е. 8 бит ( $\log_2(256)=8$  бит).

### 6.1. Восприятие цвета человеком и животными

Цвет — это ощущение, которое возникает в сознании человека при воздействии на его зрительный аппарат электромагнитного излучения с длиной волны в диапазоне от 380 до 760 нм. Эти ощущения могут быть вызваны и другими причинами: болезнь, удар, мысленная ассоциация, галлюцинации, и др.

Способность к цветоощущению возникла в процессе эволюции как реакция адаптации, как способ получения сведений об окружающем мире и способ ориентирования в нем. Каждый человек воспринимает цвета

индивидуально, отлично от других людей. Однако у большей части людей цветовые ощущения очень схожи.

Физической основой цветовосприятия является наличие специфических светочувствительных клеток в центральном участке сетчатки глаза, так называемых палочек и колбочек.

Различают три вида колбочек, по чувствительности к разным длинам волн света (цветам). Колбочки S-типа чувствительны в фиолетово-синей, M-типа — в зелено-желтой, и L-типа — в желто-красной частях спектра.

Наличие этих трех видов колбочек (и палочек, чувствительных в изумрудно-зеленой части спектра) даёт человеку цветное зрение.

В ночное время зрение обеспечивают только палочки, поэтому ночью человек не может различать цвета.

Каждое животное видит мир по-своему. Сидя в засаде, лягушка видит только движущиеся предметы: насекомых, на которых она охотится, или своих врагов. Чтобы увидеть всё остальное, она должна сама начать двигаться.

Сумеречные и ночные животные (например, волки и другие хищные звери), как правило, почти не различают цветов.

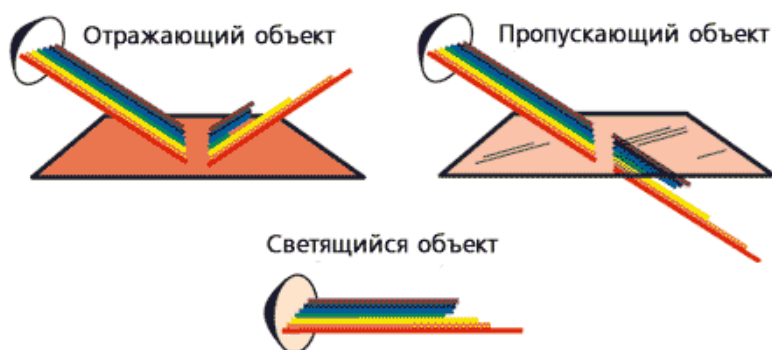
Стрекоза хорошо различает цвета только нижней половиной глаз, верхняя половина смотрит в небо, на фоне которого добыча и так хорошо заметна.

О хорошем зрении насекомых мы можем судить хотя бы по красоте цветков растений - ведь эта красота предназначена природой именно для насекомых - опылителей. Но мир, каким они его видят, сильно отличается от привычного для нас. Цветки, которые опыляют пчелы, обычно не окрашены в красный цвет: пчела этот цвет воспринимает, как мы - черный. Зато, вероятно, многие невзрачные на наш взгляд цветы приобретают неожиданные великолепие в ультрафиолетовом спектре, в котором видят насекомые. На крыльях некоторых бабочек (например, лимонницы, боярышницы) имеются узоры, скрытые от человеческого глаза и видимые только в ультрафиолетовых лучах. Когда муравьев в ходе опыта стали облучать сильными

ультрафиолетовыми лучами, они побежали укрываться "в тень" не под защиту пропускавшей ультрафиолет темной дощечки, а под прозрачное, на наш взгляд, стекло, задерживавшее эти лучи.

## 6.2. Цветовые модели RGB и CMYK

Все объекты окружающего мира можно разделить на: излучающие (светящиеся: солнце, лампа, монитор), отражающие излучение (бумага) и пропускающие (стекло).



В зависимости от того, является объект излучающим или отражающим для представления описания его цвета в виде числового кода используются две обратных друг другу цветовые модели: RGB или CMYK.

Модель *RGB* используется в телевизорах, мониторах, проекторах, сканерах, цифровых фотоаппаратах... Эта модель является *аддитивной* (суммарной), что означает, что цвета в этой модели *добавляются к черному (black)* цвету.

*Основные* цвета в этой модели: *красный (Red)*, *зеленый (Green)*, *синий (Blue)*. Их парное сочетание в равных долях дает *дополнительные* цвета: *желтый (Yellow)*, *голубой (Cyan)* и *пурпурный (Magenta)*.

$$R+G=Y; G+B=C; B+R=M.$$

Сумма всех трех основных цветов в равных долях дает *белый (White)* цвет:  $R+G+B=W$ .

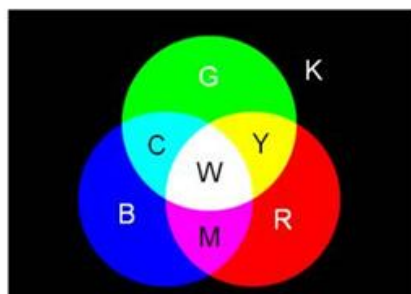
Цветовая модель *CMYK* используется в полиграфии при формировании изображений, предназначенных для печати на бумаге. Основными цветами в

ней являются те, которые являются дополнительными в модели RGB, т.к. они получаются *вычитанием* цветов RGB из белого цвета. Поэтому модель CMYK называется *субтрактивной*.

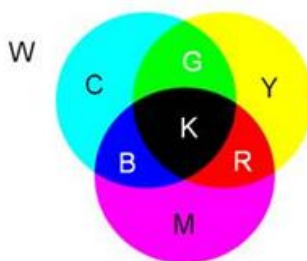
$$C=W-R; M=W-G; Y=W-B.$$

В свою очередь парное сочетание в равных долях цветов модели CMY дает цвета модели RGB. Всем известно, что если смешать на бумаге желтую и голубую краску, получится зеленый цвет. На языке цветовых моделей, это описывается выражением:  $Y+C=G$ , кроме того,  $C+M=B$  и  $M+Y=R$ .

В теории, сумма  $C+M+Y=K$ , т.е. дает *черный (black)* цвет, но поскольку реальные типографские краски имеют примеси, их цвет не совпадает в точности с теоретически рассчитанным голубым, желтым и пурпурным. Особенно трудно получить из этих красок черный цвет. Поэтому в модели CMYK к триаде CMY добавляют черный цвет K. От слова black для обозначения черного цвета взята последняя буква, и т.к. буква B уже используется в модели RGB для обозначения синего цвета.



Излучающий объект RGB.



Отражающий объект CMYK.

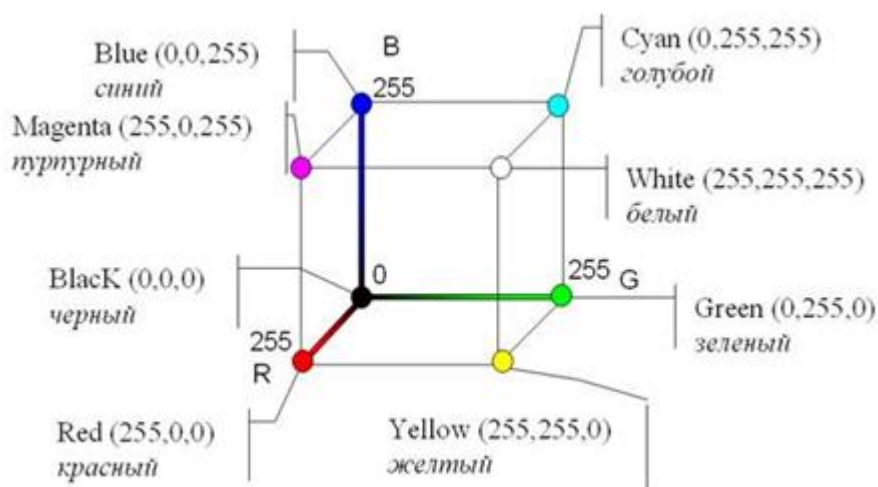
Если кодировать цвет одной точки изображения тремя битами, каждый из которых будет являться признаком присутствия (1) или отсутствия (0) соответствующей компоненты системы RGB, то мы получим все восемь различных цветов описанных выше моделей.

|   | <i>R</i> | <i>G</i> | <i>B</i> | <i>Цвет</i>             |
|---|----------|----------|----------|-------------------------|
| 1 бит на<br>каждый<br>компонент<br><br>RGB<br><br>$2^3=8$ | 1        | 1        | 1        | W (white / белый)       |
|   | 1        | 1        | 0        | Y (yellow / желтый)     |
|   | 1        | 0        | 1        | M (magenta / пурпурный) |
|   | 1        | 0        | 0        | R (red / красный)       |
|   | 0        | 1        | 1        | C (cyan / голубой)      |
|   | 0        | 1        | 0        | G (green / зеленый)     |
|   | 0        | 0        | 1        | B (blue / синий)        |
|   | 0        | 0        | 0        | K (black / черный)      |

На практике же, для сохранения информации о цвете каждой точки цветного изображения в модели RGB обычно отводится 3 байта (т.е. 24 бита) - по 1 байту (т.е. по 8 бит) под значение цвета каждой составляющей. Таким образом, каждая RGB-составляющая может принимать значение в диапазоне от 0 до 255 (всего  $2^8=256$  значений), а каждая точка изображения, при такой системе кодирования может быть окрашена в один из  $2^{3*8}=2^{24}=16\,777\,216$  цветов. Такой набор цветов принято называть True Color (правдивые цвета), потому что человеческий глаз все равно не в состоянии различить большего разнообразия.

### 6.3. Цветовой куб

Изменяющиеся в диапазоне от 0 до 255 координаты RGB образуют цветовой куб. Любой цвет расположен внутри этого куба и описывается своим набором координат, показывающем в каких долях смешаны в нем красная, зеленая и синяя составляющие.



| Изображение        | Основа кодирования  | Памяти на пиксель |     | Кол-во цветов           |
|--------------------|---------------------|-------------------|-----|-------------------------|
|                    |                     | байт              | бит |                         |
| Черно-белое        | Bitmap              |                   | 1   |                         |
| Оттенки серого     | 256 градаций серого | 1                 | 8   | $2^8=256$               |
| Цветное излучающее | RGB                 | 3                 | 24  | $2^{24}=16\ 777\ 216$   |
| Цветное отражающее | CMYK                | 4                 | 32  | $2^{32}=429\ 4967\ 296$ |

### Цветовая модель HSB

Две описанные выше модели удобны скорее для компьютеров, чем для нас с вами. Человеку гораздо проще не синтезировать цвет из отдельных составляющих, а выбирать его, ориентируясь на более естественные параметры: тон, насыщенность, яркость. Именно эти три параметра и стали основой для модели HSB (Hue, Saturation, Brightness), она же HSL (Hue, Saturation, Lightness).

Параметр тона Hue (читается «хью») — это чистый цвет сам по себе — один из цветов спектра (радуги). В модели HSB он представлен как замкнутый круг, положение конкретного оттенка на котором указывается в градусах от 0 до 359.

Параметр Saturation — это насыщенность. Чем меньше насыщенность, тем ближе цвет к серому и наоборот: с увеличением насыщенности цвет становится сочнее. Lightness, соответственно, определяет долю белого в итоговом цвете.

### Цветовая модель Lab

В попытке совместить цветовой охват моделей RGB и CMYK была создана модель Lab, не привязанная к среде вывода. Параметр модели  $L$  показывает общую яркость пикселей, параметром  $a$  передаются цвета от темно-зеленого до ярко-розового с разными вариациями насыщенности и яркости, а параметром  $b$  — от светло-синего до ярко-желтого. Модель Lab обеспечивает наибольшую совместимость, цветовой охват и скорость. Из-за

своей универсальности Lab широко используется способными в ней разобратся профессионалами.

Чем меньше размер точки, тем больше разрешающая способность, а значит, выше качество изображения. Величина разрешающей способности выражается в dpi (dot per inch – точек на дюйм), т.е. количество точек в полоске изображения длиной один дюйм (1 дюйм=2,54 см.)

Количество информации, которое используется для кодирования цвета точки изображения, называется **глубиной цвета**. В процессе дискретизации используются различные **палитры цветов** (наборы цветов, которые могут принять точки изображения). Количество цветов **N** в палитре и количество информации **I**, необходимое для кодирования цвета каждой точки, могут быть вычислены по формуле:  $N=2^I$

**ЗАДАЧА 1:**

Рассчитать необходимый объем видеопамяти (в Мбайтах) для графического режима с разрешением 800 x 600 точек и глубиной цвета 24 бита на точку.

Решение:

Находим K – количество точек на экране:  $800 * 600 = 480\ 000$

Необходимый объем видеопамяти :

$24 \text{ бит} * 480\ 000 = 11\ 520\ 000 \text{ бит} =$

$1\ 440\ 000 \text{ байт} =$

$= 1406,25 \text{ Кбайт} =$

$= \underline{1,37 \text{ Мбайт}}$

Ответ: 1,37 Мбайт

**Задача 2:**

Определите информационный объем (в Кбайтах) растрового изображения размером 1024 \* 768 пикселей, состоящего из 64 цветов.

Решение:

Находим количество точек (пикселей)

$K = 1024 * 768 = 786\ 432.$



Находим глубину цвета по формуле:

$$64 = 2^i, \text{ глубина цвета } i = 6 \text{ бит.}$$

Переведём в более крупные единицы измерения количества информации:

Задача 3:

Цветное с палитрой из 256 цветов растровое изображение имеет размер  $100 * 100$  точек. Какой информационный объём имеет изображение?

Решение:

Находим глубину цвета по формуле:

$$256 = 2^i, \text{ глубина цвета } i = 8 \text{ бит.}$$

Переведём в более крупные единицы измерения количества информации:

$$I = \frac{100 * 100 * 8}{8 * 1024} = 9,8 \text{ Кбайт}$$

Задача 4:

Для хранения растрового изображения размером  $128 * 128$  пикселя отвели 4 Кб памяти. Каково максимально возможное число цветов в палитре изображения?

Решение:

$$128 * 128 = 16384 \text{ пикселей}$$

$$4 \text{ КБайта} = 4096 \text{ байт} = 32768 \text{ бит}$$

$$32768 / 16384 = 2 \text{ бита / пиксель}$$

$$2^2 = 4$$

Ответ: 4

## **7. СЖАТИЕ (АРХИВАЦИЯ) РАЗЛИЧНЫХ ВИДОВ ИНФОРМАЦИИ**

Дискретное двоичное представление информации обычно имеет некоторую избыточность. Часто в информации присутствуют последовательности одинаковых битов или их групп. Объем информации имеет большое значение не только для хранения, но также непосредственно влияет на скорость передачи информации по компьютерным сетям. Поэтому были разработаны специальные

методы (алгоритмы) сжатия информации (data compression), с помощью которых можно существенно уменьшить ее объем. Существуют как универсальные алгоритмы, которые рассматривают информацию, как простую последовательность битов, так и специализированные, которые предназначены для сжатия информации определенного типа (изображений, текста, звука и видео). Все алгоритмы сжатия оперируют входным потоком информации, минимальной единицей которой является, бит, а максимальной – несколько бит, байт или несколько байт.

Основными техническими характеристиками процессов сжатия и результатов их работы являются:

степень сжатия (compress rating) или отношение (ratio) объемов исходного и результирующего потоков;

скорость сжатия – время, затрачиваемое на сжатие некоторого объема информации входного потока, до получения из него эквивалентного выходного потока;

качество сжатия – величина, показывающая, насколько сильно упакован выходной поток, при помощи применения к нему повторного сжатия по этому же или иному алгоритму.

Все способы сжатия можно разделить на две категории: обратимое и необратимое сжатие.

**Необратимое сжатие** - такое преобразование входного потока информации, при котором выходной поток, основанный на определенном формате информации, представляет собой объект, достаточно похожий по внешним характеристикам на входной поток, однако отличается от него объемом.

Степень сходства входного и выходного потоков определяется степенью соответствия некоторых свойств объекта (до сжатия и после), представляемого данным потоком информации. Такие подходы и алгоритмы используются для сжатия информации растровых графических файлов, видео и звука. При таком подходе используется свойство структуры данного формата файла и

возможность представить информацию приблизительно схожую по качеству для восприятия человеком. Поэтому, кроме степени или величины сжатия, в таких алгоритмах возникает понятие качества, т.к. исходная информация в процессе сжатия изменяется. Под качеством можно понимать степень соответствия исходной и результирующей информации, оцениваемое субъективно, исходя из формата информации. Для графических файлов такое соответствие определяется визуально, хотя имеются и соответствующие интеллектуальные алгоритмы и программы. Необратимое сжатие невозможно применять в областях, в которых необходимо иметь точное соответствие информационной структуры входного и выходного потоков. Данный подход реализован в популярных форматах представления фотоинформации – JPEG, TIFF, GIF, PNG и др., аудио информации – MP3, видеоинформации – MPEG-4.

**Обратимое сжатие** всегда приводит к снижению объема выходного потока информации без изменения его информативности, т.е. без потери информационной структуры.

Из выходного потока, при помощи восстанавливающего или декомпрессирующего алгоритма, можно получить входной, а процесс восстановления называется декомпрессией или распаковкой и только после процесса распаковки информация пригодна для использования в соответствии с их внутренним форматом.

## **7.1. Способы обратимого сжатия информации**

### **1) Сжатие способом кодирования серий (RLE)**

Наиболее известный простой подход и алгоритм сжатия информации обратимым путем – это кодирование серий последовательностей (Run Length Encoding – RLE). Суть методов данного подхода состоит в замене цепочек или серий повторяющихся байтов или их последовательностей на один кодирующий байт и счетчик числа их повторений.

Например:

44 44 44 11 11 11 11 01 33 FF 22 22 – исходная последовательность

03 44 04 11 00 03 01 33 FF 02 22 – сжатая последовательность.

Первый байт во второй последовательности указывает, сколько раз нужно повторить следующий байт. Если первый байт равен 00, то затем идет счетчик, показывающий, сколько за ним следует неповторяющихся байт информации (00 03). Данные методы, как правило, достаточно эффективны для сжатия растровых графических изображений (BMP, PCX, TIF, GIF), т.к. последние содержат достаточно много длинных серий повторяющихся последовательностей байтов. Недостатком метода RLE является достаточно низкая степень сжатия.

## 2) Алгоритм Хаффмана

Сжимая файл по алгоритму Хаффмана, первое, что необходимо сделать – прочитать файл полностью и подсчитать сколько раз встречается каждый символ из расширенного набора ASCII. Если учитывать все 256 символов, то не будет разницы в сжатии текстового и EXE файла. После подсчета частоты вхождения каждого символа, необходимо сформировать бинарное дерево для кодирования с учетом частоты вхождения символов.

Приведем пример сжатия по алгоритму Хаффмана.

Пусть файл имеет длину 100 байт и в нем присутствует 6 различных символов. Подсчитаем вхождение каждого из символов в файл и получим следующую таблицу:

|                 |    |    |    |   |    |    |
|-----------------|----|----|----|---|----|----|
| Символ          | A  | B  | C  | D | E  | F  |
| Число вхождений | 10 | 20 | 30 | 5 | 25 | 10 |

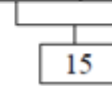
Отсортируем символы по частоте вхождения:

|                 |    |    |    |    |    |   |
|-----------------|----|----|----|----|----|---|
| Символ          | C  | E  | B  | F  | A  | D |
| Число вхождений | 30 | 25 | 20 | 10 | 10 | 5 |

Далее возьмем из последней таблицы 2 символа с наименьшей частотой. В нашем случае это D(5) и F(10) или A(10), можно взять любой из них, например, A.

Сформируем из «узлов» D и A новый «узел», частота вхождения для которого будет равна сумме частот D и A:

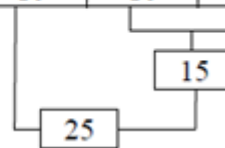
|                 |    |    |    |    |    |   |
|-----------------|----|----|----|----|----|---|
| Символ          | С  | Е  | В  | Ф  | А  | Д |
| Число вхождений | 30 | 25 | 20 | 10 | 10 | 5 |



Номер в рамке – сумма частот символов А и D. Теперь мы снова ищем два символа с самыми низкими частотами вхождения. Исключем из просмотра D и А и рассмотрим вместо них новый «узел» с суммарной частотой вхождения. Самая низкая частота теперь у F и нового «узла».

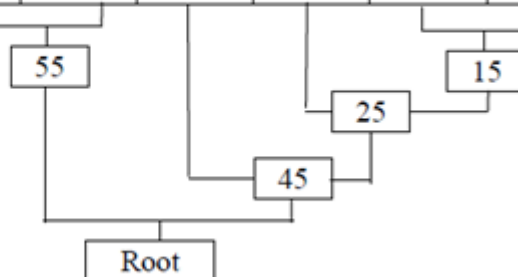
Снова сделаем операцию слияния узлов:

|                 |    |    |    |    |    |   |
|-----------------|----|----|----|----|----|---|
| Символ          | С  | Е  | В  | Ф  | А  | Д |
| Число вхождений | 30 | 25 | 20 | 10 | 10 | 5 |



Просматриваем таблицу снова для следующих двух символов (В и Е). Продолжаем это, пока все «дерево» не сформировано, т.е. пока все не сведется к одному узлу.

|                 |    |    |    |    |    |   |
|-----------------|----|----|----|----|----|---|
| Символ          | С  | Е  | В  | Ф  | А  | Д |
| Число вхождений | 30 | 25 | 20 | 10 | 10 | 5 |



Теперь, когда наше дерево создано, можно кодировать файл. Мы должны всегда начинать из корня (Root). Кодируя первый символ (лист дерева С с наибольшей частотой), прослеживаем вверх по дереву все повороты ветвей и если делаем левый поворот, то запоминаем бит = 0, и аналогично бит = 1 для правого поворота. Так для С, мы будем идти влево к 55 (и запомним 0), затем снова влево (0) к самому символу. Код Хаффмана для нашего символа С – 00. Для следующего символа (Е) получается – лево, право, что выливается в последовательность 01. Выполнив эту процедуру для всех символов, получим:

C = 00 ( 2 бита) E = 01 ( 2 бита ) V = 10 ( 2 бита ) F = 110 ( 3 бита ) A = 1101 ( 4 бита ) D = 1111 ( 4 бита )

При кодировании заменяем символы на новые коды, при этом те символы, которые встречаются наиболее часто, имеют самые короткие коды. Таблицу кодирования запоминаем в том же архивном файле для последующей разархивации.

### **Арифметическое кодирование**

Совершенно иное решение предлагает так называемое арифметическое кодирование. Арифметическое кодирование является методом, позволяющим упаковывать символы входного алфавита без потерь при условии, что известно распределение частот этих символов и является наиболее оптимальным, т.к. достигается теоретическая граница степени сжатия. Каждый символ после кодирования представляется как некоторое дробное число из интервала  $[0, 1)$  с весом, пропорциональным вероятности его появления.

### **Алгоритм Лемпеля-Зива-Велча (Lempel-Ziv-Welch – LZW)**

Данный алгоритм отличают высокая скорость работы как при упаковке, так и при распаковке, достаточно скромные требования к памяти и простая аппаратная реализация. Недостаток – низкая степень сжатия по сравнению со схемой двухступенчатого кодирования.

### **Двухступенчатое кодирование. Алгоритм Лемпеля-Зива**

Гораздо большей степени сжатия можно добиться при выделении из входного потока повторяющихся цепочек – блоков, и кодирования ссылок на эти цепочки с построением хеш-таблиц от первого до n-го уровня с последующим кодированием Хаффмена или арифметическим кодированием. Метод принадлежит Лемпелю и Зиву и обычно называется LZ-compression.

### ***Перечень программ сжатия с кратким указанием алгоритмов их работы***

**PKPAK:** Метод Packed – алгоритм RLE. Метод Crunched – алгоритм LZW. Метод Squashed – двухпроходное статическое кодирование Хаффмена.

**ZIP:** Метод Shrinked – модифицированный алгоритм LZW с частичной очисткой словаря и переменной длиной кода.

Метод Imploded – модифицированный алгоритм Лемпеля-Зива и статическое кодирование Хаффмена.

**LHArc:** Алгоритм Лемпеля-Зива и динамическое кодирование Хаффмена.

**LHA:** Алгоритм Лемпеля-Зива и статическое кодирование Хаффмена.

**ARJ:** Алгоритм Лемпеля-Зива и оригинальный метод кодирования **WinRar:** Использование высокоэффективного оригинального алгоритма автора (А. П. Рошал).

## **8. ЗАКОНОДАТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ О ЗАЩИТЕ КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ**

Нормативы и законы в области информационных технологий и защиты информации должен знать каждый грамотный специалист. Текст законов можно найти в правовых информационных системах «Консультант+», «Гарант» и др., а также в Интернете. Законодательство РФ достаточно часто модифицируется и дополняется. Например, вместо Закона об авторском праве в 2006 г. была принята часть четвертая Гражданского кодекса, последняя редакция которого по состоянию на сентябрь 2008 г. опубликована в редакции Федерального закона от 30.06.2008 N 104-ФЗ.

### **8.1. Уголовный Кодекс РФ о преступлениях в сфере компьютерной информации**

(<http://www.consultant.ru/online/base/?req=doc;base=LAW;n=93082>)

#### **Статья 272. Неправомерный доступ к компьютерной информации**

1. Неправомерный доступ к охраняемой законом компьютерной информации, то есть информации на машинном носителе, в электронно-вычислительной машине (ЭВМ), системе ЭВМ или их сети, если это деяние повлекло уничтожение, блокирование, модификацию либо копирование информации, нарушение работы ЭВМ, системы ЭВМ или их сети, - наказывается штрафом в размере до двухсот тысяч рублей или в размере

заработной платы или иного дохода осужденного за период до восемнадцати месяцев, либо исправительными работами на срок от шести месяцев до одного года, либо лишением свободы на срок до двух лет (в ред. Федерального закона от 08.12.03 № 162-ФЗ).

2. То же деяние, совершенное группой лиц по предварительному сговору или организованной группой либо лицом с использованием своего служебного положения, а равно имеющим доступ к ЭВМ, системе ЭВМ или их сети, - наказывается штрафом в размере от ста тысяч до трехсот тысяч рублей или в размере заработной платы или иного дохода осужденного за период от одного года до двух лет, либо исправительными работами на срок от одного года до двух лет, либо арестом на срок от трех до шести месяцев, либо лишением свободы на срок до пяти лет (в ред. Федерального закона от 08.12.03 № 162-ФЗ).

### **Статья 273. Создание, использование и распространение вредоносных программ для ЭВМ**

1. Создание программ для ЭВМ или внесение изменений в существующие программы, заведомо приводящих к несанкционированному уничтожению, блокированию, модификации либо копированию информации, нарушению работы ЭВМ, системы ЭВМ или их сети, а равно использование либо распространение таких программ или машинных носителей с такими программами - наказываются лишением свободы на срок до трех лет со штрафом в размере до двухсот тысяч рублей или в размере заработной платы или иного дохода осужденного за период до восемнадцати месяцев (вред. Федерального закона от 08.12.03 № 162-ФЗ).

2. Те же деяния, повлекшие по неосторожности тяжкие последствия, - наказываются лишением свободы на срок от трех до семи лет.

### **Статья 274. Нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети**

1. Нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети лицом, имеющим доступ к ЭВМ, системе ЭВМ или их сети, повлекшее



уничтожение, блокирование или модификацию охраняемой законом информации ЭВМ, если это деяние причинило существенный вред, - наказывается лишением права занимать определенные должности или заниматься определенной деятельностью на срок до пяти лет, либо обязательными работами на срок от ста восьмидесяти до двухсот сорока часов, либо ограничением свободы на срок до двух лет.

2. То же деяние, повлекшее по неосторожности тяжкие последствия, - наказывается лишением свободы на срок до четырех лет.

Законодательство РФ о защите программ для ЭВМ, как объекта авторского права

Часть 4 Гражданского кодекса РФ определяет права на результаты интеллектуальной деятельности.

### **Статья 1225. Охраняемые результаты интеллектуальной деятельности и средства индивидуализации**

1. Результатами интеллектуальной деятельности и приравненными к ним средствами индивидуализации юридических лиц, товаров, работ, услуг и предприятий, которым предоставляется правовая охрана (интеллектуальной собственностью), являются:

- 1) произведения науки, литературы и искусства;
- 2) программы для электронных вычислительных машин (программы для ЭВМ);
- 3) базы данных; ...

Статья 1256. Действие исключительного права на произведения науки, литературы и искусства на территории Российской Федерации

Исключительное право на произведения науки, литературы и искусства распространяется:

- 1) на произведения, обнародованные на территории Российской Федерации или необнародованные, но находящиеся в какой-либо объективной форме на территории Российской Федерации, и признаются за авторами (их правопреемниками) независимо от их гражданства;...

## **Статья 1261. Программы для ЭВМ**

Авторские права на все виды программ для ЭВМ (в том числе на операционные системы и программные комплексы), которые могут быть выражены на любом языке и в любой форме, включая исходный текст и объектный код, охраняются так же, как авторские права на произведения литературы. Программой для ЭВМ является представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств в целях получения определенного результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения.

Кроме того, в части 4 Гражданского кодекса присутствуют специальные статьи, посвященные программам для ЭВМ:

**Статья 1262. Государственная регистрация программ для ЭВМ и баз данных**  
**Статья 1280. Свободное воспроизведение программ для ЭВМ и баз данных. Декомпилирование программ для ЭВМ**

**Статья 1296. Программы для ЭВМ и базы данных, созданные по заказу**

**Статья 1297. Программы для ЭВМ и базы данных, созданные при выполнении работ по договору.**

Многие положения других статей относятся ко всем объектам авторского права, которыми являются и программы для ЭВМ и базы данных.

Ответственность за нарушение авторских прав изложена в статье 146 Уголовного кодекса РФ.

**1.3.3 Федеральный закон РФ «Об информации, информационных технологиях и о защите информации»**

Статья 1. Сфера действия настоящего Федерального закона

1. Настоящий Федеральный закон регулирует отношения, возникающие при:

1) осуществлении права на поиск, получение, передачу, производство и распространение информации;

2) применении информационных технологий;

3) обеспечении защиты информации.

2. Положения настоящего Федерального закона не распространяются на отношения, возникающие при правовой охране результатов интеллектуальной деятельности и приравненных к ним средств индивидуализации.

Дальнейшие статьи содержат многочисленные положения правового регулирования в сфере информации, информационных технологий и защиты информации. Большое внимание уделяется вопросам ограничения доступа к информации, создания государственных информационных систем, защиты информации.

## **8.2. Требования к организации рабочих мест пользователей ПК**

Главным документом, определяющим данные требования, является СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

Документ содержит разделы:

I. Общие положения и область применения

II. Требования к ПЭВМ

III. Требования к помещениям для работы с ПЭВМ

IV. Требования к микроклимату, содержанию аэроионов и вредных химических веществ в воздухе на рабочих местах, оборудованных ПЭВМ

V. Требования к уровням шума и вибрации на рабочих местах, оборудованных ПЭВМ

VI. Требования к освещению на рабочих местах, оборудованных ПЭВМ

VII. Требования к уровням электромагнитных полей на рабочих местах, оборудованных ПЭВМ

VIII. Требования к визуальным параметрам ВДТ (видеодисплейный терминал), контролируемым на рабочих местах

IX. Общие требования к организации рабочих мест пользователей ПЭВМ

X. Требования к организации и оборудованию рабочих мест с ПЭВМ для взрослых пользователей

XI. Требования к организации и оборудованию рабочих мест с ПЭВМ для обучающихся в общеобразовательных учреждениях и учреждениях начального и высшего профессионального образования

XII. Требования к оборудованию и организации помещений с ПЭВМ для детей дошкольного возраста

XIII. Требования к организации медицинского обслуживания пользователей ПЭВМ

XIV. Требования к проведению государственного санитарно-эпидемиологического надзора и производственного контроля.

Документ содержит также 14 приложений.

Некоторые самые существенные требования по организации рабочих мест пользователей ПЭВМ (так здесь называют ПК; ВДТ - видеодисплейный терминал), о которых не должен забывать простой пользователь, приведены в следующих разделах документа:

3.4. Площадь на одно рабочее место пользователей ПЭВМ с ВДТ на базе электронно-лучевой трубки (ЭЛТ) должна составлять не менее 6 м<sup>2</sup>, в помещениях культурно-развлекательных учреждений и с ВДТ на базе плоских дискретных экранов (жидкокристаллические, плазменные) - 4,5 м<sup>2</sup>.

При использовании ПЭВМ с ВДТ на базе ЭЛТ (без вспомогательных устройств - принтер, сканер и др.), отвечающих требованиям международных стандартов безопасности компьютеров, с продолжительностью работы менее 4-х часов в день допускается минимальная площадь 4,5 м<sup>2</sup> на одно рабочее место пользователя (взрослого и учащегося высшего профессионального образования).

4.4. В помещениях, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.

6.1. Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.

9.1. При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора) должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

9.4. Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600 - 700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

9.6. Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ.

Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

9.7. Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

*Полный текст документа можно найти в Интернете по адресу <http://lib.herzen.spb.ru/postuplenia/gigienichjrebov.pdf>.*

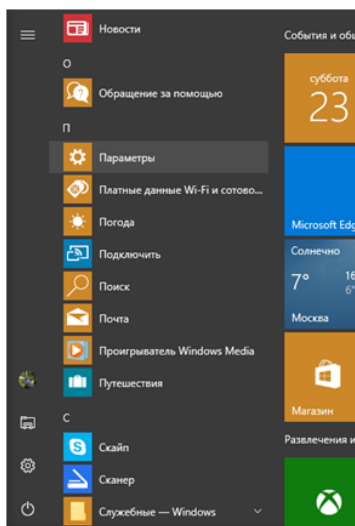
## **9. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ**

Наиболее известны и распространены персональные компьютеры на базе процессоров фирмы Intel и их аналогов (ранее их называли IBM-совместимые ПК) и ПК фирмы Apple - Macintosh.

Историю развития средств вычислительной техники можно найти во многих учебниках по информатике или в Интернете, например на сайте Свободной энциклопедии - Википедии (см. <http://ru.wikipedia.org/wiki/Компьютер>).

Аппаратное обеспечение - все те компоненты, из которых состоит компьютер, а также оборудование для организации локальных и глобальных сетей - так называемое компьютерное «железо» (hardware).

### **как открыть параметры в Windows 10**



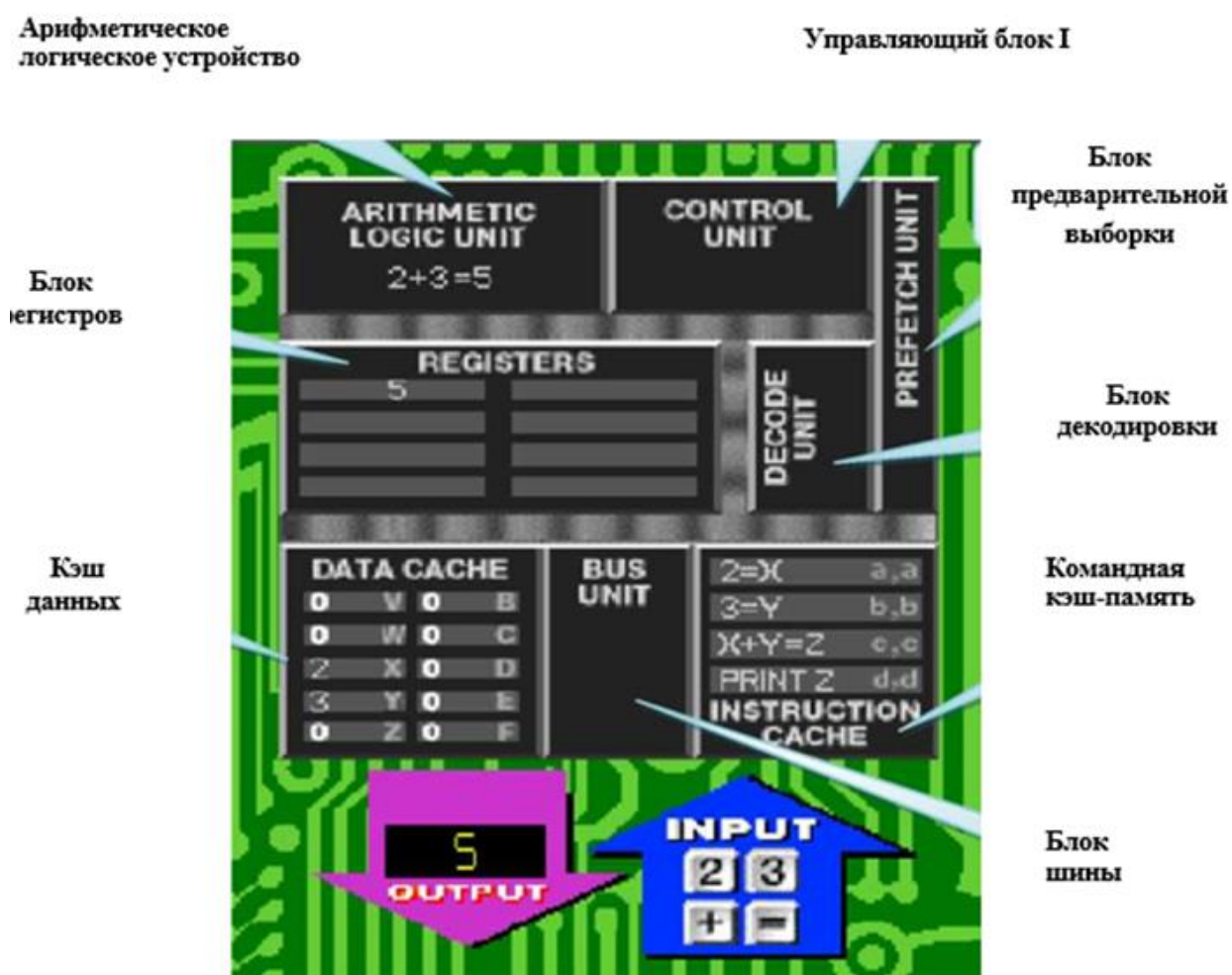
### **9.1. Процессор**

Процессор - специальная микросхема, которая выполняет операции по обработке информации в компьютере.

На небольшой кремниевой пластине размещены сотни миллионов транзисторов-переключателей и каналов передачи данных. Кроме центрального процессора (CPU) в современных компьютерах значительную роль играет процессор видеокарты, который занимается обработкой видеоинформации. В 2010 году при переходе на 32 нм технологию корпорация Intel разместила в одном корпусе центрального процессора также вторую микросхему - видеоядро HD Graphics (GPU).

От процессора в значительной степени зависит скорость работы ПК (лимитирующим фактором также может быть объем оперативной памяти). Процессор имеет сложную архитектуру, свою высокоскоростную буферную память (кэш), использует специальные технологии обработки информации.

**Простейшая принципиальная схема микропроцессора Intel : (по данным фирмы Intel).**



Принцип работы центрального процессора можно представить следующим образом. Информация для обработки под управлением блока предварительной выборки поступает из системной памяти через блок шины в кэш данных процессора, команды обработки информации - в командную кэш-память. Блок декодировки раскодирует команды, преобразуя их в двоичный код, который пересылается в управляющий блок и в кэш данных, давая им указание о том, как с полученной командой поступать дальше. Арифметическое логическое устройство выполняет готовые к исполнению команды и заносит результаты в блок регистров. Далее содержимое регистров передается в системную память или на внешние устройства. Более подробное описание выполнения процессором простой операции сложения  $2+3$  можно найти по адресу <http://www.intel.com/plt/cd/corporate/emea/rus/museum/mpuworks.swf>, здесь же в глоссарии приведено краткое описание функций всех устройств

Скорость работы процессора зависит в настоящее время, прежде всего от типа и архитектуры процессора, а также от его тактовой частоты и объема кэш-памяти. Процессоры Intel Pentium и Core используют технологию конвейерной обработки данных, в результате чего за один такт выполняется несколько машинных операций. Многоядерные процессоры позволяют увеличить производительность ПК за счет одновременного выполнения нескольких программ пользователя на разных ядрах или выполнения одной программы на нескольких ядрах, если она предусматривает параллельную многопроцессорную обработку данных. Скорость обработки информации процессором может также лимитироваться скоростью поступления этой информации из оперативной памяти.

В 1990 году прогнозировалось - тактовая частота процессора Intel возрастет к 2000 году до 900 МГц, количество транзисторов в нём - до 40 млн. штук, к 2005 году - 10 ГГц, 1 млрд. транзисторов. Однако в 2005 г. стало ясно, что прогресс пошел другим путем - развитием многоядерных процессоров без увеличения их рабочей частоты и количества транзисторов на чипе (но с



увеличением частоты системной шины FSB до 1333 МГц и с поддержкой 64-разрядных вычислений и памяти).

Рейтинг процессоров Intel можно посмотреть на сайте [http://www.intel.com/ru\\_RU/consumer/rating.htm](http://www.intel.com/ru_RU/consumer/rating.htm).

## 9.2. Чипсет

Чипсет (chipset) - набор микросхем материнской платы для обеспечения работы процессора с памятью и внешними устройствами.

Раньше компьютер имел до 2-х сотен микросхем на материнской плате. Современные компьютеры содержат две основные большие микросхемы чипсета, либо одну микросхему :

1. Трехчиповая компоновка (третий чип - процессор):

◆ северный мост (North Bridge) - один из двух вариантов:

- контроллер-концентратор памяти (MCH), который обеспечивает работу процессора с памятью и с видеоподсистемой или

- контроллер ввода-вывода (IOH) - для процессоров с интегрированным контроллером памяти;

◆ южный мост (South Bridge) - контроллер-концентратор ввода-вывода (ICH), обеспечивающий работу с внешними устройствами.

2. Двухчиповая компоновка для некоторых процессоров Core i7 и Core i5

кроме процессора используется только одна большая микросхема чипсета (PCH - Platform Controller Hub), обеспечивающая работу с внешними устройствами.

Разработкой чипсетов для настольных ПК занимаются 5 компаний: Intel, AMD, NVIDIA, VIA и SIS.

## 9.3. Материнская плата

Материнская плата - печатная плата, на которой осуществляется монтаж микросхем и других компонентов компьютерной системы.

Название происходит от английского motherboard, иногда используется сокращение MB или слово mainboard - главная плата, в русских источниках используется также название системная плата.

На материнской плате располагаются микросхемы чипсета, разъемы для подключения центрального процессора, оперативной памяти, дисковых устройств, графической платы, звуковой платы и дополнительных внешних устройств.

Существуют различные спецификации (форм-факторы), описывающие конструктивные особенности материнских плат.

Форм-фактор - стандарт (спецификация), определяющий размеры материнской платы, расположение крепежных отверстий, сокета центрального процессора, слотов оперативной памяти, интерфейсов шин, портов ввода/вывода, разъемов для подключения питания.

Наиболее известен форм-фактор ATX (Advanced Technology extended), с достаточно большим размером, что позволяет производителям размещать на материнской плате наибольшее количество устройств. Полные спецификации приведены на сайте <http://www.formfactors.org/formfactor.asp>.

Пример расположения портов ввода/вывода на задней панели материнской платы, приведенный в спецификации ATX

Наиболее важными микросхемами материнской платы являются северный и южный мост чипсета. Именно чипсет определяет, в значительной степени, особенности материнской платы и то, какие устройства могут подключаться к ней.

В настоящее время фирма Intel ввела классификацию своих материнских плат по их назначению (таблица), полный список приведен по адресу <http://www.intel.com/cd/products/services/emea/rus/motherboards/desktop/323255.htm>

#### Материнская плата Intel DG41TY

Материнская плата имеет радиатор пассивного охлаждения для микросхем чипсета MCH, кроме того, имеет возможность подключения

дополнительных вентиляторов для поддержания необходимого температурного режима работы микросхем.

Существует еще одна маленькая, но очень важная микросхема материнской платы – BIOS (Basic Input-Output System – базовая система ввода-вывода), а также устройства для ее питания – микросхема BP\_BIOS и аккумуляторная батарейка (BATTERY). Назначение этой микросхемы с постоянным питанием и наличием собственной энергонезависимой памяти (перепрограммируемая Flash EEPROM):

конфигурирование компьютера (программа BIOS Setup) и сохранение параметров конфигурации;

выполнение программы часов и календаря компьютера;

при включении компьютера:

-- выполнение программы самотестирования;

-- поиск главной загрузочной записи (MBR) на дисках;

-- далее MBR выполняет сканирование дисков для поиска загрузочного раздела, и, если он найден, запускает в работу операционную систему.

В программе BIOS Setup (которая запускается нажатием клавиши Delete или другой в момент включения компьютера) есть возможность настроить порядок устройств для поиска загрузчика операционной системы - если необходимо, ее можно загрузить с CD, DVD диска или даже через USB-порт. Эта возможность необходима для установки новой операционной системы на компьютер (или ее переустановки).

Назначение слотов (разъемов) на материнской плате – подключение различных плат сопряжения с внешними устройствами и реализация дополнительных возможностей:

— PCI Express x16 - только для соответствующих видеоплат;

— PCI Express x1 - новые звуковые, сетевые платы;

— PCI - старые звуковые, сетевые платы, внутренние модемы, FM-, TV-тюнеры.

PCI (Peripheral Component Interconnect bus) - шина для подсоединения периферийных устройств. Впервые массово применяться начала для Pentium-систем.

Шина PCI Express x16 пришла на смену шине AGP, ранее использовавшейся для подключения видеокарт.

Внутренние порты, расположенные на материнской плате, служат для подключения устройств:

- Serial ATA - дисковые устройства (HD, DVD) соответствующего типа;
- ATA IDE - дисковые устройства (HD, CD, DVD) соответствующего

типа.

ATA (Advanced Technology Attachment) - спецификация интерфейса для подключения дисковых устройств. IDE (Integrated Drive Electronics) - диск со встроенным контроллером управления для подключения с помощью параллельного интерфейса спецификации ATA.

Serial ATA - новый последовательный интерфейс подключения дисков.

Кабели подключения IDE и SATA

Внешний порт – разъем на задней или передней панели системного блока, используется для подключения различных устройств:

- USB 2.0 – принтеры, сканеры, внешние модемы, TV-тюнеры, клавиатура, мышь, внешние дисковые накопители, Flash Drive и многое другое;
- IEEE-1394 – цифровые видеокамеры, дис-ковые устройства;
- PS/2 – клавиатура и мышь;
- при наличии интегрированной видеоподсистемы разъем DVI – цифровой выход на монитор или VGA – аналоговый;
- при наличии интегрированной аудиоподсистемы разъемы для подключения колонок и микрофона.

Кроме того, могут присутствовать устаревшие на сегодняшний день порты:

- COM – мышь, модем и пр.;

– LPT – принтер, сканер.

Новые интерфейсы подключения внешних устройств – USB, SATA, IEEE-1394 (FireWire), DVI – последовательные шины (Serial Bus) с не-большим количеством проводов (4 – 7) в кабеле. Параллельные шины ATA IDE, LPT имеют значительно большее количество проводников в кабеле (IDE 40 или 80 проводов, LPT – 18) и, соответственно, контактов на разъемах. Скорость работы современных последовательных шин выше, чем параллельных.

Наиболее часто используется в настоящее время порт USB (Universal Serial Bus – универсальная последовательная шина), которая имеет 2 типа: USB и USB 2.0. Особенности этой шины:

- высокая скорость работы – до 480 Мб/с для USB 2.0;
- длина кабеля подключения – до 5 м;
- количество подключаемых устройств (включая размножители) – до 127;
- возможно подключение устройств с различными скоростями обмена;
- возможность подключать и отключать устройства при работающем компьютере;
- возможность подавать питание напряжением 5 V на маломощные периферийных устройства (ток до 500 mA);
- кабель подключения имеет различные разъемы для подключения к компьютеру (тип «А») и к внешнему устройству (тип «В»).

Шина USB предоставляет в полной мере возможность использования технологии Plug&Play для периферийных устройств, т. е. при подключении нового устройства к работающему компьютеру (наиболее часто в настоящее время – флэш-накопителей, внешних жестких дисков, «мышь» на ноутбуках и т. п.) происходит его автоматическое распознавание и установка соответствующих драйверов, необходимых для работы с этим устройством. Для использования большого количества USB портов применяются USB-концентраторы (хабы), как внешние, так и встроенные в монитор, клавиатуру и пр.

В прайс-листах фирм по продаже компьютеров обычно приводится краткое обозначение материнской платы примерно следующего вида:

MB Intel DG45FC LGA775, G45, DDR2 800, SATA2 RAID, PCI-E, 8ch Audio, GBL, DVI+HDMI, mini-ITX,

что означает:

материнская плата Intel DG45FC, разъем процессора LGA775, чипсет G45, память DDR2 с частотой 800 МГц, дисковый интерфейс SATA2 с аппаратной поддержкой дисковых массивов (RAID), шина и слоты PCI Express, звуковой адаптер 7.1 (8ch Audio), интегрированный сетевой адаптер Gb Ethernet (GBL), видеоадаптер с выходами DVI и HDMI, форм-фактор mini-ITX (171,45 мм x 171,45 мм).

#### **9.4. Оперативная память**

Оперативная память – один из основных компонентов компьютера, предназначенный для хранения информации (программ и обрабатываемых программами данных) во время исполнения программ на компьютере.

Кроме основной оперативной памяти (RAM – Random Access Memory, память с произвольным доступом) современные персональные компьютеры имеют память видеокарты (с которой работает процессор видеокарты), а также сверхоперативную память центрального процессора (кэш первого, второго и иногда третьего уровня). Для долговременного хранения информации используются дисковые и пр. устройства, которые иногда называют внешней памятью

Чипсет, используемый на компьютере, определяет тип оперативной памяти, который можно использовать.

Современная оперативная память – динамическая (DRAM), содержимое ее остается неизменным в течение очень короткого промежутка времени, поэтому память должна периодически обновляться (регенерироваться). Запоминающим элементом динамической памяти является конденсатор, который может находиться в заряженном или разряженном состоянии. Если конденсатор заряжен, то в ячейку памяти записана логическая 1, если разряжен

– логический 0. В идеальном конденсаторе заряд может сохраняться неограниченное время, в реальном конденсаторе существует ток утечки, поэтому информация, записанная в динамическую память, со временем будет утрачена в результате разрядки конденсаторов. Единственным способом регенерации хранимой в памяти информации является выполнение операций чтения и повторной записи данных, выполняемых через определенные промежутки времени (например, каждые 2 мс (миллисекунда (1/1000 доля секунды)) всех ячеек памяти. В эти моменты процессор находится в состоянии ожидания. Регенерация памяти происходит также при выполнении каждой операции чтения данных.

Ячейки памяти организованы в матрицу, состоящую из строк и столбцов. Полный адрес ячейки данных включает два компонента – адрес строки и адрес столбца. При чтении информации из памяти вначале считывается информация всех ячеек памяти для строки с заданным адресом и помещается в буфер ввода/вывода. Далее в соответствии с адресом столбца данные выбираются из буфера ввода/вывода и поступают на выход динамической памяти. Микросхемы памяти состоят из отдельных матриц, каждая из которых имеет собственную линию чтения/записи. В этом случае одновременно считывается или записывается несколько бит информации. Количество линий определяет разрядность шины ввода/вывода. Современные микросхемы памяти 64-х разрядные.

Емкость матрицы памяти (глубиной адресного пространства, address depth) и их количество (разрядность) определяет общий объем микросхемы памяти, которые размещаются на модулях памяти (DIMM – модули). DIMM-модуль (Dual In-Line Memory Module) имеет с двух сторон электрически независимые контакты для подключения к шине контроллера памяти через слот, в который они вставляются. Наиболее распространенными являются 240-контактные 64-разрядные модули DIMM, имеющие по 120 контактов с каждой стороны.

Хотя число контактов у модулей DDR2 и DDR3 одинаково, слоты, предназначенные для памяти разного типа, отличаются расположением «ключа» - прорези между контактами. Поэтому, установить DDR3 SDRAM в DDR2 DIMM и наоборот не удастся. SDRAM – микросхема синхронной динамической памяти (Synchronous DRAM), в которой процессы записи и считывания информации строго привязаны к тактам системной шины.

DDR SDRAM (Double Date Rate – удвоенная скорость передачи данных) является следующим поколением SDRAM. В отличие от SDRAM память этого типа дает возможность выполнять два обращения к памяти за время одного такта системной шины. DDR2 и DDR3 – последующие модификации памяти DDR SDRAM, позволяющие за один такт записывать или считывать 4 (DDR2) или 8 (DDR3) блоков данных, имеют пониженное напряжение питания (DDR - 2,5 В, DDR2 - 1,8 В, DDR3 - 1,5 В) и энергопотребление.

### **9.5. Устройства хранения информации**

Устройства хранения информации (внешняя память) – компоненты компьютера, позволяющие практически неограниченное время сохранять большие объемы информации без потребления электроэнергии (энергонезависимые).

Первыми такими устройствами для ПК были Floppy-дискеты (FDD) и сменные дискеты – вначале пятидюймовые (5,25") емкостью 360 Кб и 1,2 Мб, затем трехдюймовые (3,5") емкостью 1,44 Мб. В настоящее время применяются редко в связи с широким распространением устройств флэш-памяти емкостью в несколько гигабайт.

Примечание. На больших ЭВМ в эпоху до появления ПК для хранения информации использовались бумажные перфокарты, перфоленты, магнитные ленты и диски.

#### **Винчестер**

Следующий этап в развитии устройств хранения информации для ПК – винчестеры (или жесткие диски – Hard Disk Drive, HDD), которые в настоящее время являются обязательным компонентом каждого настольного



персонального компьютера. В принципе, могут существовать бездисковые сетевые рабочие станции и терминалы, с загрузкой операционной системы с сервера и работой с дисковыми устройствами сервера, но на практике такие ПК встречаются достаточно редко. В субноутбуках винчестер может отсутствовать, вместо него используется флэш-память большого объема.

Первый винчестер с обозначением 30/30 был создан корпорацией IBM в 1973 г. и позволял на двухстороннем алюминиевом диске сохранять 30 + 30 Мбайт данных. Название 30/30 совпадает с обозначением американской винтовки типа «винчестер», откуда и произошло наименование устройств хранения информации на жестких дисках.

Современные винчестеры имеют емкость 500 Гбайт и более. Винчестер представляет собой плоскую, герметически закрытую металлическую коробку, внутри которой на общей оси находятся несколько жестких алюминиевых или стеклянных пластинок круглой формы

Винчестер SATA 1000 GB Samsung HD103UJ, 32MB кэш, 7200 об/мин (со снятой верхней крышкой)

Выпускаются винчестеры в основном двух размеров дисков (форм-фактор) – 3.5” и 2.5” (последние преимущественно для использования в ноутбуках). Поверхность каждого диска покрыта тонким ферромагнитным слоем (веществом, способным длительное время сохранять состояние намагниченности). Для чтения и записи данных в винчестерах используются плавающие на воздушной подушке магнитные головки. Зазор между магнитным слоем и головкой составляет десятые доли микрон. Для перемещения головок используется электромагнит, а для отслеживания местоположения применяется автоматическая система регулирования, которая по магнитным меткам на поверхности опорного диска позиционирует головки в нужном месте.

Пакет дисков вращается непрерывно с большой скоростью (7200 – 15000 об/мин). Запись данных осуществляется следующим образом. При изменении силы тока, проходящего через записывающую головку, происходит изменение

напряженности магнитного поля в щели между поверхностью и головкой, что приводит к изменению напряженности магнитного поля на небольшом участке ферромагнитного покрытия диска. Намагниченные участки и не намагниченные соответствуют двоичным кодам 1 и 0. Для чтения данных используется магниторезистивный принцип, когда сопротивление полупроводниковой пленки, из которой сделана головка чтения, меняется в зависимости от напряженности магнитного поля на поверхности ферромагнитного диска. Электрический сигнал с головки усиливается и передается на обработку в контроллер жесткого диска, расположенный в чипсете. Вся электроника, предназначенная для управления двигателем привода дисков и электромагнитом головок, а также для чтения и записи данных, располагается на небольшой печатной плате, укрепленной снаружи корпуса.

Винчестер может иметь до десяти дисков. На поверхности дисков размечаются траектории записи информации – концентрические окружности, которые называются дорожками или треками (track) и сектора на каждой дорожке. Каждая дорожка имеет свой номер. Дорожки с одинаковыми номерами, расположенные одна над другой на разных дисках образуют цилиндр. Каждый сектор на дорожке также имеет номер, нумерация начинается с единицы. В один сектор может быть записано 511 байт двоичной информации: 512 байт отведено для записи нужной информации, остальные под заголовок (префикс), определяющий начало и номер секции и окончание (суффикс), где записана контрольная сумма, нужная для проверки целостности хранимых данных. Секторы и дорожки образуются во время низкоуровневого форматирования диска на заводе-изготовителе. В новых винчестерах число секторов на внешних дорожках больше, чем на внутренних, и для них используют линейный способ адресации блоков LBA (Linear Block Addressing), когда все секторы нумеруются последовательно от первого сектора нулевой дорожки нулевого цилиндра до последнего на крайней внутренней дорожке. К современным винчестерам неприменима операция низкоуровневого

форматирования пользователем с помощью специальных программ (возможна для гибких дисков и старых винчестеров).

Несколько секторов образуют кластер – наименьшую адресуемую единицу на диске для используемой файловой системы. Форматирование с разметкой кластеров и созданием файловой системы на диске выполняется с помощью специальной программы, входящей в состав операционной системы (иногда на стадии установки этой системы). Неформатированный диск не может быть использован для записи файлов. До форматирования жесткий диск можно разбить на логические диски (раздел винчестера или том). Это удобно, поскольку наличие не-скольких логических дисков упрощает структуризацию данных, хранящихся на жестком диске.

Современные винчестеры могут безотказно работать более 100 000 часов, система автоматической дефектации (S.M.A.R.T., Self-Monitoring, Analysis and Reporting Technology) незаметно для пользователя заменяет плохие кластеры на резервные (если в BIOS включен режим S.M.A.R.T.). Если на винчестере появляются плохие сектора и их количество постоянно увеличивается, то такой винчестер лучше заменить на новый. В настоящее время на винчестеры производителями устанавливается 1 – 3-летний гарантийный срок. Винчестеры фирм Seagate и Hitachi объемом 500 Гб состоят из 5 дисков с плотностью записи 100 Гб на диск. Hitachi и Samsung успешно эксплуатируют в своих винчестерах диски объемом 125 Гб. Seagate наряду с WD имеют в составе своих продуктовых линеек винчестеры с плотностью записи 133 Гб на диск. Лидеры рынка дисков класса 7200/3.5”, компании Seagate, Maxtor и WD, выпускают также внешние жесткие диски, выполненные в отдельном корпусе с блоком питания, интерфейсом USB или IEEE1394 (FireWire).

### **Дисковод CD-ROM**

В 1995 году в базовой конфигурации ПК появился первый привод оптических дисков – CD-ROM (Compact Disk Read Only Memory, постоянное запоминающее устройство компакт-дисков). Устройство использовало

многослойные компакт-диски диаметром 120 мм и толщиной 1,2 мм, емкость диска 650 – 700 Мбайт. CD-диск состоит из 4-х слоев (сверху-вниз):

- 1) защитный слой из поликарбоната;
- 2) слой для записи информации;
- 3) отражающий слой;
- 4) основа из поликарбоната.

Процесс изготовления диска состоит из операций напылением се-ребра или золота отражающего слоя на основу, нанесения на него прозрачного слоя для записи информации и выдавливания на нем углублений, образующих спиральную дорожку, идущую от центра диска к его краю. Для штамповки диска используют матрицу-прототип (мастер-диск) будущего диска. После этого на поверхность диска наносят защитный слой из прозрачного пластика.

CD-ROM считывает информацию с диска с помощью лазерного луча с длиной волны 780 нм, который по разному отражается от поверхности диска (land – ленд) и углублений на поверхности (pit – пит). Минимальный размер пита составляет 0,88 мкм, шаг дорожек – 1,5 мкм.

При считывании информации луч лазера отражается от лендов и попадает в фотоприемник, от питов свет тоже отражается, но из-за расфокусировки луча в фотоприемник не попадает. Далее высокочастотный информационный сигнал преобразуется компаратором в последовательность прямоугольных импульсов. Длительность каждого из полученных таким способом прямоугольных импульсов преобразуется в двоичный код.

Основные характеристики CD-ROM:

– скорость передачи данных – измеряется в кратных долях скорости проигрывателя аудио компакт-дисков (150 Кбайт/сек) и характеризует максимальную скорость с которой накопитель пересылает данные в оперативную память компьютера, например, 2-скоростной CD-ROM (2x CD-ROM) будет считывать данные с скоростью 300 Кбайт/сек., 50-скоростной (50x) - 7500 Кбайт/сек.;

– время доступа – время, нужное для поиска информации на диске, измеряется в миллисекундах.

### **Дисковод CD-RW**

Устройство используется для записи информации на диски CD-R (однократная запись) и CD-RW (CD-ReWritable – перезаписываемый диск). Внешне похож на CD-ROM и совместим с ним по размерам дисков и форматам записи. Запись данных осуществляется с помощью специального программного обеспечения или средств операционной системы. Скорость записи современных накопителей CD-R составляет 2x-48x. Диск CD-R или CD-RW имеет 4 слоя (сверху-вниз):

1. защитный слой из поликарбоната;
2. активный слой для записи информации;
3. отражающий слой;
4. основа из поликарбоната.

Может присутствовать также этикетка диска, нанесенная краской на защитный слой. Во время записи информации активный слой CD-R диска нагревается лазерным лучом (длина волны 780 нм) и в месте нагрева перестает пропускать свет. В результате отражающая способность на участках, подвергнутых действию луча лазера, уменьшается. Одним из типов активных слоев CD-R диска, широко используемых на сегодняшний день, является цианин (cyanine).

Активный слой CD-RW диска под воздействием лазерного луча при записи информации изменяет свое агрегатное состояние с переходом из кристаллического состояния в аморфное. Участки диска с аморфным состоянием активного слоя имеют меньшую степень отражения лазерного луча при считывании информации. Процесс стирания информации с CD-RW заключается в обработке поверхности диска лучом лазера меньшей интенсивности, чем при записи, в результате чего активный слой полностью возвращается в кристаллическое состояние. Основа CD-R и CD-RW имеет не гладкую поверхность: при изготовлении она получает разметку – сплошную

спиральную канавку (Pregroove), необходимую для работы системы автослежения, заполненную органическим красителем, который немного хуже отражает свет, чем пластик основы. Для синхронизации частоты тактового генератора со скоростью вращения диска в CD-R канавка выполнена не в виде ровной спирали, а с микроскопическими отклонениями – вобуляцией

### **Флэш-память**

Флэш-память появилась довольно давно (первые образцы были разработаны компанией Toshiba еще в 1984 г.), однако ее массовое использование началось с широким распространением цифровых фотокамер. Сегодня производители выпускают флэш-память нескольких типов:

– флэш-карты Compact Flash (CF), SmartMedia (SM), MultiMedia Card (MMC), SecureDigital (SD), Memory Stick PRO (MS PRO), Memory Stick (MS) и xD-Picture (xD) – для работы с ними необходимо устройство чтения флэш-карт;

– USB-флэш-память (так называемые «флэшки»); последняя самодостаточна и не требует применения дополнительных устройств для записи и чтения информации, имеет разъем для подключения к USB-порту ПК. Полное название флэш-памяти Flash Erase EEPROM (Electrically Erasable Programmable ROM) можно перевести как «быстро электрически стираемое программируемое постоянное запоминающее устройство». Другими словами, флэш-память – это энергонезависимая (не потребляющая энергии при хранении данных) перезаписываемая (данные можно стереть и записать заново при помощи электрического тока) память, содержимое которой можно быстро стереть (Flash Erase). Флэш-память – это полупроводниковая память особого типа. Ее элементарная ячейка, в которой хранится один бит информации, представляет собой не конденсатор, а полевой транзистор со специальной электрически изолированной областью, которую называют «плавающим затвором» (floating gate). Электрический заряд, помещенный в эту область, способен сохраняться многие годы. При записи одного бита данных ячейка заряжается, заряд помещается на плавающий затвор, при стирании он с затвора снимается и ячейка разряжается.

Преимущества флэш-памяти по сравнению с другими средствами переноса и хранения данных очевидны – высокая надежность и ударопрочность (в результате отсутствия движущихся компонентов и простоты механической конструкции носителей и накопителей), малое энергопотребление и компактность. Однако у нее есть и недостатки: ограниченное число циклов перезаписи (в настоящее время до 1 млн.) и относительно медленная работа.

Последнее обстоятельство связано с принципиальной особенностью работы флэш-памяти NAND-архитектуры: нельзя перезаписать содержимое одной отдельно взятой ее ячейки – можно только стереть содержимое достаточно большого блока ячеек памяти и потом записать туда новую информацию. Изменение содержимого одного бита (байта) данных во флэш-памяти происходит поэтапно: сначала с микросхемы флэш-памяти в буфер считывается блок данных, потом этот блок стирается в микросхеме, затем в буфере изменяется нужный бит (байт) и блок данных снова записывается в микросхему.

На базе флэш-памяти разработаны массивы хранения данных (например, Sun Storage F5100 Flash Array). Они предназначены для повышения быстродействия баз данных и оптимизации архитектур сетей хранения данных с целью увеличения производительности при снижении затрат. Объем хранения данных в Sun F5100 Flash Array на базе твердотельных флэш-накопителей может достигать двух терабайт. Sun F5100 обладает производительностью 1,6 млн операций чтения и 1,2 млн операций записи в секунду, имеет пропускную способность 12,8 Гбайт/с. Устройство потребляет 300 Вт. По информации Sun, его рабочие параметры сопоставимы с системой хранения данных класса предприятия с 3 тыс. жестких дисков, которые занимают более 14 стоек и потребляют в десять раз больше электроэнергии (40 кВт).

### **Стример**

Стример – устройство для резервного копирования данных (backup) с жестких дисков на магнитные ленты. Считается, что жесткие диски не

обеспечивают должного уровня безопасности длительного хранения информации. Для более надежного архивного хранения данных рекомендуется применять стримеры. Основной задачей резервного копирования является обеспечение сохранности оперативной информации, представляющей большое значение для любых предприятий. Потери организаций, работающих с банковскими счетами клиентов, от одного часа простоя, вызванного временным сбоем в работе компьютерного оборудования, могут исчисляться миллионами долларов. Потери, вызванные полной утратой коммерческой информации, могут привести к полному краху предприятия. Ежедневное проведение резервного копирования (а лучше два, три раза в день), хоть и является наилучшим выходом с точки зрения безопасности данных, но для хранения нескольких копий требуется слишком много места на магнитных лентах. Поэтому, кроме полного резервного копирования, применяется инкрементальное и дифференциальное копирование. В этом случае после первоначального полного копирования всей необходимой информации выполняется копирование только новой и измененной информации. Для инкрементального метода полное копирование периодически повторяется.

## **10. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

Все программное обеспечение (ПО, software) по назначению и использованию можно разделить на 2 большие группы – системное и прикладное.

Системное ПО – совокупность программ для обеспечения работы компьютера.

Прикладное ПО – комплекс программ для решения задач определённого класса конкретной предметной области.

В то же время, в состав основных системных средств – операционных систем (ОС) – входят и компоненты прикладного ПО, например, простейшие текстовые редакторы, программы для работы с мультимедиа, с Интернетом и другие. В составе прикладного ПО можно выделить особую группу – средства разработки ПО, назначение которых – разработка новых программ, в том числе



и операционных систем. Так что эти 2 большие группы имеют некоторое пересечение по своим компонентам. Многие другие группы ПО также имеют общие компоненты, например система автоматизации управления предприятиями работает с системой управления базой данных (СУБД), имеет средства разработки, может обмениваться информацией с САПР, и т. п.

### Классификация программного обеспечения

#### Классификация программного обеспечения



### 10.1. Рыночная классификация ПО

**Лицензионные** или **коммерческие программы** (*commercial ware*) распространяется на коммерческой основе. Для получения права на использование таких программ необходимо купить экземпляр дистрибутива такой программы. Порядок использования программы оговаривается в лицензионном соглашении, подлинность ПО подтверждается сертификатом (**Certificate of Authenticity**), имеющим уникальный код каждой копии дистрибутива (product key).

**Бесплатные программы** (*freeware*) – распространяются авторами бескорыстно. Чаще всего это небольшие программы, написанные одним или несколькими программистами для себя, и предложенные ими для общего пользования. Такие программы можно найти в Интернете.

**Условно-бесплатные программы** (*shareware*) занимают промежуточное положение между коммерческими и бесплатными программами. Самая массовая группа программ, в которую входят практически все утилиты, а часто и весьма серьезные, нужные многим программные пакеты. Эти программы предоставляются бесплатно, однако по истечении определенного срока необходимо заплатить их автору или распространителю небольшую сумму. В противном случае программа либо не загружается, либо начинает надоедать просьбами о её регистрации.

**Пробные версии программ** (*trialware*) – это полноценные версии коммерческих систем, которые можно использовать какое-то время бесплатно. По истечении этого времени программы прекращают работать.

**Демо-версии** (*demoware*) – демонстрационные версии коммерческих программ и игр с ограниченными возможностями. Например, с выключенной функцией сохранения документов в текстовом редакторе, с ограниченным количеством записей в базе данных информационной системы или с несколькими турами в играх.

## 10.2. Операционные системы

**Операционная система** – комплекс программ, обеспечивающих работу компьютера.

Функции современной ОС для ПК:

1. Управление конфигурацией компьютера.
2. Управление процессами, потоками и заданиями.
3. Управление памятью.
4. Обеспечение информационной безопасности.
5. Управление подсистемой ввода-вывода.
6. Управление внешней памятью.
7. Управление файловой системой.
8. Поддержка сетей.

В управлении конфигурацией компьютера (в ОС **Windows**) ключевую роль играют данные, сохраняемые в реестре этой системы (база данных

общесистемных и пользовательских параметров) и инструментарий управления **Windows (WMI)**.

Процесс в ОС – исполняемая программа и ее данные в закрытом адресном пространстве, данные защиты и прочие объекты, принадлежащие процессу.

Поток – некоторый компонент внутри процесса, который получает процессорное время для выполнения.

Задание (job) – совокупность процессов, управляемых как единая группа. ОС управляет виртуальной памятью, которую проецирует на физическую память компьютера.

Информационная безопасность предусматривает защиту любых разделяемых системных объектов (файлов, каталогов, процессов, потоков и т. д.), аутентификацию паролей при регистрации, предотвращение несанкционированного доступа, аудит событий в системе (ведение журналов событий).

Диспетчер ввода-вывода ОС подключает приложения и системные компоненты к виртуальным, логическим и физическим устройствам, а также определяет инфраструктуру, поддерживающую драйверы устройств.

Управление внешней памятью включает организацию работы с жесткими и гибкими дисками, с оптическими дисками, магнитными лентами, флэш-памятью и другими устройствами.

Управление файловой системой, определяющей методы хранения информации на дисках, важнейшая функция любой ОС. Многие ОС поддерживают работу с несколькими файловыми системами, например, **Windows 2000, XP** и его более новые версии могут управлять файловыми системами **NTFS** (основная для магнитных дисков), **FAT12, FAT16, FAT32** (были основными для **DOS** и **Windows 3.1 - Me**), **CDFS** и **UDF** (для лазерных дисков). Для управления используется драйвер файловой системы, который через драйверы устройств внешней памяти работает с информацией логических томов. Основные функции работы с информацией файловой системы –

**CreateFile** (создать файл), **ReadFile** (читать файл) и **WriteFile** (записать в файл).

Поддержка сетей присутствует во всех современных ОС. В ОС **Windows** появилась с версии 3.11. К четырем базовым типам сетевого программного обеспечения относятся сетевые службы, программные интерфейсы сетевого программного обеспечения (сетевые API), протоколы и драйверы устройств сетевых адаптеров. Многоуровневая архитектура поддержки сетей обеспечивает ее легкую расширяемость при появлении новых сетевых протоколов и нового оборудования.

### **Развитие операционных систем для ПК**

Первый персональный компьютер PC появился в конце 1981 года (IBM 5150 PC: 4.77-MHz Intel 8088 CPU, 64KB RAM, 40KB ROM, 5.25-дюймовый флоппи-диск) с операционной системой PC-DOS 1.0. Далее на рынке появились так называемые IBM-совместимые персональные компьютеры. Операционная система этих компьютеров называлась **MS DOS 1.0** – корпорация **Microsoft** предоставила в распоряжение фирм, производящих эти машины, точную копию операционной системы PC-DOS. Всего было выпущено 12 версий MS DOS: 1.0, 1.1, 2.0, 3.0, 3.3, 4.0, 4.01, 5.0, 6.0, 6.2, 6.21, 6.22. Версия 7 входила в состав **Windows 95** и отдельно не выпускалась.

Все версии **MS DOS** были однозадачными ОС с текстовым интерфейсом и управлением из командной строки. Правда, основной программой того времени, облегчавшей работу с файловой системой, был **Norton Commander** – файловый менеджер, имевший два окна для показа папок (directory – каталогов, как их тогда называли) и файлов.

Основой **MS DOS** был 1 файл – командный процессор **command.com**. Существовали еще 2 скрытых файла – **ibmbio.com** и **ibmdos.com** (или их аналоги) – модуль расширения BIOS и модуль обработки прерываний, а также достаточное количество утилит (служебных программ) и драйверов внешних устройств (программ для работы с клавиатурой, мышкой, принтерами и пр. периферией).

Некоторые команды **MS DOS** могут использоваться и в настоящее время в командном интерпретаторе **Windows** (окно программы **cmd.exe**, похожее на сеанс **MS DOS**, но им не являющееся, обеспечивающем текстовый интерфейс между пользователем и этой ОС.

Для работы с командной строкой надо нажать клавиши Windows + R.

2. В появившемся окне напишите "cmd" и нажать Enter:

Некоторые команды

dir – просмотреть список файлов текущего каталога (папки);

ver – узнать версию операционной системы;

cd <имя> – сменить каталог;

md <имя> – создать подкаталог;

C: \> D: – сменить текущий;

copy... – копирование файлов;

time, date – настроить время и дату на системных часах ПК и др.

Полный список выводится по команде help, содержит 86 команд.

Первые версии **Windows** использовали **MS DOS** для своей инсталляции (установки на компьютер) или запускались из MS DOS.

Современные версии **Windows** не используют для этих целей MS DOS, но могут запускать некоторые из программ, написанных для **MS DOS** в окне или в полноэкранном режиме. Однако, сложные программы для **MS DOS** (например, игры), не запускаются в последних версиях **Windows**. Для этой цели можно использовать программу **DosBox** (Free Software Foundation, Inc), версии которой существуют для многих современных ОС. История развития ОС **Microsoft Windows** началась 20 ноября 1985 г. с выходом первой версии. Выпуском этой версии фирма **Microsoft** заявила о своих намерениях разработки перспективной многозадачной операционной системы с графическим интерфейсом для быстро развивающихся IBM-совместимых персональных компьютеров.

**Windows 95** уже не графическая оболочка пользователя для **MS DOS** (как это было в предыдущих версиях), а полноценная многозадачная 32-

разрядная операционная система. Хотя пользователи могли видеть окно **MS DOS** в процессе загрузки, система заменяла собой **MS DOS 7.0** после того, как загружалась полностью.

Система **Windows XP** создавалась на основе усовершенствованного кода **Windows 2000**, причем были разработаны различные версии для пользователей домашних компьютеров и бизнес-пользователей: **Windows XP Home Edition** и **Windows XP Professional**. 30 января 2007 года была выпущена в продажу еще одна написанная полностью с нуля ОС – **Microsoft Windows Vista**. 22 октября 2009 года в продаже официально появилась Windows 7.

### 10.3. Файловые системы

*Файловая система* – методы организации хранения и использования информации во внешней памяти компьютера, которой являются магнитные и лазерные диски, флэш-память и прочие ее виды.

Каждая операционная система использует одну основную файловую систему, а также дополнительно может работать с некоторыми другими файловыми системами.

*Файл* – упорядоченная совокупность информации на диске, имеющая свое имя. Для файловой системы **NTFS** теоретически файл – единица информации, имеющая номер в Master File Table (**MFT**, общей таблице файлов), однако на практике все же основным атрибутом файла – его имя.

Сохранение информации о файлах обеспечивает файловая система. Правила именования файлов, способы доступа к данным файла, структура его данных зависят как от типа используемой файловой системы, так и от типа файла. Файловая система берет на себя размещение информации файла на диске, сохранение информации об этом в таблице файлов диска и обеспечение доступа к этим данным.

Кроме имени каждый файл может иметь и другие свойства:

- расширение имени файла, характеризующее его тип (символы после последней точки, справочник типов см. <http://open-file.ru/>);
- атрибуты (архивный, только чтение, скрытый, шифрованный и пр.);

- время создания, изменения, последнего открытия;
- права доступа (безопасность);
- метаданные (название, тема, ключевые слова, авторы и пр.).

Каждый логический раздел жесткого диска может быть отформатирован в той или иной файловой системе. Например, основной файловой системой ОС **Windows** на ядре **NT** является **NTFS**. Однако ОС **Windows** может также работать и с информацией, хранящейся на разделах диска с файловой системой **FAT** и **FAT32**

**FAT32** является самой старой из рассматриваемых файловых систем и чаще всего используется на переносных флеш-накопителях — флешках или SD-картах.

**NTFS** используется **Windows** как основная файловая система для диска, на котором установлена эта операционная система, и она также отлично подходит и для других встроенных дисков и разделов жесткого диска компьютера под управлением **Windows**.

**exFAT** является более современным аналогом устаревшей системы **FAT32** и её поддерживает большее количество устройств, чем **NTFS**, но всё ещё не так много, как «классическую» **FAT32**.

К преимуществам в данном случае можно отнести тот факт, что **FAT32** стала своеобразным стандартом и используется до сих пор во всех сменных носителях по умолчанию. Если вы сегодня купите флешку или SD-карту — «с завода» на ней будет установлена файловая система **FAT32**. Сделано это в первую очередь для того, что бы ваш сменный носитель могли поддерживать не только современные компьютеры и гаджеты, но и старые устройства и игровые консоли, которые имеют **USB** порт и умеют работать только с файловой системой **FAT32**.

Однако, в связи с возрастом этой системы, она имеет и некоторые недостатки, основным из которых является ограничение по размеру файла и всего тома. Каждый отдельный файл в этой файловой системе не может иметь

размер больше 4 гигабайт, а весь раздел с файловой системой FAT32 не может быть более 8 терабайт.

**NTFS** — это более современная и технологичная файловая система, о чем говорит даже расшифровка аббревиатуры её названия -«new technology file system». Больше всего её любит операционная система Windows, что в общем и не удивительно — ведь и то, и другое, разработано компанией Microsoft.

Начиная с версии операционной системы от Microsoft под названием XP, в которой система NTFS впервые и стала стандартном, при установке Windows диалоговое окно обязательно попросит вас отформатировать системный раздел именно в эту файловую систему. На данный момент считается, что об ограничениях файловой системы NTFS вам, теоретически, не придётся беспокоиться ещё долгие годы.

Помимо отсутствия серьезных ограничений на размер файла и раздела, NTFS имеет несколько дополнительных преимуществ, таких как: поддержка прав доступа к файлам (для повышения безопасности данных), журналирование изменений (для восстановления файловой структуры в случае сбоя), шифрование, дисковые квоты, жёсткие ссылки, и другие современные функции, которые делают NTFS идеальной для системного диска.

К несчастью, NTFS не совместима с большинством других операционных систем, так как она была разработана под требования ОС Windows. С ней отлично работают все версии операционной системы от Microsoft начиная с XP и заканчивая последней на сегодняшний день Windows 10, но вот другие ОС имеют значительные ограничения при работе с ней.

К примеру, Mac OS может только считывать данные с дисков с файловой системой NTFS, но не может записывать на них. Некоторые редкие дистрибутивы линукса могут иметь возможность записи на диски с NTFS, но большинство всё же ограничены лишь чтением информации. Ни одна из версии Playstation не может работать с NTFS, так же как и Xbox 360 от Microsoft, и лишь новая Xbox One поддерживает эту файловую систему.



**exFAT** была впервые представлена в 2008 году и является наиболее современной из рассматриваемых в этой статье файловых систем, её поддержка была добавлена в Windows начиная с версии XP при помощи обновлений ОС.

Файловая система exFAT была создана и оптимизирована для использования на внешних накопителях — флешках, SD-картах и внешних жестких дисках, и призвана заменить собой устаревшую систему FAT32. Она является самой легкой и простой файловой системой благодаря отсутствию различных специализированных особенностей, присущих NTFS, равно как и практически полному отсутствию ограничений на размер файла и раздела, которые имеет FAT32.

Также exFAT обладает лучшей совместимостью с различными операционными системами, чем NTFS, и устройства с ней отлично читаются и перезаписываются как на Windows и Mac OS, так и на Linux (при условии установки некоторого программного обеспечения).

Так как exFAT поддерживается операционной системой Mac OS по умолчанию, то вероятно она будет поддерживаться и большинством других современных девайсов, с которыми работают устройства от Apple, например, цифровыми камерами.

Современные версии игровых консолей, такие как Xbox One и Playstation 4, также поддерживают устройства с файловой системой exFAT, в отличие от их предыдущих версий (Xbox 360 и Playstation 3).

При организации разделов в первый сектор диска (цилиндр 0, головка 0, сектор 1) заносится главная загрузочная запись – **Master Boot Record (MBR)**. В ней содержатся:

- 1) исполняемый код – программа начальной загрузки (внесистемный загрузчик – non-system bootstrap);

- 2) таблица разделов – таблица разбиения диска (Disk Partition Table).

В таблице разбиения диска содержатся сведения о том, с каких цилиндров, головок и секторов начинаются и какими заканчиваются имеющиеся на диске разделы. В этой таблице также содержатся указания для

системной **BIOS**, какой из разделов является загрузочным, т.е. где следует искать основные файлы операционной системы. Сектора и дорожки формируются во время низкоуровневого форматирования при изготовлении жесткого диска (размер сектора равен 512 байт).

Поскольку файловые системы являются общим хранилищем файлов, принадлежащих разным пользователям, системы управления файлами должны обеспечивать авторизацию доступа к файлам. В общем виде подход состоит в том, что по отношению к каждому зарегистрированному пользователю данной операционной системы для каждого существующего файла указываются действия, которые разрешены или запрещены данному пользователю или группе пользователей. В современных ОС для ПК predeterminedены 3 основные группы с разными правами: **администраторы, опытные пользователи, пользователи.**

В файловых системах **FAT** и **NTFS** может существовать несколько логических дисков (каждый винчестер может быть одним разделом или может быть разбит на несколько логических дисков, общие сетевые ресурсы при необходимости также могут быть подключены как логические диски). Каждый логический диск имеет имя – латинская буква с двоеточием. А: и В: зарезервированы для флоппи-дисководов; жесткие, лазерные, сетевые диски – С:, D: и т. д.. С этого имени начинается дерево каталогов диска (см. рисунок 3.3). Полное имя файла начинается с имени диска, далее может идти перечисление папок (путь к файлу), последним идет имя фай-ла. Например, E:\Program Files\Microsoft Office\Office11\ Winword.exe. Таким образом, в этих системах может существовать несколько изолированных файловых деревьев. Каждый файл имеет расширение (символы после последней точки в имени файла – Winword.exe, Заяв-ка12.12.2006.doc).

На ПК может использоваться также ОС **Linux**, которая имеет собственную файловую систему. Принципиальное отличие этой файловой системы – отсутствие имен дисков и расширений файлов. Это централизованная файловая система, вся совокупность каталогов и файлов

представлена как единое дерево. Для подключения дисков используется операция монтирования дисков к общей файловой системе.

CD и DVD диски используют файловые системы, предназначенные специально для работы с оптическими дисками (ISO 9660, **CDFS**, **UDF** и пр.).

В файловой системе **FAT** каждый логический раздел диска (том) имеет собственную таблицу размещения файлов (File Allocation Table – **FAT**), которая выполняет две функции: содержит информацию распределения данных каждого файла по кластерам раздела диска в форме списка связей модулей распределения и указывает, какие кластеры свободны.

### Структура тома в **FAT**

Том, отформатированный для использования файловой системы **FAT**, размечается по **кластерам** (минимальная адресуемая единица в файловой системе). Размер кластера по умолчанию определяется размером тома. В название версии **FAT** входит число, которое указывает разрядность, применяемую для идентификации кластеров на диске. 12-разрядный идентификатор кластеров в **FAT12** ограничивает размер дискового раздела 212 (4096) кластерами. **FAT16** – за счет 16-разрядных идентификаторов кластеров – может адресовать до 216 (65 536) кластеров. В **Windows 2000** размер кластера **FAT16** варьируется от 512 байт до 64 Кб, поэтому размер **FAT16**-тома ограничен 4 Гб. Размер кластеров, используемых **Windows 2000**, зависит от размера тома

Пользователь имеет возможность указать другой размер кластера, однако устанавливаемый размер не может быть меньше размера по умолчанию. Ограничение системы **FAT** на размер логического диска составляет 4 Gb. Файловую систему **FAT**, не рекомендовалось использовать для томов, размер которых превышал 511 Mb.

Файловая система **FAT32** впервые появилась в **Windows 95 OSR2**, затем использовалась в **Windows 98** и **Windows ME**. **FAT32** использует 32-разрядные идентификаторы кластеров, но при этом резервирует старшие 4 бита, так что эффективный размер идентификатора кластера составляет 28 бит. Поскольку

максимальный размер кластеров **FAT32** равен 32 Кб, теоретически **FAT32** может работать с 8-терабайтными томами

Помимо большего предельного числа кластеров преимуществом **FAT32** перед **FAT12** и **FAT16** является тот факт, что место хранения корневого каталога **FAT32** не ограничено predetermined областью тома, поэтому его размер не ограничен. Кроме того, для большей надежности **FAT32** хранит вторую копию загрузочной записи. В **FAT32**, как и в **FAT16**, максимальный размер файла равен 4 Гб, поскольку длина файла в каталоге описывается 32-битным числом.

Файловая система **NTFS** обеспечивает производительность, надежность и совместимость, которые не в состоянии предоставить файловая система **FAT**. **NTFS** разрабатывалась с целью обеспечения высокой скорости и надежности выполнения операций с файлами (включая чтение, запись, поиск) при использовании дисков больших объемов. **NTFS** использует 64-разрядные индексы кластеров. Это позволяет **NTFS** адресовать тома размером до 16 экзабайт (16 миллиардов Гб).

**NTFS** поддерживает контроль доступа к папкам и файлам как для локальных, так и сетевых ресурсов.

Как и другие файловые системы, **NTFS** делит все полезное место диска на кластеры – адресуемые блоки данных, размер которых может быть от 512 байт до 64 Кбайт, стандартом считается кластер размером 4 Кбайт.

Одно из важнейших свойств **NTFS** – *восстанавливаемость*. При неожиданной остановке системы целостность метаданных тома **FAT** может быть утрачена, что вызовет повреждение структуры каталогов и значительного объема данных. **NTFS** ведет журнал изменений метаданных путем протоколирования транзакций, поэтому целостность структур файловой системы может быть восстановлена без потери информации о структуре файлов или каталогов (однако данные файлов могут быть потеряны). Транзакция – операция, выполняющаяся по принципу «все или ничего». При неудачном выполнении операции в файловой системе происходит *откат* (roll back), после

которого система воз-вращается в исходное состояние, в котором она была до начала транзакции.

В **NTFS** все данные, хранящиеся на томе, содержатся в файлах. Это относится и к структурам данных, используемым для поиска и выборки файлов, к начальному загрузочному коду и к битовой карте, в которой регистрируется состояние пространства всего тома (метаданные **NTFS**). Хранение всех видов данных в файлах позволяет файловой системе легко находить и поддерживать данные, а каждый файл может быть защищен дескриптором защиты. Кроме того, при появлении плохих секторов на диске **NTFS** может переместить файлы метаданных. Диск **NTFS** делится на две части. Первые 12% диска отводятся под **MFT** зону – пространство, в котором растёт файл **MFT** (Master File Table, главная таблица файлов). Это сделано для того, чтобы файл **MFT** не был фрагментирован. Но когда все остальное место на диске заполняется – зона **MFT** может несколько раз уменьшаться в два раза, пока это возможно (на больших дисках эта зона – десятки Гб и полностью под файл **MFT** не используется).

#### Структура тома в NTFS

**MFT** реализована как массив записей о файлах и папках (номер в **MFT**, имя, размер, положение на диске отдельных фрагментов и т. д). Размер каждой записи о файле фиксирован и равен 1 Кб. Кроме **MFT** в корневом каталоге каждого тома **NTFS** имеется набор файлов метаданных с информацией, необходимой для реализации структуры файловой системы. Имена всех файлов метаданных **NTFS** начинаются со знака доллара (\$), хотя эти знаки скрыты. Так, имя файла **MFT** – **\$MFT** . Остальные файлы **NTFS**-тома являются обычными файлами и каталогами.

#### Файловые системы CD и DVD дисков – **CDFS** и **UDF**.

**CDFS** или файловая система CD-ROM, имеет относительно простой формат, определенный в 1988 году применительно к носителям CD-ROM, предназначенных только для чтения. В ОС **Windows** реализована **CDFS** совместимая со стандартом ISO 9660, которая поддерживает длинные имена

файлов в соответствии с ISO 9660 Level 2. Формат **CDFS** очень прост и поэтому ему присущ ряд ограничений:

- длина имен файлов и каталогов не должна превышать 32 символов;
- глубина вложения каталогов не может превышать 8 уровней.

**CDFS** считается устаревшим форматом, поскольку принят новый стандарт для компакт-дисков – Universal Disk Format (**UDF**). **Windows** поддерживает файловую систему **UDF**, совместимую со стандартом ISO 13346. Ассоциация **OSTA** (Optical Storage Technology Association) определила **UDF** в 1995 году как формат магнитооптических носителей, главным образом DVD-ROM, предназначенный для замены **CDFS**.

## 11. ОС MICROSOFT WINDOWS

История развития ОС **Microsoft Windows** началась 20 ноября 1985 г. с выходом первой версии. Выпуском этой версии фирма **Microsoft** заявила о своих намерениях разработки перспективной многозадачной операционной системы с графическим интерфейсом для быстро развивающихся IBM-совместимых персональных компьютеров.

**Windows 1.0** могла использовать 256 цветов, имела окна изменяемого размера, область свернутых программ (первоначальную концепцию панели задач), способность настраивать вид окон. **Microsoft** включила в нее то, что мы теперь называем **Панелью управления**. Она имела элементы интерфейса, которые присутствуют в сегодняшних версиях **Windows** как поля редактирования, кнопки, полосы прокрутки и пункты меню.

В интерфейсе **Windows 1.0** окна могли быть развернуты, свернуты или расположены рядом (tiled). Окна не могли пересекаться, что было неудобно при работе с большим их количеством.

В систему были включены приложения: менеджер файлов, календарь, картотека, записная книжка, калькулятор, часы и телекоммуникационные программы. Пользователь мог переключаться между несколькими запущенными программами, т.е. система была многозадачной.

**Windows for Workgroups 3.11** с поддержкой одноранговых сетей вышел 15 февраля 1994 г.

Чуть раньше, в августе 1993 г. появилась новая ОС – **Windows NT 3.1** («Новая технология» – New Technology, так официально трактовали аббревиатуру **NT** в **Microsoft**). **Microsoft** создал новую операционную систему с нуля. **Windows NT** был первой операционной системой **Windows**, которая была предназначена для организации выделенного файлового сервера в составе локальной сети и включала поддержку высокопроизводительных клиент-серверных бизнес-приложений. Операционная система имела ряд новых ключевых особенностей: приоритетный многозадачный планировщик для **Windows**-приложений, интегрированная работа с сетями, разграничения прав пользователей для сервера домена, поддержка **OS/2** и **POSIX** подсистем, поддержка многопроцессорной архитектуры, новая более надежная файловая система **NTFS**.

Реально ОС **Windows** начала широко использоваться в 1995 г. с выходом версии **Windows 95**, имевшей новый графический интерфейс, очень удобный в работе. Основная масса прикладного программного обеспечения стала выпускаться для этой ОС. **Windows 95** уже не графическая оболочка пользователя для **MS DOS** (как это было в предыдущих версиях), а полноценная многозадачная 32-разрядная операционная система. Хотя пользователи могли видеть ок-но **MS DOS** в процессе загрузки, система заменяла собой **MS DOS 7.0** по-сле того, как загружалась полностью. ОС на ядре **NT** получила широкое распространение среди обычных пользователей с выпуском 25 октября 2001 г. ОС **Microsoft Windows XP**, которая имела новый графический интерфейс, использовала, как основ-ную, файловую систему **NTFS** и имела возможность разграничения прав пользователей на локальном компьютере. Система **Windows XP** создавалась на основе усовершенствованного кода **Windows 2000**, причем были разработаны различные версии для пользователей домашних компьютеров и бизнес-пользователей: **Windows XP Home Edition** и **Windows XP Professional**.

30 января 2007 года была выпущена в продажу еще одна написанная полностью с нуля ОС – **Microsoft Windows Vista**.

22 октября 2009 года в продаже официально появилась Windows 7.

22 января 2015 года представители компании Microsoft провели предфинальную презентацию новой операционной системы Windows 10, разработки которой велись на протяжении 2014 года. Прямая трансляция презентации велась на сайте компании.

Windows 10 — операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT. После Windows 8.1 система получила номер 10, минуя 9. Серверный аналог Windows 10 — Windows Server 2016.

Система призвана стать единой для разных устройств, таких как персональные компьютеры, планшеты, смартфоны, консоли Xbox One и пр. Доступна единая платформа разработки и единый магазин универсальных приложений, совместимых со всеми поддерживаемыми устройствами. Windows 10 поставляется в качестве услуги с выпуском обновлений на протяжении всего цикла поддержки. В течение первого года после выхода системы пользователи могли бесплатно обновиться до Windows 10 на устройствах под управлением лицензионных копий Windows 7, Windows 8.1 и Windows Phone 8.1. Среди значимых нововведений — голосовая помощница Кортана, возможность создания и переключения нескольких рабочих столов и др. Windows 10 — последняя «коробочная» версия Windows, все последующие версии будут распространяться исключительно в цифровом виде.

Пользовательское соглашение Windows 10 позволяет компании Microsoft собирать многочисленные сведения о пользователе, историю его интернет-деятельности, пароли к точкам доступа, данные, набираемые на клавиатуре, и многое другое.

По состоянию на июль 2017 года Windows 10 имеет долю среди используемых в мире операционных систем для доступа к сети Интернет 35,7



% и занимает первое место в мире по популярности, опередив в апреле предыдущего лидера — Windows 7.

После Windows 8.1 система получила номер 10, минуя 9. Как объяснил глава Microsoft по маркетингу Windows Тони Профет, пропуск девятого номера в наименовании операционной системы Windows не связан с совместимостью. Сторонние приложения определяли бы её, как систему из семейства Windows 9х, которые выпускались с 1995 по 2000 год. Множество программ для определения версии Windows используют подобный код:

```
if (osName.startsWith("Windows 9")) {  
    /* Windows 95 & Windows 98 */  
}  
else {  
}
```

### 11.1. Интерфейс Windows 10

Microsoft продолжает попытки усидеть на двух стульях. Пытается создать единый графический интерфейс и для компьютеров и для мобильных устройств. В сравнении с Windows 8, сделан шаг назад относительно традиционного для Windows меню "Пуск". Если в Windows 8 этого меню совсем не было, а вместо него был "Стартовый экран", то в Windows 10 вернули огрызок меню "Пуск" (каким оно было в Windows 7) и туда же прилепили уменьшенный в размерах "Стартовый экран"

В интерфейс добавлен новый визуальный элемент для переключения между открытыми программами. Task View (Представление задач). В нем показаны все окна на весь экран, а кроме того там же можно создать дополнительные Рабочие столы

Открыть этот элемент можно кнопкой с панели задач, или с клавиатуры **Win + Tab**. Теперь и в Windows есть такая же фишка какая давно уже есть в Linux и даже в Mac OS. В этом представлении, в правом нижнем углу, есть кнопка для создания дополнительных Рабочих столов. Новый Рабочий стол можно создать и с клавиатуры - **Win + Ctrl + D**. Созданные

Рабочие столы будут отображаться внизу экрана Task View. Между Рабочими столами можно переходить не только через Task View, но и с клавиатуры: **Ctrl+Win+стрелка влево/вправо**.

Доработан элемент графического интерфейса Snap (появившийся еще в Windows 7). В Windows 10 туда добавлены две новые возможности. Теперь можно не только два окна (по горизонтали) разместить, но и четыре. Для того, чтобы Snap разместил окно программы только на четверть экрана, нужно перетащить это окно в один из углов экрана. Для размещения двух окон, окно нужно перетаскивать на боковую грань экрана.

Наверное самое главное новшество это возможность голосового управления Windows - голосовая помощница Cortana. Но она есть пока только в английской версии.

А еще новый браузер Edge, но он пока не конкурент Мозилле и Гуглу.

Облачное хранилище OneDrive теперь вкорячено в систему плотнее. Отключить его посложнее чем раньше.

Такая же нездоровая канитель с "Защитником Windows" - через графический интерфейс его уже не отключить. Нужно лезть в реестр.

### **Windows наблюдает за вами**

Это не с десятки началось конечно. Просто с каждой новой версией увеличивается количество отправляемой с вашего компьютера информации.

Однако Microsoft, начиная с версии 8 дает вам возможность отключать если не все, то по крайней мере многие параметры "слежки" за вашим компьютером и за вами.

В установщике Windows 10 (и 8) есть кнопка "Настроить", которая выводит три последовательных экрана с более чем десятком параметров слежения - все эти параметры можно отключить. Кроме этого, уже после установки можно отключить еще несколько параметров через настройки.

Просто будьте внимательны и не нажимайте кнопку "Настройки по умолчанию" в установщике.

## 12. ЗАЩИТА ОТ ВИРУСОВ

**Компьютерный вирус** – вредоносная программа, способная к саморазмножению и распространению при передаче информации с одного компьютера на другой.

Официально считается что термин «компьютерный вирус» впервые употребил сотрудник Лехайского университета (США) Ф. Коэн в 1984 г. на 7-й конференции по безопасности информации, проходившей в США. В настоящее время кроме классических компьютерных вирусов существуют также сетевые вирусы (черви и троянские программы), а также прочие вредоносные программы.

Типы вирусов:

- файловые;
- загрузочные;
- макро-вирусы;
- скриптовые;
- сетевые.

Файловые вирусы различными способами внедряются в исполняемые файлы, либо создают файлы-двойники.

Пример, показывающий, насколько сложными являются подобные программы – вирус WinNT.RemEx (Remote Explorer). При запуске файла вируса его код копируется в системный каталог WinNT System32\Drivers под именем IE403R.SYS и запускается как системный сервис, который один раз в 10 минут запускает одну из своих подпрограмм – либо заражения, либо «заметания следов». Процедура заражения сканирует случайно выбранные локальные и доступные сетевые папки, ищет EXE-файлы и заражает их. При заражении вирус компрессирует файл-жертву, записывает в него свой код и добавляет компрессированный первоначальный код файла в его конец. При запуске зараженного EXE-файла вирус копирует его во временный файл, распаковывает и выполняет исходный файл программы. Вирус компрессирует файлы методом «deflate» и использует для этого встроенную в свой код библиотеку GZIP. В

зависимости от случайного счетчика вирус также портит случайно выбранные файлы: компрессирует их тем же методом и затем шифрует криптоалгоритмом средней сложности. Процедура «заметания следов» запускается следом за заражением и уничтожает следы жизнедеятельности вируса. Вирус закрывает сообщения об ошибках, произошедших при заражении файлов, и выполняет другие действия (по данным Лаборатории Касперского <http://www.kav.ru>).

Загрузочные вирусы записывают себя в загрузочный сектор диска (boot-сектор), либо в сектор, содержащий системный загрузчик винчестера (Master Boot Record), либо меняют указатель на активный boot-сектор. Данный тип вирусов был достаточно распространен в 1990-х годах, но практически исчез с переходом на 32-разрядные операционные системы и отказом от использования дискет как основного способа обмена информацией. Теоретически возможно появление загрузочных вирусов, заражающих оптические диски и USB-флэш-память, но на текущий момент такие вирусы не обнаружены.

Макро-вирусы заражают файлы данных различных популярных программ. Многие текстовые, табличные и графические редакторы, системы проектирования, имеют свои языки программирования для работы с различными объектами, присутствующими в файлах данных этих систем (например, Visual Basic for Application (VBA) в системе **Microsoft Office**). Текст программ на языке VBA сохраняется в модулях процедур (макросах), хранящихся в файлах документов (см. в меню системы **Microsoft Office 2003** раздел Сервис - Макрос). Макровирусы являются программами на языках, встроенных в такие системы обработки данных. Например, при открытии документа Word проверяет его на наличие процедуры с именем AutoOpen, если она присутствует, то Word выполняет ее. Для своего размножения вирусы этого класса используют возможности языков этих систем и при их помощи переносят себя из одного зараженного файла (документа, таблицы или другого файла данных) в другие. По данным Лаборатории Касперского количество макровирусов – несколько сотен.

Скриптовые вирусы написаны на различных скрипт-языках (VBS – Visual Basic Script, JS – Java Script, BAT, PHP и пр.), которые широко используются в ОС **Windows** и Web-программировании. Они либо заражают другие скрипт-программы, присутствующие в виде программных файлов или фрагментов текста html-страниц, либо являются частями многокомпонентных вирусов. Пример: Virus.VBS.Redlof – написан на языке VBS и зашифрован. При первом запуске создает файл со своим исполняемым кодом в системном каталоге **Windows** с именем Kernel.dll. Кроме того, копирует себя во все каталоги на других дисках зараженного компьютера в виде файла на-стройки отображения файлов и папок – folder.htt. Зараженный файл folder.htt получает управление и копируется вирусом во все папки при их открытии при помощи Проводника. Вирус дописывает себя во все НТМ файлы, находящиеся в каталоге **Windows** \web и таким образом он также получает управление при открытии данных файлов.

**Сетевые вирусы** используют для своего распространения протоколы и команды компьютерных сетей и электронной почты.

Сетевые черви подразделяются на почтовые (Email-Worm), использующие интернет-пейджеры типа ICQ (IM-Worm), использующие IRC-средства Интернет-общения (IRC-Worm), заражающие Интернет-серверы и прочие.

Почтовые черви после инсталляции в системе могут рассылать себя по всем адресам, обнаруженным в адресной книге **Windows** и в файлах электронной почты. Другие виды сетевых червей могут рассылать URL-ссылку на себя или другие сайты по адресам, обнаруженным в базах Интернет-пейджеров или IRC-клиентов, открывать полный сетевой доступ на диски компьютера и пр.

Троянская программа – любая программа, которая втайне от пользователя выполняет на компьютере какие-либо нежелательные для него действия. По своим функциям подразделяются на:

бэкдор (Backdoor) – несанкционированный доступ к удалённому администрированию компьютера;

PSW-троянцы – похищение паролей;

Trojan Clicker – открывает на компьютере веб-ссылки без ведома пользователя;

Trojan Downloader – загрузка из интернета каких-либо файлов без ведома пользователя;

Trojan-Dropper – устанавливает без ведома пользователя другие вредоносные программы на зараженный компьютер и запускает их;

Trojan Proxy – запускает без ведома пользователя прокси-сервер на зараженном компьютере;

прочие троянцы.

К вредоносным программам относятся также **руткиты** (rootkits) – утилиты, используемые хакерами для сокрытия своей вредоносной деятельности на зараженном компьютере. Для этого они модифицируют операционную систему компьютера. **Руткиты** не только скрывают свое собственное присутствие, но также и действия, которые предпринимает злоумышленник. **Руткит** может скрывать присутствие других вредоносных программ на компьютере, изменяя файловые данные, ключи реестра или активные процессы.

#### ***Особенности алгоритмов работы вирусов:***

- резидентность;
- использование стелс-алгоритмов;
- самошифрование и полиморфичность;
- использование нестандартных приемов.

Резидентный вирус при инфицировании компьютера оставляет в оперативной памяти свою резидентную часть, которая затем перехватывает обращения операционной системы к объектам заражения и внедряется в них. Резидентные вирусы находятся в памяти и являются активными вплоть до выключения компьютера или перезагрузки операционной системы.

Резидентными можно считать макро-вирусы, поскольку они постоянно присутствуют в памяти компьютера на все время работы зараженного редактора.

Нерезидентные вирусы не заражают память компьютера и сохраняют активность ограниченное время. Некоторые вирусы оставляют в оперативной памяти небольшие резидентные программы, которые не распространяют вирус. Такие вирусы считаются нерезидентными.

Использование стелс-алгоритмов позволяет вирусам полностью или частично скрыть себя в системе. Наиболее распространенным стелс-алгоритмом является перехват запросов ОС на чтение/запись зараженных объектов. Стелс-вирусы при этом либо временно лечат их, либо «подставляют» вместо себя незараженные участки информации. В случае макро-вирусов наиболее популярный способ – запрет вызовов меню просмотра макросов.

Самошифрование и полиморфичность используются практически всеми типами вирусов для того, чтобы максимально усложнить процедуру детектирования вируса. Полиморфные вирусы (polymorphic) – это достаточно труднообнаруживаемые вирусы, не имеющие сигнатур, т.е. не содержащие ни одного постоянного участка кода. В большинстве случаев два образца одного и того же полиморфного вируса не будут иметь ни одного совпадения. Это достигается шифрованием основного тела вируса и модификациями программы-расшифровщика.

***По деструктивным возможностям вирусы подразделяются на:***

безвредные, т.е. никак не влияющие на работу компьютера (кроме уменьшения свободного места на диске в результате своего размножения);

неопасные, влияние которых ограничивается уменьшением свободного места на диске и графическими, звуковыми и пр. эффектами;

опасные вирусы, которые могут привести к серьезным сбоям в работе компьютера;

очень опасные, в алгоритм работы которых заведомо заложены процедуры, которые приводят к порче программ, уничтожению файлов.

Полную информацию о многочисленных известных вирусах можно найти на сайтах фирм, производителей антивирусных программ, например, на сайте Лаборатории Касперского <http://www.kav.ru>.

### 13. АНТИВИРУСНЫЕ ПРОГРАММЫ

Для борьбы с компьютерными вирусами используются антивирусные программы. Так как компьютеры простых пользователей, не подключенные к Интернету, вскоре перестанут существовать, первой программой, устанавливаемой на ПК после установки операционной системы, должна быть антивирусная система. Интернет в настоящее время – источник повышенной вирусной опасности, особенно при посещении хакерских и развлекательных сайтов!

В состав современных антивирусных систем входят, как правило, несколько программных компонентов:

- **сканер** – программа проверки заданных дисков, папок или файлов по указанию пользователя, по расписанию или по команде монитора, программы проверки электронной почты или скриптов;

- **монитор**, запускающий программу-сканер для проверки всех запускаемых на ПК программ и открываемых файлов данных, начиная с момента запуска ОС;

- программы проверки файлов электронной почты и скриптов в файлах Интернета, поступающих на компьютер;

- программа обновления антивирусных баз;

- центр управления, позволяющий задать расписание сканирования папок, обновления баз и прочие параметры работы.

В антивирусных сканерах для поиска известных вирусов используются два основных метода: сигнатурный поиск и эвристический анализ. Сигнатурой или маской вируса является некоторая постоянная последовательность кода, специфичная для этого конкретного вируса. Все сигнатуры размещены в антивирусной базе – специальном хранилище, в котором программа-антивирус хранит характерные коды вредоносных программ. Метод эвристического



анализа проверяет не код подозрительного файла, а его действия. Для того чтобы размножиться, вирус должен копировать свое тело в память, открывать другие исполняемые файлы и записывать туда свое тело, записывать данные в секторы жесткого диска и т.д. Есть характерные действия и у сетевого червя – доступ к адресной книге и сканирование жесткого диска на предмет обнаружения адресов электронной почты. Благодаря этому эвристический анализатор способен обнаружить даже те вредоносные коды, сигнатуры которых еще неизвестны.

Антивирусные сканеры можно разделить на две категории – универсальные и специализированные. Универсальные сканеры рассчитаны на поиск и обезвреживание всех типов вирусов вне зависимости от операционной системы, на работу в которой рассчитан сканер. Специализированные сканеры предназначены для обезвреживания ограниченного числа вирусов или только одного их класса, например макро-вирусов. Специализированные сканеры, рассчитанные только на макро-вирусы, часто оказываются наиболее удобным и надежным решением для защиты систем документооборота в средах Microsoft Word и Excel.

К достоинствам универсальных сканеров относится большое количество вирусов, которые они знают и могут обнаруживать и обезвреживать. Недостатки – относительно невысокая скорость работы и большой размер антивирусной базы, которую необходимо систематически обновлять (желательно каждый день или чаще).

Существуют также CRC-сканеры, принцип работы которых основан на подсчете контрольных сумм файлов и системных секторов. Эти CRC-суммы затем сохраняются в базе данных этой программы, как, впрочем, и некоторая другая информация: длины файлов, даты их последней модификации и т.д. При последующем запуске CRC-сканеры сверяют данные, содержащиеся в базе данных, с реально подсчитанными значениями. Если информация о файле, записанная в базе данных, не совпадает с реальными значениями, то CRC-сканеры сигнализируют о том, что файл был изменен или заражен вирусом.

Если на компьютере найден вирус, необходимо сделать следующее:

1. Если компьютер подключен к сети, необходимо отключить его от сети и проинформировать системного администратора. Это защитит сервер и другие рабочие станции от заражения. Если же вирус уже заразил сервер, отключение от сети не позволит ему вновь проникнуть на компьютер после лечения. Подключение к сети возможно лишь после того, как будут вылечены все серверы и рабочие станции.

2. При помощи антивирусной программы нужно вылечить зараженные файлы и затем проверить их работоспособность. Если возможно, следует восстановить зараженные файлы из архивной копии, предварительно проверив архив на отсутствие вирусов. После лечения от вируса желательно выполнить сканирование всех дисков компьютера.

3. Если обнаружен резидентный файловый вирус, следует убедиться в том, что программе-сканеру данный вирус известен, и она обезвреживает его резидентную часть. Полезно ознакомиться и с другими характеристиками вируса: типами заражаемых вирусом файлов, действиями вируса и прочее. Источником подробной информации может служить «Вирусная энциклопедия» Лаборатории Касперского (<http://www.viruslist.com/ru/viruses/encyclopedia>).

Сертификацией антивирусных программ занимается **ICSA** – International Computer Security Association – Международная ассоциация по компьютерной безопасности (основана в 1992 году). В тестированиях, проводимых **ICSA Labs**, используется вредоносный код как из собственной коллекции, так и из списка **WildList** (список вирусов неформальной международной организации *WildList Organization International*). По результатам исследований продуктам выдается сертификат **ICSA** – его удостоиваются те антивирусы, которые способны обнаружить 100% вирусов из списка **WildList**, выпущенного за месяц до испытаний, и не менее 90% вирусов из собственной коллекции **ICSA**.

## 14. СОЗДАНИЕ И ПРОСМОТР СПЕЦИАЛЬНЫХ ФОРМАТОВ ДОКУМЕНТОВ

К подобным форматам относятся широко используемые **PDF**-документы, **DjVu**-формат и новый формат **Windows Vista – XPS** (XML Paper Specification). Эти форматы создаются путем преобразования документов, подготовленных в текстовых редакторах, в новый формат, предназначенный, в основном, для чтения этих файлов. Чтение файлов данных форматов (\*.pdf, \*.djvu, \*.xps) может выполняться в специ-альных программах или в интернет-обозревателях (при установленных для этого дополнительных программных компонентах).

**PDF-формат** (Portable Document Format, формат документов корпорации Adobe Systems) – платформонезависимый формат, предназначенный для представления в электронном виде полиграфической продукции. Позволяет сохранять точный внешний вид исходного документа за счет внедрения необходимых шрифтов для текста, векторных и растровых изображений, форм и мультимедиаобъектов. Поддерживает цветовые схемы RGB и CMYK, несколько типов сжатия растровой информации. Документы могут быть зашифрованы с заданием пароля на открытие документа или на его изменение, печать, копирование фрагментов и пр. Однако в Интернете легко можно найти программы, снимающие парольную защиту.

Для просмотра **PDF**-документов можно использовать бесплатную программу **Adobe Reader** или программы других разработчиков (например, **Foxit Reader**). Чтобы создавать, добавлять, пересматривать, редактировать и коллективно использовать информацию, содержащуюся в PDF-файлах следует использовать программы **Adobe Acrobat Professional** и **Adobe Acrobat Standard**.

**DjVu** – формат электронных документов, использующий программную технологию, сочетающую повышенную компрессию и высокое качество изображения. **DjVu** удобен для распространения сканированных документов, фотографий и прочих изображений через Интернет. Существуют

компьютерные программы просмотра файлов формата **DjVu** для различных платформ, в том числе в виде плагинов к браузерам, программных приложений для **PDA** устройств, многие из них являются бесплатными. Степень сжатия изображений в **DjVu** значительно выше, чем в JPEG. Так же как и **PDF**, **DjVu** документ может содержать текстовый слой, что позволяет производить такие привычные операции как поиск по странице, копирование и вставка фрагментов текста и пр. В настоящее время документы многих онлайн-библиотек, архивов технической документации и любительских Интернет-сайтов представлен в **DjVu**-формате.

## **15. РАБОТА С ИНФОРМАЦИЕЙ В ИНТЕРНЕТЕ**

**Интернет** – глобальная компьютерная сеть, объединяющая компьютеры тысяч региональных сетей и миллионов пользователей всего мира.

С функциональной точки зрения, Интернет – новая коммуникационная платформа, соединяющая людей и организации в самых разных сферах жизни и деятельности. Интернет (Internet – Interconnected networks, связанные сети) обеспечивает пользователю ПК доступ к информации, размещенной на Интернет-серверах. Интернет – инструмент, позволяющий людям всего мира свободно получать любую необходимую им информацию и свободно общаться друг с другом. Ежемесячная аудитория интернета в России по состоянию на октябрь 2016 — март 2017 года достигла 87 млн, человек, что составляет 71 % от всего населения страны.

| № ↕ | Страна ↕   | Пользователи <sup>[2]</sup> ↕ | % от населения <sup>[3]</sup> ↕ |
|-----|--|-------------------------------|---------------------------------|
| -   | Земля  | 3 467 560 991                 | 48                              |
| 1   |  КНР              | 692,152,618                   | 50.30                           |
| 2   |  Индия            | 350,480,000                   | 27.51                           |
| 3   |  США              | 239,882,242                   | 74.55                           |
| 4   |  Бразилия         | 122,796,320                   | 59.08                           |
| 5   |  Япония           | 118,131,030                   | 93.33                           |
| 6   |  Россия           | 105,311,724                   | 75.00                           |
| 7   |  Нигерия          | 86,436,611                    | 47.44                           |
| 8   |  Мексика          | 72,945,992                    | 57.43                           |
| 9   |  Германия         | 70,675,097                    | 87.59                           |
| 10  |  Великобритания   | 59,538,545                    | 92.00                           |
| 11  |  Индонезия        | 56,612,527                    | 21.98                           |
| 12  |  Франция          | 54,536,418                    | 84.69                           |
| 13  |  Вьетнам          | 49,265,575                    | 52.72                           |
| 14  |  Республика Корея | 45,213,802                    | 89.90                           |
| 15  |  Турция          | 42,275,017                    | 53.74                           |

### 15.1. История Интернета

Зарождение Интернета связано с программой Агентства передовых исследовательских проектов в области обороны (DARPA) при Министерстве обороны США. Первая исследовательская программа DARPA, посвященная системе глобальной коммуникации, была начата 4 октября 1962 года. Возглавлявший программу Дж. Ликлайдер (J. C. R. Licklider) опубликовал работу «Galactic Network», в которой предсказывал возможность существования в будущем глобальной компьютерной связи между людьми, имеющими мгновенный доступ к программам и базам данных из любой точки земного шара. Большое значение для разработки принципов организации компьютерных сетей имела теория коммутации пакетов передачи данных, которую Леонард Клейнрок (Leonard Kleinrock) впервые опубликовал в июле 1964 года. При коммутации пакетов передаваемая информация разбивается на части, к каждой части присоединяется заголовок, который содержит полную адресную информацию. При коммутации же каналов во время передачи информации пара компьютеров соединяется «один к одному».

В 1966 году DARPA приступило к разработке проекта компьютерной сети **ARPANET**. Проект имел сугубо военные цели, отвечающие задачам управления военными и гражданскими объектами в период войны. Принципиально важна была децентрализация, позволяющая сети функционировать даже при уничтожении нескольких узлов сети.

В декабре 1969 года была сдана в эксплуатацию экспериментальная сеть, объединявшая четыре узла со скоростью передачи данных 56 Кбит/с. В 1970-х годах сеть в основном использовалась для пересылки электронной почты, тогда же появились первые списки почтовой рассылки, новостные группы и доски объявлений. Впоследствии к сети **ARPANET** были подключены тысячи серверов и отдельных сетей (главным образом университетских и государственных), что привело к созданию так называемой «**ARPA Internet**» – прародительницы современной сети **Интернет**

В 1983 году сеть **ARPANET** перешла с протокола NCP на TCP/IP (протокол управления передачей/Интернет-протокол), как основу архитектуры. **ARPANET** сыграл важную роль в развитии Интернета. Однако, к тому времени стали возникать проблемы при расширении сетей, наиболее очевидными из которых являлись перегрузки на линиях связи. Помехой развития был также набор правил для пользователей сети, обязательный для всех (так называемая Acceptable Usage Policy – AUP). Согласно этому своду правил, запрещалось использование сети **ARPANET** в коммерческих целях.

В 1985 году Национальный научный фонд (National Science Foundation – **NSF**) приступил к созданию новой сети **NSFNET**. **NSF** финансировал создание сети, объединившей пять суперкомпьютерных центров и предложил любым региональным и университетским компьютерным центрам, находившимся в пределах физической досягаемости от этой сети подсоединиться к ней. Это был зародыш сети Интернет в нынешнем ее виде. Целый ряд университетов подсоединились к сети **NSF**, чтобы получить доступ к суперкомпьютерам. Кроме исследовательских задач сеть использовалась для передачи электронной почты, файлов и новостей.

Большое значение для развития Интернета имела разработка международного стандарта локальных сетей Ethernet. Локальные сети произвели революцию, позволив существенно увеличить производительность труда работников управленческого аппарата, разработать и внедрить автоматизированные системы управления предприятиями. Но они были явно недостаточными для крупных корпораций, имеющих отделения в разных городах и даже странах. Поэтому вполне естественным стал интерес их к подключению своих локальных сетей к общенациональной сети **NSFNET**.

Всемирная паутина – World Wide Web (**WWW**, или просто **Web**) – появилась после разработки в 1990 году проекта гипертекстовой разметки документов. Автор разработки – Тим Бернер-Ли – программист Европейского центра ядерных исследований (CERN), расположенного в Женеве. В документе 1990 года рассматривались понятия, имеющие принципиальное значение для гипертекстовых документов: общий протокол обмена документами (протокол **HTTP**); общий документальный протокол для поставщиков информации и потребителей (язык разметки гипертекстов); поддержка поиска по индексу и возможность просмотра этих документов при помощи браузеров текста и графики. Слово **Web** также используется для обозначения многих компонентов **WWW**: Web-браузер, Web-страница, Web-почта, Web-чат, Web-конференция и пр.

В 1993 году **NSF** предложил радикальным образом изменить архитектуру сети:

создать разветвленную сеть точек доступа к сети (**NAP**), посредством которых провайдеры (поставщики Интернет-услуг) могли бы осуществлять обмен трафиком;

реализовать проект арбитража маршрутизации в сети (Routing Arbiter project – **RA**), что облегчило бы согласование правил работы в сети и управление адресами между несколькими провайдерами, подключенными к **NAP**;

утвердить единого провайдера для службы обеспечения высокоскоростной магистральной сети (very high-speed Backbone Network Service – **vBNS**);

обеспечить транзит трафика по существующим сетям в целях поддержки межрегиональных соединений путем подключения к провайдерам, которые подключены к **NAP** или напрямую к **NAP**. Любой избранный для этой цели провайдер должен иметь соединения по крайней мере с тремя **NAP**.

В 1994 году **NSF** объявил о строительстве четырех **NAP** в Сан-Франциско, Нью-Йорке, Чикаго и Вашингтоне.

Согласно терминологии **NSF**, точка доступа к сети **NAP** – это высокоскоростной коммутатор или сеть коммутаторов, к которой подключается определенное число маршрутизаторов для обмена трафиком. Любая **NAP** должна работать со скоростью не ниже 100 Мбит/с и иметь возможность повышения пропускной способности по запросу или в зависимости от нагрузки.

30 апреля 1995 года опорная сеть **NSF** была закрыта, а архитектура **NAP** превратилась в Интернет.

## 15.2. Структура Интернета

Современная сеть Интернет представляет собой объединение операторов Интернет-услуг (провайдеров), у которых в нескольких регионах имеются узлы, называемые точками присутствия (Points Of Presence – **POP**). Все подключения клиентов к Интернет-провайдерам осуществляются через серверы доступа или маршрутизаторы, расположенные на **POP** провайдера. Строгая иерархическая структура в Интернете отсутствует. Решением проблем функционирования Интернета, разработкой новых технологий и стандартов сети, распределением доменных имен выполняет достаточно большой круг общественных и научных организаций. Ведущей среди них является специальная негосударственная некоммерческая организация, расположенная в США, – **Internet Society (ISOC)**. Членами **ISOC** могут быть как частные лица, так и организации, но правом голоса на выборах в управляющие органы **ISOC** обладают только частные лица.



В рамках **ISOC** действуют несколько специальных групп:

1) **Совет по архитектуре Интернета – IAB** (Internet Architecture Board) – занимается перспективами развития семейства протоколов TCP/IP: рассматривает различные идеи усовершенствования, принципы построения новых систем и пр.;

2) **Рабочая инженерная группа по Интернету – IETF** (Internet Engineering Task Force) – занимается оперативным разрешением текущих проблем эксплуатации и развития, а также разработкой новых стандартов и протоколов;

3) **Центр сетевой информации – InterNIC** занимается регулированием IP-адресов, доменных имен, справочных служб и хранением документов по стандартам Интернета.

В 1998 г. функции **InterNIC** были переданы специально созданной организации **ICANN** (Internet Corporation for Assigned Names and Numbers), которая занимается вопросами стратегического регулирования назначения IP-адресов и доменных имен. Международной организацией оперативного управления распределением IP-адресов и доменных имен является **IANA** – Internet Assigned Numbers Authority (Уполномоченная организация по распределению нумерации в сети Интернет).

**IANA** определила Региональных регистраторов сети Интернет (Regional Internet Registry – **RIR**), обеспечивающих распределение пространства IP-адресов и назначения номеров автономным системам в сети Интернет по всему земному шару:

Американский реестр адресов сети Интернет (American Registry for the Internet Numbers – **ARIN**);

Европейский сетевой координационный центр (Reseaux IP Europeens Network Coordination Center – **RIPE NCC**);

Азиатско-Тихоокеанский сетевой информационный центр (**APNIC**);

Региональный центр Латинской Америки и Карибских островов (Regional Latin-American and Caribbean IP Address Registry – **LACNIC**).

Общее администрирование и техническое сопровождение российского национального домена (**.RU**) первоначально (в 1993 году) было поручено нейтральной организации – Российскому научно-исследовательскому Институту развития общественных сетей (**РосНИИРОС**).

7 апреля 1994 года **IANA** занесла в международную базу данных национальных доменов верхнего уровня российский национальный домен **RU**.

В 2001 году был учрежден российский Координационный центр национального домена-на сети Интернет. В 2002 году Координационному центру переданы полномочия по выработке правил регистрации доменных имен в домене **RU**, аккредитации регистраторов и разработке перспективных проектов, связанных с развитием российского национального домена. В целях сохранения стабильности функционирования домена **RU** на **РосНИИРОС** возложены функции Технического центра домена **RU**.

.рф (Российская Федерация) — национальный домен верхнего уровня для России. Первый в Интернете домен на кириллице. Отличием от введённого ранее домена «.ru» является то, что в домене «.рф» все имена второго уровня пишутся исключительно кириллицей.

По словам главы делегации Министерства связи и массовых коммуникаций России Владимира Васильева, предварительное решение о введении нового домена было принято на 32-й Международной конференции ICANN в Париже. За введение домена также высказался президент России Дмитрий Медведев. Некоторые скептически отнеслись к созданию кириллической зоны.

На прошедшей в Каире 2—7 ноября 2008 года 33-й конференции ICANN было принято Решение о выделении России кириллического домена верхнего уровня «.рф».

Презентация домена произошла 30 октября 2009 в Сеуле.

16 ноября 2009 года Координационный центр домена **RU** подал заявку на делегирование домена «.рф» в ICANN.

1 декабря 2014 года начался третий этап регистрации в доменной зоне первого уровня .tatar. Право зарегистрировать свои доменные имена получили некоммерческие организации, владельцы торговых марок (знаков) и СМИ.

На предыдущих двух регистрировались: владельцы товарных знаков, внесенных в Депозитарий ICANN (первый этап); органы государственной власти РТ, органы управления государственными внебюджетными фондами, органы местного самоуправления и государственных учреждений РТ (второй этап).

Операторы Интернет-доступа по функционально-географическому признаку разделяются на следующие типы:

магистральные операторы (Internet Backbone Provider – **IBP**) – операторы межрегиональных IP-сетей, обеспечивающие перенос Интернет-трафика между регионами страны и за рубеж;

конечные провайдеры (Интернет-сервис-провайдеры – **ISP**), обеспечивающие предоставление Интернет-услуг конечным пользователям:

- региональные **ISP** – действующие на региональном уровне;
- локальные **ISP** – действующие на локальном уровне.

Обычному пользователю для получения доступа в сеть необходимо воспользоваться услугами локального Интернет-провайдера. Кроме операторов Интернет-доступа существуют также операторы контент-ресурсов – компании, деятельность которых направлена на информационное наполнение сети Интернет и предоставление конечным пользователям услуг информационного характера.

Различают следующие типы операторов контент-ресурсов:

операторы дата-центров, предоставляющие возможность своим пользователям разместить их информацию на серверах и сделать ее доступной в сети Интернет;

провайдеры приложений (Application Service Provider – **ASP**) – компании, предоставляющие пользователям программные приложения, использующие Интернет-технологии и доступные из сети Интернет;

контент-провайдеры (Content Service Provider – **CSP**) – компании, предоставляющие пользователям Интернета ту или иную информацию или службу (контент).



Сетевой протокол – это установленный язык, на котором происходит общение программ. Пересылка данных представляет собой перемещение какого-то потока битов по кабелю. Для того чтобы он доходил до целевого компьютера и представлялся в нем в виде данных, требуется определенный набор правил. Именно они и прописаны в стандартных протоколах. Про них обычно говорят, что они имеют уровень вложенности. Как это понимать? Есть физический уровень, который представляет собой перечень определений, к примеру, каким может быть сетевой кабель, толщина его жил и прочие параметры. Допустим, речь идет об исправном кабеле. Тогда пакеты данных будут отправляться по нему. Но какой из компьютеров будет их принимать? Тут в работу включается канальный уровень, при этом в заголовке пакетов указывается физический адрес каждой машины – определенное число, вшитое в сетевую карту. Его называют MAC-адресом. (<http://fb.ru/article/148419/setevoy-protokol---eto-cto-osnovnyie-setevyie-protokolyi>)

Все компьютеры, участвующие в передаче данных в сети (в настоящее время и в локальной при использовании в качестве сетевой операционной системы Windows), используют протокол **TCP/IP**.

На самом деле протокол **ТСР/ИР** – это два разных протокола, определяющих различные аспекты передачи данных в сети:

– **ТСР** (Transmission Control Protocol) – протокол управления передачей данных, использующий автоматическую повторную передачу пакетов, содержащих ошибки; этот протокол отвечает за разбиение передаваемой информации на пакеты и правильное восстановление информации из пакетов получателя;

– **ИР** (Internet Protocol) – протокол межсетевого взаимодействия, отвечающий за адресацию и позволяющий пакету на пути к конечному пункту назначения проходить по нескольким сетям.

Схема передачи информации по протоколу **ТСР/ИР** такова: протокол **ТСР** разбивает информацию на пакеты и нумерует все пакеты; далее с помощью протокола **ИР** все пакеты передаются получателю, где с помощью протокола **ТСР** проверяется, все ли пакеты получены; после получения всех пакетов протокол **ТСР** располагает их в нужном порядке и собирает в единое целое.

1) на прикладном уровне используются протоколы:

- *DNS*            • *HTTPS*        • *POP3*        • *Telnet*
- *FTP*            • *IMAP*        • *SMTP*        • *XMPP*
- *HTTP*         • *LDAP*        • *SSH*         • *SNMP*

2) на сеансовом уровне/уровне представления используются:

- *SSL*            • *TLS*

3) на транспортном уровне:

- *TCP*            • *UDP*

4) на сетевом уровне:

- *BGP*            • *IGMP*            • *OSPF*            • *EIGRP*
- *ICMP*         • *IP*                • *RIP*              • *IS-IS*

5) на канальном уровне:

- *Ethernet*     • *SLIP*            • *Frame relay*   • *PPP*
- *HDLC*

### 15.3. Адресация в Интернете

Каждый компьютер, подключенный к Интернету имеет два равноценных уникальных адреса: цифровой **ИР**-адрес и символический доменный

адрес **DNS** (Domain Name System, доменная система имен). В настоящее время существует и используется две версии цифровой адресации: **Ipv4** и **Ipv6**

**IPv6** (англ. *Internet Protocol version 6*) — новая версия протокола IP, призванная решить проблемы, с которыми столкнулась предыдущая версия (IPv4) при её использовании в Интернете, за счёт использования длины адреса 128 бит вместо 32. Протокол был разработан IETF.

**IP**-адрес представляет собой уникальную четырехоктетную (32-битовую) величину, выраженную в десятичных числах, разделенных точками, в форме W.X.Y.Z, где точки используются для разделения октетов (например, 10.0.0.1). Поле адреса размером 32 бита состоит из двух частей: адрес сети или связи – сетевая часть адреса, и адрес компьютера в сети (хоста) – идентифицирующий хост в сетевом сегменте.

Для **Ipv4**, кроме обычного распределения адресов по классам, используются также методы использования масок переменной длины (**VLSM** – Variable-Length Subnet Mask) и бесклассовая междоменная маршрутизация (**CIDR** – Classless Interdomain Routing).

Вся информация по получению и распределению адресов находится на Web-сервере ARIN ([www.arin.net](http://www.arin.net)).

### Распределение адресного пространства по регионам

| Адресное пространство       | Кому выдано                 | Когда выдано |
|-----------------------------|-----------------------------|--------------|
| 61.0.0.0 – 61.255.255.255   | Тихоокеанский регион        | апрель 1997  |
| 62.0.0.0 – 62.255.255.255   | Европа                      | апрель 1997  |
| 193.0.0.0 – 195.255.255.255 | Европа                      | май 1993     |
| 199.0.0.0 – 199.255.255.255 | Северная Америка            | май 1993     |
| 200.0.0.0 – 200.255.255.255 | Центральная и Южная Америка | май 1993     |
| 201.0.0.0 – 201.255.255.255 | Центральная и Южная Африка  | май 1993     |
| 202.0.0.0 – 203.255.255.255 | Тихоокеанский регион        | май 1993     |
| 204.0.0.0 – 205.255.255.255 | Северная Америка            | май 1993     |
| 206.0.0.0 – 206.255.255.255 | Северная Америка            | март 1994    |
| 207.0.0.0 – 207.255.255.255 | Северная Америка            | апрель 1995  |
| 208.0.0.0 – 208.255.255.255 | Северная Америка            | ноябрь 1995  |
| 209.0.0.0 – 209.255.255.255 | Северная Америка            | апрель 1996  |
| 210.0.0.0 – 210.255.255.255 | Тихоокеанский регион        | июнь 1996    |
| 211.0.0.0 – 211.255.255.255 | Тихоокеанский регион        | июнь 1996    |
| 212.0.0.0 – 212.255.255.255 | Европа                      | октябрь 1997 |
| 213.0.0.0 – 213.255.255.255 | Европа                      | март 1999    |
| 216.0.0.0 – 217.255.255.255 | Северная Америка            | апрель 1998  |

Протокол **IPv6** выделяет на адрес 128 бит вместо 32 для IPv4. Новая версия IP-протокола **IPv6** даст Интернету 1 миллиард адресов в квадрате, которых должно хватить на много лет.

Формат и семантика адресов **IPv6** описаны в документе RFC-1884. Версия ICMP **IPv6** рассмотрена в RFC-1885.

Существует три стандартные формы для представления **IPv6** адресов в виде текстовых строк:

Основная форма имеет вид x:x:x:x:x:x:x, где 'x' шестнадцатеричные 16-битовые числа. Например:

fedc:ba98:7654:3210:FEDC:BA98:7654:3210      1080:0:0:0:8:800:200C:417A

Начальные нули в каждом из конкретных полей не пишутся, но в каждом поле должна быть, по крайней мере, одна цифра

**Доменный адрес DNS**, в отличие от цифрового, является текстовым и легче запоминается человеком. Пример доменного адреса: maestro.it3207.agtu.ru. Здесь maestro – имя реального компьютера в домене it3207 (третьего уровня), принадлежащем домену более высокого второго уровня agtu, ru – обозначение группы русскоязычных доменов первого уровня. Доменов первого (высшего) уровня немного – около 250. Большая часть из них – так называемые географические домены. Например, .ru (Russia, Россия), .jp (Japan, Япония), .us (United States, США), .uk (United Kingdom, Великобритания), .ca (Canada, Канада), .au (Australia, Австралия) и т. д.. Негеографические домены верхнего уровня – .com (для коммерческих компаний), .net (для сетевых ресурсов), .edu (образовательные учреждения), .mil (военные организации), .org (некоммерческие организации), .gov (правительственные ведомства), .int (интернациональные корпорации) и др. В процессе передачи данных доменный адрес преобразуются в **IP**-адрес. В сети существуют специальные DNS-серверы, отвечающие за адресацию некоторой части сети и сохраняющие информацию о соответствии **IP**-адресов и **DNS**-имен.

Для приобретения в собственность имени домена второго уровня (например [www.agtu.ru](http://www.agtu.ru)), необходимо обратиться к одному из регистраторов доменных имен. Аккредитацией регистраторов в домене.ru занимается Координационный центр национального домена сети Интернет. За зарегистрированные имена необходимо ежегодную плату (например, в Центре регистрации доменов RU-Center около 600 руб. в домене .ru, 450 руб. в доменах NET, COM, ORG, BIZ, INFO).

Для поиска нужной информации в сети используется адрес ресурса (**URL**-адрес, англ. Uniform Resource Locator), содержащий имя протокола по которому нужно обращаться к требуемой информации, адрес сервера и имя файла на этом сервере.

В общем виде формат URL выглядит так: **method://host.domain[:port]/path/filename** где: **method** - одно из значений, перечисленных ниже:

- file – файл на локальной системе;
- http – файл на World Wide Web сервере;
- gopher – файл на Gopher сервере;
- wais – файл на WAIS (Wide Area Information Server) сервере;
- news – группа новостей телеконференции Usenet;
- telnet – выход на ресурсы сети Telnet;
- ftp – файл на FTP-сервере.

**host.domain** – доменное имя в сети Интернет. **port** – номер UDP-порта (User Datagram Protocol – протокол пользовательских датаграмм).

Пример: ***http://www.agtu.ru/archive/index.html***.

Префикс **http://** указывает, что далее следует адрес Web-страницы, **www.agtu.ru** – зарегистрированное доменное имя и, одновременно, имя Интернет-сервера, **/archive** описывает каталог с именем **archiv** на сервере, а **index.html** – имя файла. Часто имена файлов на Интернет-серверах выглядят непривычно. Это связано с тем, что большинство серверов работают под управлением операционной системы UNIX, в которой имена файлов чаще всего



не имеют расширения файла, идентифицирующего его тип. Основной кодировкой кириллицы в текстовых файлах на UNIX-серверах является KOI8-R. В Интернете в настоящее время основным протоколом работы с информацией является гипертекстовый протокол **http** (hyper text transfer protocol) – основной режим работы современных Интернет-обозревателей (браузеров). В состав операционной системы Windows входит **Internet Explorer**, но пользователи пользуются другими, более популярными браузерами.

Основными протоколами работы с электронной почтой (e-mail) являются **SMTP** и **POP3**. Отправка сообщений электронной почты осуществляется по протоколу SMTP (Simple Mail Transfer Protocol, 25-й UDP-порт), а прием по протоколу POP3 (Post Office Protocol, 110-й UDP-порт). Для работы с этими протоколами служат программы Outlook Express в ОС Windows, Microsoft Office Outlook, и пр.

При использовании электронной почты, работающей по протоколам POP3/SMTP, каждому абоненту присваивается уникальный почтовый адрес, формат которого имеет вид: <имя пользователя> @ < имя почтового сервера>, например a.ivanov@agtu.ru.

Организация информации в Интернете в таком виде, когда файлы связаны между собой информационными связями в виде гипертекстовых ссылок, называется всемирной паутиной и обозначается **www**. Под «паутиной» подразумеваются не линии связи, по которым идет сигнал, а связи информационные.

Набор файлов на одном Интернет-сервере, связанных между собой гипертекстовыми ссылками, называется **сайтом**. Например, сайт в Интернете может содержать информацию о фирме. Он имеет свой «входной» файл – домашнюю страницу (home page), которая содержит в виде меню и названий разделов гипертекстовые ссылки на остальные части сайта. Обычно структура ссылок имеют иерархическую или более сложную структуру.

## 15.4. Поиск информации в Интернете

**Поисковая система** – комплекс программных и аппаратных средств для автоматического просмотра ресурсов Интернет, индексации их содержания и предоставления услуг по поиску информации Интернет-пользователям. Поисковые системы могут отличаться по эффективности поиска, по языку поиска (русский, английский и др.) и по некоторым другим возможностям. Например, одни поисковые системы находят информацию только в виде Web-страниц, другие могут просматривать и группы новостей, и файловые серверы.

**Другие популярные сервисы:**

**Электронная почта в Интернете**

**RRS-каналы (вил представления списка новостей)** отображаются в браузере, почтовом клиенте или агрегаторе, как список заголовков. Заголовки обновляются автоматически. Выбрав любой заголовок, можно прочитать информацию по этой теме.

**Общение в Интернете в реальном времени** - службы мгновенных сообщений или Интернет-пейджеры и чат (chat) двух видов – IRC (Internet Relay Chat) и Web-чат. Самой известной **IM** ((Instant Messaging) – обмена мгновенными сообщениями )-программой (Интернет-пейджером) является программа **ICQ** (аббревиатура ICQ созвучна английской фразе «I seek you» – я ищу тебя, произносится ай-сик-ю или, в РФ, «аська»). Доля **ICQ** среди аналогичных программ в РФ составляет более 70%, **Microsoft Manager** – менее 10%, **QIP** – 7%, **Skype** – 4%, прочие (**Jabber, Google Talk, Miranda** и др.) – 8%. Число зарегистрированных пользователей **ICQ** более 50 млн. человек.

**Блоги** – личные электронные дневники на Web-страницах.

В Интернете имеются сайты, главное назначение которых – размещение блогов всех желающих, например, *www.livejournal.com*. Пользователям Web-почты также часто предоставляется возможность вести свои дневники на страницах почтового сайта, например, очень большое количество блогов можно найти на @mail.ru. Особенность этого вида общения – любой желающий может

читать блоги и писать свои комментарии к записям автора блога, каждая такая запись имеет дату и время создания.

**IP-телефония (Интернет-телефония)** – технология, которая использует Интернет для передачи речевых сигналов. При разговоре по ip-телефону голосовые сигналы (слова, которые мы произносим) преобразуются в сжатые пакеты данных. Эти пакеты данных посылаются через Интернет другой стороне. Когда пакеты данных достигают адресата, они декодируются в исходные голосовые сигналы.

**Интернет вещание** – радиоканалы и телеканалы, которые можно слушать и смотреть на Web-страницах сайтов, посвященных этому или через мультимедийные приложения – Windows Media Player, WinAmp, Real-Player и пр.

**Электронная коммерция** – форма взаимодействия продавцов и покупателей, при которой выбор и заказ товаров осуществляется через компьютерные сети, а расчеты между покупателем и поставщиком осуществляются с использованием электронных документов или средств платежа. При классификации электронного бизнеса выделяют несколько категорий:

**B2B** (business-to-business) – бизнес для бизнеса – электронная коммерция между двумя компаниям;

**B2C** (business-to-consumer) – бизнес, ориентированный на конечного физического потребителя;

**C2C** (consumer-to-consumer) – бизнес, обеспечивающий взаимодействие между большим количеством физических потребителей;

**C2B** (consumer-to-business) – системы обработки ценовых заявок, по которым потребители хотели бы приобрести товары и услуги;

**B2A** (business-to-administration) – бизнес, оперирующий сделками между частными компаниями и правительственными организациями (администрацией);

**C2A** (consumer-to-administration) – бизнес, построенный на организации взаимоотношений физических лиц и государственных служб.

При осуществлении электронной коммерции используются различные платежные системы. В зависимости от вида денежных средств системы расчетов в сети можно разделить на две большие группы:

системы расчетов, работающие с реальными деньгами (расчеты наличными, банковскими и почтовыми переводами, платежные системы, работающие с банковскими счетами клиентов);

системы расчетов, использующие электронную валюту (цифровые деньги), эмитируемую платежными системами. К этому типу относятся специализированные платежные системы в Интернете.

Цифровые деньги – это электронный эквивалент реальных денег. Они эмитируются платежной системой и зачисляются на электронные счета клиентов в обмен на деньги, которые клиент вводит в систему банковским переводом или путем внесения наличных.

### **15.5. Обеспечение конфиденциальности информации в Интернете**

Конфиденциальность передаваемой информации обеспечивается ее шифрованием. При помощи процедуры шифрования отправитель сообщения преобразует его из простого текста в набор символов, не поддающийся прочтению без применения специального ключа, известного получателю. Получатель сообщения, используя ключ, преобразует переданный ему набор символов обратно в текст. Процесс преобразования с помощью ключа простого текста в за-шифрованное сообщение и обратно называется алгоритмом шифрования. Обычно алгоритмы шифрования общеизвестны и не являются секретом. Конфиденциальность передачи и хранения зашифрованной информации обеспечивается за счет конфиденциальности ключа.

Ключ к шифру – конкретный набор символов и процедур, применяемых при шифровании и дешифровании сообщений. Обычно степень защищенности информации зависит не только от алгоритма шифрования, но и от длины ключа, измеряемой в битах. Чем длиннее ключ, тем лучше защита, но тем

больше вычислительных процедур необходимо провести компьютеру для шифрования и дешифрования передаваемой информации, что замедляет передачу данных. Существует два вида алгоритмов шифрования:

1) **Симметричные**. В алгоритмах этого вида и для шифрования, и для дешифрования информации применяется один и тот же секретный ключ, известный и отправителю, и получателю информации.

2) **Асимметричные**. Алгоритмы этого вида используют два ключа: один – для шифрования, другой – для дешифрования сообщения. Один из таких ключей является закрытым (секретным), другой – открытым (общедоступным).

В настоящее время в системах документооборота широко используется электронная цифровая подпись (**ЭЦП**), которая является электронным эквивалентом собственноручной подписи. **ЭЦП** служит не только для аутентификации отправителя сообщения, но и для проверки его целостности.

При использовании **ЭЦП** для аутентификации отправителя сообщения применяются открытый и закрытый ключи. Процедура похожа на осуществляемую в асимметричном шифровании, но в данном случае закрытый ключ служит для шифрования, а открытый – для дешифрования.

Алгоритм применения **ЭЦП** состоит из ряда операций:

- 1) генерируется пара ключей: открытый и закрытый;
- 2) открытый ключ передается заинтересованной стороне (получателю документов, подписанных стороной, сгенерировавшей ключи);
- 3) отправитель сообщения шифрует его своим закрытым ключом и передает получателю по каналам связи;
- 4) получатель дешифрует сообщение открытым ключом отправителя.

Создать зашифрованное сообщение, при расшифровке которого открытым ключом получается исходный текст, может только обладатель закрытого ключа, т. е. отправитель сообщения. Использовать для этого открытый ключ невозможно.

Для защиты данных, передаваемых через Интернет, используется протокол **SSL** (Secure Socket Layer). Этот протокол основан на комбинации

алгоритмов асимметричного и симметричного шифрования. Протокол может работать в трех режимах:

- 1) при взаимной аутентификации сторон;
- 2) при аутентификации сервера и анонимности клиента;
- 3) при взаимной анонимности сторон.

При установлении соединения по протоколу SSL для данной сессии связи генерируется разовый ключ, который служит для симметричного шифрования данных, передаваемых в течение данной сессии. Разовый ключ генерируется на этапе установления соединения. При этом используются асимметричные алгоритмы шифрования.

## **16. ОСНОВЫ HTML. РАЗРАБОТКА WEB-САЙТА**

### **16.1. Web-сайты и Web-страницы**

Web-сайт состоит из Web-страниц, объединенных гиперссылками. Web-страницы могут быть интерактивными и могут содержать мультимедийные и динамические объекты.

Создание Web-сайтов реализуется с использованием языка разметки гипертекстовых документов HTML.

Технология HTML состоит в том, что в обычный текстовый документ вставляются управляющие символы (тэги) и в результате мы получаем Web-страницу.

Вид Web-страницы задается тэгами, которые заключаются в угловые скобки.

Тэги могут быть одиночными или парными, для которых обязательно наличие открывающего и закрывающего тегов (такая пара тегов называется *контейнером*).

Закрывающий тэг содержит прямой *слэш* (/) перед обозначением.

HTML-код страницы помещается внутрь контейнера <HTML></HTML>

Заголовок Web-страницы заключается в контейнер <HEAD></HEAD>

Название Web-страницы содержится в контейнере < TITLE ></TITLE>

<HTML>

<HEAD>

<TITLE>Компьютер</TITLE>

</HEAD>

</HTML>

Основное содержание страницы помещается в контейнер

<BODY></BODY>

<HTML>

<HEAD>

<TITLE>Компьютер</TITLE>

</HEAD>

<BODY>

Все о компьютере

</BODY>

</HTML>

Размер шрифта для имеющихся в тексте заголовков задается тэгами от <H1> (самый крупный) до <H6> (самый мелкий)

<H1>Все о компьютере</H1>

Задать тип выравнивания заголовка для тэга заголовка позволяет атрибут ALIGN

Выравнивание по правой границе окна реализуется с помощью записи ALIGN="right", а по центру — ALIGN="center"

<H1 ALIGN="center">Все о компьютере</H1>

С помощью тэга FONT и его атрибутов можно задать параметры форматирования шрифта любого фрагмента текста. Атрибут FACE позволяет задать гарнитуру шрифта (например, FACE="Arial"), атрибут SIZE — размер шрифта (например, SIZE=4), атрибут COLOR — цвет шрифта (например, COLOR="blue")

<FONT COLOR="blue">

<H1 ALIGN="center">Все о компьютере</H1>

</FONT>

Заголовок целесообразно отделить от остального содержания страницы горизонтальной линией с помощью одиночного тэга <HR>

Разделение текста на абзацы производится с помощью контейнера <P></P>

Например, поместим на титульную страницу текст, разбитый на абзацы с различным выравниванием:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Компьютер</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<FONT COLOR="blue">
```

```
<H1 ALIGN="center">Все о компьютере</H1>
```

```
</FONT>
```

```
<HR>
```

```
<P ALIGN="left">На этом сайте вы сможете получить  
различную информацию о компьютере, его программном обеспечении  
и ценах на компьютерные комплектующие.</P>
```

```
<P ALIGN="right">Терминологический словарь познакомит вас с  
компьютерными терминами, а также вы сможете заполнить  
анкету.</P>
```

```
<BODY>
```

```
</HTML>
```

## 16.2. Вставка изображений

Для вставки изображения используется тэг <IMG> с атрибутом SRC

```
<IMG SRC="COMP6.gif">
```

Расположить рисунок относительно текста различными способами позволяет атрибут ALIGN тэга <IMG>, который может принимать пять различных значений:

TOP (верх),



MIDDLE (середина),

BOTTOM (низ),

LEFT (слева)

RIGHT (справа).

```
<IMG SRC="COMP6.gif" ALT="Компьютер" ALIGN="right" >
```

### 16.3. Гиперссылки

Первая титульная страница должна предоставлять посетителю Web-сайта возможность начать путешествие по сайту. Для этого на титульную страницу должны быть помещены гиперссылки на другие страницы сайта.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Заголовок страницы</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
</BODY>
```

```
</HTML>
```

Создать пустые страницы «Программы», «Словарь», «Комплекующие» и «Анкета» и сохранить их в файлах с именами software.htm, glossary.htm, hardware.htm и anketa.htm в каталоге сайта.

### 16.4. Панель навигации

Панель навигации будет представлять собой абзац, выровненный по центру, в котором указатели гиперссылок разделены пробелами (&nbsp;):

```
<P ALIGN="center">
```

```
[Программы] &nbsp; [Словарь] &nbsp;
```

```
[Комплекующие] &nbsp; [Анкета]
```

```
</P>
```

```
<P ALIGN="center">
```

```
[<A HREF="software.htm">Программы</A>] &nbsp;
```

```
[<A HREF="glossary.htm">Словарь</A>] &nbsp;
```

```
[<A HREF="hardware.htm">Комплекующие</A>] &nbsp;
```

[<A HREF="anketa.htm">Анкета</A>]

</P>

## 16.5. Ссылка на электронный адрес

<ADDRESS>

<A HREF="mailto:mailbox@provaider.ru">E-mail:

mailbox@provaider.ru </A>

</ADDRESS>

Списки

Список располагается внутри контейнера <OL></OL>, а каждый элемент списка определяется тэгом <LI>. С помощью атрибута TYPE тэга <OL> можно задать тип нумерации: арабские цифры (по умолчанию), "I" (римские цифры), "a" (строчные буквы) и т.д. Пример:

Программное обеспечение

<OL>

<LI>Системные программы

<LI>Прикладные программы

<LI>Системы программирования

</OL>

Вложенный нумерованный список располагается внутри контейнера <UL></UL>, а каждый элемент списка определяется также тэгом <LI>. С помощью атрибута TYPE тэга <UL> можно задать вид маркера списка: "disc" (диск), "square" (квадрат) или "circle" (окружность). Пример:

Для прикладных программ:

<UL>

<LI TYPE="square"> текстовые редакторы;

<LI TYPE="square"> графические редакторы;

<LI TYPE="square"> электронные таблицы;

<LI TYPE="square"> системы управления базами данных.

</UL>

Страницу «Словарь» мы представим в виде словаря компьютерных терминов.

Для этого используем контейнер списка определений `</DL>`. Внутри него текст оформляется в виде *термина*, который выделяется непарным тэгом `<DT>`, и *определения*, которое следует за тэгом `<DD>`

```
<DL>
```

```
<DT> Процессор
```

```
<DD> Центральное устройство компьютера, производящее обработку информации в двоичном коде.
```

```
<DT> Оперативная память
```

```
<DD> Устройство, в котором хранятся программы и данные.
```

```
</DL>
```

## 16.6. Формы

Форма заключается в контейнер `<FORM></FORM>`

Текстовые поля: создаются с помощью тэга `<INPUT>` со значением атрибута `TYPE="text"`. Атрибут `NAME` является обязательным и служит для идентификации полученной информации. Значением атрибута `SIZE` является число, задающее длину поля ввода в символах.

Для того чтобы анкета «читалась», необходимо разделить строки с помощью тэга перевода строки `<BR>`. Пример:

```
<FORM>
```

```
Пожалуйста, введите ваше имя: <BR>
```

```
<INPUT TYPE="text"
```

```
NAME="name" SIZE=30> <BR>
```

```
E-mail: <BR>
```

```
<INPUT TYPE="text"
```

```
NAME="e-mail" SIZE= 30 > <BR>
```

```
</FORM>
```

## 16.7. Переключатели

Группа переключателей («радиокнопок») создается с помощью тэга `<INPUT>` со значением атрибута `TYPE="radio"`. Все элементы в группе должны иметь одинаковые значения атрибута `NAME`. Например, `NAME="group"`.

Еще одним обязательным атрибутом является `VALUE`, которому присвоим значения "schoolboy", "student" и "teacher". Значение атрибута `VALUE` должно быть уникальным для каждой «радиокнопки», так как при ее выборе именно они передаются серверу. Пример:

Укажите, к какой группе пользователей вы себя относите:

`<BR>`

`<INPUT TYPE="radio" NAME="group" VALUE="schoolboy">`учащийся`<BR>`

`<INPUT TYPE="radio" NAME="group" VALUE="student">`студент`<BR>`

`<INPUT TYPE="radio" NAME="group" VALUE="teacher">`учитель`<BR>`

## 16.8. Флажки

Если из предложенного перечня он нужно выбрать одновременно несколько вариантов, то их помечают флажками.

Флажки создаются в тэге `<INPUT>` значением атрибута `TYPE="checkbox"`.

Флажки, объединенные в группу, могут иметь одинаковые значения атрибута `NAME`. Например, `NAME="group"`.

Еще одним обязательным атрибутом является `VALUE`, которому присвоим, например, значения "www", "e-mail" и "ftp".

Какие из сервисов Интернета вы используете наиболее часто:

`<BR>`

`<INPUT TYPE="checkbox" NAME="group" VALUE="WWW">`  
`WWW<BR>`

```
<INPUT TYPE="checkbox" NAME="group" VALUE="e-mail"> e-mail  
<BR>
```

```
<INPUT TYPE="checkbox" NAME="group" VALUE="ftp"> FTP<BR>
```

### 16.9. Поля списков

Выясним, какой из браузеров предпочитает посетитель сайта. Перечень браузеров представим в виде раскрывающегося списка, из которого можно выбрать только один вариант. Для реализации такого списка используется контейнер `<SELECT></SELECT>`, в котором каждый элемент списка определяется тэгом `<OPTION>`. Выбираемый по умолчанию элемент задается с помощью атрибута `SELECTED`. Пример:

Какой из браузеров вы предпочитаете? <BR>

```
<SELECT NAME="browsers">
```

```
<OPTION SELECTED> Internet Explorer
```

Internet Explorer

```
<OPTION>
```

Netscape Navigator

```
<OPTION> Opera
```

```
<OPTION> Neo Planet
```

```
</SELECT> <BR>
```

### 16.10. Текстовая область

Создается такая область с помощью тэга `<TEXTAREA>` с обязательными атрибутами: `NAME`, задающим имя области, `ROWS`, определяющим число строк, и `COLS` — число столбцов области. Пример:

Какую еще информацию вы хотели бы видеть на нашем сайте? <BR>

```
<TEXTAREA NAME="resume"
```

```
ROWS=4 COLS=30>
```

```
</TEXTAREA <BR>
```

### 16.11. Отправка данных из формы

Отправка введенной в форму информации или очистка полей от уже введенной информации осуществляется с помощью кнопок. Кнопки создаются

с помощью тэга <INPUT>. Для создания кнопки, которая отправляет информацию, атрибуту TYPE необходимо присвоить значение "submit", а атрибуту VALUE, который задает надпись на кнопке, — значение "Отправить".

Для создания кнопки, которая производит очистку формы, атрибуту TYPE необходимо присвоить значение "reset", а атрибуту VALUE - значение "Очистить".

```
<INPUT TYPE="submit" VALUE="Отправить"> <INPUT TYPE="reset"
VALUE="Очистить">
```

## 17. ЗНАКОМСТВО С XML

Язык XML (Extensible Markup Language) был разработан рабочей группой XML Working Group консорциума World Wide Web Consortium (W3C). Вот как описывают его создатели:

"Расширяемый язык разметки Extensible Markup Language (XML) представляет собой составную часть языка SGML... Он предназначен для облегчения использования языка SGML в Web и выполнения задач, которые в настоящее время реализуются с помощью языка HTML. XML разработан с целью усовершенствовать применение и взаимодействие языков SGML и HTML."

Это выдержка из спецификации версии 1.0 XML, созданной XML Working Group в феврале 1998 г. Весь документ вы можете найти на сайте W3C по адресу <http://www.w3.org/TR/1998/REC-xml-19980210>.

XML – язык разметки, разработанный специально для размещения информации в World Wide Web, аналогично языку гипертекстовой разметки HTML (Hypertext Markup Language), который изначально стал стандартным языком создания Web-страниц. Поскольку язык HTML полностью удовлетворяет всем нашим потребностям, возникает вопрос: для чего понадобился совершенно новый язык для Web? В чем состоят его преимущества и достоинства? Как он взаимодействует с HTML? Заменит ли он HTML, или только усовершенствует его? Наконец, что собой представляет язык

SGML, частью которого является XML, и почему нельзя использовать для Web-страниц собственно SGML?

### **17.1. Предназначение XML**

Хотя набор HTML-элементов был существенно расширен по сравнению с первой версией HTML, язык HTML по-прежнему не пригоден для представления многих типов документов. Ниже приведены примеры документов, которые не могут быть адекватно описаны с помощью языка HTML.

Документ, который не содержит типовых компонентов (заголовков, абзацев, списков, таблиц и т.д.) Например, в языке HTML отсутствуют элементы, необходимые для отображения музыкальных символов или математических уравнений.

База данных, такая как каталог книг. Вы можете использовать HTML-страницу, чтобы хранить и отображать информацию из статической базы данных (например, перечень книг и их описание). Однако, если вам понадобится осуществить сортировку, фильтрацию, поиск и обработку информации, придется снабдить каждую из составных частей информации соответствующей меткой (как в программе, работающей с базами данных, такой как Microsoft Access). В языке HTML не предусмотрено соответствующих элементов.

Документ, который вы хотите представить в виде иерархической структуры. Допустим, вы пишете книгу и хотите разбить ее на части, главы, разделы А, В, С и т.д. В дальнейшем программа может использовать данную структуру документа для создания оглавления, оформления различных уровней в структуре с помощью различных стилей, извлечения определенных разделов, а также обработки информации иными способами. Однако элемент типа заголовок в HTML содержит лишь описание собственно текста. Например, содержимое Web-сайта

Поскольку внутри элемента типа заголовок вы не задаете вложенные элементы текста, которые относятся к разделам документа, эти элементы не

могут быть использованы для представления иерархической структуры документа.

Язык XML позволяет преодолеть эти ограничения.

Описание на языке XML представляет собой *операторы*, написанные с соблюдением определенного синтаксиса. Когда вы создаете XML-документ, то вместо использования ограниченного набора определенных элементов вы имеете возможность создавать ваши собственные элементы и присваивать им любые имена по вашему выбору – именно поэтому язык XML является расширяемым (*extensible*). Следовательно, вы можете использовать XML для описания практически любого документа, от музыкальной партитуры до *базы данных*. Например, вы можете описать перечень книг, подобно представленному в следующем XML-документе.

```
<?xml version="1.0"?><INVENTORY>
  <BOOK><TITLE>The Adventures of Huckleberry Finn</TITLE>
  <AUTHOR>Mark Twain</AUTHOR>
  <BINDING>mass market paperback</BINDING>
  <PAGES>298</PAGES>
  <PRICE>$5.49</PRICE>
</BOOK>
<BOOK>
  <TITLE>Moby-Dick</TITLE>
  <AUTHOR>Herman Melville</AUTHOR>
  <BINDING>trade paperback</BINDING>
  <PAGES>605</PAGES>
  <PRICE>$4.95</PRICE>
</BOOK>
<BOOK>
  <TITLE>The Scarlet Letter</TITLE>
  <AUTHOR>Nathaniel Hawthorne</AUTHOR>
  <BINDING>trade paperback</BINDING>
```



```
<PAGES>253</PAGES>  
<PRICE>$4.25</PRICE>  
</BOOK>  
</INVENTORY>
```

Примечание. Для описания базы данных в XML предусмотрена возможность работы с несколькими форматами (например, формат .mdb Access или .dbf dBase): язык XML построен на принципе открытых и доступных стандартов.

Имена элементов в XML-документе (такие как INVENTORY, BOOK и TITLE в приведенном выше примере) не являются определениями языка XML. Вы всего лишь назначаете эти имена при создании определенного документа. Для ваших элементов вы можете выбирать любые корректно заданные имена (LIST вместо INVENTORY, либо ITEM вместо BOOK).

Совет. Когда присваиваете имена в XML-документе, старайтесь делать их по возможности наиболее информативными. Одним из преимуществ XML-документа является то, что каждому фрагменту информации может быть присвоено информативное описание.

В предыдущем примере XML-документ имеет иерархическую структуру в виде дерева с элементами, вложенными в другие элементы, и с одним элементом верхнего уровня (в нашем примере – INVENTORY) – он носит название элемент Документ или корневой элемент, – который содержит все другие элементы. Структуру описанного в примере документа можно представить, как показано на рисунке.

Таким образом, с помощью XML можно описать иерархическую структуру документа, такого как книга, содержащего части, главы и разделы.

## 17.2. Создание XML-документов

Поскольку в XML нет типовых элементов, может показаться, что в нем вообще отсутствуют какие-либо стандарты. Тем не менее, язык XML имеет строго определенный синтаксис. Например, в отличие от HTML, каждый элемент XML должен содержать начальный и конечный *тег* (либо специальный

пустой *тег*). Любой вложенный элемент должен быть полностью определен внутри элемента, в состав которого он входит.

На деле гибкость в создании ваших собственных элементов требует строгого соблюдения синтаксиса. Это обусловлено тем, что структура *XML*-документов должна быть понятной для программы, которая обрабатывает и отображает информацию, содержащуюся в этих документах. Строгий *синтаксис* придает *XML*-документу предсказуемую форму и облегчает написание программы обработки. Основное назначение языка *XML* – облегчить работу с документами в *Web*.

### 17.3. Отображение XML-документов

При отображении *HTML*-страницы *браузер* определяет, что элемент *H1*, например, является заголовком верхнего уровня, и отображает его в соответствующем формате. Это определяется тем, что данный элемент является частью *HTML*-стандарта. Но каким образом *браузер* либо другая *программа* определяет, как обрабатывать и отображать элементы созданного вами *XML*-документа (такого как *BOOK* или *BINDING* в нашем примере), если вы сами составили эти элементы?

Есть три основных способа сообщить браузеру (в частности, *Microsoft Internet Explorer 5*), как обрабатывать и отображать каждый из созданных вами *XML*-элементов.

Таблица стилей. С помощью данного метода вы связываете таблицу стилей с *XML*-документом. Таблица стилей представляет собой отдельный файл, содержащий инструкции для форматирования индивидуальных *XML*-элементов. Вы можете использовать либо каскадную таблицу стилей (*Cascading Style Sheet* – *CSS*), которая также применяется для *HTML*-страниц, либо расширяемую таблицу в формате языка стиливых таблиц (*Extensible Stylesheet Language* – *XSL*), обладающую значительно более широкими возможностями, нежели *CSS*, и разработанную специально для *XML*-документов.

Связывание данных. Этот метод требует создания *HTML*-страницы, связывания с ней *XML*-документа и установления взаимодействий стандартных

HTML-элементов на странице, таких как SPAN или TABLE, с элементами XML. В дальнейшем HTML-элементы автоматически отображают информацию из связанных с ними XML-элементов.

Написание сценария. В этом методе вы создаете HTML-страницу, связываете ее с XML-документом и имеете доступ к индивидуальным XML-элементам с помощью специально написанного кода сценария (JavaScript или Microsoft Visual Basic Scripting Edition [VBScript]). Браузер воспринимает XML-документ как объектную модель документа (Document Object Model – DOM), состоящую из большого набора объектов, свойств и команд. Написанный код позволяет осуществлять доступ, отображение и манипулирование XML-элементами.

#### 17.4. SGML, HTML и XML

Обобщенный *структурированный язык* разметки (Structured Generalized Markup Language – SGML) является родоначальником всех языков разметки. Языки HTML и XML образованы из SGML (хотя и различными способами). SGML определяет базовый *синтаксис*, но дает вам возможность создавать собственные элементы (отсюда термин обобщенный в названии языка). Чтобы использовать SGML для описания определенного документа, вы должны продумать соответствующий набор элементов и структуру документа. Например, чтобы описать книгу, вы должны использовать созданные вами элементы с именами BOOK, PART, CHAPTER, INTRODUCTION, A-SECTION, B-SECTION, C-SECTION и т.д.

Набор наиболее употребительных элементов, используемых для описания документа определенного типа, называется SGML-приложением. (SGML-приложение также включает в себя правила, устанавливающие способы организации элементов, а также другие особенности их взаимодействия.) Вы можете определить ваше собственное SGML-приложение, чтобы описать тип документа, с которым вы работаете, либо в теле основной программы должно быть определено SGML-приложение для описания типовых документов. Наиболее известным примером последнего типа приложений является HTML,

который представляет собой *SGML-приложение*, разработанное в 1991 г. для описания *Web*-страниц.

Казалось бы, язык *SGML* вполне подходит для описания *Web*-документов. Однако разработчики из консорциума *W3C* считали, что он является слишком сложным и фундаментальным, чтобы эффективно представлять информацию в *Web*. Гибкость и большое обилие средств, поддерживаемых *SGML*, затрудняет написание программного обеспечения, необходимого для обработки и отображения *SGML*-информации в *Web*-браузерах. Следовало бы приспособить часть языка *SGML* специально для помещения информации в *Web*. В 1996 г. группа *XML Working Group* разработала ветвь языка *SGML*, назвав его расширяемым языком разметки – *Extensible Markup Language*.

*XML* является упрощенной версией *SGML*, приспособленной для *Web*. Как и *SGML*, *XML* дает возможность разрабатывать собственные наборы элементов при описании определенного документа. Как и в *SGML*, в теле программы может быть определено *XML-приложение* (или словарь), которое содержит набор наиболее употребительных элементов общего назначения и структуру документа, которая может быть использована для описания документа определенного типа (например, документов, содержащих математические формулы или векторную графику). *Синтаксис XML* более простой, чем *SGML*, что облегчает восприятие *XML*-документов, а также написание программ браузеров, кодов и *Web*-страниц для доступа и представления информации документа.

### **17.5. Заменит ли XML HTML?**

На сегодняшний день ответ на этот вопрос отрицательный. *HTML* по-прежнему остается основным языком для сообщения браузеру, как отображать информацию в *Web*.

В браузерах вы можете открывать *XML*-документы с *вложенными таблицами* стилей непосредственно браузером, не используя *HTML*-страницы. Однако в двух других основных методах отображения *XML*-документов –

связывании данных и DOM-сценариях – *отображение XML-документов* осуществляется через *HTML Web-страницы*. (Даже при применении метода таблиц стилей в случае, если вы используете язык *XSL*, вам потребуется воспользоваться *HTML*, чтобы сообщить браузеру, каким образом форматировать *XML-данные*.)

Не заменяя *HTML*, *XML* в настоящее время используется в сочетании с ним, существенно расширяя возможности *Web-страниц* для:

виртуального представления документов любого типа;

сортировки, фильтрации, упорядочения, поиска и манипулирования информацией иными способами;

представления информации в структурированном виде.

Как заявляют сами разработчики, *XML* был создан для взаимодействия с *HTML* и совместного с ним использования.

### **17.6. Стандартные XML-приложения**

Вы можете использовать *XML* не только для описания отдельного документа. Индивидуальный *пользователь*, компания или комитет по стандартам может определить необходимый набор элементов *XML* и структуру документа, которые будут применяться для особого класса документов. Подобный набор элементов и описание структуры документа называют **XML-приложением** или **XML-словарем**.

Например, организация может определить *XML-приложение* для создания документов, описывающих молекулярные структуры, людские ресурсы, *мультимедиа* презентации или содержащих векторную графику.

*XML-приложение* обычно определяется созданием **описателя типа документа (DTD)**, который является допустимым компонентом *XML-документа*. *DTD* построен по схеме *базы данных*: он устанавливает и определяет имена элементов, которые могут быть использованы в документе, порядок, в котором элементы могут появляться, доступные к применению атрибуты элементов и другие особенности документа. Для практического использования *XML-приложения* вы обычно включаете его *DTD* в ваш *XML-*

документ; наличие *DTD* в документе ограничивает круг элементов и структур, которые вы будете использовать, вследствие чего ваш документ отвечает стандартам данного приложения.

Преимущества применения стандартных *XML*-приложений при разработке ваших документов состоят в том, что вы можете совместно использовать документы со всеми другими пользователями приложения, а документ может обрабатываться и отображаться с помощью программного обеспечения, которое уже создано для данного приложения.

### **17.7. XML-приложения, повышающие качество XML-документов**

Кроме *XML*-приложений для описания определенных классов документов имеется несколько *XML*-приложений, которые вы можете применять внутри *XML*-документа любого типа. Эти приложения облегчают создание документа и улучшают его качество. Ниже приведены примеры таких приложений.

**Extensible Stylesheet Language (XSL)** позволяет вам создавать мощные стилевые таблицы с использованием синтаксиса *XML*.

**XML Schema** позволяет разрабатывать подробные схемы для ваших *XML*-документов с использованием стандартного синтаксиса *XML*, что является более мощной альтернативой применения *DTD*.

**XML Linking Language (XLink)** дает возможность связывать ваши *XML*-документы. Он поддерживает множественные целевые ссылки и другие полезные функции, обеспечивая большую свободу по сравнению с механизмом организации ссылок в *HTML*.

**XML Pointer Language (XPointer)** позволяет определять гибкие целевые ссылки. При совместном использовании *XPointer* и *XLink* вы можете организовывать ссылки на любое место в целевом документе – а не только переходы к специально выделенным пунктам.

### **17.8. Реальное использование XML**

**Работа с базами данных.** Подобно традиционным базам данных *XML* может быть использован для присвоения метки каждому полю информации

внутри каждой записи базы данных. (Например, можно пометить каждое имя, адрес и номер телефона внутри записей списка адресов.) После этого вы сможете отображать данные различными способами и организовывать поиск, сортировку, фильтрацию и иную обработку данных.

**Структурирование документов.** Иерархическая структура XML-документов идеально подходит для разметки структуры таких документов, как романы, научные труды, пьесы. Например, вы можете использовать XML для разметки пьесы на акты, сцены, размечать действующих лиц, сюжетные линии, декорации и т. д. XML-разметка дает возможность программам отображать или распечатывать документ в необходимом формате; находить, извлекать или манипулировать информацией в документе; генерировать оглавления, резюме и аннотации; обрабатывать информацию иными способами.

**Работа с векторной графикой** (VML – Vector Markup Language).

**Мультимедиа презентации** (SMIL – Synchronized Multimedia Integration Language, HTML + TIME – HTML Timed Interactive Multimedia Extensions).

**Описание каналов.** Каналы представляют собой Web-страницы, которые автоматически рассылаются подписчикам. (CDF – Channel Definition Format).

**Описание программных пакетов и их взаимосвязей.** Такие описания обеспечивают распространение и обновление программных продуктов в сети. (OSD – Open Software Description).

**Взаимодействие приложений через Web с использованием XML-сообщений.** Эти сообщения являются независимыми от операционных систем, объектных моделей и компьютерных языков. (SOAP – Simple Object Access Protocol).

**Отправка электронных бизнес-карт через e-mail.**

**Обмен финансовой информацией.** Обмен информацией в открытом и понятном формате осуществляется между финансовыми программами (такими как Quicken и Microsoft Money) и финансовыми институтами (банками, общественными фондами). (OFX – Open Financial Exchange).

**Создание, управление и использование сложных цифровых форм для коммерческих Internet-транзакций.** Подобные формы могут включать оцифрованные подписи, которые делают их признанными юридически. (XFDL – Extensible Forms Description Language).

**Обмен запросами по приему на работу и резюме (HRMML – Human Resource Management Markup Language).**

**Форматирование математических формул и научной информации в Web.** (MathML – Mathematical Markup Language).

**Описание молекулярных структур (CML – Chemical Markup Language).**

**Кодирование и отображение информации о ДНК, РНК и цепочках (BSML – Bioinformatic Sequence Markup Language).**

**Кодирование генеалогических данных (GeDML – Genealogical Data Markup Language).**

**Обмен астрономическими данными (AML – Astronomical Markup Language).**

**Создание музыкальных партитур (MusicML – Music Markup Language).**

**Работа с голосовыми сценариями для доставки информации по телефону.** Голосовые сценарии могут быть использованы, например, для генерирования голосовых сообщений, справок о наличии товаров и прогнозов погоды (VoxML).

**Обработка и доставка информации курьерскими службами.** Служба Federal Express, например, уже использует XML для этих целей.

**Представление рекламы в прессе в цифровом формате (Ad Markup).**

**Заполнение юридических документов и электронный обмен юридической информацией (XCL – XML Court Interface).**

**Кодирование прогнозов погоды (OMF – Weather Observation Markup Format).**

**Обмен страховой информацией.**



**Обмен новостями и информацией с использованием открытых Web-стандартов (XMLNews).**

**Представление религиозной информации и разметка текстов богослужений (ThML – Theological Markup Language, LitML – Liturgical Markup Language).**

### **СПИСОК ЛИТЕРАТУРЫ**

1. Волкова В.Н. Теория информационных процессов и систем. - М.: Юрайт, 2016. - 504 с.
2. Информатика / Под ред. Н.В. Макаровой. - СПб.: Питер, 2012. - 160 с.
3. Информатика. Базовый курс / Под ред. С.В. Симоновича. - СПб.: Питер, 2015. - 640 с.
4. Информационные технологии в экономике управления / Под ред. В.В. Трофимова. - М.: Юрайт, 2015. - 482 с.
5. Кудряшов Б.Д. Теория информации. - СПб.: Питер, 2009. - 320 с.
6. Ляхович В.Ф., Молодцов В.А., Рыжикова Н.Б. Основы информатики. - М.: КноРус, 2016. - 348 с.
7. Михеева Е.В. Информационные технологии в профессиональной деятельности. - М.: Академия, 2013. - 384 с.
9. Могилев А.В., Пак Н.И., Хеннер Е.К. Информатика. - М.: Академия, 2012. - 848 с.