# 超级用户（root）提示符

$ 普通用户提示符

Shell 脚本将许多指令汇集到一起，让使用者能够一个操作从而执行多个命令，方便管理员进行设置和管理。直接用Shell来写程序，编写的程序效率很高。

Shell脚本一般由以下几部分构成：

（1）Shell关键字：例如if...else、for do...done

（2）Shell命令：例如export、echo、exit、pwd、return

（3）Linux命令：例如date、rm、mkdir

（4）文本处理功能：例如awk、cut、sed、grep

（5）函数：通过函数把一些常用的功能放在一起。例如，/etc/init.d目录中的大部分或全部系统Shell脚本所使用的函数都包含在文件/etc/initd/functions中。

（6）控制流语句：例如 if.….then.….else或执行重复操作的 Shell 循环。

shell补充的快捷键

ESC+T：调换光标前的两个单词

小实验：运行第一个shell脚本程序（同时复习VIM的用法）

（1）桌面上新建文本文件，命名为：test

（2）在桌面打开终端

（3）vim打开进入命令模式

（4）i 进入输入模式

输入命令

#!/bin/bash

（脚本第一行：告诉Shell使用什么程序解释脚本的特别提示。）

echo hello,world!

esc 由输入模式返回到命令模式

:wq底线命令模式，保存并退出

1. 执行脚本

第一种：采用bash+脚本相对路径或绝对路径

sh test

第二种：采用脚本绝对路径或相对路径直接运行(需要赋予文件执行权限）

./test

第三种：采用source或. +脚本相对路径或绝对路径

source ./test

. ./test

变量-父子Shell

在Shell环境内嵌套一个Shell，那么第一个Shell就是新开Shell的父Shell，而新开的Shell就是第一个Shell的子Shell。子Shell和父Shell最大的区别就是，环境变量的集成关系，如在子环境设置的当前变量，父环境变量是不可见的。

小实验：

ps -f

查看父子shell

变量

变量为程序或脚本访问内存中的可被修改的一块数据提供了简单的方式。Shell内的变量可以被指定为任意的数据类型。

Shell变量有两种的类型：系统变量（环境变量）、用户自定义的变量（本地变量或Shell变量）

从变量的作用域角度：有全局变量和局部变量。

小实验：

env 输出所有变量

printenv 输出所有变量

printenv LANGUAGE 输出LANGUAGE 变量的值：zh\_CN

变量-赋值

可以使用操作符“=”给变量赋值。输入的次序是：变量名、赋值操作符、赋予的值。在赋值操作符“=”的周围，不要有任何空格，否则将会出现错误。

小实验：

varNname=varValue

echo $varName

输出：varValue

错误示范：

错误变量定义：

varName = varValue

varName =varValue

varName= varValue

变量-命名规则

变量名必须以字母或下划线字符“\_”开头，后面跟字母、数字或下划线字符，第一个字符不能为数字。不要使用?、\*和其他特殊字符命名变量。

有效的Shell变量名示例如下:

USERNAME

LD\_LIBRARY\_PATH

\_var

var1

无效的变量名示例如下:

?var=123

User\*name=yantaol

变量-只读变量和撤销变量

定义只读变量语法（定义后无法被更改）：

readonly varName=varValue

撤销变量(不能删除只读变量）

unset varName

小实验：

readonly var\_r=read

echo $var\_r

var\_r=read1

出现提示：只读变量

unset var\_r

出现提示：无法取消设定：只读...

var\_unset =unset1

echo $var\_unset

unset var\_unset

echo $var\_unset

变量-特殊变量

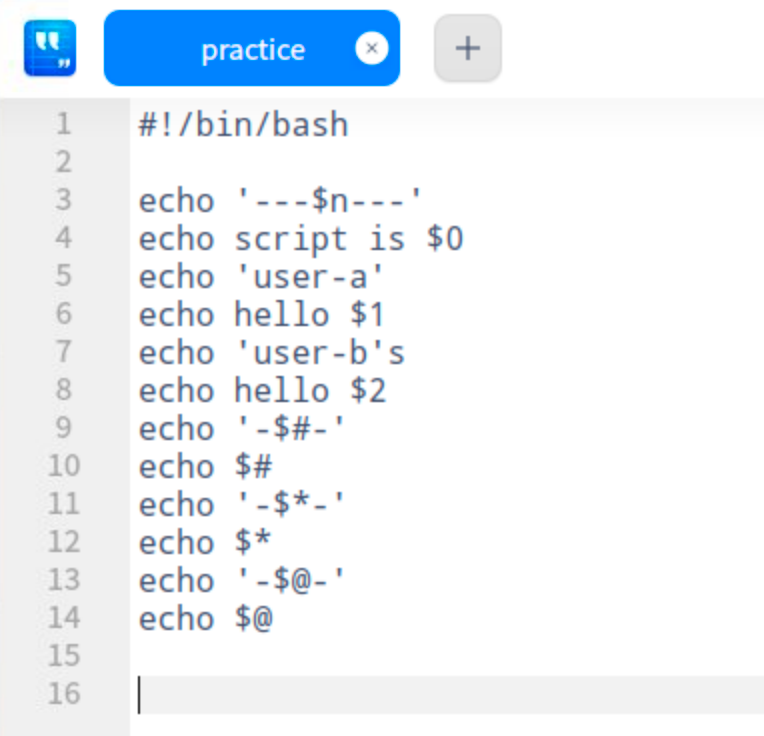
$n：n为数字，$0代表脚本名称，$1-$9代表1-9个参数，十以上的参数，需要用大括号包含,如${10}，主要作用是读取脚本的传入参数。

$#：获取所有参数的个数，常用于循环及判断参数的个数

$\*：代表命令行中的所有参数，把参数看作一个整体

$@：代表命令行中的所有参数，把参数整理为一个集合，常循环使用

$?：shell内查看最后一次脚本执行的状态，如果返回0，则正常，如果返回其他数值则表示异常



（需要重新截图，变量输入的内容可以替换）