

引言

大卫·卡特勒，穿着白色的锐步鞋，白色的裤子和印着“Over the Line”图形的体恤衫。他突然出现在构建实验室，视察有史以来最大、最复杂的个人计算机程序。

现在是早上10:20，这个称为 Windows NT 的程序最新的一次“构建”晚了数个小时。卡特勒是 NT 开发团队的负责人，他对这次延期很生气，对前一天粗糙的测试生气，对整个世界都很生气。他知道小的失误积累起来一定会拖延进度，而他在努力推动项目前进。

卡特勒强调要经常对 NT 做构建或采样，这样他的250名程序员就可以“吃狗粮”了。那是一种糟糕的经历，就像你从头开始建一座房子，同时又住在里面。构建来的越早，他的团队就可以越早测试他们最新做出来的东西，发现问题，并进行改进。

卡特勒皱着眉头，跌落进一个转椅中，怒视着面前的电脑屏幕。他敲了几个键，发出一阵叹息。

卡特勒的不快也传染了其他人。三个构建员，他们负责在计算机的辅助下拼合 NT，在卡特勒情绪激动的时候在他身后忙着自己的事情。一个没穿鞋的战战兢兢的年轻人，摆弄着三个橡皮球。他被一堆计算机围在中间。艾瑞莎·弗兰克林的声音，透过“感觉良好医生”，从墙上的立体声音箱里传出来，充满了房间。

卡特勒对杂耍和艾瑞莎没有兴趣，发出一阵怒吼。他跳起来，把椅子扔出老远，气冲冲地走出实验室。不穿鞋的杂耍者和首席构建员关于 Roloids 陷入了剧烈的争吵，怒目相向。这天一早就充满了敌意。

几分钟以后，卡特勒回到了构建实验室，甚至更加沮丧。他的罗圈腿和结实的胳膊让人想起大力水手。“你们在浪费该死的整个上午，还没有把这个该死的东西搞定，”他呵斥道。然后，他看起来很沮丧地告诉玩杂耍的人：“如果今天你能把这个[构建]搞定，给我电话”。

他的脸更红了，卡特勒再次气呼呼地离开。他的粗线条让他无法掩饰自己的情绪。“你缓解压力的方法就是释放它，”他说。他的做法并没有什么独特之处。门边墙上画着一个圈，那标记了有次他猛踢了一脚的现场，脚趾都踢破了。不久，他用拳头锤墙，本来没什么，但因为他刚好打到一个钉头上，结果折断了一根手指，引起一阵轰动。

卡特勒急躁起来态度不佳，但也是可以理解的。对他来说，现在时间就是一切。他已经比原计划晚了一年，他的团队经过了数年的努力已经疲惫不堪，灰心丧气。现在只有那个巨大的目标在支持着他们。这个团队250名成员的目标是开发一个非常强大和通用的计算机软件，有朝一日每个人都会需要它。在他们面前的是数以千计的程序缺陷，或者称为编码错误，以及从未停止过的对 NT 基本设计的怀疑。这里 NT 的意思是“新技术”。

NT 不是一个单独的个体，它是有许多相互关联的程序组成的一个操作系统。这个软件

将个人电脑变成了一个精密的思考工具，帮助个人或组织控制最有价值的资产：信息。NT 通过一连串令人眼花缭乱的普通动作发挥其能力。好像一眨眼的功夫，它可以打开一个计算机文档，将文字和图片在屏幕上移来移去，打印一封信，计算一系列数字，同时能保证好几个文字处理软件，电子表格和其他应用程序不会互相影响。NT 最深奥的优点是隐藏起来的。正是那些不可见的动作在支撑着计算机，就像人类那些下意识的动作一样——比如呼吸、眨眼、听声音——正是他们维系了人类的生命。

NT 异常的复杂。这个软件由六百万行代码构成，是最错综复杂的人类手工艺品之一。”没人能够完全理解它，”卡特勒说。

像 NT 这样复杂的系统需要一个富有的资助人。有了威廉·盖茨，NT 的资助人实际上是最富有的。盖茨是世界最大的软件制造商微软公司的共同创始人和首席执行官。盖茨希望利用 NT 将他的软件领域从其已经垄断的桌面软件扩展到网络。在1980年代，微软的 DOS 和 Windows 系统软件定义了大多数人使用计算机的方式。在1990年代，这个公司的目标是定义可以电子化的，能够将工人和商业、消费者和家庭联系起来的软件。盖茨希望，NT 的开发将是实现这个伟大梦想的第一步。

盖茨还想要给过去半个世纪以来的计算机革命带来全新的变化。NT 揭开了数十年来将软件从硬件的控制下解放出来的最新章节。直到现在，一台计算机的结构仍然决定了其软件的样子。如果某种特定型号的计算机消亡了，那么软件也会随之而消亡。NT 将软件放在了计算的中心，通过利用当前最好的那些计算机的特性，给程序赋予了一定的通用性——即使在他们选择的那些硬件消亡之后，这些软件仍将存活下来。

NT 的成功立刻成了一个软件的入门读物，一个程序员社区的肖像，以及关于信息时代坚韧的管理复杂而危险任务的情节剧。随着人类和机器都变得完全而毫无保留的依赖于计算机，从空中交通管制到医疗诊断，好的代码对社会的平稳发展至关重要。而编写优秀的代码逐渐成了大团队的工作。然而这样的团队经常会沦为平庸，因为单就其大小就足以滋生官僚主义，没有结果。每个大的团队都面临这样的挑战：在组织好那些才华横溢的成员的同时，又要鼓励领导能力和灵活性。

卡特勒在这种平衡术上表现杰出；他是能够让盖茨梦想成真的唯一合适人选。他既是明星球员，又是明星教练；他设定优先级，亲自编写关键代码，评审其他人的工作。但是他的领导也是有代价的。大多数他的跟随者们过着一维的生活：工作就是他们的全部。朋友退居幕后；婚姻紧张甚至破裂；孩子们被忽视，或者推迟生孩子的计划；业余爱好也没有了。计算机代码意味着一切。如果还能照顾到什么个人的梦想，那也只是为了减轻创造 NT 的痛苦。

那些不愿意沉浸在卡特勒世界中的人——有些人顽固地这样做——可能冒着引起他们领导愤怒的风险，也许更糟，失去他的尊重。那些屈从于他的愿望的人则获得了丰厚的回报，赚到价值数百万美元的微软股票奖金。然而即使这些人也无法确保自己的饭碗。技术变化的步

伐以及竞争的压力让他们的工作生死攸关。他们在工作上倾注了所有的精力，因为别无选择。

卡特勒体现了今天工作场所里冷静的一面。他拒绝承认工作和休闲，职业和家庭，住宅和办公室之间的差别。他非常严厉的道德规范奇怪地令其助手们兴奋。在他带来的世界里，伟大的成就在阴冷的背景下出现。对手们是懒散的、困惑的、没有竞争力的。每个团队成员都渴望超越自己的弱点。“我们的工作非常非常艰苦，”卡特勒说。“若干年以后，这个团队的人回过头来看的时候将充满自豪。他们会说，这是我有生以来最大的成就，我的生活从来没有如此简单。我不用担心自己的职业生涯，不用担心自己的幸福，也不用担心能否处理好与老板和朋友的关系。我只有一件事情需要关心——那就是交付这个产品。并且尽己所能做到最好。”

最初构思 Windows NT 的时候，卡特勒只是模糊地看到前面的道路。开发这个最复杂的计算机软件将他和他的忠实追随者们带入了一个数字荒原的深处。他们一度好像永远也不会在这次探险中找到出口。有些人一度在路上迷失了自己。另外一些则看到他们的生活处在已知和未知的模糊边界之上。直到最后，他们终于知道，每个有价值的创造都交织着爱和狂热。

这就是他们的故事。

目录

译者序

引言

1 编码勇士

2. 代码之王

3. 部落

4. 死胡同

5 嗥叫的熊

第六章 狗粮

第七章 交付模式

8 死亡行军

9 臭虫缠身

10 观止

后记

致谢

译者序

很多人都知道微软公司，它依靠软件变得非常富有。关于微软是如何从一个只有十来个人的小公司成长起来的故事，很多人都听过。听了后，不少人都觉得它很幸运。那么除了幸运之外，微软还有什么过人之处么？很基本的问题，它是如何开发软件的呢？这个问题容易

回答，很多人都能讲上几句。但如果只是讲讲有哪些“工种”，立哪些规矩，也很乏味。要是能从头到尾的讲讲一个很大的产品是怎么从无到有开发出来的，该多有趣。

微软的软件产品很多，但是在其产品线中，起着中流砥柱作用的当属操作系统，即 Windows，中文称视窗。Windows 有众多版本，但不论是 Windows 2000、Windows XP、Windows Vista，还是即将问世的 Windows 7，都是基于经典的 Windows NT 内核。NT 内核从1988年开始酝酿，1989年开始编码，历时将近五年，于1993年7月正式发布，即 Windows NT 3.1（在本书中简称 NT）。通常，软件的版本号是从1.0开始的，但 NT 的第一个版本就是3.1。

NT 的最初开发计划是18个月，但实际用了快5年，是计划的好几倍。软件项目延迟是常有的事，但延迟了这么久还没有被枪毙掉也是有些不寻常的。在行业外的人看来，编写软件的人，特别是所谓的高手，大多有些古怪之处。NT 团队最初只有几个人，结束时发展到两百多人，其中一定不乏特立独行或者离经叛道之士。这些人聚到一起，吃顿饭都可能争吵起来，何况要像绣花一样合作写软件呢，而且是写操作系统这样的复杂软件。

弹指一挥间，距 NT 团队成立的日子差不多20年了。亲历过那段“美好的旧日时光（the good ole days）”的人，都早已声名显赫、腰缠万贯了。请他们讲一下从前的故事，不是一件容易的事。

那么如何满足一下好奇心，或者冠冕堂皇一些，“师夷长技”呢？幸运的是，有一本书很好的记录了开发 NT 的全过程，它是用英文写的，名字叫《Showstopper》。

“Showstopper”这个英文单词很有趣，在观看戏剧或者其它演出时，人们使用这个词来形容节目格外精彩，让人拍手叫绝，以至于被观众的掌声和喝彩声所打断，不得不等人们安静下来后才能继续。在日常用语方面，它也是一个很好的词，人们用它来形容超乎寻常的美丽和迷人……

但当这个词被引入到计算机特别是软件领域后，它的含义发生了根本性的变化，它代表的是最严重的问题。因为在软件领域，人们经常使用臭虫（英文为 BUG）这个词来称呼软件中的瑕疵或者问题，并根据严重程度，将臭虫分成若干个等级，而“Showstopper”用来指最严重的那类臭虫，它如此严重，以至于影响了产品的核心功能，不解决它产品就不能用。如果是在开发阶段，Showstopper 的存在会导致产品就不能发布，其它很多工作也无法进展，整个项目不得不延期……

《Showstopper》一书以时间为主线，生动地描述了开发 NT 的整个过程，讲述了5年中的很多精彩故事，特别是 NT 团队如何解决开发中的 Showstopper 类问题的动人情节。开发软件不是一件简单的事。分析需求、定义功能、设计架构、编写代码、测试集成……，每个环节都不容易做。但难度最大的，最难以控制进度的，其难度又经常被低估的，就是定位和去除臭虫，也就是所谓的软件调试。如果把计算机系统中的一个比特位看作是一个水滴，那么今天的典型计算机系统就都是一片海洋，定位臭虫的过程就好像是在这个海洋中捞针。

而去除 Showstopper 类型的臭虫，更是难上加难，因为这样的问题往往关系着整个产品或项目的成败，关系着很多人的事业和命运。

在介绍紧张的开发过程的同时，作者用简炼的笔法，为我们描绘了一大批个性鲜明的人物，包括他们的学习工作经历、性格、家庭、婚恋，以及他们面对困难和压力时的反应。在作者的妙笔之下，几十个人物跃然纸上。他们有些是公司的管理者，有些是项目经理，有些是团队的技术首领、小组长，有些是普通的程序员、测试员、构建员和技术编辑。他们各司其职，为共同的目标努力，尽管也有争执、摩擦、甚至冲突……

我在四年前买了这本书，放在案头，有空的时候就拿出来读一读，有些内容读了很多遍，但是每一次都有新的感受，能受到不同的启发，有技术方面的，有项目管理方面的，也有处世和人生方面的。

2005年的时候，我开始写一本软件调试方面的书，旨在帮助人们提高去除软件臭虫的效率。在写作过程中，我还经常拿起《Showstopper》，有时是为了小憩一下，有时是为了汲取灵感。有一天，我突然冒出一个想法，为什么不把这么好的一本书翻译成中文，让更多国内的读者可以看到呢？于是我开始想办法联系书的作者。费了些周折后，终于找到了两个邮件地址。于是便写了一封简单的邮件，发了出去，时间是在2007年的5月8日。信写的不是很长，因为担心作者收不到，白费功夫。没想到，第二天一早我就收到了回信。信落款处的署名是 Gregg，也就是作者全名 G. Pascal Zachary 中 G 的完整形式。Gregg 格说，他很高兴收到我的信，愿意让他的书在中国出版，末尾他还说：

Plus, Chinese readers are keen on books about management of tech projects.
(此外，中国的读者很喜欢如何管理科技项目的书。)

是啊，这何尝不是一本软件工程管理方面的好书呢？尽管大多时候我是将其当作报告文学来读的。

接下来的问题就是找一家国内的出版社。这并不像我预想的那样顺利。前两次尝试都很快就失败了。过了几个月后，又有了第三次机会，为了让这次机会成功的可能性大些，我特意邀请了两位编辑见面来谈。我先是花了些时间告诉他们这不是一本专门介绍技术的书。然后我讲了书中的一些精彩内容和这本书对于程序员、测试员、项目经理、公司老板以及历史爱好者的意义。在我滔滔不绝、激情澎湃的讲了一个多小时后，两位编辑有兴趣了，想看一下这本书，然后推荐给社里的领导来审判。过了几周后，我得到了答复：社里的领导不同意。这时已经是2007年底了。当时，我自己写的书已经完成了大半，在写第5篇——“软件的可调试性”时，我很自然地想到了《Showstopper》中的故事是一个很好的案例，征得了 Gregg 的同意后，在我的书中有了“Showstopper 和未雨绸缪”这一节。

2008年初，我又向另一家出版社推荐。这次很顺利，出版社很快决定购买版权。2008年3月，Gregg 很高兴地发邮件告诉我他已经和中国的一家出版社签约了。之后的一些细节

又花了些时间。2008年秋，《Showstopper》的翻译工作开始了。

翻译难。尽管在打算翻译这本书的时候就预计了这个任务会很难，但是实际感受到的难度还是超出了预料。这本书的作者是《华尔街杂志》的著名记者，也许是因为写惯了针砭时弊的新闻稿，他的文笔很是犀利，用词泼辣，豪放不羁。不管是书中的小人物，还是大人物，作者都不加任何粉饰，直言其原貌，甚至缺点，毫不留情。另外，为了力求生动，作者在很多地方使用了口语化的描述方式，用了很多俚语和固定用法的短语。这些特点，使这本书让人读起来感觉很过瘾，但是却给翻译带了很大的挑战。很多俚语和短语是词典里查找不到的，有些即使能查到，但是词典上给出的词意看起来也不符。更何况，如果完全照搬词典上的解释，翻译出来的东西就会失去原作的风采。

就拿书名为例吧。作者使用 Showstopper 这个词作为书名，是有一语双关之妙的。一方面是使用了这个词在计算机领域的含义，即最严重的臭虫，这是对这本书主要内容的一种直白表达；另一方面当然还是使用了这个词的传统含义，把打造 NT 的过程比喻成一场“戏”：一场虽没有刀光剑影，但也绝对堪称紧张激烈的戏；一场发生在高科技领域，但也处处彰显人性光芒的戏；一场有欢乐，也有悲伤的戏；一场让人拍手叫绝的戏……

如何翻译这个书名呢？从打算翻译这本书的时候，我就开始考虑这个问题，也与一些朋友一起探讨过，希望用“头脑风暴”激发出一个好的中文名字。关于这个词在计算机领域的用法，在我们尝试过的词典中，都没有包含这种用法。关于传统用法，词典中一般也只是一句说明，没有给出中文的对应词汇。在想出的很多种译法都被否决后，我们想到了保留这个词不译，这是最省事和稳妥的做法，但还是觉得那样做有些不好，特别是对于那些不太熟悉这个英文单词的人来说。

最近几年，我越来越喜欢阅读古代的文学作品，尤其是当写作或者翻译中遇到困难时，我经常翻看古人的作品寻找答案或者灵感。这一次，在翻看了几本书都没什么收获后，有一天，当我在书柜中搜索其它可看的书时，我看到了一本小小的书，128开本，大约一寸多厚，是典型的“口袋书”。拿起来一看，是上海古籍出版社出的袖珍本《古文观止》。在我很小的时候，就听说过这本书，但是当时还不知道这个书名的含义。后来慢慢明白，观止是“好到极致”的意思，出自著名的历史典籍《左传》。原文是说，吴国公子季札到鲁国访问，请求观赏一下周朝的音乐，于是鲁国派了乐队为他演奏，季札一边观赏，一边称赞。当看到歌舞《韶箛》时，他发出由衷的赞叹：“德至矣哉，大矣！如天之无不帙也，如地之无不载也！虽有盛德，其蔑以加于此矣。观止矣！若有他乐，吾不敢请已！”成语叹为观止就出自于此。

看来汉语中的观止本来也是用来形容精彩的演出，“观”意为观赏、欣赏，“止”即停止。这与 SHOWSTOPPER 这个词何其相似，SHOW 即表演，STOP 即停止，二者加到一起就是令演出停止，后面的 ER 是英文中名词的常用后缀。看来，中国的古人和西方的圣贤不约而同的使用了同一种方式来表达他们对美好事物的终极赞叹，最由衷的情感和最美丽的语言是跨越地

域差异的。想到这里，我如获至宝，决定将“SHOWSTOPPER”翻译为“观止”。

除了封面上列出的译者外，还有很多人为了翻译这本书倾注了智慧和汗水。刘颖翻译了第6章，王宇翻译了第1章的前半部分（初稿）。本书原作者 Gregg 在百忙中回答了我提出的很多问题。我的英文老师 Barry October 帮助我消除了很多疑问。高端调试网站 (ADVDBG.ORG) 的很多网友 (KernelPanic、neilshu、dos、applepc、merryzhao 等) 为翻译这本书提出了很多宝贵的意见，是与他们讨论后，才决定了本书中人名和地名的翻译方式（译为中文，但是在每章第一次出现时在中文后标注英文）。在此向上面的各位朋友以及曾关心过这本书的引进和翻译工作的所有人表示衷心的感谢！

因为译者的水平有限，书中定有错误和不当之处，敬请读者批评指正。

张银奎

2009年2月

第1章 编码勇士 (1)

戴夫·卡特勒 (Dave Cutler) 是在逆境中长大的。他很小的时候就学会了照顾自己，坚持自见，以及如何寻找方法跨过前进路上的障碍。

他1942年3月13日出生于密歇根州首府兰辛。兰辛是一座汽车城，许多生产汽车以及汽车零部件的制造商都在这里安家。卡特勒的父亲名尼尔 (Neil)，他几乎一生都在兰辛的奥斯莫比 (Oldsmobile) 工厂工作，先是在工厂的运输部门工作，后来成为工厂的看门人。

尼尔·卡特勒是一个聪明而且严谨的人，不过他很安静而且没有什么野心。他自小就一直被风湿病折磨着，这也使得他由于太虚弱而不能参加运动。他的视力也很差，使他不能很好的享受户外活动的乐趣。一种苦楚萦绕着他。他不喜欢交际；从某些方面看来，就像一个隐士。在家的時候，他情绪无常，脾气很不好，容易发怒。他喜欢喝酒。

尼尔的妻子叫阿丽塔 (Arleta)，她抚养着他们的儿子戴夫和戴夫的姐姐邦尼 (Bonnie)，她们住在离兰辛八英里的迪维特 (DeWitt) 小镇，她们的房子就在尼尔父母的楼上。迪维特小镇大约有一千人口，四周都是农田，居民大部分是退休了的农民，他们离开农场搬到镇上来往。当戴夫八岁的时候，卡特勒一家从小镇搬出，到了一处有四十英亩大小的开阔地带。这块土地不适合种田，而且还没有房子。尼尔自己修建了一个小小的房子，有一面完全从泥土切割而来。那时，阿丽塔已经又生了两个孩子。这个家庭一起在一个大屋子里生活。阿丽塔维护着一个大花园，他们还在这块土地上种了葡萄树。上千颗树在那里扎根生长。

从十岁的时候开始，戴维德·卡特勒就寻找机会挣钱。放暑假的时候，他把大部分时间都用来为周围的农民干活，包括修建牲口棚以及做一些零散的工作。有一个暑假，他在一所化肥工厂工作。还有一年，他和一个小伙伴一起捡旧报纸，积攒了整整一拖车后卖给废品回收站。

在十几岁的时候，卡特勒被拉去参加体育比赛。因为毕业班上只有34个同学，他所在的小高中硬让他“服役”。他参加了跑步、棒球、橄榄球和篮球项目。在篮球队中，他是副队长，在橄榄球队中，他是四分卫。在一场比赛中，他两次达阵得分，其中一次得分几乎穿越全场。他跑的非常快。

当地的报纸把卡特勒看做一个明星，记载了他的业绩。尼尔几乎从不看他儿子的比赛。在他儿子大学三年级时，有一次，所有队员的父亲都被邀请观看一场比赛，当然尼尔也去了，不过他说他不喜欢运动，不过阿丽塔怀疑是嫉妒让他的丈夫故意不去看比赛。

父子之间的关系并不亲密。甚至还在高中的时候，卡特勒就从家里搬出去过一段时间，先和棒球教练一家生活，后来又和邦尼住。在学校里，卡特勒不用太努力就学的很好。到1960年7月毕业时，卡特勒对自己的前途很有把握。在他体内的某个地方生长着被傲慢包裹着信心和一种信念，他相信他无论干什么都能成为最棒的。其他人也被他这种乐观向上的心态所感染。在他高中的年鉴中，同学们在的照片下，用了这样一句话来描述他的特质：“只有他自己才可能与他匹敌”。

高中毕业时，位于密西根的一所很小的学院愿意招收卡特勒，同意给他体育和学术方面的多项奖学金。卡特勒接受了。这个学院的名字叫奥利维特（Olivet）。在大一学年时，他打的是四分卫，他在场上喊叫和指挥自己的队友就像一个职业球员一般。他传球的技术很棒，而且奔跑的速度很快，百米速度在11秒以内。他那时大概五尺九寸高，175磅重，两条腿又粗又壮。他的教练斯图·帕赛尔（Stu Parsell）把他称为是“万里挑一的好球员”，并对他的闪避能力感到非常吃惊。卡特勒是一个“诡计多端”的球员，他承认他喜欢从别人身边跑过去。

在比赛中间休息时，卡特勒会让大家聚到一起，听他排兵布阵。他不允许别人有不同意见，对队友的失误会大声斥责，并且告诉他们：“这里是我的地盘，既然即加入进来了，那么就闭上嘴巴。”当有人捣乱时，他说：“我真想揍他们，告诉他们要做什么…离开这里做他们的工作。”

在帕塞尔教练看来，卡特勒知道他不能光靠运动的技巧，还需要队友的配合。“他很聪明，知道凭自己一个人的力量是赢不了比赛的，”帕塞尔继续说到：“他拉拢其他的选手跟着他，他们都很拥护他。”卡特勒总是通过自己的模范带头来领导别人，而且蛾”知道他想要的是是什么”，因此队友们很响应他的号召。

卡特勒的体育天才在他大二那年达到了巅峰。在过去的几年中，奥利维特彗星队（Olivet Comets）连续失败了二十一场，长期受压抑的他们在1961年的秋天终于爆发了。在卡特勒这个舵手的带领下，这支队伍在前八场比赛中都赢了。然而，在最后一场比赛中，灾难降临了。当比赛进行到一半的时候，卡特勒在中场抢断，滚到右边，准备开始他的四分卫冲刺。在这个赛季，他已经使用这种方法得过分。这一次，他身边一片空旷，卡特勒沿着边线倾斜着身

躯全力奔跑，刚好经过他们队的座椅，因为距离很近，帕塞尔教练几乎能抓到他。接下来，一个后卫朝他猛冲过来，分身挡住了卡特勒的去路。卡特勒想从他的身上跳过去，但是他的身体与这个后卫的身体成十字形撞到了一起。他重重的摔到了地上，他的腿断了，他的这个赛季结束了。

第1章 编码勇士（2）

在下一个赛季时，卡特勒想返回比赛，但是在开赛前一天的晚上，医生告诉他，如果他上场的话，他的腿就有可能永远都不会好了。卡特勒无奈退出了比赛。

橄榄球生涯结束后，卡特勒把所有精力都放在了学习上。他数学很棒，想过要学理科，但是最后还是决定学工科。在他1965年1月毕业时，有人让他到通用汽车公司做计算机编程的工作。和其他大公司一样，通用汽车公司也开始在上世纪五十年代末期把它的业务记录从纸上转移到计算机里。但是卡特勒并不渴望加入通用汽车公司。他对电脑一窍不通，在他的印象里，计算机似乎很危险，甚至不吉祥。在19世纪60年代中期，许多人对计算机都有这种反乌托邦的看法。对这些咯吱咯吱咀嚼数字的机器，人们还抱着怀疑的态度，有的时候还很反感，因为他们有控制人类的倾向。计算机似乎总是要人类来服从它们的意志，强迫人们除了服从电脑的命令外多做不了什么。

这使计算机的名声很不好，也让人们对为计算机编程这样的工作感觉不好。几乎没有人愿意说自己是个程序员，如果有人真的这样承认，那么人们会觉得他很奇怪。在卡特勒从奥里维特学院毕业前几年时，最顶尖的程序员在荷兰，有一个物理学家在他的结婚证上注明自己是程序员。令他沮丧的是，官方人员以没有这样的工作拒绝了他的结婚申请。

卡特勒对程序员工作持着一种非常墨守成规的看法，他很在意一份工作是否能给他带来自尊和地位。对于一个出生相对贫穷、并决心依靠自己的力量攀登经济高峰，不向权势低头的年轻人来说，编程“似乎是种非常缺乏创造力并且枯燥的工作”。那些做程序员工作的人遵循着“一套固定不变的规则”，不是能规划自己蓝图的领导者。

他不想搭上软件的边，直接拒绝了通用汽车公司。相反他进了杜邦公司。他很快就适应了这个稳健繁荣的化学巨人。他保持着短发和军人那样的胡须。他第一个想到的就是要挣钱；他和一个在大学里认识的女人结了婚，而且已经是一个孩子的父亲了。

杜邦公司把卡特勒安排到一个帮客户寻找材料用途的部门工作。他接手的第一个工作是为一种新的方法建模，斯科特纸业（Scott Paper）打算用这种方法来制作泡沫保温棉，用在夹克衫和其他外套上。这个模型非常复杂，需要使用电脑来创建。于是卡特勒只好放下了所有的事情去了一所由 IBM 经营的学校，在那里他可以学到怎样给 IBM 电脑编程序。

卡特勒在学校里呆了一周。他感觉到自己的锐气第一次受挫了。编程“是最奇怪的事情，因为你过去习惯了做一些事情，而且你觉得你做的是对的，”后来他说：“但实际上错了。你

只是没有注意到它是不对的。电脑对差不多正确是没有一点宽慰和原谅的。差不多就意味着你还是错的。”

即使对于很有经验的程序员来说，他们也会发现他们的工作是非常枯燥无趣的。当然，在那时，还没有人自己拥有电脑。通常是十几个的程序员共同分享一台大型计算机。这种大型计算机的大小足以把整个房间塞满，它能够成批的把很多工作一下子处理完。在批处理任务中，程序员把指令打到穿孔卡片上，向队列中增加一个栈，然后等待结果。因为这种大型计算机很贵，所以批处理任务的日程安排是非常严格的。很多时候，要花好几个小时甚至更长的时间才知道一个程序的命运。如果失败了，单单去纠正穿孔错误就可能要花上程序员一整天的时间。

卡特勒回到了杜邦公司决定在程序上有所成就。这个工作激起了他的兴趣，因为在程序世界里，他是那个环境的控制者。他还发现他有一种神奇的能力，可以一下子把七零八落的不同程序片段都记在脑子里。他开始渴望编程。他没有耐性排队等待杜邦公司电脑设备，所以他在半夜工作，那时用计算机很便宜而且他能够不慌不忙的汇编和修正他的卡片。”几乎没有任何人在那儿，”他回忆说：“出错误的时候我可以再试一次，在我想用的时候就用，不想用的时候就不用。”

相对于整夜编程来说，做保温泡沫并没有让卡特勒晚上睡不着。不到一年的时间，他已经被电脑的魅力深深地吸引了。在电脑上找到解决问题的最佳方法之后，他很快就对问题失去了兴趣，而疯狂的爱上了电脑。事实上，卡特勒找到了人生目标。”我真正想要的是在电脑上工作，而不是用他们来解决问题。”

于是，卡特勒开始寻找一个需要编程的新工作，他找到了杜邦公司的另一个部门，这个部门需要有人来帮助维护 Univac 制造的中央电脑。二十世纪五十年代时，Univac 生产的电脑在处理数据方面是最好的，但是到了二十世纪60年代后期，这个公司开始走下坡路了。杜邦公司叫卡特勒提高那些使用多年了的 Univac 电脑的可靠性，这意味着要调整机器的操作系统。在那时之前，卡特勒甚至从来没有考虑过操作系统。不过，公司的电脑专家看起来知道的也不是很多，于是他全身心的投入去研究。

可以把计算机程序大致分为两类。应用程序，简称为“应用（apps）”，是软件世界中看得到的那一部分。这一类包括被普通用户使用的程序。举例来说，应用软件可以跟踪定单或者存货，检索名字和电话号码，准备要打印的文档或者控制时事通讯的设计。

另一方面，操作系统也是软件世界中看不到的那一部分。他们是计算机的心脏，在背后跳动。表面看起来似乎所有的事情都是应用软件做的，但是事实上很多任务都是操作系统做的，比如打开文件、关闭文件、为要储存信息的建立目录、指挥计算机的输入、输出、存储和网络设备。

在数字计算机逐步成形的年代里，也就是在第二次世界大战之后，在设计者们眼里，无

论是操作系统还是应用软件都是以后再考虑的事情。和由程序组成的“软件”不同，由电子电路组成的“硬件”太难了，以至于工程师们不敢想像何时能完成它。硬件中，最重要的一类是真正执行送给计算机的指令的电路系统，也就是处理器。第二类设备用来把数据输入进来或者从电脑输出出去。第三类设备用来储存信息。第四类设备允许把信息通过专门的电缆或者电话线发给另一台计算机。

通常，只有在硬件到位后才考虑软件问题。因此，设计计算机时并不知道以后运行什么软件，程序员只能做电脑硬件允许做的事情。艾兹格·迪科斯彻（E. W. Dijkstra）是计算机编程方面的著名理论家，他曾经概括了在计算机形成时期人们对编写编码的普遍态度。他这样说：

可怜的程序员是什么处境呢？说句实话，他们很难被注意到。首先，起初的计算机太大了，你根本就不能移动它们，除此之外，它们需要太多维护工作，以至于很自然的，机器是在哪里开发的，人们放在哪里使用它。其次，程序员做的某种程度上来说根本看不见的工作没有什么魅力：你可以把机器展示给参观者们，那比几页代码的吸引力要大几个数量级。但最重要的是，程序员自己对所作工作的看法也很平常：他的工作的意义来源于计算机这种超棒的机器的存在。因为这种机器是独一无二的，所以他知道他的编程只在本地有意义。因为机器只会使用不长一段时间，所以他知道他的编码的延续价值很少或者没有。

编程工作表面上看起来很简单。只是写个请求给电脑。这种请求应该是以电脑能够明白的方式表达的。而且，这种方式只有特定的电脑能够明白。同样的请求，完全以同样的方式编写，送给有着不同设计和电路的计算机，就会变得莫名其妙。

除了成为特定计算机的奴隶外，最早的计算机程序是很简陋的。在二战之前，当时的计算机主要是机械的，很多时候，程序的价值和人工反转开关、重新安排线路或切换次轮的价值差不多。在二十世纪三十年代时，要让那个时代最强大的机械似电脑——微分分析仪（Differential Analyzer）——去解决一个新的问题，准备工作就要花很多多天。十年之后，要让早期的数字计算机去解决一个比较难的问题，仍然要花几天时间做设置工作。

更灵活一些的机器从打孔卡片或者纸带上面读取用户的请求，不过卡片或者纸带还是手工喂给机器的。这种初级的编程方式迫切需要进步。

突破在1944年发生了，约翰·冯·诺依曼（John von Neumann）是出生在匈牙利的数学家，住在美国，他提出了存储程序的概念。这个想法与这个领域中的其他人的想法很相似，但是冯·诺依曼最清楚的看到了它的意义。使用存储程序，送给计算机执行的指令可以保存在机器自己的内存中，用对待数据的同样方式来对待。这将大大提高启动一个程序的速度，而且可以很容易的修改程序，或者从一个程序切换到另一个程序。

第1章 编码勇士（3）

随着存储程序的概念在初生的计算机文化中的广泛传播，编程变得越来越引人注目，很快吸引了一些拥护者。不过这是很艰难的一件事。数字计算机有两种状态，开或者关，所以它只能对只包含一（开）和零（关）的二进制的消息做出响应。程序中的所有元素最终必须使用这两个数字表示出来，这样使得即使普通的数学运算也很容易变得很复杂，令人眼花缭乱。正如一个观察家所说的，在二十世纪四十年代，为计算机编程是“令人发疯一般的困难。”

不常时间后，程序员们找到了更简单的方法来产生二进制串。他们首先设计了特殊的打字机，可以自动的吐出二进制代码。后来，他们转移到更友好的“汇编”语言，使用字符和符号来代表一和零。使用汇编写程序是一大进步，但是还需要严格准寻晦涩的计算机指令集。程序员必须深刻理解指令集才能写出高效的汇编代码。而且，不同型号的计算机使用的指令集是不同的，取决于它们的微处理器是如何设计的。这意味着一旦某种计算机被废弃不用了，那么程序员费劲力气学会的这种汇编语言知识就会变得一钱不值了。

到二十世纪五十年代时，严重依赖计算机的各种机构开始意识到了软件是令人头痛的事，而且是代价很高的，其中最主要的机构就是美国军方的三大分支。站在计算机领域前沿的程序员们开始寻找方法，以便可以更简单的编写出高效的程序。在1951年，格蕾丝·莫里·赫伯（Grace Murray Hopper），美国海军军械储备局的一名数学家，构思出了一种称为编译器的程序，它可以把程序员的指令翻译成二进制串，也就是最终控制计算机的机器语言。理论上来说，编译器正是可以把程序员从硬件的虐待和使大脑麻木的二进制代码中解放出来的东西。

赫伯的成果催生了无数人的努力，来简化编写代码的难度。或许，最重要的当属 IBM 开发的成为公式翻译器（Formula Translation）的编译器，也就是 Fortran。它包含了三十二条指令，诸如 PUNCH（打孔）、READ DRUM（读取磁鼓）和 IF（如果） DIVIDE（除法） CHECK（检查），它们来源于计算机硬件需要的实际二进制术语。截止到二十世纪五十年代后期，Fortran 产生了巨大的影响。“现在，任何人只要具有好的逻辑思维和愿望就可以学习计算机编程，”一个计算机历史学家曾经这样写道：“你不再必须是熟悉计算机内部原理和汇编语言的专家。使用 Fortran 的简单命令，你可以让计算机听从你的吩咐，而且编译器会自动的把你的指令翻译为高效的机器代码。”

尽管 Fortran 可以让程序员可以使用同一套指令来为任意数量的计算机编程，但是要在不同的机器上运行 Fortran 程序时，经常还需要修改。而且，Fortran 是针对科学计算和工程问题的。于是针对其它目标的其他语言出现了，比如面向商业的通用语言（Common Business-Oriented Language）（简称 Cobol）。没过多久，便有很多种语言可供程序员选择

了,而且,很多时候他们的职业路线就是由他们所选择的和学习的最好的编程语言所决定的。

赫伯深信克服因为计算机语言激增而导致的困难将成为未来最大的技术挑战之一。“对我来说,编程不仅是一种重要的实践技术,”她在1961发表的一次讲演中说:“它还需要在基础知识方面做巨大的投入。”有些出乎意料的是,她担心最大的前进障碍会来自程序员他们自己。就像转变到一种新的宗教信仰一样,他们经常表现出思想保守,包围了他们的狂热。“程序员是非常古怪的一个群体,”她观察发现。

他们成长的非常快速,很短时间内就变得很专业,而且很快就感染上了一定程度的抵制改变的毛病。有一个程序员,我听说他差点和客户吵起来,因为他不想修改它的系统,他几次走进我的办公室说:“我们一直是使用这种方法来做事的。”这是因为这个原因,现在我在我的办公室里挂了一个逆时针方向走动的钟。

在二十世纪六十年代初期,IBM 尝试在软件领域掀起一场革命,能让同一个程序在任意数量的计算机上运行。IBM 计划开发一系列机器,这些机器覆盖了大多数市场,它们是由一个操作系统控制的。这个耗资五百万美元开发的 System/360产品线取得了巨大的成功。但是它的产生过程是很痛苦的,主要是因为创建软件的难度和成本被大大低估了。第一台360硬件是在1964年问世的,直到五年后它的所有软件才能很好的运行。到那时,IBM 在编写软件方面所花的钱差不多已经与设计硬件一样多。这使 IBM 公司的经理们非常吃惊,也非常生动的显示出了“影响计算机科技进步的最大障碍”和管理大型软件项目的问题。

在杜邦公司,戴夫·卡特勒发现自己陷入了驯化计算机这项大任务的第一线。操作系统的不断标准化大大提高了计算机的实用性。但是编写应用软件的主要压力还是落在了计算机购买者他们自己身上。这产生了一种没有想到的效应,那就是使很多大公司成为孕育程序员的基地。杜邦让卡特勒创造一个程序来分析实验室里的实验数据。这个工作很需要技巧,因为需要两台计算机一前一后一起工作。一台接收数据,将数据放入到一个文件然后发送到第二台机器,第二台机器分析数据并把结果发送给研究人员。

人们希望这个程序是所谓的“实时”系统,它能使计算机接收到信息后立即做出回应,而不是几个小时或者几天后才给出结果。第一个实时系统是在风轮计算机上创建出来的,供空军和海军跟踪敌人的飞行器,并指导美国的攻击机命中目标。

实时程序是非常有价值的一种创新。对于像杜邦这样努力寻找新材料和材料用途的公司来说,快速响应的帮助比延迟相应要大很多很多。杜邦想让卡特勒的程序运行在从数字仪器公司(DEC)购买的小型计算机上。DEC 是十年前由风轮计算机的一个工程师创立的,DEC 是小型计算机领域升起的一颗新星,它大胆打破传统。在过去,计算机设计者们提倡可以让许多任务共享运算能力的大型主机。小型计算机的价格经常低于十万美元,这第一次让人们可能分配一台计算机专门做单一的任务,比如跟踪零件的轨迹,实验数据或者机器工具的操作。

尽管有用，但是像 DEC 的 PDP 这样的小型机通常只配备很少的软件。购买者不得不自己设计他们需要的软件。在接下来的几年中，卡特勒编写出了实时程序和其它一些 PDP 软件，成为为 DEC 计算机设计代码的一位专家。

但是，卡特勒开始厌倦杜邦了。他想要到一家计算机公司工作，于是他选择了 DEC。因为钻到了 IBM 产品线的一个空子，DEC 不仅成长的非常迅速，而且摒弃了传统的商业模式。它没有征购土地建设办公场地，而是把波斯顿郊外的一个废弃工厂改造为员工的办公室，这家工厂位于马萨诸塞州的梅那特。DEC 也没有坚持 IBM 青睐的从上到下的管理方式，相反，DEC 允许工程师实践他们自己的想法，即使冒着重复别人的风险。

卡特勒很渴望为 DEC 的计算机编写软件——他如此渴望，以至于和 DEC 销售人员的一次偶遇便引发了他到这家公司去参加工作面试。随着硬件的不断发展，DEC 很需要编写代码的强人。卡特勒刚好满足了这一需要。于是在1971年，卡特勒接受了 DEC 的工作，搬到了马萨诸塞州，这一年他二十九岁。

卡特勒很快就博得了 DEC 的青睐。没过多久，他就成为这家公司的一位软件之星，并让他带领了一个开发代码的团队。他完完全全被他的工作所吸引了。在那时，他的第一次婚姻已经破裂了，第二次正朝着浅滩行驶。正像他自己所承认的，他根本不是会照顾家的男人。当他终于和第二个妻子散伙后，他发誓再也不会结婚。“婚姻是一种错误，你只能犯两次，”他说。

卡特勒是一流的程序员。他全身心的投入，沉迷其间，争强好斗。他有非常坚强的意志和信心。他投入非常大的精力来注意细节。而且，他在设计和编写代码时不停地追求更好。“大多数人学会如何把一件事做得很漂亮后，以后一生便一直做这个，”一个同事观察发现：“但他不是卡特勒。卡特勒会从自己的成功中学习。下一次，他会做的更好。所以每次，他都上升到一个新的高度。”这一点是很令人吃惊的，因为在科技方面取得成功的很多人也没有这样的特质。而且，他非常的投入——除了分配到他手中的软件任务外，他对其它任何脑力活动都没有热情——这是很有好处的：“对可能干扰他的任何人和任何事，他不仅置之不理，而且还会对其进行攻击和诋毁。”

第1章 编码勇士（4）

卡特勒骂人的功夫也炉火纯青。反复无常而且脾气倔犟，他可能根本不顾及脸面，大喊大叫，根本不顾及丧失最起码的礼貌。他发脾气时，嘴里动不动就冒出一连串污言秽语。每次新的爆发都把他的形象又磨砺一番。陌生人是受不了的。有一次，一位女工程师在放满打印机的房间里第一次遇到卡特勒，卡特勒正在那里把卡在打印机里的碎纸往外抠。这位女工程师以期待的眼神看了卡特勒一眼，这时正好卡特勒也抬眼看见了她。于是，卡特勒咆哮起来了：“你就是那个把打印机搞垮的混蛋么？”

尽管他的态度没有给任何人留下好印象，卡特勒还是赢得了称赞，因为他在为 DEC 的

DPD-11计算机构建实时操作系统的工作中起到了关键作用。他出色的完成了一项非常讲究技巧的任务，可以把程序缩减的更小。这是至关重要的，因为程序越小，它便运行的更快，消耗的内存也越少。卡特勒用了一种很简单的方式来表达他希望代码更紧凑的愿望，他在他的办公桌上放了一个橡皮戳，上面刻着：大小就是目标（Size Is The Goal）。当他觉得程序员要增加的功能会大大增加内存需求时，他就会使用这个戳盖在驳回备忘录上。这个戳让程序员们对增加代码更加谨慎，同时也提供了一个笑料。很快，很多男员工的房间里都出现了一句刺眼的短语：“大腿就是目标（Thighs Is The Goal）”。

对于他的同事来说，卡特勒是一位英雄，一个不能用平常规则来评判的超人。“他让你感觉你是他的一位同伴，一起成就伟大的同伴，而且你再也不会有更好的和更值得信赖的同盟，”一个同事这样说。卡特勒希望所有人都表现得很完美。为了传达紧急的消息，他会把大家召集到一起。他喜欢奚落他的伙计们，但是为了改善他们的代码质量，他会不遗余力。有时，他甚至捡起被扔到垃圾桶里的代码，用红笔纠正了上面的错误后还给代码的作者。

卡特勒不仅很严厉和冷酷无情，他还要求他的团队要绝对诚实。“如果你对他花言巧语放烟雾弹，或者你试图欺骗他，那么他会残忍无情的，”一个同事说：“他不会只是让你自杀，他会攻击，而且他不会忘记。”

他也很吝啬表扬别人。罗杰·海恩（Roger Heinen）是卡特勒很喜欢的一名程序员，他经常渴望他的导师可以对他说句好话。“当你不在那里犯了错，他批评的很快，”海恩说。在经历了这样的一次打击后，海恩想知道卡特勒是不是还很看重他。因为太害怕而不敢问，他把手伸进办公桌里，取出一张字条，那是卡特勒一段时间以前写给他的。纸条写的很简单：“罗杰，你真的把工作做得很好。谢谢，戴夫。”

一个强有力的贵人让卡特勒有可能在 DEC 红了起来。在早期，卡特勒对待编码的认真态度和积极主动的领导风格吸引了戈登·贝尔（Gordon Bell）的注意，戈登·贝尔是 DEC 的最高工程官员，也是大多数重要计算机产品的设计者。贝尔是很少见的既有技术天赋，又有策划才能的人，他在1960年加入 DEC，当时他二十六岁，刚 MIT 毕业。在接下来的十五年中，他“领导、论证、思考并建立了很多计算机策略，这些策略把 DEC 变成 IBM 的最强大挑战者。”

工程师们很喜欢贝尔，喜欢他的温和活泼。他的关心问题的非常广泛，这种广度既让人高兴，又让人生气。他的思想会到处漫游，以至于那些真正想和他谈话的人要努力把他隔离起来，防止他的注意力转移。“要和戈登交谈，你必须把他带到小汽车里，行驶而且不让他打开收音机，”一个人说。

在1975年年初，贝尔觉得 DEC 正在失去前进的动力，尽管财务报表还很不错。当时，PDP-11计算机已经面世五年了，而且 IBM 正准备引入它的第一款真正的小型机。贝尔觉得 DEC 需要大大改进的机器来在 IBM 面前保持领导地位。经过了整整一个冬天，贝尔构思出了一个

新的计算机产品线，通过这个产品线他认为会让 DEC 继续走在竞争对手的前面，而且同时也会让那些已经在 PDP-11 上投资了很多钱的客户满意。这个新的产品线被称为 Vax，会运行一个新的操作系统，被称为 VMS，VMS 会允许用户既可以运行新的程序，又可以运行的为 PDP-11 编写的旧程序。“向后兼容”意味着 Vax 一问世后，就可以在上面运行许多现存的程序，而不是要等待程序员编写新的程序，这样的事情以前经常发生。同时，Vax 的设计是“可调整的（scalable）”，这意味着同一个软件可以在不同配置的 Vax 计算机上运行。用户可以在保留他们熟悉的软件前提下，切换到更高配置的 Vax。

这一切表明 Vax 会成为计算机历史上的一道分水岭。在组建团队时，贝尔挑选了 DEC 中最聪明的工程师。他要卡特勒来领导创建 VMS 操作系统的团队，在贝尔心中，卡特勒是 DEC 的一流程序员。

把卡特勒和其他四个高级工程师召集到一起后，贝尔在 1975 年四月 1 日开始了 Vax 项目。这个小组开了很多火辣的会议来反复推敲设计概念。“他们开始时很安静，先花十五分钟简要介绍议题，接下来就不可避免的进入到了大声争论，表面看来一片混乱，而且像是很敌视对方。但不知什么原因，他们总是能高调的结束，工程师们面带微笑鱼贯而行走出会议室。”

贝尔帮助卡特勒，不让他受到那些爱管闲事的局外人的干扰。“我不会容忍任何人插进来影响戴夫做事，”他说。即便如此，VMS 项目还是让卡特勒承受了巨大的压力。这种压力体现出来了。有一些早晨，当他从床上爬起来时，会感到头晕眼花，甚至不能把身体站直。化验表明他的血压非常高。他立刻开始吃药来把血压降下来。作为一种预防措施，DEC 让 VMS 团队中的其他经理也检查了血压。做检查的护士说，卡特勒的血压是见到过的最高值。

经常受卡特勒唇枪舌剑之苦的同事们不由自主的会拿卡特勒的健康问题来奚落他。在异常非常激烈的争论中，一个同事开玩笑说悬而未决的等待令他痛苦不堪。还不如卡特勒的心脏受了伤害，那样他们就可以不再为这个项目忧虑了。“难道你就不能一下子把这个项目做完了事？”他解释说。

很清楚，这个时候卡特勒需要有人来帮助他开发 VMS 软件。这个团队很小——最多时只有十个人——而且卡特勒是高级程序员。但是卡特勒不会求人帮忙。他从来不做那样的事。他有一句他很喜欢的话：“当把所有事情说一遍然后再做完时，要说的话比要做的事多得多。”

当卡特勒忙着做事时，这个项目的总经理（被卡特勒戏称为“老板”的男人）打算雇佣一个高级程序员来帮助卡特勒把 VMS 软件弄成样子。当卡特勒离开一周去休假时，这个老板面试了最好的候选人，并在卡特勒回来时雇用了他。卡特勒不觉得他需要一个签署代码的人，但是实际情况证明，这个新来者在技术方面也是胜任的。卡特勒默认了这个帮助，什么也没有说代表了他的感谢。

1977 年十月，第一台 Vax 计算机问世了。它的巨大成功把卡特勒推到了一流软件设计师

的前列。举例来说，贝尔就毫不犹豫的称卡特勒是这个世界上操作系统的最好编写者。财富和荣誉——成就的体现——都送到了卡特勒面前。但这些没有一样能改变他的脾气。DEC 中日益增涨的官僚作风令他讨厌。“我是一个实干家，但是做事情变得越来越艰难，”他说。令他最难以接受的是，他再不能以他习惯的管理方式来指挥 VMS 项目。因为这个程序现对公司的前途如此重要，所以一群经理都希望来改进它。卡特勒发怒了：“这个项目给所有的汤姆、迪克和哈里提供了机会，无论你要做什么，他们都过来找茬，挡住项目的去路。当然，应当有一定数量的审查。那是正常的。但是和这个项目根本不相干的人也会突然出现，询问为什么你在做这个，为什么你在做那个？”

为了少受滋扰，卡特勒离开了 VMS 团队。他想要摆脱 DEC 这种荒谬可笑的气氛。有一天，他威胁说要离开公司，告诉贝尔他希望能自己成立一家公司。贝尔的回应让卡特勒无法拒绝：“带上你想要的任何人。去你想去的任何地方。做任何你想做的事。而 DEC 会支付所有费用。告诉我你需要多少钱，我们会给你提供资金。”

卡特勒兴高采烈的接受了这个机会。在1981年春天，他把他的忠实伙伴聚集到了一起，希望搬到离 DEC 的马萨诸塞办公室尽可能远的地方。他在西海岸区域旅行寻找新的家。在西雅图呆了一天，卡特勒做出了决定。吸引他的有那里的山（他喜欢滑雪）、野物（还喜欢打猎）和太平洋。他在一家可以俯瞰秀秀湾（Shilshole Bay）的饭店享受了一顿很好的海鲜大餐，这让他的情绪焕发了活力。因为从波士顿到西雅图没有直飞的航班，如果他的公司主管们想到他这里来就要花更多的时间，这一点进一步增加了这个地方的魅力。

卡特勒很高兴有了自己掌管的实验室。“我们知道我们想要建造什么，而且我们可以就在这里做所有决定，”他说。贝尔准备好了推销卡特勒的想法。在这个实验室里，卡特勒的穿着很随便，大家很熟悉的汗衫和胶底运动鞋，连那种军用的所谓“高质量 T 恤衫”他现在都不穿了，在东部时，他经常穿这样的 T 恤衫。每周有几个晚上，他会邀请所有人去酒馆，在那里，他会毫无顾忌的喝下大量当地酿造的一种酒，称为 Red Hook，卡特勒将其当作是天赐的琼浆。在冬季，他会带领他的伙伴们去滑雪旅行，很多时候，他比年轻十几岁的人滑得还好。“和他一起滑雪可不是好玩的，”一个同事说：“我只能在后面吃他掀起的雪。”

卡特勒是有远大志向的，但是人们对他的计划反响不一。他希望和软件一起设计整个计算机。DEC 同意把他的一款计算机推向市场，但是这种支持在1983年贝尔离开 DEC 后消失了。“当戈登离开时，我们在 DEC 的支持网络开始消散了，”海恩说。一下子，卡特勒缺少了神通。“很清楚搬到西雅图是一个错误，”一个同事说：“距离总公司那么远，他没有能力来有效的拥护他的想法。”

不过，他还有他的自由。在他的名片上，他毫不害羞的把自己称为“最高指挥官（Supreme Commander）”。他的同事把一些有趣的照片收集在一起，成为实验室的“年册”，有一张照片，卡特勒穿着没有袖子的白色汗衫，一条很便宜的条纹领带松松垮垮的绕在他的脖子上，这是

卡特勒故意嘲弄公司斯文礼仪。还有一张照片，据说是他的办公桌，可以明显看到的有三样东西：一个喷雾罐，上面写着“牛屎驱除剂（Bull Shit Repellent）”；一个由“英国鸡务局（British Chicken Authority）”签发的证书，上面写着：你的公鸡鼓起的够么？还有一个喝水的杯子，上面带着一句标语：快乐是紧紧的女阴。

可以把卡特勒的粗话平衡一下的是，他信仰和追求卓越。平庸是留给弱者的，要成就伟大就要有愿意去打破荒谬或者突破传统（而且不管什么后果）。”首先，质量是每个人都必须坚信的东西，”卡特勒在年册里写道：“这意味着要管理到所有层面，从上面到最下面的工人。不给那些总是顺着市场部门的风向倒的无能经理们留一点空间。也不给工程师们留空间去通过降低质量伪装做了更多的工作。要站起来对质量问题说不有时需要勇气，不过我项目里的所有人都拿到了这样的许可。如果某个地方的某个笨蛋想要发布没有准备好的软件，那么我们会告诉他滚蛋。”

在1985年，卡特勒勉强赢得了DEC的批准，设计一个新的计算机系列，称为Prism。它的硬件是基于内部创造的一款芯片。它的操作系统称为Mica，会是一个全新的设计，尽管它有能力运行Vax应用程序。卡特勒召集近两百人做这个项目，一切完全像一个独立的公司那样运作。他设计的第一款Prism计算机是供科学家使用的，供他们做专门的向量计算，以非常高的速度进行计算。这样的运算可以帮助科学家们模拟所有东西，从天气到核爆炸，到合成大洋的洋底。通常，这样的数学密集型仿真需要所谓的超级计算机，价格在几百万美元。因为创新的硬件和软件，第一款Prism计算机的价格可能只是超级计算机的十分之一。

正当卡特勒的团队抓紧开发Prism时，DEC其他部门的竞争者对他的计划下手了。到1988年六月时，卡特勒已经被要求修改和放弃很多个项目，直到有一天，他飞到DEC总部，被告知Prism取消了，这时他已经不那么震惊了。DEC希望解散卡特勒的硬件小组，强制要求近一百名工程师在公司的其他部门寻找工作。卡特勒觉得这是“最后的稻草”。在返回西雅图的飞机上，他喝了很多酒。第二天早上，也就是1988年的六月18日，当他走进挤满员工的会议室时，他似乎还在醉酒的状态中。他无法忍受DEC拆散他的人。他可以承受从这里走开；他有二十万美金的年薪和一套两百万美金的房子，位于普及特海湾（Puget Sound）。

在说话前，他看了看大家。一些人开始哭了；其他人像冻僵了一样。几个跟随卡特勒多年的老战友甚至无法面对被挫败了的卡特勒，他们去酒吧避开了这次会议。

“你们都显得这么悲伤，”卡特勒说。

然后，他告诉大家：Prism死了。他无能为力来阻止这件事发生。他哭了。

努力控制住自己后，卡特勒给这个实验室里的所有人一个月假期一带薪的。

第2章 代码之王（1）

不管比尔·盖茨是多么的富有和强大，他母亲还是热衷于讲述她对儿子的教育失败——关

于保持整洁。

盖茨是西雅图城的一个殷实之家的独子。他的母亲，玛丽，是西北银行巨头的后代，也是华盛顿大学的校务委员。而他的父亲，威廉 II，是一位在市中心工作的事业有成的律师。

聪明而任性的盖茨从小就喜欢和他的父母对着干。他出生于1955年10月28日，他和其他婴儿潮出生的一代一样，特别懒散。他少年时代的房间简直就是——用一个词来说——肮脏不堪。他父母对此感到很担心，再三命令他打扫卫生。他却从来不听。

玛丽只好惩罚盖茨，一旦她发现他房间里有乱扔在地上的东西，她就拿走。但盖茨似乎并不在意他的东西少了，即使是当玛丽开始拿走她在地上发现的衣服的时候。

他的衣服很快就要被拿光了，可是盖茨看起来却一点反应都没有。

玛丽后来觉得要采取专业措施了。她找来一个顾问，来调解这个僵局。这个顾问与盖茨，他的两个姐姐和他的父母先后进行面谈。然后，下了个处方：别管盖茨了。

玛丽照此做了。她让她的儿子糟蹋他自己的房间，只是要求他关上门。可就连这样小小的要求，他都不总能做到。

盖茨曾是个瘦瘦的小孩，皮肤苍白，脸颊有雀斑，头发是褐色的，眼睛是淡蓝的，他有一副歪歪的咧嘴笑容和一个对他的脸显得有些过大的鼻子。他的眼镜似乎从来没有扶正过，总是向一边倾斜，使得他看起来有些傻气。

当大多数年轻小伙子把热情留给姑娘或者运动的时候，盖茨却拥抱电脑。他所在的贵族高级中学拥有给学生尽情使用的一流电脑。编程中的干脆的公正吸引着盖茨。坚固的逻辑，而不是主观意见，造成了编码间的矛盾也明确地解决了矛盾。”运行你的程序是一个绝对的考验，”盖茨回忆起他第一次付出努力的时候，曾说道，”你写一个程序，运行它，测试它是否能工作。”

盖茨的笨拙只是表面现象，他真正具有善抓商机的天赋。差不多就在他学写程序的同时，他便开始尝试销售它们。他因此遭受了市场的严峻考验。这在七十年代早期对于一个十几岁的程序员来说是不平常的。他的大多数同龄人把他们自己看成是对抗社会的黑客，他们把计算机技术当成是摧毁建成物的工具。但盖茨并不这样。他设计了一个编有交通数据的小程序，把他的 Traf-0-Data 系统买给了当地的好几个城镇。他希望西雅图城也变成他的一个客户。但是当一位市政官员到他家里看他的演示时，这个程序却运行失败了。心痛的盖茨，想努力保留住官员的兴趣，央求道：“告诉他，妈妈——告诉他这真的有用！”

在1973年的秋季，盖茨，这位数学高才生，成了哈佛大学的一名大一新生。在学校，他经常打扑克，不停地玩电脑游戏，而且看起来总是忙忙碌碌。他会经常坐在他的房间里设想他的未来，”像哲学家似的，一个抑郁的家伙，总在琢磨我应该怎样对待我的人生。”在1974

年12月，他在保罗·艾伦那里得到了一个启发，艾伦是比他大三届的高中校友。那天，这个曾经帮助盖茨设计交通程序的艾伦，带着一本最新发行的《大众电子》来到了哈佛。这本杂志的1975年元月刊封面赫然登着一台名叫牛郎星的电脑。它是发烧友的梦想。不到2000美元的牛郎星电脑完全具有价值比它贵几十到上百倍的其他类型电脑的功能。这秘密是什么呢？微处理器。加利福尼亚州圣克拉拉的英特尔几年之前发明的微处理器把一个电脑的关键元件压制在一个指甲大小的硅片或者说芯片上。这个芯片于是变成了牛郎星的“心脏”。

作为一场革新的先驱者，牛郎星成了第一批大量销售的个人电脑。这是第一种以人为本而不是以任务为本的电脑。计算机界的保守派们完全忽视了这种理念的重要性。大型计算机的痴迷者们总在炫耀着大型机处理大批工作的优势。但是大型计算机就如长龙巴士一样——只要所有人都想同时到达同一个目的地，效率的确不错。而个人电脑就像一辆小汽车；司机想到哪，它就能去哪。人们不需再按照大型计算机的时间计划表来安排工作，只要有一台微机，任何人就可在任何时间做电脑操作。

个人电脑的推广者们看到了它强烈的号召力。这使效率不高的计算机变成一种消费品，就如《大众电子》的编辑介绍牛郎星时所称：“家家户户有电脑的时代……已经到来了！”

这些话语使盖茨和艾伦振奋起来，他们看到了牛郎星的缺陷：它没有软件。买家只能自己写，如果他们写不了，那从本质上说，买的机器就没用。这正是盖茨和艾伦可以解决的问题。

牛郎星首先需要一种编程语言。盖茨和艾伦打电话给制造牛郎星的公司，该公司位于新墨西哥州阿尔伯克基城，向他们承诺将提供一种编程语言。他们选择了在20世纪60年代开发出的 Basic，它主要是为 DEC 公司生产的微型计算机设计的。Basic(初学者通用符号指令代码)比 Fortran 要更适用于短程序，也更容易学，因为它的指令比较简单。这种语言十分流行，它的原作者是两名达特茅斯学院的教授，他们宣称这组程序没有所有权，任何人都可以免费地使用或者修改它。

不到六个星期里，盖茨和艾伦就为牛郎星完成了一套 BASIC 版本。他们俩达成了名叫“微软”的合作伙伴关系，以此兜售这个程序。艾伦乘飞机到新墨西哥州完成了这笔交易。很快，微软的 BASIC 卖得非常好，虽然它要卖到500美元一套。这样的好市场使盖茨离开了哈佛，并没再回去过。（译者注：2007年，比尔·盖茨辍学创业32年后，获得母校哈佛大学荣誉法学学位）

牛郎星从微机舞台上退下了，但微软的 BASIC 语言仍是重要的编程工具。原因是盖茨和艾伦把它与英特尔的通用芯片线做了匹配，而英特尔的芯片是牛郎星和后来许多个人电脑的“心脏”。BASIC 语言的价格占到一台小小电脑的整个成本的很大一部分，这引起了很多电脑爱好者不满。他们开始复制程序，免费地到处发送，声称软件应是一种公共资源。盖茨坚决反对这种把软件看作是电子世界中的“水”和“空气”的说法。他公然指责盗版者为窃贼。“大多

数的编程爱好者应该注意到，你们当中的大部分用的是偷来的软件”，他在一本电脑杂志上写道。他讥讽地说，嘲笑他们这种共有理念，他刻薄地评论道：“硬件是要钱买的，但是软件却是用来共享的。有谁关心写软件的人是否得到报酬了呢？”

第2章 代码之王（2）

谁关心？比尔·盖茨。

个人电脑极大地改变了人们工作和娱乐的方式。从70年代后期开始，不计其数的人们开始卖掉他们的打字机，更换成台式电脑。一切的一切，包括从图案设计到文字处理到档案管理，在电子技术的帮助下，都能更容易更快地完成了。1980年，IBM，世界上最大的计算机生产商决定开始生产个人电脑。IBM 有自己生产一切部件的传统，但在当时的情势下，IBM 得把个人电脑的两块最重要的部分外包出去，因为这要便捷得多。

第一块是微处理器，即芯片，就是让个人电脑成为可能的东西。这芯片的能量决定于压在纤细硅片的开关的数量，和这些开关之间的连线。那时，最好的芯片有几百到几千个开关，但是设计者们已经开始想象拥有几百万开关的芯片了。每个开关就是一个智能的载体，相当于一条大脑神经。在微处理器上的开关总是非开即关，不是1就是0。芯片的特别构造有利于执行计算机的程序指令，程序最终被转译成一串串的0或者1。协同工作的开关越多，程序执行得越快。然而每种微处理器都有自己独特的界面，或者说独特的接收指令的方式。结果，有不同指令集的芯片是不相容的，这意味着为一种芯片设计的软件不能适用到另一种芯片上。

操作系统是 IBM 个人电脑需要从外部买入的另一部分，于是它关注还处于萌芽状态的微机软件市场，寻找供应商。微软的主要竞争对手当时并不看好 IBM，但盖茨却产生了极大的兴趣，在一瞬间就判断出这笔生意的重要性。他甚至没有一个操作系统，但是他知道可以上哪儿去找一个。西雅图也是硬盘操作系统（DOS）作者的故乡，DOS 可以控制英特尔类型个人电脑的基本功能。这种电脑恰好是 IBM 准备经营的个人电脑。虽然那时的 DOS 还做得比较粗糙，但它能成功。只花了5万块美金，盖茨就买到了这组程序，但他并没有告诉 DOS 原作者 IBM 的兴趣。后来，有些人把这次购买叫做“世纪性之交易”，但也许叫“世纪性之偷盗”更确切些。盖茨一直否认在他财富背后隐藏有巨大的罪恶，但如果承认有，也就是他购买 DOS 的行为。

当然，如果盖茨没有保留他可继续将 DOS 卖给 IBM 的竞争者的权利的话，DOS 也不会那么值钱。这样的策略——使盖茨走上发财之路——显得更加清晰了，尤其当 IBM 个人电脑最终成为方兴未艾的个人电脑市场的行业标准的时候。在20世纪80年代中期，除了苹果电脑，每一个微机产商都觉得，唯一一条能和 IBM 竞争的路子即是销售和 IBM 个人电脑一样的克隆机。制造一个克隆机除了需要其他硬件，还要的就是从微软那里购得 DOS 的使用权。于是在很长的一段时间内，DOS 成了微软的某种意义上的铁饭碗：而对个人电脑生产商而言，购买 DOS

便成了市场准入的首付代价。

其实盖茨后来并不仅仅依靠 DOS。他想要占领个人电脑软件的所有方面，从办公室的处理文档到家庭的休闲娱乐。除了 DOS，微软还销售 BASIC，其他编程工具，还有应用程序，比如文字处理软件，电子表格等。盖茨也期待着微软正在创建的一个程序的诞生，名叫 Windows。DOS 的缺点是仅仅为打开一个文件，复制一个档案，或其他任何很普通平常的操作，都要求使用者记住看起来没多大意义的指令组合。为了使人们更容易地操作电脑，Windows 呈现了一个可视菜单界面，可以显示常用的指令、文件、程序以及其他硬盘里储存的东西。用户可以指挥屏幕上的小箭头来选择指令（通过使用连接在键盘上的，手掌大小的一种定位设备，名叫鼠标）。

1988年夏天，比尔·盖茨成了新生代企业界大亨的杰出代表：软件业巨富。正如石油在上世纪产生了一批富豪，“野猫”程序员们使自己成为了美国自力更生致富人群中的一员。因病离开微软的保罗·艾伦当时的身价就过亿。其他主要的软件公司的奠基人都值这个数。盖茨在他33岁的时候，就成了美国最年轻的亿万富翁。但钱并不是他的动力。“对我来说，跟一伙聪明的工程师一起工作，研发出产品，然后你走出去看到人们确实在使用它们，这才是更大的乐趣所在。”他说。“比起用与软件不相关的方式挣钱，我宁愿免费去做软件。”当他被标榜为企业家时，他总感到不高兴，宣称，“如果可能，你朝我大脑里面看看，它其实充满着关于技术的想法。”

但是，他的行动却都是在做生意。他已经五年没有写过一行代码了，不过他一直在检查微软的大部分代码。检查过程就像是法庭的诉讼过程，而盖茨扮演着检察官的角色。他经常用一些简单的问题发问，但随后便刨根揭底地穷问下去，他坚持说他总比他雇用的工程师们更能发现问题。他并不介意工程师们为他们自己辩护，甚至向他大吼大叫，因为他也如此对待他们。在检查代码的时候，他是不顾及任何人的感情的，他发表意见时，会带有攻击性言语，诸如“这是有史以来最愚蠢的代码。”

第2章 代码之王（3）

尽管盖茨不时地对程序员们发难，他还是十分欣赏他们，而且一直让他们自己管理项目，这样工程师们可以一边写程序，一边考虑如何管理和经营。虽然双重责任带来了更多压力，但盖茨总想避免让没有编程经验或者只有过时知识的职业经理人来管理公司。仅依靠管理专家来领导软件团队——或者软件公司，这简直是破坏性的。他们既不能从辨认出哪些是有前途的程序，也不能评估进度或产品的设计。在管理专家经营的公司里，这些职业经理们到后来总是不相信他们的程序工程师，因为他们不懂也不能进行任何控制。

盖茨的检验标准是不固定的：他具有在对的时间改变风向的窍门。这个世界里，人们往往只抱住一个坚定信念不放，但盖茨不会墨守成规。然而，他也会情绪化和感到焦虑，有时候甚至是个妄想狂。他会想象失败的下场，也幻想他被敌人包围的场景。这些想法使盖茨“

害怕地奔跑”，不过他认为这是有利的条件。自满“是毁灭之路”，他说，“你必须一直想谁将取代你。”

盖茨的父母和朋友们都担心他过于严格要求自己了。毕竟，微软已经是全世界最成功的年轻企业之一。但是，这并没有给盖茨多大的安慰，他好象被他自己的名声弄得心烦意乱。当事情看起来发展不顺的时候，他的钱和名望对他来说就啥也不是。就如他父亲说的，他的态度是“所有的东西都在急转直下。”

1988年的夏天就是这样一个“急转直下”的时间。盖茨的专注工作使他卷入了一次桃色事件。在这几个月之前苹果电脑起诉微软，声称 Windows 非法抄袭苹果机的系统程序。这起案件开始了软件知识产权法律概念不清晰定义的历史，判决结果也很不明确。一直以来，这个案子让盖茨感到愤愤不平，因为它给盖茨的成功来自仿造的说法添油加醋。

接下来是 IBM，他的重要同盟。自从微软的 DOS 变成个人电脑的标准操作系统之后，IBM 就想在软件上做得别具一格，显示出与其他克隆个人电脑的不同。在1987年4月，这两家公司就公开地允诺要研发出取代 DOS 的操作系统，取名叫 OS/2。不但这名字是销售史上的一大败笔，IBM 和微软也没有及时给 DOS 迷们提供可以在新的操作系统上运行老程序的方法。刚开始时，盖茨大肆宣扬 OS/2，他发誓说 OS/2 取代 DOS 只需不到3年时间。在1988年夏天，仅仅18个月之后，OS/2 对他来说变成了一场灾难。第一版本设计得相当糟糕，不仅不能运行 Windows 程序，而且也没有一个方便的图形化用户界面，可以一点屏幕目录上的图标，便可选择运行程序的界面。尽管发展不顺，盖茨并不想与 IBM 散伙，因为 IBM 是他最大的客户，也是 OS/2 的积极推动者。他不敢从 IBM 的轨道上偏离出来，不过他也不想成为电脑生产商的囚鸟。就在他保证他会继续支持 OS/2 的时候，他开始筹划另一个项目。这时，他就像一个家族企业中一个野心勃勃的儿子，幻想着一旦他的父亲突然过世，他将如何经营下去。

对于管理，盖茨有一个简单的方法：让你周围都是极富智慧的人。“聪明人”，他说，“是在掌握充分事实后，能推算出所有可能性那种人。”定义聪明人的概念是比较难，可每当盖茨碰到一个聪明人的时候，他总能识别得出。这些聪明人清一色是男性，所以他们的聚会总洋溢着男生俱乐部的味道。那年夏天有位新来的成员，他27岁，是理论物理学界的落荒者。

内森·麦沃尔德曾是著名物理学家斯蒂芬·霍金的得意门生，不过后来他和普林斯顿的一些博士同僚们在加利福尼亚州的伯克利市成立了一家软件公司。是什么吸引了麦沃尔德或其他物理学家转行到编程序呢？科学研究越来越依赖计算机，天才们在信封背面涂写公式的时代几乎消逝。物理学家们常常把编程看作是达到目的的一种手段。麦沃尔德发现他对软件的着迷替代了他对物理科学的喜欢。当他和他的朋友写出 IBM 的 TopView 图形程序的克隆版本时，他的公司一举成名，因为克隆版本更小，更快。IBM 正考虑把 TopView 做为个人电脑的软件界面——就是可以被消费者看到的部分。为了能与 IBM 并驾齐驱，盖茨想要一个 TopView 的克隆版本，于是他买下了麦沃尔德的公司。

后来 IBM 很快把 TopView 抛弃了，不过，这次并购的最大益处是把聪明人带到了微软。麦沃尔德和盖茨十分合拍，不久后，这位物理学完成了一系列关于软件业发展趋势的生动的备忘录，大大刺激了盖茨的好奇心。在其中的一份备忘录中，麦沃尔德主张，微软的现金奶牛，DOS，面临着两个近在咫尺的威胁。

第一个威胁是新型芯片，很富潜力，速度比英特尔芯片快多了，但英特尔芯片又是 DOS 和它所有应用程序的“用武之片”。超速新型芯片是由名为 RISC（精简指令集计算机）技术支持的，这种技术使芯片可以集中于执行普通操作。万一 RISC 芯片代替了英特尔在个人电脑里心脏的地位，微软帝国很可能崩塌。

第二个威胁是一个名为 Unix 的操作系统。自从10年前，AT&T 的贝尔实验室发明了 Unix 后，它便逐渐受到科学家，工程师的欢迎，甚至受到了一些企业的宠爱。Unix 主要有两大魅力。一是技术：这个系统可以同时处理很多个计算任务，很容易与其他计算机连接，而且不需要修改大量代码就可以工作在其他任何类型的硬件上。二是实惠：实质上，它是免费的。AT&T 拥有 Unix 的核心技术，但它允许其他人去改编和扩充 Unix，而只象征性地收取费用。这鼓励着学者和工程师们去完善 Unix。许多公司开始销售装着各种 Unix 操作系统的电脑。

然而，Unix 有个致命缺点：压根就没有通用版存在。多年来，不同版本的 Unix 简直如遍地开花，所以为一个版本写的应用程序，只有修改后才能运用到另一个上。DOS 为消费者，应用程序编写者和计算机制造商提供了一个单一的目标，而 Unix 却没有。Unix 的这种支离破碎对微软来说是个福音。但是，Unix 能装在有 RISC 芯片的电脑的前景还是令盖茨坚信，他需要一个“Unix 杀手”。他让麦沃尔德试着去写一种“可移植的”的操作系统，意思是该系统可以在正在开发中的任何一种 RISC 芯片上运行。

第2章 代码之王（4）

麦沃尔德有着卷曲的胡须和头发，这使他看起来像是一位被遗忘的俄国革命家。他在学术界专家中做了些调查，询问他们的意见，然后成立了一个小组，他命名为精神病组，因为他的不少同事都认为试图写这样一种“可移植的”程序本身就是一个“疯魔想法”。

离盖茨微软办公室只有几英里的地方，大卫·卡特勒在 DEC 实验室里因为棱镜项目的取消而心乱如麻。DEC 担心他使已经一落千丈的实验室的士气变得更糟，于是让卡特勒在几天后搬到另一栋办公楼。即使他们会继续给卡特勒发薪，但他觉得他被隔离了，他打算离开 DEC。他想立即辞职，但他在马萨诸塞州的老板却坚持让他守住，许诺说会给他在公司里安排另一个职位。卡特勒心里琢磨着：DEC 只是试图“让我再开心一会儿，使我不至于在公司里到处捣乱。”

卡特勒被隔离的消息传出来了。1988年8月4日，麦沃尔德团队的一个员工冲到麦沃尔德在微软的办公室，这个员工曾经在 DEC 和卡特勒一起工作过，他告诉麦沃尔德一些小道消息：卡特勒最近的项目被取消了，他想离开 DEC。

麦沃尔德把这新闻又告诉给盖茨。“这简直是惊人的巧合，”盖茨说。他从没见过卡特勒，但是很了解卡特勒的事业，也在考虑要雇佣他。“我太想要一个可移植的（操作系统）了，”盖茨说，“问题不在于是否我们应该组成团队，而在于何时能组成团队去开发它。”

盖茨认为卡特勒将是这个团队的最佳领导人选，但他抑制住自己的热情。据说卡特勒桀骜不驯，且不容易合群。盖茨估计卡特勒的这些怪脾气会使他拒绝加入微软。但盖茨就是想吸纳他进来，于是他请第三方帮忙安排了一次见面。

卡特勒和盖茨间有着巨大反差。卡特勒，结实健壮，意识力坚强，蓝领家庭出生。他说话粗俗，还带着胡椒味；他举重，喝啤酒，不时以打猎为乐。而相反，盖茨的背景则相当优越：出生于富裕家庭，贵族高中，哈佛，并快速地成了一名优秀的企业家。生活没有给盖茨留下伤疤。他的衣着就像大学二年级学生，假日里他阅读书籍，或者举行被称为“微软竞赛”的夏日游戏：把朋友们分成不同小组来比赛，看谁能够赢，项目五花八门，比如看看谁做的沙滩城堡最漂亮。

他们见面的那天，盖茨很热情地问候卡特勒，而且开始试探他的能力。“我听说过他，但是有意思的是，我过去所知道的都是间接渠道来的。”盖茨说，“我也不知道有多少（关于他的传说）是谣言。”

盖茨很快就看出卡特勒是个颇具实力的人才。他颇感欣慰，和卡特勒聊起了自己的可移植操作系统的愿望以及该愿望是如何符合微软的前途。卡特勒并没有被盖茨的财富和权力震慑住，他并不掩饰自己对微软和其产品的轻蔑。他认为微软的编码写得很糟糕，而DOS，微软的皇冠之珠，简直就是一个玩具。他想给强大的计算机们创造一个“真正的”操作程序——一个值得信赖的，能管理网络和处理大量信息的程序——总而言之，是DOS现在或将来无法超越的东西。

盖茨听卡特勒能这样说，非常高兴。这也是他自己对微软未来的一个大致设想。虽然在总的观点上，他们达成一致，但他们仍然有许多分歧需要解决：除非卡特勒可以从DEC公司带一打或更多的工程师来，包括一批计算机硬件工程师，他才愿意加入微软。盖茨很欢迎软件人员，可是微软不制造计算机，所以 he 不想接收硬件工程师。卡特勒拒绝把他们排除在外。“我喜欢做硬件，”他坚持道，尖锐地强调他宁愿自己组建公司，也不愿离开他的硬件团队。

会面结束后，麦沃尔德也赞成雇佣卡特勒的计算机工程师，他说他们可以在微软制造一些硬件，这样公司做软件实验或优化一个新的操作系统也会容易一些，硬件工程师还可以建议客户如何设计个人电脑才能充分发挥软件功能。其实盖茨并不需要太多别人的说服，因为他太渴望得到卡特勒了，他把他看作是“为这使命而生的人”，况且“聘用这些硬件家伙们的成本也能承受。”

的确，钱不是问题。微软的利润相当高，每投入一美圆就能有25美分令人振奋的回报。另外，它给从DEC来的员工的薪水甚至还没他们原来的高（比如说卡特勒，薪水只有他原来

的一半)。但公司的薪酬也有一定的吸引力，微软给予了每个新成员许多优惠购买公司股票的权利，相对于当时的市场价，有15%的价格优惠。这些举措很可能使 DEC 的背叛者们变成百万富翁，如果他们在微软待的时间足够长，甚至四年或五年，长到他们完成开发可移植软件程序的任务的话。微软允诺卡特勒本人可认购许多股票，这使他觉得：“盖茨真的想要（我）去做这个。”

对卡特勒来说，这整个的报酬——股票认购权，自主选择他的核心团队和开发另一个重大的操作程序的机会——证明了他作为一名软件工程师和一位编程团队领袖的崇高的荣誉和丰富的经验。他四十六岁，已是软件行业的一匹老马了，却使盖茨更对他感到格外信任。全世界只有为数不多的几家公司能充当卡特勒的老板。在 DEC，戈登·贝尔助长了他的雄心；现在盖茨在微软也为他做同样的事情。卡特勒并不指望他和盖茨能完全磨合，但对盖茨给了他一次重新开始的机会，他还是心存感激。他感到他身负另一个重要的工程，要担心的仅仅是盖茨也许会变化无常，取消掉他的项目或者恶意地用无休无止的要求来摧毁他。

其实盖茨并不能完全确定卡特勒真是微软下一代软件产品的制造者，他还不能预见那么远的未来。他只是“超乎想象地兴奋”，微软的二管家斯蒂芬·鲍尔默说。“我们都手舞足蹈：卡特勒加入了微软！这简直难以置信——天堂里的绝配啊！”

卡特勒于1988年10月31日登陆微软。几天后，罗伯·肖特，卡特勒的首席计算机硬件工程师辞去了 DEC 的工作，成为第一位加入卡特勒新团队的成员。肖特是爱尔兰科克人，也是卡特勒的最亲密的朋友。他在硬件领域的成功并成为佼佼者的道路是个非典型的例子。肖特结束了他电子学方面的学业后，开始在爱尔兰高威市的 DEC 工厂工作。20世纪70年代早期，他搬到美国，白天在 DEC 上班，晚上则进修工程学和计算机学。一个晚上，他做完他的课后作业后，他把他打印出来，作业内容是关于汇编语言的一个程序。当他去拿打印稿的时候，他没找到，于是他又重新打印。当他再去打印机旁边时，他发现卡特勒在修正他的作业。想到这，肖特突然笑起来，“他就是在玩我的家庭作业，”肖特说，“这太搞笑了。”

1982年起，肖特开始为卡特勒工作，他搬到西雅图实验室。两年之后，卡特勒便让肖特管理计算机硬件设计了。

当要解决技术难题的时候，肖特有着坚韧不拔的精神，但是他会很老练地和卡特勒相处，他能轻易地看出卡特勒的情绪，也常常用一些俏皮话来缓解他快要爆发的脾气。虽然肖特对卡特勒十分忠诚，但他也意识到他将在微软开始的重要的旅程也许会令他和其他人筋疲力尽。“大卫的领导风格是他自己做大量的工作，也促使其他人做大量的工作，”他说。“有意思的是，真的是很复杂的技术活也不能让他慢下一点点。不管有多复杂，他总是用同样的速度进行。”

在一名安保人员的陪同下，肖特离开了 DEC 的大本营。他驾车来到微软的企业园区，这企业园区能盖24栋大楼（那时候公司只有不到10栋），有好几个操场和慢跑道，一个图书馆，

足够的停车位，还有很多自助餐厅。办公楼建得不高，颜色较深，风格朴素。但是每个人都有自己的办公室，按照自己的意愿去装修。每幢楼的侧楼里，都有装满冷果汁和软饮料的一整墙面那么大的冰箱，员工们可以免费地随便喝。比如说，麦沃尔德每天至少要喝六大盒轻怡百事可乐。他的朋友开玩笑说，“他是纽特牌甜味剂永远不会有致命量的活证明。”

肖特在微软园区弯曲的道路上开着，他来到二号楼，走进去，打量着陌生的环境。他突然怀念起 DEC 来，“我们在 DEC 有漂亮宽敞的办公场所，有我们自己的目标，我们也知道我们在干什么。”他发现卡特勒独自坐在楼上的一间办公室。卡特勒还没有给来客用的电脑或椅子，他看起来也不自在。于是肖特坐在地上。两个男人互相瞧着对方，大笑起来。然后肖特说：“也许，这将是灾难。”

第3章 部落（1）

卡特勒加入微软之举导致了 DEC 大批员工离职。他并非要掠夺他以前的老板，而仅仅是因为他的同事把他看作是一位高科技摩西（译者注：摩西是《圣经》故事中的犹太人古代领袖），他也从同事对他的忠诚中受益。不到一个星期，微软就从卡特勒原来的实验室中招聘了7位顶尖级编程工程师。

核心队伍中的每个人都在卡特勒手下工作了好几年，全都是男性，只有一位在三十岁以下，这样，这支队伍要比微软的平均年龄要大得多。第一天，微软的人事专员召集所有的新到者例行公事地宣讲公司政策和惯例。有一部影片讲述了微软的崛起，里面有一段是对盖茨和他母亲玛丽的动情的采访。卡特勒的一位下级后来打趣说：“这段采访不错，只要你不吐出来。”

这个上午的最后，新职员们被要求签署劳动合同，劳动合同的内容包括要求员工们对敏感问题保密，公司拥有员工的知识产权等。其中第10条，禁止雇员离开微软后一年内加入同行业竞争公司。这一条的给被雇用者的单方法律义务过重，卡特勒的部下们，虽然以前就被告知过，但还是提出抗议。他们拒绝签署该合同。他们说，如果 DEC 也给他们这样的限制，微软将不得不在法律程序上花一番功夫才能雇佣他们。所以为什么要签呢？公司也不退步：如果不签合同，他们就不可以工作。都快到中午了，卡特勒的部落们撤出了房间。虽然他们是因为肚子饿了去吃午饭，但这看起来像是个罢工。后来有人把这件棘手事告诉了卡特勒：也许他的同僚们最终不能来微软了。

卡特勒迅速解决了纠纷。微软的律师终于同意，删除第10条。卡特勒的同事们签下了修改后的合同。

这个已和解的僵局，尽管短暂，却给微软的其他员工传递了一个信息：卡特勒那伙人与众不同。他们自视更有才华，更成熟，也，的确，比微软的原先的员工要优越些。表面上，这个结论看起来合乎情理。首先，他们要比普通的微软员工（园区里被称作“微软人”）老道，有经验；微软人几乎都20多岁，大多都没有在其他地方工作过。DEC 的这些背叛者们在工作上

有条不紊，遵循技术教科书上的习惯做法，而相反，微软人经常手忙脚乱地解决问题。卡特勒的同志们非常认真地对待工作，而微软人有时在走廊里投掷碰碰球或在办公室里摆弄吉他。

他们之间风格上的差别很明显，卡特勒部落把微软人嘲笑为“微废人”。同时，微软人被卡特勒部落排斥在外。“卡特勒和他的部落们一进来，就是一个非常紧实的团伙。”一位程序员说，“他们直接把这种态度表现出来，简直让其他人心里发毛。”

卡特勒对此一笑了之，他不喜欢考虑精神生活的阴暗面。但他也感觉到他对这里的适应要比他预期的困难一些。在外界和竞争者的眼里，微软是座丰碑，可其实，微软更像一块块封建领地勉强拼凑起来的国家，而不是一个联邦国。虽然是盖茨在统治，但是他过度地依赖一些程序领主和经理人，这些人因为自己和微软的关系变得富裕起来。在 DEC，卡特勒是一颗明星，他希望在微软也能享受到明星级待遇。但，这里没几个人知晓他；他感到自己默默无闻。“我在这边没有威信。”他说。

卡特勒本能地变得内向起来，他在他的部落里寻求安慰。他的部落包括六位硬件工程师和几位程序员。他们占据了二号大楼的一个走廊，他们办公室里的桌椅随时都可以凑到一块去。刚来的头几天，这个团队莫名其妙地平静。“零方向，”一个成员说，“我们不知道给我们的是什么，也不知道怎么去发展。”他们几乎无事可做，就学习个人电脑的里里外外。尽管他们很有经验，可是都从没给一台个人电脑写过程序。他们感到很迷惑。诸如“它就是……整个的文化冲击”和“没有一见钟情”之类的说法可以概括出他们的反应。

卡特勒的小组里弥漫着沉闷的情绪。“真令人沮丧，”一个小组成员说，“开始的几个星期我们都那样觉得，包括大卫。”在他们心里，棱镜项目被取消的这块阴霾还未散去，所以很担心微软是否最终象 DEC 一样对待他们。“在我们头脑里，完成一个项目是最最重要的事情，”另一位成员说，“即使我们不喜欢这里，我们也不会让项目流产，我们宁愿先完成项目，然后在第二天辞职。”

盖茨还沉浸在获得卡特勒加盟的巨大喜悦中。他以前总是因为公司的产品功能有限而备受指责，现在他终于有了可以把大型计算机的优点带到个人电脑来的软件人才。微软宣称它的软件总有一天能进行任何一种工作，卡特勒的到来为这誓言增添不少可信性。盖茨不确定微软什么时候将用上卡特勒的操作系统，所以他在公众面前绝口不提他的软件明星。在1988年11月10日，微软承认了聘请卡特勒，但宣称只是为了请他来指导开发 OS/2 的升级版本。在一次发言中，卡特勒被形容为“是一个能管理大项目的人，而且能准时把最高质量的产品呈现在人们面前。”卡特勒并不发表意见。在 DEC，他从不接受采访，在微软，他也坚持要求他本人不见媒体。他甚至警告盖茨：“如果你带媒体来见我，我就会做出使你永远不会再把他们带来的举动。”

为了庆祝卡特勒和他的部落的到来，盖茨在微软园区的一栋还未竣工的建筑里举行了一

个啤酒比萨饼聚会。在大伙面前，盖茨介绍了新成员，激动地谈论来自他们的“伟大的东西”。盖茨极少公开地偏爱某一个程序员，但现在卡特勒显然成了他的宠臣。盖茨看起来甚至敬畏卡特勒，也不敢去打扰他。“他是一个对工作很认真的人……极其专业，”盖茨说，“这使他变得难打交道。如果你要去见大卫，你得这样想，天哪，我要说的话是不是傻乎乎的？”

盖茨担心卡特勒总被包围在熟悉的面孔之中，可能会和微软的其他员工搞得太疏远，于是他请史蒂夫·鲍尔默，卡特勒的名义上司，去教他公司的工作方式。

鲍尔默被选作为史蒂夫的指导员是个聪明的决定。如果盖茨是太极中的阳，那鲍尔默就是阴。鲍尔默是底特律人，他的父亲是汽车行业的一位主管，他在哈佛认识了盖茨，这两个人成了亲密的朋友。在商学院和宝洁待了一段时间后，从没写过一组程序的鲍尔默在1980年加入微软，成为公司的第一任总经理。他富有热情，为人诙谐，他常常不是在说话，而是在喊话。他与为人冷淡的盖茨不一样，他总是在走廊里走来走去，拿着一本黄颜色的标准信笺簿，他在上面记着“行动事项”。他喜欢与人击掌来问候人。在和员工面谈的时候，他不是抱着个篮球坐在地上，就是对着墙扔着网球，紧张地来回跑动。

第3章 部落（2）

卡特勒觉得盖茨是被外界吹捧上去的，他更喜欢汇报工作给鲍尔默，他觉得鲍尔默是微软成功的关键人物。“史蒂夫是我这种类型的人，”卡特勒说，“是一个精力旺盛真正追求卓越的人。”

鲍尔默对卡特勒的立场是让他自由发挥，至少在一定领域内。“大卫是来开发（操作系统的）内部架构的（操作系统），而我们可以界定（消费者可以看到的）外部的的设计，”鲍尔默说，“这样（分工）很有成效，因为他可以对内部进行创新，外部设计呢，就得把市场需要和技术结合起来。”

这样的安排也还是导致了分歧。鲍尔默总是提醒卡特勒在 OS/2 上运行的程序也要在他在开发中的可移植操作系统中运行。而另一方面，只想专心于加速完成他自己程序的卡特勒，感觉鲍尔“总是要的比我们能拿出来的多。”比如说，卡特勒拒绝允诺开发出能运行现有的 DOS 和 Windows 程序的操作系统。“与 DOS 兼容？与 Windows 兼容？没有人会想要那个。”卡特勒说，“为什么我们应该考虑这些？”

争执偶尔会升温。卡特勒曾经为了跟鲍尔默强调他的观点，双拳重捶他的桌子，并且嚷嚷说他不会“用狗屎去污染（NT）。”鲍尔默悄悄地走到门口，然后轻轻地敲几下门，说：“别捶了，”他朝里大声说，“不起建设性的作用，我也可以捶，而且跟你一样震天动地。”

内森·麦沃尔德也想让卡特勒和他的部落融入到微软中来。他希望这个项目从一开始就非常强大，也希望卡特勒的可移植编码中能够用到他的“精神病团队”的一些实验结果到去。然而，卡特勒只答应去看看“精神病团队”的工作成果，但令他失望的是，他们几乎没多少文

档记录或者只有一些没多大意思的设计。与其去改造“精神病”，卡特勒宁愿全将它们抛弃。

但关于选择 RISC 芯片作为卡特勒软件团队开发的主攻对象，麦沃尔德一锤定音。他选择了英特尔公司一款尚未完成的芯片，英特尔公司也在为标准个人电脑生产芯片。同时，英特尔也在推广它自己的 RISC 芯片，代号为 i860，是用来在这风起云涌的计算机行业中做挡风草的。这两家公司有着共同的愿望：要在浪尖上生存！

卡特勒曾经在 DEC 测试过 i860，他认为它有“严重问题”，但是他同意麦沃尔德的决定。毕竟，麦沃尔德是他的为数不多的同盟者之一。微软的许多人认为公司应该只为现存的个人电脑硬件写软件，不应该在可移植软件上浪费时间和钱。因为计算机潮流的走向是无人可以保证的。“虽然比尔（盖茨）和史蒂夫（鲍尔默），还有我很支持你的工程，但是很多其他人并不这样想。”麦沃尔德解释说，“他们不能理解，或者认为这样的想法太好高骛远或者太疯狂了。别让这些（批评）影响你和你的工作。”

让卡特勒的部落和微软人打成一片，盖茨还有一个绝招：他把微软的中坚骨干程序员史蒂文·伍德分配到卡特勒的“清一色 DEC”部落中去。这步棋下得很关键，因为“我必须知道他们是否能接受一个外来人”。盖茨的理由是，如果卡特勒连微软的一位最优秀最能干的程序员都不能接受的话，那很可能整个新到的团队都不能适应微软。

伍德果然不负众望。他是一位能力惊人的程序写手，他1983年加入微软。之前，他在耶鲁大学攻读计算机博士学位，但最终没有完成博士论文。他具有软件业优秀的两大传统精神：自视清高，超然出世，厌恶商业伎俩；却又是实用主义，坚持“想就去做”的观点，同时也是很聪明的自学型人才。他接受的正规教育使他坚守的信念很难被动摇，当然，不容否认的是，他足智多谋。

伍德不是一般人。他走路迟钝缓慢，表情又有些鬼鬼祟祟，看起来他似乎不太喜欢日光。他选了一个没有窗户的办公室，尽管在微软，带窗户的办公室是显示公司地位的唯一外在标志（因为微软有任意停放的车位，有自助餐厅，带司机的豪华轿车，还有公司喷气式飞机）。在这样一个打破常规的公司里，伍德跟反对派对着干，如果有人对他不尊敬，他也对那些人不客气。

伍德是牙医的儿子，他于1951年出生在西雅图的美丽的郊外。和其他男孩一样，他喜欢玩卡丁车、模型飞机和模型轮船。当对一个旧的轮船模型厌倦了的时候，他和他的弟弟就在模型上点一把火，然后把它漂在附近的池塘里。旧飞机模型被打发得更为壮观，他们使它们沿着长长的金属线下滑，如果成功的话，可以把旧飞机模型们送到一个仿造的航空母舰上。如果多些鬼点子，调皮的男孩们会在把飞机模型送上金属线前点燃它们。

伍德十四岁的时候，他已经有六英尺高了，这也是他成年后的身高，而体重只有100磅。那时的他象一根笨拙的竿子，还有点害羞。他爸爸意识到公立中学不能满足伍德，就把儿子送到湖滨高中，也就是几年后盖茨也来就读的私立精英高级中学。

伍德讨厌湖滨高中，来这里读书的都是西雅图富家子弟，他感觉自己就像一只流浪猫。有些同学有车，甚至还有同学开着飞机，伍德在坚持独立意识的教友派信徒家庭长大，他周围同学们唾手可得的钱让他感到不自在。他在学校里煎熬着，考试成绩拿 B 等。他唯一一次使他“出众”的经历是，他后来难过地回忆道，“我拿着学校的钥匙被捉了，因为有同学看见我们去打开一间仓库。”伍德和他的朋友被校长羞辱，还被在许多同学面前被拽到另一边去，这件事现在想起来都令人难为情。

1969年伍德就上大学了，为了逃避越战征兵。伍德开始漂泊。换了几所学校，并省吃俭用地在邮政局里打工，后来，他读了物理系。“突然之间，学校变得有意思起来。”他说。但是他不能想象他将终身从事物理专业。“我可以预见，如果你是天才，在从事这个领域的20年间，可以只做重要的研究，”他说，“如果你不是天才，你只能做天才的仆人。”

伍德转读了一些计算机课程，毕业后，他在西雅图的一家地区电话公司从事编程工作。四年后，他考取了耶鲁大学，攻读计算机专业。一年后，伍德开始质疑他从常青藤联盟教育中的获得的收益。当他的妈妈庆祝他获得硕士学位时，他告诉母亲：“妈妈，它简直就是厕所里的手纸。我得到它，只是因为不想因考试不及格而退学。”

伍德在耶鲁大学度过了六年，他努力扎实地学习，当时，研究领域的人们推崇自我鼓吹和自我膨胀，这已成大学里的风气，但伍德未受传染。伍德觉得他的教授们与现实脱节了，“他们在他们有限的领域里是非常聪明的人，但是他们没有能力来判断我做的是好还是坏。”

第3章 部落（3）

伍德在耶鲁大学得到的唯一收获是在那里遇到了他的妻子。这发生在1982年的11月，伍德已经31岁了，不过之前他从未跟女性约会过，在他们系里工作的一位行政助理吸引了他的注意。她是一个7岁男孩的母亲，当伍德打电话给她提出约会的请求时，她还不认识他。他们着魔般坠入爱河，很快就谈及结婚的事了。伍德开始四处打听工作，因为他有了离开耶鲁的理由。当时，他的一个耶鲁同学在微软做程序员，他帮忙安排了一个面试。伍德飞到西雅图去见盖茨。盖茨在面试时，不停地在椅子上摇晃，拿着一只钢笔在自己的手上涂涂画画，墨水都飞到他自己的衬衫袖子上。而被微软前景所吸引的伍德，装作没注意到。

伍德在微软的第一个项目就是 Windows 的原始版，这个项目就让伍德引人注目起来，因为盖茨认为 Windows 是微软能从 IBM 独立出来的主要出路。接着伍德接手了另一个备受瞩目的艰难的任务——OS/2，IBM 把它当作是 DOS 的接班产品而大力支持。1988年11月，他结束了 OS/2 的开发工作，他认为做个项目的一年是“有趣而荒谬”的。一年中，他还在 IBM 的英国和佛罗里达州波卡拉顿市的实验室里度过了整整12个星期。这一年经历使伍德对 OS/2 彻底失望，OS/2 能同时运行多种程序（DOS 则不可能），但只要任何应用程序出现问题，整个 OS/2 就不能运转了。伍德像丢“一包垃圾”一样放弃了 OS/2，还认为这是“好设计不可能出自委员会（译者注：“委员会设计”：指很多人同时进行设计，却没有统一的想法）”的最好例证。

疏离了他的 OS/2 同事，伍德需要另找地方着陆。因为他不能容忍与“无能的人”共事，自然而然他的早期前途看起来比较暗淡。他发现计算机这块新领域的时候，他既没觉得它像个家，也没觉得它意味着成功。而现在，计算机对他来说，意味着家，也意味着成功的所在。他在微软的经历既可以说是不同寻常的，又可说是十分典型的。像这样一个软件明星，对卡特勒的团队来说，肯定是锦上添花。他是一流的程序员，工作效率极高，精力充沛，他可以让自己发挥得很好，也能代表微软的形象。盖茨心里乐哉乐哉。

伍德为加入 NT 小组而激动万分，他在那儿发现了“一些聪明人”。但他也感觉怪怪的，因为卡特勒的部落想依靠他来帮助他们来适应他们的新家，可是“连我自己也不是很擅长微软方式”。

卡特勒的团队就操作系统的想法跟伍德聊了好几个小时，他们希望这是为个人电脑创造出的最先进的操作系统。“我就在一个新的沙箱里。”伍德说，“我们没有产品。我们没有日程进度。我们只有乐趣！”

他们也不赶时间。鲍尔默希望这个部落花六个月到九个月时间来理出技术思路，了解微机特性，最后能列出一张“可交付成果”清单。这样的节奏适合卡特勒，他想“一举成功”。许多程序员的工作方式是步进式的，通过多个版本来逐步完善一段代码。每个版本都为他们的想象和推理起到扮演着堡垒的角色。卡特勒则相反，他在开写之前，先在头脑里形成代码图，然后再高精确度地写下代码。“我不是那种尽可能快地写出代码，然后再一遍又一遍地修改的那种人。”他还说：“另一方面，我不怕重写任何东西。如果它的结果并不是我想要的，我不怕全部推翻，重新来过。”

对于编程，卡特勒是个实用主义者。他相信形式优先于功能，但他也不是完全屈从于结构。他总是有正确的开始，而结束时，他的代码也只有极少的错误。通常，他先为团队写出最抽象凝练的代码。他把一个操作系统预先分为几十组代码，每组代码都附着详细的要求。编写代码需求本身就是一项艺术。工程师们早就从他们的职业经验里得知，一个糟糕的代码需求会导致整个项目的死亡，而一个好的代码需求则是一座灯塔，当程序员迷失方向时，灯塔会帮助到回到正确的航线上。然而代码需求不仅仅对它的创作者有用，它还能帮助所有参与开发程序的人。每段代码共生共存，整个程序就好比一个精致的生态系统，如果一段代码有问题，甚至是和其他代码没有直接联系的时候，也可能会摧毁整个程序。

依赖代码需求有两个缺陷。写出一份代码需求就要好几个月。一旦编程开始，代码需求也需要不停地修改更新，否则它就会因过时而不起作用。过时的代码需求可能会起到误导的作用，会导致混乱。只要代码需求在有效期内，就需要专人来维护。这个工作由卡特勒在 DEC 实验室的首席程序员卢·皮亚佐利担当。

皮亚佐利是个高高瘦瘦的男人，很少挺立站直或提高他的音量。“大卫是位实干家，”他说，“我喜欢给实干家工作。”皮亚佐利除了是位天才程序员，还是一位程序员教练。他

的管理哲学很简单：让你的人们快乐。他认为他的程序员们迟早会做对的事情。他一直是位重实际的人。如果有人因疲劳过度或生气而不能继续工作，皮亚佐利会询问什么能缓解他的烦恼。有时候，程序员会给一个直白的答案，会要求要一个大点儿的显示器或是一个装数据的大硬盘。皮亚佐利有一笔基金，通常可以满足这些要求。也有时候，程序员的心情太过抑郁，而疗方也比较贵重。曾经有个程序员想要一个月的休假去游玩意大利，并在哥斯达黎加的海岸潜水。皮亚佐利通常会这么解决这样的要求：你做这个，做这个，再做这个，做完后，就给你安排旅行。善解人意的皮亚佐利是最能衬托卡特勒这朵红花的绿叶。“如果说大卫是盐，卢就是糖。”伍德这样评价。当人们害怕和卡特勒讲话的时候，他们会把他们的苦恼和烦闷诉说给皮亚佐利听。当卡特勒非常生气，忍不住要惩罚一个人的时候，他先向皮亚佐利发火，皮亚佐利再依情况，或是平静地把坏消息传达给当事人，或是不告诉任何人。许多部落成员对皮亚佐利都有极大的感激之情。“卢维护着一批自大、脾气糟糕的人，如果没有他，这帮人说不定会自相残杀。”其中一个成员说。“当有些人因为受不了卡特勒而提出辞职的时候，他能把人留住。”然而，伍德却抱怨说卡特勒和皮亚佐利用的是红脸白脸的伎俩。“麻烦就是，大卫太野蛮了，卢又太好了。”他曾经写道。

其实，也没什么太多需要做的。”卢真的比我随和得多，”卡特勒承认，“他会这么说”这里出现了一个问题，但我不准备做任何事情去解决它，因为它自己会修好。最终，每个人都能发现错误的地方。所以我才不会花这份心力。’”

第3章 部落（4）

皮亚佐利的禅宗般的生活态度也反映在他自己成为软件精英的曲折道路上。他给卡特勒做了十年的助手。在20世纪70年代早期，他先在NASA（美国国家航空和航天管理局）做一名程序员。航天局的工作十分依赖于计算机，从控制任务到分析天文资料。航天局给皮亚佐利的工作任务是给PDP-11写程序，PDP-11是DEC生产的计算机，它的操作系统正是卡特勒早期写的。几年后，皮亚佐利厌倦了他的一万美元联邦月薪的生活，加入DEC，成为一名巡回程序员，他的工作是帮助消费者把软件应用到他们购买的硬件上。在DEC的一次由卡特勒教授的培训课程上，皮亚佐利与卡特勒碰面了，他们一见如故，就卡特勒喜欢的程序足足谈了一个晚上。

几年来，他们一直保持联系。1981年，皮亚佐利搬到佛吉尼亚州林奇堡市，去负责DEC在那一带的消费者。他和女朋友买了一块80英亩的农田，过起了“回到地面上”的生活。他的女友是来自阿拉巴马州的一位高中老师，来到这儿后，在附近的一个镇里的学校教书。皮亚佐利很少在下午五点后还继续工作，他和女朋友在农田里度过了不少时间。他们过着田园诗般的生活，不过从一开始他们就知道这种日子不会持续太长。他们养火鸡和家鸡，烤面包；夏日里收集木柴，冬天就生起篝火。他们允许当地的农民在他们的地里放牧，作为交换，他们会请求农民为他们喂鸡。他们还有一个存烟草的仓库，有时候他们也把它出租给邻居。

1985年8月，卡特勒在他的位于西雅图以外的实验室里为皮亚佐利提供了一份编程的工作。皮亚佐利接受了，很快就以他的勤奋和镇静的态度给卡特勒留下深刻的印象。后来，卡特勒就让他去负责 Mica 软件，Mica 软件是棱镜工程的核心软件。而棱镜的取消打击了皮亚佐利。他从未出产过一个商业程序（他所有的编码都是在内部被公司和客户使用）。“出货”也许是衡量程序员的生产成果的唯一真实的标尺。“如果你没完成，”卡特勒说，“所有绝妙的好点子都无任何意义。”

DEC 剥夺了皮亚佐利经历“完成”的机会。他发誓不能再让任何公司那样对待他。在他最后的给 DEC 朋友的电子邮件中，他表达了加入微软的欢欣，写道：“希望在我头发变白之前，我们能生产出一个商品。”

皮亚佐利认为微软给了他第二次在软件业能一鸣惊人的机会，他注定要充分利用它。他在 DEC 喜忧参半的经历在这个项目上也帮了他不少忙，因为 Mica 软件本质上可说是 NT 的预演软件。对皮亚佐利来说，过去的经历是序幕，他渴望着继续这场大戏。

究竟什么是操作系统？卡特勒的团队想创造出什么呢？

我们来描绘一下上世纪初一个富裕的英国家庭。把电脑——硬件——想象成一座大房子，这个家庭的住所。房子里有管道，照明，砖，灰浆墙，窗户和门——所有实体的物件应有尽有。

接下来，想象一下计算机软件就是生活在房子里的人。家政人员住楼下，他们提供全套及时的家政服务。管家立在门边，司机在洗车，保姆在熨着衬衫，厨师提供一日三餐并烤着面包，园艺师在草坪上扫落叶。所有活动，看起来都像是分别在按照自己的进度进行，但实际上全部都由管家来协调和组织。这就是楼下居民的生活，在某种意义上，它是存在于背景中的。

接着考虑一下楼上的人。他们是楼下人辛苦劳作的全部理由。丈夫需要一个司机，不仅是为了省心，更是因为他想要外出旅行。妻子请来一个厨师，这样她的家人就可以吃得很好。孩子们从园艺师的工作中受益，他清除院子里的碎石片，让孩子们在室外可以安全地嬉戏。

对楼上家庭和楼下忠诚的仆人的生活的描述清晰地说明了软件世界里的重要分工。楼上的居民是应用软件：文字处理，电子表格，数据库，电子排版和大量其他能满足人们需要和愿望的程序。而楼下的居民则共同发挥了操作系统的功能：他们的是服务的领域，有些是自动的，有些则有特殊请求的。这些服务是提供优越生活物质条件的基础。

卡特勒对他的操作系统有三个期望：

可移植性：这可是软件业的圣杯——创造出一个可以适合每一种硬件的操作系统。当时，每种商业操作系统都是为特定系列的芯片而量身定做的，所以只能在装载这些芯片的电脑里发挥作用。程序员用最接近于目标硬件的汇编语言写了大量的编码。将软件和硬件联系在一起。当硬件成本远大于软件成本时，没有人介意这些。不过，后来当消费者发现在应用程序

上的投资大过于购买硬件时，他们就不愿意抛弃他们已安装的但已过时的应用软件。NT 便是创造一个通用系统的机会，它将在电脑世界开创一个新纪元。

可靠性：在苹果机上应用的是 OS/2 和 DOS 操作系统，一个应用程序的运行失败往往会导致整个系统停止工作或崩溃，数据消失，迫使使用者重新启动机器。如果许多应用程序在同一时间运行——NT 的未来特色之一——其中一个的运行失败会导致其他应用程序的停止运行。卡特勒想用“防弹”系统来结束这种不必要的崩溃。有了“防弹”系统，个人电脑的价值会被增大，因为它们能可靠地支持在商业世界中所谓“关键任务”的服务，比如航空订票或者银行自动柜员机。卡特勒和他的团队知道如何来保证可靠性，他们的 DEC 操作系统是在电脑还是十分昂贵和稀有的时代诞生出来的，卡特勒解释说：“如果系统死了一个小时，那就糟糕透了。首先，当时你不能象现在那样，对一台电脑做那么多操作，所以每一分钟都损失惨重。”

个人化：NT 应该灵活十足，它将拥有不止一种用户界面，或“个性”。当时的个人电脑操作系统往往只有一种个性，也就是说，为 DOS 写的应用程序就只能运行在 DOS 上。NT 将会支持 OS/2 的表示层管理器界面，但它也会设计成可以——如果微软选择这么做——支持其他诸如 DOS 或 Windows 或还没发明出来的操作系统的个性。

在构思 NT 的时候，卡特勒及其部落们首先攻克可移植性的难题。

可移植性的代码比较容易编写但效率比较低。理论上，用高级语言写出来的代码在经过编译后，可以在不同的微处理器上运行，而高等级的语言写出来的程序相对较慢，所占空间也大。微软的 DOS，DEC 的 VMS 和苹果的麦金塔操作系统，用的都是汇编语言，因为它写出来的程序小而紧凑，充分地发挥了特定硬件的作用。紧凑的编码固然是好的，因为它节约了内存的空间。内存空间包含了很多芯片，名叫 DRAM（动态随机存取存储器），就好比一个汽油箱。汽油箱越大，汽车就跑得越远。操作系统越小，就有更多油留给其他程序使用。

因为 NT 是可移植的，所以卡特勒知道它会很大。问题是，将有多大？这取决于他的程序员们有多聪明，以及 NT 自己的设计特色和功能。卡特勒需要花很多时间才能知道 NT 究竟需要多大一个内存油箱。可是，他已经意识到 NT 会消耗很多内存——也许比现有的个人电脑操作系统所需的要多得多。

这让比尔盖茨感到紧张。NT 最少需要一个带有 8 兆内存的电脑，这是当时一台新电脑标准配置的 8 倍以上。盖茨担心消费者会不用 NT，因为它意味着消费者需要为它再花几百美元购买额外的内存。虽然内存芯片的价格可能会大幅度下降，但 NT 的内存要求仍然是大大的未知。

这个任务落到了皮亚佐利头上，他来负责设计和编写 NT 中管理内存代码（他在卡特勒的实验室里曾写过类似的内存管理程序）。内存管理要确保操作系统的每部分和每个运行的应用程序在它们需要时都能有足够的内存空间，所以它自己要一直保持运行状态。这是个复

杂的任务，因为，当一个应用程序需要内存而当时没有时，就需要内存管理程序来精确地搜寻出可用的内存空间。而且“每个人都需要内存，而且他们想立马得到。”，使这项任务显得更加艰巨。

第3章 部落（5）

为了更好地理解内存管理程序的作用，我们来想象五个饭店，它们共用一个停车场，但是车位有限，假设只有100个车位。想在这五个饭店的任何一家里吃饭，在这个停车场里停车是唯一可行之路。一个孤单的车仆负责车进车出，和代泊每一辆车。饭店象征应用程序，停车场象征电脑的内存，车仆则代表操作系统里的内存管理程序。早上，只有一家饭店营业，所以有足够的闲置车位，车仆的管理工作也很轻松。午饭时间则显得压力重重，因为五个饭店都在营业，对车位的要求很紧张。

车仆于是开始筋疲力尽。他不仅需要留意每一个车位的状态，而且必须迅速无误地代泊每一辆车，并在客人离开的时候，准确地找到它们。既然停车位占用的是物理空间，那它肯定能被扩张：增加车位成本很高，但是它能缓解停车场的阻塞。而更好的解决办法是，车仆必须更聪明更敏捷地工作。当车位已满，车仆必须在入口等待车位空出来。优秀的车仆能够指挥这些车辆的，所以当停车场暂时不能运作的时候，没有人会引起争吵。而无能的车仆却常常不能很好地管理未被停放的车辆，所以场地充满着混乱。然后，每个客人都得离开他/她的饭局，找到他/她的车，重新再停一次车。

车仆的工作形象地显示，稍不注意，局面就失去把握。皮亚佐利想创造出一个聪明绝顶的车仆，但他坚持用陈旧的方式。

对可靠性的要求极大地影响了操作系统的整体设计。什么是隔离应用程序的最佳方式呢？这样当应用程序出现故障时不会使 NT 也失败？卡特勒的解决方案是把操作系统分成两个主要的部分。其中一个“内核”，它不直接和应用程序关联，所以不会被它们影响。另一个是操作系统的图形化的，可视的部分。

我们回到楼上一楼下的比喻来领悟一个系统内核的要领。想象下家政服务团的管家，他是多么地勤劳：他24小时不分昼夜地为楼上的家庭服务，随叫随到，处理各种要求。如果需要做顿饭，他就去做；如果衣服需要洗，他就去洗。如果他不在，他也会安排好一个候补。他把房子里的一切保持得井然有序。楼上成员的任何意外的行为都不会使楼下的生活休止——从不。

这就是卡特勒想要的操作系统的大致情形。就像楼上的家庭，应用程序可能会有意外的或不合理的行为。在程序的范畴里，当所有其他程序都运行失败的时候，唯一能保持运行的就是系统内核。它用给应用程序强加一定的限制来保护它自己，最重要的是，只有它，而不可能是应用程序，来操控硬件。

系统内核本身的结构就很复杂，包括很多组件，组件之间互相调用，或者直接调用硬件关联。系统核心的组件常常调用底层代码——“微核”，它是 NT 里的终极统领。卡特勒曾在 DEC 写过一个微核，打算也为 NT 写一个。这看起来像极了他在他团队中的地位，微核在操作系统里也颇具权威。

内核设计有两个主要的优点。第一，当一个程序突然停止的时候，内核允许用户保持其它的应用程序的正常运行，还可以启动新的应用程序，从而保证了系统的可靠性。第二，内核可以使操作系统显示出多样的个性。每个操作系统的个性都通过调用内核来向计算机发出命令。OS/2 将会是这些个性之一，这是盖茨的要求。但是 NT 也要能支持其他个性，成为一个潜在的通用的软件平台。

卡特勒内核设计的令人眩目的优点——可靠性和多样的个性——可是有代价的：速度，或者说缺乏速度。这里有严格的逻辑性。把一种个性加入模块里，意味着系统发出的调用命令的数量起码翻了一倍。比如说，一个应用程序必须先调用 OS/2 模块，然后 OS/2 调用内核。用卡特勒的话来说，个性变成了内核的“客户”，内核为客户“服务”。

当盖茨第一次听到这种客户-服务模式时，他犹疑了，想到：“这样做的话，会有巨大的额外开销。”他对他的观点有充分的证据。“我的头脑里清楚地知道这个额外开销有多大，所以我认为我们不能这样去做。”他坚持。但是卡特勒有力地为客户-服务模式辩护，说它不会降低性能。他提供了数学数据来支持他的模型，他部落里的一个成员还写了一篇十二页的论文来说明其他设计方案是不可行的。

在如此激辩之后，盖茨对 NT 的基本设计还是不放心的。他比卡特勒更了解消费者对程序运行速度的关注。他们会把 NT 和 DOS 和 OS/2 比较，而后者是用汇编语言写成的，可更好地发挥硬件的性能。固然，微芯片总有一天会变得很快，可以抵消 NT 的缓慢。但是让软件任意遨游的超快芯片的到来还得等5年到10年。盖茨担心舆论早把 NT 定死在速度的十字架上。

卡特勒坚持他的立场。他不耐烦地发誓说他的团队会写出有魔力的代码，他们高超的编程技巧技术会弥补客户-服务模式的不足。盖茨将信将疑，但也默许他们这么去做，他想：“嘿，这是他们的项目，是他们的项目。他们认为那样可以。”

在1989年年初时，NT 的设计蓝图初具雏形，这时卡特勒开始恢复他的真本性，不再像刚到微软时那么彬彬有礼了。他总是不耐烦，传达的信息也总是生硬而简洁明了。他不喜欢重复自己说过的话。“如果他给你解释过一次，他希望你再也不要问他了。”一个团队成员说。卡特勒要求别人要紧跟他的节奏。当有人请他重新做一次解释时，他也会去做，“这一次声音只是有点响亮”，一位在那年夏天加入卡特勒团队的微软人如此说。“然后，如果你再请求他进一步阐明他的意思，这时，他的脸开始变红了。就好象，因为你不明白他的话，他就要发疯。所以我学着对他所说的每一句话都听得很仔细，有时候我也不懂装懂。然后我回到办公室，再思考他所说的话，或者我去问卢[皮亚佐利]，‘大卫到底是什么意思呢？’”

卡特勒从来不为他的缺乏耐性或火暴脾气抱歉。“我靠压力过活。”他自吹。他把他的团队看作是建立在分享共同的欢乐，牺牲和秘密的一个小社会。他既是严格的执政人，又是体贴的族长。他对自己的要求永远都比别人的多。在我们现在的社会，借口是换取宽容的通行货币，但卡特勒从不接受。“我希望每个人都做到他们自己的最好——一直如此。”他说。

第3章 部落（6）

对他的亲密战友来说，卡特勒就像自然之力，但随着时间的过去，他脾气的爆发成了日常例行公事。和卡特勒不熟的人常常被警告说要保护好他们自己。“当你看到卡特勒要发作的时候，赶紧躲掉。”卡特勒的一位朋友对一位微软人说。后者一开始觉得这个建议未免太夸张了些，可是后来她听到卡特勒办公室方向传来恐怖的吼声，她问：“这是什么噪音？”这就是卡特勒发脾气的声音。

卡特勒的脾气可谓是一触即发，但有时他的咆哮只是故意做样子。“我向人们大吼大叫时，并没有发疯。”他说，“他们可能认为我疯了，可是我真的没有疯。”不过，要分辨出到底是不是装样子，谁也说不准。当卡特勒开始讥笑微软为“这个渣滓公司”时，很多人都觉得这是他的真实意思。

麦沃尔德想缓和一下紧张的气氛。他邀请了公司里称为“架构师”的优秀软件工程师们，去会见卡特勒等人。可是事与愿违，原因出现在戈登·莱特温身上，他是设计 OS/2 架构的元老。卡特勒曾被要求主持改进的 OS/2 项目，对此莱特温觉得自己受到了冷落。他还觉得麦沃尔德和卡特勒都对 RISC 芯片的前景和可移植程序的期望值过高。当他和卡特勒团队对话的时候，莱特温显得盛气凌人。这让卡特勒感到不愉快，他本来就不喜欢 OS/2 的“不值一文的设计”，他认为 OS/2 能力不足，可靠性也不强，压根儿就不能承载更多的功能。在测试 OS/2 之后，卡特勒感到“心口疼痛”，因为他已承诺，NT 将可以运行 OS/2 的程序。“这样，我们在努力制造最先进的艺术品[可移植系统]，还要让它和垃圾 OS/2 兼容。”他说。

莱特温觉得卡特勒不能理解微软为了赢得 IBM 而所做的妥协——卡特勒不会去做的妥协，因为 IBM 对 NT 的发展不提供任何帮助。莱特温发现设计 OS/2 的时候，好的点子不是被摒弃，就是处于无休止的争论中。他对卡特勒的自负也感到很愤怒，他认为卡特勒把自己看作是普罗米修斯，“从众神那儿带来火，以此来照亮（微软的）可怜的众生。但我们的确没这么想过。”

在1989年3月9日，阿尔丽塔·卡特勒打电话告诉她的儿子，他77岁的老父亲不幸因心脏病去世。卡特勒似乎无动于衷，不过他母亲也没吃惊。儿子与他父亲之间有鸿沟，她说：“他对此一点感情都没有。”然而不少人怀疑尼尔对儿子的冷淡是卡特勒保持愤怒的“源头”。不管他和他部落中的程序员们有多么亲密的关系，也不论他通过他的成就获得了多么大的财富，生气这两个字总是与他如影随形。它们紧紧地陪伴着他，就像是执拗而熟悉的朋友。当他在令人沮丧的时刻发作脾气的时候，他的行为就仿佛是被压抑青年时期的卡特勒的反抗，

那时他父亲对他所取得的成就毫无感觉。就如卡特勒高中时期的一位足球教练，拉里·车切斯所说：“大卫年轻时，缺少的是一位可以鼓励他的父亲，对他说：大卫，你做得很好。放松点。”

内尔·卡特勒望子成龙，希望儿子能通过竞争或竞赛赢得尊重和别人的倾慕。“不管大卫做什么，他都得证明他做得比别人好。”车切斯回忆道。他连最休闲的娱乐也不放过，也要拼命力争上游。“如果你有位父亲说你已经做得很好，你就不用一直去证明它了，但是大卫不停地证明给他父亲看。”

关于父亲的死，卡特勒几乎无话可说：“这基本不影响我什么，我们向来不亲近。”跟往常一样一样，他言语不能够及的，他通过行动来表达。

他没出席他父亲的葬礼。

卡特勒的圈子并不适合喜欢看限制级电影的人。蜂拥来面试卡特勒秘书职位的女士们都会被问到这样一个问题：“关于 fuck（英语脏话，可译为他妈的）这个词，你怎么看？”

“这是我最喜欢的单词！”卡丽·威尔逊获得了这份工作。她曾在微软的销售部门做过助理。卡特勒的坏脾气似乎并不太影响威尔逊的状态，她很快就开始享受卡特勒使者的权利了。通常有任何请求只要提一下的他的名字都会得到快速的响应。她说：“卡特勒就好比是你自己的罗特韦尔犬，他对别人可能比较凶，但从那样对我。他很支持我的工作。”

在部落工作了几个月之后，威尔逊迎来了三十岁的生日。为了帮她庆祝生日，她被带到肖特的硬件实验室，他们在那里举行了一个聚会，还邀请了脱衣舞男为她助兴。不过，她有点被激怒了，于是想报复一下。在下一个男同事生日聚会的时候，她找了一个女朋友来表演脱衣舞。她朋友还带来了手提式音箱。她打开音乐，开始跳舞，向过生日的小伙子慢慢靠近。突然，她停了下来，看着小伙子说：“他们没付够钱，要我给你脱衣服，没门儿。”然后带着手提式音箱扬长而去。小伙子楞住了，而卡特勒则狂笑不止。

工作中，卡特勒可以在一眨眼工夫从笑脸菩萨变成一头怒狮。他不喜欢吊儿郎当，建议办公室里禁止娱乐。他从不考虑别人的感受。他的并不是故意想让所有人都感觉很糟，而是希望工作时别去顾忌情绪问题。“我们来这儿是工作的。”他说。一个领导不应该“好像走在鸡蛋壳上，四处陪小心。伙计，今天你感觉怎么样呀？情绪不错，可以工作吧？你能来上班，简直太棒了。”“嘿，我们可是拿薪水的，对吗？所以我希望每个人时刻保持最好的工作状态。我希望他们做好自己的工作：如果他们一直做得很好，他们和我相处得也很好。可如果他们不好好工作，那跟我肯定相处得不好。”

技术争论既是实验室里的福音，又是祸根。工程和发明总是有不同的方法到达同样的结果。关于技术的恳切争吵在所有的技术公司都很常见。程序员们把一些分歧看作是“信仰不同”。偏执狂喜欢争得你死我活才罢休；中立派则往往会说，两者都可以。可是不能用逻辑

说服对方的偏执狂——只能扔下狠话了。

使偏执狂争论不休的古怪问题之一，是编写 C 语言程序使用的符号系统。C 语言是使用最广泛的计算机编程语言之一。在过去的几年中，微软公司已经形成了自己的一套命名规范，就是“匈牙利表示法”。它的创作者是出生在布达佩斯的查尔斯·西蒙尼，他也许是微软最有影响力的程序员。匈牙利表示法把大量的信息用一种紧凑的形式表达出来，但是它是由一些晦涩难懂的缩略语组成的，缺少使用英文单词的传统语言那种人们熟悉得表达能力。

第3章 部落（7）

两种不同风格必然会导致数不尽数的讨论。下面有两行指令，它们的内容是一致的，上面那行用的是匈牙利表示法，而下面那行则不是：

```
pfi->pbufCur->ibCur = ibFile;  
File->CurrentBuffer-> Offset = File Offset
```

卡特勒和他的部落们反对使用匈牙利语言。这让伍德很高兴，他曾说匈牙利表示法“是我见过的最愚蠢的东西”。他还补充道，“编程风格之争的确浪费了宝贵的资源，但匈牙利表示法也许浪费的更多。不过，很多宗教也这样。”

关于匈牙利表示法这个“宗教”问题，伍德是站在卡特勒这边的。可是，没有任何迹象表明伍德会成为卡特勒的又一膜拜者。他觉得前 DEC 的男人们有些古怪，他们“围着卡特勒走在鸡蛋壳上，神经兮兮，惟恐得罪了他。”他们当中，有些人害怕卡特勒，而有些人对卡特勒却崇拜得不得了。他们对卡特勒的热情真是太直露了，有个微软人问道：“难道卡特勒是弥塞亚？”

从某种意义上说，答案是肯定的。卡特勒给他的部落成员们以生活的目的，在某种程度上，他像一个天生的领导人一样振奋起他们的精神。他浑身散发的自信力，能使他的追随者们感觉到事物都是正确的和好的。

然而，伍德更能客观地看待卡特勒，他对卡特勒没那么多热爱之情。“史蒂夫不惧怕大卫，”卡特勒的一个亲密战友说，“他并不把大卫看作是一个大英雄，他对大卫的指令持怀疑态度。”

伍德有实力为他自由的心灵买单。他在微软的工作使他拥有价值几百万美元的公司股票——和经济上的保障——所以在卡特勒面前，他不需要做任何妥协。如果 he 不想工作，他随时都可以停下来。

在整个团队中，只有他意识到卡特勒的事业到了十字路口——他的时代也许过去了。许多工程师的黄金期都不长，每个伟大的工程师或发明家的命运大体都这样。伍德觉得卡特勒有许多需要证明的东西。另外，他还觉得卡特勒从 DEC 的灰烬中走出来，到微软来领导 NT 的

开发的这个故事，也许激励的意味要比知识经验本身多。这个故事应该是这样：卡特勒因为被 DEC 从背后捅了一刀，并被无情地逐出公司，所以到微软来了。卡特勒的火暴脾气扼杀了棱镜，伍德觉得，“卡特勒的过错和 DEC 一样多”。

在一些小的方面，伍德把他对卡特勒的不敬流露了出来。卡特勒十分喜欢给他的代码增加大量的注释。这些被程序员们称为“编码注释”是写在程序里的，这样，对编码原作者意图感兴趣的人就可以看到了。读这些注释，虽然有点无聊，但是很有用。原始的代码编写者离开之后，代码还会存在。注释可以帮助理解代码背后的动机和原因。如果想修改或改进一个程序，编写良好的注释作用很大。卡特勒的注释很漂亮，但做得有些过头，实际上他的每行编码都有注释。伍德的异义部分是因为他写代码比卡特勒多很多，所以几乎没多少时间来写注释。“注释的问题是代码总在进化，”伍德说，“我同时在忙五件活儿，代码也天天在变，那注释也得相应地修改。”当然，卡特勒对伍德的难题有个简单的解决方式：修改代码的同时，修改注释。

伍德对文档化其工作的不情愿态度让卡特勒“生气，史蒂夫是个很难管的家伙，难管极了，”卡特勒说。“他随时可以退休。所以他的态度就像是：我想做什么就做什么。”卡特勒向皮亚佐利抱怨对伍德的不满，但也只没别的办法。皮亚佐利觉得与伍德争论他从没有改变过得习惯是没有意义的。而且，有一个未被“改造”的伍德总比完全没有伍德好。

剑拔弩张。卡特勒说：“我这里没有小事。”有一次，他和伍德就一些失误当面对质。他心里怒火燃烧，说话的声音就像是大口大口地吞水。他站在伍德面前几乎碰到他的鼻子。伍德心里嘀咕：“他会对我动手么？”

卡特勒没动手。

卡特勒和伍德之间的矛盾总算控制在合理范围内，但他和达瑞尔·鲁宾的摩擦却没有控制好。鲁宾是微软的首席网络软件工程师，他既聪明又富有幽默感：他是软件业里的伍迪·艾伦（译者注：伍迪·艾伦是一位20世纪后半期重要的美国电影导演和喜剧演员）。卡特勒到来时，他曾十分激动，认为他是“行业中的巨人之一”。他在大学里就用过卡特勒的程序，对卡特勒的设计“超乎一般地熟悉”。但是当他第一次见到卡特勒本人的时候，鲁宾被吓坏了。卡特勒大步迈进房间，“就像空军上校... 胡须剃得干干净净，寸短的头发，还有那固执，严肃，坚忍不拔的神态。”鲁宾觉得做软件的人不应该那么一板一眼。

网络设计已成为微软的一个痛处。它纵然是微机和工作站的典型特征，可是在个人电脑世界里，它算是迟到者。除了苹果公司的麦金塔系统使电脑可以简单有效地共享文件和打印机，消费者还没找到可以连接不同品牌微机的标准模式。犹他州的诺维尔公司抓住了这一机遇，它的 Netware 程序成为同行中的佼佼者。它成功地把很多电脑连接起来，可以共享一台打印机，也可以处理在别的电脑上的文件。打印和文件服务，虽然是平凡的事务，却是个人电脑网络设计的活力源泉。诺维尔的成功很大部分得益于服务的传达很快，微软不仅难以超

越，就连追赶都有困难。同时，鲁宾在率领一个团队开发 OS/2 的网络体系 MAN 局域网，这是微软试图赶超诺维尔的最后希望。

卡特勒将他的严谨观点也带到了网络设计上。他认为诺维尔的 Netware 经常出问题，而且缺乏防止入侵者进入操作文件和打印机的有效设置。他认为，如果有一种操作系统能够提供可靠安全的网络，人们肯定趋之若鹜，哪怕速度慢一点。

鲁宾觉得卡特勒错了。消费者最在乎的是软件的速度。微软除了提供最快的编码，别无选择。鲁宾坚持。而且完成这个任务的最佳人选就是鲁宾，“因为我高度关心性能问题，而卡特勒则不太在乎。”而鲁宾的要求“创造了与卡特勒之间必然冲突”，卡特勒高调强调他的观点是正确的。鲁宾的一位同盟者把纷争点归纳得很透彻：“我们说：别再去买教训了。他们说：我们如此聪慧，绝不可能犯错误。”

他们俩谁也不肯让步。鲁宾跟卡特勒说别插手网络设计的工作。卡特勒冷冰冰地回应：“我就是想告诉鲁宾，他是这个公司的网络设计的架构师，这没有问题。但我们准备做自己的他妈的（网络体系），不用他告诉我们去做什么。”

后来卡特勒对网络设计的干预慢慢淡去，鲁宾写了嘲讽卡特勒大男子主义的一系列新闻快讯来庆祝这胜利。

华盛顿州雷德蒙——今天有四位程序员死于荒诞事件。大卫·N·卡特勒把他的阳物扔到二楼会议室桌上，阳物冲破地板直坠楼下的实验室，砸死了程序员，还损坏了三台计算机。而卡特勒的阳物只有一点损坏。

华盛顿——一次四英尺高海啸，或者说大浪摧毁了西海岸线大部分地方。联邦紧急事件管理机构称这是美国有史以来最严重的自然灾害，估计损失将达5000亿美元。

华盛顿州雷德蒙——营救人员在被海啸摧毁的微软总部现场发现了第一位幸存者。

大卫·N·卡特勒被发现时，他靠在一座喷泉旁边。据报道，海啸到来之前，他在和微软的另一名职员进行撒尿比赛。

看起来，卡特勒唯一的损伤是有些脱水。