



链滴

# crontab 使用

作者: [someone35514](#)

原文链接: <https://ld246.com/article/1544853672847>

来源网站: 链滴

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

## crontab 定时任务

通过 crontab 命令，我们可以在固定的间隔时间执行指定的系统指令或 shell script 脚本。时间间隔的单位可以是分钟、小时、日、月、周及以上的任何组合。这个命令非常适合周期性的日志分析或数据备份等工作。

### 19.1. 命令格式

```
crontab [-u user] file
```

```
crontab [-u user] [-e | -l | -r]
```

### 19.2. 命令参数

- 

- 

- u user: 用来设定某个用户的 crontab 服务;

- 

- file: file 是命令文件的名字,表示将 file 做为 crontab 的任务列表文件并载入 crontab。如果在命令行中没有指定这个文件, crontab 命令将接受标准输入(键盘)上键入的命令,并将它们载入 crontab。

- 

- 

- e: 编辑某个用户的 crontab 文件内容。如果不指定用户,则表示编辑当前用户的 crontab 文件。

- 

- 

- l: 显示某个用户的 crontab 文件内容,如果不指定用户,则表示显示当前用户的 crontab 文件内容。

- 

- 

- r: 从/var/spool/cron 目录中删除某个用户的 crontab 文件,如果不指定用户,则默认删除当前用户的 crontab 文件。

- 

- 

- i: 在删除用户的 crontab 文件时给确认提示。

- 

- 

### 19.3. crontab 的文件格式

分 时 日 月 星期 要运行的命令

- 

- 

- 第 1 列分钟 0~59

- 

- 

- 第 2 列小时 0~23 (0 表示子夜)

- 

- 

- 第 3 列日 1~31

- 

- 

- 第 4 列月 1~12

- 

- 

- 第 5 列星期 0~7 (0 和 7 表示星期天)

- 

- 

- 第 6 列要运行的命令

</li>

</ul>

<h2 id="19-4-常用方法">19.4. 常用方法</h2>

<h3 id="创建一个新的crontab文件">创建一个新的 crontab 文件</h3>

<p>向 cron 进程提交一个 crontab 文件之前，首先要设置环境变量 EDITOR。cron 进程根据它来决定使用哪个编辑器编辑 crontab 文件。99% 的 UNIX 和 LINUX 用户都使用 vi，如果你也是这样，么你就编辑 \$HOME 目录下的 .profile 文件，在其中加入这样一行:</p>

<p>EDITOR=vi; export EDITOR</p>

<p>然后保存并退出。不妨创建一个名为 cron 的文件，其中是用户名，例如， davecron。在该文中加入如下的内容。</p>

<h2 id="-put-your-own-initials-here-echo-the-date-to-the-console-every">(put your own initials here)echo the date to the console every</h2>

<h2 id="15minutes-between-6pm-and-6am">15minutes between 6pm and 6am</h2>

<p>0,15,30,45 18-06 \* \* \* /bin/echo 'date' &gt; /dev/console</p>

<p>保存并退出。注意前面 5 个域用空格分隔。</p>

<p>在上面的例子中，系统将每隔 15 分钟向控制台输出一当前时间。如果系统崩溃或挂起，从最所显示的时间就可以一眼看出系统是什么时间停止工作的。在有些系统中，用 tty1 来表示控制台，以根据实际情况对上面的例子进行相应的修改。为了提交你刚刚创建的 crontab 文件，可以把这个新建的文件作为 cron 命令的参数:</p>

<p>\$ crontab davecron</p>

<p>现在该文件已经提交给 cron 进程，它将每隔 15 分钟运行一次。同时，新建文件的一个副本已被放在 /var/spool/cron 目录中，文件名就是用户名(即 dave)。</p>

<h3 id="列出crontab文件">列出 crontab 文件</h3>

<p>使用 -l 参数列出 crontab 文件:</p>

<p>\$ crontab -l</p>

<p>0,15,30,45 18-06 \* \* \* /bin/echo <code>date</code> &gt; dev/tty1</p>

<p>可以使用这种方法在 \$HOME 目录中对 crontab 文件做一备份:</p>

<p><span class="language-math"> crontab -l &gt; HOME/mycron</span></p>

<p>这样，一旦不小心误删了 crontab 文件，可以用上一节所讲述的方法迅速恢复。</p>

<h3 id="编辑crontab文件">编辑 crontab 文件</h3>

<p>如果希望添加、删除或编辑 crontab 文件中的条目，而 EDITOR 环境变量又设置为 vi，那么就用以 vi 来编辑 crontab 文件:</p>

<p>\$ crontab -e</p>

<p>可以像使用 vi 编辑其他任何文件那样修改 crontab 文件并退出。如果修改了某些条目或添加了的条目，那么在保存该文件时，cron 会对其进行必要的完整性检查。如果其中的某个域出现了超出许范围的值，它会提示你。我们在编辑 crontab 文件时，没准会加入新的条目。例如，加入下面的条:</p>

<h2 id="DT-delete-core-files-at-3-30am-on-1-7-14-21-26-26-days-of-each-month">DT:delete core files,at 3.30am on 1,7,14,21,26,26 days of each month</h2>

<p>30 3 1,7,14,21,26 \* \* /bin/find -name 'core' -exec rm {} ;</p>

<p>保存并退出。</p>

<p>注解</p>

<p>最好在 crontab 文件的每一个条目之上加入一条注释，这样就可以知道它的功能、运行时间，为重要的是，知道这是哪位用户的定时作业。</p>

<h3 id="删除crontab文件">删除 crontab 文件</h3>

<p>\$crontab -r</p>

<h2 id="19-5-使用实例">19.5. 使用实例</h2>

<h3 id="实例1-每1分钟执行一次myCommand">实例 1: 每 1 分钟执行一次 myCommand</h3>

<ul>

<li>

<ul>

<li>

<ul>

<li>



<p>1. 脚本中涉及文件路径时写全局路径; </p>

<p>2. 脚本执行要用到 java 或其他环境变量时, 通过 source 命令引入环境变量, 如:</p>

```
<p>cat start_cbp.sh</p>
```

```
<p>! /bin/sh</p>
```

```
<p>source /etc/profile</p>
```

```
<p>export RUN_CONF=/home/d139/conf/platform/cbp/cbp_jboss.conf</p>
```

```
<p>/usr/local/jboss-4.0.5/bin/run.sh -c mev &&</p>
```

<p>3. 当手动执行脚本 OK, 但是 crontab 死活不执行时, 很可能是环境变量惹的祸, 可尝试在 crontab 中直接引入环境变量解决问题。如:</p>

```
<p>0 * * * * . /etc/profile; /bin/sh /var/www/java/audit_no_count/bin/restart_audit.sh</p>
```

### <p>每条任务调度执行完毕, 系统都会将任务输出信息通过电子邮件的形式发送给当前系统用户, 这日积月累, 日志信息会非常大, 可能会影响系统的正常运行, 因此, 将每条任务进行重定向处理非常重要。例如, 可以在 crontab 文件中设置如下形式, 忽略日志输出:</p> ``` <p>0 */3 * * * /usr/local/apache2/apachectl restart &&/dev/null 2&&&1</p> ``` <p>"/dev/null 2&&&1" 表示先将标准输出重定向到/dev/null, 然后将标准错误重定向到标准输出, 由于标准输出已经重定向到了/dev/null, 因此标准错误也会重定向到/dev/null, 这样日志出问题就解决了。</p> <p>系统级任务调度主要完成系统的一些维护操作, 用户级任务调度主要完成用户自定义的一些任务。可以将用户级任务调度放到系统级任务调度来完成 (不建议这么做), 但是反过来却不行, root 用户任务调度操作可以通过 "crontab -uroot -e" 来设置, 也可以将调度任务直接写入/etc/crontab 文件, 需要注意的是, 如果要定义一个定时重启系统的任务, 就必须将任务放到/etc/crontab 文件, 即使在 root 用户下创建一个定时重启系统的任务也是无效的。</p> <p>新创建的 cron job, 不会马上执行, 至少要过 2 分钟才执行。如果重启 cron 则马上执行。</p> <p>当 crontab 失效时, 可以尝试/etc/init.d/crond restart 解决问题。或者查看日志看某个 job 有没有执行/报错 tail -f /var/log/cron。</p> <p>千万别乱运行 crontab -r。它从 Crontab 目录 (/var/spool/cron) 中删除用户的 Crontab 文件。删除了该用户的所有 crontab 都没了。</p> <p>在 crontab 中 % 是有特殊含义的, 表示换行的意思。如果要用的话必须进行转义 %, 如经常用的 date '+%Y%m%d' 在 crontab 里是不会执行的, 应该换成 date '+%Y%m%d'。</p> <p>更新系统时间时区后需要重启 cron, 在 centos 中服务名为 cron:</p> ``` <p>$service cron restart</p> ``` ``` <p>ubuntu 下启动、停止与重启 cron:</p> ``` ``` <p>$sudo systemctl start crond.service</p> ``` ``` <p>$sudo systemctl stop crond.service</p> ``` ``` <p>$sudo systemctl restart crond.service</p> ``` 原文链接: [crontab 使用](#)