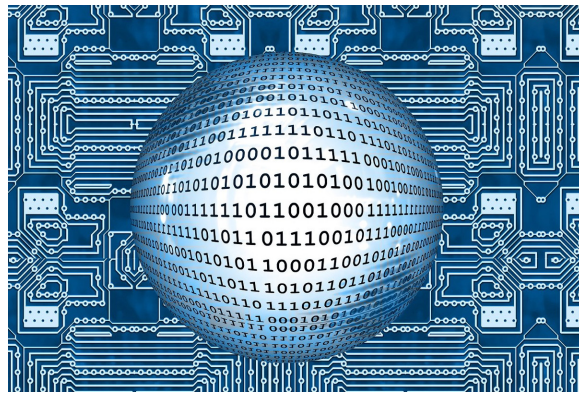


# 零與壹的世界

王士弘

July 11, 2019

我們大家活在一個信息技術（IT）主導的世界。IT 幾乎無所不在，以許許多多的方式充實和便利我們的生活。IT 依賴數字技術。“數字”一詞來自數字計算機，那是信息技術的基礎。數字計算機的特點是只用 0 和 1 來代表和存儲信息，絕無例外。計算機內所有數據和所有程序都以 0 和 1 來編碼。換句話說，計算機只使用 0 和 1 來完成所有事情，多麼奇妙和不可思議！



電腦裡面是一個 0 和 1 的世界。但這怎麼行呢？0 和 1 如何播放音樂或視頻？聽懂語言？送人上月球？自動駕駛汽車？這些聽起來都很複雜。但基本上真的也很簡單。

## 電腦硬件

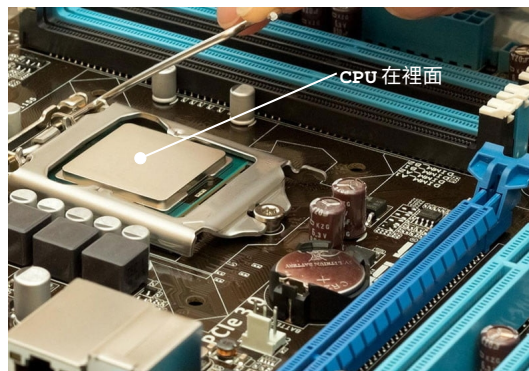
數字計算機能快速處理信號。每個信號是用電流或電壓的有無來代表。這樣的信號是計算機內最小的數據單位，叫作比特（bit）也叫位。

一個位可以代表任何正反兩面：開/關，是/否，陰/陽，上/下，左/右，真/假，還有 1/0。當然還有很多其他可能性。但通常將它們用二進制數 1 和 0 來表示。八個位組成一個字節（byte）。四個或八個字節組成一個字（word）。位，字節，字，這些是計算機中代表信息的單元。

中央處理器（CPU）是計算機的大腦，是處理信息的地方。指甲大小的矽芯片上的現代 CPU 可能包含幾十億個晶體管（transistor）。每個晶體管非常小，

約 70 個矽原子，肉眼肯定是看不見的。這些晶體管構成 CPU 中的部件以便用來存儲和處理電子信號。

計算機的主存儲器或主記憶（main memory）是用來存放 CPU 要立即處理的信息。CPU 快速從主記憶取得數據，處理以後將結果立即存回主記憶中。



最簡單的主記憶類型是 DRAM（動態隨機存取記憶）。一個 DRAM 位可以由單個晶體管和單個電容器形成，使用電容器的充電和放電狀態來表示 1 和 0。CPU 的指令集中的每條指令由它的數字操作碼（opcode）來代表。指令處理信息以字為單位。先進 CPU 速度極快，每秒執行超過 1000 億條指令。

DRAM 內存儲的信息具有揮發性（volatile）。如果電源關閉，DRAM 會失去其內容。這與用於長期數據存儲的硬盤，閃存盤和光盤形成對比。這就是為什麼每次電腦重啟時，它需要較長的時間將操作系統從磁盤帶回主記憶。主記憶通常是 4 到 16GB（1GB=10<sup>9</sup> 字節），而硬盤要大得多，接近幾個 TB（1TB=10<sup>12</sup> 字節）。

這就是數字計算機硬件的本質，它決定了計算機內部信息的表示和處理必須以 0 和 1 的形式。多麼有趣，又有多難呢？

## 整數

沉浸在電腦的 0 和 1 世界之前，我們先來看看我們人類自己的世界吧。在拼音語文中，單詞和數字是由一組字母組成。例如英語大小寫 A-Z 和數字（0 到 9）。不同語文使用不同的字母。在計算機內部，只有兩個字母，即符號 0 和 1。在這個陌生的世界裡，一切都必須以 0 和 1 表示。重要的是要意識到 0 和 1 只是方便的符號來表示比特的兩種狀態。在這裡我們必須忘記它們日常作為數字的含義。如果你願意何不把 1 視為“有”而 0 為“無”。現在，訣竅是有系統的使用一串比特來表示其他信息。

首先我們來看看如何使用比特來表示整數：零，一，二，三等等。如果用三個比特，我們可以代表多少個整數？以下是所有的排列模式，共 8 個：

0 0 0, 0 0 1, 0 1 0, 0 1 1, 1 0 0, 1 0 1, 1 1 0, 1 1 1

在零與壹的世界裡，這些是所有可能的 3 個字母的單字。它們可以用來代表 8 個整數，0 到 7。這方法不是任意的，而正是二進制數表示法。我們日常使用的數字是十進制 (decimal)，每進一位加一個十的次方。例如，

$$\text{十進制 } 209 = 2 \times 10^2 + 0 \times 10 + 9 \text{ (二百零九)}$$

同樣，二進制 (binary)，每進一位加一個二的次方。例如，

$$\text{二進制 } 101 = 1 \times 2^2 + 0 \times 2 + 1 \text{ (五)}$$

越來越大的數字需要更多的比特來表示。使用 32 位，我們可以代表  $2^{32}$  不同的數字，足以覆蓋正負  $2^{31} - 1$  範圍內的整數。使用 64 位，我們可以覆蓋遠遠更多。

二進制數的算術法則與十進制完全相似。在電腦裡易於執行。

## 文字

數字是最基本的，但計算機還需要處理其他類型的資訊，其中可能最重要的就是文字，包含**字符** (character) 和**文本** (text)。同樣我們可以用比特模式來代表每個字符，事實上每個字符是用它獨特的二進制數來代表。例如，美國信息交換標準代碼 (US-ASCII) 使用一個字節中的 7 位 (覆蓋 0 到 127) 來表示典型鍵盤上的 128 個字符：0-9，A-Z，a-z，標點符號，符號和製表符。例如，

'0'	00110000 (48)	'9'	00111001 (57)
'A'	01000001 (65)	'Z'	01011010 (90)
'a'	01100001 (97)	'z'	01111010 (122)

請注意字符“A”的位模式也可以代表整數 65。還要注意，與直覺相反，字符“9”的位模式與數字 9 的位模式不同。因此，“9”作為一個字符基本上是不同的於一個數字的。

**統一碼** (Unicode) 是一個字符編碼的國際標準。它涵蓋來自世界上大多數書寫系統，包括 100 種語言/文字，超過 110,000 個字符。統一碼聯盟是一個國際合作組織，負責發布並更新統一碼標準。

**文本** (textual) 文件內容由字符組成，也就可用一連串的字符編碼來代表。運用統一碼我們能在一個文件中混合使用幾乎所有已知的語言文字。這在通過互聯網 (Internet) 日益緊密的世界中尤其有利。UTF-8 是一種特別有效的統一碼形式，大多數網頁都是用它編寫的。

## 理解斷章取義

相同的符號，單詞或短語在不同的地方有不同的含義，取決於它使用的場合及上下文。例如‘和’，總和，和平，和麵，0 和 1，零和思維。

作為符號，位模式也不例外。你可能已經注意到同樣的位模式能代表一個整數或者一個字符。例如，位模式 01000001 代表 65 或者字符 ‘A’。問題是我們怎麼知道它到底代表什麼。答案是看它的意義背景 (context)。也就是說**相同的位模式可以跟據不同背景來代表不同的意義**。

在編碼裡面對於任何位模式我們必須提供一個背景來認定它是數字，字符還是別的。意義背景可以明確指定，也可以從模式使用的環境(上下文)推斷出來。例如，編碼表達式  $x + 5$  中的變量  $x$  是數字而不是別的。因此在計算機程序中，每個變量的**數據類型** (data type) 必須是明確的。

斷章取義會有嚴重後果。在 1999 年，美國宇航局 (NASA) 的火星氣候衛星發射後在火星大氣層中燒毀，因為工程師們將英制數據誤認為公制。就是在日常生活中，我們也要牢記避免和防範斷章取義。例如，在電子郵件中應一律避免使用諸如“今天”，“昨天”之類的話。

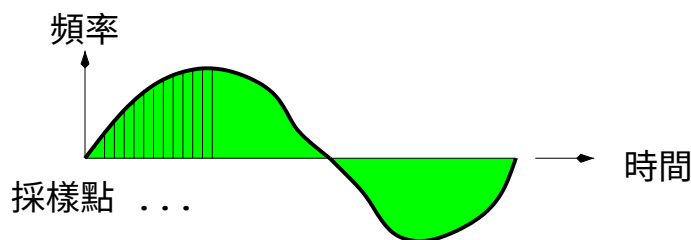
對計算和計算機的了解還能帶來許多生活中思維的導引和提示，那就是能啟發**計算思維** (Computational Thinking or CT)。世界公認計算思維是現代思考和教育的重要工具，能讓人避免風險，提高效率。作者希望能在後續文章中詳細闡釋計算思維。

## 千里眼順風耳

數字世界讓我們擁有千里眼順風耳，這要歸功於快速的音頻和視頻。雖然數字和字符比較容易用位模式表示，但是更複雜的信息如音樂，圖片，或者視頻怎麼辦呢？

首先看一下音樂。聲音是以音波形式通過空氣傳播的。音階從低到高是連續變化而有無限多的值。

過去，模擬 (analog) 電腦處理音波自麥克風生成的電子波。這種模擬信號難以存儲，傳輸，或複製而有失真問題。數字計算機無法直接存儲和處理音波一類的連續值信息，必須先將該值**數字化** (digitize)，變成一系列數字，然後用 0 和 1 來表示。數字化的方法也不難，一個連續值，例如聲波，可以通過多個採樣點的值來變成一系列數字，採樣點越多，結果越精確。









如此，一個連續的波可由各個採樣點處的一系列的數值來表示。這些數就可以用 0 和 1 來表示。數字化後的聲波也沒有失真問題，更可以通過巧妙的壓縮算法，如 MP3，來進一步減少數據量和提高傳輸速度。

## 有圖為證

能夠表示顏色，圖像或照片就不難了。一個圖像基本上是平面上不同的顏色組成的。因此以數字方式表示顏色是第一要件。在屏幕上每一個亮點是由不

同比例的紅，綠，藍形成的。這就是廣泛使用的 RGB 系統，它用一個三重數  $(r, g, b)$  來代表一種顏色。其中每個數  $r, g$  或  $b$  的範圍為 0 到 255。例如  $(255,0,0)$  為全紅色， $(0,255,0)$  為全綠色， $(0,0,255)$  為全藍色， $(0,0,0)$  為純黑色， $(255,255,255)$  為純白色。這樣，每個顏色用三個字節來代表，總共可以代表  $256^3$  種不同的顏色。

紅		$(255,0,0)$	青		$(0,255,255)$
綠		$(0,255,0)$	黃		$(255,255,0)$
藍		$(0,0,255)$	品紅		$(255,0,255)$

CYMK (青，黃，品紅，黑) 系統是完全類似地。RGB 用於屏幕顯示，而 CMYK 用於打印。

一張圖片可以通過垂直和水平線將它劃成許多小方格。每格稱為一個**像素** (pixel)，有自己的顏色。**光柵圖形** (raster graphics) 是用像素來拼成圖片。像素越多，圖片就越清晰。**矢量圖形** (vector graphics) 是用  $x-y$  坐標，線和其他幾何元素來畫出圖片。它們有各自的優點和缺點。光柵圖形常用於相片。矢量圖形更適合用於漫畫，徽標和圖標。

依時間順序結合一序列的圖像和聲音就能夠代表視頻。可以想像，高/超清視頻數據量大，需要強大的計算力才能在屏幕上快速呈現。通常，電腦會添加圖形硬件，例如圖形處理單元 (GPU) 和圖形卡，來大大加速圖像的顯示。此外，許多高效的圖像/視頻數據壓縮算法也能夠減小視頻體量並提高傳輸速度，同時盡量保持圖像品質。

## 程序也是數據

電腦的強大功能來自各種程序，包括操作系統和應用程序。在電腦里程序是如何表示的呢？

機器語言程序是由 0 和 1 表示的，CPU 可以直接運行，但是很難用來編程。高級編程語言，諸如 C++ 和 Java，允許使用類似英語詞組和數學表達式的方式來編寫程序。這樣程序文件就是文本文件，讓程序設計更方便，更不容易出錯。可是執行前這種程序必須通過另一個叫編譯器或解釋器的軟件來處理，並翻譯成機器語言。

電腦和別的機器的根本區別在於電腦是**通用機器**。因為當它運行另一個程序時，它就變成了另一個機器。硬件不變只變軟件就能執行新任務。計算機能夠完成任何可編程的任務，其他機器能嗎？

因為**程序就是數據**所以程序可以被程序處理和操作。編譯器就是一個例子。更進一步，一個程序甚至能夠自己改造自己的編程而變得更有效，就像人給自己做大腦手術。會學習，能訓練的程序是例子。

## 到頭來

電腦的本質帶來了一個充滿了位，字節，和字的神奇世界。那是一個語言只有兩個字母的世界，一個所有信息都由 **0** 和 **1** 代表的世界，一個所有指令都由 **0** 和 **1** 組成的世界，一個飛快運轉的世界。

僅使用 **0** 和 **1** 進行處理也許看起來很愚蠢，但這種簡單性正是縮小尺寸，降低成本和提高速度的基礎。現在電腦功能雖然強大，但是好戲還在後面呢！

附筆: 如果只有兩個符號的世界如此美妙，為什麼不進一步走向一個符號的世界呢？那不是更好嗎？考慮一下，然後看看答案。

**不可能只有一個符號。有有必有無，有無必有有。因此，零與壹的世界就是了。**