

在所有的问题之中，最先要解决就是：如何才能够让 Linux 做我们想让它做的事情？

我们可以用各种方式做到这一点：例如，一个 Linux 程序员，可以用编程语言通过操作系统的编程接口与 Linux 进行交互。如果，如果足够牛的话，甚至可以考虑撇开编程接口，直接与 Linux 内核进行交互。但是，作为一个普通的 Linux 系统管理员，我们还是倾向于通过现成的操作命令接口（User Command Interface）与 Linux 打交道。与 Linux 的“核 (kernel)”对应，操作命令接口很形象地被称为“壳 (shell)”。

简单来说，Shell 就是一种能让你与 Linux 之间进行“对话”的软件。Shell 等待你输入，向 Linux 解释你的输入，并且处理各种各样的系统的输出结果，把这些结果展示给你。这种交互可以是即时交互（从键盘输入，并且可以立即得到响应），也可以是非即时交互（shell 脚本<sup>1</sup>）。

我们所使用的 CentOS 系统上的默认 Shell 叫做 Bash(The GNU's Bourne Again Shell)。Bash 也是绝大部分 Linux 平台上的默认 Shell。

### 注意



- ◆ 事实上，Linux 为用户提供了两种截然不同的“壳”：一种是命令行式 shell（Command Line Interface shell，即 CLI shell），用户利用这些字符命令或者 shell 脚本来组织和控制作业的执行，或者对计算机系统进行管理。另一种是图形界面 shell（Graphical User Interface shell 即 GUI shell），用户通过点击，拖动图像图标来完成同样的事情。
- ◆ 在 Linux 中，传统意义上的 shell 指的就是命令行式的 shell，在下面课程中，如果没有特别注明，所提到的 shell 就是命令行式的 shell。

当文本模式下登入系统或在图形界面下打开终端，首先将看到的是一个提示符(prompt)。提示符本身就包含了相当多的信息：CentOS 的标准提示符包括了登录用户名、登入的主机名、用户当前所在的目录(working directory)和提示符号，例如：

命令序列 错误!文档中没有指定样式的文字。 -1

```
[stu@localhost~]$
```

其中，'Stu'为登录用户名，'localhost'为登录主机名，'~' 表示当前用户正处在 stu 用户的 家目录中，'\$'则表示当前登录用户为普通用户。



### 注意

- ◆ 根据 Bourne Shell 的传统，普通用户的提示符以'\$'结尾，而根用户以'#'结尾
- ◆ '~' 符号不是一个固定的目录名称，而是一个“Shell 变量”，代表使用者的“家目录”，举例来说，如果我以 dev01 用户身份登录系统，dev01 的家目录是'/home/dev01'，那么 '~'就代表'/home/dev01'，如果我以

<sup>1</sup> Shell 脚本是放在文件中的一串 shell 和操作系统命令，它们可以被重复使用。本质上，shell script 是命令行命令简单的组合到一个文件里面。

---

---

root 身份登录系统，而 root 的家目录是'/root'，那么 '~'就代表'/root'。至于到底“家目录”和“Shell 变量”到底是什么，我们在后继的教程中会详细讨论。

---

---

这时就可以输入命令了，为熟悉 shell，我们选择的三个有代表性的命令来小试牛刀，这三个命令分别为 whoami、echo 和 nano。



### 注意

- ◆ Linux 中所谓的命令(Command)，本质上就是一个与命令同名的可执行（Runnable）的二进制文件或者 Shell 脚本。
- ◆ 这些可执行文件一般都放在/bin、/sbin、/usr/bin 或者/usr/sbin 等目录中，当要执行命令的时候，Shell 就会在这些目录或者其他用户指定的目录中（我们在后面会详细讨论如何指定这写目录）搜索相应可执行文件，找到以后就运行，如果找不到文件，那么 Shell 就会在终端中输出相应的错误信息。

---

---

首先来试一试 whoami，在命令提示符后键入 whoami 命令，然后回车，命令将输出当前登录用户的用户名：

命令序列 错误!文档中没有指定样式的文字。-2

```
[stu@localhost ~]$ whoami
stu
[stu@localhost ~]$
```

whoami 是 Linux 中最简单的命令之一，不能带参数，只有两个选项，所以我们可以看得很清楚，要通过 Shell 给 Linux 下达命令很简单：只要在提示符后敲进命令，然后再回车，就可以执行这个命令，并在终端里输出相应的结果（如果有的话），命令结束后，再给出新的提示符；如果出错，那么就会在终端中输出相应的错误信息，然后再给出新的提示符。

命令序列 错误!文档中没有指定样式的文字。-3

```
[stu@localhost ~]$ whoAmi
bash: whoAmi: command not found
[stu@localhost ~]$
```



### 注意

- ◆ 切记，Linux 区分大小写。换一句话说，whoami 和 WHOAMI 或 whoAmi 是不一样的。



### 重要

- ◆ 选项(option): 包括一个或多个字母的代码，一般来说前面有一个 '-'
- 
-

---

符号或者'--'符号（'-'或者'--'符号是必要的，Linux 用它来区别选项和参数），选项可以看作是一种开关，一般用于控制命令的行为。事实上，Linux 中的命令在正常情况下最少有两个选项，--help 和—version。

- ◆ 参数(parameter)：一个字符串，参数一般用来向命令传递一些时运行所需的信息（例如文件或主机名称等），Linux 大多数命令都被设计为可以接纳参数。
  - ◆ 命令本身、选项和参数中间用空格来分隔，不论空多少格，shell 都视为一格；
- 



## 命令

---

用法：whoami [选项]...

显示与当前的有效用户 ID 相关联的用户名。

- ◆ --help 显示此命令帮助信息并退出
  - ◆ --version 显示命令版本信息并退出
- 

接接下来来看第二个命令 echo，这个命令就相对复杂许多了，不仅有很多选项，还可以跟参数。我们首先来看命令最简单的使用方法，在命令提示符后键入 echo 命令，空格之后跟上一个字符串"Hello, Linux world."，然后回车，命令将在屏幕上回显此字符串"Hello, Linux world." 这个命令带了一个参数就是字符串"Hello, Linux world."

命令序列 错误!文档中没有指定样式的文字。-4

```
[stu@localhost ~]$ echo "Hello, Linux world."
Hello, Linux world.
[stu@localhost ~]$
```

echo 命令可以带选项，如在 date 后跟上'-n'选项，表示在回显指定字符串后不换行：

命令序列 错误!文档中没有指定样式的文字。-5

```
[stu@localhost ~]$ echo -n "Hello, Linux world."
Hello, Linux world.[stu@localhost ~]$
```

或者跟上'-e'选项，来输出以反斜杠开头的转义字符，其中，'\t'表示制表符，'\n'表示换行：

命令序列 错误!文档中没有指定样式的文字。-6

```
[stu@localhost ~]$ echo -e 序号\t姓名\t成绩\t备注\n1\t张三\t59\t补考
序号 姓名 成绩 备注
1 张三 59 补考
[stu@localhost ~]$
```



## 注意

- ◆ 所有的 ASCII 码都可以用 “\” 加数字（一般是 8 进制数字）来表示。而 C 语言中定义了一些字母前加“\”来表示常见的那些不能显示的 ASCII 字符，如 \0, \t, \n 等，就称为转义字符，因为后面的字符，都不是它本来的 ASCII 字符意思了。
- ◆ echo 命令中的转义字符与 C 语言中的转义字符基本一样，可以参考 echo 命令的帮助文档



## 命令

用法: /bin/echo [选项]... [字符串]... 或: /bin/echo 长选项  
将 STRING 回显到标准输出。

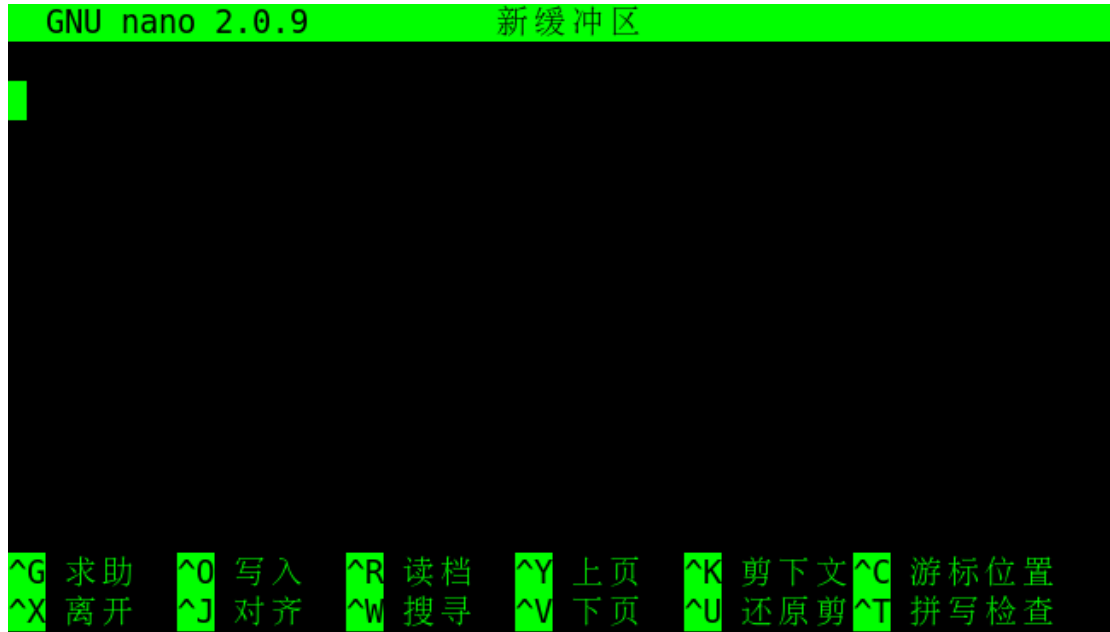
- ◆ -n 不尾随换行符
- ◆ -e 启用解释反斜杠的转义功能
- ◆ -E 禁用解释反斜杠的转义功能(默认)
- ◆ --help 显示此帮助信息并退出
- ◆ --version 显示版本信息并退出

若 -e 可用，则以下序列即可识别：

- ◆ \\ 反斜杠
- ◆ \a 响铃声
- ◆ \b 退格
- ◆ \c 不再产生新的输出
- ◆ \e 转义符
- ◆ \f 换页
- ◆ \n 新行
- ◆ \r 回车
- ◆ \t 水平制表符
- ◆ \v 竖直制表符
- ◆ \0NNN 字节数以八进制数 NNN (1 至 3 位)表示
- ◆ \xHH 字节数以十六进制数 HH (1 至 2 位)表示

注意：BASH 中可能 内含自己的 echo 程序版本，它会覆盖这里所提及的相应版本，可能在命令具体行为上有些许区别。

最后我们来看 nano 命令，nano 是一个文本编辑工具，我们在第一部分的知识点 4 中提到过。nano 命令与上面两个命令不太一样，这个命令有自己的界面，我们在命令行下输入 nano，回车，就来到了 nano 界面中，这时我们就可以在光标处输入文本了，我们可以用箭头键来移动光标，也可以按 Alt+M 打开鼠标支持来移动光标：



命令序列 错误!文档中没有指定样式的文字。-7 nano 界面

输入一段文字后，如果要保存，就按 **Ctrl+O**，填入文件名，回车，显示“已写入××行”字样，表示文件保存成功。