Linux循序渐进(19) PDF转换可能丢失图片或格式，建议阅读原文
https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E5_BE_AA_E5_BA_c103_144223.htm shell是用户和Linux操作系统之间的接口。Linux中有多种shell，其中缺省使用的是Bash。本章讲述了shell的工作原理，shell的种类，shell的一般操作及Bash的特性。 什么是shell Linux系统的shell作为操作系统的外壳，为用户提供使用操作系统的接口。它是命令语言、命令解释程序及程序设计语言的统称。 shell是用户和Linux内核之间的接口程序，如果把Linux内核想象成一个球体的中心，shell就是围绕内核的外层。当从shell或其他程序向Linux传递命令时，内核会做出相应的反应。 shell是一个命令语言解释器，它拥有自己内建的shell命令集，shell也能被系统中其他应用程序所调用。用户在提示符下输入的命令都由shell先解释然后传给Linux核心。 有一些命令，比如改变工作目录命令cd，是包含在shell内部的。还有一些命令，例如拷贝命令cp和移动命令rm，是存在于文件系统中某个目录下的单独的程序。对用户而言，不必关心一个命令是建立在shell内部还是一个单独的程序。 shell首先检查命令是否是内部命令，若不是再检查是否是一个应用程序（这里的应用程序可以是Linux本身的实用程序，如ls和rm，也可以是购买的商业程序，如xv，或者是自由软件，如emacs）。然后shell在搜索路径里寻找这些应用程序（搜索路径就是一个能找到可执行程序的目录列表）。如果键入的命令不是一个内部命令并且在路径里没有找到这个可执行文件，将会显示一条错误信息。如果能够成功找到命令，该内部命令或应用程序将被分解为系统调用

并传给Linux内核。 shell的另一个重要特性是它自身就是一个解释型的程序设计语言，shell程序设计语言支持绝大多数在高级语言中能见到的程序元素，如函数、变量、数组和程序控制结构。shell编程语言简单易学，任何在提示符中能键入的命令都能放到一个可执行的shell程序中。 当普通用户成功登录，系统将执行一个称为shell的程序。正是shell进程提供了命令行提示符。作为默认值（TurboLinux系统默认的shell是BASH），对普通用户用“ $ ”作提示符，对超级用户（root）用“ # ”作提示符。 一旦出现了shell提示符，就可以键入命令名称及命令所需要的参数。shell将执行这些命令。如果一条命令花费了很长的时间来运行，或者在屏幕上产生了大量的输出，可以从键盘上按ctrl c发出中断信号来中断它（在正常结束之前，中止它的执行）。 当用户准备结束登录对话进程时，可以键入logout命令、exit命令或文件结束符（EOF）（按ctrl d实现），结束登录。 我们来实习一下shell是如何工作的。 $ make work make:***No rule to make target‘ work. Stop. $ 注释：make是系统中一个命令的名字，后面跟着命令参数。在接收到这个命令后，shell便执行它。本例中，由于输入的命令参数不正确，系统返回信息后停止该命令的执行。 在例子中，shell会寻找名为make的程序，并以work为参数执行它。make是一个经常被用来编译大程序的程序，它以参数作为目标来进行编译。在“ make work ”中，make编译的目标是work。因为make找不到以work为名字的目标，它便给出错误信息表示运行失败，用户又回到系统提示符下。 另外，用户键入有关命令行后，如果shell找不到以其中的命令名为名字的程序，就会给出错误信息。例如，如果用户键入： $

myprog bash:myprog:command not found $ 可以看到，用户得到了一个没有找到该命令的错误信息。用户敲错命令后，系统一般会给出这样的错误信息。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com