

第1章 开源软件的起源

事物的起源往往是混乱的。有关开放源代码起源的故事也不例外。

开源软件的故事像是一个复杂而曲折的迷宫，是 UNIX 操作系统历史的一部分，而 UNIX 操作系统本身就是错综复杂的。在本章中，我们将看看开放源代码微不足道的起源。

1.1 开端

在计算机发展的早期，人们共享可读的源代码是一种普遍存在的行为。很多计算机开发过程都发生在大学和企业的研究部门，如 AT&T 的贝尔实验室。这些机构有着悠久的开放与协作传统，因此，即使代码没有正式进入公共领域，但是也得到广泛分享。

计算机公司将软件与自己的系统一起交付，通常还包括源代码。用户经常要自己修改软件，以便它能支持新的硬件或者增加新的功能。当时许多供应商的态度是，软件是使用硬件时所必需的，但独立销售是万万不行的。

事实上，在早期，计算机系统的用户往往需要编写他们自己的软件。IBM 704 计算机（如图 1-1 所示）的第一个操作系统，即 GM-NAA I/O 输入/输出系统，是由通用汽车研究所的 Robert L. Patrick 和北美航空的

Owen Mock 在 1956 年编写的。(操作系统是支持计算机基本功能的软件，例如调度任务、执行应用程序以及控制包括存储设备在内的一些外围设备。)



图 1-1 1957 年 IBM 704 在 NASA Langley

直到 20 世纪 70 年代末，澳大利亚新南威尔士大学的 John Lions 对 UNIX 操作系统的第六版源代码进行注释时，共享代码的文化在某些圈子里仍然很浓厚。“Lions 注释”的复印本在大学计算机科学系和其他地方广为流传。虽然 UNIX 操作系统的代码所有者 AT&T 在技术上不允许这种随意的、非正式的共享，但这是当时的常态。

1.1.1 啊，UNIX 操作系统

对软件进行修改使其在新的或不同的硬件上运行的想法在 UNIX

操作系统上得到发展。1973—1974 年，人们用 C 语言重写了 UNIX 操作系统（V4 版本）。C 语言是 Dennis Ritchie 在贝尔实验室新开发的一种编程语言。按照当时的标准，使用一种高级编程语言重写 UNIX 操作系统虽然不是完全创新，但却是一种不寻常的、新的甚至有争议的做法。

按照当时的情况，更典型的做法是使用特定机器架构的汇编语言。虽然编写好的代码很有挑战性，也很耗时，但是，由于汇编程序代码和特定架构的机器代码指令（可以直接在硬件上执行）之间密切相关，因此，汇编程序是非常高效的。而在当时计算机性能有限的条件下，效率是非常重要的。

然而，由于用 C 语言重写，UNIX 操作系统可以相对容易地被修改并安装到其他机器上。也就是说，它是“可移植的”，这确实不同寻常。当时的常态是为每个新的硬件平台编写一个新的操作系统和一套支持系统以及应用软件。

当然，为了进行这些修改，你需要源代码。

AT&T 愿意提供这项服务有几个原因。其中一个特别重要的原因是，在 1975 年第六版发布后，AT&T 开始向大学和商业公司以及美国政府授权使用 UNIX 操作系统。但这些许可证不包括任何支持或错误修复，这样做的目的就是“追求软件商业化”。而 AT&T 认为根据协议条款，它无权这样做，因为它是作为一个受管制的电话服务垄断机构来运作的。共享软件源代码允许持有者自行修复和将 UNIX 操作系统“移植”到不兼容的新系统中。（我们将在第 3 章中再次讨论版权和许可证的话题。）

然而，20 世纪 80 年代初期，整个行业对共享软件源代码的宽松态度开始走到尽头。

1.1.2 不再有免费的午餐了吗

1982 年，AT&T 与美国联邦贸易委员会签署了同意法令，规定分拆贝尔地区运营公司（如图 1-2 所示）。该法令允许 AT&T 将软件作为一项业务进行发展。在其他方面，该法令允许 AT&T 进入计算机行业。不久之后，AT&T 开始开发 UNIX 操作系统的商业版本。

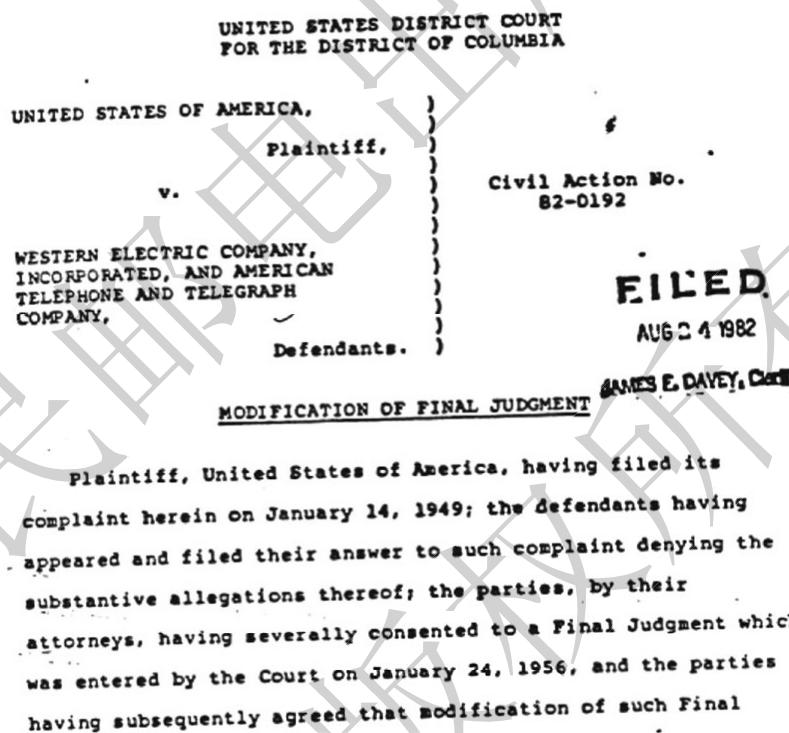


图 1-2 AT&T 与美国联邦贸易委员会于 1982 年签署的一项同意法令

由于 AT&T UNIX 操作系统的许可证持有者们开发并发布了各种专有的 UNIX 版本，这些版本都或多或少地与对方不兼容，这就导致了接下来数十年的“UNIX 混战”。这是一段极其复杂的历史。除了注意到它创造的垂直简仓与 UNIX 操作系统所取代的小型机和大型机没有什么不

同以外，它也与开源的发展没有什么关系。

在同一时期，AT&T 以其新的 UNIX 操作系统实验室子公司的名义与加州大学伯克利分校就其 UNIX 操作系统的衍生产品（又称“分叉”）——伯克利软件发行版（Berkeley Software Distribution, BSD）发生了法律纠纷。准确地说，这其实是与加州大学伯克利分校的计算机系统研究小组（Computer Systems Research Group, CSRG）的争论。

加州大学伯克利分校计算机系统研究小组曾是 AT&T 的教育许可机构之一。随着时间的推移，该研究小组修改并增加了授权版本的 UNIX 操作系统的功能，并在 1978 年开始将这些附加功能以 BSD 发布。之后，BSD 增加了许多重要的功能，包括彻底重新架构和重写许多关键子系统，同时增加了许多全新的组件。由于其宽泛的变化和改进，BSD 版本的 UNIX 操作系统逐渐被视为一种全新的，甚至更好的 UNIX 操作系统。许多 AT&T 的授权者最终将大量 BSD 版本的代码纳入他们自己的 UNIX 版本中。（由于不同公司对 AT&T 风格和伯克利风格的青睐度不同，这进一步加剧了 UNIX 战争。）

加州大学伯克利分校继续开发 BSD 版本的代码，并逐步取代大多数仍在 AT&T 许可下的标准 UNIX 工具。这最终促成了 1991 年 6 月 Net/2 的发布。Net/2 是一个几乎完整的操作系统，表面上可以自由再分发。这反过来又导致 AT&T 起诉加州大学伯克利分校侵犯版权。

UNIX 操作系统曾是许多源代码共享的中心，但是它的商业化导致了一个更加分裂和封闭的 UNIX 环境。

1.1.3 PC 是一种不同的文化

但分享精神也受到更广泛的侵蚀。

20世纪80年代，PC领域越来越多地被IBM PC及其克隆产品所支配。而这些计算机上运行的是Microsoft公司的操作系统。在很大程度上没有什么开源的显著特征。在某种程度上，这可能反映了一个事实：PC相对标准化的系统架构使得源代码的可移植性优势变得不再重要。

此外，人们购买PC的费用中并不包括开发软件所需的大多数工具，而购买这些工具的费用可能会很快增加。Microsoft公司的DOS操作系统包含了BASIC编程语言解释器，但即使以当时要求不高的标准来看，这款工具也是过时的，无法用于正式场合。当Borland公司更现代的Turbo Pascal在1984年首次亮相时，价格仅为50美元，鉴于典型的编程语言包要数百美元，因此这是一次彻底的创新。编程库和其他资源——包括大部分书本、杂志和其他离线的信息——增加了费用。相对随意的业余爱好者并不适合对一款软件进行改动。

当然，人们在IBM PC上编程。随着时间推移，一个非常健康的自由软件和共享软件开发社区应运而生。

笔者就是其中之一。

共享软件——至少在当时这个词被普遍使用——是一种先试后买的软件。当时，零售的盒装软件可以卖到数百美元，但无法保证它一定能在你的计算机上正常工作，如果不能正常使用，那就要祝你能够顺利退货了。

笔者编写的软件是一款小型DOS文件管理器——一个35KB的汇编程序，名为Directory Freedom。它是笔者和另一个开发人员共同开发的。当时我们参考了《计算机杂志》(PC Magazine)上的汇编代码列表。这个程序虽然没赚到太多钱，但还是有一定的用户基础，至今笔者仍然时不时会收到关于它的电子邮件。笔者还将自己编写的各种实用程序上传到本地的电子公告板系统(BBS)。

但是，发布源代码从来都不是一件特别重要的事情。

1.1.4 打破社区

类似的商业化动态也在其他地方上演。麻省理工学院人工智能实验室，这个在 Steven Levy 所著的《黑客》(*Hackers*) 一书中被誉为“纯粹的黑客天堂，科技广场的修道院，人们在那里以黑客为生，以黑客为命”的地方正在发生变化。在这里，Lisp 正在被商业化。

Lisp 编程语言是人工智能研究的主要工作，但它需要很高的算力，在当时的普通计算机上不能很好地运行。因此，在将近 10 年的时间里，人工智能实验室的成员一直在试图优化系统以运行 Lisp。直到 1979 年，这项工作已经取得了进展，商业化看起来是一个合理的选择。

最终，两家公司——Symbolics 和 Lisp Machines Inc. 成立了。但这是一个混乱且激烈的过程，导致开放合作的减少和实验室成员的大量流失。

Richard Stallman 是人工智能实验室的一员，他并没有去前景更好的 Lisp Machines Inc.，但他还是因为实验室社区的分裂而受到很大的影响。Stallman 之前编写过被广泛使用的 Emacs 编辑程序。正如 Glyn Moody 在《反叛的代码》(*Rebel Code*)(Perseus Publishing, 2001) 中写道：“Stallman 为 Emacs 立下一条‘非正式’的规定，即任何人对 Emacs 做出任何改进都必须把它们发回给他。”

他在 Symbolics 与 Lisp Machines Inc. 两家公司的竞争中感受到专有代码的影响。他认为缺乏共享阻碍了软件社区的形成。这也促使他决定开发一个自由且可移植的操作系统。关于 Stallman 作为自由软件倡导者，还有一个广为流传的故事。因为拒绝访问实验室新安装的激光打印机

(Xerox 9700) 的软件源代码， Stallman 不能像以前一样通过修改实验室激光打印机的源代码给打印机发送指令。

1.1.5 自由软件进入混战

1983 年， Richard Stallman 在互联网（当时仍称为 ARPANET）的 Usenet 新闻组上宣布：“从这个感恩节开始，我将编写一个完全与 UNIX 操作系统兼容的软件系统，称为 GNU (Gnu's Not UNIX) 并将它免费送给所有能使用它的人。”如图 1-3 所示。



图 1-3 Stallman 的自由软件基金会 (Free Software Foundation, FSF)

和 GNU 项目通常被认为是自由和开源软件运动的开端

[来源：自由艺术许可证下的 Victor Siame (vcopovi@wanadoo.fr)]

作为辩解，他接着写道：

我认可的黄金法则要求，如果我喜欢一个程序，就必须把它分享给其他喜欢它的人。出于良知，我不能签署一份保密协议或软件许可协议。为了能在不违反自己原则的情况下继续使用计算机，我决定组装一个包含足够多的自由软件的系统，这样就可以在没有任何非自由软件的情况下继续干下去。

它将基于 UNIX 操作系统的模式，也就是说，它将由模块化的组件

组成，例如实用程序和 C 语言编译器，这些都是构建一个工作系统所必需的。该项目于 1984 年开始，时至今日，因为 GNU Hurd 操作系统的内核一直没有完成，所以事实上还没有“GNU 操作系统”。没有内核，它们就没有办法与硬件通信，也就没有办法运行实用程序、应用程序或其他软件。

然而，Stallman 确实完成了他的操作系统的许多其他组件。关键在于，这些组件包括从源代码构建一个正常运行的操作系统以及从命令行执行基本的系统任务所需的部分。因为 UNIX 操作系统的一个特点是设计模块化，所以，修改和改编 UNIX 操作系统的部分内容而不是一次性全盘替换整个系统是完全可行的(这一事实对后来 Linux 操作系统的发展至关重要)。

1.1.6 建立自由的基础

然而，从开源起源的角度来看，同样重要的还有 1985 年的 GNU 宣言 (GNU Manifesto)、1986 年的自由软件定义和 1989 年的 GNU 通用公共许可协议 (General Public License, GPL)，这些规范正式确立了防止“限制自由软件之自由”的原则。

GPL 要求，如果以二进制 (即机器可读的形式，无论是原始形式还是修改后的形式) 发布 GPL 授权下的程序，那么必须提供可读的源代码。通过这种方式，用户可以在原始程序和其他人改进的基础上进行开发。但是，如果用户自己进行更改并发布它们，那么还必须保证这些更改可供其他人使用。这就是所谓的“copyleft”或出于彼此义务的互惠许可。

20 世纪 80 年代末，自由和开源软件仍处于起步阶段。(事实上，“开放源代码”这个词甚至还没有被创造出来。) 那时 Linux 操作系统还

没有诞生。BSD 版本的 UNIX 操作系统很快就卷入一场与 AT&T 的官司中。互联网还没有完全商业化。但是，特别是在事后看来，我们可以开始辨别出今后将变得重要起来的模式：协作、回馈以及帮助人们了解规则并适当合作的框架。

但是，真正让 Linux 操作系统和开源崭露头角的是 20 世纪 90 年代的互联网繁荣。尽管这一阶段的开源最终只是一个更重要的故事的第一幕。在这样的背景下，开源即将崛起，而计算机硬件和软件的格局也将发生根本的变化。

1.2 零散的硬件和软件

将时钟拨回到 1991 年。一个名叫 Linus Torvalds 的芬兰大学生在 Usenet 新闻组上发布消息说，作为一种爱好，他开始在 UNIX 模式下开发一个自由操作系统。Stallman 最初的 GNU 项目的许多部分已经完成。在阳光明媚的加利福尼亚州，加州大学伯克利分校已经推出了第一个可自由分发的 UNIX 版本。

自由软件显然已经到来，只是它还没有成为计算机领域中非常重要的部分。

1.2.1 垂直筒仓无处不在

当时，计算机市场格局非常分散。UNIX 市场卷入了一场自相残杀的专有系统战争中。许多其他类型的私有计算机公司依然存在——只是它们已经过了鼎盛时期。

这当中比较知名的是马萨诸塞州的“128 号公路”小型计算机公司。

之所以这样命名，是因为这些公司都位于这条公路上或附近。这条公路环绕着美国东北部海岸的波士顿和剑桥等邻近城市。然而，也有许多供应商为商业和科学计算构建和销售操作系统。大多数人使用自己的硬件设计，从芯片到磁盘驱动器、磁带驱动器、终端等。如果用户买了一台 Data General 计算机，他也许会从同一家公司或少数山寨的计算机附加设备供应商那里购买内存、卷轴磁带机、磁盘驱动器，甚至机柜。

这些公司的软件基本上也是一次性的。一家典型的公司通常会编写自己的操作系统（或几个不同的操作系统），此外还有数据库、编程语言、实用程序和办公应用程序。笔者在 Data General 工作时，该公司有大约 5 种不同的非 UNIX 小型机操作系统和几个不同版本的 UNIX 操作系统。

这些公司中许多人都在考虑向他们自己的 UNIX 版本全面转变。但这主要是针对另一种定制的硬件和 UNIX 操作系统的变体。

那时，大多数计算机系统仍然庞大且昂贵。“Big Iron”是常用的俚语。许多分析师的报告中都充斥着对它们复杂多样的架构的分析和比较。

即使是“小型企业”或“部门”服务器（通常被称为不需要“玻璃房”数据中心等特殊条件的系统），也可能会花费数万美元。

1.2.2 筒仓转向自己的一边

然而，越来越多的 PC 开始被固定在桌子下面，用于处理不太繁重的工作。Novell 公司的软件 NetWare 是此类系统的常见选项之一，专门处理打印或存储文件等常见任务。UNIX 操作系统也有面向大众市场的版本。最常见的是一家名为 Santa Cruz Operation 的公司。该公司通过从 Microsoft 公司购买 AT&T 许可的 Xenix 变体而进入 UNIX 业务。许多年

后，一家名为 SCO 或 SCO 集团的公司发起了一系列与 Linux 操作系统相关的多年诉讼，IBM 和其他公司也卷入其中。

从更广泛意义上来说，整个计算机系统领域正在发生翻天覆地的变化。正如半导体制造商 Intel 公司首席执行官 Andy Grove 在《只有偏执狂才能生存》(*Only the Paranoid Survive*) (Penguin Random House, 1999) 一书中所述，20 世纪 90 年代，计算机行业发生了根本性的变革。

正如我们所看到的，历史上的计算机行业是以垂直栈组织的。这些垂直栈越来越多地旋转成一个更水平的结构（如图 1-4 所示）。UNIX 战争时代见证了从紧密集成的专有服务器栈到单一供应商的转变。但互联网的兴起和许多与之相关的市场力量，才真正扭转垂直栈的局面。

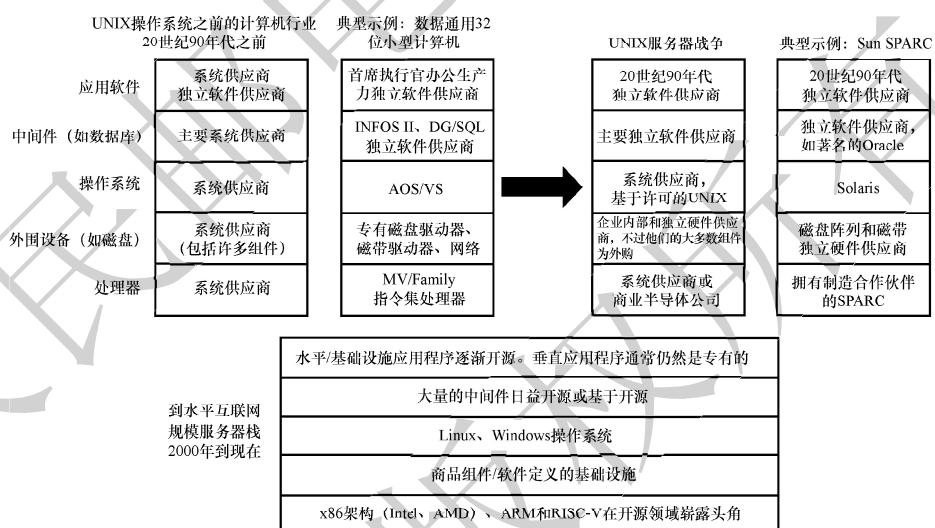


图 1-4 垂直栈的转变

这不是一个纯粹的转变。过去仍然有（现在也有）专有处理器、服务器和操作系统。

但市场越来越多地转向一种模式，即系统供应商从 Intel 公司购买计

算机的中央处理器，通过其他供应商购买各种基本标准化的芯片和组件，从其他公司购买操作系统和其他软件。然后，他们通过直销、邮购和零售的方式销售这些“行业标准”服务器。

在此期间，AMD 公司也在 Intel 公司的授权下生产兼容的 x86 架构处理器，但是随着时间的推移两家公司陷入了各种合同纠纷。AMD 公司后来取得了一些成功，但在很大程度上仍处于 Intel 公司的阴影之下。

PC 模式正在接管服务器空间。

Grove 将其描述为 PC 的 10 倍力量。旧模式的紧密整合可能会有所欠缺，但是，通过一定程度的自己动手，你就可以只花费几千美元让所有东西一起工作，并获得越来越多与工程工作站相媲美的功能。要知道，在过去，你可能需要花数万美元向某家专有的 UNIX 操作系统供应商购买这些功能。

1.2.3 大众市场的哪种操作系统将会胜出

随着 x86 架构日益占据主导地位，现在只需要确定哪个操作系统主导这个水平栈。另一个问题是，谁将更广泛地主导这个水平栈平台的重要方面，如应用程序、数据库以及刚刚开始变得重要的领域，如 Web 服务器。但是，这些都不是紧迫的问题。

答案并不明显。Microsoft 公司著名的 DOS 操作系统和 Windows 操作系统的最初版本都是为单用户 PC 设计的。它们不能像 UNIX 操作系统那样支持多用户，所以并不适合商业用户。商业用户需要能够轻松地共享数据和资源的操作系统。Novell NetWare 是一款多用户的替代品，它在文件和打印机共享方面做得非常好，但它不是一个通用的操作系统。（Novell 公司进行各种尝试来扩充 NetWare 的功能，但是大多以失败告

终。) 另外，虽然小型系统也有 UNIX 选项，但是它们并不是真正的大众市场，也受到前面描述的 UNIX 操作系统分裂的影响。

1.2.4 Microsoft 公司放手一搏

Microsoft 公司决定在其台式机统治地位的基础上以相似的方式统治服务器。

Microsoft 公司对下一代操作系统的首次尝试以失败告终。IBM 和 Microsoft 公司在 1985 年 8 月签署了“联合开发协议”，并基于以开发出 OS/2 操作系统。然而，1990 年 Windows 3.0 操作系统在台式机上获得成功后，两家公司越来越无法消除它们在技术和文化上的差异。例如，IBM 公司主要专注于销售运行在自己系统上的 OS/2 操作系统（从其将备受瞩目的 PC 系列命名为 PS/2 便可见一斑），而 Microsoft 公司则希望 OS/2 操作系统能够在众多供应商提供的各种硬件上运行。

因此，Microsoft 公司开始同步开发一个重新架构的 Windows 版本。Bill Gates 在 1988 年聘用了 Dave Cutler。Cutler 领导的团队为 DEC 公司的 VAX 计算机生产线创建了 VMS 操作系统。G. Pascal Zachary 的著作《观止：微软创建 NT 和未来的夺命狂奔》(*Showstopper! The Breakneck Race to Create Windows NT and the Next Generation at Microsoft*) (Free Press, 1994) 详细记载了 Cutler 推动开发这一新操作系统努力。作者在书中把他描述为一位才华横溢，有时甚至咄咄逼人的首席架构师。

Cutler 对 OS/2 操作系统的评价很低。有人说他对 UNIX 操作系统的评价也很低。在《观止：微软创建 NT 和未来的夺命狂奔》一书中，一位团队成员曾提到：

他认为 UNIX 是由一群博士组成的委员会设计的垃圾操作系统。整个设

计毫无灵魂，因此，他一直想设计出超越 UNIX 的操作系统。而这也是他第一次有机会这样做。

Cutler 负责设计一种新的操作系统，1993 年发布时将其命名为 Windows NT 操作系统。

IBM 公司继续自己开发 OS/2 操作系统，但它未能吸引到应用程序开发人员，所以最终被终止。Microsoft 公司的成功是以牺牲 IBM 公司为代价取得的。这是一个早期的例子，说明了开发人员的重要性以及开发人员思维共享越来越重要。Bill Gates 和 Microsoft 公司早就意识到这一趋势并发挥了相当大的优势。它后来成为开源社区成功的一个关键因素。

1.2.5 Windows NT 操作系统蓄势待发

运行在 Intel 公司的中央处理器上的 Windows NT 操作系统是一款突破性的产品。事实上，Microsoft 公司和 Intel 公司变得如此成功且占据主导地位，以至于人们越来越多地使用“Wintel”一词指代整个行业中占主导地位的系统类型。到 20 世纪 90 年代中期，UNIX 和 NetWare 等其他操作系统都在走下坡路。

Windows NT 操作系统主要基于小型服务器从 UNIX 操作系统手中夺取份额，但许多人认为自己看到了 Wintel 无处不在的未来。除了 Sun 公司的首席执行官 Scott McNealy 以外，UNIX 系统供应商都开始押宝 Windows NT 操作系统。

具有讽刺意味的是，如果没有 Windows NT 操作系统，UNIX 操作系统可能会征服一切。对此，Jeff Atwood 写道：

这个世界已经完全演变成 UNIX 和 Windows NT 阵营。如果没有 Windows NT 操作系统，无论 UNIX 操作系统是好是坏，我们现在可能都在运行它。

这确实发生在苹果公司身上，甚至该公司的下一代 Copland OS 都没有起步。而现在他们正在使用基于 UNIX 的 OS X 操作系统。

UNIX 可能仍然是拥有许多处理器的大型系统的首选操作系统；Windows NT 操作系统最初是针对较小的系统进行优化的。但显而易见，Cutler 设计的 Windows NT 操作系统完全能够扩展并且作为 UNIX 操作系统的替代品。一旦达到这个目的，它很难不围绕着工业标准，就像 Intel 公司的 x86 架构的处理器那样。因为大批量销售的产品单位成本更低，所以 Windows NT 操作系统更容易在新的栈及其各个水平栈上下建立起合作关系和各种集成。

1.3 互联网进入主流

但是，这并不是“游戏结束”的时候。有些事情正在发生。互联网正在起飞，Linux 操作系统的第一个版本已经发布。

到 1990 年，互联网已经以某种形式存在了几十年。它起源于美国国防部高级研究计划局在 20 世纪 60 年代委托的工作，目的是建立与计算机网络的容错通信。然而，除了少数连接到早期网络的机构研究人员以外，其他人可能都没有听说过它。20 世纪 90 年代，基于各种原因，互联网从默默无闻中脱颖而出，其中最重要的原因是英国科学家 Tim Berners-Lee 在欧洲核子研究中心工作时发明了万维网——对今天的许多人来说，万维网就是互联网的同义词——这个欧洲研究组织运营着世界上最大的粒子物理实验室。

1.3.1 从纵向扩展到横向扩展

20 世纪 90 年代后期的互联网建设让很多人受益，包括高端昂贵的

硬件和软件的供应商。Sun 公司、网络解决方案供应商 Cisco 公司、存储磁盘阵列供应商 EMC 公司和数据库巨头 Oracle 公司，这四家公司被称为“互联网四骑士”。似乎每一家风投支持的初创公司都需要给这四家公司开一张大额支票。

然而，许多互联网基础设施（比如 Web 服务器）以及高性能的科学计算集群都运行在大量小型系统上。UNIX 供应商，尤其是 Sun 公司，很乐意为这些目的出售自己的小盒子，但这些小盒子的价格通常比“工业标准”硬件和软件的正常价格高得多。

使用更多、更小的服务器是行业重心从“纵向扩展”向“横向扩展”进行广泛转移的一部分。20 世纪 80 年代，首次出现并流行的客户机/服务器和网络/分布式计算风格的成熟推动了分布式计算。这似乎是 Microsoft 公司的强项。

1.3.2 互联网服务器需要操作系统

然而，Windows NT 操作系统对这些应用来说也并不理想。在小型企业和大型企业的复制站点等市场上，它正在迅速地从 UNIX 操作系统手中夺取份额，以至于像 Santa Cruz 这种向市场销售的供应商都面临着巨大的损失。鉴于定制和技术的原因，网络基础设施和科学计算的角色在历史上大多偏向于 UNIX 操作系统。例如，UNIX 模块化混合匹配的性质使得它长期以来一直受到工匠和 DIY 爱好者的欢迎。因此，过渡到 Windows NT 操作系统既不自然也不容易。

BSD UNIX 操作系统是一个明显的替代方案。它确实取得了一些进展，但非常有限，其中的原因很复杂，也不完全清晰。加州大学伯克利分校与 AT&T 的法律纠纷造成的持续影响挥之不去。BSD 许可证并不像 GPL 一样，要求必须发布修改后的源代码。而且由于 BSD 采用集中式的

开发方式，其社区也不太欢迎外界的贡献者。笔者将稍后讨论其中的一些内容。无论如何，BSD UNIX 操作系统最终并没有对软件领域产生重大影响。

1.4 加入 Linux 操作系统阵营

随着时间的推移，Linux 操作系统与其他开源软件（如 GNU、Apache Web 服务器和许多其他类型的软件）相结合，填补了这一空白。

我们回想一下芬兰大学生 Linus Torvalds。他受 MINIX 的启发开始编写操作系统内核。MINIX 最初是由 Andrew Tanenbaum 出于教育目的而创建的 UNIX 版本。

内核是计算机操作系统的根本。事实上，一些纯粹主义者认为，内核才是操作系统，其他都是更广泛的操作环境的一部分。在任何情况下，它都是计算机启动时最先加载的程序之一。内核将应用软件与计算机的硬件连接起来，并且将管理系统硬件的业务从“用户空间”中抽象出来，而试图使用计算机做一些事情的人关心的就是这些。

1.4.1 一个新的*NIX

Linux 是类 UNIX（有时是*NIX）操作系统家族的一员。UNIX 和类 UNIX 操作系统之间的区别是复杂的、不清晰的，坦率地说，也并不有趣。最初，“UNIX”一词指的是 AT&T 开发的特定产品。后来它扩展为以 AT&T 为授权方的产品。现在，Open Group 拥有 UNIX 的商标，但是这个术语已经成为一个通用术语，在那些无争议的真正的 UNIX 操作系统和那些由于商标或技术特性而被更好地描述为类 UNIX 操作系统之间没有任何有意义的区别。

1991 年，Torvalds 在互联网早期流行的分布式讨论板 comp.os.minix Usenet 新闻组上向全世界宣布了他当时的业余项目。很快他发布了 0.01 版的 Linux 内核。

到了第二年，Torvalds 在 GPL 下重新授权了 Linux 内核。其他人创建了第一批 Linux 发行版，以简化安装并开始将使用操作系统所需的许多组件打包到一起。到 1993 年，超过 100 名开发人员开发 Linux 内核，除此之外，他们还将它调整到 GNU 环境。1994 年 Linux 1.0 版面市，增加了一个图形用户界面到 XFree86 项目，Red Hat 和 SUSE 也推出了商业化的 Linux 发行版。

1.4.2 Linux 操作系统越来越流行

就像 20 世纪 70 年代中期的 UNIX 操作系统和 80 年代中期的 Solaris 操作系统一样，Linux 操作系统最初在大学和计算研究机构中流行，随后开始应用于许多网络基础设施，例如文件和打印共享、Web 和文件服务以及类似的任务。

用户可以免费下载，也可以用便宜的价格购买存储了它的磁盘。在这种情况下，我们需要明确“便宜”的含义。从 20 世纪 90 年代初到 90 年代中期，仍然是 Egghead Software（一家美国计算机软件零售商）等软件零售店和《计算机购物者》（*Computer Shopper*）等大型出版物的时代。《计算机购物者》是一本大画幅杂志，其内容充斥着计算机设备和软件的广告，巅峰时期超过 800 页。消费者习惯于购买盒装软件，包括前面提到的 Windows NT 操作系统，这些软件的价格动辄数百美元。对一家习惯于从 Oracle 等公司购买商业应用程序的公司来说，这似乎是一笔很划算的交易，但大学生和年轻的专业人士并不这么认为。笔者的一位前同事回忆说，他从一家折扣零售商那里花 6 美元就能买到一张 Red Hat Linux

操作系统的光盘，这让他非常震惊，因为价格非常便宜。

1.4.3 黯然失色的 UNIX 操作系统

Linux 操作系统也与 UNIX 程序和功能兼容，有许多软件开发工具可供使用（同样是免费的或便宜的）。它拥有大型集群或某台服务器运行所需的所有应用程序。Linux 操作系统的低成本也意味着它经常被引入公司内部，甚至不需要 IT 管理人员知道，更不用说批准了。

20 世纪 90 年代末，Linux 操作系统和开源还没有形成今天这样的主导力和影响力。但是 Linux 操作系统在 x86 架构服务器上的市场份额已经超过了 UNIX 操作系统。它运行着世界 500 强公司中的超级计算机，是许多基础设施产品的基础，如同在网络繁荣时期销售的“服务器设备”。

甚至到 2000 年，Linux 操作系统已经吸引了来自世界各地的成千上万名开发者。开源开发模式正在发挥作用。

1.5 Linux 操作系统无可避免吗

在这一点上，值得稍作停顿。回顾 2000 年，我们提出一个相当简单的问题：“我们所看到的景象是命中注定的，还是 2000 年计算机行业一连串不可能事件的结果？”毕竟，正如我们所看到的，整个行业都准备将操作系统市场（以及计算机软件市场的很大一部分）让给 Microsoft 公司。当然，几乎每个人都是正确的。Linus Torvalds 和 Linux 操作系统的横空出世是一个将计算机行业推向完全不同的发展方向的随机的、不可能的事件吗？

答案的一部分取决于你是否认同英雄史观。这一理论主要归功于苏格兰哲学家和散文家 Thomas Carlyle，他在 1840 年发表了一系列关于英雄主义的演讲，后来出版了《论英雄、英雄崇拜和历史中的英雄》(*On Heroes, Hero-Worship, and The Heroic in History*)。Carlyle 曾说过：“世界的历史不过是伟人的传记。”这反映了他的信念，即英雄通过自己的个人物质和神圣的灵感来塑造历史。

但这也取决于我们如何看待互联网时代巅峰时期的计算机行业。一种观点可能会把焦点集中在强势的 Microsoft 公司身上，Microsoft 公司在这个时期横扫一切。但另一种观点可能正确地指出，互联网正在颠覆一切。Microsoft 公司对源自 UNIX 操作系统和网络的技术反应迟缓，而这些技术很大程度上脱离了 Microsoft 公司赖以崛起的 PC 世界。

至于所有这些行业的共同智慧呢？长期担任 Sun 公司工程师和 Oxide Computer 联合创始人的 Bryan Cantrill 对行业智慧并不感冒。他认为：

……这个行业错了。这个行业之前已经错了很多次，所以这不应该是惊天动地的大事或新鲜事。但是那些接受 Windows 操作系统的公司有非常严重的深层结构问题。这是一种屈服的行为。而且，这根本不是前瞻性思维。我认为它们一直都是这样的。从 DEC 公司、HP 公司到 IBM 公司，可能最可悲的是 SGI 公司，因为 SGI 公司绝对应该是一个独立的思考者，但行业认为它需要放弃它的未来以给 Windows 操作系统让路。

他接着补充：

我认为，即使它不是 Linux 操作系统，也会是 BSD UNIX 操作系统的变体之一，而它实际上就是 x86 架构上的 UNIX 操作系统。这是互联网的崛起，也是 SMP（对称多处理）在较小程度上的崛起，然后是商业微处理器作为性能最高的微处理器的崛起。Linux 操作系统在这些经济大趋势下抢占了先机，但并没有真正促成这些趋势。

1.6 开源加速

开源软件诞生于 20 世纪，但在 21 世纪，它的巨大影响已经成为一种现象。

在某种程度上，这是因为计算方式的改变有利于开源软件的发展，例如 Linux 操作系统。毫无疑问，开源也起到某种反馈回路的作用，使这些趋势得以放大。

正如我们所看到的，许多早期的开源项目都与网络化的、横向扩展的计算密切相关。最初，这似乎与许多企业 IT 部门处理基础设施和应用程序的方式不一致，后者倾向于纵向扩展和单体架构。需要更多的能力吗？那么请升级服务器，请支付更高的费用，谢谢。

1.6.1 新的企业 IT 模型

然而，到 21 世纪初，许多组织正在改变它们对企业 IT 的看法。正如当时笔者的分析师同事 Jonathan Eunice 在 2001 年的一份研究报告中所写的：

……我们必须明白，构成今天企业计算的要素实际上正在快速发展。它每天都在向网络计算领域迈进并进一步演变。企业 IT 越来越多地通过标准化的基础设施来实施并通过 Internet 协议网络连接进行交付。IT 部门越来越多地以这种方式组织任务和数据中心，服务提供商也是如此。

在“Big Iron”UNIX 供应商的世界中，为了最大化单台服务器的性能和可靠性，它们在垂直可伸缩性、故障转移集群、资源管理和其他特性方面投入了大量工作。随着时间的推移，Linux（和 Windows）操作系

统在与这些需求相关的特性上投入了大量精力。但是，世界越来越不重视这些需求，而是将注意力转移到更为分布式、以网络为中心的工作负载上，这些工作负载更接近最初对开源的需求。事实上，一种观点认为，21世纪初由IBM等公司资助的一些Linux操作系统开发工作过度专注于使Linux操作系统成为更好的、纵向扩展的UNIX操作系统。

1.6.2 诞生于网络

在新世纪中，一种新的商业模式也加速了对开源软件的需求。可以毫不夸张地说，如果没有开源软件，这种商业模式是不可能实现的。仅亚马逊网络服务（Amazon Web Services, AWS）这一家云服务供应商可能就拥有数百万台服务器。在这种规模下，Amazon公司也许可以授权一个操作系统或其他软件的源代码，然后根据它们的需要进行调整。然而，像Google公司、Facebook公司和几乎所有的互联网公司一样，它们主要运行的是开源软件。

在某种程度上，这只是成本问题。特别是对于那些提供服务的公司，它们只能通过广告或其他方式间接盈利（如Google公司免费的Gmail），如果它们需要按照传统的方式为软件付费，那么经济上是否可行不得而知。

这并不是说这些公司使用的所有东西都是免费的。通常情况下，对技术成熟的组织来说，与一些供应商建立商业关系是有意义的。许多大型企业都需要专业的财务、其他类型的软件和软件服务，这些公司同样需要。把所有事情都放在公司内部处理是没有意义的。

尽管如此，从某种意义上说，这些公司几乎是一种新类别的系统供应商，它们生产和构建大量的软件（甚至优化的硬件）以供内部使用。

1.6.3 是构建还是购买

不过，每家公司都需要决定内部的研发重点是什么。这是 Nick Carr 2003 年在《哈佛商业评论》(*Harvard Business Review*) 上发表的一篇题为《IT 无关紧要》(*IT Doesn't Matter*) 的文章的中心论点。软件几乎渗透到企业的各个方面，这使得 Carr 的一些论点显得越来越夸张。(当时许多人也这样认为。) 由于软件在许多层面发挥着竞争优势，公司越来越少地把它当作一种可以轻易外包的纯商品。然而，更广泛一点来说，他认为企业应该专注于那些能够真正实现差异化和获取价值的领域。

开源软件使构建和购买之间的决策权衡变得更容易，因为这不再是一个选择。从历史上看，软件往往是一个要么接受要么放弃的命题。如果你是一个非常重要的客户，而软件不像你希望的那样工作，你可以向供应商提出个性请求，它们可能会在一两个版本中做出回应。通常你可以选择从供应商处购买定制产品。即使这样，你仍然会在改变产品的过程中遇到很多麻烦。

有了开源，公司不需要从头开始构建整个系统，可以选择在某处使用开放源代码而不做任何更改——这可能需要供应商的全力支持，或者根据自己的特殊需求对源代码进行调整和扩充。

1.6.4 打破现状

实际上，这些关于经济的争论回避了开源快速发展这个最重要的原因，特别是在一家公司的背景下。这些原因包括超越了将开源仅仅视为代码的狭隘视角。它要求把开源看作一种非常好的开发模式，同时也是个人和公司进行合作的一种新型方式。

这就是 21 世纪开源的真实故事。随着时代变迁，它早已不是比专有软件更便宜的问题了。噢，开源软件通常是这样的。而且在某些方面，这是典型的颠覆性创新，即它并不完全是关于价格的，但通常在一定程度上涉及价格（如图 1-5 所示）。

颠覆性创新模式

这张图对比了产品性能轨迹（①②线表示产品或服务如何随着时间的推移而改进）和客户需求轨迹（③⑤线表示客户对性能的支付意愿）。随着现有公司推出更高质量的产品或服务以满足高端市场（利润率最高的地方），它们超越了低端客户和许多主流客户的需求。这为进入者在现有企业忽视的低利润领域找到立足点留下了机会。处于颠覆性轨迹上的进入者提高了它们的产品的性能并进入高端市场（对他们来说利润率最高的地方），挑战现有企业的主导地位。

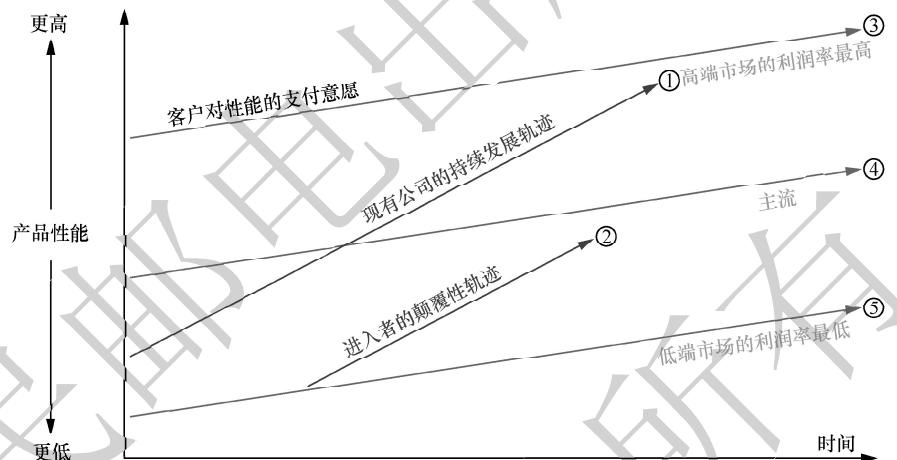


图 1-5 颠覆性创新模式

（来源：Clayton M. Christensen、Michael Raynor
和 Rory McDonald，hbr 网站版权所有）

“颠覆性创新”是哈佛商学院教授 Clayton Christensen 提出的一个术语，用来描述“一种产品或服务最初在市场底层的简单应用中扎根，然后不断向上发展，最终取代现有竞争对手的过程”。

在本书中，我们已经介绍了一个例子——PC 和服务器颠覆了传统的 UNIX 供应商，而 UNIX 操作系统供应商又颠覆了早期的专有系统，例如小型机。

Christensen 还写道：“颠覆性创新让处于市场底层的全新消费者群体能够获得产品或服务，而在过去，只有拥有资金或技能的消费者才能获得这种产品或服务。”相对于 UNIX 操作系统，Linux 操作系统符合这个定义（Windows 操作系统也是如此）。事实上，随着时间的推移，大量现有的 UNIX 操作系统业务已经转向 Linux 操作系统。这让那些想要少花钱的大型用户和技术复杂的用户受益。

但它也为那些可能负担不起专有 UNIX 操作系统的新用户和企业提供了机会。可以说，Linux 操作系统也将成为 Windows 操作系统的颠覆者，因为它在很大程度上将 Windows 操作系统挡在网络基础设施和科学计算市场之外。而在没有 Linux 操作系统或另一种免费 UNIX 操作系统替代品的情况下，Windows 操作系统最终可能会默认占领这些市场。

我们可以将类似的论点应用到许多其他类别的开源软件——数据库、应用服务器和其他企业的中间件、编程工具、虚拟化等。这些开源软件早期的主要优点是用户可以从网上下载并免费试用。但是，实际上，它们在其他方面并不比专有的替代软件更好。

1.6.5 从颠覆到创新发生的地方

然而，在过去 10 年左右的时间里，情况开始发生变化。早期的云计算项目针对的是那些想要自己动手搭建云计算平台的人。这可能是第一个大的转折点，而大数据是另一个转折点。今天，云计算是云原生、容器、人工智能以及许多与它们相关的项目。共同之处在于，开源允许多个来源的创新以强大的方式重新组合与混合。这创造了一种情况，即大量的创新最先出现在开源领域。

在某种程度上，这可能是因为独立的开源社区可以以强大的方式进行集成和组合。

1.6.6 生态系统的崛起

云原生应用程序开发和基础设施空间可能是技术领域中互补和竞争项目结合的显著例子。起初，云原生主要涉及容器本身。在操作系统中，容器是一种隔离应用程序和应用程序组件的高效、轻量级的方法。但是 Kubernetes、Prometheus 和其他数百个项目对它进行扩展，这些项目涵盖容器编排、监控、日志、跟踪、测试和构建、服务发现，以及你能想象到的任何可能对大规模开发、安全、部署和管理分布式应用程序有用的东西。我们还看到云原生和其他领域的持续交集。例如，软件定义的存储和软件定义的网络功能都以不同的方式被容器化。

人工智能和机器学习是另一个重要的领域，其中许多工具和构建模块由开源主导。这包括像 Python 这样的编程语言，像 Kafka 这样的事件系统，像 Jupyter notebooks 这样的协作和共享工具，以及像 Ceph 这样的分布式存储。你还可以看到这个空间和云原生之间的交集，因为像 Open Data Hub 这样的项目使用容器平台来简化复杂的软件部署。

1.6.7 打破单体应用

随着计算领域发展得更加分布式与灵活，开源也随之进行广泛演变。

软件既塑造了计算环境，又反映了计算环境。历史上，大量的专有软件集中开发大型应用程序，而要使用这些应用程序则需要花费昂贵的许可费以及安装费用和定制费用。应用程序之间以及应用程序和数据之间的连接需要更昂贵、更复杂的软件，比如 EDI(Electronic Data

Interchange，电子数据交换）。这符合专有的商业模式。

复杂性实际上是一件好事（至少从供应商的角度来看）。它将客户与单一的供应商捆绑在一起产生咨询费，同时通过所有需要的选项创造大量的追加销售机会，使所有的东西都能一起运作。复杂性也意味着没有真正的方法来确定软件能否像广告宣传的那样工作。在实践中，笔者曾经见过不止一款高端计算机系统管理应用程序最终沦为架子上一个非常昂贵的文件夹，因为它从来没有真正完成应该完成的工作。

即使在 Web 出现之后，最初的尝试仍然反映了传统的做事方式，例如面向服务的架构（Service-Oriented Architecture，SOA），至少在最初的形式中是这样的。SOA 背后的许多核心思想是现代架构模式（如微服务）的一部分，例如将功能分离成不同的单元，通过明确定义的共享格式传递数据来相互通信。然而，在最初实现时，SOA 经常被重量级 Web 服务标准和协议所拖累。

相比之下，今天的分布式系统通常以轻量级协议（如 REST）为特征；可以混合、匹配和改编开源软件组件；同时是一种倾向于编码参考实现而不是经过重量级标准设置过程的哲学。开源将 UNIX 风格的方法（如高度模块化设计）更广泛地引入到平台软件和应用程序中。

1.6.8 Linux 操作系统和开源已经到来

2003 年，IBM 公司播放了名为《神童》(*Prodigy*) 的电视广告。它描绘了一个小男孩坐在那里汲取智慧之珠的能量。好莱坞导演 Penny Marshall 说：“一切都取决于时机，孩子！”哈佛大学教授 Henry Louis Gates 说：“共享数据是走向社区的第一步。” Muhammad Ali 说：“说出你的想法。不要退缩。”在广告的结尾，有人问这个孩子是谁，“他的名字叫 Linux。”

这则广告由奥美广告公司制作，画面引人注目，令人不忍切换频道。时任 IBM 公司全球广告主管的 Lisa Baird 表示，该广告的目标客户是“首席执行官、首席财务官和产品经理”。对这则广告的投资反映了在世纪之交具有前瞻性的个人和组织是如何看待开源的。

Irving Wladawsky-Berger 很早就认识到 Linux 操作系统的潜力并在世纪之交的关键时刻为 IBM 公司运营 Linux 战略。他在 2011 年 LinuxCon 的一场主题演讲中发表了自己的观点。“我们并不认为 Linux 只是另一个操作系统，就像我们不将互联网视为另一个网络。”他接着说，“我们把它看作是未来创新的平台，就像我们看待互联网一样。”

许多人认为 IBM 公司对 Linux 操作系统的投资是大型企业将 Linux 操作系统视为安全选择的重要认可。AWS 的 Matt Asay 告诉笔者：“开源领域发生的最重大的事情之一是 IBM 公司对 Linux 操作系统的 10 亿美元投资……几乎一夜之间，对话发生了变化。”科技记者 Steven Vaughan-Nichols 对此表示赞同，他说：“IBM 公司的认可意味着一个操作系统获得了《财富》(*Fortune*) 50 强企业的官方认可，而在此之前，它仍然被视为只有真正的书呆子才会使用的东西。”

正如 Wladawsky-Berger 在 2020 年对笔者讲述的那样：

我还清楚地记得 1999 年 12 月，我给 IBM 公司负责人 Sam Palmisano 打了电话。我说：“Sam，工作组建议我们应该使用 Linux 操作系统。” Sam 说：“好吧，Irving，我们会这么做的。”我对 Sam 说：“你想什么时候宣布？” Sam 说：“现在怎么样？”我说：“Sam，现在是圣诞节，也许我们应该等到新年。”在 2000 年 1 月的第二个星期，我们发布了一个重要的公告，宣布 IBM 公司将在所有产品中支持 Linux 操作系统。

同样值得注意的是，像 Red Hat 公司、SUSE 公司和其他一些小型公司开始建立业务。他们不仅使用，而且明确地以开源为基础。

2000 年，许多人可能还不清楚 Linux 不仅仅是一个拥有特殊许可证的类 UNIX 操作系统，或者更广泛地说，开源是软件或业务的重要组成部分。然而到 2020 年，很明显，Linux 操作系统和广泛的开源在推动软件创新方面发挥着重要作用。考虑到商业能力与软件密不可分的关系，这意味着开源的发展将推动商业能力向前发展。