

## 鸟哥的 Linux 私房菜

为取得较佳浏览结果, 请爱用 [firefox](#) 浏览本网页



## 第一章、Linux 是什么

最近更新日期: 2008/07/26

众所皆知的, Linux 的核心原型是 1991 年由托瓦兹(Linus Torvalds)写出来的, 但是托瓦兹为何可以写出 Linux 这个操作系统? 为什么他要选择 386 的计算机来开发? 为什么 Linux 的发展可以这么迅速? 又为什么 Linux 是免费的? 以及目前为何有这么多的 Linux 版本(distributions)呢? 了解这些东西后, 才能够知道为何 Linux 可以免除专利软件之争, 并且了解到 Linux 为何可以同时个人计算机与大型主机上面大放异彩! 所以, 在实际进入 Linux 的世界前, 就让我们来谈一谈这些有趣的历史故事吧! ^\_^

1. [Linux 是什么](#)
  - 1.1 [Linux 是什么?](#)
  - 1.2 [Linux 之前, Unix 的历史](#)
  - 1.3 [关于 GNU 计划](#)
2. [Torvalds 的 Linux 发展](#)
  - 2.1 [与 Minix 之间](#)
  - 2.2 [对 386 硬件的多任务测试](#)
  - 2.3 [初次释出 Linux 0.02](#)
  - 2.4 [Linux 的发展: 虚拟团队的产生](#)
  - 2.5 [Linux distributions](#)
3. [Linux 的特色](#)
  - 3.1 [Linux 的特色](#)
  - 3.2 [Linux 的优缺点](#)
  - 3.3 [其他 Linux 相关话题: 标准与授权](#)
4. [重点回顾](#)
5. [本章习题](#)
6. [参考数据与延伸阅读](#)
7. [针对本文的建议: http://phorum.vbird.org/viewtopic.php?t=23871](#)

---

### Linux 是什么?

我们知道 Linux 这玩意儿是在计算机上面运作的, 所以说 Linux 就是一组软件。问题是这个软件是操作系统还是应用程序? 且 Linux 可以在哪些种类的计算机上面运作? 而 Linux 源自哪里? 这些我们都得来谈一谈先!

---

### Linux 是什么?

我们在[第零章、计算机概论](#)里面有提到过整个计算机系统的概念，计算机是由一堆硬件所组成的，为了有效率的控制这些硬件资源，于是乎就有操作系统的产生了。操作系统除了有效率的控制这些硬件资源的分配，并提供计算机运作所需要的功能(如网络功能)之外，为了要提供程序设计师更容易开发软件的环境，所以操作系统也会提供一整组系统呼叫接口来给软件设计师开发用喔！

知道为什么要讲这些了吗？嘿嘿！没错，因为 Linux 就是一套操作系统！如同下图所示，Linux 就是核心与系统呼叫接口那两层。至于应用程序算不算 Linux 呢？当然不算啦！这点要特别注意喔！

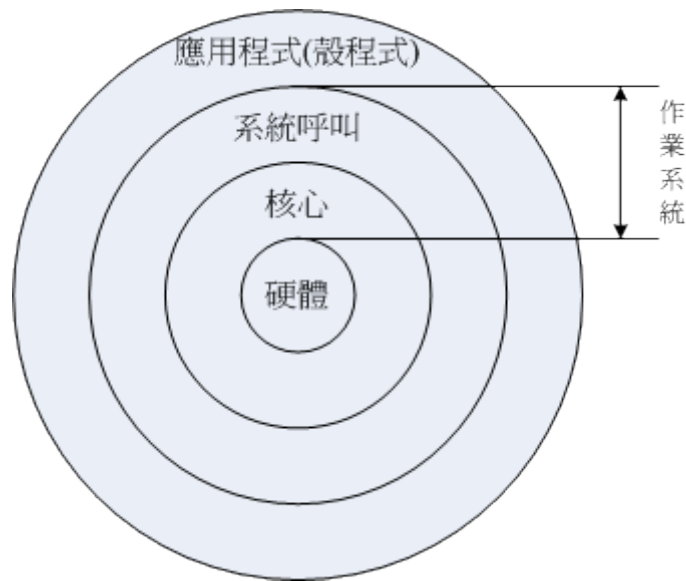


图 1.1.1、操作系统的角色

由上图中我们可以看到其实核心与硬件的关系非常的强烈。早期的 Linux 是针对 386 来开发的，由于 Linux 只是一套操作系统，因此很多工程师在下载了 Linux 核心并且实际安装之后，就只能看着计算机开始运作了！接下来这些高级工程师为了自己的需求，再在 Linux 上面安装他们所需要的软件就是了。

**Tips:**

Torvalds 先生在写出 Linux 的时候，其实该核心仅能『驱动 386 所有的硬件』而已，所谓的『让 386 计算机开始运作，并且等待用户指令输入』而已，事实上，当时能够在 Linux 上面跑的软件还很少呢！



由于不同的硬件他的功能函数并不相同，例如 IBM 的 Power CPU 与 Intel 的 x86 架构就是不一样！所以同一套操作系统是无法在不同的硬件平台上面运作的！如果你想要让 x86 上面跑的操作系统能够在 Power CPU 上运作时，就得要将操作系统进行修改才行。如果能够参考硬件的功能函数并据以修改你的操作系统程序代码，那经过改版后的操作系统就能够在另一个硬件平台上

面运作了。 这个动作我们通常就称为『软件移植』了！

例题：

请问 Windows 操作系统能否在苹果公司的麦金塔计算机 (MAC) 上面安装与运作？

答：

由上面的说明中，我们知道硬件是由『核心』来控制的，而每种操作系统都有他自己的核心。在 2006 年以前的苹果计算机公司是请 IBM 公司帮忙开发硬件 (所谓的 Power CPU)，而苹果计算机公司则在该硬件架构上发展自家的操作系统 (就是俗称的麦金塔，MAC 是也)。Windows 则是开发在 x86 架构上的操作系统之一，因此 Windows 是没有办法安装到麦金塔计算机硬件上面的。

不过，在 2006 年以后，苹果计算机转而请 Intel 设计其硬件架构，亦即其硬件架构已经转为 x86 系统，因此在 2006 年以后的苹果计算机若使用 x86 架构时，其硬件则『可能』可以安装 Windows 操作系统了。不过，你可能需要自己想些方式来处理该硬件的兼容性啰！

Tips:

Windows 操作系统本来就是针对个人计算机 x86 架构的硬件去设计的，所以他当然只能在 x86 的个人计算机上面运作，在不同的平台当然就无法运行了。也就是说，每种操作系统都是在他专门的机器上面运行的喔！这点得要先了解。不过，Linux 由于是 Open Source 的操作系统，所以他的程序代码可以被修改成适合在各种机器上面运行的，也就是说，Linux 是具有『可移植性』，这可是很重要的一个功能喔！ ^\_^



Linux 提供了一个完整的操作系统当中最底层的硬件控制与资源管理的完整架构，这个架构是沿袭 Unix 良好的传统来的，所以相当的稳定而功能强大！此外，由于这个优良的架构可以在目前的个人计算机 (X86 系统) 上面跑，所以很多的软件开发者将他们的工作心血移转到这个架构上面，那就是很多的应用软件啦！

虽然 Linux 仅是其核心与核心提供的工具，不过由于核心、核心工具与这些软件开发者的整合，使得 Linux 成为一个更完整的、功能强大的操作系统啰！约略了解 Linux 是何物之后，接下来，我们要谈一谈，『为什么说 Linux 是很稳定的操作系统呢？他是如何来的？』

---

💧Linux 之前，Unix 的历史

早在 Linux 出现之前的二十年 (大约在 1970 年代)，就有一个相当稳定而成

熟的操作系统存在了！那就是 Linux 的老大哥『Unix』是也！怎么这么说呢？他们这两个家伙有什么关系呀？这里就给他说一说啰！

众所皆知的，Linux 的核心是由 Linus Torvalds 在 1991 年的时候给他开发出来的，并且丢到网络上提供大家下载，后来大家觉得这个小东西 (Linux Kernel) 相当的小而精巧，所以慢慢的就有相当多的朋友投入这个小东西的研究领域里面去了！但是为什么这的小东西这么棒呢？然而又为什么大家都可以免费的下载这个东西呢？嗯！等鸟哥慢慢的唬 xx... 喔不！听我慢慢的道来！

- 1969 年以前：一个没有完成的梦想：Bell, MIT 与 GE 的『Multics』系统

早期的计算机并不像现在的个人计算机一样普遍，他可不是一般人碰的起的呢～除非是军事或者是高科技用途，或者是学术单位的学术研究，否则真的很难接触到。非但如此，早期的计算机架构还很难使用，除了指令周期并不快之外，操作接口也很困扰的！因为那个时候的输入设备只有卡片阅读机、输出设备只有打印机，用户也无法与操作系统互动 (批次型操作系统)。

在那个时候，写程序是件很可怜的事情，因为程序设计者，必须要将程序相关的信息在读卡纸上面打洞，然后再将读卡纸插入卡片阅读机来将信息读入主机中运算。光是这样就很麻烦了，如果程序有个小地方写错，哈哈！光是重新打卡就很惨，加上主机少，用户众多，光是等待，就耗去很多的时间了！

在那之后，由于硬件与操作系统的改良，使得后来可以使用键盘来进行信息的输入。不过，在一间学校里面，主机毕竟可能只有一部，如果多人等待使用，那怎么办？大家还是得要等待啊！好在 1960 年代初期麻省理工学院 (MIT) 发展了所谓的：『兼容分时系统 (Compatible Time-Sharing System, CTSS)』，它可以让大型主机透过提供数个终端机 (terminal) 以联机进入主机，来利用主机的资源进行运算工作。架构有点像这样：

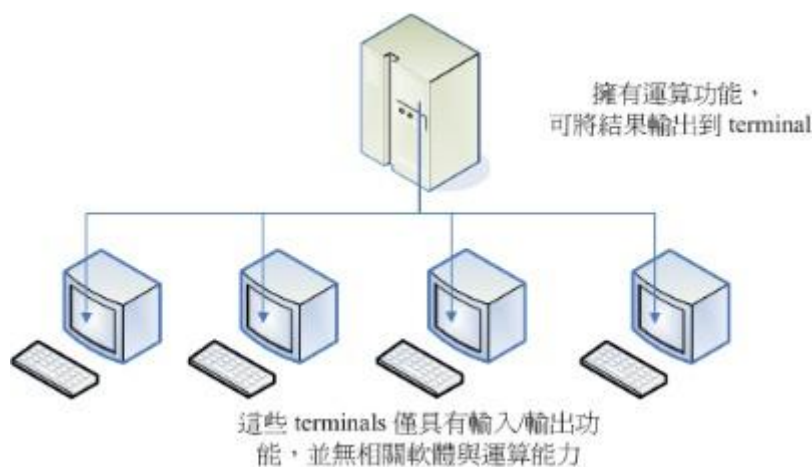


图 1.2.1、早期主机与终端机的相关性图标

### Tips:

这个兼容分时系统可以说是近代操作系统的始祖呢！他可以让多个使用者在某一段时间内分别使用 CPU 的资源，感觉上你会觉得大家是同时使用该主机的资源！事实上，是 CPU 在每个使用者的工作之间进行切换，在当时，这可是个划时代的技术喔！



如此一来，无论主机在哪里，只要在终端机前面进行输入输出的作业，就可利用主机提供的功能了。不过，需要注意的是，此时终端机只具有输入/输出的功能，本身完全不具任何运算或者软件安装的能力。而且，比较先进的主机大概也只能提供 30 个不到的终端机而已。

为了更加强化大型主机的功能，以让主机的资源可以提供更多使用者来利用，所以在 1965 年前后，由贝尔实验室(Bell)、麻省理工学院(MIT)及奇异公司(GE，或称为通用电器)共同发起了 Multics 的计划，Multics 计划的目的是想要让大型主机可以达成提供 300 个以上的终端机联机使用的目标。不过，到了 1969 年前后，计划进度落后，资金也短缺，所以该计划虽然继续在研究，但贝尔还是退出了该计划的研究工作。（注：Multics 有复杂、多数的意思存在。）

### Tips:

最终 Multics 还是有成功的发展出他们的系统，完整的历史说明可以参考：<http://www.multicians.org/>网站内容。Multics 计划虽然后来没有受到很大的重视，但是他培养出来的人才相当优秀的！^\_^



- 1969 年：Ken Thompson 的小型 file server system

在认为 Multics 计划不可能成功之后，[贝尔研究室](#)就退出该计划。不过，原本参与 Multics 计划的人员中，已经从该计划当中获得一些点子，[Ken Thompson](#) 就是其中一位！

Thompson 因为自己的需要，希望开发一个小小的操作系统以提供自己的需求。在开发时，有一部 DEC(Digital Equipment Corporation)公司推出的 PDP-7 刚好没人使用，于是他就准备针对这部主机进行操作系统核心程序的撰写。本来 Thompson 应该是没时间的(有家有小孩的宿命？)，无巧不巧的是，在 1969 年八月份左右，刚好 Thompson 的妻儿去了美西探亲，于是他有了额外的一个月的时间好好的待在家将一些构想实现出来！

经过四个星期的奋斗，他终于以汇编语言(Assembler)写出了一组核心程序，同时包括一些核心工具程序，以及一个小小的文件系统。那个系统就是 Unix 的原型！当时 Thompson 将 Multics 庞大的复杂系统简化了不少，于是同实验室的朋友都戏称这个系统为：Unics。（当时尚未有 Unix 的名称）



Thompson 的这个文件系统有两个重要的概念，分别是：

- 所有的程序或系统装置都是档案
- 不管建构编辑器还是附属档案，所写的程序只有一个目的，且要有效的完成目标。

这些概念在后来对于 Linux 的发展有相当重要的影响喔！

**Tips:**

套一句常听到的广告词：『科技始终来自于人性』，当初 Thompson 会写这套 Unix 核心程序，却是想要移植一套名为『太空旅游』的游戏呢！ ^\_^



- 1973 年: Unix 的正式诞生, Ritchie 等人以 C 语言写出第一个正式 Unix 核心

由于 Thompson 写的那个操作系统实在太好用了，所以在贝尔实验室内部广为流传，并且数度经过改版。但是因为 Unics 本来是以汇编语言写成的，而如[第零章计算器概论](#)谈到的，汇编语言具有专一性，加上当时的机器都不太相同，所以每次要安装到不同的机器都得要重新编写汇编语言，真不方便！

后来 Thompson 与 Ritchie 合作想将 Unics 改以高阶程序语言来撰写。当时现成的高阶程序语言有 B 语言。但是由 B 语言所编译出来的核心效能不是很好。后来 [Dennis Ritchie](#) 将 B 语言重新改写成 C 语言，再以 C 语言重新改写与编译 Unics 的核心，最后正名与发行出 Unix 的正式版本！

**Tips:**

这群高级黑客实在很厉害！因为自己的需求来开发出这么多好用的工具！C 程序语言开发成功后，甚至一直沿用至今呢！你说厉不厉害啊！这个故事也告诉我们，不要小看自己的潜能喔！你想作的，但是现实生活中没有的，就动手自己搞一个来玩玩吧！



由于贝尔实验室是隶属于美国电信大厂 [AT&T](#) 公司的，只是 AT&T 当时忙于其他商业活动，对于 Unix 并不支持也不排斥。此外，Unix 在这个时期的发展者都是贝尔实验室的工程师，这些工程师对于程序当然相当有研究，所以，Unix 在此时当然是不容易被一般人所接受的！

需要特别强调的是，由于 Unix 是以较高阶的 C 语言写的，相对于汇编语言需要与硬件有密切的配合，高阶的 C 语言与硬件的相关性就没有这么大了！所以，这个改变也使得 Unix 很容易被移植到不同的机器上面喔！

- 1977 年：重要的 Unix 分支：BSD 的诞生

虽然贝尔属于 AT&T，但是 AT&T 此时对于 Unix 是采取较开放的态度，此外，Unix 是以高阶的 C 语言写成的，理论上是具有可移植性的！亦即只要取得 Unix 的原始码，并且针对大型主机的特性加以修订原有的原始码(Source Code)，就可能将 Unix 移植到另一部不同的主机上头了。所以在 1973 年以后，Unix 便得以与学术界合作开发！最重要的接触就是与加州柏克莱 (Berkeley) 大学的合作了。

柏克莱大学的 Bill Joy 在取得了 Unix 的核心原始码后，着手修改成适合自己机器的版本，并且同时增加了很多工具软件与编译程序，最终将它命名为 Berkeley Software Distribution (BSD)。这个 BSD 是 Unix 很重要的一个分支，Bill Joy 也是 Unix 业者『Sun(升阳)』这家公司的创办者！Sun 公司即是以 BSD 发展的核心进行自己的商业 Unix 版本的发展的。(后来可以安装在 x86 硬件架构上面 FreeBSD 即是 BSD 改版而来！)

- 1979 年：一个措手不及的版权宣告！

由于 Unix 的高度可移植性与强大的效能，加上当时并没有版权的纠纷，所以让很多商业公司开始了 Unix 操作系统的发展，例如 AT&T 自家的 System V、IBM 的 AIX 以及 HP 与 DEC 等公司，都有推出自家的主机搭配自己的 Unix 操作系统。

但是，如同我们前面提到的，操作系统的核心(Kernel)必须要跟硬件配合，以提供及控制硬件的资源进行良好的工作！而在早期每一家生产计算机硬件的公司还没有所谓的『协议』的概念，所以每一个计算机公司出产的硬件自然就不相同啰！因此他们必须要为自己的计算机硬件开发合适的 Unix 系统。例如在学术机构相当有名的 Sun、Cray 与 HP 就是这一种情况。他们开发出来的 Unix 操作系统以及内含的相关软件并没有办法在其他的硬件架构下工作的！另外，由于没有厂商针对个人计算机设计 Unix 系统，因此，在早期并没有支持个人计算机的 Unix 操作系统的出现。

#### Tips:

如同兼容分时系统的功能一般，Unix 强调的是多人多任务的环境！但早期的 286 个人计算机架构下的 CPU 是没有能力达到多任务的作业，因此，并没有人对移植 Unix 到 x86 的计算机上有兴趣。



每一家公司自己出的 Unix 虽然在架构上面大同小异，但是却真的仅能支持自身的硬件，所以啰，早先的 Unix 只能与服务器(Server)或者是大型工作站(Workstation)划上等号！

但是这个高度开放的 Unix 系统在 1979 年有了重大的转折～因为 AT&T 由于商业的考虑，以及在当时现实环境下的思考，于是将 Unix 的版权收回去。因此，AT&T 在 1979 年发行的第七版 Unix 中，特别提到了『不可对学生提供原始码』的严格限制！同时，也造成 Unix 业界之间的紧张气氛，并且也引爆了很多的商业纠纷～

**Tips:**

AT&T 这一版开发出来的 Unix 被称为 System V，目前被称为纯种的 Unix 指的就是 System V 以及 BSD 这两套啰！



- 1984 年之一：x86 架构的 Minix 诞生

关于 1979 年的版权声明中，影响最大的当然就是学校教 Unix 核心原始码相关学问的教授了！想一想，如果没有核心原始码，那么如何教导学生认识 Unix 呢？这问题对于 Andrew Tanenbaum(谭宁邦)教授来说，实在是很伤脑筋的！不过，学校的课程还是得继续啊！那怎么办？

既然 1979 年的 Unix 第七版可以在 Intel 的 x86 架构上面进行移植，那么是否意味着可以将 Unix 改写并移植到 x86 上面了呢？在这个想法上，谭宁邦教授于是乎自己动手写了 Minix 这个 Unix Like 的核心程序！在撰写的过程中，为了避免版权纠纷，谭宁邦完全不看 Unix 核心原始码！并且强调他的 Minix 必须能够与 Unix 兼容才行！谭宁邦在 1984 年开始撰写核心程序，到了 1986 年终于完成，并于次年出版 Minix 相关书籍，同时与新闻组(BBS 及 News)相结合～

**Tips:**

之所以称为 Minix 的原因，是因为他是个 Mini 的 Unix 系统啰！^\_^



这个 Minix 版本比较有趣的地方是，他并不是完全免费的，无法在网络上提供下载！必须要透过磁盘/磁带购买才行！虽然真的很便宜～不过，毕竟因为没有在网络上流传，所以 Minix 的传递速度并没有很快速！此外，购买时，随磁盘还会附上 Minix 的原始码！这意味着使用者可以学习 Minix 的核心程序设计概念喔！（这个特色对于 Linux 的启始开发阶段，可是有很大的关系喔！）

此外，Minix 操作系统的开发者仅有谭宁邦教授，因为学者很忙啊！加上谭宁邦始终认为 Minix 主要用在教育用途上面，所以对于 Minix 是点到为止！没错，Minix 是很受欢迎，不过，使用者的要求/需求的声音可能就比较没有办法上升到比较高的地方了！这样说，你明白吧？^\_^

- 1984 年之二：GNU 计划与 FSF 基金会的成立

Richard Mathew Stallman(史托曼)在 1984 年发起的 GNU 计划，对于现今的自由软件风潮，真有不可磨灭的地位！目前我们所使用得很多自由软件，几乎均直接或间接帮助于 GNU 这个计划呢！那么史托曼是何许人也？为何他会



发起这个 GNU 计划呢？

- 一个分享的环境：

[Richard Mathew Stallman](#) (生于 1953 年，网络上自称的 ID 为 RMS) 从小就很聪明！他在 1971 年的时候，进入黑客圈中相当出名的人工智能实验室 (AI Lab.)，这个时候的黑客专指计算机功力很强的人，而非破坏计算机的怪客 (cracker) 喔！

当时的黑客圈对于软件的着眼点几乎都是在『分享』，所以并没有专利方面的困扰！这个特色对于史托曼的影响很大！不过，后来由于管理阶层的问题，导致实验室的优秀黑客离开该实验室，并且进入其他商业公司继续发展优秀的软件。但史托曼并不服输，仍然持续在原来的实验室开发新的程序与软件。后来，他发现到，自己一个人并无法完成所有的工作，于是想要成立一个开放的团体来共同努力！

- 使用 Unix 开发阶段：

1983 年以后，因为实验室硬件的更换，使得史托曼无法继续以原有的硬件与操作系统继续自由程序的撰写～而且他进一步发现到，过去他所使用的 Lisp 操作系统，是麻省理工学院的专利软件，是无法共享的，这对于想要成立一个开放团体的史托曼是个阻碍。于是他便放弃了 Lisp 这个系统。后来，他接触到 Unix 这个系统，并且发现，Unix 在理论与实际上，都可以在不同的机器间进行移植。于是他开始转而使用 Unix 系统。

因为 Lisp 与 Unix 是不同的系统，所以，他原本已经撰写完毕的软件是无法在 Unix 上面运行的！为此，他就开始将软件移植到 Unix 上面。并且，为了让软件可以在不同的平台上运作，因此，史托曼将他发展的软件均撰写成可以移植的型态！也就是他都会将程序的原始码公布出来！

- GNU 计划的推展：

1984 年，史托曼开始 [GNU](#) 计划，这个计划的目的是想要：建立一个自由的开放的 Unix 操作系统 (Free Unix)。但是建立一个操作系统谈何容易啊！而且在当时的 GNU 是仅有自己一个人单打独斗的史托曼～这实在太麻烦，但又不能不做这个计划，于是史托曼反其道而行～『既然操作系统太复杂，我就先写可以在 Unix 上面运行的小程序，这总可以了吧？呵呵！』在这个想法上，史托曼便开始了程序的写作。在写作期间，为了不让自己吃上官司，他绝对不看专利软件的原始码！为了这个计划，他开始使用原本 Unix 上面跑的软件，并自行撰写功能与 Unix 原有专利软件相仿的软件。

但不论是什么软件，都得要进行编译成为二进制文件 (binary file) 后才能够执行，因此他便开始撰写 C 语言的编译程序，那就是现在相当有名的 GNU C Compiler (gcc)！这个点相当的重要！这是因为 C 语言编译程序版本众多，但都是专利软件，如果他写的 C 编译程序够棒，效能够佳，那么将会大大的

让 GNU 计划出现在众人眼前！

**Tips:**

计算机仅认识 0/1 的数据，但是人类不认识啊！人类对于纯文本的数据(就是所谓的 ASCII 文件格式)比较有感觉。但是偏偏计算机又不认识 ASCII 格式的文字，很头痛，不是吗？后来就有所谓的『编译程序』来辅助我们。

我们一般使用文本编辑器以 ASCII 纯文本格式来撰写程序，再透过所谓的『编译程序(compiler)』将刚刚完成的文本文件『编译』成为计算机认识的二进制制(binary file)的档案，以让计算机认识且可以执行的程序啊！



但开始时并不顺利，为此，他先转而将 Emacs 编辑器写成可以在 Unix 上面跑的软件，并公开公布原始码。Emacs 是一种程序编辑器，他可以在用户撰写程序的过程中就进行程序语法的检验，此一功能可以减少程序设计师除错的时间！因为 Emacs 太优秀了，因此，很多人便直接向他购买。

此时因特网尚未流行，所以，史托曼便借着 Emacs 以磁带(tape)出售，赚了一点钱，进而开始全力撰写其他软件。并且成立自由软件基金会(FSF, Free Software Foundation)，请更多工程师与志工撰写软件。终于还是完成了 GCC，这比 Emacs 还更有帮助！此外，他还撰写了更多可以被呼叫的 C 函式库(GNU C library)，以及可以被使用来操作操作系统的基本接口 BASH shell！这些都在 1990 年左右完成了！

**Tips:**

如果纯粹使用文本编辑器来编辑程序的话，那么程序语法如果写错时，只能利用编译时发生的错误讯息来修订了，这样实在很没有效率。Emacs 则是一个很棒的编辑器！注意！是编辑(editor)而非编译(compiler)！他可以很快的立刻显示出你写入的语法可能有错误的地方，这对于程序设计师来说，实在是一个好到不能再好的工具了！所以才会这么的受欢迎啊！



- GNU 的通用公共许可证：

到了 1985 年，为了避免 GNU 所开发的自由软件被其他人所利用而成为专利软件，所以他与律师草拟了有名的通用公共许可证(General Public License, GPL)，并且称呼他为 copyleft(相对于专利软件的 copyright!)。关于 GPL 的相关内容我们在下一个小节继续谈论，在这里，必须要说明的是，由于有 GNU 所开发的几个重要软件，如：

- Emacs
- GNU C (GCC)
- GNU C Library (glibc)
- Bash shell

造成后来很多的软件开发者可以藉由这些基础的工具来进行程序开发！进一步壮大了自由软件团体！这是很重要的！不过，对于 GNU 的最初构想『建立一个自由的 Unix 操作系统』来说，有这些优秀的程序是仍无法满足，因为，当下并没有『自由的 Unix 核心』存在...所以这些软件仍只能在那些有专利的 Unix 平台上工作~~一直到 Linux 的出现...更多的 FSF 开发的软件可以参考如下网页：

- <https://www.fsf.org/resources>
- 1988 年：图形接口 XFree86 计划

有鉴于图形用户接口(Graphical User Interface, GUI) 的需求日益加重，在 1984 年由 MIT 与其他第三方首次发表了 X Window System，并且更在 1988 年成立了非营利性质的 XFree86 这个组织。所谓的 XFree86 其实是 X Window System + Free + x86 的整合名称呢！而这个 XFree86 的 GUI 界面更在 Linux 的核心 1.0 版于 1994 年释出时，整合于 Linux 操作系统当中！

#### Tips:

为什么称图形用户接口为 X 呢？因为由英文单字来看，Window 的 W 接的就是 X 啦！意指 Window 的下一版就是了！需注意的是，X Window 并不是 X Windows 喔！



- 1991 年：芬兰大学生 Linus Torvalds 的一则简讯

到了 1991 年，芬兰的赫尔辛基大学的 Linus Torvalds 在 BBS 上面贴了一则消息，宣称他以 bash, gcc 等工具写了一个小小的核心程序，这个核心程序可以在 Intel 的 386 机器上面运作，让很多人很感兴趣！从此开始了 Linux 不平凡的路程！

---

## 🐼 关于 GNU 计划

- 自由软件的活动：

1984 年创立 GNU 计划与 FSF 基金会的 Stallman 先生认为，写程序最大的快乐就是让自己发展的良好的软件让大家来使用了！而既然程序是想要分享给大家使用的，不过，每个人所使用的计算机软硬件并不相同，既然如此的话，那么该程序的原始码(Source code)就应该要同时释出，这样才能方便大家

修改而适用于每个人的计算机中呢！这个将原始码连同软件程序释出的举动，就称为自由软件(Free Software)运动！

此外，史托曼同时认为，如果你将你程序的 Source code 分享出来时，若该程序是很优秀的，那么将会有很多人使用，而每个人对于该程序都可以查阅 source code，无形之中，就会有一票人帮你除错啰！你的这支程序将会越来越壮大！越来越优秀呢！

- 自由软件的版权 GNU GPL:

而为了避免自己的开发出来的 Open source 的自由软件被拿去做成专利软件，于是 Stallman 同时将 GNU 与 FSF 发展出来的软件，都挂上 GPL 的版权宣告～这个 FSF 的核心观念是『版权制度是促进社会进步的手段，版权本身不是自然权力。』对于 FSF 有兴趣或者对于 GNU 想要更深入的了解时，请参考[朝阳科技大学洪朝贵教授](http://people.ofset.org/~ckhung/a/c_83.php)的网站

[http://people.ofset.org/~ckhung/a/c\\_83.php](http://people.ofset.org/~ckhung/a/c_83.php)，或直接到 GNU 去：  
<http://www.gnu.org> 里面有更为深入的解说！

#### Tips:

为什么要称为 GNU 呢？其实 GNU 是 GNU's Not Unix 的缩写，意思是说，GNU 并不是 Unix 啊！那么 GNU 又是什么呢？就是 GNU's Not Unix 嘛！.....如果你写过程序就会知道，这个 GNU = GNU's Not Unix 可是无穷循环啊！忙碌～



另外，什么是 Open Source 呢？所谓的 source 是程序发展者写出的源代码，Open Source 就是，软件在发布时，同时将作者的原始码一起公布的意思！

- 自由(Free)的真谛:

那么这个 GPL (GNU General Public License, GPL) 是什么玩意儿？为什么要把自由软件挂上 GPL 的『版权宣告』呢？这个版权宣告对于作者有何好处？首先，Stallman 对 GPL 一直是强调 Free 的，这个 Free 的意思是这样的：

"Free software" is a matter of liberty, not price. To understand the concept, you should think of "free speech", not "free beer". "Free software" refers to the users' freedom to run, copy, distribute, study, change, and improve the software

大意是说，Free Software(自由软件)是一种自由的权力，并非是『价格！』举例来说，你可以拥有自由呼吸的权力、你拥有自由发表言论的权力，但是，这并不代表你可以到处喝『免费的啤酒！(free beer)』，也就是说，自由软件的重点并不是指『免费』的，而是指具有『自由度，freedom』的软件，史托曼进一步说明了自由度的意义是：使用者可以自由的执行、复制、再发

行、学习、修改与强化自由软件。

这无疑是个好消息！因为如此一来，你所拿到的软件可能原先只能在 Unix 上面跑，但是经过原始码的修改之后，你将可以拿他在 Linux 或者是 Windows 上面来跑！总之，一个软件挂上了 GPL 版权宣告之后，他自然就成了自由软件！这个软件就具有底下的特色：

- 取得软件与原始码：你可以根据自己的需求来执行这个自由软件；
- 复制：你可以自由的复制该软件；
- 修改：你可以将取得的原始码进行程序修改工作，使之适合你的工作；
- 再发行：你可以将你修改过的程序，再度的自由发行，而不会与原先的撰写者冲突；
- 回馈：你应该将你修改过的程序代码回馈于社群！

但请特别留意，你所修改的任何一个自由软件都不应该也不能这样：

- 修改授权：你不能将一个 GPL 授权的自由软件，在你修改后而将他取消 GPL 授权～
- 单纯贩卖：你不能单纯的贩卖自由软件。

也就是说，既然 GPL 是站在互助互利的角度上去开发的，你自然不应该将大家的成果占为己有，而取消 GPL 授权的！对吧！因此你当然不可以将一个 GPL 软件的授权取消，即使你已经对该软件进行大幅度的修改！那么自由软件也不能贩卖吗？当然不是！还记得上一个小节里面，我们提到史托曼藉由贩卖 Emacs 取得一些经费，让自己生活不至于匮乏吧？是的！自由软件是可以贩卖的，不过，不可仅贩卖该软件，应同时搭配售后服务与相关手册～这些可就需要工本费了呢！

- 自由软件与商业行为：

很多人还是有疑问，目前不是有很多 Linux 开发商吗？为何他们可以贩卖 Linux 这个 GPL 授权的软件？原因很简单，因为他们大多都是贩卖『售后服务！』所以，他们所使用的自由软件，都可以在他们的网站上面下载！（当然，每个厂商他们自己开发的工具软件就不是 GPL 的授权软件了！）但是，你可以购买他们的 Linux 光盘，如果你购买了光盘，他们会提供相关的手册说明文件，同时也会提供你数年不等的咨询、售后服务、软件升级与其他协力工作等等的附加价值！

所以说，目前自由软件工作者，他们所赖以维生的，几乎都是在『服务』这个领域呢！毕竟自由软件并不是每个人都会撰写，有人有需要你的自由软件时，他就会请求你的协助，此时，你就可以透过服务来收费了！这样说，自由软件确实还是具有商业空间的喔！



### Tips:

很多人对于 GPL 授权一直很疑惑，对于 GPL 的商业行为更是无法接受！关于这一点，鸟哥在这里还是要再次的申明，GPL 是可以从事商业行为的！而很多的作者也是藉由这些商业行为来得以取得生活所需，更进一步去发展更优秀的自由软件！千万不要听到『商业』就排斥！这对于发展优良软件的朋友来说，是不礼貌的！



上面提到的大多是与用户有关的项目，那么 GPL 对于自由软件的作者有何优点呢？大致的优点有这些：

- 软件安全性较佳；
- 软件执行效能较佳；
- 软件除错时间较短；
- 贡献的原始码永远都存在。

这是因为既然是 Open Source 的自由软件，那么你的程序代码将会有很多人帮你查阅，如此一来，程序的漏洞与程序的优化将会进展的很快！所以，在安全性与效能上面，自由软件一点都不输给商业软件喔！此外，因为 GPL 授权当中，修改者并不能修改授权，因此，你如果曾经贡献过程序代码，嘿嘿！你将名留青史呢！不错吧！ ^\_^

不过，就鸟哥的观点来看，GPL 对于程序开发者的优点是相当多的，不过，对于不熟悉程序的一般人来说，GPL 的优点其实不太容易看出来～首先，虽然他是随手可得自由软件，不过，你也必须要会使用基本的编译程序才行吧！（呵呵！这也是你为何要察看鸟哥的网站的原因吧！ ^\_^）这对于一般人来说并不容易！当然啦，如果每个人都跟 Stallman 一样神，那商业公司就不用存在啦！嘿嘿！对于不懂程序的人来说，商业公司是一个很快速的解决之道啊！而对于我们广大的读者群来说，认识了/学习了 Linux 与自由软件的相关技巧后，对于未来真的是有很不错的帮助喔！



## Torvalds 的 Linux 发展

我们前面一节当中，提到了 Unix 的历史，也提到了 Linux 是由 Torvalds 这个芬兰人所发明的。那么为何托瓦兹可以发明 Linux 呢？凭空想象而来的？还是有什么渊源？这里我们就来谈一谈啰！



## 与 Minix 之间

[Linus Torvalds](#) (托瓦兹，1969 年出生) 的外祖父是赫尔辛基大学的统计学家，他的外祖父为了让自己的小孙子能够学点东西，所以从小就将托瓦兹带

到身边来管理一些微计算机。在这个时期,托瓦兹接触了汇编语言(Assembly Language),那是一种直接与芯片对谈的程序语言,也就是所谓的低级语言。必须要很了解硬件的架构,否则很难以汇编语言撰写程序的。

在 1988 年间,托瓦兹顺利的进入了赫尔辛基大学,并选读了计算机科学系。在就学期间,因为学业的需要与自己的兴趣,托瓦兹接触到了 Unix 这个操作系统。当时整个赫尔辛基只有一部最新的 Unix 系统,同时仅提供 16 个终端机(terminal)。还记得我们上一节刚刚提过的,早期的计算机仅有主机具有运算功能,terminal 仅负责提供 Input/Output 而已。在这种情况下,实在很难满足托瓦兹的需求,因为.....光是等待使用 Unix 的时间,就很耗时~为此,他不禁想到:『我何不自己搞一部 Unix 来玩?』不过,就如同 Stallman 当初的 GNU 计划一样,要写核心程序,谈何容易~

不过,幸运之神并未背离托瓦兹,因为不久之后,他就知道有一个类似 Unix 的系统,并且与 Unix 完全兼容,还可以在 Intel 386 机器上面跑的操作系统,那就是我们上一节提过的,谭宁邦教授为了教育需要而撰写的 Minix 系统!他在购买了最新的 Intel 386 的个人计算机后,就立即安装了 Minix 这个操作系统。另外,由于 Minix 这个操作系统是有附上原始码的~所以,托瓦兹也经由这个原始码学习到了很多的核心程序设计的设计概念喔!

---

## 💡对 386 硬件的多任务测试

事实上,托瓦兹对于个人计算机的 CPU 其实并不满意,因为他之前碰的计算机都是工作站型的计算机,这类计算机的 CPU 特色就是可以进行『多任务处理』的能力。什么是多任务呢?理论上,一个 CPU 在一个时间内仅能进行一个程序,那如果有两个以上的程序同时出现到系统中呢?举例来说,你可以在现今的计算机中同时开启两个以上的办公软件,例如电子表格与文字处理软件。这个同时开启的动作代表着这两个程序同时要交给 CPU 来处理~

啊!CPU 一个时间点内仅能处理一个程序,那怎么办?没关系,这个时候如果具有多任务的 CPU 就会在不同的程序间切换~还记得前一章谈到的 CPU 频率吧?假设 CPU 频率为 1GHz 的话,那表示 CPU 一秒钟可以进行  $10^9$  次工作。假设 CPU 对每个程序都只进行 1000 次运作周期,然后就得要切换到下个程序的话,那么 CPU 一秒钟就能够切换  $10^6$  次呢!(当然啦,切换工作这件事情也会花去一些 CPU 时间。不过这里暂不讨论)。这么快的处理速度下,你会发现,两个程序感觉上几乎是同步在进行啦!

### Tips:

为什么有的时候我同时开两个档案(假设为 A, B 档案)所花的时间,要比开完 A 再去开 B 档案的时间还要多?现在是否稍微可以理解?因为如果同时开启的话, CPU 就必须要在两个工作之间不停的切换~而切换的动作



还是会耗去一些 CPU 时间的！所以啰，同时启用两个以上的工作在一个 CPU 上， 要比一个一个的执行还要耗时一点。这也是为何现在 CPU 开发商要整合两个 CPU 于一个芯片中！ 也是为何在运作情况比较复杂的服务器上，需要比较多的 CPU 负责的原因！

早期 Intel x86 架构计算机不是很受重视的原因，就是因为 x86 的芯片对于多任务的处理不佳， CPU 在不同的工作之间切换不是很顺畅。但是这个情况在 386 计算机推出后，有很大的改善。托瓦兹在得知新的 386 芯片的相关信息后，他认为，以性能价格比的观点来看， Intel 的 386 相当的便宜，所以在性能上也就稍微可以将就将就 ^\_^。最终他就贷款去买了一部 Intel 的 386 来玩。

早期的计算机效能没有现在这么好，所以压榨计算机效能就成了工程师的一项癖好！托瓦兹本人早期是玩汇编语言的，汇编语言对于硬件有很密切的关系，托瓦兹自己也说：『我始终是个性能癖』^\_^。为了彻底发挥 386 的效能，于是托瓦兹花了不少时间在测试 386 机器上！他的重要测试就是在测试 386 的多功效能。首先，他写了三个小程序，一个程序会持续输出 A、一个会持续输出 B， 最后一个会将两个程序进行切换。他将三个程序同时执行，结果，他看到屏幕上很顺利的一直出现 ABABABAB..... 他知道，他成功了！

^\_  
—

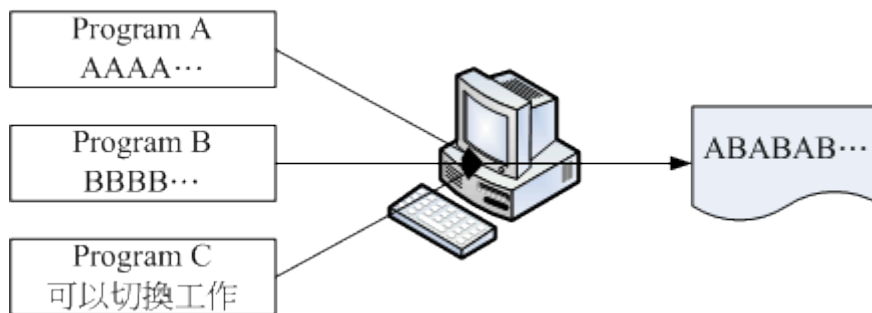


图 2.2.1、386 计算机的多任务测试

#### Tips:

要达到多任务(multitasking)的环境，除了硬件(主要是 CPU)需要能够具有多任务的特性外， 操作系统也需要支持这个功能喔！一些不具有多任务特性的操作系统， 想要同时执行两个程序是不可能的。除非先被执行的程序执行完毕，否则， 后面的程序不可能被主动执行。至于多任务的操作系统中，每个程序被执行时，都会有一个最大 CPU 使用时间，若该工作运作的时间超过这个 CPU 使用时间时， 该工作就会先被丢出 CPU 的运作中，而再度的进入核心工作排程中，等待下一次的 CPU 运作。



这有点像在开记者会啦，主持人(CPU)会问『谁要发

问』？一群记者(工作程序)就会举手(看谁的工作重要！)，先举手的自然就被允许发问，问完之后，主持人又会问一次谁要发问，当然，所有人(包括刚刚那个记者)都可以举手！如此一次一次的将工作给他完成啊！ ^\_^ 多任务的环境对于复杂的工作情况，帮助很大喔！

---

## 初次释出 Linux 0.02

探索完了 386 的硬件之后，终于拿到 Minix 并且安装在托瓦兹的 386 计算机上之后，托瓦兹跟 BBS 上面一堆工程师一样，他发现 Minix 虽然真的很棒，但是谭宁邦教授就是不愿意进行功能的加强，导致一堆工程师在操作系统功能上面的欲求不满！这个时候年轻的托瓦兹就想：『既然如此，那我何不自己来改写一个我想要的操作系统？』于是他就开始了核心程序的撰写了。

撰写程序需要什么呢？首先需要的是能够进行工作的环境，再来则是可以将原始码编译成为可执行文件的编译程序。好在有 GNU 计划提供的 bash 工作环境软件以及 gcc 编译程序等自由软件，让托瓦兹得以顺利的撰写核心程序。他参考 Minix 的设计理念与书上的程序代码，然后仔细研究出 386 个人计算机的效能优化，然后使用 GNU 的自由软件将核心程序代码与 386 紧紧的结合在一起，最终写出他所需要的核心程序。而这个小玩意竟然真的可以在 386 上面顺利的跑起来～还可以读取 Minix 的文件系统。真是太好了！不过还不够，他希望这个程序可以获得大家的一些修改建议，于是他便将这个核心放置在网络上提供大家下载，同时在 BBS 上面贴了一则消息：

```
Hello everybody out there using minix-
I'm doing a (free) operation system (just a hobby,
won't be big and professional like gnu) for 386(486)
AT clones.

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 want. Any suggestions are
welcome,
but I won't promise I'll implement them :-)
```

他说，他完成了一个小小的操作系统，这个核心是用在 386 机器上的，同时，他真的仅是好玩，并不是想要做一个跟 GNU 一样大的计划！另外，他希望能够得到更多人的建议与回馈来发展这个操作系统！这个概念跟 Minix 刚好背道而驰呢！这则新闻引起很多人的注意，他们也去托瓦兹提供的网站上下载

了这个核心来安装。有趣的是，因为托瓦兹放置核心的那个 FTP 网站的目录为：Linux，从此，大家便称这个核心为 Linux 了。（请注意，此时的 Linux 就是那个 kernel 喔！另外，托瓦兹所丢到该目录下的第一个核心版本为 0.02 呢！）

同时，为了让自己的 Linux 能够兼容于 Unix 系统，于是托瓦兹开始将一些能够在 Unix 上面运作的软件拿来在 Linux 上面跑。不过，他发现到有很多的软件无法在 Linux 这个核心上运作。这个时候他有两种作法，一种是修改软件，让该软件可以在 Linux 上跑，另一种则是修改 Linux，让 Linux 符合软件能够运作的规范！由于 Linux 希望能够兼容于 Unix，于是托瓦兹选择了第二个作法『修改 Linux』！为了让所有的软件都可以在 Linux 上执行，于是托瓦兹开始参考标准的 POSIX 规范。

#### Tips:

POSIX 是可携式操作系统接口 (Portable Operating System Interface) 的缩写，重点在规范核心与应用程序之间的接口，这是由美国电器与电子工程师学会 (IEEE) 所发布的一项标准喔！



这个正确的决定让 Linux 在起步的时候体质就比别人优良～因为 POSIX 标准主要是针对 Unix 与一些软件运行时候的标准规范，只要依据这些标准规范来设计的核心与软件，理论上，就可以搭配在一起执行了。而 Linux 的发展就是依据这个 POSIX 的标准规范，Unix 上面的软件也是遵循这个规范来设计的，如此一来，让 Linux 很容易就与 Unix 兼容共享互有的软件了！同时，因为 Linux 直接放置在网络下，提供大家下载，所以在流通的速度上相当的快！导致 Linux 的使用率大增！这些都是造成 Linux 大受欢迎的几个重要因素呢！

---

### Linux 的发展：虚拟团队的产生

- 单一个人维护阶段

Linux 虽然是托瓦兹发明的，而且内容还绝不会涉及专利软件的版权问题。不过，如果单靠托瓦兹自己一个人的话，那么 Linux 要茁壮实在很困难～因为一个人的力量是很有限的。好在托瓦兹选择 Linux 的开发方式相当的务实！首先，他将释出的 Linux 核心放置在 FTP 上面，并请告知大家新的版本信息，等到用户下载了这个核心并且安装之后，如果发生问题，或者是由于特殊需求亟需某些硬件的驱动程序，那么这些使用者就会主动回报给托瓦兹。在托瓦兹能够解决的问题范围内，他都能很快速的进行 Linux 核心的更新与除错。

- 广大黑客志工加入阶段

不过，托瓦兹总是有些硬件无法取得的啊，那么他当然无法帮助进行驱动程



序的撰写与相关软件的改良。这个时候，就会有些志工跳出来说：『这个硬件我有，我来帮忙写相关的驱动程序。』因为 Linux 的核心是 Open Source 的，黑客志工们很容易就能够跟随 Linux 的原本设计架构，并且写出兼容的驱动程序或者软件。志工们写完的驱动程序与软件托瓦兹是如何看待的呢？首先，他将该驱动程序/软件带入核心中，并且加以测试。只要测试可以运行，并且没有什么主要的大问题，那么他就会很乐意的将志工们写的程序代码加入核心中！

总之，托瓦兹是个很务实的人，对于 Linux 核心所欠缺的项目，他总是『先求有且能跑，再求进一步改良』的心态！这让 Linux 使用者与志工得到相当大的鼓励！因为 Linux 的进步太快了！用户要求虚拟内存，结果不到一个星期推出的新版 Linux 就有了！这不得不让人佩服啊！

另外，为因应这种程序代码的加入，于是 Linux 便逐渐发展成具有模块的功能！亦即是将某些功能独立出于核心外，在需要的时候才加载到核心中。如此一来，如果有新的硬件驱动程序或者其他协议的程序代码进来时，就可以模块化，大大的增加了 Linux 核心的可维护能力！

#### Tips:

核心是一组程序，如果这组程序每次加入新的功能都要重新编译与改版的话会变成如何？想象一下，如果你只是换了显示适配器就得要重新安装新的 Windows 操作系统，会不会傻眼？模块化之后，原本的核心程序不需要更动，你可以直接将他想成是『驱动程序』即可！ ^\_^



#### • 核心功能细部分工发展阶段

后来，因为 Linux 核心加入了太多的功能，光靠托瓦兹一个人进行核心的实际测试并加入核心原始程序实在太费力～结果，就有很多的朋友跳出来帮忙这个前置作业！例如考克斯(Alan Cox)、与崔迪(Stephen Tweedie)等等，这些重要的副手会先将来自志工们的修补程序或者新功能的程序代码进行测试，并且结果上传给托瓦兹看，让托瓦兹作最后核心加入的原始码的选择与整并！这个分层负责的结果，让 Linux 的发展更加的容易！

特别值得注意的是，这些托瓦兹的 Linux 发展副手，以及自愿传送修补程序的黑客志工，其实都没有见过面，而且彼此在地球的各个角落，大家群策群力的共同发展出现今的 Linux，我们称这群人为虚拟团队！而为了虚拟团队数据的传输，于是 Linux 便成立的核心网站：<http://www.kernel.org>！

而这群素未谋面的虚拟团队们，在 1994 年终于完成的 Linux 的核心正式版！version 1.0。这一版同时还加入了 X Window System 的支持呢！更于 1996 年完成了 2.0 版。此外，托瓦兹指明了企鹅为 Linux 的吉祥物。

### Tips:

奇怪的是，托瓦兹是因为小时候去动物园被企鹅咬了一口念念不忘，而正式的 2.0 推出时，大家要他想一个吉祥物。他在想也想不到什么动物的情况下，就将这个念念不忘的企鹅当成了 Linux 的吉祥物了.....



Linux 由于托瓦兹是针对 386 写的，跟 386 硬件的相关性很强，所以，早期的 Linux 确实是不具有移植性的。不过，大家知道 Open source 的好处就是，可以修改程序代码去适合作业的环境。因此，在 1994 年以后，Linux 便被开发到很多的硬件上面去了！目前除了 x86 之外，IBM、HP、Sun 等等公司出的硬件也都有被 Linux 所支持呢！

- Linux 的核心版本之谜

Linux 的核心版本编号有点类似如下的样子：

2 . 6 . 20

主版本.次版本.释出次数

如前所述，因为对于 Linux 核心的开发者太多了，以致于造成 Linux 核心经常性的变动。但对于一般家庭计算机或企业关键应用的话，常变动的核心并不适合的。因此托瓦兹便将核心的发展趋势分为两股，并根据这两股核心的发展分别给予不同的核心编号，那就是：

- 主、次版本为奇数：  
如 2.5.xx，这种核心版本主要用在测试与发展新功能，所以通常这种版本仅有核心开发工程师会使用。如果有新增的核心程序代码，会加到这种版本当中，等到众多工程师测试没问题后，才加入下一版的稳定核心中；
- 主、次版本为偶数：  
如 2.6.xx，等到核心功能发展成熟后会加到这类的版本中，主要用在一般家庭计算机以及企业版本中。重点在于提供使用者一个相对稳定的 Linux 作业环境平台。

Linux 核心版本与 distribution 的版本并不相同，很多朋友常常上网问到：『我的 Linux 是 9.x 版，请问....』之类的留言，这是不对的提问方式，因为所谓的 Linux 版本指的应该是核心版本，而目前最新的核心版本应该是 2.6.26(2008/08) 才对，并不会有 9.x 的版本出现的。

妳常用的 Linux 系统则应该说明为 distribution 才对！因此，如果以 CentOS 这个 distribution 来说，妳应该说：『我用的 Linux 是 CentOS 这个 distribution，版本为 5.x 版，请问....』才对喔！

### Tips:

当妳有任何问题想要在 Linux 论坛发言时, 请务必仔细的说明妳的 distribution 版本, 因为虽然各家 distributions 使用的都是 Linux 核心, 不过每家 distributions 所选用的软件以及他们自己发展的工具并不相同, 多少还是有点差异, 所以留言时得要先声明 distribution 的版本才行喔! ^\_^



## Linux distributions

好了, 经过上面的说明, 我们知道了 Linux 其实就是一个操作系统最底层的核心及其提供的核心工具。他是 GNU GPL 授权模式, 所以, 任何人均可取得原始码与可执行这个核心程序, 并且可以修改。此外, 因为 Linux 参考 POSIX 设计规范, 于是兼容于 Unix 操作系统, 故亦可称之为 Unix Like 的一种。

### Tips:

鸟哥曾在上课的时候问过同学: 『什么是 Unix Like 啊?』可爱的同学们回答的答案是: 『就是很喜欢 (like) Unix 啦!』囧 rz... 那个 like 是『很像』啦! 所以 Unix like 是『很像 Unix 的操作系统』哩!



### • 可完全安装的 Linux 发布套件

Linux 的出现让 GNU 计划放下了心里的一块大石头, 因为 GNU 一直以来就是缺乏了核心程序, 导致他们的 GNU 自由软件只能在其他的 Unix 上面跑。既然目前有 Linux 出现了, 且 Linux 也用了很多的 GNU 相关软件, 所以 Stallman 认为 Linux 的全名应该称之为 GNU/Linux 呢! 不管怎么说, Linux 实在很不错, 让 GNU 软件大多以 Linux 为主要操作系统来进行开发, 此外, 很多其他的自由软件团队, 例如 sendmail, wu-ftp, apache 等等也都有以 Linux 为开发测试平台的计划出现! 如此一来, Linux 除了主要的核心程序外, 可以在 Linux 上面运行的软件也越来越多, 如果有心, 就能够将一个完整的 Linux 操作系统搞定了!

虽然由 Torvalds 负责开发的 Linux 仅具有 Kernel 与 Kernel 提供的工具, 不过, 如上所述, 很多的软件已经可以在 Linux 上面运作了, 因此, Linux + 各家软件就可以完成一个相当完整的操作系统了。不过, 要完成这样的操作系统..... 还真难~ 因为 Linux 早期都是由黑客工程师所开发维护的, 他们并没有考虑到一般使用者的能力.....

为了让使用者能够接触到 Linux, 于是很多的商业公司或非营利团体, 就将 Linux Kernel (含 tools) 与可运行的软件整合起来, 加上自己具有创意的工具程序, 这个工具程序可以让用户以光盘/DVD 或者透过网络直接安装/管理 Linux 系统。这个 Kernel + Softwares + Tools 的可完全安装的咚咚, 我

们称之为 Linux distribution，一般中文翻译成可完全安装套件，或者 Linux 发布商套件等。

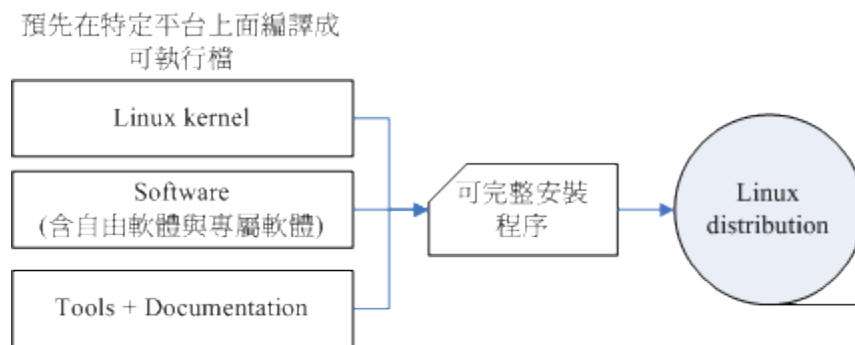


图 2.5.1、Linux 可完全安装发布套件

#### Tips:

由于 Linux 核心是由黑客工程师写的，要由原始码安装到 x86 计算机上面成为可以执行的 binary 档案，这个过程可不是人人都会的～所以早期确实只有工程师对 Linux 有兴趣。一直到一些社群与商业公司将 Linux 核心配合自由软件，并提供完整的安装程序，且制成光盘/DVD 后，对于一般使用者来说，Linux 才越来越具有吸引力！因为只要一直『下一步』就可以将 Linux 安装完成啊！ ^\_^



由于 GNU 的 GPL 授权并非不能从事商业行为，于是很多商业公司便成立来贩卖 Linux distribution。而由于 Linux 的 GPL 版权宣告，因此，商业公司所贩卖的 Linux distributions 通常也都可以从 Internet 上面来下载的！此外，如果你想要其他商业公司的服务，那么直接向该公司购买光盘来安装，也是一个很不错的方式的！

- 各大 Linux Distributions 的主要异同：

不过，由于发展 Linux distributions 的社群与公司实在太多了，例如在台湾有名的 Red Hat, SuSE, Ubuntu, Fedora, Debian 等等，所以很多人都很担心，如此一来每个 distribution 是否都不相同呢？这就不需要担心了，因为每个 Linux distributions 使用的 kernel 都是 <http://www.kernel.org> 所释出的，而他们所选择的软件，几乎都是目前很知名的软件，重复性相当的高，例如网页服务器的 Apache，电子邮件服务器的 Postfix/sendmail，文件服务器的 Samba 等等。

此外，为了让所有的 Linux distributions 开发不致于差异太大，还有 Linux Standard Base (LSB) 等标准来规范开发者，以及目录架构的 File system Hierarchy Standard (FHS) 标准规范！唯一差别的，可能就是该开发者自家所开发出来的管理工具，以及套件管理的模式吧！所以说，基本上，每个 Linux distributions 除了架构的严谨度与选择的套件内容外，其实差异并

不太大啦！ ^\_^ 。大家可以选择自己喜好的 distribution 来安装即可！

**Tips:**

事实上鸟哥认为 distributions 主要分为两大系统，一种是使用 RPM 方式安装软件的系统，包括 Red Hat, Fedora, SuSE 等都是这类；一种则是使用 Debian 的 dpkg 方式安装软件的系统，包括 Debian, Ubuntu, B2D 等等。



底下列出几个主要的 Linux distributions 发行者网址：

- Red Hat: <http://www.redhat.com>
- Fedora: <http://fedoraproject.org/>
- Mandriva: <http://www.mandriva.com>
- Novell SuSE: <http://www.novell.com/linux/>
- Debian: <http://www.debian.org/>
- Slackware: <http://www.slackware.com/>
- Gentoo: <http://www.gentoo.org/>
- Ubuntu: <http://www.ubuntu.com/>
- CentOS: <http://www.centos.org/>

**Tips:**

到底是要买商业版还是社群版的 Linux distribution 呢？如果是要装在个人计算机上面做为桌面计算机用的，建议使用社群版，包括 Fedora, Ubuntu, OpenSuSE 等等。如果是用在服务器上面的，建议使用商业版本，包括 Red Hat, SuSE 等。这是因为社群版通常开发者会加入最新的软件，这些软件可能会有一些 bug 存在。至于商业版则是经过一段时间的磨合后，才将稳定的软件放进去。



举例来说，Fedora 兜出来的软件套件经过一段时间的维护后，等到该软件稳定到不容易发生错误后，Red Hat 才将该软件放到他们最新的释出版本中。所以，Fedora 的软件比较容易常常改版，Red Hat 的软件就不容易更版。

- Linux 在台湾

当然发行套件者不仅于此。但是值得大书特书的，是中文 Linux 的延伸计划：CLE 这个套件！早期的 Linux 因为是工程师发展的，而这些工程师大多以英文语系的国家为主，所以 Linux 对于国人的学习是比较困扰一点。后来由国人发起的 CLE 计划：<http://cle.linux.org.tw/> 开发很多的中文套件及翻译了很多的英文文件，使得我们目前得以使用中文的 Linux 呢！另外，目前



正在开发中的还有台南县卧龙小三等老师们发起的众多自由软件计划，真是造福很多的朋友啊！

- 自由软件技术交流网: <http://freesf.tnc.edu.tw/index.php>
- B2D: <http://b2d.tnc.edu.tw/>

此外，如果只想看看 Linux 的话，还可以选择所谓的可光盘开机进入 Linux 的 Live CD 版本，亦即是 KNOPPIX 这个 Linux distributions 呢！台湾也有阿里巴巴兄维护的中文 Live CD 喔！

- <http://www.knoppix.net/>
- 中文 KNOPPIX: <http://knoppix.tnc.edu.tw/>

#### Tips:

对于没有额外的硬盘或者是没有额外的主机的朋友来说，KNOPPIX 这个可以利用光盘开机而进入 Linux 操作系统的 Live CD 真的是一个不错的选择！你只要下载了 KNOPPIX 的映象档，然后将他刻录成为 CD，放入你主机的光驱，并在 BIOS 内设定光盘为第一个开机选项，就可以使用 Linux 系统了呢！



如果你还想要知道更多的 Linux distributions 的下载与使用信息，可以参考：

- <http://distrowatch.com/>
- 选择适合你的 Linux distribution

那我到底应该要选择哪一个 distributions？就如同我们上面提到的，其实每个 distributions 差异性并不大！不过，由于套件管理的方式主要分为 Debian 的 dpkg 及 Red Hat 系统的 RPM 方式，目前鸟哥的建议是，先学习以 RPM 套件管理为主的 RHEL/Fedora/SuSE/CentOS 等台湾使用者较多的版本，这样一来，发生问题时，可以提供解决的管道比较多。如果你已经接触过 Linux 了，还想要探讨更严谨的 Linux 版本，那可以考虑使用 Debian，如果你是以效能至上来考虑，那么或许 Gentoo 是不错的建议！

总之，版本很多，但是各版本差异其实不大，建议你一定要先选定一个版本后，先彻头彻尾的了解他，那再继续玩其他的版本时，就可以很快的进入状况。鸟哥的网站仅提供一个版本，不过是以比较基础的方式来介绍的，因此，如果能够熟练俺这个网站的话，呵呵！哪一个 distributions 对你来说，都不成问题啦！

不过，如果依据计算机主机的用途来分的话，在台湾鸟哥会这样建议：

- 用于企业环境: 建议使用商业版本，例如 Red Hat 的 RHEL 或者是 Novell 的 SuSE 都是很不错的选择！毕竟企业的环境强调的是永续的经营，

你可不希望网管人员走了之后整个机房的主机都没有人管理吧！由于商业版本都会提供客户服务，所以可以降低企业的风险喔！

- 用于个人或教学的服务器环境：要是你的服务器所在环境如果当机还不会造成太大的问题的话，加上你的环境是在教学的场合当中时（就是说，唔！经费不足的环境啦！）那么可以使用『号称』完全兼容商业版 RHEL 的 CentOS。因为 CentOS 是抓 RHEL 的原始码来重新兜起来的一个 Linux distribution，所以号称兼容于 RHEL。这一版的软件完全与 RHEL 相同，在改版的幅度较小，适合于服务器系统的环境；
- 用于个人的桌面计算机：想要尝鲜吗？建议使用很炫的 Fedora/Ubuntu 等 Desktop(桌面环境)使用的版本！如果不想要安装 Linux 的话，那么 Fedora 或 CentOS 也有推出 Live CD 了！也很容易学习喔！

---

## Linux 的特色

Linux 是 Torvalds 先生所开发出来的，基于 GPL 的版权宣告之下，可以在 x86 的架构下运作，也可以被移植到其他的大型主机上面。由于开发的相关理念与兼容的问题，因此，我们也可以称 Linux 为 Unix Like 操作系统的一种。

### Tips:

其实 Unix-Like 可以说是目前服务器类型的操作系统的统称啦！因为，不论是 FreeBSD, BSD, Sun Unix, HP Unix, Red Hat Linux, Mandrake Linux 等等，都是由同一个祖先 Thompson 所写的『Unix』来的，因此，这些咚咚都被统称为 Unix-Like 的操作系统啰！



---

## Linux 的特色

那么这个系统有什么特异功能呢？简单的说：

- 自由与开放：  
由于 Linux 是基于 GPL 的授权之下，因此他是自由软件，也就是任何人都可以自由的使用或者是修改其中的原始码的意思！这种开放性架构对科学界来说是相当重要的！因为很多的工程师由于特殊的需求，常常需要修改系统的原始码，使该系统可以符合自己的需求！而这个开放性的架构将可以满足各不同需求的工程师！因此当然就有可能越来越流行啰！以鸟哥来说，目前环境工程界的空气质量模式最新版 [Models-3/CMAQ](#) 就是以 Linux 为基准平台设计的呢！
- 配备需求低廉：

Linux 可以支持个人计算机的 x86 架构，系统资源不必像早先的 Unix 系统那般，仅适合于单一公司 (例如 Sun) 设备！单就这一点来看，就可以造成很大的流行啰！不过，如果你想要在 Linux 下执行 X Window 系统，那么硬件的等级就不能太低了！

- 功能强大而稳定：  
而且由于 Linux 功能并不会输给一些大型的 Unix 工作站，因此，近年来越来越多的公司或者是团体、个人投入这一个操作系统的开发与整合工作！例如 IBM 与升阳公司都有推出 x86 的 Linux 服务器呢！
- 独立作业：  
另外，由于很多的软件套件逐渐被这套操作系统拿来使用，而很多套件软件也都在 Linux 这个操作系统上面进行发展与测试，因此，Linux 近来已经可以独力完成几乎所有的工作站或服务器的服务了，例如 Web, Mail, Proxy, FTP.....。

目前 Linux 已经是相当成熟的一套操作系统啰！而且不耗资源又可以自由取得！呵呵，可以说造成微软相当大的压力呀！此外，由于他的系统硬件要求很低，加上目前很多的人由于『Intel 的阴谋』（呵呵！开玩笑的，因为 Tom 的硬件评论 (<http://www.big5.tomshardware.com/>) 网站常常这样取笑 Intel 的说！呵！很好笑！）而造成手边有相当多的淘汰掉的硬件配备，Linux 在这些被淘汰的硬件中就可以执行的相当的顺畅与稳定！因此也造成相当多朋友的关注啰！

这也是造成 Linux 成为最近几年来最受瞩目的操作系统之一，如前所述，他会受到瞩目的原因主要是因为他是『free』的，就是可以自由取得的操作系统啦！然后他是开放性的系统，也就是你可以随时的取得程序的原始码，这对于程序开发工程师是很重要的！而且，虽然他是 Free 的自由软件，不过功能却很强大！另外，Linux 对于硬件的需求是很低的，这一点更造成它流行的主因，因为硬件的汰换率太快了，所以很多人手边都有一些很少在用的零件，这些零件组一组就可以用来跑 Linux 了，反正做一个工作站又不用使用到屏幕 (只要主机就可以啰)，因此 Linux 就越来越流行啰！

#### Tips:

也就是因为 Linux 具有 1. 硬件需求低、 2. 架构开放、 3. 系统稳定性及保密性功能够强、 4. 完全免费，所以造成一些所谓『反微软联盟』的程序设计高手不断的开发新软件！以与 Microsoft 进行抗衡！



---

#### Linux 的优缺点

那干嘛要使用 Linux 做为我们的主机系统呢？这是因为 Linux 有底下这些优点：

- 稳定的系统：  
Linux 本来就是基于 Unix 概念而发展出来的操作系统，因此，Linux 具有与 Unix 系统相似的程序接口跟操作方式，当然也继承了 Unix 稳定并且有效率的特点。常听到安装 Linux 的主机连续运做一年以上而不曾当机、不必关机是稀松平常的事；
- 免费或少许费用：  
由于 Linux 是基于 GPL 授权下的产物，因此任何人皆可以自由取得 Linux，至于一些『安装套件』的发行者，他们发行的安装光盘也仅需要些许费用即可获得！不同于 Unix 需要负担庞大的版权费用，当然也不同于微软需要一而再、再而三的更新你的系统，并且缴纳大量费用啰！
- 安全性、漏洞的快速修补：  
如果你常玩网络的话，那么你最常听到的应该是『没有绝对安全的主机』！没错！不过 Linux 由于支持者日众，有相当多的热心团体、个人参与其中的开发，因此可以随时获得最新的安全信息，并给予随时的更新，亦即是具有相对的较安全！
- 多任务、多使用者：  
与 Windows 系统不同的，Linux 主机上可以同时允许多人上线来工作，并且资源的分配较为公平，比起 Windows 的单人多任务系统要稳定的多啰！这个多人多任务可是 Unix-Like 上面相当好的一个功能，怎么说呢？你可以在一部 Linux 主机上面规划出不同等级的用户，而且每个用户登入系统时的工作环境都可以不相同，此外，还可以允许不同的使用者在同一个时间登入主机，以同时使用主机的资源。
- 使用者与群组的规划：  
在 Linux 的机器中，档案的属性可以分为『可读、可写、可执行』等参数来定义一个档案的适用性，此外，这些属性还可以分为三个种类，分别是『档案拥有者、档案所属群组、其他非拥有者与群组者』。这对于项目计划或者其他计划开发者具有相当良好的系统保密性。
- 相对比较不耗资源的系统：  
Linux 只要一部 p-100 以上等级的计算机就可以安装并且使用愉快啰！还不需要到 P-4 或 AMD K8 等级的计算机呢！不过，如果你要架设的是属于大型的主机（服务上百人以上的主机系统），那么就需要比较好一点的机器了。不过，目前市面上任何一款个人计算机均可以达到这一个要求啰！
- 适合需要小核心程序的嵌入式系统：  
由于 Linux 只要几百 K 不到的程序代码就可以完整的驱动整个计算机硬件并成为一个完整的操作系统，因此相当适合于目前家电或者是小电子用品的操作系统呢！那就是当红炸子鸡『嵌入式』系统啦！Linux 真的是很适合例如手机、数字相机、PDA、家电用品等等的微电脑操作系统呢！ ^\_^

- 整合度佳且多样的图形用户接口(GUI):  
自从 1994 年 Linux 1.0 后就加入的 X Window 系统, 在众多黑客的努力之下终于与 Linux 有高度整合, 且主要的绘图卡公司 (Intel, NVidia, ATI 等) 都有针对 Linux 推出最新的驱动程序, 因此 Linux 的 GUI 已经有长足的进步了! 另外, Linux 环境下的图形接口不只有一种呢! 包括大家耳熟能详的 KDE (<http://www.kde.org/>) 以及 GNOME (<http://www.gnome.org>) 都是很常见的!

反正 Linux 好处说不完啦! 不过虽然 Linux 具有这样多的好处, 但是他先天上有一个足以致命的地方, 使他的普及率受到很大的限制, 就是 Linux 需要使用『指令列』的终端机模式进行系统的管理! 虽然近年来有很多的图形接口开发使用在 Linux 上面, 但毕竟要熟悉 Linux 还是以指令列来使用是比较好的, 因此要接受 Linux 的玩家必须比较要能熟悉对计算机下指令的行为, 而不是用鼠标点一点 icon 就行了! Linux 还可以改进的地方:

- 没有特定的支持厂商:  
因为在 Linux 上面的所有套件几乎都是自由软件, 而每个自由软件的开发可能并不是公司团体, 而是非营利性质的团体。如此一来, 在你 Linux 主机上面的软件若发生问题, 该如何是好? 好在由于目前 Linux 商业界的整合还不错, 目前在台湾比较具名的 Red Hat 与 SuSE 均有设立了服务点。你可以经由这个服务点来直接向他们购买/咨询相关的软硬件问题呢! 不过, 如果你并非选择有专门商业公司的 Linux distributions 时? 怎么办? 没有专人到府服务呢~这点倒是还不需要太担心, 因为拜网络风行之赐, 你要问的问题几乎在网络上都可以找到答案喔! 看你有没有用心去找就是了!
- 游戏的支持度不足:  
在现代这个时候, 敢说你们家的桌面计算机里面完全没有游戏的小朋友应该不多了! 游戏软件也是个应用程序, 所以它与操作系统的关系就相当密切了。可惜的是目前很多游戏开发商并没有在 Linux 平台上面开发大型游戏, 这间接导致 Linux 无法进入一般家庭说。
- 专业软件的支持度不足:  
这是鸟哥到学校教书后才发现的一件事, 目前很多专业绘图软件公司所推出的专业软件并不支持 Linux 操作系统, 这让同学很难在不同的平台上面操作相同的软件! 唉! 很伤脑筋~
- 教育训练作的还不够好:  
如果能够在国小就教导小朋友使用自由软件, 那么长大自然就会使用自由软件了! 在台湾目前政策方面还是相当的摇摆不定, 希望未来能够给自由软件一些机会。

老实说, 这些缺点绝大部分都不是 Linux 本身的问题, 倒是一些政策面与商业方面的考虑, 才是最大的困扰。不过, Linux 与其他的操作系统一样, 就是一个工具而已! 希望大家能够在快乐中学习到 Linux 的精髓啦! ^\_^



---

## 🐧 其他 Linux 相关话题：标准与授权

除了 Linux 的优缺点我们要知道之外，还有一些议题大家也要来了解了解！

- 依循标准：

Linux 有个优良的传统，那就是支持比较公认而正式的标准，例如开发时就使用的 POSIX 规范就是一个例子。此外，由于 Linux 是一个没有『标准规格』样式的新鲜玩意儿，如果大家都自己玩自己的，那么可想而知的是，未来想要在 Linux 这个操作系统上面发展软件与硬件的厂商一定会无所适从的！

为了让软件开发商、与硬件发展者有一个依循的方向，因此而有 Linux Standard Base (LSB) 及 File system Hierarchy Standard (FHS) 这些个玩意儿的诞生！所以，各个 distributions 也都要遵循 LSB 上面的规范，软硬件发展者也都会依循 LSB，所以啰，我们才会常常说，各大 distributions 虽然在提供的工具与创意上面有所不同，但是基本上，他们的 Linux 架构都是很类似的！因此，你只要玩一套 Linux distribution 也就够了！其他的 distribution 应该就难不倒你啰！

- FHS: <http://www.pathname.com/fhs/>
- LSB: <http://www.linuxbase.org/>
- 服务器、工作站、终端机...

由于网络的盛行，我们或许常常会听到一些名词ㄟ~所以，底下先来介绍几个简单的网络名词吧！^\_^，来认识一下什么是服务器(Server)、工作站(Workstation)与终端机(Terminal)？简单的来说，你可以这样认为：

- 服务器(Server)：  
提供 Internet 一种以上的网络服务的主机，例如 yahoo 提供的是 WWW 的服务，那么 yahoo 就可以称之为服务器了！那么，你自己架设一台 mail server 呢？呵呵！那也是一个小型的服务器啰！所以，你必须清楚的知道，服务器是有规模大小之分的！目前一部 Linux 上面就可以架设多个服务器软件啰！例如 WWW 服务器的 Apache 软件、FTP 服务器的 Wu-ftp 软件等等的；
- 工作站(Workstation)：  
根据[计算器概论](#)中的定义，基本上，在个人计算机流行以前所制造的计算机系统称为工作站计算机，例如升阳公司的 SPARC 工作站。工作站可以视为仅提供一群特定人士，作为数值分析、科学用途的机器。鸟哥之前在读书时待的的研究室有一部 SPARC 机器，他仅提供我们研究室内部几部计算机的联机，主要的功能就是在跑空气质量的数值模式啦！但工作站也能够被当成服务器哩！因为如果这部工作站主机开启

了 Internet 的服务后，例如将 SPARC 上面的 mail server 开启之后，那么这部机器就可以称之为服务器了！

- 终端机(Terminal):  
简单的说，就是 end-user(就是你啦)前面的那部计算机！呵呵，例如鸟哥都是使用我的工作机(Windows) 连上我们的主机来工作，那么这一部 Windows 的计算机，就可以称为 terminal 机器啰！不过，更狭义的说，『终端机』本身应该是不具备任何可以作业的软件，在终端机上面一定要连上 Server 之后，才能进行各项作业！那才是最狭义的终端机啰！例如我们前面说过的[早期的大型主机联机模式](#)！

大致上的分别可以用上面的观点来看！

- 几个常见的授权模式与定义

现在市面上有好多的软件，有的是自由软件，有的是专利软件。有的专利软件免费，有的自由软件要钱～啊！好烦啊！怎么分辨这些东西？其实，鸟哥并不是律师，对于法律也不十分懂，不过，还是有几个授权模式可以来谈一谈～

## Open Source

软件以 Open Source 的方式释出时，表示除了可执行的软件本身外，一定伴随着原始码的释出喔！通常 Open Source 的软件有几个好处：

1. 程序设计师通常会等到程序成熟之后才会释出(免得被笑，^\_^)，所以通常程序在雏形的时候，就已经具有相当的优良体质；
2. Open Source 的精神，相信当程序原设计人将程序原始码释出之后，其他的程序设计师接受这份原始码之后，由于需要将程序改成自己所需的样式，所以会经由本身的所学来加以改良，并从中加以改良与除虫，所以程序的 debug 功能会比传统的 close source 来的快！
3. 由于程序是伴随原始码的，因此，系统将会不易存在鲜为人知的木马程序或一些安全漏洞，相对而言，会比较更加的安全！

Open source 的代表授权为 GNU 的 GPL 授权及 BSD 等等，底下列出知名的 Open Source 授权网页：

- GNU General Public License:  
<http://www.gnu.org/licenses/licenses.html#GPL>  
目前有 version 2, version 3 两种版本，Linux 使用的是 version 2 这一版。更多与 GPL 相关的说明可以参考[附录 A](#)的中文化条文
- Berkeley Software Distribution (BSD):  
[http://en.wikipedia.org/wiki/BSD\\_license](http://en.wikipedia.org/wiki/BSD_license)

使用 BSD source code 最常接触到的就是 BSD 授权模式了！这个授权模式其实与 GPL 很类似，而其精神也与 Open Source 相呼应呢！

- Apache License, Version 2.0:  
<http://www.apache.org/licenses/LICENSE-2.0>  
Apache 是一种网页服务器软件，这个软件的发布方式也是使用 Open source 的。只是在 apache 的授权中规定，如果想要重新发布此软件时(如果你有修改过该软件)，软件的名称依旧需要定名为 Apache 才行！

Close Source:

相对于 Open Source 的软件会释出原始法，Close source 的程序则仅推出可执行的二进制程序(binary program)而已。这种软件的优点是有专人维护，你不需要去更动他；缺点则是灵活度大打折扣，用户无法变更该程序成为自己想要的样式？此外，若有木马程序或者安全漏洞，将会花上相当长的一段时间来除错！这也是所谓专利软件(copyright)常见的软件出售方式。

虽然专利软件常常代表就是需要花钱去购买，不过有些专利软件还是可以免费提供大众使用的！免费的专利软件代表的授权模式有：

- Freeware:  
<http://en.wikipedia.org/wiki/Freeware>  
不同于 Free software, Freeware 为『免费软件』而非『自由软件！』虽然它是免费的软件，但是不见得要公布其原始码，端看释出者的意见啰！这个东西与 Open Source 毕竟是不太相同的东西喔！此外，目前很多标榜免费软件的程序很多都有小问题！例如假藉免费软件的名义，实施用户数据窃取的目的！所以『来路不明的软件请勿安装！』
- Shareware:  
<http://en.wikipedia.org/wiki/Shareware>  
共享件这个名词就有趣了！与免费软件有点类似的是，Shareware 在使用初期，它也是免费的，但是，到了所谓的『试用期限』之后，你就必须要选择『付费后继续使用』或者『将它移除』的宿命～通常，这些共享件都会自行撰写失效程序，让你在试用期限之后就无法使用该软件。

---

## 重点回顾

- 计算机主要以二进制作为单位，而目前常用的磁盘容量单位为 bytes，其单位换算为 1Byte = 8bits，其他的以 1024 为其倍数，如 1GByte=1024MBytes 等等。
- 操作系统(Operation System)主要在管理与驱动硬件，因此必须要能

够管理内存、管理装置、负责行程管理以及系统呼叫等等。因此，只要能够让硬件准备妥当(Ready)的情况，就是一个阳春的操作系统了。

- 最阳春的操作系统仅在驱动与管理硬件，而要使用硬件时，就得需要透过应用软件或者是壳程序(shell) 的功能，来呼叫操作系统操纵硬件工作。因此，目前称为操作系统的，除了上述功能外，通常已经包含了日常工作所需要的应用软件在内了。
- Unix 的前身是由贝尔实验室(Bell lab.)的 Ken Thompson 利用汇编语言写成的，后来在 1971-1973 年间由 Dennis Ritchie 以 C 程序语言进行改写，才称为 Unix。
- 1977 年由 Bill Joy 释出 BSD (Berkeley Software Distribution)，这些称为 Unix-like 的操作系统。
- 1984 年由 Andrew Tannenbaum 制作出 Minix 操作系统，该系统可以提供原始码以及软件；
- 1984 年由 Richard Stallman 提倡 GNU 计划，倡导自由软件(Free software)，强调其软件可以『自由的取得、复制、修改与再发行』，并规范出 GPL 授权模式，任何 GPL (General Public License) 软件均不可单纯仅贩卖其软件，也不可修改软件授权。
- 1991 年由芬兰人 Linus Torvalds 开发出 Linux 操作系统。简而言之，Linux 成功的地方主要在于：Minix(Unix), GNU, Internet, POSIX 及虚拟团队的产生。
- Linux 本身就是个最阳春的操作系统，其开发网站设立在 <http://www.kernel.org>，我们亦称 Linux 操作系统最底层的数据为『核心(Kernel)』。
- 目前 Linux 核心的发展分为两种版本，分别是稳定版本的偶数版，如 2.6.X，适合于商业与家用环境使用；一种是发展中版本如 2.5.X 版，适合开发特殊功能的环境。
- Linux distributions 的组成含有：『Linux Kernel + Free Software + Documentations(Tools) + 可完全安装的程序』所制成的一套完整的系统。



## 本章习题

(要看答案请将鼠标移动到『答：』底下的空白处，按下左键圈选空白处即可察看)

- 一个完整的操作系统至少要能够完整的控制整个硬件，请问，操作系统应该要控制硬件的哪些单元？

至少要能够控制：(1)input/output control, (2)device control, (3)process management, (4)file management. 等等！

- 核心的功能在于管控整个系统的硬件，这包括了 CPU 运算单元的管理，输入/输出的管理，内存的管理等等。那么请问一个较为完整的操作系统，应该包含哪些部分？

应包含 Kernel + Kernel Tools + Applications 等等

- 一个 GBytes 的硬盘空间，等于几个 KBytes?

$1\text{GBytes} = 1024\text{ MBytes} * 1024\text{ KBytes/MBytes} = 1048576\text{ KBytes}$

- 你在你的主机上面安装了一张网络卡，但是开机之后，系统却无法使用，你确定网络卡是好的，那么可能的问题出在哪里？该如何解决？

因为所有的硬件都没有问题，所以，可能出问题的地方在于系统的核心(kernel) 不支持这张网络卡。解决的方法，到网络卡的开发商网站，下载支持你主机操作系统的驱动程序，安装网卡驱动程序后，就可以使用了。

- 我在一部主机上面安装 Windows 操作系统时，并且安装了显示适配器的驱动程序，他是没有问题的。但是安装 Linux 时，却无法完整的显示整个 X Window。请问，我可不可以将 Windows 上面的 显示适配器驱动程序拿来安装在 Linux 上？

不行！因为核心不同，针对硬件所写的驱动程序也会不相同，编译程序也不同，当然，驱动程序也无法在两个操作系统间兼容。这也是为何开发商在他们的网站上面，都会同时提供许多不同操作系统的驱动程序之故。

- 我在 Windows 上面玩的游戏，可不可以拿到 Linux 去玩？

当然不行！跟上一题相似的，因为游戏也是一个应用程序(application)，他必须要使用到核心所提供的工具来开发他的游戏，所以这个游戏是不可在不同的平台间运作的。除非这个游戏已经进行了移植。

- 什么是软件的移植？

程序是由程序代码(通成为 ASCII 格式)经过编译程序编译成为 binary file 之后，才能够在该操作系统上面执行。因此，你可以将你的程序的程序代码修改成可以适应 其他操作系统的环境，并且加以编译，使程序可以在其他平台上运作，这个修改的动作即为移植。

- Linux 本身仅是一个核心与相关的核心工具而已，不过，他已经可以驱动所有的硬件，所以，可以算是一个很阳春的操作系统了。经过其他应用程序的开发之后，被整合成为 Linux distributions。请问众多的 distributions 之间，有何异同？

相同：利用同样的 Linux kernel (<http://www.kernel.org>)，几乎相同的自由软件套件（例如 GNU 里面的



gcc/glibc/vi/apache/bind/sendmail... ), 几乎相同的操作接口 (例如均使用 bash/KDE/GNOME 等等)。

不同: 使用的 kernel 与各软件的版本可能会不同; 各开发商加入的应用工具不同, 使用的套件管理模式不同 (debian 与 RPM)

- Unix 是谁写出来的? GNU 计划是谁发起的?

Unix 是 Ken Thompson 写的, 1973 年再由 Dennis Ritchie 以 C 语言改写成功。至于 GNU 与 FSF 则是 Richard Stallman 发起的。

- GNU 的全名为何? 他主要由那个基金会支持?

GNU 是 GNU is Not Unix 的简写, 是个无穷循环! 另外, 这个计划是由自由软件基金会 (Free Software Foundation, FSF) 所支持的! 两者都是由 Stallman 先生所发起的!

- 我要如何取得 Linux distribution 的可安装光盘?

目前各大版本的 Linux distribution 大致上仍然依附在 GPL 这个版权宣告上, 因此在网络上面都可以轻易的下载, 若有兴趣的话可以到各发展的厂商主网页去下载。然而为了带宽的节省起见, 建议在台湾以映像站台来下载速度上会快上很多, 特别建议南台湾朋友可以到义守大学的 FTP 站 ( <http://ftp.isu.edu.tw/> ) 下载资料!

- 简单的说明一下什么是 GNU 的 GPL ?

1. 任何个人或公司均可释出自由软件 ( free software );
2. 任何释出自由软件的个人或公司, 均可由自己的服务来收取适当的费用;
3. 该软件的原始码 ( Source Code ) 需要随软件附上, 并且是可公开发表的;
4. 任何人均可透过任何正常管道取得此一自由软件, 且均可取得此一授权模式。

- 何谓多人 ( Multi-user ) 多任务 ( Multitask )?

Multiuser 指的是 Linux 允许多人同时连上主机之外, 每个使用者皆有其各人的使用环境, 并且可以同时使用系统的资源!

Multitask 指的是多任务环境, 在 Linux 系统下, CPU 与其他例如网络资源可以同时进行多项工作, Linux 最大的特色之一即在于其多任务时, 资源分配较为平均!

- 简单说明 GNU General Public License ( GPL ) 与 Open Source 的精神:

1. GPL 的授权之软件，乃为自由软件 (Free software)，任何人皆可拥有他；2. 开发 GPL 的团体(或商业企业)可以经由该软件的服务来取得服务的费用；3. 经过 GPL 授权的软件，其属于 Open source 的情况，所以应该公布其原始码；4. 任何人皆可修改经由 GPL 授权过的软件，使符合自己的需求；5. 经过修改过后 Open source 应该回馈给 Linux 社群。

- 有个朋友问我说『Linux 是什么？』我该如何回答比较好？

简单的说，Linux 就是一个操作系统，或者说，Linux 是操作系统最底层的核心。这个核心可以管理整个计算机硬件，让计算机硬件可以完整的运作起来，并等待用户输入指令。最早 Linux 是由 Torvalds 在 1991 年写出来的，后来由于他承接了 Unix 的良好传统：稳定性高、多人多任务的环境设计优良、要求配备较低等优点，所以很多软件开发商在这个核心上面开发，而某些厂商将这些软件与核心整合成为可以完全安装的光盘，而成为目前大家常常听到的 Linux 操作系统了。

- 什么是 POSIX？为何说 Linux 使用 POSIX 对于发展有很好的影响？

POSIX 是一种标准规范，主要针对在 Unix 操作系统上面跑的程序来进行规范。若你的操作系统符合 POSIX，则符合 POSIX 的程序就可以在你的操作系统上面运作。Linux 由于支持 POSIX，因此很多 Unix 上的程序可以直接在 Linux 上运作，因此程序的移植相当简易！也让大家容易转换平台，提升 Linux 的使用率。

- Linux 的发展主要分为哪两种核心版本？

主要分为奇数的发展中版本(develop)，如 2.5，及偶数的稳定版本，如 2.6。

- 什么是 Linux 的 Live CD？

所谓的 Live CD 就是将完整的 Linux distribution 放置到一片光盘（目前也有 DVD 版本了）当中，然后透过重新启动以『光盘开机』，就能够不使用硬盘直接进入 Linux 系统的环境。

- 简单说明 Linux 成功的因素？

1. 藉由 Minix 操作系统开发的 Unix like，没有版权的纠纷；
2. 藉助于 GNU 计划所提供的各项工具软件，gcc/bash 等；
3. 藉由 Internet 广为流传；
4. 藉由支持 POSIX 标准，让核心能够适合所有软件的开发；
5. 托瓦兹强调务实，虚拟团队的自然形成！



## 参考数据与延伸阅读

- Multics 计划网站: <http://www.multicians.org/>。
- 葛林穆迪着, 杜默译, 『Linux 传奇』, 时报文化出版企业。  
书本介绍: <http://findbook.tw/book/9789571333632/basic>
- 网络农夫, 2001, Unix 简史:  
<http://netlab.cse.yzu.edu.tw/~statue/freebsd/docs/csh/>
- Ken Thompson 的个人网站:  
<http://plan9.bell-labs.com/cm/cs/who/ken/index.html>
- Dennis Ritchie 的个人网站:  
<http://cm.bell-labs.com/cm/cs/who/dmr/>
- Richard Stallman 的个人网站: <http://www.stallman.org/>
- GNU 计划: <http://www.gnu.org>
- XFree86 的网站: <http://www.xfree86.org/>
- 洪朝贵老师的 GNU/FSF 介绍:  
[http://people.offset.org/~ckhung/a/c\\_83.php](http://people.offset.org/~ckhung/a/c_83.php)
- 维基百科对 Linus Torvalds 的介绍:  
[http://en.wikipedia.org/wiki/Linus\\_Torvalds](http://en.wikipedia.org/wiki/Linus_Torvalds)。
- POSIX 的相关说明:  
维基百科: <http://en.wikipedia.org/wiki/POSIX>  
IEEE POSIX 标准: <http://standards.ieee.org/regauth/posix/>

---

2002/06/25: 第一次完成

2003/01/26: 重新修订, 加入一些历史事件、重新编排与加入 FAQ

2003/02/28: 加入百资以及 distrowatch 两个网站的推荐!

2005/05/31: 旧有的资料放于 [此处](#)

2005/06/02: 做了大幅度的改版, 很多数据参考了网络农夫及 Linux 传奇等书籍, 建议大家要多看看网络农夫的大作喔!

2005/06/08: 将原本的 binary / compiler / Emacs 的地方再说明一下! 比较容易了解那是什么! 顺便加入习题

2005/07/21: 网络农夫的网站结束了~真伤心~只好提供网络农夫之前发表的文章连结了!

2005/08/03: 感谢网友 babab 的来信告知, 修订了国家高速网络中心网址:  
<http://www.nchc.org.tw>

2005/10/24: 经由网友的回报, 洪朝贵老师已经调职到树德大学, 因此整个连结内容已作修订。

2006/05/31: 加入了重点回顾的项目啦!

2006/06/06: 感谢网友 "Warren Hsieh" 兄的提醒, 由于麦金塔在 2006 年后使用 Intel 的 x86 硬件架构, 故 Windows 是可能可以在上面安装的!

2008/07/23: 因为加入了计算器概论的章节, 所以本文做了挺大幅度的修改! 原本针对 FC4 的版本请点选[这里](#)。

2007/07/26: 将整份文章重新校阅过, 修订一些文辞, 也将格式调整为适合

的 XHTML 了！

2007/07/29：将主、次核心版本加强说明！

---

2002/02/03 以来统计人数

703542



本网页主要以 [firefox](#) 配合分辨率 1024x768 作为设计依据

<http://linux.vbird.org> is designed by [VBird](#) during 2001-2009. [Aerosol Lab](#).