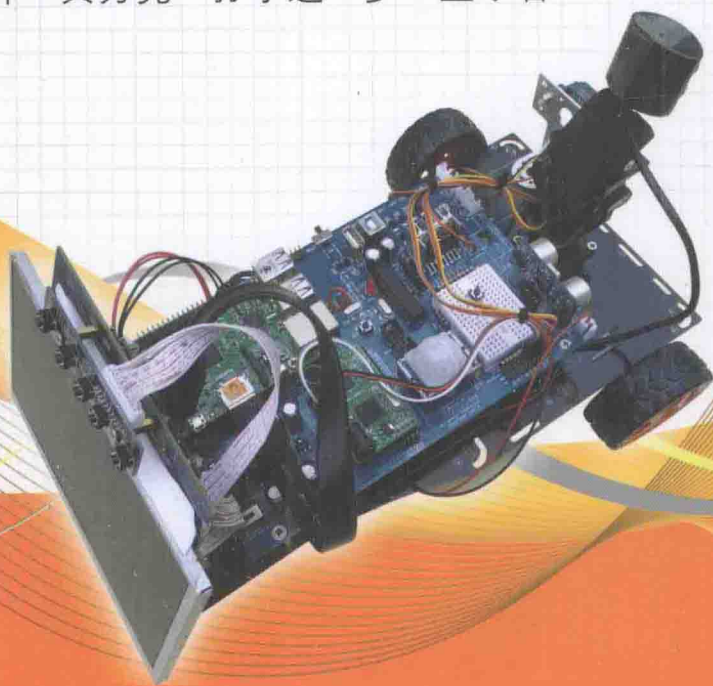


智能家居机器人 设计与控制

ZHINENG JIAJU JIQIREN
SHEJI YU KONGZHI

谭立新 朱 林 黄秀亮 孙小进 罗 坚◇著



智能家居机器人设计与控制

谭立新 朱 林 黄秀亮 孙小进 罗 坚 著

 北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

全书详细介绍了嵌入式 Linux 系统应用程序设计的方法与技巧,结合视觉、触觉、听觉等传感器将机器人推向智能化。本书以服务机器人的各种复杂功能下手,从简单的基本运动、机器人的视觉系统、机器人的触觉系统及机器人的听觉系统逐步进行详细的讲解,到最后搭建成一个功能完善的服务机器人,以不同的传感器分配不同的案例进行详细的边实验边学习的步骤讲解,每个案例都经过现场调试运行,是多年开发经验的结晶。

版权专有 侵权必究

图书在版编目(CIP)数据

智能家居机器人设计与控制/谭立新等著. —北京:北京理工大学出版社, 2015. 1
ISBN 978-7-5640-9939-8

I. ①智… II. ①谭… III. ①智能机器人-设计 IV. ①TP242. 6

中国版本图书馆 CIP 数据核字(2014)第 257547 号

出版发行/北京理工大学出版社有限责任公司

社 址/北京市海淀区中关村南大街 5 号

邮 编/100081

电 话/(010) 68914775 (总编室)

82562903 (教材售后服务热线)

68948351 (其他图书服务热线)

网 址/<http://www.bitpress.com.cn>

经 销/全国各地新华书店

印 刷/三河市华骏印务包装有限公司

开 本/787 毫米×1092 毫米 1/16

印 张/12.5

字 数/290 千字

版 次/2015 年 1 月第 1 版 2015 年 1 月第 1 次印刷

定 价/52.00 元

责任编辑/钟 博

文案编辑/胡卫民

责任校对/孟祥敬

责任印制/李志强

图书出现印装质量问题,请拨打售后服务热线,本社负责调换

前言

随着机器人技术逐渐成熟，机器人已在军事、工业生产、农业、科研上被大量使用，科学家们在机器人能代替人做更多的事情上面花费了更多的精力，从而出现了研究热潮。家庭服务机器人是机器人服务人类的一个研究方向，现代老年化程度日渐攀升、年轻人工作压力增大、人们对生活品质的追求提高，因此，人们在利用机器人提供服务方面有了更多的需求，现在出现了大量的扫地机器人和某些特定的医疗服务机器人、助老机器人等。家庭服务机器人的研究发展领域还是有很大空间的，特别是更加智能的服务机器人的研究，我国的服务机器人研究还处在一个初级阶段，所以我们必须加强对相关研究人员的培养，这对于我们的服务机器人产业具有十分重要的战略意义。

机器人学是多学科的综合产物，集成度较高，特别需要强大的计算能力和并行处理能力，本书利用 ARM 为核心的强大硬件处理器搭载 Linux 嵌入式系统作为机器人的控制中心和数据处理中心，利用各种传感器及其他微控制器为控制和感应系统进行模块化设计，是电子信息专业、计算机专业、机电一体化专业的理想学习平台。

现在高职院校的专业教材很多都是按照知识体系编写的，重理论、轻实践，重原理、轻应用。学生学习了一个学期，不知道这个知识在实践中有什么作用、为什么要学习、与今后的工作有什么联系以及如何应用等。本书按照项目设计制作或者工作过程展开，将学科知识融入项目或者产品的制作过程中，回归人类认识自然的本源方式。书中典型工程案例的实践过程按照“任务驱动”的模式组织，通过“实践—归纳—推理—再实践”这一螺旋式上升的模式使学生获取系统的科学知识和实践技能，并将企业的“6S”考核体系融入教学考核评价中。我们将交互式智能家居机器人作为典型工程案例以培养学生的职业能力，这是本书的主要探索方向，也是本书的主要特色。

本教材的每个项目主要由以下几个部分组成：

(1) 项目描述，介绍该项目的工程实践应用与技术要求及其实现的关键技术；介绍案例与课程的联系、技术要点及职业要求。

(2) 教学目标，从知识、能力、态度等方面介绍各项目的教学目标，包括方法能力、专业能力、社会能力以及团队合作能力等。

(3) 知识准备，介绍项目设计制作所必需的知识，不求知识的完整性与全面性，以“必需、够用”为度。

(4) 任务实现，介绍实现任务所需的材料、场地、技术资料及实现的步骤、相关的技术要求、现场要求、工艺与技术文件的撰写方法等。

(5) 考核评价，对作品的质量、技术、职业素养、完成过程进行评价。

(6) 拓展提高，让读者学会触类旁通、举一反三，强调学生知识的迁移、技能的强化、职业能力的拓展等。

本书从嵌入式系统与机器人的设计入手，详细介绍了嵌入式 Linux 系统应用程序设计的方法与技巧，结合视觉、触觉、听觉等传感器将机器人应用技术进一步推向智能化。作者在

实用的基础之上,通过大量工程实践详细介绍了服务机器人的软件设计,从简单的基本运动、机器人的视觉系统、机器人的触觉系统及机器人的听觉系统逐步进行细致的讲解,到最后搭建成一个功能完善的服务机器人。

与同类书相比,本书具有以下特色:

(1) 结构清晰,内容合理,讲解从零开始,循序渐进,将大量实例呈现在读者面前,并将基础知识融合在实例中,以“先练后讲”的方式充分讲解枯燥无味的概念性知识点。

(2) 提供了 30 个嵌入式系统应用于机器人的训练实例,这些实例典型、实用、易学易懂,所有程序都通过调试并被下载到机器人身上进行过运行实践,涵盖了嵌入式系统的大量开发技术。

(3) 对机器人项目的开发步骤和设计思路进行了详细讲解,穿插介绍了开发中的注意事项,对程序代码进行了详细注释,有利于读者理解和巩固知识点。读者既可以单独成模块地使用每个项目,也可以将项目内容结合在一起开发出新的有趣的项目。

本书适合作为计算机、自动化、电子等相关专业的本科和高职院校的专业教材,也可作为学习嵌入式系统技术、传感器技术、嵌入式技术和人工智能技术的主导教材或辅助教材,还可供其他智能控制技术和机器人爱好者使用。

由于时间仓促和水平有限,书中难免有不妥之处,请读者不吝赐教,多提宝贵意见。

谭立新等
于湘江之滨
2014 年仲秋

目 录

项目 1 开发软件平台环境的搭建与配置	1
1.1 项目描述	1
1.2 教学目标	1
1.3 知识准备	1
1.3.1 了解嵌入式系统 Linux 使用的命令行	1
1.3.2 了解 Linux 文件与文件管理命令	2
1.3.3 更多 Linux 命令	5
1.3.4 特殊的控制键	6
1.3.5 了解 Linux 进程机制命令	7
1.3.6 了解 sudo 与权限	7
1.4 任务实现	8
1.4.1 安装 Raspbian Linux 发行版本操作系统	8
1.4.2 配置 Raspbian Linux 操作系统	10
1.4.3 通过 apt-get 管理软件	13
1.4.4 使用 Pi Store 安装软件和多媒体	15
1.4.5 设置 Raspbian Linux 固件	15
1.4.6 把控制器变成多媒体中心	17
1.4.7 在控制器上面玩游戏	19
1.4.8 使用 Raspbian Linux 系统编译 Python 程序	20
1.5 考核评价	24
任务 1 Raspbian Linux 系统安装	24
任务 2 使用 Raspi-config 设置系统	24
任务 3 启动进入桌面	24
任务 4 通过 apt-get 安装新软件	24
任务 5 通过 apt-get 删除软件	24
任务 6 使用 Python 编写简单的 Hello 程序	24
1.6 拓展提高	24
任务 1 升级固件或内核	24
任务 2 设置视频输出	26
任务 3 测试和设置音频系统	27
项目 2 嵌入式交互式智能家居机器人控制系统	28
2.1 项目描述	28
2.2 教学目标	28
2.3 知识准备	28

2.3.1 智能家居机器人控制系统硬件主板	28
2.3.2 智能家居机器人控制系统的特性	29
2.3.3 智能家居机器人控制系统电路及传感器资源	30
2.4 任务实现	34
2.4.1 使用输入输出接口	34
2.4.2 数字信号输出: 点亮 LED	34
2.4.3 数字信号输入: 读取按钮状态	36
2.4.4 在 Python 中安装并测试 GPIO	38
2.4.5 让 LED 闪烁	39
2.4.6 读取按钮状态	40
2.4.7 在智能家居机器人系统中定位串口	42
2.4.8 串口通信	44
2.5 考核评价	46
任务 1 制作定时台灯项目	46
任务 2 制作简易发音板项目	49
任务 3 在智能家居机器人系统中驱动底层控制系统	51
项目 3 交互式智能家居机器人运动模块设计	56
3.1 项目描述	56
3.2 教学目标	56
3.3 知识准备	56
3.3.1 关于直流电机	56
3.3.2 关于 L298 驱动器	57
3.3.3 底层控制板的基本电机编程技巧	59
3.3.4 导航传感器选择	62
3.3.5 家居巡航导航策略	63
3.4 任务实现	64
3.4.1 方案设计	64
3.4.2 硬件电路设计	64
3.4.3 家居机器人导航组装	65
3.4.4 智能家居机器人导航程序设计	67
3.4.5 智能家居机器人导航避开“死角”的算法设计	69
3.4.6 调试程序	72
3.4.7 利用嵌入式系统监测运行状态	72
3.5 考核评价	76
任务 1 实现智能家居机器人家庭巡航	76
任务 2 嵌入式智能家居机器人的实现	76
3.6 拓展提高	76
任务 智能家居机器人智能导航	76

项目 4 交互式智能家居机器人视觉交互控制模块设计	77
4.1 项目描述	77
4.2 教学目标	77
4.3 知识准备	78
4.3.1 Python 基础编程	78
4.3.2 计算机视觉库 SimpleCV 的运行条件	82
4.3.3 计算机视觉库 SimpleCV 基础编程	82
4.3.4 关于摄像头的选择	84
4.3.5 关于视觉处理的基本概念	85
4.4 任务实现	85
4.4.1 测试摄像头	85
4.4.2 安装并测试 SimpleCV	86
4.4.3 图像数据无线传输的实现	87
4.4.4 图像变换与增强处理	90
4.4.5 图像背景去除和轮廓提取的实现	99
4.4.6 图像分割和运动物体检测的实现	100
4.4.7 图像模式识别——颜色识别算法的实现	105
4.4.8 图像模式识别——形状识别算法的实现	109
4.5 考核评价	119
任务 1 条码识别	119
任务 2 人脸检测	120
任务 3 二维码识别	121
4.6 拓展提高	122
任务 利用智能家居机器人编写一个互动型照相馆	122
项目 5 交互式智能家居机器人听觉交互控制模块设计	125
5.1 项目描述	125
5.2 教学目标	125
5.3 知识准备	125
5.3.1 关于语音识别	125
5.3.2 听觉交互控制模块硬件的设计思路	127
5.3.3 语音感知、获取的原理	128
5.4 任务实现	129
5.4.1 方案设计	129
5.4.2 听觉交互控制模块硬件的设计与制作	129
5.4.3 语音分析处理与模式识别设计	130
5.4.4 语音识别与机器人动作控制程序设计	133
5.4.5 程序代码的编辑、调试及运行	135
5.5 考核评价	135
任务 1 交互式智能家居机器人的人机对话设计	135

任务2 接收主人发出的控制命令并响应	136
5.6 拓展提高	136
任务 智能家居语音控制机器人的动作互动	136
项目6 交互式智能家居机器人触觉交互控制模块设计	137
6.1 项目描述	137
6.2 教学目标	137
6.3 知识准备	137
6.3.1 超声波传感器	137
6.3.2 超声波测距的原理及计算公式	138
6.3.3 关于非接触式红外触觉探测	139
6.3.4 非接触式红外触觉探测传感器介绍及其控制方式	140
6.3.5 非接触式温度测量	140
6.3.6 非接触式温度测量传感器介绍及其控制方式	141
6.4 任务实现	143
6.4.1 方案设计	143
6.4.2 总体硬件电路设计	144
6.4.3 非接触式超声波触觉软件的编写	144
6.4.4 非接触式红外触觉探测软件的编写	148
6.4.5 非接触式温度触觉软件的编写	149
6.4.6 程序代码的编辑、调试及运行	153
6.5 考核评价	153
任务1 编写感应人体接近动作的响应程序	153
任务2 用超声波判断人体离机器人的距离	153
任务3 智能家庭机器人非接触式人体温度测量服务	154
6.6 拓展提高	154
任务 基于非接触式多传感器的机器人对周围环境的监测	154
项目7 交互式智能家居机器人交互模块整体软硬件调试	155
7.1 项目描述	155
7.2 教学目标	155
7.3 知识准备	155
7.4 任务实现	156
7.4.1 综合方案设计	156
7.4.2 整体硬件电路设计	156
7.4.3 多模块智能程序的编写	156
7.4.4 程序代码的编辑、调试及运行	162
7.4.5 烧写可执行文件	162
7.5 考核评价	162
任务 触觉和听觉导航视觉综合案例	162
7.6 拓展提高	171

任务 触觉、听觉导航、无线数据传输和视觉信息采集显示综合编程	171
附录 A Linux 操作系统	177
A1.1 初次使用	177
A1.2 浏览文件系统	179
A1.3 编辑文本文件	179
A1.4 管理用户	181
A1.5 管理进程	183
A1.6 关闭和重启智能家居机器人控制系统	185
A1.7 获取帮助	185
附录 B 安装所需元器件清单	186
附录 C 调试所需仪器仪表清单	187
参考文献	188

项目 1

开发软件平台环境的搭建与配置



1.1 项目描述

本项目的主要学习内容包括：了解 Linux 嵌入式系统、安装 Raspbian Linux 发行版本操作系统、通过 apt-get 管理软件使用 Pi Store 安装软件和多媒体、使用 Raspi-config 设置系统、使用 Python 编写简单的程序。

1.2 教学目标

通过本项目的学习，学生应掌握交互式智能家居机器嵌入式系统 Raspbian Linux 发行版本的安装、Linux 系统的基本操作、如何配置 Raspbian Linux 操作系统、如何通过 apt-get 管理应用软件和使用 Pi Store 安装软件、如何让控制系统连接网络、如何通过网络更新安装软件及如何使用自带的 Python 编写程序。后续复杂的功能都是基于 Raspbian 操作系统实现的，所以掌握本项目的内容尤为重要。本项目的内容为基础软件的安装，包括大量的安装过程和配置环节，学生可以按照本项目所讲的安装步骤进行同步操作，从而为后续更加复杂的内容打下坚实的基础。

1.3 知识准备

1.3.1 了解嵌入式系统 Linux 使用的命令行

本小节所介绍的命令和概念都是标准 Linux 系统中所使用的，具有很强的通用性，非常值得学习。

在开始前，先打开一个 LX(LXTerminal)终端(见图 1.3.1)，可以按照任务需要在 Raspbian Linux 系统的控制系统中实现部分软件的安装。在操作终端时，有两个基本技巧非常有用：自动补全与命令历史。通常情况下，只需要输入命令或者文件名的前几个字符，然后按【Tab】键，终端就会尝试以当前目录中的文件或常用目录中的程序(Shell 通常会在“/bin”和“/usr/bin”目录中搜索可执行程序)来补全所输入的字符串。按键盘上的上下箭头方向键，可以调用以前执行过的命令。如果不小心在命令中打错了字，可以用这个方法把错误的命令找出来，修改后再重新执行。

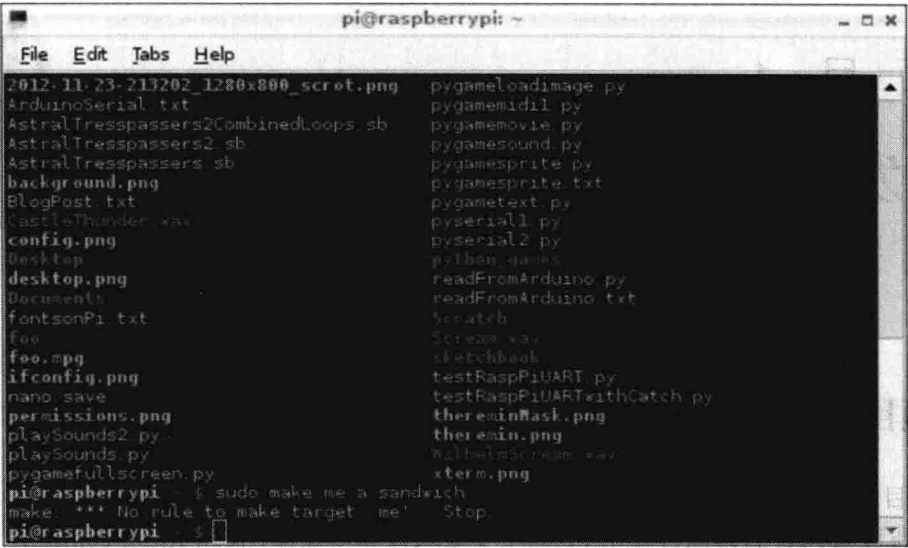


图 1.3.1 通过 LX（LXTerminal）终端访问命令行（Shell）

1.3.2 了解 Linux 文件与文件管理命令

表 1.3.1 列出了文件系统中的部分重要目录，它们中的大部分都是按照 Linux 标准来设置的，标准决定了一些文件应该放在哪个目录中，其中有一部分是 Raspbian Linux 所特有的。通过访问 “/sys” 目录中的文件，可以操作智能家居机器人控制板上的大部分硬件。

表 1.3.1 Raspbian 文件系统中的部分重要目录

目录	描 述
/	
/bin	所有用户都可以运行的程序和命令
/boot	系统启动时所需要的文件
/dev	表示系统上的各种设备的特殊文件
/etc	配置文件
/etc/init.d	启动各种服务的脚本
/etc/X11	X11 配置文件
/home	各用户的主目录
/home/pi	Pi 用户的主目录
/lib	内核模块与驱动程序
/media	可移动存储的挂载点
/proc	包含操作系统和各个运行中的进程相关信息的虚拟目录
/sbin	用于系统管理与维护的程序

续表

目录	描 述
/sys	Raspbian Linux 上的一个特殊的目录，可用于操作设备
/tmp	用于让各个程序创建临时文件
/usr	所有用户都可以使用的程序和数据
/usr/bin	包含了操作系统中大部分的程序
/usr/games	游戏都在这里
/usr/lib	用于支持常用程序库文件
/usr/local	存放一些这台机器专用的软件
/usr/sbin	更多系统管理程序
/usr/share	一些在多个程序间共享的数据，如图表与字体
/usr/src	Linux 是开放源代码的，这里面就包含了它的代码
/var	系统日志或脱机缓存文件
/var/backups	一些系统关键文件的备份
/var/cache	存放应用程序（如 apt-get 和网页浏览器）的缓存文件
/var/log	所有的系统及服务程序的日志
/var/mail	如果配置了电子邮件功能，所有用户的邮件都保存在这里
/var/spool	待处理的数据（如接收到的电子邮件、打印任务）

在命令提示符前面，可以看到当前所处的工作目录。在 Linux 文件中，主目录会被缩写为波浪线（~）。刚刚打开 LX 终端时，它会把当前的工作目录切换到主目录下，命令提示符会显示为：

```
pi@raspberrypi ~ $
```

有关命令提示符的详细解释如下：

```
pi①② raspberrypi③ ~③ $④
```

- ① 你的用户名：Pi，后面加上@符号。
- ② 你的主机名（默认的主机名为“Raspberrypi”）。
- ③ Shell 的当前工作目录，默认是主目录（~）。
- ④ 命令提示符：输入的一切内容都会出现在它的右侧。

按回车键执行输入的命令。

使用 cd（change directory，改变目录）命令可以在文件系统的各个目录间切换，下面的两条命令对于 Pi 用户来说具有相同的效果（把当前工作目录切换到主目录）：

```
cd /home/pi/  
cd ~
```

如果传给 cd 命令的路径以斜线开头，则表示这是一个绝对路径，否则这个路径被认为是相对于当前工作目录位置的相对路径。它可以用“.”和“..”分别表示当前目录和上一层目

录。例如，把当前目录切换到系统根目录：

```
pi@raspberrypi ~ $ cd ..
pi@raspberrypi /home $ cd ..
```

也可以直接用“/”切换到根目录：

```
pi@raspberrypi ~ $ cd /
```

切换完目录后，可以用ls命令列出当前目录下的文件：

```
pi@raspberrypi / $ ls
bin  dev  home  lost+found  mnt  proc  run  selinux  sys  usr
boot  etc  lib  media      opt  root  sbin  srv      tmp  var
```

大部分命令都允许增加一些参数或开关来改变默认的行为。例如，ls命令的“-l”参数可以让显示的文件列表更为详尽，显示出文件的大小、修改时间和权限：

```
pi@raspberrypi ~ $ ls -l
total 8
drwxr-xr-x 2 pi pi 4096 Oct 12 14:26 Desktop
drwxrwxr-x 2 pi pi 4096 Jul 20 14:07 python_games
```

用“-a”参数可以列出所有文件，包括隐藏文件：

```
pi@raspberrypi ~ $ ls -la
total 80
drwxr-xr-x 11 pi pi 4096 Oct 12 14:26 .
drwxr-xr-x 3 root root 4096 Sep 18 07:48 ..
-rw----- 1 pi pi 25 Sep 18 09:22 .bash_history
-rw-r--r-- 1 pi pi 220 Sep 18 07:48 .bash_logout
-rw-r--r-- 1 pi pi 3243 Sep 18 07:48 .bashrc
drwxr-xr-x 6 pi pi 4096 Sep 19 01:19 .cache
drwxr-xr-x 9 pi pi 4096 Oct 12 12:57 .config
drwx----- 3 pi pi 4096 Sep 18 09:24 .dbus
drwxr-xr-x 2 pi pi 4096 Oct 12 14:26 Desktop
-rw-r--r-- 1 pi pi 36 Sep 18 09:35 .dmrc
drwx----- 2 pi pi 4096 Sep 18 09:24 .gvfs
drwxr-xr-x 2 pi pi 4096 Oct 12 12:53 .idlerc
-rw----- 1 pi pi 35 Sep 18 12:11 .lessht
drwx----- 3 pi pi 4096 Sep 19 01:19 .local
-rw-r--r-- 1 pi pi 675 Sep 18 07:48 .profile
drwxrwxr-x 2 pi pi 4096 Jul 20 14:07 python_games
drwx----- 4 pi pi 4096 Oct 12 12:57 .thumbnails
-rw----- 1 pi pi 56 Sep 18 09:35 .Xauthority
-rw----- 1 pi pi 300 Oct 12 12:57 .xsession-errors
-rw----- 1 pi pi 1391 Sep 18 09:35 .xsession-errors.old
```

使用 `mv` 命令可以改变文件的名字；使用 `touch` 命令可以创建一个新的空文件：

```
pi@raspberrypi ~ $ touch foo
pi@raspberrypi ~ $ ls
foo Desktop python_games
pi@raspberrypi ~ $ mv foo baz
pi@raspberrypi ~ $ ls
baz Desktop python_games
```

删除文件用 `rm` 命令；删除空目录，可以使用 `rmdir` 命令。如果要删除非空目录，使用 `rm-r` 命令。传给 `rm` 命令的“-r”参数意味着要求 `rm` 程序递归进入所有的子目录删除所有的文件。

如果想了解一个命令所包含的所有参数，可以使用 `man` 命令（或在命令后面增加“-help”参数）：

```
pi@raspberrypi ~ $ man curl
pi@raspberrypi ~ $ rm --help
```

若要创建新的目录，则使用 `mkdir` 命令；若要把一个目录下的所有文件打包在一起，则用 `tar`（tape archives 的缩写）命令。常常能看到很多文件与源代码以 `tar` 包的形式发布，并且它们常常还会用 `gzip` 进行一次压缩。其操作步骤为：

```
pi@raspberrypi ~ $ mkdir myDir
pi@raspberrypi ~ $ cd myDir
pi@raspberrypi ~ $ touch foo bar baz
pi@raspberrypi ~ $ cd ..
pi@raspberrypi ~ $ tar -cf myDir.tar myDir
pi@raspberrypi ~ $ gzip myDir.tar
```

这时，就把 `myDir` 目录打包成了一个名为“`myDir.tar.gz`”的文件，这个文件可以很方便地通过网络或电子邮件传递。

1.3.3 更多 Linux 命令

Linux 如此成功的原因之一在于它的设计哲学：通过一系列可以组装在一起的简单小模块来构建一个非常复杂的系统。所以，需要学习一些有关“管道”和“重定向”的基础知识，它们是实现这个设计哲学的基础。

管道是把两个程序联系起来的一种方式，通过管道，可以把其中一个程序的输出作为另一个程序的输入。所有的 Linux 程序都可以从标准输入（通常表示为 `stdin`）读入数据，向标准输出（表示为 `stdout`）输出数据并把错误信息抛向标准错误输出（表示为 `stderr`）。通过管道可以把一个程序的 `stdout` 与另一个程序的 `stdin` 相连（见图 1.3.2）。管道的操作如下：

```
pi@raspberrypi ~ $ ls -la | less
```

（按【Q】键退出 `less` 程序）

还可以尝试通过管道传输更多的数据：

```
pi@raspberrypi ~ $ sudo cat /boot/kernel.img | aplay
```

运行这个命令前，建议先把音量调小一些。这个命令把系统内核对于 0 和 1 的数据传递

给音乐播放器，当成音乐来播放，这就是“内核之声”。

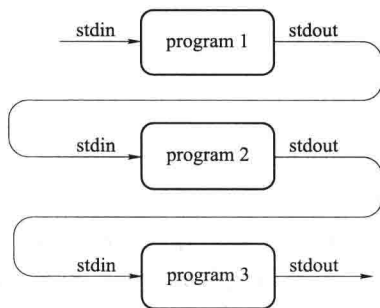


图 1.3.2 管道可以把小程序连接在一起并形成更大的任务

本书后面项目中的命令还会用到重定向，也就是指把程序的标准输出发送到一个文件中去。Linux 中有很多的东西都可以被当成文件来使用（如 Pi 的通用输入输出接口），所以重定向是一个非常有用的功能。可使用“>”操作符来实现重定向：

```
pi@raspberrypi ~ $ ls > directoryListing.txt
```

1.3.4 特殊的控制键

除了前面提到的自动补全（【Tab】键）和调用历史命令（光标向上键）的控制键，Shell 中还提供了一些常用的控制键。

（1）【Control-C】：中断当前正在运行的程序。在有些交互式应用程序（如文本编辑器）中其可能无效。

（2）【Control-D】：退出 Shell 软件。必须在命令提示符后紧跟着输入才有效（在“\$”提示符后不要输入“Control-D”以外的其他任何字符）。

（3）【Control-A】：把光标移动到行首位置。

（4）【Control-E】：把光标移动到行末位置。

当需要在屏幕上显示一个文件的内容，并且要分屏显示时，可以使用 less 命令：

```
pi@raspberrypi ~ $ ls > flob.txt
```

```
pi@raspberrypi ~ $ less flob.txt
```

如果想把一个文件的内容一次性全部输出到标准输出上，可以用 cat 命令（concatenate 的缩写）。在需要把一个文件的内容传到另一个程序中或者把它重定向到其他地方时，这个命令非常有用。

例如，用 cat 命令可以非常方便地实现把一个文件复制为另一个文件的功能（在第二个 cat 命令中，两个文件被连接成一个新的文件）。

```
pi@raspberrypi ~ $ ls > wibble.txt
```

```
pi@raspberrypi ~ $ cat wibble.txt > wobble.txt
```

```
pi@raspberrypi ~ $ cat wibble.txt wobble.txt > wubble.txt
```

用 tail 命令可以查看一个文件的最后几行内容（如一个日志文件中的最后几条），用 head 命令则可以查看前几行。如果要在一个或多个文件中搜索一个特定的字符串，可以用强大的 grep 命令：

```
pi@raspberrypi ~ $ grep Puzzle */*
```

grep 命令能支持正则表达式“非常”的语意，这使它成了一个非常强大的工具。正则表达式不太容易让人理解，这也是很多人认为 Linux 对初学者来说很难运用的一个重要原因。

1.3.5 了解 Linux 进程机制命令

智能家居机器人控制系统运行的每一个程序都是一个独立的进程，所以任何时候都有很多个进程同时运行。当把控制系统启动完成时，系统会运行大约 75 个进程，每个进程负责一项任务或提供一个服务。可以用 top 命令查看正在运行的进程，与此同时也可以看到 CPU 和内存的使用情况。top 命令能让人看到占用资源更多的进程，ps 命令可以用来列出所有的进程和它们对应的进程 ID (PID)：

```
pi@raspberrypi ~ $ ps -aux | less
```

如果要中断一个出错或失去响应的进程，可以用 ps 命令找出这个进程的 ID，然后用 kill 命令来中断它的运行：

```
pi@raspberrypi ~ $ kill 95689
```

对于一些系统进程，人们是没有权限去中断它的运行的，但可以通过稍后介绍的 sudo 命令来获得需要的权限。

1.3.6 了解 sudo 与权限

Linux 是一个多用户操作系统，其一条最基本的原则就是每个用户可以在他们自己的目录下创建、删除或者修改属于他们自己的文件。root 用户（管理员）可以修改文件系统中的任何文件，以避免错误地对文件进行操作，日常使用时应该尽量避免使用 root 用户登录。



如果以 pi 用户登录，很难对系统造成什么致命性的破坏，但以管理员身份操作则可能对系统造成严重的破坏，这也许是不小心的错误操作引起，也有可能是故意的。所以，在使用 sudo 命令时应该特别小心，尤其在移动或删除文件时，更应慎之又慎。当然，如果确实已经把系统破坏了，也还是可以通过 SD 卡安装新系统来修复的。

通过使用 sudo 或者类似的工具可以使普通用户临时获得管理员权限，以执行安装软件包之类的操作，这样就可以避免直接以 root 用户登录所带来的安全风险。在修改系统配置或与硬件打交道时，常常会使用到 sudo。

系统中的每一个文件都属于某一个用户或某一个用户组，用 chown 和 chgrp 命令可以修改文件所属的用户或用户组。若要执行这两个命令，则必须拥有 root 权限。

```
pi@raspberrypi ~ $ sudo chown pi garply.txt
```

```
pi@raspberrypi ~ $ sudo chgrp staff plugh.txt
```

每个文件都被设置了相关的权限，决定了一个文件是否可以被读、写或执行。这些权限可以按文件的拥有者、文件所属的用户组或全部用户分别设置（见图 1.3.3）。

rw	x	rw	x	rw	x	2	pi	pi	4096	Oct 14 21:29	foo
User			Group			User name		Group name	Size	Date	name
Everyone											

图 1.3.3 对文件所有者、文件所属组和全部用户的权限设置