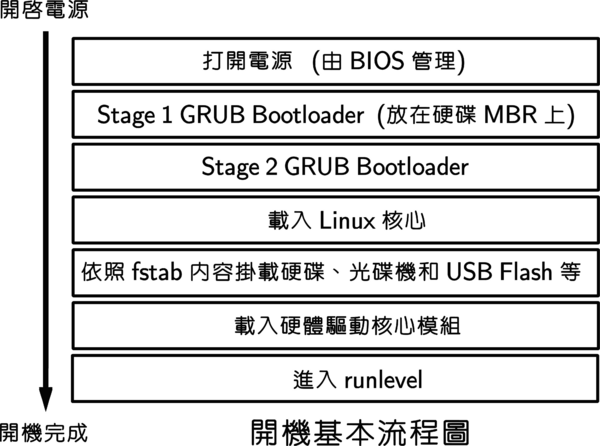
<https://www.dbtsai.com/linux/index.php?title=6.10_-_Ch13_%E6%B7%B1%E5%85%A5%E7%9E%AD%E8%A7%A3Ubuntu%E7%B3%BB%E7%B5%B1&variant=zh-tw>

## 開機流程

當電腦電源開啟時，系統把控制權交給BIOS，BIOS會開始對硬體作檢查並初始化。接下來BIOS會找硬碟的MBR(Master Boot Record)磁區的開機程式並執行！MBR只有512 bytes，在今天的複雜應用無法在MBR放上完整的bootloader開機程式，所以Linux把bootloader GRUB拆成stage 1和stage 2，stage 1放在MBR上。當stage 1上的bootloader從BIOS取得控制權後，會先驅動硬碟，接下來從硬碟內找到stage 2並執行stage 2的bootloader。  
Stage 2的bootloader執行後，會由/boot/grub/menu.lst取得可用的核心和系統。接下來顯示並讓我們選擇使用的核心或者改成從Windows XP開機的選單。若選擇Linux，它會依照menu.lst內的核心檔案位置去載入該核心映像檔，接著該核心會開始初始化硬體並依照fstab把該掛載的硬碟或者光碟機等檔案設備掛載進來。  
為了減少核心大小，有一些硬體驅動程式會編成模組而不會全部塞到核心裡面。此時系統還有一些額外的硬體還沒有驅動，所以核心就會把這些硬體所需要的模組裝起來。最後核心會執行/sbin/init這個程式，這個程式會讀取/etc/inittab裡面的設定，並且進入runlevel，跑完這些script後，系統已經正式的開機完成了。

## 圖13.1



## fstab開機磁碟掛載與UUID新架構

fstab是開機時核心應該要掛載哪些硬碟分割區或光碟等等的設定檔。例如開機後想要把hda1掛載到/，hda2掛載到/home就是把設定寫到這個檔案。傳統上會在/etc/fstab裡面直接指定hdaX掛載到某個目錄底下。

### 為何引入UUID與新的管理方式

以前設定fstab時，大致上如以下範例的內容。第一欄表某個磁碟裝置，第二欄表該設備將要被掛載到的位置，第三欄表該分割區或裝置所使用的檔案格式。最後在加上一些掛載參數和是否開機時要fsck檢驗和備份時要不要dump到。  
**傳統的 /etc/fstab設定檔：**

# <檔案系統> <掛載點> <格式> <參數> <dump> <pass>

/dev/hda4 / reiserfs defaults 0 1

/dev/hda3 /boot ext3 defaults 0 2

/dev/hda6 /home reiserfs defaults 0 2

/dev/hda5 none swap sw 0 0

/dev/hdc /media/cdrom0 udf,iso9660 user,noauto 0 0

檔案系統的設備是SCSI或SATA時，設備代號是sdX，照順序排列。當有設備出現問題時，使用者或許會把它拔掉，使設備代號順序不會有空缺，由後面的遞補，此時fstab裡的設定就會和實際硬體的設備位置不一樣，導致系統無法開機。為了解決這種問題，Ubuntu Edgy引入了UUID的功能，它會在每個分割區和硬碟產生唯一的識別碼，所以fstab不再使用/dev/hdaX來指定裝置，而是使用該裝置的UUID當成裝置代號，當然使用傳統的方式也是可以被允許的，而預設安裝好後是採用UUID的方式。IDE硬碟不會遇到這種問題，畢竟每一個IDE插槽都對應到個別的代號。

**在edgy引入UUID的 /etc/fstab：**

# <檔案系統> <掛載點> <格式> <參數> <dump> <pass>

# 取代原本的hda4

UUID=956cab3e-a68b-45c0-9350-7f0da4810e02 / reiserfs defaults 0 1

# 取代原本的hda3

UUID=633055b1-7736-42ad-a73d-8e1f59b6737a /boot ext3 defaults 0 2

# 取代原本的hda6

UUID=9bd96654-3ac0-4027-9724-d6219fa17f82 /home reiserfs defaults 0 2

# 取代原本的hda5

UUID=af7a979a-a383-441c-a865-9efab61b8cce none swap sw 0 0

在Ubuntu edgy以後，使用者可採用新的設定，或傳統的明確指定方式。在SCSI介面的硬碟中，建議採用UUID設定方法。

## Runlevel原理和設定

Linux採用SysV管理模式，把開機要啟動的服務建立軟連結到/etc/rcX.d時，開機後會依照inittab執行不同的runlevel。如inittab若定義開機後執行runlevel 5，系統開機後會去跑/etc/rcS.d內的指令，接下來執行/etc/rc5.d內的指令。因此在Linux中若要開機後自動啟用某個服務，只需把該服務的執行檔建立一個軟連結到 /etc/rcX.d裡面。  
SysV的方式很古老，它只能依照runlevel內的link依照順序一個個去執行。但今日有些服務需要其他服務先啟用，需要排定runlevel內的執行順序。有些服務和硬體有關，但現在即插即用流行，無法預期開機的時候那些硬體已就緒，無法在runlevel就驅動相關服務。典型的例子就是fstab，在Linux中會依照fstab的內容去掛載設備，但開機時未必該設備已經裝上去。

### init.d內的各種伺服器控制檔

如同前面提到了，開機時系統會依照inittab去跑不同的runlevel，而在那些runlevel裡面都只是軟連結到 /etc/init.d裡面。這樣的好處在於不同的runlevel共用相同的伺服器控制檔。首先看一下/etc/init.d裡面所有的檔案吧！

username@ubuntu:~ $ ls /etc/init.d/

apache2 killprocs rc

apport klogd rc.local

atd linux-restricted-modules-common rcS

bittorrent makedev README

bluetooth mdadm reboot

bluetooth.dpkg-old mdadm-raid rmnologin

checkroot.sh mountnfs-bootclean.sh ssh

console-setup mysql vsftpd

hdparm portmap waitnfs.sh

x11-common

‧‧‧(略)‧‧‧

可在控制檔後加上參數start啟用服務，stop停止服務或restart來重新啟動服務。也可用reload來只載入新的設定檔，或者用force-reload強迫載入新的設定檔。每個服務支援的參數不一定一樣，可執行該控制檔得知此控制檔有支援哪些參數。  
**控制檔操作範例 ：**

*username@ubuntu:~ $ sudo /etc/init.d/apache2 start # 啟用Apache*

*username@ubuntu:~ $ sudo /etc/init.d/apache2 stop # 關閉Apache*

*username@ubuntu:~ $ sudo /etc/init.d/apache2 restart # 重新啟動Apache*

username@ubuntu:~ $ /etc/init.d/apache2 # 得知此控制檔所有可用參數

Usage: /etc/init.d/apache2 start|stop|restart|reload|force-reload

### inittab預設runlevel層級

在開機後，系統會先讀取 /etc/inittab這個設定檔，決定系統進去哪一個runlevel。runlevel可想像成系統提供很多種不同的工作模式。具體而言是可以在runlevel 5啟用圖形介面，runlevel 3不啟用圖形介面，可以分別在/etc/rc5.d裡設定啟用圖形介面，在/etc/rc3.d設定不啟用圖形介面！

|  |  |  |
| --- | --- | --- |
| **runlevel** | **目錄** | **簡介** |
| 0 | /etc/rc0.d | 關機 |
| 1 | /etc/rc1.d | 單人模式 |
| 2 | /etc/rc2.d | 多使用者模式 |
| 3 | /etc/rc3.d | 多使用者模式 |
| 4 | /etc/rc4.d | 多使用者模式 |
| 5 | /etc/rc5.d | 多使用者模式 |
| 6 | /etc/rc6.d | 重新開機 |

修改開機時預設的runlevel很簡單，讀者先如以下範例，編輯 /etc/inittab吧！

**/etc/inittab內容：**

# /etc/inittab: init(8) configuration.

# 開機後會進入的runlevel

id:2:initdefault:

‧‧‧(略)‧‧‧

如以上內容，Ubuntu預設的runlevel是2！若改成6或0則系統會在開機後跑runlevel 0然後又自動關機，或者重新開機！其實runlevel 0和6是提供系統關機和開機時使用的。在Linux中提供init這個程式來改變目前系統的runlevel。  
**改變系統的runlevel：**

username@ubuntu:~ $ sudo init 0 # 進入runlevel 0，等效於關機

username@ubuntu:~ $ sudo init 6 # 進入runlevel 6，等效於重新開機

RedHat的Linux通常runlevel 3是文字模式，runlevel 5是圖形模式，在Ubuntu預設runlevel 2到5是一樣的！

**觀看runlevel 2的運作方式：**

username@ubuntu:~ $ ls /etc/rc2.d/ -l

(略) S10sysklogd -> ../init.d/sysklogd

(略) S11klogd -> ../init.d/klogd

(略) S13gdm -> ../init.d/gdm

(略) S20vsftpd -> ../init.d/vsftpd

(略) S91apache2 -> ../init.d/apache2

‧‧‧(以下略)‧‧‧

以上範例中，軟連結的檔名是S或K開頭，且後面有一個數字。若是S開頭，開機後進入runlevel 2時執行，順序依後面的數字由小到大啟動。若是K開頭的話，依數字由小到大關閉。通常在runlevel 2到5不會看到K開頭的檔案，這些關閉服務的runlevel通常只在runlevel 0和6中出現。在runlevel 0中，讀者會發現同時有K開頭和S開頭的檔案，此時就會先把所有的K開頭檔案跑完，再跑S開頭的檔案。  
以上面例子來說，進入runlevel 2後，會依序啟動sysklogd、klogd、gdm、vsftpd和apache2。在跑runlevel的時候，是透過 /etc/init.d/rc這程式去找runlevel內到底有哪些檔案，它看到S13gdm就會自動翻譯成 /etc/init.d/gdm start，所以在runlevel裡面，若檔案名稱命名錯誤的話，就會自動被忽略。1. 進入runlevel後，先執行關閉的指令 (K開頭)，接下來才跑啟動服務的指令 (S開頭)。2. 不論是K開頭或者S開頭都是依照大小，從小的開始執行，並且數字可以重覆。  
若想要在runlevel 3不要進入視窗介面，要怎樣做呢？先把runlevel 3進入視窗介面的啟動砍了，就達成啦！

**在runlvel 3時進入文字介面：**

# 先砍這個soft link，數字可能會在不同版本的Ubuntu改變！ GDM基本上是

# 提供X Window的登入介面，所以在開機時不啟動這個的話，就沒有視窗介面啦！

username@ubuntu:~ $ sudo rm /etc/rc3.d/S13gdm

username@ubuntu:~ $ vi /etc/inittab

(略)

id:2:initdefault: # 把這裡的2改成3，下次開機就直接進入runlevel 3

(略)

# 若在文字模式想要在進入視窗，可以用以下指令

username@ubuntu:~ $ sudo /etc/init.d/gdm start

以上的範例可修改讓系統開機後預設進入runlevel 3，並且不會啟動視窗介面。這種技巧對伺服器特別有用，因伺服器幾乎是很少用到視窗介面，把它關了可以節省資源，當讀者需要的時候，再用 /etc/init.d/gdm start把它打開。

### 管理某服務是否開機後啟用

在Ubuntu裡面修改runlevel有以下幾種方式。

* 第一種方式是手工刪除或建立軟連結。
* 第二種方式是使用sysv-rc-conf來使用光棒模式設定runlevel，這程式需要額外安裝。

修改runlevel是修改開機時預設啟動的服務或程式，在修改完runlevel後不會立即生效，需重新開機後才會生效。假設把 /etc/rc3.d/S91apache2砍了，只有在下次開機時才不會啟用Apache2！若要立即停止，需執行/etc/init.d/apache2 stop！ 預設Ubuntu從runlevel 2到5都是一樣的內容，而預設在inittab是runlevel 2。  
手工建立或移除runlevel程序，看目前有哪些服務可以加到runlevel裡面：

username@ubuntu:~ $ ls /etc/init.d/

alsa-utils festival mountnfs-bootclean.sh README anacron gdm reboot apache2 glibc.sh mysql hdparm networking

(以下略)

從以上範例中看到的控制script可以加到runlevel裡面讓它自動啟用唷！  
**觀看runlevel 2的運作方式（其他的runlevel也是使用相同方式）：**

username@ubuntu:~ $ ls /etc/rc2.d/ -l

(略) S10sysklogd -> ../init.d/sysklogd

(略) S11klogd -> ../init.d/klogd

(略) S13gdm -> ../init.d/gdm

(略) S20vsftpd -> ../init.d/vsftpd

(略) S91apache2 -> ../init.d/apache2

‧‧‧(以下略)‧‧‧

**在runlevel 2中移除Apache2 ：**

username@ubuntu:~ $ sudo rm /etc/rc2.d/S91apache2

**在runlevel 2中加入Apache2 ：**

username@ubuntu:~ $ sudo ln -s /etc/init.d/apache2 /etc/rc2.d/S91apache2

數字最好和原本的一樣，改前面可能相依的服務沒啟。改太後面會讓該服務比較晚啟動，有時候會影響後面的服務。

ubuntu在6.10開始用upstart替代init，主要腳本都在/etc/event.d下面，默認情況下/etc下沒有inittab文件。

ubuntu下使用sysv-rc-conf管理服務程序。Ubuntu運行級別Linux系統任何時候都運行在一個指定的運行級上，不同的運行級的程序和服務都不同，所要完成的工作和要達到的目的不同，系統可以在這些運行級之間進行切換，以完成不同的工作。要察看當前runlevel，可以輸入命令

*$ runlevel*

切換runlevel可以使用下面命令實現

*$ init [0123456sS]* ##S s是某些版本的選項，以為single user mode，與1大概相同。

*$ apt-get install sysv-rc-conf*

*$ sudo sysv-rc-conf*

運行等級 S：開機進程中第一個運行等級。/etc/init.d/rcS腳本被調用開啟並且/etc/rcS.d目錄下的所有進程將被執行。

運行等級 1：單用戶模式。/etc/rc1.d目錄下的所有進程將被執行。

運行等級 2，3，4，5：在debian系統裡是多用戶環境，可能不包含圖形用戶界面。同樣的，在相應目錄下的進程將被運行。

運行等級 0：關閉計算機

運行等級 6：重起計算機

