

COMPUTER

高等院校计算机技术



“十二五”规划教材

Linux 程序设计

◎ 严 冰 刘加海 季江民 主编



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

严 冰 刘加海 季江民 主编



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

内容简介

本书内容包括:Linux 终端命令、Shell 程序设计, Linux 系统 C 语言开发工具, Linux 环境下系统函数的使用, 文件 I/O 操作, 进程控制与进程间的通信, 线程及线程同步互斥, Linux 网络程序设计, Linux 图形程序设计, 设备驱动程序设计基础, 终端 I/O 通信, 最后给出 9 个主题的程序设计实例。

本书结构合理、概念清楚、由浅入深、实用性强, 是一本技能型的 Linux 程序设计教材, 适合 Linux 环境下软件工程技术使用, 也可以作为软件、计算机等专业本、专科学生的 Linux 程序设计、嵌入式程序设计教材。

图书在版编目 (CIP) 数据

Linux 程序设计 / 严冰, 刘加海, 季江民主编. —杭州:
浙江大学出版社, 2012. 2
ISBN 978-7-308-09569-3

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

中国版本图书馆 CIP 数据核字 (2012) 第 009678 号

Linux 程序设计

严 冰 刘加海 季江民 主编

责任编辑 周卫群

封面设计 刘依群

出版发行 浙江大学出版社

(杭州市天目山路 148 号 邮政编码 310007)

(网址: <http://www.zjupress.com>)

排 版 杭州中大图文设计有限公司

印 刷 浙江省良渚印刷厂

开 本 787mm×1092mm 1/16

印 张 31.25

字 数 800 千

版 次 2012 年 2 月第 1 版 2012 年 2 月第 1 次印刷

书 号 ISBN 978-7-308-09569-3

定 价 59.00 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

Linux 操作系统的应用领域越来越广泛, Linux 早已涉足政府、军队、金融、电信和证券等方方面面。随着 Linux 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, Linux 操作系统给世界各国都提供了一个重要机会, 具有 Linux 专业知识的及开发经验的人才越来越受到用人单位的欢迎。本书 Linux 程序设计知识涵盖面广, 内容安排合理。从 Linux 终端命令、Shell 程序设计、Linux 环境下 C 语言开发工具、系统函数的使用、文件 I/O 操作、进程控制与进程间的通信、线程及线程同步互斥、网络程序设计、图形程序设计、设备驱动程序、终端 I/O 通信, 到最后给出 9 个主题的程序设计实例, 把许多工程开发中的知识基本都涉及到了, 是一本非常不错的 Linux 编程入门与工程应用参考书。

为了致力于开源人才培养, 为了改变“linux 是专业人士的事情”的看法, 本书的读者定位于有志于 Linux 程序设计的初学者和中级 C 程序员。本书概念清晰、论述简单明了、由浅入深, 容易理解, 以完整案例为引导, 采用非常丰富的实例, 讲解 Linux 环境下 C 程序设计的精华, 并且这些案例来自实际工程, 因而在实际开发时可以直接引用。对于刚接触 Linux 程序开发的技术人员既很有帮助也很实用。

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。此后根据多年的教学经验曾给 LUPA(Linux 高校推进联盟)写了《Linux 软件工程师(C 语言)实用教材》一书, 由倪光南院士写序, 在读者中深受欢迎。根据教学过程中发现的问题及教学的需要, 在《Linux 软件工程师(C 语言)实用教材》基础上重新编写了《Linux 程序设计》一书。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;
- Shell 程序设计;
- Linux 环境下 C 程序编辑、编译、调试方法;
- 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 环境下的图形程序、游戏程序设计;
- 网络程序设计;
- 串行程序设计;
- 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。

本书由浙江大学城市学院严冰、刘加海、计算机学院季江民编写,全书由刘加海统稿。

本书备有教学大纲与教学课件,并希望在近来出版与本书配套的《Linux C 程序设计实验与辅导》一书,在实践一书给出 17 周的 C 程序设计的实验、重点与难点提示、习题解答提示等。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。

第 1 章 Linux 基本命令	1
1.1 shell 基础	1
1.2 shell 命令基础	3
1.2 文件系统命令	11
1.3 文件管理工具	24
1.4 文件权限与文件共享	27
1.5 进程、管道和重定向	32
思考与实验	39
第 2 章 shell 编程	44
2.1 bash 脚本的建立和运行	44
2.1.1 bash 脚本的建立	44
2.1.2 运行 bash 脚本	45
2.2 shell 的变量	45
2.2.1 环境变量和用户定义变量	45
2.2.2 变量声明和赋值	46
2.2.3 变量引用和引号使用	48
2.2.4 命令替换	49
2.2.5 输入命令	49
2.3 shell 脚本位置参数的传递	50
2.4 控制结构语句	52
2.4.1 if-then-elif-else-fi 语句	52
2.4.2 for 语句	54
2.4.3 while 语句	56
2.4.4 until 语句	57
2.4.5 case 语句	58
2.5 其他几个有用的语句	59
2.5.1 break 和 continue 语句	59
2.5.2 exit 语句	59
2.5.3 printf 语句	60
2.6 数值处理	61

2.6.1	let 命令	62
2.6.2	\$((expression))扩展	62
2.6.3	expr 命令	63
2.7	数组	65
2.8	函数	67
2.9	调试脚本程序	70
2.10	综合实例	71
	思考与实验	82
第3章	Linux 系统 C 语言开发工具	84
3.1	第一个 Linux C 语言程序	84
3.2	编译 C 语言程序	85
3.2.1	gcc 编译器	85
3.2.2	函数库	87
3.3	make 工具	90
3.3.1	make 命令	90
3.3.2	make 规则	90
3.3.3	Makefile 中的变量	92
3.4	gdb 调试工具	93
	思考与实验	100
第4章	Linux 环境下系统函数的使用	103
4.1	数学函数的使用	103
4.2	字符函数的使用	107
4.3	系统时间与日期函数的使用	109
4.4	环境控制函数	113
4.5	内存分配函数	116
4.6	数据结构中常用函数	121
	思考与实验	125
第5章	文件 I/O 操作	126
5.1	Linux 系统文件的属性	126
5.1.1	Linux 文件类型	127
5.1.2	Linux 文件权限	129
5.1.3	Linux 文件的其他属性	133
5.2	不带缓存的文件 I/O 操作	135
5.2.1	文件的创建	135
5.2.2	文件的打开和关闭	136
5.2.3	文件的读写操作	138
5.2.4	文件上锁	144

5.3 带缓存的流文件 I/O 操作	149
5.3.1 流文件的打开和关闭	150
5.3.2 流文件的读写操作	152
5.3.3 文件的定位	158
5.4 特殊文件的操作	160
5.4.1 目录文件的操作	160
5.4.2 链接文件的操作	165
思考与实验	167
第 6 章 进程控制	170
6.1 进程简介	170
6.2 Linux 进程控制	171
6.2.1 进程的相关函数	173
6.2.2 进程创建	173
6.2.3 进程终止	180
6.2.4 僵尸进程	183
6.3 Linux 守护进程	193
6.3.1 守护进程及其特性	193
6.3.2 编写守护进程的要点	194
6.3.3 守护进程的编写	196
思考与实验	202
第 7 章 进程间的通信	204
7.1 进程间的通信	204
7.2 信号	205
7.2.1 信号及其使用简介	205
7.2.2 信号操作的相关函数	207
7.3 管道	218
7.3.1 低级管道操作	219
7.3.2 命名管道	223
7.3.3 高级管道操作	228
7.4 消息队列	231
7.5 共享内存	235
7.5.1 内存映射	235
7.5.2 UNIX System V 共享内存	238
思考与实验	246
第 8 章 线程	248
8.1 线程概述	248
8.2 线程创建	249

8.3 线程同步与互斥	255
8.3.1 互斥锁	255
8.3.2 条件变量	259
8.3.3 信号量	263
思考与实验	269
第9章 网络程序设计	272
9.1 TCP/IP 简介	272
9.1.1 TCP/IP 概述	272
9.1.2 TCP/IP 模块结构	273
9.2 网络编程	274
9.2.1 端口	274
9.2.2 socket 端口	274
9.2.3 socket 套接口	274
9.2.4 socket 套接口的数据结构	275
9.2.5 基于 TCP 协议的客户端/服务器程序的常用函数	276
9.2.6 TCP 编程	276
9.2.7 UDP 编程	288
9.3 网络高级编程	295
思考与实验	305
第10章 Linux 的图形编程	317
10.1 Linux 的图形编程简介	317
10.2 安装和使用 SDL 图形开发库	318
10.3 初始化图形模式	318
10.4 基本绘图函数的应用	324
10.5 图片与文字显示	335
10.6 动画	342
10.7 三维绘图	347
10.8 游戏程序入门	354
思考与实验	365
第11章 设备驱动程序设计基础	377
11.1 设备驱动程序的概念	377
11.1.1 设备的分类	377
11.1.2 设备文件的查看	378
11.1.3 主设备号与次设备号	378
11.1.4 设备驱动相关的数据结构	378
11.2 驱动程序的设计流程	379
11.2.1 字符驱动程序设计流程	379

11.2.2 驱动程序流程设计举例.....	380
11.3 Linux 字符设备驱动程序设计	383
11.3.1 字符设备驱动程序数据结构.....	383
11.3.2 字符设备驱动程序的基本框架.....	387
11.4 字符设备驱动程序实例——虚拟字符设备.....	389
11.4.1 结构体设计.....	390
11.4.2 设备驱动读、写函数的设计	390
11.4.3 字符设备驱动程序设计步骤.....	392
11.4.4 字符设备驱动程序测试.....	394
思考与实验.....	396
第 12 章 串行通信	401
12.1 串行通信概述.....	401
12.1.1 同步通信.....	401
12.1.2 异步通信.....	402
12.2 串行通信程序的设计.....	402
12.2.1 串行通信程序设计流程.....	402
12.2.2 打开通信端口.....	403
12.2.3 设置串口属性.....	404
12.2.4 串口通信程序设计.....	406
思考与实验.....	413
第 13 章 程序设计实例	414
13.1 Shell 程序设计实例	414
13.2 系统函数的应用实例.....	418
13.3 文件操作.....	423
13.4 进程控制实例.....	429
13.5 进程通信.....	434
13.6 网络程序设计.....	445
13.7 图形程序设计.....	462
13.8 驱动程序设计.....	478
13.9 串口通信程序设计.....	480
附录 SDL 库的安装	483

第 1 章 Linux 基本命令

本章重点

1. shell 的基本概念
2. Linux 环境 bash 的基本命令

本章导读

Linux 操作系统是目前最流行的操作系统之一,广泛应用于嵌入式系统中,它是一个基于 GNU 的自由软件(free software)。

本章主要介绍基本的 Linux shell 命令。

1.1 shell 基础

每个 Linux 系统发行版本中都包含了多种 shell。目前使用最为广泛的 shell 包括 bash、TC shell 和 Korn shell。通常默认的登录 shell 是 bash。系统管理员可以为你指定使用哪种 shell 作为登录 shell,也可以通过命令来改变自己的默认登录 shell。比如说,如果你的默认登录 shell 是 bash,但是你更喜欢用 TC shell,你可以通过命令 tcsh 或者 chsh 来改变默认登录 shell。

各种发行版本的 Linux 系统中并不一定把所有的 shell 都安装在系统中,作者使用的 fedora core 系统中最常用的几种 shell 如表 1.1 中列出。各种 shell 程序放在/bin/目录下。

表 1.1 常用 shell 程序

shell 名称	存放的位置	程序名
Bourne shell	/bin/sh->bash	bash
Bourne Again shell	/bin/bash	bash
C shell	/bin/csh->tcsh	tcsh
TC shell	/bin/tcsh	tcsh
Korn shell	/bin/ksh	ksh

1. shell 的环境变量

shell 环境变量具有特殊的意义,它们的名字一般比较短,bash 的环境变量名通常由大写字母组成。

用户在任何时候都可以更改大多数 shell 环境变量的值,如果需要修改 bash 环境变量的

值,就在初始化文件/etc/profile 和/etc/csh.cshrc 中进行修改。用户可以将用户创建的变量变成全局变量,同样,可以将环境变量变成全局变量,这个工作也在初始化文件中自动完成。

例如,在搜索路径中增加了两个目录,~/bin 和.(“.”表示当前目录)。而且使~/bin 成为最先被搜索的目录,而当前目录则成为最后被搜索的目录,可使用以下命令与环境参数:

```
[root@localhost root]# PATH = ~/bin: $ PATH:.
```

表 1.2 列出了部分环境变量,更多的环境变量将在第 2 章中给出。

表 1.2 部分 bash 环境变量

环境变量名	含义
CDPATH	cd 命令访问的目录的别名
EDITOR	用户在程序中使用的默认的编辑器
ENV	Linux 查找配置文件的路径
HOME	主目录的名字
PATH	存放搜索命令或者程序的所有目录
PS1	shell 提示符
PS2	shell 的二级提示符
PWD	当前工作目录的名字
TERM	用户使用的控制台终端的类型

2. shell 元字符

除了字母和数字,很多其他字符对于 shell 都有特殊的含义。这些字符被称为 shell 元字符(shell metacharacters),如表 3 所示。如果不以特殊方式指明,在 shell 命令中,这些字符不能作为文本字符使用。所以,不要在文件名中使用这些字符。而且在命令中使用这些字符时,不需要在它们的前面或者后面加上空格。

表 1.3 shell 中的元字符及其作用

元字符	功能
回车换行	把命令输入后要按回车键
空格	命令行中的分隔符
TAB	命令行中的分隔符
#	以#开头是注释行
“	引用多个字符,但是允许替换
‘	引用多个字符,括号中字符按原义解释
\$	表示一行的结束,或引用变量时使用
&	使命令在后台执行
()	在子 shell 中执行命令
[]	匹配[]中一个字符

续表

元字符	功能
{ }	在当前 shell 中执行命令,或实现扩展
*	匹配 0 个或者多个字符
?	匹配单个字符
-	紧跟后面的字符开始的行,或作为否定符号
.	替换命令
	管道符
;	顺序执行命令的分隔符
<	输入重定向符号
>	输出重定向符号
/	用作根目录或者路径名中的分割符
\	转义字符;转义回车换行字符;或作为续行符
!	启动历史记录列表中的命令和当前命令
%	指定一个作业号时作为起始字符
~	表示主目录

shell 元字符允许你在一个命令行中指定若干个目录中的若干个文件。先只给出一些简单的例子解释一些常用元字符的含义,如 *, ?, ~ 和 []。

字符 ? 是一个匹配任何单个字符的通配符。

字符 * 则匹配 0 个或者多个字符。

符号 [] 表示区间中的任一字符。

例:字符串“?.txt”可以用来表示一个字符后跟“.txt”的所有文件名,如:a.txt, l.txt, @.txt。

例:字符串“lab1 \ / c”表示 lab1/c。注意,在这里用反斜线号(\)来处理“消除了特殊意义”的斜线号(/)。

例:下面的这条命令显示当前目录中所有由 2 个字符组成,且以.html 为结尾的文件。而且这些文件名的第一个字符是数字,第二个字符是大写或者小写的字母。

```
[root@localhost root]# ls [0-9][a-zA-Z].html
```

在这里,[0-9]表示从 0 到 9 的任何数字,[a-zA-Z]表示任何大写或者小写的字母。

1.2 shell 命令基础

1. 修改密码

使用 passwd 命令修改密码。下面是 passwd 命令示例,your_username 是你的登录名。

```
[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

2. 获取帮助

Linux 系统的发行版通常没有纸质的参考手册。但是 Linux 提供了详尽的联机帮助文档。可以用 man 和 info 工具得到 Linux 的联机帮助文档,也可通过 Internet 找到 Linux 的各种帮助信息。

使用--help 选项获取帮助

大多数 GNU 工具或命令都有--help 选项,用来显示使用命令的一些帮助信息。使用 ls 命令的帮助信息如下所示:

```
$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of-cftuSUX nor--sort.

Mandatory arguments to long options are mandatory for short options too.
-a,--all                do not hide entries starting with .
-A,--almost-all        do not list implied . and ..
    --author             print the author of each file
-b,--escape             print octal escapes for nongraphic characters
    --block-size=SIZE   use SIZE-byte blocks
-B,--ignore-backups     do not list implied entries ending with ~
.....
```

如果显示的信息超出了一个屏幕,可以通过管道使用 more 程序分屏显示帮助信息。例如:

```
[root@localhost root]# ls--help | more
```

man 命令

man 命令可用来访问在线手册页。通过查看 man 页可以得到有关程序或命令的更多相关主题信息和 Linux 的更多特性。以下是 man 命令语法。

命令语法:

```
man [options] command-list
```

常用选项:

-S 'section' 指定 man 命令所查找章节号为“section”文档

用 man 命令显示这些文档,根据主题分成八个章节。表 1.4 列出了手册的八个章节和它们包含的内容。

表 1.4 Linux 帮助手册的章节

节	描述
1	用户命令
2	系统调用
3	语言函数库调用(C,C++等)
4	设备和网络界面
5	文件格式
6	游戏和示范
7	troff 的环境,表格和宏
8	关于系统维护的命令

帮助手册对每个 Linux 命令、系统调用以及库函数调用都有描述。一个命令往往对应多页格式化描述。这里的格式由七个部分组成:名字、概要、描述、文件列表、相关信息、错误/警告和已知 bug。用户可以使用 man 命令来阅读帮助手册。在屏幕上显示一页帮助手册时,这页的左上角显示命令名,以及在括弧中的命令所属的章节号,就像 ls(1)。

帮助手册是多页文本文档,每个主题的帮助手册需要一个以上的满屏文本来显示全部内容。按键盘上的空格键,可以一次一满屏地显示帮助手册。按<Q>键退出浏览帮助手册。

info 命令

GNU 软件和其他一些自由软件还使用名为 info 的在线文档系统。可以通过特殊的程序 info 或通过 emacs 编辑器中的 info 命令来在线浏览全部的文档。对文档作者来说,info 系统的优点是它的文件可以由排版印刷文档使用的同一个源文件自动生成。

例:当输入 info passwd 命令后,屏幕显示如下内容:

```
File: * manpages * , Node: passwd, Up: (dir)
```

```
PASSWD(1)                               User utilities                               PASSWD(1)
NAME
    passwd-update a user's authentication tokens(s)
SYNOPSIS
    passwd [-k] [-l] [-u [-f]] [-d] [-S] [username]
DESCRIPTION
    Passwd is used to update a user's authentication token(s).
    Passwd is configured to work through the Linux-PAM API. Essentially,
    it initializes itself as a "passwd" service with Linux-PAM and utilizes
    configured password modules to authenticate and then update a user's
    password.
.....
```

由于屏幕上的信息来自于可编辑文件,所以不同的系统显示结果可能有所不同。当看到 info 上面的初始屏幕后,可以使用各种 info 命令。下面列出几个最常用键盘命令:

- <? >或<CTRL-H>键:列出 info 命令
- <SPACE>键:滚动翻页
- <Q>键:退出

info 系统包含它自己的一个 info 形式的帮助页。如果按下<? >或<Ctrl+H>键,将看到一些帮助信息,其中包括如何使用 info 的指南。

3. 获取用户和系统信息的命令

在登录到 Linux 系统时,你要了解用户 id,你登录上的计算机或系统,以及那台计算机上的操作系统的信息。这些工作可以通过下面的命令完成:

whoami 命令:在屏幕上显示你的用户 id。

hostname 命令:显示登录上的主机的名字。

uname 命令:显示关于运行在计算机上的操作系统的信息。

下面的会话显示了在命令行上键入这些命令时系统是怎样回答它们的。

```
[root@localhost root]# whoami
root
[root@localhost root]# hostname
localhost.localdomain
[root@localhost root]# uname
Linux
```

4. 显示系统运行时间

可以用 uptime 命令显示系统的运行时间(从最近一次启动开始,系统已经运行的时间)和其他一些有用的统计数据,比如当前系统中有多少登录的用户。这个命令并不需要任何参数。下面的这个例子显示了这条命令的输出。

```
[root@localhost root]# uptime
9:43am up 58 min, 1 users, load average: 0.4, 0.12, 0.17
```

5. 显示日期和时间

可以用 date 命令来显示当前的日期和时间,超级(root)用户可以使用 date 命令来修改系统时钟。

例:显示当前的时间和日期,如下所示:

```
[root@localhost root]# date
五 9 月 28 16:19:27 UTC 2011
```

例:将时间设置为 9 月 4 日的下午 14:20:15,不改变年份。

```
[root@localhost root]# date 0941420.15
二 9 月 4 14:20:15 UTC 2011
```

6. vi / vim 文本编辑器

vi 是 Linux/Unix 世界里最常用的全屏编辑器,所有的 Linux 系统都提供该编辑器,而 Linux 也提供了 vi 的加强版——vim,同 vi 是完全兼容,存放路径为/usr/bin/vim,vim 软件及

有关信息可以从 www.vim.org 获得。vi 虽然不易学习,但它强大的功能和高度灵活性,与操作系统的兼容性,却是最好的,而且是 UNIX 类操作系统使用人数最多的文本编辑器。

多数的 Linux 系统发行版本中 vi 命令是 vim 的别名,你可以通过 alias 命令或 which vi 命令查看一下,所以,当启动 vi 命令时,实际运行的是 vim 程序。在这里,不对 vi 和 vim 加以区别,统一使用 vi 命令。

vi 文本编辑器的命令语法如下:

命令语法:vi [options] [filename]

常用选项:

- + n 从第 n 行开始编辑文件
- + /exp 从文件中匹配字符串 exp 的第一行开始编辑

vi 中的操作主要有两类模式:

- 命令模式(command mode),由键盘命令序列(vi 编辑器命令)组成,完成某些特定动作。
- 插入模式(insert mode),允许你输入文本。

图 1.1 说明 vi 文本编辑器的一般结构,说明如何在模式间进行切换。在 vi 中执行的键盘命令是大小写敏感的,例如:大写的<A>可在当前行末尾的最后一个字符后添加新文本,而小写的<a>则在当前光标所在字符后添加新文本。

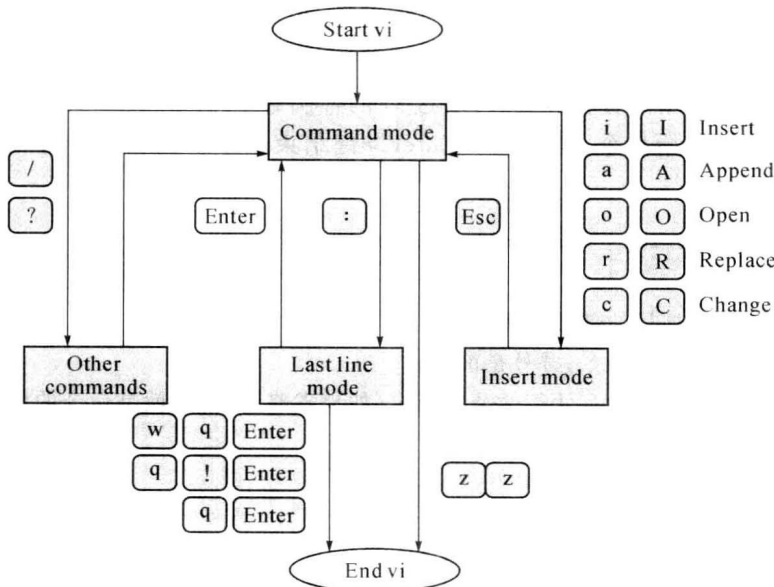


图 1.1 vi 文本编辑器的操作模式

vi 的进入与退出

在系统提示符下键入命令 vi,后面跟上你要编辑或创建的文件名,vi 自动装入所要编辑的文件或是开启一个新文件。

退出 vi 编辑器,可以在命令行方式下使用命令“:wq”或者“:q!”,前者的功能是写文件并从 vi 中退出,后者的功能是从 vi 中退出,但不保存所作的修改。

vi 的插入方式

vi 编辑器预设的是以命令模式开始。当从命令模式转变成插入模式时有三个键,分别是