

第一章计算机和数据的存储第一节 了解计算机很奇怪的一件事，现在的 C 语言教材，一上

来就是变量，表达式，一点也不提**计算机原理**方面的事，把 C 语言当作语言，数学来学，只是生硬的列出一些语法规则，学生们难以理解，只好死记硬背，没有把握住本质的东西，也就无法随心所欲，**其实 C 语言是一门非常好学的语言**，核心的语法规则是非常少的，而且都是和计算机息息相关，所以必须要学习好计算机的一些基础，掌握好计算机的基础知识对学习 C 语言至关重要，对将来其它语言的编程都极其有益，C 语言还有一个优点，就是写法非常简洁明了，和 c++，java，C#，javascript 等语言有着非常相似的书写方式，所以学好 C 语言，可以轻松的过渡到大量其它编程语言。讲完 C 以后，将陆续推出 java，C#，C++，javascript，ruby，python 等，不过都是在此教程的基础之上进行，所以 C 语言乃基础之基础，想要学编程，强烈建议从 C 语言开始。下面开始了解计算机。**按照冯诺依曼的理论，计算机含有 5 个部分，运算器、控制器、存储器、输入和输出设备。**现在我们通常把**运算器、控制器和在一起，称为 CPU(中央处理器)**，**存储器就是我们说的内存和硬盘**，输入设备就是鼠标，键盘等，输出设备主要是显示器、打印机、绘图仪等。其中 CPU 和内存最关键，**CPU 是执行指令的部件，内存是一些连续的存储单元，用来存储指令和数据，每个单元可以存 8 位二进制数，也称为 1 个字节，计算机里存储器都是以字节为单位的，在描述网络速度的时候通常以位为单位，大家注意区别。CPU 要执行的指令就存在内存里，CPU 按照指定顺序连续地执行内存中的指令，这就是冯诺依曼体系的基本工作原理。**什么是指令，就是一些二进制的数，**比如 10010001 是一条加法指令，10011011 是一条存储指令，CPU 遇见不同的指令就会完成不同的操作，CPU 能够执行的指令是生产厂家生产时就设计好的，比如 8086 基本指令有 70 条，那么 CPU 就只认识这 70 条指令，CPU 在不断升级，除了运行速度加快还会增加一些新的指令，CPU 的指令分三大类：传送，计算，跳转，其实我们的编程语言也就能完成这些功能。其中传送主要是管理存储器，管理数据如**

何存储，是最重要的，计算比较简单，主要是数学计算，跳转让计算机有了智能。什么是数据，数据可以是要计算的数字，也可以是一篇文章，一幅画，一首歌，这些东西必须先存到内存里，才能让 CPU 传送到显卡，声卡，我们才能听见看见。在内存里指令和数据是分开存放的，我们在编写程序时用到的内存通常是指存储数据用的数据区，在一条指令里，往往也分操作码和操作数，

操作数也是一种数据，只不过它是指令的一部分，存储在指令区，这种数据称为操作数，以后也会用到。什么是程序，程序就是由一批指令和数据组合在一起构成的二进制数据集。但如果要用二进制来写程序，那估计肯定有人骂街了，我们通常用能看懂的字母和阿拉伯数字来写程序，第一种被发明的编程语言叫汇编语言，它给每个二进制指令都给起了一个英文名字，叫做助记码，我们用助记码写起程序来要好看多了，但是 CPU 是看不懂的，我们必须把它重新翻译回二进制的才行，我们管这个叫编译，这个工作起初是手工完成的，后来有人发明了软件可以自动翻译，就出现了编译器，用汇编语言写程序比起用二进制进步了一大步，可效率还是很低，于是新的语言--高级语言出现了，据说是先有了 A 语言，又有了 B 语言，最后出现了 C 语言，由于其优秀的品质，PK 掉了同时代的众多语言，直到现在还广为流传。

有些人还有些迷惑，三大件之一的硬盘怎么还没出场，其实已经出场了，我们写好的源程序都要存储到硬盘上，编译好的可执行程序也要存储到硬盘上，内存是 RAM,掉电就内容没有了，与之对应的是 ROM,存永久保，存储器还有另一种分类方法:随机存储和线性存储，随机存储就是我们在访问的时候没有搜索的过程，只要说出地址，立即访问，线性存储有一个查找的过程，告诉哪个柱面，扇区，磁道，通过电机旋转找到后才能访问数据，CPU 只能访问随机存储器，所以内存必须是随机存储器，光盘，硬盘都属于线性存储，只能用来做外部存储器，现在比较流行的闪盘属于随机存储器，可它还是 ROM,既可以当内存，也可以当外存。硬盘上保存的有一个非常重要的程序，叫做操作系统，其实，操作系统是由众多的程

序文件和数据文件组成的，没有操作系统我们是很难与电脑沟通的，操作系统负责管理整个内存，把系统需要执行的程序加载到内存里并执行，这就是机器启动的过程，用鼠标双击一个可执行程序，操作系统就会到硬盘将其取出，加载到内存，这就是程序加载过程，我们上边说的程序就是这样被加载到内存中，才能由 CPU 来执行，除了管理内存，操作系统还监视着每个程序的执行情况，出现异常情况结束程序，如果内存不够了，操作系统会先把一部分内存里的东西搬到硬盘上，等用的时候再搬回来，硬盘上的这块地方被称为虚拟内存，使用虚拟内存来回移动数据本来就浪费时间，再加上硬盘的读取速度较慢，所以就机器就显得很慢，加大物理内存，少使用虚拟内存对计算机的性能有很大的提高。除此之外，操作系统的另一项重要功能就是硬盘上文件的管理，不同的操作系统对硬盘进行管理的方式都不一样，这个管理方式叫做文件系统，按道理说一台 Linux 的机器，挂了一块 Windows 系统下使用的硬盘，是无法对它进行读取，但是已经有人开发了 Linux 下访问 Windows 文件系统的软件，因此可以识别，并进行读取，但是反过来就不行了，除非有人开发了相应的软件才可以。总结：1. 计算机的核心部件：CPU 和内存 2. CPU 执行指令，内存中存储二进制的指令和数据 3. 可执行程序由指令和数据构成，写程序时使用高级语言，经过编译，连接后生成可执行的程序。4. 线性存储与随机存储，ROM 和 RAM,硬盘,操作系统

第二节数据的存储

计算机最擅长的就是存储，存储也是计算机的灵魂，所以这本教程是紧紧围绕着存储来展开的，理解并掌握了存储，你基本上就懂了如何编写程序，我们刚才讲到了内存里可以存储数据，那么我们都存储什么样的数据？又是如何来存储的呢？

1. 数的存储

这里说的数就是我们计算中用到的表示大小的数，我们习惯以十进制表示，计算机只能存储二进制数，我们把它转换成相应的二进制数来存储就可以了。例如 254 11111110 通常写做 FE 5 00000101 051)

进制转换：1.n 进制à十进制方法：

下面这个等式大家估计都能看的懂。

$$3679 = 3 \times 10^3 + 6 \times 10^2 + 7 \times 10^1 + 9 \times 10^0$$

这是十进制数的计算方法，对于 n 进制的数，把 10

换成 n 就可以了，所以记住这个等式，任何进制转 10 进制都会了。如：10011001 是二进制数转化成 10 进制就是 $1 \times 2^7 + 0 + 0 + 1 \times 2^4 + 1 \times 2^3 + 0 + 0 + 1 \times 2^0 = 128 + 16 + 8 + 1 = 153$ 。十进制转 n 进制方法：像刚才一样，想想 10 进制数怎么找出每一位的值，3679 / 10 余数 9 商 367 最低位 9367 / 10 余数 7 商 36 上一位 736 / 10 余数 6 商 3 上一位 63 / 10 余数 3 商 0 上一位 3 用要转换的数除以 10，余数就是的最低位数，然后用商接着除 10，得到余数就是上一位，一直除下去，直到商为 0 结束。容易出错的地方：用 260 除以 16 的时候另外列个算式，两个算式很像，不小心容易搞混。

3. 十六进制，二进制相互转换方法：

十六进制 -> 二进制 1 位变 4 位，不够 4 位，前边补零如：104 -> 0001 0000 0100 二进制 -> 十六进制 4 位变 1 位，如：101 1100 -> 5 c 注意：电脑里用的都是二进制，但我们总是看到很多十六进制的数，为什么呢，原因：二进制，十六进制转换方便，口算即可，十六进制书写起来，看起来都要比二进制方便很多，因此，在需要使用二进制的地方通常用十六进制代替。注意：任何存储都以字节为单位，每个字节能存 2 位十六进制数。所以我们经常看到是两位 16 进制数。

2) 多字节存储

一个字节只能存储 8 位 2 进制数，最大只能存 255，那么要存更大的数怎么办，我们可以使用多个单元来存一个数，连起来就能构成一个大数，我们通常把低位存在最右边，然后依次存储高位，内存是连续的，只有前后，没有左右，一个数由多个字节组成的话，一定要明确高位低位的顺序，低位在前，高位在后。这个很关键，要记住。比如 258 写成 16 进制就是 01 02，每个字节可以存两位 16 进制，现在需要两个字节，存储的时候 02 在前，01 在后，更多字节，依次类推。

3) 符号数存储

数字有正负之分，当存储一个有符号的数时，通常用最高位代表符号位，1 代表负数，0 代表正数。计算机中所有带符号数都以补码形式存储。补码定义：正数的补码是其本身，负数的补码符号位不变，其余位取反加 1 如果大存储空间存一个小点的数，高位的字节也不能空着，正数我们可以在高位补 0，不影响大小，负数我们必须在高位补 1，才不影响大小。

2. 字

符的存储要存储源程序,存储一篇文章,存储一串字母和数字,这都是字符的存储,和数字不一样,它不表示大小,只表是一个字母或数字,我们通常采用编码的方式来存储,最常用的编码要属 Asc 码了,每个字母和数字都有一个数字对应,都可以变成二进制进行储存,但要存中文,Asc 码就不行了,我们国家有自己的编码标准 Gb2312,哈哈,汉字也可以存了,不过存一个汉字要用两个字节,当然读出来还要按照 Gb2312 的标准进行解码,再查找字库进行显示,有些软件会自动根据文件的开始几个字节或某一指定内容来自动选择编码标准,将来要编写国际化的程序,编码一定要考虑好。

3. 图片的存储

图片是由点组成的,一幅图片大小是 200×300 ,那么他就有 60000 (200×300) 个点,每个点都有自己的颜色,这些颜色组合在一起就构成了图片,如果每个点颜色我们用一字节来存,那么最多只能有 256 种颜色,这样存储的图片颜色会看起来有些失真,如果用两个字节表示颜色,能表示的颜色就有 65536 种,基本上超过了人眼的分辨能力,看上去没有失真,如果我们要更好的表现色彩,会用 3 个字节 24 位,甚至 4 个字节 32 位来存储颜色,图片更逼真,所需要的空间就越大。*视频将图片连续播放出来就会给人动的效果,如果没有压缩,视频会占用大量的存储空间,视频基本上都是压缩存放的。

4. 声音的存储

声音就是一种波形,我们把波形分成很小的段,每段的幅值平均一下用一个二进制数表示,存储起来,这叫做采样,就相当于把声音存储起来了,通常每秒钟要分 44.1k 段才能达到不失真的效果。采样频率低的话,声音质量会下降。

总结:

1. 数的存储二进制,十六进制,数制之间的转换,多字节存储,带符号数存储.
2. 字符的存储通过编码对字符进行数字化后存储