

第十四章 系統管理程式設計

14-1 帳戶查詢程式

管理帳戶是系統管理員入門的必修課程，如何增加、刪除或暫停某一帳號登入系統，可能藉由前面所介紹的命令操作即可達成。但如果管理大量帳戶（如班級同學），僅靠操作命令來達成，可能要浪費不少時間。我們既然學習了 Shell script 程式設計，為何不利用它來製作一些功能較強的管理工具呢？這裡作者僅提出幾個簡單範例，讓讀者了解 Shell script 程式設計的功能何在，有了這些簡單概念之後，相信讀者很快的即可發展出功能更強的管理工具出來。

當系統管理員最起碼需知道，在您的系統內到底有哪些一般帳戶可以登入。接下來，介紹兩個基本查詢程式。

14-1-1 查詢系統內帳戶名稱

在 `/etc/passwd` 檔案，每一行表示一筆紀錄，並描述一個帳戶。其中 UID（使用者識別碼）如果超過 500 大多給一般帳戶使用，我們可以利用這一點來查詢，到底系統有哪些一般帳戶。UID 位於 `/etc/passwd` 檔案的第三個欄位，我們只要將該欄位內容大於或等於 500 的第一個欄位名稱印出即可。我們可利用 `awk` 命令查詢，如下：

- (1) 所先指定欄位分隔標示冒號 (`-F :`)；
- (2) 命令語句是：第三欄位大於 500 (`$3 > 500`)，
- (3) 則印出第一欄位內容 (`{print $1}`)，
- (4) 並導入 `/etc/passwd` 檔案。

執行此命令時，將 `/etc/passwd` 檔案內以每一行為一筆資料，一筆接一筆進行比對，再印出結果，操作如下：

```
[tsnien@Secure-1 book]$ awk -F : '$3 >= 500 {print $1}' /etc/passwd  
tsnien
```

```
user1
nien
guest
nien1
admin
grad0
mis49411119
guest1
student
```

14-1-2 查詢帳戶名稱

當使用者要求建立帳戶時，大多會提供帳戶名稱 (Login name)。基本上，帳戶名稱是不可以重複的，我們編寫一程式檢視使用者輸入的帳戶名稱是否重複。比對方法很簡單，輸入名稱與 `/etc/passwd` 中第一個欄位內容相比對，如果相同則表示該名稱已被使用。我們利用 `Awk` 命令稿製作，程式內容如下：(`$ vi ex14_2`)

```
# file name: ex14_2
# 功能 Check account
# awk script

# 以下是檔案搜尋之前執行敘述 BEGIN { ...}
BEGIN {
    flag=0;
    printf("please enter name=>");
    getline name <"/dev/stdin";
}

# 開始搜尋檔案內一筆接一筆紀錄 { ...}
{
    if ($1 == name)
        flag=1;
}

# 以下是搜尋完畢後執行敘述 END { ...}
END {
    if (flag == 0)
        printf("%s 帳戶可正常開啟\n", name);
    else
```

```
printf("%s 該帳戶名稱已被使用，請另選其它名稱\n", name);  
}
```

執行結果如下：

```
[root@Secure-1 ~]# awk -F: -f ex14_2 /etc/passwd  
please enter name=>tsnien  
tsnien 該帳戶名稱已被使用，請另選其它名稱  
[root@Secure-1 ~]# awk -F: -f Check_account /etc/passwd  
please enter name=>typeo  
typeo 帳戶可正常開啟
```

14-2 單一帳戶管理程式

雖然利用 `useradd` 命令可以增加單一帳戶，但每次操作此命令之前，必須再次的詳細閱讀操作語法，我們期望發展一個交談式工具，讓我們隨時都可方便的利用它來管理帳戶。

14-2-1 交談式建立單一帳戶

我們利用 Shell script 製作一個簡單範例，如下 (`$vi ex14_3`):

```
#!/bin/bash  
# 功能: Add one user  
echo "歡迎使用 ex14_3 增加一個帳戶"  
  
# 輸入帳戶參數  
read -p "請輸入帳戶名稱 =>" name  
read -p "請輸入群組名稱 =>" groupName  
read -p "請輸入帳戶註解 =>" comment  
read -p "請輸入帳戶密碼 =>" pass  
  
# 執行 useradd 命令  
useradd -g $groupName -c $comment $name  
  
# 設定帳戶密碼  
echo $pass | passwd --stdin $name
```

執行結果如下：

```
[root@Secure-1 book]# ex14_3
歡迎使用 ex14_3 增加一個帳戶
請輸入帳戶名稱 =>user1
請輸入群組名稱 =>users
請輸入帳戶註解 =>資管系學生
請輸入帳戶密碼 =>csuguest
Changing password for user user1.
passwd: all authentication tokens updated successfully.
[root@Secure-1 book]# cat /etc/passwd | grep user1
user1:x:864:100:資管系學生:/home/user1:/bin/bash
[root@Secure-1 book]# ls -l /home |grep user1
drwxr-xr-x  3 user1      users              4096  8月  5 10:19 user1
```

14-2-2 設定帳戶使用期限

雖然利用 `passwd` 命令可以設定帳戶的密碼使用期限，但每次下命令時都必須詳加閱讀命令格式，我們可以寫一個簡單程式，利用交談式輸入各項參數，程式範例如下 (**\$vi ex14_4**):

```
#!/bin/bash
echo "歡迎使用 ex14_4 管理帳戶使用期限"

# 輸入密碼參數
read -p "請輸入帳戶名稱 =>" name
read -p "請輸入密碼最長使用期間 =>" maxDays
read -p "請輸入密碼過期前幾天警告 =>" warnDays
read -p "請輸入密碼到期後幾天鎖定帳戶 =>" inactiveDays

# 執行 passwd 命令
passwd -x $maxDays -w $warnDays -i $inactiveDays $name
```

執行結果如下：

```
[root@Secure-1 book]# ex14_4
```

```

歡迎使用 ex14_4 管理帳戶使用期限
請輸入帳戶名稱 =>user1
請輸入密碼最長使用期間 =>30
請輸入密碼過期前幾天警告 =>3
請輸入密碼到期後幾天鎖定帳戶 =>0
Adjusting aging data for user user1.
passwd: Success
[root@Secure-1 book]# cat /etc/shadow |grep user1
user1:$1$MY6d/Gs8$pwej4.ukFnd4eX5m80y7L1:14461:0:30:3:0::

```

14-2-3 建立帳戶並產生密碼

系統管理者依照客戶要求建立帳號後，大多會利用亂數產生一個密碼，再通知使用者利用此密碼登入系統後，再變更自行設定的密碼。我們製作一個不但可建立帳戶，也順便產生密碼的程式如下：(**\$vi ex14_5**)

```

#!/bin/bash
echo "歡迎使用 ex14_5 增加帳戶並產生密碼"

# 輸入帳戶參數
read -p "請輸入帳戶名稱 =>" name
read -p "請輸入群組名稱 =>" groupName
read -p "請輸入帳戶註解 =>" comment

# 執行 useradd 命令
useradd -g $groupName -c $comment $name

# 產生亂數密碼，至少 8 個字元
pass="" # 設定密碼為空字串
until [ ${#pass} -ge 8 ] # 測試密碼是否超過 8 個字元
do
    pass_nu=$((($RANDOM*50/32767+30)) # 利用亂數產生
    pass_ch=`printf "\x${pass_nu}` # 將所產生的亂數轉換為字元
    pass=$pass$pass_ch # 與上次產生的密碼串接
done

```

```
done
echo "$name 帳號的密碼是 $pass"

# 設定帳戶密碼
echo $pass | passwd --stdin $name
```

執行結果如下：

```
[root@Secure-1 book]# ex14_5
歡迎使用 ex14_5 增加帳戶並產生密碼
請輸入帳戶名稱 =>user1
請輸入群組名稱 =>users
請輸入帳戶註解 =>資工系學生
user1 帳號的密碼是 gf2RsQYe
Changing password for user user1.
passwd: all authentication tokens updated successfully.
```

14-3 大量帳戶管理程式

許多情況需要一次建立多個帳戶，譬如開學時後，老師必須幫每位修課學生開啟帳戶，如果一個接一個利用命令（`adduser`、`passwd`）建立的話；學期結束又要一個接一個刪除，可能非常耗時。我們可利用 Shell 寫一個簡單程式，一次可大量增加或刪除帳戶。

14-3-1 建立大量帳戶

大量帳戶大多有一個共用名稱（如 `user`）、起始與結束號碼（如 `10~12`）、以及相同群組（如 `users`）等等，依照此慣例程式範例如下（`$ vi ex14_6`）：

```
#!/bin/bash
echo "歡迎使用 ex14_6 增加多個帳戶"

# 輸入帳戶參數
read -p "請輸入整體共用名稱 =>" mainName
read -p "請輸入起始號碼 =>" startNum
read -p "請輸入結束號碼 =>" endNum
```

```
read -p "請輸入群組名稱 =>" groupName
read -p "請輸入帳戶註解 =>" comment
read -p "請輸入帳戶共用密碼 =>" pass
if [ "$mainName" == "" ]; then
    echo "帳戶共用名稱不空白"; exit 1
fi

count=$startNum
while [ $count -le $endNum ]; do
    useradd -g $groupName -c $comment ${mainName}$count
    echo ${pass}$count | passwd --stdin ${mainName}$count
    echo "${mainName}$count 帳號的密碼是 ${pass}$count"
    count=`expr $count + 1`
done
echo "完成大量帳戶設定"
```

執行結果如下：

```
[root@Secure-1 book]# ex14_6
歡迎使用 ex14_6 增加多個帳戶
請輸入整體共用名稱 =>user
請輸入起始號碼 =>10
請輸入結束號碼 =>12
請輸入群組名稱 =>users
請輸入帳戶註解 =>電子系學生
請輸入帳戶共用密碼 =>guest
Changing password for user user10.
passwd: all authentication tokens updated successfully.
user10 帳號的密碼是 guest10
Changing password for user user11.
passwd: all authentication tokens updated successfully.
....
passwd: all authentication tokens updated successfully.
user12 帳號的密碼是 guest12
完成大量帳戶設定
```

14-3-2 刪除大量帳戶

與 ex14_6 範例的概念相同，程式範例如下 (`$ vi ex14_7`):

```
#!/bin/bash
echo "歡迎使用 ex14_7 刪除多個帳戶"

# 輸入帳戶參數
read -p "請輸入整體共用名稱 =>" mainName
read -p "請輸入起始號碼 =>" startNum
read -p "請輸入結束號碼 =>" endNum

# 開始刪除多個帳戶
count=$startNum
while [ $count -le $endNum ]; do
    userdel -r ${mainName}$count          # 共用名稱與號碼連結
    echo "已刪除 ${mainName}$count 帳戶了"
    count=`expr $count + 1`
done
echo "完成刪除大量帳戶"
```

執行結果如下：

```
[root@Secure-1 book]# ex14_7
歡迎使用 ex14_7 刪除多個帳戶
請輸入整體共用名稱 =>user
請輸入起始號碼 =>10
請輸入結束號碼 =>12
已刪除 user10 帳戶了
....
已刪除 user12 帳戶了
完成刪除大量帳戶
```

14-3-3 刪除大量檔案

當學期結束後刪除大量帳戶，開學時又新增大量帳戶，好像有點麻煩。新學期也可以

再利用舊帳戶，只要將帳戶內的檔案刪除即可。我們可以編寫一只範例，可以同時刪除大量家目錄內的檔案，範例如下 (**\$vi ex14_8**)：

```
#!/bin/bash
echo "歡迎使用 ex14_8 清除多帳戶家目錄內檔案"

# 輸入帳戶參數
read -p "請輸入整體共用名稱 =>" mainName
read -p "請輸入起始號碼 =>" startNum
read -p "請輸入結束號碼 =>" endNum

count=$startNum
while [ $count -le $endNum ] ; do
    user_d=$mainName$count

    # 由 /etc/passwd 尋找使用者家目錄位置
    home_d=`echo $user_d | awk -F: 'BEGIN {getline temp < "/dev/stdin"}
            {if ($1 == temp) print $6}' /etc/passwd`

    # 搜尋家目錄下有哪些檔案或目錄，並填入 files 陣列內
    files=`ls $home_d`

    # 刪除 files 陣列內所登錄的檔案或目錄
    for file in $files ; do
        rm -r -f $home_d/"$file
    done
    echo "已刪除 $user_d 家目錄($home_d)下所有檔案"
    count=`expr $count + 1`
done
echo "完成刪除多帳戶的家目錄內檔案與目錄"
```

14-3-4 刪除超過 500 K 檔案。

程式範例如下 (**ex14_9**)：

```
#!/bin/bash
echo "歡迎使用 ex14_9 刪除帳戶內較大的檔案"
```

```
# 輸入帳戶參數

read -p "請輸入整體共用名稱 =>" mainName

read -p "請輸入起始號碼 =>" startNum

read -p "請輸入結束號碼 =>" endNum

read -p "請輸入檔案大小(如 500k) =>" Size

count=$startNum
while [ $count -le $endNum ] ; do
    homeDir=`eval echo ~${mainName}$count`
    Files=`find $homeDir -size +$Size -type f -print`
    for file in $Files ; do
        rm -f $file
    done
    count=`expr $count + 1`
done
echo "完成刪除帳戶內較大的檔案"
```

14-3-5 刪除未使用之檔案

我們也可以編寫一程式，搜尋某些帳戶內太久沒有使用的檔案，並將它刪除，範例如下：

(\$vi ex14_10)

```
#!/bin/bash
echo "歡迎使用 ex14_10 刪除帳戶內較舊的檔案"

# 輸入帳戶參數

read -p "請輸入整體共用名稱 =>" mainName

read -p "請輸入起始號碼 =>" startNum

read -p "請輸入結束號碼 =>" endNum

read -p "請輸入舊檔案天數(如 50) =>" Days

count=$startNum
while [ $count -le $endNum ] ; do

    # 利用 eval 命令取出用戶的家目錄
```

```
homeDir=`eval echo ~${mainName}$count`

# 將家目錄下所有舊檔案儲存於 Files_1 陣列內
Files_1=`find $homeDir -atime +$Days -print`

# 將家目錄下舊隱藏檔名稱儲存於 Files_2 陣列內
Files_2=`find $homeDir -atime +$Days -name ".*" -print`
for file_1 in $Files_1 ; do
    flag=1
    for file_2 in $Files_2 ; do
        if [ $file_1 == $file_2 ] ; then
            flag=0
        fi
    done
    if [ $flag == "1" ] ; then
        rm -i $file_1
    fi
done
count=`expr $count + 1`
done
echo "完成刪除帳戶內較大的檔案"
```

執行此範例最大的困擾的是沒有超過 30 天未使用的檔案，但我們還是可以利用 touch 命令設定某一檔案被修改時間，如下（將 file_1 設定為 2007 年 3 月 20 日 10 點 45 分被修改，請自行輸入練習）：

```
[root@Secure-1 book]# touch -t 200703201045 file_1
[root@Secure-1 book]# ls -l file_1
-rw-r--r-- 1 root root 0  3月 20  2007 file_1
```