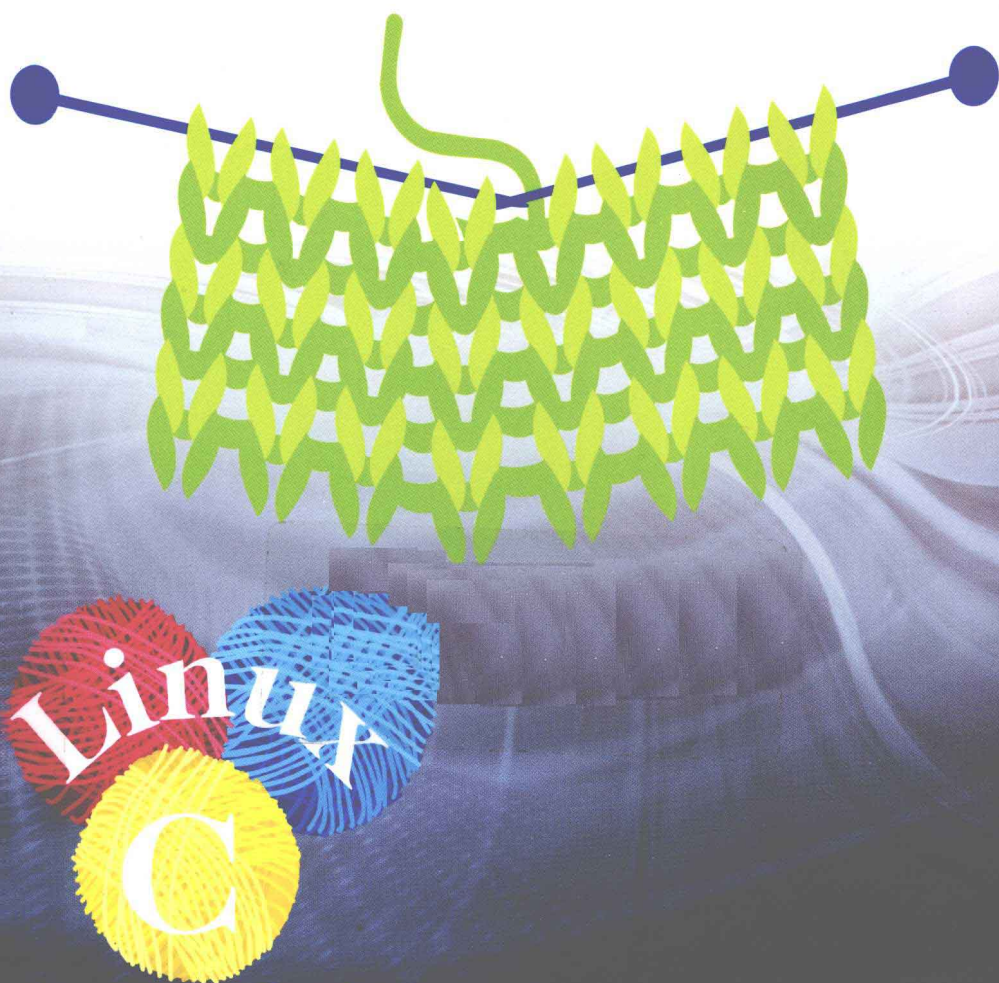


Linux下C语言 应用编程

杨 铸 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

内 容 简 介

本书深入浅出地讲解了 Linux 下 C 应用程序开发所需的基本工具、知识和技巧,主要包括:开发工具的使用和技巧、文件 I/O 编程、多进程编程、进程间通信、多线程编程、网络编程等内容。可以让读者轻松、快速入门 Linux 下非图形界面应用编程。

本书适合大学本、专科学生,培训机构学生,自学人员以及研究生学习 Linux 下 C 语言应用编程;同时也适合想转入到 Linux 下进行开发的 Windows 应用软件开发工程师。

图书在版编目(CIP)数据

Linux 下 C 语言应用编程 / 杨铸编著. — 北京:北京航空航天大学出版社, 2012. 9

ISBN 978-7-5124-0905-7

I. ①L… II. ①杨… III. ①

Linux 操作系统—程序设计②C 语言—程序设计 IV.

①TP316.89②TP312

中国版本图书馆 CIP 数据核字(2012)第 186148 号

版权所有,侵权必究。

Linux 下 C 语言应用编程

杨 铸 编著

责任编辑 卫晓娜

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@gmail.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:17.25 字数:378 千字

2012 年 9 月第 1 版 2012 年 9 月第 1 次印刷 印数:4 000 册

ISBN 978-7-5124-0905-7 定价:36.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

前言

创作动机

其一, Linux 操作系统以它的开源与包容, 已经在当今计算机软件领域占据非常重要的一席, 使用的人越来越多, 而且越来越多的程序员也从 Windows 下的应用程序开发转入 Linux 下的应用程序开发。可以预见, 在不远的将来, Linux 下的应用程序开发将得到长足的发展, 也将很有前途。虽然都是应用程序开发, 但与 Windows 下的应用程序开发相比较; Linux 下的应用程序开发, 无论是在开发工具还是开发所需知识上, 都有很多的不同, 因此这就需要转入 Linux 下开发的程序员重新学习一些知识, 其中最主要的就是开发工具的使用和 POSIX API 的使用。如何使得这些有应用程序开发经验的程序员以最短的时间掌握 Linux 下应用程序开发的精髓, 帮助他们在转型过程中少走弯路, 就是本书创作的初衷之一。也许有人会说, Linux 下非图形界面应用程序开发, 已经有非常经典的名家之作, 例如: 本书的参考文献[1]和[2], 本书岂能与它们比肩。的确, 这些名家之作, 知识之全面、内容之经典绝非本书作者所能超越, 事实上本书的很多素材皆来源于这些名家之作, 作者启蒙之初也是靠拜读这些名作。但正是因为如此, 作者才有切肤之感。名作兼容并蓄, 内容全面, 因此书的厚度非常可观, 如果你有足够的时间, 慢慢阅读、品尝将非常惬意, 无奈当今 IT 界的飞速发展, 已不容许程序员如此惬意, 因此本书的目的之一就是为程序员准备速成之法。因此, 在此也郑重提醒读者, 在阅读本书快速入门之后, 可能的话, 尽量抽出一些时间阅读名家之作。

其二, 作者在为本科生讲授“C 语言程序设计”这门课程时, 深深感到由于学生基础、课时以及国内教育体制与习惯所限, 大多数情况下, 这门课程基本就变成了仅仅讲解 C 语言基本语法的课程, 学生在学习过程中写的都是些三、四十行的用来简单练习 C 语言语法的程序, 更谈不上进行大量的程序调试。这样的结果是, 学生仅仅学习了 C 语言表面语法的東西, 基本就没有在实战中去理解 C 语言的机会。要实战 C 语言, 需要一个媒介, 就目前情况而言, 选择有二: 在 Windows 下用 C 语言做实战开发, 或者在 Linux 下用 C 语言做实战开发。作者个人愚见, 当选后者, 将 C 语言的学习和 Linux 下的无图形界面 C 应用程序开发结合起来, 这样将会大大提高大学本科学生学习 C 语言的兴趣和效果。

前言

其三,写完《深入浅出嵌入式底层软件开发》后,渐渐感到自下往上理解计算机软件系统的整体架构虽然是较为可行的一种方法,但由于底层软件不易调试、不直观,且涉及的知识又是当前上层程序开发人员较少接触到的,因此理解起来有很大困难。当从一个方向不能实现突破的话,那么从两个方向进行夹逼,也许就是最明智的选择。其实,上层开发的知识与底层开发的知识是可以相互补充和相互印证的,很好地掌握任何一方的知识都有助于理解另一方的知识,文件描述符的本质就是一个最好的例子。当程序员通过上层开发知道了应用程序需要文件描述符作为访问硬盘文件的接口,通过底层开发知道了 Linux 字符设备驱动需要用到内核的文件表,两边一夹逼,就会恍然大悟,文件描述符的本质其实就是内核文件描述符表的数组下标,此时将会理解 Linux 下字符设备驱动与应用程序是如何通过文件描述符关联的,进而将激发程序员对 Linux 下的 VFS 文件系统所扮演角色进行一探究竟的冲动,也许当解决这个问题后,也就打通了任督二脉,上下贯通了。

本书内容及阅读建议

第 1 章讲解 Linux 下 C 应用程序开发环境的搭建,以及主要开发工具(编辑器 vim、编译器 gcc、调试器 gdb、工程管理工具 make)的入门级使用。通过本章的学习,将使读者可以迅速入门 Linux 下 C 应用程序开发。特别说明,由于本章的目的是让读者以最快的速度进入开发的状态,所以这些工具的使用都是最初级和最基本的,随着读者开发的深入,会发现这点知识实在是不敷使用,此时读者可以查阅本书第 7 章相关内容。

第 2 章讲解文件 I/O 编程。文件 I/O 的 API 是最基本的 POSIX API,是 Linux 下无图形界面应用程序开发最基本的知识,虽然简单,但意义重大。特别提醒读者,在基本学完了第 2 章的知识点后,务必结合随书所配程序代码 myls.c 阅读并理解第 2.6 节的内容,学习 myls.c 将有助于读者将散落的知识串联起来,形成具有实战意义的程序,作者相信它也能激发读者继续学习的兴趣。

第 3 章讲解多进程编程的基本知识,核心是 fork、exec、wait 三大系统调用。学习起来可能会有些难度,如果读者在学习过程中碰到了麻烦的话,建议先补一补操作系统进程并发运行的基本知识后,再阅读本章。不过不需要太花时间深入了解进程并发运行,只要能有个基本概念和理解,阅读本章就应该足够了。与第 2 章的 myls.c 一样,本章的 shellv2.c 是串接本章知识点的灵魂程序,务请读者认真阅读理解。

第 4 章讲解传统的进程间通信的几种方法——信号、管道、共享内存、信号量、消息队列。读者应该掌握每种方法,不过就作者个人的观点而言,最重要、最常用的当属信号和管道。

第 5 章讲解多线程编程的基础知识。线程编程是一个很大的 topic,足以写整整一本书,不过本书的目的是快速入门,因此仅仅介绍了多线程编程最基本的知识和

API,如果读者想深入学习多线程编程的话,建议专门阅读有关多线程编程的经典之作。

第6章讲解网络编程,也就是传说中的 socket 编程,这部分也许是本书中最能引起初学者兴趣的部分,因为它最为实用。本部分主要介绍了 TCP socket 编程,简单介绍了 UDP socket 编程。这里要特别予以说明的是,本章只是介绍了 Linux 下 socket 编程的皮毛(虽然它们的确是常用的部分),如果读者想深入学习 socket 编程的话,那么本书参考文献[2]就是必读的经典之作。

第7章相对第1章讲解了开发工具(包括:编辑器 vim、编译器 gcc、调试器 gdb、工程管理工具 autoconf、程序库的制作)更多常用的知识和技巧。初学者不必急于阅读,等在实际开发中感觉到需要使用开发工具的一些功能,而第1章又没有学习到的话,可以查阅第7章。

感谢

本书由杨铸负责编写并统编全部书稿,李奎编写了本书部分章节,对他踏实刻苦的钻研精神和认真负责的敬业精神,在此表示深深的谢意。

感谢 IFL 嵌入式小组的唐攀,还有成宝宗,不辞辛劳地提供了本书所用到的部分素材。

感谢我的父母,是你们从小对我朴实无华的谆谆教导,在我心灵的深处种下了要勤奋学习、要努力工作、要懂得感恩的火种,你们给了我强大的精神鼓励和支持,使得本书得以顺利完成。

感谢北京航空航天大学出版社胡晓柏主任对本书的支持和关怀,正是他耐心的鼓励和支持,才使得本书在最短的时间内与读者见面。

感谢甘洪江、蒋军、蒋跃胜、彭川、汤世宇、唐保青、喻甫军、赵冬、党旭和李欢,对本书的认真校对,并提出修改意见。

感谢丁晓峰、葛红艳、关东升、关杰、何桂忠、何凯霖、胡德昆、黄曼绮、江军、刘鹏、刘志强、柳斌、罗福强、罗佳、马伯骊、马锋、马林、孙夏玉、王柱、肖瑶、修宸、杨建光、杨剑、杨菊英、张云和对本书的写作和出版提供的帮助。

感谢中国操作系统旗舰企业——中标软件有限公司,你们的国产自主操作系统和办公软件产品,为我国信息化建设构建了安全基础平台。中标麒麟操作系统系列产品主要以操作系统技术为核心,重点打造自主可控、安全可靠等差异化特性,是国家重点项目——“核高基”项目所指定的操作系统,旗下产品包括:中标麒麟安全操作系统、中标麒麟桌面操作系统、中标麒麟通用服务器操作系统、中标麒麟高级服务器操作系统、中标麒麟安全云操作系统、中标麒麟安全邮件服务器、中标麒麟高可用集

前 言

群软件等核心产品。本书的案例代码正是运行在你们免费授权提供的中标麒麟桌面操作系统上,在此对你们表示感谢。

最后,要感谢伟大的互联网,本书不少的素材都来源于互联网。正是由于互联网的普及,使得各种技术资料的获得和技术知识的交流变得非常容易,从而使得本书的创作事半功倍。

另外,由于版权问题,请读者上网搜索自行下载书中涉及的部分软件和资料,也可访问作者博客:<http://user.qzone.qq.com/308337370>,作者提供相关内容的下载链接。书中的代码也可在北航出版社网站:www.buaapress.com.cn 的下载专区下载获得。

限于笔者水平有限,书中难免有遗漏和不足之处,恳请广大读者批评指正,联系方式是 E-mail:scyz@263.net,并开通了 QQ 技术讨论群:47753328。

作者 2012 年于
成都少城公园
重庆西永微电园
北京维亚大厦

目 录

第 1 章 Linux 下 C 语言编程环境	1
1.1 Linux 操作系统及相关开发工具的安装与使用	1
1.1.1 虚拟机软件 vmware 的安装	1
1.1.2 Linux 操作系统的安装	12
1.2 基本 Shell 命令的使用	25
1.2.1 常用命令	25
1.2.2 帮助命令	29
1.2.3 搜索命令	30
1.2.4 重定向与管道	32
1.2.5 文件类型与权限	32
1.3 编辑器 vim 的使用	34
1.4 编译器 gcc 的使用	36
1.4.1 gcc 简介	36
1.4.2 使用 gcc	38
1.4.3 gcc 常用选项	40
1.5 调试器 gdb 的使用	41
1.5.1 gdb 概述	41
1.5.2 使用 gdb	45
1.6 make 与 Makefile 文件	46
1.6.1 Makefile 简介	46
1.6.2 Makefile 基本规则	46
1.6.3 使用带宏的 Makefile	48
1.6.4 Makefile 隐含规则	50
第 2 章 文件 I/O 编程	51
2.1 文件 I/O 介绍	51

目 录

2.2 文件描述符.....	51
2.3 基本 API	52
2.3.1 open	52
2.3.2 read	53
2.3.3 write	54
2.3.4 close	55
2.3.5 lseek	55
2.3.6 基本 API 综合使用	57
2.3.7 fcntl	58
2.3.8 ioctl	60
2.3.9 select	61
2.3.10 poll	64
2.4 stat 的使用	67
2.4.1 stat 的基本使用	67
2.4.2 文件类型的判定	69
2.4.3 权限设定的判定	71
2.4.4 获取主、次设备号	72
2.4.5 获取时间及时时间格式的转换	72
2.5 目录操作	73
2.5.1 打开目录	73
2.5.2 读取目录	74
2.5.3 关闭目录	74
2.5.4 使用范例	74
2.5.5 回绕目录 rewinddir	75
2.6 文件 I/O 编程综合案例分析——mysls.c	75
2.6.1 用于命令选项的宏定义	76
2.6.2 子函数功能	76
2.6.3 main 函数分析	77
2.6.4 子函数 show_link_content 的实现	77
2.6.5 子函数 show_file 的实现	78
2.6.6 子函数 show_dir 的实现	79
2.6.7 子函数 my_get_opt 的实现	80
2.6.8 子函数 get_file_list 的实现	80
第 3 章 多进程编程	81
3.1 进程基础知识	81

3.1.1 进程概念	81
3.1.2 进程的状态及其转换	82
3.1.3 进程标识	83
3.2 精解 Linux 下 C 进程内存布局	83
3.2.1 C 进程内存布局说明	83
3.2.2 C 进程内存布局验证	84
3.2.3 环境变量的获取与设置	85
3.3 进程控制天字第 1 号系统调用——fork	87
3.3.1 fork 的机制与特性	87
3.3.2 fork 使用实例分析	88
3.4 揭秘文件描述符的本质	89
3.4.1 文件描述符的本质是数组元素的下标	89
3.4.2 fork 对文件描述符的影响	91
3.4.3 标准输入、输出文件描述符与 dup2	92
3.5 父子进程同步的功臣——wait	94
3.5.1 wait 的作用	94
3.5.2 调用 wait 的实例	94
3.5.3 waitpid	97
3.6 进程控制地字第 1 号系统调用——exec	97
3.6.1 exec 的机制和用法	98
3.6.2 exec 的使用实例	98
3.6.3 exec 与 fork 合作	100
3.6.4 关于 exec 函数的 6 种形式	103
3.7 gdb 调试多进程程序的技巧	103
3.8 进程的消亡	104
3.8.1 启动例程与 main 函数	104
3.8.2 exit 函数与 _exit 函数	105
3.8.3 I/O 流与 I/O 库缓存	105
3.8.4 Exit handler	106
3.9 守护进程的编写	109
3.9.1 进程组、对话期与控制终端	109
3.9.2 编程规则与步骤	111
3.9.3 出错记录	112
第 4 章 进程间通信	115
4.1 进程间通信概述	115

目 录

4.2 进程间的传令兵——信号	116
4.2.1 信号概述	116
4.2.2 信号的捕获与处理	117
4.2.3 实例分析	118
4.2.4 使用 gdb 调试信号	121
4.2.5 pending signal	121
4.2.6 sigaction	125
4.3 进程间的动脉——管道	127
4.3.1 无名管道	127
4.3.2 有名(命名)管道	132
4.4 进程间的高速公路——共享内存	136
4.4.1 共享内存的原理	137
4.4.2 主要 API	137
4.4.3 利用共享内存进行进程间通信的实例	139
4.5 高速公路上的红绿灯——信号量	141
4.5.1 使用共享内存存在的同步问题	141
4.5.2 信号量的实现原理	142
4.5.3 使用信号量同步共享内存访问的设计	143
4.5.4 信号量编程的主要 API	143
4.5.5 使用信号量同步共享内存访问的实例分析	145
4.6 进程间的邮局——消息队列	150
第 5 章 多线程编程初步	154
5.1 线程的概念和优势	154
5.2 多线程编程的基本 API	155
5.2.1 创建线程 pthread_create	155
5.2.2 结束线程 pthread_exit	155
5.2.3 等待线程结束 pthread_join	156
5.2.4 多线程编程实例分析	156
5.3 多线程的同步与互斥	159
5.3.1 互斥锁	159
5.3.2 信号量	162
5.4 线程属性	165
5.4.1 创建线程时指定属性	165
5.4.2 线程创建后改变属性	167
5.5 使用 gdb 调试多线程程序	173

第 6 章 网络编程	174
6.1 socket 编程所需网络通信基础知识	174
6.1.1 客户端程序和服务端程序	174
6.1.2 常用的命令	174
6.1.3 TCP/UDP 介绍	174
6.1.4 数据封包与解包	174
6.1.5 TCP 连接建立过程——3 次握手	178
6.1.6 TCP 连接终止过程——4 分节终止序列	178
6.2 服务器和客户机的信息函数	179
6.2.1 字节序列转换	179
6.2.2 地址格式转换	181
6.2.3 IP 和域名的转换	182
6.2.4 服务信息函数	183
6.3 TCP socket 编程	183
6.3.1 编程模型	184
6.3.2 主要 API	185
6.3.3 实例分析	187
6.3.4 令人困惑的“不能绑定到指定地址”的错误	191
6.3.5 套接口地址结构几点特别说明	194
6.3.6 迭代服务器与并发服务器	195
6.4 UDP socket 编程	200
6.4.1 编程模型	200
6.4.2 主要 API	201
6.4.3 实例分析	202
6.4.4 特别说明	205
6.5 高级套接字函数	206
6.5.1 send 和 recv	206
6.5.2 sendmsg 和 recvmsg	206
6.5.3 shutdown	207
第 7 章 Linux 下 C 开发环境使用进阶	209
7.1 编辑器 vim 的使用	209
7.1.1 命令行模式的操作	209
7.1.2 命令行模式切换到输入模式	211
7.1.3 最后行模式的操作	211

目 录

7.1.4	vim 的注意事项	212
7.2	编译器 gcc 的使用	213
7.2.1	gcc 警告提示功能	213
7.2.2	库依赖	215
7.2.3	gcc 代码优化	216
7.2.4	gcc 编译加速	218
7.2.5	gcc 的错误类型及对策	218
7.3	程序库文件的制作、链接、加载精解	219
7.3.1	linux 下 ELF 文件格式	219
7.3.2	库函数的编写者如何制作库文件	223
7.3.3	应用程序的编写者如何使用(链接)库文件	224
7.3.4	应用程序的使用者如何加载库文件	225
7.3.5	静态库文件的制作和使用	226
7.4	调试器 gdb 的使用	227
7.4.1	gdb 的命令概貌	227
7.4.2	gdb 中运行 Linux 的 shell 程序	229
7.4.3	在 gdb 中配置程序运行环境	229
7.4.4	调试已运行的程序	230
7.4.5	暂停/恢复程序运行	230
7.4.6	查看栈信息	237
7.4.7	查看源程序	239
7.4.8	查看运行时数据	242
7.4.9	改变程序的执行	249
7.5	autoconf 和 automake 生成 Makefile 文件	253
7.5.1	引子	253
7.5.2	模拟需求	253
7.5.3	工具简介	254
7.5.4	生成 Makefile 的来龙去脉	255
7.5.5	Configure.in 的八股文	256
7.5.6	实战 Makefile.am	258
参考文献		261
后 记		262

第 1 章

Linux 下 C 语言编程环境

1.1 Linux 操作系统及相关开发工具的安装与使用

古语云：工欲善其事，必先利其器。因此本节先介绍 Linux 下 C 语言程序开发环境的搭建。

1.1.1 虚拟机软件 vmware 的安装

本书基于 Linux 进行程序开发，因此必须安装 Linux 操作系统，而不是大多数人习惯使用的 Windows 操作系统。考虑到大多数人习惯在 Windows 环境下进行工作，因此为了减少初学者的陌生感，本节采用在电脑上安装虚拟机软件 vmware7.0，然后再在该软件下安装 Linux 操作系统的方式。

特别说明：本书介绍的虚拟机使用两个硬盘，40 G 的硬盘用于挂载 root 分区(/)并制作 snapshot，这样在系统损坏时，可以快速的一键恢复；80 G 硬盘用于挂载 work 分区(/work)，并设置为不受 snapshot 影响的独立硬盘，以后将在这个分区上编辑、编译软件，这样在系统出错后使用 snapshot 恢复时，不会破坏学习成果。

请按照如下的一系列操作建立虚拟机。

1. 在 Windows 上安装 vmware workstation 7.0 软件

该软件可以从 vmware 的官方网站 <http://www.vmware.com> 下载。

2. 启动 vmware，新建客户虚拟机

如图 1.1 所示，选择 File→New→Virtual Machine 选项新建虚拟机，接着按图 1.2～图 1.8 依次设置安装方式、虚拟机版本、安装操作系统方式、选择哪种操作系统、虚拟机的名字及存储位置、CPU 数目、内存容量。然后进行以下设置。

(1) 在主机能联通互联网的情况下，按图 1.9 选择虚拟机与主机的互联方式为 NAT，使得虚拟机可以通过主机联通互联网。

(2) 按图 1.10～图 1.14 设置硬盘相关的一些配置。图 1.13 中选择“Split disk into 2 GB files”，表示使用多个小于 2GB 的文件来表示一个很大的硬盘。如果 Windows 的硬盘格式为 FAT32，因为 FAT32 支持的最大文件大小为 4 GB，请务必选择

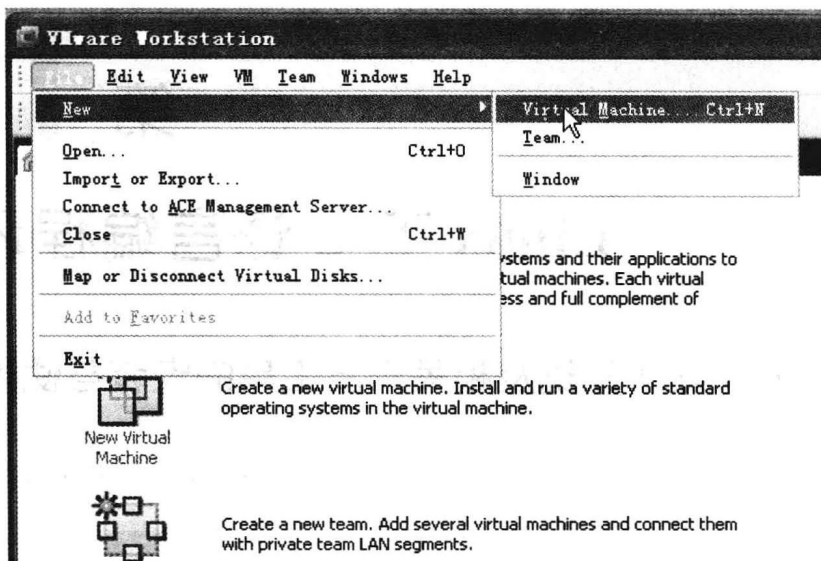


图 1.1 选择新建虚拟机



图 1.2 选择定制虚拟机

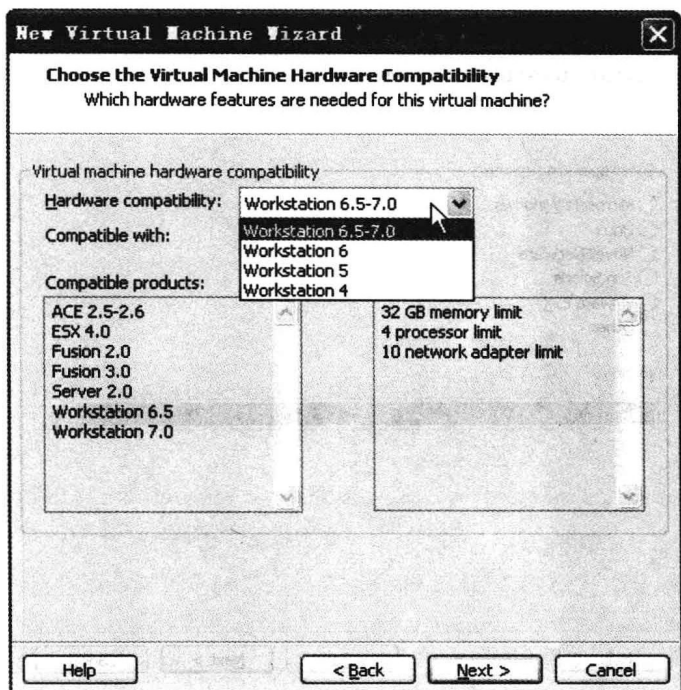


图 1.3 选择虚拟机版本

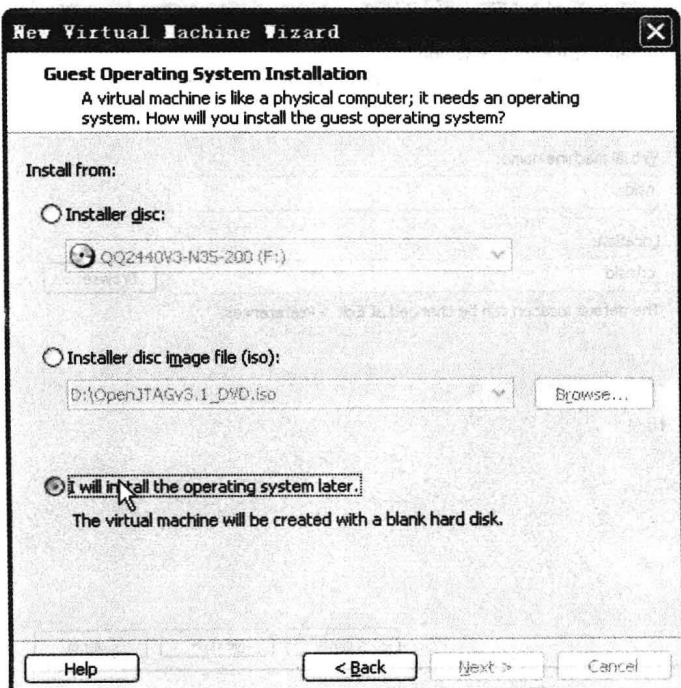


图 1.4 选择虚拟机安装操作系统方式

第 1 章 Linux 下 C 语言编程环境

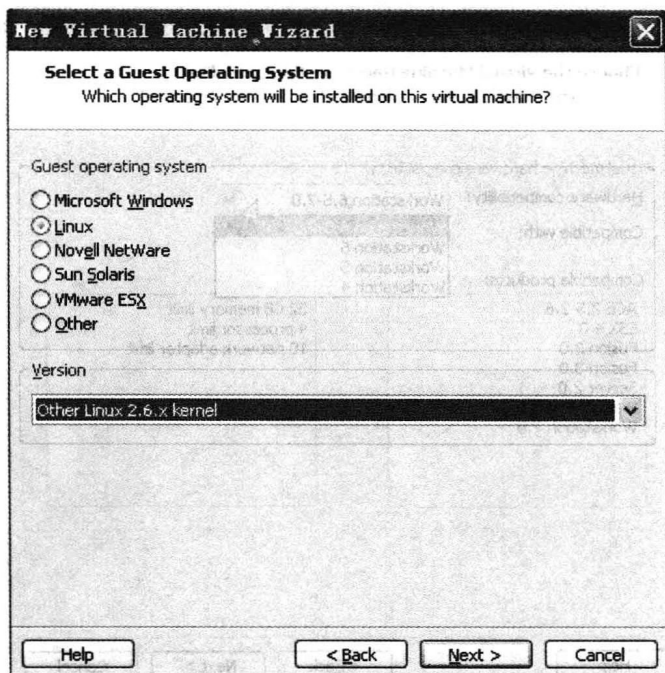


图 1.5 选择虚拟机操作系统

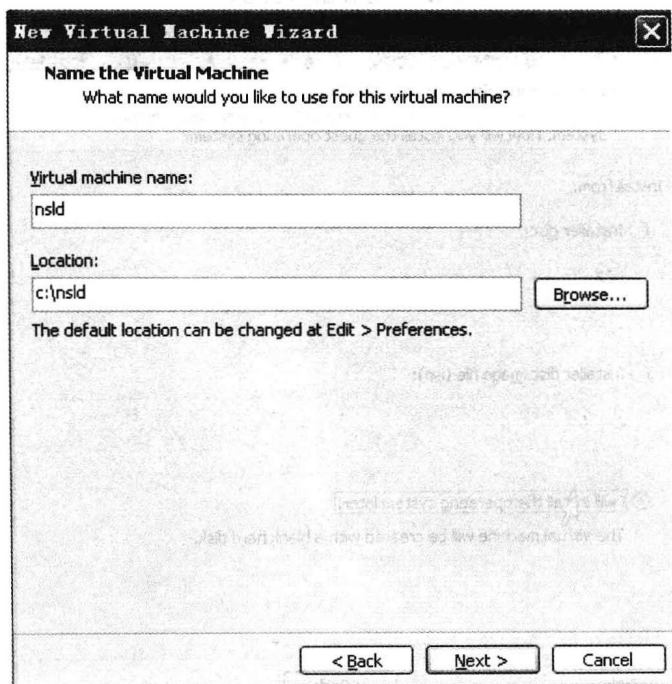


图 1.6 设置虚拟机名字及存储位置

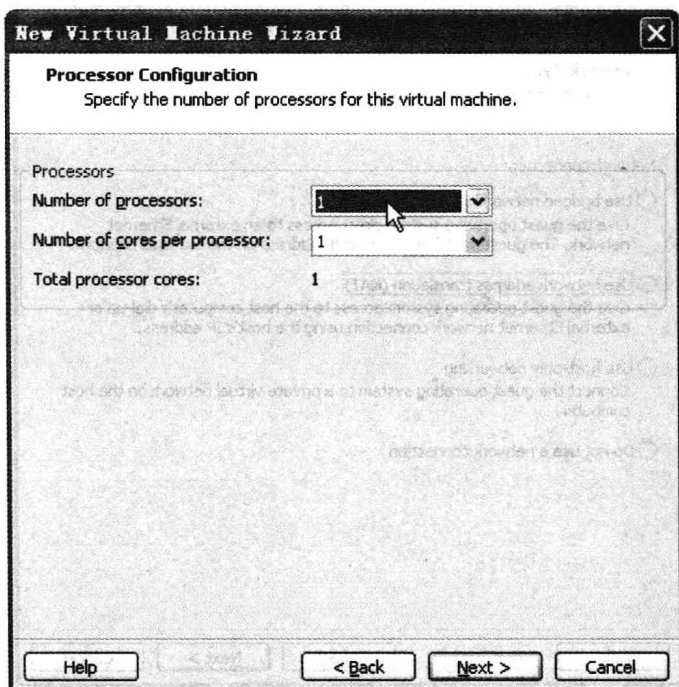


图 1.7 设置虚拟机 CPU 数目

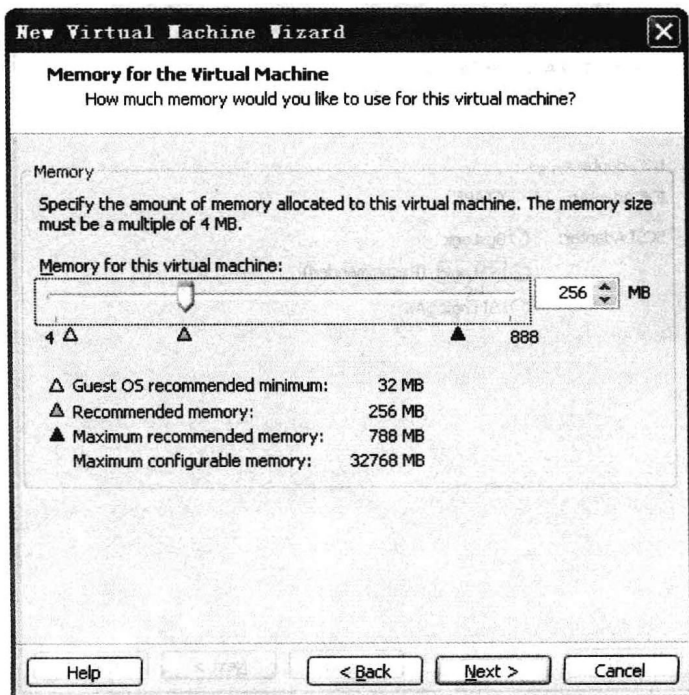


图 1.8 设置虚拟机内存数量