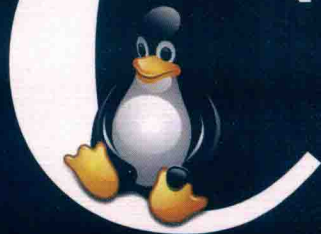


Linux C

编程完全解密



闫敬 吴淑坤 编著

- ✿ 由浅入深地讲解Linux C编程基础
- ✿ 精选实用性强的项目案例辅助教学
- ✿ 配有大量实例源代码进行实战练习



清华大学出版社



Linux C

编程完全解密



闫敬 吴淑坤 编著

清华大学出版社
北京

内 容 简 介

本书以 Red Hat 9.0 和 Ubuntu 12.01 为平台,系统地介绍了 Linux 操作系统下的各种 shell 命令以及在此平台下进行 C 语言开发的步骤和方法,并通过大量实例讲解在 Linux 下进行 C 语言开发的方法和技巧。

本书共 13 章,包括 Linux 操作系统概述, Linux 的基本操作, 文本编辑器, Linux 下的 C 语言开发基础, 静态库和动态库, make 工程管理, 文件编程, shell 脚本的开发, 进程管理, 进程间通信 (IPC), POSIX 线程, Linux 网络编程, Linux 驱动程序和嵌入式基础。同时还配有综合项目实战环节的源代码,使读者能够在此基础上举一反三,提升开发应用项目的能力。

本书结构合理、内容全面、深入浅出、通俗易懂,具有很强的参考性和实用性。可作为普通高等院校计算机及相关专业进行 Linux 平台下 C 语言程序设计的教材,也可作为程序开发员的参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Linux C编程完全解密 / 闫敬, 吴淑坤编著. — 北京: 清华大学出版社, 2019
ISBN 978-7-302-53128-9

I. ①L… II. ①闫… ②吴… III. ①Linux操作系统—程序设计②C语言—程序设计
IV. ①TP316.89②TP312.8

中国版本图书馆 CIP 数据核字 (2019) 第 104360 号

责任编辑: 魏 莹

封面设计: 杨玉兰

责任校对: 周剑云

责任印制: 宋 林

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

社总机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 17.75

字 数: 431 千字

版 次: 2019 年 8 月第 1 版

印 次: 2019 年 8 月第 1 次印刷

定 价: 69.00 元

产品编号: 078326-01

前言

Linux 系统基于开源软件思想，是当下最流行的操作系统之一。随着 Linux 系统的发展和广泛应用，其已占据绝大多数嵌入式系统和 PC 服务器的市场份额，且桌面系统的普及率也在逐年上升。越来越多的开发者希望了解 Linux 系统开发技术，特别是基于 Linux 系统上流行的 C 语言开发技术。

本书全面介绍了 Linux 系统上 C 语言开发技术，大量实例贯穿全书，由浅入深，力求使读者在学习后，能够掌握 Linux 平台下的开发技能，并通过大量的项目实战，培养综合实践能力。

本书特点

1. 提供 Ubuntu 免安装文件和 Red Hat 安装镜像

为了让读者更好地按照本书的内容进行学习，做到无障碍学习测试，作者提供了开发工具和环境，包括两个 Linux 发行版本和 VMware 10 的安装文件。

2. 配有大量实例源代码

为了让读者更加快速、直观地学习本书内容，在每个章节都安排了实例讲解。另外，一些重要章节的课后上机习题也附有源代码供读者参考。

3. 循序渐进，由浅入深

工欲善其事，必先利其器。为了方便读者学习，本书首先介绍了 Linux 操作系统平台下终端命令的使用方法和技巧。在此基础上，学习如何在 Linux 系统下进行 C 语言的开发。通过基础实例讲解和项目实战逐步培养学习能力，能够举一反三，具备一定的应用开发能力。

4. 项目实战案例

本书安排了两个项目实战，分别是模拟 ATM 功能和局域网内的聊天程序。项目实战旨在培养综合运用知识的能力，能够对所学知识进行有效整合，提高项目开发的能力和水平。

本书内容

第1章：介绍 Linux 操作系统的发展背景和特点。Linux 系统是一个开放的操作系统，在学习的同时，可根据自身需要安装 Linux 操作系统，为实践各种开发技能打好基础。

第2章：介绍了 Linux 系统的基本操作，包括 shell 命令的基本使用、目录及文件的相关操作命令、软件包的安装等内容。

第3章：介绍了 Linux 平台下的 vi 文本编辑器，通过本章的学习，会使用该编辑器编写 C 源文件。

第4章：介绍了 Linux 系统下 C 语言开发步骤，包括编译 C 语言的编译器 gcc 基本概念及编译过程和 IDE 集成开发环境的安装和使用。

第5章：介绍了动态库和静态库的应用，包括库的概述，静态库和动态库的创建步骤以及通过实例讲解静态库和动态库的区别。

第6章：介绍了 make 工程管理，包括 make 工程管理的作用，如何为项目编写规则文件，automake 的使用方法以及相关的实例讲解。

第7章：介绍了文件编程，包括基本的 I/O 函数的使用方法和技巧，文件锁的概念和使用，以及 ATM 项目实战。

第8章：介绍了 shell 脚本的开发，包括 shell 编程基础和 shell 脚本的语法，为了更好地掌握本章内容，特别布置了 shell 脚本设计示例。

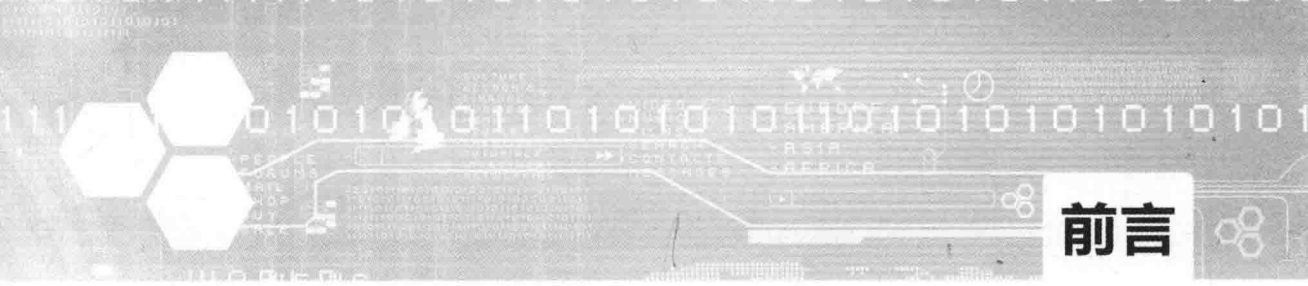
第9章：介绍了 Linux 系统下的进程管理，包括进程的概念，进程控制等。其中进程控制中主要介绍了进程创建函数 fork、进程等待函数 wait 及 waitpid 函数的用法。

第10章：介绍了进程间的通信机制，包括管道、信号、信号量、共享内存、消息队列等通信机制的特点和应用，并进行了实例讲解。

第11章：介绍了 POSIX 线程，包括线程创建，线程等待，线程销毁和线程同步。本章最后通过项目实战——聊天室的设计与实现，对本章内容进行了练习。

第12章：介绍了 Linux 下的网络编程的方法和步骤，包括网络编程的基础知识，socket 实现本地通信和网络通信的编程思路和步骤，最后通过多客户通信的实例将本章内容与第9章内容结合运用，以提高综合运用能力。

第13章：介绍了 Linux 在嵌入式方向的应用，包括 Linux 驱动程序和嵌入式开发基础，嵌入式 Linux 开发的一般流程及开发中需要注意的问题，最后总结了 Linux 嵌入式开发的应用特点。



前言

本书由唐山师范学院的闫敬、吴淑坤老师编写完成。书中实例代码经过严格测试可以直接运行。同时对参与本书代码录入、语法校正等工作的同事们表示由衷的感谢!

由于作者水平有限,时间仓促,书中难免有错误和不妥之处,敬请读者批评指正。

编 者

第 1 章 Linux 操作系统概述 1

1.1 认识 Linux 操作系统.....	2
1.1.1 Linux 操作系统发展背景	2
1.1.2 Linux 操作系统的特点	2
1.1.3 Linux 操作系统的应用现状	3

1.2 Linux C 开发概述	3
1.2.1 Linux C 开发简介	3
1.2.2 IEEE POSIX	4
1.2.3 Linux C 开发工具	4

1.3 小结	4
习题	4

第 2 章 Linux 的基本操作..... 7

2.1 shell 初体验	8
2.1.1 虚拟终端	8
2.1.2 shell 命令	8

2.2 shell 常用命令	9
2.2.1 目录和文件操作命令	9
2.2.2 文件归档及压缩	15
2.2.3 软件包的安装	17

2.3 深入 shell	18
2.3.1 通配符	18
2.3.2 重定向	18
2.3.3 管道	19
2.3.4 自动补全	19
2.3.5 用户操作命令	20

2.3.6 关机与重启	21
-------------------	----

2.4 硬链接与软链接 22

2.4.1 硬链接	23
2.4.2 软链接	24

2.5 小结	24
习题	25

第 3 章 文本编辑器 27

3.1 vi 编辑器概述	28
3.1.1 vi 的三种工作模式	28
3.1.2 vi 的初体验	28

3.2 命令模式下的文本块操作	29
3.2.1 行的定位	29
3.2.2 文本块的复制、移动和删除	30
3.2.3 撤销和重复	31
3.2.4 字符串的查找	31

3.3 末行模式下的常用操作	31
3.4 vi 环境定制.....	32
3.5 小结	33
习题	33

第 4 章 Linux 下的 C 语言

开发基础 35

4.1 C 语言开发的基本步骤.....	36
4.1.1 gcc 编译工具	36

4.1.2	gcc 编译过程详解	37	5.4	静态库和动态库的区别	62
4.1.3	gcc 编译多文件	39	5.4.1	实例测试	63
4.2	头文件	40	5.4.2	验证环节	63
4.2.1	头文件的编辑和使用	40	5.5	综合举例	65
4.2.2	进一步理解头文件	42	5.6	小结	66
4.2.3	头文件重复包含	43	习题		66
4.3	gdb 调试工具	44	第 6 章	make 工程管理	69
4.3.1	gdb 调试基本命令	44	6.1	make 概述	70
4.3.2	gdb 初体验	46	6.1.1	make 命令和 Makefile 文件	70
4.3.3	gdb 的断点调试	48	6.1.2	Makefile 文件编写	70
4.4	IDE 工具 CodeBlocks	51	6.1.3	Makefile 文件中的变量	72
4.4.1	CodeBlocks 的安装	51	6.1.4	Makefile 通配符	73
4.4.2	CodeBlocks 的使用	52	6.1.5	Makefile 文件的默认规则	74
4.5	小结	55	6.1.6	Makefile 文件中的伪目标	75
习题		55	6.1.7	make 的条件执行	76
第 5 章	静态库和动态库	57	6.2	automake 的使用	76
5.1	库的概述	58	6.2.1	automake 生成 Makefile 步骤	76
5.1.1	为什么使用库	58	6.2.2	实例讲解	77
5.1.2	库的特点	58	6.3	小结	80
5.2	静态库	59	习题		80
5.2.1	静态库的创建	59	第 7 章	Linux 下的文件编程	81
5.2.2	静态库的使用	60	7.1	概述	82
5.3	动态库	61	7.1.1	Linux 下的系统调用	82
5.3.1	动态库的创建	61	7.1.2	基本 I/O 函数	82
5.3.2	动态库的使用	62	7.1.3	文件描述符	83

7.2 基本 I/O 操作	83	8.3.1 变量	117
7.2.1 open 函数	84	8.3.2 程序结构	118
7.2.2 close 函数	86	8.3.3 函数	123
7.2.3 write 函数	87	8.3.4 命令及其执行	123
7.2.4 read 函数	88	8.3.5 调试脚本	125
7.2.5 lseek 函数	90		
7.3 文件锁	91	8.4 shell 脚本设计示例	125
7.3.1 Linux 下的文件锁机制	93	8.4.1 查看主机网卡流量	125
7.3.2 文件锁的使用	93	8.4.2 监控 CPU 和内存的 使用情况	126
7.3.3 实例讲解	94	8.4.3 查找日期为某一天	126
7.4 目录操作	97	8.5 小结	127
7.4.1 mkdir 函数	97	习题	127
7.4.2 rmdir 函数	98		
7.4.3 getcwd 函数	99	第 9 章 进程管理	129
7.4.4 opendir 函数	99	9.1 进程概述	130
7.4.5 readdir 函数	100	9.1.1 进程的概念	130
7.4.6 closedir 函数	101	9.1.2 进程 ID	130
7.5 项目实战	101	9.1.3 进程调度	131
7.6 小结	110	9.2 进程控制	132
习题	110	9.2.1 进程控制块	132
第 8 章 shell 脚本的开发	113	9.2.2 进程创建函数 (fork)	133
8.1 shell 编程基础	114	9.2.3 进程等待函数 (wait)	138
8.2 管道和重定向	114	9.2.4 waitpid 函数	139
8.2.1 输出重定向	114	9.2.5 vfork 函数	142
8.2.2 输入重定向	115	9.2.6 进程终止函数 (exit)	143
8.2.3 管道	115	9.2.7 exec 函数	145
8.3 shell 脚本的语法	116	9.3 小结	146
		习题	146

第 10 章 进程间通信 (IPC)..... 149

10.1 进程通信概述..... 150

10.1.1 管道 150

10.1.2 信号 153

10.1.3 信号的产生方式 154

10.1.4 信号的处理方式 154

10.2 信号量 156

10.2.1 信号量定义 157

10.2.2 信号量功能 157

10.2.3 使用信号量 164

10.3 共享内存 167

10.3.1 认识共享内存 167

10.3.2 共享内存函数 168

10.4 消息队列 173

10.4.1 消息队列概述 174

10.4.2 消息队列函数 176

10.4.3 消息队列编程实例 179

10.4.4 ATM 的实现..... 183

10.5 小结..... 189

习题 189

第 11 章 POSIX 线程 191

11.1 认识线程 192

11.2 多线程编程 194

11.2.1 线程创建函数 195

11.2.2 多线程中的线程等待 198

11.2.3 线程私有数据 199

11.2.4 保证多线程编程中函数的

可重入性 202

11.3 线程的属性 203

11.3.1 常用线程属性 203

11.3.2 常用线程属性函数 205

11.4 线程的销毁 206

11.5 线程的同步 209

11.5.1 用信号量进行同步 209

11.5.2 用互斥量进行同步 211

11.6 聊天室的实现..... 214

11.7 小结..... 219

习题 219

第 12 章 Linux 网络编程 221

12.1 网络编程常识..... 222

12.1.1 TCP/IP 协议概述 222

12.1.2 IP 地址与端口 223

12.1.3 网络字节序和主机
字节序 224

12.2 简单的本地通信 225

12.2.1 socket 实现本地通信 226

12.2.2 相关 API 详解 226

12.3 socket 实现网络通信..... 230

12.3.1 基于 TCP 的网络编程..... 230

12.3.2 相关 API 详解 230

12.3.3 基于 UDP 的网络编程 234

12.4 守护进程 237

12.4.1 守护进程概念 237

12.4.2 守护进程的编写要点 238

12.5 多客户通信	242
12.6 小结.....	245
习题	246

第 13 章 Linux 驱动程序和嵌入式基础 249

13.1 Linux 驱动程序与嵌入式开发的基础知识	250
13.1.1 嵌入式 Linux 的特点	250
13.1.2 嵌入式 Linux 的系统开发平台	251
13.1.3 嵌入式 Linux 开发的一般流程	251
13.1.4 嵌入式 Linux 驱动程序	252
13.2 Linux 驱动程序与嵌入式开发中的注意事项	253
13.2.1 Linux 的内存空间划分	253

13.2.2 Linux 的内存管理和 IO 寻址	253
13.2.3 Linux 的时基问题	254

13.3 Linux 驱动程序和嵌入式开发的设备类型	254
13.3.1 字符设备特点	255
13.3.2 块设备特点	255

13.4 Linux 嵌入式开发的应用特点.....	255
13.4.1 常用的调试方法	255
13.4.2 可移植性问题	256

13.5 小结.....	256
习题	257

附录 习题答案	259
---------------	-----

```
selection at the end -add back the deselected mirror modifier object
r_ob.select= 1
r_ob.select=1
context.scene.objects.active = modifier_ob
("Selected" + str(modifier_ob)) # modifier ob is the active ob
mirror_ob.select = 0
= bpy.context.selected_objects[0]
data.objects[one.name].select = 1

print("please select exactly two objects, the last one gets the modifier unless
--- OPERATOR CLASSES -----
```

第 1 章

Linux 操作系统概述

Linux 是一套免费使用和自由传播的类 UNIX 操作系统，是一个基于 POSIX 和 UNIX 的多用户、多任务、支持多线程和多 CPU 的操作系统。它能运行主要的 UNIX 工具软件、应用程序和网络协议，且支持 32 位和 64 位硬件。Linux 继承了 UNIX 以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。

1.1 认识 Linux 操作系统

Linux 操作系统诞生于 1991 年 10 月 5 日。Linux 有多个版本，且都使用了 Linux 内核。Linux 可安装在各种计算机硬件设备中，例如手机、平板电脑、路由器、视频游戏控制台、台式计算机、大型机和超级计算机。

严格来讲，Linux 这个词本身只表示 Linux 内核，但实际上人们已经习惯用 Linux 来形容整个基于 Linux 内核，并且使用 GNU 的工具和数据库的操作系统。

1.1.1 Linux 操作系统发展背景

Linux 操作系统核心最早是由芬兰的 Linus Torvalds 于 1991 年 8 月在芬兰赫尔辛基大学上学时发布的。后来经过众多世界顶尖软件工程师的不断修改和完善，Linux 得以在全球普及开来，应用于服务器领域及个人桌面版，在嵌入式开发方面更是具有其他操作系统无可比拟的优势。

Linux 是一套免费的 32 位多人多工的操作系统，其稳定性、多工能力与网络功能是许多商业操作系统无法比拟的。另外，Linux 的最大特色在于源代码完全公开，任何人皆可自由取得、散布，甚至修改源代码。

1.1.2 Linux 操作系统的特点

Linux 的基本思想有两点：第一，一切都是文件，即系统中的所有内容都归结为一个文件，包括命令、硬件和软件设备、操作系统、进程等；第二，每个软件都有确定的用途。对于操作系统内核而言，都被视为拥有各自特性或类型的文件。至于说 Linux 是基于 UNIX 的，很大程度上也是因为这两者的基本思想十分相近。下面介绍 Linux 的特点。

1. 完全免费

Linux 是一款免费的操作系统，用户可以通过网络或其他途径免费获得，并可以任意修改其源代码，这是其他操作系统做不到的。正是由于这一点，来自全世界的无数程序员根据自己的兴趣和灵感参与了 Linux 的修改、编写工作，这让 Linux 不断升级。

2. 完全兼容 POSIX 1.0 标准

这使得可以在 Linux 下通过相应的模拟器运行常见的 DOS、Windows 程序。这为用户从 Windows 转到 Linux 奠定了基础，消除了他们的疑虑。

3. 多用户、多任务

Linux 支持多用户，每个用户的文件设备都有自己特殊的权利，保证了用户之间互不影响。多任务则是 Linux 最主要的一个特点，Linux 可以使多个程序同时并独立运行。

4. 良好的界面

Linux 同时具有字符界面和图形界面。在字符界面用户可以通过键盘输入相应的指令来进行操作。图形界面则是类似于 Windows 的 X-Window 系统，用户可以使用鼠标对其进行操作，其环境和 Windows 相似，可以说是一个 Linux 版的 Windows。

5. 支持多种平台

Linux 可以运行在多种硬件平台上,如具有 x86、680x0、SPARC、Alpha 等处理器的平台。此外 Linux 还是嵌入式操作系统,可以运行在掌上电脑、机顶盒或游戏机上。2001 年 1 月份发布的 Linux 2.4 版内核已经能够完全支持 Intel 64 位芯片架构。同时 Linux 也支持多处理器技术,多个处理器同时工作,使系统性能大大提高。

1.1.3 Linux 操作系统的应用现状

互联网产业的迅猛发展,促使云计算、大数据产业形成并快速发展,云计算、大数据作为一个基于开源软件的平台, Linux 占据了核心优势。据 Linux 基金会调查,86% 的企业已经使用 Linux 操作系统进行云计算、大数据平台的构建,成为最受青睐的云计算、大数据平台操作系统。

目前企业大量使用 Linux 作为服务器, Tomcat、jobss 这一类都是搭建在 Linux 上的,以及需要学习的数据库 Mysql、Oracle、DB2、Greenplum 等也都是使用 Linux 搭建的。

在全球超级计算机 TOP500 强操作系统排行榜中, Linux 的占比长期保持在 85% 以上,且呈快速上升趋势。根据 2016 年的排行榜, Linux 的占比已高达 98.80%。其实在各企业的服务器应用领域, Linux 系统的市场份额也越来越接近这个比例,这足以说明 Linux 的表现非常出色。

1.2 Linux C 开发概述

Linux 的本质只是操作系统的核心,负责控制硬件、管理文件系统、程序进程等。Linux Kernel(内核)并不负责提供用户强大的应用程序,它没有编译器、系统管理工具、网络工具、Office 套件、多媒体、绘图软件等,其系统无法发挥强大功能,用户也无法利用这个系统工作,因此有人提出以 Linux 为核心再集成搭配各式各样的系统程序或应用工具程序组成一套完整的操作系统,而经过如此组合的 Linux 套件即称为 Linux 发行版。

国外封装的 Linux 以 Red Hat(又称为“红帽 Linux”）、Ubuntu、Open Linux、SUSE、Turbo Linux 等最为成功;国内 Linux 发行版做得相对成功的是红旗和中软两个版本。

1.2.1 Linux C 开发简介

Linux C 开发和以前学过的 C 语言有什么本质区别呢?C 语言学习的主要内容包括:

- (1) C 的语法。
- (2) 标准 C 的库函数。

而 Linux 下的 C 开发课程学习的是 Linux 系统调用,也就是说如何使用 Linux 操作系统提供的函数,这是内核提供的函数,而系统调用属于底层调用,适合硬件编程,比如驱动等的编程。

应用程序既可以使用系统调用,也可以使用库函数。系统调用通常提供一种最小接口,而库函数通常提供比较复杂的功能,实际上也可以将库函数理解为对系统调用的封装。C 库函数和系统调用的关系及差别如图 1-1 所示。

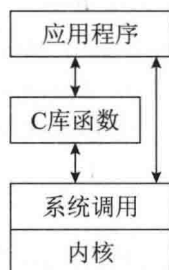


图 1-1 C 库函数和系统调用的关系

1.2.2 IEEE POSIX

POSIX 是由 IEEE(电气和电子工程师协会)制定的标准族。POSIX 是指可移植操作系统接口 (Portable Operating System Interface)。它定义了操作系统应该为应用程序提供的接口标准,是 IEEE 为要在各种 UNIX 操作系统上运行的软件而定义的一系列 API 标准的总称,目的是提升应用程序在各个 UNIX 系统环境之间的可移植性。

1.2.3 Linux C 开发工具

在 Linux 操作系统下,C 语言编辑器一般采用 vi、gedit,其中 vi 是使用比较广泛的编辑器,文件名后缀为 .c。编译工具通常采用 gcc 编译器。gcc 是 GNU 推出的基于 C/C++ 的编译器,是开放源代码领域应用最广泛的编译器,其功能强大,编译代码支持性能优化。目前,gcc 可以用来编译 C/C++、Java 等语言程序,开发人员可根据需要选择安装支持的语言。具体的开发步骤及编译过程详见本书第 4 章。

1.3 小结

本章首先介绍了 Linux 操作系统的发展背景、特点以及应用现状,然后介绍了 Linux 下的 C 编程的主要内容和主要步骤,最后简单介绍了 Linux 下 C/C++ 语言源程序的编译工具 gcc。希望读者通过本章的学习,对 Linux 操作系统以及 Linux 下的 C 编程有一个基本认识。

习 题

一、填空题

1. Linux 操作系统的发行版本中,国外比较著名的是_____。
2. 在 Linux 下进行 C 开发,用到的编译工具是_____。
3. _____是指可移植操作系统接口。

4. Linux 操作系统的特点是_____。
5. Linux 操作系统是_____操作系统的一个克隆版本。

二、简答题

1. 简述 Linux 系统发展历程及比较著名的 Linux 版本。
2. 简述 Linux 系统中库文件的作用。
3. 简述 Linux 程序设计的特点。

三、上机题

选择一种 Linux 发行版本，将其安装到自己的计算机上。

