**[Linux介绍及架构简介](http://km.makeblock.com/pages/viewpage.action?pageId=16812597)**

[转至元数据结尾](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&refresh=1505277967634&focusedCommentId=16812637&refresh=1505278070042#page-metadata-end)

* 创建： [王活生](http://km.makeblock.com/display/%7Eshenghuowang@makeblock.cc)，最新修改： [昨天3:58 下午](http://km.makeblock.com/pages/diffpagesbyversion.action?pageId=16812597&selectedPageVersions=5&selectedPageVersions=6)

[转至元数据起始](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&refresh=1505277967634&focusedCommentId=16812637&refresh=1505278070042#page-metadata-start)

第一部分 Linux介绍

第二部分 Linux架构

第三部分 Linux如何学习

第四部分 学习资料整理（见评论）

**1.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？为什么 Linux 还不用钱？

**1.1 Linux是什么**

       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！

       所以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！

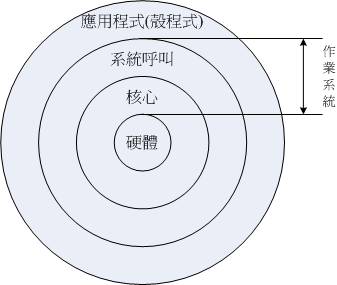


                   图1.1.1、操作系统的角色

      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渐渐的将他们的工作心血移转到这个架构上面，所以 Linux 操作系统也有很多的应用软件啦！

      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提供的软件的整合， 使得Linux成为一个更完整的、功能强大的操作系统啰！

**1.2 Linux历史**

**1.2.1 Unix与Minix**

      在计算机出现早期，程序员是使用硬件语言在裸硬件上进行开发的。缺少操作系统就意味着在某个时间只有一个应用程序（和一个用户）可以使用这些庞大而又昂贵的设备。在 20 世纪 60 年代，MIT（Massachusetts Institute of Technology）和一些公司为 GE-645 开发了一个名为 Multics（Multiplexed Information and Computing Service）的实验性的操作系统， 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：

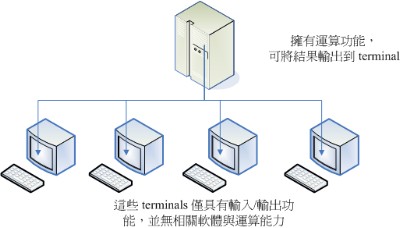


                                  图1.1.2、兼容分时系统

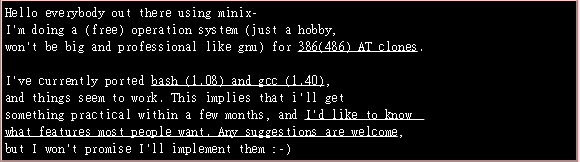
        这个操作系统的开发者之一 AT&T 后来退出了 Multics，并在 1970 年开发了自己的名为 Unics 的操作系统。与这个操作系统一同诞生的是 C 语言，C 语言就是为此而开发的，然后它们使用 C 语言对操作系统进行了重写，使操作系统开发具有可移植性。

二十年后，Andrew Tanenbaum 创建了一个微内核版本的 UNIX®，名为 MINIX（代表 minimal UNIX），它可以在小型的个人计算机上运行。这个开源操作系统在 20 世纪 90 年代激发了 Linus Torvalds 开发 Linux 的灵感

**1.2.2 Linnux 0.02诞生**

[Linus Torvalds](http://en.wikipedia.org/wiki/Linus_Torvalds)(托瓦兹）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。

托瓦兹写出了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。 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：



        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。 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， 另外，托瓦兹所丢到该目录下的第一个核心版本为0.02！)

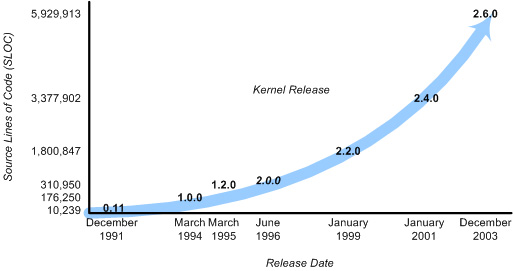


                                                 图1.1.3 主要Linux内核发行版简史

**1.3 Linux distributions**

     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。

**1.3.1 可完全安装的Linux发布套件**

       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 + 各种软件』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。

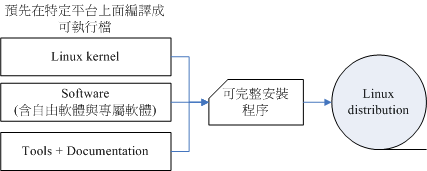


                               图1.3.1、Linux可完全安装发布套件

**1.3.2 各大Linux Distributions的主要异同：支持标准！**

       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](http://www.kernel.xn--org-4d2ez75eoz1am53b/)，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，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！

* [FHS: http://www.pathname.com/fhs/](http://www.pathname.com/fhs/)
* [LSB: http://www.linuxbase.org/](http://www.linuxbase.org/)

     几个主要的Linux distributions发行者网址：

* [Red Hat: http://www.redhat.com](http://www.redhat.com/)
* [Fedora: http://fedoraproject.org/](http://fedoraproject.org/)
* [Mandriva: http://www.mandriva.com](http://www.mandriva.com/)
* [Novell SuSE: http://www.novell.com/linux/](http://www.novell.com/linux/)
* [Debian: http://www.debian.org/](http://www.debian.org/)
* [Slackware: http://www.slackware.com/](http://www.slackware.com/)
* [Gentoo: http://www.gentoo.org/](http://www.gentoo.org/)
* [Ubuntu: http://www.ubuntu.com/](http://www.ubuntu.com/)
* [CentOS: http://www.centos.org/](http://www.centos.org/)

**2.Linux架构**

**2.1 实时操作系统和Linux操作系统的区别**

                                                                           图2.1.1 vxworks与Linux区别

**2.2 Linux体系结构**



                      图2.2.1 Linux操作系统体系结构

      最上面是用户（或应用程序）空间。这是用户应用程序执行的地方。用户空间之下是内核空间，Linux 内核正是位于这里。GNU C Library （glibc）也在这里。它提供了连接内核的系统调用接口，还提供了在用户空间应用程序和内核之间进行转换的机制。这点非常重要，因为内核和用户空间的应用程序使用的是不同的保护地址空间。每个用户空间的进程都使用自己的虚拟地址空间，而内核则占用单独的地址空间。

      Linux 内核可以进一步划分成 3 层。最上面是系统调用接口，它实现了一些基本的功能，例如 read 和 write。系统调用接口之下是内核代码，可以更精确地定义为独立于体系结构的内核代码。这些代码是 Linux 所支持的所有处理器体系结构所通用的。在这些代码之下是依赖于体系结构的代码，构成了通常称为 BSP（Board Support Package）的部分。这些代码用作给定体系结构的处理器和特定于平台的代码。

**2.3 Linux内核主要组件**

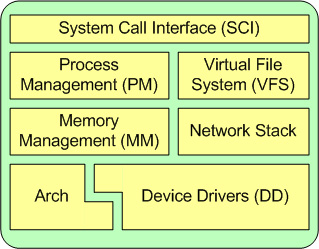


                    图2.3.1 Linux内核组件

  虚拟文件系统（VFS）是 Linux 内核中非常有用的一个方面，因为它为文件系统提供了一个通用的接口抽象。VFS 在 SCI 和内核所支持的文件系统之间提供了一个交换层

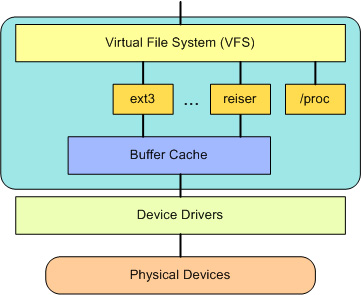


     图2.3.2 VFS在用户和文件系统之间提供了一个交换层

     在 VFS 上面，是对诸如 open、close、read 和 write 之类的函数的一个通用 API 抽象。在 VFS 下面是文件系统抽象，它定义了上层函数的实现方式。它们是给定文件系统（超过 50 个）的插件。文件系统的源代码可以在 ./linux/fs 中找到。

文件系统层之下是缓冲区缓存，它为文件系统层提供了一个通用函数集（与具体文件系统无关）。这个缓存层通过将数据保留一段时间（或者随即预先读取数据以便在需要是就可用）优化了对物理设备的访问。缓冲区缓存之下是设备驱动程序，它实现了特定物理设备的接口。

**3.Linux如何学习**

**3.1 学习路径**

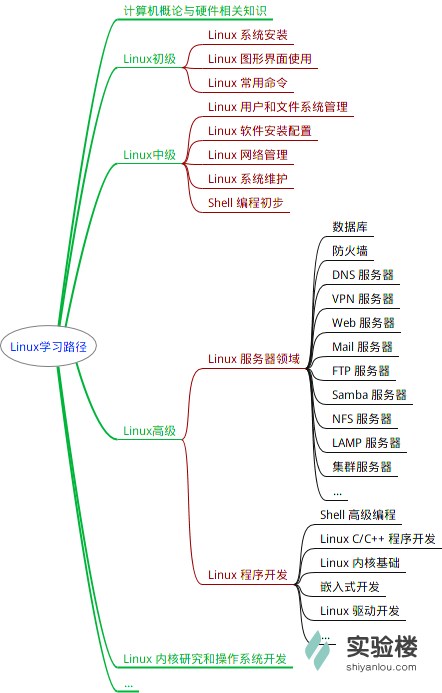


                              图3.1.1 Linux学习路径

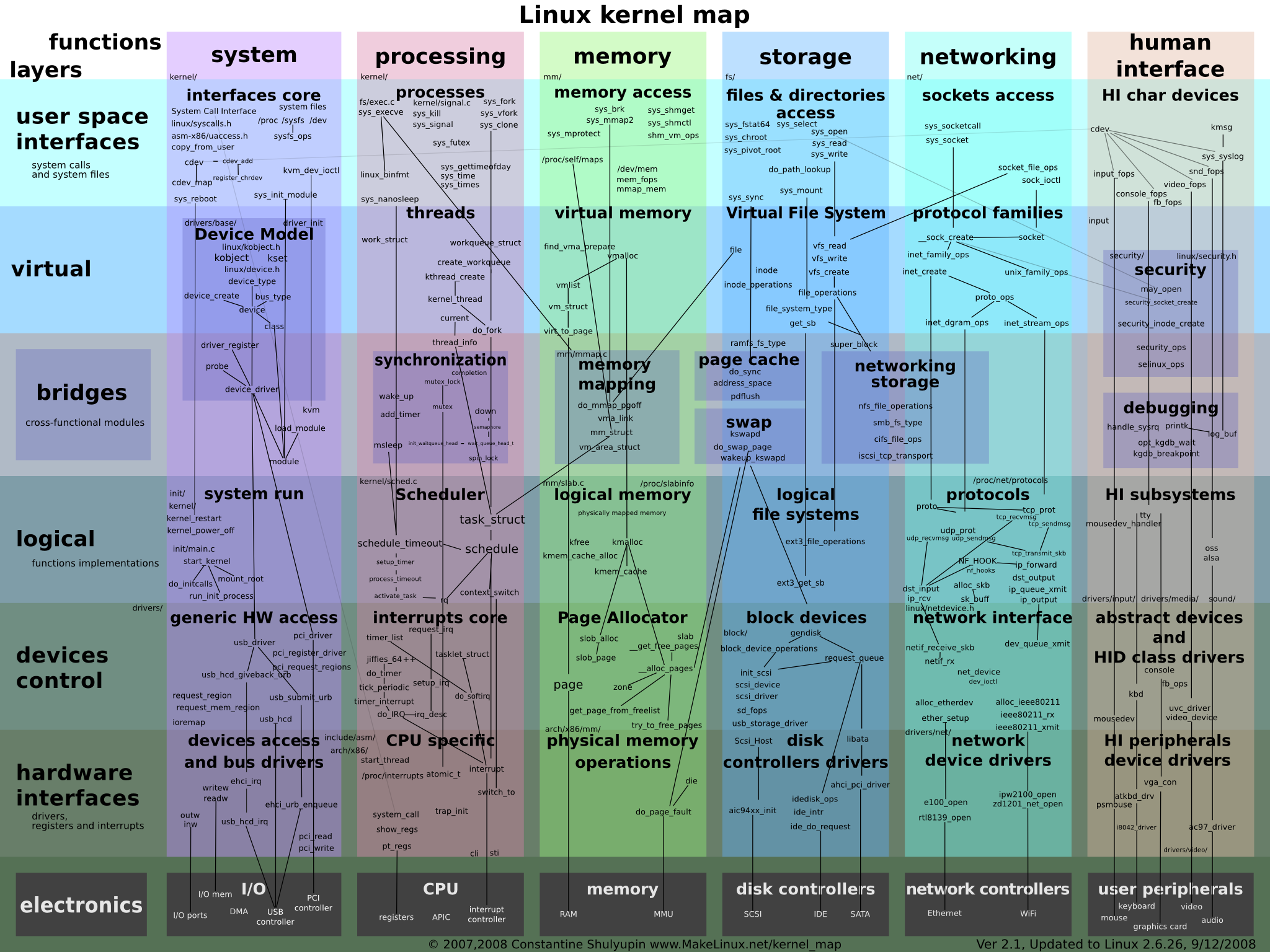


                                                                                                                             图3.1.2 Linux内核学习路线

* [电脑基础 http://www.study-area.org/compu/compu.htm](http://www.study-area.org/compu/compu.htm)
* [网路基础 http://www.study-area.org/network/network.htm](http://www.study-area.org/network/network.htm)

**3.2 选择一本易读的工具书**

      每个人的阅读习惯都不太一样，所以，除了大家推荐的书籍之外， 最好亲眼看过该本书籍，确定您可以吸收的了书上的内容。

      《鸟哥的私房菜--基础学习篇》

      《鸟哥的私房菜--服务器》                服务器方向

      《Linux设备驱动开发详解》              内核开发/驱动开发方向

**3.3 实践**

找一部计算机，赶快安装一个Linux distribution，然后快点进入Linux的世界里面晃一晃！ 相信对於你自己的Linux能力必然大有斩获！除了自己的实作经验之外，也可以参考网路上一些善心人士整理的实作经验分享！ 例如最有名的Study-Area(http://www.study-area.org)等网站。

在人类记忆的曲线中， 你必须要『不断的重复练习』才会将一件事情记得比较熟！同样的，学习Linux也一样， 如果你无法经常摸索的话，那么，抱歉的是，学了后面的， 前面的忘光光！学了等於没学。所以，除了要实践之外，还得要常摸！

        列出几个学习网站来提供大家做为参考实作的依据：

* [Study-Area http://www.study-area.org](http://www.study-area.org/)
* [鸟哥的私房菜馆 http://linux.vbird.org](http://linux.vbird.org/)
* [卧龙大师的网路技术文件 http://linux.tnc.edu.tw/techdoc/](http://linux.tnc.edu.tw/techdoc/)
* [台湾 Linux 社群 http://www.linux.org.tw/](http://www.linux.org.tw/)
* [狼主的网路实验室 http://netlab.kh.edu.tw/index.htm](http://netlab.kh.edu.tw/index.htm)
* [大南国小（林克敏主任文件集）http://freebsd.lab.mlc.edu.tw/](http://freebsd.lab.mlc.edu.tw/)
* [吴仁智的文件集 http://www.cses.tcc.edu.tw/~chihwu/](http://www.cses.tcc.edu.tw/%7Echihwu/)

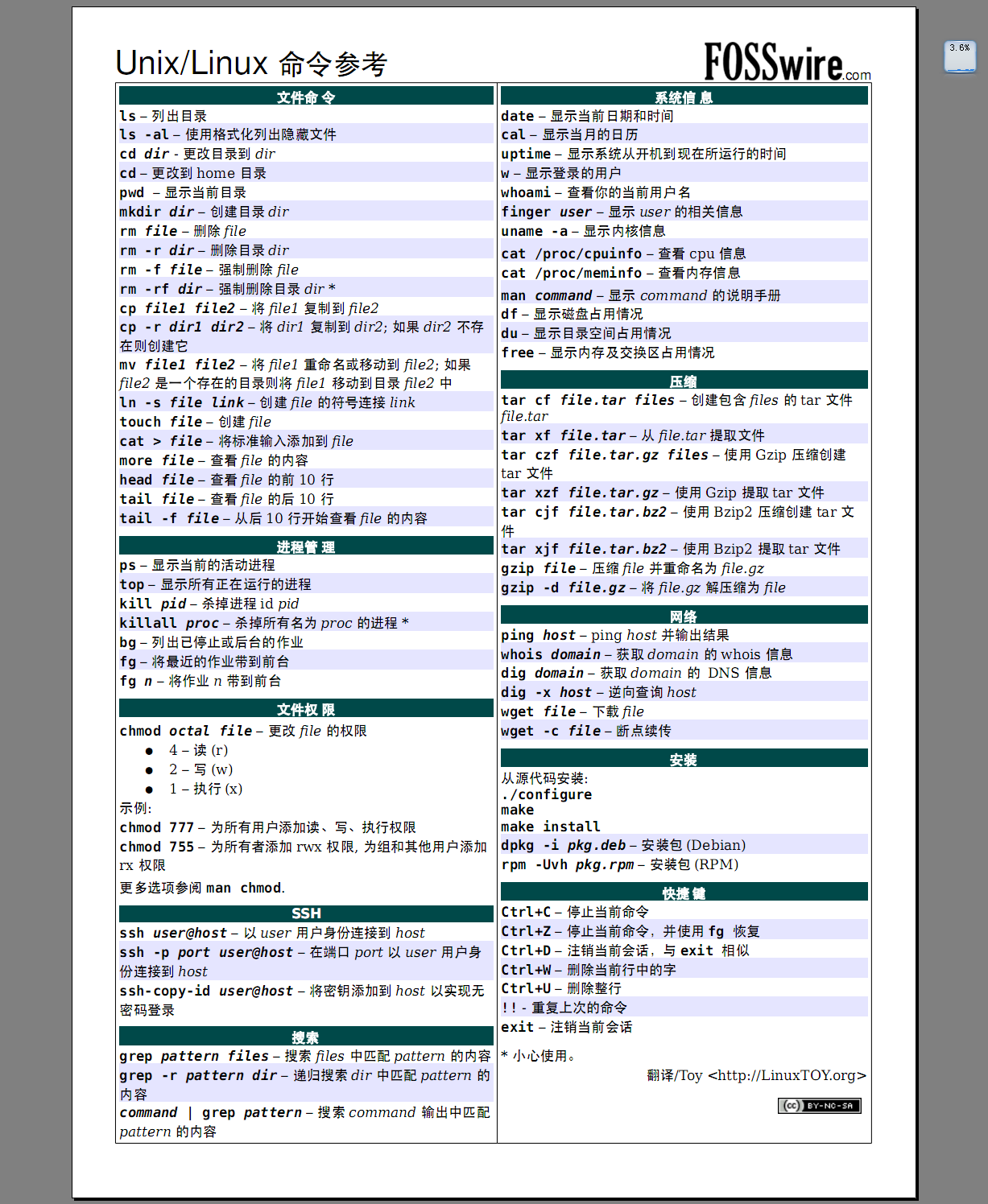
参考资料

* [GNU 站点](http://www.gnu.org/licenses) 介绍了 GNU GPL，它涵盖了 Linux 内核及其附带的大量有用的应用程序。另外还介绍了一个比 GPL 限制更少的许可 Lesser GPL （LGPL）。
* [UNIX](http://en.wikipedia.org/wiki/Unics)、 [MINIX](http://en.wikipedia.org/wiki/Minix) 和 [Linux](http://en.wikipedia.org/wiki/Linux) 在 Wikipedia 上都有介绍，另外还详细介绍了操作系统系列。
* [GNU C Library](http://www.gnu.org/software/libc/)，即 glibc，是标准 C 库的实现。它用于 GNU/Linux 操作系统，也可用于 [GNU/Hurd](http://directory.fsf.org/hurd.html) 微内核操作系统。
* [uClinux](http://www.uclinux.org/) 是 Linux 内核的一个移植，它可以在一些缺少 MMU 的系统上执行。这允许 Linux 内核在很小的嵌入式平台上运行，例如 PalmPilot PalmPilot Personal Digital Assistants (PDAs) 上使用的 Motorola DragonBall 处理器。
* “[使用 Linux 系统调用的内核命令](http://www.ibm.com/developerworks/cn/linux/l-system-calls/index.html)”（developerWorks，2007 年 3 月）对 SCI 进行了介绍，这是 Linux 内核中非常重要的一层，具有 glibc 的用户空间支持，可以在用户空间和内核之间启用函数调用。
* “[Linux 调度器内幕](http://www.ibm.com/developerworks/cn/linux/l-scheduler/index.html)”（developerWorks，2006 年 6 月）介绍了 Linux 2.6 中引入的新 O(1) 调度程序，这是一个非常高效的算法，可以扩展到很大数量的进程（线程），并且可以利用 SMP 系统。
* “[使用 /proc 文件系统来访问 Linux 内核的内容](http://www.ibm.com/developerworks/cn/linux/l-proc.html)”（developerWorks，2006 年 3 月）介绍了 /proc 文件系统，这是一个虚拟文件系统，为用户空间的应用程序提供了一种全新的方法与内核进行通信。这篇文章展示了 /proc 以及可加载内核模块。
* “[服务器诊所: 使虚拟文件系统工作](http://www.ibm.com/developerworks/cn/linux/l-sc12/index.html)”（developerWorks，2003 年 4 月）深入介绍了 VFS 层，它允许 Linux 通过一个通用接口支持各种不同的文件系统。这个相同的接口也可以用于其他类型的设备，例如 socket。
* “[Linux 引导过程内幕](http://www.ibm.com/developerworks/cn/linux/l-linuxboot/index.html)”（developerWorks，2006 年 5 月）介绍了 Linux 的引导过程，该过程将启动一个 Linux 系统，并且不管是从硬盘、软盘、USB 内存条还是通过网络引导系统，该过程是相同的。
* “[Linux 初始 RAM 磁盘（initrd）概述](http://www.ibm.com/developerworks/cn/linux/l-initrd.html)”（developerWorks，2006 年 7 月）介绍了初始 RAM 磁盘，它将引导过程与所引导的物理介质隔离开来。
* “[使用 SCTP 优化网络](http://www.ibm.com/developerworks/cn/linux/l-sctp/)”（developerWorks，2006 年 2 月）介绍了一种非常有用的网络协议：流控制传输协议，它的操作与 TCP 类似，但是增加了很多有用的特性，例如消息传递、 多宿主和多流。如果您对网络协议感兴趣，那么 Linux 与 BSD 一样，都是一种非常好的操作系统。
* “[Linux slab 分配器详解](http://www.ibm.com/developerworks/cn/linux/l-linux-slab-allocator/)” （developerWorks，2007 年 5 月）介绍了 Linux 内存管理中最有用的一个工具：slab 分配器。这种机制源自于 SunOS，不过它在 Linux 内核中找到了合适的位置。
* “[虚拟 Linux](http://www.ibm.com/developerworks/cn/linux/l-linuxvirt/)”（developerWorks，2006 年 12 月）介绍了 Linux 是如何充分利用具有虚拟化能力的处理器。
* “[Linux 和对称多处理](http://www.ibm.com/developerworks/cn/linux/l-linux-smp/)”（developerWorks，2007 年 3 月）讨论了 Linux 怎样利用提供芯片级多处理能力的芯片。
* “[探索 Linux 内核虚拟机](http://www.ibm.com/developerworks/cn/linux/l-linux-kvm/)”（developerWorks，2007 年 4 月）介绍了最近引入内核的虚拟化技术，它可以将 Linux 内核转换成其他虚拟化的操作系统的系统管理程序。
* 有关在用户空间对 Linux 进行编程的更多信息，请参考 Tim 撰写的 [*GNU/Linux Application Programming*](http://www.charlesriver.com/Books/BookDetail.aspx?productID=91525) 一书。
* 在 [developerWorks Linux 专区](http://www.ibm.com/developerworks/cn/linux/) 中可找到适合 Linux 开发人员的更多资源，包括 [Linux 教程](http://www.ibm.com/developerworks/cn/views/linux/tutorials.jsp) 以及上月 [读者最喜欢的 Linux 文章和教程](http://www.ibm.com/developerworks/cn/linux/top10/)。

[赞同2个人](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&refresh=1505277967634&focusedCommentId=16812637&refresh=1505278070042)表示赞同

* 无标签
* [编辑标签](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&refresh=1505277967634&focusedCommentId=16812637&refresh=1505278070042)

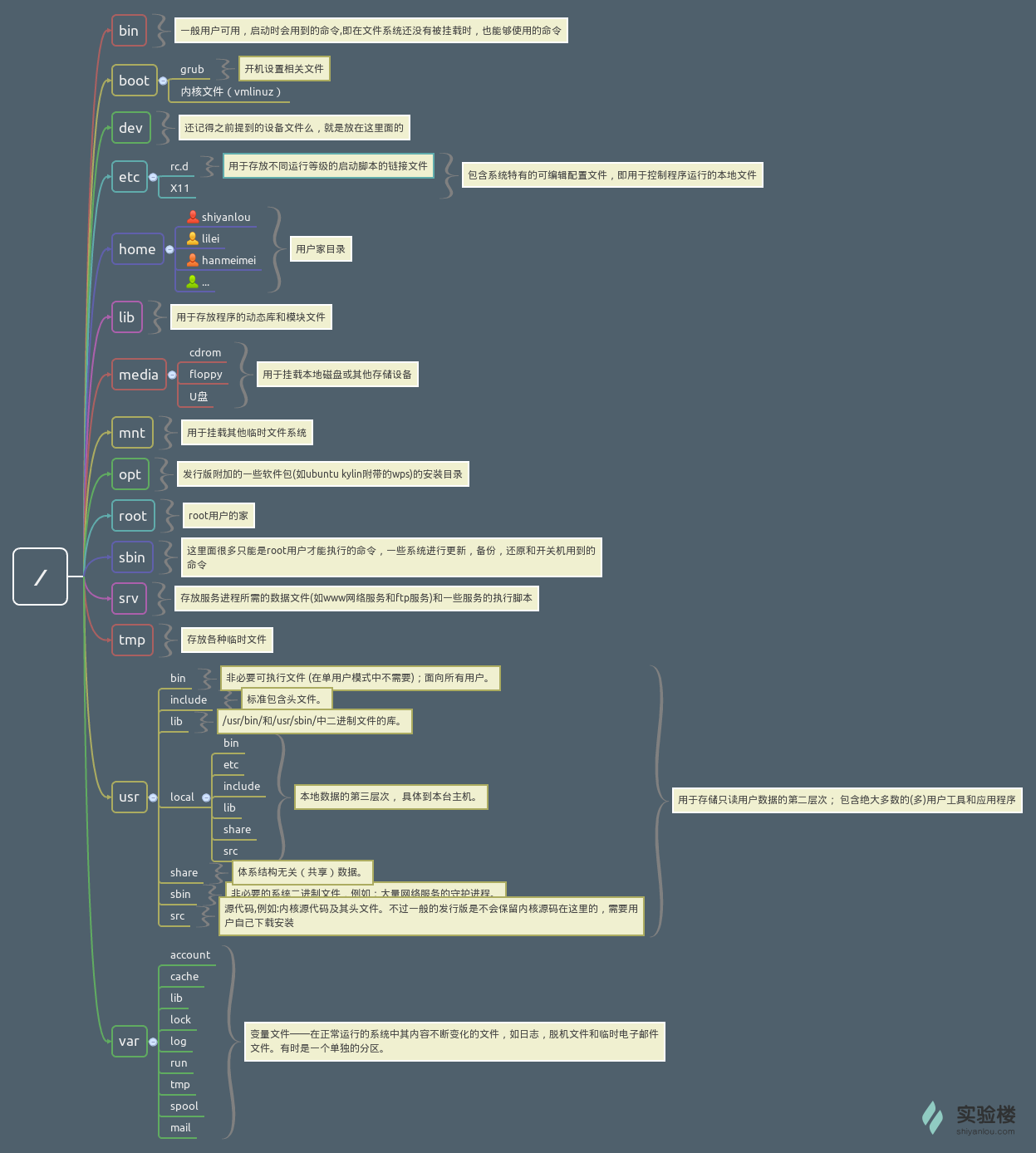
linux命令参考



* + [回复](http://km.makeblock.com/pages/replycomment.action?commentId=16812636&pageId=16812597)
  + [编辑](http://km.makeblock.com/pages/editcomment.action?commentId=16812636&pageId=16812597)
  + [删除](http://km.makeblock.com/pages/removecomment.action?commentId=16812636&pageId=16812597&atl_token=2f5d6c51b634014749422af7a1e3c73dcc4a1d20)
  + [赞同](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&refresh=1505277967634&focusedCommentId=16812637&refresh=1505278070042)
  + [昨天12:44 下午](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&focusedCommentId=16812636#comment-16812636)

[**王活生**](http://km.makeblock.com/display/%7Eshenghuowang@makeblock.cc) **发表：**

FHS（文件系统目录标准）



* + [回复](http://km.makeblock.com/pages/replycomment.action?commentId=16812637&pageId=16812597)
  + [编辑](http://km.makeblock.com/pages/editcomment.action?commentId=16812637&pageId=16812597)
  + [删除](http://km.makeblock.com/pages/removecomment.action?commentId=16812637&pageId=16812597&atl_token=2f5d6c51b634014749422af7a1e3c73dcc4a1d20)
  + [赞同](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&refresh=1505277967634&focusedCommentId=16812637&refresh=1505278070042)
  + [昨天12:46 下午](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&focusedCommentId=16812637#comment-16812637)

[**王活生**](http://km.makeblock.com/display/%7Eshenghuowang@makeblock.cc) **发表：**

**Linux资料**

* + [只要一部计算机，就可以创造出无限的世界](http://www.diguage.com/archives/22.html)，来个开胃菜，准备迎接下面的大餐！
  + [程序猿技术练级攻略](http://coolshell.cn/articles/4990.html)
  + [应该知道的Linux技巧](http://coolshell.cn/articles/8883.html)
  + [命令行是你的至交好友](http://www.oschina.net/translate/the-command-line-is-your-best-friend)
  + [Debuggex-可视化的正则表达式调试器](http://www.debuggex.com/)，正则表达式是玩转Linux的必备技能！这个必须掌握，理解得越透彻越好；
  + [LinuxCast](http://www.linuxcast.net/)，国人做的Linux视频网，里面有成套的、完整的Linux学习视频；
  + [永远的Unix](http://fanqiang.chinaunix.net/index_b.shtml)，这里荟萃了很多关于Unix/Linux的资料；
  + [28个Unix/Linux的命令行神器](http://coolshell.cn/articles/7829.html)
  + [简明VIM教程](http://coolshell.cn/articles/5426.html)
  + [vim 正则表达式 很强大](http://qianjigui.iteye.com/blog/368449)
  + [普通人的编辑利器——Vim](http://www.diguage.com/archives/51.html)
  + [Vimer的程序世界](http://www.vimer.cn/)
  + [手把手教你把Vim改装成一个IDE编程环境(图文)](http://blog.csdn.net/wooin/article/details/1858917)
  + [VIM 文件编码识别与乱码处理](http://edyfox.codecarver.org/html/vim_fileencodings_detection.html)
  + [StackOverFlow上关于vim最有信息量的答案](http://stackoverflow.com/questions/1218390/what-is-your-most-productive-shortcut-with-vim/1220118#1220118)
  + [在线交互式Vim教程](http://www.openvim.com/tutorial.html)，挺有意思，建议你全屏玩，否则估计显示不全；
  + [在线交互式Vim教程（中文版）](http://www.lx138.com/vim.php)
  + [Vim Adventures](http://vim-adventures.com/)，一个以Vim快捷键为闯关工具的游戏。D瓜哥玩到了第三关，看你能玩多少关；
  + [只用VIM制作EPUB](http://video.sina.com.cn/v/b/94538718-2170808240.html)
  + [谁说Vim不是IDE？（一）](http://www.cnblogs.com/chijianqiang/archive/2012/10/30/vim-1.html)
  + [谁说Vim不是IDE？（二）](http://www.cnblogs.com/chijianqiang/archive/2012/10/31/vim-2.html)
  + [谁说Vim不是IDE？（三）](http://www.cnblogs.com/chijianqiang/archive/2012/11/06/vim-3.html)
  + [谁说Vim不是IDE？（四）](http://www.cnblogs.com/chijianqiang/archive/2012/12/17/vim-4.html)
  + [All commands](http://www.commandlinefu.com/commands/browse)，非常强大的Linux命令在线查询工具，谁用谁知道！！
  + [Linux Wiki](http://linux-wiki.cn/wiki/%E9%A6%96%E9%A1%B5)，里面的指令讲解挺详细，排版也很好！
  + [Linux.cn](http://linux.cn/portal.php)，关注Linux的论坛；
  + [AWK简明教程](http://coolshell.cn/articles/9070.html)
  + [sed简明教程](http://coolshell.cn/articles/9104.html)
  + [Unix目录结构的来历](http://www.ruanyifeng.com/blog/2012/02/a_history_of_unix_directory_structure.html)
  + [Unix传奇(上篇)](http://coolshell.cn/articles/2322.html)
  + [Unix传奇(下篇)](http://coolshell.cn/articles/2324.html)
  + [Unix 40年：昨天，今天和明天](http://coolshell.cn/articles/1023.html)
  + [是UNIX还是Unix？](http://www.ituring.com.cn/article/37154)，哎，看来读的书还是少啊！
  + [批量重命名文件](http://linux-wiki.cn/wiki/zh-hans/%E6%89%B9%E9%87%8F%E9%87%8D%E5%91%BD%E5%90%8D%E6%96%87%E4%BB%B6)，这篇文章告诉你，Linux中水很深，一个简单的rename指令都可以堪比回字的写法。所以，好好学。
  + [理解inode](http://www.ruanyifeng.com/blog/2011/12/inode.html)
  + [KVM虚拟化原理与实践](http://smilejay.com/kvm_theory_practice/)，最新颖的Linux虚拟机技术！
  + [agentzh 的 Nginx 教程](http://openresty.org/download/agentzh-nginx-tutorials-zhcn.html)，nginx是目前使用最广泛的Web服务器！
  + [Nginx开发从入门到精通](http://tengine.taobao.org/book/index.html)，淘宝出品；
  + [一些鲜为人知的但却很有趣的Unix/Linux命令（一）](http://www.diguage.com/archives/88.html)
  + [一些鲜为人知的但却很有趣的Unix/Linux命令（二）](http://www.diguage.com/archives/90.html)
  + [一些鲜为人知的但却很有趣的Unix/Linux命令（三）](http://www.diguage.com/archives/96.html)
  + [一些鲜为人知的但却很有趣的Unix/Linux命令（四）](http://www.diguage.com/archives/111.html)
  + [一些鲜为人知的但却很有趣的Unix/Linux命令（五）](http://www.diguage.com/archives/8.html)
  + [危险的Shell，邪恶的HTML](http://www.soimort.org/posts/154/)
  + [回复](http://km.makeblock.com/pages/replycomment.action?commentId=16812638&pageId=16812597)
  + [编辑](http://km.makeblock.com/pages/editcomment.action?commentId=16812638&pageId=16812597)
  + [删除](http://km.makeblock.com/pages/removecomment.action?commentId=16812638&pageId=16812597&atl_token=2f5d6c51b634014749422af7a1e3c73dcc4a1d20)
  + [赞同](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&refresh=1505277967634&focusedCommentId=16812637&refresh=1505278070042)
  + [昨天1:00 下午](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&focusedCommentId=16812638#comment-16812638)

[**王活生**](http://km.makeblock.com/display/%7Eshenghuowang@makeblock.cc) **发表：**

**网络相关**

* + [互联网协议入门（一）](http://www.ruanyifeng.com/blog/2012/05/internet_protocol_suite_part_i.html)，从上到下，从下到上，透彻地分析互联网中的主要协议！
  + [互联网协议入门（二）](http://www.ruanyifeng.com/blog/2012/06/internet_protocol_suite_part_ii.html)
  + [SSH原理与运用（一）：远程登录](http://www.ruanyifeng.com/blog/2011/12/ssh_remote_login.html)
  + [SSH原理与运用（二）：远程操作与端口转发](http://www.ruanyifeng.com/blog/2011/12/ssh_port_forwarding.html)
  + [趣解HTTP状态码](http://www.diguage.com/archives/33.html)
  + [CDN技术介绍](http://rdc.taobao.com/blog/cs/?p=106)
  + [CDN技术实践](http://www.diguage.com/archives/27.html)
  + [回复](http://km.makeblock.com/pages/replycomment.action?commentId=16812641&pageId=16812597)
  + [编辑](http://km.makeblock.com/pages/editcomment.action?commentId=16812641&pageId=16812597)
  + [删除](http://km.makeblock.com/pages/removecomment.action?commentId=16812641&pageId=16812597&atl_token=2f5d6c51b634014749422af7a1e3c73dcc4a1d20)
  + [赞同](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&refresh=1505277967634&focusedCommentId=16812637&refresh=1505278070042)•[杨敬民](http://km.makeblock.com/display/%7Eyangjingmin) 表示赞同
  + [昨天12:55 下午](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&focusedCommentId=16812641#comment-16812641)

[**王活生**](http://km.makeblock.com/display/%7Eshenghuowang@makeblock.cc) **发表：**

**编程相关**

* + [Web开发：我希望得到的编程学习路线图](http://blog.jobbole.com/15213/)，这只是一个**学习指导路线**。大家可以把其中Ruby相关的，换成JSP、ASP.NET或者PHP等。另外，Ruby很好玩，感兴趣的可以学一学。（如果Java或者C#等没有学会的话，还是不要学了，Ruby的语法太灵活了，你会被搞蒙的。）
  + [23种设计模式的形象比喻](http://www.blogjava.net/qcyycom/archive/2012/04/16/374791.html)
  + [JDK里的设计模式](http://coolshell.cn/articles/3320.html)
  + [每个程序员都应该了解的内存知识](http://www.oschina.net/translate/tag/%E8%AE%A1%E7%AE%97%E6%9C%BA%E5%9F%BA%E7%A1%80)
  + [网站开发人员应该知道的62件事-思维导图](http://www.ccforward.net/2013/02/62_things_every_web_developer_should_know.html)，总结的非常好，建议认真看看学习一下；
  + [Codecademy](http://www.codecademy.com/)，在线交互式编程学习网站！很有趣！
  + [MySQL（微博）](http://weibo.com/ourmysql)，里面分享了很多干货
  + [图解SQL的Join](http://coolshell.cn/articles/3463.html)
  + [字符编码笔记：ASCII，Unicode和UTF-8](http://www.ruanyifeng.com/blog/2007/10/ascii_unicode_and_utf-8.html)，如果你写过程序，肯定就会遇到过乱码问题。那么，你可以看看这篇文章！
  + [Redis 设计与实现](http://www.redisbook.com)，可能一时半会用不上，备用吧。
  + [Joel Spolsky给计算机专业学生的七个建议](http://blog.jobbole.com/535/)
  + [想学编程？这样开始](http://blog.jobbole.com/37540/)
  + [Node入门](http://www.nodebeginner.org/index-zh-cn.html)，一本全面的Node.js教程 。现在JavaScript不仅仅能做网页，也可以搞服务端开发了。
  + [权威的前端性能指南](http://browserdiet.com/zh/)
  + [回复](http://km.makeblock.com/pages/replycomment.action?commentId=16812642&pageId=16812597)
  + [编辑](http://km.makeblock.com/pages/editcomment.action?commentId=16812642&pageId=16812597)
  + [删除](http://km.makeblock.com/pages/removecomment.action?commentId=16812642&pageId=16812597&atl_token=2f5d6c51b634014749422af7a1e3c73dcc4a1d20)
  + [赞同](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&refresh=1505277967634&focusedCommentId=16812637&refresh=1505278070042)
  + [昨天2:23 下午](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&focusedCommentId=16812642#comment-16812642)

[**王活生**](http://km.makeblock.com/display/%7Eshenghuowang@makeblock.cc) **发表：**

**网页开发**

* + [JavaScript 开发进阶：理解 JavaScript 作用域和作用域链](http://www.cnblogs.com/lhb25/archive/2011/09/06/javascript-scope-chain.html)，JavaScript的作用域以及作用域链必须非常熟悉；
  + [jQuery选择器大全](http://www.cnblogs.com/keepfool/archive/2012/06/02/2532203.html)，jQuery是任何一个搞网站开发人员，甚至移动应用开发人员必学JavaScript库；
  + [jQuery最佳实践](http://www.codeproject.com/Tips/573781/jQuery-Best-Practices)
  + [从零开始学习jQuery (五) 事件与事件对象](http://www.cnblogs.com/zhangziqiu/archive/2009/05/06/jQuery-Learn-5.html)，jQuery事件对象是比较容易忽略的地方，希望这个帖子能把这部分补上；这个系列的帖子都不错，建议看看；
  + [jQuery教程](http://www.cnblogs.com/zhangziqiu/tag/jQuery%E6%95%99%E7%A8%8B/)，一个比较系统的jQuery教程；
  + [Bootstrap](https://github.com/twitter/bootstrap)，Github上最火爆的项目。
  + [Bootstrap中文网](http://www.bootcss.com/)，专门的Bootstrap中文网；
  + [Bootstrap中文手册](http://xiemin.me/bootstrap-2.3.0/index.html)
  + [50个必备的实用jQuery代码段](http://www.cnblogs.com/gumutianqi/archive/2011/10/17/2215186.html)
  + [jsFiddle](http://jsfiddle.net/)，可以在网上调试JavaScript代码。还可以共享出来。
  + [HTML语义简析](http://justineo.github.com/slideshows/semantic-html/#/)
  + [梦想天空](http://www.cnblogs.com/lhb25/)，里面有超多设计相关的资源；
  + [优设](http://www.uisdc.com/)，网页设计。
  + [imageCSS](http://imagecss.gucheen.pro/index.html)
  + [LessCSS中文网](http://www.lesscss.net/)，CSS的一个“变体”，解决传统CSS的一些问题，不过最好还是要编译成CSS才能被浏览器识别；
  + [浏览器的工作原理](http://www.html5rocks.com/zh/tutorials/internals/howbrowserswork/)
  + [浏览器是如何工作的？](http://blog.csdn.net/zzzaquarius/article/details/6532299)
  + [分享8个超棒的免费高质量图标搜索引擎](http://www.cnblogs.com/lhb25/archive/2011/04/15/2009563.html)
  + [微软出版的IE兼容性测试工具Modern.IE](http://www.modern.ie/zh-cn)，可以试试，挺不错；
  + [web前端性能优化进阶路](http://blogread.cn/it/article/6217)
  + [分享一些D瓜哥攒的比较好的Web开发资料](http://www.diguage.com/archives/67.html)，这里分享了更多的关于网页开发的资料！
  + [回复](http://km.makeblock.com/pages/replycomment.action?commentId=16812643&pageId=16812597)
  + [编辑](http://km.makeblock.com/pages/editcomment.action?commentId=16812643&pageId=16812597)
  + [删除](http://km.makeblock.com/pages/removecomment.action?commentId=16812643&pageId=16812597&atl_token=2f5d6c51b634014749422af7a1e3c73dcc4a1d20)
  + [赞同](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&refresh=1505277967634&focusedCommentId=16812637&refresh=1505278070042)
  + [昨天12:57 下午](http://km.makeblock.com/pages/viewpage.action?pageId=16812597&focusedCommentId=16812643#comment-16812643)

[**王活生**](http://km.makeblock.com/display/%7Eshenghuowang@makeblock.cc) **发表：**

**挨踢书籍推荐**

* + [深入理解计算机系统（原书第2版）](http://book.douban.com/subject/5333562/)，老赵的文章，[“老赵书托（3）：深入理解计算机系统”](http://blog.zhaojie.me/2009/11/recommended-reading-3-csapp.html)，专门推荐这本书。自己看，不解释了；
  + [计算机程序的构造和解释](http://book.douban.com/subject/1148282/)，还是 老赵的文章，[“老赵书托（2）：计算机程序的构造与解释”](http://blog.zhaojie.me/2009/07/recommended-reading-2-sicp.html)；
  + [代码大全（第2版）](http://book.douban.com/subject/1477390/)，讲软件构建的，软件工程方面的书；学好编程后，可以看看，规范一下自己的编码习惯；
  + [计算机网络（第5版）](http://book.douban.com/subject/10510747/)，计算机网络是编程的基础课程，这个必须学好；
  + [现代操作系统（第3版）](http://book.douban.com/subject/3852290/)，和上面的《计算机网络》是同一作者，面试的时候，尤其是比较大的公司，会考察这方面的内容，基础课程，建议看看；
  + [Unix编程艺术](http://book.douban.com/subject/5387401/)，作者是Unix的开发人员，开源运动的推动者，主要讲解Unix/Linux的”野史”；当然，也讲解了很多Unix/Linux系统设计原理和设计思想。推荐！
  + [编程珠玑（第二版）](http://book.douban.com/subject/3227098/)
  + [编程之美](http://book.douban.com/subject/3004255/)，据说国内很多公司的算法题都是从这里出的，想找个好工作就认真看看吧；
  + [算法设计手册](http://book.douban.com/subject/4048566/)，据说这本书提供了大量可用于生产的代码，D瓜哥计划阅读一下；
  + [大话设计模式](http://book.douban.com/subject/2334288/)，这本书会告诉你，其实技术也很好看。
  + [重构](http://book.douban.com/subject/4262627/)，讲解如何重构代码，从中可以思考自己的代码，提高编码水平；

以上书籍注重基本功训练，这些都是基础东西，一通百通，建议都看看。下面给大家推荐一些更实用的数据。

* + [鸟哥的Linux私房菜.基础学习篇](http://book.douban.com/subject/4889838/)，Linux也是挨踢程序猿的基本技能；
  + [JavaScript高级程序设计（第3版）](http://book.douban.com/subject/10546125/)，JavaScript最棒的入门教程，没有之一；
  + [高性能网站建设指南](http://book.douban.com/subject/3132277/)
  + [高性能网站建设进阶指南](http://book.douban.com/subject/4719162/)，和上一本书《高性能网站建设指南》同一个作者，都是前端性能牛人Steve Souders的大作！貌似也只有影印版了。