命令参数的路径：要使用反斜杠'**\**'，不要使用正斜杠'/' 如：del d:\test2\file\my.txt

命令参数的路径：若存在空格，应使用双引号将路径引起来 如：del "d:\program files\file\my.txt"

文件及目录名中不能包含下列任何字符：\ / : \* ? " < > |

**rem** // 在批处理文件中添加注解，其后的命令不会被执行，但会回显

:: // ::也可以起到rem的注释作用，且不会有回显

任何以冒号:开头的字符行, 在批处理中都被视作标号（label）, 而直接忽略其后的所有内容 有效标号：冒号后紧跟一个以字母数字开头的字符串，goto语句可以识别 无效标号：冒号后紧跟一个非字母数字的一个特殊符号，goto无法识别的标号，可以起到注释作用，::常被用作注释符号

**0. 获取帮助**

**command /?** // 查看command命令帮助说明

**1. 中断命令执行**

**Ctrl + Z**

**2. 文件/目录**

**cd** 切换目录

例：cd // 显示当前目录

例：cd .. // 进入父目录

例：cd /d d: // 进入上次d盘所在的目录（或在直接输入：d:）

例：cd /d d:\ // 进入d盘根目录

例：cd d: // 显示上次d盘所在的目录

例：cd /d d:\src // 进入d:\src目录

例：cd prj\src\view // 进入当前目录下的prj\src\view文件夹

**pushd popd** 使用栈来维护当前目录

md d:\mp3 // 在C:\建立mp3文件夹 md d:\mp4 // 在D:\建立mp4文件夹 cd /d d:\mp4 // 更改当前目录为d:\mp4 pushd c:\mp3 // 将当前目录d:\mp4入栈，并切换当前目录为c:\mp3 popd // 将刚才保存的d:\mp4弹栈，并设置为当前目录

**dir** 显示目录中的内容

例：dir // 显示当前目录中的子文件夹与文件

例：dir /b // 只显示当前目录中的子文件夹与文件的文件名

例：dir /p // 分页显示当前目录中的子文件夹与文件

例：dir /ad // 显示当前目录中的子文件夹

例：dir /a-d // 显示当前目录中的文件

例：dir c:\test // 显示c:\test目录中的内容

例：dir keys.txt // 显示当前目录中keys.txt的信息

例：dir /S // 递归显示当前目录中的内容

例：dir key\* // 显示当前目录下以key开头的文件和文件夹的信息

例：dir /AH /OS // 只显示当前目录中隐藏的文件和目录，并按照文件大小从小到大排序

**tree** 显示目录结构

例：tree d:\myfiles // 显示d:\myfiles目录结构

**ren** 文件或目录重命名

例：ren rec.txt rec.ini // 将当前目录下的rec.txt文件重命名为rec.ini

例：ren c:\test test\_01 // 将c盘下的test文件夹重命名为test\_01

例：ren Logs.txt Logs-%date:~0,4%%date:~5,2%%date:~8,2%\_%time:~0,2%%time:~3,2%.txt // 将当前目录下的Logs.txt文件重命名为Logs-20150114\_2135.txt或Logs-20150114\_ 812.txt（注意：小时只有个位数时会多一个空格，可以使用字符串替换：将空格替换成0）

**md** 创建目录

例：md movie music // 在当前目录中创建名为movie和music的文件夹

例：md d:\test\movie // 创建d:\test\movie目录

**rd** 删除目录

例：rd movie // 删除当前目录下的movie空文件夹

例：rd /s /q d:\test // 使用安静模式删除d:\test（除目录本身外，还将删除指定目录下的所有子目录和文件）

**copy** 拷贝文件

例：copy key.txt c:\doc // 将当前目录下的key.txt拷贝到c:\doc下（若doc中也存在一个key.txt文件，会询问是否覆盖）

例：copy jobs c:\doc // 将当前目录下jobs文件夹中文件（不递归子目录）拷贝到c:\doc下（若doc中也存在相应的文件，会询问是否覆盖）

例：copy key.txt c:\doc\key\_bak.txt // 将当前目录下的key.txt拷贝到c:\doc下，并重命名为key\_bak.txt（若doc中也存在一个key\_bak.txt文件，会询问是否覆盖）

例：copy /Y key.txt c:\doc // 将当前目录下的key.txt拷贝到c:\doc下（不询问，直接覆盖写）

例：copy key.txt + // 复制文件到自己，实际上是修改了文件日期

例：copy /Y key1.txt + key2.txt key.txt // 将当前目录下的key1.txt与key2.txt的内容合并写入key.txt中（不询问，直接覆盖写）

例：copy /B art\_2.7z.\* art\_2.7z // 将当前目录下的art\_2.7z.开头的所有文件（按照名称升序排序）依次合并生成art\_2.7z

例：copy /B art\_2.7z.001+art\_2.7z.002 art\_2.7z // 将当前目录下的art\_2.7z.001、art\_2.7z.002文件合并生成art\_2.7z

**xcopy** 更强大的复制命令

例：xcopy c:\bat\hai d:\hello\ /y /h /e /f /c // 将c:\bat\hai中的所有内容拷贝到d:\hello中 注意：需要在hello后加上\ 表示hello为一个目录，否则xcopy会询问hello是F，还是D

例：xcopy c:\bat\hai d:\hello\ /d:12-29-2010 // 将c:\bat\hai中的2010年12月29日后更改的文件拷贝到d:\hello中

**move** 移动文件

例：move \*.png test // 将当前目录下的png图片移动到当前目录下test文件夹中 （若test中也存在同名的png图片，会询问是否覆盖）

例：move /Y \*.png test // 将当前目录下的png图片移动到当前目录下test文件夹中 （不询问，直接覆盖写）

例：move 1.png d:\test\2.png // 将当前目录下的1.png移动到d盘test文件夹中，并重命名为2.png （若test中也存在同名的png图片，会询问是否覆盖）

例：move test d:\new // 若d盘中存在new文件夹，将当前目录下的test文件夹移动到d盘new文件夹中；若不存在，将当前目录下的test文件夹移动到d盘，并重命名为new

**del** 删除文件 注意：目录及子目录都不会删除

例：del test // 删除当前目录下的test文件夹中的所有非只读文件（子目录下的文件不删除；删除前会进行确认；等价于del test\\*）

例：del /f test // 删除当前目录下的test文件夹中的所有文件（含只读文件；子目录下的文件不删除；删除前会进行确认；等价于del /f test\\*）

例：del /f /s /q test d:\test2\\*.doc // 删除当前目录下的test文件夹中所有文件及d:\test2中所有doc文件（含只读文件；递归子目录下的文件；删除前不确认）

++++++++++++++++++++++

/ar、/ah、/as、/aa 分别表示删除只读、隐藏、系统、存档文件 /a-r、/a-h、/a-s、/a-a 分别表示删除除只读、隐藏、系统、存档以外的文件

++++++++++++++++++++++

例：del /ar \*.\* // 删除当前目录下所有只读文件

例：del /a-s \*.\* // 删除当前目录下除系统文件以外的所有文件

**replace** 替换文件【即使这个文件在使用，仍然可以替换成功】

例：replace d:\love.mp3 d:\mp3 // 使用d盘下的love.mp3强制替换d盘mp3目录中的love.mp3文件

**mklink** 创建符号链接（win7引入）；创建的符号链接文件上会有一个类似快捷方式的箭头

win7下的mklink命令通过指定参数可以建立出不同形式的文件或目录链接，分为硬链接(hard link)、符号链接(symbolic link)和目录联接(junction)三种。

(1) 符号链接(symbolic link)

　建立一个软链接相当于建立一个文件（或目录），这个文件（或目录）用于指向别的文件（或目录），和win的快捷方式有些类似。

删除这个链接，对原来的文件（或目录）没有影像没有任何影响；而当你删除原文件（或目录）时，再打开链接则会提示“位置不可用”。

(2) 目录联接(junction)

　作用基本和符号链接类似。区别在于，目录联接在建立时会自动引用原目录的绝对路径，而符号链接允许相对路径的引用。

(3) 硬链接(hard link)

　建立一个硬链接相当于给文件建立了一个别名，例如对1.txt创建了名字为2.txt的硬链接；

若使用记事本对1.txt进行修改，则2.txt也同时被修改，若删除1.txt，则2.txt依然存在，且内容与1.txt一样。

建立链接请注意：  
a、建立文件或目录链接限于 NTFS 文件系统；符号链接（目录联接）的建立可以跨分区（如：在d盘可以建立c盘文件或目录的链接），硬链接只能建立同一分区内的文件指向  
b、硬链接只能用于文件，不能用于目录；目录联接只能用于目录；符号链接则均可以；  
c、硬链接不允许对空文件建立链接，符号（软）链接可以。

+++++++++++++++++++++++++++++++++

mklink [[/d] | [/h] | [/j]] Link Target

/d　　 创建目录符号链接。黙认为文件符号链接。 /h　　 创建硬链接，而不是符号链接。 /j　　　创建目录联接。 Link　　指定新的符号链接名称。 Target　指定新链接引用的路径(相对或绝对)。

+++++++++++++++++++++++++++++++++

例：mklink /j "C:\Users" "D:\Users" // 创建D盘Users目录联接到C盘，并命名为Users

**attrib** 查看或修改文件或目录的属性 【A：存档 R：只读 S：系统 H：隐藏】

例：attrib 1.txt // 查看当前目录下1.txt的属性

例：attrib -R 1.txt // 去掉1.txt的只读属性

例：attrib +H movie // 隐藏movie文件夹

**assoc** 设置'文件扩展名'关联到的'文件类型'

例：assoc // 显示所有'文件扩展名'关联

例：assoc .txt // 显示.txt代表的'文件类型'，结果显示.txt=txtfile

例：assoc .doc // 显示.doc代表的'文件类型'，结果显示.doc=Word.Document.8

例：assoc .exe // 显示.exe代表的'文件类型'，结果显示.exe=exefile

例：assoc .txt=txtfile // 恢复.txt的正确关联

**ftype** 设置'文件类型'关联到的'执行程序和参数'

例：ftype // 显示所有'文件类型'关联

例：ftype exefile // 显示exefile类型关联的命令行，结果显示 exefile="%1" %\*

例：ftype txtfile=C:\Windows\notepad.exe %1 // 设置txtfile类型关联的命令行为：C:\Windows\notepad.exe %1

当双击一个.txt文件时，windows并不是根据.txt直接判断用notepad.exe打开 而是先判断.txt属于txtfile'文件类型'；再调用txtfile关联的命令行：txtfile=%SystemRoot%\system32\NOTEPAD.EXE %1

**forfiles** 递归目录执行命令

例：forfiles /p . /m .svn /s /c "cmd /c svn up -r12005" // 在当前目录下查找含有.svn的文件或目录（递归子目录），并对该目录执行指定版本号svn更新

例：forfiles /p c:\myfiles /m .svn /s /c "cmd /c svn up -r12005" // 在c:\myfiles目录下查找含有.svn的文件或目录（递归子目录），并对该目录执行指定版本号svn更新

**3. 文件查看**

**type** 显示文本文件内容

例：type c:\11.txt // 显示c盘中11.txt的文本内容

例：type conf.ini // 显示当前目录下conf.ini的文本内容

例：type c:\11.txt | more // 分页显示c盘中11.txt的文本内容

**more** 逐屏的显示文本文件内容

例：more conf.ini // 逐屏的显示当前目录下conf.ini的文本内容 【空格：下一屏 q：退出 】

**4. 注册表命令**

**reg** 注册表相关操作

参数说明：

KeyName [\Machine]FullKey Machine为远程机器的机器名 - 忽略默认到当前机器。 远程机器上只有 HKLM 和 HKU。 FullKey ROOTKEY+SubKey ROOTKEY [ HKLM | HKCU | HKCR | HKU | HKCC ] SubKey 所选ROOTKEY下注册表项的完整名 /v 所选项之下要添加的值名 /ve 为注册表项添加空白值名<无名称> /t RegKey 数据类型 [ REG\_SZ | REG\_MULTI\_SZ | REG\_DWORD\_BIG\_ENDIAN | REG\_DWORD | REG\_BINARY | REG\_DWORD\_LITTLE\_ENDIAN | REG\_NONE | REG\_EXPAND\_SZ ] 如果忽略，则采用 REG\_SZ /s 指定一个在 REG\_MULTI\_SZ 数据字符串中 用作分隔符的字符；如果忽略，则将""用作分隔符 /d 要分配给添加的注册表ValueName的数据 /f 不提示，强行改写现有注册表项

例：reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v MyApp /t REG\_SZ /d "c:\tools\myapp.exe" /f // 强制添加一条开机启动c:\tools\myapp.exe程序的注册表项

例：reg add "HKLM\SOFTWARE\ScmClient" /v AgreementConfirmed /t REG\_SZ /d 1 /f // 解决32位xp打开ioa后，弹出的框关不掉问题

例：reg add "HKCU\ControlPanel\Desktop" /v WaitToKIllAppTimeOut /t REG\_SZ /d 10000 /f // 强制添加一条加速关闭应用程序的注册表项

例：reg add "hkcu\software\Unity Technologies\Unity Editor 4.x" /v JdkPath\_h4127442381 /t REG\_SZ /f // 将JdkPath\_h4127442381设置为空

例：reg add "HKCR\\*\shell\WinDbg\command" /t REG\_SZ /d "\"D:\Program Files (x86)\windbg\windbg.exe\" -z \"%1\" " /f // 强制添加windbg打开dump文件到右键菜单的注册表项（不指明/v，键值将写入默认值名中）

例：reg add "HKCR\\*\shell\WinHex\command" /t REG\_SZ /d "\"D:\software-setup\system\winhex\winhex.exe\" \"%1\" " /f // 强制添加winhex到右键菜单的注册表项（不指明/v，键值将写入默认值名中）

注册表中%1 %2 %3 %4的含义： -- %1表示文件列表，%2表示默认打印机，%3表示驱动器，%4表示端口

例：reg add "hkcu\software\microsoft\windows\currentversion\internet settings" /v AutoConfigURL /t REG\_SZ /d "[http://txp-01.tencent.com/proxy.pac](https://link.zhihu.com/?target=http%3A//txp-01.tencent.com/proxy.pac)" /f // 为IE设置代理：[http://txp-01.tencent.com/proxy.pac](https://link.zhihu.com/?target=http%3A//txp-01.tencent.com/proxy.pac" \t "_blank)

例：reg add "hkcu\software\Sysinternals\Process Monitor" /v EulaAccepted /t REG\_DWORD /d 1 /f // 为Procmon.exe工具（Process Monitor为其属性面板上的描述名）添加License同意

例：reg delete "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v MyApp /f // 强制删除值名的MyApp的注册表项

例：reg delete "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\taskmgr.exe" /f // 强制删除让任务栏里的任务管理器为灰色的注册表项

例：reg delete HKEY\_CURRENT\_USER\Environment /v HTTP\_proxy /f // 删除http代理

例：reg delete HKEY\_CURRENT\_USER\Environment /v HTTPS\_proxy /f // 删除https代理

例：reg copy "hkcu\software\microsoft\winmine" "hkcu\software\microsoft\winminebk" /s /f // 强制复制winmine下所有的子项与值到winminebk中

例：reg export "hkcu\software\microsoft\winmine" c:\regbak\winmine.reg // 导出winmine下所有的子项与值到c:\regbak\winmine.reg文件中

例：reg import c:\regbak\winmine.reg // 导入c:\regbak\winmine.reg文件到注册表中

例：reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\IEXPLORE.EXE" /s // 查询ie的安装路径

例：reg query HKCR\.dsw /ve // 查询.dsw默认值

例：reg query HKEY\_CURRENT\_USER\Software\Tencent\QQGame\SYS /v GameDirectory // 查询QQGame安装路径

**5. @#@**

**&**顺序执行多条命令，而不管命令是否执行成功

例：cd /d d:\src&work.exe /o c:\result.txt // 先将当前工作目录切换到d:\src下，然后执行work.exe /o c:\result.txt命令

**&&** 顺序执行多条命令，当碰到执行出错的命令后将不执行后面的命令

例：find "ok" c:\test.txt && echo 成功 // 如果找到了"ok"字样，就显示"成功"，找不到就不显示

**||** 顺序执行多条命令，当碰到执行正确的命令后将不执行后面的命令

例：find "ok" c:\test.txt || echo 不成功 // 如果找不到"ok"字样，就显示"不成功"，找到了就不显示

**|** 管道命令

例：dir \*.\* /s/a | find /c ".exe" // 先执行dir命令，然后对输出结果（stdout）执行find命令（输出当前文件夹及所有子文件夹里的.exe文件的个数）

例：dir \*.\* /s/a 2>&1 | find /c ".exe" // 先执行dir命令，然后对输出结果（stdout）和错误信息（stderr）执行find命令（输出当前文件夹及所有子文件夹里的.exe文件的个数）

**>** 将当前命令输出以覆盖的方式重定向

例：tasklist > p1.txt // 将tasklist的输出结果（stdout）以覆盖的方式重定向到p1.txt文件中（注：tasklist的输出结果就不会打印到屏幕上了）

例：tasklist 1> p1.txt // 等同于：tasklist > p1.txt

例：dir bin 2> p1.txt // 输出结果（stdout）打印在屏幕上，错误信息（stderr）以覆盖的方式重定向到p1.txt中（注：bin目录不存在时，会输出错误信息）

例：dir bin > p1.txt 2>&1 // 将错误信息（stderr）重定向到输出结果（stdout），然后将输出结果（stdout）以覆盖的方式重定向到p1.txt中（注：bin目录不存在时，会输出错误信息）

例：dir bin 2> p1.txt 1>&2 // 将输出结果（stdout）重定向到错误信息（stderr），然后将错误信息（stderr）以覆盖的方式重定向到p1.txt中（注：bin目录不存在时，会输出错误信息） 注：与上条命令结果一致

例：tasklist >nul // 屏幕上不打印tasklist的输出结果（stdout），错误信息（stderr）仍会打印

例：dir bin 2>nul // 屏幕上不打印命令的错误信息（stderr），输出结果（stdout）仍会打印（注：bin目录不存在时，会输出错误信息）

例：dir bin >nul 2>&1 // 将命令的错误信息（stderr）重定向到输出结果（stdout），然后不打印输出结果（stdout）【屏幕上错误信息（stderr）和输出结果（stdout）都不打印】（注：bin目录不存在时，会输出错误信息）

例：dir bin 2>nul 1>&2 // 将命令的输出结果（stdout）重定向到错误信息（stderr），然后不打印错误信息（stderr）【屏幕上错误信息（stderr）和输出结果（stdout）都不打印】（注：bin目录不存在时，会输出错误信息）

**>>** 将当前命令输出以追加的方式重定向

例：tasklist >> p2.txt // 将tasklist的输出结果（stdout）以追加的方式重定向到p2.txt文件中（注：tasklist的输出结果就不会打印到屏幕上了）

例：tasklist 1>> p2.txt // 等同于：tasklist >> p2.txt

例：dir bin 2>> p2.txt // 输出结果（stdout）打印在屏幕上，错误信息（stderr）以追加的方式重定向到p2.txt中（注：bin目录不存在时，会输出错误信息）

例：dir bin >> p2.txt 2>&1 // 将错误信息（stderr）重定向到输出结果（stdout），然后将输出结果（stdout）以追加的方式重定向到p2.txt中（注：bin目录不存在时，会输出错误信息）

例：dir bin 2>> p2.txt 1>&2 // 将输出结果（stdout）重定向到错误信息（stderr），然后将错误信息（stderr）以追加的方式重定向到p2.txt中（注：bin目录不存在时，会输出错误信息） 注：与上条命令结果一致

**<** 从文件中获得输入信息，而不是从屏幕上，一般用于date time label等需要等待输入的命令

例：date <temp.txt // temp.txt中的内容为2005-05-01

编号Handle说明0stdin键盘输入1stdout在命令提示窗口上打印输出结果（默认）2stderr在命令提示窗口上打印错误信息（默认）3-9undefined应用程序自己定义和指定

**@** 命令修饰符 在执行命令前，不打印出该命令的内容

例：@cd /d d:\me // 执行该命令时，不打印出命令的内容：cd /d d:/me

**,** 在某些特殊的情况下可以用来代替空格使用

例：dir,c:\ // 相当于：dir c:\

**;** 当命令相同的时候,可以将不同的目标用;隔离开来但执行效果不变。如执行过程中发生错误则只返回错误报告但程序还是会继续执行

例：dir c:\;d:\;e:\ // 相当于顺序执行：dir c:\ dir d:\ dir e:\

**echo.**// 输出一个"回车换行"，空白行

**echo off** // 后续所有命令在执行前，不打印出命令的内容

**echo on** // 后续所有命令在执行前，打印出命令的内容

**echo 123** // 输出123到终端屏幕

**echo "Hello World!!!"** // 输出Hello World!!!到终端屏幕

**echo %errorlevel%** // 每个命令运行结束，可以用这个命令行格式查看返回码；默认值为0，一般命令执行出错会设errorlevel为1

**echo test > p1.txt** // 输出test的字符串到当前目录中的p1.txt文件中（以覆盖的方式）

**set** // 显示当前用户所有的环境变量

**set path** // 查看path的环境变量值（准确的说是查看以path开头的环境变量）

**set path=** // 清空path变量

**set path=d:\execute** // 将path变量设置为d:\execute（注：修改的path只会影响当前回话，也不会存储到系统配置中去；当前cmd窗口关闭，新设置的path也就不存在了）

**set path=%path%;d:\execute** // 在path变量中添加d:\execute（注：修改的path只会影响当前回话，也不会存储到系统配置中去；当前cmd窗口关闭，新设置的path也就不存在了）

**path** // 显示当前path变量的值

**path ;** // 清除所有搜索路径设置并指示cmd.exe只在当前目录中搜索

**path d:\xxx;%PATH%** // 将d:\xxx路径添加到path中

---------------------------------------------------

**set p=aa1bb1aa2bb2** // 设置变量p，并赋值为aa1bb1aa2bb2

**echo %p%** // 显示变量p代表的字符串，即aa1bb1aa2bb2

**echo %p:~6%** // 显示变量p中第6个字符以后的所有字符，即aa2bb2

**echo %p:~6,3%** // 显示第6个字符以后的3个字符，即aa2

**echo %p:~0,3%** // 显示前3个字符，即aa1

**echo %p:~-2%** // 显示最后面的2个字符，即b2

**echo %p:~0,-2%** // 显示除了最后2个字符以外的其它字符，即aa1bb1aa2b

**echo %p:aa=c%** // 用c替换变量p中所有的aa，即显示c1bb1c2bb2

**echo %p:aa=%** // 将变量p中的所有aa字符串置换为空，即显示1bb12bb2

**echo %p:\*bb=c%** // 第一个bb及其之前的所有字符被替换为c，即显示c1aa2bb2

**set p=%p:\*bb=c%** // 设置变量p，赋值为 %p:\*bb=c% ，即c1aa2bb2

**set /a p=39** // 设置p为数值型变量，值为39

**set /a p=39/10** // 支持运算符，有小数时用去尾法，39/10=3.9，去尾得3，p=3

**set /a p=p/10** // 用 /a 参数时，在 = 后面的变量可以不加%直接引用

**set /a p="1&0"** // &运算要加引号。其它支持的运算符参见set/?

---------------------------------------------------

**cls** 清除屏幕

**ver** 显示当前windows系统的版本号

**winver** 弹框显示当前windows系统信息

**vol** 显示当前分区的卷标

**label** 显示当前分区的卷标，同时提示输入新卷标

**label c:system** 设置c盘的卷标为system

**time** 显示或设置当前时间

例：time /t // 显示当前时间

例：time // 设置新的当前时间（格式：hh:mm:ss），直接回车则表示放弃设置

**date** 显示或设置当前日期

例：date /t // 显示当前日期

例：date // 设置新的当前日期（格式：YYYY/MM/DD），直接回车则表示放弃设置

**title** 正在做命令行测试 // 修改当前cmd窗口的标题栏文字为正在做命令行测试

**prompt orz:** // 将命令提示符修改为orz:

**print 1.txt** // 使用设置好的打印机来打印1.txt文本文件

**call ff.bat** // 调用执行ff.bat脚本（ff.bat脚本执行完原脚本才会往下执行）

**start** 运行某程序或命令

例：start /max notepad.exe // 最大化的方式启动记事本

例：start /min calc.exe // 最小化的方式启动计算器

例：start /min "" d:\Proxifier.exe // 最小化的方式启动Proxifier代理工具

例：start tasklist // 启动一个cmd实例窗口，并运行tasklist

例：start explorer f:\ // 调用资源管理器打开f盘

例：strat iexplore "[http://www.qq.com](https://link.zhihu.com/?target=http%3A//www.qq.com)" // 启动ie并打开[http://www.qq.com](https://link.zhihu.com/?target=http%3A//www.qq.com" \t "_blank)网址

例：start ff.bat // 启动开始执行ff.bat（启动ff.bat脚本后，原脚本继续执行，不会等ff.bat脚本执行完）

**exit** 退出当前cmd窗口实例

例：exit 0 // 退出当前cmd窗口实例，并将过程退出代码设置为0（0表示成功，非0表示失败）

例：exit /B 1 // 退出当前bat脚本，并将ERRORLEVEL系统变量设置为1

**pause** 暂停批处理程序，并显示出：请按任意键继续....

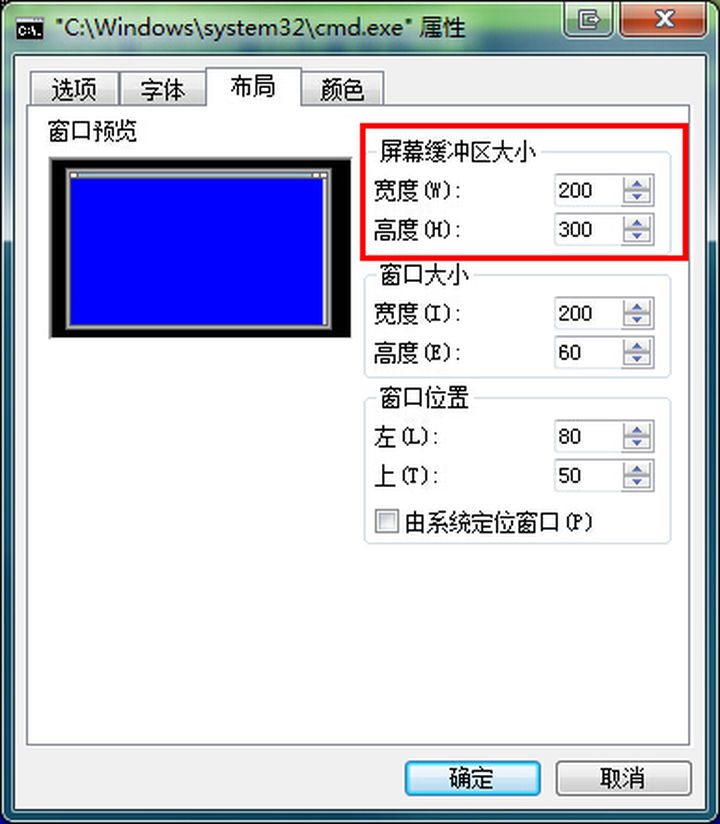
**color** 设置当前cmd窗口背景色和前景色（前景色即为字体的颜色）

例：color // 恢复到缺省设置

例：color 02 // 将背景色设为黑色，将字体设为绿色

-------------------------------------- 0 = 黑色 8 = 灰色 1 = 蓝色 9 = 淡蓝色 2 = 绿色 A = 淡绿色 3 = 浅绿色 B = 淡浅绿色 4 = 红色 C = 淡红色 5 = 紫色 D = 淡紫色 6 = 黄色 E = 淡黄色 7 = 白色 F = 亮白色 --------------------------------------

**mode con cols=200 lines=60 & color 9f**设置DOS窗口颜色为9f，大小：200行 60列（若屏幕缓冲区大小的宽度w<200或高度h<60,最终DOS的窗口就会为w行，h列）



**systeminfo** 查看当前计算机的综合信息

**systeminfo | findstr /i "初始安装日期 系统启动时间"** 只查看当前计算机的初始安装日期和系统启动时间

**wmic** 查看硬件的信息 -- C:\Windows\System32\wbem\WMIC.exe

例：wmic logicaldisk // 查看计算机上各个盘的相关信息

例：wmic LogicalDisk where "Caption='C:'" get FreeSpace,Size /value // 获取C盘的剩余空间大小与总大小（单位：Byte）

例：wmic os get Caption,InstallDate,OSArchitecture /value // 获取当前os的Caption、安装日期以及系统架构信息

**wmic** 查看进程信息

例：wmic process where Caption="buyticket.exe" get commandline,ExecutablePath,ProcessId,ThreadCount /value // 查看名为"buyticket.exe"所有进程命令行，exe全路径，PID及线程数

例：wmic process where Caption="buyticket.exe" get ExecutablePath,HandleCount /value // 查看名为"buyticket.exe"所有进程的exe全路径及当前打开的句柄数

例：wmic process where Caption="buyticket.exe" get ExecutablePath,VirtualSize,WorkingSetSize /value // 查看名为"buyticket.exe"所有进程的exe全路径、当前虚拟地址空间占用及物理内存工作集

**logoff** 注销当前用户

**shutdown** 关闭、重启、注销、休眠计算机

例：shutdown /s // 关闭计算机

例：shutdown /s /t 3600 // 一小时后，关闭本地计算机

例：shutdown /a // 终止系统关闭

例：shutdown /r // 关闭并重启本地计算机

例：shutdown /m 192.168.1.166 /r // 关闭并重启ip为192.168.1.166的计算机

+++++++++++++++++++++

远程关机权限的获取： 1）修改远程pc的“本地安全策略”，为指定的用户开放权限 在WindowsXP默认的安全策略中，只有Administrators组的用户才有权从远端关闭计算机，如果要给xxxx用户远程关机的权限。 可利用WindowsXP的“组策略”或“管理工具”中的“本地安全策略”来实现。 1.命令行运行gpedit.msc打开“组策略编辑器“； 2.导航到“计算机配置/Windows设置/安全设置/本地策略/用户权利指派”； 3.修改“从远端系统强制关机”，添加xxxx用户即可。

2）获得远程IPC管理权限 如果配置第一步后还出现“拒绝访问。”，则需要在运行shutdown命令前先运行如下命令 net use \\[ip地址或计算机名]\ipc$ password /user:xxxx 其中password为帐号xxxx的登录密码。

+++++++++++++++++++++

例：shutdown /g // 关闭并重启计算机，重启后重新启动所有注册的应用程序

例：shutdown /l // 注销本地计算机

例：shutdown /h /f // 休眠本地计算机（强制正在运行的应用程序关闭，不前台警告用户）

例：shutdown /s // 关闭计算机

**regsvr32** 注册或反注册com组件

例：regsvr32 /s clock.ocx // 以无声的方式注册clock.ocx组件

例：regsvr32 /u myCommon.dll // 卸载myCommon.dll组件

**format** 格式化磁盘

例：format J: /FS:ntfs // 以ntfs类型格式化J盘 【类型有:FAT、FAT32、exFAT、NTFS或UDF】

例：format J: /FS:fat32 /Q // 以fat32类型快速格式化J盘

**chkdsk /f D:** // 检查磁盘D并显示状态报告；加参数/f表示同时会修复磁盘上的错误

**subst** 磁盘映射 -- 磁盘映射信息都保存在注册表以下键值中：HKEY\_CURRENT\_USER\Network

例：subst // 显示目前所有的映射

例：subst z: \\com\software // 将\\com\software共享映射为本地z盘

例：subst y: e:\src // 将e:\src映射为本地y盘

例：subst z: /d // 删除z盘映射

**cmdkey** 凭据（保存的用户名和密码）

例：cmdkey /list // 列出可用的凭据

例：cmdkey /list:10.12.190.82 // 列出指定目标的凭据

例：cmdkey /list:Domain:target=10.12.190.82 // 列出指定目标的凭据

例：cmdkey /add:Domain:target=10.12.190.82 /user:LiLei /pass:123456 // 添加凭据

例：cmdkey /delete:Domain:target=10.12.190.82 // 删除指定目标的凭据

**cscript** 执行vbs脚本

例：cscript /Nologo mac.vbs // 执行mac.vbs脚本，显示本机mac地址

-------mac.vbs----------

Dim mc,mo Set mc=GetObject("Winmgmts:").InstancesOf("Win32\_NetworkAdapterConfiguration") For Each mo In mc If mo.IPEnabled=True Then MsgBox "本机网卡MAC地址是: " & mo.MacAddress Exit For End If Next

--------------------------

**6. net命令**

**net start** // 查看已经启动的服务

**net start "Task Scheduler"** // 开启任务计划服务

**net stop "Task Scheduler"** // 关闭任务计划服务

**net start dnscache** // 开启dns缓存服务

**net stop dnscache** // 关闭dns缓存服务

**net share** // 查看当前用户下的共享目录

**net share workFile /delete** // 取消名为workFile的共享状态

**net share xxx=c:\360Downloads** // 将c:\360Downloads设为共享，并取名为xxx

**net share ipc$** // 开启ipc$共享

**net share ipc$ /del** // 删除ipc$共享

**net share c$ /del** // 删除c盘共享

**net use \\192.168.1.166\ipc$ " " /user:" "** // 建立192.168.1.166的ipc空链接

**net use \\192.168.1.166\ipc$ "123456" /user:"administrator"** // 直接登陆后建立192.168.1.166的ipc非空链接（用户名为administrator 密码为123456）

**net use h: \\192.168.1.166\c$ "123456" /user:"administrator"** // 直接登陆后映射192.168.1.166的c盘到本地为h盘（用户名为administrator 密码为123456）

**net use h: \\192.168.1.166\c$** // 登陆后映射192.168.1.166的c盘到本地为h盘

**net use \\192.168.1.166\ipc$ /del** // 删除ipc链接

**net use h: /del** // 删除本地的h盘的映射

**net view** // 查看本地局域网内开启了哪些共享

**net view \\192.168.1.166** // 查看192.168.1.166的机器上在局域网内开启了哪些共享

**net time \\127.0.0.1** // 查看本地机器的日期及时间

**net time \\localhost** // 查看本地机器的日期及时间

**net time \\192.168.1.166** // 查看192.168.1.166机器的日期及时间

**net time \\192.168.1.166 /set** // 设置本地计算机时间与192.168.1.166主机的时间同步，加上参数/yes可取消确认信息

**net user** // 查看当前机器上的用户

**net user Administrator** // 查看当前机器上的Administrator用户的信息

**net user Guest /active:yes** // 启用Guest用户

**net user dev 123456 /add** // 新建一个名为dev，密码为123456的用户

**net localgroup administrators dev /add** // 把名为dev的用户添加到管理员用户组中，使其具有管理员权限

**net user dev /del** // 删除名为dev的用户

**7. 进程操作**

**tasklist** // 显示当前运行的进程信息（可查看PID）

**taskkill** 结束指定的进程

例：taskkill /im notepad.exe // 结束名为notepad.exe的进程

例：taskkill /pid 1230 /pid 1241 /pid 1253 /t // 结束pid为1230、1241和1253的进程以及由它们启动起来的子进程

例：taskkill /f /im cmd.exe /t // 强制结束有名为cmd.exe的进程以及由它启动起来的子进程

**8. 网络操作**

**ping** // 用于检测网络是否通畅，以及网络时延情况（工作在ICMP协议上）

例：ping [http://baidu.com](https://link.zhihu.com/?target=http%3A//baidu.com) // 测试与baidu服务器的连接情况

例：ping chen-pc0 // 测试机器名为chen-pc0的连接情况

例：ping 220.181.111.86 // 测试与ip为220.181.111.86的连接情况

例：ping -l 65500 -n 10 [http://qq.com](https://link.zhihu.com/?target=http%3A//qq.com) // 向[http://qq.com](https://link.zhihu.com/?target=http%3A//qq.com" \t "_blank)发送10次65500字节的ping

例：ping -n 6 127.0.0.1 // 对当前主机执行6次ping操作（花费时间为5s）

例：ping -t [http://baidu.com](https://link.zhihu.com/?target=http%3A//baidu.com) // 不断地测试baidu服务器的连接情况 【Ctrl+Pause Break：查看ping的统计信息；Ctrl+C：终止当前任务】

a. 首先查本地arp cache信息，看是否有对方的mac地址和IP地址映射条目记录 b. 如果没有，则发起一个arp请求广播包，等待对方告知具体的mac地址 c. 收到arp响应包之后，获得某个IP对应的具体mac地址，有了物理地址之后才可以开始通信了,同时对ip-mac地址做一个本地cache d. 发出icmp echo request包，收到icmp echo reply包

**ipconfig /all** // 查看本地ip地址等详细信息

**ipconfig /displaydns** // 显示本地dns缓存的内容

**ipconfig /flushdns** // 清除本地dns缓存的内容

**nslookup**[**http://www.cnblogs.com**](https://link.zhihu.com/?target=http%3A//www.cnblogs.com) // 获取[http://www.cnblogs.com](https://link.zhihu.com/?target=http%3A//www.cnblogs.com" \t "_blank)的域名解析

服务器: [http://gm-captiva.tencent.com//DNS](https://link.zhihu.com/?target=http%3A//gm-captiva.tencent.com//DNS)服务器的主机名 Address: 10.6.18.41//DNS服务器IP

非权威应答: 名称: [http://www.cnblogs.com//](https://link.zhihu.com/?target=http%3A//www.cnblogs.com//)解析的域名URL Address: 42.121.252.58//解析回的IP

**nslookup -d**[**http://www.cnblogs.com**](https://link.zhihu.com/?target=http%3A//www.cnblogs.com) // 打印出[http://www.cnblogs.com](https://link.zhihu.com/?target=http%3A//www.cnblogs.com" \t "_blank)的域名解析所有记录

**netstat -a** // 查看开启了哪些端口

**netstat -n** // 查看端口的网络连接情况

**netstat -v** // 查看正在进行的工作

**netstat -p tcp** // 查看tcp协议的使用情况

**tracert 182.140.167.44** // 查看本机到达182.140.167.44的路由路径

**route print** // 显示出IP路由

**telnet 182.140.167.44 8000** // 探测182.140.167.44是否使用TCP协议监听8000端口（注意：telnet命令不支持UDP端口检测）

说明：如果端口关闭或者无法连接，则显示不能打开到主机的链接，链接失败；端口打开的情况下，链接成功，则进入telnet页面（全黑的），证明端口可用。

用于探测指定IP的端口号，只是telnet的一个基本功能；

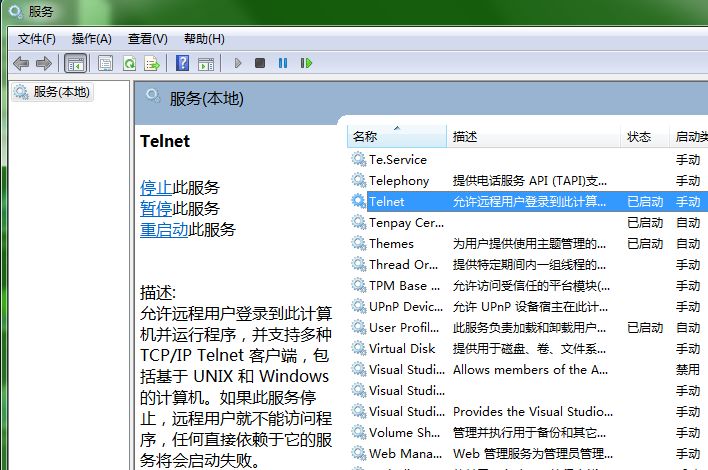
远程登录到网络中的计算机，并以命令行的方式远程管理计算机才是telnet命令的强大之处。

**windows telnet服务器(默认端口：23)环境配置过程如下**： [参考1](https://link.zhihu.com/?target=http%3A//winsystem.ctocio.com.cn/Longhorn/472/8756972.shtml)

a. 安装telnet服务器



b. 启动Telnet服务



c. 关闭windows防火墙 注：若不想关闭防火墙，则需要在Windows防火墙 -- 高级设置里面对Telnet服务器的访问规则进行配置



**ftp 46.19.34.198 21** // 连接46.19.34.198 ftp服务器（21为端口号），然后会要求输入用户名与密码；连接成功后，具体如何使用可以键入?来查看帮助说明

**arp** 显示和修改地址解析协议(ARP)使用的“IP到mac”的地址转换表

例：arp -a // 显示arp缓存表

**at** 计划任务（必须保证“Task Scheduler”服务启动 net start "task scheduler"）

例：at // 查看所有的计划任务

例：at /delete /yes // 停止所有任务计划（不需要确认）

例：at 1 // 开启id为1的计划任务

例：at 1 /delete /yes // 停止id为1的计划任务（不需要确认）

例：at 12:42 shutdown –s –t30 // 到12:42 ，电脑会出现“ 系统关机 ”对话框，并默认 30 秒延时自动关机

例：at cmd /c dir > c:\test.out // 如果命令不是exe文件，必须在命令前加上cmd /c

例：at 6:00AM /every:Saturday task.bat // 在每周六早上6点，电脑定时启动task.bat批处理文件

例：at \\chen 12:00 shutdown /r // 到12:00时，关闭名为chen的计算机

例：at \\192.168.1.166 12:00 shutdown /r // 到12:00时，关闭ip为192.168.1.166的计算机

**9. 文本处理**

**edit config.ini** // 编辑config.ini文件（会进入edit字符编辑器；按alt，可以选择对应的菜单） win7 x64下没有该命令

**find** 文件中搜索字符串

例：find /N /I "pid" 1.txt // 在1.txt文件中忽略大小写查找pid字符串，并带行号显示查找后的结果

例：find /C "exe" 1.txt // 只显示在1.txt文件中查找到exe字符串的次数

例：find /V "exe" 1.txt // 显示未包含1.txt文件中未包含exe字符串的行

**findstr** 文件中搜索字符串

例：findstr "hello world" 1.txt // 在1.txt文件中搜索hello或world

例：findstr /c:"hello world" 1.txt // 在1.txt文件中搜索hello world

例：findstr /c:"hello world" 1.txt nul // 在1.txt文件中搜索hello world，并在每行结果前打印出1.txt: 注：findstr只有在2个及以上文件中搜索字符串时才会打印出每个文件的文件名，nul表示一个空文件

例：findstr /s /i "Hello" \*.\* // 不区分大小写，在当前目录和所有子目录中的所有文件中的hello

例：findstr "^[0-9][a-z]" 1.txt // 在1.txt中搜索以1个数字+1个小写字母开头子串的行

编辑于 2018-04-17