

# UNIX OS 简史

作者：网络农夫 (黄天彦)

整理：Livelybear

出处：[http://www.europa.idv.tw/UNIX-history/UNIX\\_History-01.html](http://www.europa.idv.tw/UNIX-history/UNIX_History-01.html)

## 先前的一个理想

UNIX 系统自 1969 年 [Ken Thompson](#) 与 [Dennis Ritchie](#) 在美国贝尔电话实验室 (Bell Telephone Laboratories) 发展出雏形至今，已历经近 30 来年。而 "UNIX" 这个字典上查不到其原意的怪字，其实是戏谑 MULTICS (MULTiplexed Information and Computing System) 操作系统的大而无当所产生的谐音字。

在 1957 年 10 月，前苏联发射了第一枚人造卫星，此举让当时的美国总统艾森豪威尔决定投下巨额的经费用以支持及发展科学，美国高等研究计划署 (ARPA, Advanced Research Projects Agency) 便是在这个时空下设立了，该单位负责推动系统发展等相关计划，成为当时美国电子计算机发展的重要推手。



Dennis Ritchie



Ken Thompson



J.C.R. Licklider

1960 年代是大型计算机的发展年代，当时的麻省理工学院因最先实现了兼容分时系统 (CTSS, Compatible Time-Sharing System)，在电子计算机领域享有相当崇高的地位。1963 年，麻省理工的里克莱德 (J. C. R. Licklider, 1915~1990) 推动了 MAC 计划，MAC 以 IBM 的大型计算机做为主体，连接了将近 160 台终端机，这些终端机就四散在学区以及教职员的家中，可以让 30 位使用者同时共享计算机资源。这项计划到了 1965 年便不堪负荷，于是麻省理工便决定开发更大型的分时计算机系统。新的计划便是 -- MULTICS。一个计算机史上最为庞大的分时计算机系统，企图连接 1000 部终端机，支持 300 位使用者同时上线的分时计算机系统。她面临的是，操作系统的分时观念还在各学术与研究机构探索成形中，计算机硬件亦需重新设计的双重挑战。

当时，麻省理工原本找 IBM 来配合这项计划，但 IBM 正忙着应付自己的问题而无意配合 MULTICS 计划。此时，通用电子公司 (General Electric Company) 也就是奇异公司正好在发展自己的大型主机，见机不可失，便极力邀请麻省理工参予她们的 GE 645 大型主机的规格制定。有了奇异热心主动的计算机硬件配合，麻省理工找上的不能贩售计算机却人才济济的贝尔电话实验室来负责承包软件工程。于是乎，MULTICS 的计划便在 1965 年由麻省理工学院、奇异公司及贝尔电话实验室这三个成员开始共同发展。

1969 年，MULTICS 计划在历经四年的奋战后，仍旧未达到原先规划设计的理想，贝尔电话实验室决定退出计划。功能未达原始设计理想的 MULTICS 还是安装在奇异公司的 GE 645 大型计算机上供麻省理工使用。奇异公司在该计划草草结束后不到一年便完全淡出大型计算机市场。日后，MULTICS 计划被嘲解为 Many Unnecessarily Large Table In Core

Simultaneously。

农夫我个人认为，MULTICS 计划诞生在大型计算机将开始鼎沸的 1965 年，夭折于大型计算机最为辉煌的 1969 年。她如果适时在 1960 年代末期成功的话，绝对可以助长当时已经普遍被计算机权威人士视为理想的『计算机公用事业』，至少可以让大型计算机的发展与资源集中的应用模式就不至于会在 1970 年代初期就迅速萎缩。因为 MULTICS 计划如果成功，至少能让当时的大型计算机的应用规模大上 10 倍左右。然而，MULTICS 计划失败了。她严重地打击了当时依赖大型计算机主机的计算机公用事业业者在发展上的信心。更由于没有相似的计划后继进行，使得集中式的大型计算机主机没有明显的 使用效能提升，而加速催化计算机工业的转变，以寻找新的道路。另一方面，MULTICS 计划失败的经验亦让当时参与该计划的软件工程师们得到相当宝贵的经验与正面的影响。

几年后，就在 AT&T，MULTICS 计划这个不同凡响的失败换来的一个不同凡响的成功。一个戏谑她的名字诞生了 .... UNIX。

## 一个游戏的开始



1969 年贝尔实验室的计算机科学研究中心 ([Computing Science Research Center](#)) 成员退出 MULTICS 计划的同时，贝尔实验室本身其实也没有一套完善便利的交谈式计算机服务环境。在其中不少工程师们也正为了改善程序设计环境努力着，Ken Thompson、Dennis Ritchie 和其同事们在当时草拟一个新的档案系统架构，这个档案系统也就是早期的 UNIX 操作系统的档案系统的前身。当时的 Ken Thompson 忙着使用 Fortran 语言将原本在 Multics 系统中开发的 game 叫 "Space Travel" (太空旅游) 转移到 GECOS System 上开发。当时 GECOS System 大型计算机的 CPU Time 相当昂贵 (一秒要 75 块美金)，同时控制 "spaceship" (宇宙飞船) 的效果不甚理想，于是 Ken Thompson 不得不寻找替代的开发环境。Thompson 看上了一台很少被人使用的 Digital Equipment Corporation PDP-7 迷你计算机，当时

PDP-7 使用的是 Graphic-11 显示器，具有不错的图形处理能力。于是 Ken Thompson 便与 Dennis Ritchie 联手将程序设计转移到 PDP-7 型计算机上。Ken Thompson 在移转工作环境的同时为了得到较好的发展环境，便与 Dennis Ritchie 共同动手设计一套包含 File System、Process Subsystem 及一小组 Utility 的操作系统，当时这套系统仅能支持 2 个使用者使用。由于贝尔实验室对于 MULTICS 计划失败的阴霾还未消散，[Brian Kernighan](#) 这位仁兄开玩笑地戏称这套新的操作系统为 UNiplexed Information and Computing System，缩写为 UNICS，之后大家取谐音便叫她为 "UNIX"，没想到这个开玩笑的名字会被人叫到今天。



## 初期的自由发展

事实上该套 "UNIX" 系统在当时仅是私下的被使用，也并没有得到多大的重视，一直到 1971 年的一个正式的计划，UNIX 才正式被搬上台面。

1970 年，当时贝尔实验室的专利部门（Patent department）缺乏一套文书处理系统，为了设计开发的需要，于是买了一台 PDP-11 计算机。当时 PDP-11 计算机的交机过程并不顺利，处理器先到，硬盘则多等了好几个月。当 PDP-11 一切准备妥当后，他们便将 UNIX 移植到拥有 512K bytes 硬盘的 PDP-11/20 型计算机上，并在此系统之下开发了一套文书处理工具。而这套工具便是后来 nroff / troff 的前身。那时的 UNIX 提供 16K bytes 给系统、8K bytes 给使用程序，档案最大的极限是 64K bytes。而此套含有文书处理工具的系统，也正式获得贝尔实验室的专利部门采用，系统名称并被编为 "First Edition"。在 UNIX 移植成功后 Thompson 用 B 语言为它添加了 Fortran Compiler，但因为 B 语言属于一种解译语言（interpretive language），执行成效并不是很好，于是 Ritchie 又将它 -- Compiler 发展成可产生机器码、允许定义数据形态及结构，Ritchie 称它为 C 语言。1973 年并以 C 语言改写全部 UNIX 原始程序，UNIX 于是首度出现正式版本--V5 (第五版)。

此时的 UNIX 慢慢地在贝尔实验室内部蔓延开来，装机数也变成了 25 部之多。由于当时的贝尔实验室实际上是掌控在美国电信电话公司（AT&T）及其子公司西方电器公司的手上，实验室主要是负责研究改进西方电器公司制造的和美国电信电话公司在贝尔系统中使用的电信设备。同时根据军方合同，从事与国防有关的研究与改进的工作。而 AT&T 本身由于有反托拉斯法的限制并不能从事于任何有关计算机方面的销售，所以 AT&T 的主管阶层们对于当时 UNIX 的发展并没有太多的支持，因而当时贝尔实验室内部对于 UNIX 的发展并不是相当在意也无意于将之推广。不过为了应付实验室内各部门日益增加的 UNIX 使用者与相关技术支持需求，还是成立了 UNIX System Group（简称 USG）。但该组织也仅只是提供技术上的支持，并未赋予继续发展的任务。所以当时的 UNIX 发展，全靠 AT&T 的工程师们的努力。这段期间 UNIX 的发展完全没有组织及系统性可言，而玩家尽是一些工程师们，于是乎种下了 UNIX 日后较难以被一般人所接受的命运。

## 走出贝尔实验室

1974 年 Thompson 与 Ritchie 共同在 [Communications of the ACM](#) 发表了一篇 UNIX 论文 "[UNIX Time-Sharing System](#)" 得到相当大的回响。1975 年 UNIX 发表第六版（V6），其提供的强大功能更胜过当时昂贵大计算机的操作系统，其最大特点是以高级语言写成，仅需要做少部份程序的修改便可移植到不同的计算机平台上。UNIX V6 版本并附有完整的程序原始码在 1976 年正式从贝尔实验室内部传播到各大学及研究机构，UC Berkeley 也就是依据这个版本开始研究并加以发展，并在 1977 年发表 1 BSD（1st Berkeley Software Distribution）版本的 UNIX OS，其后续的发展更为 UNIX OS 贡献良多且影响深远，此点稍后再为你说明。同年 UNIX 因它提供良好程序发展环境、网络传输服务与及时服务（Real-Time Services），而广得各电话公司采用。Interactive System Corporation 更因 Value Added Reseller (VAR) 运用 UNIX 来强化办公室自动化环境，成为第一家应用 UNIX 操作系统的公司。此年 UNIX 亦被修改并第一次装到 Interdata 8/32 型计算机上。这也是 UNIX 操作系统首次安装在非 PDP 型的计算机上。自此 UNIX 系统开始被移植改装到各型微处理机及新计算机上。

## 一个稳定的基石

1978 年 UNIX 发表对今日影响最重大的 UNIX 第七版（UNIX Time-Sharing System, Seventh Edition）也就是 V7。此版本包含 Fortran 77 compiler、Shell（只有 Bourne Shell）、文件处理工具（nroff/troff、roff、MS mocomp 等）、UNIX-to-UNIX-file-Copy（用来支持两台 UNIX 机器间的档案传输）、数据处理工具（AWK、SED 等强悍的工具）、除错工具



(ADB)、程序发展工具(MAKE)、Lexical analyzer generator (LEX、YACC 等)、简单的绘图工具、并支持 C 语言及 LINT verifier, 主要执行于 PDP-11 及 Interdata 8/32 型计算机上。在当时那个年代来说其系统的架构与功能已经是相当的完备的了。Bourne Shell 的原作者称她为 "improvement over all preceding and following Unices", 在今日也有人称这个版本是 "last true Unix"。由此可见 V7 在 UNIX 发展里程上的扮演了相当重要的基石角色。

当农夫我还在联华电子工作时, 王林正同事是我工作上的导师兼撞球球友。在他出国留学前送我一本 V7 的使用手册。这本书对我的帮助很大。虽已事隔多年, 但指导之恩永生难忘。在此表示对王林正前辈的由衷感谢。

事实上, V7 的使用手册分成两个部分, Volume 1 中有一篇简短的系统简介以及所有指令的使用说明。Volume 2 则是比较深入的单元介绍, 此部份原本分为 2a, 2b 两部份, 儒林出版社发行时则将之合并为一册。我手上有的便是这一本手册。

前些日子, 我在网络上闲逛, 意外地在 Bell Laboratories 发现了 V7 的使用手册。总算宿愿得偿, 看到了 Volume 1。我个人认为, V7 的使用手册, 事实上写得相当好, 比起市面上某些书籍介绍的还要完整。有心学好 UNIX 的人, 最好能花点时间看看 Volume 2 的部份, 该份文件您可以在到以下的网址找到。 <http://plan9.bell-labs.com/7thEdMan/bswv7.html>

在当时 DEC 公司推出了一款 32-bit supermini 主机 -- VAX, 搭配的 VAX 的操作系统叫做 VMS。这款迷你级计算机的硬件无可挑剔 (直到今日她的稳定度仍是被诸多老一辈的系统管理者所赞许的), 但 DEC 对 VMS 操作系统的支持性却让贝尔实验室的工程师们宁愿使用 UNIX OS。而这项工作则是由 John Reiser 和 Tom London 所共同完成。他们以 V7 为基础转移 UNIX OS 到 VAX 计算机上使用。这个版本被称为 UNIX V32。同时为了转移的方便性, 他们把 32-bit 的 VAX 当成是大一点的 PDP-11(因为 DEC 的 PDP-11 型计算机是 16-bit), 同时为了执行的效率, V32 放弃使用 VAX 硬件提供的一项 paging 功能(DEC 的 VMS OS 有支持 paging 功能, 也由于 V32 舍弃这项功能, 所以 V32 没有虚拟内存的功能)。即使是如此, V32 支持的地址已高达 4Gb。就这样没有支持 paging 功能的 V32 开始被广泛的安装在 VAX 的机器上运作。



DEC 则是在 1984 年左右推出来自己的 UNIX OS, 叫做 ULTRIX。

## 一个重要的延续及发展 -- BSD UNIX

时间回到 1973 年 11 月, Ken Thompson 和 Dennis Ritchie 在印第安纳 [Purdue 大学](#) 的一场操作系统原理的座谈会。会场上、坐着一位 [柏克莱大学 \(U.C. Berkeley\)](#) 教授, 名字叫 Bob Fabry。当天的 K&R 所发表的 UNIX 立刻引发 Bob Fabry 的极度兴趣。当时的柏克莱还是处在使用大型计算机主机、批次执行程序阶段, 并没有像 UNIX 这样的交谈式作业环境。会后, 他便决定将 UNIX 带回柏克莱。



于是柏克莱的计算器科学、数学与统计三个系所合买的一台 PDP-11/45, 准备用来迎接 UNIX。1974 年 1 月, Bell Labs 寄来了一卷 V4 的磁带, 学生 Keith Standiford 便开始进行安装 V4 的工作。安装时 Standiford 碰到了问题, 便转向 Bell Labs 求援。人在新泽西州的 Thompson 便透过柏克莱这端速度只有 300-baud 的调制解调器在线进行侦错。

在 UNIX 的发展史上，这是 Bell Labs 与 柏克莱的第一次接触。

完成除错后，V4 便顺利地柏克莱这台新买的 PDP-11/45 计算机上工作了。当时这台是三个系所合买的，计算器科学好不容易装上了 UNIX，却碰到数学与统计系所要使用 DEC's RSTS system，所以在一阵协调后，UNIX 与 DEC's RSTS system 以 8:16 小时的比例分配，供三个系所轮流使用。一段时日后，具交谈式功能的 UNIX 在效能上的表现得到绝大多数学生们喜爱，纷纷将自己的计划转向 UNIX 的时段。而一天占了 16 个小时的批处理时段却乏人问津。

当时 Eugene Wong 与 Michael Stonebraker 教授，看上了 UNIX 提供的便利性，便打算将他们的 INGRES 数据库计划重原先批处理的计算机环境转移到 UNIX 系统上面。在 1974 年，他们为这执行计划添购了一台新的 PDP-11/40 计算机，上面安装了 V5。这个计划也就是柏克莱的第一个将作业环境转移到 UNIX 的案子。UNIX 作业环境的需求，在柏克莱迅速地成长。为了应付需求，Michael Stonebraker 与 Bob Fabry 教授决定再申请购买两台 PDP-11/45。1975 年初，DEC 推出 PDP-11/70，价格差不多等于两台 PDP-11/45，但功能强过 PDP-11/45，所以他们便决定改购买一台 PDP-11/70。

这台机器引来了 Ken Thompson、碰上 Bill Joy 以及日后产生了 1BSD。她就宛如是一块 UNIX 史上的地标，沿袭自 Bell Labs，竖立在柏克莱，承先启后并开创新局。农夫个人认为，她应该被供在博物馆。



Bill Joy

当这台机器在 1975 年终运达柏克莱时；同一时间，Thompson 受邀回母校(柏克莱)当客座教授，科目就是 UNIX。Thompson 在校期间与 Jeff Schriebman 和 Bob Kridle 一起动手将新版的 V6 安装在 PDP-11/70。

1975 年，一位密执安州大学的毕业生来到了柏克莱，他的名字就是 Bill Joy。当时 Joy 和同学 Chuck Haley (tar 就是他写的)喜欢一起泡在计算机房里面，Thompson 也时常插上一脚。他们成功地改善了 Pascal 的解译与侦错的能力，同时还提升了解译与执行的速度。另外换装上 ADM-3 的屏幕后，他们觉得 ed 文字编辑指令并不合用；于是根据另外一个相似的 em 指令，发展了自己的觉得满意的文字编辑工具，也就是指令 ex。

1976 年夏天，Thompson 结束了他的休假回到 Bell Labs。此时的 Joy 和 Haley 已经开始着手探索 UNIX kernel，甚至还做了一些修改。1977 年初，Joy 制作了一卷磁带，上头写着 "Berkeley Software Distribution."，这就是 1BSD。其中包含新的 Pascal compiler 与 ex 编辑器。

次年，来了几台新屏幕 -- ADM-3a，这种屏幕支持光标地址显示，Joy 在这种屏幕上完成了有人爱不释手；有人恨之入骨的文字编辑器 -- vi。接着不久，Joy 便发现一个问题，老旧的屏幕装备，还是会被用在其它的计算机上。为了支持上的方便，Joy 针对此现象设计了一个接口，用来管理、支持不同的屏幕装备。这个接口就是现在的 termcap。1978 年中，包含了功能加强的 Pascal 与 vi 及 termcap 的 "Second Berkeley Software Distribution," 也就是 2BSD，迅速的取代了原先版本。1979 年，至少有 75 部 PDP-11 的机器上安装 2BSD 在运作着。自此在 DEC PDP-11 系列上执行的 BSD 版本便一直以 2.xBSD 作为识别。由于 PDP-11 计算机实在相当长寿，持续到今日农夫我仍然在网络上发现过关于 PDP 计算机的网站。似乎到今日它们仍旧在某些地方默默地工作着。2.xBSD 最近的一次改版是在 1987 年，使用 4.3 BSD 为主架构改写，版本定为 2.10 BSD。

在 BSD UNIX 中登场的重要功能当中，有一个直到今日仍然叫人又爱又恨的指令 -- vi。我接触过不少学习 UNIX OS 的人，大部分的人对 vi 的使用与掌握都不算顺手，其中恨死这个指令的也大有人在，前些日子农夫我还看到某个网站公开讨论起 vi 是否阻碍了 UNIX 的

发展？实在夸张了一点！

Bill Joy 多次公开地说，他要是知道 vi 会如此受"欢迎"的话，他宁愿当初没有写 vi 这只程序。不过 Bill Joy 也说过，当时他原本还想加入一项 Multiple Windows in vi 的功能，不过当他在写这部分程序的时候，磁带机坏了，所以 Bill 只好在没有备份的情况下继续工作，想不到"屋漏偏逢连夜雨"，程序写到一半，他使用的硬盘也跟着挂了。在无可挽救又没有备份磁带的情况下，Bill 宣告放弃为 vi 增加 Multiple Windows 这项功能。事后 Bill 为前一版的 vi 写好使用说明后就继续作其它的事。所以 vi 就长成今天那付德性。农夫我认为这或许是福不是祸！搞不好当初要是连 Multiple Windows 这项功能一起发表的话，上头的图可能就是遗照了。

当时有位 Richard Fateman 教授，原先使用一台 PDP-10 上进行着他的 Macsyma 研究计划。但他需要更大的内存地址来执行程序，所以在 1978 年初，他看上了当时迪吉多新发表的 VAX-11/780。好不容易，他联合了其它的部门才凑足购买 VAX 的经费。刚开始时，机器原本安装的是 VMS 操作系统。不过别的成员要执行 UNIX 操作系统，于是 Fateman 安装上了 V32。但问题来了，V32 并不支持虚拟内存，Fateman 便找上了 Domenico Ferrari 教授，希望他与他的研究小组能为 UNIX 加上这项功能。当时一位学生叫 Ozalp Babaoglu，他想到了一些解决的方法似乎可行，但因为牵涉到 VAX 硬件与 UNIX kernel 的问题，于是他找上了 Joy 帮忙。就在只有一台 VAX 的状况下，他们努力奋战着。1979 年 1 月，在 VAX 上支持虚拟内存的 UNIX 版本终于诞生，V32 从此走入历史。紧接着 Peter Kessler 与 Marshall Kirk McKusick 为他加上了 Pascal；Joy 则动手将 2BSD 上的 ex、vi、C shell 等工具转移了过来。这个版本就是 3BSD。一个首次支持虚拟内存、demand paging 和 page replacement 的 UNIX OS。

## UNIX 与 DARPA 交会

1970 年代末，美国国防部先进研究计划机构 (Defense Advanced Research Projects Agency -- 简称 DARPA) 正在为 AI(Artificial Intelligence), VLSI 及计算器视觉等研究(vision research)找寻一个可共通作业的计算机环境。硬件方面的首选是迪吉多的 VAX 主机。配合的操作系统是 VMS。这样的组合因拥有相当接近 DARPA 需求的功能被列入优先的考量，但在 DARPA 与 DEC 商谈对于 VMS 的支持事宜之后，DARPA 并没有得到满意的答案。这迫使他们考虑朝向 UNIX 发展。但当时 UNIX OS(指的就是32V) 搭配 VAX，最大的缺憾就是没有支持虚拟内存；但此时已经有人克服了。

当时，Bob Fabry 教授写了一份建议书给 DARPA，建议他们以柏克莱支持虚拟内存的 3BSD 为基础，发展成为计划所需。这份企划书引起了 DARPA 的高度兴趣。随后 3BSD 也实际获得了 DARPA 相关计划成员们的美好风评，也因此最后柏克莱大学打败了卡奈基梅隆大学与 [BBN\(Bolt Baranek & Newman, Inc.\)](#)，让 Bob Fabry 成功地获得了 DARPA 的资助合约。这份合约开始于 1980 年 4 月，为期 18 月。此后的 DARPA 便以 UNIX OS 为标准操作系统。Bob Fabry 教授在取得 DARPA 合约后，依约成立了一个支持机构，也就是 Computer Systems Research Group 简称 CSRG。Bob Fabry 找上了 Bill Joy 来负责软件开发。Joy 迅速地以先前的 3BSD 为基础，整合新的功能。如 Job Control(作者是 Jim Kulp)、auto reboot、1K block file system。同时也整合入 Pascal compiler、Franz Lisp system、enhanced mail handling system。这就是在 1980 年所发表的 4BSD。没多久她便被安装在将近 500 台 VAX 上。

DARPA 采用了这个版本作为当时 DARPA 的标准 UNIX 操作系统。

树大招风，当时，有位在 Stanford Research Institute 的仁兄叫 David Kashtan，写了一份关于 VMS 与 BSD UNIX 在 VAX 上的执行效率评估。该份报告指出 BSD UNIX 在效率



上不如 VMS 来的好。Joy 知道这件事之后，花了不到一个星期的时间，重新调整 UNIX kernal。然后也写了一份报告，证明他们的 BSD 在 VAX 上要比 VMS 优越多多。1981 年 6 月，这个 Joy 调整过的系统，加上了 Robert Elz 写的 auto configuration，以 4.1BSD 的版本发表了。

当时的 DARPA 对柏克莱 4.1BSD 的表现相当满意，于是续签了两年的新约，金额更是先前合约的 5 倍。其中有一半的金额用在资助柏克莱继续发展 BSD UNIX。钱多的相对代价就是要求高。当时，DARPA 对 UNIX 的期望开出了明确的目标：更迅速、更有效率的档案系统、支持程序可执行地址达 multi-gigabyte、提供弹性的解译沟通能力、具整合支持网络能力。在此同时，为了达到计划的目标，DARPA 成立的一个指导委员会；主要的成员有柏克莱的 Bob Fabry, Bill Joy, Sam Leffler、BBN 公司的 Alan Nemeth and Rob Gurwitz、贝尔实验室的 Dennis Ritchie、史丹佛大学的 Keith Lantz、卡内基·梅伦大学 Rick Rashid、麻省理工学院 Bert Halstead、信息科学协会 Dan Lynch、DARPA 的 Duane Adams and Bob Baker 以及加州·洛杉矶大学的 Jerry Popek。

不久，Joy 便开始整合早先 BBN 的 Rob Gurwitz 所发表的 TCP/IP protocols，不过他对 BBN 这些程序的执行效率并不满意，于是 Joy 与 Sam Leffler 重新写的一版自己的程序。另外，并加入了一些支持网络的工具 rcp, rsh, rlogin, rwho。他们称她为 4.1aBSD，这个版本并没有正式发表，在 1982 年 4 月开始供内部使用。虽是如此，在 4.2BSD 未正式发表之前，她还是繁殖的到处都是。6 月，4.1aBSD kernal 加上了新完成的档案系统，版本更新为 4.1bBSD。

rcp, rsh, rlogin, rwho 这群指令。因安全机制上的理由，逐渐被另一群新的指令群所取代，新的指令群叫 SSH (Secure Shell)。SSH 相关网址(<http://www.ssh.org>)。

1982 年的春季末，已厌倦了在柏克莱环境的 Bill Joy，答应受邀加入当年刚创办的 Sun Microsystems, Inc.，成为 SUN 的第四号创办人。那年的整个夏季他就在两地奔走。之后他对修改中的弹性解译沟通机制及改写 UNIX kernal 到一个段落之后，由 Leffler 接手了他的工作。由于合约期限的因素，Leffler 在 1983 年 4 月发表了 4.1cBSD，提供给参予 DARPA 各项相关计划的成员试用。6 月，DARPA 的指导委员会第二次会议召开，验收与检讨最新版的 BSD 成果。继续整合 UNIX 系统的 Leffler，在 1983 年 8 月，发表了 4.2BSD。她达到了 DARPA 的预定的需求；足以应付 CAD/CAM 影像处理与 AI 研究的高速的档案系统及扩展强化的虚拟内存功能；提供能分散处理的解译沟通机制；支持 56-Kbit 的 ARPA Internet 网络连结，以及 10-Mbit/s Ethernet 的局域网络；还有经过重组架构已模块化的 kernal code，提供更有效率的计算机平台移植。

SUN 以生产 RISC 架构的工作站计算机为主，使用的正是以 BSD 为基础所的 UNIX OS。在当时以不逊色于大型计算机的多人多任务、具网络沟通功能的 UNIX OS、加上价格低廉的硬件（相对于 mini 级计算机而言），广获得工程界的青睐，而 mini 级大计算机的命运自此注定开始逐渐式微。计算机软件的应用因为有了网络于是也开始朝向 Client-Server 的架构发展。

1982 年，SUN 有了自己的操作系统 -- SunOS 1.0 -- 承袭自 4.1BSD。一直到 1990 年 11 月，发表 SunOS 4.1.1 版同时冠上 Solaris 1.0 时，SUN 才算开始向 System V 版本靠拢。SunOS 4.1.1 可算是以 BSD 为主体再附上 System V 工具的 UNIX 混血儿。但这其实是个商业考量的过渡性做法(后文会加以说明)。而 SunOS 4.1.x 版的字眼也仅延续到 1994 年的 SunOS 4.1.4 为止，她后继的版本是 Solaris 1.3。真正延续到今日的 Solaris 版本，则是始于 1992 年 7 月的 Solaris 2.0(SUN OS 5.0)。

在商业有所成就的 SUN Microsystems 对 UNIX OS 的发展倒也做了些重大贡献；如 1984 年发表的 NFS(Network File System)与其后在 1986 年发表的 PC-NFS。

## 商业化的不平坦历程 -- UNIX 版本的战争

UNIX 商业化实质上即意味着将产生各种独立化的 UNIX 版本，这点大概是最显而易见的事实。如果以商品要具备独特性与独占性的利益来做考量的话，其实一点也不意外。因此 UNIX 开始衍生的相当多的版本。这种现象，对使用者以开发应用程序的厂商而言，已经造成了某成程度上困惑。然而，一种无所适从的无力感其实才刚开始。

1984 年 1 月 1 日，AT&T 这个拥有 1495 亿美元资产、1,009,000 位员工的庞大巨兽，终于被格林法官 (Harold H. Greene) 以反托拉斯法 (antitrust) 强制拆解成七家 RBOCs (Regional Bell Operating Companies)。AT&T 也因而在一夕间解体成为区域性网络公司，从此失去了长途电话的垄断性地位。这种时空的转变让 AT&T 对 UNIX 的态度有了 180 度的转变(其实，农夫我指的是收费的态度)。

先前已经提过 70 年代初期的 AT&T，已经在长途电话市场上占有绝对垄断的优势，因而被美国政府的限制不得涉足与从事计算机与其它行业，也正因而造就了 UNIX 发展初期的自由开放。直到 1979 年，AT&T 才宣布要将 UNIX 商业化的计划。1981 年 11 月，AT&T 属下的 USG 发表了 System III。次年又更新为 System IV。稍后于 1983 年，AT&T 将 CRG, USG 合并成立了 UNIX System Development Lab. 一般简称为 USL，从其名称就不难清楚她将要扮演的角色。该年 System V 上市了。此时 AT&T 发觉每次版本更新都得花不少宣传费，实在不划算，所以决定在 System V 以后，名字就不再做变动了。1984 年，System V Release 2 发表，简称为 SVR2。在这个版本中，才终于看到来自 BSD 版本的 Virtual memory 功能，农夫我不得不惊叹 AT&T 的稳健作风。SVR3 则是到了 1986 年才发表，随后 1987 年又发表了 SVR3.2。

1987 年，在工作站市场上已占有一席之地的 SUN，找上了 AT&T，打算将 System V 与 BSD 这两大版本归为一统。1988 年初，双方更签订了合作合约，AT&T 取得 SUN 的一席董事，同时亦有权买下 SUN 百分之二十的股份。这项合作计划，原本有机会整合当时版本纷乱的 UNIX OS。但那是理想。实际上这个计划反而让 UNIX 族群里的其它成员恐慌万分，特别是 IBM、DEC、HP 这几个产业龙头。为了抵制这项行动，他们组织了一个反对联盟。因此「开放软件基金会」也就是 Open Software Foundation 简称 OSF 在 1988 年正式诞生；成员除了前面的三巨头外，尚有多达三十几家计算机硬件制造厂商与系统咨询顾问公司，也相继以行动投入到此反对的行列中。然而 AT&T 与 SUN 也不示弱地组织了 UNIX International，也就是 UNIX 国际公司，成员数量虽然不比 OSF 阵营来的多，但如果她是 Intel、Toshiba、Unisys、Motorola、Fujitsu，这几个大块头，那也是很够看头的。

企业自身的利益在现实世界里始终是以个体的考量为优先，所以这两大阵营始终没能再达成任何共识，就连当时所制定的 UNIX 统一标准规格，严格来说也从不曾被实现过。这种企业利益上的冲突与矛盾其实也存在于同一个阵营中不同的成员之间。两大阵营对峙，可以说是 UNIX 有史以来最重大的产业冲突事件。由于商业利益的政治考量大过技术问题的考量，也因此奠定了 UNIX 将继续分裂下去的命运。AT&T 在 1989 年发表了 SVR4，SUN 在日后也将她的 SunOS 4.1.1 开始冠上 Solaris 的字眼，以行动靠拢 SVR4。OSF 则是在 1990 年发表了 OSF/1。UNIX 版本的问题因而更加混乱了。但有趣且可笑的是，开放系统 -- Open System，这个双方都标榜的理念与观念却因此在计算机产业界引起了回响，这点倒是原先所始料未及的。

不久 AT&T 撤销了对 SUN 的投资，同一个阵营的成员彼此也因而劳燕分飞。USL 在 1991 年正式转变了一家独立的商业公司。但 UNIX 在商业市场上的价值却出现了变化...



## 让 UNIX 自由 -- Networking Release 2

自从 UNIX 走出贝尔实验室后，研究机构与学术界就扮演了继承与发展的双重角色。在 1979 到 1984 年这段期间，UNIX 的拥有者 AT&T，对于学术界的授权政策尚可用『大方』来形容；同时也对学术界做某种程度的资助与合作。当时的学术界，得助于 AT&T 的大方授权与分享程序原始码，研习 UNIX 这个分时操作系统开始在学术界蔚为一股风气，甚至可以说是一种潮流或一种流行。其中，像柏克莱 BSD 对 UNIX 的贡献，就是一个公开的事实。但早期的 BSD 使用者，是必需向 AT&T 支付授权金的。这点，从产业界资助学术界的角度来看是一点也不值得惊讶的。因为资金的援助为了就是取得其成果。所以当时基于 AT&T 原始码所发展的成果，均归属 AT&T 所有。也因而 AT&T 掌控了 UNIX 的所有权。到了 1984 年以后，AT&T 开始更积极地保护 UNIX 的原始码；AT&T 甚至还要求各大学的使用人员签订保密条约，想藉此防堵 UNIX 的原始码从学术单位流出，以影响到商业利益。

在 DARPA 资助柏克莱从事 BSD OS 发展的过程中，诞生了 TCP/IP 这项广泛影响现今计算机与因特网的通讯协议。由于 DARPA 对于资助开发的软件项目有明文规定接受资助者必须无条件地释出程序的原始码，所以 TCP/IP 的原始码与程序的版权并不属于 AT&T 所有。这点在现今看来其意义是不凡的。也正因为有此一条件，柏克莱的 CSRG(Computer System Research Group)因应 BSD Vendors 需求，在 1989 年 6 月发表了 Networking Release 1，她包含了 TCP/IP source code 以及一些工具，提供给当时正开始起步发展的个人计算机制造业者使用。Networking Release 1 授权收费仅 1000 美元，而且不需要 AT&T 的商业授权，取而代之的是柏克莱大学的开放式授权。

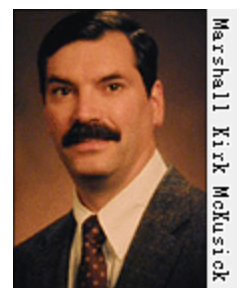
农夫我看柏克莱授权方式，几乎可以说是一种良心式授权方式，在实质的运用上她完全没有限制。她允许原始码或执行档在任何情况下修改并且允许 将修改后的程序从事商业行为而无须任何回馈，当然也没有绝对要求开发者必须要释出原始码。如果你改都不改地加以販售，她也没有意见。但有一点不可违反的限制，就是必须在衍生物的版权声明上提到柏克莱的贡献。这种做法在日后，也没有多少改变，而这样的授权方式也成为了柏克莱的授权精神。



Keith Bostic

由于 Networking Release 1 所得到的响应实在远超过 CSRG 成员的预估。这个不算差的成果，让柏克莱的 CSRG 觉得有必要释出更多属于 BSD 的程序原始码。于是激发 CSRG 的成员 [Keith Bostic](#) 开始组织志愿工作者从事一项就算不能够惊天也足以动地的程序写作计划。计划的主要目的在当时还真让人感到有点“乌托邦”。农夫我个人喜欢戏称她为『解放 UNIX 计划』。

这项计划大体上分成两个部分，操作系统工具 (Utility) 与核心(kernel)。而且参与人员必须在完全没有参考 AT&T UNIX source code 的情况下进行撰写程序的工作。因为只有在这种条件下，写出来的程序代码，才能摆脱 AT&T 的著作权束缚。当然这也绝对不是一件容易的事。Keith Bostic 四处奔走，组织了超过四百名热心的软件工程师，经过了长达十八个月的奋战之后，操作系统主要的工具与链接库才算改写完成。Marshall Kirk McKusick 负责改写当时的核心程序。但系统核心的部分，由于长期以来柏克莱与 AT&T 一直就彼此分享 UNIX 原始码，所以各自所加上去的程序代码早已混杂难分了。为了彻底的厘清双方各自撰写的部分，他们下决心进行逐行比对。首先花了好几个月的时间，将核心程序每一行每一个档案都建立转换比对的数据库。然后接着进行移除来自 AT&T 32V 的程序代码并改写她们。即使是如此，仍旧有 6 只程序



Marshall Kirk McKusick

让他们束手无策，因而无法将核心程序彻底完整地改写。最后，他们还是决定将他们所做的所有成果发表。授权的方式沿用 Networking Release 1 的授权方式，授权的磁带依旧是 1000 美金。这个版本就是 Networking Release 2，也有人称她为 4.3BSD NET/2。发表的时间在 1991 年 6 月。虽然这是个不完整的操作系统。但在今日看来，却有着划时代的意义 -- UNIX OS 自由了。

## 谁是"老大哥" -- 侵权诉讼

AT&T 的 USL 在 1991 年正式转变了一家公司。当然，这意味着她将更重视 UNIX 在商业上的利益。当时的 UNIX OS 早已称霸高阶的计算机市场；从 Cray 超级计算机、IBM 的大型计算机主机、迷你级计算机到工作站，均是 UNIX 的天下(这一点，直到现在21世纪，仍旧没有多大的改变)。即使在 80 年代中期后开始迅速发展的个人计算机，虽然当时被戏称为是玩具计算机，但也仍旧有像 XENIX<sup>[注1]</sup>，Interactive UNIX<sup>[注2]</sup> 等几种向 AT&T 缴过税的商业化版本。UNIX 简直就是 AT&T 的一棵摇钱树。

但这一切在 Networking Release 2(以后简写为 Net/2)出现之后，起了变化！

首先，一位 i386 处理器的玩家名叫 Bill Jolitz，在拿到 Net/2 之后，很快地就将 Net/2 kernal 缺少的程序补齐了。BSD kernal 这时可算是大功告成了。当时 Bill Jolitz 将他们放在因特网与其它人共享他的原始码，并且得到了不少正面的响应。由于这个版本是使用在 i386 微处理器的个人计算机上，所以就命名为 386BSD，在 1992 年 2 月正式发表。这该算是 BSD 首度功能完整且版权独立的版本。Bill Jolitz 是当时唯一的 kernal 维护者。在他离开这个计划之后，继起的 BSD 玩家们延续了这个版本，日后衍生出了 FreeBSD，然后又从其中分支出裂 NetBSD 版本。

另一个将 Net/2 完整化的是一家叫 [Berkeley Software Design, Incorporated](#) 的公司，简称 BSDI<sup>[注3]</sup>。由于 Net/2 的版权声明中，宣称其源文件的合法性，并且允许使用者，从事衍生物的商业行为，所以 BSDI 将他们修改后的系统命名为 BSD/386。他们并将成果打包，刊登广告以 995 美金的售价贩售 BSD/386，而且含原始码，而且还提供免费服务电话的咨询，电号号码是"1-800-ITS-Unix"。时间大约是在 1992 年 1 月。当时，USL 的 System V 含 source code 的价格大约是 BSD/386 价格的一百倍左右。这可惊动了老大哥 AT&T。并且正式地书面严重警告 BSDI 违反的注册商标法(电话号码里有 Unix 的字眼)，并公开宣称 AT&T 拥有 UNIX 的注册商标。BSDI 再次刊登广告公开反击 AT&T，声明她的商业行为完全合法。果不期然，BSDI 的博命演出让双方手牵手走上法庭。

AT&T 的 USL 控告 BSDI 剽窃他的 UNIX 原始码，要求法官还他公道。在听证会上，BSDI 祭出早已准备好的法宝：自己在无任何 AT&T source code 的条件下写出的合法档案，以孰造?BSD 授权的 Net/2 source code。前面的证据足以让 BSDI 立于不败之地，后者让 BSDI 置身在暴风圈外。BSDI 的辨证获得了法官的采信。但 At&T 岂会就此罢手，他们将焦点转移到 Net/2 的 BSD 授权上面，并且重新提出控诉，被告的对象变成了 BSDI 与柏克莱大学；同时 AT&T 还申请法庭禁止 BSDI 一切的 BSD/386 销售行为。就这样，柏克莱大学也对号入座了。

农夫我认为，毕竟 AT&T 是营利企业，她得维护她的商业利益，这点是天经地义的事。虽然柏克莱大学与 AT&T 在 UNIX 发展上有着非比寻常的关系，但商业利益是现实的。企业资助学术界的研究计划，多半是基于商业上的考量；我相信，学术界的少数高层在寻求奥援时不会不明白这一点，即使这有可能让大部分的学术人士无法接受或不愿接受。不管如何，这一记醒棍倒再次挑起了这一点事实。

成为被告的柏克莱大学，只好无奈地面对这场无情的商业诉讼。但他们也同样不甘示弱地对 AT&T 的 System V 著作权提出质疑，因为在 AT&T 的 UNIX 授权声明中完全没有提及柏克莱的贡献。所以柏克莱反控 AT&T 违反 BSD 的授权条款。柏克莱的反击让战况越演越烈，诉讼案一路从 AT&T 的老家新泽西州的联邦法庭打到柏克莱大学的所在地加州法院，但依旧没有结果。

到了 1993 年，官司还在进行中，但 AT&T 却已经打包 USL 准备以一亿美元的价格找寻买主了。最后 AT&T 将 USL 以八千万美元代价的卖给了 [Novell](#)。而新买主也当仁不让地加入了这场混战。但却也因此，战况露出了一线平息的曙光。诉讼案在 1994 年 1 月宣告终结，以庭外和解收场。实际的协议内容仅有当事人知情。

如果从胜负的角度来看这场诉讼，或许柏克莱与 BSDI 是胜利的一方。但如果从 UNIX 发展的脚步来看这场诉讼，就可能根本没有任何一方是胜利者了。

事件平息后的 1994 年 6 月，柏克莱的 CSRG 风光地发表了 BSD 4.4 Lite。在这个版本中，有 70 个档案引用的一份新修改的版权声明，阐述的 AT&T 与 BSD 双方的贡献，并明确地给予档案自由散播的权利。但不知为何，应该有能力完整发表的 BSD 4.4 Lite 还是缺少了三的档案。当时，农夫我也很高兴地买了一本 BSD4.4-Lite CD-ROM Companion，含一张光盘，现在拿在手上，看来总觉得有点呆。

掌握 UNIX source code 以及 UNIX 商标的 Novell，将 UNIX 商标交给 X/open 管理，自己则发展了一套命名为 UNIXWare 的操作系统。推出后市场的反应并不热络。不久，Novell 与 SCO 接头，在 SCO 保证继续支持 UNIXWare 的条件下，UNIX 在 1995 年二次易主，新主人是 [SCO](#)<sup>[注4]</sup>。

备注：

\*1 Intel 在 1978 年发表 4.77 MHz 的 8086 微处理器。1980 年，Microsfot 便以 V7 为基础，发表了在微处理器(microprocessor-based computers)上执行的版本也就是 XENIX。到了 1982 年，一家成立于 1979 年的软件公司 Santa Cruz Operation，成为微软的合作开发厂商。之后她这家公司便一直致力于这个领域里延续到今日，缩写就是今日的 SCO。

\*2 Interactive IS/1 (以 V6 为主体)。这个版本后来演化为比较让人熟知的名字 -- Interactive UNIX。后来因为 Sun Microsystems 致力发展 Solaris for X86，被财力雄厚的 Sun Microsystems 合并了，如今已经不见踪迹了。

\*3 就在我反复修改这段文稿的时候，BSDI 这家公司已经被 Wind River 合并了，改名为 iXsystems。  
2001/05/03

\*4 2001 年 5 月 4 日，[Caldera International, Inc.](#) 正式并购了 SCO 的服务器软件部及SCO专业服务部这两个部门，新的控股公司名为 Caldera, Inc

## GNU 计划 -- 开启了新大道

在 1983 年 9 月 27 日，麻省理工学院人工智能实验室(MIT Artificial Intelligence Lab)的 [Richard M. Stallman](#) (以下简称为 RMS)，在 net.unix-wizards 以及 net.usoft 的 newsgroups 贴上了一份标题为 "[new UNIX implementation](#)" 的讯息。这就是如今广为人知的 [GNU 计划](#) 的开始。在那则被视为「[GNU 宣言](#)」草稿的讯息中，RMS 阐述个人的理念与计划的目的 -- 完成一个命名为 GNU 的 "Free UNIX" 操作系统，希望藉此寻唤理念想同者共襄盛举。

『如果喜欢一个程序的话，那我就应该分享给其它喜欢这个程序的人』，这是 RMS 的座右铭。此点也似乎正是促使其决心运作 GNU 计





划的原动力。当时的 RMS 是想写出一套免费的操作系统。能够让每个人如空气般地自由的取得与使用。选择“UNIX 兼容”为设计的主要原因是；RMS 表明，UNIX 并非他个人理想中的操作系统；他仅阅读一些相关数据，但未曾使用过 (MIT 使用操作系统是 "ITS--Incompatible Timesharing System")；但他认为 UNIX 操作系统具有优良的本质特性。他相信如果 GUN 与 UNIX 兼容将更容易令人接受。所以 RMS 承袭 MIT 用递归缩写命名的传统为 GNU 释译界定 **Gnu is Not Unix**。

1984 年 1 月，RMS 为了展开他的理想而决心离开已经待了十几年的 MIT AI Lab.。当他向他老板 Patrick Winston 辞职时，Winston 试图挽留地说：「你还是要辞职？」。RMS 不为所动的回答：「是」。Winston 显然得到预料中的答案，于是接着说出了思绪里关怀：「你想要保留你的钥匙吗？」。于是 RMS 就从此开始专心地"失业"在他的老东家。一个人窝在他原来的旧办公室中，规划着如何开始他的 GNU 计划。但想开发一套新的 UNIX 兼容的操作系统，即使是财力、人力资源雄厚的顶级计算机公司，也绝对不是一件说想做就能够做到的事。当拟妥他的 [GNU 宣言](#)之后，他正式向全世界呼唤、表明其将所为。种子落地了。



GNU 计划的第一只程序要算是孤军奋战的 RMS 在 1984 年 9 月开始撰写的 Emacs 编辑器。1985 年初，Emacs 已进入可用的阶段。于是 RMS 将她放在 pre.ai.mit.edu 这台机器的 FTP server 上，免费地让 amonymous 的到访者自由下载使用。不久后，Emacs 强悍的功能引发了一些玩家们的注意，由于附上了 source code，玩家们能自己动手为它添加新的功能或除错，很快地，Emacs 获得了相当热烈的回响。随着名声渐播，开始有人相继地加入 GNU 计划的程序写作阵营。"此道不孤"让 RMS 倍感振奋与喜悦。

当时的因特网并未十分普及。所以有不少人虽然对 Emacs 程序有兴趣，却没办法经由 FTP 的管道取得，因而有人透过其它管道向 RMS 询问能如何取得时，这可让当时处在失业状态的 RMS 看到能够支持他继续奋战下去的资金来源--贩售"自由软件"。

一个人、一个独立的个人，要想在现实中实行自己的理念，最先得接受"现实"。唯有接收它是事实，实行理念的 道路，才获得比较稳固的起点与开始。 -- 网络农夫如是说。

想着、写着，脑中突然掠过一丝感受(所以顺便记录在这个地方)。不管如何，RMS 真的开始以一卷磁带 150 块美金的代价，服务有需要的人。也因为基于这个开始与基础，RMS 当年便创立了自由软件基金会 -- [Free Software Foundation](#) (以后简称为FSF)。这对 GNU 计划而言，意味着它已跨越个人化理念的构思阶段，并进入了有群体组织化的运作阶段。同时，RMS 也制定出了属于 GNU 计划的软件版权。RMS 使用 "copyleft" 用来形容她，其实就是与著作权版权(copyright) "对立"之意。这也就是 GPL -- General Purpose License (通用公共授权)。GNU 计划的种子，就这样生根发芽了。

从贩卖 GNU 自由软件扩展到其它的相关软件与参考手册，提供软件技术支持，并接受计算机器材与资金的捐助(捐助者依法享有一定额度的减税)，为企业代训软件人才。FSF 努力地开辟财源却仍旧是运作资金捉襟见肘。RMS 本人并不支薪。而 FSF 聘请软件工程师的待遇，也仅是软件业界薪资水平的一半。但这绝不表示 GNU 计划的软件水准是半桶水。GCC 编译器是 GNU 计划在 1987 年 3 月开始发表的免费编译器，当时的版本是 0.9 测试版。如今最新的版本则是 3.0。这个编译器可以说是今日自由软件写作的基石。**GCC 所解译的机器码，其可靠度绝对不逊于商业化的编译器产品，甚至可以说是优越过商业编译器。**

90 年代初，GNU 计划暨已完成了质量与数量均十分可观的系统工具。这些工具被广泛的使用在当时各种工作站的 UNIX 系统上。虽然已有如此的成果，但仍称不上是完整的操作系统。他们缺少一支属于自己的"核心程序(kernel)"。

UNIX 在 4.2BSD 之后，越写越大 kernal 开始带来一些不便与问题。因而当时便开始有另一个写作理念逐渐在发展--微核心(microkernal)理念。

1985 年, [卡内基大学\(Carnegie Mellon University 简称 CMU\)](#) 暨以 4.3BSD 为发展基础, 将之一拆为二, 分成 micro kernal 与 single server 两个部分。该计划的名称为 "[Mach](#)"。这个计划成了微核心发展的技术先河。GNU 原本有意直接采用 "Mach" 计划的成果。但无奈, 这一等, 从80年代中等到了90年代初, 在几经商量之后, 他们打算采用微核心的写法, 成立自己的计划, 名称叫 "[Hurd](#)"。这项计划, 如今仍在奋战中, 虽然 microkernal 的做法让他们吃了不少苦头; 但可喜的是, 0.2, 0.3 测试版本已经发表。

直到 21 世纪的今日, RMS 依旧努力不懈地耕耘着他的梦土。尽管他本人认为还尚未完全地实现他的「GNU 宣言」; 但他执着于理念的行动, 已凝聚了相当数量的自由软件写作族群们, 在这些人与群体的努力下, 一条新的大道其实已经被开拓出来了, 她通往一个新的世界。大道旁, 枝叶已然繁茂的树荫下, 可口果实一如礼物般地为所有的人成熟。人们称她 -- Linux。

## 新世代的焦点 -- Linux

1990 年代中期, 因特网因出现 World Wide Web, HTML 这种新型态的应用, 而开始迅速的延烧全世界。一夕间, 架设因特网主机的需求激增。这时有一套可以免费取得, 并且能让 x86 计算机升格成 UNIX 级主机的免费操作系统, 开始了吸引全世界目光。在传媒与计算机工程师们的竞相走告下成为了这个新世代的焦点, 这个新的名字就是 Linux。

当然, 这套媒体吹捧的当红炸子鸡, 可非一人之功, 一夕即成的。

Linux 是一套版权彻彻底底与 AT&T 无关的 UNIX-like OS。原始核心程序的创作者是芬兰籍的 Linus Benedict Torvalds(现今他仍旧是核心程序的维护者)。操作系统里大部分的系统工具, 来自于 RMS 行之多年的 GNU 计划成果, 以及其它的自由软件写作计划产生的软件, 如 X Windows、KDE、Gnome 等窗口接口。由于构成操作系统的主要部分均奉行 GPL 版权, 所以市面上有相当多样的安装套件, 目前较广为人知的有 RedHat、Slackware、SuSE、Debian GNU/Linux...。也因此, 这套操作系统, 可说是包含了无数字自由软件写作者的共同心血。如此的一套操作系统其实也就是 RMS 多年来想要达成的宿愿 -- "Free UNIX"。所以, RMS 本人总认为该将名称改做"GNU/Linux"。因此, 也有人用 GNU/Linux 来称呼这个操作系统。



Torvalds 打从十岁出头当他外公的"键盘手"开始, 到了中学就已成了不折不扣的计算机迷。1990 年, 当他就读赫尔辛基大学(University of Helsinki)信息系二年级, 选修一门「C 语言与 UNIX 操作系统」的课程, 因而疯狂地迷恋上了 UNIX 操作系统。那年正好赫尔辛基大学正好添购的一台 VAX, 安装 Ultrix 操作系统。连接了 16 台终端机供授课师生使用。有所限制的计算机资源, 对一位计算机迷来说是极痛苦忍受的。Torvalds 开始作梦想"搞"一套可以在自己计算机上跑的 UNIX。

1991 年 1 月, Torvalds 利用 "学生贷款" 加上去年的 "圣诞红包", 以分期付款的方式买了一台 386 DX33 个人计算机(他的第三台计算机)。他选择安装的操作系统则是在学术界颇负盛名的 Minix<sup>[注5]</sup>。在几番奋战下, 就绪运作的 Minix OS 功能性却多方面无法满足 Torvalds 的需求, 因而激发了他重头来的欲念。于是 Torvalds 在他的 386 DX33 上逐步探索并撰写出他自己的核心程序。他网络上释放的第一个版本是 1991 年 9 月 17 日的 0.01 版。虽然她是个简陋的开始, 但由于 Torvalds 本人持续维护与网友回馈贡献, 原本一个人所撰写的核心程序竟在不知不觉中逐渐转化成 "虚拟团队" 的运作模式。

然而, 一般计算机使用者, 需要的是可安装运作的操作系统(农夫我习惯以"安装套件"称之), 而非单一的操作系统核心。当时英国的曼彻斯特电算中心(Manchester Computer Center, 简称 MCC)便根据 0.12 版核心程序制作了一套名为 MCC Interim 的安装套件。随后各地的安装套件有如雨后春笋般地出现; 如美国德州 Dave Safford 的 TAMU(Texas A&M University)版、Martin Junius 的 MJ 版、Peter McDonald 的 SLS(Softlanding

Linux Systeem)版等非商业安装套件的出现。在安装需求日增的情况下, Linux 安装套件创造出了一块新的需求市场。这一线商机, 让非商业安装套件的也开始出现在商业市场上。Slackware 大概可算是最早出现的商业安装套件了。到如今, 商业与非商业的安装套件则已多得数不清了。

随着使用人数激增, 核心程序的版本与功能也开始加速演化, 但仍不失于稳健。1994 年 3 月 13 日, 核心程序 1.0 正式发表。其安装套件在功能上的整合已急起直追当时商业版的 UNIX OS。此时的 Linux OS 已拥有数十万名使用者。当时赫尔辛基大学还以此为由举办了一场名为"Linux 首度正式发表会"。就在芬兰电视台与众多传媒的郑重其事的报导下, Torvalds 成了芬兰人的自豪, Linux OS 宛如刚诞生的"超新星", 闪闪发亮地展现在众人眼前。

早期的 Linux 核心程序曾被 Andrew Tanenbaum 指出, 过度紧密地与 x86 处理器结合, 所以他认为 Linux 核心程序将无法移植到别的处理器上。这点显然与 UNIX OS 的可移植性大不相同。当时的情况的确是如此, 这或多或少与 Torvalds 本人受限于拥有的硬件资源有关。但当 Linux 的使用族群拓展开来之后, 便开始有人主动地将她移植到不同的平台上。像 Dave Miller 即以不下于 Torvalds 狂热与学习精神将 Linux 成功地移植到 SUN 的 SPARC 工作站上。此外如 Amiga、Atari、PowerPc、MIPS R4000 也陆续见到 Linux 的身影。这些移植严格地从技术的角度来看, 仅能说是"个案"。但这已激发 Torvalds 的兴趣。真正撼动 Linux 核心的移植是对 Alpha 处理器。



1994 年 5 月, 在 DEC 使用者协会上, Digital 的工程师 John Hall(外号 Maddog)碰上了 Torvalds, 双方一见如故。Maddog 力劝 Torvalds 将 Linux 移植到 Alpha 芯片上, 并主动提供了一台 Alpha 计算机供 Torvalds 研究使用。当年可说是全世界最快的 64-bits Alpha 芯片是 DEC 引以为豪的一项成就, 其架构与功能均优越过同一时期的 Intel 32-bits 处理器。这种技术性的挑战吸引了 Torvalds 的投入。这项移植, 但这对原先以 x86 微处理器为写作基础的 Linux 核心程序而言, 实在不是一件小工程。在 Torvalds 与 DEC 相关人员的将近一年奋战后, Linux 核心程序脱胎换骨, 成功地移植到 Alpha 处理器上(与 x86 处理器使用同一套程序代码)。

1995 年 3 月, 被戏谑是 Linux'95 的 1.2 版核心程序正式发表, 支持 Intel x86、DEC Alpha、SUN SPARC、MIPS 等处理器。

1996 年 6 月, 核心程序版本由 1.3 直接跃升为 2.0 版。Torvalds 本人正式钦定了一只"企鹅"作为 Linux 的标志。同时也开始支持对称式多重处理器(Symmetric Multi-Processing, 简称 SMP)架构的计算机; 而支持的处理器则又多了 Motorola 68k 和 PowerPc。在自由软件团体们的努力与计算机产业业界的支持投入之下, Linux 具备的功能逼近商业版 UNIX OS。当然, Linux 要达到"成熟"与"稳定", 实际上还有好长的一段路要走。

时至今日, 散播在全球各地的 Linux 虚拟发展团体, 仍旧持续地发展中。能维持到什么时候? 这在将来的历史自有答案。但至少在今, 一个 RMS 奋斗的目标 -- 可自由分享程序代码的操作系统, 已可贵地呈现在我们的眼前。

◇备注:

\*5 Minix 是 Andrew Tanenbaum 教授为教学目的而撰写的操作系统。在教育界可算是一套学习 UNIX 基础的好范本。

## 新文明世纪 自由共享



到此，这一段关于 UNIX 发展的文字，已从过去的历史当中走回到了今日 ... 21 世纪的今日。本文也接近尾声了。请诸位原谅农夫将以极为自身的历史感受，来作为本文的结语。

阅读与探究历史，是农夫个人在年轻时即有的一点小癖好。通常我无法忍受对自己喜好事物的缘由一无所知。所以我会 想办法去探究她由谁所创、因何而生与发展的沿革。也正因为如此，我才会为 UNIX 这个当初我没能从英文字典上找到的怪字，写了这么一篇文章。

然而在 UNIX 的发展过程当中，我惊讶地发现了一项有别于我探索 20 世纪历史的东西。我相信诸君应该清楚，20 世纪是人类文明史上最为血腥残暴的一段岁月。在其间，多数民族的上个世代所遭逢的苦难，都是空前的。哲学家柏林(Isaiah Berlin)回顾 20 世纪的感受，说了以下这样的一段话。

「我的一生--我一定得这么说一句--经历了二十世纪，却不曾遭逢个人苦难。然而在我的记忆之中，它却是西方史上最可怕的一个世纪。」

的确，每当我阅读 20 世纪的相关史料，我就更能加倍地感受到这份莫名的幸运。我生长在台湾，这块回顾其历史仅能以"悲土"称之的岛上，她的苦难直至今日亦尚未完全结束。尽管多数年轻的一代已然淡忘，来自何方，归往何处。身为一个中国人，站立在这块似乎仍将被同胞武力相向的孤岛上....我已不清楚历史伤口会因得到同胞的爱而痊愈，还是再次因人类残暴掠夺的天性而迸裂.....抱歉，离题了。

我想说的是，在 20 世纪末的因特网时代中，我感受到了令人喜悦地，根源于心、跨越既有疆界藩篱的自由共享文明。这相较于 20 世纪初将"战争"视为文明象征的人类而言，实属无价可贵的进展。即使这文明仍仅是刚播下的种子。但我相信，她将如贝聿铭所言：

「你永无法明确知道你已播种的东西何时可以收割；或许只有一次收成，或许可重复收成。你也许遗忘曾播种了些什么，一种经验，一种感受，与某人的关系，抑或一种哲学及一项传统。然后，忽然间就开花了，被全然不同的环境所唤醒。这种盛开可以冲破藩篱及整个时代。」

多 希望亲眼看到，几个世代后的某日，人类彼此掠夺的行为如天花一般地在人类社会中绝迹；而，共享已成为人类整体奉行的道德公理。如果这样的一个社会是我们今日所企求的；那么，这个方向与希望，就值得你我花一生的精力去努力。当然，这仅只是一个人的希望，我也清楚这世间并非如此美好。但，如果因假设一事物不可能做到，而决定不去做；那是假设得到胜利，而非真实的事实。或许过去的历史，曾经证实正义、公理、平等与理想的胜利，不过是短暂的昙花一现；那又如何。只要我们不放弃希望，希望就有机会成为真实。今日，所有的美好均因此得来，明日也是。

这几年来，我已看到不少因特网上诸君们的努力。我也相信这崭新文明的种子，有朝一日将展现出令人赞叹、愉悦的美景。未来存在我们尚未发现的国度。我相信，我们能发现未曾走过的通道，打开不曾打开的门，进入玫瑰园中.....那会是一个崭新的文明。

网络农夫 2001/12/12 文定

参考文献：

[20 Years of Berkeley Unix : From AT&T-Owned to Freely Redistributable](#)

Computer : A History of the Information Machine by Martin Campbell-Kelly, William Aspray

Free for All: How LINUX and the Free Software Movement Undercut the High-Tech Titans by Peter Wayner

Just for Fun: The Story of an Accidental Revolutionary by Linus Torvalds, David Diamond

Life With Unix : A Guide for Everyone by Don Libes, Sandy Ressler

[The Creation of the UNIX Operating System](#)

Rebel Code: Linux and the Open Source Revolution by Glyn Moody

Where Wizards Stay Up Late : The Origins of the Internet by Katie Hafner, Matthew Lyon

Copyright©2004, 2003, 2002, 2001 网络农夫 (黄天彦).

本文为自由共享文件，不提供任何明示或默示的保证。作者同意任何人在不违背 GNU Free Documentation License, 版本 1.1 或后继版本规范下，对本文进行拷贝、传播及修改等行为。任何型态的衍生物，必须注明文件来源(<http://www.europa.idv.tw> 网络农夫)并保留本段版权声明或引用 "[GNU Free Documentation License](#)"。