

COMPUTER

高等院校计算机技术



“十二五”规划教材

Linux

程序设计实践与编程技巧

◎ 主 编 刘加海 骆建华
副主编 王晓池 杨 锴



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

高等院校计算机技术“十二五”规划教材

Linux 程序设计实践与编程技巧

主 编 刘加海 骆建华

副主编 王晓池 杨 皓



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

内容简介

本书是浙江大学出版社出版的《Linux 程序设计》一书的辅导书，书中内容主要分两大部分。一是程序设计要点与技巧；二是实践部分，包含 16 个实验，每个实验有 6~7 个任务。本书包含：终端基本命令、Shell 程序设计、Linux 环境下 C 程序编译技巧、Linux 环境下 C 语言系统函数的应用、Linux 环境下文件的 I/O 操作、进程的控制与进程调度、线程、Linux 环境下的图形与游戏程序设计、网络程序设计、字符驱动程序设计、串行程序设计的编程知识要点，编程技巧与程序设计实例。并包含以上各部分及 Linux 环境与 Windows 环境资源共享设置的实验。

本书重点给出 Linux 程序设计的要点与技巧，希望能够为本科生、研究生、嵌入式工程技术人员、Linux 程序设计师及 Linux 程序爱好者提供有效的帮助。

图书在版编目 (CIP) 数据

Linux 程序设计实践与编程技巧 / 刘加海，骆建华主编.
—杭州：浙江大学出版社，2013.6

ISBN 978-7-308-11315-1

I. ①L… II. ①刘… ②骆… III. ①Linux 操作系统—程序设计 IV. ① TP316.89

中国版本图书馆 CIP 数据核字(2013)第 060654 号

Linux 程序设计实践与编程技巧

主 编 刘加海 骆建华

副主编 王晓池 杨 皓

责任编辑 武晓华

封面设计 刘依群

出版发行 浙江大学出版社

(地址：杭州市天目山路 148 号 邮编：310007)

(网址：<http://www.zjupress.com>)

排 版 杭州理想广告有限公司

印 刷 浙江省良渚印刷厂

开 本 787mm×1092mm 1/16

印 张 20.25

字 数 520 千

版 次 2013 年 6 月第 1 版 2013 年 6 月第 1 次印刷

书 号 ISBN 978-7-308-11315-1

定 价 39.00 元

版权所有 翻印必究 印装差错 负责调换

浙江大学出版社发行部邮购电话 (0571)88925591

前 言

PROFACE

随着 Linux 操作系统在各个行业的广泛应用,企业对 Linux 人才的需求也将持续升温,具有 Linux 专业知识及开发经验的人才越来越受到用人单位的欢迎。

本书是浙江大学出版社出版的《Linux 程序设计》一书的辅导书,书中内容主要分两大部分:一是程序设计要点与技巧,二是实践部分。第一部分是编程技巧,包含了 Linux 基本命令操作、Shell 程序设计、makefile 工程文件、Linux 环境下 C 语言系统函数的应用、Linux 环境下文件的 I/O 操作、进程的控制与进程通信、线程、Linux 环境下的图形程序与游戏程序设计、网络程序设计、串行程序设计等内容的要点、应用技巧及程序设计实例。实践部分包含 16 个实验,每个实验有 6~7 个任务,含终端基本命令、Shell 程序设计、Linux 环境下 C 程序编译技巧、Linux 环境下 C 语言系统函数的应用、Linux 环境下文件的 I/O 操作、进程的控制与进程调度、线程、Linux 环境下的图形与游戏程序设计、网络程序设计、字符驱动程序设计、串行程序设计、Linux 环境与 Windows 环境资源共享设置。

十多年来,笔者一直在浙江大学计算机学院、浙江大学软件学院、LUPA 全国 Linux 师资培训中心讲授《Linux 程序设计》这门课程,此课程深受本科生、研究生、高校教师的欢迎,此后笔者根据多年的教学经验编写了《Linux 软件工程师(C 语言)实用教材》一书,此书由倪光南院士写序,在读者中深受欢迎。应广大读者的需求,笔者根据教学过程中发现的问题及教学的需要,编写了《Linux 程序设计》一书,此书由浙江省教育厅推荐为国家“十二五”规划教材。在使用中学生希望有本对于在实践方面与编程中遇到的难点给予指导的辅导书,所以又花了两年多时间编写了这本《Linux 程序设计实践与编程技巧》一书,书中部分内容选取浙江大学计算机学院、软件学院 2006—2012 级学生的作业,在此对这些同学表示感谢。希望本书能够对学习 Linux 程序设计的本科生、研究生、嵌入式工程技术人员及 Linux 程序爱好者提供帮助。

通过本书的学习,可以掌握以下内容:

- Linux 基本操作技巧;
- Shell 程序设计技巧;
- makefile 工程文件;
- Linux 环境下 C 语言编辑、编译、调试方式与技巧;



- Linux 环境下文件的 I/O 操作技巧;
- 进程的控制与进程调度技巧;
- 线程编程技巧;
- Linux 环境下的图形程序、游戏程序设计技巧;
- 网络程序设计技巧;
- 串行案例设计技巧;
- 典型案例设计技巧。

本书在讲解讨论内容时,首先给出实例,通过例子介绍程序设计的方法,通过大量的例子与清晰的程序流程使读者迅速掌握相关知识、编程技能与技巧,并通过大量的思考题帮助读者迅速提高程序设计能力。

本书中 16 个实验的设计,更是符合人们学习程序设计的心理特点,在每一个实验中首先给出程序的调试,在程序调试的基础上再根据给出的问题,对程序的关键语句学会自己设计,最后给出程序设计项目,培养读者完全独立设计程序的能力。如果比喻着学习“走路”,这相当于在每个实验中完成了让您看着别人走路、扶着您走路、让您独立行走的全过程,相信这 16 个实验会极大提高读者的程序设计能力。

本书由浙江大学城市学院刘加海教授,上海交通大学航空航天学院骆建华教授,浙江大学计算机学院季江民副教授,浙江大学城市学院严冰副教授、杨锐讲师、王晓池讲师,浙江大学宁波理工学院唐云廷副教授编写,全书由刘加海统稿。由于时间仓促及作者水平有限,书中难免存在疏漏和不妥之处,敬请广大读者批评指正,批评与建议请发到邮件 Ljhqyyq@yahoo.com.cn,以便及时修订。



目 录

CONTENTS

程序设计要点与技巧

第 1 章 基本命令 // 3

- 1.1 知识要点 // 3
 - 1.1.1 常用终端命令 // 3
 - 1.1.2 常用元字符*、?、~、[]的应用 // 7
 - 1.1.3 重定向符|、>>、>、< // 8
 - 1.1.4 单引号与双引号的作用 // 9
 - 1.1.5 文本编辑命令 vi 或 vim 的应用 // 10
 - 1.1.6 Linux 常用系统变量 // 11

第 2 章 Shell 编程 // 13

- 2.1 知识要点 // 13
 - 2.1.1 Shell 程序运行方式 // 13
 - 2.1.2 Shell 中变量的使用 // 13
 - 2.1.3 Shell 编程中参数替换 // 14
 - 2.1.4 Shell 编程中大段文字显示技巧 // 14
 - 2.1.5 随机数产生 // 15
 - 2.1.6 日期输出格式设置 // 15



- 2.1.7 在 Shell 中的算术表达方法 // 16
- 2.1.8 Shell 程序设计中分支语句 // 18
- 2.1.9 Shell 程序设计中循环 // 19
- 2.1.10 Shell 中函数的格式 // 20
- 2.1.11 Shell 中数组的使用 // 21
- 2.1.12 文件与文件夹的判断 // 22
- 2.1.13 某路径下文件总容量的判断 // 23
- 2.1.14 菜单程序的框架 // 23
- 2.2 程序设计实例 // 24

第 3 章 Linux 系统 C 语言开发工具 // 32

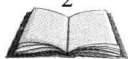
- 3.1 知识要点 // 32
 - 3.1.1 gcc 的使用 // 32
 - 3.1.2 makefile 工程文件 // 33
- 3.2 程序设计实例 // 35

第 4 章 Linux 环境下系统函数的使用 // 37

- 4.1 知识要点 // 37
 - 4.1.1 随机函数的应用 // 37
 - 4.1.2 系统时间函数的应用 // 38
 - 4.1.3 系统函数 // 39
 - 4.1.4 数据结构中的函数 // 40
- 4.2 程序设计实例 // 41

第 5 章 Linux 环境下文件 I/O 操作 // 48

- 5.1 知识要点 // 48
 - 5.1.1 文件操作 // 48
 - 5.1.2 文件控制特性的判断 // 51
- 5.2 程序设计实例 // 53





第6章 进程控制 // 61

- 6.1 知识要点 // 61
 - 6.1.1 进程调度命令 at // 61
 - 6.1.2 进程创建函数 fork // 62
 - 6.1.3 僵尸进程 // 63
 - 6.1.4 wait 与 waitpid 函数 // 64
 - 6.1.5 僵尸进程的避免 // 65
 - 6.1.6 守护进程 // 65
- 6.2 程序设计实例 // 66

第7章 进程间的通信 // 78

- 7.1 知识要点 // 78
 - 7.1.1 Linux 进程间的通信方式 // 78
 - 7.1.2 进程间通信的特点 // 78
 - 7.1.3 管道通信的函数 // 79
 - 7.1.4 信号传送和处理 // 80
 - 7.1.5 消息队列应用 // 81
 - 7.1.6 共享内存函数 mmap 应用 // 83
- 7.2 程序设计实例 // 84

第8章 线程 // 99

- 8.1 知识要点 // 99
 - 8.1.1 线程与进程 // 99
 - 8.1.2 多线程和多进程的对比 // 100
 - 8.1.3 线程中的常用函数 // 100
 - 8.1.4 线程中互斥锁的实现 // 101
 - 8.1.5 线程中信号量的应用 // 102
- 8.2 程序设计实例 // 103





第9章 网络程序设计 // 113

9.1 知识要点 // 113

9.1.1 Socket 接口 // 113

9.1.2 Sockaddr 和 Sockaddr_in 结构类型 // 113

9.1.3 TCP 编程 // 114

9.1.4 UDP 编程 // 115

9.1.5 常用操作函数 // 116

9.2 程序设计实例 // 119

第10章 Linux 图形程序设计 // 134

10.1 知识要点 // 134

10.1.1 SDL 库 // 134

10.1.2 常用 SDL 库函数 // 135

10.2 程序设计实例 // 137

第11章 设备驱动程序设计基础 // 147

11.1 知识要点 // 147

11.1.1 设备驱动程序概况 // 147

11.1.2 字符设备驱动程序 // 148

11.2 程序设计实例 // 150

第12章 串行通信 // 162

12.1 知识要点 // 162

12.1.1 串行通信 // 162

12.1.2 串行通信程序设计流程 // 163

12.1.3 串行通信程序设计步骤 // 163

12.2 程序设计实例 // 165





实践部分

- Linux 程序设计实验报告 1**——操作系统基本命令使用 // 175
- Linux 程序设计实验报告 2**——Shell 程序设计 1 // 180
- Linux 程序设计实验报告 3**——Shell 程序设计 2 // 185
- Linux 程序设计实验报告 4**——Linux 系统 C 开发工具 // 190
- Linux 程序设计实验报告 5**——Linux 环境系统函数的应用 // 198
- Linux 程序设计实验报告 6**——Linux 文件 I/O 操作 1 // 206
- Linux 程序设计实验报告 7**——Linux 文件 I/O 操作 2 // 212
- Linux 程序设计实验报告 8**——进程控制 // 224
- Linux 程序设计实验报告 9**——进程通信 1 // 233
- Linux 程序设计实验报告 10**——进程通信 2 // 240
- Linux 程序设计实验报告 11**——Linux 线程程序设计 // 251
- Linux 程序设计实验报告 12**——Linux 网络程序设计 // 259
- Linux 程序设计实验报告 13**——Linux 图形程序设计 // 267
- Linux 程序设计实验报告 14**——虚拟字符驱动程序设计 // 280
- Linux 程序设计实验报告 15**——Linux 串行通信程序设计 // 289
- Linux 程序设计实验报告 16**——Windows 与 Linux 操作系统间资源共享 // 303



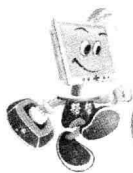


程序设计要点与技巧

Linux程序设计实践与编程技巧

第 1 章

基本命令



1.1 知识要点



1.1.1 常用终端命令

1. 在 Linux 中，字符是区分大小写的。

2. 建立目录命令 `mkdir`

例如，在目录 `/mnt` 下建立文件夹 `usb`

```
mkdir /mnt/usb
```

3. 在目录中搜索文件 `find`

```
find [path][options][expression]
```

常用参数：

—name: 支持通配符*和?。

—atime *n*: 搜索在过去 *n* 天读取过的文件。

—ctime *n*: 搜索在过去 *n* 天修改过的文件。

—group groupname: 搜索所有组为 groupname 的文件。

—user 用户名: 搜索所有文件属主为用户名 (ID 或名称) 的文件。

—size *n*: 搜索文件大小是 *n* 个 block 的文件。

—print: 输出搜索结果，并且打印。



应用技巧:

(1) 根据文件名查找

例如, 想要查找一个文件名是 `lilo.conf` 的文件, 可以使用如下命令:

```
find / -name lilo.conf
```

`find` 命令后的参数 “/” 表示搜索整个硬盘, 例如搜索 `/etc` 路径下的文件名 `smb.conf` 应用参数。

```
find /etc -name smb.conf
```

(2) 根据部分文件名查找方法

例如, 想要查找文件名中包含字符 `abvd` 的所有文件。

```
find / -name '*abvd*'
```

在/目录中查找所有的包含有 `abvd` 这 4 个字符的文件。

(3) 使用混合查找方式查找文件

例如, 在 `/etc` 目录中查找大于 500000 字节, 并且在 24 小时内修改的某个文件, 则可以使用 `-and` (与) 把两个查找参数链接起来组合成一个混合的查找方式。

```
find /etc -size +500000c -and -mtime +1
```

(4) 复制文件夹时, 使用命令 `cp`, 要加上参数 `-r`, 才能递归复制, 不然只会将目录下的文件复制过来, 而目录下的文件夹不会复制过来。

4. 修改密码

Linux 终端里使用 `passwd` 命令修改密码可以很方便地修改密码:

```
[root@localhost root]# passwd
```

Changing password for your_username

New password: 新密码

Retype new password: 再输入一次新密码

Passwd: all authentication tokens updated successfully

如果忘记了密码, 可以试图使用以下 2 种方法解决问题:

(1) 在系统进入单用户状态, 直接用 `passwd root` 去更改;

(2) 用安装光盘引导系统, 进行 `linux rescue` 状态, 将原来/分区挂载做法如下:

```
cd /mnt
```

```
mkdir hd
```

```
mount -t auto /dev/hdaX(原来/分区所在的分区号) hd
```

```
cd hd
```

```
chroot ./
```

```
passwd root
```

5. 文件权限设置命令 `chmod`

使用方式:

```
chmod [-cfvR] [--help] [--version] mode file...
```

说明:

Linux/Unix 的档案存取权限分为三级: 档案拥有者, 群组, 其他。利用 `chmod` 可以控制档案如何被他人所存取。





mode: 权限设定字符串,格式如下: [ugoa...][[+|=][rwxX]...][, ...], 其中 **u** 表示该档案的拥有者, **g** 表示与该档案的拥有者属于同一个群体(group)者, **o** 表示其他以外的人, **a** 表示这三者皆是。

+ 表示增加权限, - 表示取消权限, = 表示唯一设定权限。

r 表示可读取, **w** 表示可写入, **x** 表示可执行, **X** 表示只有当该档案是子目录或者该档案已经被设定过为可执行。

-c: 若该档案权限确实已经更改, 才显示其更改动作。

-f: 若该档案权限无法被更改也不要显示错误讯息。

-v: 显示权限变更的详细资料。

-R: 对目前目录下的所有档案与子目录进行相同的权限变更(即以递归的方式逐个变更)。

-help: 显示辅助说明。

-version: 显示版本。

例如:

将档案 **file1.txt** 设为所有人皆可读取:

```
chmod ugo+r file1.txt
```

或

```
chmod a+r file1.txt
```

将档案 **file1.txt** 与 **file2.txt** 设为该档案拥有者, 与其所属同一个群体者可写入, 但其他以外的人则不可写入:

```
chmod ug+w,o-w file1.txt file2.txt
```

将 **ex1.py** 设定为只有该档案拥有者可以执行:

```
chmod u+x ex1.py
```

将目前目录下的所有档案与子目录皆设为任何人可读取:

```
chmod -R a+r *
```

此外 **chmod** 也可以用数字来表示权限如 **chmod 777 file**

语法为: **chmod abc file**

其中 **a**、**b**、**c** 各为一个数字, 分别表示 User, Group, 及 Other 的权限。

r=4, **w=2**, **x=1**

若为 **rwX** 属性则 $4+2+1=7$;

若为 **rw**-属性则 $4+2+0=6$;

若为 **r-x** 属性则 $4+0+1=5$ 。

例如:

```
chmod a=rwx file
```

和

```
chmod 777 file
```

效果相同

```
chmod ug=rwx,o=x file
```

和





```
chmod 771 file
```

效果相同

若用 `chmod 4755 filename` 可使此文件具有 root 的权限。

6. 磁盘挂载

在 Linux 中所有的设备都以文件的形式显示在 `/dev` 当中，对于老的 PATA 的硬盘，会以 `hda` 开头，对于目前采用比较多的 SATA 硬盘会以 `sda` 开头，光驱是以 `cdrom` 开头的，`cciss` 是惠普的磁盘阵列开头。

如果需要使用一个设备，可以通过 `mount` 命令来挂载设备。

例如：

```
mount -t 文件格式 /dev/cciss/c0d0p2 /home
```

对于 Linux 中基本是 `ext` 格式，较新的系统会采用 `ext 4` 格式，老一些不支持 `ext 4` 的版本会采用 `ext 3` 格式。

7. 回收站的位置

回收站的位置用 “`~/.Trash/`” 表示。

8. 下载命令 wget

可以下载互联网上面的文件到本地，格式为：

```
wget URL地址
```

例如：

```
$wget http://www.rarlab.com/rar/rarlinux-3.8.0.tar.gz
```

9. 时间显示与时间设置的命令 date

```
date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
```

查询：

```
[root@localhost root]# date
```

```
日 12月 9 07:09:14 CST 2012
```

修改：

```
[root@localhost root]# date 12242322.21
```

```
一 12月 24 23:22:21 CST 2012
```

date命令的使用：

直接输入 `date` 命令，将显示当前的系统时间。

使用 “`date 月日時分年.秒`” 的格式改变当前系统时间

使用 “`date+格式`” 按格式输出系统时间

常用的格式有：

`%a,%A`：输出星期几（`%a` 为简写，`%A` 为全称）

`%b,%B`：输出月份名（`%b` 为简写，`%B` 为全称）

`%d`：一个月内的第几天

`%H`：小时

`%m`：月份

`%M`：分钟

`%N`：纳秒数（可以用来生成随机数）





%S: 秒

%Y: 年

例如:

```
[root@localhost root]# date +%Y-%m-%d
2012-12-10
```

在 Linux 中有硬件时钟与系统时钟等两种时钟。date 可以改变系统时钟, 而 hwclock 可以改变硬件时钟。例如:

```
[root@localhost root]# date
一 12月 24 23:46:36 CST 2012
[root@localhost root]# hwclock
2012年12月09日 星期日 07时35分19秒 -0.771641 seconds
[root@localhost root]#
```

思考执行结果:

```
t_date=$(date +%m%d)
n_date=$(date +%w)
```

10. 检查磁盘命令 fdisk

检查磁盘, 列出结果中有关/dev/sd的磁盘信息

```
/sbin/fdisk -l |grep /dev/sd
```

11. 终止进程命令 kill

例如: 在后台执行编辑命令vi, 然后应用命令kill终止vi的执行。

```
[root@localhost root]# vi &
[2] 22633
[1]+  Stopped                  vim
[root@localhost root]# kill 9 -22633
[2]+  Stopped                  vim
Vim: CVim: 拦截到信号(signal) TERM
Vim: 结束.
[root@localhost root]#
```



1.1.2 常用元字符*、?、~、[]的应用

通配符是由 shell 处理的, 它只出现在命令的“参数”里。当 shell 在“参数”中遇到了通配符时, shell 会将其当作路径或文件名去在磁盘上搜寻可能的匹配。若符合要求的匹配存在, 则进行代换(路径扩展), 否则就将该通配符作为一个普通字符传递给“命令”, 然后再由命令进行处理。