

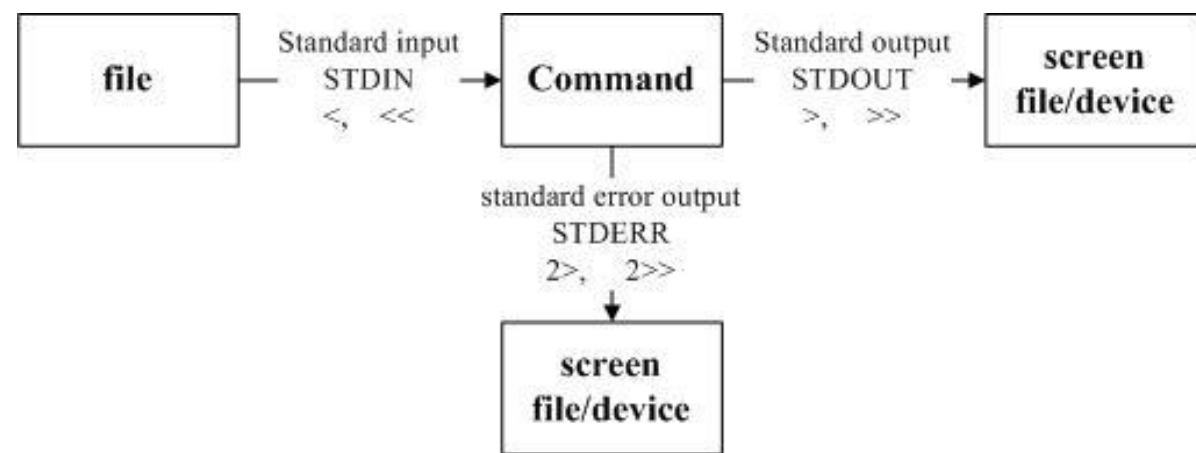
bash 指令連續下達與資料 流重導向

Reading material

- bash 指令連續下達與資料流重導向
 - http://linux.vbird.org/linux_basic_train/unit08.php#8.2

資料流重導向

- 資料流重導向
 - STDIN: 標準輸入裝置，就是鍵盤
 - STDOUT: 標準輸出裝置，就是螢幕
 - STDERR: 標準錯誤輸出，也是螢幕
 - 將『資料傳導到其他地方去』，
 - 例如輸出至檔案存起來，而不是螢幕
 - 或由檔案讀取內容，而不是從鍵盤



standard output 與 standard error output

- 標準輸出: 指令執行所回傳的正確的訊息
- 標準錯誤輸出: 指令執行失敗後，所回傳的錯誤訊息
- 標準輸入 (stdin)：代碼為 0，使用 < 或 << ；
- 標準輸出 (stdout)：代碼為 1，使用 > 或 >> ；
- 標準錯誤輸出(stderr)：代碼為 2，使用 2> 或 2>> ；
- >：以覆蓋的方法將『正確的資料』輸出到指定的檔案或裝置上；
- >>：以累加的方法將『正確的資料』輸出到指定的檔案或裝置上；
- 2>：以覆蓋的方法將『錯誤的資料』輸出到指定的檔案或裝置上；
- 2>>：以累加的方法將『錯誤的資料』輸出到指定的檔案或裝置上；

- 不做重導向

```
[jimmy@localhost ~]$ find /home -name .bashrc  
/home/jimmy/.bashrc  
find: '/home/user1': Permission denied  
find: '/home/user2': Permission denied
```

- 重導向stdout

```
[jimmy@localhost ~]$ find /home -name .bashrc > out  
find: '/home/user1': Permission denied  
find: '/home/user2': Permission denied  
[jimmy@localhost ~]$ cat out  
/home/jimmy/.bashrc  
[jimmy@localhost ~]$
```

- 重導向stderr

```
[jimmy@localhost ~]$ find /home -name .bashrc 2> err  
/home/jimmy/.bashrc  
[jimmy@localhost ~]$ cat err  
find: '/home/user1': Permission denied  
find: '/home/user2': Permission denied  
[jimmy@localhost ~]$
```

- Stdout , stderr重導至不同輸出

```
[jimmy@localhost ~]$ find /home -name .bashrc > out 2> err
[jimmy@localhost ~]$ cat out
/home/jimmy/.bashrc
[jimmy@localhost ~]$ cat err
find: '/home/user1': Permission denied
find: '/home/user2': Permission denied
[jimmy@localhost ~]$
```

- Stdout , stderr重導至相同輸出

```
[jimmy@localhost ~]$ find /home -name .bashrc > all_out 2>&1
[jimmy@localhost ~]$ cat all_out
/home/jimmy/.bashrc
find: '/home/user1': Permission denied
find: '/home/user2': Permission denied
[jimmy@localhost ~]$
```

例題：

- 以一般用戶身份執行找檔名的任務，指令為『 `find /etc -name '*passwd'` 』，螢幕輸出什麼資訊？請了解 `*` 的意義。
- 一般用戶對於系統設定檔原本就有很多的無權限目錄，因此請將錯誤訊息直接丟棄 (導向於 `/dev/null`)
- 將最終螢幕的輸出轉存成為 `~/find_passwd.txt` 檔案

standard input

- 有些指令在執行時，你需要敲擊鍵盤(stdin)才行
 - cat 這個指令就是將敲擊鍵盤的字由螢幕輸出
- 重導向stdio，由檔案輸入

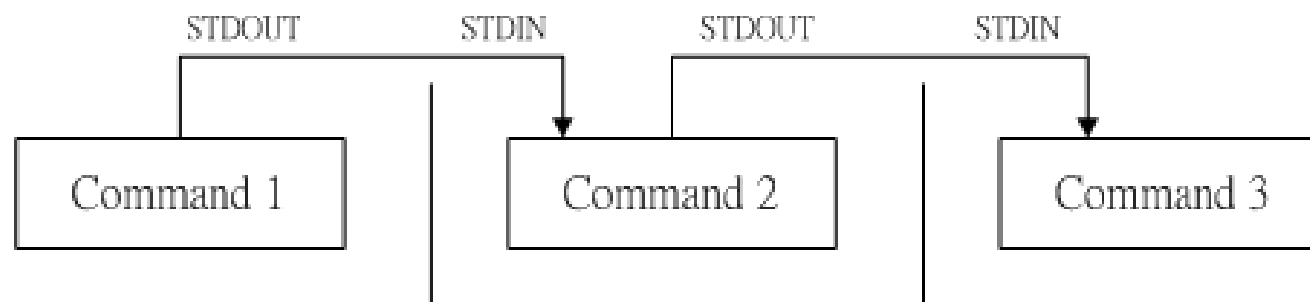
```
[jimmy@localhost ~]$ cat < /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[jimmy@localhost ~]$
```


例題：

- 將 `/etc/hosts` 透過 `cat` 以 `<` 的方式讀入後，用 `>` 輸出到 `mycat.txt` 檔案內

管線 (pipe)

- **bash** 命令執行的時候有輸出的資料會出現，如果這群資料必需要經過幾道手續之後才能得到我們所想要的格式，可以透過**PIPE**來處理
 - 管線僅會處理 **standard output**，對於 **standard error output** 會予以忽略
 - 管線命令必須要能夠接受來自前一個指令的資料成為 **standard input** 繼續處理才行。



常見的管線指令

- `grep` : 擷取特殊關鍵字的功能
- `awk '{print $N}'` : 以空白為間隔，印出第 `N` 個欄位的項目
 - `cat /etc/passwd | awk -F ":" '{print $5}'`
- `sort` : 進行資料排序
 - `du | sort -k 1 -n`
- `cut` : 裁切資料，包括透過固定符號或者是固定字元位置
 - `cat /etc/passwd | cut -d : -f 5`
 - `ls -al | cut -c 1-10`
- `wc` : 計算資料的行數、字數、字元數
 - `cat /etc/hosts | wc -l`

例題：

- 以一般使用者執行 `find /home -name .bashrc`，輸出如下，

```
[jimmy@localhost ~]$ find /home -name .bashrc  
/home/jimmy/.bashrc  
find: '/home/user1': Permission denied  
find: '/home/user2': Permission denied
```

- 若要用 `find /home -name .bashrc | grep Permission > out` 將 `permission denied` 的資料存至 `out`，請問 `out` 內有無結果？
- 要如何修正。

例題：

- 查看 `/etc` 底下共有多少檔案檔名結尾為 `.conf` 的檔案個數

下達連續指令

- 指令是可以連續輸入的，直接透過分號 (;) 隔開每個指令

```
[student@localhost ~]$ date; uptime; uname -r  
五 7月 29 22:16:04 CST 2016  
22:16:04 up 14 days, 23:35, 2 users, load average: 0.00, 0.01, 0.05  
3.10.0-327.el7.x86_64
```

例題：

- 可以將多個指令的輸出重導向至檔案存起來，觀查下面二個指令的差別

```
[student@localhost ~]$ date; uptime; uname -r > myfile.txt  
[student@localhost ~]$ (date; uptime; uname -r ) > myfile.txt
```

具有指令相依性的 && 與 ||

- 分號 (;) 是直接連續下達指令，指令間不必有一定程度的相依性。但當指令之間有相依性時，就能夠使用 && 或 || 來處理。這兩個處理的方式如下：
 - `command1 && command2`
當 `command1` 執行回傳為 0 時(成功)，`command2` 才會執行，否則就不執行。
 - `command1 || command2`
當 `command1` 執行回傳為非 0 時(失敗)，`command2` 才會執行，否則就不執行。

例題：

- 當不存在 `/dev/shm/check` 時，就建立該目錄，若已經建立該目錄，就不做任何動作。
- 當 `/dev/shm/check` 存在時，就將該目錄刪除，否則就不進行任何動作