

Linux操作系统

9 shell流程控制

主讲：杨东平
中国矿大计算机学院

流程控制语句

- 单分支 if 语句
- 双分支 if 语句
- 多分支 if 语句
- case 语句
- for 循环
- while 循环和 until 循环

网络安全与网络工程系杨东平 jsxhbc@163.com

Linux操作系统

2018年9月26日7时52分

2

单分支 if 语句

➢ 语法规则1:

```
if [ 条件判断式 ]
then
    条件成立时执行的命令
...
fi
```

➢ 语法规则2(写成一行, 适用于终端命令提示符):

```
if [ 条件表达式 ]; then 条件成立时执行的命令; fi
```

➢ 几个注意点:

- ❖ if 语句使用 fi 结尾(其它流程控制语句存在类似情况)
- ❖ [条件判断式] 就是使用 test 命令判断, 所以中括号和条件判断式之间必须有空格
- ❖ 写在一行上时的各部分之间需要用 “;” 分隔

网络安全与网络工程系杨东平 jsxhbc@163.com

Linux操作系统

2018年9月26日7时52分

3

单分支 if 语句(续)

➢ 例1(视频: 18 流控: JudgeRoot.sh): 判断登录的用户是否 root

```
#!/bin/bash
# JudgeRoot.sh
```

```
test=$(env | grep "USER" | cut -d "=" -f2)
```

```
if [ "$test" == root ]
then
    echo "Current user is root."
fi
```

```
root@localhost ~1# env
HOSTNAME=localhost.localdomain
TERM=linux
SHELL=/bin/bash
HISTSIZE=1000
USER=root
LS_COLORS=rs=0:di=01;34:ln=01;3
```

```
root@localhost ~1# chmod 755 JudgeRoot.sh
root@localhost ~1# ./JudgeRoot.sh
Current user is root.
```

网络安全与网络工程系杨东平 jsxhbc@163.com

Linux操作系统

2018年9月26日7时52分

4

单分支 if 语句(续)

➢ 例2(视频: 19 流控: JudgeUsage.sh): 判断分区使用率

```
#!/bin/bash
# JudgeUsage.sh
```

```
rate=$(df -h | grep "/dev/sda1" | awk '{print $5}' | cut -d "%" -f1)
```

```
if [ $rate -ge 7 ]
then
    echo "Warning! /dev/sda1 is full!"
fi
```

```
root@localhost ~1# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup-lv_root 10G  755M  8.9G   8% /
tmpfs            499M   0    499M   0% /dev/shm
/dev/sda1        499M  32M  428M  7% /boot
```

```
root@localhost ~1# chmod 755 JudgeUsage.sh
root@localhost ~1# ./JudgeUsage.sh
Warning! /dev/sda1 is full!
```

网络安全与网络工程系杨东平 jsxhbc@163.com

Linux操作系统

2018年9月26日7时52分

5

双分支 if 条件语句: if else 语句

➢ 语法:

```
if [ 条件判断式 ]
then
    条件成立时执行的程序段
else
    条件不成立时执行的另一个程序段
fi
```

网络安全与网络工程系杨东平 jsxhbc@163.com

Linux操作系统

2018年9月26日7时52分

6

双分支 if 条件语句: if else 语句(续)

➤ 例(视频: 20 流控: testIfElse.sh):

```
#!/bin/bash
# testIfElse.sh

num1=$((2*3))
num2=$((1+5))
if test $[num1] -eq $[num2] # if [ $(num1) -eq $(num2) ]

then
    echo "equals"
else
    echo "not equals"
fi

[root@localhost ~]# chmod 755 testIfElse.sh
[root@localhost ~]# ./testIfElse.sh
equals
```

网络安全与网络工程系靳东平 jxshbc@163.com Linux操作系统 2018年9月26日7时52分 7

多分支 if 条件语句

➤ 语法:

```
if [ 条件判断式1 ]
then
    当条件判断式1成立时执行的程序段1
elif [ 条件判断式2 ]
then
    当条件判断式2 成立时执行的程序段2
...省略更多条件...
else
    当所有条件都不成立时最后执行的程序段
fi
```

网络安全与网络工程系靳东平 jxshbc@163.com Linux操作系统 2018年9月26日7时52分 8

多分支 if 条件语句(续)

➤ 例(视频: 21 流控: JudgeFileType.sh): 判断用户输入的是不是文件

```
#!/bin/bash
# JudgeFileType.sh

read -p "Please input a filename:" file
#接收键盘的输入并赋予变量file

if [ -z "$file" ]
#判断file变量是否为空
then
    echo "Error,Please input a filename"
    exit 1
elif [ !-e "$file" ]
#判断file的值是否存在
then
    echo "Your input is not a file!"

    exit 2
elif [ -f "$file" ]
#判断file是否为普通文件
then
    echo "$file is a regular file!"
elif [ -d "$file" ]
#判断file是否为目录文件
then
    echo "$file is a directory!"
else
    echo "$file is an other file!"
fi

[root@localhost ~]# chmod 755 JudgeFileType.sh
[root@localhost ~]# ./JudgeFileType.sh
Please input a filename:JudgeFileType.sh
JudgeFileType.sh is a regular file!
[root@localhost ~]# ./JudgeFileType.sh
Please input a filename:root
Your input is not a file!
```

网络安全与网络工程系靳东平 jxshbc@163.com Linux操作系统 2018年9月26日7时52分 9

多分支 case 条件语句

➤ case 语句匹配一个值或一个模式, 如果匹配成功, 执行相匹配的命令

➤ 语法:

```
case 值 in
模式1)
    变量值匹配模式1时执行的命令序列
;;
模式2)
    变量值匹配模式2时执行的命令序列
;;
.....
*)
    变量值与以上模式都不匹配时默认执行的命令序列
;;
esac
```

网络安全与网络工程系靳东平 jxshbc@163.com Linux操作系统 2018年9月26日7时52分 10

多分支 case 条件语句(续)

➤ 语法说明:

- ❖ case 行尾必须为单词 in, 每个模式必须以右括号")" 结束
- ❖ 双分号 ;; 表示命令序列结束, 类似与其他语言中的 break
- ❖ 匹配模式中可使用方括号表示一个连续的范围, 如 [0-9]; 使用竖杠符号 "|" 表示或
- ❖ 最后的 "*" 表示默认模式, 当使用前面的各种模式均无法匹配该变量时, 将执行 "*" 后的命令序列
- ❖ case 后的值可以为变量或常数

网络安全与网络工程系靳东平 jxshbc@163.com Linux操作系统 2018年9月26日7时52分 11

多分支 case 条件语句(续)

➤ 例(视频: 22 流控: testCase1.sh): 提示输入 1

到 4, 与每一种模式进行匹配

```
#!/bin/bash
# testCase1.sh
```

```
echo 'Input a number:'
read Num
case $Num in
1) echo 'You select 1'
;;
2) echo 'You select 2'
;;
3) echo 'You select 3'
;;
4) echo 'You select 4 or 5'
;;
*) echo 'default'
;;
esac
```

```
[root@localhost ~]# chmod 755 testCase1.sh
[root@localhost ~]# ./testCase1.sh
Input a number:
4
You select 4
[root@localhost ~]# ./testCase1.sh
Input a number:
4
You select 4 or 5
[root@localhost ~]# ./testCase1.sh0
bash: ./testCase1.sh0: No such file or directory
[root@localhost ~]# ./testCase1.sh
Input a number:
0
default
```

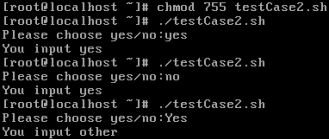
网络安全与网络工程系靳东平 jxshbc@163.com Linux操作系统 2018年9月26日7时52分 12

多分支 case 条件语句(续)

➤例(视频: 23 流控: testCase2.sh): 判断用户输入

```
#!/bin/bash
# testCase2.sh

read -p "Please choose yes/no:" -t 30 choose
case $choose in
    "yes")
        echo "You input yes"
        ;;
    "no") echo "you input no"
        ;;
    *) echo "you input other"
        ;;
esac
```



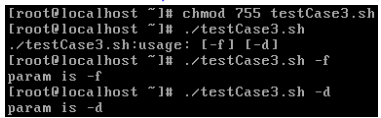
网络安全与网络工程系靳东平 jsxhbc@163.com Linux操作系统 2018年9月26日7时52分 13

多分支 case 条件语句(续)

➤例(视频: 24 流控: testCase3.sh): 命令行输入

```
#!/bin/bash
# testCase3.sh

option=$1
case ${option} in
    -f) echo "param is -f"
        ;;
    -d) echo "param is -d"
        ;;
    *)
        echo "$0:usage: [-f] | [-d]"
        exit 1 #退出码
        ;;
esac
```



网络安全与网络工程系靳东平 jsxhbc@163.com Linux操作系统 2018年9月26日7时52分 14

for 循环

- 语法1:
for 变量 in 值1 值2 值3...
do
 程序块
done
- 语法2:
for 变量 `命令`
do
 程序块
done
- 语法3:
for ((初始值; 循环控制条件; 变量变化))
do
 程序块
done

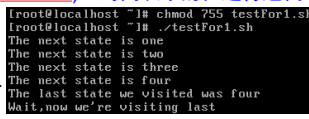
网络安全与网络工程系靳东平 jsxhbc@163.com Linux操作系统 2018年9月26日7时52分 15

for 循环(续)

➤例(视频: 25 流控: testFor1.sh): 对列表中的值进行迭代

```
#!/bin/bash
# testFor1.sh

for t in one two three four
do
    echo "The next state is $t"
done
echo "The last state we visited was $t"
t=last
echo "Wait,now we're visiting $t"
```



若遇到 / . 等特殊符号时, 可以使用反斜杠 \ 来转义特殊字符, 或者用双引号括起来

网络安全与网络工程系靳东平 jsxhbc@163.com Linux操作系统 2018年9月26日7时52分 16

for 循环(续)

➤例(视频: 26 流控: testFor2.sh): 若遇到 / . 等特殊符号时, 可以使用反斜杠 \ 来转义特殊字符, 或者用双引号括起来

```
#!/bin/bash
# testFor2.sh

for t in "I'm" a student
do
    echo "$t"
done
```



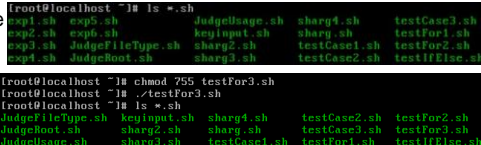
网络安全与网络工程系靳东平 jsxhbc@163.com Linux操作系统 2018年9月26日7时52分 17

for 循环(续)

➤例(视频: 27 流控: testFor3.sh): 批量删除文件

```
#!/bin/bash
# testFor3.sh

for file in `ls exp?.sh`
do
    rm -f $file
done
```



网络安全与网络工程系靳东平 jsxhbc@163.com Linux操作系统 2018年9月26日7时52分 18

for 循环(续)

➤例(视频: 28 流控: testFor4.sh)求1~100的和

```
#!/bin/bash
# testFor4.sh

sum=0
for (( i=1; i<=100; i++ ))
do
    sum=$(( $sum + $i ))
done
echo "1+2+...100=$sum"
```

```
[root@localhost ~]# chmod 755 testFor4.sh
[root@localhost ~]# ./testFor4.sh
1+2+...+100=5050
```

网络安全与网络工程系蔡东平 jsxhbc@163.com Linux操作系统 2018年9月26日7时52分 19

while 循环

➤语法:

```
while [ 条件判断式 ]
do
    条件判断式成立时执行的命令
done
```

网络安全与网络工程系蔡东平 jsxhbc@163.com Linux操作系统 2018年9月26日7时52分 22

while 循环(续)

➤例(视频: 29 流控: testWhile.sh): 计算1+2+...100之和

```
#!/bin/bash
# testWhile.sh

i=1
s=0
while [ $i -le 100 ]
do
    s=$(( $s + $i ))
    i=$(( $i + 1 ))
done
echo "1+2+...100=$s"
```

```
[root@localhost ~]# chmod 755 testWhile.sh
[root@localhost ~]# ./testWhile.sh
1+2+...100=5050
```

网络安全与网络工程系蔡东平 jsxhbc@163.com Linux操作系统 2018年9月26日7时52分 23

until 循环

➤语法:

```
until [ 条件判断式 ]
do
    条件判断式不成立时执行的命令
done
```

➤注意: 循环体执行到条件判断式成立时为止

网络安全与网络工程系蔡东平 jsxhbc@163.com Linux操作系统 2018年9月26日7时52分 24

until 循环(续)

➤例(视频: 30 流控: testUntil.sh)从键盘输入数据, 直到输入 "exit" 为止

```
#!/bin/bash
# testUntil.sh

read -p "Please input data until 'exit': " EXIT
until [ $EXIT = "exit" ]
do
    read -p "Please input data until 'exit': " EXIT
done
echo "Program stopped!"
```

```
[root@localhost ~]# chmod 755 testUntil.sh
[root@localhost ~]# ./testUntil.sh
Please input data until 'exit':aswqgr
Please input data until 'exit':sfas
Please input data until 'exit':q
Please input data until 'exit':c
Please input data until 'exit':exit
Program stopped!
```

网络安全与网络工程系蔡东平 jsxhbc@163.com Linux操作系统 2018年9月26日7时52分 25