7 case条件语句

`case`语句

`case`语句**相当于多分支的`if/elif/else`语句**。

`if`语句看起来略微复杂，`case`语句更加**简洁工整**。

`case`语句常应用在实现**系统服务启动脚本**等应用场景中。

一、`case`语句语法

1.1 `case`语句基本格式

`case`语句有固定的语法格式。

Shell | 复制代码

```shell
case 变量 in
  条件表达式1)
    代码块1
    ;;
  条件表达式2)
    代码块2
    ;;
  条件表达式3)
    代码块3
    ;;
  *)
    无匹配后代码块
esac
```

在`case`语句中，程序首先会获取`case`语句中**变量的值**，然后**依次检测**结构中的**条件表达式与变量值是否匹配**，如果**匹配**则执行后面的**代码块**，执行到**双分号(`;;`)停止**，在**无一匹配**的情况下匹配**最后的默认`*`**，并执行后面的默认命令(**此处的双分号可以省略**)。 **只要满足一个条件表达式**就会**跳出`case`语句主体**，执行`esac`关键字后面的命令。

⚠️`case`语句中的**条件表达式**通常为**常量或正则表达式。**

案例：判断星期日

读取一个数字，判断星期日。1–7对应星期一至星期天，1–7之外的数字提示错误。

Shell | 复制代码

```
[root@Shell ~]# vi weekday.sh
#! /bin/bash
read -p "请输入一个整数: " num
case $num in
    1)
        echo "Monday"
        ;;
    2)
        echo "Tuesday"
        ;;
    3)
        echo "Wednesday"
        ;;
    4)
        echo "Thursday"
        ;;
    5)
        echo "Friday"
        ;;
    6)
        echo "Saturday"
        ;;
    7)
        echo "Sunday"
        ;;
    *)
        echo "error"
esac
[root@Shell ~]# . weekday.sh
请输入一个整数: 2
Tuesday
[root@Shell ~]# . weekday.sh
请输入一个整数: 8
error
```

上述案例等价于下面使用 `if/elif/else` 条件语句编写的代码。

```shell
#!/bin/bash
read -p "请输入一个整数: " num
if ((num==1)); then
    echo "Monday"
elif ((num==2)); then
    echo "Tuesday"
elif ((num==3)); then
    echo "Wednesday"
elif ((num==4)); then
    echo "Thursday"
elif ((num==5)); then
    echo "Friday"
elif ((num==6)); then
    echo "Saturday"
elif ((num==7)); then
    echo "Sunday"
else
    echo "error"
fi
```

1.2 `case` 语句的正则表达式支持

`case` 语句中的条件表达式支持简单的**正则表达式**，常用的条件表达式如下。

条件表达式	说明
`*`	任意个字符
`?`	任意单个字符
`[abc]`	a、b或c其中之一
`[a-n]`	a-n之间的任一字符
`|`	多重选择

案例：识别用户的输入信息

判断输入信息为 `yes` 或 `no`，其他情况提示“无效输入”。

可以输入简写 `y` 或全写 `yes`，可以输入简写 `n` 或全写 `no`，**不区分大小写**。

Shell | 复制代码

```
[root@Shell ~]# vi validate_input.sh
#! /bin/bash
# 获取输入内容
read -p "您确定需要执行该操作吗(y|n)?" key
case $key in
    # 输入为y或yes时（忽略大小写）的判断条件
    [Yy] | [Yy][Ee][Ss])
        echo "注意:您选择的是yes."
        ;;
    # 输入为n或no时（忽略大小写）的判断条件
    [Nn] | [Nn][Oo])
        echo "您选择的是no."
        ;;
    *)
        echo "无效的输入"
        ;;
esac
[root@shell ~]# . validate_input.sh
您确定需要执行该操作吗(y|n)?Y
注意:您选择的是yes.
[root@Shell ~]# . validate_input.sh
您确定需要执行该操作吗(y|n)?y
注意:您选择的是yes.
[root@Shell ~]# . validate_input.sh
您确定需要执行该操作吗(y|n)?yes
注意:您选择的是yes.
[root@shell ~]# . validate_input.sh
您确定需要执行该操作吗(y|n)?YeS
注意:您选择的是yes.
```

⚠️ `[Yy]` 表示 `Y` 或 `y` 均匹配。

使用 `|` 分隔多个模式匹配，表示**或者关系**，匹配任意模式即可成功。

案例：判断字符类型

输入一个字符，判断字符的类型。

```shell
[root@Shell ~]# vi detect_char.sh
#!/bin/bash
read -n 1 -p "请输入一个字符: " char
echo
case $char in
    [a-zA-Z])
        echo "字母"
        ;;
    [0-9])
        echo "数字"
        ;;
    [,.?!])
        echo "符号"
        ;;
    *)
        echo "error"
esac
[root@Shell ~]# . detect_char.sh
请输入一个字符: 1
数字
[root@Shell ~]# . detect_char.sh
请输入一个字符: a
字母
[root@Shell ~]# . detect_char.sh
请输入一个字符: ,
符号
[root@Shell ~]# . detect_char.sh
请输入一个字符: -
error
```

二、`case`语句实例

2.1 `case`删除用户判断

`case`语句结合`read`命令（读入用户输入的内容），与对应的变量名建立关联，如果用户输入正确的内容，返回一个结果；如果输入其他内容，返回另外一个结果。

案例：`if`语句实现删除用户

使用`if`语句实现提示用户输入信息并赋值给`user`变量，如果返回值不等于0，则显示没有这个用户，否则用户存在。然后，根据脚本提示信息删除用户操作。

Shell 复制代码

```shell
#!/bin/bash
read -p "Please input a username: " user
# 打印信息提示用户输入，输入信息赋值给 user 变量
if id -u $user >/dev/null 2>&1 ; then
    # 读取操作指令
    read -p "Are you sure?[y/n]: " action
    # 判断指令是否为y、Y、yes、YES
    if [ "$action" = "y" -o "$action" = "Y" -o "$action" = "YES" -o "$action" = "yes" ]
        then
        userdel -r $user
        echo "$user is deleted"
    fi
else
    echo "no such user: $user"
fi
```

案例：`case` 语句实现删除用户

```shell
[root@Shell ~]# vi delete_user1.sh
#!/bin/bash
read -p "Please input a username: " user
# 打印信息提示用户输入，输入信息赋值给 user 变量
if id -u $user >/dev/null 2>&1 ; then
    # 读取操作指令
    read -p "Are you sure?[y/n]: " action
    # 判断指令是否为y、Y、yes、YES
    case "$action" in
        y|Y|yes|YES)
            userdel -r $user
            echo "$user is deleted!"
            ;;
        *)
            echo "error"
    esac
else
    echo "no such user: $user"
fi
[root@Shell ~]# . delete_user1.sh
Please input a username: test1
Are you sure?[y/n]: y
test1 is deleted!
[root@Shell ~]# . delete_user1.sh
Please input a username: test1
no such user: test1
```

2.2 `case` 实现系统工具箱

系统工具箱指一些列系统工具软件的集合，如查看内存大小、磁盘负载、CPU大小。

案例：系统工具箱

采用 `cat` 命令打印菜单，如果用户输入 `h` ，则打印出菜单；如果用户输入 `f` ，则执行磁盘分区命令；如果用户输入 `d` ，则执行磁盘空间使用情况；如果用户输入 `m` ，则执行内存使用情况； 如果用户输入 `u` ，则执行 `uptime` 命令，这个命令主要用于获取主机运行时间和查询Linux系统 负载等信息；如果用户输入 `q` ，则跳出整个循环；如果用户输入为空则不显示内容，否则显示错误。

Shell | 复制代码

```
[root@Shell ~]# vi system_tools.sh
#!/bin/bash
menu() {
cat <<-EOF
    ############################################
    #  h.      help    #
    #  f.      disk partition  #
    #  d.      filesystem mount        #
    #  m.      memory  #
    #  u.      system load     #
    #  q.      exit    #
    ############################################
EOF
}
menu
while true
do
    read -p "Please input[h for help]: " action
    clear
    case "$action" in
    h)
        menu
        ;;
    f)
        fdisk -l
        ;;
    d)
        df -Th
        ;;
    m)
        free -m
        ;;
    u)
        uptime
        ;;
    q)
        break
        ;;
    "")
        ;;
    *)
        echo "error"
    esac
done
echo "finish……"
```

```
[root@Shell ~]# . system_tools.sh
     ###########################################
     #  h.      help    #
     #  f.      disk partition  #
     #  d.      filesystem mount        #
     #  m.      memory  #
     #  u.      system load     #
     #  q.      exit    #
     ###########################################
Please input[h for help]: h
     ###########################################
     #  h.      help    #
     #  f.      disk partition  #
     #  d.      filesystem mount        #
     #  m.      memory  #
     #  u.      system load     #
     #  q.      exit    #
     ###########################################
Please input[h for help]: d
Filesystem              Type      Size  Used Avail Use% Mounted on
/dev/mapper/centos-root xfs        18G  858M   17G   5% /
devtmpfs                devtmpfs  479M     0  479M   0% /dev
tmpfs                   tmpfs     489M     0  489M   0% /dev/shm
tmpfs                   tmpfs     489M  6.7M  483M   2% /run
tmpfs                   tmpfs     489M     0  489M   0% /sys/fs/cgroup
/dev/sda1               xfs       497M  125M  373M  25% /boot
tmpfs                   tmpfs      98M     0   98M   0% /run/user/0
Please input[h for help]: m
              total        used        free      shared  buff/cache   avai
lable
Mem:            977         111         732           6         134
  724
Swap:          2047           0        2047
Please input[h for help]: q
finish……
```

小结

- `case` 语句格式
- `case` 语句支持的正则表达式

课程目标

- 知识目标：熟练掌握 `case` 语句的基本语法。
- 技能目标：能够根据实际需求利用 `case` 语句实现条件流程控制。

课外拓展

- 进一步了解 `case` 语句的应用场景。

参考资料

- `help case` 或者 `man bash` 后输入 `/case word`
- 《Linux Shell核心编程指南》，丁明一，电子工业出版社