

## **cron 与 anacron: 如何在 Linux 中计划任务**

### **内容**

在本篇中，我们会解释 cron 和 anacron，并向你展示如何在 Linux 中设置 anacron。我们也会比较这两个工具。

要[在一个给定时间或者稍后安排一个任务](#)，你可以使用 at 或者 batch 命令，要使命令能够重复运行，你可以使用 cron 以及 anacron 工具。

[cron](#) - 是一个用于运行计划任务如系统备份、更新等的守护进程。它适合在那些 24X7 不间断运行的机器如服务器上运行的计划任务。

命令/脚本被写在 cron 任务脚本中，它是在 crontab 文件中被安排的。系统默认的 crontab 文件是 /etc/crontab，但是每个用户也可以创建自己的 crontab 文件来在特定时间运行用户定义的命令。

要创建一份个人 crontab 文件，只要输入：

```
$ crontab -e
```

### **如何在 Linux 中设置 anacron**

anacron 用于以天为单位的频率运行命令。它的工作与 cron 稍有不同，它假设机器不会一直开机。

cron 也适合在那些不会 24X7 运行如笔记本以及桌面电脑的机器上运行每日、每周以及每月的计划任务（LCTT 译注：不适合按小时、分钟执行任务）。

假设你有一个计划任务（比如备份脚本）要使用 cron 在每天半夜运行，也许你以及睡着，那时你的桌面/笔记本电脑已经关机。你的备份脚本就不会被运行。

然而，如果你使用 anacron，你可以确保在你下次开启桌面/笔记本电脑的时候，备份脚本会被执行。

### **anacron 如何在 Linux 工作**

anacron 任务被列在 /etc/anacrontab 中，任务可以使用下面的格式（anacron 文件中的注释必须以 # 号开始）安排。

```
period delay job-identifier command
```

从上面的格式中：

**period** - 这是任务的频率，以天来指定，或者是 @daily、@weekly、@monthly 代表每天、每周、每月一次。你也可以使用数字：1 - 每天、7 - 每周、30 - 每月，或者 N - 几天。

**delay** - 这是在执行一个任务前等待的分钟数。

**job-id** - 这是写在日志文件中任务的独特名字。

**command** - 这是要执行的命令或 shell 脚本。

要浏览示例文件，输入：

```
$ ls -l /var/spool/anacron/
total 12
-rw----- 1 root root 9 Jun 1 10:25 cron.daily
-rw----- 1 root root 9 May 27 11:01 cron.monthly
-rw----- 1 root root 9 May 30 10:28 cron.weekly
```

这是实际发生的：

anacron 会检查任务是否已经在 **period** 字段指定的时间被执行了。如果没有，则在等待 **delay** 字段中指定的分钟数后，执行 **command** 字段中指定的命令。

一旦任务被执行了，它会使用 **job-id**（时间戳文件名）字段中指定的名称将日期记录在 /var/spool/anacron 目录中的时间戳文件中。

现在让我们看一个例子。这个会每天运行 /home/aaronkilik/bin/backup.sh 脚本：

```
@daily 10 example.daily /bin/bash /home/aaronkilik/bin/backup.sh
```

当机器在 backup.sh 期望被运行时是关机的，**anacron** 会在机器开机十分钟之后运行它，而不用再等待 7 天。

这里有两个你应该理解的 **anacrontab** 文件的重要变量：

**START\_HOURS\_RANGE** - 这个设置任务开始运行的时间范围（也就是任务只在这几个小时内运行）。

**RANDOM\_DELAY** - 这定义添加到用户定义的任务延迟的最大随机延迟（默认为 45）。

这是你的 **anacrontab** 文件可能看上去的样子。

**Anacron – /etc/anacrontab:**

```
# /etc/anacrontab: configuration file for anacron
# See anacron(8) and anacrontab(5) for details.
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
HOME=/root
LOGNAME=root
# These replace cron's entries
```

```
1 5 cron.daily run-parts --report /etc/cron.daily
7 10 cron.weekly run-parts --report /etc/cron.weekly
@monthly 15 cron.monthly run-parts --report /etc/cron.monthly
@daily 10 example.daily /bin/bash /home/aaronkilik/bin/backup.sh
```

下面是 `cron` 以及 `anacron` 的比较，帮助你理解何时用他们其中一个。

<b>cron</b>	<b>anacron</b>
它是守护进程	它不是守护进程
适合服务器	适合桌面/笔记本电脑
可以让你以分钟级运行计划任务	只能让你以天为基础来运行计划任务
关机时不会执行计划任务	如果计划任务到期，机器是关机的，那么它会在机器下次开机后执行计划任务
普通用户和 <code>root</code> 用户都可以使用	只有 <code>root</code> 用户可以使用（使用特定的配置启动普通任务）

`cron` 和 `anacron` 主要的区别在于 `cron` 能在那些持续运行的机器上有效地运行，而 `anacron` 是针对那些会在一天内或者一周内会关机的机器。