

引言

I want to tell you a story.

我想给大家讲个故事。

No, not the story of how, in 1991, Linus Torvalds wrote the first version of the Linux kernel. You can read that story in lots of Linux books. Nor am I going to tell you the story of how, some years earlier, Richard Stallman began the GNU Project to create a free Unix-like operating system. That’s an important story too, but most other Linux books have that one, as well.

故事内容不是 Linus Torvalds 在1991年怎样写了 Linux 内核的第一个版本， 因为这些内容你可以在许多 Linux 书籍中读到。我也不是来告诉你， 更早之前，Richard Stallman 是如何开始 GNU 项目， 设计了一个免费的类Unix 的操作系统。那也是一个很有意义的故事， 但大多数 Linux 书籍也讲到了它。

No, I want to tell you the story of how you can take back control of your computer.

我想告诉大家一个你如何才能夺回计算机管理权的故事。

When I began working with computers as a college student in the late 1970s, there was a revolution going on. The invention of the microprocessor had made it possible for ordinary people like you and me to actually own a computer. It’s hard for many people today to imagine what the world was like when only big business and big government ran all the computers. Let’s just say, you couldn’t get much done.

在20世纪70年代末，我刚开始和计算机打交道时，正进行着一场革命，那时的我还是一名大学生。微处理器的发明，使普通老百姓（就如你和我）真正拥有一台计算机成为可能。今天， 人们难以想象，只有大企业和强大的政府才能够拥有计算机的世界，是怎样的一个世界。 简单说，你做不了多少事情。

Today, the world is very different. Computers are everywhere, from tiny wristwatches to giant data centers to everything in between. In addition to ubiquitous computers, we also have a ubiquitous network connecting them together. This has created a wondrous new age of personal empowerment and creative freedom, but over the last couple of decades something else has been happening. A single giant corporation has been imposing its control over most of the world’s computers and deciding what you can and cannot do with them. Fortunately, people from all over the world are doing something about it. They are fighting to maintain control of their computers by writing their own software. They are building Linux.

今天，世界已经截然不同了。计算机遍布各个领域，从小手表到大型数据中心，及大小介于它们之间的每件东西。除了随处可见的计算机之外，我们还有一个无处不在的连接所有计算机的网络。这已经开创了一个奇妙的， 个人授权和创作自由的新时代，但是在过去的二三十年里，正在发生另一些事情。一个大公司不断地把它的 管理权强加到世界上绝大多数的计算机上，并且决定你对计算机的操作权力。幸运地是，来自世界各地的人们， 正积极努力地做些事情来改变这种境况。通过编写自己的软件，他们一直在为维护电脑的管理权而战斗着。 他们建设着 Linux。

Many people speak of “freedom” with regard to Linux, but I don’t think most people know what this freedom really means. Freedom is the power to decide what your computer does, and the only

way to have this freedom is to know what your computer is doing. Freedom is a computer that is without secrets, one where everything can be known if you care enough to find out.

一提到 Linux，许多人都会说到“自由”，但我不认为他们都知道“自由”的真正涵义。“自由”是一种权力，它决定你的计算机能做什么，同时能够拥有这种“自由”的唯一方式就是知道计算机正在做什么。“自由”是指一台没有任何秘密的计算机，你可以从它那里了解一切，只要你用心的去寻找。

为什么使用命令行

Have you ever noticed in the movies when the “super hacker,”— you know, the guy who can break into the ultra-secure military computer in under thirty seconds —sits down at the computer, he never touches a mouse? It’s because movie makers realize that we, as human beings, instinctively know the only way to really get anything done on a computer is by typing on a keyboard.

你是否注意到，在电影中一个“超级黑客”坐在电脑前，从不摸一下鼠标，就能够在30秒内侵入到超安全的军用计算机中。这是因为电影制片人意识到，作为人类，本能地知道让计算机圆满完成工作的唯一途径，是用键盘来操纵计算机。

Most computer users today are only familiar with the graphical user interface (GUI) and have been taught by vendors and pundits that the command line interface (CLI) is a terrifying thing of the past. This is unfortunate, because a good command line interface is a marvelously expressive way of communicating with a computer in much the same way the written word is for human beings. It’s been said that “graphical user interfaces make easy tasks easy, while command line interfaces make difficult tasks possible” and this is still very true today.

现在，大多数的计算机用户只是熟悉图形用户界面（GUI），并且产品供应商和此领域的学者会灌输给用户这样的思想，命令行界面（CLI）是过去使用的一种很恐怖的东西。这就很不幸，因为一个好的命令行界面，是用来和计算机进行交流沟通的非常有效的方式，正像人类社会使用文字互通信息一样。人们说，“图形用户界面让简单的任务更容易完成，而命令行界面使完成复杂的任务成为可能”，到现在这句话仍然很正确。

Since Linux is modeled after the Unix family of operating systems, it shares the same rich heritage of command line tools as Unix. Unix came into prominence during the early 1980s (although it was first developed a decade earlier), before the widespread adoption of the graphical user interface and, as a result, developed an extensive command line interface instead. In fact, one of the strongest reasons early adopters of Linux chose it over, say, Windows NT was the powerful command line interface which made the “difficult tasks possible.”

因为 Linux 是以 Unix 家族的操作系统为模型写成的，所以它分享了 Unix 丰富的命令行工具。Unix 在20世纪80年代初显赫一时(虽然，开发它在更早之前)，结果，在普遍地使用图形界面之前，开发了一种广泛的命令行界面。事实上，很多人选择 Linux（而不是其他的系统，比如说 Windows NT）是因为其可以使“完成复杂的任务成为可能”的强大的命令行界面。

这本书讲什么

This book is a broad overview of “living” on the Linux command line. Unlike some books that concentrate on just a single program, such as the shell program, bash, this book will try to convey how to get along with the command line interface in a larger sense. How does it all work? What can it do? What’s the best way to use it?

这本书介绍如何生存在 Linux 命令行的世界。不像一些书籍仅仅涉及一个程序，比如像 shell 程序，bash。这本书将试着向你传授如何与命令行界面友好相处。它是怎样工作的？它能做什么？使用它的最好方法是什么？

This is not a book about Linux system administration. While any serious discussion of the command line will invariably lead to system administration topics, this book only touches on a few administration issues. It will, however, prepare the reader for additional study by providing a solid foundation in the use of the command line, an essential tool for any serious system administration task.

这不是一本关于 **Linux** 系统管理的书。然而任何一个关于命令行的深入讨论，都一定会牵涉到 系统管理方面的内容，这本书仅仅提到一点儿管理方面的知识。但是这本书为读者准备好了学习更多内容的坚实基础， 毕竟要胜任系统管理工作也需要良好的命令行使用基本功。

This book is very Linux-centric. Many other books try to broaden their appeal by including other platforms such as generic Unix and MacOS X. In doing so, they “water down” their content to feature only general topics. This book, on the other hand, only covers contemporary Linux distributions. Ninety-five percent of the content is useful for users of other Unix-like systems, but this book is highly targeted at the modern Linux command line user.

这本书是围绕 **Linux** 而写的。许多书籍，为了扩大自身的影响力，会包含一些其它平台的知识， 比如 Unix, MacOS X 等。这样做，很多内容只能比较空泛的去讲了。另一方面， 这本书只研究了当代 Linux 发行版。虽然，对于使用其它类 Unix 系统的用户来说， 书中95%的内容是有用的，但这本书主要面向的对象是现代 Linux 命令行用户。

谁应该读这本书

This book is for new Linux users who have migrated from other platforms. Most likely you are a “power user” of some version of Microsoft Windows. Perhaps your boss has told you to administer a Linux server, or maybe you’re just a desktop user who is tired of all the security problems and want to give Linux a try. That’s fine.here.All are welcome

这本书是为已经从其它平台移民到 Linux 系统的新手而写的。最有可能，你是使用某个 Windows 版本的高手。或许是老板让你去管理一个 Linux 服务器，或许你只是一个桌面用户，厌倦了系统出现的各种 安全防御问题，而想要体验一下 Linux。很好，这里欢迎你们！

That being said, there is no shortcut to Linux enlightenment. Learning the command line is challenging and takes real effort. It’s not that it’s so hard, but rather it’s so vast. The average Linux system has literally thousands of programs you can employ on the command line. Consider yourself warned; learning the command line is not a casual endeavor.

不过一般来说，对于 Linux 的启蒙教育，没有捷径可言。学习命令行富于挑战性，而且很费气力。这并不是说 Linux 命令行很难学，而是它的知识量很大，不容易掌握。Linux 操作系统， 差不多有数以千计的命令可供用户操作。由此可见，要给自己提个醒，命令行可不是轻轻松松就能学好的。

On the other hand, learning the Linux command line is extremely rewarding. If you think you’re a “power user” now, just wait. You don’t know what real power is — yet. And, unlike many other computer skills, knowledge of the command line is long lasting. The skills learned today will still be useful ten years from now. The command line has survived the test of time.

另一方面，学习 Linux 命令行会让你受益匪浅，给你极大的回报。如果你认为， 现在你已经是高手了。别急，其实你还不知道什么才是真正的高手。不像其他一些计算机技能， 一段时间之后可能就被淘汰了， 命令行知识却不会落伍，你今天所学到的，在十年以后， 都会有用处。命令行通过了时间的考验。

It is also assumed that you have no programming experience, but not to worry, we’ll start you down that path as well.

如果你没有编程经验，也不要担心，我会带你入门。

这本书的内容

This material is presented in a carefully chosen sequence, much like a tutor sitting next to you guiding you along. Many authors treat this material in a “systematic” fashion, which makes sense from a writer’s perspective, but can be very confusing to new users.

这些材料是经过精心安排的，很像一位老师坐在你身旁，耐心地指导你。许多作者用系统化的方式讲解这些材料，虽然从一个作者的角度考虑很有道理，但对于 Linux 新手来说，他们可能会感到非常困惑。

Another goal is to acquaint you with the Unix way of thinking, which is different from the Windows way of thinking. Along the way, we’ll go on a few side trips to help you understand why certain things work the way they do and how they got that way. Linux is not just a piece of software, it’s also a small part of the larger Unix culture, which has its own language and history. I might throw in a rant or two, as well.

另一个目的，是想让读者熟悉 Unix 的思维方式，这种思维方式与 Windows 不同。在学习过程中，我们会帮助你理解为什么某些命令那样工作，以及它们是如何工作的。Linux 不仅是一款软件，也是 Unix 文化的一小部分，它有自己的语言和历史渊源。同时，我也许会说过激的话。

This book is divided into five parts, each covering some aspect of the command line experience. Besides the first part, which you are reading now, this book contains:

这本书共分为五部分，每一部分讲述了不同方面的命令行知识。除了第一部分，也就是你正在阅读的这一部分，这本书还包括：

- Part 2 – Learning The Shell starts our exploration of the basic language of the command line including such things as the structure of commands, file system navigation, command line editing, and finding help and documentation for commands.
- Part 3 – Configuration And The Environment covers editing configuration files that control the computer’s operation from the command line.
- Part 4 – Common Tasks And Essential Tools explores many of the ordinary tasks that are commonly performed from the command line. Unix-like operating systems, such as Linux, contain many “classic” command line programs that are used to perform powerful operations on data.
- Part 5 – Writing Shell Scripts introduces shell programming, an admittedly rudimentary, but easy to learn, technique for automating many common computing tasks. By learning shell programming, you will become familiar with concepts that can be applied to many other programming languages.

• 第二部分 — 学习 shell 开始探究命令行基本语言，包括命令组成结构，文件系统浏览，编写命令行，查找命令帮助文档。

• 第三部分 — 配置文件及环境 讲述了如何编写配置文件，通过配置文件，用命令行来 操控计算机。

• 第四部分 — 常见任务及主要工具 探究了许多命令行经常执行的普通任务。类似于 Unix 的操作系统，例如 Linux, 包括许多经典的命令行程序，这些程序可以用来对数据进行 强大的操作。

• 第五部分 — 编写 Shell 脚本 介绍了 shell 编程，一个无可否认的基本技能，能够自动化许多 常见的计算任务，很容易学。通过学习 shell 编程，你会逐渐熟悉一些关于编程语言方面的概念， 这些概

念也适用于其他的编程语言。

怎样阅读这本书

Start at the beginning of the book and follow it to the end. It isn’t written as a reference work, it’s really more like a story with a beginning, middle, and an end.

从头到尾的阅读。它并不是一本技术参考手册，实际上它更像一本故事书，有开头，过程，结尾。

前提条件

To use this book, all you will need is a working Linux installation. You can get this in one of two ways:

为了使用这本书，你需要安装 Linux 操作系统。你可以通过两种方式，来完成安装。

1. Install Linux on a (not so new) computer. It doesn’t matter which distribution you choose, though most people today start out with either Ubuntu, Fedora, or . If in doubt, try Ubuntu first. Installing a modern Linux distribution can be ridiculously easy or ridiculously difficult depending on your hardware. I suggest a desktop computer that is a couple of years old and has at least 256 megabytes of RAM and 6 gigabytes of free hard disk space. Avoid laptops and wireless networks if at all possible, as these are often more difficult to get working.

2. Use a “Live CD.” One of the cool things you can do with many Linux distributions is run them directly from a CDROM without installing them at all. Just go into your BIOS setup and set your computer to “Boot from CDROM,” insert the live CD, and reboot. Using a live CD is a great way to test a computer for Linux compatibility prior to installation. The disadvantage of using a live CD is that it may be very slow compared to having Linux installed on your hard drive. Both Ubuntu and Fedora (among others) have live CD versions.

1. 在一台（不用很新）的电脑上安装 Linux。你选择哪个 Linux 发行版安装，是无关紧要的事。虽然大多数人一开始选择安装 Ubuntu, Fedora, 或者 OpenSUSE。如果你拿不定主意，那就先试试 Ubuntu。由于主机硬件配置不同，安装 Linux 时，你可能不费吹灰之力就装上了，也可能费了九牛二虎之力还装不上。所以我建议，一台使用了几年的台式机，至少要有256M 的内存，6G 的硬盘可用空间。尽可能避免使用 笔记本电脑和无线网络，在 Linux 环境下，它们经常不能工作。

2. 使用“Live CD.”许多 Linux 发行版都自带一个比较酷的功能，你可以直接从系统安装盘 CDROM 中运行 Linux，而不必安装 Linux。开机进入 BIOS 设置界面，更改引导项，设置为“从 CDROM 启动”。

Regardless of how you install Linux, you will need to have occasional superuser (i.e., administrative) privileges to carry out the lessons in this book.

不管你怎样安装 Linux，为了练习书中介绍的知识，你需要有超级用户（管理员）权限。

After you have a working installation, start reading and follow along with your own computer. Most of the material in this book is “hands on,” so sit down and get typing!

当你在自己的电脑上安装了 Linux 系统之后，就开始一边阅读本书，一边练习吧。本书大部分内容 都可以自己动手练习，坐下来，敲入命令，体验一下吧。

Why I Don’t Call It “GNU/Linux”

In some quarters, it’s politically correct to call the Linux operating system the “GNU/Linux operating system.” The problem with “Linux” is that there is no completely correct way to

name it because it was written by many different people in a vast, distributed development effort. Technically speaking, Linux is the name of the operating system’s kernel, nothing more. The kernel is very important of course, since it makes the operating system go, but it’s not enough to form a complete operating system.

Enter Richard Stallman, the genius-philosopher who founded the Free Software movement, started the Free Software Foundation, formed the GNU Project, wrote the first version of the GNU C Compiler (gcc), created the GNU General Public License (the GPL), etc., etc., etc. He insists that you call it “GNU/Linux” to properly reflect the contributions of the GNU Project. While the GNU Project predates the Linux kernel, and the project’s contributions are extremely deserving of recognition, placing them in the name is unfair to everyone else who made significant contributions. Besides, I think “Linux/GNU” would be more technically accurate since the kernel boots first and everything else runs on top of it.

In popular usage, “Linux” refers to the kernel and all the other free and open source software found in the typical Linux distribution; that is, the entire Linux ecosystem, not just the GNU components. The operating system marketplace seems to prefer one-word names such as DOS, Windows, MacOS, Solaris, Irix, AIX. I have chosen to use the popular format. If, however, you prefer to use “GNU/Linux” instead, please perform a mental search and replace while reading this book. I won’t mind.

为什么我不叫它“GNU/Linux”

在某些领域，把 Linux 操作系统称为“GNU/Linux 操作系统”， 则政治立场正确。但“Linux”的问题是， 没有一个完全正确的方式能命名它，因为它是由许许多多，分布在世界各地的贡献者们，合作开发而成的。 从技术层面讲，Linux 只是操作系统的内核名字，没别的含义。当然内核非常重要，因为有它， 操作系统才能运行起来，但它并不能构成一个完备的操作系统。

Richard Stallman 是一个天才的哲学家，自由软件运动创始人，自由软件基金会创办者，他创建了 GNU 工程， 编写了第一版 GNU C 编译器（gcc），创立了 GNU 通用公共协议（the GPL）等等。他坚持把 Linux 称为“GNU/Linux”，为的是准确地反映 GNU 工程对 Linux 操作系统的贡献。然而，尽管 GNU 项目早于 Linux 内核，项目的贡献应该得到极高的赞誉，但是把 GNU 用在 Linux 名字里， 这对其他为 Linux 的发展做出重大贡献的程序员来说，就不公平了。而且，我觉得要是叫也要叫 “Linux/GNU” 比较准确一些， 因为内核会先启动，其他一切都运行在内核之上。

在目前流行的用法中，“Linux”指的是内核以及在一个典型的 Linux 发行版中所包含的所有免费及开源软件； 也就是说，整个 Linux 生态系统，不只有 GNU 项目软件。在操作系统商界，好像喜欢使用单个词的名字， 比如说 DOS, Windows, MacOS, Solaris, Irix, AIX. 所以我选择用流行的命名规则。然而， 如果你喜欢用“GNU/Linux”，当你读这本书时，可以在脑子里搜索并替换“Linux”。我不介意。

拓展阅读

Here are some Wikipedia articles on the famous people mentioned in this chapter:

Wikipedia 网站上有些介绍本章提到的名人的文章，以下是链接地址：

- http://en.wikipedia.org/wiki/Linux_Torvalds
- http://en.wikipedia.org/wiki/Richard_Stallman

The Free Software Foundation and the GNU Project:

介绍自由软件基金会及 GNU 项目的网站和文章：

- http://en.wikipedia.org/wiki/Free_Software_Foundation
- <http://www.fsf.org>
- <http://www.gnu.org>

Richard Stallman has written extensively on the “GNU/Linux” naming issue:

Richard Stallman 用了大量的文字来叙述“GNU/Linux”的命名问题，可以浏览以下网页：

- <http://www.gnu.org/gnu/why-gnu-linux.html>
- <http://www.gnu.org/gnu/gnu-linux-faq.html#tools>

[Go to Table of Contents](#)