

# BAT 语法

作者: [yscxy](#)

原文链接: <https://ld246.com/article/1608024188724>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

 <https://b3logfile.com/bing/20181018.jpg?imageView2/1/w/960/h/540/interlace/1/q/100>

BAT 文件语法和技巧 (bat 文件的编写及使用) (转载一)

比较有用的东比较有用的东西

首先, 批处理文件是一个文本文件, 这个文件的每一行都是一条 DOS 命令 (大部分时候就好象我们在 DOS 提示符下执行的命令行一样), 你可以使用 DOS 下的 Edit 或者 Windows 的记事本(notepad) 任何文本文件编辑工具创建和修改批处理文件。

其次, 批处理文件是一种简单的程序, 可以通过条件语句(if)和流程控制语句(goto)来控制命令流的流程, 在批处理中也可以使用循环语句(for)来循环执行一条命令。当然, 批处理文件的编程能力与 C 语言等编程语句比起来是十分有限的, 也是十分不规范的。批处理的程序语句就是一条条的 DOS 命令 (包括内部命令和外部命令), 而批处理的能力主要取决于你所使用的命令。

第三, 每个编写好的批处理文件都相当于一个 DOS 的外部命令, 你可以把它所在的目录放到你的 DOS 搜索路径(path)中来使得它可以在任意位置运行。一个好的习惯是在硬盘上建立一个 bat 或者 batch 目录 (例如 C:\BATCH), 然后将所有你编写的批处理文件放到该目录中, 这样只要在 path 设置上 c:\batch, 你就可以在任意位置运行所有你编写的批处理程序。

第四, 在 DOS 和 Win9x/Me 系统下, C:盘根目录下的 AUTOEXEC.BAT 批处理文件是自动运行处理文件, 每次系统启动时会自动运行该文件, 你可以将系统每次启动时都要运行的命令放入该文件, 例如设置搜索路径, 调入鼠标驱动和磁盘缓存, 设置系统环境变量等。下面是一个运行于 Windows 98 下的 autoexec.bat 的示例:

```
<table>
<thead>
<tr>
<th>1234567</th>
<th><code>@</code><code>ECHO</code>&lt;&lt;code>&lt;&lt;code>OFF</code> PATH C:\WINDOWS;C:
WINDOWS\COMMAND;C:\UCDOS;C:\DOSTools;C:\SYSTOOLS;C:\WINTOOLS;C:\BATCH <cod
>LH SMARTDRV.EXE /X</code> LH DOSKEY.COM /INSERT<code>LH CTMOUSE.EXE</cod
> SET</code> <code>TEMP=D:\TEMP ``SET</code> <code>TMP=D:\TEMP</code></th>
</tr>
</thead>
</table>
```

\*\* 批处理的作用

\*\*简单的说, 批处理的作用就是自动的连续执行多条命令。

这里先讲一个最简单的应用: 在启动 wps 软件时, 每次都必须执行 (&gt; 前面内容表示 DOS 示符):

```
C:&gt;cd wps<br>
C:\WPS&gt;spdos<br>
C:\WPS&gt;py<br>
C:\WPS&gt;wbx<br>
C:\WPS&gt;wps<br>
```

如果每次用 WPS 之前都这样执行一遍, 您是不是觉得很麻烦呢?

好了, 用批处理, 就可以实现将这些麻烦的操作简单化, 首先我们编写一个 runwps.bat 批处理件, 内容如下:

```
<table>
<thead>
<tr>
<th>12345678</th>
<th><code>@</code><code>echo</code>&lt;&lt;code>&lt;&lt;code>off</code> c:<code>cd</code>\wp
<code>spdos</code> py <code>wbx</code> wps <code>cd</code>&lt;&lt;code>&lt;&lt;code>/co
e</code></th>
</tr>
</thead>
</table>
```

以后, 我们每次进入 wps, 只需要运行 runwps 这个批处理文件即可。

\*\*常用命令

\*\*<br>

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，我们就从他开始学起。<br>

echo 表示显示此命令后的字符<br>

echo off 表示在此语句后所有运行的命令都不显示命令本身<br>

@ 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（只能影当前行）。<br>

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将无法返回当前文件并执行当前文件的后续命令）。<br>

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，等待用户任意键后继续<br>

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）。</p>

<p>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：根目录中所有文件写入 a.txt 中，启动 UCdos，进入 WPS 等功能。</p>

<p> 批处理文件的内容为： 命令注释：</p>

<p> @<a href="https://ld246.com/member/echo" aria-name="echo" class="tooltipp d\_user" target="\_blank">echo</a> off 不显示后续命令行及当前命令行<br>

dir c:.* &gt;a.txt	将 c 盘文件列表写入 a.txt 
call c:\ucdos\ucdos.bat	调用 uc dos 
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 uc dos	进入 uc dos 目录 
wps	运行 wps </p>

<p>批处理文件的参数</p>

<p>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），这需要一个参数表示符"%”。</p>

<p>%[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。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。</p>

<p>例 2：C:根目录下有一批处理文件名为 f.bat，内容为：</p>

<thead>
<tr>
<th>12</th>
<th><code>@<code>echo</code>&lt;/code> &lt;/code>&lt;/code>off </code> format %1</code> </th>
</tr>
</thead>
</table>

<p>如果执行 C:&gt;f a:<br>

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:</p>

<p>例 3：C:根目录下一批处理文件名为 t.bat，内容为:</p>

<thead>
<tr>
<th>123</th>
<th><code>@<code>echo</code>&lt;/code> &lt;/code>&lt;/code>off </code> type</code> <code>%1</code> <code>%2</code> </th>
</tr>
</thead>
</table>

<p>那么运行 C:&gt;t a.txt b.txt<br>

%1 : 表示 a.txt<br>

%2 : 表示 b.txt<br>

于是上面的命令将顺序地显示 a.txt 和 b.txt 文件的内容。<br>

\*\*<br>

特殊命令\*\*</p>

<p>if goto choice for 是批处理文件中比较高级的命令，如果这几个你用得很熟练，你就是批处理件的专家啦。</p>

<p>一、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:</p><ol>

<li>if [not] "参数" == "字符串" 待执行的命令</li>

</ol>

<p>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。<p>

<p>例: if "%1"=="a" format a:</p>

<ol start="2">

<li>if [not] exist [路径]文件名 待执行的命令<br>

如果有指定的文件，则条件成立，运行命令，否则运行下一句。</li>

</ol>

<p>如: if exist c:\config.sys type c:\config.sys<br>

表示如果存在 c:\config.sys 文件，则显示它的内容。</p>

<ol start="3">

<li>if errorlevel &lt; 数字 &gt; 待执行的命令</li>

</ol>

<p>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 error level 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。</p>

<p>如 if errorlevel 2 goto x2</p>

<p>二、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。</p>

<p>如:</p>

<p>goto end</p>

<p>:end<br>

echo this is the end</p>

<p>标号用":字符串"来定义，标号所在行不被执行。</p>

<p>三、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。</p>

<p>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 命令语法稍有不同，请用 choice /?查看用法。</p>

<p>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): </p>

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

<p>描述:<br>

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。</p>

<p>参数列表:<br>

/C choices 指定要创建的选项列表。默认列表是 "YN"。</p>

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

<p> /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。<p>

<p> /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。如果指了 0，就不会有暂停，默认选项<br>

会得到选择。</p>

<p> /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;同时，必须用 /T 指定 nnnn。</p>

<p> /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。 </p>

<p> /? 显示帮助消息。 </p>

<p> 注意:<br>

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，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。 </p>

<p>示例:<br>

CHOICE /?<br>

CHOICE /C YNC /M "确认请按 Y，否请按 N，或者取消请按 C。" <br>

CHOICE /T 10 /C ync /CS /D y<br>

CHOICE /C ab /M "选项 1 请选择 a，选项 2 请选择 b。" <br>

CHOICE /C ab /N /M "选项 1 请选择 a，选项 2 请选择 b。" <br>

<br>

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。" <br>

屏幕上会显示： <br>

确认请按 Y，否请按 N，或者取消请按 C。 [Y,N,C]?<br>

<br>

<br>

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）:</p>

<table>

<thead>

<tr>

<th>12345678910111213141516</th>

<th><code>@</code><code>echo</code>&lt;&code>&code>off </code> choice /C dme /M <code>"defrag,mem,end" </code> if errorlevel 3 goto end <code>if errorlevel 2 goto mem </code> if errorlevel 1 goto defrag <code>&nbsp;&lt;/code>&lt;&code>&code>;defrag </code> c:\dos defrag <code>goto end </code>&nbsp;</code> <code>:mem <code>mem </code> goto nd <code>&nbsp;&lt;/code>&lt;&code>&code>;end </code> echo <code>good bye </code></th>

</tr>

</thead>

</table>

<p>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。 </p>

<p>\*\*四、for 循环命令，只要条件符合，它将多次执行同一命令。 <br>

\*\*<br>

语法： <br>

对一组文件中的每一个文件执行某个特定命令。 </p>

<p>FOR %%variable IN (set) DO command [command-parameters]</p>

<p>%%variable 指定一个单一字母可替换的参数。 <br>

(set) 指定一个或一组文件。可以使用通配符。 <br>

command 指定对每个文件执行的命令。 <br>

command-parameters 为特定命令指定参数或命令行开关。 </p>

<p>例如一个批处理文件中有一行:<br>

for %%c in (\*.bat \*.txt) do type %%c</p>

<p>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。 </p>

<p>批处理示例</p>

<ol>

<li>IF-EXIST</li>

</ol>

<ol>

</li></li>

</ol>

<p>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: <br>

```
@<a href="https://ld246.com/member/echo" aria-name="echo" class="tooltipped__user" target="_blank">echo</a> off<br>
```

```
IF EXIST \AUTOEXEC.BAT TYPE \AUTOEXEC.BAT<br>
```

```
IF NOT EXIST \AUTOEXEC.BAT ECHO \AUTOEXEC.BAT does not exist</p>
```

<p>然后运行它: <br>

```
C:>TEST1.BAT</p>
```

<p>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, 批处理就会显示你该文件不存在。</p>

<ol start="2">

</li></li>

</ol>

<p>接着再建立一个 test2.bat 文件, 内容如下: <br>

```
@<a href="https://ld246.com/member/ECHO" aria-name="ECHO" class="tooltipped__user" target="_blank">ECHO</a> OFF<br>
```

```
IF EXIST %1 TYPE %1<br>
```

```
IF NOT EXIST %1 ECHO %1 does not exist</p>
```

<p>执行:<br>

```
C:>TEST2 AUTOEXEC.BAT<br>
```

该命令运行结果同上。</p>

<p>说明: <br>

(1) IF EXIST 是用来测试文件是否存在的, 格式为<br>

```
IF EXIST [路径 + 文件名] 命令<br>
```

(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。</p>

<ol start="3">

<li>更进一步的, 建立一个名为 TEST3.BAT 的文件, 内容如下: <br>

```
@<a href="https://ld246.com/member/echo" aria-name="echo" class="tooltipped__user" target="_blank">echo</a> off<br>
```

```
IF "%1" == "A" ECHO XIAO<br>
```

```
IF "%2" == "B" ECHO TIAN<br>
```

```
IF "%3" == "C" ECHO XIN</li>
```

</ol>

<p>如果运行: <br>

```
C:>TEST3 A B C<br>
```

屏幕上会显示:<br>

```
XIAO<br>
```

```
TIAN<br>
```

```
XIN</p>
```

<p>如果运行: <br>

```
C:>TEST3 A B<br>
```

屏幕上会显示<br>

```
XIAO<br>
```

```
TIAN</p>
```

<p>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。</p>

<ol start="2">

<li>IF-ERRORLEVEL</li>

</ol>

<p>建立 TEST4.BAT, 内容如下: <br>

```
@<a href="https://ld246.com/member/ECHO" aria-name="ECHO" class="tooltipped__user" target="_blank">ECHO</a> OFF<br>
```

```

XCOPY C:\AUTOEXEC.BAT D:IF ERRORLEVEL 1 ECHO 文件拷贝失败<br>
IF ERRORLEVEL 0 ECHO 成功拷贝文件</p>
<p>然后执行文件:<br>
C:&gt;TEST4</p>
<p>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"。</p>
<p>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：<br>
@<a href="https://ld246.com/member/ECHO" aria-name="ECHO" class="tooltipped__user" t
rget="_blank">ECHO</a> OFF<br>
XCOPY C:\AUTOEXEC.BAT D:<br>
IF ERRORLEVEL 0 ECHO 成功拷贝文件<br>
IF ERRORLEVEL 1 ECHO 未找到拷贝文件<br>
IF ERRORLEVEL 2 ECHO 用户通过 ctrl-c 中止拷贝操作<br>
IF ERRORLEVEL 3 ECHO 预置错误阻止文件拷贝操作<br>
IF ERRORLEVEL 4 ECHO 拷贝过程中写盘错误</p>
<p>无论拷贝是否成功，后面的：</p>
<p>未找到拷贝文件<br>
用户通过 ctrl-c 中止拷贝操作<br>
预置错误阻止文件拷贝操作<br>
拷贝过程中写盘错误</p>
<p>都将显示出来。</p>
<p>以下几个就是几个常用命令的返回值及其代表的意义：<br>
backup<br>
0 备份成功<br>
1 未找到备份文件<br>
2 文件共享冲突阻止备份完成<br>
3 用户用 ctrl-c 中止备份<br>
4 由于致命的错误使备份操作中止</p>
<p>diskcomp<br>
0 盘比较相同<br>
1 盘比较不同<br>
2 用户通过 ctrl-c 中止比较操作<br>
3 由于致命的错误使比较操作中止<br>
4 预置错误中止比较</p>
<p>diskcopy<br>
0 盘拷贝操作成功<br>
1 非致命盘读/写错<br>
2 用户通过 ctrl-c 结束拷贝操作<br>
3 因致命的处理错误使盘拷贝中止<br>
4 预置错误阻止拷贝操作</p>
<p>format<br>
0 格式化成功<br>
3 用户通过 ctrl-c 中止格式化处理<br>
4 因致命的处理错误使格式化中止<br>
5 在提示"proceed with format (y/n) ?"下用户键入 n 结束</p>
<p>xcopy<br>
0 成功拷贝文件<br>
1 未找到拷贝文件<br>
2 用户通过 ctrl-c 中止拷贝操作<br>
4 预置错误阻止文件拷贝操作<br>
5 拷贝过程中写盘错误</p>
<ol start="3">
<li>IF STRING1 == STRING2</li>
</ol>

```

<p>建立 TEST5.BAT, 文件内容如下: <br>

```
@<a href="https://ld246.com/member/echo" aria-name="echo" class="tooltipped__user" target="_blank">echo</a> off<br>IF "%1" == "A" formAT A:</p>
```

<p>执行: <br>

```
C:&gt;TEST5 A<br>
```

屏幕上就出现是否将 A:盘格式化的内容。 </p>

<p>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, 注意不能使用保留号) 括起来。 <br>

如: if [%1]==[A] 或者 if %1\*==A\* </p>

```
<ol start="5">
```

```
<li>GOTO</li>
```

```
</ol>
```

<p>建立 TEST6.BAT, 文件内容如下: <br>

```
@<a href="https://ld246.com/member/ECHO" aria-name="ECHO" class="tooltipped__user" target="_blank">ECHO</a> OFF<br>
```

```
IF EXIST C:\AUTOEXEC.BAT GOTO _COPY<br>
```

```
GOTO _DONE<br>
```

```
:_COPY<br>
```

```
COPY C:\AUTOEXEC.BAT D:<br>
```

```
:_DONE</p>
```

<p>注意: <br>

(1) 标号前是 ASCII 字符的冒号":", 冒号与标号之间不能有空格。 <br>

(2) 标号的命名规则与文件名的命名规则相同。 <br>

(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。 </p>

```
<ol start="6">
```

```
<li>FOR</li>
```

```
</ol>
```

<p>建立 C:\TEST7.BAT, 文件内容如下: <br>

```
@<a href="https://ld246.com/member/ECHO" aria-name="ECHO" class="tooltipped__user" target="_blank">ECHO</a> OFF<br>
```

```
FOR %%C IN (*.BAT *.TXT *.SYS) DO TYPE %%C</p>
```

<p>运行: <br>

```
C:&gt;TEST7</p>
```