

深入研究嵌入式操作系统的绝佳教材

一步步 写 嵌入式操作系统

——ARM编程的方法与实践

李无言 著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

一步步写 嵌入式操作系统

—ARM编程的方法与实践

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书是一本介绍怎样去实际编写一款嵌入式操作系统的书,所涉及的内容包括操作系统基本原理以及这些原理在嵌入式平台中的实现方法。全书共分九章,从最基本的嵌入式编程方法开始,逐渐深入到中断管理、内存管理、设备管理、文件系统管理以及进程管理等操作系统核心部分,为读者系统地呈现了一个操作系统的全貌。

另外,本书遵循理论联系实际的基本原则,在阐述基本原理的同时,还给出非常详尽的示例代码,以及对这些代码的讲解。读者研读这些代码,不但可以进一步巩固对操作系统理论知识的理解,更可以以此为基础,去实现一个属于自己的嵌入式操作系统。也许写操作系统是很多人的梦想,或者对有些人来说根本不敢想象,希望通过学习本书,您可以超越您的想象,实现您的梦想!

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

一步步写嵌入式操作系统:ARM 编程的方法与实践 / 李无言著. —北京:电子工业出版社,2011.1
ISBN 978-7-121-12240-8

I. ①—… II. ①李… III. ①微处理器, ARM—系统设计 IV. ①TP332

中国版本图书馆 CIP 数据核字(2010)第 218180 号

策划编辑:袁金敏

责任编辑:贾 莉

印 刷:北京机工印刷厂

装 订:三河市胜利装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱

邮编 100036

开 本:720×1000 1/16

印张:17.25

字数:274 千字

印 次:2011 年 1 月第 1 次印刷

印 数:4000 册

定价:39.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系电话:(010) 68279077; 邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

本书作者多年的嵌入式领域研发、管理和教学经验，促成了本书的诞生。

本书的目的

操作系统是一个古老的话题，它的出现和发展，对于计算机技术来说，意义非凡。这种发展不仅仅体现在传统的计算机中，在强调精简、高效率和实时性的嵌入式领域，操作系统也发挥着不可替代的作用。

熟悉嵌入式技术的读者朋友多少都能列举出一些典型的嵌入式操作系统，如ECOS、UCOS、Linux、VxWorks，等等，这些操作系统凭借自身的优势，在嵌入式领域的各种应用中安营扎寨，各占据了一席之地，国内基于这些嵌入式操作系统的书籍也是琳琅满目、层出不穷。

本书写作的一个根本目的是能够体现出嵌入式操作系统的基本原理和结构特点，于是，本书力求从嵌入式操作系统的实现方法这一角度出发，去讲解嵌入式系统的基本原理，为大家展示一个嵌入式操作系统的全貌，不拘泥于某款具体的操作系统，不局限在操作系统上的应用程序这一范畴。

一次跟Android领军人物高焕堂先生聊天时，他的一番话让我感触颇深。他说，国外先进的开发工具、平台和操作系统就好比是武器，而中国人喜欢拿着别人给的先进武器去打仗（做应用层开发），一旦有一天我们跟外国人打起来，人家拿走我们的武器，我们就真的是一筹莫展了。

这句话很有道理，中国计算机技术整体水平的提高需要以大量自主研发的开发工具、平台架构以及操作系统为基础。不过，目前我们离这样的一个目标还相去甚远。

本书强调实践，力求能够帮助读者编写出属于自己的嵌入式操作系统。如果读者以本书为基础（或者哪怕从中得到了一丝灵感）开发出一些优秀的嵌入式操作系统，那将会是非常令人高兴的事情！

本书的特点

目前，市面上与操作系统理论相关的书有很多，与这些书相比，本书特点十分鲜明。

第一，本书的内容立足于嵌入式技术，以目前最流行的ARM体系结构为基础，为您展示出嵌入式环境下操作系统的基本原理和实现方法。这是一本学习嵌入式技术，尤其是系统级技术的首选教材。

第二，本书是以实践的方式讲述全书内容的，重视理论联系实际。操作系统涉及的每一个角落，如进程、内存、中断、文件系统、驱动程序，都有若干段代码供读者实践。实际上，读者只需要将书中的代码拼接起来，就可以构成一个结构完整的操作系统内核。

第三，涉猎广泛。为了让读者能够全方位地理解操作系统的理论和实现方法，书中涉及了高级C语言编程、汇编语言、算法、ARM体系结构等诸多领域，分析研究了包括freeRTOS、uCOSII、u-boot、Linux、Minix等在内的操作系统和引导程序的源代码。这些必备的知识 and 概念都将成为您深入研究任何一款操作系统的绝佳入门内容。

第四，本书语言通俗易懂。书中在描述各种操作系统概念或原理时，力求使用通俗的语言浅显地说明问题，这与一些学术性质的操作系统读物有明显的不同。

谁适合读这本书

本书是写给那些想了解操作系统原理的人，也适合那些想要学习ARM技术的开发者。当然，对于那些致力于开发属于自己的嵌入式操作系统的读者们来说，本书是一部绝佳的入门指南。

当然，我们也希望本书的读者至少具备一些基本的C语言编程基础。除了C语言之外，如果您对微机原理之类的知识也略知一二，那么在阅读本书

的时候，将会觉得更加轻松。本书不需要读者了解嵌入式技术，因为这些知识在书中都会涉及，但如果您也知道一些关于ARM的事，那么无论是重写书中的代码还是深入理解书中所讲述的操作系统原理，都会水到渠成。

学习本书的方法

本书在知识的宽度和篇幅上做了平衡。希望展示给读者一个操作系统的全貌，保证读者能够自己动手完成操作系统的编写，而这些都需要我们在开发的过程中不断学习书中所涉猎的知识点。毫无疑问，这些内容是相当庞大的。

为了避免在书中罗列各种技术手册的细节，在本书的编写过程中，我们遵循了只对用到的知识深入讲解的原则。这样做的弊端是，读者会对很多知识点有深入的理解，但却不全面。

因此，我们推荐阅读本书的最佳方法是，以本书为主线，以其他本书中未提到的相关知识、参考资料为辅助。当然，如果读者的确是时间有限，那么单凭书中所涉猎的知识，也足以完成读者学习和实践的过程了。

另外，这里还想强调的是，一定要多多实践，实践将成为您快速掌握一门新技术的不二选择。

为了方便读者获取书中涉及的相关资料 and 工具，我们开设了一个网站，读者朋友可以去访问www.leeos.org以获取帮助。

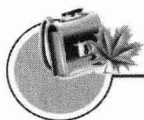
最后，希望所有的读者在读完本书之后，都能够有所收获，希望每位读者都能编写出属于自己的嵌入式操作系统！

目 录

第 1 章 搭建工作环境	1
1.1 选择合适的开发环境	1
1.1.1 准备 Cygwin 开发环境	2
1.1.2 使用 Linux 开发环境	7
1.2 开发工具的使用	8
1.2.1 编译器的选择和安装	8
1.2.2 编辑器的选择和使用	10
1.3 虚拟硬件的安装和使用	13
1.3.1 SkyEye 的安装	15
1.3.2 SkyEye 的使用	15
1.4 总结	17
第 2 章 基础知识	18
2.1 使用 C 语言写第一段程序	18
2.2 用脚本链接目标文件	22
2.3 用汇编语言编写程序	25
2.4 汇编和 C 的混合编程	29
2.4.1 过程调用标准	30
2.4.2 混合编程的例子	31
2.5 Makefile	33
2.6 总结	35
第 3 章 操作系统的启动	36
3.1 启动流程	36
3.1.1 ARM 的启动过程	38
3.1.2 ARM 操作系统解读	39
3.1.3 正式开始写操作系统	46
3.1.4 让启动代码运行起来	52
3.2 MMU	56
3.2.1 页表	58
3.2.2 页权限	64
3.2.3 cache 和 write buffer	66

3.2.4	激活 MMU	69
3.3	GCC 内联汇编	71
3.4	总结	78
第 4 章	打印函数	79
4.1	打印函数实例	80
4.1.1	变参函数是如何工作的	81
4.1.2	亲自实现一个可变参数函数	83
4.1.3	实现打印函数中的格式转换	87
4.2	实现自己的打印函数	91
4.3	总结	99
第 5 章	中断处理	100
5.1	ARM 的中断	101
5.1.1	统一的异常和中断处理	102
5.1.2	独立的中断处理	114
5.2	简单的中断处理实例	118
5.2.1	解决异常向量表的问题	118
5.2.2	简单的中断处理代码	120
5.2.3	S3C2410 中的定时器	122
5.2.4	让中断处理程序运行起来	124
5.3	复杂的中断处理实例	126
5.3.1	提高中断的效率	126
5.3.2	中断嵌套的实现	130
5.4	更优秀的中断嵌套方法	134
5.5	总结	139
第 6 章	动态内存管理	140
6.1	伙伴算法	141
6.1.1	伙伴算法的原理	143
6.1.2	伙伴算法的实现	146
6.2	slab	169
6.2.1	使用 slab 的前提条件	170
6.2.2	slab 的组成	171
6.2.3	通过 slab 进行内存分配	176
6.2.4	内存空间的释放	177
6.2.5	slab 的销毁	178
6.3	kmalloc 函数	179

6.4 总结	183
第 7 章 框架	184
7.1 驱动程序框架	184
7.1.1 基于存储设备的实例	186
7.1.2 运行存储设备实例	192
7.2 文件系统框架	194
7.2.1 文件系统的原理	195
7.2.2 文件系统框架的实现	197
7.2.3 romfs 文件系统类型	200
7.2.4 实现 romfs 文件系统	204
7.2.5 让代码运行起来	210
7.3 总结	215
第 8 章 运行用户程序	217
8.1 二进制程序的运行方法	218
8.2 可执行文件格式	222
8.2.1 ELF 格式的组成结构	223
8.2.2 操作 ELF 格式文件的方法	226
8.2.3 运行 ELF 格式的应用程序	230
8.3 系统调用	232
8.3.1 用户和内核的运行空间	232
8.3.2 实现一个系统调用	235
8.3.3 运行系统调用程序	244
8.4 总结	246
第 9 章 进程	247
9.1 进程的实现原理	247
9.2 进程的实现	252
9.2.1 改写中断处理程序	252
9.2.2 抽象调度函数	256
9.2.3 新进程的产生	258
9.2.4 多个进程同时运行	262
9.3 总结	265
结束语	266
参考资料	267



第 1 章

搭建工作环境

“工欲善其事，必先利其器”。在我们开始写操作系统代码之前，花些时间来学习工具的使用其实是非常必要的。从过程上看，操作系统的开发与普通应用程序的开发并没有太大的区别。正像开发应用程序那样，要开始一个操作系统的开发，首先也必须要解决开发环境搭建的问题。

搭建开发环境归根结底是要解决四个问题：

- 在什么样的系统环境下开发？
- 使用什么样的编辑工具？
- 怎样编译程序？
- 程序如何运行？

为了能够帮助读者学习本书的内容，在本章中我们将会围绕上述四个问题展开深入的讨论，以帮助读者顺利地编写出属于自己的嵌入式操作系统。

1.1 选择合适的开发环境

每一个计算机爱好者都有自己心仪的操作系统。有些开源软件和嵌入式技术的爱好者也许偏好 Linux，一些追求时尚和个性的朋友或许钟爱 MacOS，但对于绝大多数读者来说，Windows 操作系统应该还是最熟悉的。

本来使用什么样的操作系统，不应该成为限制开发的理由。但这里需要说明的是，本书的所有示例代码都是在一个叫做 Gentoo 的 Linux 发行版中开发的。书中所使用的绝大多数工具在 Linux 下都有原生的支持。因此如果有可能，还是建议大家使用 Linux 操作系统来学习和编写本书的示例。

然而，要求那些不熟悉 Linux 操作系统的开发者在短时间内学会使用 Linux 也并不现实。好在 Windows 操作系统在兼容性和应用程序多样性等方面是无人能敌的，于是在 Windows 下，我们同样也找到了一套解决方案来编译和运行本书的代码，它就是 Cygwin。

Cygwin 是一个在 Windows 平台上运行的 UNIX 模拟环境，是 Cygnus Solutions 公司开发的自由软件。它对于学习 UNIX/Linux 操作环境、从 UNIX 到 Windows 的应用程序移植，或者进行某些特殊的开发工作（尤其是使用 gnu 工具集在 Windows 上进行嵌入式系统开发）都非常有用。

下面我们就来介绍一下 Cygwin 环境的安装和使用方法。如果您是 Linux 用户，就可以跳过这部分，继续阅读下一小节的内容。

1.1.1 准备 Cygwin 开发环境

首先，我们需要从 www.Cygwin.org/Cygwin/setup.exe 处下载最新版的 Cygwin 安装程序。

接下来运行这个安装程序，如图 1-1 所示。

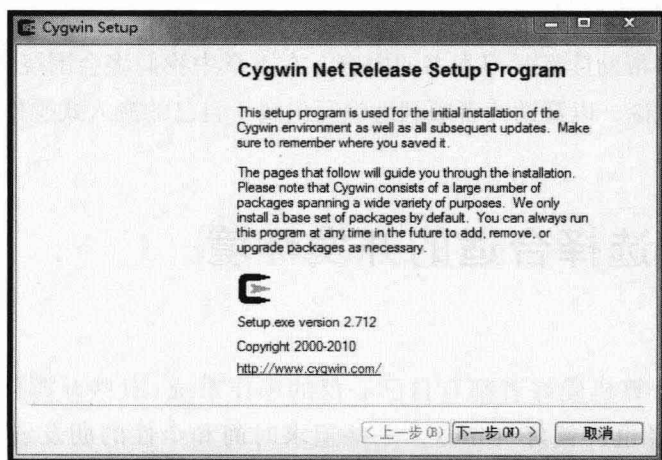


图 1-1 Cygwin 的安装

在图 1-1 的对话框中单击“下一步”按钮，进入安装类型选择页面，如图 1-2 所示。

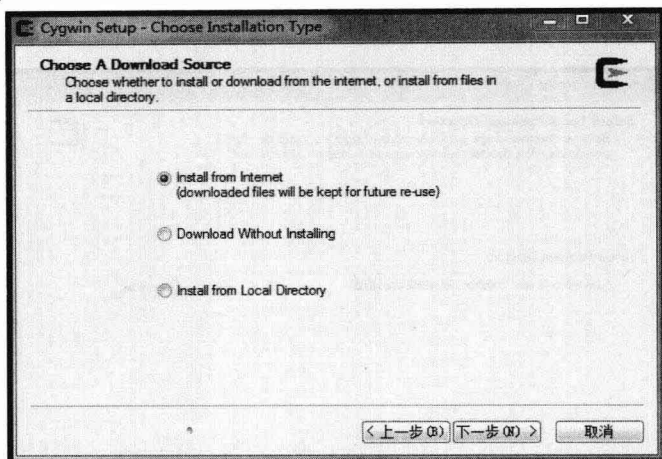


图 1-2 Cygwin 安装类型选择页面

此处如果没有特殊的要求，可以采用默认的“Install from Internet”单选项，通过网络安装 Cygwin 系统。当然，这要求读者的电脑能够上网才可以。

继续单击“下一步”按钮，进入安装路径选择页面，如图 1-3 所示。在这里，读者可以根据自己系统的情况选择某一安装路径。

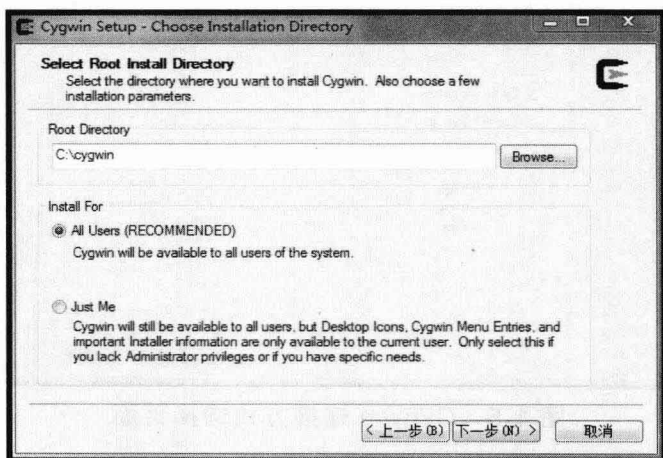


图 1-3 Cygwin 安装目录选择页面

再次单击“下一步”按钮，程序进入软件包下载目录选择页面。之前我

们已经选择了通过网络安装的方法来进行安装，那么在这个页面中，我们就需要指定一个本地文件夹，保存从网上自动下载的各种应用软件包，如图 1-4 所示。

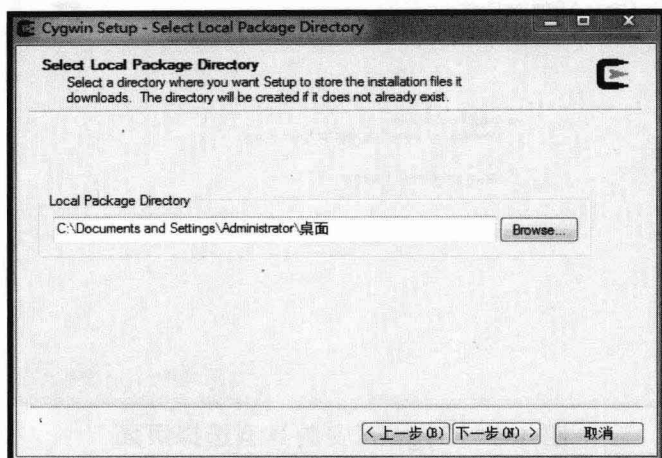


图 1-4 Cygwin 软件包下载目录选择页面

之后程序将打开连接方式选择页面，如果没有特殊要求，这一步选中默认的“Direct Connection”单选项即可，如图 1-5 所示。

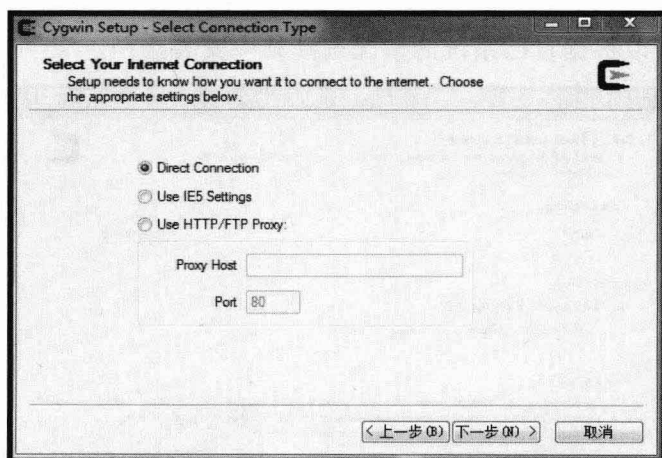


图 1-5 Cygwin 连接方式选择页面

单击“下一步”按钮，进入下载站点选择页面。通常我们会选择 <http://www.cygwin.cn/> 作为下载源，如图 1-6 所示，对于国内的用户来说，这个站点的连接速度最快。

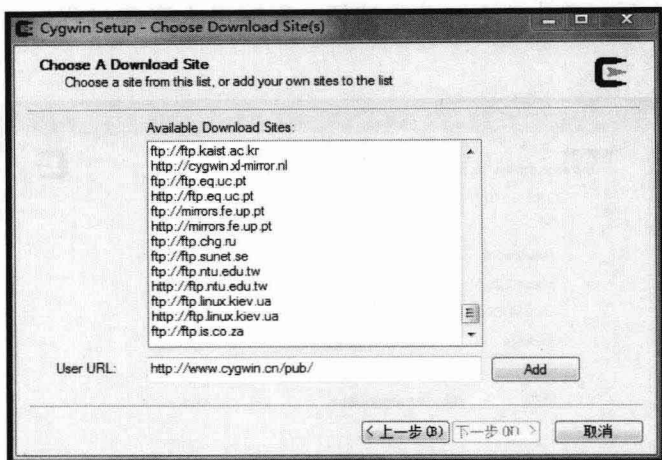


图 1-6 Cygwin 下载站点选择页面

如果是首次运行 Cygwin 安装程序，则需要在“User URL”文本框中添加“http://www.cygwin.cn/pub/”这一地址，单击“Add”按钮。在其他情况下，只需在“Available download Sites:”列表框中选择“http://www.cygwin.cn”站点即可。下载源选择完成后，单击“下一步”按钮，进入软件包选择页面，如图 1-7 所示。

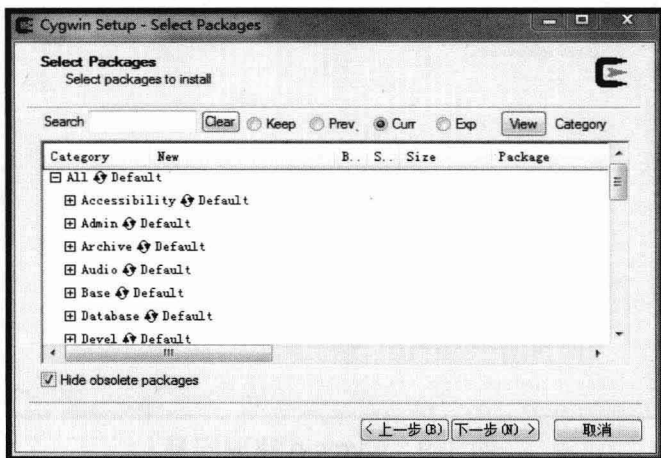


图 1-7 Cygwin 软件包选择页面

在该页面中，我们可以选择需要安装的软件包。由于 Cygwin 默认安装的软件包不能完全满足我们自己的操作系统开发，因此需要至少选择两个额外的软件包进行安装。单击“Devel”类前面的“+”，将该分类展开，从中选择“make”和“gcc4”两组软件包，而其他软件包按照默认方式处理即

可，完成操作后，单击“下一步”按钮，程序进入安装过程，如图 1-8 所示，显示安装进度。

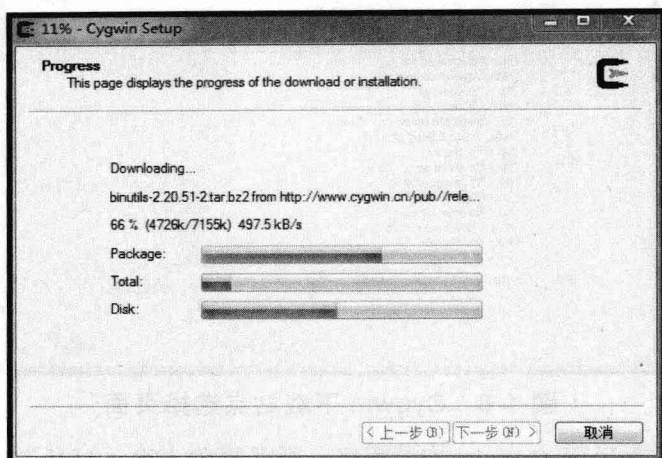


图 1-8 Cygwin 安装过程

片刻之后，Cygwin 的安装就完成了，如图 1-9 所示。

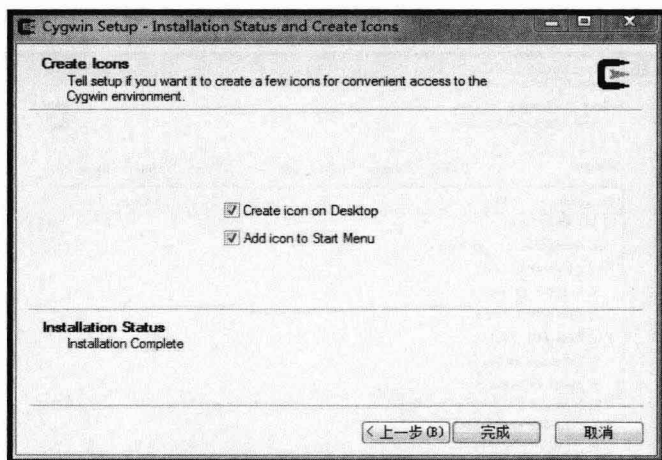


图 1-9 Cygwin 完成安装

单击“完成”按钮，退出 Cygwin 的安装过程。

Cygwin 的使用方法非常简单，只需要单击桌面上的 Cygwin 图标，就会弹出一个类似于 Windows 命令行的页面，如图 1-10 所示。



图 1-10 Cygwin Shell 页面

简单地讲，如图 1-10 所示的操作页面实际上是在 Windows 操作系统中模拟出来的类 UNIX 系统中的 Shell 页面，我们可以输入一些 UNIX 下常用的命令，如 `ls`、`cd` 等，这些命令的使用方法和运行结果与 UNIX 系统是一致的。对于那些使用 Windows 操作系统的朋友们来说，属于我们自己的操作系统也将在这样一个环境中诞生！

1.1.2 使用 Linux 开发环境

如果读者朋友已经准备使用 Linux 来开发属于自己的操作系统，又没有把握在电脑中直接安装某个 Linux 发行版，那么一个折中的办法就是使用虚拟机，就好像我们又多了一台电脑一样。

目前流行的虚拟机软件五花八门，各有优劣。使用什么样的虚拟机安装 Linux 系统，其实并没有太大的差别。在这些虚拟机中安装 Linux 操作系统的大致步骤如下。

(1) 选择读者喜欢的 Linux 发行版，主流的如 `ubuntu`、`redhat`，个性的如 `gentoo`、`archLinux`，极端的如 `LFS`。

(2) 从各发行版的网站下载安装光盘镜像，这些光盘镜像有的可能很小，需要通过网络安装的方式进行，有的也许很大，直接一张光盘就可以完

成一个基本系统的安装。

(3) 将下载下来的光盘镜像插入到虚拟机中, 根据各 Linux 发行版的相关文档或软件提示进行 Linux 的安装。

出于篇幅的原因, 本书不对虚拟机下安装 Linux 发行版的详细步骤进行介绍。因为如果介绍详细步骤, 一连串的关于 Linux 原理、基本操作方法等问题都将被牵扯出来, 本书很有可能就会变成一本 Linux 入门级书籍。为避免喧宾夺主, 我们建议只了解 Windows 操作系统的朋友使用 Cygwin 进行开发。对于那些了解 Linux 的朋友们来说, 对于 Linux 如何安装自然会非常清楚, 因此也无须多言。

1.2 开发工具的使用

自己动手写操作系统的第二个重要工作是选择合适的编译器和编辑器。

1.2.1 编译器的选择和安装

由于我们的操作系统需要运行在 ARM 体系结构中。因此, 在琳琅满目的 ARM 编译器中选择出一款, 用来编译操作系统源程序, 就成为了必须先要解决的问题。于是我们选择了 GCC。

GCC 的大名, 相信读者应该都会如雷贯耳。简单地说, GCC 就是由 GNU 基金会组织开发的一套开源的编译器。根据相关许可, 我们可以自由地下载和使用这套编译器, 可以随意地修改和重新发布它的源代码, 整个过程不需要付一分钱或承担任何责任。这就是我们选择 GCC 作为开发工具的一个最主要的原因。

GCC 的免费获得和使用并不等于质量低劣或功能欠缺。它的功能不逊于任何一款商用 ARM 编译器。这也是我们选择 GCC 作为操作系统开发工具的第二个原因。

接下来我们就来介绍一下 GCC 编译器的安装过程。

GCC 是源码开放的, 所以理论上, 我们完全可以去网站下载 GCC 源代码, 然后在本地进行编译, 生成一个能够编译 ARM 体系结构下程序