

克里斯·安德森：有件事情很奇怪。你的软件，Linux，运行于数百万计算机中，可以说是它驱动了网络。我想，全世界正在使用 将近15亿台的安卓设备。每一台里都有你的软件。这非常了不起。你肯定有一个特别高大上的 总部来实现这一切。我以前真这么想——直到我看到这张照片，我被震惊了。真的惊了——这就是Linux的总部。

Chris Anderson: This is such a strange thing. Your software, Linux, is in millions of computers, it probably powers much of the Internet. And I think that there are, like, a billion and a half active Android devices out there. Your software is in every single one of them. It's kind of amazing. You must have some amazing software headquarters driving all this. That's what I thought -- and I was shocked when I saw a picture of it. I mean, this is -- this is the Linux world headquarters.

(笑声)

(Laughter)

(掌声)

(Applause)

林纳斯·托瓦兹：好吧，让大家失望了。我不得不说，这张照片里最有趣的地方，被最多的人问起的地方，就是那张会动的办公桌。这是我办公室最有趣的部分 虽然我现在不怎么用它了。我觉得两者是有联系的。

Linus Torvalds: It really doesn't look like much. And I have to say, the most interesting part in this picture, that people mostly react to, is the walking desk. It is the most interesting part in my office and I'm not actually using it anymore. And I think the two things are related.

我的工作方式..... 我不想要有太多的外来刺激。大家能看到，我的墙是淡绿色的。据说精神病医院的墙就是这种颜色。

The way I work is ... I want to not have external stimulation. You can kind of see, on the walls are this light green. I'm told that at mental institutions they use that on the walls.

(笑声)

(Laughter)

这种颜色能让你平静，不会刺激到你。

It's like a calming color, it's not something that really stimulates you.

其实大家看不到我的电脑（主机），只能看到显示器，我对电脑最主要的要求——并不是性能强劲，尽管我也爱——就是它必须没有任何声音。我认识一些在谷歌工作的人，他们在家都有小型数据中心，我可没这习惯。我的办公室也许是你见过最无聊的一个。我喜欢独自一人，享受安静。如果有猫进来，它会坐在我大腿上。我喜欢听猫打呼噜，而不是听特烦的电脑风扇的声音。

What you can't see is the computer here, you only see the screen, but the main thing I worry about in my computer is -- it doesn't have to be big and powerful, although I like that -- it really has to be completely silent. I know people who work for Google and they have their own small data center at home, and I don't do that. My office is the most boring office you'll ever see. And I sit there alone in the quiet. If the cat comes up, it sits in my lap. And I want to hear the cat purring, not the sound of the fans in the computer.

CA：太了不起了，你的这种工作方式，才能让你运行如此庞大的一个技术帝国——确实是一个帝国——这是开源（软件）强大力量的一个最好例证。

CA: So this is astonishing, because working this way, you're able to run this vast technology empire -- it is an empire -- so that's an amazing testament to the power of open source.

请告诉我们你是如何理解开源的， 和它又是如何一步一步发展成Linux的。

Tell us how you got to understand open source and how it lead to the development of Linux.

LT: 我现在还是独自工作。 真的，我独自一人在家工作， 经常穿着浴袍干活。 当一个摄影师来的时候，我才会穿戴整齐， 所以（这张照片里）我是穿着衣服的。

LT: I mean, I still work alone. Really -- I work alone in my house, often in my bathrobe. When a photographer shows up, I dress up, so I have clothes on.

(笑声)

(Laughter)

我一直以来就是这样工作的。 Linux也是这样诞生的。 Linux并不是一个合作的产物。 它是我一系列项目中的一个， 纯粹出于自己当时的需要， 部分原因是我需要得到结果， 但更重要的原因是我享受编程。 这段旅程的终点， 在25年后的今天，我们仍未达到。 当年我只是想做一个 完全属于自己的项目， 我压根就没想过开源这件事。

And that's how I've always worked. I mean, this was how I started Linux, too. I did not start Linux as a collaborative project. I started it as one in a series of many projects I had done at the time for myself, partly because I needed the end result, but even more because I just enjoyed programming. So it was about the end of the journey, which, 25 years later, we still have not reached. But it was really about the fact that I was looking for a project on my own and there was no open source, really, on my radar at all.

但在那之后， 随着项目越来越大， 你会开始想让别人知道。 感觉就像“哇，快来看看我的成果！” 但说实话，最开始并没这么好。 我把它放到网上供人下载， 当时甚至还不是开源的。 当然，所有代码都是开放的， 但我并没有想过 要用我们今天所熟知的开源方法 来改进它。 我当时想的是，“看，这是我折腾了半年的成果， 我想听听大家的意见。”

And what happened is ... the project grows and becomes something you want to show off to people. Really, this is more of a, "Wow, look at what I did!" And trust me -- it was not that great back then. I made it publicly available, and it wasn't even open source at that point. At that point it was source that was open, but there was no intention behind using the kind of open-source methodology that we think of today to improve it. It was more like, "Look, I've been working on this for half a year, I'd love to have comments."

有人来找过我。 在赫尔辛基大学， 我有个朋友是做开源软件的—— 当年还叫“免费软件”—— 他让我了解到，“嘿， 你可以用那些已有的开源许可证。” 我考虑了一段时间。 我其实有点担心商业利益牵涉其中。 我想，很多人一开始会跟我一样， 担心自己的工作成果会被人利用，不是吗？ 于是我决定，“不管了。” 然后.....

And other people approached me. At the University of Helsinki, I had a friend who was one of the open source -- it was called mainly "free software" back then -- and he actually introduced me to the notion that, hey, you can use these open-source licenses that had been around. And I thought about it for a while. I was actually worried about the whole commercial interests coming in. I mean, that's one of the worries I think most people who start out have, is that they worry about somebody taking advantage of their work, right? And I decided, "What the hell?" And --

CA: 然后某一天, 一些人对你的代码提出了些建议, 你觉得 “哇哦, 真是不错, 我都没想到这一点。对改进软件很有帮助。”

CA: And then at some point, someone contributed some code that you thought, "Wow, that really is interesting, I would not have thought of that. This could actually improve this."

LT: 其实最开始没有多少人直接改代码, 而是提出了很多想法。让别人看一下你所做的项目——我想在其他行业也是这样——但在编程的时候就更是如此——有人对你的代码感兴趣, 仔细研究过, 能给你反馈, 给你建议。对我而言是非常重要的一件事。

LT: It didn't even start by people contributing code, it was more that people started contributing ideas. And just the fact that somebody else takes a look at your project -- and I'm sure it's true of other things, too, but it's definitely true in code -- is that somebody else takes an interest in your code, looks at it enough to actually give you feedback and give you ideas. That was a huge thing for me.

当时我21岁, 还很年轻, 但在那之前, 我人生的一半时间都在敲代码。之前的每一个项目, 我都是独自完成的, 有人开始评论我的代码, 给予反馈, 真是头一遭。他们甚至还没开始改代码, 但对我来讲已经足以载入史册, 因为我觉得, “我爱世人!” 别误会我, 其实我并不合群。

I was 21 at the time, so I was young, but I had already programmed for half my life, basically. And every project before that had been completely personal and it was a revelation when people just started commenting, started giving feedback on your code. And even before they started giving code back, that was, I think, one of the big moments where I said, "I love other people!" Don't get me wrong -- I'm actually not a people person.

(笑声)

(Laughter)

我并不爱世人。

I don't really love other people --

(笑声)

(Laughter)

我爱电脑, 我喜欢通过发邮件互动, 因为这样能给你缓冲。但我真正喜欢的是人们给出评论, 参与我的项目。这样能让我的项目快速成长。

But I love computers, I love interacting with other people on email, because it kind of gives you that buffer. But I do love other people who comment and get involved in my project. And it made it so much more.

CA: 有没有那样一个时间点, 你发现自己一直在做的项目突然开始成型了, 你对自己说, “等一下, 这事儿也许能做成, 不单单是一个我能得到良好反馈的个人项目, 而是能在整个科技界掀起轩然大波的大事”?

CA: So was there a moment when you saw what was being built and it suddenly started taking off, and you thought, "Wait a sec, this actually could be something huge, not just a personal project that I'm getting nice feedback on, but a kind of explosive development in the whole technology world"?

LT: 还真没有。对我而言, 最重要的时刻并不是它变大的那一天, 恰恰是在它还不够大的时候。对我而言, 当我不再独自工作, 有10个人, 或者100个人加入进来——这才是一种突破。之后事情就顺其自然了。对我来说, 从100人到100万人, 并不是那么重要。

其实我的意思是，如果你.....

LT: Not really. I mean, the big point for me, really, was not when it was becoming huge, it was when it was becoming little. The big point for me was not being alone and having 10, maybe 100 people being involved -- that was a big point. Then everything else was very gradual. Going from 100 people to a million people is not a big deal -- to me. Well, I mean, maybe it is if you're --

(笑声)

(Laughter)

如果你是想把自己的成果卖掉，那确实就很重要——别误解我的意思。如果你感兴趣的是科技本身，是这个项目本身，那么最重要的就是找到志趣相投的人。之后这个群体会自发壮大。并没有一个特别明显的时间点，我会觉得“哇哦，这事儿快成了！”因为——这个时间非常漫长。

If you want to sell your result then it's a huge deal -- don't get me wrong. But if you're interested in the technology and you're interested in the project, the big part was getting the community. Then the community grew gradually. And there's actually not a single point where I went like, "Wow, that just took off!" because it -- I mean -- it took a long time, relatively.

CA: 所有我采访的技术专家们 都对你赞不绝口，说你极大改进了他们的工作。不光是Linux，还有Git，是一种软件开发管理系统。简单介绍一下它以及你在其中扮演的角色。

CA: So all the technologists that I talk to really credit you with massively changing their work. And it's not just Linux, it's this thing called Git, which is this management system for software development. Tell us briefly about that and your role in that.

LT: 我们遇到的问题之一，（在项目开始后）一段时间后才开始显现，就是当你..... 从10个或者100个人参与一个项目，增长到1万人的时候，而这一关于（Linux）内核，我们碰到的问题是，每次更新都有1000人参与，而更新频率是每两、三个月一次。这其中有些人作用不大。有很多人能做出小小的改进。

LT: So one of the issues we had, and this took a while to start to appear, is when you ... When you grow from having 10 people or 100 people working on a project to having 10,000 people, which -- I mean, right now we're in the situation where just on the kernel, we have 1,000 people involved in every single release and that's every two months, roughly two or three months. Some of those people don't do a lot. There's a lot of people who make small, small changes.

但要进行维护，所需要的工作量就会大大增加。我们吃尽了苦头。有时候甚至整个程序都仅仅是为了对源代码进行维护。CVS曾经是最流行的，我恨透了它，连碰都不想碰，我喜欢更基础、更有趣的方法，但其他人又不喜欢。

But to maintain this, the scale changes how you have to maintain it. And we went through a lot of pain. And there are whole projects that do only source-code maintenance. CVS is the one that used to be the most commonly used, and I hated CVS with a passion and refused to touch it and tried something else that was radical and interesting and everybody else hated.

CA: (笑声)

CA: (Laughs)

LT: 我们陷入了僵局，成千上万的人想参与进来，但很多时候，我成为了那个断点，我无法让自己跨出那一步，同上千人合作。

LT: And we were in this bad spot, where we had thousands of people who wanted to participate, but in many ways, I was the kind of break point, where I could not scale to the point where I could work with thousands of people.

因此Git是我的第二个大项目， 它存在的意义就是维护我的第一个大项目。 事实上这就是我的工作方式。 我编程并不是为了..... 我编程是因为好玩， 但我也想做一些有意义的事情， 因此我设计每一个程序 仅仅是因为我自己需要 而且.....

So Git is my second big project, which was only created for me to maintain my first big project. And this is literally how I work. I don't code for -- well, I do code for fun -- but I want to code for something meaningful so every single project I've ever done has been something I needed and --

CA: 因此， 实际上， Linux和Git的出现， 都是意外的收获， 仅仅因为你不想跟太多人一起工作。

CA: So really, both Linux and Git kind of arose almost as an unintended consequence of your desire not to have to work with too many people.

LT: 千真万确。没错。

LT: Absolutely. Yes.

(笑声)

(Laughter)

CA: 太牛了。 LT: 可不是吗。

CA: That's amazing. LT: Yeah.

(掌声)

(Applause)

你带来了技术革新， 不是一次，而是两次， 我们试图去探寻背后的原因。 你给了我们一些线索，但是..... 这是一张你小时候的照片， 你手里拿着魔方。 你刚才提到， 你从10岁、11岁就开始编程了， 人生的一半，

And yet, you're the man who's transformed technology not just once but twice, and we have to try and understand why it is. You've given us some clues, but ... Here's a picture of you as a kid, with a Rubik's Cube. You mentioned that you've been programming since you were like 10 or 11, half your life.

你是那种电脑天才，技术宅吗？ 你在学校里是无所不能的明星吗？ 你是个什么样的孩子？ Were you this sort of computer genius, you know, übernerd, were you the star at school who could do everything? What were you like as a kid?

LT: 没错，我觉得自己就是宅男的原型。 我是说，当年..... 我并不喜欢跟人打交道。 那是我弟弟。 显然我对魔方更有兴趣， 完全无视我弟弟。

LT: Yeah, I think I was the prototypical nerd. I mean, I was ... I was not a people person back then. That's my younger brother. I was clearly more interested in the Rubik's Cube than my younger brother.

(笑声)

(Laughter)

我还有个妹妹，她不在照片里， 当我们全家聚在一起的时候—— 我家人不算多，但我有好几个侄子侄女—— 她会事先给我打预防针。 在我进屋之前， 她会说， “好的， 你要注意什么什么.....” 因为我其实..... 我是个技术宅。 我喜欢电脑， 喜欢数学， 喜欢物理。 我擅长这些。 但也没到出类拔萃的地步。 就像我妹妹说的， 我最突出的品质就是不放弃。

My younger sister, who's not in the picture, when we had family meetings -- and it's not a huge family, but I have, like, a couple

of cousins -- she would prep me beforehand. Like, before I stepped into the room she would say, "OK. That's so-and-so ..." Because I was not -- I was a geek. I was into computers, I was into math, I was into physics. I was good at that. I don't think I was particularly exceptional. Apparently, my sister said that my biggest exceptional quality was that I would not let go.

CA: 好的, 那我们就来聊聊这个, 这很有意思。你不轻言放弃。也就是说, 不是因为技术宅或者聪明, 而是因为.....固执?

CA: OK, so let's go there, because that's interesting. You would not let go. So that's not about being a geek and being smart, that's about being ... stubborn?

LT: 没错, 就是固执。就像, 我开始做一件事情, 我不会说, “好的, 我不干了, 我要去干点别的..... 看, 比如那个!”

LT: That's about being stubborn. That's about, like, just starting something and not saying, "OK, I'm done, let's do something else -- Look: shiny!"

我发现我在做其他事情的时候也会这样。我在硅谷生活了7年。一直在同一家公司上班, 那可是在硅谷哦, 从未换过。这简直闻所未闻。尤其在硅谷这个地方。硅谷最大的特点就是, 人们总在跳槽, 一会这一会那。而我不是这样。

And I notice that in many other parts in my life, too. I lived in Silicon Valley for seven years. And I worked for the same company, in Silicon Valley, for the whole time. That is unheard of. That's not how Silicon Valley works. The whole point of Silicon Valley is that people jump between jobs to kind of mix up the pot. And that's not the kind of person I am.

CA: 但在Linux系统发展的过程中, 这种固执有时会让你和别人产生冲突。给我们讲讲这个。维持已有成果的质量是不是非常必要? 你会如何来形容之后发生的事情?

CA: But during the actual development of Linux itself, that stubbornness sometimes brought you in conflict with other people. Talk about that a bit. Was that essential to sort of maintain the quality of what was being built? How would you describe what happened?

LT: 我不确定这是不是非常必要。还是说回到“我并不合群”这一点, 有的时候我还..... 可以说, 不太理解他人的感受, 经常会说话伤到别人。对此我感到很抱歉。

LT: I don't know if it's essential. Going back to the "I'm not a people person," -- sometimes I'm also ... shall we say, "myopic" when it comes to other people's feelings, and that sometimes makes you say things that hurt other people. And I'm not proud of that.

(掌声)

(Applause)

但与此同时, 有人告诉我, 我应该友善一些。然后我会跟他们说, 可能你是友善, 但你应该更加富有侵略性一些, 他们会把这个当成我的不友善。

But, at the same time, it's -- I get people who tell me that I should be nice. And then when I try to explain to them that maybe you're nice, maybe you should be more aggressive, they see that as me being not nice.

(笑声)

(Laughter)

我想说的是我们是不同的。我不是个合群的人, 对此我并不感到骄傲, 但这就是我。而我喜欢开源软件的一点就是, 它能让形形色色的人在一起合作。我们不必相互喜欢——有

时候我们甚至互相讨厌。是真的，我们经常吵得不可开交。但是你会，你会发现许多事情——甚至都没法同意保留不同意见，因为大家的兴趣点真的不一样。

What I'm trying to say is we are different. I'm not a people person; it's not something I'm particularly proud of, but it's part of me. And one of the things I really like about open source is it really allows different people to work together. We don't have to like each other -- and sometimes we really don't like each other. Really -- I mean, there are very, very heated arguments. But you can, actually, you can find things that -- you don't even agree to disagree, it's just that you're interested in really different things.

回到我刚才说过的，我担心商人们利用我们的劳动成果，但没过多久，我发现，这些商人都是非常非常好的人。他们做那些我不感兴趣的事，关注点与我完全不同。他们使用开源软件的方法跟我不一样的。但正因为是开源的，他们可以这么干，而且实际上效果很好。我甚至认为跟我使用的效果一样好。我们需要合群的人，需要能与人交流的人，需要温暖友好的人 就像.....（笑声）喜欢拥抱你，带你走进这个圈子。但不是每个人都能做到。比如我就做不到。我关注的是技术。有的人关注用户界面。我死也做不来用户界面。举个例子，如果我被困在一个孤岛上，逃出生天的唯一办法是设计一个漂亮的用户界面，我宁愿死在岛上。

And coming back to the point where I said earlier that I was afraid of commercial people taking advantage of your work, it turned out, and very quickly turned out, that those commercial people were lovely, lovely people. And they did all the things that I was not at all interested in doing, and they had completely different goals. And they used open source in ways that I just did not want to go. But because it was open source they could do it, and it actually works really beautifully together.

（笑声）

And I actually think it works the same way. You need to have the people-people, the communicators, the warm and friendly people who like --

人跟人是不同的，我并不是在找借口，只是想解释给大家听。

（Laughter）

CA：我们上周见面的时候，你聊到自己的一些其他特点，我觉得很有意思。是关于品味的。

really want to hug you and get you into the community. But that's not everybody. And that's not me. I care about the technology. There are people who care about the UI. I can't do UI to save my life. I mean, if I was stranded on an island and the only way to get off that island was the make a pretty UI, I'd die there.

我这里有一些图片。我觉得这一段代码就没什么品位，而这一段要好一些，相信大家一看便知。这两段代码有什么不一样？

（Laughter）

LT：这是..... 在座的有哪些人编过程？

So there's different kinds of people, and I'm not making excuses, I'm trying to explain.

CA：我的天啊。

CA：Now, when we talked last week, you talked about some other trait that you have, which I found really interesting. It's this idea called taste.

LT：我敢保证，刚刚举手的每一位，肯定做过单向链表。它是在..... 这个，第一段不怎么

漂亮的代码，基本上是在你刚开始学编程的时候学到的。你不需要真的理解这段代码。
And I've just got a couple of images here. I think this is an example of not particularly good taste in code, and this one is better taste, which one can immediately see. What is the difference between these two?

我觉得最有意思的是最后一个“if”语句。因为在单向链表中——这里是试图将一个已存在的入口移出链表——不同之处在于这是第一个入口还是一个位于中间的入口。如果是第一个入口，你需要将指针改向第一个入口。而如果位于中间，你需要将指针改向上一个入口。这是两种完全不同的情况。

LT: So this is -- How many people here actually have coded?

CA: 这一个比较好。

CA: Oh my goodness.

LT: 这一个比较好。它没有if语句。这完全不影响——你不必了解这里为什么没有“if”语句，你需要了解的是有时候你可以换个角度看问题，重写代码，排除特例，完美覆盖所有情况。这就是好的代码。同时也很简单。这是最基本的原则。其实这都不重要——当然，细节非常重要。

LT: So I guarantee you, everybody who raised their hand, they have done what's called a singly-linked list. And it's taught -- This, the first not very good taste approach, is basically how it's taught to be done when you start out coding. And you don't have to understand the code.

对我来说，我愿意与之共事的人，必须有好的品位，这就是如何.....我举的这个例子很傻，没什么意义，因为实在太短。好的品位体现在更长的代码里。好的品位体现在能看清全局甚至有一种直觉，知道怎么把事情做漂亮。

The most interesting part to me is the last if statement. Because what happens in a singly-linked list -- this is trying to remove an existing entry from a list -- and there's a difference between if it's the first entry or whether it's an entry in the middle. Because if it's the first entry, you have to change the pointer to the first entry. If it's in the middle, you have to change the pointer of a previous entry. So they're two completely different cases.

CA: 好的，让我们来总结一下。你有品位，当然是对于程序员们而言，你很有品位——

CA: And that's better.

(笑声)

LT: And this is better. It does not have the if statement. And it doesn't really matter -- I don't want you understand why it doesn't have the if statement, but I want you to understand that sometimes you can see a problem in a different way and rewrite it so that a special case goes away and becomes the normal case. And that's good code. But this is simple code. This is CS 101. This is not important -- although, details are important.

LT: 对在座有些人而言，是这样。

To me, the sign of people I really want to work with is that they have good taste, which is how ... I sent you this stupid example that is not relevant because it's too small. Good taste is much bigger than this. Good taste is about really seeing the big patterns and kind of instinctively knowing what's the right way to do things.

CA: 你是个非常有才华的电脑程序员，你异常固执。但肯定不止这些。因为你是改变了未来的人。你一定有实现未来宏伟愿景的能力。你是个有远见的人，是吗？

CA: OK, so we're putting the pieces together here now. You have

taste, in a way that's meaningful to software people. You're --

LT: 其实在过去两天里，我在TED大会 觉得不太自在， 因为有很多关于未来的想象，对吧？ 我并不是一个有远见的人。 我没有制定未来五年的计划。 我是一名工程师。 而且我觉得，真是—— 我是说——我非常乐意跟梦想家在一起 他们行走四方，仰望苍穹， 看着满天星辰说，“我想到那儿去。” 但我是低头看路的那种人， 我只想填好眼前这个坑， 不让自己掉进去。 我是这样的人。

(Laughter)

(欢呼)

LT: I think it was meaningful to some people here.

(掌声)

CA: You're a very smart computer coder, and you're hellish stubborn. But there must be something else. I mean, you've changed the future. You must have the ability of these grand visions of the future. You're a visionary, right?

CA: 上周你跟我聊过他们俩。 他们是谁？ 你跟他们有什么关系？

LT: I've actually felt slightly uncomfortable at TED for the last two days, because there's a lot of vision going on, right? And I am not a visionary. I do not have a five-year plan. I'm an engineer. And I think it's really -- I mean -- I'm perfectly happy with all the people who are walking around and just staring at the clouds and looking at the stars and saying, "I want to go there." But I'm looking at the ground, and I want to fix the pothole that's right in front of me before I fall in. This is the kind of person I am.

LT: 这其实是技术界的一个老故事了， 特斯拉和爱迪生之争， 特斯拉被认为是一名有远见的科学家， 总是有疯狂的想法。 大家都爱特斯拉。 你看，还有人把自己公司也叫这个名字。

(Cheers)

(笑声)

(Applause)

另一个人是爱迪生， 经常被嘲讽说他是一个单调乏味的人， 而且—— 他最有名的一句话就是，“天才是1%的灵感加上99%的汗水。” 而我是爱迪生一派的， 尽管人们经常讨厌他。 如果你比较一下这两位， 特斯拉很能抓眼球， 但真正改变世界的是谁？ 爱迪生也许不是个讨喜的人， 但他做了很多事—— 他也许没那么天才， 不是那么有远见。 我觉得自己更像爱迪生而不是特斯拉。

CA: So you spoke to me last week about these two guys. Who are they and how do you relate to them?

CA: TED大会这一周的主题是梦想—— 宏伟、大胆的梦想。 而你却像碎梦人。

LT: Well, so this is kind of cliché in technology, the whole Tesla versus Edison, where Tesla is seen as the visionary scientist and crazy idea man. And people love Tesla. I mean, there are people who name their companies after him.

LT: 我会试着收敛一些。

(Laughter)

CA: 太好了。

The other person there is Edison, who is actually often vilified for being kind of pedestrian and is -- I mean, his most famous quote is, "Genius is one percent inspiration and 99 percent perspiration." And I'm in the Edison camp, even if people don't always like him. Because if you actually compare the two, Tesla has kind of this mind grab these days, but who actually changed the world? Edison may not

have been a nice person, he did a lot of things -- he was maybe not so intellectual, not so visionary. But I think I'm more of an Edison than a Tesla.

(笑声) 我们接受你, 我们爱你。

CA: So our theme at TED this week is dreams -- big, bold, audacious dreams. You're really the antidote to that.

谷歌和其他许多公司 使用你的软件, 挣了非常多的钱。你会觉得不爽吗?

LT: I'm trying to dial it down a bit, yes.

LT: 不会。原因有很多, 其中一个原因是, 我过得不错。真心不错。

CA: That's good.

另一个原因就是——如果我不将它开源, 而过于纠结的话, Linux就不会发展成今天这样。而且Linux让我有了许多经历, 尽管我不喜欢, 比如在公开场合讲话, 但同时, 这也是一种历练。相信我。很多事情让我成为了一个快乐的人, 我觉得当初的选择没有错。

(Laughter) We embrace you, we embrace you.

CA: 开源软件这个想法——我想这是最后一个话题——开源软件这个想法已经在全世界实现了吗? 还是说还有潜力可挖, 还有很多可以做的?

Companies like Google and many others have made, arguably, like, billions of dollars out of your software. Does that piss you off?

LT: 我有两个想法。我认为开源软件如此成功的原因之一 在于归根结底, 代码是一个非黑即白的事情。我们有非常明确的方法来分辨, 一段代码是行还是不行。代码要么运行成功, 要么失败, 这就意味着没什么可争议的。我们有很多争议, 不是吗? 在其他许多领域——人们探讨过公开政治和其他类似的事情——有时候真的很难去说, 没错, 你可以在其他领域也采取同样的原则, 因为在那些地方不是非黑即白的, 有灰色, 也有其他颜色。

LT: No. No, it doesn't piss me off for several reasons. And one of them is, I'm doing fine. I'm really doing fine.

因此, 科学界的开源显然是一种回归。科学最初是开源的。但之后变得越来越封闭, 只存在那些昂贵的科学期刊上。开源让科学回归了, 我们有了arXiv和开放期刊。维基百科也改变了世界。还有其他许多例子, 我觉得未来还会有更多。

But the other reason is -- I mean, without doing the whole open source and really letting go thing, Linux would never have been what it is. And it's brought experiences I don't really enjoy, public talking, but at the same time, this is an experience. Trust me. So there's a lot of things going on that make me a very happy man and thinking I did the right choices.

CA: 但你不是个有远见的人, 这些事跟你没啥关系。

CA: Is the open source idea -- this is, I think we'll end here -- is the open source idea fully realized now in the world, or is there more that it could go, are there more things that it could do?

LT: 当然。

LT: So, I'm of two minds there. I think one reason open source works so well in code is that at the end of the day, code tends to be somewhat black and white. There's often a fairly good way to decide, this is done correctly and this is not done well. Code either works or it doesn't, which means that there's less room for arguments. And we have arguments despite this, right? In many other areas -- I mean, people have talked about open politics and things like that -- and it's really hard sometimes to say that, yes, you can apply the same principles in some other areas just because the black and white turns into not just gray, but different colors.

(笑声)

So, obviously open source in science is making a comeback. Science was there first. But then science ended up being pretty closed, with very expensive journals and some of that going on. And open source is making a comeback in science, with things like arXiv and open journals. Wikipedia changed the world, too. So there are other examples, I'm sure there are more to come.

得靠你们去实现了，对吗？

CA: But you're not a visionary, and so it's not up to you to name them.

CA: 没错。

LT: No.

林纳斯·托瓦兹， 谢谢你的Linux， 谢谢你为互联网做的一切， 谢谢你为安卓手机做的一切。

(Laughter)

谢谢你来参加TED大会， 让我们更加了解你。

It's up to you guys to make them, right?

LT: 谢谢。

CA: Exactly.

(掌声)

Linus Torvalds, thank you for Linux, thank you for the Internet, thank you for all those Android phones.