

DOS 命令与批处理

Author: Marlous

E-mail: goonecat@foxmail.com

Date: 2018/10/18

第一章 DOS 命令

一 DOS 常用命令

1 目录命令

- `命令 /?`
作用：查看该命令帮助。
- 将旧文件夹 A 替换为新文件夹 A：

```
rd /S /Q D:\A
D:
cd D:\document
mkdir A
xcopy /E C:\A D:\document\A
```

先删除整个 D 盘旧 A 文件夹；进入 D 盘并创建文件夹 A；将新 A 文件夹中的所有内容复制过去（命令可以删除整个文件夹，但复制只能复制某个文件夹中的全部内容）。

- `md/mkdir`
作用：创建一个子目录（make directory）。
语法：`md[C:][path] <subPath>`
- `cd`
作用：改变或显示当前目录（change directory）。
语法：`cd [C:][path]`
PS：路径可以使用绝对路径和相对路径两种。
`cd\` 表示退回到根目录。
`cd..` 表示退回到上级目录。
如果只有 `cd` 而没有参数，则只显示当前路径。
注意：子目录中一定有两个“特殊目录”，即“.”和“..”，其中一点表示当前目录，两点表示上一层目录。
从简单实用的角度来看，我们只要学会逐层进入（`cd` 下一层某目录名），和逐层退出（`cd..`）就可以解决所有问题。
- `rd /s /q`
作用：删除文件夹及文件。

- `rd`
作用：删除空子目录（remove directory）。
语法：`rd [c:][path]`
PS: `rd`是专门删除空子目录的命令。
- `del`
作用：删除文件命令。
注意两条：一是不能删除非空目录；二是不能删除当前目录。
- `dir`
作用：主要用来显示一个目录下的文件和子目录。(directory)
语法：`dir [C:][path][filename][/o][/s][/p][/w][/a]`
PS: 斜杠表示后面的内容是参数。
`/p` 显示信息满一屏时，暂停显示，按任意键后显示下一屏
`/o` 排序显示。o后面可以接不同意义的字母
`/w` 只显示文件名目录名，每行五个文件名。即宽行显示
`/s` 将目录及子目录的全部目录文件都显示
`/a` 显示隐藏文件
- `path`
作用：设置可执行文件的搜索路径，只对文件有效。
语法：`path[盘符1:][路径1][盘符2:][路径2]...`
PS: 当运行一个可执行文件时，dos会先在当前目录中搜索该文件，若找到则运行之；若找不到该文件，则根据`path`命令所设置的路径，顺序逐条地到目录中搜索该文件
- `tree`
作用：显示指定驱动器上所有目录路径和这些目录下的所有文件名。
语法：`tree [盘符:][[/f][>prn]]`
- `deltree`
作用：删除目录树。
语法：`DELTREE [C1:][path1]`
PS: 这个命令将整个指定目录树全部消灭，而不管它是否是只读、隐藏与否。使用应特别小心。它是一个危险命令。

2 磁盘命令

- `format`
作用：磁盘格式化。
语法：`format <盘符> [/s][/4][/q]`
- `unformat`
作用：对进行过格式化误操作丢失数据的磁盘进行恢复。
语法：`unformat <盘符> [/1][/u][/p][/test]`
- `chkdsk`
作用：显示磁盘状态、内存状态和指定路径下指定文件的不连续数目。
语法：`chkdsk [盘符:][路径][文件名][/f][/v]`
PS: PS: 例如要检查A盘使用情况，就输入`chkdsk A:`，检查c盘使用情况，就输入`chkdsk C:`，如果直接输入`chkdsk`，就检查当前磁盘的使用情况。
- `diskcopy`
作用：复制格式和内容完全相同的软盘。
语法：`diskcopy[盘符1:][盘符2:]`
- `label`
作用：建立、更改、删除磁盘卷标。

语法: `label[盘符:][卷标名]`

- `vol`

作用: 查看磁盘卷标号。

语法: `vol[盘符:]`

- `scandisk`

作用: 检测磁盘的fat表、目录结构、文件系统等是否有问题, 并可将检测出的问题加以修复。

语法: `scandisk[盘符1:][{[盘符2:]...}[/all]]`

- `defrag`

作用: 整理磁盘, 消除磁盘碎片。

语法: `defrag[盘符:][/f]`

PS: 选用/f参数, 将文件中存在盘上的碎片消除, 并调整磁盘文件的安排, 确保文件之间毫无空隙。从而加快读盘速度和节省磁盘空间。

3 文件命令

- `copy`

作用: 拷贝一个或多个文件到指定盘上。

语法: `copy [源盘][路径](源文件名) [目标盘][路径](目标文件名)`

- `xcopy`

作用: 复制指定的目录和目录下的所有文件连同目录结构。

语法: `xcopy [源盘:][源路径名] [目标盘符:][目标路径名][/s][/v][/e]`

PS: xcopy是copy的扩展, 可以把指定的目录连文件和目录结构一并拷贝, 但不能拷贝隐藏文件和系统文件。

- `move`

作用: 移动或重命名文件夹, 移动文件。(不能跨盘符, 跨盘符可先复制再删除)。

- `type`

作用: 显示ascii码文件的内容。

语法: `type [C:][path][filename.ext]`

PS: type命令用来在屏幕上快速、简便地显示文本文件的内容, 扩展名为TXT的文件是文本文件。

- `ren`

作用: 对指定磁盘、目录中的一个文件或一组文件更改名称(rename)。

语法: `ren[盘符:][路径]〈旧文件名〉〈新文件名〉`

PS: 改名操作只限于某个文件某组文件的名称, 它不会更改文件所在的目录。

- `fc`

作用: 比较文件的异同, 并列出差异处。

语法: `fc[盘符:][路径名]〈文件名〉[盘符:][路径名][文件名][/a][/c][/n]`

- `attrib`

作用: 修改指定文件的属性。

语法: `attrib[文件名][r][-r][a][-a][h][-h][-s]`

- `del`

作用: 删除指定的文件。

语法: `del[盘符:][路径]〈文件名〉[/p]`

4 程序状态与配置

- `ipconfig`

作用: ipconfig这个命令查看计算机当前的网络配置信息。

ps: Ipconfig /all: 完全显示计算机的网络信息, IP地址、MAC地址及其他相关的信息, 都可以显示出来。

Ipconfig /release: 释放计算机当前获得的IP地址。对于使用动态IP地址的单位来说, 如果发现机器无法上网, 而计算机从DHCP服务器处获得的IP地址等相关信息不完全, 可以将该地址释放。

Ipconfig /renew: 从DHCP服务器重新获得IP地址。释放了IP地址及相关信息之后, 必须重新获得一个IP地址, 直接输入此命令之后, 便可以从DHCP服务器处获得一个IP地址。如果不用此命令, 要想重新获得一个IP地址信息, 需要重新启动计算机或注销计算机才行。

- `netstat`

作用: netstat命令可以查看单位的网络是否被攻击。

语法: `netstat [-a][-n][-b]`

PS: -a: 显示所有连接和监听端口

-n: 以数字形式显示地址和端口号

-b: 显示包含于创建每个连接或监听端口的可执行组件。另外, 使用该参数之后, 还可以显示占用TCP协议端口的一些程序名称。

- `netstat -aon|findstr "80"`

查看特定端口。

- `tasklist|findstr "2448"`

查看此 pid 的进程。

- `tasklist`

作用: 将整个计算机的进程显示出来, 同任务管理器。

语法: tasklist

- `kill -f 进程名` 加-f参数后强制结束某进程 (为系统的附加工具, 默认是没有安装的, 在安装目录的support/tools文件夹内)

- `ntsd`

作用: ntsd 是一条dos命令, 功能是用于结束一些常规下结束不了的死进程。

使用:

1.利用进程的PID结束进程

命令格式: `ntsd -c q -p pid`

命令范例: `ntsd -c q -p 1332` (结束explorer.exe进程)

2.利用进程名结束进程

命令格式: `ntsd -c q -pn .exe` (.exe 为进程名, exe不能省)

命令范例: `ntsd -c q -pn explorer.exe`

3.taskkill结束进程

命令格式: `taskkill /pid 1234 /f` (也可以达到同样的效果)

5 其他

- `ping`

- `tracert`

- `arp`

- `shutdown`

- `net`

`net view`: 使用此命令查看远程主机的所有共享资源。命令格式为 `net view \IP`。

`net use`: 把远程主机的某个共享资源影射为本地盘符, 图形界面方便使用。命令格式为 `net use x: \IP\sharename`。

`net start`: 使用它来启动远程主机上的服务。用法: `net start servername`

`net stop`: 入侵后发现远程主机的某个服务碍手碍脚, 怎么办? 利用这个命令停掉就ok了, 用法和net start同。

`net user`: 查看和帐户有关的情况, 包括新建帐户、删除帐户、查看特定帐户、激活帐户、帐户禁用等。

1, `net user abcd 1234 /add`, 新建一个用户名为abcd, 密码为1234的帐户, 默认为user组成员。

2, net user abcd /del, 将用户名为abcd的用户删除。

3, net user abcd /active:no, 将用户名为abcd的用户禁用。

4, net user abcd /active:yes, 激活用户名为abcd的用户。

5, net user abcd, 查看用户名为abcd的用户的情况

`net localgroup`: 查看所有和用户组有关的信息和进行相关操作。

`net time`: 这个命令可以查看远程主机当前的时间。

- `at`

必须正在开启Task scheduler服务才能使用 at 命令。可以在命令窗口中输入net start schedule启动它。

作用: 这个命令的作用是安排在特定日期或时间执行某个特定的命令和程序。

用法: at time command \computer

- `ftp`

作用: 首先在命令行键入ftp回车, 出现ftp的提示符, 这时候可以键入“help”来查看帮助(任何DOS命令都可以使用此方法查看其帮助)。

1.ftp

2.open 主机IP ftp端口

3.录入用户名和密码, 就可以进行相应操作了。

dir 跟DOS命令一样, 用于查看服务器的文件, 直接敲上dir回车, 就可以看到此ftp服务器上的文件。

cd 进入某个文件夹。

get 下载文件到本地机器。

put 上传文件到远程服务器。这就要看远程ftp服务器是否给了你可写的权限了, 如果可以, 呵呵, 该怎么利用就不多说了, 大家就自由发挥去吧。

delete 删除远程ftp服务器上的文件。这也必须保证你有可写的权限。

bye 退出当前连接。

quit 同上。

- `telnet`

第二章 批处理

参考文章: <http://www.cn-dos.net/newdos/dosuse.htm#dosbat>

参考文章: [百度百科](#)

参考文章: [脚本之家](#)

一 概述

批处理是一种简化的脚本语言, 也称作宏。它应用于 DOS 和 Windows 系统中, 它是由 DOS 或者 Windows 系统内嵌的命令解释器(通常是 COMMAND.COM 或者 CMD.EXE) 解释运行。类似于 Unix 中的 Shell 脚本。

二 具体命令

1 命令

- `echo`:

`echo on` 或 `echo off` 来打开或关闭回显, `echo off` 等效 `@`, 但它是一个单独的命令, 而不能像@那样放

在其它命令之前，常在开头用 `@echo off` 不显示批处理过程。

显示指定的信息。通常显示在屏幕上。在实际应用中我们会把这条命令和重定向符号结合来实现输入一些命令到特定的文件中。

- `rem` :
用来注释。
- `pause` :
暂停，会挂起批处理。显示 Press any key to continue
- `call` :
从一个批处理程序调用另一个批处理程序，调用完后继续执行原来的批文件,是调用，可以相互传值。
例 `call c:\document\test.bat`
- `start`
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，是执行，只能单向传值给被调用程序。
例 `start calc.exe`
- `goto` :
程序指针跳转到指定的标签，从标签后的第一条命令开始继续执行批处理程序。
例

```
goto labeltest
:labeltest
XXX
```

- `set` :
显示、设置或删除变量。
显示变量：set 或 set s 前者显示批处理当前已定义的所有变量及其值，后者显示所有以 s 开头的变量及值。
设置和调用变量： `set a=7`，调用变量加百分号 `echo %a%`
- `choice` :
选择语句。
例：

```
set /p choice=是否显示当前时间？ (y/n)
if /i not %choice% EQU n echo 当前时间是： %date% %time%
pause
```

2 符号

- `@` :
不显示此条命令。 `echo off` 不显示所有命令。
- `>` 和 `>>` :
将输出信息重定向到指定的设备或文件。系统默认输出到显示器。后者为追加。
- `<` :
将输入信息来源重定向为指定的设备或文件。系统默认从显示器读取输入信息。
例：从 a.txt 文件输入给变量 input，注意 set 命令有个参数 /p

```
@echo off
set /p input=<a.txt
echo content:
echo %input%
pause
```

- `|`:
管道命令，将管道符号前面命令的输出结果重定向输出到管道符号后面的命令中去，作为后面命令的输入。
- `^`:
转义符。
例 `echo aaaa^>test.txt`
- `&`、`&&`、`||`:
`&`: 它的作用是用来连接多个 DOS 命令，并把这些命令按顺序执行，而不管是否有命令执行失败。
`&&`: 当 `&&` 前面的命令成功执行时，执行 `&&` 后面的命令，否则不执行。
`||`: 当 `||` 前面的命令失败时，执行 `||` 后面的命令，否则不执行。

三 语句结构

1 if 语句

1. 字符串比较:

`/i` 则不区分字符串大小写；选择 `not` 项，则对判断结果进行逻辑非。
== 等于，EQU 等于，NEQ 不等于，LSS 小于，LEQ 小于或等于，GTR 大于，GEQ 大于或等于。
也可以用数学符号，但注意要转义。

2. 存在判断:

存在判断的功能是判断文件或文件夹是否存在。

```
if not exist filename (command1) else (command2)
rem 单个命令可不加括号。
```

3. 定义判断:

定义判断的功能是判断变量是否存在，即是否已被定义。
例:

```
set var=111
if defined var (echo var=%var%) else echo var尚未定义!
```

4. 结果判断:

```
masm test.asm
if errorlevel 1 pause & edit test.asm
link %1.obj
```

先对源代码进行汇编，如果失败则暂停显示错误信息，并在按任意键后自动进入编辑界面；否则用 link 程序连接生成的 obj 文件，这种用法是先判断前一个命令执行后的返回码（也叫错误码，DOS 程序在运行完后都有返回码），如果和定义的错误码符合（这里定义的错误码为 1），则执行相应的操作（这里相应的操作为 pause & edit %1.asm 部分）。

2 for 语句

`for /参数 %变量 in (集) do 命令`（命令行用一个百分号，批处理中一定用两个。），分四种参数 D L R F，并且有的参数还可附加另外的选项。

for 命令会在每次循环中，把 in (集) 中读取到的值赋于这个变量，以便其后的命令中引用。由系列文件、字符串或由命令产生的内容形成的集合（当然可用通配符 * ?，还可引用环境变量），for 命令是按一定顺序和规律分次读取集中内容，赋值给变量，并执行 do 后的命令，进行循环下一轮，直至集中内容读取完毕。

- /d 参数是指定仅对目录而不是文件。例 `for /d %%a in (c:*) do echo %a` 表示列出 c 盘根目录下的目录。
- /r 此参数后可以带有路径，在 /r 之后的那个路径，指包含它之下的整个目录树。
- /l `for /L %变量 in (起始值, 每次增值, 结束时的比较值) do 命令`，例: 在 D 盘建立 aa1~ aa5 五个文件夹
`for /L %i in (1,1,5) do mk d:\aa%i`。
- /f 会打开（集）里的文件，使 for 命令能处理文本文件的读取和添加删除替换等编辑性的操作。/f 后还可跟参数，无则默认以空格为分隔。

假定 d:\abc.txt 内容如下：

姓名 性别 年龄 等-级

张三 男 36 A-1

李四 男 29 B-2

赵六 女 31 A-2

/f 后无参数：

`for /f %c in (d:\abc.txt) do @echo %c`

则屏幕上显示：

姓名

张三

李四

赵六

/f 后有参数：

`for /f "skip=1 tokens=1,4 delims= " %c in (d:\abc.txt) do @echo %c %d`

显示为：

张三 A-1

李四 B-2

赵六 A-2

按需要是否“忽略几行”（skip=）、“用什么刀来切分”（delims=）始终空格都会被切分、“最多只需取哪几段”（tokens=）将集里形成的字符串，逐行地分段赋给 % 或 %% 后的变量及可能顺延扩展出的变量，以执行 do 后的命令，每一行即为一轮循环。

四 字符串

- 截取字符串：`%string:~0,5%` 表示从第 0 个字符开始，长度为 5。
- 替换字符串中的某个字符：`%string:hi=hello%` 表示将 string 变量的字符串中的 hi 换成 hello。

- 字符串合并：直接写在一起，`%string1%string2%`。
- 扩充字符串：扩充”这个词来自于微软自己的翻译，意思就是对表示文件路径的字符串进行特殊的处理，该路径字符串变量用 `%x` 表示，例

```
echo 完全路径：%0
echo 去掉引号：%~0
echo 所在分区：%~d0
echo 所处路径：%~p0
echo 文件名：%~n0
echo 扩展名：%~x0
echo 文件属性：%~a0
echo 修改时间：%~t0
echo 文件大小：%~z0
```

其中的 `%0` 是批处理里面的参数，代表当前运行的批处理的完全路径。类似的还有 `%1-%9`，分别代表传递来的第 1-9 个参数。

凡是 `%~` 打头的操作符，都是文件名或环境变量的分离操作。

五 数值计算

和 C 语言类似。

六 传递参数

传参数：

```
例：把参数传给 test.bat
call test.bat a b c
```

接收参数：

```
例：test.bat 的内容，用 %1 %2 ... %9 来对应传入的参数。
rem 输出传入的参数 a b c
echo %1 %2 %3
```

第三章 批处理实例

一 发布博客

```
@echo off
rem @author: Marlous 2018/10/13
rem 脚本说明：自动发布博客脚本
```

```
rem ---设置---博客所在盘符，目录。
set disk=E:
set blogPath=E:\WorkZone\Repositories\my-blog

%disk%
cd %blogPath%

call hexo clean
call hexo g
call hexo d

timeout /t 3
```

二 测试博客

```
@echo off
rem @author: Marlous 2018/10/13
rem 脚本说明：测试 HEXO 博客，自动打开浏览器。

rem ---设置---博客所在盘符，目录。
set disk=E:
set blogPath=E:\WorkZone\Repositories\my-blog

%disk%
cd %blogPath%

start hexo server

rem ---设置---测试地址，火狐浏览器所在目录，路径有空格用引号包裹，防止误认为多个参数。
set testPath=http://localhost:4000/

rem 等待 3 秒。
timeout /t 3

set browsePath="D:\Program Files\Mozilla Firefox\firefox.exe"
start %browsePath% %testPath%
```

三 备份博文

```
@echo off
rem @author: Marlous 2018/10/13
rem 脚本说明：自动备份博文到备份文件夹，推送到远端仓库。
rem 1.先删除备份文件夹；2.进入备份文件夹盘符，具体位置；3.创建备份文件夹；4.将要备份文件夹内容复制到备份文件夹；5.推送到远端仓库。

rem ---设置---要删除的备份文件夹。
set backupDocumentPath=E:\WorkZone\Repositories\files-backup\hexo_blog_backup\source
rd /S /Q %backupDocumentPath%
```

```
rem ---设置---进入备份文件夹（被删除）所在盘符。
set disk=E:
%disk%

rem ---设置---,进入备份文件夹（被删除）所在位置。
set backupDocumentSite=E:\WorkZone\Repositories\files-backup\hexo_blog_backup
cd %backupDocumentSite%

rem ---设置---创建的备份文件夹名称。
set backupDocumentName=source
mkdir %backupDocumentName%

rem 将要备份的文件夹中内容复制到备份文件夹中。
rem ---设置---需要做备份的文件夹路径。
set documentPath=E:\WorkZone\Repositories\my-blog\source
xcopy /E %documentPath% %backupDocumentPath%

rem 获取系统时间变量，年月日 yyyy/mm/dd，第一个数字为起始，第二个为长度。
set yyyy=%DATE:~0,4%
set mm=%DATE:~5,2%
set dd=%DATE:~8,2%

rem 添加所有文件到缓冲区，做成版本（添加 commit 内容），推送到远程端。
git add -A
git commit -m "v%yyyy%%mm%%dd%"
git push origin master

rem pause>nul 进行从定向，所有的提示都不显示。
timeout /t 3
```