编译内核详细配置

badicoubid 发表于 2006-6-29 16:42:00

一篇编译内核的详细配置文章[转]

前言之前言：谁是这篇文章的读者？

不习惯读英文资料的非 LINUX 高手
声卡不响光驱不转连不上网等等，只要有问题就行
开发嵌入式操作系统

现在流行的 ODL(only disk linux)中做内核部分，那些文章不介绍此点内容。我正在做
，完工后整理资料。
BY THE WAY ，想成为 LINUX 高手吗？你需要熟练掌握 KERNEL COMPILE 、XCONFIGRATER
、LINUXCONFIG、 SAMBA 四大利器，你随时可以找到后三者的大量相关资料，但内核编译
就只好啃为数不多的英文了。
笔者耗时 3 月，搜集并整理大量资料，在儿童节前连续工作 18 个小时后，给小企鹅送了这份礼物。
笔者自信是目前为止 KERNEL 编译方面最完备的中文资料（将不断翻译补充），这可是毕业论文哪！
为什么要放网上呢？首先找这方面资料太难了，文章对各位 LINUX 爱好者会有所帮助。其次，取之于网用之于网。
欢迎使用这篇文章，请随便引用，这才符合 LINUX 自由软件的精神嘛，不过别忘了提提我的名字，就算为我的辛苦付了点稿酬。
介于内核方面资料较少，欢迎在这篇文章中添加和修改内容，但不要过多删除，笔者列表要加入你的名字，让我们为把它变成数百页的资料而努力。OK，交个朋友吧， 我是玉玉安，email ：ziboyuyuan@263.net <ZIBOyuyuan@263.net>
目录

序言
近几年，linux 大行其道，令不满 windows 蓝屏的使用者跃跃欲试，结果发现 linux 安装不及 windows 方便，界面不及 windows 友好，配置不及 windows 容易，软件不及 windows 丰富
，以至浅尝辄止。
其实, Linux 有 windows 无可比拟的两个优势：网络应用和嵌入式技术，这也正是未来最有前途的方向。同时此课题是从理论上设计具有嵌入式 Linux 操作系统器件的重要组成部分。
如要涉足这两个方面，就必须对内核有深刻理解。当然，您可以从源代码入手，但前提是您拥有程序设计和操作系统等多方面专业知识，否则，就利用现成的 Linux kernel 从编译内核开始吧。不幸的是，内核编译方面的资料匮乏。以上两个原因使我写这篇论文成为必要。我可以自信得说，这是目前最详尽的内核编译方面的中文资料。
值得一提的是，我在搜集翻译资料的过程中，获得了操作系统、程序设计、硬件设备、网络通信等各方各面的知识，极大的拓宽了视野，真正学有所获。
感谢所有为 Linux 发展作出贡献的人，感谢所有 Linux 中文网站为促进 Linux 在中国的发展作出的不懈努力，他们是我搜集资料的来源。
特别感谢我的指导老师——官伯然教授和高斌博士，正是他们的辛勤指导让我顺利完成毕业设计。

西电科大 ：李玉元 2001/6/1
备注：#----------后跟小的选项
##---------后还有更细选项
注意------上机实践结果
第一章 内核编译的基础
第一节 内核简介
内核，是一个操作系统的核心，它负责管理系统的进程、内存、设备驱动程序、文件和网络系统，决定着系统的性能和稳定性。 就好比 DOS 下的 IO．SYS 和 MSDOS．SYS 一体，我
们可以把这两个文件叫做 DOS 的核心。Linux 也有它的核心，通常在根目录下，一个叫 vmlinuz 的文件。我们用这个文件来控制我们的整台 PC，包括周边设备和软硬磁盘机、CD－ROM、声卡等。简单地说，核心就是操作系统本身。没有了它，就像一个无人住的家，没有人去维持这个家的动作。一个安定的家需要一个很用心已能当机立断的主人：一部电脑也需要很有效率已稳定的核心，也就是操作系统。因此，核心是整个系统维持下去的关键。
Linux 的一个重要的特点就是其源代码的公开性，所有的内核源程序都可以在/usr/src/linux 下找到，大部分应用软件也都是遵循 GPL 而设计的，你都可以获取相应的源程序代码。任何一个软件工程师都可以将自己认为优秀的代码加入到其中，由此引发的一个明显的好处就是 Linux 修补漏洞的快速以及对最新软件技术的利用。而 Linux 的内核则是这些特点的最直接的代表。
想象一下，拥有了内核的源程序对你来说意味着什么？首先，我们可以了解系统是如何工作的。通过通读源代码，我们就可以了解系统的工作原理，这在 Windows 下简直是天方夜谭。其次，我们可以针对自己的情况，量体裁衣，定制适合自己的系统，这样就需要重新编译内核。在 Windows 下是什么情况呢？相信很多人都被越来越庞大的 Windows 整得莫名其妙过。再次，我们可以对内核进行修改，以符合自己的需要。这意味着什么？没错，相当于自己开发了一个操作系统，但是大部分的工作已经做好了，你所要做的就是要增加并实现自己需要的功能。在 Windows 下，除非你是微软的核心技术人员，否则就不用痴心妄想了。
先介绍一下编译核心的选项，希望能对大家消除对内核的神秘感有所帮助。
1.代码成熟等级
2..处理器类型和特色
3.对模块的支持
4.基本的选择
5.即插即用支持
6.块设备支持
7.网络选项
8.电话支持
9.SCSI 设备的支持
10.I2O 接口适配器
11.网络设备支持
12.配置业余无线广播
13.红外支持
14.ISDN 的文件系统
15.旧型的光驱类型(非 IDE 界面的光驱)

16.字符设备
17.USB 支持
18.文件系统
19.控制台驱动
20.声卡驱动
21.Kernel hacking
第二节 内核版本
由于 Linux 的源程序是完全公开的，任何人只要遵循 GPL，就可以对内核加以修改并发布给他人使用。Linux 的开发采用的是集市模型（bazaar，与 cathedral--教堂模型--对应），为了确保这些无序的开发过程能够有序地进行，Linux 采用了双树系统。一个树是稳定树（stable tree），另一个树是非稳定树（unstable tree）或者开发树（developm ent tree）。一些新特性、实验性改进等都将首先在开发树中进行。如果在开发树中所做的改进也可以应用于稳定树，那么在开发树中经过测试以后，在稳定树中将进行相同的改进。一旦开发树经过了足够的发展，开发树就会成为新的稳定树。开发数就体现在源程序的版本号中；源程序版本号的形式为 x.y.z：对于稳定树来说，y 是偶数；对于开发树来说，y 比相应的稳定树大一（因此，是奇数）。确定是以〞 root 〞的身份签入，然后 cd 到 /usr/src 。uname -r 这个指令将会显示版本。内核版本的更新可以访问<h ttp://www.kernel.org/>。
第三节 编译原因
Linux 作为一个自由软件，在广大爱好者的支持下，内核版本不断更新。新的内核修订了旧内核的 bug，并增加了许多新的特性。如果用户想要使用这些新特性，或想根据自己的系统度身定制一个更高效，更稳定的内核，就需要重新编译内核。
通常，更新的内核会支持更多的硬件，具备更好的进程管理能力，运行速度更快、更稳定，并且一般会修复老版本中发现的许多漏洞等，经常性地选择升级更新的系统内核是 Linux 使用者的必要操作内容。

为了正确的合理地设置内核编译配置选项，从而只编译系统需要的功能的代码，一般主要有下面四个考虑：
---自己定制编译的内核运行更快（具有更少的代码）
---系统将拥有更多的内存（内核部分将不会被交换到虚拟内存中）
---不需要的功能编译进入内核可能会增加被系统攻击者利用的漏洞
---将某种功能编译为模块方式会比编译到内核内的方式速度要慢一些
以上是针对成熟的 Linux 套件如 Redhat Linux 而言,我的目的是为建造嵌入式 Linux 操作系统做准备，也是必由之路。
第四节 准备工作
第一部分 新版本内核的获取和更新
Linux 内核版本发布的官方网站是<http://www.kernel.org/>，国内各大 ftp 上一般都可以找到某些版本的内核。新版本的内核的发布有两种形式，一种是完整的内核版本，另外一种是 patch 文件，即补丁。完整的内核版本比较大，比如 linux-2.4.0-test8.tar.b z2 就有 18M 之多。完整内核版本一般是.tar.gz（.tgz）文件或者是.bz2 文件，二者分别是使用 gzip 或者 bzip2 进行压缩的文件，使用时需要解压缩。patch 文件则比较小，一般只有几十 K 到几百 K，极少的会超过 1M。但是 patch 文件是针对于特定的版本的，需要找到
自己对应的版本才能使用。

编译内核需要 root 权限。把需要升级的内 拷贝到/usr/src/下（下文中以 2.2.16 的内核的 linux-2.2.16tar.gz 为例），命令为

#cp linux-2.2.16tar.gz /usr/src

先查看当前/usr/src 的内容，注意到有一个 linux 的符号链接，它指向一个类似于 linux -2.2.14（对应于现在使用的内核版本号）的目录。首先删除这个链接：

#cd /usr/src

#rm -f linux

现在解压下载的源程序文件。如果所下载的是.tar.gz（.tgz）文件，使用命令：

#tar -xzvf linux-2.2.16tar.gz

如果下载的是.bz2 文件，例如 linux-2.2.16tar.bz2，使用命令

#bzip2 -d linux-2.2.16tar.bz2

#tar -xvf linux-2.2.16tar

现在再来看一下/usr/src 下的内容，发现现在有了一个名为 linux 的目录，里面就是需要升级到的版本的内核的源程序。还记得那个名为 linux 的链接么？之所以使用那个链接就是防止在升级内核的时候会不慎把原来版本内核的源程序给覆盖掉了。现在也需要同样处理：

#mv linux linux-2.2.16

#ln -s linux-2.2.16 linux

如果还下载了 patch 文件，比如 patch-2.2.16，就可以进行 patch 操作（下面假设 patch-2.2.16 已经位于/usr/src 目录下了，否则需要先把该文件拷贝到/usr/src 下）：

#patch -p0 < patch-2.2.16

第二部分 准备主机板和相关硬件的说明手册

其实也不用太详细，只要知道您的硬件是属于哪一类型就行了。例如：有一张 SCSI 卡，那就要知道这张卡的名字，有一台 cd-rom,就要知道这台光驱是哪一种牌子的，是否为标准的 IDE/ATAPI 界面，还是另有专属接口卡呢？或者，主机版是否有支持 Triton 芯片（通常 586 以上的电脑常有），这些信息能帮助我们，使得设定变得清楚且容易。

因此，不管您有什么使用手册，准备好吧。即使现在不用，将来还是会用到的（设 X－window system 时要显示卡的手册）。

第三部分 检查声卡的 IRQ 设定和其种类

如果配有一张声卡，除了要知道卡的种类外（例如 Sound Blaster）还需要知道这张卡的 IRQ 地址。一般来说，卢卡的 IRQ 地址是 5 或 7 而 IO 地址则为 220。DMA 则 l，不过，有时不

同的声卡可能会有不同的设定。因为稍后的选项里，就会要填入这些数字。

第四部分 编译核心的硬件需求

在编译核心时，确定您的 RAM 最好在 8MB 以上， 否则可能会很慢而且问题会很多，记得查

看 swap 有没有打开（用 free 指令）。此外，最好不要超频，不然很有可能会发生 signal 11 的错误，使得编到一半的核心停了下来，其实编译核心就好比编译程序一样，只是因为构成核心的程序太多了，因此我们能小心尽量小心。

第二章 内核编译的流程

概述编译的流程：

编译开始----- make mrproper;检查所需的连接

配置核心
编译核心
编辑/etc/lilo.conf
重新启动新核心
重新启动机器
发现并修理故障（仔细看我的文章，应该没多少问题了）
第一节 编译开始
通常要运行的第一个命令是：
#cd /usr/src/linux
#make mrproper
该命令确保源代码目录下没有不正确的目标.o 文件以及文件的互相依赖。如使用刚下载的完整的源程序包进行编译，本步可以省略。而如果多次使用了这些源程序编译内核，那么最好要先运行一下这个命令。
确保/usr/include/目录下的 asm、linux 和 scsi 等链接是指向要升级的内核源代码的。它们分别链向源代码目录下的真正的、该计算机体系结构（对于 PC 机来说，使用的体系结构是 i386）所需要的真正的 include 子目录。如：asm 指向/usr/src/linux/include/asm-i386 等。若没有这些链接，就需要手工创建，按照下面的步骤进行：
cd /usr/include
rm -r asm linux scsi
ln -s /usr/src/linux/include/asm-i386 asm
ln -s /usr/src/linux/include/linux linux
ln -s /usr/src/linux/include/scsi scsi

这是配置非常重要的一部分。删除掉/usr/include 下的 asm、linux 和 scsi 链接后，再创建新的链接指向新内核源代码目录下的同名的目录。这些头文件目录包含着保证内核在系统上正确编译所需要的重要的头文件。也是上面又在/usr/src 下"多余"地创建了个名为 linux 的链接的原因之一.
一旦万事俱备，转到/usr/src/linux。现在你也许想停下细读一下文档文件，实际上如果你有些特别的硬件，或几种光驱驱动程需要自己动手设置，他们通常这样做，当引导时这些驱动程序将给出警告，这并不碍事他们照常工作少，阅读扩展名为.txt .h. c 的文件。通常我发现他们具有共性且易于配置。如果你不想冒险，你没必要做。记住你照样可以解开 tar 文件(或再次安装.rpm 文件)恢复前的文件。
第二节 配置内核 核心内容
接下来的内核配置过程比较烦琐，但是配置的适当与否与日后 Linux 的运行直接相关，有必要了解一下选项的设置。

配置内核可以根据需要与爱好使用下面命令中的一个：
#make config（基于文本的最为传统的配置界面，不推荐使用）
#make menuconfig（基于文本选单的配置界面，字符终端下推荐使用，必须安装 ncurses-dev 和 tk4-dev 库）
#make xconfig（基于图形窗口模式的配置界面，Xwindow 下推荐使用）
#make oldconfig（如果只想在原来内核配置的基础上修改一些小地方，会省去不少麻烦）

如果不能使用 Xwindow，那么就使用 make menuconfig 好了。界面虽然比上面一个差点，总比 make config 的要好多了。
选择相应的配置时，有三种选择，它们分别代表的含义如下：

Y－-将该功能编译进内核

N－-不将该功能编译进内核

M－-将该功能编译成可以在需要时动态插入到内核中的模块

在每一个选项前都有个括号，但有的是中括号有的是尖括号，还有一种圆括号。 用空格键选择时可以发现，中括号里要么是空，要么是"*"，而尖括号里可以是空，"*"和"M"。这表示前者对应的项要么不要，要么编译到内核里；后者则多一样选择，可以编译成模块。而圆括号的内容是要在所提供的几个选项中选择一项。
在编译内核的过程中，最烦杂的事情就是这步配置工作了，不清楚到底该如何选取这些选项。实际上在配置时，大部分选项可以使用其缺省值，只有小部分需要根据用户不同的需要选择。选择的原则是将与内核其它部分关系较远且不经常使用的部分功能代码编译成为可加载模块，有利于减小内核的长度，减小内核消耗的内存，简化该功能相应的环境改变时对内核的影响；不需要的功能就不要选；与内核关心紧密而且经常使用的部分功能代码直接编译到内核中。下面对选项分别加以介绍。

1．Code maturity level options 代码成熟等级
此处只有一项：prompt for development and/or incomplete code/drivers，如果要试验现在仍处于实验阶段的功能，比如 khttpd、IPv6 等，就必须把该项选择为 Y 了；否则可以把它选择为 N。 在 Linux 的世界里，每天都有许多人为它发展支持的 driver 和加强它的核心。但是有些 driver 还没进入稳定的阶段。但其作者很欢迎其他人去测试这些 driver 并提出一些 bugs。这个问题是说，有一些 drive 还在做测试中，问您是否要选择这些 drive 或支持的程序码。
如果键入 Y，往后将会出现一些还在测试中的东西给您做选择。（像 Java 的程序码和 PCI bridge），台则就键入 N。

2. Processor type and features 处理器类型和特色
#Processor family (386, 486/Cx486, 586/K5/5x86/6x86, Pentium/K6/TSC,PPro/6x86MX) [PPro/6x86MX] ------选择处理器类型，缺省为 Ppro/6x86MX。它会对每种 CPU 做最佳化，让它跑得快又好。一般来说，没有选择正确的 CPU 并不会有重大的影响（特别是选择 386 ，这样编译出来的核心也许会比较小但它的速度可能就会变慢了）。所以，最好要知道您的 CPU 是哪一种。不过，如果您的 gCC 编译器是 2.7.0 版以前的。那么只能选择 386 或是 486。
#High Memory Support------内核支持的最大内存数，缺省为 1G。可以支持到 4G、6.4G，一般可以不选.
#Math emulation------这项询问是否需 L1nux 核心模拟数学浮点运算器。如果有 486Dx、AMD 以及 Pentium 机器的话，这个选项就不必选了，因为它们都有内建的浮点运算器。协处理器是在 386 时代的宠儿，现在早已不用了。不过，对于有内建浮点运算器的人来说，选了这个选项并不会因此让内建的浮点运算器失效。但它会增大核心约 45KB。
#MTTR （memory type range register）support------选择该选项，系统将生成/proc

/mtrr 文件对 MTRR 进行管理，供 X server 使用。同时用来启动 pentinum pro 和 pentinum II 的特殊功能，如果你用的不是这类 CPU 就选 N，否则也仅仅是使内核变大而已。
#Symmetric multi-processing support------对称多处理支持。除非有多个 CPU，否则就不用选了。

3. Loadable module support 对模块的支持.
首先，了解一点关于模块的知识。模块就像你特意插入核心中的某些东西，如果办公室有一个小网络并且有时想用一下(但并不经常)，也许你想把网卡编译成一个模块。使用这个模块，机器必运行和存取/libs 下的模块，意思是驱动程序(IDE,SCSI 等但必须是 NFS 支持的网卡)，文件系统(通常是 ext2 但也可以是 nfs)和核心类型(最好是 elf)必须编译在内核并且不能是模块，模块只有核心引导时才起作用，驱动程序(来网络)的存取，和文件系统安装。这些文件必须编译在核心内否则将能安装启动分区。如果安装启动分区和网络，你需要网络系统文件，和已经编译的网卡。为什么要使用模块? 模块化使核心变的更简捷，它减少核心释放大量的受保护的空间。模块的安装和卸载使用的空间是可重复分配利用的。如果你打开机器有 90%以上的时间用到一个模块，编译它。运用这类模块是浪费内存的，原因是一旦你编译了模块它们同样将占用大量的内存，核心需要一些代码来挂上模块。记住，核心在保护空间运行，但模块并不是。这么说，并不经常使用我的设备，把它编译成只支持 ext2,ide 和 elf。而一直使用的网卡，把其它的编译成模块：如 a.out, java, floppy, iso9960, msdos, minix, vfat,smb,nfs,smcultra(ethern etcard),serial,printer,sound,ppp,等等。它们许多只是在这或那用上那么几分钟。严格的说,这样做会使核心增大许多而降低它的执行速度。这时我们就可以把这些可能会用的驱动程序编译成一个一个的模块，在需要用的时候才用 insmod 这个指令加入核心，不用的时候也能 rmmod 把它从核心移除，或是用 lsmod 察看目前所载入的模块。这里面有三项：
#Enable loadable module support------除非准备把所有需要的内容都编译到内核里面，否则该项应该是必选的。
#Set version information on all module symbols------通常，我们更新核心版本之后，模块要重新的编译。这个选项使您不必更新编译模块而能使用以前的模块。可以不选它。但如果您选 y，则按照它的说明，您必须有 genksyms 这个程序（可用 whereis 指令查看有无此程序）。
#Kernel module loader------让内核在启动时有自己装入必需模块的能力，建议选上。
注意：在开机就会 mount 上来的 partition 的 FS 、device driver 记得要 compiler 进 kernel，不能把它弄成 modules。请不要夸张到为了完全模组化而忘了把 ext2fs 和 IDE dirver compiler 进 kernel 里。

4. General setup 普通的属性设置
这部分内容非常多，一般使用缺省设置就可以了。下面介绍一下经常使用的一些选项：
#Networking support------网络支持。因为在 Linux 里面，有虚拟的网络设备(1ookback)，可以模拟整个网络。而且，一些程序需要它。必须，没有网卡也建议你选上。注意：选 N，则 7.(Networking options 网络选项)和 11.(Network device support 网络设备支持) 不会出现。
#Limit memory to low 16MB ------大部分的人这一选项 N。除了主机板没有办法处理 16MB 以上的内存，或者有超过 16MB 以上的内存但却常常发生一些很奇怪的问题。这时，您可以试试这个选项。有些主机板对超 16MB 内存的处理并不是很好，通常这些都是旧型

的主机板。还有，在说明文件中有提到，如果内存超过 64MB 的话，用 LILO 加一些参数给 Linux 核心（例：mem=80M），并且把您主机板上的 Cache 加到 512K。这样，整体效率才能
提升。
#PCI support ------PCI 支持。如果使用了 PCI 的卡，当然必选。
#PCI bios support ------主机板是否有 PCI 界面。如果有，则您必须回答 y。
PCI 是 586 电脑的主要界面（一些 486 主机板上也有），这个界面能让您插入所谓的 PCI 显示卡，或是 PCI 的网络卡等。这种界面是现在电脑的主要趋势，因此如果有 PCI 的插槽。您就可以选 Y。除了一些很旧很旧但有支持 PCI 的主机板外（这些有 bugs 的旧型主机板可能会因为这个选项而让核心挂掉）。
#PCI access mode (BIOS, Direct, Any) [Any] ------设置 Linux 探测 PCI 设备的方式。
选择“BIOS”，Linux 将使用 BIOS；选择“Direct”，Linux 将不通过 BIOS；选择“Any”，Linux 将直接探测 PCI 设备，如果失败，再使用 BIOS。
#Support for hot-pluggabel devices ------热插拔设备支持。支持的不是太好，可不选。
#PCMCIA/CardBus support------PCMCIA/CardBus 支持。有 PCMCIA 就必选了。
#PCI bridge optimization (experimental) ------在某些支持 BIOS 上，它能让存取速度加快，建议是选 Y。
#Backward-compatible /proc/pci------设备兼容，自己看 help。
#System V IPC 如果将来想编译 dosemu（DOS 模拟器），则这个选项一定要选，它是一个让各个程序（process）同步且能彼此交换数据的函数库和一些系统的调用，没它，很多的程序将会无法执行。
#BSD Process Accounting------
#Sysctl support------除非你的内存少的可怜，否则你应该启动这个功能，启用该选项后内核会大 8K，但能让你直接改变内核的参数而不必重新开机。
#Kernel support for A.OUT binaries ------a．out 的执行文件是比较古老的可执行码，用在比较早期的 UNIX 系统上。Linux 最初也是使用这种码来执行程序，一直到 ELF 格式的可执行码出来后，有愈来俞多的程序码随着 ELF 格式的优点而变成了 ELF 的可执码。将来势必完全取代 a．out 格式的可执行码。但目前由于沿有许多的程序还没有取代过来，所以只好选择 Y，等将来有一天，全部的程序都变成了 ELF 的天下时，那时再 disable 掉。
#Kernel support for Linux/Intel ELF binaries ------由上所述，这个当然 y 哩，因为目前 gcc－2.7.0 以上的都有支持 ELF 了，如果没有选择这一项，可能会使用相当多的程序因此无法执行。
注意：编译模块成 ELF 和编译支持 ELF 二进制。不编译适当的支持’’gotcha’’是明智的，如果机器结构是 Pentium 或 486 你将得到高效的代码，但一个 386 的核心将运行在 32-bit compatible clone；一个 Pentium 核心将不。为大多机器制作一张紧急启动盘，最好在 386 下编译，而 386 并不能运行在 Pentium 下编译的核心。
另外一点要注意的，你不能同时把 a.out 和 ELF 支援编译成 modules，否则当你为了能够使用 insmod 而用 insmod 来载入 a.out/ELF modules 时会有 Catch/22 状况发生。如果你的系统主要是 ELF 而你偶尔会需要用到 a.out，你可以把 a.out 支援编译为 modules，否则你最好把它直接放入 kernel 之中。如果你还没进入 ELF 的世纪，在 compiler kernel 时可以直接把 ELF 支援去掉。
#Kernel support for JAVA binaries ------ 这一项是正在做测试中的产品，但是如果想写有关 Java 的程序，希望它能在 Linux 的机器上跑。那么，可以选择把它编成一个模块

或是直接把它编进核心里。
#Power Management support ------电源管理支持。
##Advanced Power Management BIOS support------高级电源管理 BIOS 支持。这通常是用在笔记本电脑上的东西，如果您有 APM 的 BIOS，支持省电的设备的（有电池的那种），那么您可以选上这项，一般人这一项是选 n，以避免一些可能会发生的问题。后有 8 个选项。
#Parallel port support ------串口支持。

5. Plug and Play configuration 即插即用支持
Linux 对即插即用目前支持的不如 Windows，好有些情况下会和其他设备产生冲突（I/O，DMA，IRQ 等）。这个选项对 PCI 设备没有影响，因为他们天生就是 PNP 设备。。
#Plug and Play support (CONFIG_PNP) ------ 选择“y”，内核将自动配置即插即用设备。原来 PNP 还有这个意思。
ISA Plug and Play support ------ 选择“y”，内核将自动配置基于 ISA 总线的即插即用设备。

6. Block devices 块设备支持
这个就得针对自己的设备情况来选了：
#Normal PC floppy disk support (CONFIG_BLK_DEV_FD) [Y/m/n/?]------普通 PC 软盘支持。
#Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support ------选择“y”，内核将提供对增强 IDE 硬盘、CDROM 和磁带机的支持。在硬盘没有做得很大的时候，一般的 IDE 卡和 BIOS 只能支持小于 540 MB 的硬盘。不但如此，那时也只能支持二颗硬盘。但现今的硬盘动不动就是 1GB 以上，今年主流是 30~50G，而且常常都会超过一二颗硬盘。如此一来，新的主机板就开始支持加强型的 IDE 界而（Enhanced IDE），以支持到 540MB 以上的硬盘。所以，如果您的 IDE 界面是 Enhanced 的，请您选 Y，底下就会出现八部分 IDE 界面的
选项，这些选项能加快您的 IDE 界面的速度和对某些芯片做一些最佳化。但如果您的硬盘或光盘全都是 SCSI 界面的，那么选 N 以跳过下面选项。
#use old disk-only driver on primary interface------通常是选择 N。因为我们有其他新的 drivers 可用。这个选项的意思是说，如果您的 IDE 界面是很旧很旧的那种的。那么，就可以使用这个 drives 驱动那个旧型的 IDE 界面（可装二台硬盘；或是一台硬盘，一台光盘）。而现今流行的 Enhanced IDE 则有两个界面，共可以接四台硬盘。我们稍后会有 driver 支持它。
＃i nclude IDE／ATAPI CDROM support------如果希望核心支持 IDE／ATAPI 界而的光驱，选择 Y。如果有光驱，但它附有一张接口卡，必须把排线接到那张专属接口卡上；或者是接到声卡上的，则这个选项也需要选 N，稍后我们会有非 IDE 的光驱厂牌让我们挑选。现今的光驱通常是 IDE／ATAPI 界而的，所以这个选项通常是 Y。
#Support removable IDE interfaces（PCMCIA）------这个选项对大部分的人全选 n，除非您有 PCMCIA 的东西，这通常是笔记本电脑上看得到的东西。 PCMCIA 是一个组织，在
以前是设计内存条的。但现在他们对于 PC CARDS 定了一个标排，并很广泛的应用在 1apt ap 的电脑上。不但有所谓的 PCMCIA 的硬盘，甚至有网络卡、SCSI 卡等，不过，大部分的

人并不需要这个选项。
这个选项选完后，以下则是 Linux 核心对几种芯片的 IDE 界面做修正或是加强它。
#CMD640 chipset bugfix / support------ 很多 486 和 586 的主机板都是用 CMD640 的芯片，它是 Neptune 芯片和 SIS 芯片的结合。不过，这种芯片有它的缺点，在许多的情形下，它会造成数据的流失和错误。如果您选了这一项，则 Linux 核心会为您小心的寻找这些错误并修正它。而且，它会打开对二个 IDE 界面的支持。不过，在它的说明文件中提到，如果您的主机板没有 PCI 界面只有 VESA 总线界面却希望有这项功能的话，则您必须传一些
参数给核心（ideo＝cmd640__vlb）。如果不确定上面所说的，选择 Y。
#CMD640 enhanced support------一般来说，对于硬盘的存取速度来说，有所谓的 PIO MODES 值设定，现今的 IDE 界面及 BIOS 应该都能侦测到正确的硬盘 PIO MODE 值了。此值
愈高表示硬盘的存取的速度愈快。可是，有些主机板的 B10S 还是旧式的，不能抓到比较高的 PIO MODE 值，如此一来，便不能发挥整台硬盘的效率。这个设定告诉读者说：如果您的 IDE 界面是 CMD640 为基础的界面，但是您的 BIOS 并不能抓到正确的 PIO MODE 值，那么
，这个选项可以自动的找到硬盘正确的 PIO MODE 值。
#RZ1000 chipset bugfix / support------这个选项如同前面的 CMD640 一样。不过，它的芯片是 RZ1000 的芯片，这种芯片是以 Neptune 芯片为主的一种芯片，而目、有很多的 486 和 586 的主机板都在使用它。可以查查主机板的说明书或是 IDE 接口卡的说明书做确定。文件上提到，选择这个将会降低一些速度，但是数据能百分之百的正确。
#Intel 82371 PIIX（Triton I / II） DMA support------对于 586 的 Pentium 电脑来说，有相当多的主机板都是用 Intel 的 Triton 芯片，使用这种芯片的最大好处是支持直接内存存取 DMA，而节省您的 CPU 时间。在以往还没有 DMA 这个东西时，读取硬盘需要耗用许多的
CPU 时间。如此一来，CPU 被占用，就不能充分的发挥它的功效。后来，DMA 出来后，硬盘
的读取便靠 Triton 或其他有支持 DMA 的 IDE 界面的芯片，直接与它们做沟通，而节省了大量的 CPU 时间，但这必须您的硬盘和主机板有同时支持 DMA 的 IDE 界面的芯片，直接与
它们做沟通而节省了大量的 CPU 时间。但这必须您的硬盘和主机板有同时支持 DMA MODE
才行。
##other IDE chipset support------
如果这上选项选 y，则会出现下列六种其他的芯片或厂牌供您选择。
＊ NOte： most of these also require special kernel boot paraaeters
ALI M14xx support
DTC－2278 support
Holtek HT65608 support
PROMISE DC4030 support
QDI QD6580 support
UMC 8672 support
上面这六种厂牌的芯片依硬件配备而使用，但它们有共同的特点就是必须传一些参数给核心。如果找不到您的芯片，那么也没关系。上这些选项只不过对这些芯片做最佳化罢

了。
Additional Block Devices 其他的块设备
#Loopback device support------大部分的人这一个选项都选 N，因为没有必要。这个选项的意思是说，可以将一个文件挂成一个文件系统。如果要烧光盘片的，那么您很有可能在把一个文件烧进去之前，看看这个文件是否符合 IS09660 的文件系统的内容，是否符合您的需求。而且，可以对这个文件系统加以保护。不过，如果您 想做到这点的话，您必须有最新的 mount 程序，版本是在 2.5X 版以上的。而且如果您希望对这个文件系统加上保护，则您必须有 des.1.tar.gz 这个程序。注意：此处与网络无关。
#Multlnle devices driver support------这个选项可以让把整个硬盘分区变成一个单独的区块设备，您必须有 md035．tgz 这个程序。而且在做这件事之前请将您的硬盘备份，因为它尚在测试阶段。一般人对这个选项是选 N。
#RAM disk support------如果使用过 DOS 下的 ramdrive 程序，应该能了解这个选项的意义。它可以把内存当成硬盘来做存取就如同一般的硬盘一样，可以 format 它，或是放一些文件在里头。然后，当您关机这些数据也随着之而去了。如果的 RAM 够大，可以考虑玩玩这选项，但一般人都不需要。
#XT hard disk support------支持 XT 的古董硬盘，这是 IBM 电脑时代的东西，如果您还有这种很旧很旧的硬盘。那么，您可以把它编进核心或是编成一个模块。大部分的人这个选项都是选择 N 的。
#Compaq SMART2 support------
#Mulex DAC960/DAC1100 PCI RAID Controller support------RAID 镜像用的。
#Logical volume manager（LVM）support------逻辑卷管理支持。
#Multiple devices driver support------多设备驱动支持。
#RAM disk support------RAM 盘支持。

7. Networking options 网络选项
这里配置的是网络协议。
#Packet socket ------ 选择“Y”，一些应用程序将使用 Packet 协议直接同网络设备通讯，而不通过内核中的其它中介协议。
#Kernel／User Network link driver------这个是在测试中的程序码，一般人不需要用。依它的说明，它允许在核心、模块或程序间的某些部分间，彼此做双向的沟通。如果想使用 arpd，则这个程序码就要加进核心里。
#TCP/IP networking------选择“Y”，内核将支持 TCP/IP 协议。这个选项无论如何请您选择 Y，即使没有网络卡，或是没有连到网络上的设备，在 linux 上仍有所谓的 lookback 设备而且有些程序需要这个选项。在说明文件中提到，如果您没有打开这个设定，则 X－window system 可能会有问题（回为它也需要 TCP／IP）。
#Network firewalls------选择“Y”，内核将支持防火墙。Firewalls 依英文看是防火墙。在网络愈来愈发达的今天，网络安全的考虑也愈来愈重要了。在局域网上找一台电脑来保护自己的考虑也愈来愈多了。可以在一局域网上找一台电脑来保护自己区域内的电脑。这样的结果是，所有外部的电脑如果要连进内部的电脑就必须通过这台装有 Fire walls 电脑的同意。所以，如果您希望这台电脑有着过滤网络的功能的话，那么这个选项要选 Y。而且，等一下有个 IP firewalling 的选项也要选 y；但下面的 forwarding／gat ewaying 要选 n，如此才能让它正常动作。大部分的人这个选项选 N。
#Network alasing------允许有多个 IP 地址。
#IP：forwarding／gatewaying ------ 和 Firewall 相反，这个选项是用来疏导网络的

。一个 gateway（也就是 router），要帮忙疏导两个网络间的数据传送。这台机器必须要有两张网络卡，连接两个个同的网络，做疏导网络的工作。如果选择了这个选项，则表示想让这台负责做 router。那么，就必须有两张网络卡了。另外有一种情形是，如果您有 MODEM（通过串列界面以 SLIP 和 PPP 协议）和网络卡，并用它们来连上 Internet。这时您也一样可以执行 IP－routing 服务，也需把这个选项打开。

#IP：multicasting------所谓的 multicasting 是群组广播，它是用在视频会议上的协议，如果想送一个网络封包（网络的数据），同样的一份数据将送往十部机器上。您可以连续送十次给十台机器（点对点的传送），也可以同时送一次，然后让十台机器同时接收到。当然后者比前者好，由于视频会议要求是最好每个人都能同时收到同一份信息，所以如果您有类似的需要，这个选项就要打开。同时您还必须去找相关的软件。

#IP： accounting------如果您打开这个选项，您就可以在／proc／net 下看到系统对于整个网络状况的纪录。所以一般的人这个选项都是选 y。而且，如果您设计把这台 Linux 机器当 router 用，那么读者可以因为这个选项而获得许多有关于网络 lP 控制的信息和它的输送情形。不过，您必须在底下的选择中选择 proc 系统（其实 proc 文件系统一定要选，不选很多程序会不能用！）

#IP：aliasing support------也许您只有一张网络卡，但经由这个设定，您可以拥有数个 IP 地址。假设您已经有一个 lP 地址了，您还想再加入其他的 IP 地址，这时，您可以依下面的程序来做。

在 shell 下键入：

sunlyy：／#ifconfig eth0：0 其他的 IP 地址（这个 IP 地址不能与其他机器重复）

sunlyy：／# route－add－host 其他的 IP 地址 dev eth0： 0

如此您就可以同时拥有两个 IP 地址了。当然，如果想把这个 IP 地址去除，那么可以键人：

sunslyy：／# ifconfig eth0：0－ IP 地址

这样您就可以把加入的 IP 地址去除。不过在使用此选项前，前面的那一个 aliasing Net work 选项也要选上去。

#IP： PC／TCP compatibility mode------大部分人都选 n。除非在使用 DOS 下的 NCSA－TCP／IP 软件连进 Linux 机器时遇到了困难，或者有不相容的情形出现。这时，您可以试着把这个选项打开，看看是否能解决这个问题。

#IP Reverse ARP------如果您的 Linux 希望提供 bootd 的服务，就是让没有硬盘或软盘也能够开机并且上网络，只要它们有网络卡有连接到网络的话。此时，您必须执行一个指令叫 rarp 来设定哪些电脑的网络卡可以如此。不过一般人都没有这个需要，所以答 n。

#IP：Disable Path MTU Discovery （normall yenable）------ 大部分的人这个选项是选择 N。除非发现用 DOS 下的 ncsa 的 telnet 程序连到 Linux 机器上出问题。这是很多人的

问题，如果发生了 DOS 下的 telnet 程序不能连进 Linux 时，除了可以改用 Nsysutel 的 teln et 程序来解决外，还可以在编译核心的时候，把这一项选 Y。 MTU（Maximal Transfer Unit）叫做最大的传输单位，是说我们一次送往网络的信息大小。而 Path MTUD iscov ery 的意思是，当 Linux 发现一些机器的传输量比较小时，我们会分送网络信息给它。如此可以增加网络的速度，所以我们大部分都选 N，也就是 Enable。

#Ip：Dorp source routed frames------通常我们一个网络的封包在丢出去后就不管它了，不过，在 TCP／IP 协议里，您可以设定让那些帮您绕路的机器回送一个是否这个封包已经送达了的消息。不过，这会导至网络安全上的问题，所以很少用，一般来说我们选择 Y。

#IP：Allow large windows（not recommended if＜16Mb of memory＝）------如果有超过 16MB 以上的内存，那么建议打开这个选项，可以增加传输的速度。在一般长距离的网络传输下要预备传输的数据可以先储存在缓冲区，等到对方的回应时再一次会过去。因此，您必须有内存来作为缓冲区。

#The IPX Protocol------IPX 是一种传输协议，它是 Novell 的一种网络协议，通常用在区域或是 Windows 的网络下。如果您希望 Novell 的机器资源共享（例如用他们的打印机或是硬盘），那么这个选项则要选 y。至于存取文件的格式是 NCPFS 的格式，稍后把这个文件系统选上来，以便支援这个文件系统。 如此一来，您就可以通过 Novell 的 IPX 通讯协议去存取它们的数据了。或者，您希望从 dosemu（DOS 的模拟器）里用 IPX 协议，这时也要把它选进来。

#Full internal IPX network------提供了一个完整的内部 IPX 网络，预设选项是 N，因为它可能会让一些应用的服务程序（RIP／SAP）当掉。

#Appletalk DDP------AppleTalk 是存在于苹果机上的一种通讯协议，用来苹果电脑之间的网络通迅，通过 AppleTalk，彼此的电脑间可以打印和分享文件。如果您需要连上这样的网络，可以把这个选项打开，如此就能加入他们与这些电脑做沟通了，或者把这项编成一个模块亦可。

#Amateur Radio AX.25 Level2------

#Bridging（EXPERIMENTAL）------

选这两个可以让 Linux 变成一个网络上的网桥，用来做不同网络间的沟通，通常一般人不需要。

#Qos and/or fair queueing（服务质量公平调度）也支持了，还有 kHTTPd，不过这些都还在实验阶段。

8. Telephony Support 电话支持

原来是 Linux 下可以支持电话卡，这样你就可以在 IP 上使用普通的电话提供语音服务了。记住，电话卡可和 modem 没有任何关系。

9. SCSI support SCSI 设备支持

如果有 SCSI 设备，就回答 Y 。现在一般 PC 机不会有 SCSI。接着会有提示要求更进一步的资讯，像是你是否要支援光驱，硬盘，还有你使用的是那一种 SCSI 界面卡。这部份请参阅 SCSI-HOWTO ，有更详细的说明。如果你的启动分区是 SCSI 设备，不要选择 SCSI 模

块支持。在一般的 SCSI 后是 SCSI 低级设备驱动程序。再次重申，模块仅仅是用在不在启动分区的设备。

#SCSI disk support------指硬盘而言，如果有 SCSI 硬盘，那么就要选这个选项。

#SCSI tape support------指磁带机而言，如果您有 SCSI 的磁带机，那么就要选这个选项。

#SCSI CDROM support------指 CDROM，如果您有 SCSI 光驱，这一项一定要选。

#SCSI generic support------指其他有关 SCSI 的东西，也许您有一台 SCSI 的扫描器或是烧录机，或是其他有关 SCSI 的配备，您就要选这一项。而且，除此之外，您还必须准备关于这些配备的软件。

##Some SCSI devices（e.g. CD jukebox）support multiple LUNs

#Probe a11 LUNs on each SCSI device------通常这个选项大部分的人都不会选。我们举个例子来说，如果您的 SCSI 光驱是那种多片装的，就是一台光驱，但可以一次放好几

片光盘片的那种。这种我们叫做 Lun。
#Verbose SCSI error reporting（kernel size+=12K）------如果认为您的 SCSI 硬件配备有些问题，想了解一下它出现的错误信息。那么您可以把这个选项选 y，Linux 核心会告诉您有关于您的 SCSI 配备的问题（如果有的话）。不过，它会增加核心约 12KB 左右。
##SCSI low—level drivers
下面总共有接近 30 张的 SCSI 卡，您可以依需求做选择 SCSI 卡牌子。
c AIA1542 support
AdaPtec AIHA1740 support
AdaDtec AHA274X／284X／294X support
AdaPte 7000FASST SCSI support
AdaPtec AHA152X／2825 support
Advansys SCSI support
Always IN2000 SCSI support
Advansys SCSI support
Always IN2000 scsi support
AM53／79C974 PCI SCSI sppport
Buslogic SCSI Support
DTC3180／3280 SCSI support
EATA ISA／EISA（DPT PM2011／021／012／022／122／322） support
EATA－DMA（DPT，NEC，AT&T，SNI，AST，01ivetti，Alphatronix）support
EATA－PIO（old DPT PM2001，PM2012A） support
Future Domain 16xx SCSI support
Generic NCR5380／53c400 SCSI support
NCR53c405a SCSI support
NCR53c7，8xx SCSI support
NCR53CSXX SCSI support
IOMEGA Parallel Port ZIP drive SCSI support
PAS16 SCSI SUpport
Qlogic FAS SCSI support
Qlogic lSP SCSI support
Seagate ST-02 and future Domain TMC-8xx SCSI support
Trantor T128／T128F／T228 scsi support
Ultrastor 14F／34F support
Ultrastor SCSI support
10. I2O device support
这个也不清楚，帮助里说是这个需要 I2O 接口适配器才能支持的，在智能 Input/Output（I2O）体系接口中使用，又是要硬件，不选了。
11. Network device support 网络设备支持
上面选好协议了，现在该选设备了，内容多得很。还好里面大概分类了，有 ARCnet 设备、Ethernet（10 or 100 Mbit）、Ethernet（1000Mbit）、Wireless LAN（non-hamradio）、Token Ring device、Wan interfaces、PCMCIA network device support 几大类。耐心点，一般说来都能找到自己用的网卡。如果没有，你只好自己到厂商那里去要驱动了。 如果这个选项没有打开的话，那么以下的选项将不会出现。它是在选择网络卡或是网络的设备。例如，PLIP，PPP，SLIP，还有各式各样的网络卡，所以这个选项通常是

选 y。
#Dummy net driver support------如果有 SLIP 或 PPP 的传输协议，那么要把这一项打开。因为一来它不会让您的 Linux 核心增大。二来，对某些应用程序来说，它可以让我们模拟出来的 TCP／IP 环境更像 TCP／IP 环境。如果您没有 SLIP 或 PPP 协议，就不用打开了。
#EQL（serial line load balancing） support------如果有两个 MODEM，两条电话线，而且用 SLIP 或 PPP 协议，可以用这个 Driver 以便让您的 MODEM 有两倍的速度。当然，在网
络的另一端也要有同样的设备。
#PLIP（parallel port） support------依字面上看，它是一种利用打印机的接口（平行接口），然后利用点对点来模拟 TCP／IP 的环境。它和 SLIP／PPP 全都属于点对点通讯，您可以把两台电脑利用打印机的连接接口串联起来，然后，加入此通讯协议。如此一来，这两部电脑就等于一个小小的网络了。不过，如果电脑有提供打印服务的话，这个选项最好不要打开，不然可能会有问题（因为都是用平行接口）。
#PPP（point－to－point）support------点对点协议，近年来，PPP 协议已经慢慢的取代 SLIP 的规定了，原因是 PPP 协议可以获取相同的 IP 地址，而 SLIP 则一直在改变 IP 地址，
在许多的方面，PPP 都胜过 SLIP 协议。
#SLIP（serial line） support------这是 MODEM 族常用的一种通讯协议，必须通过一台 Server（叫 ISP）获取一个 IP 地址，然后利用这个 IP 地址，可以模拟以太网络，使用有关 TCP／IP 的程序。
##Ethernet（10 or 100Mbit）
如果您在学校接了校园网络并且使用网络卡，那么这个选项一定要选 y，否则以下对网络卡的选择将不会出现。或是您有网络卡，这时您同样的也要选 y。之后，下面会列出许多网络卡让您选择。像我们平常用的都是 NE2000 相容卡。
#3COM cards
#AMD LANCE and PCnet（AT1500 and NE2100） support
#Western Digital／SMC cards
##other ISA Cards（CONFIG．ISA）------选 y，以下才会列出有关 ISA 的网络卡。包括 NE2000 的兼容卡。
Cabletron E21xx support
DEPCA， DE10x， DE200， DE201， DE202， DE422 support
EtherWORKS 3（DE203， DE204， DE205） support
EtherExpress 16 support
HP PCLAN＋（27247B and 27252A） support
HP PCLAN（27245 and other 27xxx series） support
HP 10／100VG PCLAN（ISA， EISA， PCI） support
NE2000／NE1000 support
SK．G16 support
EISA， VLB， PCI and on board controllers ------选择网络卡，包括直接附在主机板上的那种。如果选择 y，则底下会列出其他的网络卡让您做选择，这些卡对于一般人来说很少会去用到。所以大部分的人这项是选 N 的。
#Pocket and portable adaptors------通常用在可携式的电脑上，这类型的网络卡（口袋型的），由于体积很小在安装和取下方面很方便，因此笔记本相关电脑上便常常采用这种网络卡。

#Token Ring driver support------Token Ring 是 IBM 电脑上的网络。它叫令牌环网络，和以太网络是很类似的东西。如果您希望使用的 Token Ring 网络卡以便连接到这种网络，那么选 Y，一般人都选 N。
#ARCnet support------这也是一种网络卡，通常一般人用不到，所以选 n。如果您有这样的网络卡，请看 Documentation／networking／arcnet．txt 的说明。
补：block 选项中的有关 loop 的应选上
否则不能 mount iso 文件
应该将其修改一下

12. Amateur Radio support 业余无线广播
可以用来启动无线网络的基本支持，目前的无线网络可以通过公众频率传输数据，如果你有此类设备就可以启用，具体请参考 AX25 和 HAM HOWTO 文档。

13. IrDA（infrared）support 红外支持

14. ISDN subsystem
如果使用 ISDN 上网，这个就必不可少了。ISDN（Integrated Services Digital Networ k），它的中文名称是综合数字服务网络，是一个利用电话线，把声音，影片信息以数字的方式传送的数字网络，它需要电话交换机设备有支持 ISDN，这通常需要电信局来做安装，对于在家工作的人来说，ISDN 可能是最舒适最便宜的一种方式，因此有愈来愈多的人使用它。不过，除非是公司，不然一般人很少会使用到 ISDN 的，所以这部分的选项大都选 N。如果您选择 Y，则下面会出现一些有关 ISDN 的问题。如果需要用到 ISDN，可以去
看看杂志的介绍。只要是有关网络的杂志应该都会有介绍。还需要启用 Support synchr onous PPP 选项（参考 PPP over ISDN)。

15. Old CD-ROM drivers（not SCSI、 IDE）非 SCSI/IDE 口的光驱
如用 IDE 的 CD-ROM，不选。
以下是选择非 IDE／ATAPI 和 SCSI 界面的光驱，这些光驱通常有自己专属的接口卡也是比较旧型的光驱类型。如果有这些光驱，则这个选项要选 y，否则选 n。如果您选择 n，则会跳过以下光驱的选项。
Aztech／ orchid／okano／Wearnes／IXC／CyDROM CDROM support
Goldstar R420 CDROM support
Matsllshita／panasonic／Creative，longshine，TEAC CDRW Support
Mitsllmi（no XA／MUltisessIon） CDROM Support
Mitsumi（XA／Multisession） CDROM support
optics Storage DCLPHIN 8000AT CDROM support
Philips／LMS CM206 CDROM support
Sanyo CDR－h94A CDROM sunnort
Soft configurable CDROM interface card support
Sony CDU3lA CDROM sunnort

Sony CDU535 CDROM support

16. Character devices 字符设备
所谓字符设备通常是指以字符为单位做处理的设备，例如终端机就是其中一项。原则上，我们对于这些选项的选择也是以预设为主。这个内容又太多了，先使用缺省设置，需要的话自己就修改。把大类介绍一下吧：

15. Old CD-ROM drivers（not SCSI、 IDE）非 SCSI/IDE 口的光驱
如用 IDE 的 CD-ROM，不选。
以下是选择非 IDE／ATAPI 和 SCSI 界面的光驱，这些光驱通常有自己专属的接口卡也是比较旧型的光驱类型。如果有这些光驱，则这个选项要选 y，否则选 n。如果您选择 n，则会跳过以下光驱的选项。
Aztech／ orchid／okano／Wearnes／IXC／CyDROM CDROM support
Goldstar R420 CDROM support
Matsllshita／panasonic／Creative，longshine，TEAC CDRW Support
Mitsllmi（no XA／MUltisessIon） CDROM Support
Mitsumi（XA／Multisession） CDROM support
optics Storage DCLPHIN 8000AT CDROM support
Philips／LMS CM206 CDROM support
Sanyo CDR－h94A CDROM sunnort
Soft configurable CDROM interface card support
Sony CDU3lA CDROM sunnort
Sony CDU535 CDROM support

16. Character devices 字符设备
所谓字符设备通常是指以字符为单位做处理的设备，例如终端机就是其中一项。原则上，我们对于这些选项的选择也是以预设为主。这个内容又太多了，先使用缺省设置，需要的话自己就修改。把大类介绍一下吧：
#Digiboard PC／Xx Support------这是一张叫 Digiboard PC／XX 卡的 driver，这种卡上面有很多个 serial port 的插槽（一般来说只有两个），可以用来连接很多个 MODEM，在民间的 BBS 站很常用到，如果您有这样的东西，您必须选这项为 y。有兴趣的读者可以读读 Documentation／digiboard．txt 的内容。
#Cyclades async mux support------同上，这也是一种能接很多个 serial port 插槽的卡的驱动程序。
#Stallion multlport serial support------同上，这也是其中一种卡。
#SDL RISCom／8 card support------这也是其中的一个支持 muti－serial 卡的 driver。
#Parallel printer support------有打印机的或是使用到并行接口的人这一项一定要选。除非是用 serial 的打印机。还有如果您有使用 PLIP，那么这项也请选上。
#Mouse support------大部分的人这一项并不用选 y。因为大部分的人是用 serial 的鼠标，除非有些人是用一种附有接口卡的鼠标，这时这个选项才要选上，如果您选 y，则底下会列出您的 BUS 鼠标所用的接口卡。
#support for user misc device modules------除非您有所谓的触摸式显示器或是光笔等东西，否则这一项选 n。
#QIC－02 tape support------非 SCSI 界面的磁带机，除非您有，否则选 n。

#Ftape（QIC－80／Travan）support------如果有磁带机，而这个磁带机是接在软盘控制卡上，这个选项才要选 y。
#Watchdog Timer support------一般人不需要这个选项，如果您选上这个选项，则您要用 mknod 在／dev 下建一个 watchdog 的文件。请看 Documentation／watchlog．txt 的解释。
#Enhanced Real Time Clock Support------关于系统上 Clock 的东西，您必须自己用 mknod 在／dev／下建立一个文件叫 rtc。如此一来，在／proc／下将可以看到 rtc 的信息。有关于 rtc 的内容请看 Documentation／rtc．txt。一般是选 n。

17. USB supprot USB 支持
很多 USB 设备，比如鼠标、调制解调器、打印机、扫描仪等，在 Linux 都可以得到支持，根据需要自行选择。

18. File systems 文件系统
Linux 上有支持约二十几种的文件系统，有支持某个文件系统的意思是，可以存取某个文件系统的数据或是做拷贝动作。在这些文件系统中，通常的选择方法是按照原来预设的方式，不过，在其中，EXT2FS 那个选项无论如何一定要选，因为那是 Linux 系统所使用的文件系统。其他的则依需求做选择。
通常是 ext2 而让其余的使用模块。
#Kernel automounter support------选择“y”，内核将提供对 automounter 的支持，使系统在启动时自动 mount 远程文件系统。
#Standard（ minix ）------新的套件不再建立 minix 文件系统，而且很多人不使用它，但是把它配置在核心里仍然是个好主意。某些〞rescue-disk 〞 程序会用到它，而且仍然有许多磁片可能用 minix 文件系统，因为 minix 文件系统对于处理磁片方面是最好的。当初 Linus 是因为对 Minix 这个小型的操作系统有很深的经验，所以才写出 Linux 这个操作系统。Minix 文件系统通常用在磁盘上，有时会用到它。
#Extended fs ------ 这是扩充文件系统的第一版，现在已经不再使用。
#Second extented fs------这是现在新发行的套件所广泛采用的文件系统，你可能会有其中一种。这个是 linux 文件系统，请务必选 y，如果问我说选 n 会有什么后果，我也不知道。除非您能把 Linux 装在 DOS 的目录下。
#xiafs filesystem-----这个文件系统曾经一度很普遍，但是在写这份文件时，我已经不知道有任何人在使用它了。
#DOS FAT fs ------DOS FAT 文件格式的支持，可以支持 FAT16、FAT32。这个选项是 DOS 的文件系统，如果您没有选 y，则下面的 MSDOS，VFAT，umsdos 将不会出现。
#msdos------DOS 文件系统的格式。如果你想要在 linux 下使用你硬盘中的 MS-DOS 分割区，或是想将用 MS-DOS 格式化的磁盘挂进来的话，回答 y 。
#VFAT（Windows95） fs ------windows95 所支持的文件系统，是我们常说的 vfat 文件系统，如果您的系统中装有 windows95，那么选择这个文件系统将对以让您看到 windows95 的长文件名。
#umsdos： UNIX like fs on top of std MSDOS FAT fs ------ 如果把 Linux 装进 DOS 的一个目录下那么您则要选择这一项。不建议这样做，因为如此一来，就不能看到长文件名了。而且效率上，并不是很好。
#ums-dos------相当 slick 的文件系统，它能使 MS-DOS 文件系统拥有更多的特性，像是长档名等等。这对那些不使用 MS-DOS 的人（像我）并不是很有用。

#/proc------这是最 slick 的文件系统之一。它不是你硬盘分割区里的任何东西，不占用硬盘的空间，而是核心与程序之间的文件系统介面，它表示的只是内存里头的状况和各个程序执行的情形，它也记录了您硬件上配备。。许多程序工具（像〞ps 〞）都会用到它。如果已经将它安装好了，有空不妨试试看〞cat /proc/meminfo 〞或者是〞 cat /proc/devices 〞。有些 shells ，像是 rc ，会用 proc/self/fd（在其它系统上为 /dev/fd ）来处理输出入。几乎可以确定你在这里得要回答 y ，有许多重要的 Linux 标准工具是靠它来运作的，否则有些指令会出问题。

#Root file system on NFS------一般不选，除非您的电脑上没有硬盘，希望通过网络由别人的硬盘开机过 Linux 如此一来才有需要选这项。同时对方也要执行 rarp 的服务。

#System V and Coherent------这是为 System V 以及 Coherent 的分割区而设的。如果希望支持 System V 或 Xenix 的相关 UNIX 系统的 FS 并读取它们的数据，那么才有必要选这
个，否则一般来说这个选项是选 n 的。

#Quota support ------Quota 可以限制每个用户可以使用的硬盘空间的上限，在多用户共同使用一台主机的情况中十分有效。

#ISO 9660 CD-ROM file system support------光盘使用的就是 ISO 9660 的文件格式。

#Mandatory lock support------有些很特殊的 database 应用软件会用到它，一般人这个选项是选 n。而且，如果选 y 的话，必须有最新版的 NFS 软件，最新版的 samba 软件等。

#NTFS file system support------ntfs 是 NT 使用的文件格式。

#UFS filesystem support-----这是 BSD，SunoS，FreeBSD，NetBSD 或 Nextstep 所使用的文件系统。如果您在电脑上有这些操作系统的话，那么可以选这一项。否则一般人都选 n。

##Network File Systems------网络文件系统

NFS------如果你在网络环境下而且想要分享档案，回答 y 。如果希望挂上别的电脑的文件系统，那么这个选项一定要选进去。它可以让您利用网络把别人的硬盘当成自己的来使用（把它变成一个目录）。对于一般人来说，这个选项是选 y。

#SMB filesystem support------这个文件系统让您可以挂上 windows95 或 windowsNT 的文件系统，也就是您也可以抓到在 windows 下，网上邻居上的电脑。

#SMB long filename support------支持 windows95 的长文件名。

#NCP filesystem support------NCP 是一种网络的通讯协议，用在跑 IPX 协议上，它可以利用 IPX 协议让两台电脑之间的文件共享，并做沟通。如果您想挂上有关 Novell 的 Netware 文件系统，那么这个选项就选上去吧。

##Partition Types-----分区类型，该选项支持一些不太常用的分区类型，用户如果需要，在相应的选项上选择“y”即可。

##Native Language Support------本地语言支持

附：不知道需要那些文件系统怎末办?

键入〞 mount 〞它看起来会像这样：

sunlyy% mount

/dev/hda1 on / type ext2 (defaults)

/dev/hda3 on /usr type ext2 (defaults)

none on /proc type proc (defaults)

/dev/fd0 on /mnt type msdos (defaults)

仔细看看每一行；在〞 type 〞后面的那个字就是文件系统的格式。在这个例子中，我的 / 和 /usr 分割区是 second extended 格式，我使用 /proc ，而且挂有一张以 msd

os (bleah) 为文件系统格式的磁片。如果你有使用 /proc ，可以试试″cat /proc/fi
lesystems ″。它会给你一份目前使用的核心所支援的文件系统列表。

19. Console drivers 控制台驱动
#VGA text console ------ 选择“y”，用户就可以在标准的 VGA 显示方式下使用 Linux 了。一般使用 VGA text console 就可以了，标准的 80*25 的文本控制台。
#Video mode selection support

20. Sound sound 声卡驱动
如果你能在列表中找到声卡驱动那自然最好，否则就试试 OSS 了。阅读帮助文件从列表中小心的选取。确信为你声卡真确的选择了 I/O 和 IRQ。声卡的 MPU I/O 是 0 选项。一般是 33
0，如果不对不必担心。模块的好处就是在核心编译以后你还能重新编译、安装模块并挂上核心。
如果有声卡，请去了解一下声卡的 IRQ 和 DMA 等信息，并了解是属于哪一种的。现在大部分的人使用的都是 Sound Blaster 或是它的相容卡。有关这类的信息请看各个声卡的说明书。当选 Y 时，出现下面的画面。就依您的声卡来做选择吧！
Pro Audio Spectrum 16 support
Sound Blaster（SB，SBPro，SB16，clone） support
Generic OPL2／0P13 FM synthesizer support
Gravis Ultrasound support
MPU-401 support（NOT for SB16）
6850 CART Midi support
PSS（ECho－AD12111）support（NOT for SB16）
16 bit samplins option of GUS（N0T__GUS__MAX）
GUS MAX support
Microsoft Sound System support
EnsolliQ Soundscape support
MediaTrix AudioTrix Pro support
Support forMAD16 and／or Mozart based cards

20. Sound sound 声卡驱动
如果你能在列表中找到声卡驱动那自然最好，否则就试试 OSS 了。阅读帮助文件从列表中小心的选取。确信为你声卡真确的选择了 I/O 和 IRQ。声卡的 MPU I/O 是 0 选项。一般是 33
0，如果不对不必担心。模块的好处就是在核心编译以后你还能重新编译、安装模块并挂上核心。
如果有声卡，请去了解一下声卡的 IRQ 和 DMA 等信息，并了解是属于哪一种的。现在大部分的人使用的都是 Sound Blaster 或是它的相容卡。有关这类的信息请看各个声卡的说明书。当选 Y 时，出现下面的画面。就依您的声卡来做选择吧！
Pro Audio Spectrum 16 support
Sound Blaster（SB，SBPro，SB16，clone） support
Generic OPL2／0P13 FM synthesizer support
Gravis Ultrasound support

MPU-401 support（NOT for SB16）
6850 CART Midi support
PSS（ECho－AD12111）support（NOT for SB16）
16 bit samplins option of GUS（N0T__GUS__MAX）
GUS MAX support
Microsoft Sound System support
EnsolliQ Soundscape support
MediaTrix AudioTrix Pro support
Support forMAD16 and／or Mozart based cards
Support for Crystal CS4232 based（PnP） cards
Support for Turtle Bench Wave Front （Maul，Tropez）synthesizers
／dev／dsp and ／dev／audio support------ 这个选项通常是必要的。因此大部分的人选 y，如果没有这选项，则很多的游戏将没有声音效果。
#MIDI interface support------支持 MIDI 界面。
#FM synthesizer（YM3812／0PL－3） support
#I／O base for SB Check from manual of the card------声卡的 I／O 地址。括号是常用的选项。
#Sound Blaster IRQ Check from manual of the card-----声卡的 IRQ，通常是 1 或 5。
#Sound Blaster DMA 0，1 for 3-----声卡的 DMA，通常是 1。
#Sound Blaster 16 bit DMA 5， 6 or 7（usel for 8 bitcards）（SB．DMA2）-----5
#MPU401 I／0 base of SB16， Jazz16 and ES1688 Check from manual of the card------0
#SB MPU401 IRQ（Jazzl6， SM Wave and ES1688） Usel with SB16------1
#Audio DMA buffer size 4096，16384，32768 or 65536------65536
##Additional low level drivers------如有其他种类的声卡，则这项要选上去，以下会列出其他的声卡供选择。

21. Kernel hacking 安全模式
通俗的说，这是 windows 安全模式,找不到明确解释，就引用这个说法。＞这是从 Linus 的 README 里摘录的：

第四节 启用内核

通常，核心安装叫做 vmlinuz。过去 Unix 使用者共同起了这名字。"z"表示压缩，"v"和"m"意思是"virtual"(虚拟）和"sticky（粘性的）"，各自属于内存和磁盘管理。建议保留 vmlinuz 核心，直到知道它工作。
为了能够使用新版本的内核，还需要做一些改动：
#cp /usr/src/linux/System.map /boot/System.map-2.2.16
#cp /usr/src/linux/arch/i386/bzImage /boot/vmlinuz-2.2.16

以上这两个文件是刚才编译时新生成的。下面修改/boot 下的两个链接 System.map 和 vml
inuz，使其指向新内核的文件：
#cd /boot
#rm -f System.map vmlinuz
#ln -s vmlinuz-2.2.16 vmlinuz
#ln -s System.map-2.2..16 System.map
注意：要保留 vmlinuz 核心，以下列步骤进行
#cp /usr/src/linux/System.map /boot/System.map-2.2.16
#cp /usr/src/linux/arch/i386/bzImage /boot/vmlinuz-2.2.16
#cd /boot
#rm -f System.map
#ln -s System.map-2.2..16 System.map
现在#vi /etc/lilo.conf,增加如下一段：
image=/boot/vmlinuz-2.2.16 是设定为已经安装的核心
label=linux2.2.16 则是由 lilo 用来告诉你现在要启动的是那个核心或作业系统
,
read-only
root=/dev/hda2 则是这个特别的作业系统的根目录 /
其中 root=/dev/hda2 一行要根据需要自行加以修改。
运行：#/sbin/lilo -v 保存执行命令：lilo 你将看到核心标签，第一个是星号。如果你
没有看到新核心的标签或 LILO 出现错误，你需要重新对/etc/lilo.conf 工作（看下面的
LILO 分析）。
确认对/etc/lilo.conf 的编辑无误，现在重新启动系统：
#shutdown -r now
不建议使用热启动或 ctrl+Alt+del 键。在一些情况下，文件系统不完全卸栽会损坏打开
的文件。在 LILO 提示时，如果你需要启动旧的核心或使用一些参数启动，如果你没看见
启动提示，你可以试用 shift 或 ctrl 键，这样启动提示就出现了。一旦出现，按 tab 看核
心标签。输入标签和可选参数启动。通常，在/etc/lilo.conf 文件指定的时间后自动启
动核心。启动时，你可能看见一些出错信息就象 SIOCADDR。这常常显示模块（一般是网
络模块）没有引导。处理这事很简单，如果有此一错，"VFS，cannotmount root",你就
不要在核心中编译适当的磁盘或文件系统支持。
在机器重启后出现 LILO 时按 TAB 键，输入 linux2.1.16，新内核发挥作用了。

附录：LILO 分析（技术性强，仅供参考）

第一部分 LILO 介绍

LILO（Linux Loader）是 Linux 自带的一个优秀的引导管理器，使用它可以很方便地引导
一台机器上的多个操作系统。与其他常用的引导加载程序相比，LILO 引导方式显得更具
有艺术性，对其深入的理解，将有助于我们方便地处理多操作系统、网络引导、大硬盘
及大内存等诸多棘手的问题。
LILO 的引导机制------众所周知，计算机的最初启动是由 BIOS 控制的，在对一些硬件（
如：内存、键盘等）初始化之后，它会试图加载硬盘的主引导记录（MBR）或软盘的引导

扇区。MBR 可通过两种方式运行，其一是定位到活动分区并加载相应的引导扇区，然后由引导扇区完成该分区内操作系统的基本组件的加载；其二是直接从一指定分区中加载信息，并通过它装入任一分区的操作系统，诸如 LILO、OS/2 boot loader 及 Partition Magic 等引导加载程序都可以配置成这种方式。软盘的引导扇区相当于硬盘活动分区的引导扇区，它通常用于装入软盘上的操作系统。由此可见，只要把 LILO 安装在 MBR、活动分区
或者引导软盘上，就能接管计算机的控制权，然后由 LILO 完成后继的引导过程。LILO 中建有一个引导表地址编码，借此它的引导程序就能定位到 Linux 的内核文件，这种地址编码既可以按照柱面/磁头/扇区(CHS)模式，又可以采用 LBA 的线性块号模式，因此，即使对某些 SCSI 控制程序 LILO 也能运转良好。
当 LILO 定位到配置文件后，经过预引导过程，就显示提示符：　LILO boot:
此时，系统允许选择引导不同的操作系统或者不同的内核配置，按 Tab 键显示可选项列表，然后输入可选项或者直接回车选择缺省配置，如果选择了引导 Linux，还可以直接传递参数到系统内核。
和其他系统的引导加载程序相比，LILO 具有更大的灵活性，其引导方式也更丰富多彩。
●当 LILO 被安装在硬盘的 MBR、活动分区或引导软盘上时，作为原引导程序的替身，它能
引导任一硬盘任一分区上的 Linux 和其他操作系统；除了引导扇区，它没有任何隐含文件，也不需要使用特定的分区，它的配置文件可以在任何分区、甚至是存放在与 Linux 毫不相干的 DOS 分区的某个子目录下；它能引导几个不同的内核配置，甚至是几个不同的内核；　它能引导同一机程序上的多个 Linux 版本；可达 16 个。
●它能从网络上引导 Linux。
●LILO 的灵活性使得其配置变得相当复杂，当有多个系统共存时，建议先安装其他操作系统，最后再装 Linux，这样，设置 LILO 对其他系统的引导会相对简单一些。

第二部分 LILO 参数

通常我们谈到 LILO，会涉及到两个方面——LILO 引导程序和 LILO 安装命令/sbin/lilo。为了不至于混淆这两个概念，本文将用 LILO 表示 LILO 引导程序，而 lilo 表示/sbin/lilo。一般地，LILO 使用一个文本文件/etc/lilo.conf 作为其配置文件。lilo 读取 lilo.conf，按照其中的参数将特定的 LILO 写入系统引导区。任何时候，修改了/etc/lilo.conf，都必须重新运行 lilo 命令，以保证 LILO 正常运 lilo.conf 使用的配置参数很多，配置起来也相当复杂。下面以 RedHat Linux 为例作一些初步探讨，RedHat 的 lilo 程序包版本为 0.20，别的 Linux 发行版本可能会有所出入，但不会太大。
lilo.conf 文件中的配置参数分为两部分，一部分是全局参数，另一部分是引导映像参数。引导映像参数作用于每一个引导映像区。如果某一引导映像参数（例如：password 与全局参数的定义相抵触，则以该引导映像参数的定义为准，但仅限于该引导映像区。LILO 的引导参数有很多，在此只对一些比较重要的参数作一介绍。与 Linux 系统其他的配置文件一样，“#”号后的一行文字表示注释。
1.“boot=”此参数指明包含引导扇区的设备名（如：/dev/had），若此项忽略，则从当前的根分区中读取引导扇区。
2.“root=”此参数告诉内核启动时以哪个设备作为根文件系统使用，其设定值为构造内核时根文件系统的设备名，可用的设备名有：
(1)/dev/hdaN~/dev/hddN：ST-506 兼容硬盘，a 到 d 上的 N 个分区

(2)/dev/sdaN~/dev/sdeN：SCSI 兼容硬盘，a 到 e 上的 N 个分区
(3)/dev/xdaN~/dev/xdbN：XT 兼容硬盘，a 到 b 上的 N 个分区
(4)/dev/fdN：软盘，A：（N=0）或 B：（N=1）
(5)/dev/nfs：由网络取得根文件系统的标志
3.“nfsroot=”若需通过 NFS 提供根文件系统来引导无盘工作站，此参数为内核指定了网络根文件系统所在的机程序、目录及 NFS，其格式为：nfsroot=〈server_ip〉:〈root_dir〉〈,nfs_options〉)
4.“nfsaddrs=”设定网络通讯所需的各种网络界面地址，如无此参数，则内核会试图用反向地址解析协定(RARP)或启动协定(BOOTP)找出这些参数，其格式为：nfsaddrs=〈客户端 IP〉:〈服务端 IP〉:〈网关 IP〉:〈子网屏蔽〉:〈客户端名称〉:〈网络设备名 〉:〈auto〉
5.“image=”指定 Linux 的内核文件。
6.“delay=”设定引导第一个映像前的等待时间。
7.“disk=”此参数为某一特殊的硬盘定义非标准参数。
8.“append=”为内核传递一个可选的参数行，其典型的应用是为不能完全由系统自动识别的硬盘指定参数，如：append = "hd=64,32,202"
9.“label=”此参数为每个映像指定一个名字，以供引导时选择。
10.“read-only”设定以只读方式挂入根文件系统，用于文件系统一致性检查（fsck）。
11.“install=”安装一个指定文件作为新的引导扇区，缺省为/boot/boot.b。
12.“loader=”说明所使用的链加载程序(chain loader)，缺省为/boot/chain.b，如果不是从首硬盘或软盘启动，那么，此选项必须说明。
13.“table=”说明包含分区表的设备名，如果此参数忽略，引导加载程序将不能传递分区信息到已引导的操作系统。当此参数指向的分区表被修改时，必须重新运行/sbin/lilo。
14.“init=”内核初始化时执行的程序，通常过程为 init、getty、rc 和 sh，版本 1.3.43 以来的 Linux 内核能够执行/sbin/init 说明的命令行，若在引导过程中出现问题，则可设置 init=/bin/sh 直接跳到 Shell。
15.“ramdisk_start=”由于内核不能放在压缩的内存文件系统映像内，为使内核映像能够和压缩的内存映像放在一张软盘内，加入“ramdisk_start=〈offset〉”，这样内核才能开始执行。
16.“mem=”此参数的目的之一是为 Linux 指定使用的内存数量：如 mem=96MB，目的之二
是指定 mem=nopentium 告诉内核不要使用 4 MB 分页表。
17.“vga=”设置显示模式，如 80×50、132×44 等。
18.“linear” 产生用于替换硬盘 sector/head/cylinder 地址（硬盘几何参数）的 linear 扇区地址。linear 地址在运行时产生并且不依赖于硬盘几何参数。某些 SCSI 硬盘和一些以 LBA 方式使用的 IDE 硬盘可能会需要使用这个参数。注意：在将 LILO 安装到软盘上时
不能使用“linear”参数。
19.“prompt” 给出“boot:”提示，强制 LILO 等待用户的键盘输入，按下回车键则立即引导默认的操作系统，而按下 Tab 键则打印可供选择的操作系统。当“prompt”被设置而“timeout”没有被设置时，系统会一直处于等待状态而不引导任何操作系统。不设置该参数时，LILO 不给出“boot:”提示而直接引导默认操作系统，除非用户按下了 Shift、

Ctrl、Alt 三键中的任何一个。大多数情况下，如果你的硬盘上有多个操作系统，建议使用参数，它留给用户一个选择的余地。
20.“timeout=” 设置等待键盘输入的时长，单位是 0.1 秒。超过这段时间没有输入则为超时，系统将自动引导缺省的操作系统。如果不设置本参数，缺省的超时时间长度为无穷大。
21.“other=” 设置包含非 Linux 操作系统，如 DOS、SCO UNIX、Windows 95 等系统引导映像的文件或设备。
22．alias=name 给当前操作系统起一别名。

第三部分 LILO 典型配置方法

通常情况下，Linux 的安装程序自身就可以完成 LILO 的安装配置，从而较好地解决多重系统的引导问题，如果系统不能自动完成这种配置，则可以通过手工修改配置文件/etc/lilo.conf 来实现不同条件下的引导。
1.当系统能自动完成配置时
对于这种情况只有一个建议：将 LILO 安装到 Linux 分区的根上，而不是 MBR 这个多事地带
。假设当前 hda1 中装有 DOS/Windows，hda2 中安装了 Linux，则/etc/lilo.conf 的内容大致如下：
boot=/dev/hda2＃指定引导位置
compact
delay=50＃延时 5 秒
root=current＃根在当前分区
image=/boot/vmlinuz＃指定 linux 的内核文件
label=linux＃用 linux 为代表名称
other=/dev/hda1＃其他操作系统所在的分区
table=/dev/hda＃指定包含分区表的硬盘
label=dos＃用 dos 为代表名称
2. 当系统无法自动完成配置时
系统无法自动完成配置的情况不外乎两种：
（1） BIOS 不能直接看到 Linux 的根分区；
（2） BIOS 只能读写标准 IDE 硬盘的前 504MB。
这时，必须遵循一个最基本的原则：建立一个 BIOS 能存取的较小的 Linux 分区，其中包含内核文件、映射文件及链加载程序等必要内容，而根则可以是另外一个独立的分区。至
通常情况下，Linux 的安装程序自身就可以完成 LILO 的安装配置，从而较好地解决多重系统的引导问题，如果系统不能自动完成这种配置，则可以通过手工修改配置文件/etc/lilo.conf 来实现不同条件下的引导。
1.当系统能自动完成配置时
对于这种情况只有一个建议：将 LILO 安装到 Linux 分区的根上，而不是 MBR 这个多事地带
。假设当前 hda1 中装有 DOS/Windows，hda2 中安装了 Linux，则/etc/lilo.conf 的内容大致如下：
boot=/dev/hda2＃指定引导位置
compact

delay=50＃延时 5 秒
root=current＃根在当前分区
image=/boot/vmlinuz＃指定 linux 的内核文件
label=linux＃用 linux 为代表名称
other=/dev/hda1＃其他操作系统所在的分区
table=/dev/hda＃指定包含分区表的硬盘
label=dos＃用 dos 为代表名称
2. 当系统无法自动完成配置时
系统无法自动完成配置的情况不外乎两种：
（1） BIOS 不能直接看到 Linux 的根分区；
（2） BIOS 只能读写标准 IDE 硬盘的前 504MB。
这时，必须遵循一个最基本的原则：建立一个 BIOS 能存取的较小的 Linux 分区，其中包含内核文件、映射文件及链加载程序等必要内容，而根则可以是另外一个独立的分区。至于配置上的其他细节，我们通过以下实例来进行说明。
第四部分 lilo.conf 配置实例
有了这些基础知识，我们可以很容易地按照自己的意图配置 LILO。
例一．lilo.conf 文件
boot=/dev/hda #将 LILO 安装在 MBR。LILO 作为主引导管理器
message=/boot/message #注释为/boot/message
compact #产生一个更小的“map”文件
map=/boot/map #指定“map”文件为/boot/map
install=/boot/boot.b
password=zhoudi #设置口令
vga=normal #80x25 文本模式
linear #使用“linear”地址
prompt #提示用户键盘输入
timeout=50 #超时时长为 5 秒
default=dos #缺省引导 label 为 dos 的操作系统
image=/boot/vmlinuz-2.0.34-1#设定 Linux 所用核心
#设置 Linux 核心引导映像
label=linux #标识为 linux
root=/dev/hda1 #设置根文件系统
read-only #LILO 以只读方式载入根文件系统
#设定 MS-DOS 或 Windows 95
other=/dev/hda2 #DOS 分区为第一个 IDE 硬盘的第二分区
label=dos #标识为 dos
table=/dev/hda #主设备为第一个 IDE 硬盘
#设定 SCO UNIX 注意：SCO 分区必须设为活动（active）分区并将 LILO 安装在 MBR 上。
other=/dev/hda3
label=sco
table=/dev/had

这个例子中，LILO 是作为主引导管理器来管理机器上所有操作系统的。LILO 也可作为二级引导管理器，这只要将“boot”参数改为根分区就可做到。例如： boot=/dev/h

da1 以这种方式使用 LILO 时，Linux 根分区必须用 DOS 或 Linux 的 fdisk 程序将其设置为活
动分区，并且这种方式只对硬盘主分区（不是扩展或逻辑分区）有效。
例二．一个标准的 IDE 大硬盘需安装 Linux 和 DOS/Windows。
对于大硬盘问题，很多人只知道低于 1024 个柱面的限制，而不知为什么标准的 IDE 硬盘只能认前 504MB。其实，BIOS 的 int13 调用是采用三个位元组的 CHS 编码，10 位为柱面号，8
位为磁头号，6 位为扇区号。可能的柱面号码是 0～1023，可能的磁头号码是 0～255，而磁道上可能的扇区号码是 1～63，以这 24 位最多可以定址 8455716864 个位元组(7.875GB)。但不幸的是，标准的 IDE 介面容许 256 个扇区／磁道、65536 个柱面及 16 个磁头。它自己
本身可以存取 2 3 7 = 137438953472(128 GB)，但是加上 BIOS 方面 63 个扇区与 1024 个柱面
的限制后只剩 528482304(504MB)可以定址得到。
对策：在硬盘的前 500MB 中划分 350MB(/dev/hda1)给 DOS，150MB(/dev/hda2)给 Linux，在相应的配置文件中应说明硬盘的参数。

```
boot=/dev/hda
... ...
disk=/dev/hda
bios=0x80
sectors=63
heads=16
cylinders=2100
image=/vmlunuz
append="hd=2100,16,23"
root=/dev/hda2
label=linux
```

例三．如果你有一块超过 8 G 的大硬盘，并且需要把 Linux 安装在比较靠后的位置,可以在安装的时候，选择 linear 模式，并且给它加上硬盘参数.
安装时候的硬盘参数可以这样写: hd?=CYLs, HEADs, SECs 其中的大写字母需要用实际的硬盘参数来替换，这些参数可以从硬盘的标签上查到，也可以看看 BIOS 设置里硬盘参数对应 LBA 模式的那一行．问号是根据硬盘确定的，实际使用时，它可以是 a, b, c, d 四个字母中的一个．比如: hda=1869,63,255 这是 IBM 15.2G 硬盘的参数.
当然进入了 Linux 以后，可以通过编辑 /etc/lilo.conf 加上这个文件，然后运行一遍 lilo 达到同样的目的．下面是本人未加参数前的 lilo.conf 的内容:

```
boot = /dev/hda
map = /boot/map
install = /boot/boot.b
prompt
timeout = 50
image = /boot/vmlinuz
label = linux
root = /dev/hda1
```

initrd = /boot/initrd-2.2.12-20.img
read-only
按照 linear 方式加入参数以后是如下格式:
boot = /dev/hda
map = /boot/map
install = /boot/boot.b
prompt
linear <-----加进了这一行
timeout = 50
image = /boot/vmlinuz
label = linux
root = /dev/hda1
initrd = /boot/initrd-2.2.12-20.img
read-only
append = "hda=1869,63,255" <------加进了这一行 注意, append 参数是针对每个系统引导记录的, 一定要放在 image 的下面或者是 other 的下面,这样它才可以发挥作用.
当再次起动 Linux 系统的时候, LILO 就按照线性模式对系统进行引导.除了在硬盘上寻址定位的方式不同以外, 对其他方面没有什么影响.
第三章内核编译的应用
第一节嵌入式 Linux 技术
第二节你的 Linux 有多大？（及实践结果）
这可不是我写的，只是用来参考制造 small kernel,效果还可以。
最小的 Linux kernel
我使用的是 Mandrake 内核的 2.2.15，我没有修改任何一行程序码，完全只靠修改组态档得到这些数据。
首先，使用 make xconfig 把所有可以拿掉的选项都拿得。
不要 floppy
不要 SMP,MTRR
不要 networking,SCSI
把所有的 block device 移除，只留下 old IDE device
把所有的 character device 移除
把所有的 filesystem 移除，只留下 minix
不要 sound 支援
相信我，我已经把所有的选项都移除了。这样做之后，我得到了一个 188K 的核心。不过这个核心恐怕很难发挥 Linux 的功能，因此我决定把网络加回去。把 General 中的 network support 加回去，重新编译，核心变成 189 K。10K 换个 TCP/IP stack，似乎是很上算的生意。
不过有 stack 没有 driver 也是惘然，所以我把 embedded board 常用的 RTL8139 的 driver 加回去，195K。如果你需要 DOS 档案系统，那大小成为 213K。如果 minix 用 ext2 换代，则大小成长至 222K。
不过大家要注意，那里的大小指的是核心档的大小。那和所需要的随取记忆体是二回事。这个数字代表的意义是你需要多小的 ROM 来存放你的核心。
Linux 所需的记忆体大约在 600~800 K 之间。1MB 可能可以开机了，但可能不太有用。因为可能连载入 C 程序库都有困难。2MB 应该就可以做点事了，但可能要到 4MB 以上才

可以执行一个比较完整的系统。
看到这里，是不是觉得 Linux 真的有点大。好吧! 那我们就来看看谁占用了这些空间，下面这个列表是从 222K 这个核心做出来的。
wc
arch/i386/kernel/kernel.o
arch/i386/mm/mm.o
kernel/kernel.o
mm/mm.o fs/fs.o
ipc/ipc.o
fs/filesystems.a
net/network.a
drivers/block/block.a
drivers/char/char.a
drivers/misc/misc.a
drivers/net/net.a drivers/pnp/pnp.a
/usr/src/smalllinux/arch/i386/lib/lib.a
/usr/src/smalllinux/lib/lib.a
/usr/src/smalllinux/arch/i386/lib/lib.a
结果如下 ：
243 2250 81946 arch/i386/kernel/kernel.o
42 316 10569 arch/i386/mm/mm.o
173 1541 74660 kernel/kernel.o
266 2307 68053 mm/mm.o
222 3139 123193 fs/fs.o
49 602 21600 ipc/ipc.o
263 2940 106504 fs/filesystems.a
137 1510 65512 net/network.a
92 719 39178 drivers/block/block.a
230 2308 87556 drivers/char/char.a
1 1 8 drivers/misc/misc.a
83 721 25680 drivers/net/net.a
1 1 8 drivers/pnp/pnp.a
20 187 9526 /usr/src/smalllinux/arch/i386/lib/lib.a
23 150 7714 /usr/src/smalllinux/lib/lib.a
20 187 9526 /usr/src/smalllinux/arch/i386/lib/lib.a
1865 18879 731233 total
先说明一下，这里的大小和最终的大小有点差别，但大致还是可以做个参考。这边显示 730K 实际上大约在 600K 左右。很显然的，filesystem 相当的大。大约在 230K 左右，占了 1/3 的体积。记忆体管理占了 80K，和核心其它部份的总合差不多。TCP/IP stack 占了 65K，驱动程序占了 120K。SysV IPC 占了 21K，必要的话可以拿掉，核心档应该可以再小个 10K 左右。 所以如果要减核心大小，应该动那里呢? 答案应该很明显，当然是档案系统。Linux 的 VFS 减化了档案系统的设计，buffer cache, directory cache 增加了系统的效率。但这些对整个系统都在 flash 上的 embedded 系统而言根本就用处不大。如果可以把它们对拿掉，核心可以马上缩小 20K 左右。如果跳过整个 VFS，

直接将档案系统写成一个 driver 的型式，应该可以将 230K 缩减至 50K 左右。整个核心缩到 100K 左右。
从上面的数据来看，ucLinux 所减小的 mm 部份反到省的不多，主要是 mm 除了 virtual memory 之外，也要处理 memory allocation 的部份，这部份是省不得的。如果二者齐做，则 100K 以下的 Linux 核心不是不可能的事。
实践：成功编译 196k 的核心，但不能用来启动 redhat，它太庞大了。280k 可以启动，正在向更小努力。
结束语
论文终于写完了，有必要回顾一下我的毕业设计过程。
对于一个非计算机专业的学生来说，这个课题更具有挑战性。在 Linux 大行其道的今天，系统及网管方面的资料随处可见，但内核资料匮乏，让人不知所措。开始只要是 Linux 的书籍就看，一个月后觉得视野大开，逐渐明确了方向。搜集翻译资料成了重要内容，有时通宵在网上找资料，再在系统上不断检验。论文核心------编译流程开始就得到老师指导，配置内核 21 大项数百个知识点可以说是一条一条积累起来的，让我很有成就感。论文除了详细叙述怎么做，也说明了原因，甚至还有附录。我在突出重点的同时，又联系了我课题以外的内容------嵌入式 Linux 操作系统，因为我知道我的课题是综合课题-----设计具有嵌入式操作系统的器件（理论）------的前期工作。论文集网络文章之大