

03.Shell条件测试

- 03.Shell条件测试
 - 1.文件测试
 - 2.数值比较
 - 3.字符串比较

徐亮伟, 江湖人称标杆徐。多年互联网运维工作经验, 曾负责过大规模集群架构自动化运维管理工作。擅长Web集群架构与自动化运维, 曾负责国内某大型电商运维工作。

个人博客"[徐亮伟架构师之路](#)"累计受益数万人。

笔者Q:552408925、572891887

架构师群:471443208

Shell 条件测试

格式1: test 条件表达式

格式2: [条件表达式]

格式3: [[条件表达式]]

1.文件测试

[-e dir|file]

[-d dir] 是否存在,而且是目录

[-f file] 是否存在, 而且是文件

[-r file] 读权限

[-x file] 执行权限

[-w file] 写权限

[-L file]

1.常见使用方式

```
[ ! -d /bbb ] && mkdir /test  
[ -d /bbb ] || mkdir /test
```

2.脚本使用方式

```
#!/usr/bin/bash
```

```
#定义备份目录站点
back_dir=/var/mysql_back

# test -d $back_dir || mkdir -p $back_dir
if [ ! -d $back_dir ];then
    mkdir -p $back_dir
fi
echo "开始备份..."
```

2.数值比较

数值比较 [整数 1 操作符 整数 2]

[1 -gt 10] 大于

[1 -lt 10] 小于

[1 -eq 10] 等于

[1 -ne 10] 不等于

[1 -ge 10] 大于等于

[1 -le 10] 小于等于

1.条件测试, 脚本使用案例, 创建用户

```
#!/usr/bin/bash
read -p "Please input a username: " user

#if id $user &>/dev/null; then

id $user &>/dev/null
if [ $? -eq 0 ]; then
    echo "user $user already exists"
else
    useradd $user
    if [ $? -eq 0 ];then
        echo "$user is created."
    fi
fi
```

2.查看磁盘/当前使用状态, 如果使用率超过80%则报警发邮件

```
//怎么看磁盘使用率, 怎么获取对应的值
df -h|grep "/"$|awk '{print $5}'|awk -F '%' '{print $1}'

//怎么进行数值比对
```

```

if [ $Disk_Free -ge 30 ];then
    echo "Disk Is Use:${Disk_Free}%" > /tmp/disk_use.txt
fi

//脚本
[root@bgx]# cat disk_free.sh
#!/usr/bin/bash

Disk_Free=$(df -h|grep "/"$|awk '{print $5}'|awk -F '%' '{print $1}')

if [ $Disk_Free -ge 30 ];then
    echo "Disk Is Use:${Disk_Free}%" > /tmp/disk_use.txt
fi

```

3.查看内存/当前使用状态，如果使用率超过80%则报警发邮件

```

[root@bgx]# cat mem.sh
Mem_Total=$(free -m|grep "^M"|awk '{print $2}')
Mem_Use=$(free -m|grep "^M"|awk '{print $3}')
Mem_B=$((($Mem_Use*100)/$Mem_Total))

if [ $Mem_B -ge 30 ];then
    echo -e "\033[31m Memory Is Err ${Mem_B}% \033[0m"
else
    echo -e "\033[32m Memory Is OK ${Mem_B}% \033[0m"
fi

```

3.字符串比较

1.字符串比对

```

[root@bgx]# [ "$USER " = "root" ];echo $?
0
[root@bgx]# [ "$USER " == "root" ];echo $?
0
[root@bgx]# BBB=""
[root@bgx]# echo ${#BBB}
0
[root@bgx]# [ -z "$BBB" ] 字符长度为 0
[root@bgx]# echo $?
0
[root@bgx]# [ -n "$BBB" ] 字符长度不为 0
[root@bgx]# echo $?
0

```

1

//小结: 变量为空 或未定义: 长度都为0

2.字符串比对

```
[root@bgx]# [ "$USER" = "root" ];echo $?  
0  
[root@bgx]# [ "$USER" = "bgx" ];echo $?  
1  
[root@bgx]# [ "$USER" != "alice" ];echo $?  
0
```

3.多整数比对条件

```
[root@bgx]# [ 1 -lt 2 -a 5 -gt 10 ];echo $?  
1  
[root@bgx]# [ 1 -lt 2 -o 5 -gt 10 ];echo $?  
0  
[root@bgx]# [[ 1 -lt 2 && 5 -gt 10 ]];echo $?  
1  
[root@bgx]# [[ 1 -lt 2 || 5 -gt 10 ]];echo $?  
0
```

4.正则比对

```
[root@bgx]# [ "$USER" = "root" ];echo $?  
0  
[root@bgx]# [ "$USER" =~ ^r ];echo $?  
bash: [: =~: binary operator expected  
2  
[root@bgx]# [[ "$USER" =~ ^r ]];echo $? //使用正则  
0  
  
//判断变量是不是数字  
[root@bgx]# num=123  
[root@bgx]# [[ "$num10" =~ ^[0-9]+$ ]];echo $?  
0  
[root@bgx]# [[ "$num20" =~ ^[0-9]+$ ]];echo $?  
1
```

5.判断用户输入的是否是数字

```
[root@bgx]# cat test02.sh
#!/bin/bash

read -p "请输入一个数值: " num
if [[ ! "$num" =~ ^[0-9]+$ ]];then
    echo " 你输入的不是数字, 程序退出!!!"
    exit
fi
echo "Number Is $num"
```

6.批量创建用户脚本, 仅个人使用

```
#!/usr/bin/bash
read -p "Please input number: " num
read -p "Please input prefix: " prefix

for i in `seq $num`
do
    user=$prefix$i
    useradd $user
    echo "123" |passwd --stdin $user &>/dev/null
    if [ $? -eq 0 ];then
        echo "$user is created."
    fi
done
```

7.批量创建用户脚本, 改进版

```
#!/usr/bin/bash
read -p "Please input number: " num
if [[ ! "$num" =~ ^[0-9]+$ ]];then
    echo "error number!" && exit 1
fi

read -p "Please input prefix: " prefix
if [ -z "$prefix" ];then
    echo "error prefix"
    exit
fi

for i in `seq $num`
do
    user=$prefix$i
    useradd $user
    echo "123" |passwd --stdin $user &>/dev/null
```

```
if [ $? -eq 0 ];then
    echo "$user is created."
fi
done
```

8.批量创建用户脚本, 完善版