

目录

写在前面	iii
第一章 Slackware简介	1
1.1 什么是Linux?	1
1.1.1 GNU的世界	1
1.2 Slackware是什么	2
1.3 开源与自由软件	2
第二章 帮助	4
2.1 系统帮助	4
2.1.1 man	4
2.1.2 /usr/doc文件夹	5
2.1.3 HOWTO及mini-HOWTO	6
2.2 在线帮助	6
2.2.1 官网及论坛	6
2.2.2 E-mail支持	6
2.2.3 Slackware的邮件列表	7
2.2.4 非官方的网站及论坛	7
第三章 安装	9
3.1 获取Slackware	9
3.1.1 官方CD及套装	9
3.1.2 通过互联网	10
3.2 系统要求	10
3.2.1 软件包系列	10
3.2.2 安装方法	11
3.3 启动安装器	13
3.4 分区	15
3.5 setup程序	18
3.5.1 帮助	18
3.5.2 键映射	19

3.5.3	添加交换分区	20
3.5.4	选择安装位置	20
3.5.5	安装源	21
3.5.6	选择安装包	21
3.5.7	安装	22
3.5.8	配置	23
第四章	系统配置	29
4.1	系统一览	29
4.1.1	文件系统布局	29
4.1.2	查找文件	31
4.1.3	/etc/rc.d文件夹	32
4.2	选择内核	35
4.2.1	Slackware CD-ROM上的/kernels文件夹	35
4.2.2	从源码编译内核	36
4.2.3	2.4.x版本的内核编译	36
4.2.4	2.6.x版本的内核编译	38
4.2.5	使用内核模块	38
第五章	网络配置	39
5.1	NFS	39
附录 A	GNU通用公共许可证	40

写在前面

无心插柳

笔者是从Slackware 13.1开始接触Slackware的，光盘中就附有Slackbook，一本很好的官方指南。但由于年代久远（第二版是2005年发行的，写作时是2012年），Slackware已经发生了很大的改变，其中的很多内容已不再适用。在网上发现2009年是就已经出了Slackbook 3.0的beta版，可正式版仍迟迟不出。一方面为了学习Slackware，另一方面为了为Slackware尽一点绵薄之力，遂决定翻译Slackbook。

笔者为Slackbook 3.0的beta版翻译了六章，由于beta版的内容实在过于简单，与Slackbook 2正式版实有天壤之别，最终决定放弃。但Slackbook 2为已经过时，实用价值大打折扣，遂决定在Slackbook 2基础之上，以翻译为主要途径，对其中过时的内容加以更新，提供一本好用的Slackware中文指南。

书中会尽量以笔者的使用经验对相关的内容进行陈述，尽量使用第一人称。另外，由于很多部分是翻译而来，其中会夹杂着以原作者为第一人称的描述。请见谅。另外，笔者水平有限，如有不恰当之处，敬请批评指正。

目标受众

Slackware Linux 操作系统作为一个功能强大的平台，是为基于Intel处理器的计算机设计的。它的设计目标是成为一个稳定，安全，实用的高端服务器以及功能强劲的工作站。

本书的初衷是带你走进Slackware Linux操作系统。这并不意味着本书涵盖了Slackware发行版的方方面面，而是为你展示它能做些什么，并且教会你使用该系统的基础知识。

如果你是Slackware的老手，那么我们希望本书可以作为参考手册使用。当然，我们也希望，如果有人向你问起Slackware，你可以向他们介绍介绍本书。Slackware，谁用谁知道。

当然，这本书不可能像小说一样吸引人，我们会尽我们所能地写得更有趣。写得好的话说不定还会有人请我们写剧本？当然，我们最希望的是你能从中学习到你认为有用的知识。

现在，准备被亮瞎吧！

为什么写本新的？

在现存的Linux发行版中，Slackware Linux算得上是爷爷辈的，但这并不意味着它与时代脱节。Slackware的确尝试过保持纯正的Unix风格，但还是没能逃过时代的进步的历程。子系统不一样了，窗口管理器变来变去，人们又发明了新的方法来管理复杂的现代操作系统。我们确实反对为了改变而改变，但当事物进化了，相应的文档就变得陈旧了，一切都是无法避免的——包括书籍。

本书中的一些约定

为了保持一致性，同时方便大家阅读，全书遵守一些约定。

印刷上的约定

Italic 斜体字*Italic*用于表示命令、强调、及技术名词第一次出现的地方。

Monospace 等宽字体monospaced用于表示错误信息、命令、环境变量、端口名、主机名、用户名、组名、设备名、变量及代码段。

Bold 粗体字用于表示例子中的用户输入。

用户输入

输入用粗体表示，用以区别其它的字符。至于键组合，我们用‘+’来连接各个字符以表示要同时按下，例如：

```
Ctrl+Alt+Del
```

意味着我们必须同时按下*Ctrl*，*Alt*，以及*Del*键。

有时候是要先后键入的，我们用逗号来分隔它们，例如：

```
Ctrl+X, Ctrl+S
```

就代表我们要先同时输入*Ctrl*和*X*键，接着同时输入*Ctrl*键和*S*键。

例子

以E: \>开始的例子代表这是一个MS-DOS®命令。除非特别说明，这些命令都是在Microsoft® Windows®“命令提示符”之下运行的。

```
D:\> rawrite a: bare.i
```

以#开始的例子代表执行一个命令时必须使用Slackware中超级用户的权限执行。你可以直接以root登陆，也可以以普通用户登陆，之后再使用su(1)来获得超级用户的权限。

```
# dd if=bare.i of=/dev/fd0
```

以%开始的例子代表执行命令时只须使用普通用户权限。除非特别说明，我们将使用C-shell语法来设置环境变量以及其它的shell命令。

```
% top
```

致谢

这个项目是由许多人花费几个月的贡献累积而成的。仅凭我一己之力，是不可能凭空创造出的。我们要感谢许许多多的人，感谢他们无私的奉献：Keith Keller 写了无线网络方面的内容；Joost Kremers单枪匹马完成了emacs这一章；Simon Williams完成了安全这章；Jurgen Phillippaerts完成基础网络命令的内容；Cibao Cu Ali G Colibri带来的启发以及在裤子上揣一脚¹。还有数不清的人为我们提出了建议及文章的修正。下面是一个完整的列表：Jacob Anhoej, John Yast, Sally Welch, Morgan Landry, 及Charlie Law。同时，我也要感谢Keith Keller为这个项目管理邮件列表，感谢Carl Inglis为我们的管理网页。最后但并非不重要，我要感谢Patrick J. Volkerding创建了Slackware Linux，以及David Cantrell, Logan Johnson及Chris Lumens，是他们完成了Slackware Linux 精要的第一版。如果没有他们的初始框架，本书的一切都不可能发生。还有许多其它的人在本项目中的大大小小的方面做了贡献，但在此没有列出。希望他们能原谅我记忆不好。

Alan Hicks。2005年5月。

¹原文为a good kick in the pants. 译者对这些幽默不是很理解。

第一章 Slackware简介

1.1 什么是Linux?

Linux是一个操作系统的内核，是Linus Torvalds于1991年作为一个私人的项目完成的。最初他开始这个项目的目的只是为了不花钱就能得到一个基于Unix的操作系统。另外，他还想学习386处理器的细节。Linux是在遵守通用公共许可（见第1.3节及附录A中对该许可的介绍）下发行的，对于公众它是免费的，且任何人都能自由地学习并进行改善。现在，Linux已经成为操作系统市场上的一个重要的竞争者。人们已经把它移植到了许多架构上运行，包括HP/Compaq的Alpha，Sun公司的SPARC及UltraSPARC，以及Motorola的PowerPC芯片（在如苹果公司的Macintosh及IBM的RS/6000等计算机上运行）。当然世界上，即使没有上千，也有成百的程序员在开发Linux。在Linux上可以运行如Sendmail、Apache及BIND等软件，这些都是当前流行的用来搭建英特网服务器的软件。要时刻记得，Linux这个词代表的是内核——一个操作系统的核心。这个核心的作用是控制计算机的处理器、内存、硬件驱动及外围设备。这也是Linux的所有功能：控制系统的运转并保证程序正常运行。一些公司或个人将Linux内核与一些程序绑定在一起构成一个操作系统。我们将每一个绑定称作一个Linux发行版。

1.1.1 GNU的世界

Linux 内核项目最早是由Linux Torvalds在1991年时独自努力实现的，但正如牛顿说过的“我看得远，是因为站在巨人的肩上”一样，正当Linus Torvalds准备开发内核时，自由软件基金会已经有了协作软件的想法了。他们将他们的想法命名为GNU，GNU是一个递归的缩写词，全称为“GNU’s Not Unix”。GNU的软件从Linux元年1月1日¹开始运行在Linux内核上。人们用他们编译器gcc来编译内核。直至今今天，许多GNU工作仍是每个主流Linux 发行版的基础，从gcc到gnutar。因此，许多自由软件基金会的支持者都坚持认为他们所做的工作有着不小于Linux内核的功劳。他们强烈建议所有的Linux发行版都应该叫作GNU/Linux发行版。

在这个主题上引起了很多口水战，只输于vi与Emacs间的圣战了。本书的目的并不是煽动这个已经炙手可热的讨论的战火，而是为新手阐明术语。当你看到GNU/Linux时，它表示Linux发行版，但你看Linux时，它可能指的是内核，也可能指的是一个发行版。这还是很

¹原文为‘day 1’

难区分的。特别的，因为GNU/Linux很饶舌，所以一般不用。

1.2 Slackware是什么

Slackware，由Patrick Volkerding于1992年末发起，并最早发布于1993年7月17日，它是第一个得到广泛使用的Linux发行版。Volkerding最早认识Linux是在需要为一个项目找一个廉价的LISP解释器的时候。那时候可用的发行版中有一个叫SLS Linux，是Soft Landing Systems公司的产品。Volkerding用的是SLS Linux，并在找到bugs时进行修复。最终，他决定将所有修复的bug合并到他自己的私人发行版中，让他和他的朋友使用。这个私人发行版很快就小有名气，所以Volkerding就决定将它命名为Slackware并对公众开放。在此过程中，Patrick为Slackware添加了一些新的东西：一个很友好的基于菜单系统安装程序，以及软件包管理的概念，包管理让用户在自己的系统上方便地添加、删除及更新软件包。

Slackware能成为现存的最古老的发行版，有诸多原因。例如它从不试图模仿Windows，它尽可能地保持近似Unix。它并不试图用绚丽的指点GUI（图形用户接口(Graphical User Interfaces)）来隐藏一些操作。相反地，它让用户看到底层的内容，以赐予用户最大的可控制性。它的开发并不是为了赶什么进度，新版本只在准备好的时候才出。

Slackware适用于那些喜欢学习，喜欢通过配置自己的系统来实现想做的事的那些人。Slackware的稳定性和简单性是多年来人们不断使用它的原因。Slackware目前享有的美名是——一个坚固的服务器，一个严肃的工作站。你会发现，Slackware可以运行几乎所有的窗口管理器或桌面环境，也可以不运行其中的任意一个。Slackware提供了强大的服务，每个服务器能使用的地方，都能见到Slackware的身影。Slackware的用户是Linux用户中满意度最高的。废话，我们当然会这么说了。:~)

1.3 开源与自由软件

在Linux社区中，有两种主要的意识形态运动。自由软件运动（我们下面会说到）的目标是使所有的软件都有免费的知识产权。这项运动的追随者们认为，知识产权的限制阻碍了科技的发展并与社区的优点相违背。开源运动的目标与前者差不多，但走了一条更实际的路线。这项运动的追随者们的论点是以使源代码自由可得所带来的商业及技术的好处为基础，而不是一个精神与论理上的律条来推动自由软件运动。

这个运动的底端是一些想更精密控制他们软件的组织。

自由软件运行是由自由软件基金会带头的，而该基金会正是为GNU项目筹款的。自由软件远不止是一个意识形态。一个用烂的说法是“free²指的是演讲的自由，而不是啤酒的免费。”。本质上，自由软件是尝试同时保证用户与开发人员的一些权利。这些自由权利包括基于任何目的地运行程序的权利，自由修改源码的权利，重新发布源码的权利以及共享你所做的修改的权利。为了保证这些自由，他们创建了GNU通用公共许可（GPL）。简单地说，GPL的内容是：任何人在发布一个遵守GPL许可的编译后的程序时，也必须提供源代码，并且只要做出的修改也以源码的形式提供，就可以对原先的源代码进行任意的修改。这就保证了一旦一个程序

²free在英语中即指自由，也指免费

为社区“打开”了，那么除非得到其中每部分代码（包括所做的修改）的作者许可，那么这个程序就不能被“关闭”。Linux下的程序绝大多数是遵守GPL许可的。

要注意一件事，GPL并没有说明有关价格的事。也许就像听起来很奇怪一样，你可以对自由³软件收费。许可中“自由”的部分是针对源代码说的，而不是针对软件的价格据说的。（然而，一旦有人告诉你或给你一个遵守GPL的编译后的软件，那么他也有义务给你软件的源码。）

另一个流行的许可是BSD许可，与GPL不同的是，BSD许可并不要求发布程序的源码。遵守BSD许可的软件只要满足几个条件就可以重新以源码或二进制的形式发布。程序作者的凭证并不作为程序某种形式上的广告。它也免除了作者任何因使用该软件而造成损失带来的责任。Slackware中包含的许多软件是以BSD许可发布的。

站在年轻的开源运动前线的，是称为Open Source Initiative的组织，它单独存在，以获取开源软件的支持，也就是，软件总是可以在得到可执行程序的同时得到它的源码。它们并不提供一个特定的许可，而是支持多种开源许可。

OSI背后的想法是通过让公司自己撰写自己的开源许可，并使许可通过OSI的认证来争取更多的公司参与开源运动。许多公司同意发布源码，但并不想使用GPL许可。由于不能改变GPL许可，所以OSI为它们提供了撰写自己的许可的机会，这个许可最后由该组织认证。

虽然自由软件基金会与Open Source Initiative的工作是互相帮助的，但却不是一个东西。自由软件基金会使用一个特殊的许可并提供该许可下的软件，而Open Source Initiative则寻求所有的开源许可，包括自由软件基金会的那个。谈到使人们自由获得源码的领域时通常分为两项运动，但这分立的两种不同意识形态追求的目标是一致的，因此要信任双方所做出的努力。

³free，有免费的意思

第二章 帮助

在配置一个程序或安装一个硬件时，我们时常需要查看某个命令的帮助。也可能是你想对一个命令更深入地了解，或者查看它别的选项。好在我們有很多方式来得到我们寻求帮助。在安装Slacware时，其中的“F”系列就包含了FAQ和HOWTO文档，你可以选择安装它。一般程序都会包含关于选项、配置文件及使用方法的帮助。

2.1 系统帮助

2.1.1 man

`man`命令（“manual”的缩写）是Unix及Linux系统中最为传统的在线文档。“man手册页”由特定格式的文件构成，涵盖了绝大多数命令，并与软件本身一同发布。很自然地，执行命令`man somecommand`就显示指定命令的man手册页，本例中，则应该显示我们虚构的命令`somecommand`的手册页。

你可能立即会想到，man手册页的数量会急剧上升，最后连高级用户都会被弄得晕头转向的。因此，man手册被分为几个节。这个设定已经存在很久了；久到我们会经常看到一些命令、程序甚至是编程库函数在引用时都附上了man的节号。例如：

你可能会看到诸如`man(1)`的引用。其中的数字告诉我们“man”的文档在第一节（用户命令）；你也可以用命令`man 1 man`命令来为“man”指定查看手册的第一节。在一个同名的项有多个手册页时，指定man应查看的节号是很有用的。

节号	内容
第 1 节	用户命令（只含介绍）
第 2 节	系统调用
第 3 节	C库调用
第 4 节	设备及特殊文件
第 5 节	文件格式及协议（即， <code>wtmp</code> 、 <code>/etc/passwd</code> 及 <code>nfs</code> 等）
第 6 节	游戏（只含介绍）
第 7 节	一些约定，宏包等（即 <code>nroff</code> , <code>ascii</code> 等）
第 8 节	系统管理（只含介绍）

表 2.1: man手册分节

除了`man(1)`，还可以使用其它命令：`whatis(1)`及`apropos(1)`。这些命令的目的是使我们更容易在man系统中找到有用信息。

`whatis`命令会为系统命令给出非常简洁的描述，就像放在口袋的小抄一样。

例如：

```
$ whatis whatis
whatis []          (1) - search the whatis database for complete words
```

`apropos`命令的作用是：给定一个关键词，在man手册页中搜索含有该关键词的内容。

例如：

```
$ apropos wav
SDL_FreeWAV        (3) - Frees previously opened WAV data
SDL_FreeWAV []     (3) - Frees previously opened WAV data
SDL_LoadWAV         (3) - Load a WAVE file
SDL_LoadWAV []     (3) - Load a WAVE file
cdda2wav []        (1) - a sampling utility that dumps CD audio
data into wav sound files
cwaves []          (6) - languid sinusoidal colors
fadeplot []        (6) - draws a waving ribbon following a sinusoidal path
interference []    (6) - decaying sinusoidal waves
oggdec []          (1) - simple decoder, Ogg Vorbis file to PCM
audio file (WAV or RAW)
pilot []           (1) - wav - Decodes Palm Voice Memo files to
wav files you can read on your desktop
wavelan []         (4) - AT&T GIS WaveLAN ISA device driver
```

如果你想深入了解这些命令，请阅读它们的man手册以获取更多信息。:)

2.1.2 /usr/doc文件夹

我们构建的多数软件包含有一些文档：README文件、使用说明及许可文件等。源码中包含的任何文档都安装在系统的/usr/doc文件夹。每个文件都会（一般而言是的）在/usr/doc中创建自己的文件夹，并将文档放在该文件夹中，该文件夹的名字遵守如下约定：

```
/usr/doc/$program-$version
```

其中，`$program`表示你想查找的程序的名字，`$version`字段（很明显）表示在系统中安装的软件的版本号。

例如，想阅读关于命令`man(1)`的文档，你可能想用`cd`命令切换到目录：

```
$ cd /usr/doc/man-$version
```

如果对应的man手册页没有提供足够的信息，或者没有你想找的那些信息，那么接下来可以考虑阅读/usr/doc文件夹下的相关内容。

2.1.3 HOWTO及mini-HOWTO

归功于开源社区的最真挚的精神，我们才有了HOWTO/mini-HOWTO。这些文件跟名字一样，是一些如何完成某些任务的文档或指南。如果你选择了安装HOWTO文档，那么它们会被安装在/usr/doc/Linux-HOWTOs，mini-HOWTO会被安装在/usr/doc/Linux-mini-HOWTOs。

该软件包中还含有FAQ集，FAQ是Frequently Asked Questions，中文含义为“常问问题”。这些文档是以问答的形式写成的。如果你只是想快速地解决一个问题，那么通常选择查看FAQ是极其有用的。如果在安装时选择安装FAQ，那么它们会被安装在/usr/doc/Linux-FAQs目录下。

在你不知道如何处理一些东西时，这些文档通常是值得阅读的。它们涵盖了相当大的范围，而且解决方法通常很详细，详细到不可思议。好东西是吧！

2.2 在线帮助

除了Slackware中可以提供安装的这些文档外，在网上还在大量的在线资源可供我们学习。

2.2.1 官网及论坛

Slackware官网¹

Slackware的官网有时比较过时，但还是有关于最新的Slackware版本的一些信息。有一个时期官网上是有在线帮助的，但后来多了很多人来骚扰生事，导致维护这个论坛变得很不容易，于是Patrick就把论坛关了。如果想找的话，可以在<http://www.userlocal.com/phorum/>上找到之前数据的一个可搜索的归档。

在<http://slackware.com>上的论坛挂掉之后，一些其它的站点如雨后春笋般崛起以提供Slackware的论坛支持。经过再三思量，Pat决定将<http://www.linuxquestions.org>作为Slackware的官方论坛。

2.2.2 E-mail支持

据说只要购买官方的CD，就可以通过电子邮件得到开发人员的免费安装支持。话虽如此，请记住，我们Slackware的开发人员（及绝大多数的用户）是“老派²”。这意味着我们更倾向于帮助那些真的有兴趣的“自助者”。无论谁给我们发email问问题，我们都会尽我们所能帮助他们。然而，请在发email前查阅相关的文档及网站（尤其是FAQ及下列的一些论坛）。因此通过这些途径你可能更快得到答案，同时我们也可以少回一些email，显然，我们也才能够更快地帮助那些需要帮助的人们。

¹ www.slackware.com

² The Old School

技术支持的email是support@slackware.com³。其它的邮箱地址及联系信息都在官网上给出。

2.2.3 Slackware的邮件列表

我们有许多邮件列表，不论是摘要式的，还是正常格式的。请查阅如何订阅邮件列表的说明。

要订阅邮件列表，请发送邮件到

majordomo@slackware.com

邮件正文写上“subscribe [列表名]”。列表名的描述在下面（使用下面列出的名字中的任意一个）。

邮件列表的完整归档可以在Slackware的官网上找到：<http://slackware.com/lists/archive>

slackware-announce

slackware-announce 邮件列表是关于新版本的声明，主要的一些更新及其它一般的信息

slackware-security

slackware-security 邮件列表是关于安全问题的声明。任何直接涉及到Slackware的任何攻击或者漏洞都会立即在这份列表上发布。

这些列表还有对应的摘要版本。这意味着你可以每天得到一条大的信息而不是每天得到一大堆的信息。由于Slackware的邮件列表并不允许用户发表信息，因此该列表的流量很小，所以多数用户认为摘要版没什么用。但如果你需要的话，只要订阅slackware-announce-digest或slackware-security-digest即可。

2.2.4 非官方的网站及论坛

网站

Google (<http://www.google.com>)

搜索引擎的功夫大师。当你积极地、真心地想找到一个主题的每一个信息：它是不二之选。

Google:Linux(<http://www.google.com/linux>)

专门搜索Linux版块。

Google:BSD(<http://www.google.com/bsd>)

专门搜索BSD的内容。Slackware是一个类Unix的通用系统，通用到经常可以在这里找到一些与Slackware百分百相关的详细信息。很多时候，一个BSD内容的搜索会比一般的面向大众的Linux搜索得到更多的技术细节。

³作者没有给他们发过邮件，不知道该邮箱的有效性。

Google:Groups(<http://groups.google.com>)

搜索几十年来Usenet上的文章来发掘你的智慧之光。

<http://userlocal.com>

一个关于知识、好建议、第一手经验及有趣文章的宝库。通常我们会在这里听到Slackware世界的新进展。

网络资源**linuxquestions.org ⁴**

Slackware 官方认可的论坛。

LinuxISO.org Slackware论坛 ⁵

“一个下载Linux和寻求帮助的地方”。

alt.os.linux.slackware FAQ ⁶

另一个FAQ。

Usenet小组 (NNTP)

Usenet一直以来都是geek们聚集并互相帮助的地方。其中有几个致力于Slackware的内容。但里面的人更多的是在行的人。

alt.os.linux.slackware

alt.os.linux.slackware, 更为人所知的名字是aols (不要和AOL®混起来!), 在遇到Slackware的问题时, 是获取相关的技术帮助的最活跃的地方。就像所有的Usenet新闻组一样, 一些不帮忙的参与者 (“山精”们⁷) 总是遭到大家的非议。学会无视那些山精们, 认出那些真心帮助别人的人, 对于使这个资源最大化利用很重要。

⁴<http://www.linuxquestions.org/questions/forumdisplay.php?forumid=14>

⁵貌似和Linuxquestions.org合并了, 新网址是iso.linuxquestions.org/slackware

⁶貌似挂掉了。<http://wombat.san-francisco.ca.us/perl/fom>

⁷斯堪的那维亚神话中的, 邪恶的巨怪或顽皮的侏儒

第三章 安装

使用Slackware前，当然要先获取Slackware并进行安装。不论是购买Slackware 的CD或是从网上免费下载都是很容易的。只要对自己的电脑有一定的基础知识，并且想进一步学习，安装Slackware是非常容易的。安装程序本身就像是一步一步进行的。因此，我们很快就能熟悉这个过程并快速地运行它。事实上，Slackware鼓吹自己是所有功能完善的Linux发行版中安装时间最短的版本之一。

3.1 获取Slackware

3.1.1 官方CD及套装

官方的Slackware CD套装可以从Slackware公司得到。CD套装包含6个CD¹，第一张CD包含基本安装所需要的所有软件，具体包括A/AP/D/E/L/N系列及可启动的安装器，内核，testing/，及Slackwbook。第二张CD包含一些文档及X系统，具体包括F/K/T/TCL/X/XAP/Y、L系列的源码及/testing kernel source。第三张CD含有KDE系列及A/AP/E/F/安装器的源码。第四张CD含有KDEI、/extra软件包及D系列的源码。第五张CD含有KDE/XAP系列的源码。第六张CD包含/pasture软件包、K/N/T/TCL/X/Y的源码及USB和PXE的安装器。

你也可以购买一个套装，其中含有6个CD及Slackbook的印刷版，还有一些简洁的Slackware工具，可供你显摆你作为geek的自尊。我们倾向于通过Slackware store来从网上购买Slackware商品。

<http://store.slackware.com>

你也可以通过打电话或发邮件来订购。

方式	详细联系信息
电话	1-(925)674-0783
网页	http://store.slackware.com
Email	orders@slackware.com
写信	114 Claremont Drive, Brentwood, CA 94513

表 3.1: Slackware Linux公司的联系信息

¹这里以Slackware 13.37为例，参见<http://slackware.com/getslack/torrents.php>

3.1.2 通过互联网

Slackware可以从网上免费下载，你也可以发送email邮件问一些支持问题，但我们会优先考虑那些购买官方CD的人。为什么这么说呢？因为我们收到了一大堆的邮件，但我们的时间是有限的，所以在发送email之前，请先阅读第二章的帮助。

Slackware官网为：

<http://www.slackware.com>

Slackware的FTP主站为：

<ftp://ftp.slackware.com/pub/slackware>

请记住，虽然我们的FTP站点是为公众开放的，但带宽有限。在下载时请优先考虑离你近一点的镜像站下载。你可以在我们的网址上找到一个镜像站列表：

<http://www.slackware.com/getslack>

3.2 系统要求

一个简单安装的Slackware的最小要求²如下：

硬件各类	具体要求
处理器	586架构以上
内存	32MB以上
硬盘空间	1GB以上
多媒体驱动器	4倍速的CD-ROM以上

表 3.2: 系统要求

如果你有可启动的CD盘，那么就不需要软盘驱动器了。当然，这表明如果你没有光盘驱动器，那么你就得有一个软盘驱动器来进行网络安装。如果采用NFS方式安装的话，还需要一张网卡。请参见NFS一节获取更多信息。另外，如果你并不是在全新的系统上安装Slackware，还可以采用从硬盘安装。

声称只要1G的硬盘有点狡猾了。因为对于最小安装而言1G空间是可以的，但如果你是采用完全安装，那么至少要有2GB的硬盘空间³，另外还得准备出存放私人文件的硬盘空间。多数用户并不采用完全安装，事实上，许多人在只有100MB硬盘空间的机器上运行Slackware。

Slackware可以安装到内存少，硬盘小或CPU落后的系统上，但如果这么做，就要付出点代价。如果你正面临这样的情况，那么可以查看CD上的LOWMEM.TXT文件，它包含了一些有用的信息。

3.2.1 软件包系列

由于要求简洁的原因，历史上Slackware被分为软件系列。那时候，人们要想连接到FTP服务器上，只能通过奇慢无比的波特率300的调制解调器，所以Slackware被拆分成不同的集合，

²这是Slackbook 2的内容，对于Slackware 13.37不知是否还适用。

³Slackware 13.37的完全安装需要5.6GB左右

而这些集合的大小适合存放于软盘上，所以用户只需要下载和安装他们感兴趣的软件集合。今天，Slackware中使用软件包系列的目的，主要是用来对软件包进行分类。用软盘安装的日子已不复存在。下面是对软件包系列的一个简要描述。

系列	内容
A	基本系统，其中包含的软件足以让我们启动并运行一个系统，且有一个文本编辑器及基本的通信软件
AP	一些不需要X Window系统的应用软件。
D	软件开发工具。包括编译器、调试器、解释器及man手册等。
E	GNU Emacs
F	FAQ、HOWTO及其它一些文档。
K	内核源码。
KDE	KDE桌面环境。一个外观类似MacOS及Windows的X桌面环境。其中还包含作为KDE依赖的Qt库。
KDEI	KDE桌面的国际化语言包
L	库文件。其它程序要用到的动态链接库。
T	teTex文档系统。
TCL	工具命令语言。包括Tk、TclX及TkDesk等。
X	基本的X Window系统
XAP	主要桌面环境中不包含的一些X应用程序（如Ghostscript及firefox 等）。
Y	BSD控制台游戏。

表 3.3: 软件包系列

3.2.2 安装方法

本节中介绍有一些方法相当古老，笔者只尝试过CD/DVD安装及硬盘安装，其它都没试过，只能照书翻译。

软盘

早先的时候是可以通过软盘的方法安装Slackware的，但随着软件包的增大（是的，一些单独的软件在增大），软盘的方法已经不得不被淘汰了。最后一个能用软盘安装的版本是Slackware 7.1的部分安装。A系列和N系列几乎可以完全安装，这就提供了安装该发行版其它软件的一个基础系统。如果你考虑使用软盘安装（尤其在老的硬件上），那么我们推荐用其它的方法安装或者用Slackware老的版本。Slackware 4.0和7.0在这种情况下还是很可靠的。

注意，如果你想采用CD安装，但却没有可启动的CD，那么还是需要软盘的，同样，对NFS安装也如此。

现在多数的电脑连软驱都没有了，更别说软盘了，如果你是一个新手，完全不用考虑这种方法，当然，老手也几乎不需要考虑。

光驱

可启动的CD可以在Slackware Linux公司的官网上获得（参见获取Slackware这一节）。采用基于CD/DVD的安装全更容易一些。如果你没有可启动CD的话，那就需要从软驱启动了。另外，如果你的硬件使用启动CD上的内核有问题的话，你也可能需要使用特制的软盘了。

对于Slackware 8.1版本，我们使用了一种新的方法制作启动CD，这种方法对于一些特定的BIOS芯片工作得不太好（相比于当前多数Linux CD遇到的问题而言，这个问题简直不值一提）。如果你遇到这种情况，我们还是建议你使用软盘启动。

第3.2.2节和第3.2.2节提供了关于选择及创建启动软盘的信息，也许会很有用。

NFS

NFS（the Network File System 网络文件系统）是一个远程机器上使用的文件系统。NFS安装使我们可以从网络上的另一台机器上安装Slackware。作为获取源的那台机器要为安装机导出Slackware发行版的目录树。当然，这需要关于NFS的一定知识，在第5.1节中有介绍。NFS可以通过PLIP（parallel port并行端口）、SLIP及PPP（一个调制解调器连接）进行安装。然而，我们建议最好通过网卡进行连接。毕竟通过打印机端口来安装一个操作系统是非常非常慢的。

启动盘

软盘已经几乎被淘汰，这里按slackbook 2进行翻译，以防某些万一需要的情况。Slackware 13.37的光盘已经明确写明，不再支持使用其中的内核制作启动盘了。

所谓启动盘指的是一张软盘，通过它我们真正启动并开始安装。它包括一个压缩的内核镜像，用来在安装过程中控制硬件。因此，这是必须要有的（除非你是从CD启动，就像在光驱这一节中讨论的一样）。启动盘镜像位于bootdisks/目录下。

Slackware中，可以选择不止一个启动盘（具体而言是16个。）。在文件bootdisks/README.TXT中所有启动盘的完整列表及对应的描述。然而，多数用户可以使用bare.i（IDE设备使用）或scsi.s（SCSI设备用）的启动盘镜像。

参见第3.2.2节以获取关于从镜像制作启动盘的介绍。

启动之后，会有提示要求插入根磁盘。我们建议你按照启动盘的提示进行。

根磁盘

这节的内容也是被淘汰的。

根磁盘包含了setup程序及安装过程中使用的一个文件系统。这个也是必须的。根磁盘镜像位于rootdisks/文件夹下。我们需要从install.1及install.2两文件制作两个根硬盘。在这个目录下，你还可以找到network.dsk、pcmcia.dsk、rescue.dsk及sbootmgr.dis的磁盘镜像。

追加盘

如果你执行的是NFS安装或安装系统中带有PCMCIA设备，那么你需要一个追加盘。追

加盘的镜像也是在rootdsk/文件夹下，文件名为network.dsk及pcmcia.dsk。近来也追加了rescue.dsk及sbootmgr.dsk盘。急救盘是一个安装在软盘上的，可以在4MB内存上运行的一个根磁盘镜像。它包含了一些基本的网络工具及vi编辑器，我们可以用它在一个破坏了的系统上进行快速的修复。sbootmgr.dsk是用来启动其它设备的。

根磁盘在加载之后会指示你使用其它的追加盘的。

制作启动盘

选择好了启动盘镜像，就需要将它放到软盘中，根据使用的操作系统的不同，这个过程也有一些小差别。如果运行的是Linux（或其它类Unix的操作系统），你需要使用dd命令，假设使用的是bare.i镜像文件，软驱对应的设备文件是/dev/fd0，那么制作bare.i软盘的命令为

```
% dd if=bare.i of=/dev/fd0
```

如果运行的是Microsoft的操作系统，你需要使用RAWRITE.EXE程序，在我们的发行版中也包含了该程序，它与软盘镜像在同一个文件夹下。同样的，假设用的是bare.i镜像，软驱为A:，在DOS提示符下，输入下面的命令即可。

```
C:\ rawrite a: bare.i
```

3.3 启动安装器

启动安装器很简单，只要将Slackware安装盘插入你的CD或DVD驱动器，之后重启就行了。另外，你可能要先进入BIOS修改启动顺序，把光驱的启动顺序放在硬盘的启动顺序之前。有一些计算机可以在系统启动时，按下某些键来动态的改变启动顺序。由于每台电脑都是不同的，我们不可能为所有的电脑提供一个修改启动顺序的说明，但修改的方法在几乎所有的机器上都是很容易的。

一旦你的电脑从CD启动了，你就会被带到一个界面，要求你输入特定的一些内核参数，从这里开始，你就可以像使用急救盘那样使用这个安装器了。一些系统可能要求特定的内核参数才能启动。但这是极少数的情况，多数情况下，只要按下回车键，启动内核就可以了。

```
Welcome to Slackware version 13.37 (Linux kernel 2.6.37.6)!
```

```
If you need to pass extra parameters to the kernel, enter them at the prompt  
below after the name of the kernel to boot (huge.s etc).
```

```
In a pinch, you can boot your system from here with a command like:
```

```
boot: huge.s root=/dev/sda1 rdinit= ro
```

```
In the example above, /dev/sda1 is the / Linux partition.
```

```
This prompt is just for entering extra parameters.  If you don't need to enter
any parameters, hit ENTER to boot the default kernel "huge.s" or press [F2]
for a listing of more kernel choices.
```

启动过程中，你应该会看到屏幕上刷了一排排了文字。不必惊慌，这是很平常的。这是内核启动过程中，对硬件进行检测并准备载入操作系统（现在载入的是安装器）时产生的信息。如果你感兴趣的话，之后可以用`dmesg(1)`来阅读这些信息。通常，如果你的硬件有问题，这些信息对于排除问题是十分重要的。一旦内核完成对硬件的检测，屏幕上的信息就会停下，让你选择对非标准美式键盘的支持。

```
<OPTION TO LOAD SUPPORT FOR NON-US KEYBOARD>

If you are not using a US keyboard, you may not load a different
keyboard map.  To select a different keyboard map, please enter 1
now.  To continue using the US map, just hit enter.

Enter 1 to select a keyboard map: _
```

输入1并输入**Enter**（回车），就能看到一张键盘映射表。选择符合你的键盘类型的那一项就可以继续了。

```
Welcome to the Slackware Linux installation disk! (version 13.37)

#####  IMPORTANT!  READ THE INFORMATION BELOW CAREFULLY.  #####

- You will need one or more partitions of type 'Linux' prepared.  It is also
  recommended that you create a swap partition (type 'Linux swap') prior
  to installation.  For more information, run 'setup' and read the help file.

- If you're having problems that you think might be related to low memory, you
  can try activating a swap partition before you run setup.  After making a
  swap partition (type 82) with cfdisk or fdisk, activate it like this:
    mkswap /dev/<partition> ; swapon /dev/<partition>

- Once you have prepared the disk partitions for Linux, type 'setup' to begin
  the installation process.

- If you do not have a color monitor, type:  TERM=vt100
  before you start 'setup'.
```

```
You may now login as 'root'.
```

```
slackware login: root
```

其它的一些发行版会直接启动进入一个专门制作的安装程序，与之不同，Slackware 的安装器会让你进入一个载入系统内存的轻量Linux发行版。之后，我们就可以使用这个轻量的Linux手工运行安装程序，当然，我们也能将它作为一个应急启动系统，在系统不能启动时对系统进行修复。现在，你已经用root登陆了（安装器中不需要密码），这时就应该设置硬盘了。在这个节点上，如果你希望得到一个加密的根分区，你可以安装支持RAID或LVM的软件，但这个主题已经超出了本书的范畴。如果你希望使用这些先进的工具，我强烈建议你查看CD上的README_RAID.TXT，README_LVM.TXT及README_CRYPT.TXT文件。大多数用户并没有这种需求，因此应该直接跳到下一步——分区。

3.4 分区

与其它发行版不同的是，Slackware的安装器中没有使用图形化的磁盘分区工具；而取而代之的是fdisk(8)及cfdisk(8)，二者都是命令行工具。cfdisk基于curses库⁴，fdisk则不然。不管你用哪一个区别都不是很大，本书中，我们只讨论fdisk。

要对你的硬盘正确分区，首先要了解如何识别它们。在Linux中，所有的硬件都是用一个特殊的称为设备文件的文件来识别。这些文件都（不失一般性地）放在/dev文件夹下。硬盘，不论是古老的IDE（PATA）盘，还是串行的ATA（SATA）盘，内核都把它当作SCSI设备。因此，会为它们创建一个诸如/dev/sda的设备节点。如果你不知道你的硬盘的设备节点名是什么，fdisk可以帮你。

```
root@slackware:/# fdisk -l

Disk /dev/sda: 72.7 GB, 72725037056 bytes
255 heads, 63 sectors/track, 8841 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

这里可以看到，我的系统有个72.7G的硬盘，位置是/dev/sda。你也可以看到一些额外的信息（这个例子中，实际上有3个SCSI硬盘，由一个RAID控制器控制，所以看起来只有一个硬盘）。fdisk的参数[-l]用来告诉fdisk，同时显示硬盘本身及在硬盘上检测到的所有分区，但这并不会对硬盘做任何改变。要对我们的硬盘进行分区，我们需要告诉fdisk具体如何操作。

```
root@slackware:/# fdisk /dev/sda

The number of cylinders for this disk is set to 8841.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
```

⁴为程序提供字符终端用户接口。通俗地说：为字符终端（命令行）提供更好的显示，可理解为终端下的GTK等。译者注。

- 1) software that runs at boot time (e.g., old versions of LILO)
- 2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)

Command (m for help):

现在，我们告诉fdisk应该对哪个盘分区，fdisk在显示一串讨厌的警告信息后，进入了命令模式。1024个柱面的限制早已不成问题，Slackware引导装载程序完全可以从拥有更大柱面的磁盘中启动。键入[m]然后输入回车，就可以得到更多的信息来告诉我怎么做。

```
Command (m for help): m
Command action
  a   toggle a bootable flag
  b   edit bsd disklabel
  c   toggle the dos compatibility flag
  d   delete a partition
  l   list known partition types
  m   print this menu
  n   add a new partition
  o   create a new empty DOS partition table
  p   print the partition table
  q   quit without saving changes
  s   create a new empty Sun disklabel
  t   change a partition's system id
  u   change display/entry units
  v   verify the partition table
  w   write table to disk and exit
  x   extra functionality (experts only)
```

现在你就知道了什么样的命令会有什么样的行为，是时候开始为我们的磁盘分区了。最低的要求是必须要有一个单独的/分区，并且，你也应该创建一个交换分区。你也可能想单独分一个/home分区来存储用户文件（把所有的用户文件放在单独的一个分区中，会使之后对系统升级，或是全新安装一个不同的Linux操作系统更为容易）。因此，我们继续分三个区。创建新分区的命令是[n]（你可能已经从帮助中看到了）。

```
Command: (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4):1
```

```

First cylinder (1-8841, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-8841, default 8841): +8G

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (975-8841, default 975): 975
Last cylinder or +size or +sizeM or +sizeK (975-8841, default 8841): +1G

```

这里，我们创建了两个分区。第一个的大小为8G，第二个只有1G。我们可以用[p]命令来查看当前分区情况。

```

Command (m for help): p

Disk /dev/sda: 72.7 GB, 72725037056 bytes
255 heads, 63 sectors/track, 8841 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start       End      Blocks   Id  System
/dev/sda1            1        974       7823623+  83  Linux
/dev/sda2          975       1097       987997+   83  Linux

```

这两个分区的类型都是“83”，也就是标准Linux文件系统。我们需要将/dev/sda2的类型设置为“82”，才能将其转变为交换分区。我们使用fdisk的[t]命令来完成这个工作。

```

Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 82

Command (me for help): p

Disk /dev/sda: 72.7 GB, 72725037056 bytes
255 heads, 63 sectors/track, 8841 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start       End      Blocks   Id  System
/dev/sda1            1        974       7823623+  83  Linux
/dev/sda2          975       1097       987997+   82  Linux swap

```

所谓的交换分区是一个特殊的分区，Linux内核用它来作为虚拟内存。如果你一不小心内存不够用了，内核会将内存中的一些东西移动到交换分区中，用以防止崩溃。交换分区的大小完全取决于你自己。对于交换分区的大小，大家都有争执，但一个黄金定律就是将交换分区的大小设为系统内存大小的两倍。由于我们机子内存大小为512MB，所以我决定将交换分区设置为1GB。你可能希望测试自己的交换分区大小，看它是不是能最好地工作，但一般情况下，有很大的交换分区并不会有什么坏处。有一种说法，如果你有*很多*的内存（也就是大于2GB），就没必要遵守所谓的黄金定律了。如果你想要使用睡眠功能（挂起到硬盘），要求的交换分区大小至少要 and 物理内存（RAM）的大小一样，要记住。

在这个节骨眼上不能就这么停下，保存好设置然后继续后续的操作。但我还想创建第三个分区并挂载到/home目录下。

```
Command: (me for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 3
First cylinder (1098-8841, default 1098): 1098
Last cylinder or +size or +sizeM or +sizeK (1098-8841, default 8841): 8841
```

最后，保存设置。

```
Command: (me for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
root@slackware:/#
```

现在，我们已经对磁盘完成了分区，执行安装程序的前期工作已经完成。然而，如果你分了其它的区，可以先重启一下，来保证内核能正确读取这些分区。

3.5 setup程序

现在我们分好了区，是时候运行setup程序，安装Slackware了。setup程序包括对分区格式化、安装包以及一步步地执行基本设置。只要在shell提示符下键入setup就可以了。参见图3.1。

3.5.1 帮助

如果之前你没有安装过Slackware，你可以阅读Slackware的帮助菜单，它对Slackware安装

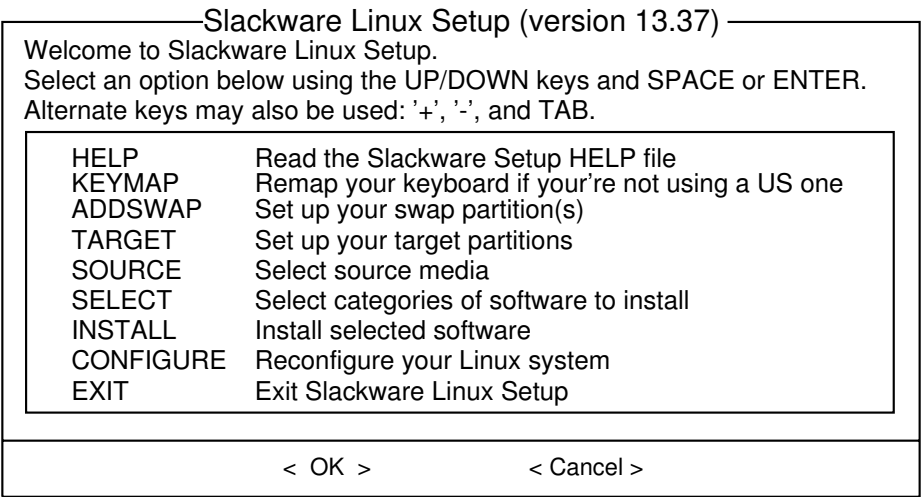


图 3.1: setup程序

器有一个基本的概述。其中的多数信息是关于对安装器的一些基本操作，这些操作也很直观。参见图3.2。

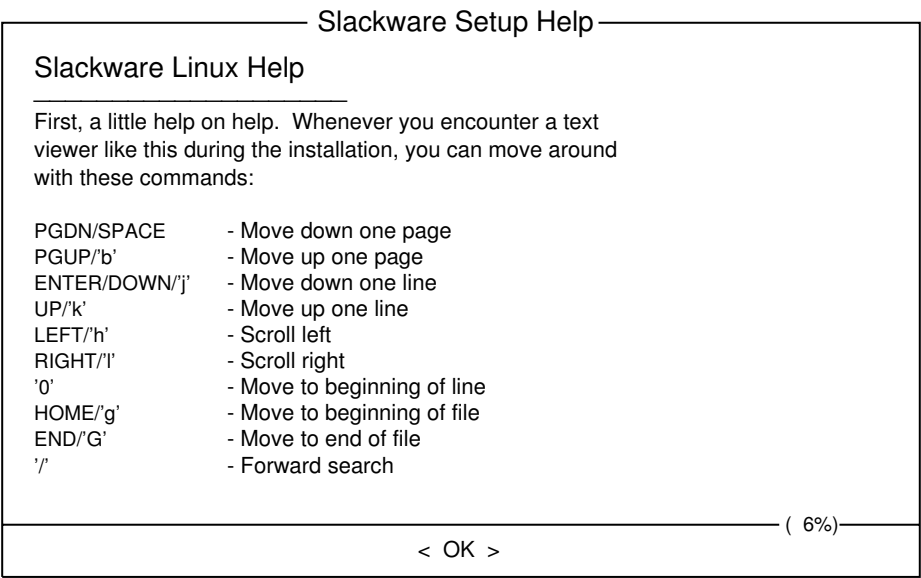


图 3.2: Slackware setup程序帮助

3.5.2 键映射

在继续之前，Slackware给了我们为键盘选择一个不同映射的机会。如果你用的是标准美式键盘，可以直接跳到下一步，但如果你用的是国际键盘，你应该在现在选择一个正确的键映射。这个步骤保证了我们从键盘输入的键与系统的理解是一致的。参见图3.3。

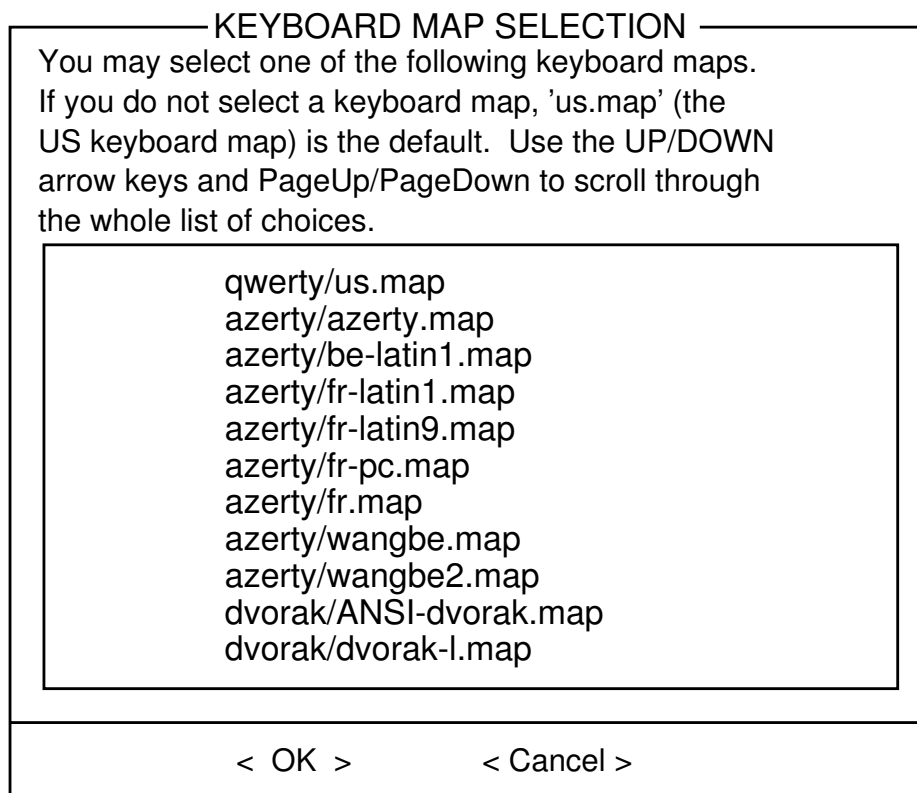


图 3.3: 键盘映射选择

3.5.3 添加交换分区

如果你创建了一个交换分区，那么这一步能让你运行其它对内存敏感的活动，诸如安装软件包之前，启用这个交换分区。交换分区是一个磁盘分区（或者是一个文件，尽管Slackware的安装器不支持交换文件），当计算机的可用内存不足时，会将活动的系统内存拷贝到这个分区中。通过这个方法让计算机能够在活动内存中切换进切换出，从而能使用比计算机实际拥有的更多的内存。本步骤会将你的交换分区加入`/etc/fstab`中，使之在你的操作系统中生效。参见图3.4。

3.5.4 选择安装位置

下一个步骤是选择根分区的位置，以及Slackware要用到的其它分区。安装程序会提示我们选择是否使用某个分区及是否对要使用的分区格式化。如果你要安装到一个新的分区，那么就必须先格式化。如果你要装到的分区中有你不想删除的数据，那么不要进行格式化。例如，如果用户有一个`/home`分区，用以存放用户数据，那么切记在安装时不要对其进行格式化。那么新安装的Slackware就不需要对这些数据进行备份和还原了。根据使用的内核，我们可以选择不同的文件系统，包括reiserfs、ext2、ext3、ext4、jfs及xfs等，一般而言采用ext3或ext4即可。参见图3.5。

进入选择安装位置后第一次选择的是安装根（/）文件系统的位置。之后，你可以根据选择映射其它的分区到该文件系统中。（例如，你可能希望另一个分区，如`/dev/sda3`，作为你

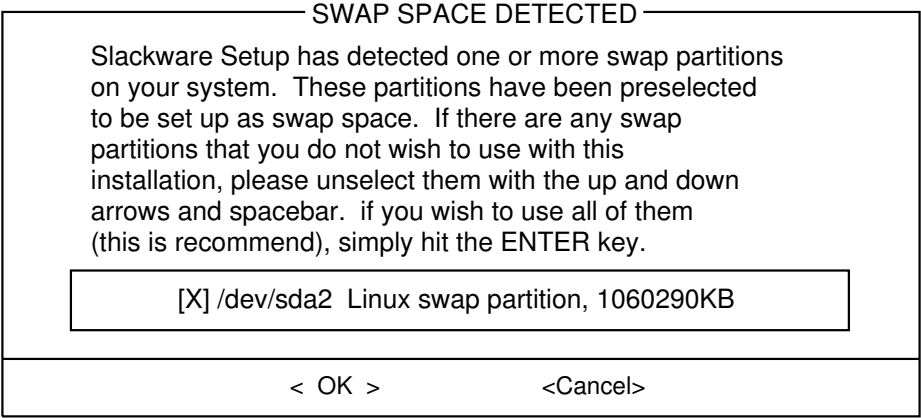


图 3.4: Slackware安装程序之添加交换分区

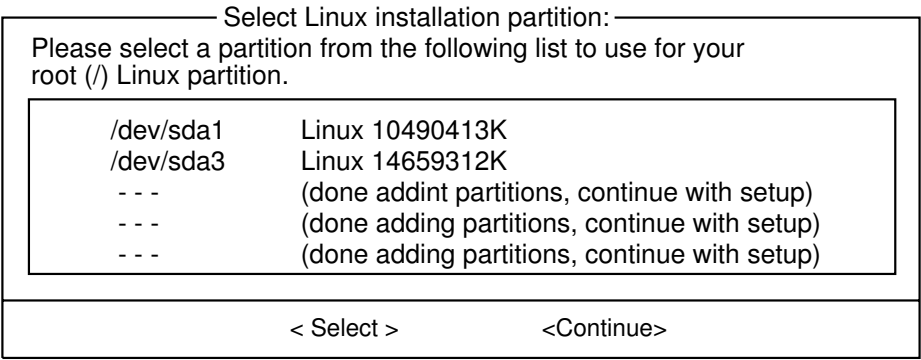


图 3.5: Slackware安装程序之选择安装位置

的Home文件系统。这只是一个例子，你可以按自己的想法进行映射。)

3.5.5 安装源

现在，你要告诉Slackware安装器，在哪可以找到Slackware的软件包。最常见的方法是用Slackware安装CD或DVD，但还可以用其它的一些方式。如果你之前已经把安装包放在之前你设置好的分区中，你可以选择从那个分区或是一个事先挂载的文件夹中安装。（你需要先使用mount(8) 命令挂载那个分区。参见第??章以获得更多信息。）另外，Slackware还提供了许多使基于网络的安装方法，如NFS共享、FTP、HTTP及Samba等方法。如果你选择网络安装，Slackware会先显示提示符，让你输入TCP/IP信息。这里，我们只讨论从DVD安装的情况，但其它的方法也很直接、简单。参见图3.6。

3.5.6 选择安装包

安装器会让你选择安装哪些集合。这些集合在第3.2.1节介绍过了。这个方法能让你方便地跳过那些你可能不想安装的软件包，如在服务器上你可能不想安装X或KDE，或者你压根就不想安装Emacs。注意“A”集合总是必须的。参见图3.7。

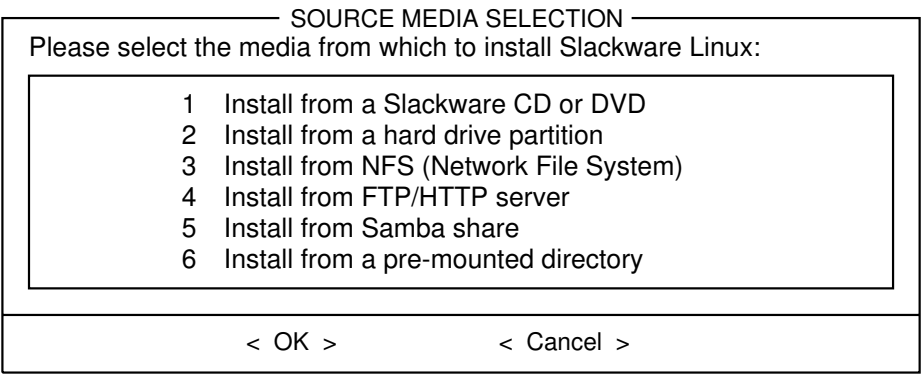


图 3.6: Slackware安装程序之选择安装源

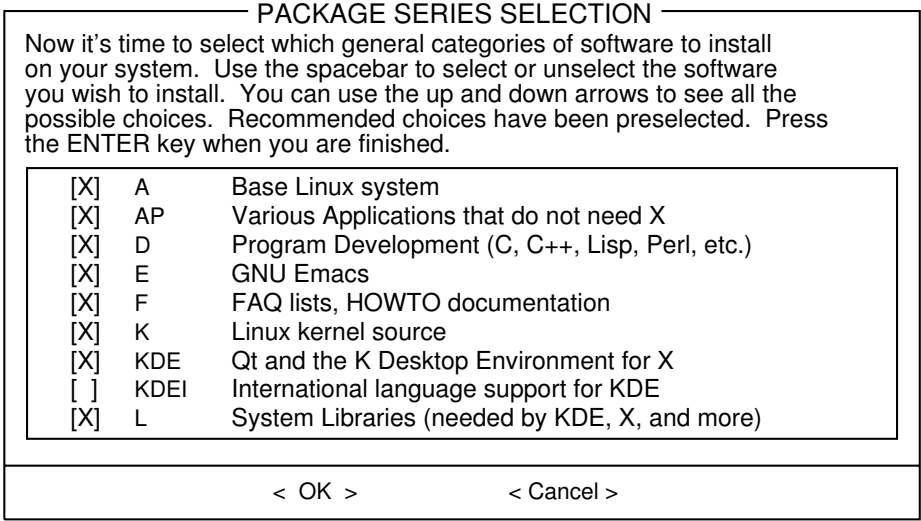


图 3.7: Slackware安装程序之选择软件包

3.5.7 安装

假设你已经完成了“target”，“source”及“select”等选项，接下来，“install”选项让我们在软件系列中选择具体要安装的软件包。如果前面的步骤未完成，它会提示你先完成之前的步骤。该选项可以让你从六个不同安装方法中进行选择：**full**——完全安装、**newbie**——新手安装、**menu**——菜单安装、**custom**——自定义安装及**tag path** 利用TAG安装。

full选项会安装选择的所有系列中的所有软件包。而不会提示其它信息。这是最容易的安装方法，因为我们不想思考到底要安装哪个软件包。当然，这个方法用到的磁盘空间也最多。

下一个选项是**newbie**。这个选项会安装选择的系列中所有必需的软件包。对于其它的软件包，它会给出提示符，让你选择“YES”、“NO”还是“SKIP”。YES表示安装，NO表示不安装，SKIP则跳到下一个系列。另外，你会看到软件包的一个描述及大小来帮助你决定是否要安装它。对于新用户来说，我们强烈推荐使用这个选项，因为它能保证我们安装了所有必需的软件包。然而，由于为每个软件包都显示提示符，所以速度慢。

menu选择是一个相对于**newbie**选项更快更高级的选项。对于每个系列，都显示一个菜单，

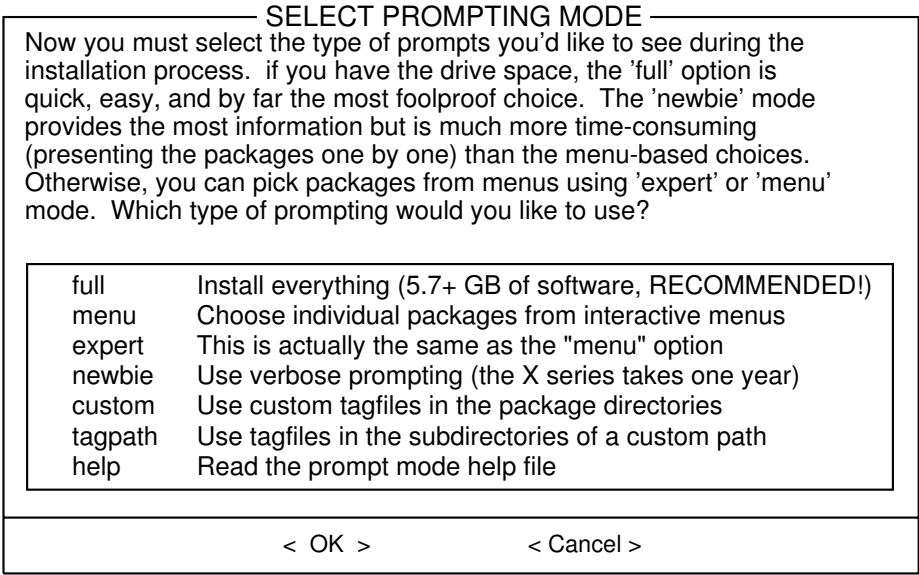


图 3.8: 选择提示方式

我们可以选择是否安装其它不是必需的软件包。必需的软件包不会显示在这个菜单上。

对于更高级的用户，install提供了**expert**选项，这个选项能让你完全控制想要安装的软件包。你可以不安装那些绝对需要的包，得到一个不能用的系统。相反的，你可以精确地控制系统中安装的东西。这个选项对于新手不建议使用，因为很可能会搬石头砸到自己的脚。

custom和**tag path**选项也是为高级用户而设的。这些选项让我们能基于预先制作的tag文件来安装系统。该功能对于批量安装Slackware很有帮助，关于使用tag文件的更多信息，请参见第//TODO:Section18.4//节。

选择了安装方法后，根据不同的方法，会有不同的响应。如果选择的是**expert**或**menu**，那么会出现一个菜单，让你选择要安装的软件包。如果选择的是**full**，会自动开始将软件包安装到目标位置中。如果选择的是**newbie**，在选择完可选的包后，才会开始安装软件包。

注意，安装时可能会出现磁盘空间不足，如果你选择了太多的包，而对应安装位置的剩余空间不足，就会出现问题。最安全的方法是先选择一些包，之后再添加其它的包。这可以通过使用slackware的包管理工具轻松完成。关于这些内容，参见第//TODO:Chapter18//章。

3.5.8 配置

配置这部分会对系统进行一个基本的配置。下面出现的画面很大部分依赖于我们安装的软件包。我们将以完全安装进行介绍。顺序可能与安装过程不太一致。

创建USB启动盘

很多年前，我们用软盘来创建启动盘，在其中存储启动数据。但现在软盘已经几乎被淘汰了，一些新的计算机甚至连软驱都没有，所以，Slackware采用USB设备来创建启动盘。基于安全的考虑，我们也建议你创建一个USB启动盘，防止之后Slackware不能启动。如果你的机

器支持从USB设备启动（现在的机器一般都支持），那么最好在这个步骤时就创建启动盘。切记，用来创建启动盘的U盘会被格式化，所以建议先对U盘中的数据进行备份。

另外，笔者尝试过创建USB启动盘，但4GB的U盘被格成了200MB左右，另外的空间反而不能使用了，虽然之后没有再尝试过，但请大家作好心理准备。

MAKE USB FLASH BOOT	
If your computer supports booting from a USB device, it is recommended that you make a USB boot stick for your system at this time. It will boot your computer straight into the root filesystem on /dev/sda1.	
Please insert a USB flash memory stick and then press ENTER to create a boot stick.	
WARNING! The existing contents of the USB stick will be erased.	
Create Skip	Make a USB Linux boot stick Skip making a USB boot stick
< OK > < Cancel >	

图 3.9: 创建USB启动盘

LILO

这里，会提示是否安装LILO（the LInux LOader；详情参见第//TODO:Section 7.1//节）如果系统中只安装了Slackware，那么选择`simple`就可以了。如果你安装的是双系统（或多系

INSTALL LILO	
LILO (Linux Loader) is a generic boot loader. There's a simple installation which tries to automatically set up LILO to boot Linux (also Windows if found). For more advanced users, the expert option offers more control over the installation process. Since LILO does not work in all cases (and can damage partitions if incorrectly installed), there's the third (safe) option, which is to skip installing LILO for now. You can always install it later with the 'liloconfig' command. Which option would you like?	
simple	Try to install LILO automatically
expert	Use expert lilo.conf setup menu
skip	Do not install LILO
< OK > < Cancel >	

图 3.10: 安装LILO

统)，那么使用`expert`选项会更好一些。关于双系统启动，请参见第//TODO:Section 7.3//节以获取更多信息。我们不推荐你使用第三个选项`do not install`，除非你知道自己在干什么，

知道自己为什么不需要安装LILO。如果选择的是expert选项，会有提示选择LILO的安装位置，通常可以选择将其安装在硬盘的MBR（主引导记录）中，Linux 根分区的superblock中，或是安装在软盘上。

是否使用UTF-8终端

从2.6.24内核起，就提供了一个标准的UTF-8终端，但由于常有一些问题，尽管你使用的是UTF8的locale，选择默认的非UTF8文本终端会更加安全一些。

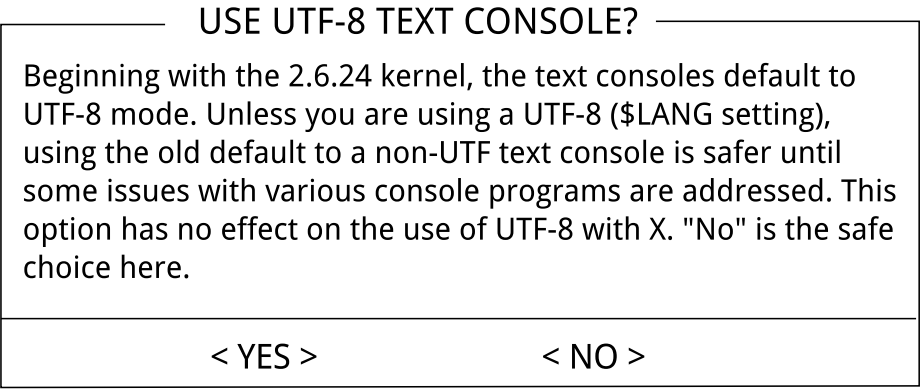


图 3.11: 是否使用UTF-8文本终端

鼠标设置

选择你的鼠标类型，一般选择ps/2类型或usb类型。之后会询问是否在启动时开启gpm(8)。选择了鼠标类型后，系统会创建链接/dev/mouse，并指向默认的鼠标设备。如果启动后鼠标无效，可以手动修改该文件。

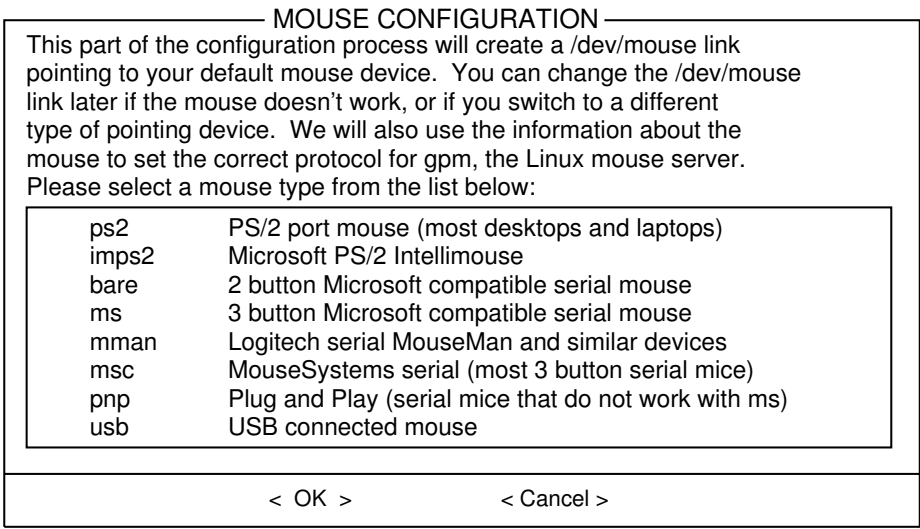


图 3.12: 鼠标设置

网络设置

网络设置实际上就是运行netconfig脚本。详情请参见第//TODO:Section 5.1//节。

选择默认启动的服务

本节是选择启动时默认运行的服务。Slackware默认选择了几个服务，请根据自己是否需要该服务进行相应的选择。如果在服务器或桌面系统上安装Slackware，那么请关闭pcmcia服务。使用空格键进行选择及反选。请记住，开启越多的服务，系统的安全性就越低。

如果你是新手，那么不要考虑安全性问题，尽管选就是了。

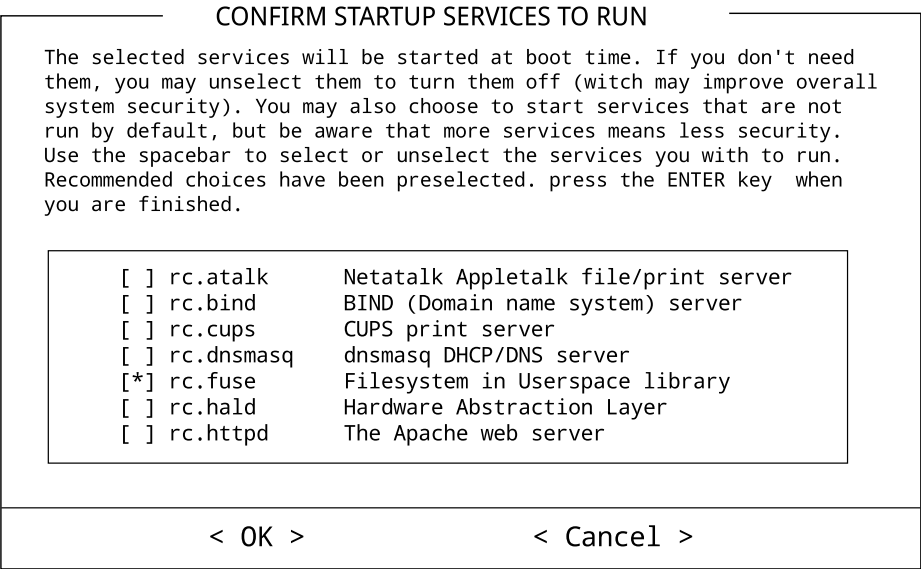


图 3.13: 选择默认启动的服务

终端字体选择

在该选项中可以选选择终端中使用的字体。

个人建议选择“No”，因为一般而言用不到字符终端，另一方面，即使使用，使用默认字体就能很好地显示，其它一些字体还易出现乱码。

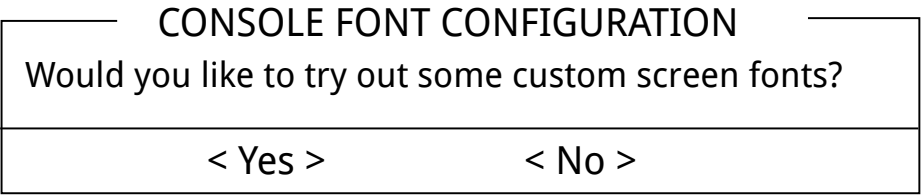


图 3.14: 文本终端字体选择

时区选择

这个步骤很直观，就是要你选择所在位置的时区。开始的选项是让我们选择是否使用UTC，一般而言，选择将时钟设置为本地时间（对应选项“No”）即可。之后选择所在的时区即可。

中国的同学们一般选择“Asia/Shanghai”，即上海时区即可。

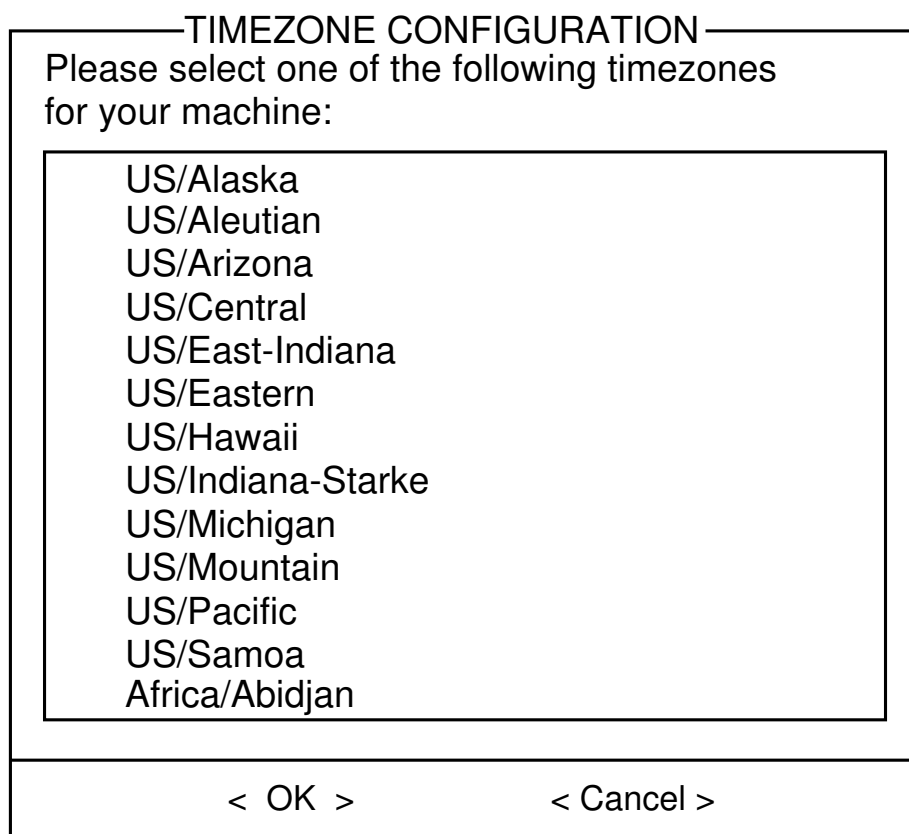


图 3.15: 时区选择

X 窗口管理器

通过本节，我们就可以选择默认的X窗口管理器，参见第//TODO:Chapter 6://章以获取关于X及窗口管理器的更多知识。

不管安装了什么软件包，最后一项配置是为root设置密码。出于安全考虑，设置root密码是不会错的，但是就如Slackware中的其它东西一样，设不设可以自己决定。

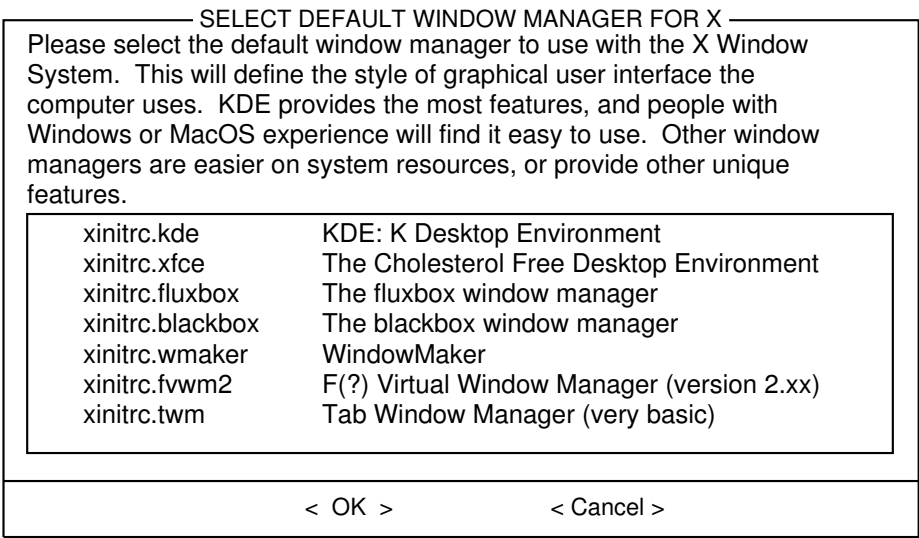


图 3.16: 选择默认的X窗口管理器

第四章 系统配置

在对系统进行进一步的配置之前，先学习关于系统的组织结构以及文件或程序的查找方法不失为一个不错的选择。我们还会介绍如何自己编译内核，如果你正打算这么做的话，本章应该很有帮助。本章的主要内容是让你了解系统的组织结构及相关的配置文件。之后我们就可以配置系统的更为高深的部分了。

4.1 系统一览

在对Linux分解认识之前，了解Linux系统的组成方式无疑是极其重要的。Linux系统与DOS、Windows或是Macintosh（类Unix的Mac OS X例外）等系统有着显著的不同，我们会在下面的几节中帮助你了解Linux的布局，从而使你能够根据自己的需求来配置系统。

4.1.1 文件系统布局

Slackware Linux与DOS或者Windows的第一个显著差别就是文件系统。首先，我们并不使用驱动器的盘符来表示分区。在Linux下，有一个主目录，我们可以类比为DOS下的C:盘。并且系统中的每个分区都挂载到该主目录的某个子目录下。听起来有点像是一个可以无限扩展的硬盘。

我们将这个主目录称为`root`目录，并用一个斜杆（/）来表示。这个概念开始时可能不好理解，甚至有些诡异，但当我们需要增加更多的空间时，这个设计的优势就能很好地体现出来了。例如，我们假设有个硬盘，挂载到/`home`目录的某个子目录上，它的空间已经用完了。虽然大多数人在安装Slackware时都分配了一个较大的`root`分区。由于分区可以被挂载到任意目录下，所以我们可以随意到商店中买一块新的硬盘，并挂载到/`home`目录下，万事OK！现在，我们的系统中就添加了新的空间，整个过程异常简单。

下面，我们要对“/”目录下的主要顶级目录夹进行一个简要的介绍。

bin

该目录下存放一些必需的程序。它包含使用系统的最低要求下的一些程序。该文件夹下包括了如shell及一些文件系统的命令（`ls`、`cp` 等等）。在安装之后，一般就不再对/`bin`目录进行改动。即使进行改动，一般也是由官方的更新所做出的改动。

boot

该目录存在LILO所需的一些文件。Slackware的内核文件也存放于此。该目录在安装后可

能会做一些小的改动。

dev

在Linux中，一切皆文件，诸如串口、硬盘或扫描仪等设备文件也以文件的形式表示。所有的设备节点都存放于/dev目录下。在许多类Unix的系统中都采用了这种策略。

etc

该目录下存放系统的配置文件。从X的配置文件、用户数据库，到系统的启动脚本，都存放于此。随着使用Slackware时间的增长，系统管理员会对该目录越来越熟悉。

home

Linux是一个多用户系统，每个用户都有一个帐户和一个独立的文件夹来存放私人文件，这个文件夹就被称为用户的主目录，默认情况下，它就存放于/home目录下。

lib

该目录用于存放一些基本操作所依赖的库文件。包括C库、动态链接库、ncurses库及内核模块等等。

media

一些设备的默认挂载点。包括CDROM、DVD等的挂载点。

mnt

该目录的作用是作为硬盘或可移动设备的临时挂载目录。其中包括CDROM及软盘的挂载点。注意，该目录与/media的区别在于该目录一般用于临时挂载，而/media一般作为默认挂载点。

opt

该目录用于安装额外的一些软件包。其中的思想是：将软件包安装到/opt/software-package目录下，之后只需要将该目录删除即可卸载该软件包。一般而言，将一些占用空间较大的软件包安装到这里(如google-chrome、matlab、libreoffice及texlive等)。

proc

该目录是个特殊的目录。严格来说，它不属于文件系统的一部分，它是一个虚拟的文件系统，为我们提供内核的相关信息。内核需要让我们知道的信息会保存为文件的形式，并存放在/proc文件夹内。我们也可以通过修改这些文件来向内核传递信息。例如我们可以执行cat /proc/cpuinfo来获取CPU相关的信息。

root

还记得系统管理员的用户名是什么吗？是的，root。与普通用户不同的是，root用户的主目录并不存放在/home/root中，而是放在/root下。这么做的原因很简单，试想，如果/home与/在不同的分区下，那么当/home不能挂载时会发生什么情况？很自然地，我们会以root用户登陆并对系统进行修复，那么如果它的主目录就在损坏的文件系统上，那么甚至连登陆都成了问题。

sbin

在启动过程中，**root**用户需要用到的一些程序就存放在这个文件夹中。普通用户不会用到该目录下的程序。

tmp

该目录为临时存储的文件夹。所有用户都有对该目录的读写权限。

usr

该目录在Linux系统中是个相当大的目录。除了上面提到的东西外，其它的内容一般都放在该文件夹下，包括程序、文档、内核源码及X Window系统等。一般情况下，我们会将软件安装到该文件夹下。

var

该目录用于存放系统日志、缓冲的数据及程序锁文件等。该目录的内容最常改变。

现在你应该有些印象了，系统有什么文件夹，什么样的文件夹中放什么东西。如果你想更详细了解文件系统的布局，可以参见**hier(7)**的man手册。下一节中，我们会讲解如何快速地查找文件，之后就不需要人工地查找了。

4.1.2 查找文件

现在我们知道了系统一些主要文件夹的功能，但这并不能真正帮我们找到一些特定的文件。是的，我们可以一个个文件夹地查找，但我们需要的是快速地查找。Slackware中有四个主要的文件搜索命令。

which

我们首先要介绍的就是**which(1)**命令。**which**一般用于快速定位某个程序的位置。它的作用是查找我们的环境变量**PATH**，并返回第一个搜索到的路径。例如：

```
% which bash
/bin/bash
```

从上例中我们可以看到，**bash**位于**/bin**目录中。对于搜索而言，**which**的功能有限，因为它只对**PATH**进行搜索。

whereis

接下来介绍命令**whereis(1)**。它与**which**类似，只是它除了搜索**PATH**外，还对man手册进行搜索，我们使用**whereis**对**bash**命令进行搜索时，它还返回如下结果：

```
% whereis bash
bash: /bin/bash /usr/bin/bash /usr/man/man1/bash.1.gz
```

`whereis`命令不仅告诉我们程序的实际位置，还告诉我们相应的文档的位置。但，一样的，该命令的功能有限。如果我们只想查找一个特定配置文件的位置呢？显然使用`which`或`whereis`命令都无法解决这个问题。

find

下面介绍`find(1)`命令，它是一个功能异常强大的命令。我们可以指定一定的搜索规则对文件系统进行搜索。例如可以使用文件名通配符、指定文件的创建或修改时间或都其它的一些规则。举个例子，如果我们想搜索系统中默认的`xinitrc`文件的位置，我们可以使用如下命令：

```
% find / -name xinitrc
/etc/X11/xinit/xinitrc
```

由于上述的`find`命令是对整个文件系统进行遍历搜索，所以要花上相当长的一段时间。并且，如果使用普通用户执行命令，在搜索那些只有`root`用户能看到的文件夹时，还会出现“权限不足”的错误，但最重要的是它的确找到了我们想要的文件。尽管如此，要是能再快一点就好了……

slocate

和`find`一样，`slocate(1)`也是对整个文件系统进行搜索，但它搜索的并不是真正的文件系统，而是一个事先生成的数据库。该数据库默认在每天凌晨时自动更新。但对于一些个人用户，凌晨时电脑一般处于关机状态，所以，我们可以手动执行`updatedb(1)`命令来更新数据库（执行该命令需要`root`权限，使用`su`命令切换到`root`用户或使用`sudo`即可获得`root`权限）。下面我们举个例子：

```
% slocate xinitrc # 这里并不需要root权限
/etc/X11/xinit/xinitrc
/etc/X11/xinit/xinitrc.fluxbox
/etc/X11/xinit/xinitrc.fvwm2
/etc/X11/xinit/xinitrc.twm
/etc/X11/xinit/xinitrc.xfce
/etc/X11/xinit/xinitrc.kde
...
```

得到的结果很长，但是运行速度很快。使用以上这些命令，我们就能够快速地找到我们想查找的文件了。

4.1.3 /etc/rc.d文件夹

`/etc/rc.d`文件夹用于存放系统的初始化文件。与System V采用`init`脚本的方法不同，Slackware为它的初始化文件采用了BSD风格的布局。`init`脚本的风格在不借助专门为此设计

的软件时，配置起来是很困难的。对于BSD风格的初始化脚本，每个运行级别都有一个单独的rc文件，而System V中，每个运行级别都有自己的一个目录，每个目录下又包含多个init脚本，采用这种设计维护进行比较方便。

初始化文件可以分为很多类别。有系统启动、运行级别、网络初始化及System V 兼容等。对于每个类别，我们都会将其它东西归为另一个类别。(As per tradition, we'll lump everything else into another category.)

系统启动

除了内核之外，Slackware运行的第一个程序是**init(8)**。该程序读取文件/etc/inittab中的配置告诉自己如何启动系统。在进行指定的运行级别之前，**init**执行/etc/rc.d/rc.S脚本，为系统作好准备。**rc.S**文件的作用是启用虚拟内存，挂载文件系统，清理一些特定的日志文件夹，初始化热插拔设备，载入内核模块，配置PCMCIA设备，设置串口，并执行System V的启动脚本（如果有的话）。很显然，**rc.S**做了很多工作，下面是一些**rc.S**调用的脚本，它们都位于/etc/rc.d目录下：

rc.S

系统真正的初始化脚本。

rc.modules

该脚本用于载入内核模块。如网卡、PPP设备及其它的一些模块。如果该脚本找到rc.netdevice文件，rc.modules也会执行它。

rc.pcmcia

查找并配置系统中的PCMCIA设备。这对于笔记本用户可能最为有用，因为它们的调制解调器或网卡可能就是PCMCIA的¹。

rc.serial

执行适当的setserial命令对串口进行适当的配置。

rc.sysvinit

查找与运行级别相关的System V初始化脚本并运行。在下面会进行更详细的介绍。

运行级别初始化脚本

系统初始化结束后，**init**会继续作运行级别的初始化。所谓的运行级别是对系统将要运行的状态的描述。听起来很繁琐？好吧，运行级别就是告诉**init**，是支持多用户还是只支持单用户；要不要开启网络支持；是使用X Window还是使用**agetty(8)**来管理登陆。下面的文件定义了Slackware中的不同运行级别。

rc.0

关机（运行级别为0）。默认情况下，该文件是rc.6的一个软链接。

¹这里的描述与前面选择启动脚本的描述不符，笔者对PCMCIA没什么概念，弄懂了再改吧

rc.4

带有多用户支持启动（运行级别为4），并使用X11的KDM、GDM或XDM作为登陆管理器。

rc.6

重启系统（运行级别为6）。

rc.K

启动单用户模式（运行级别为1）。

rc.M

多用户模式（运行级别为2或3），使用传统的基于文本的登陆管理器，这也是Slackware默认的运行级别。

网络初始化

运行级别为2、3或4时都会开启网络支持。下面这些文件就是用于网络的初始化的：

rc.inet1

由netconfig命令生成，该文件用于配置实际使用的网络接口。

rc.inet2

在rc.inet1之后执行，并开启基本的网络服务。

rc.atalk

启动AppleTalk服务。

rc.http

开启Apache网络服务器。与其它脚本相同，只要向它传递stop、start或restart参数，就可以相应地停止、启动或重启Apache服务。

rc.news

启动新闻服务器。

System V兼容化

在Slackware 7.0时引入了System V初始化脚本的兼容措施。许多其它的Linux发行版都采用System V风格的初始化布局而不是BSD风格的。System V为每一个运行级别的初始化脚本准备了一个文件夹，而BSD风格则是为每个运行级别准备了一个初始化脚本文件。

脚本rc.sysvinit会搜索存放在/etc/rc.d文件夹下的System V初始化脚本文件，并执行相应运行级别的脚本。这个兼容性措施对一些安装System V初始化脚本的商业软件包而言是很重要的。

其它文件

下面介绍的脚本文件是其它的一些初始化脚本。它们一般会通过上面介绍的脚本的调用运行，所以我们只需要修改这些脚本的内容即可。

rc.gpm

启动文本终端下的鼠标服务。开启后就可以在文本终端下进行复制与粘贴。极少数情况下，gpm会引发在X Window下使用鼠标的一些问题。所以如果在使用X window时遇到一些鼠标问题，可以尝试去掉该脚本的执行权限并停止gpm服务器。

rc.font

为文本终端加载自定义字体。（在安装过程中可进行设置。）

rc.local

包含你系统独有的一些初始化命令。在安装后该文件是空的，因为它是为系统管理员准备的。所有的初始化步骤结束后，系统会执行该脚本。

要启用某个脚本，只需要使用chmod命令为其添加执行权限。相应的，要停用某个脚本，也只需要去除相应脚本的执行权限即可。关于chmod 命令的更多信息，请参见第//TODO:Section 9.2//节。

4.2 选择内核

内核在一个操作系统中起着至关重要的作用，它提供了对硬件的访问、对进程的控制以及整个系统的控制。内核中包含了对硬件设备的支持，所以在安装的时候选择合适的内核是很重要的。

Slackware提供了不止一打的提前编译好的内核可供选择，每个内核都包含一个标准的驱动集合和一些特定的驱动。你可以选择直接使用提前编译的内核或从源码编译自己的内核。不管使用哪种方法，都要确保内核中包含系统所需的所有硬件支持。

4.2.1 Slackware CD-ROM上的/kernels文件夹

提前编译的内核可以在Slackware的CD或DVD上或FTP站点上的Slackware文件夹下的/kernels文件夹中找到。新的版本发布后，其中可用的内核也会发生改变，而该文件夹下的文档则是权威的。/kernels文件夹为每个内核准备了一个子文件夹，每个子文件夹的名字都与它们对应的启动盘名相同。在每个子文件夹中，你会找到以下文件：

文件名	作用
System.map	内核的系统映射文件
bzImage	内核本体
config	内核源码配置文件

表 4.1: kernel文件夹中的文件

要使用一个内核，将System.map及config文件拷贝到/boot目录中，并将内核镜像²复制到/boot文件夹下，重命名为vmlinuz，之后，运行/sbin/lilo(8)命令为新的内核安装LILO，重启即可。这就是安装一个新内核的步骤。

接下来的内容是过时的。以.i结尾的内核是为IDE设备准备的，这些内核不支持对SCSI。以.s结尾的内核为SCSI内核，支持IDE，同时支持SCSI。

4.2.2 从源码编译内核

新手们常问的一个问题是“我需要自己编译一个内核吗？”。答案是也也许吧。有一些情况下是需要为自己的系统编译内核的。多数用户只需使用编译好的内核，外加一些可加载的内核模块就可以构建一个可用的系统了。如果你想使用一个Slackware当前不提供的新的内核版本，或者你想对内核添加补丁以获取对一些当前内核不支持的设备的支持，那么你可能需要为自己的系统重新编译内核。如，一些有SMP功能的系统就会考虑编译一个带SMP支持的内核。许多用户还会自己编译内核来让机器跑得更快些。要是想针对某些特定的处理器做优化，编译内核也是有帮助的。

编译自己的内核其实并不难。首先是确保你的机器上安装了内核的源码。即安装是切记选择安装K系列下的软件包。当然，也要确保安装了D系列的软件包，我们需要的是D系列中的C编译器、GNU make、及GNU binutils。一般而言，如果做的工作与开发有关，那么把D系列的软件包全装上会是个不错的选择。你也可以从<http://www.kernel.org/mirrors>网站下载最新的内核源码。

4.2.3 2.4.x版本的内核编译

```
% su -  
Password:  
# cd /usr/src/linux
```

第一个步骤是让内核处于默认的状态。使用下面的命令（注意，该命令会删除.config文件，且没有任何提示，需要的话，可以先进行备份）：

```
# make mrproper
```

接下来，我们就可以根据自己的系统来配置内核了。当前的内核提供三种方式。第一种基于文本的问答系统，它会问用户一系列问题并根据用户的回答来创建配置文件。这个方法的一个问题就是如果你搞砸了，就得重新来过。多数人选择的是第二种方式——基于菜单的方式。第三种是基于X的配置工具。选择好你喜欢的方式，通过执行下面对应的命令即可：

```
# make config      (基于文本的问答系统)  
# make menuconfig  (基于文本的菜单驱动系统)  
# make xconfig     (基于X的版本，请先确保自己在使用X系统)
```

²kernel image, image不知道应该怎么翻才好。

Linux Kernel v2.2.16 Configuration

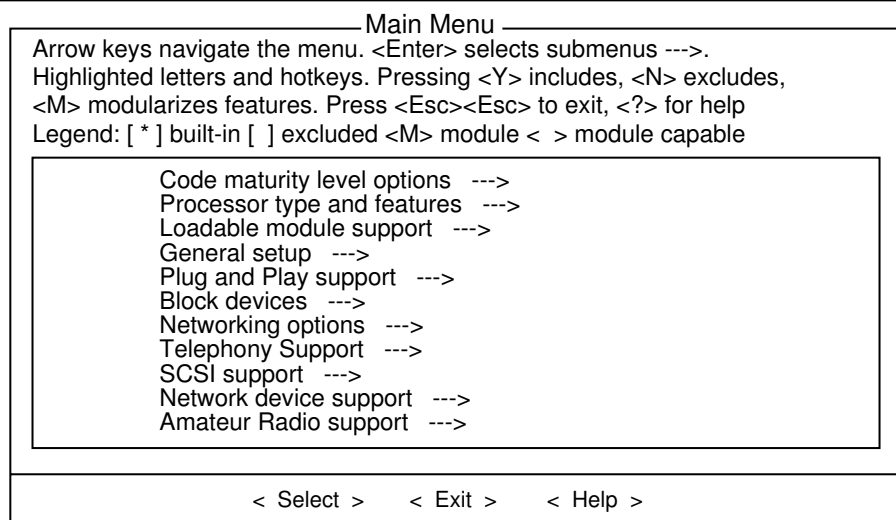


图 4.1: 内核配置菜单

对于新用户而言，`menuconfig`应该是最容易使用的方式。其中提供了帮助，解释了内核各部分的功能。配置完成后退出配置程序。它会创建一个必要的配置文件。之后我们就可以准备用来编译的源码树了：

```
# make dep
# make clean
```

下个步骤就是编译内核。首先尝试如下命令：

```
# make bzImage
```

根据你的CPU速度，这个命令会跑上一会。编译的过程中会显示一些编译器的信息。编译完成之后，我们还需要编译之前配置时标记为模块的部分。

```
# make modules
```

接下来就是安装刚编译好的内核了。要在Slackware系统上安装内核，可以使用如下的命令：

```
# mv /boot/vmlinuz /boot/vmlinuz.old
# cat arch/i386/boot/bzImage > /boot/vmlinuz
# mv /boot/System.map /boot/System.map.old
# cp System.map /boot/System.map
# make modules_install
```

之后是重新配置LILO加载新的内核。首先修改`/etc/lilo.conf`文件并为老的内核添加相应的启动项。完成后运行`/sbin/lilo`来安装新的启动块。最后重启并以新的内核启动即可。

4.2.4 2.6.x版本的内核编译

2.6版本的内核编译与2.4或2.2的内核只有小部分的差别，但在深究之前了解其中的差异还是很有必要的。编译2.6内核时，不再需要运行`make dep`及`make clean`了，并且在2.6系列内核中，编译时不再像之前的版本一样显示很详细的信息。结果就是编译过程更容易理解了，当然，这样做也有一些短处。所以如果在编译过程中出了什么错误，我们强烈建议你选择显示详细信息。只要在编译时加上`V=1`选项即可。显示更为详细的信息有助于内核开发人员记录更多信息，在别的geek帮助你时也能更有效率。

```
# make bzImage V=1
```

4.2.5 使用内核模块

内核模块可以认为是设备驱动的一个别名，我们可以在内核运行时对其动态加载与卸载。通过内核模块，我们在不选择另一个内核或自己编译内核的情况下就能添加对新硬件的支持。

模块可以在任何时刻进行加载或卸载。这使得系统管理员在更新一些特定驱动时变得很容易。编译新的模块-`i` 卸载旧的模块-`i` 加载新的模块，连重启系统都不需要。

所有的模块都存储在`/lib/modules/kernel-version`目录下（`kernel-version`处根据你的系统而定）。它们可以通过`rc.modules`文件在启动时就加载。这个文件的注释很很，并为主要的硬件提供了配置的实例。使用`lsmod(1)`可以显示当前活动的模块：

```
# lsmod
Module      Size  Used by
parport_pc  7220    0
parport     7844    0  [parport_pc]
```

就上面的例子而言，可以看到我只加载了并口的模块。要移除一个模块，使用`rmmod(1)`命令。加载模块可以使用`modprobe(1)`或`insmod(1)`命令。其中，`modprob`更为安全，因为它解决了模块间的依赖关系。

多数用户从来没有手工加载过模块。他们只使用内核自动加载器（the kernel autoloader）来管理模块。默认情况下，Slackware的内核中包括了`kmod`。`kmod`是一个内核选项，它使内核在需要时能自动加载所需的模块。想了解更多关于`kmod`及如何配置的内核，请参见`/usr/src/linux/Documentation/kmod.txt`³。再次，前提是你已经安装了内核源码软件包，或者可以从<http://kernel.org>网站上下载内核源码。

我们可以在上面涉及到的命令的相应man手册中找到更多信息，当然，还有`rc.modules`文件的相关内容。

³Slackware 13.37、内核2.6.37中已经找不到该文件。

第五章 网络配置

5.1 NFS

附录 A GNU通用公共许可证

待完成