<http://wenwen.soso.com/z/q175449738.htm>

#### linux中的shell变量的种类及各种类的用处是什么？

[ 标签：[linuxshell](http://wenwen.soso.com/z/Search.e?sp=Slinuxshell&ch=rainbow.wty.tag), [变量](http://wenwen.soso.com/z/Search.e?sp=S%E5%8F%98%E9%87%8F&ch=rainbow.wty.tag), [种类](http://wenwen.soso.com/z/Search.e?sp=S%E7%A7%8D%E7%B1%BB&ch=rainbow.wty.tag) ] [^o^& 良子雨](http://wenwen.soso.com/z/ShowUser.e?sp=af8f95c8eaaf9ed1fcff1f3bb5f53161&ch=rainbow.wty.nickname) 2010-01-25 20:05

尽量偏重于名词解释及用处！

#### 满意答案 好评率：100%

2.1.1 本地变量  
定义:是在用户现在的shell生命期的脚本中使用;  
例:[redhat@server redhat]$name="Red Hat Linux" #把变量设置为Red Hat Linux  
[redhat@server redhat]$echo ${name} or #显示变量内容  
=[redhat@server redhat]$echo $name #等同上面的语意  
Red Hat Linux #显示其设定内容,  
[redhat@server redhat]$set | grep "name" #查看刚才所设定的内容,或不加'grep "name"'查看用户自己所有的本地变量,  
[redhat@server redhat]$readonly name #把name这个变量设置为只读变量,不允许再次设置,这可参考变量定义;-\_-!  
[redhat@server redhat]$name="He SenLin" #重新赋值  
bash:name:readonly variable #[只读文件](http://wenwen.soso.com/z/Search.e?sp=S%E5%8F%AA%E8%AF%BB%E6%96%87%E4%BB%B6&ch=w.search.yjjlink&cid=w.search.yjjlink)系统  
[redhat@server redhat]$readonly #查看系统存在的只读文件  
附:重新登录一下系统,再次用  
[redhat@server redhat]$echo ${name}   
#本地变量就不存在了,  
说明本地变量只能存活在用户shell的[生命周期](http://wenwen.soso.com/z/Search.e?sp=S%E7%94%9F%E5%91%BD%E5%91%A8%E6%9C%9F&ch=w.search.yjjlink&cid=w.search.yjjlink)中,重登录就不在了,

2.1.2 [环境变量](http://wenwen.soso.com/z/Search.e?sp=S%E7%8E%AF%E5%A2%83%E5%8F%98%E9%87%8F&ch=w.search.yjjlink&cid=w.search.yjjlink)  
定义:环境变量用于所有[用户进程](http://wenwen.soso.com/z/Search.e?sp=S%E7%94%A8%E6%88%B7%E8%BF%9B%E7%A8%8B&ch=w.search.yjjlink&cid=w.search.yjjlink)<通称为子进程>,包括编辑器,脚本和应用,登录进程为父进程,shell中执行的用户进程均为子进程,不像本地变量只用于现在的shell.  
存在目录在所有用户的主目录下<即/home>/.bash\_profile或  
如何设置环境变量?第一种方法就是编辑上面的二个文件,还可以用"export"进行设定或更改;这可不像本地变量只需 name="Red Hat Linux"那样简单,但却可以把本地变量更改为环境变量;只须在本地变量结束后,输入:export <变量名>;例,我们把name由本地变量修改为环境变量,我们可以这样做;  
[root@server root]#name="Red Hat Linux" #本地变量  
[root@server root]#export name #变量name由本地升为环境  
当然也可以用下面这种方法;  
[root@server root]#export name="Red Hat Linux" #直接定义name为环境变量;  
定义后,可通过env进行查看,不知道上面的本地环境变量由哪个命令进行查看你记住了没有?,演示一下查看现在定义的环境变量;  
[root@server root]#env | grep "name"  
Red Hat Linux  
同样不加后面grep "name",这个参数也是查看所有的环境变量;更同样的是我们也可以运行readonly这个参数给name以只读属性;  
Tips:环境变量就是全部包含在本地变量中<因为它即可运行中父进程中也可运行于子进程中>,注意一点,本地变量不能运行在全部进程中,只有环境变量可以,再不懂就联想一下[人与自然](http://wenwen.soso.com/z/Search.e?sp=S%E4%BA%BA%E4%B8%8E%E8%87%AA%E7%84%B6&ch=w.search.yjjlink&cid=w.search.yjjlink)吧;-\_-!还是不会?用绝招:  
[root@server root]#you=myfriend #设置本地变量name为myfriend  
[root@server root]#echo ${you} #查看一下是否成功  
myfriend #成功  
[root@server root]#sh #更换shell环境等同重登录,这相当一个  
sh-2.05b#echo $you #用户子进程  
#没显示,说明不存在,真的不存在?  
sh-2.05b#exit #返回bash环境  
[root@server root]#echo $you #再次查看;  
myfriend #又出来了,怎么回事?  
[root@server root]#export you #把变量you设为环境变量;  
[root@server root]#sh #再试一次;  
sh-2.05b#echo $you   
myfriend #ok,you see?  
再不懂,,你就把我杀了吧,,什么,你不知道我住在哪?汗-\_-!   
再补充一点,就是在sh进程中设置环境变量是可以写回bash环境中,因为bash是父进程,只有老的才有小的,小的怎么才能有老的哩?<长大了就有老的喽,哦,也对!..乱语>

2.1.3 变量替换  
定义:用变量的值替换它的名字.  
查看方式:echo <通用本地,环境...>  
可以用echo显示可以单个变量的取值;  
[root@server root]#str="this is example" #设置本地变量  
[root@server root]#echo "can you see `echo $str`" #显示  
cat you see this is example #替换就到这了,不明白明再来  
变量替换续  
除了上面的这个方式以外还有五种哦;别急,一个一个来;-\_-!<先说好,偶不是皮条>  
格式 定义  
${variable name} 显示实际值到variable name 1  
  
${variable name :+value} 如果设置了variable name,就把value显示,   
未设置则为空; 2  
${variable name:-value} 如果设置了variable name,就显示它,未设置就显示  
value<仅显不存本地中>>; 3  
${variable name:?value} 未设置提示用户错误信息value; 4  
${variable name:=value} 如未设置就把value设置并显示<写入本地中>, 5  
变量清除:  
unset <变量name>  
[root@server root]#echo $name #变量 name是否存在?  
Red Hat Linux #存在  
[root@server root]#unset name #清除name变量   
[root@server root]#echo $name #是否成功  
#成功  
注意:如果用readonly限制了只读属性就会失败,还有怎么从环境变量中去掉那个变量呢?还记的用哪个设定环境变量的吗?对.export,只须在它后面加"-n"参数即可

**2.1.4 位置变量**  
定义:位置变量表示$0,$1,$2,....$9  
$0 $1 $2 $3 $4  
脚本名 A B C D  
向脚本中使用位置参数;  
向[系统命令](http://wenwen.soso.com/z/Search.e?sp=S%E7%B3%BB%E7%BB%9F%E5%91%BD%E4%BB%A4&ch=w.search.yjjlink&cid=w.search.yjjlink)传递参数;

例:./test<在程序中进行替换>  
[root@server root]#vi /test #创建并编辑/test文件  
#!/bin/bash  
#test  
echo "这是脚本的第一个名称: $0"  
echo "这是脚本的第二个位置参数: $1"  
echo "这是脚本的第三个位置参数: $2"  
echo "这是脚本的第四个位置参数: $3"  
echo "这是脚本的第五个位置参数: $4"  
[root@server root]#chmod u+x ./test #加执行权限;  
[root@server root]#./test A B C #加参数证验  
echo "这是脚本的第一个名称: ./test   
echo "这是脚本的第二个位置参数: A  
echo "这是脚本的第三个位置参数: B  
echo "这是脚本的第四个位置参数: C  
echo "这是脚本的第五个位置参数:   
**<向系统命令中传递位置参数>**  
[root@server root]#vi test1.sh #新建一个脚本文件  
#!/bin/bash  
#name:test1.sh  
#to call: test1.sh 755 #使用方式  
find ~ -perm $1 -print  
$2 #系统命令  
[root@server root]#chmod u+c test1.sh  
[root@server root]#./test1.sh 755 date #位置一为755,2位date  
/dev/ip  
/dev/iput  
.  
.  
.  
Fri May 13 2:31:54 CST 2006  
  
2.1.5 标准变量:  
bash默认建立一些环境变量就是标准变量,可以通俗的说就是不是你加的而是装好机子带的那些环境变量, 可在/etc/profile中进行定义,  
命令 意义  
HOME 用户主目录  
IFS 默认[分隔符](http://wenwen.soso.com/z/Search.e?sp=S%E5%88%86%E9%9A%94%E7%AC%A6&ch=w.search.yjjlink&cid=w.search.yjjlink)  
MAIL 邮箱地址   
MAILCHECK 邮箱检查间隔  
PWD 用户当前工作目录  
LOGNAME 当前用户登录名

2.1.6 特殊变量  
命令 意义  
$# 传递到脚本的参数列表  
$? 前个命令执行情况,0成功,其它值失败  
$$ 脚本运行的当前进程号  
$! [运行脚本](http://wenwen.soso.com/z/Search.e?sp=S%E8%BF%90%E8%A1%8C%E8%84%9A%E6%9C%AC&ch=w.search.yjjlink&cid=w.search.yjjlink)最后一个命令  
$\* 显示脚本全部参数

2.1.7 影响变量的命令  
declare 设置或显示变量  
-f 只显示函数名  
-r 创建只读变量  
-x 创建转出变量  
-i 创建整数变量  
使用"+"代替"-",可以颠倒选项的意思,-r除外,  
export 用于传递给子shell的变量  
-- 表明选项结束,所有后续参数都是[实参](http://wenwen.soso.com/z/Search.e?sp=S%E5%AE%9E%E5%8F%82&ch=w.search.yjjlink&cid=w.search.yjjlink),  
-f 表明在"名-值"对中的名字是函数名,  
-n 把[全局变量](http://wenwen.soso.com/z/Search.e?sp=S%E5%85%A8%E5%B1%80%E5%8F%98%E9%87%8F&ch=w.search.yjjlink&cid=w.search.yjjlink)转换成局部变量,换句话就是不再传给shell   
-p 显示全局变量列表,  
shift  
用于移动位置变量,调整位置变量,使$3的值赋给$2.$2的值赋予$1.  
  
[root@server root]#vi /test   
#!/bin/bash   
#test  
echo "这是脚本的第一个名称: $0"  
echo "这是脚本的第二个位置参数: $1"  
echo "这是脚本的第三个位置参数: $2"  
echo "这是脚本的第四个位置参数: $3"  
echo "这是脚本的第五个位置参数: $4"  
shift  
echo "这是脚本的第二个位置参数: $1"   
echo "这是脚本的第三个位置参数: $2"  
[root@server root]#chmod u+x ./test  
[root@server root]#./test A B C  
echo "这是脚本的第一个名称: ./test   
echo "这是脚本的第二个位置参数: A  
echo "这是脚本的第三个位置参数: B  
echo "这是脚本的第四个位置参数: C  
echo "这是脚本的第五个位置参数: $4"  
shift #显示的时候并没有,加在此只会明了  
echo "这是脚本的第二个位置参数: B   
echo "这是脚本的第三个位置参数: C  
给shfit加一个数试一下:  
[root@server root]#vi ./test  
#!/bin/bash   
#test  
echo "这是脚本的第一个名称: $0"  
echo "这是脚本的第二个位置参数: $1"  
echo "这是脚本的第三个位置参数: $2"  
echo "这是脚本的第四个位置参数: $3"  
echo "这是脚本的第五个位置参数: $4"  
shift 2  
echo "这是脚本的第二个位置参数: $1"   
echo "这是脚本的第三个位置参数: $2"  
[root@server root]#./test A B C D  
echo "这是脚本的第一个名称: ./test   
echo "这是脚本的第二个位置参数: A  
echo "这是脚本的第三个位置参数: B  
echo "这是脚本的第四个位置参数: C  
echo "这是脚本的第五个位置参数: D  
shift 2 #显示的时候并没有,加在此只会明了  
echo "这是脚本的第二个位置参数: C   
echo "这是脚本的第三个位置参数: D  
  
第二节 引号  
2.1 引号的必要性  
2.2 [双引号](http://wenwen.soso.com/z/Search.e?sp=S%E5%8F%8C%E5%BC%95%E5%8F%B7&ch=w.search.yjjlink&cid=w.search.yjjlink)有什么作用  
2.3 [单引号](http://wenwen.soso.com/z/Search.e?sp=S%E5%8D%95%E5%BC%95%E5%8F%B7&ch=w.search.yjjlink&cid=w.search.yjjlink)的作用  
2.4 反引号的作用  
2.5 [反斜杠](http://wenwen.soso.com/z/Search.e?sp=S%E5%8F%8D%E6%96%9C%E6%9D%A0&ch=w.search.yjjlink&cid=w.search.yjjlink)的作用