

《Linux基础》



第三讲 命令终端基础

终端的概念

- 终端 (Terminal) 也称终端设备, 是计算机网络中处于网络最外围的设备, 主要用于用户信息的输入以及处理结果的输出等。
- 在早期计算机系统中, 由于计算机主机非常昂贵, 因此一个主机 (如IBM大型计算机) 一般会配置多个终端, 这些终端本身不具备计算能力, 仅仅承担信息输入输出的工作, 运算和处理均由主机来完成。
- 在个人计算机时代, 个人计算机可以运行称为终端模拟器的程序来模拟一个终端的工作。这样的终端叫做虚拟终端。
- 右图是VT100物理终端



虚拟控制台 (virtual console)

- Linux提供了虚拟控制台，桌面环境的Ubuntu系统上使用Ctrl+Alt+[F1~F7]会在不同虚拟控制台之间切换。
- Ctrl+Alt+F7就是默认的桌面环境。

终端，控制台，shell

- 终端是一个**设备**。而shell是一个程序，主要就是获取用户输入的命令并运行，把结果返回给用户。终端最重要的事情是输入和输出，程序运行的结果要通过shell显示在终端上以此和用户产生交互。
- shell有很多不同的实现，基本功能都相同。Linux默认的shell大多都是bash。类似的还有sh、zsh、fish等。
- 早期的计算机多个终端通过串口连接到主机上，而和主机直接配套的输入输出设备是控制台。早期设备的控制台往往具备很多控制按键、开关等可以从硬件层面直接控制设备。而现在操作系统的虚拟控制台完成和终端相同的工作，有时候把虚拟控制台也成为虚拟终端。
- Ubuntu桌面环境自带一个terminal软件，是终端模拟器（虚拟终端），启动后会运行bash。

Linux命令分类

- 内部命令：属于Shell解释器的一部分，是shell程序自带的命令。
- 外部命令：独立于Shell解释器之外的程序文件，是在系统的某个路径下的可执行程序，由shell依照事先设定好的路径查找，找到后才能执行。
- type <command> 查看<command>是内部命令还是外部命令。hash表会缓存外部命令所在位置。刚开机时，hash表为空，每次运行命令，hash表会缓存此命令的路径，提高命令查找效率。
- which/whereis <command> 查看<command>命令所在的路径。

Linux命令格式

- Linux命令的通用格式为：命令 [选项] [参数]
- 选项的含义：
 - 选项：用于调节命令的具体功能
 - 以“-”引导短格式选项（单个字符），例如“-l”
 - 以“--”引导长格式选项（多个字符），例如“--color”
 - 多个短格式选项可以写在一起，例如“-al”，但是这和程序设计方式有关，如果程序不支持这种格式则不能这样使用。
- 参数：命令操作的对象，如文件、目录名等。
- 选项和参数都是人为的划分，对程序来说，比如C语言的main(int argc, char *argv[])函数，后面的选项和参数都会传递到argv，都是作为main函数的参数传递的。

获得命令帮助

- 使用help 查看Bash内部命令的帮助信息
- 命令的“--help” 选项适用于大多数外部命令 。
- 使用man命令阅读手册页：man [COMMAND]
 - 使用“↑”、“↓”方向键滚动文本
 - 使用Page Up和Page Down键翻页
 - 按q键退出阅读环境、按“/”键后查找内容
- 示例：type test查看test是shell内建命令，则使用help test查看帮助信息，而ls是独立于shell的命令，则使用man ls查看手册。

命令搜索路径

- bash会根据PATH变量的设置自动寻找输入的命令。如果有同名的命令，按照路径顺序找到后返回执行，不再继续寻找。有同名的指令可以输入路径运行。
- .profile记录了bash会在哪些目录查找命令。
- 默认的路径搜索顺序(用 ; 隔开)：
 - ~/bin ; ~/.local/bin
 - /usr/local/sbin ; /usr/local/bin ; /usr/sbin ; /usr/bin
 - /sbin ; /bin ; /usr/games ; /usr/local/games

添加其他路径

- 如果在用户主目录下存在ins目录，并希望添加到PATH环境变量中，可以在用户主目录下找到.profile文件，使用文本编辑器打开。
- 在PATH=后面的双引号中的字符串末尾添加 :/home/brave/ins，保存文件。
- 其中/home/brave是用户主目录，个人使用应换成自己的用户名。
- 或者是直接使用\$HOME/ins，\$HOME会自动替换为当前用户的主目录。
- 然后运行source .profile
- 这样就可以直接在shell中输入ins目录下存在的程序名称被shell执行了。
- 可以使用env命令显示系统中已存在的环境变量。

shell使用小技巧

- Tab键可以自动补全命令名称以及参数
- 命令历史：bash中可以使用↑、↓按键逐条翻看，允许编辑并重复执行，history命令显示所有缓存在.bash_history中的命令，history -c 清空历史命令。
- 快捷键：
 - Ctrl+a 跳转行首，Ctrl+e 跳转行尾
 - Ctrl+l 清屏
 - Ctrl+c 终止程序执行
 - Ctrl+d 退出shell，回到登录界面

终端常用命令

| 命令 | 说明 |
|-------|--------------------------------|
| ls | 列出文件、目录的信息 |
| cd | 切换当前工作目录 |
| mkdir | 创建目录 |
| rmdir | 删除空目录 |
| touch | 创建空文件 |
| rm | 删除给定的文件和目录。如果目录不为空，需要使用rm -rf |
| cp | 将一个或多个源文件或者目录复制到指定的目的文件或目录 |
| mv | 对文件或目录重新命名，或者将文件从一个目录移到另一个目录中 |
| cat | 连接文件并打印到标准输出设备上。cat经常用来显示文件的内容 |
| more | 以全屏幕的方式按页显示文本文件的内容 |
| less | 以全屏幕的方式按页显示文本文件的内容，功能比more更强大 |
| find | 在指定目录下查找文件 |

常用命令使用示例

| | |
|-------------------------|---------------------------------------|
| ls /usr | 显示/usr的目录内容 |
| ls -R /usr/ | 递归显示目录内容 |
| ls -l /usr | 显示目录/文件详细信息 |
| mkdir abc | 在当前目录下创建abc目录 |
| rmdir abc | 删除当前目录下的abc目录，abc必须是空目录 |
| rm tmp/a.sh | 删除tmp目录下的a.sh文件 |
| rm -rf tmp/ | 删除tmp目录，目录可以不为空，会删除目录下所有的内容 |
| cp tmp/a.sh sh/ | 复制tmp/a.sh文件到sh目录，tmp和sh都是当前目录的子目录 |
| cp tmp/a.sh /usr/local/ | 复制tmp/a.sh到/usr/local，/usr/local是绝对路径 |
| cp tmp/a.sh sh/b.sh | 复制tmp/a.sh到sh/目录并重命名为b.sh |

常用命令使用示例

| | |
|--|--------------------------|
| <code>cp mariadb/ -R /usr/local</code> | 递归复制mariadb目录到/usr/local |
| <code>mv tmp/a.sh sh/</code> | 移动文件 |
| <code>mv tmp/a.sh sh/b.sh</code> | 移动文件并重命名 |
| <code>mv tmp/a.sh tmp/c.sh</code> | 重命名文件 |
| <code>touch tmp/test.sh</code> | 在tmp目录创建test.sh空文件 |
| <code>cat /etc/passwd</code> | 显示/etc/passwd文件的内容 |
| <code>find /usr -name gcc</code> | 查找名称为gcc的文件，大小写敏感 |
| <code>find /usr -name gcc*</code> | 查找名称gcc开头的文件，大小写敏感 |
| <code>find /usr/ -iname gcc*</code> | 查找名称开头gcc的文件，大小写不敏感 |