

- 从初学者的角度出发，以通俗易懂的语言，配合丰富多彩的实例，详细介绍使用Linux C/C++进行企业级程序开发应该掌握的各方面知识
- 集基础知识、核心技能、高级应用、项目案例于一体，易于学习、易于掌握、易于应用



Linux C与C++ 一线开发实践

朱文伟 李建英 著



清华大学出版社



Linux C与C++ 一线开发实践

朱文伟 李建英 著



清华大学出版社
北京

内 容 简 介

Linux C/C++编程在Linux应用程序开发中占有重要的地位,掌握这项技能将在就业竞争中立于不败之地。本书是一本针对初、中级读者的、贴近软件公司一线开发实践的书。

本书共分为19章,内容包括Linux概述、搭建开发环境、语言基础、文件编程、多进程编程、进程间通信、Web编程、多线程编程、Linux下的库、TCP/IP协议基础、网络编程、网络性能测试工具iPerf简析、版本控制和SVN工具、C++跨平台开发以及安全编程等。

本书适合想全面学习Linux环境下C/C++语言编程的读者,并可作为初中级开发人员的案头查阅与参考手册,也适合作为高等院校和培训学校相关专业师生的教学参考书。

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与C++一线开发实践 / 朱文伟, 李建英著. —北京: 清华大学出版社, 2018
ISBN 978-7-302-51255-4

I. ①L… II. ①朱… ②李… III. ①Linux 操作系统—程序设计②C 语言—程序设计③C++语言—程序设计 IV. ①TP316.89 ②TP312.8

中国版本图书馆CIP数据核字(2018)第213834号

责任编辑: 夏毓彦

封面设计: 王 翔

责任校对: 闫秀华

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: 190mm×260mm

印 张: 44.75

字 数: 1152千字

版 次: 2018年12月第1版

印 次: 2018年12月第1次印刷

定 价: 129.00元

产品编号: 072288-01

前言

这是一本 Linux C/C++ 入门的经典图书。任何学过 C/C++ 语言并立志成为一名 Linux 开发工程师的朋友，都可以从本书起步。本书虽然有点厚，但内容通俗易懂，由浅入深，并且实例丰富、步骤详细、注释充分，相信大家都能看得懂。对于中高级开发人员，也可以通过本书快速上手 Linux C/C++ 的实际开发。本书并没有详细讲述 C++ 语言部分（但也进行了一定程度的叙述），而是把更多的笔墨放在 Linux 编程方面，因此书中都是实实在在的 Linux 编程“干货”。此外，实例丰富是本书的一大特点，大家应该知道，编程开发只是了解理论是不够的，只有自己上机调试运行实例，才能深刻理解编程，尤其是 C/C++。另外，为了照顾初学者，每个实例步骤都非常详细，并且从建立工程到运行工程都有着丰富的注释。最后，本书所有例子都在 CentOS 7 上用 gcc/g++ 编译通过。

本书在讲述基本编程的同时，也讲述了很多一线实践开发中经常会碰到的问题和解决方案，可以说本书是紧贴工业界的图书。希望大家能够通过本书的学习打好 Linux 开发的基础，早日成为 Linux C/C++ 开发高手。

本书示例源代码下载

本书示例源代码请扫描下面的二维码获得：



如果下载有问题，请联系 booksaga@163.com，邮件主题为“Linux C 与 C++ 一线开发实践”。

技术支持与鸣谢

虽然编者尽了最大努力，但是书中难免会出现一些疏漏，希望大家不吝指正。编者的 QQ 技术支持邮箱是 itpxw@qq.com，欢迎大家来信。

这里要感谢夫人李建英，没有她认真测试示例程序就不能顺利完成本书的写作。同时也感谢清华大学出版社的老师，他们辛勤的工作使得本书得以顺利出版。

编 者

2018 年 11 月

目 录

第 1 章 Linux 概述	1
1.1 什么是 Linux	1
1.2 Linux 的简史	2
1.3 Linux 和 Windows 的比较	4
1.4 Linux 主要应用领域	4
1.5 Linux 的版本	5
1.6 使用哪个版本的 Linux 进行学习	5
1.7 Linux 的特点	5
1.8 如何学习 Linux	6
1.9 命令行还是图形界面	6
1.10 计算机启动的基本过程	6
1.10.1 按下电源	7
1.10.2 BIOS 自检	7
1.10.3 系统引导	8
1.10.4 实模式和保护模式	10
1.11 启动内核	11
1.12 认识 Shell	11
1.13 常见的 Shell	11
1.14 图形界面和字符界面的切换	12
1.14.1 在不退出 X-Window 的情况下切换到字符界面	12
1.14.2 强行退出 X-Window 进入文本模式	12
1.14.3 设置每次开机进入字符界面	13
1.14.4 从字符界面进入图形界面	14
1.15 Shell 命令概述	14
1.16 环境变量	14

第 2 章 搭建 Linux C++ 开发环境	16
2.1 准备 Linux 虚拟机	16
2.2 连接 Linux 虚拟机	20
2.2.1 通过桥接模式连接虚拟机	21
2.2.2 主机模式	24
2.2.3 通过 NAT 模式连接虚拟机	25
2.3 通过终端工具连接 Linux 虚拟机	27
2.4 搭建 Linux 下的 C++ 开发环境	30
2.4.1 非集成开发方式	30
2.4.2 集成开发方式	31
2.5 搭建 Windows 下的 Linux C++ 开发环境	37
2.5.1 搭建非集成式的 Windows 下的 Linux C++ 开发环境	37
2.5.2 搭建集成式的 Windows 下的 Linux C++ 开发环境	39
2.6 需要掌握的开发工具	49
2.7 vi 编辑器的使用	50
2.7.1 vi 编辑器概述	50
2.7.2 vi 编辑器的工作模式	50
2.7.3 vi 的基本操作	51
2.7.4 命令行模式下的基本操作	51
2.7.5 插入模式	55
2.7.6 末行模式操作	55
2.8 gcc 编译器的使用	57
2.8.1 gcc 对 C 语言的编译过程	57
2.8.2 gcc 所支持的后缀名文件	62
2.8.3 gcc 的语法格式	62
2.8.4 gcc 常见选项	64
2.9 g++ 的基本使用	75
2.10 gdb 调试器的使用	77
2.10.1 为何要学习 gdb 调试器	77
2.10.2 gdb 简介	77
2.10.3 重要准备	77
2.10.4 启动 gdb	77
2.10.5 退出 gdb	78

2.10.6	gdb 的常用命令概述	79
2.10.7	file 命令加载程序	80
2.10.8	list 命令显示源代码	80
2.10.9	run 命令运行程序	83
2.10.10	break 命令设置断点	85
第 3 章	C++语言基础	89
3.1	C++基础知识	89
3.1.1	C++程序结构	89
3.1.2	注释	91
3.1.3	变量和数据类型	92
3.1.4	标识	92
3.1.5	基本数据类型	93
3.1.6	变量的定义和 C++11 中的 auto	94
3.1.7	变量的范围	98
3.1.8	变量初始化	98
3.1.9	常量	99
3.1.10	操作符/运算符	103
3.1.11	控制台交互	110
3.2	控制结构	115
3.2.1	条件结构	115
3.2.2	循环结构	116
3.2.3	分支控制和跳转	120
3.2.4	选择结构语句 switch	122
3.3	函数	123
3.4	函数高级话题	127
3.4.1	参数按数值传递和按地址传递	127
3.4.2	函数重载	130
3.4.3	内联函数	131
3.4.4	递归	132
3.4.5	函数的声明	133
3.5	高级数据类型	134

3.5.1	数组.....	134
3.5.2	指针.....	145
3.5.3	动态分配内存.....	155
3.5.4	结构体.....	159
3.5.5	自定义数据类型.....	165
3.6	面向对象编程	168
3.6.1	类.....	168
3.6.2	构造函数和析构函数	171
3.6.3	构造函数重载.....	173
3.6.4	类的指针.....	175
3.6.5	由关键字 struct 和 union 定义的类	176
3.6.6	操作符重载.....	176
3.6.7	关键字 this.....	179
3.6.8	静态成员.....	180
3.6.9	类之间的关系.....	182
3.6.10	多态.....	189
3.7	C++面向对象小结	195
3.8	C++高级知识	199
3.8.1	模板.....	199
3.8.2	命名空间.....	205
3.8.3	异常处理.....	209
3.8.4	预处理指令.....	213
3.8.5	预定义宏.....	215
3.8.6	C++11 中的预定义宏	216
3.9	字符串.....	218
3.9.1	字符串基础.....	218
3.9.2	搜索与查找.....	227
3.10	再论异常处理	233
3.10.1	基本概念.....	233
3.10.2	抛出异常.....	234
3.10.3	捕获异常.....	234
3.10.4	C++ 标准异常	235
3.10.5	定义新的异常.....	236

3.11 再论函数模板	237
3.12 字符集.....	239
3.12.1 计算机上的 3 种字符集	239
3.12.2 查看 Linux 系统的字符集.....	241
3.12.3 修改 Linux 系统的字符集.....	242
3.12.4 Unicode 编码的实现.....	242
3.12.5 C 运行时库对 Unicode 的支持	246
3.12.6 C++标准库对 Unicode 的支持.....	247
3.12.7 字符集相关实例	248
第 4 章 Linux 文件编程	249
4.1 文件系统.....	249
4.1.1 基本概念.....	249
4.1.2 文件系统层次结构标准	249
4.2 文件的属性信息	250
4.3 i 节点	251
4.3.1 基本概念.....	251
4.3.2 i 节点的内容	251
4.3.3 i 节点的使用状况	253
4.4 文件类型.....	254
4.4.1 普通文件.....	255
4.4.2 目录.....	255
4.4.3 块设备文件.....	256
4.4.4 字符设备文件.....	257
4.4.5 链接文件.....	257
4.5 文件权限.....	259
4.6 Linux 文件 I/O 编程的基本方式	260
4.7 什么是 I/O	260
4.8 Linux 系统调用下的文件 I/O 编程	261
4.8.1 文件描述符.....	261
4.8.2 打开或创建文件.....	262
4.8.3 创建文件.....	263
4.8.4 关闭文件.....	264

4.8.5	读取文件中的数据	266
4.8.6	向文件写入数据.....	268
4.8.7	设定文件偏移量.....	269
4.8.8	获取文件状态.....	271
4.8.9	文件锁定.....	272
4.8.10	建立文件和内存映射	276
4.8.11	mmap 和共享内存对比	279
4.9	C++方式下的文件 I/O 编程.....	280
4.9.1	流的概念.....	280
4.9.2	流的类库.....	280
4.9.3	打开文件.....	281
4.9.4	关闭文件.....	283
4.9.5	写入文件.....	283
4.9.6	读取文件.....	283
4.9.7	文件位置指针.....	285
4.9.8	状态标志符的验证	287
4.9.9	读写文件数据块.....	288
4.10	文件编程中的其他操作	290
4.10.1	获取文件有关信息	290
4.10.2	创建和删除文件目录项	293
第 5 章	多进程编程.....	296
5.1	进程的基本概念	296
5.2	进程的描述	296
5.2.1	进程的标识符.....	299
5.2.2	PID 文件	301
5.3	进程的创建	303
5.3.1	使用 fork 创建进程.....	303
5.3.2	使用 exec 创建进程	305
5.3.3	使用 system 创建进程	311
5.4	进程调度.....	312
5.5	进程的分类	315
5.5.1	前台进程.....	315

5.5.2 后台进程.....	315
5.6 守护进程.....	316
5.6.1 守护进程的概念.....	316
5.6.2 守护进程的特点.....	317
5.6.3 查看守护进程.....	317
5.6.4 守护进程的分类.....	318
5.6.5 守护进程的启动方式	319
5.6.6 编写守护进程的步骤	319
第 6 章 Linux 进程间的通信.....	323
6.1 信号.....	323
6.1.1 信号的基本概念.....	323
6.1.2 与信号相关的系统调用	328
6.2 管道.....	336
6.2.1 管道的基本概念.....	336
6.2.2 管道读写的特点.....	337
6.2.3 管道的局限性.....	337
6.2.4 创建管道函数 pipe.....	338
6.2.5 读写管道函数 read/write	338
6.2.6 等待子进程中断或结束的函数 wait	338
6.2.7 使用管道的特殊情况	342
6.3 消息队列.....	342
6.3.1 创建和打开消息队列函数 msgget.....	343
6.3.2 获取和设置消息队列的属性函数 msgctl.....	343
6.3.3 将消息送入消息队列的函数 msgsnd.....	344
6.3.4 从消息队列中读取一条新消息的函数 msgrcv.....	345
6.3.5 生成键值函数 ftok.....	346
第 7 章 C++ Web 编程	354
7.1 CGI 程序的工作方式.....	354
7.2 架设 Web 服务器 Apache.....	354
第 8 章 多线程基本编程.....	358
8.1 使用多线程的好处	358
8.2 多线程编程的基本概念	359

8.2.1	操作系统和多线程	359
8.2.2	线程的基本概念	359
8.2.3	线程的状态	360
8.2.4	线程函数	361
8.2.5	线程标识	361
8.2.6	C++多线程开发的两种方式	361
8.3	利用 POSIX 多线程 API 函数进行多线程开发	362
8.3.1	线程的创建	362
8.3.2	线程的属性	367
8.3.3	线程的结束	379
8.3.4	线程退出时的清理机会	387
8.4	C++11 中的线程类	392
8.4.1	线程的创建	393
8.4.2	线程的标识符	401
8.4.3	当前线程 <code>this_thread</code>	402
第 9 章	多线程高级编程	406
9.1	多线程的同步和异步	406
9.2	线程同步	406
9.3	利用 POSIX 多线程 API 函数进行线程同步	411
9.3.1	互斥锁	411
9.3.2	读写锁	417
9.3.3	条件变量	424
9.4	C++11/14 中的线程同步	431
9.5	线程池	434
9.5.1	线程池的定义	434
9.5.2	使用线程池的原因	435
9.5.3	用 C++实现一个简单的线程池	435
第 10 章	Linux 下的库	441
10.1	库的基本概念	441
10.2	库的分类	441
10.3	静态库	442
10.3.1	静态库的基本概念	442

10.3.2 静态库的创建和使用	442
10.4 动态库.....	445
10.4.1 动态库的基本概念	445
10.4.2 动态库的创建和使用	445
第 11 章 TCP/IP 协议基础.....	450
11.1 什么是 TCP/IP	450
11.2 TCP/IP 协议的分层结构	450
11.3 应用层.....	453
11.3.1 DNS.....	454
11.3.2 端口的概念.....	454
11.4 传输层.....	455
11.4.1 TCP 协议	455
11.4.2 UDP 协议.....	456
11.5 网络层.....	456
11.5.1 IP 协议.....	456
11.5.2 ARP 协议.....	462
11.5.3 RARP 协议	464
11.5.4 ICMP 协议.....	465
11.6 数据链路层	474
11.6.1 数据链路层的基本概念	474
11.6.2 数据链路层的主要功能	474
第 12 章 套接字基础.....	476
12.1 网络程序的架构	477
12.2 套接字的类型	478
12.3 套接字的地址结构	478
12.4 主机字节序和网络字节序	479
12.5 出错信息的获取	481
第 13 章 TCP 套接字编程.....	483
13.1 TCP 套接字编程的基本步骤.....	483
13.2 协议簇和地址簇	484
13.3 socket 地址	487
13.3.1 通用 socket 地址	487

13.3.2	专用 socket 地址	488
13.3.3	IP 地址的转换	489
13.4	TCP 套接字编程的相关函数	491
13.4.1	socket 函数	491
13.4.2	bind 函数	492
13.4.3	listen 函数	494
13.4.4	accept 函数	494
13.4.5	connect 函数	495
13.4.6	write 函数	497
13.4.7	read 函数	498
13.4.8	send 函数	498
13.4.9	recv 函数	499
13.4.10	close 函数	499
13.4.11	获得套接字地址	499
13.4.12	阻塞套接字的使用	504
13.4.13	非阻塞套接字的使用	511
第 14 章	UDP 套接字编程	525
14.1	UDP 套接字编程的基本步骤	525
14.2	TCP 套接字编程的相关函数	526
14.2.1	消息发送函数 sendto 和 sendmsg	526
14.2.2	消息接收函数 recvfrom 和 recvmsg	527
14.3	实战 UDP 套接字	529
14.4	UDP 丢包及无序问题	538
第 15 章	原始套接字编程	539
15.1	原始套接字概述	539
15.2	与标准套接字的区别	539
15.3	原始套接字的编程方法	540
15.4	面向链路层的原始套接字编程函数	540
15.4.1	创建原始套接字函数	540
15.4.2	接收函数 recvfrom	541
15.4.3	发送函数 sendto	542
15.5	以太网帧格式	545

15.6 获取网络接口的信息	547
15.7 实战链路层的原始套接字	550
15.7.1 常见的应用场景	550
15.7.2 混杂模式.....	577
15.7.3 链路层原始套接字开发注意事项	596
15.8 面向 IP 层的原始套接字编程.....	597
第 16 章 C++网络性能测试工具 iPerf 的简析	605
16.1 iPerf 概述.....	605
16.2 iPerf 的特点.....	605
16.3 iPerf 的工作原理.....	605
16.4 iPerf 的主要功能.....	606
16.5 在 Linux 下安装 iPerf.....	607
16.6 iPerf 的简单使用.....	608
16.7 iPerf 源代码概述.....	609
16.8 Thread 类	610
16.8.1 数据成员说明.....	611
16.8.2 主要函数成员.....	611
16.9 SocketAddr 类	615
16.10 Socket 类.....	617
16.10.1 Listen 函数.....	618
16.10.2 Accept 函数	620
16.10.3 Connect 函数	620
第 17 章 版本控制和 SVN 工具	623
17.1 SVN 简介	623
17.1.1 什么是 SVN.....	623
17.1.2 使用 SVN 的好处	624
17.1.3 使用 SVN 的基本流程	624
17.2 SVN 服务器的安装和配置	624
17.2.1 VisualSVN 服务器的安装和配置	624
17.2.2 SVN 客户端在 Windows 上的使用	629
第 18 章 C++跨平台开发	634
18.1 什么是跨平台	634

18.2	C++的可移植性	634
18.2.1	可移植性的概念	634
18.2.2	影响 C++语言可移植性的因素	635
18.3	设计跨平台软件的原则	638
18.3.1	避免语言的扩展特性	638
18.3.2	实现动态的处理	638
18.3.3	使用脚本文件进行管理	639
18.3.4	使用安全的数据串行化	640
18.3.5	跨平台开发中的编译及测试	641
18.3.6	实现抽象	641
18.4	建立跨平台的开发环境	642
18.4.1	跨平台开发编译器的选择	642
18.4.2	建立跨平台的 Make 系统	643
18.5	C++语言跨平台软件开发的实现	648
18.6	C++语言跨平台的开发策略	649
18.7	建立统一的工程包	650
18.8	建立跨平台的代码库	650
18.9	工厂模式与单例模式的实现	651
18.10	利用平台依赖库封装平台相关代码	651
18.11	处理器的差异控制	652
18.11.1	内存对齐	652
18.11.2	字节顺序	653
18.11.3	类型的大小	654
18.11.4	使用预编译处理类型差异	654
18.12	编译器的差异控制	655
18.12.1	实现平台无关的代码	655
18.12.2	内存管理	657
18.12.3	容错性的影响	657
18.12.4	利用日志管理异常	657
18.13	操作系统和接口库	658
18.13.1	文件描述符的限制	659
18.13.2	进程和线程的限制	659
18.13.3	操作系统抽象层	659