

# 《代码英雄》第三季（7）：与机器对话

代码英雄讲述了开发人员、程序员、黑客、极客和开源反叛者如何彻底改变技术前景的真实史诗。

## 什么是《代码英雄》

Command Line Heroes

代码英雄是世界领先的企业开源软件解决方案供应商红帽（Red Hat）精心制作的原创音频播客，讲述开发人员、程序员、黑客、极客和开源反叛者如何彻底改变技术前景的真实史诗。该音频博客邀请到了谷歌、NASA 等重量级企业的众多技术大牛共同讲述开源、操作系统、容器、DevOps、混合云等发展过程中的动人故事。

本文是《[代码英雄](#)》系列播客[《代码英雄》第三季（7）：与机器对话](#)的[音频](#)脚本。

导读：创造一台会思考的机器在 20 世纪 50 年代似乎是科幻小说。但 John McCarthy 决定把它变成现实。他从一种叫做 LISP 的语言开始。Colin Garvey 介绍了 McCarthy 是如何创造出第一种人工智能语言的。Sam Williams 介绍了早期人们对思考机器的兴趣是如何从学术界蔓延到商业界的，以及在某些项目没有兑现承诺之后，漫长的人工智能寒冬是如何最终到来的。Ulrich Drepper 解释说，人工智能的梦想超出了当时硬件所能提供的范围。

但硬件每天都在变得更强大。Chris Nicholson 指出，今天的机器有足够的处理能力来处理人工智能的资源需求——以至于我们正处于人工智能研发的革命性复苏之中。最后，Rachel Thomas 确定了 LISP 之外的人工智能语言——证明了人工智能现在正准备做的不同类型的任务。

## 00:00:05 - 播音员：

卡斯帕罗夫非常慌乱。虽然已经拼尽全力，但是他意识到，他已经输了。

## 00:00:12 - Saron Yitbarek:

1997 年的春天，国际象棋冠军加里·卡斯帕罗夫输给了计算机程序“深蓝”。对于机器智能来说，这是历史上的关键时刻。对于某些人来说，这是一次生存危机，是对人类智慧至上的挑战。但是，对于世界上的技术专家来说，这是另一种意味上的里程碑，是人工智能（AI）领域的一次飞跃。这表明，想要真正的智慧机器诞生，或许不是太过于疯狂的梦想。

## 00:00:47 - 播音员：

去赋予一台机器思考的能力仍然还是梦想，距离实现梦想还需要很多年的努力。此外，还需要做出不少惊人的突破。

## 00:00:56 - Saron Yitbarek:

我们是如何抵达这一步的？是什么导致了卡斯帕罗夫那场著名的失败？我们从哪里来，又该到哪里去？我是 Saron Yitbarek，这里是《代码英雄》，一档来自红帽的原创播客节目。在本季，我们都在探索编程语言的奥秘，揭开他们的历史和发展潜力。这一期让我们关注人工智能。你会使用什么语言，来让你的机器拥有自己的思想呢？我们的编程语言如何帮助我们抵达“深蓝时刻”，甚至是更远的地方？什么样的编程语言能最适配会思考的机器？这是我们半个多世纪以来一直在尝试解决的问题。因此，我们的故事要追溯到 1940 年代，那时，人工智能这个词汇还没有被创造出来。

## 00:01:59:

回顾二战结束是如何结束的，我们有这样一种感觉，那就是技术帮助同盟国赢得了战争。人们乐观地认为，科技可以成就一切。整整一代人都相信计算机的力量。在这一代人中诞生了人工智能的教父，一位出色的数学家——约翰·麦卡锡，他从根本上改变了我们与机器交谈

的方式。我想知道他的思想的起源，以及范式的转变是如何发生的。我的意思是，对于初学者来说，当麦卡锡和他的同事们在想象智能机器的未来时，他们到底想象了什么？

### 00:02:43 - Colin Garvey:

哇，这是一个好问题。

### 00:02:46 - Saron Yitbarek:

我和 Colin Garvey 聊了聊。他是伦斯勒理工学院科学与技术研究系的历史学家。以下是我们的一些聊天内容：

### 00:02:58 - Colin Garvey:

麦卡锡对提出 AI 究竟是什么这件事情非常谨慎。不过，举个例子来说，他可能在他最著名但并未实现的程序中描述过（这只是一个思路），人工智能是一个接受建议者。接受建议者这个说法由麦卡锡于 1960 年的名为《具有常识的程序》的论文中提出的。在开始的时候，你觉得接受建议者可能是一个会学习的机器人，这是它被发明的本意，它算是一个可以接受建议的家用机器人。你说：“不，你这样做是错误的，要那样做。”它就会明白你的意思。

### 00:03:44 - Saron Yitbarek:

那很有帮助。

### 00:03:45 - Colin Garvey:

可能会有帮助。那时候接受建议者要实现的目标是，从办公室导航到机场。换句话说，将从办公室去机场这个建议，正式化成接受建议者机器人能够接受的程序以达成目的，得出从办公室前往机场的一系列程序。它把人类的语言正式化为一系列的合乎逻辑的陈述，即需要根据现在的情况进行合适的抉择，以达到把当前情况转变为理想情况的目标。他将这些想法伪代码化，这实际上是 LISP 的起源。然后在接下来的几年里，LISP 作为他想法的实现，或者说实现他想法的语言出现了。

### 00:04:39 - Saron Yitbarek:

麦卡锡的 LISP 语言改变了游戏规则。它不仅能帮助机器理解命令，还能帮助机器理解一些常识性的逻辑。麦卡锡发现，他可以编写条件表达式，即体现规则的代码，而不是仅仅是直白的命令。实际上，LISP 有一堆重大的编程突破，条件表达式、垃圾回收、递归等等。LISP 在代码和数据中都使用了列表。这是一个简单但具有深远影响的更改。这是语言范式的转变，所有这些为麦卡锡本人称之为“人工智能”的整个领域打开了大门。想象一下，在你和机器说话的时候，并不需要把每一个细节都给编写好。你可以让那台机器自己来进行推断和推理。麦卡锡希望通过他的 LISP 语言，给机器一种智慧。好了，回到我和 Colin Garvey 的聊天。

### 00:05:41 - Colin Garvey:

LISP 是人们编写高级计算机语言的尝试的过程中，诞生的精华。

### 00:05:47 - Saron Yitbarek:

这很有意思。因为我的下一个问题是 LISP 和人工智能之间的关系。接受建议者第一次描述了 AI 的功能，它听起来也像是 LISP 的开始。能跟我聊聊更多 LISP 和人工智能之间的关系吗？

### 00:06:04 - Colin Garvey:

当然。这些早期的人工智能研究人员面临着一件事，那就是他们还在使用打孔卡进行编程。也许这些早期的编程人员，知道如何进行机器码级别的编程，这是非常困难且耗时的工作。因此，你需要一个更高级的编程语言，以便你可以用更加接近人类语言的方式编程。所以，像 LISP 这样，用列表的形式给出指令的语言——这也是其名称的来源，基于列表的处理（LCTT 译注，LISP 全称：“LISt Processing”）。从某种意义上来说，这样用列表给出的指令更加接近于人类语言，因为他们基本上都可以用通顺的逻辑去理解。因此，如果你可以读懂形式逻辑，那么基本上可以看明白 LISP 程序，或者任何基于逻辑的编程语言的程序，并对于代码中发生的事情有着更好的了解。

### 00:07:10 - Saron Yitbarek:

所以 LISP 真的帮了大忙.....嗯，我的意思是它帮助我们将我们的想法给予人工智能，让我们能够朝着实现人工智能努力，有朝一日总能达成这个梦想。不过这也让我想知道在那个时期，“智能”到底意味着什么？所以，如果我们回到 50 年代，在那个时间点上，什么是“智能”？人们是如何定义它的？因为 LISP 最初为 IBM 704 开发的，它每次做的其实只是一件事。看起来并不是很“智能”。当时的人们是如何看待“智能”的呢？

#### **00:07:43 - Colin Garvey:**

有关“智能”的定义，这自然非常有争议的。就我个人，从一个普通大众的角度而言，他们的概念是非常狭窄的。但是在当时，广泛认可的定义是：能够执行被认为有智力的举动的能力。但持这个观点的这些人实际上是数学家、逻辑学家和计算机程序员。我的意思是，说句粗俗的话，下棋的能力被认为是智力的明确标志。这一代早期的 AI 人更喜欢回避这个问题，并说，“嗯，哲学家们还没有就什么是‘智力’达成一致，但如果我们制造出一台能下棋的电脑，我想我们都同意这是‘智能’。”

#### **00:08:40 - Saron Yitbarek:**

不管怎样，这是一个起点。就像婴儿的第一步。

#### **00:08:45 - Colin Garvey:**

麦卡锡有一个梦想，让机器可以具有常识，并像人一样聪明。本质上来说，你可以和它们交谈。他开始创造一种程序语言来实现这个梦想。这就是 LISP。它模仿了人类思想的某些方面，尤其是逻辑思维能力，并使得利用计算机放大或扩展这些思想特征成为可能。所以从数学家的角度来看，他已经在实现智能机械的道路上走得很远了。

#### **00:09:32 - Saron Yitbarek:**

Colin Garvey 是伦斯勒理工学院科学技术研究系的历史学家。

随着 LISP 的推进，发展人工智能的新机会开始出现，LISP 是这一新领域的标准语言。麦卡锡离开麻省理工学院去斯坦福大学工作后，麻省理工学院的其他工程师继续从事 LISP 工作。他们把麦卡锡的语言

改进成一种叫做 **Maclisp** 的方言版本，甚至开发了 **LISP** 操作系统。看起来麦卡锡对人工智能未来的梦想正在成为现实。1968 年，麦卡锡甚至和苏格兰国际象棋大师大卫·利维打赌。麦卡锡打赌，10 年后，电脑将能够在国际象棋比赛中赢过利维。但是，如果事情进展得那么顺利，那就不是《代码英雄》的故事了。那时候麦卡锡还不知道，人工智能的冬天就要来了。

#### **00:10:45 - Saron Yitbarek:**

当麻省理工学院的那个基于 **LISP** 的操作系统，从学院衍生到公司后，情况开始发生变化。其中的一家公司 **Symbolix** 推出了 **LISP** 机器，甚至从 MIT 的 **AI** 实验室雇用了 14 名员工，而且更不切实际的研究不再是重点。

#### **00:11:25 - Sam Williams:**

这些公司原本都来自 **AI** 实验室。

#### **00:11:31 - Saron Yitbarek:**

这是记者 **Sam Williams**，他写了一本名为《**Arguing AI**》的书。

#### **00:11:36 - Sam Williams:**

**Symbolix** 可能是最突出的一个。它得到了最多的关注、最多的风险投资。约翰·麦卡锡和他的门徒们推动了所有的创新。我想，这只是一个案例。在 70 年代末期，人们普遍认为：“好吧，我们已经在学术界做了我们能够做到的事情，让我们在商界脱颖而出，让私营企业为我们投资。”

#### **00:12:01 - Saron Yitbarek:**

这在当时的环境下可以理解。那时 **AI** 似乎即将起飞，有利可图。

#### **00:12:07 - Sam Williams:**

那时候的算力便宜到中型企业也可以购买。因此，我认为许多公司看到了一个机会：“那里有一个富有潜力的市场，我们可以进入这个市

场，我们可以把关于人工智能的产出卖给那些想要具有全球竞争力的企业。”他们让投资者为这个想法大肆投资。在 80 年代初期，大量的资金涌入该领域。

**00:12:29 - Saron Yitbarek:**

我们可以讨论一下为什么这个热潮后来干涸了。但是，我现在可以肯定的一点是，事情已经被大肆炒作了。

**00:12:39 - Sam Williams:**

大量资金流入。人们认为市场非常成熟。

**00:12:44 - Saron Yitbarek:**

从一定来说，大量热钱涌入人工智能领域造成的繁荣，就像世纪之交时网络泡沫的前奏。

**00:12:52 - Sam Williams:**

这在技术发展史上并不少见，人们过度投资，但最终公司未能兑现他们的希望。所以，资金供应陷入困境。公司必须通过老式的方式赚钱——去拉客户，去生产东西。

**00:13:08 - Saron Yitbarek:**

所以那时，麦卡锡的梦想轰然崩溃，回归了现实。Williams 描述了随后的人工智能冬天。

**00:13:16 - Sam Williams:**

你会看到像 Symbolix 这样的公司市值上升到 1 亿美元，然后短短几年后就申请破产保护。在那时候有很多这样的人：“在 10 年内，我们将做到这一点，在 5 年内，你会看到这个。5 年后，你就会看到这个。”这些家伙是投机者，或者是想要从五角大楼或任何其他地方获取更多资金的学者。因为这种危险的炒作，过度炒作，让人工智能走上歧途。

**00:13:48 - Saron Yitbarek:**

人们对于人工智能的兴趣崩塌了。基础设施不到位，或许计算速度也不够高，这使一切努力都变成了徒然。但无论如何，这之中的指导思想是正确的，它只是生在了错误的时代。到了 80 年代末，人工智能研究日渐减少。

#### **00:14:06 - Sam Williams:**

我认为人们脱离人工智能这个领域的原因是，它失去了最初吸引他们进入其中时的冒险性。它曾处于技术领域的最前沿。在西海岸，很多人的精力被吸引到制造个人电脑上。从 40 年后的今天看来，试着让电脑进入美国每个家庭，比帮助财富 1000 强的企业做出更好的决策，是一个更引人注目的概念。

#### **00:14:39 - Saron Yitbarek:**

Sam Williams 是《Arguing AI》的作者。我们在这里描述的是一个计算机编程的世界，它被一分为二：一方面是约翰·麦卡锡这样的人，他们试图使用 LISP 等新型语言来探究智能的本质；另一方面，也不可避免地存在着另一群人，他们更专注于解决实际问题，试图让那些愿意为其服务付款的企业轻松一些。现实情况是，世界上的那些约翰·麦卡锡们，那些抽象的梦想家们，并没有太多机会在人工智能的可能性上进行深入探索。他们所拥有的设备甚至还比不上我们今天使用的手机。他们的愿景对于手头的硬件现实来说，真的太大了。

#### **00:15:36 - Ulrich Drepper:**

当初，我们在轰轰作响的庞大计算机上工作，和数不清的人、数不清的工作组抢夺它有限的性能。

#### **00:15:46 - Saron Yitbarek:**

这是红帽的杰出工程师 Ulrich Drepper。他解释了早期的人工智能梦想之所以失败，只是因为缺少成功所需的基本工具。

#### **00:15:55 - Ulrich Drepper:**

我们有 PDP 10，PDP 11 之类的计算机（LCTT 译注：Digital Equipment Corporation 公司的大型计算机产品），但每一台都可能



有另外 20 个人在同时使用。即使没有这些人将计算力分流走，它们也是真正的低能力机器，其性能不足以在支持 LISP 系统的必要功能的同时，运行一套机能齐全的用户界面。

**00:16:19 - Saron Yitbarek:**

随着时间的推移，新开发的硬件对 LISP 也不再友好。

**00:16:26 - Ulrich Drepper:**

我们不仅走了台式机的道路，也走了服务器的道路。在我们今天已经习惯的处理器类型中，我们的实现的一切都是基于 8 位的处理器，而我们可能有了 16 位处理器、32 位处理器之类的。想要在这样的机器上高效实现 LISP 系统，就必须绕不少圈才能成功。

**00:16:56 - Saron Yitbarek:**

因此，计算世界的外部现实对这个脆弱的人工智能领域产生了意想不到的抑制作用。

**00:17:06 - Ulrich Drepper:**

和大厂商生产的 CPU 相比，定制硬件的成本极为高昂，其发展也极为缓慢。

**00:17:21 - Saron Yitbarek:**

通过创建自定义的硬件解决方案，继续使用 LISP 朝着 AI 梦想前进并非不可能。但是，事实是，令人兴奋的新硬件将很多人的注意力推向了其他工作领域。

**00:17:35 - Ulrich Drepper:**

你只需要等待处理器的下一个版本，那个普通的处理器获得的改进，比你自已开发硬件求解，比如说实现一个 LISP 系统，所能达到的效果和收获要大得多。最终，硬件变得越来越复杂，自行定制硬件成为了一件近乎于不可能的事。

**00:18:03 - Saron Yitbarek:**

那么，约翰·麦卡锡的远景梦想是慢慢死去，还是只在那漫长的人工智能冬天里沉睡，等待时机？在 21 世纪的第一个十年里，约翰·麦卡锡的语言 LISP 的使用者不断地流失。但是，在同一时间，发生了一件惊人的事。尽管 LISP 本身逐渐消逝，与这门语言相连的梦想却被重新点燃了。人工智能的梦想重新活了起来。

**00:18:34 - Chris Nicholson:**

我想说的是，这次不一样。

**00:18:36 - Saron Yitbarek:**

这是 Chris Nicholson，开源 AI 软件公司 SkyMind 的创始人兼 CEO。Chris 认为，在经历了大起大落之后，人工智能或许终于要进入一个可持续发展的长夏了。

**00:18:52 - Chris Nicholson:**

我们正处在人工智能的夏天，许多人都对 AI 可能的应用而激动不已。但是，这种兴奋是基于研究的真正进展，而这些进展是基于其他领域的进展。我们的硬件比当初好了太多太多，人们能用 GPU 替代 CPU，也能在大规模集群中使用这些 GPU 来训练巨大的机器学习模型，从而产生真正准确的预测。为了做到这一点，他们也使用了前所未有的巨大数据集，这样的数据量在 80 年代、90 年代或者之前都是闻所未闻、见所未见的。

**00:19:30 - Saron Yitbarek:**

对于 AI 语言，巨大的数据集的兴起是关键，因为它们代表了我们理解智能的方式的关键变化。AI 研究的重点转移到了深度学习方面，而深度学习似乎成为了 AI 研究的全部。

**00:19:47 - Chris Nicholson:**

LISP 是为了操纵符号而设计的。在 LISP 诞生的时代，AI 意味着符号推理。当时，人们认为人类的思维 and 智力本身就是符号的操作。现代的人工智能，也就是让我们如此着迷的人工智能，远不止于此。现在的 AI，就机器学习而言，是一种可以从非结构化数据中学习的大型数

据处理机器。因此，你可以让机器学习算法一点一滴地拾起文本和图像，在这个过程中，随着时间的推移，这些算法将变得越来越智能。LISP 不是那样设计的。它被设计为仅在人工干预的基础上发展。然而，对于现在的机器学习算法而言，其中的学习部分意味着它们会根据所接触的数据进行自我调整。

### **00:20:40 - Saron Yitbarek:**

这就是让机器以它们自己的方式实现智能化。但是，你知道，不仅仅是我们的机器在学习以强大的新方式思考；开源工具也让人们提高了自己的水平。克里斯认为，开源的开发方式对防止另一个人工智能冬天的到来有很大的帮助。

### **00:21:03 - Chris Nicholson:**

人工智能冬天发生的一个原因是，当创意在放缓时，网络就会崩溃，资金就会枯竭。有了现在这些免费的开源工具，我们看到的不是创意的放缓，而是一种加速。创意得以被及时地测试，得以被迅速地分享。所以，这不仅仅是工具，而是预先训练好的机器学习模型，一个研究小组可以与另一个研究小组分享，或者与广大公众分享。而创意本身也被发表在 **ARXIV** 这样的平台上。**ARXIV** 是由康奈尔大学主办的。所有这些因素汇合在一起，加快了创意流动的速度。对我来说，这是对人工智能冬天的一种保险。这些因素的交汇使得人工智能成为一个超级令人兴奋领域，它们意味着人工智能的发展速度比炒作更快。

### **00:22:03 - Saron Yitbarek:**

Chris Nicholson 是 SkyMind 的创始人兼 CEO。

所以，开源能将人工智能领域从旧的专有模式所催生的寒冬中解冻出来。AI 的新时代伴随着 GPU 的巨大改进而到来。与之一并到来的还有算法上的突破，它帮助我们训练 AI 学习数据中的模式。然后，数据本身，海量的数据集正在为今天的 AI 提供动力。这一点，比任何事情都重要，这就是为什么新语言得以继续 LISP 未能完成的征途。我们向 Fast AI 的联合创始人 Rachel Thomas 询问，哪些语言适合下一代 AI。

### **00:22:50 - Rachel Thomas:**

我们目前正在探索将 Swift 用于 TensorFlow 的可能性。Swift 是 Chris Lattner 开发的一种语言，我知道也有人在使用 Julia，他们正促进着 Julia 的开发与发展。但是，就目前而言，Python 在深度学习领域有着无可比拟的巨大优势。

### **00:23:03 - Saron Yitbarek:**

这是因为 Python 非常适合处理大型数据集。

### **00:23:08 - Rachel Thomas:**

我认为过去 20 年中发生的最重要的改变是，数据量的增长和人们对于数据的越发重视。我认为，数十年前有关 AI 的许多早期工作更多地围绕一种符号系统，即某种抽象符号系统。我要说的是，目前 AI 领域中的许多工作都围绕着数据展开，例如识别图像里的某样东西，或者辨认一条影评是褒还是贬。这些都是基于数据的问题。所以，我认为，Python 成为赢家——至少是早期的赢家——的原因之一，是因为 NumPy、SciPy 和 pandas 的生态系统，以及 Python 中所有数据处理用的库都足够完善。

### **00:24:02 - Saron Yitbarek:**

在约翰·麦卡锡的团队还在麻省理工学院工作时，人工智能事业确实只有世界级的精英学者才能参与。但是自那时以来，语言和计算环境已经发生了巨大的发展，时至今日，每个人都能参与到 AI 的领域中来。我们在本季的第 1 期中听说过，Python 是许多初学者的第一语言。它使新手程序员能够加入 AI 的世界。

### **00:24:29 - Rachel Thomas:**

我只想告诉大家，实际上你只需要一年的编码经验。你不需要诸如“鬼才”这样的名号，也不必具有真正的声望或权威。但是，我们需要来自各个背景的人。实际上，不同背景的人才能够为这个领域提供许多新鲜的创意，你的创意也是我们所需要的。

### **00:24:52 - Saron Yitbarek:**

Rachel Thomas 是 Fast AI 的联合创始人，也是旧金山大学数据研究所的教授。我喜欢她为我们提供的信息。

AI 的故事是发现截然不同的人们之间的共同语言的故事。这是我们真正的任务，为充满 AI 的世界开发语言解决方案，开发能使所有人共同努力以实现下一个突破的解决方案。

**00:25:22 - 播音员 1:**

各位，你们看到了今天在这里创造的历史。AlphaGo 以伟大的风格赢得了比赛。

**00:25:32 - 播音员 2:**

是的，它甚至向我们展示了它的代码。

**00:25:34 - 发言者 7:**

甚至向我们展示了它的代码。我想要祝贺这个计划。

**00:25:40 - Saron Yitbarek:**

2015 年，就在加里·卡斯帕罗夫<sup>Garry Kasparov</sup>在国际象棋比赛中输给深蓝的几十年后，围棋世界冠军李世乭被谷歌的 AlphaGo 击败。深蓝显示出的智力威力大半依托于蛮力，而 AlphaGo 的胜利之所以让人感到惊讶，是因为它依靠神经网络和强化学习赢得了那场比赛。换句话说，AlphaGo 已经成功地朝着真正的智慧迈出了下一步。如果没有开源对人工智能的推动，这一切都不可能发生。新的语言正朝着麦卡锡的梦想而逐渐发展。我们的语言正越来越适合于最大限度地提高机器的智能。好消息是，在开源世界中，我们可以共同实现这份梦想，共同应对它所带来的挑战。

**00:26:39 - Saron Yitbarek:**

《代码英雄》是红帽的原创播客节目。如果你希望更深入的了解 LISP 或者人工智能，你可以访问节目的网站 [redhat.com/commandlineheroes](http://redhat.com/commandlineheroes)。在这里，你也能看到每一集的额外内容。下一期将是本季的最终集。在下一期中，我们将对一门重量级

语言进行探究，并揭露与之有关的许多惊人事实；这门语言对本季中迄今为止讨论过的几乎所有语言都产生过影响。这是我们对 C 的深入研究。我是 Saron Yitbarek，下期之前，编程不止。

## 什么是 LCTT SIG 和 LCTT LCRH SIG

LCTT SIG 是 LCTT Special Interest Group 特别兴趣小组，LCTT SIG 是针对特定领域、特定内容的翻译小组，翻译组成员将遵循 LCTT 流程和规范，参与翻译，并获得相应的奖励。LCRH SIG 是 LCTT 联合红帽（Red Hat）发起的 SIG，当前专注任务是《代码英雄》系列播客的脚本汉化，已有数十位贡献者加入。敬请每周三、周五期待经过我们精心翻译、校对和发布的译文。

欢迎[加入 LCRH SIG](#)一同参与贡献，并领取红帽（Red Hat）和我们联合颁发的专属贡献者证书。

---

via: <https://www.redhat.com/en/command-line-heroes/season-3/talking-to-machines>

作者: [Red Hat](#) 选题: [bestony](#) 译者: [bestony](#) 校对: [acyanbird](#), [Northurland](#), [wxy](#)

本文由 [LCRH](#) 原创编译, [Linux中国](#) 荣誉推出