



慕课大数据学习群

群号: 938632081



扫一扫二维码，入群聊。



慕课网大数据体系课程



扫码体验项目【电商+直播+中台】



扫码关注获取更多课程信息



扫码体验课程配套电子书

Linux快速上手使用

学习目标

- ◆ 掌握Linux虚拟机的安装
- ◆ 掌握Linux中高级命令的使用
- ◆ 掌握Linux中的shell编程

Linux的安装配置

安装Linux虚拟机

- ◆ 安装VMware软件
- ◆ 使用VMware安装配置Linux虚拟机
- ◆ 使用SecureCRT连接Linux虚拟机



Linux常见高级命令

Linux高级命令之文件相关

◆ vi: 文件编辑利器

◆ wc、uniq、sort、head: 文件内容统计相关命令

vi命令

◆ 编辑文件内容

◆ 复制粘贴

◆ 查找字符串

◆ 快速删除

◆ 查找某一行内容

◆ 快速跳到文件首行和末行

文件内容统计相关命令

- ◆ wc: 统计字数相关信息
- ◆ uniq: 检查重复的行列
- ◆ sort: 排序
- ◆ head: 取前N条数据

Linux高级命令之日期相关

- ◆ date: 获取当前时间
- ◆ 格式化输出: `date + "%Y-%m-%d %H:%M:%S"`

Linux高级命令之进程相关

- ◆ ps: 显示进程信息
- ◆ top: 动态监控进程信息
- ◆ netstat: 显示端口信息
- ◆ kill: 杀掉进程
- ◆ jps: 显示Java进程信息

ps、netstat命令

- ◆ ps -ef: 显示系统内的所有进程
- ◆ netstat -anp: 可以查看端口信息

jps命令

- ◆ jps：类似ps命令，不同的是ps是用来显示所有进程信息的而jps只显示Java进程信息
- ◆ 准确的说jps是显示当前用户已启动的Java进程信息，信息包括进程号和简短的进程command
- ◆ 注意：jps是Java提供的命令，只有在安装配置好Java环境之后才能使用

top命令

- ◆ top命令：主要作用在于动态显示系统消耗资源最多的进程信息；包含进程ID、内存占用、CPU占用等
- ◆ 和ps命令作用基本相同，唯一的区别是top命令能够动态显示进程信息

kill命令

- ◆ kill PID: 杀掉进程, 自杀
- ◆ kill -9 PID: 强制杀掉进程, 它杀



Linux三剑客

- ◆ grep: 查找
- ◆ sed: 编辑
- ◆ awk: 分析



Linux三剑客之grep

- ◆ grep：常用于查找文件里符合条件的字符串
- ◆ -i：忽略大小写、-n：显示该行的行号、-v：忽略包含指定字符串的内容
- ◆ 注意：grep经常和管道 "|" 结合使用，实现数据过滤查找

Linux三剑客之sed

- ◆ sed：用来自动编辑一个或多个文件、简化对文件的反复操作
- ◆ a/i：增加、d:删除、s:替换、-i:直接修改源文件

Linux三剑客之awk

- ◆ awk：是一种处理文本的语言，是一个强大的文本分析工具
- ◆ 基本格式：awk [option] programme file

Linux高级配置

Linux高级配置

- ◆ Linux分配身份证号码(ip): 静态ip设置
- ◆ Linux起名字(hostname): 临时设置+永久设置
- ◆ Linux的金钟罩铁布衫(防火墙): 临时关闭+永久关闭

Linux之shell编程

什么是shell

- ◆ shell是用户与Linux操作系统沟通的桥梁



我的第一个shell脚本

- ◆ shell脚本文件名后缀通常是.sh
- ◆ shell脚本第一行内容是：#!/bin/bash
- ◆ 注意：其它行以#开头的表示是注释

执行我的shell脚本

- ◆ 格式: `bash hello.sh` 或者 `sh hello.sh`

- ◆ `./hello.sh` 或者 `hello.sh`

- ◆ `bash -x`: shell脚本的单步执行

shell中的变量

- ◆ 变量不需要声明, 初始化也不需要指定类型

- ◆ 变量命名: 只能使用数字、字母和下划线, 且不能以数字开头

- ◆ 变量赋值是通过"`=`"进行赋值, 在变量、等号和值之间不能出现空格!

变量的分类

◆ 本地变量

◆ 位置变量

◆ 环境变量

◆ 特殊变量

本地变量

◆ 格式: VAR_NAME=VALUE

◆ 应用场景: 在shell脚本中定义一些临时变量时使用, 对当前shell进程的子shell进程及其它shell进程无效

环境变量

- ◆ 格式: `export VAR_NAME=VALUE`
- ◆ 应用场景: 用于设置临时环境变量, 对子shell进程有效, 对其它shell进程无效
- ◆ 注意: 设置永久环境变量, 需要添加到配置文件/etc/profile中, 然后执行`source /etc/profile`可立刻生效

位置变量

- ◆ `$0`、`$1`、`$2...`
- ◆ 格式: `location.sh abc xyz`
- ◆ 位置变量相当于java中main函数的args参数, 可以在shell脚本中动态获取外部参数

特殊变量

- ◆ `$?`: 上一条命令的返回状态码, 状态码在0~255之间
- ◆ `$#`: shell脚本所有参数的个数

变量和引号的特殊使用

- ◆ `"`: 单引号不解析变量 `echo '$name'`
- ◆ `"`: 双引号解析变量 `echo "$name"`
- ◆ ```: 反引号是执行并引用命令的执行结果 `echo ` $name ``

变量和引号的特殊使用

◆ `$(...)`是``的另一种写法，效果一样

◆ 注意: `echo "$name"` 和 `echo "'$name'"`

shell中的循环和判断

◆ for循环

◆ while循环

◆ if判断



for循环

- ◆ `for((i=0;i<10;i++))`
do
循环体...
done

- ◆ `for i in 1 2 3`
do
循环体...
done



while循环

- ◆ 适用于循环次数未知，或不便于使用for直接生成较大列表时

- ◆ while 测试条件
do
循环体...
done

- ◆ 测试条件为"真"，则进入循环，测试条件为"假"，则退出循环

测试条件

- ◆ 格式: test EXPR 或者 [EXPR] 中括号和表达式之间的空格不能少
- ◆ 整型测试: -gt(大于)、-lt(小于)、-ge(大于等于)、-le(小于等于)、-eq(等于)、-ne(不等于)
- ◆ 字符串测试: =(等于)、!=(不等于)

if判断

- ◆ 单分支
- ◆ 双分支
- ◆ 多分支



if判断之单分支

◆ if 测试条件
then
 选择分支
fi



if判断之双分支

◆ if 测试条件
then
 选择分支1
else
 选择分支2
fi



if判断之多分支

- ◆ if 测试条件1
then
 选择分支1
elif 测试条件2
then
 选择分支2
...
else
 选择分支n
fi



shell扩展

- ◆ 后台模式运行脚本：nohup xxx.sh &
- ◆ 标准输出(1)、标准错误输出(2)、重定向(>或者>>)
- ◆ 案例：nohup hello.sh >/dev/null 2>&1 &

Linux中的定时器crontab

Linux中的定时器crontab

◆ crontab作用于周期性被执行的命令：每天凌晨1点去"偷菜"

◆ 格式： * * * * * user-name command

- 星期中的第几天(0-6)
- 月份(1-12)
- 月中的第几天(1-31)
- 小时(0-23)
- 分钟(0-59)

Linux中的定时器crontab

- ◆ 查看crontab服务状态：systemctl status crond
- ◆ 启动/停止crontab服务：systemctl start/stop crond
- ◆ 添加定时任务：vi /etc/crontab

Linux中的定时器crontab

- ◆ 查看crontab执行日志：tail -f /var/log/cron
- ◆ 问题：如果设置任务每7分钟执行一次，那么任务分别会在什么时间点执行？

实战：在Linux上安装配置JDK

