Linux系统shell脚本判断变量文件目录+l脚本的字符串截取

先写一些基本语法：   
  
1、字符串判断   
  
str1 = str2　　　　　　当两个串有相同内容、长度时为真   
str1 != str2　　　　　 当串str1和str2不等时为真   
-n str1　　　　　　　 当串的长度大于0时为真(串非空)   
-z str1　　　　　　　 当串的长度为0时为真(空串)   
str1　　　　　　　　   当串str1为非空时为真   
  
2、数字的判断   
  
int1 -eq int2　　　　两数相等为真   
int1 -ne int2　　　　两数不等为真   
int1 -gt int2　　　　int1大于int2为真   
int1 -ge int2　　　　int1大于等于int2为真   
int1 -lt int2　　　　int1小于int2为真   
int1 -le int2　　　　int1小于等于int2为真   
  
3 文件的判断   
  
-r file　　　　　用户可读为真   
-w file　　　　　用户可写为真   
-x file　　　　　用户可执行为真   
-f file　　　　　文件为正规文件为真   
-d file　　　　　文件为目录为真   
-c file　　　　　文件为字符特殊文件为真   
-b file　　　　　文件为块特殊文件为真   
-s file　　　　　文件大小非0时为真   
-t file　　　　　当文件描述符(默认为1)指定的设备为终端时为真

[ -n "$var" ] ：判断$var变量是否有值

3、复杂逻辑判断   
  
-a 　 　　　　　 与   
-o　　　　　　　 或   
!　　　　　　　　非   
  
  
  
下面是一些使用实例：   
  
#!/bin/sh   
myPath="/var/log/httpd/"   
myFile="/var /log/httpd/access.log"   
  
#这里的-x 参数判断$myPath是否存在并且是否具有可执行权限   
if [ ! -x "$myPath"]; then   
mkdir "$myPath"   
fi   
  
#这里的-d 参数判断$myPath是否存在   
if [ ! -d "$myPath"]; then   
mkdir "$myPath"   
fi   
  
#这里的-f参数判断$myFile是否存在   
if [ ! -f "$myFile" ]; then   
touch "$myFile"   
fi   
  
#其他参数还有-n,-n是判断一个变量是否是否有值   
if [ ! -n "$myVar" ]; then   
echo "$myVar is empty"   
exit 0   
fi   
  
#两个变量判断是否相等   
if [ "$var1" == "$var2" ]; then   
echo '$var1 eq $var2'   
else   
echo '$var1 not eq $var2'   
fi   
  
  
  
  
      if list then   
          do something here   
      elif list then   
          do another thing here   
      else   
         do something else here   
      fi     
  
EX1:   
  
#!/bin/sh   
  
SYSTEM=`uname -s`    #获取操作系统类型，我本地是linux   
  
if [ $SYSTEM = "Linux" ] ; then     #如果是linux的话打印linux字符串   
echo "Linux"   
elif [ $SYSTEM = "FreeBSD" ] ; then     
echo "FreeBSD"   
elif [ $SYSTEM = "Solaris" ] ; then   
echo "Solaris"   
else   
echo "What?"   
fi     #ifend

一个查看linux系统是32位还是64位的脚本

#!/bin/bash  
if [ `uname -m | sed -e 's/i.86/32/'` -eq 32 ]  
then  
        echo 32  
else  
        echo 64  
fi  
exit 0

基本上和其他脚本语言一样。没有太大区别。不过值得注意的是。[]里面的条件判断。

----------------------------

**Linux shell脚本的字符串截取**

假设有变量 var=http://www.google.com/test.htm  
  
一 # 号截取，删除左边字符，保留右边字符。[http://blog.sina.com.cn/s/blog\_4a071ed80100lvlu.html](http://blog.sina.com.cn/s/blog_4a071ed80100lvlu.html" \t "_blank)  
echo ${var#\*//}  
其中 var 是变量名，# 号是运算符，\*// 表示从左边开始删除第一个 // 号及左边的所有字符  
即删除 http://  
结果是 ：[www.google.com/test.htm](http://www.google.com/test.htm)  
  
二 ## 号截取，删除左边字符，保留右边字符。  
echo ${var##\*/}  
##\*/ 表示从左边开始删除最后（最右边）一个 / 号及左边的所有字符  
即删除<http://www.google.com/>           
结果是 test.htm  
  
三 %号截取，删除右边字符，保留左边字符  
echo ${var%/\*}  
%/\* 表示从右边开始，删除第一个 / 号及右边的字符  
结果是：[http://www.google.com](http://www.google.com/)  
  
四 %% 号截取，删除右边字符，保留左边字符  
  
echo ${var%%/\*}  
%%/\* 表示从右边开始，删除最后（最左边）一个 / 号及右边的字符  
结果是：http:  
  
五 从左边第几个字符开始，及字符的个数  
echo ${var:0:5}  
其中的 0 表示左边第一个字符开始，5 表示字符的总个数。  
结果是：http:  
  
六 从左边第几个字符开始，一直到结束。  
echo ${var:7}  
其中的 7 表示左边第8个字符开始，一直到结束。  
结果是 ：[www.google.com/test.htm](http://www.google.com/test.htm)  
  
七 从右边第几个字符开始，及字符的个数  
echo ${var:0-7:3}  
其中的 0-7 表示右边算起第七个字符开始，3 表示字符的个数。  
结果是：test  
  
八 从右边第几个字符开始，一直到结束。  
echo ${var:0-7}  
表示从右边第七个字符开始，一直到结束。  
结果是：test.htm  
  
注：（左边的第一个字符是用 0 表示，右边的第一个字符用 0-1 表示）