

CHAPTER 1

Linux 那些事兒之我是 USB Core



- 1 · 引子
- 2 · 它從哪裡來
- 3 · 挑選
- 4 · 漫漫辛酸路
- 5 · 我型我秀
- 6 · 我是一棵樹
- 7 · 我是誰
- 8 · 好戲開始了
- 9 · 不一樣的 **Core**
- 10 · 從這裡開始
- 11 · 面紗
- 12 · 模型，又見模型
- 13 · 繁華落盡
- 14 · 介面是裝置的介面
- 15 · 設定是介面的設定
- 16 · 端點
- 17 · 裝置
- 18 · 設定
- 19 · 向左走，向右走
- 20 · 裝置的生命線（一）
- 21 · 裝置的生命線（二）
- 22 · 裝置的生命線（三）
- 23 · 裝置的生命線（四）
- 24 · 裝置的生命線（五）
- 25 · 裝置的生命線（六）
- 26 · 裝置的生命線（七）
- 27 · 裝置的生命線（八）
- 28 · 裝置的生命線（九）
- 29 · 裝置的生命線（十）
- 30 · 裝置的生命線（十一）
- 31 · 驅動的生命線（一）
- 32 · 驅動的生命線（二）
- 33 · 驅動的生命線（三）
- 34 · 驅動的生命線（四）
- 35 · 字串描述符號
- 36 · 介面的驅動
- 37 · 還是那個 **match**
- 38 · 結束語

1 · 引子

還是要說在前面，在這裡耗費青春寫 USB，並不是因為喜歡它，雖然每天都必須和它相依為伴，不離不棄，不過那可是絲毫沒有辦法的事情，非我所願。

2 · 它從哪裡來

不過，與 PCI、AGP 屬於 Intel 單獨提出的硬體標準不同，Compaq、IBM、Microsoft 等也一起參與了 USB 這個遊戲。他們一起於 1994 年 11 月提出了 USB，並於 1995 年 11 月制定了 0.9 版本，1996 年制定了 1.0 版本。因為缺乏操作平台的良好支援和大量支援它的產品，這些標準都成了空談。1998 年，USB 1.1 的出現才告成功。

為什麼要開發 USB ？

在 USB 出現以前，電腦的介面處於“春秋戰國時代”，序列埠、平行埠等多方割據，鍵盤、滑鼠、MODEM、印表機、掃描器等都要連接在這些不同種類別的介面上，一個介面只能連接一個裝置。不過電腦不可能有那麼多介面，所以擴充能力不足，而且速度也確實很有限。還有關鍵的一點是，熱抽換對它們來說也是比較危險的操作。

USB 正是為了解決速度、擴充能力、易用性等問題應景而生的。

3 · 挑選

USB 最初的設計目標就是替代串列、並行等各種低速匯流排，以一種單一型態的匯流排連接各種不同的裝置。它現在幾乎可支援所有連接到 PC 上的裝置，1999 年提出的 USB 2.0 理論上可達到 480 MB/s 的速度，2008 年公佈的 USB 3.0 標準更是提供了十倍於 USB 2.0 的傳送速率。

因此，USB 與序列埠、平行埠等的這場“挑戰”從一開始就是不平等的，這樣的開始也註定了以什麼樣的結果結束，只能說命運選擇了 USB。我們很多人都說命運掌握在自己手裡，但是從 USB 填滿“挑戰”的一生中可知道，只有變得比別人更強，命運才能掌握在自己手裡。

有了 USB 在這場挑戰中的大獲全勝，才有了 USB 鍵盤、USB 滑鼠、USB 印表機、USB 攝影機、USB 掃描器、USB 喇叭等。至於將來，“挑戰自己的，讓別人去說吧！”USB 如是說。

4 · 漫漫辛酸路

USB 的一生從 USB 1.0、USB 1.1、USB 2.0 到 USB 3.0，漫漫辛酸路，一把辛酸淚。

USB 2.0 的高速模式（High-Speed）最高已經達到了 480 MB/s，也就是說，以這個速度，你將自己從網上下載的短片備份到自己的行動硬碟上的時間長約為一秒鐘。而 USB 3.0 的 Super-Speed 模式比這個速度加強了幾乎 10 倍，達到了 4.8GB/s。

USB 走過的這段辛酸路，對我們來說，最直觀的結果也就是傳送速率加強了，過程很艱辛，結果很簡單。

USB 的各個版本都是相容的。每個 USB 2.0 控制器帶有 3 個晶片，根據裝置的識別方式，將訊號發送到正確的控制晶片。我們可將 USB 1.1 裝置連接到 USB 2.0 的控制器上使用，不過它只能達到 USB 1.1 的速度。同時也可將 USB 2.0 的裝置連接到 USB 1.1 的控制器上，不過不能指望它能以 USB 2.0 的速度執行。

顯然，Linux 對 USB 1.1 和 USB 2.0 都是支援的，並搶在 Windows 前，在 2.6.31 核心中率先對 USB 3.0 進行了支援。

5 · 我型我秀

USB 既然能一路“挑戰”走過來，也算是一個挺能“秀”的角色了，不然也不會有那麼多的擁護者。

USB 為所有的 USB 外接裝置都提供了單一、標準的連接型態，這就簡化了外接裝置的設計，也讓我們不用再去想哪個裝置對應哪個插槽的問題，就像種蘿蔔，一個蘿蔔一個坑，但是哪個蘿蔔種到哪個坑裡是不用我們關心的。

USB 支援熱抽換，而其他的例如 SCSI 裝置等只有在關掉主機的前提下才能增加或移走週邊裝置。所以說，USB 的一生不僅僅是“挑戰”的一生，也是豐富多彩的一生，可不用關機就能更換不同類別的外接裝置。

USB 在裝置供電方面提供了靈活性。USB 裝置可透過 USB 電纜供電，不然行動硬碟、iPod 等常備外接裝置也用不了了。相對應，有的 USB 裝置也可使用普通的電源供電。

USB 能夠支援從每秒幾十千位元組到幾十百萬位元組的傳輸速率，來適應不同類別的外接裝置。它可支援多個裝置同時操作，也支援多功能的裝置。多功能的裝置當然指的就是一個裝置同時有多個功能，例如 USB 喇叭。這透過在一個裝置中包含多個介面來支援，一個介面支援一個功能。

USB 可支援多達 127 個裝置。

USB 可確保固定的頻寬，這個對視訊 / 音訊裝置是利多。

6 · 我是一棵樹

如圖 1.6.1 所示，USB 子系統的拓撲也是一棵樹，它並不以匯流排的方式來佈署。

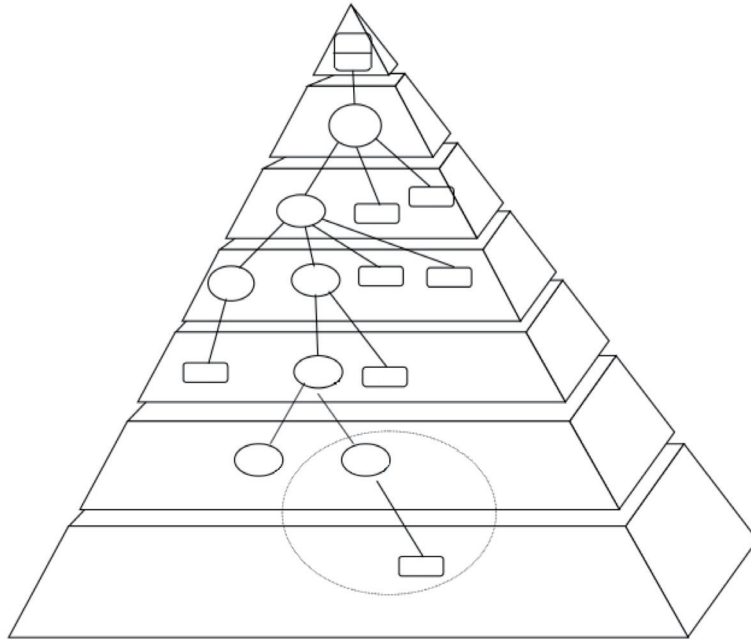


圖 1.6.1 USB 子系統的樹狀結構

USB 的大樹主要包括 USB 連接、USB Host Controller (USB 主機控制器) 和 USB 裝置三個部分。而 USB 裝置還包括了 Hub 和功能裝置 (也就是圖 1.6.1 中的 Func)。

什麼是 USB 主機控制器？控制器，顧名思義，用於控制。控制什麼？控制所有的 USB 裝置的通訊。通常，電腦的 CPU 並不是直接和 USB 裝置進行處理，而是和控制器進行處理。它要對裝置做什麼，它會告訴控制器，而非直接把指令發給裝置。然後控制器再去負責處理這件事情，它會去指揮裝置執行指令，而 CPU 就不用管剩下的事情。控制器替它去完成剩下的事情，事情辦完了再通知 CPU。否則，讓 CPU 去盯著每一個裝置做每一件事情，那是不現實的。

那麼 Hub 是什麼？在大學裡，有的宿舍裡網路介面有限，所以會有網路介面不夠用的情況出現，於是有人會使用 Hub，讓多個人共用一個網路介面，這是乙太網上的 Hub。而 USB 的世界裡同樣有 Hub，其實原理是一樣的，任何支援 USB 的電腦不會只允許你只能一個時刻使用一個 USB 裝置，例如，你插入了隨身碟，同樣還可插入 USB 鍵盤，然後再插一個 USB 滑鼠，因為你會發現你的電腦裡並不只是一個 USB 介面。這些介面實際上就是所謂的 Hub 埠。

而現實中經常是一個 USB 控制器和一個 Hub 綁定在一起，專業一點稱為“整合”，而這個 Hub 也被稱做 Root Hub。換而言之，和 USB 控制器綁定在一起的 Hub 就是系統中最根本的 Hub，其他的 Hub 可連接到它這裡，然後可延伸出去，外接別的裝置，當然也可不用別的 Hub，讓 USB 裝置直接接到 Root Hub 上。

而 USB 連接指的就是連接 USB 裝置和主機（或 Hub）的四線電纜。電纜中包括 VBUS（電源線）、GND（地線）和兩根訊號線。USB 系統就是透過 VBUS 和 GND 向 USB 裝置提供電源的。主機對連接的 USB 裝置提供電源供其使用，而每個 USB 裝置也能夠有自己的電源，如圖 1.6.2 所示。

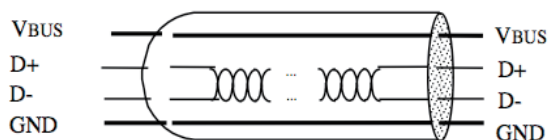


圖 1.6.2 USB 四線電纜

現在，如圖 1.6.3 所示的 USB 大樹裡只有 Compound Device 還沒有說。那麼，Compound Device 又是什麼樣的裝置？其實，在 USB 的世界裡，不僅僅有 Compound Device，還有 Composite Device，簡單的中文名字已經無法形象地表達它們的區別。正如圖 1.6.3 所示，Compound Device 是將 Hub 和連在 Hub 上的裝置封裝在一起所組成的裝置。而 Composite Device 則是包含彼此獨立的多個介面的裝置。從主機的角度看，一個 Compound Device 和單獨的一個 Hub，然後連接了多個 USB 裝置是一樣的，它裡面包含的 Hub 和各個裝置都會有自己獨立的位址，而一個 Composite Device 裡不管有多少介面，它都只有一個位址。

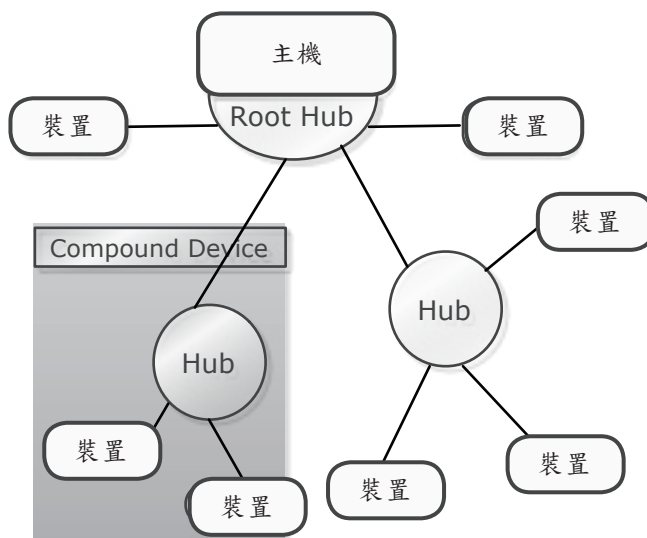


圖 1.6.3 Compound Device