УДК 004.382.7
ББК 32.971.3
Д40

Мэттью Джастис

Д40 Как на самом деле работают компьютеры. Практическое руководство по внутреннему устройству машины / науч. ред. Ю. В. Ревич; пер. с англ. С. Л. Плехановой. – М.: ДМК Пресс, 2022. – 428 с.: ил.

ISBN 978-5-97060-973-6

В этом руководстве в доступной форме излагаются основы вычислительной техники. Рассматриваются принципы электронных вычислений и использование двоичных чисел; в общих чертах показано, как функционирует аппаратное обеспечение компьютера, для чего нужна операционная система и как передаются данные по интернету. Читатель получит базовое представление о языках программирования, изучая примеры кода на C и Python.

Каждая глава содержит упражнения и практические задания (проекты), позволяющие на практике применить полученные знания.

Книга будет полезна всем, кто хочет разобраться, как работает компьютер.

УДК 004.382.7
ББК 32.971.3

Title of English-language original: How Computers Really Work: A Hands-On Guide to the Inner Workings of the Machine, ISBN 9781718500662, published by No Starch Press Inc. 245 8th Street, San Francisco, California United States 94103.

ISBN 978-1-71850-066-2 (англ.)
ISBN 978-5-97060-973-6 (рус.)

Об авторе

Мэттью Джастис – инженер-программист. Он проработал 17 лет в компании Microsoft, где выполнял различные работы, включая отладку ядра Windows, автоматизацию исправления ошибок и руководство группой специалистов, ответственных за создание диагностических инструментов и сервисов. Он работал над низкоуровневым программным обеспечением (операционная система) и над программным обеспечением высокого уровня, удаленным от основного оборудования (например, веб-приложения). Мэттью имеет научную степень в области электротехники. Когда он не занят написанием кода или сбором схем, Мэттью любит проводить время с семьей, ходить в горы, читать, создавать музыкальные аранжировки и играть в старые видеоигры.

О технических рецензентах

Доктор Билл Янг – доцент кафедры вычислительной техники Техасского университета в Остине. До поступления в Техасский университет в 2001 году он имел 20-летний опыт работы в данной отрасли. Он специализируется на формальных методах программирования и на компьютерной безопасности, но среди прочих курсов также часто преподает и компьютерную архитектуру.

Брайан Вильгельм – инженер-программист. Он имеет научные степени по математике и информатике и работает в Microsoft уже 20 лет, выполнив за это время самые разные работы – от отладки ядра Windows до разработки бизнес-приложений. Он любит читать, смотреть научно-фантастические фильмы и слушать классическую музыку.

Джон Хьюз начал собирать электрические цепи уже в раннем возрасте, а будучи подростком, перешел к проектам по электронике. Позже он получил научную степень по физике и продолжил развивать свой интерес к электронике, помогая школьникам с их проектами, пока работал ассистентом по науке. Джон преподавал электронику и физику вплоть до профессионального уровня в Великобритании и руководил школьным клубом электроники для детей в возрасте от 11 до 18 лет, создав сайт http://www.electronicsclub.info/ для поддержки клуба. Он считает, что каждый может получить удовольствие от создания проектов по электронике независимо от возраста и способностей.

КРАТКОЕ СОДЕРЖАНИЕ

СОДЕРЖАНИЕ

9

Программирование высокого уровня 199

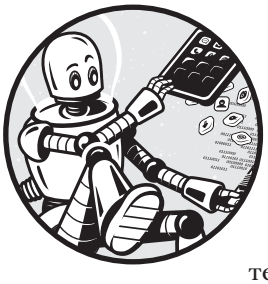
10

Операционные системы 242

Приложение B

Технические средства 406

ВВЕДЕНИЕ



Вам интересно, как работают компьютеры? Обретение глубокого понимания вычислительной техники часто достигается долгим и извилистым путем. Проблема не в отсутствии информации. Быстрый поиск в интернете покажет, что существует очень много книг и веб-сайтов, посвященных объяснению работы вычислительной техники. Программирование, информатика, электроника, операционные системы... так много информации. Это хорошо, но может напугать. С чего начать? Как одна тема связана с другой? Эта книга была написана, чтобы дать вам отправную точку для изучения ключевых концепций, касающихся вычислительной техники, и понимания того, как эти концепции связаны друг с другом.

Работая руководителем инженерного отдела, я регулярно проводил собеседования с людьми, претендующими на должности разработчиков программного обеспечения. Я разговаривал со многими кандидатами, которые умели писать код, но довольно многие из них, похоже, не знали, как на самом деле работают компьютеры. Они знали, как заставить компьютер выполнять их команды, но не понимали, что происходит за экраном. Размышления об этих интервью и воспоминания о моих собственных трудностях в попытках разобраться в компьютерах привели меня к написанию этой книги.

Моя цель – изложить основы вычислительной техники в доступной, практической форме, которая делает абстрактные понятия более понятными. В этой книге нет глубокого погружения в каждую из поднятых тем, но вместо этого в ней представлены основополагающие концепции вычислительной техники, а также связь между этими концепциями. Я хочу, чтобы вы смогли составить представление о том, как работают вычислительные машины, что позволит затем углубиться в интересующие вас темы.

Компьютеры уже повсюду, и поскольку наше общество все больше и больше зависит от технологий, нам нужны люди с широким пониманием вычислительной техники. Я надеюсь, что эта книга поможет вам обрести такое широкое понимание.

Для кого эта книга?

Эта книга предназначена для всех, кто хочет разобраться, как работают компьютеры. Вам не нужно обладать какими-либо предварительными знаниями по рассматриваемым темам, так как мы начнем с основ. С другой стороны, если у вас уже есть знания в области программирования или электроники, эта книга поможет расширить понимание в других областях. Книга написана для тех, кто склонен к самообразованию, кто знает основы математики и естественных наук, пользуется компьютерами и смартфонами, но у кого все еще есть вопросы о том, как эти устройства работают. Учителя также найдут содержание книги полезным, так как, по моему мнению, проекты являются хорошим подспорьем в школьном образовании.

Об этой книге

Эта книга рассматривает компьютеры как стек[1] (иерархию) технологий. Современное вычислительное устройство, например смартфон, состоит из нескольких технологических слоев. В нижней части этого множества слоев находится аппаратное обеспечение, в верхней части лежат приложения, а между ними имеется еще много уровней. Преимущество такой многослойной модели заключается в том, что каждый слой использует возможности всех нижних уровней, но каждый конкретный слой должен опираться только на тот слой, который находится непосредственно под ним. После изучения некоторых базовых концепций мы пройдем по технологическим уровням снизу вверх, начиная с электрических цепей и заканчивая технологиями, которые обеспечивают работу веб-сайтов и приложений. Вот что мы рассмотрим в каждой главе.

Глава 1: Принципы компьютерных вычислений. Охватывает основополагающие идеи, такие как понимание аналоговой и цифровой систем, двоичной системы счисления и кратных приставок в системе СИ[2].

Глава 2: Двоичная система в действии. Рассмотрим, как двоичная система может использоваться для представления данных и логических состояний, а также познакомимся с логическими операторами.

Глава 3: Электрические цепи. Объясняет основные понятия, касающиеся электричества и электрических цепей, включая напряжение, ток и сопротивление.

Глава 4: Цифровые схемы. Знакомит с транзисторами и логическими вентилями, а также объединяет понятия из глав 2 и 3.

---

[1] Термин «stack» (стек) означает ряд вложенных друг в друга структур различного уровня, выполняющих каждая свою функцию (подробнее см. далее по тексту). – *Прим. ред.*

[2] СИ в данном случае не название языка программирования C, а общепринятое сокращение для самой распространенной международной системы единиц (SI, от фр. Système International d'unités). – *Прим. ред.*

Глава 5: Математика в цифровых схемах. Показывает, как можно выполнять сложение с помощью цифровых схем, и рассказывает о том, как числа представляются в компьютерах.

Глава 6: Память и синхросигналы. Представляет устройства памяти и последовательные схемы, а также демонстрирует взаимодействие посредством синхросигналов.

Глава 7: Аппаратное обеспечение компьютера. Рассматривает основные составные части компьютера: процессор, память, устройства ввода/вывода.

Глава 8: Машинный код и язык ассемблера. Представляет низкоуровневый машинный код, выполняемый процессорами, а также язык ассемблера – представление машинного кода в удобочитаемом виде.

Глава 9: Программирование высокого уровня. Знакомит с языками программирования, которые не зависят от конкретных процессоров, и включает примеры кода на языках C и Python.

Глава 10: Операционные системы. Рассматривает семейства операционных систем и основные возможности операционных систем.

Глава 11: Интернет. Показывает, как работает интернет, включая общий набор используемых сетевых протоколов.

Глава 12: Всемирная паутина. Объясняет, как функционирует Всемирная паутина, и рассматривает ее основные технологии: HTTP, HTML, CSS и JavaScript.

Глава 13: Современные вычислительные технологии. Содержит обзор нескольких направлений современных вычислительных технологий, таких как приложения, виртуализация и облачные сервисы.

По мере чтения этой книги вам будут встречаться схемы и исходные коды, используемые для иллюстрации концепций. Они предназначены для наглядности обучения, когда отдается предпочтение понятности, а не производительности, безопасности и другим факторам, которые учитывают инженеры-программисты при разработке аппаратного или программного обеспечения. Другими словами, схемы и коды в этой книге могут помочь вам узнать, как работают компьютеры, но они необязательно являются примерами наилучшего способа выполнения задачи. Аналогичным образом в технических разделах книги предпочтение отдается простоте, а не полноте. Так я иногда опускаю некоторые детали, чтобы не завязнуть в сложных объяснениях.

Об упражнениях и проектах

Во всех главах вы найдете упражнения и практические задания. Упражнения – это задачи, которые вы можете решить мысленно или с помощью карандаша и бумаги. Задания же выходят за рамки умственных упражнений и часто предполагают создание схем или программирование.

Для выполнения практических заданий (проектов) вам потребуется приобрести некоторое оборудование (список необходимых компонентов вы найдете в приложении B). Я включил такие проекты, потому что считаю, что лучший способ научиться – это попробовать самому, и я призываю вас выполнять их, если вы хотите получить максимальную пользу от этой книги.

Тем не менее я изложил материалы глав таким образом, чтобы вы могли следовать за повествованием, даже если не соберете ни одной схемы или не введете ни одной строки кода.

Ответы на упражнения вы найдете в приложении A, а подробное описание каждого проекта в конце соответствующей главы. Приложение B содержит информацию, которая поможет вам начать работу с проектами, а в тексте вы найдете отсылки к этой информации, когда она будет нужна.

Копия исходного кода, используемого в проектах, доступна по адресу https:// www.howcomputersreallywork.com/code/. Вы также можете посетить страницу этой книги, https://nostarch.com/how-computers-really-work/, где мы будем предоставлять обновления.

Мое путешествие в мир компьютеров

Мое увлечение компьютерами, вероятно, началось с видеоигр, в которые я играл в детстве. Когда я гостил у бабушки с дедушкой, то часами играл в Frogger, Pac-Man и Donkey Kong на Atari 2600 моей тети. Позже, когда я учился в пятом классе, родители подарили мне на Рождество игровую приставку Nintendo Entertainment System, и я был в восторге! Где-то в процессе увлеченной игры в Super Mario Bros. и Double Dragon я начал интересоваться, как работают видеоигры и компьютеры. К сожалению, игровая приставка Nintendo не очень-то помогала мне понять, что происходит внутри нее.

Примерно в то же время моя семья купила наш первый «настоящий» компьютер, Apple IIGS, открыв для меня новые возможности для изучения того, как именно работают эти машины. К счастью, в моей средней школе был организован класс по программированию на языке BASIC для компьютеров Apple II, и вскоре я понял, что не могу перестать программировать! Я писал код в школе, приносил домой копию своей работы на дискете и продолжал работать дома. На протяжении средней и старшей школы я узнавал все больше о программировании, и мне это очень нравилось. Я также начал понимать, что, хотя BASIC и другие подобные языки программирования позволяют относительно легко указывать компьютеру, что делать, они также во многом скрывают то, как работают компьютеры. Мне захотелось узнать больше.

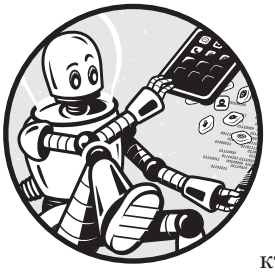
В колледже я изучал электротехнику и начал разбираться в электронике и цифровых схемах. Я посещал занятия по программированию на языке C и языке ассемблера и, наконец, получил некоторое представление о том, как компьютеры выполняют инструкции. Элемен-

ты работы компьютеров на низком уровне начали обретать смысл. Во время учебы в колледже я также начал изучать новую штуку под названием Всемирная паутина – World Wide Web; я даже создал свою собственную веб-страницу (в то время это казалось большим событием)! Я начал программировать приложения для Windows, познакомился с Unix и Linux. Эти темы иногда казались весьма далекими от аппаратных цифровых схем и языка ассемблера, и мне было интересно понять, как все это сочетается друг с другом.

После колледжа мне посчастливилось получить работу в Microsoft. За 17 лет работы там я занимался различными вопросами разработки программного обеспечения, от отладки ядра Windows до разработки веб-приложений. Этот опыт помог мне получить более широкое и глубокое понимание компьютеров. Я работал со многими невероятно умными и знающими людьми и понял, что о компьютерах всегда можно узнать больше. Понимание того, как работают компьютеры, стало для меня путешествием длиною в жизнь, и я надеюсь передать вам часть того, чему я научился, посредством этой книги.

1

ПРИНЦИПЫ КОМПЬЮТЕРНЫХ ВЫЧИСЛЕНИЙ



В наше время компьютеры повсюду: в домах, школах, офисах – вы можете найти компьютер в кармане, на запястье или даже в холодильнике. Найти и использовать компьютеры стало проще, чем когда-либо, но мало кто сегодня действительно понимает, как компьютеры работают. И это неудивительно, поскольку изучение сложных компьютерных технологий может оказаться непосильной задачей. Цель этой книги – изложить основные принципы работы вычислительной техники таким образом, чтобы любой любознательный человек с небольшими техническими способностями смог их понять. Прежде чем мы начнем погружаться в тонкости работы компьютеров, давайте уделим время знакомству с некоторыми основными принципами проведения вычислений в компьютерах.

В этой главе мы начнем с обсуждения определения компьютера. Далее рассмотрим различия между аналоговыми и цифровыми данными, а затем изучим системы счисления и терминологию, используемую для описания цифровых данных.

Определение компьютера

Начнем с базового вопроса: что такое компьютер? Когда люди слышат слово «компьютер», большинство представляет себе ноутбук или настольный компьютер, иногда называемый персональным компьютером, или ПК. Это один из классов устройств, которые рассматриваются в данной книге, но давайте мыслить немного шире. Рассмотрим смартфоны. Смартфоны, безусловно, являются компьютерами; они выполняют те же операции, что и ПК. На самом деле для многих людей сегодня смартфон является основным вычислительным устройством. Большинство пользователей компьютеров сегодня также использует интернет, работа которого обеспечивается серверами, являющимися еще одним типом компьютеров. Каждый раз, когда вы посещаете вебсайт или используете приложение, работающее через интернет, вы взаимодействуете с одним или несколькими серверами, подключенными к глобальной сети. Приставки для видеоигр, фитнес-трекеры, умные часы, умные телевизоры... все это – компьютеры!

Компьютер – это любое электронное устройство, которое может быть запрограммировано на выполнение набора логических инструкций. Учитывая это определение, становится понятно, что многие современные устройства на самом деле являются компьютерами!

УПРАЖНЕНИЕ 1-1: Найдите компьютеры в своем доме

Потратьте немного времени и посмотрите, сколько компьютеров вы сможете найти в своем доме. Когда я выполнял это упражнение со своей семьей, мы быстро нашли около 30 устройств!

Аналоговый и цифровой

Вы наверняка слышали, что компьютер называют цифровым устройством, в отличие от аналоговых устройств, таких как механические часы. Но что на самом деле означают эти два термина? Понимание различий между аналоговыми и цифровыми устройствами является основополагающим для понимания вычислительной техники, поэтому давайте рассмотрим эти два понятия подробнее.

Аналоговый подход

Оглянитесь вокруг себя. Выберите какой-нибудь предмет. Спросите себя: какого он цвета? Какого он размера? Сколько он весит? Отвечая на эти вопросы, вы описываете атрибуты или *данные* этого объекта. Теперь возьмите другой объект и ответьте на те же вопросы. Если вы

повторите этот процесс для большего количества объектов, то обнаружите, что на каждый вопрос существует множество вариантов ответов. Вы можете взять красный, желтый или синий объект. Или цвет объекта может состоять из смеси основных цветов.

Такое разнообразие вариантов относится не только к цвету. Существует потенциально бесконечное количество вариантов определенного свойства среди всех объектов нашего мира.

Одно дело описать объект словами, но, допустим, вы хотите измерить один из его атрибутов более точно. Например, если вы хотите измерить вес объекта, можете положить его на весы. Обычные (не электронные!) весы, реагируя на поставленный на них вес, будут перемещать стрелку вдоль пронумерованной линии, останавливаясь, когда стрелка достигнет положения, соответствующего весу. Возьмите показанное на весах число, и вы получите вес объекта.

Такой способ измерения общераспространен, но давайте подумаем немного о том, как мы измеряем эти данные. Положение стрелки на весах на самом деле не является весом, это представление веса. Пронумерованная линия, на которую указывает игла, позволяет нам легко преобразовать положение иглы, представляющее вес, в числовое значение этого веса. Другими словами, хотя вес является атрибутом объекта, здесь мы можем определить этот атрибут через нечто другое: через положение иглы вдоль линии. Положение иглы изменяется пропорционально в зависимости от веса, помещенного на весы. Таким образом, весы являются *аналогией*, позволяющей нам определить вес объекта через положение иглы на линии. Вот почему мы называем этот метод измерения *аналоговым* подходом.

Другим примером аналогового измерительного прибора является жидкостный (например, ртутный или спиртовой) термометр. Объем жидкости увеличивается с ростом температуры. Производители термометров используют это свойство, помещая подкрашенную жидкость в стеклянную трубку с отметками, которые соответствуют ожидаемому объему жидкости при различных температурах. Таким образом, положение поверхности жидкости в трубке служит отображением температуры. Обратите внимание, что в обоих примерах (весы и термометр), когда производим измерение, мы можем использовать отметки на приборе для преобразования положения в конкретное числовое значение. Но значение, которое мы считываем с прибора, является лишь приблизительным. Истинное положение иглы или высота столбика жидкости могут быть любыми в пределах разрешающей способности прибора, и мы округляем значение в большую или меньшую сторону до ближайшего отмеченного значения. Поэтому, хотя может показаться, что подобные инструменты могут выполнять измерения из ограниченного набора значений (например, ртутный медицинский термометр – от примерно 35 до 42° через каждую десятую, всего около 70 значений), это ограничение связано с преобразованием в число, а не с самим аналоговым принципом.

На протяжении почти всей истории человечества люди измеряли разные вещи, используя аналоговый подход. Но люди используют аналоговый подход не только для измерений. Они также придумали хитроумные способы хранения данных в аналоговом виде. Граммофон-

ная пластинка использует извилистую канавку в качестве аналогового представления записанного звука. Форма канавки изменяется на всем ее протяжении в соответствии с изменениями формы звуковой волны во времени. Канавка – это не сам звук, а аналог формы волны оригинального звука. Пленочные камеры делают нечто подобное, кратковременно подвергая пленку воздействию света через объектив камеры, что приводит к химическим изменениям на пленке. Химические свойства пленки – это не само изображение, а представление запечатленного изображения, аналогия изображения.

Переход на цифровые технологии

Какое отношение все это имеет к вычислительной технике? Оказывается, компьютерам трудно оперировать всеми этими аналоговыми представлениями. Типы используемых аналоговых систем настолько разнообразны и изменчивы, что создать вычислительное устройство, способное понимать все из них, практически невозможно. Например, создание машины, способной измерить объем ртути, сильно отличается от создания машины, способной прочитать канавки на виниловом диске. Кроме того, компьютеры требуют высоконадежного и точного представления определенных типов данных, например числовых наборов данных и программного обеспечения. Аналоговые представления данных трудно поддаются точному измерению, имеют тенденцию разрушаться со временем и теряют точность при копировании. Компьютерам необходим способ представления всех типов данных в формате, который можно тщательно обрабатывать, хранить и копировать.

Что же мы можем сделать, если не хотим представлять данные в виде аналоговых значений, которые в принципе могут уточняться до бесконечности? Вместо этого мы можем использовать цифровой подход. *Цифровая* система представляет данные в виде последовательности символов, где каждый символ принадлежит ограниченному набору значений. Такое описание может показаться несколько формальным и не очень понятным, поэтому вместо того, чтобы углубляться в теорию цифровых систем, я объясню, что это означает на практике. Почти во всех современных компьютерах данные представлены комбинациями из двух символов: 0 и 1. Вот и все. Хотя цифровая система может использовать более двух символов, добавление большего их количества приведет к увеличению сложности и стоимости системы. Набор всего из двух символов позволяет упростить техническое обеспечение и повысить надежность. Все данные в большинстве современных вычислительных устройств представлены в виде последовательности из нулей и единиц. С этого момента в данной книге, когда я буду говорить о цифровых компьютерах, вы можете считать, что я говорю о системах, которые имеют дело только с нулями и единицами, а не с каким-то другим набором символов. Легко и просто!

Этот момент стоит повторить: все на вашем компьютере хранится в виде нулей и единиц. Последняя фотография, которую вы сделали на смартфон? Ваше устройство хранит ее как последовательность из нулей

и единиц. Песня, которую вы загрузили из интернета? Нули и единицы. Документ, который напечатали на компьютере? Нули и единицы. Приложение, которое вы установили? Это было множество из нулей и единиц. Веб-сайт, который посетили? Нули и единицы.

Может показаться сомнительным, что мы можем использовать только 0 и 1 для представления бесконечного числа значений, встречающихся в природе. Как можно свести музыкальную запись или детальную фотографию к нулям и единицам? Многим кажется парадоксальным тот факт, что столь ограниченный «словарный запас» может использоваться для выражения сложных идей. Ключевым моментом здесь является то, что цифровые системы используют *последовательность* из нулей и единиц. Цифровая фотография, например, обычно состоит из миллионов нулей и единиц.

Что же конкретно представляют собой эти нули и единицы? Вы можете встретить другие термины, используемые для описания этих нулей и единиц: ложь и истина, выключенное и включенное состояние, низкий и высокий уровень и т. д. Это происходит потому, что компьютер не хранит буквально числа 0 или 1. Он хранит последовательность состояний, где каждое состояние в последовательности может иметь только два возможных значения. Каждое состояние подобно выключателю света, который либо включен, либо выключен. На практике эти последовательности нулей и единиц хранятся по-разному. На CD- или DVD-дисках нули и единицы хранятся в виде выпуклостей (0) или ровных участков (1). На флеш-накопителях нули и единицы хранятся в виде электрических зарядов. На жестком диске нули и единицы хранятся с помощью ориентированных в разных направлениях магнитных зон (доменов). Как вы увидите в главе 4, цифровые схемы для представления нулей и единиц используют уровни электрического напряжения.

Прежде чем мы продолжим, последнее замечание по поводу термина *«аналоговый»* – он часто используется просто для обозначения «нецифровой». Например, инженеры могут говорить об аналоговом сигнале, имея в виду сигнал, который непрерывно изменяется и не соотносится с цифровыми значениями. Другими словами, это нецифровой сигнал, но он не обязательно представляет собой аналогию чего-либо другого. Поэтому, когда вы видите термин «аналоговый», имейте в виду, что он не всегда означает то, что вы думаете.

Системы счисления

Итак, мы выяснили, что компьютеры – это цифровые машины, которые работают с нулями и единицами. Для многих людей эта концепция кажется странной; они привыкли, что для представления чисел в их распоряжении имеются символы от 0 до 9. Если мы ограничиваемся только двумя символами, а не десятью, как нам представлять большие числа? Чтобы ответить на этот вопрос, давайте вернемся в прошлое и рассмотрим тему из математики начальной школы о системах счисления.

Десятичные числа

Обычно мы записываем числа, используя так называемую *десятичную позиционную систему счисления*. Давайте разберемся в этом. *Позиционная система счисления* означает, что каждая позиция в написанном числе представляет собой отдельный порядок величины; *десятичная*, или *по основанию 10*, означает, что порядки величины являются множителями 10, и на каждой позиции может быть один из десяти различных символов, от 0 до 9. Посмотрите на пример позиционной системы счисления на рис. 1-1.

Рис. 1-1. Число двести семьдесят пять, представленное в десятичной позиционной системе счисления

На рис. 1-1 число двести семьдесят пять записано в десятичной системе счисления как 275. Цифра 5 стоит на месте единиц, т. е. ее значение равно 5 × 1 = 5. 7 стоит на месте десятков, т. е. ее значение равно 7 × 10 = 70. 2 стоит на месте сотен, т. е. ее значение равно 2 × 100 = 200. Общее значение – это сумма всех позиций: 5 + 70 + 200 = 275.

Легко, правда? Вы, наверное, поняли это еще в первом классе. Но давайте рассмотрим это немного подробнее. Почему крайнее правое место – это место единицы? И почему следующее место – место десятков и т. д.? Потому что мы работаем в десятичной системе, или с основанием 10, и поэтому каждая позиция – это степень десяти; другими словами, 10 умножается на себя определенное количество раз. Как видно на рис. 1-2, самая правая позиция – это 10, возведенное в степень 0, что равно 1, так как любое число, возведенное в степень 0, равно 1. Следующая позиция – это 10, возведенное в степень 1, что равно 10, а еще следующее место – это 10, возведенное в степень 2 (10 × 10), что равно 100.

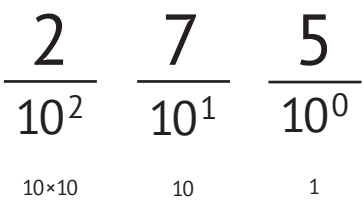


Рис. 1-2. В десятичной позиционной системе счисления каждая позиция – это степень десяти

Если бы нам нужно было представить число больше 999 в десятичной системе, мы бы добавили еще одну позицию слева, позицию тысяч, и вес этой позиции был бы равен 10 в степени 3 (10 × 10 × 10), т. е. 1000. По такой схеме мы можем представить любое большое целое число, добавляя по мере необходимости новые позиции.

Мы выяснили, почему различные позиции имеют определенный вес, но давайте продолжим копать дальше. Почему на каждой позиции используются символы от 0 до 9? При работе с десятичными системами у нас может быть только десять символов, потому что по определению каждая позиция может представлять только десять различных значений. В настоящее время используются символы от 0 до 9, но на самом деле можно использовать любой набор из десяти уникальных символов, каждый из которых соответствует определенному числовому значению.

Большинство людей предпочитает десятичную систему счисления, или систему с основанием 10. Некоторые считают, что это потому, что у нас десять пальцев на руках и десять на ногах, но какова бы ни была причина, в современном мире большинство людей читает, пишет и думает о числах в десятичной системе счисления. Конечно, это просто правило, которое мы коллективно выбрали для представления чисел. Как мы уже говорили ранее, это правило не относится к компьютерам, которые используют только два символа. Давайте посмотрим, как мы можем применять принципы позиционной системы счисления, ограничиваясь только двумя символами.

Двоичные числа

Система счисления, состоящая только из двух символов, – это система с основанием 2, или двоичная система. Двоичная система по-прежнему является позиционной системой счисления, поэтому фундаментальная техника такая же, как и у десятичной системы, но есть несколько изменений. Во-первых, каждая позиция представляет собой степень 2, а не степень 10. Во-вторых, на каждой позиции может быть только один из двух символов, а не один из десяти. Эти два символа – 0 и 1. На рис. 1-3 приведен пример представления числа в двоичной системе счисления.

Рис. 1-3. Число пять в десятичной системе, представленное в двоичной позиционной системе счисления

На рис. 1-3 представлено двоичное число: 101. Для вас это может выглядеть как сто один, но при работе с двоичными числами это на самом деле представление пяти! Если вы хотите назвать это число словами, то «один–ноль–один (двоичное)» будет хорошим способом передать написанное.

Как и в десятичной системе счисления, каждая позиция имеет вес, равный основанию, возведенному в различные степени. Так как у нас

основание 2, самая правая позиция – это 2 в степени 0, т. е. 1. Следующая позиция – это 2 в степени 1, т. е. 2, и еще следующая позиция – это 2, возведенное в степень 2 (2 × 2), т. е. 4. Так же как и в десятичной системе, для получения общего значения мы умножаем символ в каждой позиции на вес позиции и суммируем результаты. Итак, начиная справа, мы имеем (1 × 1) + (0 × 2) + (1 × 4) = 5.

Теперь вы можете попробовать самостоятельно перевести двоичную систему счисления в десятичную.

УПРАЖНЕНИЕ 1-2: Двоичное в десятичное

Переведите представленные в двоичной системе числа в их десятичные эквиваленты:

10 (двоичное) = _____(десятичное);
111 (двоичное) = _____(десятичное);
1010 (двоичное) = _____(десятичное).

Вы можете проверить ответы в приложении А. У вас все получилось верно?

Последний пример может показаться немного сложным, поскольку он вводит еще одну позицию слева – позицию восьмерки. Теперь попробуйте пересчитать в обратном направлении, из десятичной системы счисления в двоичную.

УПРАЖНЕНИЕ 1-3: Десятичное в двоичное

Переведите представленные в десятичной системе числа в их двоичные эквиваленты:

3 (десятичное) = _____(двоичное);
8 (десятичное) = _____(двоичное);
14 (десятичное) = _____(двоичное).

Я надеюсь, что это вы тоже сделали без ошибок! Сразу можно увидеть, что одновременная работа с десятичной и двоичной системами счисления может сбить с толку, поскольку такое число, как 10, обозначает десять в десятичной системе счисления или два в двоичной. С этого момента в книге, там, где есть вероятность путаницы, двоичные числа будут записываться с префиксом 0b. Я выбрал префикс 0b, потому что несколько языков программирования использует этот подход. Ведущий символ 0 (ноль) обозначает числовое значение, а b – сокращение от binary (двоичный). Например, 0b10 означает два в двоичном исчислении, тогда как 10 без префикса означает десять в десятичном.

Биты и байты

Одна позиция или символ в десятичном числе называется цифрой. Десятичное число 1247 – это четырехзначное число. Аналогично одна позиция или символ в двоичном числе называется битом (двоичной цифрой). Каждый бит может быть либо 0, либо 1. Такое двоичное число, как 0b110, является трехразрядным.

Один бит не может передать много информации; он выключен либо включен, 0 или 1. Для представления чего-либо более сложного нам нужна последовательность битов. Чтобы облегчить работу с этими последовательностями битов, компьютеры объединяют биты в наборы по восемь штук, называемые байтами. Вот несколько примеров битов и байтов (без префикса 0b, поскольку все они двоичные).

1 – это бит.
0 – это тоже бит.
11001110 – это один байт, или 8 бит.
00111000 – это тоже байт!
10100101 – еще один байт.
0011100010100101 – это два байта, или 16 бит.

ПРИМЕЧАНИЕ

Забавный факт: четырехбитное число, половина байта, иногда называется ниббл[1].

Так сколько же данных мы можем хранить в байте? Другой способ посмотреть на этот вопрос – это найти, сколько уникальных комбинаций из нулей и единиц мы можем сделать с помощью наших 8 бит? Прежде чем мы ответим на этот вопрос, позвольте мне проиллюстрировать его на примере только 4 бит, так как это будет легче представить.

В табл. 1.1 я перечислил все возможные комбинации из нулей и единиц в четырехбитном числе. Я также включил соответствующее десятичное представление этих чисел.

Как видно из табл. 1.1, мы можем представить 16 уникальных комбинаций из нулей и единиц в четырехбитном числе, расположенных по порядку от 0 до 15 в десятичном исчислении. Представление списка комбинаций битов хорошо это иллюстрирует, но мы могли бы выяснить это также несколькими способами без перечисления всех возможных комбинаций.

[1] Nibble – огрызок, кусочек. В русскоязычной информатике для половины байта приняты термины «тетрада» или «полубайт». Англоязычное «ниббл» в русскоязычной практике употребляется очень редко, потому далее в этой книге мы будем заменять его более привычными и понятными русскими терминами. – *Прим. ред.*

Таблица 1.1. Все возможные значения четырехбитного числа

Двоичное	Десятичное
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10

Мы можем определить наибольшее возможное число, которое могут представлять 4 бита, установив все биты равными единице, что даст нам 0b1111. Это 15 в десятичной системе счисления; если мы прибавим 1 для учета представления 0, то получим наши 16 комбинаций. Другой короткий путь – увеличить 2 в количество раз, равное количеству битов, в данном случае 4, что дает нам $2^4 = 2 \times 2 \times 2 \times 2 = 16$ комбинаций из нулей и единиц.

Рассмотрение 4 бит – это хорошее начало, но ранее мы говорили о байтах, которые содержат по 8 бит. Используя предыдущий подход, мы могли бы перечислить все комбинации из нулей и единиц, но давайте пропустим этот шаг и сразу перейдем к короткому пути. Возведем 2 в степень 8 и получим 256 – это и есть количество уникальных комбинаций битов в байте.

Теперь мы знаем, что четырехбитное число допускает 16 комбинаций из нулей и единиц, а байт допускает 256 комбинаций. Какое отношение это имеет к вычислительным технологиям? Допустим, в компьютерной игре 12 уровней; игра может легко хранить номер текущего уровня всего в 4 битах. С другой стороны, если в игре 99 уровней, 4 бит будет недостаточно... можно представить только 16 уровней! Байт, с другой стороны, легко справится с этим требованием 99 уровней. Компьютерным инженерам нередко приходится учитывать, сколько бит или байт потребуется для хранения данных.

Префиксы

Для представления сложных типов данных требуется большое количество битов. Для такого простого числа, как 99, требуется не более байта; видео в цифровом формате, с другой стороны, может потребовать миллиарды бит. Чтобы проще передать размеры данных, мы используем такие префиксы, как гига- и мега-. Международная система единиц (СИ), также известная как метрическая система, определяет

набор стандартных префиксов. Эти префиксы используются для описания всего, что поддается количественной оценке, а не только битов. Мы еще встретимся с ними в последующих главах, посвященных электрическим цепям. В табл. 1.2 перечислены некоторые из распространенных префиксов СИ и их значения.

Таблица 1.2. Распространенные префиксы СИ

Название префикса		Обозначение префикса		Значение	Основание 10	Наименование
Англ.	Русск.	Англ.	Русск.			
tera	тера	T	Т	1 000 000 000 000	10^{12}	триллион
giga	гига	G	Г	1 000 000 000	10^{9}	миллиард
mega	мега	M	М	1 000 000	10^{6}	миллион
kilo	кило	k	к (К)	1000	10^{3}	тысяча
centi	санти	c	с	0,01	10^{-2}	сотый
milli	милли	m	м	0,001	10^{-3}	тысячный
micro	микро	μ	мк	0,000001	10^{-6}	миллионный
nano	нано	n	н	0,000000001	10^{-9}	миллиардный
pico	пико	p	п	0,000000000001	10^{-12}	триллионный

С помощью этих префиксов, если мы хотим сказать «три миллиарда байт», мы можем использовать сокращение 3 ГБ. Или, если хотим обозначить четыре тысячи бит, можем сказать 4 Кбит. Обратите внимание, что байты обозначаются как «Б» (в англоязычном написании «b», иногда «B»), а биты как «бит» («bit»)[1].

Вы увидите, что это правило часто используется для обозначения количества битов и байтов. К сожалению, оно также часто технически неверно. Вот почему: при работе с байтами большая часть программного обеспечения на самом деле работает с основанием 2, а не с основанием 10. Если ваш компьютер говорит вам, что размер файла составляет 1 МБ, на самом деле это 1 048 576 байт! Это приблизительно один миллион, но не совсем. Это число кажется странным, не так ли? Это потому, что мы смотрим на него в десятичной системе счисления. В двоичной системе это же число выражается как 0b10000000000000000000. Это степень двух, а именно 2^{20}. В табл. 1.3 показано, как интерпретировать префиксы СИ при работе с байтами.

[1] Согласно ГОСТ 8.417-2002, в русскоязычной практике приставка «кило» в физике сокращается до маленькой буквы (кг, а не Кг, кВт, а не КВт), а в компьютерной технике до заглавной (Кбит, КБ), чтобы подчеркнуть, что речь идет о множителе 1024, а не 1000 (см. далее в тексте). В англоязычной практике этого правила не сложилось, потому там «кило» принято сокращать до маленькой буквы (kbit). Путаться между этими обозначениями не следует: повторим, что «Кбит» правильно переводить на английский как «kbit», и наоборот.
Ранее часто биты обозначали строчной «б», но это приводило к многочисленным ошибкам при чтении. Поэтому устоялось сокращение для бита полностью («бит»). Байт, согласно отечественным правилам (ГОСТ 8.417-2002), следует сокращать до заглавной буквы «Б», но часто во избежание путаницы его пишут также полностью (Кбайт, Мбайт). – *Прим. ред.*

Таблица 1.3. Значение префиксов при применении их к байтам

Название префикса		Обозначение префикса		Значение	Основание 10
Англ.	Русск.	Англ.	Русск.		
tera	тера	T	Т	1,099,511,627,776	2^{40}
giga	гига	G	Г	1,073,741,824	2^{30}
mega	мега	M	М	1,048,576	2^{20}
kilo	кило	k	К	1,024	2^{10}

Другой момент путаницы с битами и байтами связан со скоростью передачи данных в сети. Интернет-провайдеры обычно заявляют скорость передачи данных в битах в секунду по основанию 10. Так, если вы получаете 50 Мбит в секунду при подключении к интернету, это означает, что вы можете передавать только 6 МБ в секунду. То есть 50 000 000 бит в секунду, разделенных на 8 бит в одном байте, дают нам 6 250 000 байт в секунду. Разделив 6 250 000 на 2^{20}, мы получим около 6 МБ в секунду.

Префиксы СИ для двоичных данных

Для устранения путаницы, вызванной многозначностью префиксов, в 2002 году был введен новый набор префиксов (в стандарте под названием IEEE 1541) для использования в двоичных системах. Когда речь идет о степенях числа 2, вместо кило- следует использовать киби-, вместо мега- – меби- и т. д. Эти новые префиксы соответствуют значениям с основанием 2 и предназначены для использования в ситуациях, где старые префиксы ранее использовались неправильно. Например, поскольку килобайт может интерпретироваться как 1 000 или 1 024 байт, данный стандарт рекомендует использовать кибибайты для обозначения 1 024 байт, а кило- сохраняет свое первоначальное значение, так что килобайт равен 1 000 байт.

Это кажется хорошей идеей, но на момент написания данной книги эти символы еще не были широко распространены. В табл. 1.4 перечислены новые префиксы и их значения.

Таблица 1.4. Префиксы для двоичных данных согласно IEEE 1541-2002

Название префикса		Обозначение префикса		Значение	Основание 2
Англ.	Русск.	Англ.	Русск.		
tebi	теби	Ti	Ти	1 099 511 627 776	2^{40}
gibi	гиби	Gi	Ги	1 073 741 824	2^{30}
mebi	меби	Mi	Ми	1 048 576	2^{20}
kibi	киби	Ki	Ки	1024	2^{10}

Это различие важно, потому что на практике большая часть программного обеспечения, отображающего размер файлов, использует старый префикс СИ, но вычисляет размер по основанию 2. Другими словами, если ваше устройство сообщает, что размер файла составляет 1 КБ, это означает 1024 байта. С другой стороны, производители запоминающих устройств, как правило, заявляют емкость своих устройств, используя основание 10. Это означает, что жесткий диск, емкость которого заявлена как 1 ТБ, вероятно, содержит один триллион байт, но, если вы подключите это устройство к компьютеру, компьютер покажет размер около 931 ГБ (один триллион, разделенный на 2^{30}). Учитывая отсутствие распространения новых префиксов, в этой книге я продолжу использовать старые префиксы СИ.

Шестнадцатеричная система

Прежде чем мы оставим тему двоичной системы, я расскажу еще об одной системе счисления – шестнадцатеричной. Если кратко резюмировать, то наша «обычная» система счисления – десятичная, или с основанием 10. Компьютеры используют двоичную систему счисления, или с основанием 2. *Шестнадцатеричная* – это система *с основанием 16*! Учитывая то, что вы уже узнали в этой главе, вы, вероятно, понимаете, что это значит. Шестнадцатеричная система счисления – это позиционная система счисления, в которой каждая позиция представляет собой степень шестнадцати и на каждой позиции может быть один из шестнадцати символов.

Как и во всех позиционных системах счисления, крайняя правая позиция по-прежнему будет позицией единиц. Следующая позиция слева будет позицией 16, затем позиция 256 (16 × 16), затем – 4 096 (16 × 16 × 16) и т. д. Достаточно просто. Но как насчет другого требования, что на каждой позиции может быть один из 16 символов? Обычно мы используем десять символов для представления чисел, от 0 до 9. Нам нужно добавить еще шесть символов для представления других значений. Мы могли бы выбрать несколько случайных символов, например & @ #, но эти символы не имеют очевидного порядка. Вместо этого стандартом является использование A, B, C, D, E и F (прописные или строчные – неважно!). В этой схеме A обозначает 10, B – 11 и т. д., вплоть до F, которая обозначает 15. Это имеет смысл, так как нам нужны символы, которые представляют значения от нуля до значения основания за вычетом единицы. Поэтому наши дополнительные символы – это буквы от A до F. Стандартной практикой является использование префикса 0x для обозначения шестнадцатеричной системы, когда это необходимо подчеркнуть. В табл. 1.5 перечислены все 16 шестнадцатеричных символов, а также их десятичные и двоичные эквиваленты.

Таблица 1.5. Шестнадцатеричные символы

Шестнадцатеричный	Десятичный	Двоичный (4-бит)
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Что же происходит, когда вам нужно считать дальше, чем 15 в десятичной системе или 0xF? Как и в десятичной системе счисления, мы добавляем еще одну позицию. После 0xF идет 0x10, что равно 16 в десятичной системе. Затем 0x11, 0x12, 0x13 и т. д. Теперь посмотрите на рис. 1-4, где мы видим большее шестнадцатеричное число, 0x1A5.

Рис. 1-4. Число 0x1A5 в шестнадцатеричной системе, разбитое по значениям на позициях

На рис. 1-4 мы имеем число 0x1A5 в шестнадцатеричной системе счисления. Каково значение этого числа в десятичной системе? Крайняя правая позиция имеет значение 5. Следующая позиция имеет вес 16, и там стоит буква A, что в десятичной системе равно 10, поэтому средняя позиция означает 16 × 10 = 160. Крайняя левая позиция имеет вес 256, и на этом месте стоит 1, поэтому эта позиция означает 256. Таким образом, суммарное значение равно 5 + 160 + 256 = 421 в десятичной системе.

Чтобы подчеркнуть вышесказанное, отметим, что этот пример показывает, что новые символы, такие как A, имеют разное значение в зависимости от позиции, на которой они находятся. 0xA – это 10 в десятичной системе, но 0xA0 – это 160 в десятичной системе, потому что A стоит на позиции шестнадцатых.

В этот момент вы, возможно, скажете себе: «Отлично, но зачем это нужно?» Я рад, что вы спросили. Компьютеры не используют шестнадцатеричную систему счисления, как и большинство людей. И все же шестнадцатеричная система очень полезна для людей, которым необходимо работать в двоичной системе.

Использование шестнадцатеричной системы помогает преодолеть два сложных момента, связанных с работой в двоичной системе. Во-первых, большинство людей с большим трудом читает длинные последовательности из нулей и единиц. Через некоторое время все биты сбиваются в кучу. Работа с 16 и более битами утомительна и чревата ошибками. Вторая проблема заключается в том, что, хотя люди хорошо умеют работать в десятичной системе счисления, преобразование между десятичной и двоичной системами не такое уж простое. Большинству людей трудно с одного взгляда на десятичное число быстро определить, какие биты будут единицами, а какие нулями, если бы это число было представлено в двоичном виде. Но в шестнадцатеричной системе счисления преобразование в двоичную систему гораздо проще. В табл. 1.6 приведена пара примеров 16-битных двоичных чисел и их соответствующих шестнадцатеричных и десятичных представлений. Обратите внимание, что для наглядности я вставил пробелы в двоичные значения.

Таблица 1.6. Примеры 16-битных двоичных чисел как десятичных и шестнадцатеричных

	Пример 1	**Пример 2**
Двоичное	1111 0000 0000 1111	1000 1000 1000 0001
Шестнадцатеричное	F00F	8881
Десятичное	61,455	34,945

Рассмотрим пример 1 в табл. 1.6. В двоичном исчислении существует четкая последовательность: первые четыре бита равны 1, следующие восемь бит равны 0, а последние четыре бита равны 1. В десятичной системе счисления эта последовательность не просматривается. При взгляде на число 61 455 совершенно неясно, какие биты могут быть нулевыми или единичными. С другой стороны, шестнадцатеричная система отражает последовательность в двоичной системе. Первый шестнадцатеричный символ – F (что в двоичном исчислении равно 1111), следующие два шестнадцатеричных символа – 0, а последний шестнадцатеричный символ – F.

Перейдем к примеру 2, первые три набора из четырех бит равны 1000, а последний набор из четырех бит – 0001. Это легко увидеть в двоичной системе, но довольно трудно в десятичной. Шестнадцатеричная

система дает более четкую картину: шестнадцатеричный символ 8 соответствует 1000 в двоичной системе, а шестнадцатеричный символ 1 так и соответствует 1!

Надеюсь, вы заметили закономерность: каждые четыре бита в двоичной системе соответствуют одному символу в шестнадцатеричной. Если вы помните, четыре бита – это половина байта (или тетрада). Поэтому байт можно легко представить двумя шестнадцатеричными символами. 16-битное число можно представить четырьмя шестнадцатеричными символами, 32-битное число – восемью шестнадцатеричными символами и т. д. В качестве примера возьмем 32-битное число на рис. 1-5.

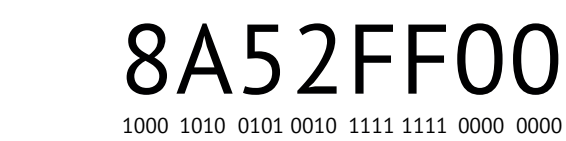


Рис. 1-5. Каждый шестнадцатеричный символ соответствует 4 битам

На рис. 1-5 мы можем переварить это довольно длинное число по полбайта за раз, что невозможно при использовании десятичного представления того же числа (2 320 695 040).

Поскольку переходить от двоичной системы к шестнадцатеричной относительно легко, многие инженеры часто используют эти две системы в тандеме, переходя к десятичным числам только в случае необходимости. Позже в этой книге я буду использовать шестнадцатеричную систему, когда это будет иметь смысл.

Попробуйте преобразовать двоичные числа в шестнадцатеричные, не проходя промежуточный этап преобразования в десятичные.

УПРАЖНЕНИЕ 1-4: Из двоичной системы в шестнадцатеричную

Переведите эти числа, представленные в двоичном формате, в их шестнадцатеричные эквиваленты. Не переводите в десятичную систему, если это возможно! Цель состоит в том, чтобы перейти непосредственно от двоичной системы к шестнадцатеричной.

10 (двоичное) = _____ (шестнадцатеричное).
11110000 (двоичное) = _____ (шестнадцатеричное).
Вы можете проверить свои ответы в приложении A.

Как только вы освоите перевод из двоичной системы в шестнадцатеричную, попробуйте перейти в обратном направлении, из шестнадцатеричной системы в двоичную.

УПРАЖНЕНИЕ 1-5: Из шестнадцатеричной в двоичную

Преобразуйте эти числа, представленные в шестнадцатеричной системе счисления, в их двоичные эквиваленты. Не переводите в десятичную систему счисления, если это возможно! Цель состоит в том, чтобы перейти непосредственно от шестнадцатеричной системы к двоичной.

1A (шестнадцатеричное) = _____ (двоичное).
C3A0 (шестнадцатеричное) = _____ (двоичное).
Вы можете проверить свои ответы в приложении A.

Выводы

В этой главе мы рассмотрели некоторые основополагающие понятия вычислительной техники. Вы узнали, что компьютер – это любое электронное устройство, которое может быть запрограммировано на выполнение набора логических инструкций. Затем вы увидели, что современные компьютеры являются скорее цифровыми, чем аналоговыми устройствами, и узнали разницу между этими понятиями: аналоговые системы – это системы, использующие для представления данных широко варьирующиеся значения, в то время как цифровые системы представляют данные в виде последовательности символов. После этого мы рассмотрели, как современные цифровые компьютеры используют только два символа, 0 и 1, и узнали о системе счисления по основанию 2, состоящей только из двух символов, или двоичной системе. Мы разобрали биты, байты и стандартные префиксы СИ (гига-, мега-, кило- и т. д.), которые можно использовать для более удобного описания размера данных. Наконец, вы узнали, как шестнадцатеричная система счисления полезна для людей, которым необходимо работать в двоичной системе.

В следующей главе мы более подробно рассмотрим, как двоичная система используется в цифровых системах. Мы рассмотрим, как двоичная система может использоваться для представления различных типов данных, и увидим, как работает двоичная логика.