

Linux 技巧
内容

概览

用 cron 和 at 调度作业

如何轻松地管理系统

在指定的时间运行作业



Ian Shields

管理调度作业 2017 年 8 月 27 日发布

配置用户对作业调度的访问

了解更多信息:
系列内容:相关主题
此内容是该系列的一部分: **Linux 技巧**

评论

Linux® 和 UNIX® 系统允许调度任务在以后执行一次，或者重复运行。本文是从 developerWorks [“任务”](#) 摘录的，讲解如何调度作业定期运行，或在指定的时间运行一次。

在 Linux 系统上，许多管理任务必须频繁地定期执行。这些任务包括轮转日志文件以避免装满服务器来执行系统时间同步。上面提到的教程更详细地介绍了这些管理任务。在本文中，学习 Lir 和 anacron 设施以及 crontab 和 at 命令。即使系统常常关机，anacron 也可以帮助调度作业。

以一定的时间间隔运行作业

以一定的时间间隔运行作业需要使用 cron 设施进行管理，它由 crond 守护进程和一组表（描述率）组成。这个守护进程每分钟唤醒一次，并通过检查 crontab 判断需要做什么。用户使用 cr 守护进程常常是在系统启动时由 init 进程启动的。

为了简单，假设希望定期运行清单 1 所示的命令。这个命令实际上只报告日期和时间，其他什么也不做。使用 crontab 设置 cron 作业，而且还可以通过输出看到作业运行的时间。设置 crontab 条目需字符串，所以适合于简单的命令和参数。在这个示例中，将从脚本 /home/ian/mycrontab.sh 运行。这可以减少处理转义字符的工作。

清单 1. 一个简单的命令示例

```

1 [ian@lyrebird ~]$ cat mycronetest.sh
2 #!/bin/bash
3 echo "It is now $(date +%T) on $(date +%A)"
4 [ian@lyrebird ~]$ ./mycronetest.sh
5 It is now 18:37:42 on Friday

```

内容

创建 crontab

概览

使用 `crontab` 命令和 `-e`（表示“edit”）选项创建 crontab。这会打开 vi 编辑器，除非在 `EDITOR` 环境变量中指定了另一种编辑器。

在指定的时间运行作业

每个 crontab 条目包含六个字段：
管理调度的作业

1. 分钟对作业调度的访问

2. 小时
了解更多信息

3. 日
相关主题

4. 月

评论

5. 星期

6. 由 sh 执行的字符串

分钟和小时的范围分别是 0-59 和 0-12，日和月的范围分别是 1-31 和 1-12。星期的范围是 0-6，0 定为 sun、mon、tue 等等。第 6 个字段包含前 5 个字段之后的所有内容，它是要传递给 sh 的行，所以如果要使用 % 或其他任何特殊字符，就要在前面加上反斜线 (\)。第一个 % 之前的所有行都作为标准输入传递。

各个与时间相关的字段可以指定一个单独的值、值的范围（比如 0-10 或 sun-wed）或者以逗号分隔的列表。表 2 给出一个 crontab 条目示例。

清单 2. 一个简单的 crontab 示例

```

1 0,20,40 22-23 * 7 fri-sat /home/ian/mycronetest.sh

```

在这个示例中，我们的命令在 7 月的每个星期五和星期六晚上 10 点到午夜之间的第 0、20、40 分钟运行。有关指定时间的其他方式的细节，参见 `crontab(5)` 的手册页。

输出

您可能想知道对来自命令的输出会如何处理。为使用 cron 而设计的大多数命令会使用 syslog 在 [102 考试准备：管理任务](#)”中的讨论)。但是，定向到 stdout 的输出会通过电子邮件发送给用能产生的输出。

清单 3. 通过电子邮件发送的 cron 输出

```
1  From ian@lyrebird.raleigh.ibm.com  Fri Jul  6 23:00:02 2007
2  Date: Fri, 6 Jul 2007 23:00:01 -0400
3  From: root@lyrebird.raleigh.ibm.com (Cron Daemon)
4  To: ian@lyrebird.raleigh.ibm.com
5  Subject: Cron <ian@lyrebird> /home/ian/mycrontest.sh
6  Content-Type: text/plain; charset=UTF-8
7  Auto-Submitted: auto-generated
8  X-Cron-Env: <SHELL=/bin/sh>
9  X-Cron-Env: <HOME=/home/ian>
10 X-Cron-Env: <PATH=/usr/bin:/bin>
11 X-Cron-Env: <LOGNAME=ian>
12 X-Cron-Env: <USER=ian>
13
14  It is now 23:00:01 on Friday
```

了解更多信息

crontab 存储在哪里？

相关主题

用 `crontab` 命令创建的 crontab 存储在 `/etc/spool/cron` 下面的一个子目录中，这个子目录与创建 crontab 的用户同名，所以上面的 crontab 存储在 `/etc/spool/cron/ian` 中。因此，与 `passwd` 命令一样，`crontab` 命令是一个用根权限运行的 suid 程序。

/etc/crontab

除了 `/var/spool/cron` 中的用户 crontab 文件之外，cron 还会检查 `/etc/crontab` 文件和 `/etc/cron.d` 目录中的文件。在这些系统 crontab 中，在第五个时间字段（星期）和命令之间增加了一个字段。这个字段指定哪个用户应该运行这个命令，一般情况下是根用户。清单 4 给出一个 `/etc/crontab` 文件示例。

清单 4. /etc/crontab

```
1  SHELL=/bin/bash
2  PATH=/sbin:/bin:/usr/sbin:/usr/bin
3  MAILTO=root
4  HOME=/
5
6  # run-parts
7  01 * * * * root run-parts /etc/cron.hourly
8  02 4 * * * root run-parts /etc/cron.daily
9  22 4 * * 0 root run-parts /etc/cron.weekly
10 42 4 1 * * root run-parts /etc/cron.monthly
```

在这个示例中，真正的工作由 `run-parts` 命令执行，它运行 `/etc/cron.hourly`、`/etc/cron.daily` 仅控制执行作业的时间。注意，这里的所有命令都作为根用户运行。还要注意，`crontab` 可以在运行命令之前执行。

anacron

内容

`cron` 适合那些连续运行的系统。对于那些常常不开机的系统，比如笔记本电脑，可以使用另一个概念（“anachronistic cron”）调度每日、每周或每月执行的作业。`anacron` 不处理每小时执行的作业。

以一定的时间间隔运行作业

`anacron` 在 `/var/spool/anacron` 中保留时间戳文件，记录作业运行的时间。当 `anacron` 运行时，它检查时间间隔所需的天数，如果需要，就运行作业。`anacron` 的作业表存储在 `/etc/anacrontab` 有不同。与 `/etc/crontab` 一样，`/etc/anacrontab` 可以包含环境设置。每个作业有四个字段：管理调度的作业

配置周期对作业调度的访问

了解更多信息

3. 作业标识符

相关主题

4. 命令

评论

周期是天数，但是可以指定为 `@monthly`，这确保作业每个月只运行一次（无论这个月中有多少件之后，到实际启动它之前等待的分钟数。可以使用这个设置防止在系统启动时集中执行作业。（/）之外的所有非空白字符。

`/etc/crontab` 和 `/etc/anacrontab` 都通过直接编辑进行更新。不使用 `crontab` 命令更新这些文件

在指定的时间运行作业

有时候，需要只运行作业一次而不是定期运行。为此，应该使用 `at` 命令。要运行的命令是从 - 有使用 `-f`，那么从 `stdin` 读取。`-m` 选项向用户发送邮件，即使命令没有 `stdout`。`-v` 选项显示在输出中。

清单 5 给出一个运行 `mycrontest.sh` 脚本的示例。清单 6 显示在运行作业之后通过邮件发送的比对应的 `cron` 作业输出要简单一些。

清单 5. 使用 `at` 命令

```
1 [ian@lyrebird ~]$ at -f mycrontest.sh -v 10:25
2 Sat Jul  7 10:25:00 2007
3
```

```
4 | job 5 at Sat Jul 7 10:25:00 2007
```

清单 6. 来自 at 的作业输出

```
1 | From ian@lyrebird.raleigh.ibm.com Sat Jul 7 10:25:00 2007
2 | Date: Sat, 7 Jul 2007 10:25:00 -0400
3 | From: Ian Shields <ian@lyrebird.raleigh.ibm.com>
4 | Subject: Output from your job 5
5 | To: ian@lyrebird.raleigh.ibm.com
6 |
7 | It is now 10:25:00 on Saturday
```

TIME

时间的设置可以非常复杂。清单 7 给出几个示例。参见 at 的手册页、`/usr/share/doc/at/timespec` 这样的文件（这个示例中的 3.1.10 是 at 包的版本号）。
在指定的时间运行作业

清单 7. at 命令使用的时间值

管理调度的作业

```
1 | [ian@lyrebird ~]$ at -f mycrontest.sh 10pm tomorrow
2 | job 14 at Sun Jul 8 22:00:00 2007
3 | [ian@lyrebird ~]$ at -f mycrontest.sh 2:00 tuesday
4 | job 15 at Tue Jul 10 02:00:00 2007
5 | [ian@lyrebird ~]$ at -f mycrontest.sh 2:00 july 11
6 | job 16 at Wed Jul 11 02:00:00 2007
7 | [ian@lyrebird ~]$ at -f mycrontest.sh 2:00 next week
8 | job 17 at Sat Jul 14 02:00:00 2007
```

at 命令还有一个 -q 选项。随着队列的增长，作业的 nice 值也会增长。还有一个 batch 命令，它与 at 命令相似，但是作业只在系统负载足够低时运行。这些特性的细节参见手册页。

管理调度的作业

列出调度的作业

可以管理 cron 和 at 作业。使用 crontab 命令和 -l 选项列出 crontab，使用 atq 命令显示用 at 调度的作业。

清单 8. 显示调度的作业

```
1 | [ian@lyrebird ~]$ crontab -l
2 | 0,20,40 22-23 * 7 fri-sat /home/ian/mycrontest.sh
3 | [ian@lyrebird ~]$ atq
4 | 16      Wed Jul 11 02:00:00 2007 a ian
5 | 17      Sat Jul 14 02:00:00 2007 a ian
6 | 14      Sun Jul 8 22:00:00 2007 a ian
7 | 15      Tue Jul 10 02:00:00 2007 a ian
```

如果希望查看 at 调度执行的实际命令，那么可以使用 at 命令并加上 -c 选项和作业号。您会注意到多数环境设置会随调度的作业一起保存。清单 9 给出清单 7 和清单 8 中作业 15 的部分输出。

清单 9. 使用 at -c 并加上作业号

```

1  #!/bin/sh
2  # atrun uid=500 gid=500
3  # mail ian 0
4  umask 2
5  HOSTNAME=lyrebird.raleigh.ibm.com; export HOSTNAME
6  SHELL=/bin/bash; export SHELL
7  HISTSIZE=1000; export HISTSIZE
8  SSH_CLIENT=9.67.219.151\ 3210\ 22; export SSH_CLIENT
9  SSH_TTY=/dev/pts/5; export SSH_TTY
10 USER=ian; export USER
11 ...
12 HOME=/home/ian; export HOME
13 LOGNAME=ian; export LOGNAME
14 ...
15 cd /home/ian || {
16     echo 'Execution directory inaccessible' >&2
17     exit 1
18 }
19 ${SHELL:-/bin/sh} << `(dd if=/dev/urandom count=200 bs=1 \
20     2>/dev/null|LC_ALL=C tr -d -c '[:alnum:]')`
21
22 #!/bin/bash
23 echo "It is now $(date +%T) on $(date +%A)"

```

注意，我们脚本文件的内容已经复制在一个 here-document 中，这个 here-document 将由 SH（如果没有设置 SHELL 变量，就使用 /bin/sh）。关于 here-document 的信息参见教程“[LPI 101 考试命令](#)”。

删除调度的作业

可以使用 cron 命令和 -r 选项删除所有调度的 cron 作业，见清单 10。

清单 10. 显示并删除 cron 作业

```

1  [ian@lyrebird ~]$ crontab -l
2  0,20,40 22-23 * 7 fri-sat /home/ian/mycronetest.sh
3  [ian@lyrebird ~]$ crontab -r
4  [ian@lyrebird ~]$ crontab -l
5  no crontab for ian

```

要删除系统 cron 或 anacron 作业，应该编辑 /etc/crontab、/etc/anacrontab 或者编辑或删除，

可以使用 atrm 命令加作业号删除用 at 命令调度的一个或多个作业。多个作业应该用空白分隔

清单 11. 用 atq 和 atrm 显示并删除作业

```
1 [ian@lyrebird ~]$ atq
2 16      Wed Jul 11 02:00:00 2007 a ian
3 17      Sat Jul 14 02:00:00 2007 a ian
4 14      Sun Jul  8 22:00:00 2007 a ian
5 15      Tue Jul 10 02:00:00 2007 a ian
6 [ian@lyrebird ~]$ atrm 16 14 15
7 [ian@lyrebird ~]$ atq
8 17      Sat Jul 14 02:00:00 2007 a ian
```

概览

配置用户对作业调度的访问

以一定的时间间隔运行作业

如果文件 `/etc/cron.allow` 存在，那么非根用户必须其中列出，才能使用 `crontab` 和 `cron` 设施。但是 `/etc/cron.deny` 存在，那么其中列出的非根用户不能使用 `crontab` 或 `cron` 设施。如果这两级用户使用这个命令。空的 `/etc/cron.deny` 文件允许所有用户使用 `cron` 设施，这是默认情况。

配置用户对作业调度的访问

`/etc/at.allow` 和 `/etc/at.deny` 文件对 `at` 设施起相似的作用。

了解更多信息

相关主题

了解更多信息

评论

如果想了解关于 Linux 管理任务的更多信息，请阅读教程“[LPI 102 考试准备：管理任务](#)”，或[rate 这个页面](#)。

相关主题

- 您可以参阅本文在 developerWorks 全球站点上的 [英文原文](#)。
- 回顾教程“[LPI 102 考试准备：管理任务](#)”（developerWorks，2007 年 7 月），了解关于 Linux 包括用户管理、备份、系统日志和 Network Time Protocol。它是 [LPI 考试准备教程系列](#) 的一部分，帮助您准备系统管理员认证。本教程还引用了这个系列中的两个教程“[LPI 101 考试准备：设备、Linux 文件系统和 Filesystem Hierarchy Standard](#)”。
- [Linux Documentation Project](#) 有很多非常有用的文档，尤其是其中的 HOWTO 指导。
- 在 [developerWorks Linux 专区](#) 中可以找到为 Linux 开发人员准备的更多教程，包括 [Linux 入门](#) 的 Linux 文章和教程。
- 使用 [IBM 试用软件](#) 构建您的下一个 Linux 开发项目，这些软件可以从 developerWorks 直接下载。