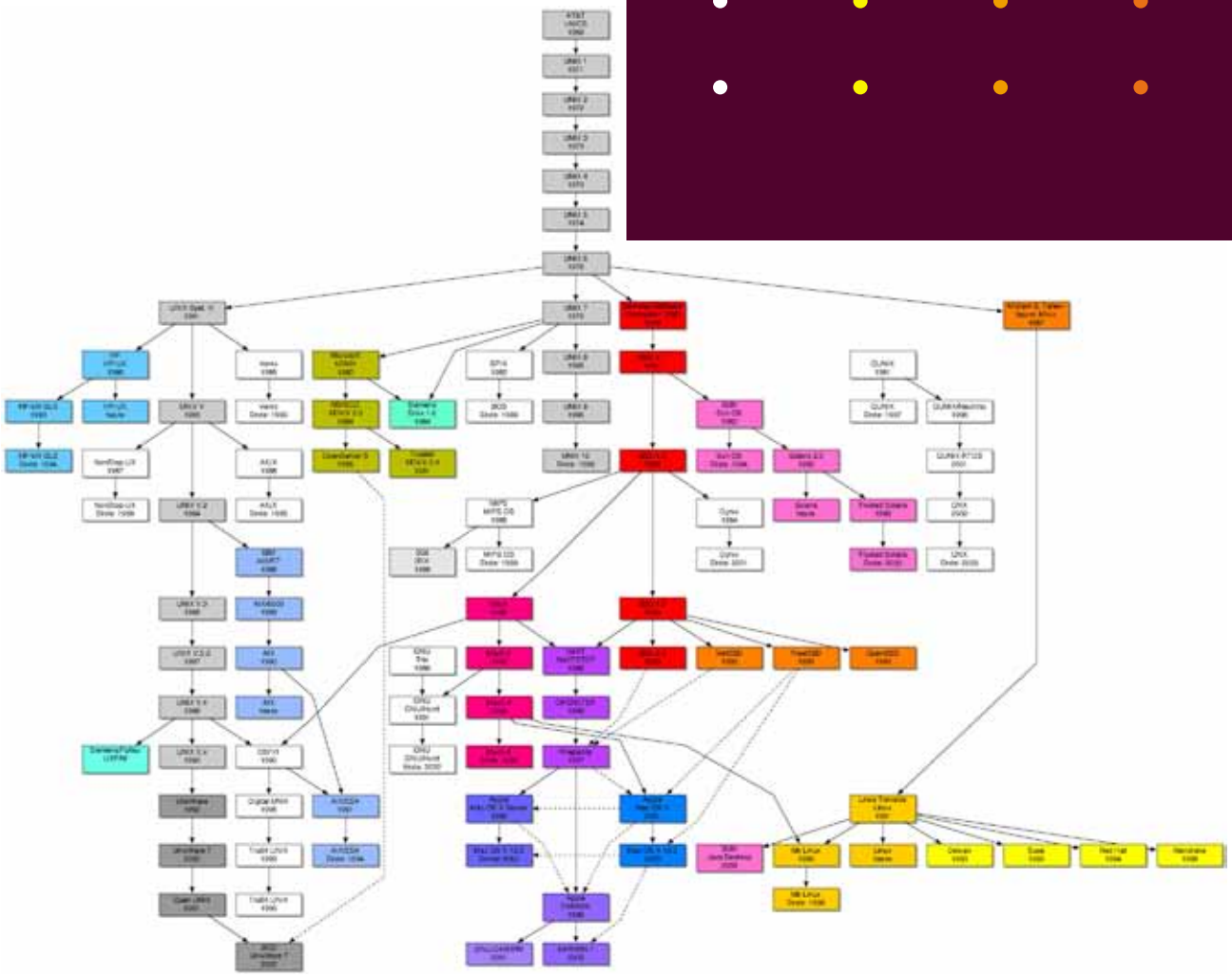
序

一切的起因



伟大的 UNIX 家族

UNIX

我们的故事从上个世纪 60 年代的美国开始。

那个时代的计算机是个新鲜玩意，非常笨重，家庭用户是没有的，都是商用或者试验，科学计算用的机器。你说你想买个电脑斗地主？把你卖给地主你也买不起呀。再说那时候的计算机不是随便一个人就会用的，那时候的计算机使用的时候是由人来输入一条条的指令，来进行各种运算的。他们输入的指令大约相当于现在的汇编指令，所以这个效率和操作难度有多高就可想而知了。那时候计算机大都没有什么操作系统，顶多有个批处理系统，可以把要输入的指令记录在某种媒介上（比如纸带）一次性输入进去，让人们省去一条条重复输入指令的麻烦。后来慢慢有了很简单的操作系统，但并不像现在我们见到的操作系统这样通用。这个时候，卖计算机的厂商要为每一型号的计算机设计不同的操作系统，一个程序如果在这个型号的计算机上写好了，拿到其他型号的计算机上是运行不了的，因为这两台机器连操作系统都不一样，怎么可能程序通用呢。

计算机要是老这样肯定是不行啦，否则你今天要玩斗地主，人家游戏公司就得专门派人到你家机器上现写出一个来——因为不同型号的计算机上的操作系统不同用嘛。这个斗地主的问题，终于还是被那个时代 IT 业界的大地主，蓝色的 IBM 公司率先着手解决了。1964 年他们公司推出了一个系列的大型机，用途、价位，各不一样，但他们上面运行的操作系统，都是 System/360。（这 360 可不是卖鞋的，也不是跟 QQ 打架的那个。）这一下获得了很大的成功，因为省去了为每一台电脑单独编写系统的成本嘛。直到今天，IBM 的大型机上依然可以运行这个 360 系统，可见其当初设计时充分考虑的兼容性。然而我们要讲的主角不是 360，而是另一个伟大的操作系统。

那时候有个聚集了很多牛人的地方，叫做贝尔实验室，是 1925 年由 AT&T 公司成立的。一帮头脑发达四肢也不一定简单的家伙整天聚在那里，研究新奇的东西，什么任意门啊，竹蜻蜓啊，记忆面包啊——呃……都不是他们发明的（发明这些的人是个日本科学家）。那贝尔实验室那帮人研究什么呢？贝尔实验室的工作可以大致分为三个类别：基础研究，系统工程和应用开发。在基础研究方面主要从事电信技术的基础理论研究，包括数学、物理学、材料科学、行为科学和计算机编程理论，反正都是大学听不懂的那几门就对了。系统工程主要研究构成电信网络的高度复杂系统。开发部门是贝尔实验室最大的部门，负责设计构成贝尔系统电信网络的设备和软件。具体来说贝尔实验室研究出来过的东西有晶体管、发光二极管、数字交换机、通信卫星、电子数字计算机、蜂窝移动通信、有声电影、立体声录音，等等。（怎么样，不比机器猫那些东西差吧？）通信网的许多重大发明都诞生自这里。

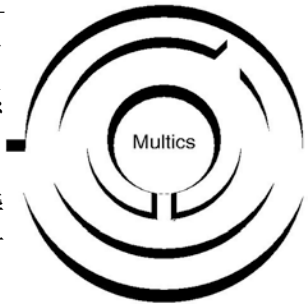


那时候还有个聚集了很多牛人的地方，叫做麻省理工学院（MIT），这是美国的一所综合性私立大学，有“世界理工大学之最”的美名。从这里走出的牛人很多，到 2009 年为止，先后有 76 位诺贝尔奖得住，都曾经在麻省理工学院学习或者工作。麻省理工学院的自然及工程科学在世界上享有极佳的盛誉，其管理学、经济学、哲学、政治学、语言学也同样优秀。另外，麻省理工研发高科技武器和美国最高机密的林肯实验室、领先世界一流的计算机科学及人工智能实验室、世界尖端的媒体实验室、和培养了许多全球顶尖首席执行官斯隆管理学院也都是麻省理工赫赫有名宝贵资产。有着毋庸置疑的实力，麻省理工自然非常不差钱，截止 2008 年底麻省理工有 101 亿美元的总资产，因为不差钱，于是该校对家庭年收入低于 75000 美元的学生一律免学费。哎，万恶的资本主义阿……不提也罢。

那时候，又有个聚集了很多牛人的地方。（哪那么多地方阿！）这地方是个公司，叫做通用电气。这公司可是个大公司，当年是个

卖灯泡的，他们的灯泡可非同一般，虽然不节能，虽然寿命不如现在的长，虽然价格比现在贵，虽然外形也不一定好看，但是——他们是第一家卖灯泡的！因为他们的老大，就是大名鼎鼎的托马斯 . 爱迪生。1876 年，发明灯泡的爱迪生同学成立了爱迪生灯泡厂，为节约蜡烛和灯油做出了突出的贡献，估计那年的五一劳动奖章肯定是他的了。到 1890 年，爱迪生同学将灯泡厂重组，成立的爱迪生通用电气公司，到 1892 年又与汤姆森－休斯顿电气公司合并，成立了通用电气公司。

好，时间到了 1965 年，这三个聚集着不少牛人的地方有一天忽然想合作一把。于是，大名鼎鼎的贝尔实验室，大名鼎鼎的麻省理工学院和大名鼎鼎的通用电气公司一起开始了一个制作操作系统的计划。为了结束长期以来计算机上面没有统一的操作系统的混乱局面，他们决定要创造出一套前无古人后无来者，上得厅堂，下得厨房，念天地之悠悠，独怆然而泣下的惊世骇俗的操作系统！具体来说吧，这个操作系统应该是一个支持多使用者、多任务、多层次的操作系统，因为这三多，所以这个操作系统就起名叫做——MULTICS。（难道你以为叫许三多？人家讲的是英语好不好。）有了这三家的强强联合，那开发的结果还用问么？这个 MULTICS 操作系统的项目在 1965 年成立，到了 1969 年就……被取消了。



咳咳，这个……其实编写操作系统也不是一件容易的事儿啦。毕竟道路是曲折滴，研究是辛苦滴，成绩还是有滴，失败捏……也是可以原谅滴嘛。

项目失败了，大家都很沮丧。在这些沮丧的人中，Kenneth Lane Thompson 只是很普通的一个。Kenneth Lane Thompson 是 1943 年出生在美国的新奥尔良的，吃着烤翅长大的他，没有辜负养育他长大的父母和那些没有了翅膀的鸡。1960 年，汤普逊考上了加州大学博克莱分校主修电气工程，顺利取得了电子工程硕士的学位。1966 年，他加入了贝尔实验室，参与了 MULTICS 项目。做项目是个很辛苦的事情，在疲劳的揉揉因熬夜而发红的眼睛后，他很想能有个电脑游戏来玩玩。然而那时候别说超级玛丽，连吃豆也没有啊！所以汤普逊同学就自己编了一个游戏，叫做星际旅行。这个星际旅行跟星际争霸那肯定是没的比的了，不过在那时候已经算是很有吸引力了。这个游戏自然是被设计运行在 MULTICS 系统上的，由于 MULTICS 系统还不完善，所以游戏运行的也不是很流畅，所以，能够顺畅的玩星际旅行，成为了汤普逊同学努力工作的动力。可是后来项目被干掉了，如果事情就这样结束，那么汤普逊同学就再也不可能流畅的玩他的星际旅行了，这是多么遗憾的事情阿。可是现实是残酷的，项目确实就是取消了，要想顺畅的玩游戏怎么办？毛主席教导我们说：自己动手，丰衣足食。我估计汤普逊没有背过毛泽东语录，但是他用自己的行动证明了这两句话的正确性。他在墙角淘换出一台 PDP-7 的机器，并且伙同其同事 Dernis Ritchie，打算将星际旅行移植到这台 PDP-7 上。

当然，要想运行这游戏，还是得有个系统。有了固定的系统，那以后再编写别的游戏就

更方便了。可是系统从哪里来？ MULTICS ？已经停工了，并且这系统绝对不是两个人可以搞定的。那怎么办？还得自己动手！于是 Ken Thompson 和 Dernis Ritchie 再次发扬自己动手的精神，用汇编语言写出来个系统，这就是最初的，非常简陋的，UNIX 的前身。这个系统不像 MULTICS 那么牛，不支持很多的用户，只能支持两个用户，（估计是为了避免做好了之后俩人抢机器玩的局面发生。也可能是为了以后俩人对战？）支持的进程也有限，其他功能也都没有 MULTICS 设计的那么复杂。相对于那个 MULTICS 系统——MULTiplexed Information and Computing System，Brian Kernighan 开玩笑地戏称他们的系统其实是："UNiplexed Information and Computing System"，缩写为 "UNICS"。后来大家取其谐音，就诞生了 UNIX 这个词。这一年，已经是 1970 年，史称 Unix 元年。直到现在，计算机中都是用 1970 年 1 月 1 日 0 点 0 分 0 秒为原点来记录时间。（计算机中的时间记录的是自 1970 年 1 月 1 日 0 点 0 分 0 秒开始，到现在经过的总秒数，再用这个秒数计算出年，月，日）后来，Brian Kernighan 觉得用汇编写的系统不好维护，于是……他发明了 C 语言（符合大牛一切自己动手的风格），然后用 C 语言又重写了一遍。从此，Unix 走上了发展的快车道，并且一直用到现在。许多世界级的大服务器，用的都是 Unix 系统。

而这一切的努力，就是为了玩个游戏。-_-b

Ken (seated) and Dennis (standing) at a PDP-11 in 1972

这回要说的，是另一个传奇人物。

Richard Stallman,1953 年出生在美国纽约曼哈顿地区，他从一出生就……没什么特别；他上小学的时候……反正我不认识他；等到他上初中的时候呢……也还没我呢。总之，他在生命的前十几年中并没有表现出什么过人的地方，因为他没遇到一个叫做电脑的东西。

高中的一个暑假，他去给 IBM 打工，花了两周的时间用 Fortran 语言编了一个数据处理的程序。这是他第一次接触计算机，或许就是这次相遇，确定了他未来行走的方向。后来，1971 年，他考上了哈佛大学，听说这学校不错，怎么也得是个区重点吧。上学的同时，他还受聘于麻省理工学院的人工智能实验室，成为了一名职业黑客（黑客这个词没有贬义）。也不知道他哪来的那么多时间，可能也是把毛概和邓论都翘了吧。在人工智能实验室的期间，他可没少干活，开发了很多有用的软件，其中最著名的就是 Emacs 编辑器。Emacs 是一个可与 vi 相抗衡的强大的编辑器，他们俩的操作方式完全不同，但却同样强大，各自用自己独有的方式，提高着人们的编辑效率。直到今天，仍然总有人争论到底 emacs 好还是 vi 好，信奉 emacs 的人和信奉 vi 的人形成了两个帮派，这俩帮派经常在大街上用板砖菜刀拼个你死我活。哦，扯远了，咱还回来说 Stallman。

那时候的 Stallman 在人工智能实验室里工作的非常 Happy，大家有 BUG 同担，有代码共享。软件工程师的世界，是一个人人为我，我为人人的世界。因为咱说过，最初的计算机就像我们的算盘一样，只是一个硬件， 没有软件的概念。后来随着电子管、晶体管的发明，计算机的电子成分才超越了机械成分，逐步演化成了现在的电子计算机，在这个过程中，出现了软件，并起到越来越重要的作用，最终成为了

计算机的灵魂。而最初的计算机软件没有什么开源不开源的概念，因为那时候软件天生就是自由的！卖计算机的同时会附带软件，包括软件的源代码和文档。用户可以根据自己的需要去进行修改软件，与别人分享软件，总之，软件是用户花钱买硬件时附带着买来的，用户想怎么玩就怎么玩。软件开发者的目的，也不是靠软件赚钱，而是靠软件支撑起硬件的功能，然后卖硬件赚钱。然而随着技术的发展，软件逐渐脱离硬件成为一个独立的产业，很多软件慢慢的只提供二进制代码而不提供源码了，这就意味着你不能修改它，并且多数还规定最终用户没有二次分发的权利。也就是说，这东西你买了，只能你用，你再给别人，不行！这就好像我买了把菜刀，然后卖菜刀的告诉我“你这把菜刀不许借给你的邻居用，也不许私给菜刀换刀把，否则我就告你！”……囧，你管的着么！？

Stallman 当时就遇到了类似这样的菜刀问题。那时候，他们实验室买的第一台打印机附带有驱动程序的源代码。他们那的黑客们可以随意修改这个驱动，根据自己的需要添加些小功能啊，改改 bug 啊之类的，这为他们的工作带来了很大的方便。后来，实验室又买了一台激光打印机，这次厂商只提供了二进制的打印机驱动程序，它是实验室里仅有的一个没有源代码的软件。出于工作的需要，Richard Stallman 想修改一下这个驱动程序，但是不行啊，没源码啊。后来 Richard Stallman 听说卡内基·梅隆大学有这个打印机的驱动程序源代码，他就去了那里，对他们说：“那啥，大家都是道上混的，谁还没个马高蹬短的时候？是兄弟的拉哥们一把，我也没啥事儿，就是我们那打印机老丢字，一遇到什么敏感的字眼就给我打成口口，我估计是驱动的问题，听说你们这有这驱动的源码，能不能给我拷一份？”对方办事效率还是挺高的，很干脆的拒绝了他。因为他们和厂商签署了一份保密协议，协议要求他们不能向别人拷贝源代码。顿时 Richard Stallman 感到他们背叛了自由的计算机社团，他非常生气，但是他选择了沉默。这只是一件小事，只是一个时代的缩影。那个时代，正处在软件向私有化转变的过程中，越来越多的软件选择了不开放源代码，不允许二次分发的发布方式。甚至 Stallman 身边的同志们也都一个一个都跑到那些靠卖私有软件挣钱的公司去打工了。而 Stallman 依然沉默。

不在沉默中爆发，就在沉默中灭亡。

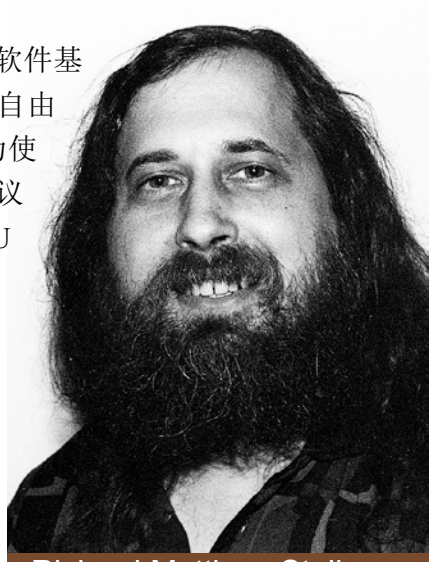
Stallman 爆发了！

他不能容忍软件世界里清新自由的空气被私有软件污染的乌烟瘴气；他不能容忍被剥夺按照自己的需求修改软件的权利和乐趣；他不能容忍自己买条皮带尺寸不够，他竟然连自己在上面多打个洞的权利都没有！

于是，他爆发了。

他要重现当年那人人为我，我为人人的合作互助的软件世界；他要把使用、复制、研究、修改、分发软件的权利还给每一个软件世界的人民；他要用自己的行动告诉人们，软件天生就该是自由的！他要开辟一个新的世界，哪怕是一个人在战斗！于是，一个宏伟的计划在他心中产生——GNU 计划。它的目标是创建一套完全自由的操作系统，因为操作系统是电脑中最重要的最基础的软件，要创造自由的软件世界，自然先要有一套自由的操作系统，然后再以此系统为中心，开发各种各样自由的软件。Richard Stallman 最早是在 net.unix-wizards 新闻

组上公布了 GNU 计划，那是 1983 年的事情。既然要做操作系统，首先得有个明确的规划和目标，目标是什么？这个操作系统要做成什么样子？这当然是要向最成功的操作系统学习，哪个？ UNIX ！ GNU 计划中的操作系统，将是一个类 Unix 的操作系统。这个系统要使用与 Unix 相同的接口标准，这样，就可以由不同的人，分期分批的创作操作系统的不同部分而不必担心相互之间协同工作的问题。

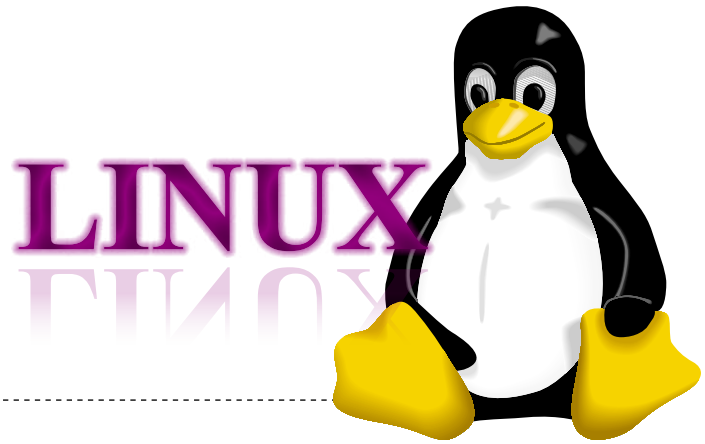

Richard Matthew Stallman

为了实施 GNU 计划，1985 年，Stallman 又创建了自由软件基金会。基金会的主要工作就是执行 GNU 计划，开发更多的自由软件。1989 年，Stallman 与基金会的一群律师们起草了广为使用的《GNU 通用公共协议证书》也就是 GPL 协议，以此协议来保证 GNU 计划中所有软件的自由性。到了 1990 年，GNU 计划中的这个系统已经初具规模，有了很多的优秀的软件。其中有很多是世界各地的黑客们无偿提供的，也有部分是利用自由软件基金会的基金雇佣程序员来开发的，当然，Stallman 自己也是身先士卒，开发了 Emacs，Gcc，gdb 等重要软件。当他看着这些丰富的自由软件的时候，感觉到那清新自由的空气，终于又回来了，以后，人们就可以拥有一个可以自由使用、自由修改、自由分发的，自由的操作系统！

不过等一下，好像还差点什么，哦，还……差个内核吧？

作为一个系统，没有内核是不行的，这么重要的部件 Stallman 当然不会忘记，所以才会有 Hurd 内核。这个内核被设计为一个遵守 POSIX 标准的微内核。所谓微内核，是相对于宏内核来说的。宏内核就像我们现在的 Linux 内核，是一个独立的程序，里面包含了进程管理，内存管理，文件管理等等功能。而微内核则将一个内核需要的功能尽量的简化并且拆分，运行起来是几个独立的程序，有的专门负责进程管理，有的专门负责内存分配，等等。内核是一个系统的核心，所以至关重要，Stallman 对 Hurd 的开发也是精益求精，非常谨慎，以至于内核的进度有些落后于其他的系统软件，当其他软件都已经有比较优秀的版本的时候，Hurd 内核依然不能够走出实验室投入真正的使用。这种情况，一直持续到 1991 年，另一个英雄的出现。

无论怎样，到今天，Stallman 理想中的自由世界，终于拉开了那沉重的幕布，展现出了自由的光彩。而 Stallman 并不满足，也确实没有满足的理由，这个自由的世界还需要成长，还需要更加丰富多彩，还需要有更多的人走进这个世界中来。于是 Stallman 奔走于世界各地，告诉人们有这么一个自由的世界，号召人们加入这个世界，鼓励人们为这个世界更加自由而付出自己的力量。他是一个执着的苦行僧，为了他的梦想，为了他的自由世界，他会一直走下去……

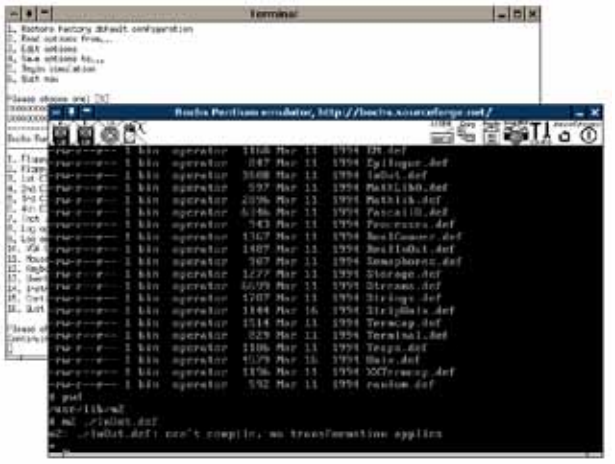


1988 年，芬兰赫尔辛基大学迎来了一位新的大学生—— Linus Benedict Torvalds。当然，那时候他的名字在学校的花名册中并不显眼，但是一年后，他大二的时候，开始有故事了。

大学二年级的时候，Linus 开始学习操作系统这们课程。那时候这门课程使用 Minix 系统进行教学。Minix 这个名字或许您听着并不熟悉，这是个专门用于教学的操作系统，他的系统结构和 Unix 系统是类似的。有人可能问：那为什么不直接用 Unix 呢？恩，Unix 确实是很先进，很有技术含量的，确实值得学习计算机科学和操作系统的同学们学习。然而要知道有一种东西叫做版权，即便你不怎么在乎这个东西，但人家学校是不能做违法的事的。Unix 并不免费，并且是天价的，广大穷苦的大学生们买不起，学校也没钱为没一名学生配备一套 Unix 系统。因此，荷兰阿姆斯特丹的 Vrije 大学的 Andrew S. Tanenbaum 教授最先深刻的体会到了这一点。他的学生们学习了计算机学习了操作系统原理，不能光啃书本啊，总得实践一下吧？总得找台机器装个操作系统用用吧？用什么操作系统来教学呢？买个 DOS 装上？虽然那时候 DOS 已经问世了，但是这么一个单用户单任务效率也不高的操作系统，实在不能指望它培养出什么软件人才。装个 Unix ？学校还不想破产。于是 Andrew S. Tanenbaum 牛人拿起键盘——咱自个儿编一个吧！然后 Minix 就诞生了。Minix 取 Mini Unix 之意，自从 1987 年被编写出来，到 1991 年发展到 1.5 版，现在有两个版本，1.5 和 2.0。因为这个操作系统的初衷只是作为一个用来学习的模型，并不是一个实用的系统，所以他的功能很简单，体积也很小，并且以后也没有进行进一步的开发和扩充。这为的是能够让学生


Linus Benedict Torvalds

在一学期内能学完整个系统。那时候 Minix 在大学中用于教学是免费的，但是用于其他用途是需要给钱的，不过现在已经彻底免费了。它作为一个操作系统，其实并不算优秀，但它是一个源代码完全开放的操作系统，这使得有理想有志向有报复的黑客们，第一次能够完整的阅读到一个操作系统的全部代码。



所以呢，Linus 他们学校的计算机上装的也是这个专门用于教学的 Minix 系统。虽然适合拿来学习，不过系统本身并不强大。这要是别人也还罢了，可是 linus 同学有个最大的爱好，就是虐待计算机。他热衷于测试计算机的能力和限制，整天研究怎么让计算机按照自己的想法去干活，怎么发挥计算机最大的性能，一定要把可怜的机器累得精疲力尽呼哧带喘直到电容爆浆，吐血身亡才算罢休。因此很快的，这个教学用的操作系统就已经不能满足 Linus 大侠的欲望了。可是似乎也没有更好的选择，上面说过了，Unix 奇贵无比，DOS 又不够优秀，而且无论 Unix 还是 DOS，他们的代码都是不开放的，只能拿来用，没法拿来折腾。于是象其他牛人一样，Linus 自己动手了。（当想要的东西不存在就自己动手创造，这充分说明他有成为大牛的潜质。）

今天我们都知道，Linus 从那时起开始了一个事业，一个神话，但在当时，他并没有想那么多，只是为了学习 Intel386 体系结构保护模式运行方式下的编程技术。他并不知道自己即将创造的是一个在世界范围广泛使用的系统，而只觉得是自己一时的异想天开。因此，一开始他把自己写的这个操作系统命名为 FREAX。就此开始了这个“异想天开”操作系统的编写。大约 1991 年 4 月份的时候，就编写出了第一个可以运行的版本——0.00 版。这个版本可以启动，运行两个进程，分别在屏幕上打印出 AAA，和 BBB，然后……就没了。虽然连句整话都不会说，不过这是一个好的开始，至少能启动了。

如果他就这么干下去，估计到今天只会有两种结果：1. 成家立业后的 Linus 经常指着他电脑 C：盘里面的一个文件夹对来访的朋友说：看，我那时候还写过一个 Freax 系统。2. Linus 为完成 Freax 系统挑灯夜战，最终累得吐血身亡，永远活在我们心中。总之是不会有 linux 这个东西了，因为一个人的力量是有限的，有道是人多力量大，众人拾柴火焰高，多个铃铛多个响，一个篱笆三个桩，三个臭皮匠还顶个诸葛亮……铛！哎呦～ 好吧，就说这么多了。总之，Linus 没有独自在家闭门造车，而是让他的操作系统和互联网，亲密接触了。

“Hello everybody out there using minix——I' m doing a (free) operating system”这是他当年在 comp.os.minix 上发布的消息，告诉大家，他正在写一个操作系统。并且，他还把他写的“异想天开”操作系统的代码上传到 ftp.funet.fi 的服务器上让大家下载，以便交流心得，共同学习。这就相当于你跑到网站上发帖子说：我研究出一种萝卜炖牛腩的方法，主料是啥啥啥，配料是啥啥啥，怎么怎么炖，大家都试试吧！（对不起，我又饿了）于是很多有兴趣的人就来尝

Linus 炖的牛腩，哦不对，是尝试 Linus 写的系统。不过当时那个服务器的管理员 Ari Lemke 看着这个异想天开的名字就不顺眼，想想，既然是 Linus 写的操作系统，又是类 Unix 的，干脆，叫 Linux 吧。

Linux 被公布在网上之后，引来大家纷纷的路过和围观，很多人觉得这个东西挺有意思，不过第一个对外发布的 0.01 版 linux 还有很多的不完善（这简直是一定的）。这里先要说一个概念，linux 是什么？确切的说，狭义的讲，linux 只是一个操作系统的内核，他只是各位的 Ubuntu 系统里面 /boot/ 目录下的那个内核文件 vmlinuz-x.x.xx-xx-generic。就好比汽车，linux 只是一个引擎，只是大家普遍的把装了 linux 这种引擎的汽车叫做 linux 汽车。那么既然 linux 只是一个内核，要想工作就还需要很多周边的支持，比如文件系统，比如一个命令行程序，比如一些基本的软件。

首先就要感谢 Richard Stallman 大牛创建的 GNU 计划，这使得 Linux 不必去从头开始开发那些最基本的软件和命令，而是直接利用 GNU 计划中的那些优秀的开源软件——前面说过了，那时候 GNU 系统除了内核以外，已经比较完善了。

有了基本的软件之后，还需要个文件系统。由于当初 linus 大侠是在 minix 系统上开发的，所以最一开始 linux 用的文件系统是借用 minix 的文件系统。可老借别人的总不是个事，还是应该有自己的文件系统，要不然你怎么好意思跟别的操作系统打招呼？这时候，来了个牛人叫 Theodore Ts'o。

Theodore Ts'o（曹予德，华裔），1990 年毕业于美国 MIT 大学计算机科学专业。他爱好广泛，喜欢烹饪，骑车，无线电报，还有折腾电脑（这都不挨着啊～），当然这不是我们的重点。他看到 linux 觉得很有意思，于是怀着极大的热情为 linux 提供了邮件列表服务以便大家一起讨论问题，后来还提供了 ftp 站点来共享 linux 的代码，并且一直用到现在。除此之外，技术上，他编写了 linux0.10 内核中的虚拟磁盘驱动程序和内存分配程序。在感觉到 linux 缺少一个自己的文件系统后，他提出并实现了 ext2 文件系统，并且 ext 系的文件系统一直都成为了 linux 世界中事实上的标准，任何一个发行版都会默认支持。现在已经发展到了 ext4 了。

另一位牛人，一个英国人——Alan Cox。他工作于英国威尔士斯旺西大学，特别爱玩电脑游戏（又一个玩游戏的，可见玩游戏也不是坏事），尤其是网游（你看你看，还是网游），不过那时候的网游不像现在这样华丽，那时候是字符界面的，能想象嘛？字符界面的网游！那种叫做 MUD——Multi-User Dungeon or Dimension。玩 MUD 当然就得有计算机啊，就得有网啊，所以 Alan Cox 就开始逐渐的对计算机和网络产生了兴趣。为了提高电脑运行游戏的速度以及网络传输的速度，他开始接触各种操作系统，为自己选择一个满意的游戏平台，争取榨干电脑的每一个指

令周期。经过仔细考虑，他买了一台 386SX 电脑，并且装了 Linux0.11 版的系统。这主要是因为预算比较紧张，即使 minix 他也买不起。（重复一下，那时候 minix 用于教学是免费的，但是其他用途要收费，包括个人用。）于是他开始使用 linux，进而学习其源代码，并对 linux 产生了兴趣，尤其是网络方面相关的代码。（整天琢磨怎么榨干他家那点带宽）在 Linux0.95 版之后，他开始为 linux 系统编写补丁程序，以后逐渐加入 Linux 的开发队伍，并成为维护 linux 内核源代码的主要人物之一。那个有点软的公司还曾经邀请他加盟，被他有点硬的拒绝了。

再有一位，Michael K. Johnson，他是著名的 linux 文档计划的发起者之一，写了《内核骇客手册》一书，曾经在 Linux Journel 工作，现在在著名的商业发行版 RedHat 的公司工作。

当然除了这些大牛，还有更多的大牛，中牛，小牛，牛犊，牛杂，牛尾，肥牛……（唉，又饿了）他们都为 linux 的发展做出了自己的贡献。他们来自不同的国家，从事不同的职业，他们甚至从未见过面，但是他们为了一个共同的目标，通过网络，一起合作，利用自己的业余时间，义务的帮助 linux 成长，才有今天这个可以合法免费使用的操作系统。这是什么精神？这就是软件国际共产主义的精神！（好吧，这个词是我造的）

2008 Linux Kernel Summit 全家福

这之后，Linux 的发展可以用“一发不可收”拾形容。很多的商业公司和民间组织都纷纷看好这个系统，纷纷加入了 Linux 的阵营，各种各样发行版满足着各种 Linux 爱好者的需求。比如做的比较大的 RedHat，浪漫的 Mandriva，扎实稳健的 Debian，灵活的 Slackware，极端的 Gentoo，以及我们这个故事的主角，从 Debian 的基础上改头换面而来的 Linux 界的新星——Ubuntu。