

<<Shell脚本编程诀窍>>

图书基本信息

书名：<<Shell脚本编程诀窍>>

13位ISBN编号：9787302297819

10位ISBN编号：7302297819

出版时间：2012-9

出版时间：清华大学出版社

作者：帕克

页数：534

字数：840000

译者：万千

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Shell脚本编程诀窍>>

内容概要

《Shell脚本编程诀窍--适用于Linux\Bash等》介绍shell脚本编程，主要针对Bourne shell与POSIX兼容的shell，但也广泛涵盖了新近的一些发展情况，尤其是bash shell。bash shell几乎总是会包含在GNU/Linux操作系统中，也包含在了大多数商业Unix中。另外，KornShell也被广泛用于大部分这样的闭源或开源操作系统中。

Steve Parker的《Shell脚本编程诀窍--适用于Linux\Bash等》分为4个部分。

第 部分介绍shell的基本功能和语法；第 部分介绍shell脚本可以使用的工具；第 部分给出了一些涵盖更广泛话题的实用脚本；第 部分是参考信息。

<<Shell脚本编程诀窍>>

作者简介

作者：（英国）帕克（Steve Parker）译者：万千 Steve Parker，是具有20年Unix经验与15年GNU/Linux经验的Unix与Linux顾问。

他编写了在线shell脚本编程教程并对其进行维护。

Steve提供IT咨询服务，还提供shell脚本编程与Unix、Linux技术的培训课程。

<<Shell脚本编程诀窍>>

书籍目录

第1部分 基本概念

第1章 Unix、GNU和Linux的历史

1.1.1 Jnix

1.1.1.1 “一切皆文件”与管道

1.1.2 BSD

1.2 GNU

1.3 Linux

1.4 本章小结

<<Shell脚本编程诀窍>>

章节摘录

版权页：插图：17.1.3潜在的陷阱 正确地进行碰撞检测会有些困难，尤其是外星人宽度大于一个单元格的时候。

让屏幕保持干净也需要谨慎进行。

对屏幕进行太多刷新会毁掉游戏，因为会导致过度闪烁。

相对而言，调用clear命令会花较长时间，并且使屏幕非常闪烁。

编写代码时，该脚本的最显著的改变在于for循环中模功能从函数调用转移为外部expr命令。

这意味着对每个外星人（无论死活）调用expr，然后替换为内嵌的（（...%2））结构。

从expr到内嵌方法的修改意味着可以改变外星人的形状。

不修改的话，执行会非常缓慢。

将模值从for循环中取出也会效率更高一些，但不会特别明显。

17.1.4脚本结构 脚本的开头与结尾是tput命令。

它可以使光标在游戏开始前消失（tput civis），然后在退出后又出现（tput cvvis）。

另一个较小的修改是在游戏结束时取消SIGALRM上的trap，这样move函数在脚本结束后就不会发送SIGALRM信号。

这些只是小细节，但可以对效果有相当大的改观。

否则，如果没有它们，效果会差很多。

该脚本由4个关键函数与一个主循环构成。

从脚本的底部往上，主循环只从键盘读取一个字符（read—n 1）。

如果是“左”或“右”指令（分别为a和1），则循环更新飞船的位置。

飞船将立刻被重新绘制。

如果按下的是开火按钮（f），且激光炮不处于使用状态（cannonY—eq0），则cannonX变量相对于飞船当前的x轴坐标进行赋值，cannonY被赋值为飞船的Y轴坐标（固定在屏幕的底部）。

只要按键移动了飞船，主循环就调用drawship。

该函数用printf语句清空屏幕的整个底行，并将飞船中的激光炮用彩色显示，用以表示是否装配了火炮

。这与move函数是独立的，并且能实时地更新飞船的移动，且与单调缓慢的外星飞船的更新不同。

drawship也在move函数中调用，这样即使在飞船没有移动的情况下，火炮的更新也能正常反映出来。

主循环的开头是move函数。

move函数使用SIGALRM在\$DELAY秒之后调用自身。

DELAY会随着时间推移越来越小，所以外星人下降的速度越来越快。

每次调用move时，外星人在它们的前进方向上移动一格。

当它们到达屏幕的边界时，就翻转direction变量使它们向相反的方向运动。

外星人在每次抵达屏幕的右边时都会向下移动一行（通过增加ceilin9变量）。

move随后对每行外星人调用一次drawrow函数。

因为bash没有多维数组，所以外星人的行数被硬编码到脚本中。

使用循环来遍历所有行是一个不错的选择，就算有6行外星人，6次调用drawrow函数也不会显得太笨拙

。在显示过程中，drawrow返回碰到炮弹的所有外星人的索引号，如果当前行中没有外星人被击中则返回0。

这样对数组的结构产生了一些额外的效果。

数组从0开始索引，但因为0在drawrow函数的返回码中具有特殊意义，所以保存外星人分值的数组row0到row5不使用0号索引。

这或许有些拙劣，但可以使每一行在调用drawrow之后进行直接赋值rowx[\$?

]=0。

如果\$?

<<Shell脚本编程诀窍>>

为0，则更新未使用的0号索引，对其他数组元素没有影响。

如果\$?

大于0，则它会指向某个外星人，然后将存储其得分值的数组元素赋值为0。

这样一来，随后对drawrow的调用不会在原来位置显示外星人，并且碰撞检测会允许炮弹通过该位置而不停下来。

更简洁的实现方法要求对drawrow的每次调用之后都有一些更加复杂的代码。

这些代码会检查返回值，并且在外星人被击中的情况下更新数组。

该脚本使用的方法使得游戏在最低限度上能快速运行，并且更重要的是易于阅读与管理。

<<Shell脚本编程诀窍>>

编辑推荐

《Shell脚本编程诀窍:适用于Linux、Bash等》是为了对shell进行较为全面的介绍，并且无论用户具有何种背景与经验都能从《Shell脚本编程诀窍:适用于Linux、Bash等》中得到一些收获。

《Shell脚本编程诀窍:适用于Linux、Bash等》主要面向中级与高级Unix与Linux管理员，以及可能感兴趣的其他高级用户。

《Shell脚本编程诀窍:适用于Linux、Bash等》假设读者至少会用一种Unix系统，并且可能已经编写了一些shell脚本，但希望提高自身脚本编写的水平。

<<Shell脚本编程诀窍>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>