

如何在 Linux 启动时自动执行命令或脚本

内容

我一直很好奇，在[启动 Linux 系统并登录](#)的过程中到底发生了什么事情。按下开机键或启动一个虚拟机，你就启动了一系列事件，之后会进入到一个功能完备的系统中，有时，这个过程不到一分钟。当你注销或者关机时，也是这样。

更有意思的是，在系统启动以及用户登录或注销时，还可以让系统执行特定的操作。

本文，我们将探讨一下在 Linux 操作系统中实现这些目标的传统方法。

注意：我们假定使用的是 **Bash** 作为登录及注销的主 Shell。如果你使用的是其他 Shell，那么有些方法可能会无效。如果有其他的疑问，请参考对应的 Shell 文档。

在启动时执行 Linux 脚本

有两种传统的方法可以实现在启动时执行命令或脚本：

方法 #1 - 使用 cron 任务

除了常用格式（分 / 时 / 日 / 月 / 周）外，[cron 调度器](#)还支持 @reboot 指令。这个指令后面的参数是脚本（启动时要执行的那个脚本）的绝对路径。

然而，这种方法需要注意两点：

- a) cron 守护进程必须处于运行状态（通常情况下都会运行），同时
- b) 脚本或 crontab 文件必须包含需要的环境变量（如果有的话，参考 StackOverflow 获取更多详细内容）。

方法 #2 - 使用 /etc/rc.d/rc.local

这个方法对于 systemd-based 发行版 Linux 同样有效。不过，使用这个方法，需要授予 /etc/rc.d/rc.local 文件执行权限：

```
# chmod +x /etc/rc.d/rc.local
```

然后在这个文件底部添加脚本。

下图说明如何分别使用 cron 任务和 rc.local 运行两个示例脚本

（/home/gacanepa/script1.sh 和 /home/gacanepa/script2.sh）。

script1.sh:

```
#!/bin/bash
DATE=$(date +%F %H:%M:%S)
DIR=/home/gacanepa
echo "Current date and time: $DATE" > $DIR/file1.txt
```

script2.sh :

```
#!/bin/bash
SITE="Tecmint.com"
DIR=/home/gacanepa
```

```
echo "$SITE rocks... add us to your bookmarks." > $DIR/file2.txt
```

启动时执行 Linux 脚本

**启动时执行 Linux 脚本 **

记住，一定要提前给两个示例脚本授予执行权限：

```
$ chmod +x /home/gacanepa/script1.sh
```

```
$ chmod +x /home/gacanepa/script2.sh
```

在登录或注销时执行 Linux 脚本

要在登录或注销时执行脚本，分别需要使用 `~.bash_profile` 和 `~.bash_logout` 文件。多数情况下，后者需要手动创建。在每个文件的底部，添加调用脚本代码，如前面例中所示，就可以实现这个功能。

总结

本文主要介绍如何在启动、登录以及注销系统时执行脚本。如果你有其他的方法可以补充，请使用下面的评论表给我们指出，我们期待您的回应！