



# 三件可以用 systemd 做的令人惊讶的事情

作者: [Alan Smithee](#) 译者: [LCTT](#) [Xingyu.Wang](#) | 2023-04-02 13:55 评论: [1](#)

它不仅仅是为了让你的电脑启动得更快。

当 **systemd** 刚问世时，有很多关于它能加快启动时间的消息。这项功能对大多数人都有吸引力（对那些不重启的人来说就不那么重要了），所以在很多方面，这也是它今天仍然拥有的声誉。虽然 **systemd** 确实是在启动过程中并行启动服务起到了作用，但它的作用远不止于此。以下是你可能没有意识到 **systemd** 可以做的三件事，但你应该好好利用。



## 1、简化 Linux ps

如果你曾经使用过 **ps**，甚至只是 **top** 命令，那么你就会知道你的电脑一直都在运行数百个进程。有时，这正是你需要的信息，以便了解你的计算机或其用户在做什么。其他时候，你真正需要的是一个总体的概览。

**systemd-cgtop** 命令提供了一个基于 <sup>(cgroup)</sup> 控制组 任务安排的计算机负载的简单视图。**控制组** 对现代 Linux 很重要，基本上是容器和 **Kubernetes** 的底层支持结构（这也是云计算可以扩展的原因），但它们也是家庭电脑上的有用结构。例如，从 **systemd-cgtop** 的输出中，你可以看到用户进程的负载，而不是系统进程：

```
1. Control Group          Proc#   %CPU   Memory Input/s Output/s
2. /                      183    5.0   1.6G   0B     3.0M
3. user.slice             4      2.8   1.1G   0B     174.7K
4. user.slice/user-1000.slice 4      2.8   968.2M 0B     174.7K
5. system.slice           65     2.2   1.5G   0B     2.8M
```

你也可以只查看你的用户空间进程，或者查看用户空间进程和内核线程。

这绝不是对 [top](#) 或 [ps](#) 的替代，而是从一个不同的、独特的角度来观察你的系统。在运行容器时，它可能是至关重要的，因为容器使用控制组。



## 2、Linux 定时任务

[Cron](#) 是 Linux 的一个经典组件。当你想安排一些事情定期发生时，你会使用 [Cron](#)。它很可靠，而且相当好地集成到你的系统中。

问题是，[Cron](#) 并不了解有些计算机会被关闭。如果你有一个安排在午夜的 [Cron](#) 任务，但你每天在 23:59 关闭你的电脑，那么你的 [Cron](#) 任务就永远不会运行。[Cron](#) 没有任何工具可以检测到一夜之间错过了工作。

作为对这个问题的回答，有一个很好的 [Anacron](#)，但它不像 [Cron](#) 那样集成的好。要让 [Anacron](#) 运行，你需要做很多设置。

第二个选择是 [systemd](#) 计时器。和 [Cron](#) 一样，它也是内置的，可以随时使用。你需要写一个单元文件，这肯定比单行的 [Crontab](#) 条目多，但也很简单。例如，这里有一个单元文件，在开机 30 分钟后运行一个假想的备份脚本，但每天只运行一次。这可以确保我的电脑得到备份，并防止它每天尝试备份超过一次。

```
1. [Unit]
2. Description=Backup
3. Requires=myBackup.service
4.
5. [Timer]
6. OnBootSec=30min
7. OnUnitActiveSec=1d
8.
9. [Install]
10. WantedBy=timers.target
```

当然，你也可以干预并提示运行一个任务。多亏了 [OnUnitActiveSec](#) 指令，[systemd](#) 不会试图运行你手动激活的作业。



## 3、运行 Linux 容器

容器使启动一个复杂的服务变得非常容易。你可以在短短几分钟内运行一个 [Mattermost](#) 或 [Discourse](#) 服务器。在某些情况下，困难的部分是在你运行容器后管理和监控它们。[Podman](#) 使得管理它们变得容易，但是用什么来管理 [Podman](#) 呢？嗯，[你可以使用 systemd](#)。

[Podman](#) 有一个内置的命令来生成单元文件，这样你的容器就可以被 [systemd](#) 管理和监控：

```
1. $ podman generate systemd --new --files --name example_pod
```

然后你所要做的就是启动服务：

```
1. $ systemctl --user start pod-example_pod.service
```

和其他服务一样，[systemd](#) 确保你的容器荚在任何情况下都能运行。它记录问题，你可以用 [journalctl](#) 和其他重要的日志来查看，你也可以用 [systemd-cgtop](#) 在控制组中监控它的活动。

它不是 [Kubernetes 平台](#)，但对于一两个容器来说，你只需要在可靠和可预测的基础上提供服务，[Podman](#) 和 [systemd](#) 是一对很棒的组合。



## 下载 systemd 电子书

systemd 的内容还有很多，你可以从作者 [David Both](#) 的新书《[systemd 实用指南](#)》中了解基础知识，以及很多实用的技巧。

via: <https://opensource.com/article/23/3/3-things-you-didnt-know-systemd-could-do>

作者: [Alan Smithee](#) 选题: [lkxed](#) 译者: [wxy](#) 校对: [wxy](#)

本文由 [LCTT](#) 原创编译, [Linux中国](#) 荣誉推出



最新评论

发表评论

来自辽宁的 Chrome 78.0|GNU/Linux 用户 2023-04-02 15:34赞 回复

systemd 把事情弄得不透明了。破坏了许多快乐。

译自: [opensource.com](#)  
原创: [LCTT https://linux.cn/article-15685-1.html](#)

作者: Alan Smithee  
译者: Xingyu.Wang

本文由 LCTT 原创翻译, [Linux 中国](#) 首发。也想加入译者行列, 为开源做一些自己的贡献么? 欢迎加入 LCTT! 翻译工作和译文发表仅用于学习和交流目的, 翻译工作遵照 [CC-BY-SA](#) 协议规定, 如果我们的工作有侵犯到您的权益, 请及时联系我们。  
[欢迎遵照 CC-BY-SA 协议规定 转载](#), 敬请在正文中标注并保留原文/译文链接和作者/译者等信息。  
文章仅代表作者的知识 and 看法, 如有不同观点, 请楼下排队吐槽 :D

★📄  
上一篇: 使用 [Terraform](#) 来管理 [OpenStack](#) 集群  
下一篇: 如何在 [RHEL 9](#) 上配置 [DHCP](#) 服务器

LCTT 译者



Xingyu.Wang

共计翻译: **994.0** 篇 | 共计贡献: **3307** 天  
贡献时间: 2014-07-25 -> 2023-08-14  
访问我的 [LCTT 主页](#) | 在 [GitHub](#) 上关注我

相关阅读

 Systemd

硬核观察 #500 谷歌称其已清楚说明 [Chrome](#) 隐

2021-12-29

Linux 启动性能分析

2022-04-12

硬核观察 #652 首个性能超百亿亿次的超算夺得

2022-05-30

Peppermint OS 现在也提供无 [systemd](#) 的

2022-08-07

systemd 已可用于 [WSL](#)

2022-09-25