如何設定與維持準確的電腦時鐘

作者: Ron Bean, *rbean@execpc.com*
譯者: Da-Wei Chiang, *dawei@sinica.edu.tw*
v, December 1996 翻譯日期: 2-16 July 1998

如何設定與維持準確的電腦時鐘.

目錄

1 介紹

個人電腦(甚至於昂貴的工作站)主機板上所使用的 real-time-clock 晶片不夠準確是眾所皆知的事情. 然而 Linux 作業系統提供了一個簡單的方法來校正這個不準確的電腦時鐘, 使得電腦時鐘就算是不倚賴外面的校時源也可以達到 *相當* 準確的地步. 但是似乎大多數人都還不知道有這個方法, 其原因如下:

1. 在一般 "如何設定 linux 作業系統" 的文件中並未述及這個方法, 而且在安裝作業系統時很難自動地完成設定 (但是理論上並不是不可能的, 如果你有一台數據機的話).
2. 若你使用"`man clock`"來查看設定時鐘的說明時, 你可能會取得非預期的 `clock(3)` 說明內容. (試著使用"`man 8 clock`").
3. 似乎大多數人並不在意現在的正確時刻是多少.
4. 那些少數在意的人往往會採用 `louie.udel.edu` 所提供的 `xntpd` 程式集以便與外面的校時源, 例如網路校時伺服器或無線電時鐘信號, 取得同步.

如果各位對本文所講述的低階的時鐘設定方法有著極高度的興趣, 我強烈建議各位花些時間來瀏覽網頁 `<http://www.eecis.udel.edu/ntp/>`, 這個網頁有著各類你所感興趣的內容,例如有關於 `xntpd` 的完整資訊以及 NIST 與 USNO 等站台的超連結. (我在本文後面也對 `xntpd` 作了些許的簡介.)

注意

如果你在同一台機器上跑多個作業系統, 你應該只讓其中一個來重新設定 CMOS 時鐘, 如此它們纔不會相互影響而導致混亂. 假使你平常在同一台機器上跑 Linux 和 Windows 二種作業系統, 而當你要尋找一些共享的校時程式便時可以忽略 Windows 作業系統的部分 (參考前面所說網頁的超連結).

2 使用“clock”程式

所有你須要知道的內容都在 `clock(8)` 的說明中, 而本文將帶領各位完成整個設定程序.

注意

要執行“clock”或是其它會影響到系統時間和 CMOS 時鐘的程式, 你必須擁有 **root** 的權限.

2.1 檢查你的作業系統

檢查你作業系統的開機執行檔中是否有類似“clock -a”或 “clock -ua”的命令. 這個命令可能放在 `/etc/rc.local` ,或 `/etc/rc.d/rc.sysinit` , 或其它類似的地方, 這要看你安裝的是那一套 Linux 作業系統.

如果找到的命令是“clock -s” 或“clock -us” , 請將“s” 更改成“a” , 接著檢查看看是否有 `/etc/adjtime` 這個檔案, 該檔案的內容只有一行看起來像是這樣:

```
0.000000 842214901 0.000000
```

這些數字依序是校正因子 (每天偏差的秒數), 上一次校正時鐘的時間 (由 Jan 1, 1970 開始起算的秒數), 上一次校正所扣除的秒數. 如果你找不到這個檔案, 請以 **root** 的身份產生此檔案, 其內容只有一行看起來像是這樣 (全部是零):

```
0.0 0 0.0
```

然後透過 shell 手動執行 “clock -a” 或 “clock -ua” 以便更新第二個數字 (如果你的時鐘被設定成協調國際時間而不是本地時間 你得使用 “u”這個參數).

2.2 量測時鐘的偏差率

首先你得知道現在到底是幾點鐘 `:-)`. 你機器的本地時間可能是不準確的. 我最喜歡的方法是撥電話到 WWV 的 (303)499-7111 電話報時台 (這是個付費電話). 如果你能夠與網路校時伺服器連上線, 你也可以使用 `xntpd` 程式集所提供的 `ntpdate` 程式來校時 (使用 `-b` 選項以避免系統弄亂 CMOS 時鐘). 或者使用 “date -s hh:mm:ss” 以手動方式設定系統時鐘, 不論那一種方法最後都得使用 “clock -w” 將系統時間寫入 CMOS 時鐘. 你必須記得最後一次設定時鐘的日期, 所以你得將該日期寫下來放在不會遺失的地方 . 如果你使用的是 `ntpdate` 程式, 執行 “date +%s” 然後寫下自從 Jan 1,1970 算起到該程式執行的秒數.

然後等待幾天或幾個星期後之後再回來看看時鐘到底偏差了多少時間. 如果你以手動的方式設定時鐘, 建議至少得等待二個星期之後, 再來計算時鐘偏差率將可得到趨近 .1 sec/day 的結果. 幾個月之後偏差率應該會趨近 .01 sec/day (有些人宣稱應該更準確, 但是我個人持保留的態度). 如果你使用 `ntpdate` 程式來設定時鐘你可能不必等待太久的時間, 然而不論使用那種方式之後都得做些微調的工作.

你可以使用 cron 在一定的時間周期內執行“`clock -a`”以便系統時間能夠與 (校正過的) CMOS時間維持一致性. 其實在你每次開機的時候都會從開機執行檔來執行這個命令, 然而只要常常做上面這個動作就可以達到你準確機器時間的目的.

注意如果系統時間的修正一次超過一秒或是時間往回修正, 有些系統的 clock 程式可能會無法執行. 如果你有這方面的問題, 則可以使用 `xntpd` 或 `ntpdate` 程式 來逐步地修正時間.

2.3 實例

2.3.1 設定時間

簽入系統並成爲 **root**. 撥電話至 (303)499-7111 (語音), 來聆聽報時的內容. 然後使用鍵盤打出下面幾個字:

```
date -s hh:mm:ss
```

一直等到嗶的一聲纔按下 enter 鍵. (如果在此處使用“`ntpdate`”程式, 則你可以省略撥電話的動作) 這個步驟在設定“系統時間”. 接著執行:

```
clock -w
```

這個步驟將系統時間寫回 CMOS 時鐘讓二者時間一致. 接著執行:

```
date +%j
```

(如果在前面使用“`ntpdate`”程式你得執行“`date +%s`”) 將其執行的結果記錄下來.

2.3.2 重新設定時間與檢查偏差率

找出你上次寫下來日期. 簽入系統並成爲 **root**. 然後執行:

```
clock -a
```

這個步驟將 CMOS 時間寫入系統時鐘讓二者時間一致. 撥電話至 (303)499-7111 (語音), 來聆聽報時的內容. 接著使用鍵盤打出下面幾個字:

```
date
```

一直等到嗶的一聲纔按下 enter 鍵, 但是當你等待的時候, 寫下報時的內容, 並且還不要掛上電話. 這個動作是讓你知道目前眞正的時間是幾點, 而此時相對你的機器上的時間是幾點. 現在使用鍵盤打出下面幾個字:

```
date -s hh:mm:00
```

其中分針部分所設定的數字就是目前報時內容 *之後的* 下一分鐘, 一直等到嗶的一聲纔按下 enter 鍵 (現在可以掛上電話了). 而 hh 的部分則使用機器本地的時針數字即可. 這個步驟在設定“系統時間”. 然後執行:

```
clock -w
```

將新的 (校正過的) 系統時間寫回 CMOS 時鐘. 然後執行:

```
date +%j
```

(或是執行 "date +%s")

現在手上有三個數字 (二個校正日期及一個正確時間) 於是你可以開始計算偏差率了.

2.3.3 計算校正因子

當你在某一分鐘執行 "date" 程式時, 你機器的本地時間是慢還是快呢? 如果是快了, 你必須將之減去一些秒數以符合正確時間, 所以你應該寫下一個負的偏差數值. 如果是慢了,你必須將之加上一些秒數以符合正確時間, 所以你應該寫下一個正的偏差數值.

現在將二個校正日期相減. 如果之前你使用的是 "date +%j", 則數值所代表意義爲一年的第幾天 (1-365, 或 1-366 如果是閏年). 如果自從你上一次設定時鐘到現在經過1 月1 日你還須要將第二個校正日期加 365 (或 366) 如果之前你使用的是 "date +%s" 則數值的單位爲秒數, 你必須將之除以 86400 以便取得日數.

如果你的檔案 /etc/adjtime 中已經有校正因子, 你必須將先前已校正過的秒數考慮進來, 如果你校正過度, 這個校正因子的正負號將與你量測到偏差數值相反; 如果你校正不足則二者正負號相同. 將舊的校正因子乘以日數, 然後加上新獲得的偏差秒數 (附註– 如果二者正負號相同, 你將取得一個較大的偏差數值, 如果二者正負號相反, 你將取得一個較小的偏差數值).

接著將這個總偏差秒數除以日數以便取得新的校正因子, 然後將檔案 /etc/adjtime 中舊的校正因子代換成新的. 最後寫下新的校正日期 (以秒數或日數爲單位) 以便下次使用.

這裡是我的 /etc/adjtime 檔案內容:

```
-9.600000 845082716 -0.250655
```

(注意每天 9.6 秒則一個月將近有 5 分鐘的偏差!)

3 xntpd 簡介

你的系統實際上有二個時鐘– 一個是當系統關機繼續使用電池電力維持計時功能的 "real time clock" (也就是所謂的 "CMOS 時鐘","硬體時鐘", 或 "RTC") 而另一個是 "kernel clock" (有時稱爲 "軟體時鐘" 或 "系統時鐘") 系統時鐘的跳動是以計時器的中斷信號爲基準並在開機時從CMOS 時鐘處載入初始值. 這二個時鐘有不同的時間偏差率, 所以二個時鐘上的時間會逐漸地產生偏差, 而且也會與"眞正的"時間產生偏差.

在 xntpd 文件中所述及的 "the clock" 就是指 "kernel clock". 當你執行 xntpd 或 timed 程式時 (或是其他使用到 adjtimex 系統呼叫的程式) linux 作業系統會假定系統時鐘比 CMOS 時鐘準確, 而且會在開機後每隔 11 分鐘重新設定一次 CMOS 時間 (一直到重開機爲止). 這意味著 "clock" 程式不再知道上一次重新設定 CMOS 時鐘的正確時間, 所以你不可以使用檔案 /etc/adjtime 中的校正因子. 你可以在開機執行檔上使用 ntpdate 程式 以便在初次執行 xntpd 程式之前與校時伺服器對時. 如果你在開機的時候一直無法連上準確的校時源, 這可能會讓你面臨到一些困境– 實際上 xntpd 不是被設計來在那種情況之下使用的.

Xntpd 包含了許多無線電時鐘的驅動程式, 並且也可以被設定在一定的周期內撥電話至 NIST 的電話撥接電腦校時服務 (當你設定好電話撥接的周期之後最好先計算一下電話帳單的花費). 在與其他的校時源失去聯絡一段時間之後這個方式也能提供系統時鐘一個校正因子.

大部分的無線電時鐘大約要花費 $3-4K, 不過你可以採用較便宜的 "gadget box" 方案 (實際上是一個 300 baud 的數據機) 他被放在你的電腦與一台短波無線電接收器之間 頻道調至 Canada 的 CHU 校時無線電台 (參見網址 <ftp://ftp.udel.edu/pub/ntp/gadget.tar.Z>). 而Heathkit WWV 接收器 ("最準確的時鐘") 仍然買的到 (但不是一組套件), 大約要花費 $4-500. 目前GPS 的訊號裡仍然包含有校時的資訊 , 而且部分的 GPS 接收器可以接上電腦的串列埠. 在最近的未來這可能會成爲低價位的解決方案.

理論上, 一般人可以自己寫程式來使用 NIST 的電話撥接電腦校時服務以便自動 計算 CMOS 時鐘與系統時鐘的偏差率. 雖然我不知道是否有單獨的程式能夠達到這個目的, 但是大部分的程式碼應該可以借用自 xntpd 程式集.