Windows 批处理(bat)语法大全

批处理(Batch)，也称为批处理脚本。顾名思义，批处理就是对某对象进行批量的处理。批处理文件的扩展名为bat。

目　录

致谢

当前文档 《Windows 批处理(bat)语法大全》 由 进击的皇虫 使用 书栈网(BookStack.CN) 进行构建，生成于 2020-04-06。

内容来源：qingqing-zhao https://www.cnblogs.com/zhaoqingqing/p/4620402.html

文档地址：http://www.bookstack.cn/books/zhaoqingqing-bat

书栈官网：https://www.bookstack.cn

书栈开源：https://github.com/TruthHun

作者：@qingqing-zhao

本文为作者原创，转载请注明出处：https://www.cnblogs.com/zhaoqingqing/p/4620402.html

本文部分内容转载
自：http://m18333611647.blog.163.com/blog/static/225533041201422111365439/

%~dp0[获取当前路径]

%~dp0 “d”为Drive的缩写，即为驱动器，磁盘、“p”为Path缩写，即为路径，目录

cd %~dp0 ：进入批处理所在目录

cd %~dp0bin\ ：进入批处理所在目录的bin目录

示例

这个示例在win10 x64测试正常

```
::作用：以管理员身份安装Apache
d:
cd %~dp0bin\
httpd.exe -k install -n "Apache24"
```

运行结果

以管理员身份运行 示例**.bat** ，执行结果如下:

```
C:\Windows\system32>d:
D:\>cd D:\Server\Apache24\bin\
D:\Server\Apache24\bin>httpd.exe -k install -n "Apache24"
```

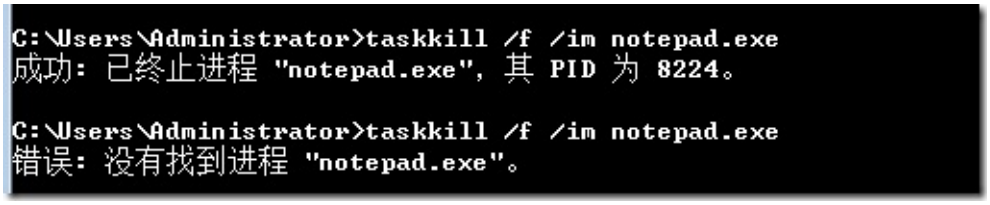
我的常用命令

%cd%[执行的路径]

当前执行的路径，并非目标文件的路径

taskkill /f /im notepad.exe [终止进程]

taskkill /?打开帮助



cmd窗口中文乱码

在CMD窗口右键/默认值，打开属性选择 “默认代码页为简体中文GBK”，

获取命令帮助 xxx /?

遇到记不清楚的命令，但记得名字，就可以键入 命令名 空格 /?就会有详细的该命令的帮助了，比如：ping /?

```
cd /?
```


C:\Users\Administrator>cd /?
显示当前目录名或改变当前目录。
CHDIR [/D] [drive:][path]
CHDIR [..]
CD [/D] [drive:][path]
CD [..]
.. 指定要改成父目录。
键入 CD drive: 显示指定驱动器中的当前目录。
不带参数只键入 CD，则显示当前驱动器和目录。
使用 /D 开关，除了改变驱动器的当前目录之外，
还可改变当前驱动器。
http://www.cnblogs.com/zhaoqingqing/


C:\Users\Administrator>ping /?
用法: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
[-r count] [-s count] [[-j host-list] | [-k host-list]]
[-w timeout] [-R] [-S srcaddr] [-4] [-6] target_name
http://www.cnblogs.com/zhaoqingqing/

查看内置命令的帮助信息

```
ver /?

cmd /?

set /?

rem /?

if /?

echo /?

goto /?

for /?

shift /?

call /?
```

其他常用的命令

```
type /?

find /?

findstr /?

copy /?
```

一、基础语法

1. 批处理文件是一个“.bat”结尾的文本文件，这个文件的每一行都是一条DOS命令。可以使用任何文本文件编辑工具创建和修改。
2. 批处理是一种简单的程序，可以用 if 和 goto 来控制流程，也可以使用 for 循环。
3. 批处理的编程能力远不如C语言等编程语言，也十分不规范。
4. 每个编写好的批处理文件都相当于一个DOS的外部命令，把它所在的目录放到DOS搜索路径(path)中，即可在任意位置运行。
5. C:\AUTOEXEC.BAT 是每次系统启动时都会自动运行的，可以将每次启动时都要运行的命令放入该文件中。
6. 大小写不敏感(命令符忽略大小写)
7. 批处理的文件扩展名为 .bat 或 .cmd。
8. 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来运行该文件。

二、参数

1）系统参数

```
%SystemRoot% === C:\WINDOWS (%windir% 同样)

%ProgramFiles% === C:\Program Files

%USERPROFILE% === C:\Documents and Settings\Administrator (子目录有“桌面”,“开始菜单”,“收藏夹”等)

%APPDATA% === C:\Documents and Settings\Administrator\Application Data

%TEMP% === C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp (%TEM% 同样)

%APPDATA% === C:\Documents and Settings\Administrator\Application Data

%OS% === Windows_NT (系统)

%Path% === %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem (原本的设置)

%HOMEDRIVE% === C: (系统盘)

%HOMEPATH% === \Documents and Settings\Administrator

枚举当前的环境变量

setlocal enabledelayedexpansion

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

2）传递参数给批处理文件

%[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。

Sample:

call test2.bat "hello" "haha" (执行同目录下的“test2.bat”文件，并输入两个参数)

在“test2.bat”文件里写:

echo %1 (打印: "hello")

echo %2 (打印: "haha")

echo %0 (打印: test2.bat)

echo %19 (打印: "hello"9)

三、批处理基本命令

0. help 命令

/? 命令

语法： 命令 /?

可显示此命令的帮助信息

Sample: type /? >>tmp.txt (把 type 命令的帮助信息写入到tmp.txt文件里)

Sample: help type (显示跟“type /?”一样)

1.Echo 命令

语法: echo [{on|off}] [message]

ECHO [ON | OFF] 打开回显或关闭回显功能。

ECHO 显示当前回显设置。

ECHO [message] 显示信息。

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；默认是on，on时会显示如： C:\文件夹路径>命令行。

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。

Sample: echo off

Sample: echo hello world (显示出“hello world”)

Sample: echo Windows Registry Editor Version 5.00 > c:\setupreg.reg (此前还没有 setupreg.reg 这个文件)

Sample: echo "SourcePath"="D:\Win2003\" >> c:\setupreg.reg (追加内容进 setupreg.reg 这个文件)

2.@ 命令

表示不显示@后面的命令，(在入侵过程中自然不能让对方看到你使用的命令啦)

@ 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)。

Sample: @echo off (此语句常用于开头，表示不显示所有的命令行信息，包括此句)

Sample: @echo please wait a minite…

Sample: @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

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

指定跳转到标签行，找到标签行后，程序将处理从下一行开始的命令。

label标签的名字可以随便起，但是最好是有意义的，字母前必须加个冒号“:”来表示这个字母是标签。

goto命令就是根据这个冒号来寻找下一步跳到到那里。经常与 if 配合使用，根据不同的条件来执行不同的命令组。

例题见“5.Pause 命令”

4.Rem 命令

语法: Rem Message…

(小技巧: 用::代替rem)

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，便于别人阅读和自己日后修改。

Sample: @Rem Here is the description.

5.Pause 命令

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

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。

显示的信息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，然后按任意键再次复制。

6.Call 命令

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。

调用另一个批处理程序，并且不终止父批处理程序。

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。

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。

Sample: call="%cd%\test2.bat" haha kkk aaa (调用指定目录下的 test2.bat，且输入3个参数给他)

Sample: call test2.bat arg1 arg2 (调用同目录下的 test2.bat，且输入2个参数给他)

注: 可以调用自身(死循环、递归)

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

Sample: start /MIN test2.bat arg1 arg2 (调用同目录下的 test2.bat，且输入2个参数给他，且本窗口最小化)

Sample: e:\"program files"\极品列车时刻表\jpskb.exe (文件路径名有空格时)

8.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

语法: 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。(注意是两个等号)

Sample: if "%1" == "a" format a:

Sample: if {%1} == {} goto noparms

2) if exist

语法: 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。

Sample: if exist config.sys edit config.sys (表示如果存在这文件，则编辑它，用很难看的系统编辑器)

Sample: if exist config.sys type config.sys (表示如果存在这文件，则显示它的内容)

3) if errorlevel number

语法: if [not] errorlevel <数字> 待执行的命令

如果程序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。(返回值必须按照从大到小的顺序排列)

Sample:

@echo off

XCOPY F:\test.bat D:\

IF ERRORLEVEL 1 (ECHO 文件拷贝失败

) Else IF ERRORLEVEL 0 ECHO 成功拷贝文件

pause

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)，称为错误码errorlevel或称返回码。

常见的返回码为0、1。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。

4) else

语法: if 条件 (成立时执行的命令) else (不成立时执行的命令)

如果是多个条件，建议适当使用括号把各条件包起来，以免出错。

Sample: if 1 == 0 (echo comment1) else if 1==0 (echo comment2) else (echo comment3)

注: 如果 else 的语句需要换行，if 执行的行尾需用“^”连接，并且 if 执行的动作需用(括起来)，否则报错

Sample: if 1 == 0 (echo comment1) else if 1==0 (echo comment2) ^

else (echo comment3)

5）比较运算符：

EQU - 等于（一般使用“==”）

NEQ - 不等于（没有 “!=”,改用“ if not 1==1 ”的写法）

LSS - 小于

LEQ - 小于或等于

GTR - 大于

GEQ - 大于或等于

9.choice 命令

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，

然后配合 if errorlevel 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 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……

Sample: choice /c:dme defrag,mem,end

将显示: defrag,mem,end[D,M,E]?

Sample:

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

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。

1) for {%variable | %%variable} in (set) do command [command-parameters]

%variable 指定一个单一字母可替换的参数。变量名称是区分大小写的，所以 %i 不同于 %I

在批处理文件中使用 FOR 命令时，指定变量建议用 %%variable而不要用 %variable。

(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。

2) 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：

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

如果集里面包含通配符，则指定与目录名匹配，而不与文件名匹配。

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

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。

c.FOR /L %variable IN (start,step,end) DO command [command-parameters]

该集表示以增量形式从开始到结束的一个数字序列。

如: (1,1,5) 将产生序列 1 2 3 4 5； 而(5,-1,1) 将产生序列 (5 4 3 2 1)。

d.有或者没有 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

参数"options"为：

eol=c - 指一个行注释字符的结尾(就一个,如“;”)

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分配。

m-n格式为一个范围。通过 nth 符号指定 mth。

如果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后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一个单引号字符为文字字符串命令并允许在 filenameset中使用双引号扩起文件名称。

3) Sample:

1. 如下命令行会显示当前目录下所有以bat或者txt为扩展名的文件名。

for %%c in (*.bat* .txt) do (echo %%c)

a. 如下命令行会显示当前目录下所有包含有 e 或者 i 的目录名。

```
for /D %%a in (e i) do echo %%a
```

b. 如下命令行会显示 E盘test目录 下所有以bat或者txt为扩展名的文件名。

```
for /R E:\test %%b in (.txt .bat) do echo %%b
```

```
for /r %%c in (*) do (echo %%c) :: 遍历当前目录下所有文件
```

c. 如下命令行将产生序列 1 2 3 4 5

```
for /L %%c in (1,1,5) do echo %%c
```

d. 以下两句，显示当前的年月日和时间

```
For /f "tokens=1-3 delims=-/. " %%j In ('Date /T') do echo %%j年%%k月%%l日
```

```
For /f "tokens=1,2 delims=: " %%j In ('TIME /T') do echo %%j时%%k分
```

e. 把记事本中的内容每一行前面去掉8个字符

```
setlocal enabledelayedexpansion
for /f %%i in (zhidian.txt) do (
set atmp=%%i
set atmp=!atmp:~8!
if {!atmp!}=={} ( echo.) else echo !atmp!
)
:: 读取记事本里的内容(使用 delims 是为了把一行显示全,否则会以空格为分隔符)
for /f "delims=" %%a in (zhidian.txt) do echo.%%a
```

4) continue 和 break

利用 goto 实现程序中常用的 continue 和 break 命令，其实非常简单

continue：在 for 循环的最后一行写上一个标签，跳转到这位置即可

break：在 for 循环的外面的下一句写上一个标签，跳转到这位置即可

Sample：(伪代码)

```
for /F ["options"] %variable IN (command) DO (
… do command …
if … goto continue
```

```
if … goto break

… do command …

:continue

)

:break
```

四、其它命令

1. ping 命令

测试网络联接状况以及信息包发送和接收状况。但是不能够测试端口。

语法: ping IP地址或主机名 [-t] [-a] [-n count] [-l size]

参数含义:

-t 不停地向目标主机发送数据;

-a 以IP地址格式来显示目标主机的网络地址;

-n count 指定要Ping多少次，具体次数由count来指定;

-l size 指定发送到目标主机的数据包的大小。

Sample: ping 192.168.0.1 -t (不停的测试192.168.0.1，按ctrl+c停止)

Sample: for /L %%a in (0,1,255) do ping 192.168.0.%%a -n 1 >> tmp.txt (ping一下所有的局域网电脑)

2. telnet 命令

测试端口使用 telnet IP地址或主机名 端口，使用tcp协议的

Sample: telnet 192.168.0.1 80 (测试192.168.0.1的80端口)

3.color 命令

设置背景及字体颜色

语法: color bf

b 是指定背景色的十六进制数字; f 指定前景颜色(即字体颜色)。

颜色值: 0:黑色 1:蓝色 2:绿色 3:湖蓝 4:红色 5:紫色 6:** 7:白色

8:灰色 9:淡蓝 A:淡绿 B:浅绿 C:淡红 D:淡紫 E:淡黄 F:亮白

如果没有给定任何参数，该命令会将颜色还原到 CMD.EXE 启动时的颜色。

如果两参数一样，视为无效输入。只有一个参数时，设置字体。

4. random 命令

产生随机数(正整数0~)

5. exit 命令

结束程序。即时是被调用的程序，结束后也不会返回原程序

6. shutdown命令

```
shutdown -s 关机
```

五、字符串处理

1）分割字符串，以查看时间为例

%源字符串:~起始值,截取长度% (起始值从0开始；截取长度是可选的，如果省略逗号和截取长度，将会从起始值截取到结尾;

截取长度如果是负数，表示截取到倒数第几个。)

"%time%" 显示如: "11:04:23.03" (完整的时间"hh:mm:ss.tt")

"%time:~0,5%" 显示"hh:mm"(即"11:04")，其中0表示从右向左移位操作的个数，5表示从左向右移位操作的个数

"%time:~0,8%" 显示标准时间格式"hh:mm:ss"(即"11:04:23"，前8个字符串)

"%time:~3,-3%"显示"mm:ss"(即从第4个开始,截去最后3个的字符串)

"%time:~3%" 显示"04:23.03"(即去掉前4个字符串)

"%time:~-3%" 显示".tt"(即最后3个字符串)

上面的字串分割格式，也可以用于其它地方，如目录路径: "%cd:~0,10%"

2）替换字符串

set a="abcd1234"

echo %a% 显示: "abcd1234"

set a=%a:1=kk% 替换“1”为“kk”

echo %a% 显示: "abcdkk234"

3）字符串合并

由于没有直接的字符串合并函数，只能用笨方法了。

set str1=%str1%%str2% (合并 str1 和 str2)

4）计算字符串长度

没有现成的函数。如下程序利用 goto形成循环，不断将字符串截短1，并记录截短的次数，到字符串变成空时的次数即长度。

set testStr=This is a test string

:: 将 testStr 复制到str，str 是个临时字符串

```
set str=%testStr%

:: 标签，用于goto跳转

:next1

:: 判断str是不是空，如果不是则执行下边的语句

if not "%str%"=="" (

:: 算术运算，使num的值自增1，相当于num++或者++num语句

set /a num+=1

:: 截取字符串，每次截短1

set "str=%str:~1%"

:: 跳转到next1标签: 这里利用goto和标签，构成循环结构

goto next1

)

:: 当以上循环结构执行完毕时，会执行下边的语句

echo testStr=%testStr%

echo testStr的长度为: %num%
```

5）截取字符串时，需要传递参数

直接 echo %args:~%num%,-5% 没办法想要的字符串，需要如下两步

```
setlocal enabledelayedexpansion

echo !args:~%num%,-5!
```

六、注册表操作

1）备份注册表，将[HKEY_LOCAL_MACHINE … Run]的内容，备份到“c:\windows\1.reg”

```
reg export HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
c:\windows\1.reg
```

```
reg export HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
c:\windows\2.reg
```

2）修改/添加注册表内容

a.一般的添加或修改

```
reg add "HKCU\Environment" /v Java_Home /t reg_sz /d "D:\Java\jdk1.6.0_07" /f
```

上句解析：“HKCU”是“HKEY_CURRENT_USER”的缩写，不用缩写用全称也可以；

添加名称为“Java_Home”的变量；类型为“reg_sz”，另一种常见类型是“reg_dword”；值为D:\Java\jdk1.6.0_07;

b.使用变量

```
set SoftWareHome=HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java
```

```
reg add "%SoftWareHome%Web Start\1.6.0_07" /v Home /t reg_sz /d "%cd%\jre1.6.0_07\bin"
/f
```

c.如果注册表的名称有空格，或者数据用特殊符号时

```
reg add "%SoftWareHome2%\HelpCommands" /v "01:Online Documentation" /t reg_sz /d
"\"%cd%\Documentation\Index.htm\"" /f
```

传入值为(值用双引号括起来的): "D:\ProgramFiles\1.work_soft\Sybase\PowerDesigner_12\Documentation\Index.htm"

```
reg add "%SoftWareHome2%\Paths" /v ReportTemplates /t reg_sz /d "%cd%\Resource
Files\Report Templates\" /f
```

传入值为(“\”结尾的): E:\Holemar\1.notes\90. Windows\Resource Files\Report Templates\

d.增加空的内容

```
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Shared
Tools\MSConfig\startupreg\IMJPMIG8.1"
```

e.添加或修改默认值

```
reg add "%vpath%InstallPath" /ve /t reg_sz /d "%cd%" /f
```

这里用“/ve”来代替一般修改时的“/v 变量名”,即可修改默认值了

3）删除注册表的内容

双引号里面的是注册表的目录，下面两句将删除这目录下的所有信息

```
reg delete "HKEY_CURRENT_USER\Software\RealVNC" /f

reg delete "HKEY_LOCAL_MACHINE\SOFTWARE\RealVNC" /f
```

双引号里面的是注册表的目录，下面一句将删除这目录下指定的某个信息

```
reg delete "HKEY_LOCAL_MACHINE\Software\RealVNC" /v VNC_Server /f
```

4）注册表的常用位置

a.系统启动项:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]

example: REG ADD HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run /v VNC_Server /t REG_SZ /d "%cd%\VNC_Server.bat" /f
```

b.系统环境变量:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment]
```

c.当前用户的环境变量:

```
[HKEY_CURRENT_USER\Environment]
```

5）修改注册表之后，结束并重新加载explorer.exe进程，可刷新注册表，令其生效

```
taskkill /f /im explorer.exe >nul

start "" "explorer.exe"
```

七、系统服务

1）停止服务：NET STOP 服务名

启动服务：NET Start 服务名

2）设置启动类型

自动： SC CONFIG 服务名 START= auto

手动： SC CONFIG 服务名 START= demand

已禁用：SC CONFIG 服务名 START= disabled

附：“START= ”等号后面必须要有一个空格。(start还有boot,system两个值)

Sample: SC CONFIG Spooler START= demand (打印机加载项，设置成手动，默认自动)

3）查看系统服务：start %SystemRoot%\system32\services.msc /s

八、setlocal与变量延迟

0）在没有开启变量延迟的情况下，某条命令行中的变量改变，必须到下一条命令才能体现。

另外例如for命令等，其后用一对圆括号闭合的所有语句也当作一行。

example:

```
set a=4

set a=5 & echo %a%
```

结果：4

也可以对这种机制加以利用，如下的变量交换

example:

```
set var1=abc

set var2=123

echo 交换前： var1=%var1% var2=%var2%

set var1=%var2%& set var2=%var1%

echo 交换后： var1=%var1% var2=%var2%
```

1）启动批处理文件中环境变量的本地化。本地化将持续到出现匹配的 endlocal 命令或者到达批处理文件结尾为止。

语法：setlocal {enableextension | disableextensions} {enabledelayedexpansion | disabledelayedexpansion}

enableextension：启用命令扩展，直到出现匹配的 endlocal 命令，无论 setlocal 命令之前的设置如何。

disableextensions：禁用命令扩展，直到出现匹配的 endlocal 命令，无论 setlocal 命令之前的设置如何。

enabledelayedexpansion：启用延迟的环境变量扩展，直到出现匹配的 endlocal 命令，无论 setlocal 命令之前的设置如何。

disabledelayedexpansion：禁用延迟的环境变量扩展，直到出现匹配的 endlocal 命令，无论 setlocal 命令之前的设置如何。

2）为了能够感知环境变量的动态变化，批处理设计了变量延迟。简单来说，在读取了一条完整的语句之后，不立即对该行的变量赋值，而会在某个单条语句执行之前再进行赋值，也就是说“延迟”了对变量的赋值。

examle:

```
setlocal enabledelayedexpansion
```

```
set a=4

set a=5 & echo !a!
```

结果： 5

变量延迟的启动语句是“setlocal enabledelayedexpansion”，并且变量要用一对叹号“!!”括起来

由于启动了变量延迟，所以批处理能够感知到动态变化，即不是先给该行变量赋值，而是在运行过程中给变量赋值，因此此时a的值就是5了

另外，启动变量延迟，“%”的变量还是不变

example2:

```
setlocal enabledelayedexpansion

for /l %%i in (1,1,5) do (

set a=%%i

echo !a!

)
```

结果，打印从1到5；如果不变量延迟，一个变量也没有打印

九、文件处理

1.删除

1) 删除一个文件或多个文件

del /s /q /f d:\test\a.bat

将直接删除d:\test\a.bat，没有任务提示

del temp* /q /f /s

将直接删除 本目录的 temp 目录的所有文件，没有任务提示

删除文件的时候可以使用“*”作通配符

2) 删除一个空目录

rd /q /s d:\test\log

将直接删除d:\test\log目录，如果log目录里面有文件将无法删除

3) 删除一个非空目录 (必须指定目录名称)

rmdir /q /s d:\test\logs

必须指定目录名称，不能使用通配符

/S 除目录本身外，还将删除指定目录下的所有子目录

/Q 安静模式，带 /S 删除目录树时不要求确认

无论里面是否有文件或文件夹将全部直接删除

2.创建目录

MKDIR [drive:]path

MD [drive:]path

路径有空格时，可以用双引号括起来，也可以用   替代

实践部分:

==========

小摘录：

1. 调用其他程序时，对文件的大小写不敏感，文件后缀也可忽略

如：start LeapFTP.exe 与 start leapftp 效果一样，都是运行“LeapFTP.exe”文件

每行的开头的字符串会自动查找程序来运行，还可用双引号引起来(文件名或目录名含空格时必须用)

如："D:\Program Files\Leap FTP.exe"

"LeapFTP.exe" 可正常运行文件，start "" "LeapFTP.exe" 也可以正常运行文件(注意，第一个参数是窗口显示的标题)

1. copy C:\test*.* D:\back (复制C盘test文件夹的所有文件(不包括文件夹及子文件夹里的东西)到D盘的back文件夹)
2. dir c:*.* > a.txt (将c盘文件列表写入 a.txt 中)
3. > 生成文件并写入内容(如果有这文件则覆盖)， >> 文件里追加内容
4. md d:\aa (创建文件夹)
5. 在命令末尾加上“>NUL 2>NUL”，表示隐蔽返回信息。
6. 等待用户输入： set /p 变量名=屏幕显示信息。 Sample：set /p pass=请输入密码：
7. 让用户按回车退出

小技巧(替代pause)，文件的最后一句：set /p tmp=操作结束，请按回车键退出…

10.设置标题： title JDK安装

11.设置屏幕显示颜色，如绿色：color 0a

12.清屏： cls

13.查看自己的IP：

for /f "tokens=15" %%i in ('ipconfig ^| find /i "ip address"') do set ip=%%i

echo %ip% (这时的 %ip% 就是自己的IP地址)

1. 修改文件的更新日期

copy 文件名+,,>nul (修改为当前时间，如果要修改为指定时间，先修改系统时间，再改回系统时间)

1. 修改文件的后缀名

ren C:\test*.jpg *.JPG

for /r %%c in (*.jpg) do (ren %%c .JPG) :: 修改当前目录下的所有文件的后缀名，包括子目录的

1. 修改文件的文件名

rename test.jpg test2.JPG

rename *.jpg* .888.JPG

1. 查看DNS、IP、Mac等

1) Win98: winipcfg

2) Win2000以上: Ipconfig /all

3) NSLOOKUP

18.查看IP上的共享资源，就可以

net view 192.168.10.8

19.共享

A.查看你机器的共享资源: net share

B.手工删除共享

net share 共享资源名称$ /d

注意$后有空格。

C.增加一个共享:

net share mymovie=e:\downloads\movie /users:3

mymovie 共享成功。 同时限制链接用户数为3人。

20.打开某网站

start iexplore.exe http://www.baidu.com

实例

1. 生成 reg 文件，运行它，再删除它

```
echo "更改windows安装文件的路径"

echo Windows Registry Editor Version 5.00 > c:\setupreg.reg

echo [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Setup] >>
c:\setupreg.reg

echo "ServicePackSourcePath"="D:\Win2003\" >> c:\setupreg.reg

echo "SourcePath"="D:\Win2003\" >> c:\setupreg.reg

:: 写入注册表

regedit /S c:\setupreg.reg

:: 删除注册表文件

del c:\setupreg.reg
```

2.调用了exe文件,结束后没有关闭，解决方法

用start命令运行文件，如:

```
start LeapFTP.exe 192.168.0.100
```

3.设置系统环境变量

```
:: 有这个环境变量，则不需再设置，直接结束

if not "%JAVA_HOME%" == "" exit

:: 设置环境变量的地址

set inputJavaHome=%cd%\jdk1.6.0_07

:: 设置环境变量，也可以设置当前用户的变量

set EnvironmentHome=HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session
Manager\Environment

echo 正在设置环境变量，请稍候……

reg add "%EnvironmentHome%" /v JAVA_HOME /t reg_sz /d "%inputJavaHome%" /f

reg add "%EnvironmentHome%" /v ClassPath /t reg_sz /d ".;%%JAVA_HOME%%\lib" /f

reg add "%EnvironmentHome%" /v Path /t reg_sz /d "%%JAVA_HOME%%\bin;%Path%" /f]
```

```
:: 刷新，令环境变量生效

taskkill /f /im explorer.exe >nul

start "" "explorer.exe"
```

4.隐藏某目录的所有文件及文件夹

```
cd /d 要隐藏的目录(如: D:)

for /f "usebackq delims=" %%A in ( dir /a /b ) do (attrib "%%A" -r +h -s)

5.在批处理中使用密码。密码为admin，输入正确，跳转到next1 ，若输入密码错误3次，则锁屏。。

@echo off

set num=0

:11

set /p pass=请输入密码:

if "%pass%"=="admin" goto next1

set /a num=%num% + 1

if %num%==3 goto no1

goto 11

:no1

%windir%\system32\rundll32.exe user32.dll,LockWorkStation

goto 11

:next1

echo 密码正确，执行下面的程式

pause
```

6.清空回收站(未成功)

```
@echo off

del /f /s /q c:\recycler*.*

::刷新屏幕

taskkill /f /im explorer.exe >nul

start "" "explorer.exe"
```

7.让系统断断续续地鸣叫

```
@echo off

:begin

:: 发出鸣叫(“”实际就是ASCII码值为7的特殊字符（蜂鸣键beep）

echo

:: 让程序暂停一小阵子

ping -n 1 -l 1 127.1>nul

goto :begin
```

8.将 FAT 卷转换成 NTFS

利用“CONVERT.exe”进行,解析如下:

```
CONVERT volume /FS:NTFS [/V] [/CvtArea:filename] [/NoSecurity] [/X]

volume 指定驱动器号(后面跟一个冒号)、装载点或卷名。

/FS:NTFS 指定要被转换成 NTFS 的卷。

/V 指定 Convert 应该用详述模式运行。

/CvtArea:filename

将根目录中的一个接续文件指定为NTFS 系统文件的占位符。

/NoSecurity 指定每个人都可以访问转换的文件和目录的安全设置。

/X 如果必要，先强行卸载卷。该卷的所有打开的句柄则无效。
```

程序如下:

```
@ ECHO OFF

@ ECHO.

@ ECHO. 说 明

@ ECHO ————————————————————————————————

@ ECHO NTFS是一种磁盘格式。该格式能存放大于4G的单个文件(如高清电影文件)，并可对

@ ECHO 文件夹进行加密，但有个缺点是DOS下无法访问。建议D盘及其后的盘使用NTFS格式，

@ ECHO C盘如非必要可以不转换，FAT32与NTFS这两种格式的读写速度几乎是没有差别的。
```

```
@ ECHO ————————————————————————————————
@ ECHO.
convert c: /fs:ntfs
:: D盘也转成 NTFS
convert d: /fs:ntfs
```

9. 获取我的文档

```
SET SF="HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"
FOR /F "tokens=2,*" %%I IN ('REG QUERY %SF% /v Personal 2^>NUL^|FIND /I "Personal"')
DO SET "myDoc=%%~J"
:: 复制文件到我的文档
XCOPY /D /E /R /Y /C "%cd%\test.txt" "%myDoc%\test\"
```

10 获取当前目录路径

```
cd ./
set CURR_PATH=%cd%
==========================================
```

实例:

```
3.IF-ERRORLEVEL
@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
```

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:\
CHO 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

==========

@echo off //不显示shell的命令。

Setlocal //环境改变只适用于这个文件。

%OS% //为当前的操作系统。

Rem //注释一行文本。

Goto 标签 //改变执行顺序，去标签位置.

: 标签 //定义一个标签。

Set 变量名=值 //定义变量

Not //取反

Netstat –na //显示当前被点用的端口.

%0 %1 %2 //用于表示批处理文件的参数0为命令,共1-9个参数。

Shift //用于向前一个参数，原1变0，原2变1.每调用一次shift向前一移动一位。

Call //调用其他批处理文件或命令。

Start 命令 参数 //指示出在另一个窗口中开始运行命令。

==========

```
:: 这段批处理程序可以自动设置Java环境变量

@echo off

IF EXIST %1\bin\java.exe (

rem 如输入正确的 Java2SDK 安装目录，开始设置环境变量

@setx JAVA_HOME %1

@setx path %path%;%JAVA_HOME%\bin

@setx classpath %classpath%;.

@setx classpath %classpath%;%JAVA_HOME%\lib\tools.jar

@setx classpath %classpath%;%JAVA_HOME%\lib\dt.jar

@setx classpath %classpath%;%JAVA_HOME%\jre\lib\rt.jar

@echo on

@echo Java 2 SDK 环境参数设置完毕，正常退出。

) ELSE (

IF "%1"=="" (

rem 如没有提供安装目录，提示之后退出

@echo on

@echo 没有提供 Java2SDK 的安装目录,不做任何设置，现在退出环境变量设置。

) ELSE (

rem 如果提供非空的安装目录但没有bin\java.exe，则指定的目录为错误的目录

@echo on

@echo 非法的 Java2SDK 的安装目录,不做任何设置，现在退出环境变量设置。

)

)
```

可能遇到问题

中文乱码

把bat文件的编码改为ANSI，UTF-8在win10我这儿会中文显示乱码。