1.5.1 Numbers

Integers from 0 to 255 can be represented in one byte using the binary (base-2) system as follows:

Decimal	Binary
0	00000000
1	00000001
2	00000010
3	00000011
4	00000100
5	00000101
6	00000110
7	00000111
8	00001000
9	00001001
10	00001010
...	...
252	11111100
253	11111101
254	11111110
255	11111111

If we use 2 bytes (16 bits), we can represent integers from 0 to $2^{16}–1$ = 65535:

Decimal	Binary
0	00000000 00000000
1	00000000 00000001
2	00000000 00000010
...	...
65534	11111111 11111110
65535	11111111 11111111

In general, *k* bits can produce 2^k different combinations of 0s and 1s. Therefore, *k* bits used as binary digits can represent non-negative integers in the range from 0 to $2^k–1$. A 32-bit memory address can identify 2^{32} = 4,294,967,296 different memory locations. So if we want to be able to address each individual byte, 32-bit addresses cover 4 GB of memory space.

❖ ❖ ❖

Copyright © 2015-2020 by Skylight Publishing

CPUs perform all their arithmetic operations on binary numbers. A CPU may have instructions that perform 16-bit, 32-bit, or 64-bit arithmetic, for instance. Since it is difficult for a human brain to grasp long sequences of 0s and 1s, programmers who have to deal with binary data often use the *hexadecimal* (or simply "*hex*") representation in their documentation and programs. The hexadecimal system is the base-16 system, which uses 16 "digits." The first ten digits are the usual 0 through 9, with the eleventh through sixteenth digits represented by the letters A through F. A byte can be split into two four-bit *quads*; each quad represents one hex digit, as follows:

Decimal	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Experienced programmers remember the bit patterns for the sixteen hex digits and can easily convert a binary number into hex and back. It is often convenient to use hex representation as an intermediate step for converting numbers from binary to decimal and back. For example:

$$700_{10} = 2\cdot16^2 + 11\cdot16 + 12 = 2BC_{16} = (\underbrace{0010}_{2}\underbrace{1011}_{B}\underbrace{1100}_{C})_2$$

$$(\underbrace{110}_{6}\underbrace{1101}_{D})_2 = 6D_{16} = 6\cdot16 + 13 = 109_{10}$$

Copyright © 2015-2020 by Skylight Publishing

The following examples show a few numbers represented in the decimal, hex, and 16-bit binary systems:

Decimal	Hex	Binary
0	0000	00000000 00000000
1	0001	00000000 00000001
12	000C	00000000 00001100
32	0020	00000000 00100000
128	0080	00000000 10000000
255	00FF	00000000 11111111
256	0100	00000001 00000000
32767	7FFF	01111111 11111111
32768	8000	10000000 00000000
65535	FFFF	11111111 11111111

For a software developer, knowing the hex system is a matter of cultural literacy. In practice, programmers who use a high-level programming language, like Java, don't have to use it very often and there are calculators and programs that can do conversions.

❖ ❖ ❖

What about negative numbers? The same bit pattern may represent an unsigned (positive) integer and a negative integer, depending on how a particular instruction interprets it. Suppose we use 32-bit binary numbers, but now we decide that they represent signed integers. Positive integers from 0 to $2^{31}-1$ can be represented as before. These use only the 31 least significant bits. As to negative integers, their representation may be *machine-dependent*, varying from CPU to CPU. Many CPUs, including the Intel family, use a method called *two's-complement arithmetic*. In this method, a negative integer x in the range from -1 to -2^{31} is represented the same way as the unsigned binary number $2^{32}-|x|$ where $|x|$ is the absolute value of x. For example, 37 and -37 will be represented as:

$$00000000\ 00000000\ 00000000\ 00100101 = 37_{10}$$
$$11111111\ 11111111\ 11111111\ 11011011 = -37_{10}$$

There are dozens of applets on the Internet that illustrate binary representation of numbers and conversions from decimal to binary.

❖ ❖ ❖

Copyright © 2015-2020 by Skylight Publishing

Real numbers are represented using one of the standard formats expected by the CPU (or a separate floating-point arithmetic unit). Like scientific notation, this representation consists of a fractional part (mantissa) and an exponent part, but here both parts are represented as binary numbers. The IEEE (Institute of Electrical and Electronics Engineers) standard for a 4-byte (32-bit) representation uses 1 bit for the sign, 8 bits for the exponent and 23 bits for the mantissa. 127 is added to the exponent to ensure that negative exponents are still represented by non-negative numbers. This format lets programmers represent numbers in the range from approximately -3.4×10^{38} to 3.4×10^{38} with at least seven digits of precision. Figure 1-5 gives a few examples.

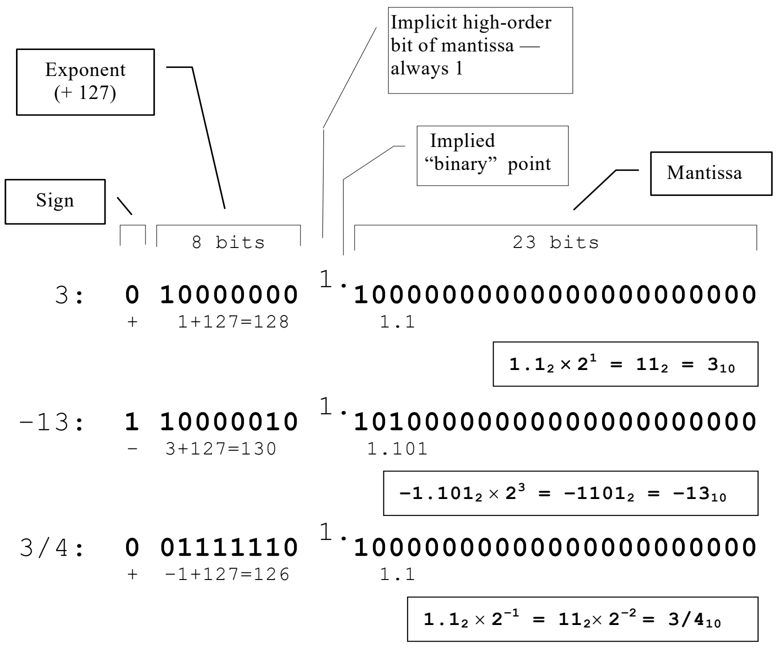

Figure 1-5. IEEE standard representation of 32-bit floating-point numbers

Copyright © 2015-2020 by Skylight Publishing

1.5.2 Characters

Characters are represented by numeric codes. These days most applications use platform-independent Unicode standard for encoding characters [1]. For example, the reason you can see the sentence written in Russian — Многие программы используют Юникод [1] — is that the software that displays this document understands the Unicode representation of Cyrillic characters.

Java programs use Unicode internally for representing characters and text strings.

Unicode uses two bytes per character. It can encode up to 65,000 characters, enough to encode the alphabets of most world languages and many special characters. Unicode has a provision to extend the character set even further, into millions of different codes.

In the old days, the two most common character codes were EBCDIC (Extended Binary Coded Decimal Interchange Code) [1], used in IBM mainframes, and ASCII (American Standard Code for Information Interchange, pronounced as'-kee), used in personal computers, printers and other devices. Both of these use one byte per character. In the PC world, the term *ASCII file* refers to a text file (in which characters are represented in ASCII code), as opposed to a *binary file* that may contain numbers, images, or any other digitized information. Normally you won't find EBCDIC-encoded data on a PC unless the file originated on a mainframe.

Unicode includes ASCII codes as a subset (called "C0 Controls and Basic Latin"). ASCII code defines 128 characters with codes from 0 to 127 and uses only the seven least-significant bits of a byte. Codes from 33 to 127 represent "printable" characters: digits, upper- and lowercase letters, punctuation marks, and so on. 32 (hex 20) is a space.

The first 32 ASCII codes (0-31) are reserved for special control codes. For example, code 13 (hex 0D) is "carriage return" (CR), 10 (hex 0A) is "line feed" (LF), 12 (hex 0C) is "form feed" (FF) and 9 (hex 09) is "horizontal tab" (HT). How control codes are used may depend to some extent on the program or device that processes them. A standard ASCII table, including the more obscure control codes, is presented in Figure 1-6.

Copyright © 2015-2020 by Skylight Publishing

Hex	0_	1_	2_	3_	4_	5_	6_	7_
_0	0 NUL	16 DEL	32 (SPACE)	48 0	64 @	80 P	96 `	112 p
_1	1 SOH	17 DC1	33 !	49 1	65 A	81 Q	97 a	113 q
_2	2 STX	18 DC2	34 "	50 2	66 B	82 R	98 b	114 r
_3	3 ETX	19 DC3	35 #	51 3	67 C	83 S	99 c	115 s
_4	4 EOT	20 DC4	36 $	52 4	68 D	84 T	100 d	116 t
_5	5 ENQ	21 NAK	37 %	53 5	69 E	85 U	101 e	117 u
_6	6 ACK	22 SYN	38 &	54 6	70 F	86 V	102 f	118 v
_7	7 BEL	23 ETB	39 '	55 7	71 G	87 W	103 g	119 w
_8	8 BS	24 CAN	40 (	56 8	72 H	88 X	104 h	120 x
_9	9 HT	25 EM	41)	57 9	73 I	89 Y	105 i	121 y
_A	10 LF	26 SUB	42 *	58 :	74 J	90 Z	106 j	122 z
_B	11 VT	27 ESC	43 +	59 ;	75 K	91 [	107 k	123 {
_C	12 FF	28 FS	44 ,	60 <	76 L	92 \	108 l	124 |
_D	13 CR	29 GS	45 -	61 =	77 M	93]	109 m	125 }
_E	14 SO	30 RS	46 .	62 >	78 N	94 ^	110 n	126 ~
_F	15 SI	31 US	47 /	63 ?	79 O	95 _	111 o	127 (NUL)

Figure 1-6. ASCII code used in personal computers and printers

Copyright © 2015-2020 by Skylight Publishing

1.6 The Internet

You basically need three things to make computers talk to each other:

1. A wire, or a radio (Bluetooth) link between them;
2. Hardware adapters at each *host* (the term used for a computer connected to a network), switches at the network junctions, and other hardware that controls the network;
3. A common language, called a *protocol*, so that the hosts can understand each other.

Of the three, the protocol is probably the most important. Once a reliable and flexible protocol is designed, the hardware and the connections will somehow follow and fall into place. Actually a protocol standard defines hundreds, even thousands of distinct protocols that function at many different levels, or, as network designers say, layers. (Figure 1-7).

The protocols in the bottom layer deal directly with the hardware — the network technology itself and various devices that are connected to it: switches, high-speed adapters, routers, and so on. These hardware protocols are constantly evolving due to changing and competing new technologies and standards set by manufacturers and professional organizations.

Copyright © 2015-2020 by Skylight Publishing

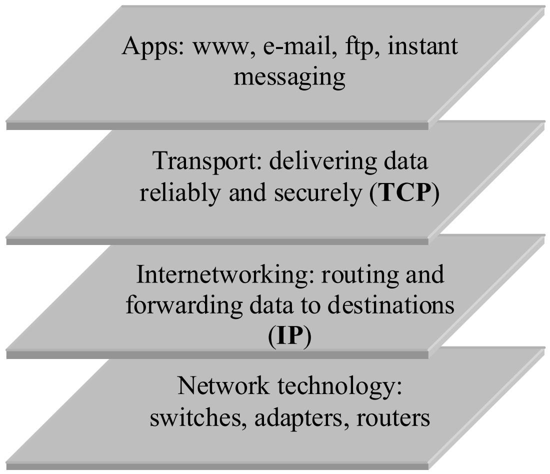
Apps: www, e-mail, ftp, instant messaging
Transport: delivering data reliably and securely (**TCP**)
Internetworking: routing and forwarding data to destinations (**IP**)
Network technology: switches, adapters, routers

Figure 1-7. TCP/IP protocols in layered network architecture

The next layer deals with routing and forwarding: how to make sure that the information from one host eventually reaches another host. These internetworking protocols must know about the general layout of the network (who is connected to whom), and be efficient and robust for reliable connections. The Internet's internetworking layer is called *IP* (the *Internet Protocol*).

The layer above internetworking, the transport protocol, is responsible for properly handling information from specific applications used on the network and for meeting the requirements of these applications: reliability, security, data compression, and so on. *TCP* (the *Transmission Control Protocol*) is the Internet's transport protocol. The *TCP/IP* combination is what really defines the Internet.

Finally, at the very top layer, there are protocols for network apps such as e-mail (using SMTP — Simple Mail Transfer Protocol), the World Wide Web (using HTTP — HyperText Transfer Protocol), file transfer (using FTP — File Transfer Protocol), instant messaging, remote terminal emulation (*telnet*), and other applications.

If we had to set a birth date for the Internet, it would probably be early September 1969 when the first computer network was tested. The network had only four nodes: University of California in Los Angeles (UCLA), Stanford Research Institute (SRI), University of California in Santa Barbara (UCSB), and the University of Utah in Salt Lake City. Here is how Dr. Leonard Kleinrock [1], a computer science professor at UCLA and one of the Internet pioneers, describes the event. Kleinrock and his group of graduate students hoped to log onto the Stanford computer and try to send it some data. They would start by typing "login" and seeing if the letters appeared on the remote terminal.

> "We set up a telephone connection between us and the guys at SRI...We typed the L and we asked on the phone, "Do you see the L?" "Yes, we see the L," came the response. "We typed the O, and we asked, "Do you see the O." "Yes, we see the O." Then we typed the G, and the system crashed! Yet a revolution had begun."

More precisely, the world's first instance of host-to-host, packet-switched data communications between networked computers had taken place.

The Internet has certainly made a lot of headway since. In 2020, over 90 percent of people in the United States and Canada and over 4.8 billion people worldwide (62%) were online [1]. The best way to explore the Internet and its history is certainly not through a book, but by getting online and browsing [1, 2]. A *browser* is an app that helps its user navigate through the Internet and presents the information that comes from the Internet back to the user. Google *Chrome*, Microsoft *Edge*, *Firefox*, and *Safari* are the four most popular browsers [1].

Copyright © 2015-2020 by Skylight Publishing

Information comes from the Internet in many different formats. The most common one is HTML (HyperText Markup Language) documents — text files with embedded formatting tags in them. You can find many HTML tutorials on the web. A common format for representing documents is PDF (Portable Document Format). Other files use standard formats for representing images, sounds, and so on. A browser and its helper modules, called *plugins*, know how to handle different formats of data and show (or play) the data to the user.

In a matter of just a few years the Internet has become a vast repository of knowledge and information (and misinformation) of all kinds and from all sources. It would be very difficult to find anything in this ocean without some guidance. *Portals* are popular web sites that arrange a large number of Internet links by category and help users navigate to the subjects they need. *Search engines* are programs based in large Internet data centers that analyze and index the contents of web pages and find web sites relevant to user queries.

As more and more individuals and businesses use computing services and resources on remote servers over the Internet, a new technology has emerged: *cloud computing*. Cloud computing may refer to different things. In its simplest form it can mean using a calendar or a calculator that runs on a remote web site or using a word processor or spreadsheet software remotely through the Internet and keeping your documents on a remote server, too, so that you can access them anywhere from any computer. Cloud computing can also mean structuring a whole enterprise's information technology needs around remote services provided by a third party vendor.

The idea is that with the Internet, computing companies become services, which, like water and electricity, are delivered to your home or business in quantity that you need. That way a business does not have to invest in expensive hardware and software; it just buys a service that meets its needs and can be easily scaled up or down according to the demand. Not everyone knows, for example, that amazon.com, besides selling books and consumer goods, sells cloud computing services through its Amazon Web Services (AWS) division [1]. Netflix, a popular movie delivery company, is an AWS's customer.

1.7 Summary

Digital electronics represents information using two states: "on" or "off," "1" or "0." Digital devices, called gates, implement simple logical operations on signals: AND, OR, NOT. All other logical and arithmetic operations can be implemented using these three simple operations.

Copyright © 2015-2020 by Skylight Publishing

The heart of a computer is a CPU (central processing unit) that can perform logical and arithmetic operations. An executable program is a sequence of CPU instructions in machine code. It must be loaded into RAM (random-access memory) before it can run. All instructions and data addresses are encoded in binary sequences of 0s and 1s. The CPU fetches instructions and data from RAM, interprets operation codes, and executes the instructions.

Computer memory (RAM) is arranged into bytes; each byte is 8 bits; each bit can hold either 1 or 0. The size of RAM is measured in kilobytes (1 KB = 2^{10} = 1024 bytes), megabytes (1 MB = 2^{20} = 1,048,576 bytes), or gigabytes (1 GB is over one billion bytes). The contents of RAM are erased when the power is turned off.

Mass storage devices have a larger memory capacity — hundreds of gigabytes or terabytes (1 TB is over one trillion bytes) — and they can hold the information permanently, but access to them is slower. Data in mass storage is arranged into files by the operating system software. The files are stored in a branching system of directories (represented as folders). The operating system also loads and runs applications, provides a GUI (graphical user interface) to users, and provides system services to programs (such as reading and writing files, supporting input devices, etc.).

All kinds of information are represented in the computer memory as sequences of 0s and 1s. Integers are represented as binary numbers. Real numbers use standard floating-point binary representation. Characters are represented as their codes in one of the standard coding systems. The most common codes are ASCII and Unicode. The latter includes thousands of characters from virtually all world alphabets.

The Internet is a network of millions of computers connected in many different ways. The Internet is based on the TCP/IP (Transmission Control Protocol / Internet Protocol), which controls information routing on the network and supports various network applications. Higher-level protocols are used in Internet applications such as e-mail, the World Wide Web, file transfer, and remote terminal emulation. A browser is a program on your computer that processes your requests for Internet connections and delivers and displays the received Internet information, web pages in particular. Portals are popular web sites that list many Internet links, arranged by category. Search engines are Internet indexing services that collect and index information from the Internet and help you find web sites relevant to your requests.

Copyright © 2015-2020 by Skylight Publishing

🕮 Exercises 🕮

The exercises for this chapter are in the book (*Java Methods: Object-Oriented Programming and Data Structures*, 3rd AP Edition, ISBN 978-0-9824775-6-4, Skylight Publishing, 2015 [1]).

Copyright © 2015-2020 by Skylight Publishing