第5篇　Linux Shell编程

本章主要内容

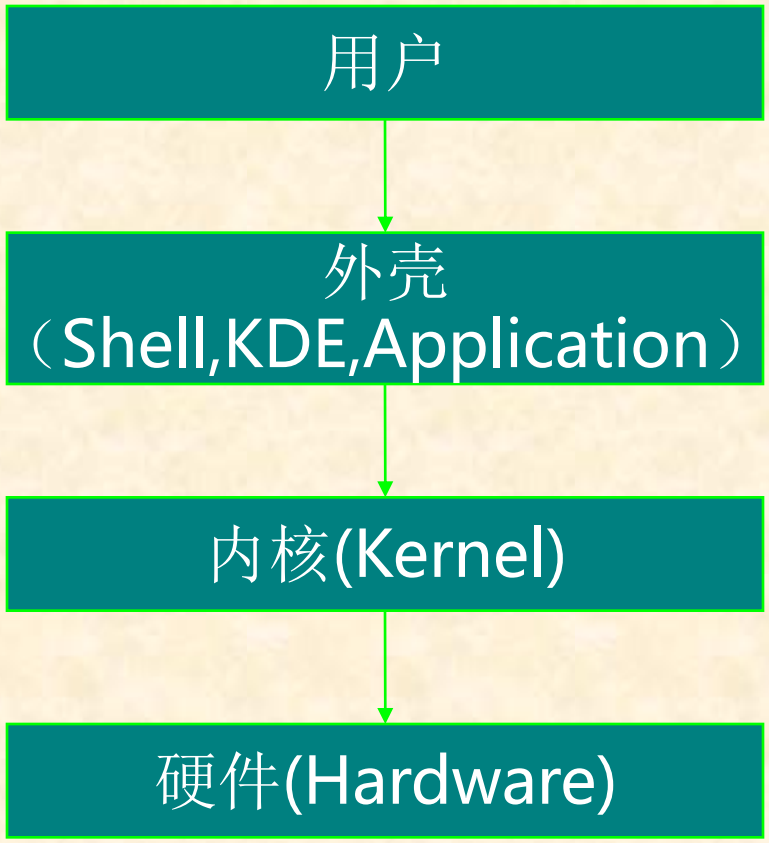
- Shell综述
- Shell变量
- Shell语法-算数运算
- Shell语法-逻辑判断
- Shell语法-控制语句
- Shell语法-函数
- Shell调试

课程进度

- Shell综述
- Shell变量
- Shell语法-算数运算
- Shell语法-逻辑判断
- Shell语法-控制语句
- Shell语法-函数
- Shell调试

Shell综述

- 命令解释器俗称外壳程序
- 用户通过shell向kernel发送指令，kernel再向硬件发送指令
- Shell版本众多，例如常听到的Bourne Shell（sh）、在Sun中默认的C Shell、 商业上常用的K SHell以及TCSH 等，每一种Shell都各有其特点。
- Linux 使用的是Bourne Again Shell（简称 bash），这个 Shell是Bourne Shell的增强版本，是基于GNU的架构下发展出来的。



Shell综述

- shell的执行方式有4 种：
  - $ bash < 脚本名 [参数]
  - $ bash 脚本名 [参数]
  - $. 脚本名 [参数]

课程进度

- Shell综述
- **Shell变量**
- Shell语法-算数运算
- Shell语法-逻辑判断
- Shell语法-控制语句
- Shell语法-函数
- Shell调试

Shell变量

- 特定变量：脚本运行时的一些相关信息

$#	传递到脚本的参数个数
$*	传递到脚本的参数，与位置变量不同，此选项参数可超过9个
$$	脚本运行时当前进程的ID号，常用作临时变量的后缀，如haison.$$
$!	后台运行的(&)最后一个进程的ID号
$@	与$#相同，使用时加引号，并在引号中返回参数个数
$-	上一个命令的最后一个参数
$?	最后命令的退出状态，0表示没有错误，其他任何值表明有错误

Shell变量

- 系统或者当前用户环境预设的变量
- 包含被操作系统或者程序所用的信息对象

系统环境变量	变量说明
SHELL	当前系统外壳程序
LOGNAME	登录名
PATH	系统搜索路径
HOME	用户工作目录
PS1	外壳程序的提示符
LANG	采用语言及字符集
~	同**HOME**
…	

Shell变量

❖ 变量声明和使用

- 变量的可以用字符(a-z A-z)数字和下划线组成，但必须以字符开头
  - 区分大小写
- Shell变量为弱变量
  - 无需声明类型

✓ 变量赋值
  - 格式：变量=值 （注意等号前后无空格）
  - 例如：a=1

✓ 变量引用
  - $加变量名 即可引用该变量的值
  - 使用 {}可以使变量名更加安全，使得变量和后续字母分开
  - 例如 echo ${a}b ；

Shell变量

- 设置环境变量
  - myvar = “Hello EveryOne”;myUser=David; exports myvar myUser
  - export myvar = “Hello EveryOne”
- env命令
  - 显示所有环境变量

    #env或者#set
  - 搜索某个环境变量是否存在并显示

    #env|grep myvar

课程进度

- Shell综述
- Shell变量
- **Shell语法-算数运算**
- Shell语法-逻辑判断
- Shell语法-控制语句
- Shell语法-函数
- Shell调试

Shell语法-算术运算

- + - * / % 分别对应加、减、乘、除、取余
- 只需将特定的算术表达式用 “$((” 和 “))” 括起来。a=$((4-2)) a的值为2

```
a=10
b=2
echo $(($a+$b))
echo $(($a-$b))
echo $(($a*$b))
echo $(($a/$b))
echo $(($a%$b))
```

课程进度

- Shell综述
- Shell变量
- Shell语法-算数运算
- **Shell语法-逻辑判断**
- Shell语法-控制语句
- Shell语法-函数
- Shell调试

Shell语法-逻辑判断

- && 即“与”，||即“或”

命令执行情况	说明
cmd1 && cmd2	1. 若 cmd1 执行完毕且正确执行($?=0)，则开始执行 cmd2。 2. 若 cmd1 执行完毕且为错误 ($?≠0)，则 cmd2 不执行。
cmd1 || cmd2	1. 若 cmd1 执行完毕且正确执行($?=0)，则 cmd2 不执行。 2. 若 cmd1 执行完毕且为错误 ($?≠0)，则开始执行 cmd2。

- 命令test 和判断符[ ]
  - 测试为真，则返回0，假返回1
  - 可以通过$?得到返回值

Shell语法-逻辑判断

- **检测文件属性**
  - -f 检测文件是否存在且是普通文件
  - -d 检测目录是否存在
  - -e 检测文件是否存在，可以是任何类型文件
  - -r 文件对于该用户是否具备可读属性
  - -w 是否可写
  - -x 是否可执行
  - 例如：
    - 判断文件~/.bashrc是否存在
      test -f ~/.bashrc 或者 [-f ~/.bashrc]
    - 判断文件~/.bashrc是否可读
      test -r ~/.bashrc 或者 [-r ~/.bashrc]

Shell语法-逻辑判断

- **算术运算：**

-eq	等于
-ne	不等于
-lt	小于
-gt	大于
-le	小于或等于
-ge	大于或等于

- **字符串运算符：**

==	字符串相同则为真
!=	字符串不相同则为真
-z $str	若变量str的长度等于0则为真
-n $str	若变量str的长度大于0则为真，-n可去掉

课程进度

- Shell综述
- Shell变量
- Shell语法-算数运算
- Shell语法-逻辑判断
- **Shell语法-控制语句**
- Shell语法-函数
- Shell调试

Shell语法-控制语句

- 选择结构
  - if语句: if then else 提供条件测试
  - case 语句：允许匹配模式、单词或值
- 循环结构
  - for：每次依次处理列表内的信息，直至循环耗尽。
  - until:不常用。条件在循环末尾，至少执行一次。
  - while:条件在循环头部。

shell语法-控制语句

- if语句：用于条件控制结构中
  - 语法为：

    if　测试条件

    then　命令1

    else　命令2

    fi
  - 关键字为if、then、else和fi
  - 例子：

```
if test -f "$1"
then echo "$1 is an ordinary file."
else echo "$1 is not an ordinary file."
fi
```

Shell语法-控制语句

- case语句允许进行多重条件选择
  - 语法为：

```
case     字符串   in
模式字符串1）   命令
                …
                命令;;
模式字符串2）   命令
                …
                命令;;
   …
模式字符串n）   命令
                …
                命令;;
esac
```

  - 关键字为：case in esac

Shell语法-控制语句

for循环控制语句

- 一种是值表方式，另一种是算术表达式方式
- 值表方式,其一般语法是：

  for 变量 [in 值表]; do 命令表; done
- 算术表达式方式,其一般语法是：

  for ((e1;e2;e3)) ; do 命令表; done
- 例如

```
n=3;s=0
for((i=0;i<=$n;i++))
do
let  s=$s+$i
done
echo $s
```

Shell语法-控制语句

- While循环控制语句，
  - 语法是：

```
while  测试条件
do
命令表
done
```

  - 关键字为： while do done
  - 例如

```
while read line
do
    echo $line
done < filename
```

课程进度

- Shell综述
- Shell变量
- Shell语法-算数运算
- Shell语法-逻辑判断
- Shell语法-控制语句
- **Shell语法-函数**
- Shell调试

Shell语法-函数

- 在shell脚本中可以定义并使用函数
- 其定义格式为:

```
[function]函数名( )
{
    命令表
}
```

- 函数应先定义，后使用
- 调用函数时，直接利用函数名，如showfile，不必带圆括号
- shell脚本与函数间的参数传递可利用位置参数和变量直接传递
- 通常，函数中的最后一个命令执行之后，就退出被调函数。也可利用return命令立即退出函数，其语法格式是:

```
return  [ n ]
```

课程进度

- Shell综述
- Shell变量
- Shell语法-算数运算
- Shell语法-逻辑判断
- Shell语法-控制语句
- Shell语法-函数
- **Shell调试**

Shell调试

- 基本的错误类型有两种：语法错误和逻辑错误。
- 语法错误是编写程序时违反了所用编程语言的规则而造成的。
- 逻辑错误通常是由于程序的逻辑关系存在问题。对此类问题需要进行程序调试。
- 基本调试方法
  - set命令打开-x选项，
  - 启动shell时使用-x选项将shell设置成跟踪模式。
  - 经常使用echo或print 输出关键信息