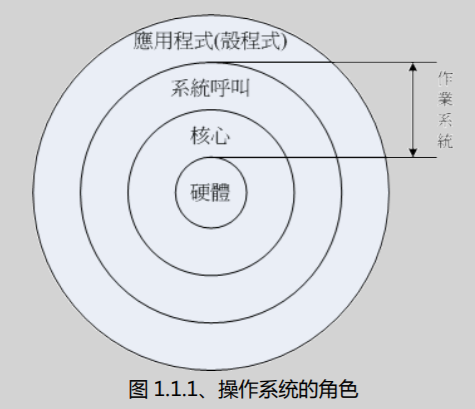
Linux

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。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。



由上图中我们可以看到其实核心（内核）和硬件的关系非常的强烈。早期的Linux系统只是一套操作系统并不含有其他的应用程序。

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行的。

举例来说，如果你想要让x86 上面跑的那套操作系统也能够在Power CPU上运作时，就得要将该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运作了。这个动作我们通常就称为“软件移植”。

因为Linux是一个开源的操作系统，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。

Linux提供了一个完整的操作系统当中最底层的硬件控制和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渐渐的将他们的工作心血转移到这个架构上面，所以Linux操作系统也有很多的应用软件了。

GNU：

GNU就是GNU is not Unix的缩写，GNU的创始人Stallman认为Unix虽然不是最好的操作系统，但是至少不会太差，而他自信有能力把Unix不足的地方加以改进，使它成为一个优良的操作系统。

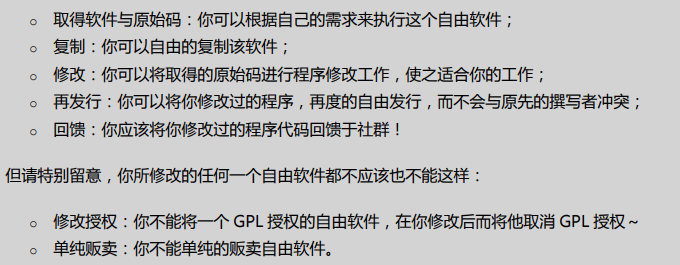
作为操作系统，GNU的发展仍未完成，其中最大的问题是具有完备功能的内核尚未被开发成功。GNU的内核，称为Hurd是自由软件基金会发展的重点，但是仍未发展成熟。在实际使用上，多半使用Linux内核，FreeBSD等替代方案。

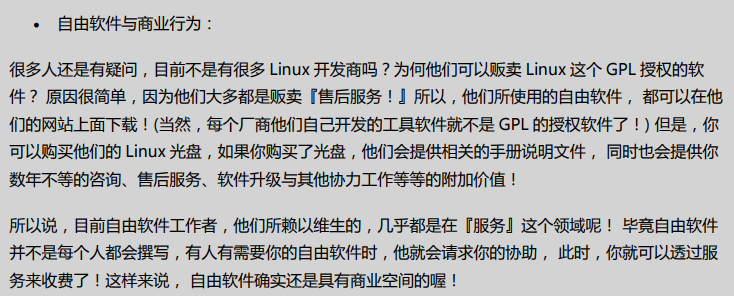
GNU的通用公共许可证：

到了1985年，为了避免GNU所开发的自由软件被其他人利用成为专利软件，所以他草拟了有名的通用公共许可证（GPL）。

其中GNU开发出了BASH SHELL。一个被使用来操作操作系统的基本接口。

自由软件：每个人都可以分享自己的软件源码，可以让所有人进行交流学习。以后你所拿到的软件原先只能在Unix上跑，但是经过原始码的修改之后（自由软件的优点），你将可以拿到Linxu或者Windows上跑。总的来说，一个软件挂上了GPL版权宣告之后，他自然就成了自由软件。





就是说自由软件可以免费使用，但是其背后的服务就是需要收费的了。

托瓦茲（Linus Torvalds）在编写出Linux核心之后为了让自己的Linux能够兼容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，他发现到有很多软件无法在Linux这个核心上运作。这个时候托瓦兹选择了修改Linux，让Linux符合软件能够运作的规范。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POSIX规范。

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。是美国电器与电子工程师学会（IEEE）发布的一项标准。

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，只要依据这些标准来设计的核心和软件，理论上，就可以搭配在一起执行了。

由于网络上志愿者的热情相助，使得Linux蓬勃发展。从之前小小的内核不断发展壮大。

但是因为Linux核心的开发者太多了，以至于造成Linux核心经常性的变动。因此托瓦兹便将核心的发展趋势分为两股。

* 主、次版本为奇数：发展中版本

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：稳定版本

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。用于提供使用者一个稳定的Linux作业环境平台。

**重点**：

Linux本身并不是一个操作系统，只是一个内核。所以真正的操作系统应该是Linux发布版即distribution。比如说“我用的是Linux的CentOS这个distribution，版本是5.X，请问。。。”

由上方的说明知道，Linux其实就是一个操作系统最底层的核心，它是GNU GPL授权模式，所以，任何人都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Unix操作系统。

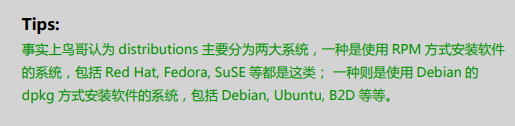
由于慢慢的很多软件已经可以在Linux上运作了，因此，“Linux各种软件”就可以完成一个相当完整的操作系统了。为了使使用者能够更加方便的接触Linux，很多商业公司就将Linux Kernel（含tools）与可运行的软件整合起来，加上自己具有创意的工具程序。这个“Kernel+Softwares+Tools的可完全安装”的东西，我们称之为Linux Distribution。

各大Linux Distributions的主要异同：支持标准

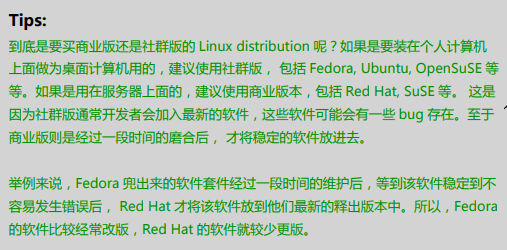
不过，由于发展Linux Distributions的公司实在太多了，例如Red Hat，Ubuntu，Fedoras等，所以很多会担心，是不是每个Distribution都不同呢？

这就不需要担心了，因为每个Linux Distribution使用的kernel都是一致的，而他们所选择的软件，几乎都是目前很知名的软件，例如网页服务器Apache，电子邮件服务器Postfix，文件服务器的Samba等。

此外，为了让所有的Linux distributions开发不至于差异太大，且让这些开发商在开发的时候有所依据，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。唯一的差别可能就是开发者自家所开发出来的管理工具，以及套件管理的模式把。



RPM：Red Hat Package Manager是Red Hat发行版专门用来管理Linux各项套件的程序。RPM套件管理方式的出现，让Linux易于安装，升级。（套件类似于压缩包）



关于授权：

以下有几个授权模式可以了解：

* Open Source（开放源码）

软件以Open Source的方式释出时，表示除了可执行的软件本身外，一定伴随这源码的释出。

Open Source的代表授权为GNU的GPL授权及BSD等。

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的软件会释出源码，Close Source的程序则仅推出可执行的二进制程序而已。这种软件的优点是有专人维护；缺点是灵活度大打折扣，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，将会花上很长的时间来来除错。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：

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源码。这个东西和Open Source毕竟是不太相同的。此外，目前很多标榜免费软件的程序很多都有小问题。例如假借免费软件的名义，实施用户数据窃取的目的。

ShareWare：

与免费软件有点类似的是，ShareWare在使用初期也是免费的，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还是将它移除的宿命。