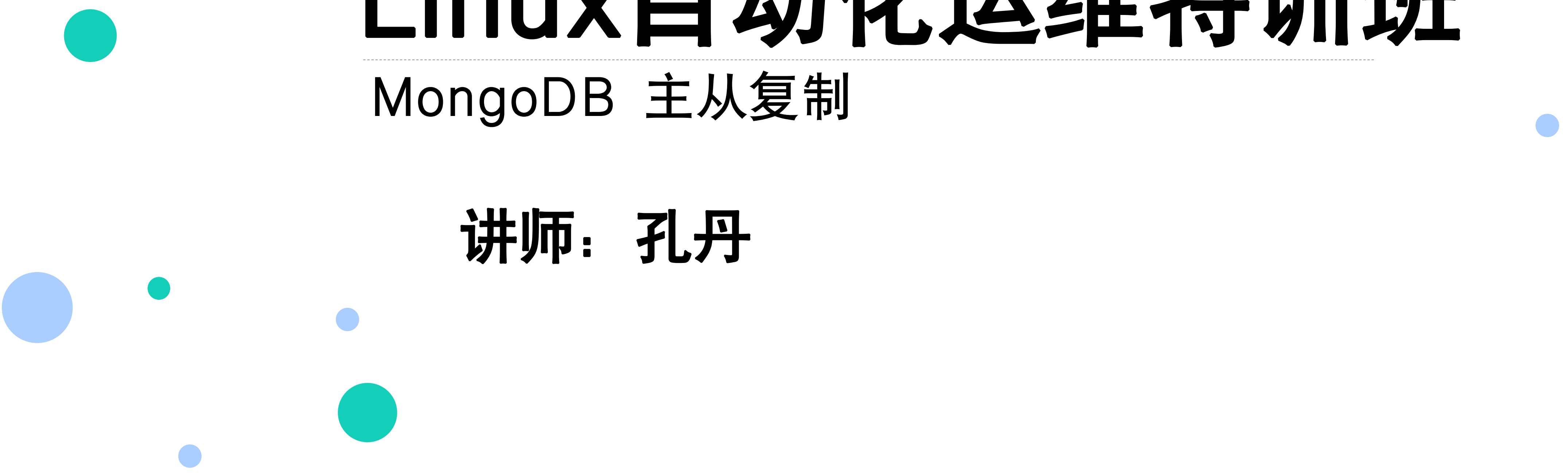




Linux自动化运维特训班

流程控制之条件判断

讲师：孔丹



大纲

- 条件判断语句：介绍基本的if、if else以及if elif语句的使用方法。
- 多条件判断语句case：主要介绍case语句的基本语法以及使用case语句来解决一些实际问题。

条件判断语句

- 条件判断语句是一种最简单的流程控制语句。该语句使得程序根据不同的条件来执行不同的程序分支。本节将介绍Shell程序设计中的简单的条件判断语句。

if条件语句的语法

□ 1、单分支结构

第一种语法：

```
if <条件表达式>  
then  
    指令  
fi
```

第二种语法：

```
if <条件表达式>;then  
    指令  
fi
```

if条件语句的语法

□ 2、双分支结构

```
if <条件表达式>  
then  
    指令序列1  
else  
    指令序列2  
fi
```

□ 3、多分支结构

```
if 条件表达式1  
then  
    命令序列1  
elif 条件表达式2  
then  
    命令序列2  
else  
    命令序列n  
fi
```

if条件语句案例

□ 1、单分支结构

示例1：编写脚本，判断当前系统剩余内存大小，如果低于100M，邮件报警管理员，使用计划任务，每10分钟检查一次。

分析：核心是如何获取当前剩余内存大小

```
[root@localhost ~]# free -m | grep "Mem:" | tr -s " " | cut -d" " -f4  
600
```

```
[root@localhost ~]# free -m | awk '/Mem:/ {print $4}'  
600
```

if条件语句案例

□ 示例2：编写脚本，判断当前脚本执行者，如果不是root用户，提示用户脚本需要root用户来执行，并退出。

分析：核心在于判断当前脚本执行者

方法1：whomai

```
[root@localhost ~]# whoami
```

```
root
```

方法2：id -u，结构为零则为root

```
[root@localhost ~]# id -u
```

```
0
```

方法3：使用环境变量

```
[root@localhost ~]# echo $USER
```

```
root
```

```
[student@localhost ~]$ echo $UID
```

```
1000
```


if条件语句案例

□ 2、双分支结构

示例 1：判断 sshd 进程是否运行，如果服务未启动则启动相应服务。

分析：核心在于判断进程是否运行的方法？

方法1：查看进程

```
[root@localhost ~]# ps -elf | grep sshd | grep -v grep | wc -l
4
```

方法2：查看端口

本地查看netstat、ss、lsof

远程查看telnet、nmap、nc

```
[root@localhost ~]# netstat -lnupt | grep 22 | wc -l
2
```

```
[root@localhost ~]# ss -lnupt | grep 22 | wc -l
2
```

```
[root@localhost ~]# nmap -p 22 127.0.0.1 | grep open | wc -l
1
```


if条件语句案例

□ 2、双分支结构

示例 2：检查主机是否存活，并输出结果

分析：最简单方式是通过ping命令检测

```
ping -c 2 -W 1 192.168.150.16 &> /dev/null
```

通过\$?判断，也可以直接 if ping -c 2 -W 1 192.168.150.16
&> /dev/null测试

if条件语句案例

□ 3、多分支结构

示例1: 两个整数比较大小。

示例2: 根据用户输入成绩, 判断优良中差。

85-100 优秀--A 70-84 良好--B

60-69 合格--C 60分以下不合格--D

```
read -p "Please enter your score (0-100): " grade
```

```
if [ $grade -ge 85 ]
```

```
then
```

```
    echo "$grade, A"
```

```
elif [ $grade -ge 70 ]
```

```
then
```

```
    echo "$grade, B"
```

```
elif [ $grade -ge 60 ]
```

```
then
```

```
    echo "$grade, C"
```

```
else
```

```
    echo "$grade, D"
```

```
fi
```

if条件语句案例

□ 3、多分支结构

示例3：根据用户输入，判断是数字、字母或其他字符。

```
read -p "Please enter a character, press enter to  
continue: " str  
if echo "$str"|grep "[a-zA-Z]" >/dev/null  
then  
    echo "Input is letter"  
elif echo "$str"|grep "[0-9]" >/dev/null  
then  
    echo "Input is number"  
else  
    echo "Input is other"  
fi
```

复合指令

□ 复合指令：即一串命令。

()和{}都是对一串的命令进行执行,但有所区别:

相同点:

()和{}都是把一串的命令放在括号里面,如果命令在一行命令之间用;

号隔开

不同点

- ✓ ()只是对一串命令重新开一个子shell进行执行,{}对一串命令在当前shell执行
- ✓ ()最后一个命令可以不用分号,{}最后一个命令要用分号
- ✓ ()里的第一个命令和左边括号不必有空格,{}的第一个命令和左括号之间必须要有一个空格
- ✓ ()和{}中括号里面的某个命令的重定向只影响该命令,但括号外的重定向则影响到括号里的所有命令

复合指令

□ 复合指令示例

```
[root@localhost scripts]# (pwd;cd /tmp;pwd)
/scripts
/tmp
# () 子shell中执行，执行完毕，当前路径不变
[root@localhost tmp]# { pwd;cd /tmp;pwd;}
/tmp
/tmp
```

exit退出程序

- ❑ exit语句的基本作用是终止Shell程序的执行。除此之外，exit语句还可以带一个可选的参数，用来指定程序退出时的状态码。

exit语句的基本语法如下：

- ❑ exit status
- ❑ 其中，status参数表示退出状态，该参数是一个整数值，其取值范围为0~255。与其他的Shell命令的一样，Shell程序的退出状态也储存在系统变量\$?中，因此，用户可以通过该变量取得Shell程序返回给父进程的退出状态码。

多条件判断语句case

□ case语句语法

case 变量名 in

值1)

指令1

;;

值2)

指令2

;;

值3)

指令3

;;

*)

默认

esac

多条件判断语句case

□ case语句示例

示例1：由用户从键盘输入一个字符，并判断该字符是否为字母、数字或者其他字符，并输出相应的提示信息。

```
read -p "Please enter a character, press enter to continue:" KEY
```

```
case "$KEY" in
    [a-z]|[A-Z])
        echo "Input is letter"
        ;;
    [0-9])
        echo "Input is number"
        ;;
    *)
        echo "Input is other characters"
esac
```

多条件判断语句case

□ case语句示例

示例2：将判断分数范围多分支语句用case语句实现

```
read -p "Please enter your score (0-100): " grade
case "$grade" in
    8[5-9]|9[0-9]|100)
        echo "$grade, A"
        ;;
    7[0-9]|8[0-4])
        echo "$grade, B"
        ;;
    6[0-9])
        echo "$grade, C"
        ;;
    *)
        echo "$grade, D"
esac
```

总结

- if条件判断
- 复合指令
- case多分支

作业

- 1、ping主机测试
- 2、判断一个用户是否存在
- 3、判断当前内核主版本是否为3，且次版本是否大于10
- 4、判断vsftpd软件包是否安装，如果没有则自动安装
- 5、判断httpd是否运行
- 6、判断指定的主机是否能ping通，必须使用\$1变量
- 7、报警脚本，要求如下：
 - 根分区剩余空间小于20%
 - 内存已用空间大于80%
 - 向用户alice发送告警邮件
 - 配合crond每5分钟检查一次
 - `[root@localhost ~]# echo "邮件正文" | mail -s "邮件主题" alice`
- 8、判断用户输入的是否是数字



谢谢观看

更多好课，请关注[万门大学APP](#)

