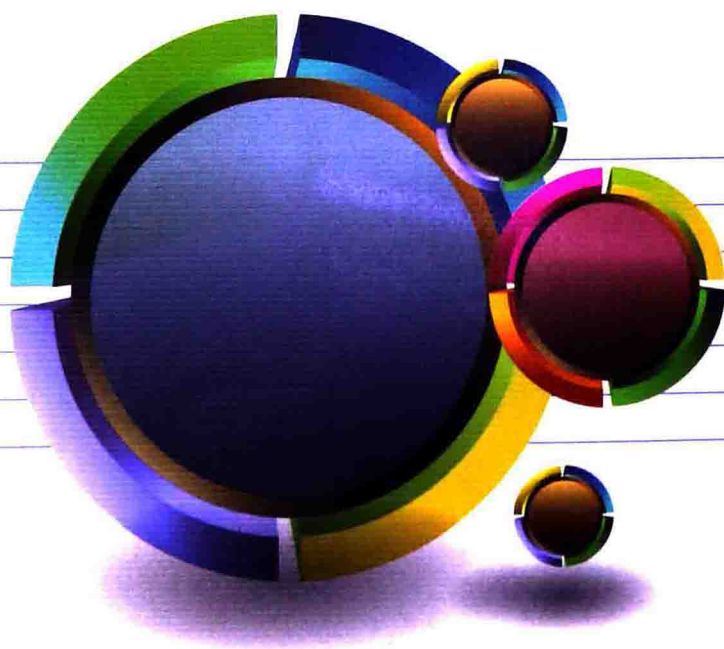


嵌入式

Linux 系统编程

深圳信盈达电子有限公司 秦立春 周中孝◎编 著



信盈达技术创新系列图书

嵌入式 Linux 系统编程

深圳信盈达电子有限公司 秦立春 周中孝 编著



电子工业出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书是配合 Linux 系统程序设计的指导教材，独立于任何 Linux 系统编程的教科书。主要内容有：Linux 基础知识，包括基本命令、vi 编辑器的用法、Shell 编程；Linux 环境下程序的编译与调试，包括 gcc/g++ 的用法、Makefile 的编写、gdb 调试器的基本用法等；Linux 文件目录操作；Linux 多进程的控制；Linux 环境下信号的处理；系统各进程间通信方式；Linux 多线程，包括线程的同步与互斥；网络 Socket 编程。

本书涵盖了 Linux 下应用层编程的全面知识，并且讲解清晰易懂。通过详细的示例，从零开始，由浅入深、循序渐进地指导初学者掌握 Linux 系统编程，培养读者实际分析问题和编程的能力，提高读者和学生的综合素质。

本书适合作为高等院校计算机专业的 Linux 系统编程教科书，也可为广大专业编程人员提供参考。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

嵌入式 Linux 系统编程/秦立春，周中孝编著. —北京：电子工业出版社，2014.8

(信盈达技术创新系列图书)

ISBN 978-7-121-23025-7

I. ①嵌… II. ①秦… ②周… III. ①Linux 操作系统—程序设计—教材 IV. ①TP316.89

中国版本图书馆 CIP 数据核字 (2014) 第 080404 号

策划编辑：杨 博

责任编辑：杨 博 文字编辑：张 迪

印 刷：北京季蜂印刷有限公司

装 订：北京季蜂印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1092 1/16 印张：18.5 字数：473.6 千字

版 次：2014 年 8 月第 1 版

印 次：2014 年 8 月第 1 次印刷

印 数：3 000 册 定价：49.80 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。

近年来，Linux 操作系统使用越来越广泛。而 Linux 操作系统的使用相比于 Windows 要困难一些，又由于 Linux 下没有很好的集成开发环境（IDE），在 Linux 下进行 C 语言编程开发比 Windows 下更不容易。而且在 Linux 下编程需要用到很多 Linux 系统 API 函数，这些函数虽然有完整的帮助文档可以查阅，但因为是英文，讲解太详细太全面，所以内容太多，我们通常没有时间对它们进行详尽的阅读，而且这些帮助文档以参考性的内容为主，这无疑增加了开发人员学习的难度。所以我们确实需要一本通俗易懂的介绍 Linux 系统下编程的书。本书是一本较为全面系统地介绍 Linux 下应用程序编程知识的书籍，涵盖了 Linux 系统下编程绝大多数的 API 函数及使用示例，足够广大开发人员参考使用。

国内关于讲解 Linux 系统编程知识的书籍比较少，国外相关著作虽然全面系统，但讲解风格却是依照西方的思维方式，过于专业和深奥，这与中国人的思维方式有些不同，所以我们在阅读那些书时难免会感到不习惯，影响学习。因此，我们需要一本按照国人容易接受的方式去讲解编程知识的参考书。本书讲解清晰易懂，内容科学合理，有助于初学者迅速掌握知识体系和精髓，在学习时间相同的情况下，学习效果也更好。

本书适合广大范围的计算机编程人员学习，只要读者具备一定的 C 语言编程基础即可。即使没用过 Linux 系统的人，阅读本书第 1 章 Linux 基本命令讲解后，也能对 Linux 系统有相当的了解和熟悉。然后讲述了 Linux 下程序编译和调试的基本知识，如 gcc、gdb 的用法，Makefile 的编写规则，这两章算是基础入门知识，接下来后面的章节都是介绍 Linux 下的 API 函数及其示例用法，程序都是精心设计的，一针见血地表达出了重点。API 函数部分包括 Linux 文件目录操作、基于文件指针和基于文件描述符的文件操作函数、多进程的控制、Linux 下信号处理、各种进程间通信方式的讲解、多线程编程、线程同步与互斥，以及网络 Socket 编程这些内容。

本书内容涵盖面广，知识点全面，语言简练而清晰，能让读者迅速掌握精髓，也很适合自学。初学者和专业开发人员都能从中受益匪浅。

本书由深圳信盈达电子有限公司秦立春、周中孝编著，在此向他们表示衷心的感谢。另外，感谢电子工业出版社，是他们认真专业的审核，让本书由粗糙的初稿变成了精美的图书。

由于时间仓促，编者水平有限，书中难免有不恰当的地方，希望广大读者批评指正，联系邮箱：niusdw@163.com，欢迎来信交流。

编著者

2014 年 2 月 27

第 1 章 Linux 基础知识	1
1.1 Linux 的历史	1
1.2 Linux 目录结构	1
1.3 Linux 常用命令	2
1.3.1 su (用户切换)	2
1.3.2 useradd (添加用户)	3
1.3.3 passwd (修改密码)	3
1.3.4 进程相关 (系统管理) 命令	4
1.3.5 磁盘相关命令 (fdisk)	4
1.3.6 pwd (显示工作目录)	4
1.3.7 cd (改变工作目录)	5
1.3.8 ls (查看文件/目录)	5
1.3.9 mkdir (创建目录)	6
1.3.10 cp (文件/目录复制)	7
1.3.11 mv (文件/目录改名、移动)	7
1.3.12 rm (文件/目录删除)	8
1.3.13 cat (查看文件内容)	8
1.3.14 more (分页显示文件内容)	9
1.3.15 less (分页显示文件内容)	10
1.3.16 管道与命令替换	11
1.3.17 umask (文件/目录创建掩码)	11
1.3.18 chgrp (改变文件/目录所属的组)	11
1.3.19 chown (更改某个文件/目录的所有者和属组)	11
1.3.20 chmod (修改文件权限)	12
1.3.21 find (文件查找)	14
1.3.22 locate (快速文件查找)	16
1.3.23 ln (建立符号链接)	16
1.3.24 whereis 和 which (查找命令所在目录)	17
1.3.25 grep (搜索文件内容)	17
1.3.26 tar (压缩解压)	18

1.3.27	gzip/gunzip 和 bzip2/bunzip2 (文件压缩/解压缩)	20
1.3.28	unzip (winzip 文件解压缩)	22
1.3.29	ifconfig 及网络相关命令	23
1.3.30	Linux 下编程环境	23
1.4	vi 编辑器	23
1.5	Linux 常用技巧	25
1.5.1	使用“Tab”键	25
1.5.2	Red Hat Linux 启动到字符界面 (不启动 XWindow)	26
1.5.3	挂接 USB 闪存	26
1.5.4	不用 samba 或 ftp, 利用 VM 自身实现本机与虚拟机的文件共享	27
1.5.5	其他常用命令	28
1.5.6	如何配置 vim 脚本功能及 vim 中设置多窗口	29
1.5.7	Shell 编程	31
第2章	Linux 环境下的编译与调试	45
2.1	gcc/g++ 编译器	45
2.1.1	gcc/g++ 编译过程	45
2.1.2	静态库和动态库	48
2.1.3	gcc 警告和优化选项	52
2.2	make 工程管理和 Makefile	54
2.2.1	Makefile 中特殊处理与伪目标	55
2.2.2	变量、规则与函数	56
2.3	gdb 调试器	60
2.3.1	gdb 的常用命令	60
2.3.2	gdb 应用举例	62
第3章	Linux 文件目录操作	64
3.1	基于文件指针的文件操作 (缓冲)	64
3.1.1	文件的创建、打开与关闭	64
3.1.2	读写文件	65
3.1.3	文件定位	68
3.1.4	标准输入/输出流	69
3.1.5	目录操作	69
3.2	基于文件描述符的文件操作 (非缓冲)	73
3.2.1	文件描述符	73
3.2.2	打开、创建和关闭文件	73
3.2.3	读写文件	75

3.2.4	改变文件大小	77
3.2.5	文件定位	77
3.2.6	原子操作	78
3.2.7	进一步理解文件描述符	78
3.2.8	文件描述符的复制	79
3.2.9	文件的锁定	81
3.2.10	获取文件信息	86
3.2.11	access 函数	88
3.2.12	标准输入/输出文件描述符	89
3.2.13	时间和日期相关函数	90
3.2.14	处理的模型（补充）	91
3.2.15	串口编程	95
3.2.16	日志	103
第 4 章	Linux 多进程	105
4.1	Linux 进程概述	105
4.1.1	进程标识	105
4.1.2	进程的用户 ID 与组 ID（进程的运行身份）	105
4.1.3	进程的状态	108
4.1.4	Linux 下的进程结构及管理	108
4.2	Linux 进程的创建与控制	109
4.2.1	fork 函数	109
4.2.2	进程的终止	112
4.2.3	wait 和 waitpid 函数	113
4.2.4	exec 函数族	116
4.2.5	system 函数	120
4.2.6	popen 函数	120
4.3	守护进程	121
第 5 章	Linux 信号处理	125
5.1	信号概念	125
5.2	signal 信号处理机制	126
5.3	sigaction 信号处理机制	128
5.3.1	信号处理情况分析	128
5.3.2	sigaction 信号处理注册	130
5.3.3	sigprocmask 信号阻塞	134
5.4	用程序发送信号	136

5.4.1	kill 和 raise 信号发送函数	136
5.4.2	sigqueue 信号发送函数	137
5.5	计时器与信号	140
5.5.1	睡眠函数	140
5.5.2	时钟处理	141
第6章	进程间通信	144
6.1	进程间通信 (IPC) 简介	144
6.2	标准管道流	145
6.3	无名管道 (PIPE)	146
6.4	命名管道 (FIFO)	149
6.4.1	创建、删除 FIFO 文件	149
6.4.2	打开、关闭 FIFO 文件	150
6.4.3	读写 FIFO	150
6.5	内存映射	151
6.6	System V 共享内存机制	156
6.7	消息队列	159
6.8	信号量	163
6.8.1	System V IPC 机制: 信号量	164
6.8.2	Posix 有名信号量	170
第7章	Linux 多线程	175
7.1	Linux 多线程概述	175
7.1.1	多线程概述	175
7.1.2	线程分类	176
7.1.3	线程创建的 Linux 实现	176
7.2	线程的创建和退出	176
7.3	线程的等待退出	178
7.3.1	等待线程退出	178
7.3.2	线程的取消	180
7.3.3	线程终止清理函数	182
7.4	线程的互斥	184
7.5	线程的同步	191
7.5.1	条件变量	191
7.5.2	信号灯	197
7.6	生产者消费者问题	200
7.7	线程的属性	204

第 8 章 Linux 网络编程	208
8.1 Linux 网络编程介绍	208
8.1.1 TCP/IP 概述	208
8.1.2 OSI 参考模型及 TCP/IP 参考模型	208
8.1.3 TCP	210
8.1.4 UDP	212
8.1.5 协议的选择	212
8.2 网络相关概念	213
8.2.1 Socket 概念	214
8.2.2 Socket 类型	214
8.2.3 Socket 的信息数据结构	215
8.2.4 数据存储优先顺序的转换	215
8.2.5 地址格式转化	216
8.2.6 名字地址转化	217
8.3 Socket 编程	220
8.3.1 使用 TCP 的流程图	220
8.3.2 使用 UDP 的流程图	237
8.3.3 设置套接口的选项 setsockopt 的用法	242
第 9 章 系统编程的例子	244
附录 A C 语言文件操作详解	257
附录 B 虚拟内存 (VM)	266
附录 C ping 命令解析	274
附录 D 大端小端	285

1.1 Linux 的历史

操作系统始于 20 世纪 50 年代。1991 年，一个计算机爱好者在 Minix（一个功能很有限的类似于 UNIX 的操作系统）的基础上开发出一款操作系统，叫 Linux。最初的 Linux 作用并不大，但是很多人开始对 Linux 感兴趣，共同开发它，使 Linux 变成了一个很有用的操作系统，但此时它的功能仍然不够好。

为创建一套完全自由的操作系统，有人发起了 GNU 计划。GNU 计划又称革奴计划，它可以让任何人自由地使用、复制、修改和发布 GNU 软件。GNU 软件都遵守 GPL 协议（GNU General Public License，GNU 通用开源许可证），或称“反版权”协议。在该计划的促进下，开发人员开发出了功能得到极大改善的新的 Linux 操作系统，于是 Linux 开始流行起来。GNU 对 Linux 的贡献非常大。

Linux 本身只是操作系统的内核，后来才添加进许多应用功能。内核非常重要，它是其他程序能够运行的基础。实际上，用户或者系统管理员运行的所有程序都运行在内核之上。

为什么要用 Linux 操作系统而不用 Windows？Linux 操作系统相比于 Windows 的优点包括：

- 开源和完全免费，这是其他操作系统做不到的；
- 支持多种平台，Linux 可运行在多种硬件的处理器平台上；
- 最突出的是它可以很好地运行在嵌入式产品上，如掌上电脑、机顶盒、游戏机等；
- Linux 系统很安全，针对 Linux 的病毒很少。

Linux 也有缺点：可扩展性不强；缺乏硬件集成/支持；缺乏洞察力等。

1.2 Linux 目录结构

/bin：该目录中存放 Linux 的常用命令，在有的版本中是一些和根目录下相同的目录。

/boot：该目录下存放的都是系统启动时要用到的程序，当用 lilo 引导 Linux 时，会用到这里的一些信息。

/dev：该目录包含了 Linux 系统中使用的所有外部设备，它实际上是访问这些外部设备

的端口，可以访问这些外部设备，与访问一个文件或一个目录没有区别。例如，在系统中输入“cd/dev/cdrom”，就可以看到光驱中的文件；输入“cd/dev/mouse”，即可看到与鼠标相关的文件。

/etc：该目录存放了系统管理时要用到的各种配置文件和子目录，如网络配置文件、文件系统、X 系统配置文件、设备配置信息、设置用户信息等。

/sbin：该目录用来存放系统管理员的系统管理程序。

/home：如果建立一个名为“xx”的用户，那么在/home 目录下就有一个对应“/home/xx”的路径，用来存放该用户的主目录。

/lib：该目录用来存放系统动态链接共享库，几乎所有的应用程序都会用到该目录下的共享库。

/lost + found：该目录在大多数情况下都是空的；但当突然停电或者非正常关机后，有些文件就临时存放在这里。

/mnt：该目录在一般情况下也是空的，可以临时将别的文件系统挂在该目录下。

/proc：可以在该目录下获取系统信息，这些信息是在内存中由系统自己产生的。

/root：如果你是以超级用户的身份登录的，这个就是超级用户的主目录。

/tmp：用来存放不同程序执行时所产生的临时文件。

/usr：用户的很多应用程序和文件都存放在该目录下。例如：

- /usr/X11R6：X - Window 目录；
- /usr/src：Linux 源代码；
- /usr/include：系统头文件；
- /usr/lib：存放常用的动态链接共享库和静态档案库。

1.3 Linux 常用命令

Linux 下的两种账号：

- root——超级用户账号（系统管理员），使用这个账号可以在系统中做任何事情；
- 普通用户——供普通用户使用，可以进行有限的操作。

用户正确地输入用户名和口令后，就能合法地进入系统，屏幕显示“[root@ localhost/ root]#”。

这时可以对系统做各种操作。注意超级用户的提示符是“#”，其他用户的提示符是“\$”。利用 whoami 可以知道当前登录的用户账号是什么身份。

用 exit 或 logout 结束当前的 Linux 会话，也可按下〈CTRL〉+ d 来结束 Linux 会话。

1.3.1 su（用户切换）

功能：切换用户。

语法：su [用户名] 注：“[]”表示可选

说明：su 命令是最基本的命令之一，常用于不同用户间切换。例如：如果登录为 user1，要切换为 user2，只要用命令“\$su user2”；然后系统提示输入 user2 口令，输入正确的口令之后就可以切换到 user2；完成之后可以用 exit 命令返回到 user1。

su 命令的常见用法是变成根用户或超级用户。如果普通用户发出不带用户名的 su 命令，则系统提示输入根口令，输入之后即可切换为根用户。

如果登录为根用户，则可以用 su 命令切换为系统上任何用户而不需要口令。

1.3.2 useradd (添加用户)

功能：添加用户。

语法：useradd 用户名

说明：useradd 可以用来建立用户账号。账号建好之后，再用 passwd 设定账号的密码。而可用 userdel 删除账号。使用 useradd 指令所建立的账号，实际上保存在“/etc/passwd”文本文件中。每建立一个用户，就在“/home”目录下建立一个主目录。

该命令必须有系统管理员权限才能执行。

常用选项：-d 指定目录，-m 指定权限，-g 为组号（把用户加入到具体的组）。

利用 id 可以查看用户信息组号，id lry 可以查看 lry 用户的组信息。

1.3.3 passwd (修改密码)

功能：添加密码和修改密码。

语法：passwd [用户名]。

说明：出于系统安全的考虑，Linux 系统中的每一个用户除了有其用户名外，还有其对应的用户密码。因此，在使用 useradd 命令添加用户时，还需使用 passwd 命令为每一位新添加的用户设置密码；用户以后还可以随时用 passwd 命令修改自己的密码。

该命令的一般格式为：passwd [用户名]。其中用户名为需要修改密码的用户名。只有超级用户可以使用“passwd 用户名”修改其他用户的密码，普通用户只能用不带参数的 passwd 命令修改自己的密码。

利用 passwd 命令修改自身密码的方法如下：

- 输入“: passwd” <Enter>；
- 在“(current) LINUX passwd:”下输入当前的密码；
- 在“new password:”提示下输入新的密码（在屏幕上看不到这个密码）；
- 系统提示再次输入这个新密码。

输入正确后，这个新密码被加密并放入“/etc/shadow”文件。选取一个不易被破译的密码是很重要的，选取密码应遵守如下规则：

- 密码应该至少有 6 位（最好是 8 位）字符；
- 密码应该是大小写字母、标点符号和数字混杂的。

超级用户修改其他用户（xxq）的密码的过程如下：

```
# passwd xxq
New LINUX password:
Retype new LINUX password:
passwd: all authentication tokens updated successfully
```

1.3.4 进程相关（系统管理）命令

ps 命令：类似于任务管理器，ps 为我们提供了一次性查看进程的方法。ps 命令所提供的查看结果并不是动态连续的；如果想对进程实时监控，应该用 top 工具。ps 常用形式为 ps -aux 和 ps -ef。

执行 ps -aux 之后结果如下：

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.1	0.1	1372	472	?	S	21:32	0:04	init
root	2	0.0	0.0	0	0	?	SW	21:32	0:00	[keventd]
...										

该命令的参数如下：

- USER：进程的属主；
- PID：进程的 ID（是唯一的数值，用来区分进程）；
- PPID：父进程；
- %CPU：进程占用的 CPU 百分比；
- %MEM：占用内存的百分比；
- NI：进程的 NICE 值，数值大，表示较少占用 CPU 时间；
- VSZ：进程的虚拟大小；
- RSS：驻留中页的数量；
- TTY：终端 ID；
- WCHAN：正在等待的进程资源；
- stat：进程状态 [运行 R、休眠 S、僵尸 Z、停止或被追踪 T、死掉的进程 X、优先级较低的进程 N、优先级高的进程 <、进入内存交换 W、非中断休眠（常规 IO）D]；
- START：启动进程的时间；
- TIME：进程消耗 CPU 的时间；
- COMMAND：命令的名称和参数。

kill 命令：通常与 ps 命令一起使用，常用的形式是“kill -9 进程 ID”（表示向指定的进程 ID 发送 SIGKILL 的信号，其中 -9 表示强制终止，可以省略，它是信号代码，可以利用 kill -l 列出所有信号），另一种形式是“pkill 进程名字”（可以直接杀死指定进程名的进程）。

top 命令：和 ps 相比，top 是动态监视系统任务的工具，top 输出的结果是连续的，比如 #top。

jobs 命令：观察后台进程。

1.3.5 磁盘相关命令（fdisk）

fdisk 命令可以查看硬盘分区情况，并可用于对硬盘分区进行管理以及给硬盘分区等。其常用功能是查看 U 盘状况，比如 fdisk -l。

1.3.6 pwd（显示工作目录）

功能：在 Linux 层次目录结构中，用户可以在被授权的任意目录下利用 mkdir 命令创建

新目录，也可以利用 `cd` 命令从一个目录转换到另一个目录。然而，没有提示符来告知用户目前处于哪一个目录中。要想知道当前所处的目录，可以使用 `pwd` 命令，该命令显示整个路径名。

语法：`pwd`

说明：此命令显示出当前工作目录的绝对路径。

举例：`pwd`

根目录以开头的“/”表示。如果 `pwd` 后面什么都没有，则显示当前所在位置；如果屏幕信息很多，用 `clear` 命令可以清除。

1.3.7 `cd`（改变工作目录）

功能：改变工作目录。工作目录是当前左边的信息所显示的目录。

语法：`cd [dirname]`

说明：该命令将当前工作目录切换至 `dirname` 所指定的目录。若没有指定 `dirname`，则回到用户的主目录。为了改变为指定目录，用户必须拥有对指定目录的执行权限和读权限。

该命令也可以使用通配符。例如，假设用户当前目录是 `/home/lry`，现需要更换到 `/home/lry/Linux` 目录中：

```
$cdLinux
```

此时，用户可以执行 `pwd` 命令来显示工作目录（相对路径的形式）：

```
$pwd/home/lry/Linux （绝对路径的形式）
```

用法：

<code>cd .</code>	到父目录，即上一级目录，相当于“向上”
<code>cd -</code>	到上一次目录，相当于“后退”
<code>cd /</code>	到根目录
<code>cd ~</code> （或者只写“ <code>cd</code> ”）	到用户主目录下

1.3.8 `ls`（查看文件/目录）

功能：`ls` 是英文“list”的简写，其功能为列出目录的内容。这是用户最常用的一个命令之一，因为用户需要不时地查看某个目录的内容。该命令类似于 DOS 下的 `dir` 命令。

语法：`ls [选项][目录或是文件]`

说明：对于每一个目录，该命令将列出其中的所有子目录与文件。对于每个文件，`ls` 将输出其文件名以及所要求的其他信息。默认情况下，输出条目按字母顺序排序。当未给出目录名或文件名时，就显示当前目录的信息。注意，Linux 文件系统不是根据后缀来执行文件的，而是根据此文件有没有执行权限来执行文件的。

常用参数：

- a 显示指定目录下所有子目录与文件，包括隐藏文件。
- A 显示指定目录下所有子目录与文件，包括隐藏文件，但不列出“.”和“..”。
- i 在输出的第一列显示文件的 i 节点号。
- l 以长格式来显示文件的详细信息。这个选项最常用。
- d 列出目录本身。

-p 在目录后面加一个“/”。

-R 递归式地显示指定目录的各个子目录中的文件。

以 -l 长格式显示文件的详细时，每行列出的信息依次是：文件类型与权限，链接数，文件所有者，文件属组，文件大小，最近修改的时间，名字。

对于符号链接文件，显示的文件名之后有“->”和引用文件路径名。

对于设备文件，其“文件大小”字段显示主、次设备号，而不是文件大小。

目录中的总块数显示在长格式列表的开头，其中包含间接块。

在用 ls -l 命令显示的信息中，开头是由 10 个字符构成的字符串，其中第一个字符表示文件类型，它可以是下述类型之一：

- 普通文件；
- 目录 (d)；
- 符号链接 (l)；
- 块设备文件 (b)；
- 字符设备文件 (c)；
- 命名管道 (p)；
- socket 文件 (s)。

后面的 9 个字符表示文件的访问权限，分为 3 组，每组 3 位。第一组表示文件所有者的权限，第二组表示同组用户的权限，第三组表示其他用户的权限。每一组的三个字符分别表示对文件的读、写和执行权限。各权限如下：

r (读)	4
w (写)	2
x (执行)	1 (对于目录，表示进入权限)
-	没有设置权限。

另外，ls 输出内容是有颜色的，比如：目录是蓝色的，压缩文件是红色的。如果没有颜色，可以加上参数“color = never”表示输出没有彩色，而“color = auto”表示自动，“color = always”表示始终有颜色。

通配符在 ls 命令中的应用：

- * 代表 0 个或多个字符
- [] 表示内部包括任何字符
- ? 表示任何单个字符

如果需要更加详细的参数描述，可以通过如下 3 种方式获得 ls 的帮助：

- ls—help
- man ls
- info ls

✎ 1.3.9 mkdir (创建目录)

功能：创建一个目录

语法：mkdir [选项] dirname

说明：该命令创建由 dirname 命名的目录。要求创建目录的用户在当前目录中 (dirname

的父目录中) 具有写权限, 并且 `dirname` 不能是当前目录中已有的目录或文件名称。

参数:

- m 对新建目录设置存取权限。也可以用 `chmod` 命令设置。
- p 可以是一个路径名称。此时若路径中的某些目录尚不存在, 加上此选项后, 系统将自动建立好那些尚不存在的目录, 即一次可以建立多个目录。

✎ 1.3.10 cp (文件/目录复制)

功能: 将给出的文件或目录复制到另一文件或目录中。

语法: `cp [选项] 源文件或目录 目标文件或目录`

说明: 该命令把指定的源文件复制到目标文件, 或者把多个源文件复制到目标目录中。

参数:

- a 该选项通常在复制目录时使用。它保留链接、文件属性, 并递归地复制目录。
- d 复制时保留链接。
- f 删除已经存在的目标文件而不提示。
- i 和 `f` 选项相反, 在覆盖目标文件之前给出提示要求用户确认。回答 `y` 时目标文件将被覆盖, 是交互式复制。
- r 若给出的源文件是一个目录文件, 此时 `cp` 将递归复制该目录下所有的子目录和文件。此时目标文件必须为一个目录名。

要说明的是, 为了防止用户在不经意的情况下用 `cp` 命令破坏另一个文件 (如用户指定的目标文件名是一个已存在的文件名, 用 `cp` 命令复制文件后, 这个文件就会被新复制的源文件覆盖), 建议用户在使用 `cp` 命令复制文件时, 最好使用 `i` 选项, 如下所示:

```
$cp -ia.txt/home/b.txt
```

该命令将文件 `a.txt` 复制到 “/home” 这个目录下, 并改名为 “`b.txt`”。若不希望重新命名, 可以使用下面的命令:

```
$cp a.txt/home
$cp -r/home/lry/root
```

该命令将 `/home/lry` 目录中的所有文件及其子目录复制到目录 `/root` 中。

✎ 1.3.11 mv (文件/目录改名、移动)

功能: 为文件或目录改名或将文件由一个目录移入另一个目录中。该命令如同 DOS 下 `ren` 和 `move` 的组合。

语法: `mv [选项] 源文件或目录 目标文件或目录`

说明: 视 `mv` 命令中第二个参数类型的不同 (是目标文件还是目标目录), `mv` 命令将文件重命名或将其移至一个新的目录中。当第二个参数类型是文件时, `mv` 命令完成文件重命名, 此时源文件只能有一个 (也可以是源目录名), 它将所给的源文件或目录重命名为给定的目标文件名。当第二个参数是已存在的目录名称时, 源文件或目录参数可以有多个, `mv`

命令将各参数指定的源文件均移至目标目录中。在跨文件系统移动文件时，mv 先复制，再将原有文件删除，而与该文件的链接也将丢失。

参数：

- i 交互方式操作。如果 mv 操作将导致对已存在的目标文件的覆盖，则系统询问是否重写，要求用户回答 y 或 n，这样可以避免误覆盖文件。
- f 禁止交互操作。在 mv 操作要覆盖某个已有的目标文件时不给任何指示；指定此选项后，i 选项将不再起作用。

如果所给目标文件（不是目录）已存在，此时该文件的内容将被新文件覆盖。为防止用户在不经意的情况下用 mv 命令破坏另一个文件，建议用户在使用 mv 命令移动文件时，最好使用 i 选项。

需要注意的是，mv 与 cp 的结果不同。mv 好像文件“搬家”，文件个数并未增加，而 cp 对文件进行复制，文件个数增加了。

示例：

- ① 将 /lry 中的所有文件移到当前目录（用“.”表示）中：

```
$mv/home/lry/* .
```

- ② 将文件 a.txt 重命名为 b.txt：

```
$mv a.txt b.txt
```

✎ 1.3.12 rm(文件/目录删除)

功能：在 Linux 中创建文件很容易，系统中随时会有文件过时且毫无用处。用户可以用 rm 命令将其删除。该命令的功能为删除一个目录中的一个或多个文件或目录，它也可以将某个目录和其下的所有文件及子目录均删除。对于链接文件，只是删除链接，原有文件均保持不变。

语法：rm [选项] 文件…

说明：如果没有使用 -r 选项，则 rm 不会删除目录。

参数：

- f 忽略不存在的文件，从不给出提示；
- r 指示 rm 将参数中列出的全部目录和子目录均递归地删除；
- i 进行交互式删除。

使用 rm 命令要格外小心；因为一旦一个文件被删除，它是不能被恢复的。例如，用户在输入 cp、mv 或其他命令时，不小心误输入了 rm 命令，当该用户按了回车键并认识到自己的错误时，已经太晚了，文件已经没有了。为了防止这种情况的发生，可以使用 rm 命令中的 i 选项来确认要删除的每个文件：如果用户输入 y，文件将被删除；如果输入任何其他内容，文件将被保留。

✎ 1.3.13 cat（查看文件内容）

功能：查看文件内容。