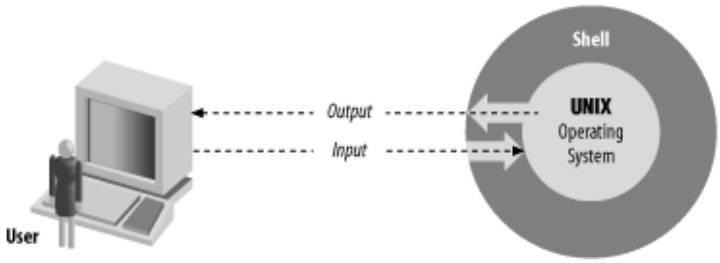
简介

什么是Shell?

Shell是命令解释器(command interpreter), 是Unix/Linux操作系统的用户接口, 程序从用户接口得到输入信息, shell将用户程序及其输入翻译成操作系统内核(kernel)能够识别的指令, 并且操作系统内核执行完将返回的输出通过shell再呈现给用户,下图所示用户、shell和操作系统的关系:



Shell也是一门编程语言, 即shell脚本, shell是解释执行的脚本语言, 可直接调用linux命令. 一个系统可以存在多个shell, 可以通过cat /etc/shells 命令查看系统中安装的shell, 不同的shell可能支持的命令语法是不相同的.

Shell种类

操作系统内核(kernel)与shell是独立的套件, 而且都可被替换. 不同的操作系统使用不同的shell; 同一个kernel之上可以使用不同的shell. 常见的shell分为两大主流:

sh

- Bourne shell(sh) , Solaris, hpux默认shell
- Bourne again shell(bash）), Linux系统默认shell

csh

- C shell(csh)
- tc shell(tcsh)

查看使用Shell

```
wuyijie@wuyijiedeMacBook-Pro ~ % echo $SHELL
/bin/zsh
```

shell脚本编程

同传统的编程语言一样，shell提供了很多特性，这些特性可以使你的shell脚本编程更为有用.

创建Shell脚本

```
touch test.sh
```

shell 脚本一般是以.sh结尾的文本文件，但它并不是必须的，因为Linux/Unix系统并不以文件后缀名来区分文件，它该是什么就是什么.

首行

#!/bin/bash

#!符号能够被内核识别成是一个脚本的开始，这一行必须位于脚本的首行，/bin/bash是bash程序的绝对路径，在这里表示后续的内容将通过bash程序解释执行.

注释

注释符号#放在需注释内容的前面

内容

可执行内容和shell结构

Shell脚本的权限

一般情况下，默认创建的脚本是没有执行权限的

```
wuyijie@wuyijiedeMacBook-Pro ~ % ls -l| grep test.sh
-rw-r--r--   1 wuyijie  staff     0  2 13 20:30 test.sh
```

没有权限不能执行，需要赋予可执行权限

```
wuyijie@wuyijiedeMacBook-Pro ~ % ls -l| grep test.sh
-rwxr-xr-x   1 wuyijie  staff     0  2 13 20:30 test.sh
```

Shell脚本的执行

1.输入脚本的绝对路径或相对路径

```
wuyijie@wuyijiedeMacBook-Pro ~ % /Users/wuyijie/test.sh
```

```
2 Hello World!
```

2.bash或sh +脚本

```
1 wuyijie@wuyijiedeMacBook-Pro ~ % bash test.sh
2 Hello World!
3 wuyijie@wuyijiedeMacBook-Pro ~ % sh test.sh
4 Hello World!
```

注:当脚本没有x权限时, root和文件所有者通过该方式可以正常执行

在脚本的路径前再加"." 或source

```
1 wuyijie@wuyijiedeMacBook-Pro ~ % ./test.sh
2 Hello World!
3 wuyijie@wuyijiedeMacBook-Pro ~ % source test.sh
4 Hello World!
```

区别：第一种和第二种会新开一个bash, 不同bash中的变量无法共享. 但是使用./脚本.sh 这种方式是在同一个shell里面执行的