

# Linux 诞生和发展的五个重要支柱

Five Vital Supports for  
the Inception & Development of  
Linux

赵炯

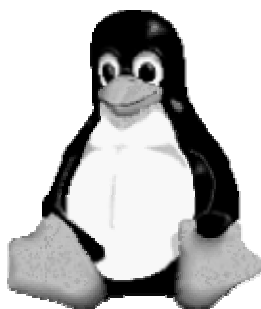
gohigh@sh163.net  
www.plinux.org

2002-12-22

©赵炯, 2002 年

## 目录

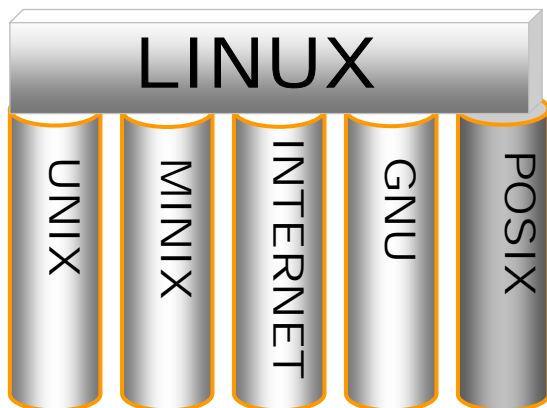
1 概述.....	1
2 起源(UNIX 操作系统) .....	2
3 MINIX 操作系统 .....	7
4 GNU 发展历史 .....	10
5 POSIX 标准 .....	14
6 LINUX 操作系统 .....	16
7 早期主要贡献者.....	33
8 结论.....	37
参考资料.....	37



## 1 概述

Linux 操作系统诞生于 1991 年的 10 月 5 日（第一次正式向外公布的时间）。以后借助于 Internet 网络，经过全世界各地计算机爱好者的共同努力下，才成为今天世界上使用最多的一种 UNIX 类操作系统，并且使用人数还在迅猛增长。

LINUX 诞生和成长的五大重要支柱：



- UNIX 操作系统 -- UNIX 于 1969 年诞生在 Bell 实验室。Linux 就是 UNIX 的一种克隆系统。UNIX 的重要性就不用多说了。
- MINIX 操作系统 -- Minix 操作系统也是 UNIX 的一种克隆系统，它于 1987 年由著名计算机教授 AST 开发完成。由于 MINIX 系统的出现并且提供源代码(只能免费用于大学内)在全世界的大学中刮起了学习 UNIX 系统旋风。Linux 刚开始就是参照 Minix 系统于 1991 年才开始开发。
- GNU 计划（项目、工程）-- 开发 Linux 操作系统，以及 Linux 上所用大多数软件基本上都出自 GNU 计划。Linux 只是操作系统的内核，没有 GNU 软件环境(比如说 bash shell)，则 Linux 将寸步难行。
- POSIX 标准 -- 该标准在推动 Linux 操作系统以后朝着正规路上发展，起着重要的作用。是 Linux 前进的灯塔。
- INTERNET -- 如果没有 Internet 网，没有遍布全世界的无数计算机爱好者的无私奉献，那么 Linux 最多只能发展到 0.13 (0.95) 版的水平。

本文主要根据这五个基本线索来追寻一下 Linux 的开发历程，它的酝酿过程，最初的发展经历。首先分别介绍其中的四个基本要素 (UNIX、MINIX、GNU 和 POSIX, Internet 的重要性显而易见，所以不用对其罗嗦)，然后根据 Linux 的创始人 Linus Toravlds 从对计算机感兴趣而自学计算机知识，到心里开始酝酿编制一个自己的操作系统，到最初 Linux 内核 0.01 版公布，以及从此如何艰难地一步一个脚印地在全世界 hacker 的帮助下最后推出比较完善的 1.0 版本这段时间的发展经过，也即对 Linux 的早期发展历史进行详细介绍。

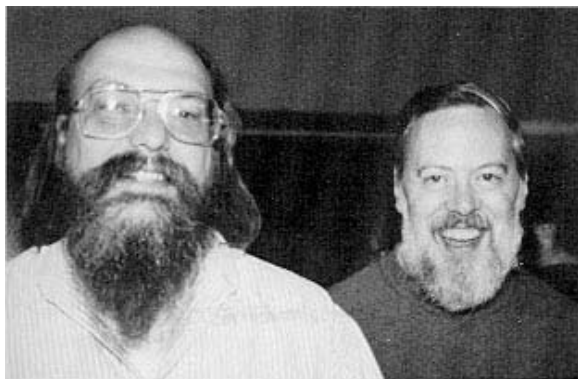
当然，目前 Linux 内核版本已经开发到了 2.5.52 版。而大多数 Linux 系统中所用到的内核是 2.4.20 版。(其中第 2 个数字奇数表示是正在开发的版本，不能保证系统的稳定性) 对于 Linux 的一般发展史，许多文章和书籍都有介绍，这里就不重复。

## 2 起源(UNIX 操作系统)

要说明 Linux (FREAX) 的发展，首先就需要概要介绍 UNIX 的历史。



Bell Labs.



K. Thompson (UC Berkeley, 从 1966 开始在 BTL 工作)

Dennis Ritchie (哈佛应用数学专业, 1968 年开始在 BTL 工作)

1969 -- 贝尔电话实验室 [BTL]

Multics (BTL + GE + MIT)

DEC PDP-7 用户程序空间只有 4K 字节。

玩星际旅行(space travel)游戏。

UNICS → UNIX

1969 年夏天 Ken 的夫人回家乡 California 度假，时为 1 个月。

在这一个月的时间内 Ken 开发出了 unix 的原型，使用的是 BCPL 语言 (Basic Combined

Programming Language)

Richie 将 BCPL 精简为 B 语言 (为了适合 PDP-7)。

1970 年在 PDP-11/20 上实现了 UNIX

1971 年 11 月 3 日 “UNIX Programmer’s Manual”, K. Thompson, D. M. Ritchie.

1972 年 Ritchie 重写 B 语言, 命名为新的 C 语言。



1972 年, DEC PDP11/20  
(PDP - Programmed Data Processor)



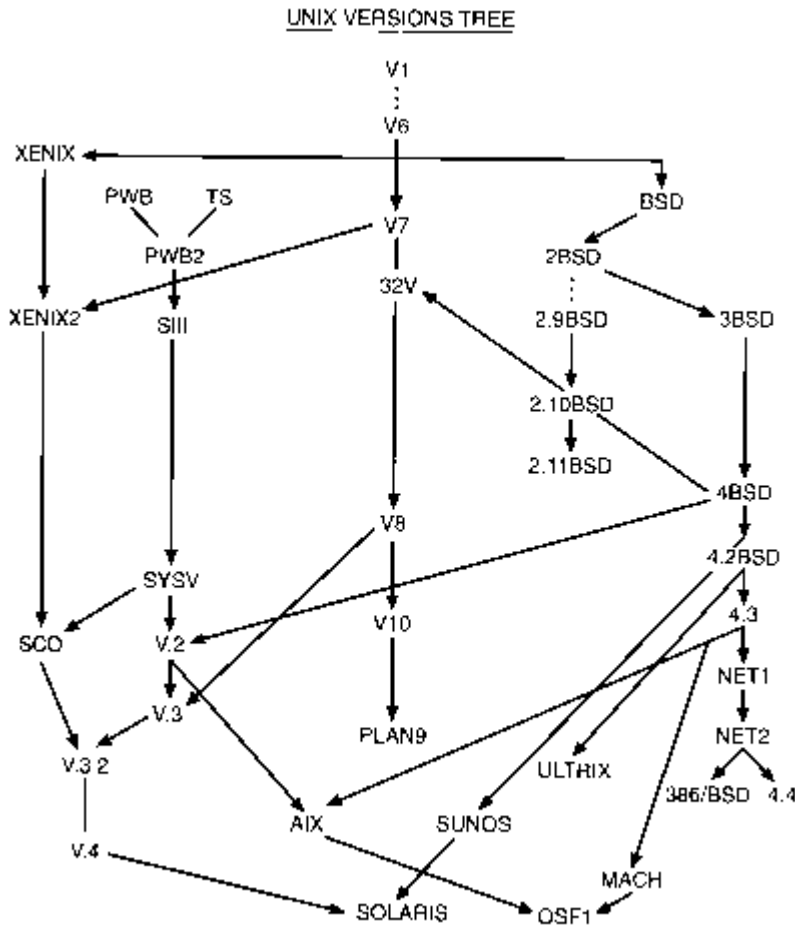
Dennis Ritchie  
<http://cm.bell-labs.com/cm/cs/who/dmr/>



<The C Programming Language>1988

<The Practice of Programming>1999

<The Unix Programming Environment>1984



UNIX 起初主要有三类：

到上世纪 80 年代末，90 年代初，基本上存在两类 UNIX：一类是 AT&T 的系统 V (System V Release - SVR)，另一类是 Berkeley 分校发布的 BSD 版本 (Berkeley Software Distribution)。而 SVR4 则基本上是这两者的综合与合并。到 91 年末，开放软件基金会 (Open Software Foundation) 作为与系统 V 的直接竞争，发布了 OSF/1 系统。

下面对系统 V、BSD 和 OSF/1 的主要发行版及其特性介绍。

AT&T 的系统 V，典型的使用平台是 Intel 硬件环境。是被移植最多的操作系统，并增加了许多 BSD 的特性（包括 csh shell，作业控制，termcap，curses 终端编程库函数，vi 编辑程序，符号连接等）。系统 V 现在已经由 UNIX 国际 (UNIX International - UI) 组织接管，该组织成员包括 AT&T，Sun 公司等。（comp.unix.sysv[23]86）

BSD (Berkeley Software Distribution)，其典型的应用环境是 VAX 计算机、RISC 芯片计算机以及许多的工作站。目前是计算机研究工作等使用较多的 Unix 系统（比如在计算机网络协议研究等领域）。由于 BSD 是大学研制开发的，并且在一定的程度上比较容易得到其源代码，因此对于 UNIX 的流行，它起了很大的作用。Unix 早期版本许多性能上的改进都始于 BSD。BSD 是由加州大学伯克利分校的 CSRG (Computer System Research Group) 组织进行研制和维护，但到 1992 年该组织被关闭了。（comp.unix.bsd），不过现在已衍生出许多分支，包括 FreeBSD、386BSD、BSD386、BSDNet 等，都是对原 BSD 系统的克隆。

开放软件基金会 OSF(The Open Software Foundation)在 1991 年末发布的 UNIX 系统称为 OSF/1。是一个与 SVR、POSIX、X/Open 等兼容的系统。OSF 组织包括许多著名的厂家,如 Apollo、Dec(现属于 Compaq)、HP(已与 Compaq 合并)、IBM 等。

目前 Unices 变种非常多,最主要的也有 100 多种,但是 Linux 是使用人数最多的一种。下面列出了一些比较有名的 UNIX 系统。

AIX: IBM's Unix, based on SVR2 (later up to SVR3.2?) with varying degrees of BSD extensions, for various hardwares. Proprietary system admin (SMIT). Both 850 and Latin-1 CPs. Quite different from most Unices and among themselves.  
Newsgroup: comp.unix.aix.  
AIX was to have been base for OSF/1 until Mach was chosen instead.  
I hope this subsection is converging :-)

AOS (IBM): 4.3BSD port to IBM PC RT (for educational institutes).  
Don't confuse with DG's proprietary OS of same name.

Arix: SV

A/UX (Apple): SV with Berkeley enhancements, NFS, Mac GUI. System 6 (later System 7) runs as guest of A/UX (opposite of MachTen).  
Newsgroup: comp.unix.aux.

3B1 (680x0): SV-based, done by Convergent for AT&T.  
Newsgroup: comp.sys.3b1.

BNR/2: stands for BSD Net/2 Release? Includes NetBSD/1, FreeBSD.

BOS for Bull's DPX/2 (680x0)

386BSD: Jolitz's port of Net/2 software. Posix, 32-bit, still in alpha

BSD/386 (80386): from BSDI, with source (augmented Net2 software)  
Newsgroup: comp.unix.bsd.

Chorus/MiXV: Unix SVR3.2 (SVR4) over Chorus nucleus, ABI/BCS.

Coherent (80286): Unix clone compatible with V7, some SVR2 (IPC).  
V4.0 is 32-bit. Newsgroup: comp.os.coherent

Consensys: SVR4.2

CTIX: SV-based, from Convergent

D-NIX: SV

DC/OSx (Pyramid): SVR4.

DELL UNIX [DELL Computer Corp.]: SVR4

DomainIX: see DomainOS below.

DomainOS (Apollo, now HP): proprietary OS; layered on top is BSD4.3 and SVR3 (a process can use either, neither or both). Development now stopped, some features now in OSF/1 (and NT). Now at SR10.4.  
Name for SR9.\* was DomainIX. Newsgroup: comp.sys.apollo.

DVIX (NT's DVS): SVR2

DYNIX (Sequent): 4.2BSD-based

DYNIX/PTX: SVR3-based

Esix (80386): pure SVR4, X11, OpenLook (NeWS), Xview

Eurix (80?86): SVR3.2 (Germany)

FreeBSD: 386bsd 0.1 with the patchkit applied, and many updated utilities.

FTX: Stratus fault-tolerant OS (68K or i860-i960 hardware)

Generics UNIX (80386): SVR4.03 (Germany)

GNU Hurd (?): vaporware from the Free Software Foundation (FSF):  
Unix emulator over Mach 3.0 kernel. Many GNU tools are very popular (emacs) and used in the PD Unices.

HP-UX (HP): old from S III (SVRx), now SVR2 (4.2BSD?) with SV utilities

Interactive SVR3.2 (80x86): pure SVR3. Interactive has been bought by Sun; will their system survive Solaris?

Idris: first Unix clone by Whitesmith.

IRIX (SGI): Version 4: SVR3.2, much BSD. Version 5.x (current is 5.2) is based on SVR4. Newsgroup: comp.sys.sgi.

Linux (386/486/586): Unix under GPL (not from FSF, though). Available with sources. POSIX compliant w/ SysV and BSD extensions. Being ported to ALPHA and PowerPC (ports for 680x0 Amigas and Ataris already exist). Newsgroup: comp.os.linux.

MachTen, Tenon Intersystems: runs as a guest of System 6, no memory protection, 4.3BSD environment with TCP, NFS.

MacMach (Mac II): 4.3BSD over Mach 3.0 microkernel, X11, Motif, GNU software, sources, experimental System 7 as Mach task. Complete with all sources (need Unix license).

Mach386: from Mt Xinu. Based on Mach 2.5, with 4.3BSD-Tahoe enhancements. Also 2.6 MSD (Mach Source Distribution).

Microport (80x86): pure SVR4, X11, OpenLook GUI

Minix (80x86, Atari, Amiga, Mac): Unix clone compatible with V7. Sold with sources. Being POSIXified (sp?). Newsgroup: comp.os.minix.

MipsOS: SVish (RISC/OS, now dropped, was BSDish)

more/BSD (VAX, HP 9000/300): Mt Xinu's Unix, based on 4.3BSD-Tahoe. Newsgroup: comp.os.xinu?

NCR UNIX: SVR4 (4.2?)

Net/2 tape (from Berkeley, 1991): BSD Unix, essentially compatible with 4.3BSD, includes only sources free of AT&T code, no low-level code. See 386BSD and BSD/386 above.

NetBSD 0.8: is actually 386bsd in a new suit. Ported to [34]86, MIPS, Amiga, Sun, Mac. What is relation to Net/2?

NextStep (Next): BSD4.3 over Mach kernel, own GUI.

NEWS-OS (Sony)

OSF/1 (DEC): DEC's port of OSF/1. I think this is now (4/93) available on DEC's latest Alpha AXP (64-bit machine).

OSx (Pyramid): Dualport of both SysV.3 and BSD4.3.

PC-IX (IBM 8086): SV

Plan 9 (AT&T): announced 1992, complete rewrite, not clear how close to Unix it is. Key points: distributed, very small, various hardwares (Sun, Mips, Next, SGI, generic hobbit, 680x0, PCs), C (not C++ as rumors had it), new compiler, "8 1/2" window system (also very small), 16-bit Unicode, CPU/file servers over high speed nets.

PowerOpen: announced Unix which is to run on the PowerPC chip (from IBM, Apple and Motorola). Mac UI, compatibility to AIX.

SCO Xenix (80x86): Versions for XT (not robust!), 286, 386 (with demand paging). Today bulk of code is from System V. Stable product.

SCO Unix (80x86): SVR3.2

Sinix [Siemens]: System V base.

Solaris (Sparc, 80386):

- 1.0: essentially same as SunOS 4.1.1, with OpenWindows 2.0 and DeskSet utilities.
- 1.0.1: SunOS 4.1.2 with multiprocessing (kernel not multithreaded); not for 386
- 2.0: (initially announced as SunOS 5.0 in 1988) based on SVR4 (with symmetric MP?), will include support for 386; with OpenWindows 3.0 (X11R4) and OpenLook, DeskSet, ONC, NIS. Both



a.out (BSD) and elf (SVR4) formats. Kerberos support. Compilers unbundled!

SunOS (680x0, Sparc, i386): based on 4.3BSD, includes much from System V. Main Sun achievements: NFS (1984), SunView (1985), NeWS (1986, postscript imaging, now in OpenWindows), OpenLook GUI standard, OpenWindows (NeWS, X11, SunView!). Newsgroup: comp.sys.sun.\*.

UHC (80x86): pure SVR4, X11, Motif

Ultrix (DEC): based on 4.2BSD with much of 4.3.

Newsgroup: comp.unix.ultrix.

UNICOS (Cray): System V base. Newsgroup: comp.unix.cray

UnixWare Release 4.2 [Univel]: SVR4.2.

UTEK (Tektronix)

Xenix (80x86): 1st Unix on Intel hardware, based on SVR2 (previously on S III and even V7). Newsgroup: comp.unix.xenix.

还有许多实时 Unix (Real-time Unices)。

对于 UNIX 的源代码，1989 年我校顾伟楠老师为我从 AT&T 公司人员那里搞到了 Unix V7 的源代码复印稿。由于当时水平有限，加上其中使用了 PDP-11 机的汇编语言，没有花时间专研下去。1993 年，朋友为我从常州实达搞来了 SYSTEM V 的部分源代码，此时源代码量已经非常庞大。国内有关 UNIX 的最初介绍，或者说比较知名的文章是尤晋元和另外 1 人对 Unix V7 版的分析报告。由计算机研究与发展以一本专刊发表。

浙江大学也在 UNIX 方面研究的比较多。

### 3 MINIX 操作系统

要说 MINIX 就要简单介绍一下 MINIX 的创始人 Andrew S. Tanenbaum (AST) 其人。



<http://www.cs.vu.nl/~ast/>

<http://www.cs.vu.nl/~ast/minix.html>

[ast@cs.vu.nl](mailto:ast@cs.vu.nl)



AST 是在荷兰 Amsterdam 的 Vrije 大学数学与计算机科学系统工作，是 ACM 和 IEEE 的资深会员(全世界也只有很少人是两会的资深会员)。共发表了 100 多篇文章，5 本计算机书籍。



AST 出生在美国纽约，但是是荷兰侨民(1914 年他的祖辈来到美国)。他在纽约上的中学、M. I. T 上的大学、加州大学 Berkeley 分校念的博士学位。由于读博士后的缘故，他来到了家乡荷兰。从此就与家乡一直有来往。后来就在 *Vrije* 大学开始教书、带研究生了。



荷兰首都 Amsterdam 是个常年阴雨绵绵的城市，而对于 AST 来说，这最好不过了，因为这样他就可以待在家里摆弄他的计算机了。

他不太喜欢窗口界面(WYSIWYG)，他认为 MS-DOS 要比 Windows 好。认为计算机要比人做事的效率高得多，因此他只使用字符界面(命令行)。他开玩笑地说“我母亲说指点人家是很不礼貌的”



他喜欢摄影，也喜欢美术。他的主页上充满了色彩缤纷的小图案。去过很多国家开会、讲学。在主页上存放有许多他在各国拍的照片。



他也养了一个宠物，一只漂亮的小白兔，名字叫 Bram。



他生活哲理是：

KISS: Keep It Simple, Stupid - 做事要简单明了，蠢蛋。

Get it right the first time. - 第一次就要做正确。

在 1980 年前后，他编写了一个编译器写作系统，称为 Amsterdam Compile Kit (ACK) (VUCK)。

MINIX 是他 1986-87 年编制的，主要用于学生学习操作系统原理。MINIX 名称的含义是 Mini-Unix。由于 MINIX 系统不含任何 AT&T UNIX 的代码，所以不受其许可证的限制。在刚

发行开始的头一个月(1987年1月),全世界范围内就有许多人对其产生了兴趣,comp.os.minix 新闻组很快就在 USENET 上被设立。几个月后该新闻组就有近万人在阅读和 post 文章(想想当时的计算机网络使用人数并不是很多)。

最早发行的 MINIX 只能用于 IBM PC 和 PC/AT 微机,但是很快就有人将其移植到了其它种类的计算机上。移植的第一种计算机是基于 68000 的机器,叫做 Atari ST。后来被逐渐移植到了 Amiga、Macintosh 等。

就象 Linux 的发展不能没有 Internet 一样,MINIX 的发展也不能低估 USENET 的作用。有很多人出了各种好点子、对程序进行修改以及为其编制程序,其中最为突出的人名字叫 Bruce Evans (bde@kralizec.zeta.org),他以非常专业的方式对 MINIX 进行了无数次的改进。

对于 MINIX 的工作机理,AST 出版了一本对其进行了详细描述。MINIX 系统也是由 Prentice Hall 出售,需要\$169 美圆!

《Operating Systems: Design and Implementation》

出版社: Prentice Hall, 1987

ISBN: 0-13-637406-9

1989、1990 年我国就已经有人翻译出版了此书:

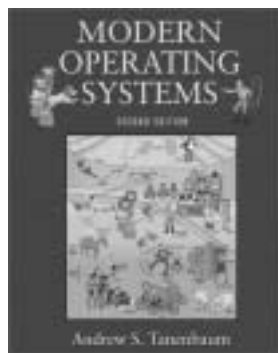
《操作系统教程: MINIX 设计与实现》

[ 陆佑珊、施振川,世界图书出版公司,1990.4。19.60 元 MINIX1.0]

以现在眼光来看,当时书的版权肯定是盗版。

但此书描述的是 MINIX 的早期版本(8086 PC/XT 硬件结构),是与最后一个小型 V7 UNIX 兼容的版本,但是还没有制定 ANSI C 和 IEEE POSIX 标准,后来的版本逐步开始向标准过渡,但并没有全部兼容,因为若要全部兼容标准,那么就违背了 AST 将其设计的很小而能用于教学目的并易于理解。(这也正是后来 Linux 的设计者 Linus 不满足的地方和开发 Linux 的主要原因之一,另外一个原因是 free)。

1997 年出版了此书的第 2 版,此时所附带的源代码已经改为具有 386 32 位特色的 2.0 版:



《操作系统: 设计与实现(第 2 版)》

王鹏, 尤晋元等译, 操作系统: 设计与实现(第 2 版), 电子工业出版社, 1998 年 8 月

到 91 年时版本是 1.5。目前主要有两个版本在使用: 1.5 版和 2.0 版,当时该操作系统在大学使用是免费的,但其它用途不是,当然目前都已经是免费的,可以从许多 FTP 上下载。而且仍然是学习 UNIX 类操作系统工作原理最佳的书之一。

作为一个操作系统，MINIX 并不是优秀者，但它同时提供了用 C 语言和汇编语言写的系统源代码。这第一次使得有抱负的程序员或 hacker 能够阅读操作系统的源代码，在当时这种源代码是软件商一直小心地守护着的。

对于 Linux 系统，他表示对 Linus 的称赞。但他认为 Linux 的发展有很大原因是因为他保持了 minix 的小型化，能让学生在一个学期内就能学完，而没有接纳全世界许多人对 Minix 的扩展要求。因此这激发了 Linus 编写 Linux。Linus 正好抓住了这个好时机。

## 4 GNU 发展历史



<http://www.gnu.org/>

GNU's Not Unix

GNU 项目开始于 1984 年，旨在开发一个类似 Unix、并且是自由软件的完整操作系统：GNU 系统。（GNU 是“GNU's Not Unix”的递归缩写，它的发音为“guh-NEW”。）各种使用 linux 作为核心的 GNU 操作系统正在被广泛的使用。虽然这些系统通常被称作“Linux”，但是严格地说，它们应该被称为 **GNU/Linux** 系统。

自由软件基金会(the Free Software Foundation - FSF)成立于 1985 年，除了软件开发的工作，FSF 还极力保护和推广自由软件。FSF 仅以发布的费用发布 GNU 软件和使用手册，并且接受支持 GNU 开发的可免税礼物。

FSF 是 GNU 项目(计划)的主要组织和开发者。自由软件基金会依靠一些公司捐助和其它商业捐助来维持。有大约 2/3 的运转资金来自个人的捐款，FSF 目的是支持和倡导软件的自由使用、学习、复制、修改和发布。

到上世纪 90 年代初，GNU 项目已经开发出许多高质量的免费软件，其中包括有名的 emacs 编辑系统、bash shell 程序、gcc 系列编译程序、gdb 调试程序等等。这些软件为 Linux 操作系统的开发创造了一个合适的环境，是 Linux 能够诞生的基础之一。以至于目前许多人都将 Linux 操作系统称为“GNU/Linux”操作系统。在 GNU 的网站上还专门有一段文字和 FAQ，解说为什么应该称为 GNU/Linux (<http://www.gnu.org/gnu/gnu-linux-faq.html>)。其中在回答为什么叫 GNU/Linux 而不是 Linux 时说道：

“大多数基于 Linux 内核发布的操作系统基本上都是 GNU 操作系统的修改版。我们从 1984 年就开始 GNU 软件，要比 Linus 开始写它的内核早许多年，而且我们开发了系统的大部分软件，要比其它项目多得多。为公平起见，我们就应该得到公平对待。”

在争名“夺利”啊！很有意思吧。不过这也说明了 GNU 项目在 Linux 操作系统从初生到发展壮大，都离不开 GNU 软件的支持。当初最早的 Linux 0.01 版也只能使用 gcc 1.4 版进

行编译。

GNU 和 FSF 都是由 Richard M. Stallman (RMS) 一手创办的。



RMS 毕业于哈佛，当他 1971 年开始在 MIT 的人工智能实验室工作的时候，就成为软件共享团体的一分子。该团体但是就已经存在了许多年了。当时的软件是鼓励自由拷贝的，而计算机业的传统就是一切为人人所共享。后来的事实证明了许多软件靠着这种共享方式取得了成功。软件的共享并非局限于他那个团体，自从计算机诞生以来就已经以某种形式存在了。就象烹调手艺一样古老。但 GNU 项目做得更进了一步。

当时 AI 实验室使用一个分时操作系统，称为 ITS (the Incompatible Timesharing System)，是由实验室人员为 DEC 的 PDP-10 设计和编制的，PDP-10 是当时的一种大型计算机之一。作为这个团体的一名成员，RMS 主要负责改进这个系统。

当时还没有“自由(免费)软件”这一称呼，但是他们所做的正是这样的一种方式。每当其他大学或公司的人想要移植和使用他们的系统时，他们就非常高兴地复制给他们。RMS 认为每当你见到某人使用一个你不熟悉但却感兴趣的程序，你肯定希望得到源代码，这样你就可以阅读研究程序的工作原理，修改程序，或者将其中的一部分用到其它新程序中去。

但在 80 年代的早期，当 DEC 公司终止了 PDP-10 这个系列计算机的生产后，该计算机的体系结构、所有投入的智慧和能力由于不能适合 80 年代更大的运行内存空间，使得 ITS 上几乎所有软件都过时了。在这之前不久这个团体就已经彻底垮台了。在 1981 年，很多公司几乎将 AI 实验室的所有人员都挖走了。当 AI 实验室购进了一台新的 PDP-10 机器时，管理员决定使用 DEC 的非免费分时系统来取代 ITS。

那个时代的先进计算机，如 VAX 或 68020 (Motorola) 都有自己的操作系统，但都不是免费的：即使想得到一个执行程序拷贝，你也要签署非公开协议。这意味着使用计算机的开始就要许诺不能帮助你的朋友。软件共享合作的社团是禁止的。软件的拥有者定下的规矩就是“若你想要与你的朋友共享软件，那么你就是个盗版者，如果你想对软件作一些改变，就求得我们帮你做”。70 年代末，以微软公司的创始人比尔·盖茨的《致电脑业余爱好者的一封信》为标志，以世界知识产权组织制定的《伯尔尼公约》为框架，软件步入了 Copyright 的时代。这种制度通过一个偶然的事件影响到了 Richard Stallman 先生。

当 Xerox 公司送了一台图像激光打印机，有时纸印光了或夹了纸，一大堆人的打印工作就全部停了下来。送来时，Xerox 随机附带了驱动程序的源代码，实验室的人就把控制打印机驱动程序的功能作了些修改，问题才得到解决。然而到了 1978 年，Xerox 送了一台叫 Dover 的新打印机给人工智能实验室，但不愿再附上源代码。Stallman 为给打印机添加功能以便机器更好工作，希望获得打印机控制程序的源代码，结果被严正拒绝。由此 Stallman 再也无法说不公开协议是纯洁清白的。

软件拥有体制的想法 - 也即不允许共享和修改软件的体制 -- RMS 认为是不利于社会

发展的，是反社会的，不道德的。基于这种想法，RMS 决定编写免费的软件。于是在 1983 年 9 月 27 日，他在 net.unix-wizards 和 net.usoft 新闻组中发布了“net UNIX implementation” - 一个免费 UNIX 系统的想法。第一句就是“Free Unix!”

```
From CSvax:pur-ee:inuxc!ixn5c!ihnp4!houxm!mhuxi!eagle!mit-vax!mit-eddie!RMS@MIT-OZ
From: RMS%MIT-OZ@mit-eddie
Newsgroups: net.unix-wizards,net.usoft
Subject: new UNIX implementation
Date: Tue, 27-Sep-83 12:35:59 EST
Organization: MIT AI Lab, Cambridge, MA
```

Free Unix!

Starting this Thanksgiving I am going to write a complete Unix-compatible software system called GNU (for Gnu's Not Unix), and give it away free to everyone who can use it. Contributions of time, money, programs and equipment are greatly needed.

To begin with, GNU will be a kernel plus all the utilities needed to write and run C programs: editor, shell, C compiler, linker, assembler, and a few other things. After this we will add a text formatter, a YACC, an Empire game, a spreadsheet, and hundreds of other things. We hope to supply, eventually, everything useful that normally comes with a Unix system, and anything else useful, including on-line and hardcopy documentation.

...

在这个 post 中，RMS 谈到了 GNU 项目的宗旨（创建一个 UNIX 内核及其所有实用工具）、为什么要发起 GNU 项目（不能忍受签署非泄密协议）、大家如何能对此项目作出贡献（捐献计算机、资金和程序）等。

在对捐献的计算机，RMS 还提出了特别条件：要求计算机不能需要特别复杂的空调设备和电源，可以在一般环境下工作<sup>©</sup>。

他希望编程人员能够一些 UNIX 工具的兼容版本程序给 GNU。

但是，若要将整个计算机的软件系统都从头开始编制，是一个非常大的计划(项目)，所以 RMS 在 GNU 项目的开始寻找已有的免费软件，不久他发现了一个大学免费使用的编译器，称为 VUCK(Vrei(Free) University Compiler Kit)。该软件可以支持多种计算机语言，包括 C 和 Pascal，并且支持多种计算机，于是 RMS 写信给作者询问 GNU 是否可以使用这个编译器，结果作者(知道这个作者是谁吗?)以嘲笑的口吻回答了他，说大学是免费的但编译程序不是的(the university was free but the compiler was not)。这刺激了 RMS 决定 GNU 项目的第一个程序将是一个多语言、多平台使用的编译程序。之后他还从 Lawrence Livermore 实验室获得过 Pascal 的编译程序，但却不适用，于是他就重起炉灶，从第一行代码写起，最后实现了目前最有名的软件之一 gcc 编译器。在这之前，在 1984 年到 1985 年初，他首先开发出了 emacs 编辑系统。

之后，为了禁止某些公司和团体将自由软件以其它方式改变成他们的私有财产，保持免费软件始终是免费软件，RMS 发布了有名的 GPL(General Public License)(Copeyleft)。

GPL 协议的主要的用途是推广自由软件的使用和学习，并且防止一些别有用心公司在对免费软件进行了一些修改后就申请版权，阻止软件的进一步推广。因此 GPL 协议要求在对

软件进行修改后，如果要再次发布，就需要将已修改的部分同时发布出来。

这后来成为 Linux 的发布所遵守的协议，极大地推动了 Linux 的发展。但是有一点要注意，如果在自己上程序中包括 GNU GPL 公用许可证的话，一定要附上英文原版，否则没有任何法律约束(<http://www.gnu.org/licenses/translations.html>)，这在 gnu 的网站上写得很清楚，因为经过翻译后需要有法律方面的专家审查后才能做到完全具有相同的含义，这点比较困难。

在 RMS 看来，将“骇客”(hacker)一词看作是“安全系统破坏者”(security breaker)是大众媒体的一种误导。作为一个 hacker，RMS 不承认这个含义，并且继续使用这个词来指“喜欢编程并且乐意在其上表现出聪明智慧的人”。



RMS 除热衷于计算机以外，还喜欢音乐和蝴蝶，上图是 RMS 拿着话筒正对着一只美丽的蝴蝶唱着情歌☺。

1999 年 10 月 29 日应 Turbo Linux 公司的约请，RMS 首次访华。出席清华大学 GNU/Linux 研发中心成立仪式，并做了“自由软件运动与 GNU/Linux 操作系统 (The Free Software Movement and the GNU/Linux Operating System)”主题演讲。



2000 年 5 月 28 日，RMS 应邀来到武汉华中科技大学演讲，在华中科技大学学术报告厅做了题为“GNU 精神与自由软件运动”的演讲。他非常支持中国的自由软件运动。



## 5 POSIX 标准

POSIX (Portable Operating System Interface for Computing Systems) 是由 IEEE 和 ISO/IEC 开发的一簇标准。该标准是基于现有的 UNIX 实践和经验，描述了操作系统的调用服务接口，用于保证编制的应用程序可以在源代码一级上在多种操作系统上移植运行。它是在 1980 年早期一个 UNIX 用户组 (usr/group) 的早期工作的基础上取得的。该 UNIX 用户组原来试图将 AT&T 的系统 V 和 Berkeley CSRG 的 BSD 系统的调用接口之间的区别重新调和集成，从而于 1984 年产生了 /usr/group 标准。1985 年，IEEE 操作系统技术委员会标准小组委员会 (TCOS-SS) 开始在 ANSI 的支持下责成 IEEE 标准委员会制定有关程序源代码可移植性操作系统服务接口正式标准。到了 1986 年 4 月，IEEE 就制定出了试用标准。第一个正式标准是在 1988 年 9 月份批准的 (IEEE 1003.1-1988)，也既以后经常提到的 POSIX.1 标准。

1989 年 POSIX 的工作被转移至 ISO/IEC 社团，并由 15 工作组继续将其制定成 ISO 标准。到 1990 年，POSIX.1 与已经通过的 C 语言标准联合，正式批准为 IEEE 1003.1-1990 (也是 ANSI 标准) 和 ISO/IEC 9945-1:1990 标准。

POSIX.1 仅规定了系统服务应用程序编程接口 (API)，仅概括了基本的系统服务标准，因此期望对系统的其它功能也制定出标准。这样 IEEE POSIX 的工作就开始展开了。在 1990 年，刚开始有十个批准的计划在进行，有近 300 多人参加每季度为期一周的会议。着手的工作有命令与工具标准 (POSIX.2)、测试方法标准 (POSIX.3)、实时 API (POSIX.4) 等。到了 1990 年上半年已经有 25 个计划在进行，并且有 16 个工作组参与了进来。

- 802.x (LAN) standards (LLC, ethernet, token ring, token bus)
- POSIX (ISO 9945?): Portable Operating System I/F (Unix, VMS and OS/2!) (only ? have been finalized at this point)
- 1003.1: library procedures (mostly system calls) -- roughly V7 except for signals and terminal I/F (1990)
- 1003.2: shell and utilities
- 1003.3: test methods and conformance
- 1003.4: real-time: binary semaphores, process memory locking, memory-mapped files, shared memory, priority scheduling, real-time signals, clocks and timers, IPC message passing, synchronized I/O, asynchronous I/O, real-time files
- 1003.5: Ada language bindings
- 1003.6: security
- 1003.7: system admin (incl. printing)



- 1003.8: transparent file access
- 1003.9: FORTRAN language bindings
- 1003.10: super computing
- 1003.12: protocol-independent I/Fs
- 1003.13: real-time profiles
- 1003.15: supercomputing batch I/Fs
- 1003.16: C-language bindings (?)
- 1003.17: directory services
- 1003.18: POSIX standardized profile
- 1003.19: FORTRAN 90 language bindings

与此同时，还有一些组织也在制定类似的标准，如 X/Open, AT&T, OSF 等。

但不久，参加者人数开始回落。原因之一是有些计划已经完成；其它原因是有些文档存在较大的分歧，进展不大。随着工业不景气情况的出现，许多组织开始缩减对标准制定的费用支出。到最近一些年来，每年参加季度会议的人数已不到 50 人。主要工作都围绕着军方需要的实时接口标准的制定以及将已经完善的扩展信息加入到基本文档中去，并有开放组织 (Open Group) 来确保这些资源的有效利用和避免重复的工作。

目前的最新标准是 IEEE 标准 1003.1-2001 可以从开放组织的网站上免费获得：  
<http://www.opengroup.org/austin/defectform.html>。

在 90 年代初，POSIX 标准的制定正处在最后投票敲定的时候，那是 1991-1993 年间。此时正是 Linux 刚刚起步的时候，这个 UNIX 标准为 Linux 提供了极为重要的信息，使得 Linux 能够与绝大多数 UNIX 系统兼容。在最初的 Linux 内核代码中 (0.01 版、0.11 版) 就已经为 Linux 与 POSIX 标准的兼容做好了准备工作。在 0.01 版的内核/include/unistd.h 文件中就已经定义了几个有关 POSIX 标准要求的常数符号，并且在注释中就写到“ok，这也许是个玩笑，但我正在着手研究它呢”。

```

1 #ifndef UNISTD_H
2 #define UNISTD_H
3
4 /* ok, this may be a joke, but I'm working on it */
5 #define POSIX_VERSION 198808L
6
7 #define POSIX_CHOWN_RESTRICTED /* only root can do a chown (I think..) */
8 /* #define POSIX_NO_TRUNC*/ /* pathname truncation (but see in kernel) */
9 #define POSIX_VDISABLE '\0' /* character to disable things like ^C */
10 /*#define POSIX_SAVED_IDS*/ /* we'll get to this yet */
11 /*#define POSIX_JOB_CONTROL*/ /* we aren't there quite yet. Soon hopefully */
12

```

1991 年 7 月 3 日在 comp.os.minix 上发布的 post 上就已经提到了正在搜集 POSIX 的资料。(当然此时还不存在 Linux 这个名称，当时 Linus 的脑子里想的可能是 FREAK ☺，FREAK 的英文含义是怪诞的、怪物、异想天开等)。其中透露了他正在进行 Linux 系统的开发，并且在 Linux 最初的时候已经想到要实现与 POSIX (UNIX 的国际标准) 的兼容问题了。

=====

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)  
Newsgroups: comp.os.minix  
Subject: Gcc-1.40 and a posix-question  
Keywords: gcc, posix  
Message-ID: <1991Jul3.100050.9886@klaava.Helsinki.FI>  
Date: 3 Jul 91 10:00:50 GMT  
Organization: University of Helsinki  
Lines: 28

Hello netlanders,

Due to a project I'm working on (in minix), I'm interested in the posix standard definition. Could somebody please point me to a (preferably) machine-readable format of the latest posix rules? Ftp-sites would be nice.  
...

Alan Cox 在 1992 年 9 月 23 日 comp.os.linux 新闻组中回答选择 386BSD 还是 Linux 时说, 通过研究了这些操作系统的源代码后, 他发现将软件移植到 Linux 上要比到 386BSD 上容易得多, 因为 Linux 要更遵守 posix 标准(Linux is more posix like.)。

因此, 可以看出, Linux 的成长一直有 POSIX 标准的辅佐, 没有 POSIX 的指导, Linux 就不会有今天如此地壮大。当然所提到的 5 个基本条件都是缺一不可的。

## 6 LINUX 操作系统



1981 年 IBM 公司推出享誉全球的 IBM PC 机。

1981-1991 年 MSDOS 主宰, windows 3.0(硬件 PC/XT, PC/AT, PC386);  
硬件价格虽然逐年下降, 但软件价格仍然居高不下。

DOS - Bill Gates 花了\$50000 美元从西雅图一个 hacker 手中购得。通过有效的商业推销, DOS 这个简陋的操作系统进入了世界各个角落。PC 用户没有选择, 虽然当时 Apple 的 MACs 是最好的, 但是其天价没人能靠近。

另一个计算机技术阵营是 Unix 世界。但是 Unix 操作系统就不仅是价格贵的问题了。为了寻求高利率, Unix 经销商将价格抬得极高, PC 小用户就根本不能靠近它。曾经一度受到 Bell Labs 的许可而可以在大学中用于教学的 UNIX 源代码一直被小心地守卫着不需公开。

对于广大的 PC 用户，软件行业的大型供应商始终没有给出有效的解决该问题的手段。

而此时出现了 MINIX 操作系统，并有一本详细的书本描述它的设计实现原理。由于 AST 的书写得非常详细，并且叙述有条有理，几乎全世界的计算机爱好者都在看这本书以理解操作系统的工作原理。其中也包括 Linus Torvalds。



当时(1991 年), Linus Benedict Torvalds 是赫尔辛基大学计算机科学系的二年级学生，也是一个自学 hacker。这个 21 岁的芬兰年轻人喜欢鼓捣计算机，测试计算机的能力和限制。但当时缺乏的是一个专业级的操作系统。MINIX 虽然很好，但只是一个用于教学目的简单操作系统，而不是一个强有力的实用操作系统。



到 1991 年，GNU 计划已经开发出了许多工具软件。最受期盼的 Gnu C 编译器已经出现，但还没有开发出免费的 GNU 操作系统。即使是 MINIX 也开始有了版权，需要购买才能得到源代码。而 GNU 的操作系统 HURD 一直在开发之中，但并不能在几年内完成。

对于 Linus 来说，已经不能等待了。从 1991 年 4 月份起，他开始酝酿并着手编制自己的操作系统。刚开始，他的目的很简单，只是为了学习 Intel 386 体系结构保护模式运行方式下的编程技术。但后来 Linux 的发展却完全改变了初衷。

1991 年初，Linus 开始学习 minix 系统时，使用的是一台 386sx 微机。

以下根据 Linus 在 comp. os. minix 新闻组上发布的消息介绍他逐步从学习 minix 系统到开发自己的 Linux 的过程。这些新闻系统到目前仍然可以从 Internet 上查阅到。  
(group.google.com 上的 comp. os. minix 组，搜索 Linus Benedict Torvalds)

Linus 第 1 次向 comp. os. minix 投递消息是在 1991 年 3 月 29 日。

题目是“gcc on minix-386 doesn't optimize”，是有关 gcc 编译器在 minix-386 上运行

的优化问题，由此看来，Linus 在 1991 年的初期已经开始深入研究了 minix 系统，并在这段时间有了改进 minix 操作系统的思想，而且在进一步学习 minix 系统中，逐步演变为想自己重新设计一个基于 Intel 80386 体系结构的新操作系统。

From: torvalds@cc.helsinki.fi  
Newsgroups: comp.os.minix  
Subject: gcc on minix-386 doesn't optimize?  
Message-ID: <1991Mar29.151930.5767@cc.helsinki.fi>  
Date: 29 Mar 91 15:19:30 GMT  
Organization: University of Helsinki  
Lines: 22

Hello everybody,

I've had minix for a week now, and have upgraded to 386-minix (nice), and duly downloaded gcc for minix. Yes, it works - but ... optimizing isn't working, giving an error message of "floating point stack exceeded" or something. Is this normal? I had problems with the crcs, so I'm not actually sure I've gotten it right (pretty sure though), but I'm somewhat surprised that gcc would use floating point in normal optimizations when the program under compilation certainly doesn't.

I've downloaded the sources (2.9Mb for just gcc, not gas etc), but due to a rather small HD I've been unable to untar them completely, so I cannot recompile or anything. I could get one of the floating point packages floating around, if that is the problem, but my understanding is that the current binary cannot take advantage of them anyway. Could somebody please tell me what's the story behind the gcc floating point?

advTHANKSance, Linus Torvalds  
torvalds@cc.helsinki.fi

PS. No it's not a big problem, even without optimization I get around 7000 dhrystones, I'm just wondering. And yes - I'll get a bigger HD as soon as my somewhat strained economy can make it 8-).



Linus 在 comp.os.minix 上的第 2 个消息中(1991 年 4 月 1 日)有这样一句话:

“RTFSC (Read the F\*\*ing Source Code :-)” ，他是在回答有人提出 minix 上的 ps.c 程序编译后运行出错时所说的第一句话。他认为答案就在源程序中。这也说明了对于学习系统软件来说，你不光需要懂得系统的工作基本原理，还需要结合实际系统，学习实际系统的实现方法。因为理论毕竟是理论，其中省略了许多枝节，而这些枝节问题虽然没有太多的理论含量，但却是一个系统必要的组成部分，就向麻雀的一根羽毛。

从 1991 年的 4 月份开始，Linus 几乎花了全部时间研究 386-minix 系统(hack the kernel)，并且尝试着移植 GNU 的软件到该系统上(GNU gcc、bash、gdb 等)。并于 4 月 13 日在 comp.os.minix 上发布说自己已经成功地将 bash 移植到了 minix 上，而且已经爱不释

手、不能离开这个 shell 软件了。



第一个与 Linux 有关的消息是在 1991 年 7 月 3 日在 comp.os.minix 上发布的(当然此时还不存在 Linux 这个名称, 当时 Linus 的脑子里想的可能是 FREAX ☺, FREAX 的英文含义是怪诞的、怪物、异想天开等)。其中透露了他正在进行 Linux 系统的开发, 并且在 Linux 最初的时候已经想到要实现与 POSIX (UNIX 的国际标准) 的兼容问题了。

```
=====  
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)  
Newsgroups: comp.os.minix  
Subject: Gcc-1.40 and a posix-question  
Keywords: gcc, posix  
Message-ID: <1991Jul3.100050.9886@klaava.Helsinki.FI>  
Date: 3 Jul 91 10:00:50 GMT  
Organization: University of Helsinki  
Lines: 28
```

Hello netlanders,

Due to a project I'm working on (in minix), I'm interested in the posix standard definition. Could somebody please point me to a (preferably) machine-readable format of the latest posix rules? Ftp-sites would be nice.

As an aside for all using gcc on minix - the new version (1.40) has been out for some weeks, and I decided to test what needed to be done to get it working on minix (1.37.1, which is the version you can get from plains is nice, but 1.40 is better :-). To my surprise, the answer turned out to be - NOTHING! Gcc-1.40 compiles as-is on minix386 (with old gcc-1.37.1), with no need to change source files (I changed the Makefile and some paths, but that's it!). As default this results in a compiler that uses floating point insns, but if you'd rather not, changing 'toplev.c' to define DEFAULT\_TARGET from 1 to 0 (this is from memory - I'm not at my minix-box) will handle that too. Don't make the libs, use the old gnu lib&libc.a. I have successfully compiled 1.40 with itself, and everything works fine (I got the newest versions of gas and binutils at the same time, as I've heard of bugs with older versions of ld.c). Makefile needs some chmem's (and gcc2minix if you're still using it).

Linus Torvalds            torvalds@kruuna.helsinki.fi

PS. Could someone please try to finger me from overseas, as I've installed a "changing .plan" (made by your's truly), and I'm not certain it works from outside? It should report a new .plan every time.



在 Linus 的下一发布的消息中(1991 年 8 月 25 日 comp.os.minix), 他向所有 minix 用户询问 “What would you like to see in minix?” (“你最想在 minix 中见到什么?”), 在该消息中他首次透露出正在开发一个(免费的)386(486)操作系统, 并且说只是兴趣而已, 代码不会很大, 也不会象 GNU 的那样专业。开发免费操作系统这个想法从 4 月份就开始酝酿了, 希望大家反馈一些对于 minix 系统中喜欢那些特色不喜欢什么等信息, 由于实际的和其它一些原因, 新开发的系统刚开始与 minix 很象(并且使用了 minix 的文件系统)。并且已经成功地 will bash(1.08 版)和 gcc(1.40 版)移植到了新系统上, 而且在过几个月就可以实用了。

最后, Linus 申明他开发的操作系统没有使用一行 minix 的源代码; 而且由于使用了 386 的任务切换特性, 所以该操作系统不好移植(没有可移植性), 并且只能使用 AT 硬盘。

对于 Linux 的移植性问题, Linus 当时并没有考虑。但是目前 Linux 几乎可以运行在任何一种硬件体系结构上。

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Keywords: 386, preferences
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
Lines: 20
```

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).



到了 1991 年的 10 月 5 日, Linus 在 comp.os.minix 新闻组上发布消息, 正式向外宣布 Linux 内核系统的诞生 (Free minix-like kernel sources for 386-AT)。这段消息可以称为 Linux 的诞生宣言, 并且一直广为流传。因此 10 月 5 日对 Linux 社区来说是一个特殊的日子, 许多后来 Linux 的新版本发布时都选择了这个日子。所以 RedHat 公司选择这个日子发布它的新系统也不是偶然的。

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Free minix-like kernel sources for 386-AT
Keywords: 386, preliminary version
Message-ID: <1991Oct5.054106.4647@klaava.Helsinki.FI>
Date: 5 Oct 91 05:41:06 GMT
Organization: University of Helsinki
Lines: 55
```

Do you pine for the nice days of minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on minix? No more all-nighters to get a nifty program working? Then this post might be just for you :-)

As I mentioned a month(?) ago, I'm working on a free version of a minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02 (+1 (very small) patch already), but I've successfully run bash/gcc/gnu-make/gnu-sed/compress etc under it.

Sources for this pet project of mine can be found at nic.funet.fi (128.214.6.100) in the directory /pub/OS/Linux. The directory also contains some README-file and a couple of binaries to work under linux (bash, update and gcc, what more can you ask for :-). Full kernel source is provided, as no minix code has been used. Library sources are only partially free, so that cannot be distributed currently. The system is able to compile "as-is" and has been known to work. Heh. Sources to the binaries (bash and gcc) can be found at the same place in /pub/gnu.

ALERT! WARNING! NOTE! These sources still need minix-386 to be compiled (and gcc-1.40, possibly 1.37.1, haven't tested), and you need minix to set it up if you want to run it, so it is not yet a standalone system for those of you without minix. I'm working on it. You also need to be something of a hacker to set it up (?), so for those hoping for an alternative to minix-386, please ignore me. It is currently meant for hackers interested in operating systems and 386's with access to minix.

The system needs an AT-compatible harddisk (IDE is fine) and EGA/VGA. If you are still interested, please ftp the README/RELNOTES, and/or mail me for additional info.

I can (well, almost) hear you asking yourselves "why?". Hurd will be

out in a year (or two, or next month, who knows), and I've already got minix. This is a program for hackers by a hacker. I've enjoyed doing it, and somebody might enjoy looking at it and even modifying it for their own needs. It is still small enough to understand, use and modify, and I'm looking forward to any comments you might have.

I'm also interested in hearing from anybody who has written any of the utilities/library functions for minix. If your efforts are freely distributable (under copyright or even public domain), I'd like to hear from you, so I can add them to the system. I'm using Earl Chews estdio right now (thanks for a nice and working system Earl), and similar works will be very wellcome. Your (C)'s will of course be left intact. Drop me a line if you are willing to let me use your code.

Linus

PS. to PHIL NELSON! I'm unable to get through to you, and keep getting "forward error - strawberry unknown domain" or something.

Grupi Shqiptar i Shtyryezuesve të Linux



Albanian Linux Users Group

[[[下面是上述文字的译稿:

你是否怀念 minix-1.1 版时的日子? 那时人们干劲十足, 自己编写设备驱动程序。你是否手头正缺少一个很好的项目, 并且非常渴望为符合自己的需要动手修改一个操作系统? 当几乎所有的程序都能在 minix 上运行时, 你是否感到非常失望? 不再有了为了调通一个巧妙的程序而整夜不睡觉的夜猫子? 那么本消息 (邮件、公告) 可能正是为你而发布的:-)。

正如我一个月前所提到的, 我正在开发一个用于 AT-386 微机类似于 minix 的操作系统。它目前已经达到了可用的程度 (当然, 能不能用还依赖于你的具体要求), 而且我很高兴把源代码拿出来广泛发布。目前它的版本是 0.02 (加上已经编制好的 (很小的) 补丁程序, 就是 0.03), 但是我已经能在它上面成功地运行了 bash/gcc/gnu-make/gnu-sed/压缩程序等。

该小巧项目的源程序可以在 nic.funet.fi (128.214.6.100) 上/pub/OS/Linux 目录中找到。该目录中含有一些 README 文件以及几个在 Linux 下运行的二进制执行程序 (bash, update 和 gcc, 你还能要求什么呢:-)。提供了完整的内核源代码, 而且没有使用 minix 的代码。库文件的源代码仅是部分免费的, 所以目前不能给出。照内核现在的样子, 系统已经可以进行编译, 并且已经可以运行。二进制执行程序 (bash 和 gcc) 的源代码可以在同一个地方的 /pub/gnu 目录中找到。

当心! 警告! 注意! 这些源代码仍然需要 minix-386 系统来进行编译 (需要 gcc-1.40, 1.37.1 可能也能用, 但没有试过), 并且如果你想运行它的话还需要 minix 来进行设置, 所以对没有 minix 的人来说, 它至今它还不是一个独立的系统, 不过我正在朝这方面努力着。你还需要有些骇客的本事来设置它, 所以对那些希望一个 minix-386 替代品的人来说, 就不用考虑 Linux 了。它目前主要是供对操作系统感兴趣的骇客使用的, 并且有能使用 minix 的 386 机



器。

该系统需要一个 AT 兼容硬盘（IDE 硬盘当然更好）以及 EGA/VGA 显示卡，如果你还感兴趣的话，就使用 ftp 下载 README/RELNOTES 文件看看，并且/或者给我 EMAIL 告之其它信息。

我能够（当然，几乎是）听到你问自己“为什么？”，Hurd 将在近年（或者两年、或者下个月，谁知道）内推出，而且我已经有了 minix。这是一个骇客为骇客们写的程序，在开发过程中我已经得到了快乐，而某些人可能也乐意阅读它，甚至为自己的需要而修改它。它仍然很小，足以理解、使用和修改，我正期望你可能有的任何建议和说明。

我也对为 minix 系统编写过工具软件/库函数的任何人的反馈信息感兴趣。如果你的软件是可以自由发布的（在版权下甚至公共域内），那么我很希望得到你们的消息，这样我就可以将它们加入到 Linux 系统中。现在我正使用着 Earl Chews 的 stdio（Earl，谢谢你的很好而又能使用的系统），很欢迎这种类似的软件。你的版权当然会保留着，如果你乐意我使用你的代码，就请告知。

Linus

]]]



从此之后，Linus 开始大肆宣扬并怂恿其他人实验、使用它的免费操作系统。下面是 Linus 抓住机会就宣传 Linux 的一个 post 例子。

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Re: Beginner asking questions.
Message-ID: <19910ct21.184940.27505@klaava.Helsinki.FI>
Date: 21 Oct 91 18:49:40 GMT
References: <19910ct21.154757.9187@convex.com>
Distribution: comp.os.minix
Organization: University of Helsinki
Lines: 29
```

Ok, it is obviously time for a small plug once more :-)

There is an experimental FREE minix-like thing which supports paging (not yet to disk, but using the MMU to share pages after a fork etc etc), written only for 386 AT's. It currently isn't very practical, even though a number of people have a running system. It is a bit faster than minix (except the floppy-disk driver which really sucks :-).  
...



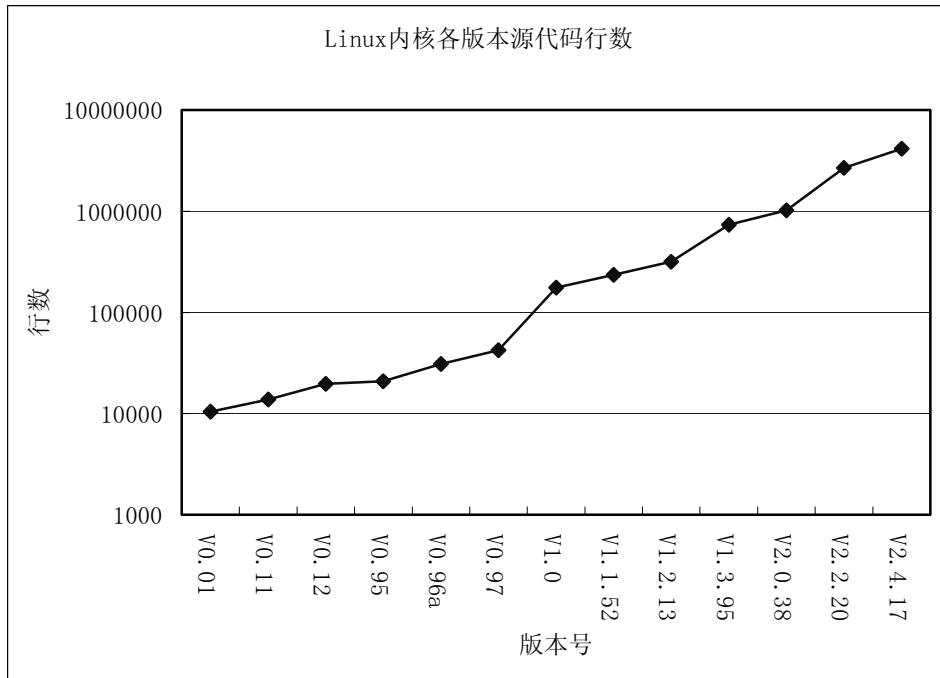
版本的变迁:

- 0.00 (1991. 2-4?) 两个进程分别显示 AAA BBB
- 0.01 (1991. 9?)
- 0.02 (1991. 10. 5)
- 0.03 (1991. 10. 5)
- 0.10 (1991. 10)
- 0.11 (1991. 12. 8)
- 0.12 (1992. 1. 15) 主要加入对数学协处理器的软件模拟程序。
- 0.95(0.13) (1992. 3. 8)
- 0.96 (1992. 5. 12) 开始加入网络支持和虚拟文件系统 VFS。
- 0.97 (1992. 8. 1)
- 0.98 (1992. 9. 29)
- 0.99 (1992. 12. 13)
- 1.00 (1994. 3. 14)
- 1.20 (1995. 3. 7)
- 2.0 (1996. 2. 9)



Linus v2.0

这是 Linus 的第一个女儿，名叫 Patricia Miranda Torvalds。出生于 199 年，Linus 称她为 Linus v2.0©



2.20 (1999.1.26)

2.40 (2001.1.4)

到现在为止最新版是

2.4.20	29	Nov	2002		26,200,000
2.5.52	16	Dec	2002		30,000,000

将 0.13 改称 0.95 版, Linus 的意思是让大家不要觉得离 1.0 版还很遥远。同时, 从 0.95 版开始, 对内核的许多改进之处(补丁程序的提供)均以其他人为主了, 而 Linus 的主要任务开始变成对内核的维护和决定是否采用某个补丁程序。



在当时看来, Linux 的优点在于:

- it's free, full source, and I try to correct bugs you find.
- it's a bit faster than minix, I think.
- uses paging for memory management (not to disk yet)
- multithreaded fs (but then you can get patches to minix that do similar stuff)
- mostly full termios and vt100-console.

- most things easy to port (easier than to minix).

而不足的地方有:

- ONLY 386/486

- early versions: there might be lots of bugs, and you might need to port/hack things to work.

- minix is recommended even for the upcoming version that doesn't absolutely need it.

- currently only VGA (EGA?) support, limited keyboard drivers (US and Finnish) etc



===== 插曲 =====

1992年1月29日中午12点12分, Andrew S. Tanenbaum(ast)的一个 post “LINUX is obsolete”(LINUX 是过时的), 在 ast 与 linus 之间引起了一场著名的舌战。当晚 11 点 Linus 就回了一则态度强硬而又激动的消息。

ast 主要认为 Linux 系统的结构是不分层次的单核式的结构, 而许多陈旧的老式的操作系统就是这个样子的; 再者, Linux 与 CPU 的体系结构紧密结合, 导致其移植困难, 或者说不能移植。由此, ast 认为 Linux 是一种过时的操作系统。

而 Linus 的回应, 充分暴露了 Linus 的学生气质: “你用这来作为 minix 不足之处的借口? 对不起, 但你输了: 我可以有比你更恰当的借口, 可 linux 仍然在各个方面将 minix 打得屁滚尿流。还没说到这么一个事实, 即 PC minix 系统中许多很好的代码还都是 Bruce Evans 编写的呢。”

“你说你写 minix 只是一种业余爱好 - 看谁利用 minix 在赚钱, 而谁又是免费送出 linux 的。说到业余爱好, 要是 minix 是免费的, 那么我对 minix 的最大抱怨就不复存在。Linux 才真正是我的业余爱好(但是是严肃的一种: 最佳的类型): 我并不凭它赚钱, 而且它还根本不是我大学学习中的任何部分。我是一直用自己的时间, 在自己的机器上开发的。”

“你的工作是作为一个教授和一个研究者: 这倒是 minix 中某些脑子有问题的人才写得出来的代码的好借口啊。我只能希望 Amoeba (ast 的另一个项目 - 采用微内核的分布式操作系统) 不要象 minix 那么臭。

就这样, 围绕着操作系统的体系结构和可移植性问题, 大家分成了两派, 两个阵营, 互相不让步, 象 Theodore Y. Ts'o 也参加了进来帮助 Linus 叫阵☺。这场争论持续了将近两个星期, 最终不了了之。因为任何事物都有它的可取之处, 并不可能一棍子大死的。好的地方可以做的更好, 不足的地方可以改进, 目前 Linux 的发展充分证明了这点。

这场争论差点使 Linus 放弃对 Linux 的继续开发，后来 Linus 也承认，只有这件事情让他动摇过继续开发 Linux 的决心。

当然，Linus 后来也认识到自己的态度太激烈，随后还写了另外的 post，向 ast 道歉。



From: ast@cs.vu.nl (Andy Tanenbaum)  
Newsgroups: comp.os.minix  
Subject: LINUX is obsolete  
Date: 29 Jan 92 12:12:50 GMT  
Organization: Fac. Wiskunde & Informatica, Vrije Universiteit, Amsterdam

I was in the U.S. for a couple of weeks, so I haven't commented much on LINUX (not that I would have said much had I been around), but for what it is worth, I have a couple of comments now.

As most of you know, for me MINIX is a hobby, something that I do in the evening when I get bored writing books and there are no major wars, revolutions, or senate hearings being televised live on CNN. My real job is a professor and researcher in the area of operating systems.

As a result of my occupation, I think I know a bit about where operating are going in the next decade or so. Two aspects stand out:

#### 1. MICROKERNEL VS MONOLITHIC SYSTEM

Most older operating systems are monolithic, that is, the whole operating system is a single a.out file that runs in 'kernel mode.' This binary contains the process management, memory management, file system and the rest. Examples of such systems are UNIX, MS-DOS, VMS, MVS, OS/360, MULTICS, and many more.

The alternative is a microkernel-based system, in which most of the OS runs as separate processes, mostly outside the kernel. They communicate by message passing. The kernel's job is to handle the message passing, interrupt handling, low-level process management, and possibly the I/O. Examples of this design are the RC4000, Amoeba, Chorus, Mach, and the not-yet-released Windows/NT.

While I could go into a long story here about the relative merits of the two designs, suffice it to say that among the people who actually design operating systems, the debate is essentially over. Microkernels have won. The only real argument for monolithic systems was performance, and there is now enough evidence showing that microkernel systems can be just as fast as monolithic systems (e.g., Rick Rashid has published papers comparing Mach 3.0 to monolithic systems) that it is now all over but the shoutin'.

MINIX is a microkernel-based system. The file system and memory management are separate processes, running outside the kernel. The I/O drivers are also separate processes (in the kernel, but only because the brain-dead nature of the Intel CPUs makes that difficult to do otherwise). LINUX is a monolithic style system. This is a giant step back into the 1970s. That is like taking an existing, working C program and rewriting it in BASIC. To me, writing a monolithic system in 1991 is a truly poor idea.

## 2. PORTABILITY

Once upon a time there was the 4004 CPU. When it grew up it became an 8008. Then it underwent plastic surgery and became the 8080. It begat the 8086, which begat the 8088, which begat the 80286, which begat the 80386, which begat the 80486, and so on unto the N-th generation. In the meantime, RISC chips happened, and some of them are running at over 100 MIPS. Speeds of 200 MIPS and more are likely in the coming years. These things are not going to suddenly vanish. What is going to happen is that they will gradually take over from the 80x86 line. They will run old MS-DOS programs by interpreting the 80386 in software. (I even wrote my own IBM PC simulator in C, which you can get by FTP from ftp.cs.vu.nl = 192.31.231.42 in dir minix/simulator.) I think it is a gross error to design an OS for any specific architecture, since that is not going to be around all that long.

MINIX was designed to be reasonably portable, and has been ported from the Intel line to the 680x0 (Atari, Amiga, Macintosh), SPARC, and NS32016. LINUX is tied fairly closely to the 80x86. Not the way to go.

Don't get me wrong, I am not unhappy with LINUX. It will get all the people who want to turn MINIX in BSD UNIX off my back. But in all honesty, I would suggest that people who want a **\*\*MODERN\*\*** "free" OS look around for a microkernel-based, portable OS, like maybe GNU or something like that.

Andy Tanenbaum (ast@cs.vu.nl)

=====



Linus 的回复:

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)  
Newsgroups: comp.os.minix  
Subject: Re: LINUX is obsolete  
Date: 29 Jan 92 23:14:26 GMT  
Organization: University of Helsinki

Well, with a subject like this, I'm afraid I'll have to reply. Apologies to minix-users who have heard enough about linux anyway. I'd

like to be able to just "ignore the bait", but ... Time for some serious flamewar!

In article <12595@star.cs.vu.nl> ast@cs.vu.nl (Andy Tanenbaum) writes:

```
>
>I was in the U.S. for a couple of weeks, so I haven't commented much on
>LINUX (not that I would have said much had I been around), but for what
>it is worth, I have a couple of comments now.
>
>As most of you know, for me MINIX is a hobby, something that I do in the
>evening when I get bored writing books and there are no major wars,
>revolutions, or senate hearings being televised live on CNN. My real
>job is a professor and researcher in the area of operating systems.
```

You use this as an excuse for the limitations of minix? Sorry, but you loose: I've got more excuses than you have, and linux still beats the pants of minix in almost all areas. Not to mention the fact that most of the good code for PC minix seems to have been written by Bruce Evans.

Re 1: you doing minix as a hobby - look at who makes money off minix, and who gives linux out for free. Then talk about hobbies. Make minix freely available, and one of my biggest gripes with it will disappear. Linux has very much been a hobby (but a serious one: the best type) for me: I get no money for it, and it's not even part of any of my studies in the university. I've done it all on my own time, and on my own machine.

Re 2: your job is being a professor and researcher: That's one hell of a good excuse for some of the brain-damages of minix. I can only hope (and assume) that Amoeba doesn't suck like minix does.

#### >1. MICROKERNEL VS MONOLITHIC SYSTEM

True, linux is monolithic, and I agree that microkernels are nicer. With a less argumentative subject, I'd probably have agreed with most of what you said. From a theoretical (and aesthetical) standpoint linux loses. If the GNU kernel had been ready last spring, I'd not have bothered to even start my project: the fact is that it wasn't and still isn't. Linux wins heavily on points of being available now.

```
> MINIX is a microkernel-based system. [deleted, but not so that you
> miss the point ] LINUX is a monolithic style system.
```

If this was the only criterion for the "goodness" of a kernel, you'd be right. What you don't mention is that minix doesn't do the micro-kernel thing very well, and has problems with real multitasking (in the kernel). If I had made an OS that had problems with a multithreading filesystem, I wouldn't be so fast to condemn others: in fact, I'd do my damndest to make others forget about the fiasco.

```
[ yes, I know there are multithreading hacks for minix, but they are
hacks, and bruce evans tells me there are lots of race conditions ]
```

#### >2. PORTABILITY

"Portability is for people who cannot write new programs"  
-me, right now (with tongue in cheek)

The fact is that linux is more portable than minix. What? I hear you say. It's true - but not in the sense that ast means: I made linux as conformant to standards as I knew how (without having any POSIX standard in front of me). Porting things to linux is generally /much/ easier than porting them to minix.

I agree that portability is a good thing: but only where it actually has some meaning. There is no idea in trying to make an operating system overly portable: adhering to a portable API is good enough. The very /idea/ of an operating system is to use the hardware features, and hide them behind a layer of high-level calls. That is exactly what linux does: it just uses a bigger subset of the 386 features than other kernels seem to do. Of course this makes the kernel proper unportable, but it also makes for a /much/ simpler design. An acceptable trade-off, and one that made linux possible in the first place.

I also agree that linux takes the non-portability to an extreme: I got my 386 last January, and linux was partly a project to teach me about it. Many things should have been done more portably if it would have been a real project. I'm not making overly many excuses about it though: it was a design decision, and last april when I started the thing, I didn't think anybody would actually want to use it. I'm happy to report I was wrong, and as my source is freely available, anybody is free to try to port it, even though it won't be easy.

Linus

PS. I apologise for sometimes sounding too harsh: minix is nice enough if you have nothing else. Amoeba might be nice if you have 5-10 spare 386's lying around, but I certainly don't. I don't usually get into flames, but I'm touchy when it comes to linux :)

===== 插曲 =====

\* 为什么选择企鹅作为 Linux 的标志?



选择企鹅作为 Linux 的标志还有一则典故。相传有一次 Linus 去南半球旅行的时候，在一个动物园内他见到一群企鹅，出于好奇心理 Linus 就上前用手去摸企鹅，谁知企鹅狠狠啮了他一口，这使得 Linus 对企鹅映象深刻。在给 Linux 系统选吉祥物时，Linus 很自然地想到了企鹅。于是就有了这么一个可爱的小企鹅深入到 Linux 的各个角落。



\* Linux 名称的来由?

Linux 操作系统刚开始时并没有被称作 Linux, Linus 给他的操作系统取名为 FREAX, 其英文含义是怪诞的、怪物、异想天开等意思。在他将新的操作系统上载到 ftp.funet.fi 服务器上时, 管理员 Ari Lemke 很不喜欢这个名称。他认为既然是 Linus 的操作系统就取其谐音 Linux 作为该操作系统的目录吧, 于是 Linux 这个名称就开始流传下来。

在 Linus 的自传《Just for Fun》一书中, Linus 解释说:

“坦白地说, 我从来没有想到过要用 Linux 这个名称发布这个操作系统, 因为这个名字有些太自负了。而我为最终发布版准备的是什么呢? Freax。实际上, 内核代码中某些



早期的 Makefile - 用于描述如何编译源代码的文件 - 文件中就已经包含有“Freax”这个名字了，大约存在了半年左右。但其实这也没什么关系，在当时还不需要一个名字，因为我还没有向任何人发布过内核代码。”

“而 Ari Lemke，他坚持要用自己的方式将内核代码放到 ftp 站点上，并且非常不喜欢 Freax 这个名字。他坚持要用现在这个名字(Linux)，我承认当时我并没有跟他多争论。但这都是他取的名字。所以我可以光明正大地说我并不自负，或者部分坦白地说我并没有本位主义思想。但我想好吧，这也是个好名字，而且以后为这事我总能说服别人，就象我现在做的这样。”

-- Linus Torvalds 《Just for fun》第 84-88 页。



## Linux 的开发模式。

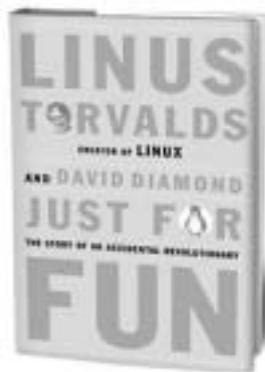
集市式的自由开发模式。

大教堂式开发模式。



2001 年，Linus 出版了个人传记《Just for Fun》，下面是书中开始的一段文字（电子版）。

<b>作者:</b> Linus Torvalds, David Diamond	<b>出版社:</b> Harper Business
<b>ISBN:</b> 0066620724	<b>页数:</b> 288
<b>封面类型:</b> 硬版	<b>日期:</b> 2001
<b>版本:</b> 第 1 版	<b>种类:</b> 传记/Linux



"I was a nerd. Geek. From fairly early on. I didn't duct-tape my glasses together, but I might as well have, because I had all the other traits. Good at math, good at physics, and with no social graces whatsoever. And this was before being a nerd was considered to be a good thing."

In this witty and engrossing narrative, Linus Torvalds, the brilliant mastermind behind the latest Internet revolution, in collaboration with writer David Diamond, chronicles his transformation from a pale, skinny Helsinki college kid to an international folk hero. What began as a childhood hobby soon became the astonishing phenomenon known as the LINUX operating system.

LINUX was created because Linus was curious to see if he could improve upon the operating systems already out there, such as UNIX. How could he create a system that could run all of his favorite software with fewer crashes and faster productivity? Linus posted his early versions of LINUX on the Internet and called upon the most brilliant minds in computer science to enhance his system. What happened next took the world by storm. Linus wasn't the first to use the power of the open source movement, but the excitement he generated when he offered his system to the world was unprecedented. What made LINUX revolutionary was its journey to become the marvel it is today -- and how the system grew from the meeting of thousands of minds around the world. Suddenly, Linus's creation was available for everyone to use, *for free!* Anyone could improve LINUX at whim. No monopolies, no trade secrets, no whispering behind closed doors. LINUX wasn't controlled by a select few -- this remarkable creation was accessible to absolutely anyone -- and still is. The LINUX system is still constantly evolving and improving every day a decade after its inception. Linus describes the history of LINUX in graspable terms and details how the system itself works, without lapsing into mindboggling technical jargon.

Part autobiography and part business philosophy, *Just for Fun* brims with biographical detail about the radical spirit and creativity of Linus Torvalds. It offers a unique glimpse into the mind of an accidental revolutionary and how the altruistic creation of LINUX flourished so successfully in the open source movement.

For general readers, Torvalds spins a witty tale of his fascinating life. Here is the story of a young man who, as a still-rising star, keeps his feet on the ground through a combination of self-deprecating humor and the realization that life is simply about having a good time. Linus's narrative teems with clever anecdotes and his captivating opinions on the future of competition in the computer world. Linus even reveals his own take on the meaning of life.

For techies, this is a candid glimpse at the life of this honored, yet unintentional hero. Linus offers a compelling look at how he sees the road ahead for LINUX and the computer industry. He also divulges how LINUX began as a small spark and spread like wildfire across the world.

Although he's the first to admit that roaring down the freeway in his shiny new BMW has its benefits, Linus never intentionally sought fame and fortune. Yet both found him when Linus Torvalds radically changed the world of technology for one selfish reason: He did it just for fun.

## 7 早期主要贡献者

这里从 Linux 早期开发者中任意选出几位作一些介绍。

### 7.1 Theodore Ts'o



<http://thunk.org/tytso/>

mail: tytso@mit.edu

Theodore Ts'o (Ted Ts'o) 是 linux 社区中的著名人物。他 1990 年毕业于 MIT 计算机科学专业。在大学时代他就积极参加学校中举办的各种学生活动。他喜欢烹饪、骑自行车，当然还有就是 hacking on Linux，后来他开始喜欢起业余无线电报运动 (Amateur radio operator)。目前他在 IBM 工作从事系统编程及其它重要事务。他还是 IETF 成员。

(Internet Engineering Task Force - 国际网络设计、操作、销售和研究者开放团体，主要从事 Internet 网络体系结构发展的研究，由一些列课题组构成，主要通过邮件列表进行工作，每年举行三次会议，IETF 是 RFC 信息库的主要维护者之一)

Linux 在世界范围内的流行也有他很大的功劳，早在 Linux 操作系统刚问世时，他就怀着极大的热情为 linux 的发展提供了 maillist，几乎是在 Linux 刚开始发布起 (1991 年开始) 就一直为 Linux 做出贡献的人，也是最早向 Linux 内核添加程序的人 (Linux 内核 0.10 版中的虚拟盘驱动程序 ramdisk.c 和内核内存分配程序 kmalloc.c)。直到目前仍然从事着与 Linux 有关的工作。他当时在北美洲地区最早设立了 linux 的 ftp 站点 (tsx-11.mit.edu)，而且至今仍然为广大 linux 用户提供服务。他对 linux 作出的最大贡献之一是提出并实现了 ext2 文件系统。该文件系统已成为 linux 世界中事实上的文件系统标准。最近他又推出了 ext3 文件系统，大大提高了文件系统的稳定性和访问效率。作为对他的推崇，第 97 期 (2002 年 5 月) 的 linuxjournal 期刊将他作为封面人物，并对他进行了采访。目前，他为 IBM linux 技术中心工作，并从事着有关 LSB (Linux Standard Base) 等方面的工作。

## 7.2 Alan Cox



homepage: <http://www.linux.org.uk/>

mail: [iiitac@PYR.SWAN.AC.UK](mailto:iiitac@PYR.SWAN.AC.UK)

Alan Cox 原工作于英国威尔士斯旺西大学(Swansea University College)。刚开始他特别喜欢玩电脑游戏，尤其是 MUD (Multi-User Dungeon or Dimension, 多用户网络游戏)。在 90 年代早期 games.mud 新闻组的 posts 中你可以找到他发表的大量 posts。他甚至为此还写了一篇 MUD 的发展史(rec.games.mud 新闻组, 1992 年 3 月 9 日, A history of MUD)。由于 MUD 游戏与网络密切相关, 慢慢的他对计算机网络开始感兴趣。为了玩游戏并提高电脑运行游戏的速度以及网络传输的速度, 他开始接触各种类型的操作系统, 为他的游戏选择一个最为满意的平台。由于没钱, 即使 Minix 他都买不起, 当 Linux 0.11 和 386BSD 发布时, 他考虑良久总算买了一台 386SX 电脑。由于 386BSD 需要数学协处理器的支持, 而 386SX 中是不带的, 所以他安装了 Linux 系统。于是他开始学习带有免费源代码的 Linux。开始对 Linux 产生了兴趣, 尤其是有关网络方面的实现。在关于 Linux 的单用户运行模式问题的讨论中, 他甚至赞叹 Linux 实现的巧妙(beautifully)。

它最早出现在 comp.os.linux 新闻组的时间是 1992 年 7 月 17 日, 回答一个关于使用 KA9Q 邮件程序的问题。Linux 0.95 版发布之后, 他开始为 Linux 系统编写补丁程序(修改程序)(记得他最早的两个补丁程序, 都没有被 Linus 采纳), 成为 Linux 上 TCP/IP 网络代码的最早使用人之一。后逐渐加入 Linux 的开发队伍, 并开始成为维护 Linux 内核源代码的主要负责人之一, 也可以说成为 Linux 社团中在 Linus 之后最为重要的人物。以后 Microsoft 公司曾经邀请他加盟, 但他却干脆地拒绝了。

从 2001 年开始他负责维护 Linux 内核 2.4.x 的代码(Linus 主要负责开发最新开发版内核的研制(奇数版, 比如 2.5.x 版)。

他建议所有想对 Linux 进行 hack 的人, 首先要从 LDP 入手, 尤其是其中 Dave Rusling 写的《The Linux Kernel》(<http://www.plinux.org/LDP/LDP/tlk/tlk.html>), 并且通过修改出错的内核程序或者编写驱动程序入手。

Linux 爱好者一般都是男生, 但世界是辽阔的, 也有许多女生喜欢 Linux, 这里就是一个 Alan Cox 夫人常去的一个女生办的 Linux 网站(<http://www.linuxchix.org/>)。



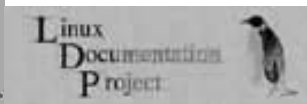
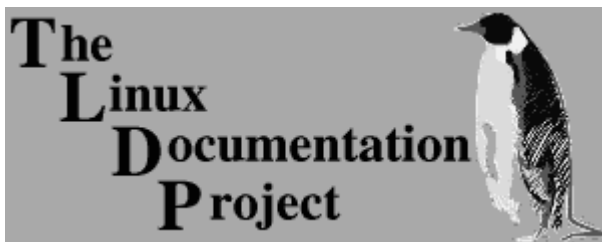
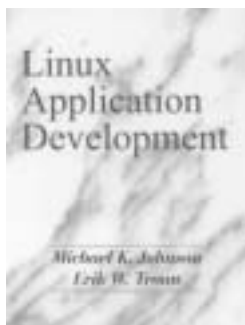
### 7.3 Donald Becker



Linux 操作系统中网络设备驱动程序的主要开发者。

目前正主持着自己的公司 Scyld Computing Corporation，主要从事 Cluster OS 的开发工作。

### 7.4 Michael K. Johnson



<http://people.redhat.com/johnsonm/>

KHG (Kernel Hacker's Guide) 的作者。

《Linux Application Development》一书的作者。最早接触 Linux 操作系统的人之一(从 0.97 版? )，是 LDP (Linux Document Project) 的发起者。曾经在 Linux Journal 工作，现在 RedHat 公司工作。

## 7.5 Werner Almesberger



<http://icapeople.epfl.ch/almesber/>

<http://www.almesberger.net/>

mail:wa@almesberger.net

奥地利人，1967 年 8 月 13 日出生于瑞士苏黎士。1992 年秋毕业于苏黎士 ETH 大学计算机科学系，硕士。之后首先在苏黎士的 IBM 研究实验室工作(93-94 年)，从事 ATM 交换控制软件的开发工作。最后在计算机通信和应用研究院攻读博士，并于 1999 年末毕业。

从 1992 年开始就介入了 Linux 的开发和推广。主要涉及的领域有文件系统、设备驱动程序、引导启动程序(LILO)等。目前人在阿根廷首都布宜诺斯艾利斯(Buenos Aires, Argentina)，主要从事 Linux 的流量控制等项目。

## 7.5 其他贡献者

Linux 系统并不是仅有这些中坚力量就能发展成今天这个样子的，还有许多计算机高手对 Linux 做出了极大的贡献，这里就不一一列举了。主要贡献者的具体名单可参见 Linux 内核中的 CREDITS 文件，其中以字母顺序列出了对 Linux 做出较大贡献的近 400 人的名单列表，包括他们的 email 地址和通信地址、主页以及主要贡献事迹等信息。

## 8 结论

8.1 蝴蝶效应(正反馈) - 如何做一只“兴风作浪”的蝴蝶。

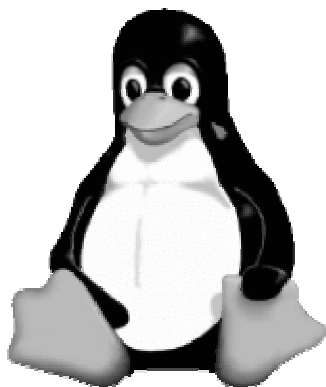
8.2 宣传手段(利用 Internet)

8.3 把握时机

8.4 毅力(两年后才推出 1.0 版)

8.5 初学 Linux 的方法。

- a. UNIX 操作系统原理(Back 的书)
- b. MINIX 操作系统实现(AST)
- c. 阅读 Linux 源代码(可选择早期内核版本, 比如 0.01 版等)。
- d. RTFSC!!!
- e. Internet 信息。



## 参考资料

- [1] POSIX 1003.1 Base Specifications Issue 6. <http://www.opengroup.org/>
- [2] Jim Isaak. The history of POSIX: A study in the standards process. IEEE Computer, Vol.23, P:89-92, July 1990
- [3] Linus B. Torvalds' s homepage. <http://www.cs.helsinki.fi/u/torvalds/>
- [4] Linus Torvalds, David Diamond. Just for Fun. Harper Business, 2001.
- [5] Andrew S. Tanenbaum' s homepage. <http://www.cs.vu.nl/~ast/minix.html>
- [6] Dennis Ritchie' s homepage. <http://cm.bell-labs.com/cm/cs/who/dmr/>
- [7] Richard M. Stallman' s homepage. <http://www.stallman.org/>
- [8] Linux kernel source code. <http://www.kernel.org/>
- [9] Linux source code cross reference. <http://www.plinux.org/lxr/>

- 
- [10] Don Marti, Richard Vernon. Interview: Interview with Ted Ts' o, Linux Journal, May 2002. <http://www.linuxjournal.com/print.php?sid=5891>
- [11] Kent Jones. A brief, serious UNIX history. [http://infocom.cqu.edu.au/Units/aut98/85321/Study\\_Material/Resource\\_Materials/UNIX\\_History/Serious/index.txt.html](http://infocom.cqu.edu.au/Units/aut98/85321/Study_Material/Resource_Materials/UNIX_History/Serious/index.txt.html)
- [12] Alan Cox homepage. <http://www.linux.org.uk/>
- [13] Theodore Ts' o homepage. <http://thunk.org/tytso/>
- [14] Michael K. Johnson' s homepage. <http://people.redhat.com/johnsonm/>
- [15] Werner Almesberger' s homepage. <http://www.almesberger.net/>
- [16] Ragib Hasan. History of Linux. <http://ragib.hypermart.net/linux/>
- [17] 武汉自由软件协会. <http://www.clinux.org/>
- [18] turbolinux 中国分公司. <http://www.turbolinux.com.cn/>