

2021 图解 Linux 全面升级

韩顺平

韩顺平教育

第 1 章 LINUX 开山篇-内容介绍	1
1.1 本套 LINUX 课程内容.....	1
1.2 LINUX 使用在那些地方.....	2
1.3 LINUX 的应用领域.....	2
1.3.1 个人桌面领域的应用.....	2
1.4 服务器领域.....	3
1.5 嵌入式领域.....	3
第 2 章 LINUX 基础篇-LINUX 入门	4
2.1 LINUX 介绍.....	4
2.1.1 概述.....	4
2.2 LINUX 和 UNIX 的关系.....	5
2.2.1 unix 是怎么来的.....	5
2.2.2 Linux 是怎么来的.....	5
2.2.3 Linux 和 Unix 关系图.....	6
第 3 章 LINUX 基础篇-VM 和 LINUX 的安装	7
3.1 安装 VM 和 CENTOS.....	7
3.1.1 基本说明.....	7
3.2 VMWARE15.5 下载.....	7
3.3 VM 安装的步骤.....	7
3.4 CENTOS 下载地址.....	8
3.5 CENTOS 安装的步骤.....	8
3.6 安装虚拟机和 CENTOS 的练习.....	9
3.7 虚拟机克隆.....	9
3.8 虚拟机快照.....	10
3.8.1 应用实例.....	10
3.9 虚拟机迁移和删除.....	10
3.10 安装 VMTTOOLS.....	11
3.10.1 介绍.....	11
3.10.2 安装 vmtools 的步骤.....	11
3.10.3 设置共享文件夹.....	12
第 4 章 LINUX 基础篇-目录结构	14
4.1 LINUX 目录结构.....	14
4.1.1 基本介绍.....	14
4.2 具体的目录结构(不用背,知道即可).....	14
第 5 章 LINUX 实操篇-远程登录到 LINUX 服务器	17
5.1 为什么需要远程登录 LINUX.....	17

5.1.1 说明: 公司开发时候, 具体的应用场景是这样的.....	17
5.2 远程登录 LINUX-XSHELL6.....	17
5.2.1 介绍.....	17
5.3 下载-安装-配置和使用.....	18
5.4 远程上传下载文件-XFTP6.....	18
5.4.1 介绍.....	18
5.4.2 Xftp6 安装配置和使用.....	19
5.4.3 细节讨论.....	19
5.5 XFTP6 和 XSHELL6 的使用.....	19
5.5.1 练习要求.....	20
第 6 章 LINUX 实操篇-VI 和 VIM 编辑器.....	21
6.1 VI 和 VIM 的基本介绍.....	21
6.2 VI 和 VIM 常用的三种模式.....	21
6.2.1 正常模式.....	21
6.2.2 插入模式.....	21
6.2.3 命令行模式.....	21
6.3 VI 和 VIM 基本使用.....	22
6.4 各种模式的相互切换.....	22
6.5 VI 和 VIM 快捷键.....	22
6.5.1 快捷键使用练习.....	22
6.6 VI 和 VIM 课堂练习.....	24
第 7 章 LINUX 实操篇-开机、重启和用户登录注销.....	25
7.1 关机&重启命令.....	25
7.1.1 基本介绍.....	25
7.1.2 注意细节.....	25
7.2 用户登录和注销.....	25
7.2.1 基本介绍.....	25
7.2.2 使用细节.....	25
第 8 章 LINUX 实操篇-用户管理.....	26
8.1 基本介绍.....	26
8.2 添加用户.....	26
8.2.1 基本语法.....	26
8.2.2 应用案例.....	26
8.2.3 细节说明.....	26
8.3 指定/修改密码.....	27
8.3.1 基本语法.....	27
8.3.2 应用案例.....	27

8.4 删除用户.....	27
8.4.1 基本语法.....	27
8.4.2 应用案例.....	27
8.4.3 细节说明.....	27
8.5 查询用户信息指令.....	27
8.5.1 基本语法.....	27
8.5.2 应用实例.....	28
8.5.3 细节说明.....	28
8.6 切换用户.....	28
8.6.1 介绍.....	28
8.6.2 基本语法.....	28
8.6.3 应用实例.....	28
8.6.4 细节说明.....	28
8.7 查看当前用户/登录用户.....	28
8.7.1 基本语法.....	28
8.8 用户组.....	28
8.8.1 介绍.....	29
8.8.2 新增组.....	29
8.8.3 删除组.....	29
8.8.4 案例演示.....	29
8.8.5 修改用户的组.....	30
8.9 用户和组相关文件.....	30
8.9.1 /etc/passwd 文件.....	30
8.9.2 /etc/shadow 文件.....	30
8.9.3 /etc/group 文件.....	30
第 9 章 LINUX 实操篇-实用指令.....	31
9.1 指定运行级别.....	31
9.1.1 基本介绍.....	31
9.1.2 应用实例.....	31
9.1.3 CentOS7 后运行级别说明.....	31
9.2 找回 ROOT 密码.....	32
9.2.1 面试题.....	32
9.2.2 课堂练习.....	32
9.3 帮助指令.....	32
9.3.1 man 获得帮助信息.....	32
9.3.2 help 指令.....	32
9.3.3 应用实例.....	32
9.3.4 百度帮助更直接.....	32

9.4 文件目录类.....	32
9.4.1 pwd 指令.....	32
9.4.2 ls 指令.....	33
9.4.3 cd 指令.....	33
9.4.4 mkdir 指令.....	33
9.4.5 rmdir 指令删除空目录.....	34
9.4.6 touch 指令.....	34
9.4.7 cp 指令.....	35
9.4.8 rm 指令.....	35
9.4.9 mv 指令.....	36
9.4.10 cat 指令.....	36
9.4.11 more 指令.....	37
9.4.12 less 指令.....	38
9.4.13 echo 指令.....	38
9.4.14 head 指令.....	39
9.4.15 tail 指令.....	39
9.4.16 > 指令 和 >> 指令.....	39
9.4.17 ln 指令.....	40
9.4.18 history 指令.....	40
9.5 时间日期类.....	41
9.5.1 date 指令-显示当前日期.....	41
9.5.2 date 指令-设置日期.....	42
9.5.3 cal 指令.....	42
9.6 搜索查找类.....	42
9.6.1 find 指令.....	42
9.6.2 locate 指令.....	43
9.6.3 grep 指令和 管道符号 	43
9.7 压缩和解压类.....	44
9.7.1 gzip/gunzip 指令.....	44
9.7.2 zip/unzip 指令.....	45
9.7.3 tar 指令.....	45
第 10 章 LINUX 实操篇-组管理和权限管理.....	47
10.1 LINUX 组基本介绍.....	47
10.2 文件/目录 所有者.....	47
10.2.1 查看文件的所有者.....	47
10.2.2 修改文件所有者.....	48
10.3 组的创建.....	48
10.3.1 基本指令.....	48

10.3.2 应用实例.....	48
10.4 文件/目录 所在组.....	48
10.4.1 查看文件/目录所在组.....	48
10.4.2 修改文件/目录所在的组.....	49
10.5 其它组.....	49
10.6 改变用户所在组.....	49
10.6.1 改变用户所在组.....	49
10.6.2 应用实例.....	49
10.7 权限的基本介绍.....	50
10.8 RWX 权限详解, 难点.....	50
10.8.1 rwx 作用到文件.....	50
10.8.2 rwx 作用到目录.....	51
10.9 文件及目录权限实际案例.....	51
10.9.1 ls -l 中显示的内容如下:	51
10.10 修改权限-CHMOD.....	52
10.10.1 基本说明:	52
10.10.2 第一种方式: + 、 -、 = 变更权限.....	52
10.10.3 第二种方式: 通过数字变更权限.....	52
10.11 修改文件所有者-CHOWN.....	53
10.11.1 基本介绍.....	53
10.12 修改文件/目录所在组-CHGRP.....	53
10.12.1 基本介绍.....	53
10.12.2 案例演示.....	53
10.13 最佳实践-警察和土匪游戏.....	54
10.14 课后练习.....	55
10.15 课堂练习 2, 完成如下操作.....	56
第 11 章 LINUX 实操篇-定时任务调度.....	57
11.1 CROND 任务调度.....	57
11.1.1 概述.....	57
11.1.2 基本语法.....	57
11.1.3 常用选项.....	57
11.1.4 快速入门.....	58
11.1.5 应用实例.....	59
11.1.6 crond 相关指令.....	59
11.2 AT 定时任务.....	60
11.2.1 基本介绍.....	60
11.2.2 at 命令格式.....	60
11.2.3 at 命令选项.....	60

11.2.4 at 时间定义.....	61
11.2.5 应用实例.....	61
第 12 章 LINUX 实操篇-LINUX 磁盘分区、挂载.....	63
12.1 LINUX 分区.....	63
12.1.1 原理介绍.....	63
12.1.2 硬盘说明.....	63
12.1.3 查看所有设备挂载情况.....	64
12.2 挂载的经典案例.....	64
12.2.1 说明:.....	64
12.2.2 如何增加一块硬盘.....	64
12.2.3 虚拟机增加硬盘步骤 1.....	65
12.2.4 虚拟机增加硬盘步骤 2.....	66
12.2.5 虚拟机增加硬盘步骤 3.....	66
12.2.6 虚拟机增加硬盘步骤 4.....	66
12.2.7 虚拟机增加硬盘步骤 5.....	67
12.3 磁盘情况查询.....	67
12.3.1 查询系统整体磁盘使用情况.....	67
12.3.2 查询指定目录的磁盘占用情况.....	68
12.4 磁盘情况-工作实用指令.....	68
第 13 章 LINUX 实操篇-网络配置.....	70
13.1 LINUX 网络配置原理图.....	70
13.1.1 画出原理图, 帮助理解.....	70
13.2 查看网络 IP 和网关.....	70
13.2.1 查看虚拟网络编辑器和修改 IP 地址.....	70
13.2.2 查看网关.....	71
13.3 查看 WINDOWS 环境中 VMNET8 网络配置 (IPCONFIG 指令).....	71
13.4 查看 LINUX 的网络配置 IFCONFIG.....	72
13.5 PING 测试主机之间网络连通性.....	72
13.5.1 基本语法.....	72
13.5.2 应用实例.....	72
13.6 LINUX 网络环境配置.....	72
13.6.1 第一种方法(自动获取):.....	72
13.6.2 第二种方法(指定 ip).....	73
13.7 设置主机名和 HOSTS 映射.....	74
13.7.1 设置主机名.....	74
13.7.2 设置 hosts 映射.....	74
13.8 主机名解析过程分析(HOSTS、DNS).....	75

13.8.1 Hosts 是什么.....	75
13.8.2 DNS.....	75
13.8.3 应用实例: 用户在浏览器输入了 www.baidu.com.....	75
第 14 章 LINUX 实操篇-进程管理(重点).....	77
14.1 基本介绍.....	77
14.2 显示系统执行的进程.....	77
14.2.1 基本介绍.....	77
14.2.2 ps 详解.....	78
14.2.3 应用实例.....	79
14.3 终止进程 KILL 和 KILLALL.....	79
14.3.1 介绍:.....	79
14.3.2 基本语法.....	79
14.3.3 常用选项.....	79
14.3.4 最佳实践.....	80
14.4 查看进程树 PSTREE.....	80
14.4.1 基本语法.....	80
14.4.2 常用选项.....	80
14.4.3 应用实例:	80
14.5 服务(SERVICE)管理.....	81
14.5.1 介绍:.....	81
14.5.2 service 管理指令.....	81
14.5.3 service 管理指令案例.....	81
14.5.4 查看服务名:.....	82
14.5.5 服务的运行级别(runlevel):.....	82
14.5.6 CentOS7 后运行级别说明.....	83
14.5.7 chkconfig 指令.....	83
14.5.8 systemctl 管理指令.....	84
14.5.9 systemctl 设置服务的自启动状态.....	84
14.5.10 应用案例:	84
14.5.11 细节讨论:	84
14.5.12 打开或者关闭指定端口.....	85
14.5.13 firewall 指令.....	85
14.5.14 应用案例:	85
14.6 动态监控进程.....	86
14.6.1 介绍:	86
14.6.2 基本语法.....	86
14.6.3 选项说明:	86
14.6.4 交互操作说明:	86

14.6.5 应用实例.....	87
14.7 监控网络状态.....	87
14.7.1 查看系统网络情况 netstat.....	87
14.7.2 检测主机连接命令 ping:	88
第 15 章 LINUX 实操篇-RPM 与 YUM.....	89
15.1 RPM 包的管理.....	89
15.1.1 介绍.....	89
15.1.2 rpm 包的简单查询指令.....	89
15.1.3 rpm 包名基本格式.....	89
15.1.4 rpm 包的其它查询指令:	89
15.1.5 卸载 rpm 包:	90
15.1.6 安装 rpm 包.....	91
15.2 YUM.....	91
15.2.1 介绍:	91
15.2.2 yum 的基本指令.....	92
15.2.3 安装指定的 yum 包.....	92
15.2.4 yum 应用实例:	92
第 16 章 LINUX 之 JAVAEE 定制篇-搭建 JAVAEE 环境.....	93
16.1 概述.....	93
16.2 安装 JDK.....	93
16.2.1 安装步骤.....	93
16.2.2 测试是否安装成功.....	93
16.3 TOMCAT 的安装.....	93
16.3.1 步骤:	94
16.3.2 测试是否安装成功:	94
16.4 IDEA2020 的安装.....	94
16.4.1 步骤.....	94
16.5 MYSQL5.7 的安装(!).....	95
第 17 章 大数据定制篇-SHELL 编程.....	96
17.1 为什么要学习 SHELL 编程.....	96
17.2 SHELL 是什么.....	96
17.3 SHELL 脚本的执行方式.....	96
17.3.1 脚本格式要求.....	96
17.3.2 编写第一个 Shell 脚本.....	96
17.3.3 脚本的常用执行方式.....	97
17.4 SHELL 的变量.....	97
17.4.1 Shell 变量介绍.....	97

17.4.2 shell 变量的定义.....	97
17.4.3 shell 变量的定义.....	99
17.5 设置环境变量.....	99
17.5.1 基本语法.....	99
17.5.2 快速入门.....	100
17.6 位置参数变量.....	100
17.6.1 介绍.....	100
17.6.2 基本语法.....	100
17.6.3 位置参数变量.....	100
17.7 预定义变量.....	101
17.7.1 基本介绍.....	101
17.7.2 基本语法.....	101
17.7.3 应用实例.....	101
17.8 运算符.....	102
17.8.1 基本介绍.....	102
17.8.2 基本语法.....	102
17.8.3 应用实例 oper.sh.....	102
17.9 条件判断.....	103
17.9.1 判断语句.....	103
17.10 流程控制.....	105
17.10.1 if 判断.....	105
17.10.2 case 语句.....	106
17.10.3 for 循环.....	107
17.10.4 while 循环.....	108
17.11 READ 读取控制台输入.....	109
17.11.1 基本语法.....	109
17.11.2 应用实例 testRead.sh.....	110
17.12 函数.....	110
17.12.1 函数介绍.....	110
17.12.2 系统函数.....	110
17.12.3 自定义函数.....	111
17.13 SHELL 编程综合案例.....	112
17.13.1 需求分析.....	112
17.13.2 代码 /usr/sbin/mysql_db.backup.sh.....	113
第 18 章 LINUX 之 PYTHON 定制篇-PYTHON 开发平台 UBUNTU.....	115
18.1 UBUNTU 介绍.....	115
18.2 安装的步骤.....	115
18.3 设置 UBUNTU 支持中文.....	115

18.4 UBUNTU 的 ROOT 用户.....	117
18.4.1 介绍.....	117
18.4.2 给 root 用户设置密码并使用.....	117
18.5 UBUNTU 下开发 PYTHON.....	117
18.5.1 说明.....	117
18.5.2 在 Ubuntu 下开发一个 Python 程序.....	118
第 19 章 LINUX 之 PYTHON 定制篇-APT 软件管理和远程登录.....	119
19.1 APT 介绍.....	119
19.2 UBUNTU 软件操作的相关命令.....	119
19.3 更新 UBUNTU 软件下载地址.....	120
19.3.1 原理介绍.....	120
19.3.2 寻找国内镜像源.....	120
19.3.3 寻找国内镜像源.....	121
19.3.4 备份 Ubuntu 默认的源地址.....	121
19.3.5 更新源服务器列表.....	122
19.3.6 更新源.....	122
19.4 UBUNTU 软件安装，卸载的最佳实践.....	123
19.5 远程登录 UBUNTU.....	123
19.5.1 ssh 介绍.....	123
19.5.2 原理示意图.....	124
19.5.3 安装 SSH 和启用.....	124
19.5.4 在 Windows 使用 XShell6/XFTP6 登录 Ubuntu.....	124
19.5.5 从一台 linux 系统远程登陆另外一台 linux 系统.....	124
第 20 章 CENTOS8.1 的使用.....	126
20.1 安装 CENTOS8.1/8.2.....	126
20.1.1 Centos 下载地址.....	126
20.1.2 CentOS8.1 安装的步骤.....	126
20.2 CENTOS8.0 VS CENTOS7.0 比较 (了解).....	126
第 21 章 LINUX 高级篇-日志管理.....	128
21.1 基本介绍.....	128
21.2 系统常用的日志.....	128
21.3 日志管理服务 RSYSLOGD.....	129
21.4 日志轮替.....	131
21.4.1 基本介绍.....	131
21.4.2 日志轮替文件命名.....	131
21.4.3 logrotate 配置文件.....	132
21.4.4 把自己的日志加入日志轮替.....	134

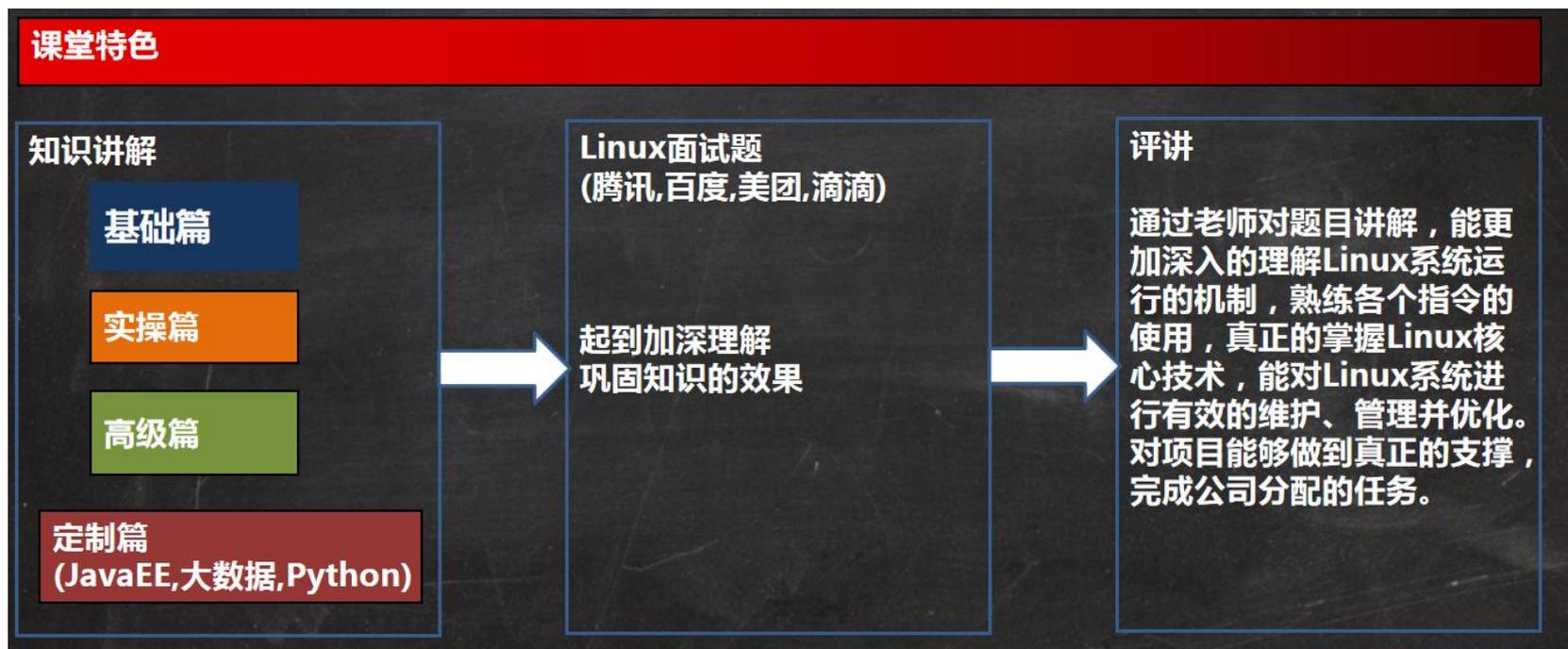
21.4.5 应用实例.....	134
21.5 日志轮替机制原理.....	134
21.6 查看内存日志.....	135
第 22 章 LINUX 高级篇-定制自己的 LINUX 系统.....	137
22.1 基本介绍.....	137
22.2 基本原理.....	137
22.3 制作 MIN LINUX 思路分析.....	137
22.4 操作步骤.....	138
第 23 章 LINUX 内核源码介绍&内核升级.....	139
23.1 为什么要阅读 LINUX 内核?.....	139
23.2 LINUX0.01 内核源码.....	139
23.2.1 基本介绍.....	139
23.2.2 linux0.01 内核源码目录&阅读.....	140
23.3 LINUX 内核最新版和内核升级.....	141
23.3.1 内核地址: https://www.kernel.org/ 查看.....	141
23.3.2 下载&解压最新版.....	141
23.3.3 linux 内核升级应用实例.....	141
23.3.4 具体步骤, 看老师演示.....	142
第 24 章 LINUX 系统-备份与恢复.....	143
24.1 基本介绍.....	143
24.2 安装 DUMP 和 RESTORE.....	143
24.3 使用 DUMP 完成备份.....	143
24.3.1 基本介绍.....	143
24.3.2 dump 语法说明.....	143
24.3.3 dump 应用案例 1.....	144
24.3.4 dump 应用案例 2.....	144
24.3.5 dump -W.....	144
24.3.6 查看备份时间文件.....	145
24.3.7 dump 备份文件或者目录.....	145
24.3.8 老韩提醒.....	145
24.4 使用 RESTORE 完成恢复.....	145
24.4.1 基本介绍.....	145
24.4.2 restore 基本语法.....	145
24.4.3 应用案例 1.....	146
24.4.4 应用案例 2.....	146
24.4.5 应用案例 3.....	147
24.4.6 应用案例 4.....	147

第 25 章 LINUX 可视化管理-WEBMIN 和 BT 运维工具.....	148
25.1 WEBMIN.....	148
25.1.1 基本介绍.....	148
25.1.2 安装 webmin&配置.....	148
25.1.3 简单使用演示.....	149
25.2 BT(宝塔).....	150
25.2.1 基本介绍.....	150
25.2.2 安装和使用.....	150
25.2.3 使用介绍，比如可以登录终端，配置，快捷安装运行环境和系统工具，添加计划任务脚本.....	150
25.2.4 如果 bt 的用户名，密码忘记了，使用 bt default 可以查看.....	151
第 26 章 LINUX 面试题-(腾讯,百度,美团,滴滴).....	152
26.1 分析日志 T.LOG(访问量)，将各个 IP 地址截取，并统计出现次数,并按从大到小排序(腾讯).....	152
26.2 统计连接到服务器的各个 IP 情况，并按连接数从大到小排序 (腾讯).....	152
26.3 问题：如忘记了 MYSQL5.7 数据库的 ROOT 用户的密码，如何找回?(滴滴).....	153
26.4 写出指令：统计 IP 访问情况，要求分析 NGINX 访问日志(ACCESS.LOG)，找出访问页面数量在前 2 位的 IP(美团).....	153
26.5 使用 TCPDUMP 监听本机，将来自 IP 192.168.200.1，TCP 端口为 22 的数据，保存输出到 TCPDUMP.LOG，用做将来数据分析(美团) >>>153	
26.6 常用的 NGINX 模块，用来做什么(头条).....	153
26.7 如果你是系统管理员，在进行 LINUX 系统权限划分时,应考虑哪些因素? (腾讯).....	154
26.8 权限操作思考题.....	155
26.9 说明 CENTOS7 启动流程，并说明和 CENTOS6 相同和不同的地方(腾讯).....	156
26.10 问题：列举 LINUX 高级命令，至少 6 个(百度).....	156
26.11 问题：LINUX 查看内存、IO 读写、磁盘存储、端口占用、进程查看命令是什么?(瓜子).....	156
26.12 使用 LINUX 命令计算 T2.TXT 第二列的和并输出 (美团).....	156
26.13 SHELL 脚本里如何检查一个文件是否存在？并给出提示(百度).....	156
26.14 用 SHELL 写一个脚本，对文本 T3.TXT 中无序的一系列数字排序，并将总和输出(百度).....	156
26.15 请用指令写出查找当前文件夹（/HOME）下所有的文本文件内容中包含有字符“CAT”的文件名称(金山).....	157
26.16 请写出统计/HOME 目录下所有文件个数和所有文件总行数的指令(在金山面试题扩展).....	157
26.17 列出你了解的 WEB 服务器负载架构(滴滴).....	157
26.18 每天晚上 10 点 30 分，打包站点目录/VAR/SPOOL/MAIL 备份到/HOME 目录下（每次备份按时间生成不同的备份包 比如按照 年月日时分秒）(滴滴).....	157
26.19 如何优化 LINUX 系统，说出你的方法 (瓜子).....	158
26.20 LINUX 面试 100 道题.....	159

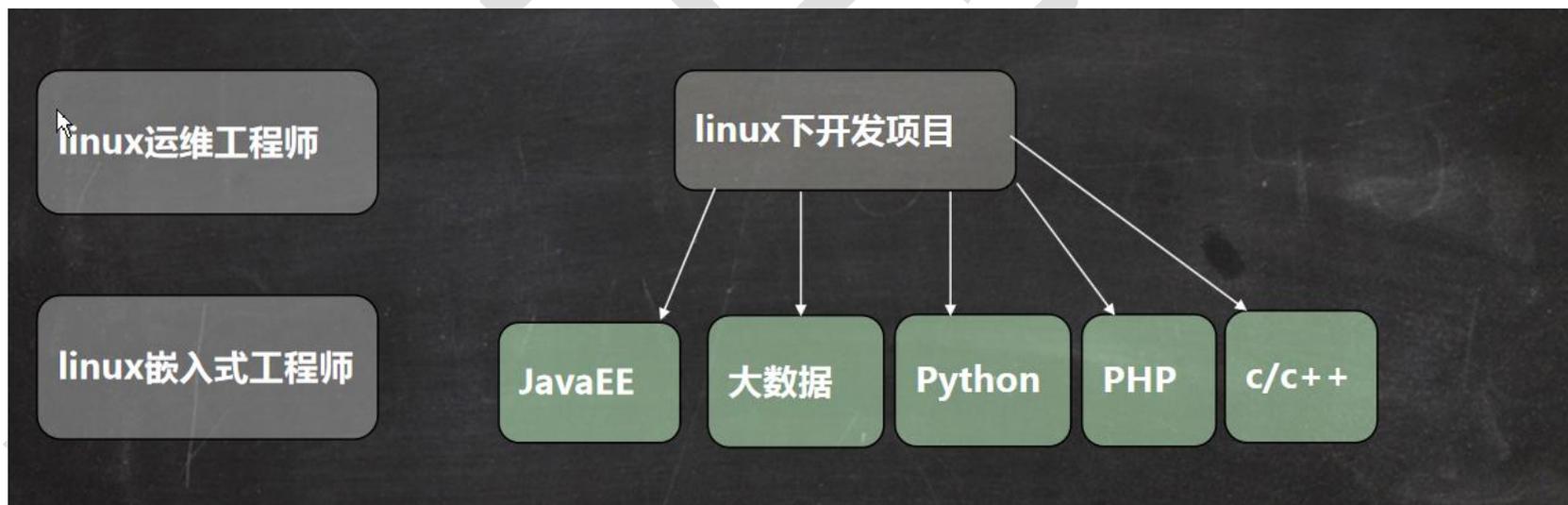
第 1 章 Linux 开山篇-内容介绍

1.1 本套 Linux 课程内容





1.2 Linux 使用在那些地方



1.3 Linux 的应用领域

1.3.1 个人桌面领域的应用

此领域是传统 linux 应用薄弱的环节, 近些年来随着 ubuntu、fedora [fi'do:rə] 等优秀桌面环境的兴起, linux 在个人桌面领域的占有率在逐渐的提高。



1.4 服务器领域

linux 在服务器领域的应用是最强的。

linux 免费、稳定、高效等特点在这里得到了很好的体现，尤其在一些高端领域尤为广泛（c/c++/php/java/python/go）。

1.5 嵌入式领域

linux 运行稳定、对网络的良好支持性、低成本，且可以根据需要进行**软件裁剪**，内核最小可以达到几百 KB 等特点，使其近些年来在**嵌入式领域**的应用得到非常大的提高

主要应用：机顶盒、数字电视、网络电话、程控交换机、手机、PDA、智能家居、智能硬件等都是其应用领域。以后在物联网中应用会更加广泛。

第 2 章 Linux 基础篇-Linux 入门

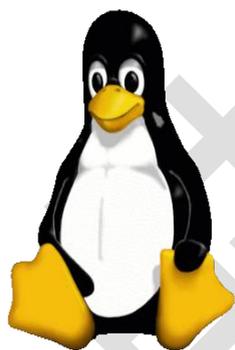
2.1 Linux 介绍

2.1.1 概述

1. linux 怎么读， 不下 10 种
2. linux 是一个开源、免费的操作系统，其稳定性、安全性、处理多并发已经得到业界的认可，目前很多企业级的项目(c/c++/php/python/java/go)都会部署到 Linux/unix 系统上。
3. 常见的操作系统(windows、IOS、 Android、 MacOS, Linux, Unix)



4. Linux 吉祥物



tux

5. xLinux 之父

Linus Torvalds

Git 创作者

世界著名黑客

小伙伴看看 linux0.01 版源码(不到 1w 行)

后面老韩给大家演示如何下载

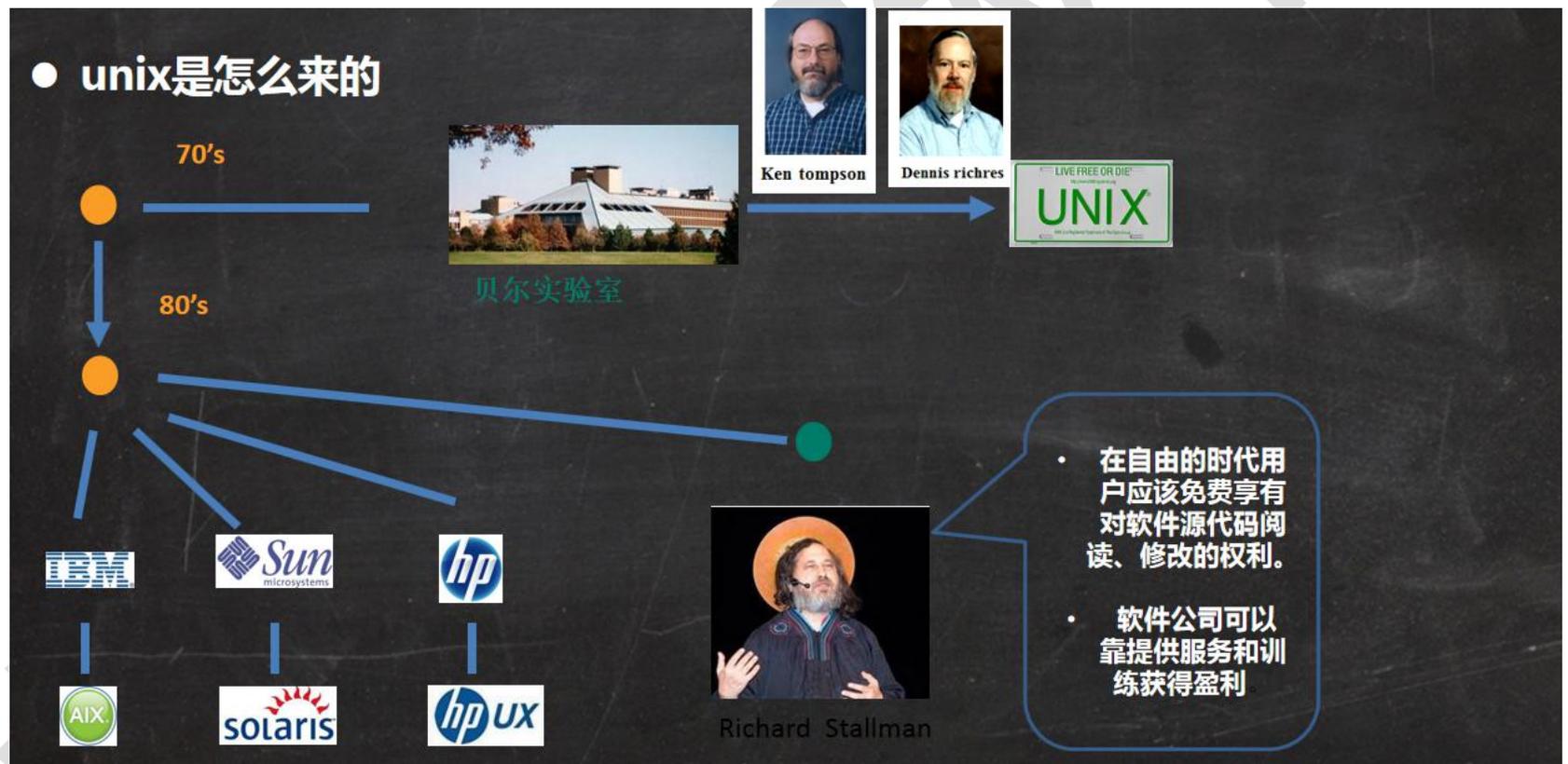
最新版本 linux 内核源码

6. Linux 主要的发行版:

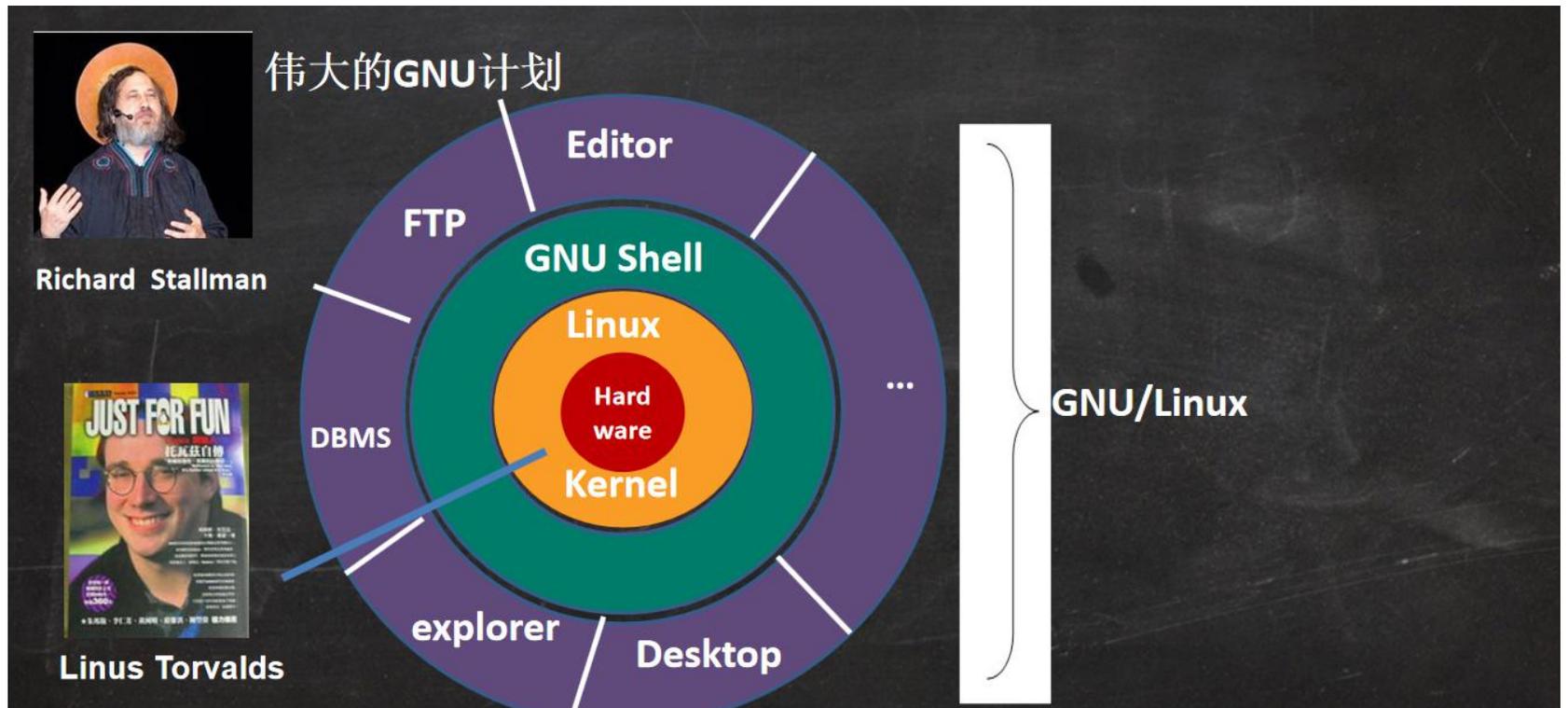
Ubuntu(乌班图)、RedHat(红帽)、CentOS、Debian[蝶变]、Fedora、SuSE、OpenSUSE [示意图]

2.2 Linux 和 Unix 的关系

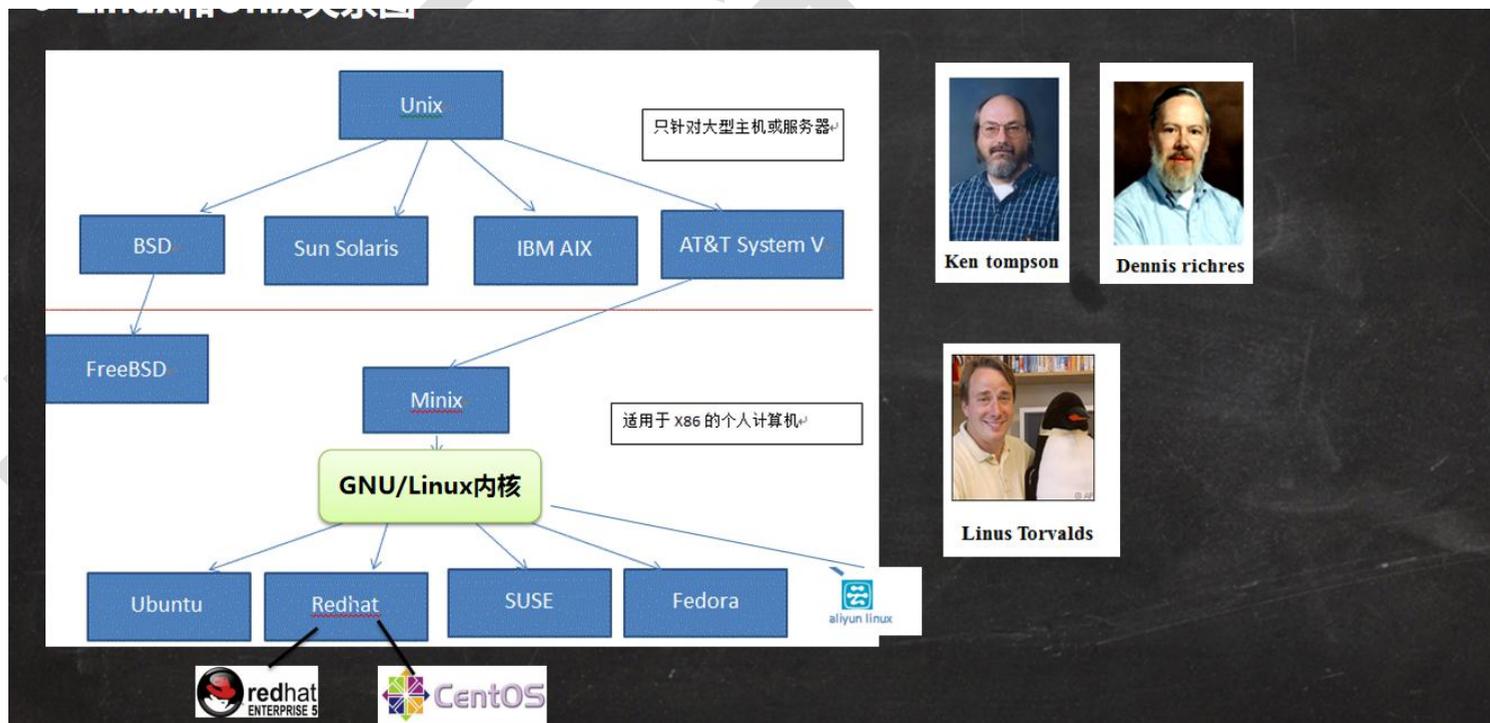
2.2.1 unix 是怎么来的



2.2.2 Linux 是怎么来的



2.2.3 Linux 和 Unix 关系图



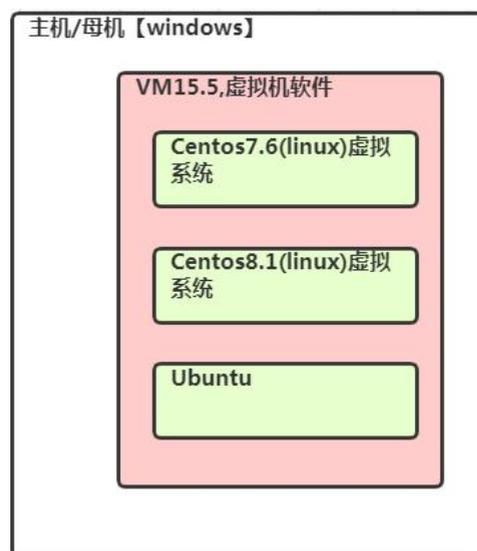
第 3 章 Linux 基础篇- VM 和 Linux 的安装

3.1 安装 vm 和 Centos

3.1.1 基本说明

学习 Linux 需要一个环境，我们需要创建一个虚拟机，然后在虚拟机上安装一个 Centos 系统来学习

1. 先安装 virtual machine 15.5
2. 再安装 Linux (CentOS 7.6/centOS8.1)
3. 原理示意图 [画]

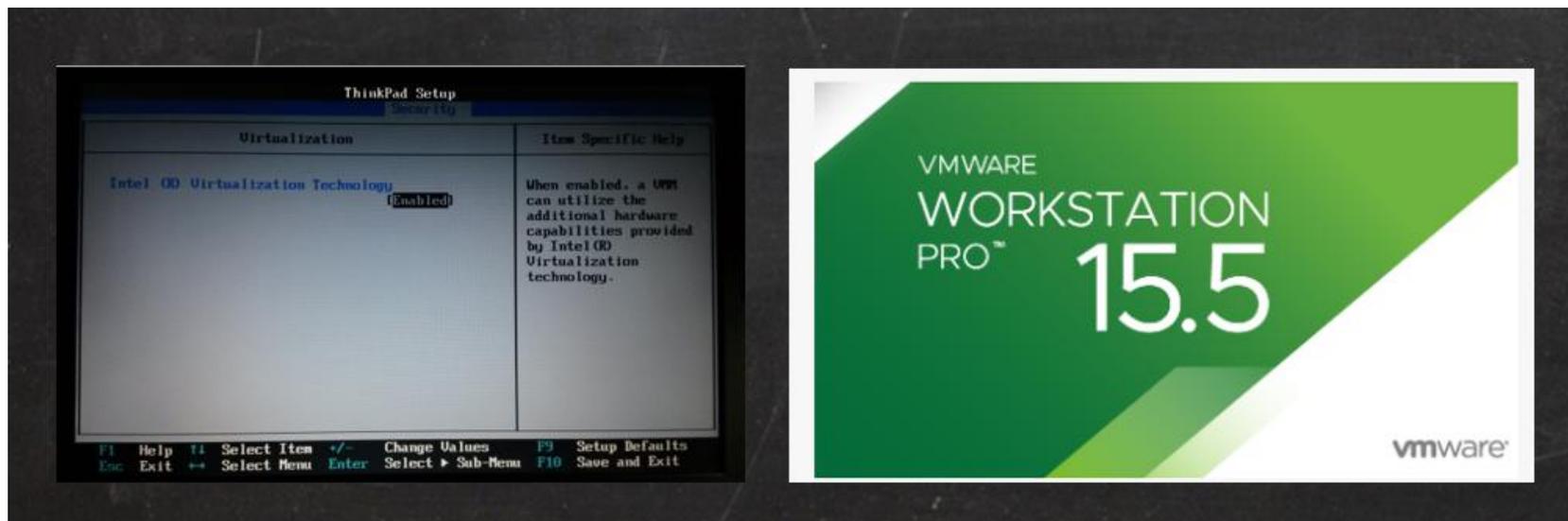


3.2 vmware15.5 下载

1. 官方地址: <https://www.vmware.com/cn.html>
2. 其它地址: <https://www.nocmd.com/windows/740.html>

3.3 VM 安装的步骤

1. 去 BIOS 里修改设置开启虚拟化设备支持 (f2, f10)
2. 安装虚拟机软件 (vm15.5), 演示一把, 1. 在 BIOS 开启 CPU 虚拟化支持
3. <https://jingyan.baidu.com/article/ab0b56305f2882c15afa7dda.html> 2. 再安装

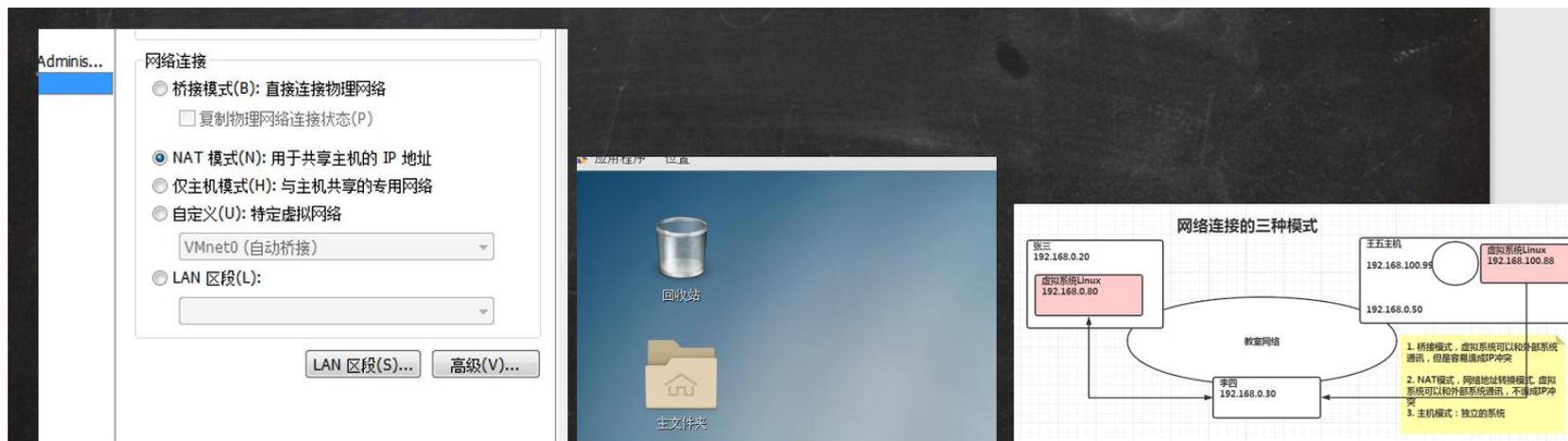


3.4 Centos 下载地址

1. CentOS-7-x86_64-DVD-1810.iso CentOS 7.6 DVD 版 4G (目前主流的生产环境)
http://mirrors.163.com/centos/7.6.1810/isos/x86_64/CentOS-7-x86_64-DVD-1810.iso
2. CentOS-8.1.1911-x86_64-dvd1.iso CentOS 8.1 DVD 版 8G (未来的主流.)
https://mirrors.aliyun.com/centos/8.1.1911/isos/x86_64/CentOS-8.1.1911-x86_64-dvd1.iso

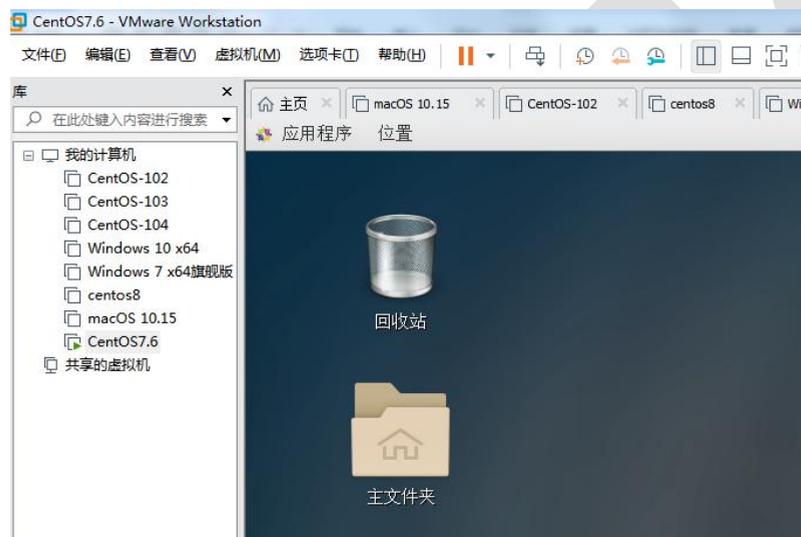
3.5 CentOS 安装的步骤

1. 创建虚拟机 [演示]
2. 开始安装系统(CentOS7.6)的步骤 [演示] 提示生成复杂密码:<https://suijimimashengcheng.51240.com/>
3. CentOS 安装难点-网络连接方式理解 [示意图]



3.6 安装虚拟机和 CentOS 的练习

学习 Linux 需要一个环境，我们需要创建一个虚拟机，然后在虚拟机上安装一个 Centos 系统来学习，请各位同学完成安装。



3.7 虚拟机克隆

如果你已经安装了一台 linux 操作系统，你还想再更多的，老韩提示大家：没有必要再重新安装，你只需要克隆就可以，看演示。

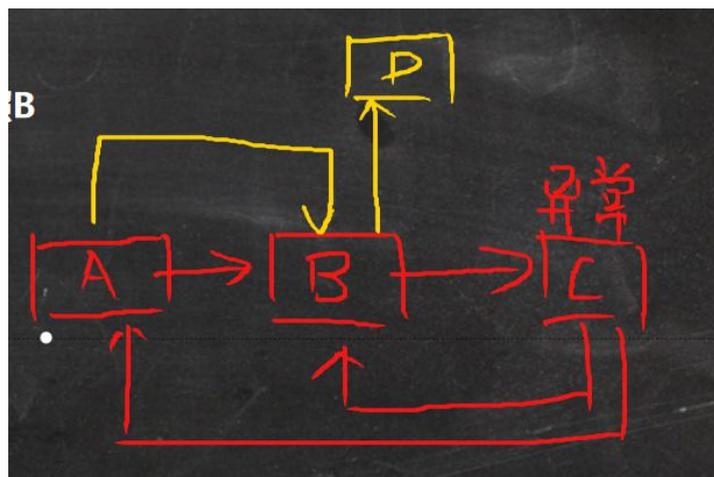
- 1) 方式 1，直接拷贝一份安装好的虚拟机文件
- 2) 方式 2，使用 vmware 的克隆操作，注意，克隆时，需要先关闭 linux 系统

3.8 虚拟机快照

如果你在使用虚拟机系统的时候(比如 linux)，你想回到原先的某一个状态，也就是说你担心可能有些误操作造成系统异常，需要回到原先某个正常运行的状态，vmware 也提供了这样的功能，就叫快照管理。

3.8.1 应用实例

- 1) 安装好系统以后，先做一个快照 A
- 2) 进入到系统。创建一个文件夹，再保存一个快照 B
- 3) 回到系统刚刚安装好的状态，即 快照 A
- 4) 试试看，是否还能再次回到快照 B
- 5) 示意图



3.9 虚拟机迁移和删除

我在前面讲过，虚拟系统安装好了，它的本质就是文件(放在文件夹的)。因此虚拟系统的迁移很方便，你可以把安装好的虚拟系统这个文件夹整体拷贝或者剪切到另外位置使用。删除也很简单，用 **vmware** 进行移除，再点击菜单->从

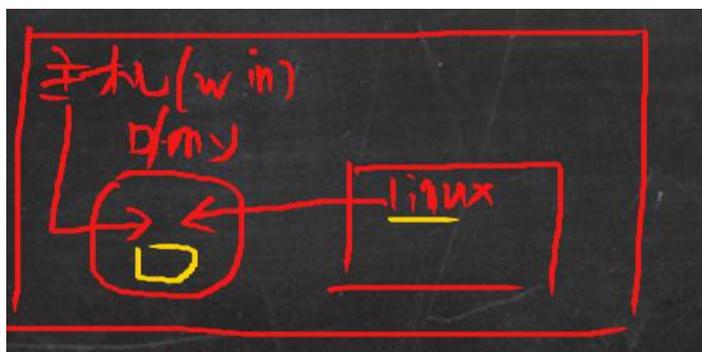
磁盘删除即可，或者直接手动删除虚拟系统对应的文件夹即可。

看老师给小伙伴演示一把（把前面克隆的系统进行迁移和删除）

3.10 安装 vmtools

3.10.1 介绍

- 1) vmtools 安装后，可以让我们在 windows 下更好的管理 vm 虚拟机
- 2) 可以设置 windows 和 centos 的共享文件夹
- 3) 示意图



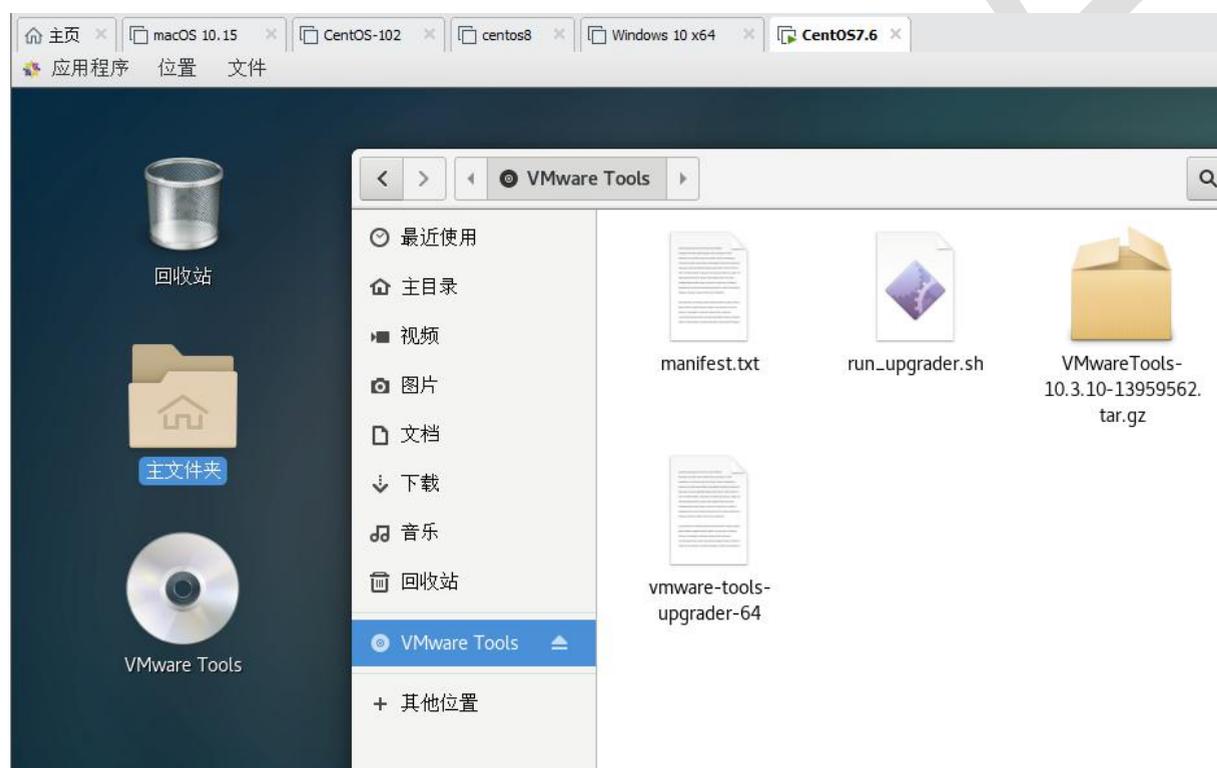
3.10.2 安装 vmtools 的步骤

- 1) 进入 centos
- 2) 点击 vm 菜单的->install vmware tools
- 3) centos 会出现一个 vm 的安装包, xx.tar.gz
- 4) 拷贝到 /opt
- 5) 使用解压命令 tar, 得到一个安装文件
cd /opt [进入到 opt 目录]
tar -zxvf xx.tar.gz
- 6) 进入该 vm 解压的目录, /opt 目录下
cd vmware...

- 7) 安装 `./vmware-install.pl`
- 8) 全部使用默认设置即可, 就可以安装成功
- 9) 注意: 安装 `vmtools` 需要有 `gcc` .

```
gcc -v
```

- 10) 示意图



3.10.3 设置共享文件夹

- 基本介绍

为了方便, 可以设置一个共享文件夹, 比如 `d:/myshare`

- 具体步骤

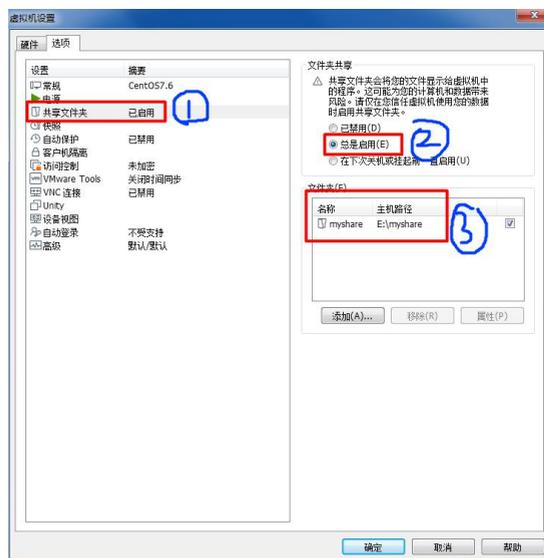
- 1) 菜单->vm->setting, 如图设置即可注意:设置选项为 `always enable` ,这样可以读写了

- 2) windows 和 centos 可共享 d:/myshare 目录可以读写文件了
- 3) 共享文件夹在 centos 的 /mnt/hgfs/ 下

- 注意事项和细节说明

- 1) windows 和 contos 就可以共享文件了，但是在实际开发中，文件的上传下载是需要使用 远程方式完成的
- 2) 远程方式登录，我们后面会具体讲解

示意图



第 4 章 Linux 基础篇-目录结构

4.1 Linux 目录结构

4.1.1 基本介绍

- 1) linux 的文件系统是采用级层式的树状目录结构，在此结构中的最上层是根目录“/”，然后在此目录下再创建其他的目录。
- 2) 深刻理解 linux 树状文件目录是非常重要的，这里我给大家说明一下。
- 3) 记住一句经典的话：在 Linux 世界里，一切皆文件(!!)
- 4) 示意图



4.2 具体的目录结构(不用背,知道即可)

- 1) /bin [常用] (/usr/bin 、 /usr/local/bin)
是 Binary 的缩写, 这个目录存放着最经常使用的命令
- 2) /sbin (/usr/sbin 、 /usr/local/sbin)
s 就是 Super User 的意思, 这里存放的是系统管理员使用的系统管理程序。
- 3) /home [常用]
存放普通用户的主目录, 在 Linux 中每个用户都有一个自己的目录, 一般该目录名是以用户的账号命名, 老师演示,

体验

4) /root [常用]

该目录为系统管理员，也称作超级权限者的用户主目录

5) /lib 系统开机所需要最基本的动态连接共享库，其作用类似于 Windows 里的 DLL 文件。几乎所有的应用程序都需要用到这些共享库

6) /lost+found 这个目录一般情况下是空的，当系统非法关机后，这里就存放了一些文件

7) /etc [常用]

所有的系统管理所需要的配置文件和子目录，比如安装 mysql 数据库 my.conf

8) /usr [常用]

这是一个非常重要的目录，用户的很多应用程序和文件都放在这个目录下，类似与 windows 下的 program files 目录。

9) /boot [常用] 存放的是启动 Linux 时使用的一些核心文件，包括一些连接文件以及镜像文件

10) /proc [不能动] 这个目录是一个虚拟的目录，它是系统内存的映射，访问这个目录来获取系统信息

11) /srv [不能动] service 缩写，该目录存放一些服务启动之后需要提取的数据

12) /sys [不能动] 这是 linux2.6 内核的一个很大的变化。该目录下安装了 2.6 内核中新出现的一个文件系统 sysfs => 【别动】

13) /tmp 这个目录是用来存放一些临时文件的

14) /dev

类似于 windows 的设备管理器，把所有的硬件用文件的形式存储

15) /media [常用] linux 系统会自动识别一些设备，例如 U 盘、光驱等等，当识别后，linux 会把识别的设备挂载到这个目录下

16) /mnt [常用]

系统提供该目录是为了让用户临时挂载别的文件系统的，我们可以将外部的存储挂载在/mnt/上，然后进入该目录就

可以查看里的内容了。 `d:/myshare`

17) `/opt` 这是给主机额外安装软件所存放的目录。如安装 ORACLE 数据库就可放到该目录下。默认为空

18) `/usr/local` [常用]

这是另一个给主机额外安装软件所安装的目录。一般是通过编译源码方式安装的程序

19) `/var` [常用]

这个目录中存放着在不断扩充着的东西，习惯将经常被修改的目录放在这个目录下。包括各种日志文件

20) `/selinux` [security-enhanced linux]

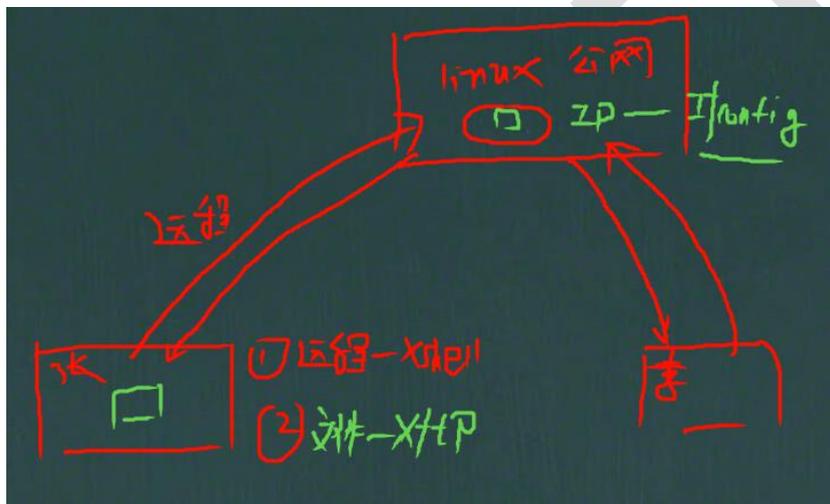
SELinux 是一种安全子系统,它能控制程序只能访问特定文件,有三种工作模式,可以自行设置.

第 5 章 Linux 实操篇-远程登录到 Linux 服务器

5.1 为什么需要远程登录 Linux

5.1.1说明: 公司开发时候, 具体的应用场景是这样的

- 1) linux 服务器是开发小组共享
- 2) 正式上线的项目是运行在公网
- 3) 因此程序员需要远程登录到 Linux 进行项目管理或者开发
- 4) 画出简单的网络拓扑示意图(帮助理解)
- 5) 远程登录客户端有 Xshell6, Xftp6, 我们学习使用 Xshell 和 Xftp6, 其它的远程工具大同小异
- 6) 示意图



5.2 远程登录 Linux-Xshell6

5.2.1介绍

- 1) 说明: Xshell 是目前最好的远程登录到 Linux 操作的软件, 流畅的速度并且完美解决了中文乱码的问题, 是目前程

序员首选的软件。

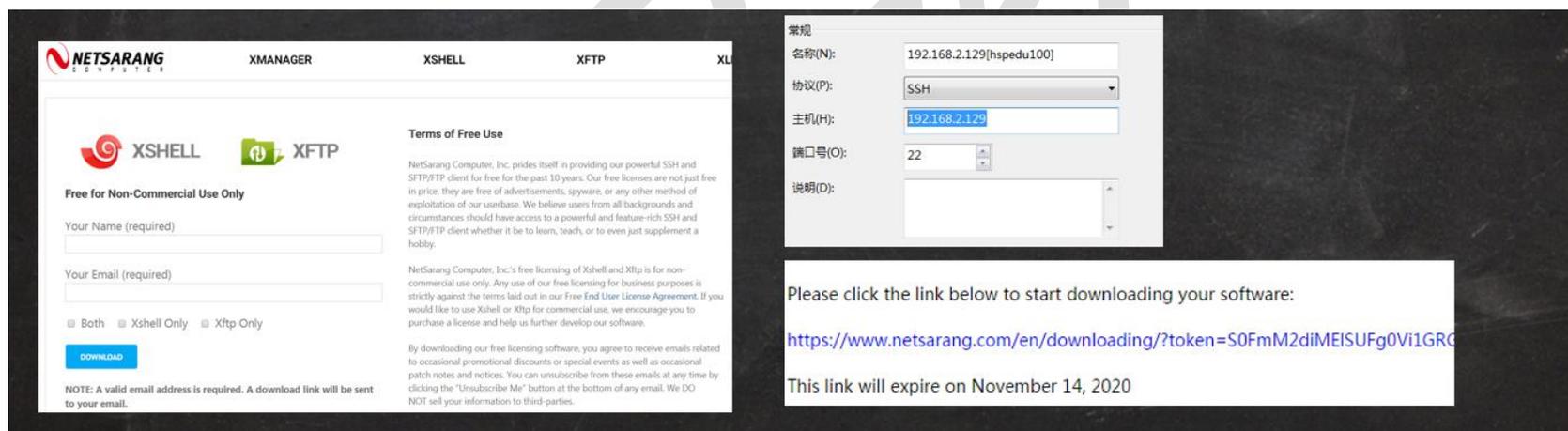
- 2) Xshell 是一个强大的安全终端模拟软件，它支持 SSH1, SSH2, 以及 Microsoft Windows 平台的 TELNET 协议。
- 3) Xshell 可以在 Windows 界面下用来访问远端不同系统下的服务器，从而比较好的达到远程控制终端的目的

5.3 下载-安装-配置和使用:

具体看老师的演示和操作

下载 free-for-home-school 版本

地址: <https://www.netsarang.com/en/free-for-home-school/>



5.4 远程上传下载文件-Xftp6

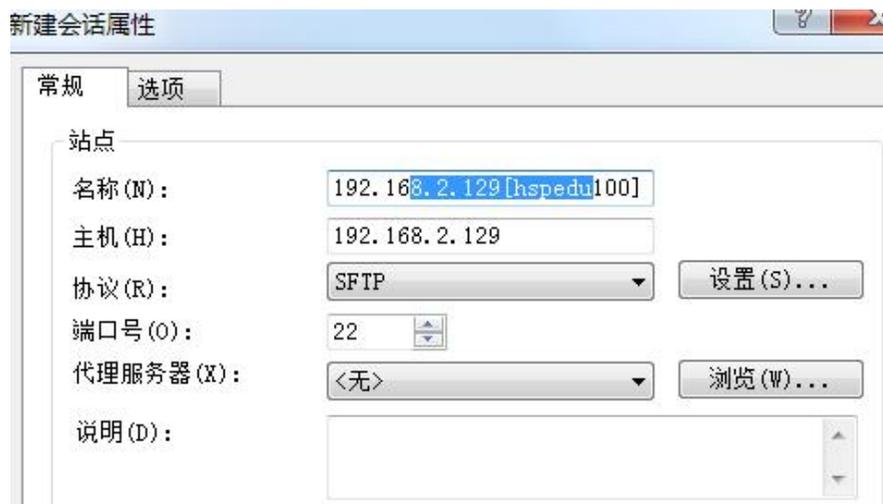
5.4.1 介绍

是一个基于 windows 平台的功能强大的 SFTP、FTP 文件传输软件。使用了 Xftp 以后，windows 用户能安全地在 UNIX/Linux 和 Windows PC 之间传输文件。[示意图]

软件下载方法在前面说过了

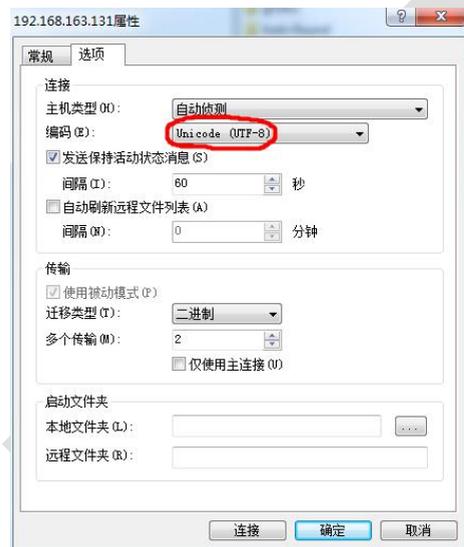
5.4.2 Xftp6 安装配置和使用

具体看老师的演示和操作



5.4.3 细节讨论

如何处理 Xftp 中文乱码问题



5.5 Xftp6 和 XShell6 的使用

5.5.1 练习要求

- 1) 通过 Xftp6 软件可以连接到 linux ,并传输一个文件(比如图片)到 /root 目录
- 2) 使用 xshell6 可以连接到 linux ,并且可以重新启动一把 linux 系统 , 提示: reboot

第 6 章 Linux 实操篇- Vi 和 Vim 编辑器

6.1 vi 和 vim 的基本介绍

Linux 系统会内置 vi 文本编辑器

Vim 具有程序编辑的能力，可以看做是 Vi 的增强版本，可以主动的以字体颜色辨别语法的正确性，方便程序设计。代码补完、编译及错误跳转等方便编程的功能特别丰富，在程序员中被广泛使用。



6.2 vi 和 vim 常用的三种模式

6.2.1 正常模式

以 vim 打开一个档案就直接进入一般模式了(这是**默认的模式**)。在这个模式中，你可以使用『上下左右』按键来移动光标，你可以使用『删除字符』或『删除整行』来处理档案内容，也可以使用『复制、粘贴』来处理你的文件数据。

6.2.2 插入模式

按下 i, I, o, O, a, A, r, R 等任何一个字母之后才会进入编辑模式，一般来说按 i 即可。

6.2.3 命令行模式

输入 esc 再输入：在这个模式当中，可以提供你相关指令，完成读取、存盘、替换、离开 vim 、显示行号等的动作则是在此模式中达成的！

6.3 vi 和 vim 基本使用

使用 vim 开发一个 Hello.java 程序, 保存. 步骤说明和演示

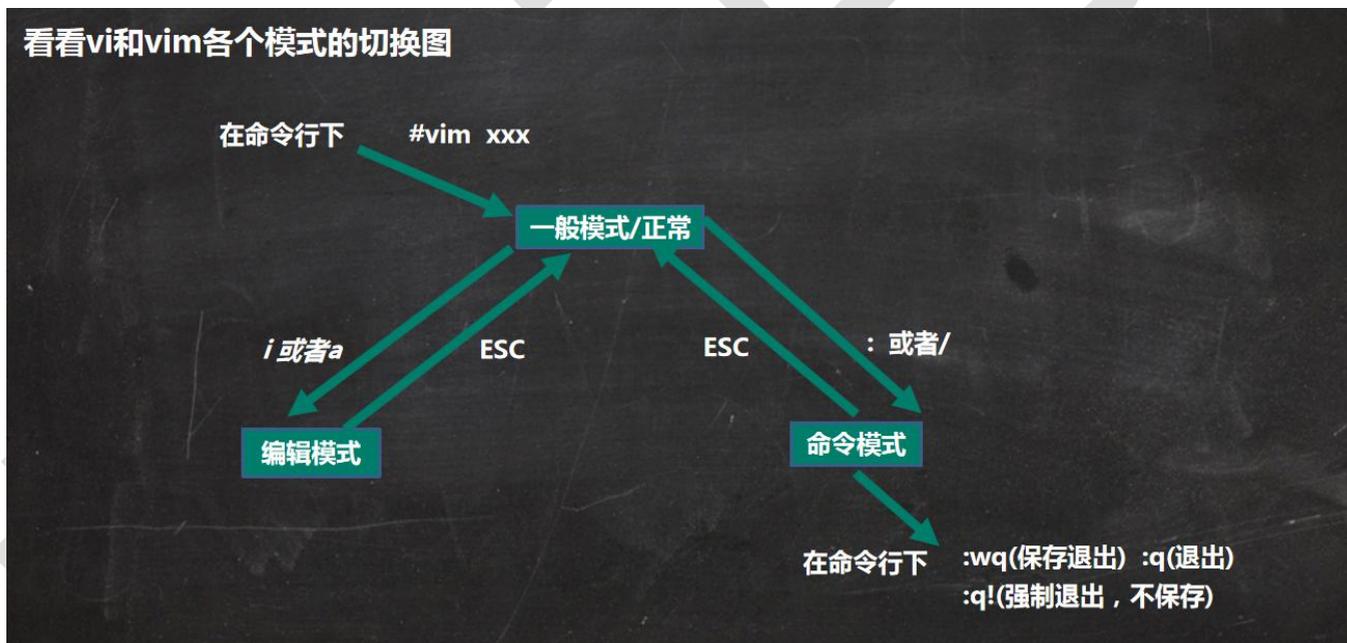
```
public class Hello {

    public static void main(String[] args) {
    public static void main(String[] args) {

        System.out.println("hello,world~");
    }

    System.out.println("hello,world~");
}
}
```

6.4 各种模式的相互切换



6.5 vi 和 vim 快捷键

6.5.1 快捷键使用练习

- 1) 拷贝当前行 yy, 拷贝当前行向下的 5 行 5yy, 并粘贴 (输入 p)。
- 2) 删除当前行 dd, 删除当前行向下的 5 行 5dd
- 3) 在文件中查找某个单词 [命令行下 /关键字, 回车 查找, 输入 n 就是查找下一个]
- 4) 设置文件的行号, 取消文件的行号.[命令行下 :set nu 和 :set nonu]
- 5) 编辑 /etc/profile 文件, 在一般模式下, 使用快捷键到该文档的最末行[G]和最首行[gg]
- 6) 在一个文件中输入 "hello",在一般模式下, 然后又撤销这个动作 u
- 7) 编辑 /etc/profile 文件, 在一般模式下, 并将光标移动到 , 输入 20,再输入 shift+g
- 8) 更多的看整理的文档
- 9) 快捷键的键盘对应图

翻译:2006-5-22

vi / vim 键盘图

Esc
命令模式

~ 转换大小写	! 外部过滤器	@ 运行宏	# prev ident	\$ 行末	% 括号匹配	^ "软"行首	& 重复 :s	* next ident	(句首) 下一句首	"soft" bol down	+ 后一行行首
跳转到标注	1	2	3	4	5	6	7	8	9	0 "硬"行首	- 前一行行首	= 自动格式化
Q 切换到 ex 模式	W 下一单词	E 词尾	R 替换模式	T back 'till	Y 拷贝行	U 撤销行内命令	I 到行首插入	O 分段(前)	P 粘贴(前)	{ 段首	}	段尾
q 录制宏	w 下一单词	e 词尾	r 替换字符	t 'till	y 拷贝	u 撤销命令	i 插入模式	o 分段(后)	p 粘贴(后)	[杂项]	杂项
A 在行末附加	S 删除行并插入	D 删除至行末	F 行内字符反向查找	G 文尾/行号	H 屏幕顶行	J 合并两行	K 帮助	L 屏幕底行	:	ex 命令	" 寄存器标识	行首/列
a 附加	s 删除字符并插入	d 删除	f 行内字符查找	g 附加命令	h ←	j ↓	k ↑	l →	:	重复 t/T/f/F	' 跳转到标注的行首	\ 未使用!
Z 退出	X 退格	C 修改至行末	V 可视行模式	B 前一单词	N 查找上一处	M 屏幕中间行	< 反缩进	> 缩进	?	向前搜索		
Z 附加命令	X 删除(字符)	c 修改	v 可视模式	b 前一单词	n 查找下一处	m 设置标注	, t/T/f/F	.	重复命令	/	向后搜索	

- 动作** 移动光标, 或者定义操作的范围
- 命令** 直接执行的命令, 红色命令进入编辑模式
- 操作** 后面跟随表示操作范围的指令
- extra** 特殊功能, 需要额外的输入
- q.** 后面跟字符参数

w,e,b命令
小写(b): `quux(foo, bar, baz);`
大写(B): `quux(FOO, BAR, BAZ);`

主要ex命令:
:w (保存), :q (退出), :q! (不保存退出)
:e f (打开文件 f),
:%s/x/y/g ('y' 全局替换 'x'),
:h (帮助 in vim), :new (新建文件 in vim)

其它重要命令:
CTRL-R: 重复 (vim),
CTRL-F/-B: 上翻/下翻,
CTRL-E/-Y: 上滚/下滚,
CTRL-V: 块可视模式 (vim only)

可视模式:
漫游后对选中的区域执行操作 (vim only)

- 备注:
- (1) 在 拷贝/粘贴/删除 命令前使用 "x (x=a..z,*) 使用命令的寄存器('剪贴板') (如: "ay\$ 拷贝剩余的行内容至寄存器 'a')
 - (2) 命令前添加数字 多遍重复操作 (e.g.: 2p, d2w, 5i, d4j)
 - (3) 重复本字符在光标所在行执行操作 (dd = 删除本行, >> = 行首缩进)
 - (4) ZZ 保存退出, ZQ 不保存退出
 - (5) zt: 移动光标所在行至屏幕顶端, zb: 底端, zz: 中间
 - (6) gg: 文首 (vim only), gf: 打开光标处的文件名 (vim only)

6.6 vi 和 vim 课堂练习

- 1) 看使用 vim / vi 编写一个 Hello.java
- 2) 把老师课堂讲的各种快捷键自己玩一把即可

第 7 章 Linux 实操篇-开机、重启和用户登录注销

7.1 关机&重启命令

7.1.1 基本介绍

- 1) shutdown -h now 立该进行关机
- 2) shutdown -h 1 "hello, 1 分钟后会关机了"
- 3) shutdown -r now 现在重新启动计算机
- 4) halt 关机，作用和上面一样.
- 5) reboot 现在重新启动计算机
- 6) sync 把内存的数据同步到磁盘.

7.1.2 注意细节

- 1) 不管是重启系统还是关闭系统，首先要运行 **sync** 命令，把内存中的数据写到磁盘中
- 2) 目前的 shutdown/reboot/halt 等命令均已经在关机前进行了 sync ， 老韩提醒: 小心驶得万年船

7.2 用户登录和注销

7.2.1 基本介绍

- 1) 登录时尽量少用 root 帐号登录，因为它是系统管理员，最大的权限，避免操作失误。可以利用普通用户登录，登录后再用 "su - 用户名" 命令来切换成系统管理员身份.
- 2) 在提示符下输入 **logout** 即可注销用户

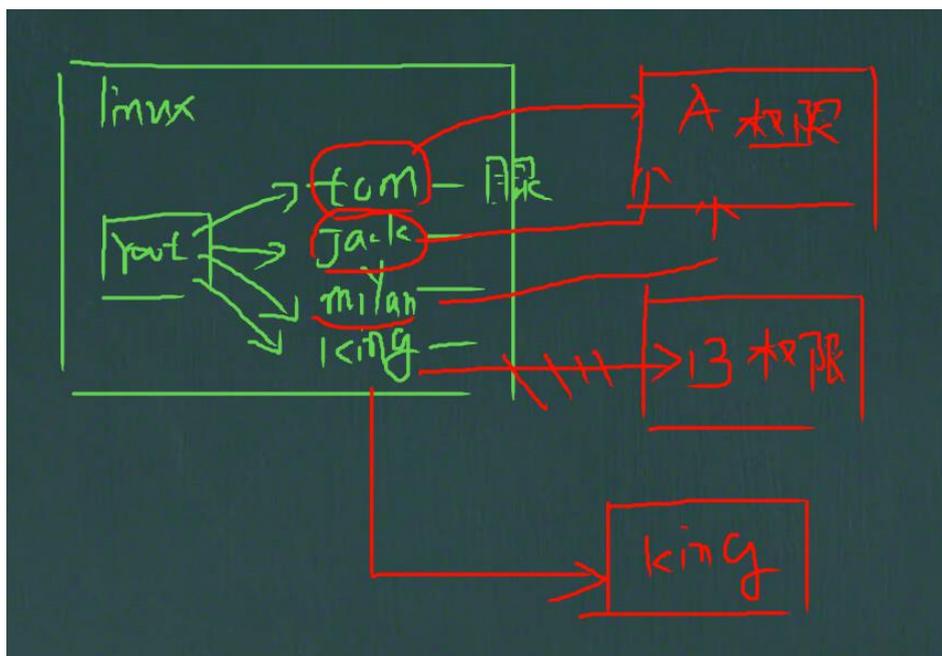
7.2.2 使用细节

- 1) **logout** 注销指令在图形运行级别无效，在运行级别 3 下有效.
- 2) 运行级别这个概念，后面给大家介绍

第 8 章 Linux 实操篇-用户管理

8.1 基本介绍

Linux 系统是一个多用户多任务的操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统



8.2 添加用户

8.2.1 基本语法

useradd 用户名

8.2.2 应用案例

案例 1: 添加一个用户 milan, 默认该用户的家目录在 /home/milan

8.2.3 细节说明

1) 当创建用户成功后, 会自动的创建和用户同名的家目录

2) 也可以通过 `useradd -d` 指定目录 新的用户名, 给新创建的用户指定家目录

8.3 指定/修改密码

8.3.1 基本语法

`passwd` 用户名

8.3.2 应用案例

给 milan 指定密码

```
passwd milan
```

补充, 显示当前用户所在的目录 `pwd`

8.4 删除用户

8.4.1 基本语法

`userdel` 用户名

8.4.2 应用案例

- 1) 删除用户 milan, 但是要保留家目录, `userdel milan`
- 2) 删除用户以及用户主目录, 比如 tom, `userdel -r tom`

8.4.3 细节说明

是否保留家目录的讨论? 一般情况下, 我们建议保留

8.5 查询用户信息指令

8.5.1 基本语法

`id` 用户名

8.5.2 应用实例

案例：请查询 root 信息

```
#id root
```

8.5.3 细节说明

当用户不存在时，返回无此用户

8.6 切换用户

8.6.1 介绍

在操作 Linux 中，如果当前用户的权限不够，可以通过 `su -` 指令，切换到高权限用户，比如 root

8.6.2 基本语法

`su -` 切换用户名

8.6.3 应用实例

创建一个用户 jack，指定密码，然后切换到 jack

8.6.4 细节说明

- 1) 从权限高的用户切换到权限低的用户，不需要输入密码，反之需要。
- 2) 当需要返回到原来用户时，使用 `exit/logout` 指令

8.7 查看当前用户/登录用户

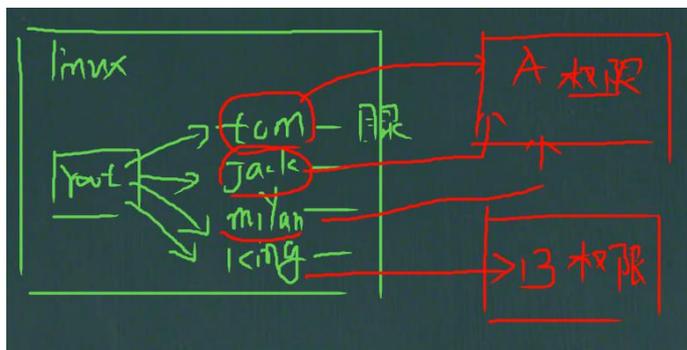
8.7.1 基本语法

`whoami/who am I`

8.8 用户组

8.8.1 介绍

类似于角色，系统可以对有共性/权限的多个用户进行统一的管理



8.8.2 新增组

指令 : `groupadd` 组名

案例演示

8.8.3 删除组

指令(基本语法): `groupdel` 组名

8.8.4 案例演示

- 1) 增加用户时直接加上组
- 2) 指令(基本语法): `useradd -g 用户组 用户名`
- 3) 增加一个用户 `zwj`, 直接将他指定到 `wudang`
- 4) `groupadd wudang`
- 5) `useradd -g wudang zwj`

8.8.5 修改用户的组

指令(基本语法): `usermod -g 用户组 用户名`

- 案例演示

创建一个组 `mojiao`

把 `zwj` 放入到 `mojiao`

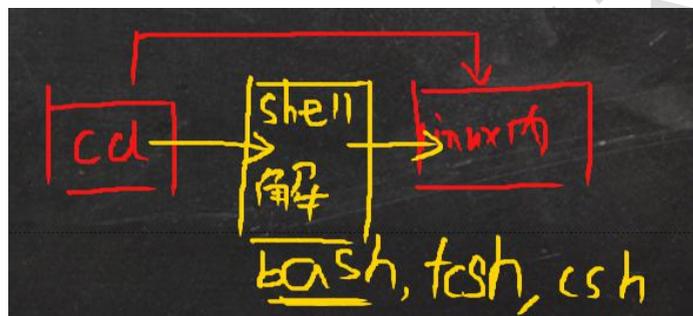
指令: `usermod -g mojiao zwj`

8.9 用户和组相关文件

8.9.1 /etc/passwd 文件

用户 (user) 的配置文件, 记录用户的各种信息

每行的含义: 用户名:口令:用户标识号:组标识号:注释性描述:主目录:登录 Shell



8.9.2 /etc/shadow 文件

口令的配置文件

每行的含义: 登录名:加密口令:最后一次修改时间:最小时间间隔:最大时间间隔:警告时间:不活动时间:失效时间:标志

8.9.3 /etc/group 文件

组(group)的配置文件, 记录 Linux 包含的组的信息

每行含义: 组名:口令:组标识号:组内用户列表

第 9 章 Linux 实操篇-实用指令

9.1 指定运行级别

9.1.1 基本介绍

运行级别说明：

- 0 : 关机
- 1 : 单用户【找回丢失密码】
- 2: 多用户状态没有网络服务
- 3: 多用户状态有网络服务
- 4: 系统未使用保留给用户
- 5: 图形界面
- 6: 系统重启

常用运行级别是 3 和 5 ， 也可以指定默认运行级别， 后面演示

9.1.2 应用实例

命令：init [0123456] 应用案例： 通过 init 来切换不同的运行级别， 比如动 5-3 ， 然后关机。

9.1.3 CentOS7 后运行级别说明

在 centos7 以前， /etc/inittab 文件中

进行了简化 ， 如下：

multi-user.target: analogous to runlevel 3

graphical.target: analogous to runlevel 5

To view current default target, run:

systemctl get-default

To set a default target, run:

systemctl set-default TARGET.target

9.2 找回 root 密码

9.2.1 面试题

如何找回 root 密码

9.2.2 课堂练习

- 1) 假设 root 密码忘记了，请问如何找回密码
- 2) 设置运行级别，linux 运行后，直接进入命令行终端(3)

9.3 帮助指令

9.3.1 man 获得帮助信息

基本语法：man [命令或配置文件]（功能描述：获得帮助信息）

案例：查看 ls 命令的帮助信息 `man ls`

在 linux 下，隐藏文件是以 . 开头，选项可以组合使用 比如 `ls -al`，比如 `ls -al /root`

9.3.2 help 指令

基本语法：help 命令（功能描述：获得 shell 内置命令的帮助信息）

9.3.3 应用实例

案例：查看 cd 命令的帮助信息

9.3.4 百度帮助更直接

如果英语不太好的，也可以直接百度靠谱。

9.4 文件目录类

9.4.1 pwd 指令

基本语法：pwd（功能描述：显示当前工作目录的绝对路径）

应用实例：案例：显示当前工作目录的绝对路径

9.4.2ls 指令

基本语法: `ls [选项] [目录或是文件]`

✓ 常用选项

-a : 显示当前目录所有的文件和目录, 包括隐藏的。

-l : 以列表的方式显示信息

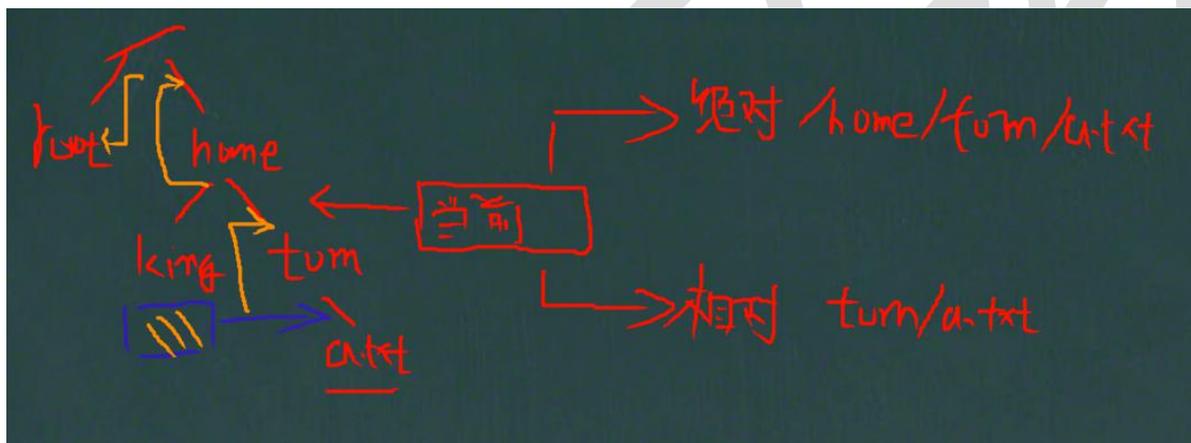
✓ 应用实例

案例: 查看当前目录的所有内容信息

9.4.3cd 指令

基本语法: `cd [参数]` (功能描述: 切换到指定目录)

理解: 绝对路径和相对路径



`cd ~` 或者 `cd` : 回到自己的家目录, 比如 你是 root , `cd ~` 到 `/root`

`cd ..` 回到当前目录的上一级目录

✓ 应用实例

案例 1: 使用绝对路径切换到 root 目录 , `cd /root`

案例 2: 使用相对路径到/root 目录, 比如在 `/home/tom` , `cd ../../root`

案例 3: 表示回到当前目录的上一级目录 , `cd ..`

案例 4: 回到家目录 , `cd ~`

9.4.4mkdir 指令

- `mkdir` 指令用于创建目录
 - 基本语法：`mkdir [选项] 要创建的目录`
- 常用选项
 - `-p` : 创建多级目录
- 应用实例

案例 1: 创建一个目录 `/home/dog`

```
mkdir /home/dog
```

案例 2: 创建多级目录 `/home/animal/tiger`

```
mkdir -p /home/animal/tiger
```

`rmdir` 指令

9.4.5 `rmdir` 指令删除空目录

- 基本语法
 - `rmdir [选项] 要删除的空目录`
- 应用实例
 - 案例: 删除一个目录 `/home/dog`
- 使用细节
 - `rmdir` 删除的是空目录, 如果目录下有内容时无法删除的。
 - 提示: 如果需要删除非空目录, 需要使用 `rm -rf` 要删除的目录
 - 比如: `rm -rf /home/animal`

9.4.6 `touch` 指令

- `touch` 指令创建空文件
- 基本语法

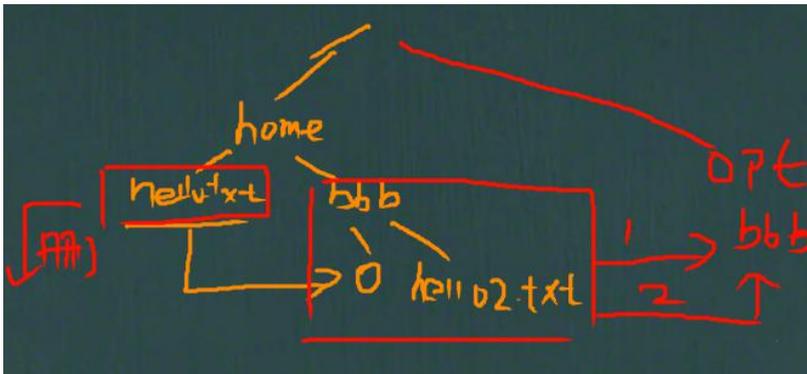
touch 文件名称

➤ 应用实例

案例: 在/home 目录下, 创建一个空文件 hello.txt

9.4.7 cp 指令

cp 指令拷贝文件到指定目录



➤ 基本语法

cp [选项] source dest

➤ 常用选项

-r : 递归复制整个文件夹

➤ 应用实例

案例 1: 将 /home/hello.txt 拷贝到 /home/bbb 目录下

```
cp hello.txt /home/bbb
```

案例 2: 递归复制整个文件夹, 举例, 比如将 /home/bbb 整个目录, 拷贝到 /opt

```
cp -r /home/bbb /opt
```

➤ 使用细节

强制覆盖不提示的方法: \cp , \cp -r /home/bbb /opt

9.4.8 rm 指令

说明: rm 指令移除文件或目录

✓ 基本语法

`rm` [选项] 要删除的文件或目录

✓ 常用选项

`-r` : 递归删除整个文件夹

`-f` : 强制删除不提示

✓ 应用实例

案例 1: 将 `/home/hello.txt` 删除 , `rm /home/hello.txt`

案例 2: 递归删除整个文件夹 `/home/bbb` , `rm -rf /home/bbb` [删除整个文件夹, 不提示]

✓ 使用细节

强制删除不提示的方法: 带上 `-f` 参数即可

9.4.9 mv 指令

`mv` 移动文件与目录或重命名

✓ 基本语法

`mv oldNameFile newNameFile` (功能描述: 重命名)

`mv /temp/movefile /targetFolder` (功能描述: 移动文件)

✓ 应用实例

案例 1: 将 `/home/cat.txt` 文件 重新命名为 `pig.txt`

案例 2: 将 `/home/pig.txt` 文件 移动到 `/root` 目录下

案例 3: 移动整个目录 , 比如将 `/opt/bbb` 移动到 `/home` 下 `mv /opt/bbb /home/`
试试.

9.4.10 cat 指令

`cat` 查看文件内容

✓ 基本语法

cat [选项] 要查看的文件

✓ 常用选项

-n : 显示行号

✓ 应用实例

案例 1: /etc/profile 文件内容, 并显示行号

✓ 使用细节

cat 只能浏览文件, 而不能修改文件, 为了浏览方便, 一般会带上管道命令 | more

cat -n /etc/profile | more [进行交互]

9.4.11 more 指令

more 指令是一个基于 VI 编辑器的文本过滤器, 它以全屏幕的方式按页显示文本文件的内容。more 指令中内置了若干快捷键(交互的指令), 详见操作说明

✓ 基本语法

more 要查看的文件

操作说明, 如图

操作	功能说明
空白键 (space)	代表向下翻一页;
Enter	代表向下翻『一行』;
q	代表立刻离开 more , 不再显示该文件内容。
Ctrl+F	向下滚动一屏
Ctrl+B	返回上一屏
=	输出当前行的行号
:f	输出文件名和当前行的行号

✓ 应用实例

案例: 采用 more 查看文件

/etc/profile

9.4.12 less 指令

less 指令用来分屏查看文件内容，它的功能与 more 指令类似，但是比 more 指令更加强大，支持各种显示终端。less 指令在显示文件内容时，并不是一次将整个文件加载之后才显示，而是根据显示需要加载内容，对于显示大型文件具有较高的效率。

✓ 基本语法

less 要查看的文件

操作说明

操作	功能说明
空白键	向下翻动一页；
[pagedown]	向下翻动一页
[pageup]	向上翻动一页；
/字串	向下搜寻『字串』的功能；n：向下查找；N：向上查找；
?字串	向上搜寻『字串』的功能；n：向上查找；N：向下查找；
q	离开 less 这个程序；

✓ 应用实例

案例：采用 less 查看一个大文件文件 /opt/杂文.txt

```
less /opt/杂文.txt
```

9.4.13 echo 指令

echo 输出内容到控制台

✓ 基本语法

echo [选项] [输出内容]

✓ 应用实例

案例：使用 echo 指令输出环境变量，比如输出 \$PATH \$HOSTNAME, echo \$HOSTNAME

案例: 使用 echo 指令输出 hello,world!

9.4.14 head 指令

head 用于显示文件的开头部分内容, 默认情况下 head 指令显示文件的前 10 行内容

✓ 基本语法

head 文件 (功能描述: 查看文件头 10 行内容)

head -n 5 文件 (功能描述: 查看文件头 5 行内容, 5 可以是任意行数)

✓ 应用实例

案例: 查看/etc/profile 的前面 5 行代码

```
head -n 5 /etc/profile
```

9.4.15 tail 指令

tail 用于输出文件中尾部的内容, 默认情况下 tail 指令显示文件的前 10 行内容。

✓ 基本语法

1) tail 文件 (功能描述: 查看文件尾 10 行内容)

2) tail -n 5 文件 (功能描述: 查看文件尾 5 行内容, 5 可以是任意行数)

3) tail -f 文件 (功能描述: 实时追踪该文档的所有更新)

✓ 应用实例

案例 1: 查看/etc/profile 最后 5 行的代码

```
tail -n 5 /etc/profile
```

案例 2: 实时监控 mydate.txt, 看看到文件有变化时, 是否看到, 实时的追加 hello,world

```
tail -f /home/mydate.txt
```

9.4.16 > 指令 和 >> 指令

> 输出重定向和 >> 追加

✓ 基本语法

1) `ls -l > 文件` (功能描述: 列表的内容写入文件 a.txt 中 (覆盖写))

2) `ls -al >> 文件` (功能描述: 列表的内容追加到文件 aa.txt 的末尾)

3) `cat 文件 1 > 文件 2` (功能描述: 将文件 1 的内容覆盖到文件 2)

4) `echo "内容" >> 文件` (追加)

✓ 应用实例

案例 1: 将 /home 目录下的文件列表 写入到 /home/info.txt 中, 覆盖写入

`ls -l /home > /home/info.txt` [如果 info.txt 没有, 则会创建]

案例 2: 将当前日历信息 追加到 /home/mycal 文件中

指令为: `cal >> /home/mycal`

9.4.17 ln 指令

软链接也称为符号链接, 类似于 windows 里的快捷方式, 主要存放了链接其他文件的路径

✓ 基本语法

`ln -s [原文件或目录] [软链接名]` (功能描述: 给原文件创建一个软链接)

✓ 应用实例

案例 1: 在/home 目录下创建一个软连接 myroot, 连接到 /root 目录

`ln -s /root /home/myroot`

案例 2: 删除软连接 myroot

`rm /home/myroot`

✓ 细节说明

当我们使用 `pwd` 指令查看目录时, 仍然看到的是软链接所在目录。

9.4.18 history 指令

查看已经执行过历史命令,也可以执行历史指令

✓ 基本语法

history (功能描述: 查看已经执行过历史命令)

✓ 应用实例

案例 1: 显示所有的历史命令

```
history
```

案例 2: 显示最近使用过的 10 个指令。

```
history 10
```

案例 3: 执行历史编号为 5 的指令

```
!5
```

9.5 时间日期类

9.5.1 date 指令-显示当前日期

✓ 基本语法

1) date (功能描述: 显示当前时间)

2) date +%Y (功能描述: 显示当前年份)

3) date +%m (功能描述: 显示当前月份)

4) date +%d (功能描述: 显示当前是哪一天)

5) date "+%Y-%m-%d %H:%M:%S" (功能描述: 显示年月日时分秒)

✓ 应用实例

案例 1: 显示当前时间信息

```
date
```

案例 2: 显示当前时间年月日

```
date "+%Y-%m-%d"
```

案例 3: 显示当前时间年月日时分秒

```
date "+%Y-%m-%d %H:%M:%S"
```

9.5.2 date 指令-设置日期

- ✓ 基本语法

date -s 字符串时间

- ✓ 应用实例

案例 1: 设置系统当前时间，比如设置成 2020-11-03 20:02:10

```
date -s "2020-11-03 20:02:10"
```

9.5.3 cal 指令

查看日历指令 cal

- ✓ 基本语法

cal [选项] (功能描述: 不加选项, 显示本月日历)

- ✓ 应用实例

案例 1: 显示当前日历 cal

案例 2: 显示 2020 年日历 : cal 2020

9.6 搜索查找类

9.6.1 find 指令

find 指令将从指定目录向下递归地遍历其各个子目录, 将满足条件的文件或者目录显示在终端。

- ✓ 基本语法

```
find [搜索范围] [选项]
```

- ✓ 选项说明

选项	功能
-name<查询方式>	按照指定的文件名查找模式查找文件
-user<用户名>	查找属于指定用户名所有文件
-size<文件大小>	按照指定的文件大小查找文件。

✓ 应用实例

案例 1: 按文件名: 根据名称查找/home 目录下的 hello.txt 文件

```
find /home -name hello.txt
```

案例 2: 按拥有者: 查找/opt 目录下, 用户名称为 nobody 的文件

```
find /opt -user nobody
```

案例 3: 查找整个 linux 系统下大于 200M 的文件 (+n 大于 -n 小于 n 等于, 单位有 k,M,G)

```
find / -size +200M
```

9.6.2 locate 指令

locate 指令可以快速定位文件路径。locate 指令利用事先建立的系统中所有文件名称及路径的 locate 数据库实现快速定位给定的文件。Locate 指令无需遍历整个文件系统, 查询速度较快。为了保证查询结果的准确度, 管理员必须定期更新 locate 时刻

✓ 基本语法

locate 搜索文件

✓ 特别说明

由于 locate 指令基于数据库进行查询, 所以第一次运行前, 必须使用 updatedb 指令创建 locate 数据库。

✓ 应用实例

案例 1: 请使用 locate 指令快速定位 hello.txt 文件所在目录

which 指令, 可以查看某个指令在哪个目录下, 比如 ls 指令在哪个目录

```
which ls
```

9.6.3 grep 指令和 管道符号 |

grep 过滤查找，管道符，“|”，表示将前一个命令的处理结果输出传递给后面的命令处理。



✓ 基本语法

grep [选项] 查找内容 源文件

✓ 常用选项

选项	功能
-n	显示匹配行及行号。
-i	忽略字母大小写

✓ 应用实例

案例 1: 请在 hello.txt 文件中，查找 "yes" 所在行，并且显示行号

写法 1: cat /home/hello.txt | grep "yes"

写法 2: grep -n "yes" /home/hello.txt

9.7 压缩和解压类

9.7.1 gzip/gunzip 指令

gzip 用于压缩文件，gunzip 用于解压的

✓ 基本语法

gzip 文件 （功能描述：压缩文件，只能将文件压缩为*.gz 文件）

gunzip 文件.gz （功能描述：解压缩文件命令）

✓ 应用实例

案例 1: gzip 压缩，将 /home 下的 hello.txt 文件进行压缩

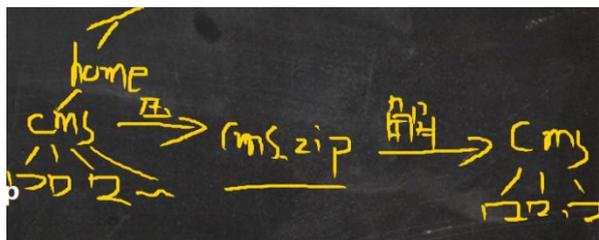
```
gzip /home/hello.txt
```

案例 2: gunzip 压缩, 将 /home 下的 hello.txt.gz 文件进行解压缩

```
gunzip /home/hello.txt.gz
```

9.7.2 zip/unzip 指令

zip 用于压缩文件, unzip 用于解压的, 这个在项目打包发布中很有用的



✓ 基本语法

zip [选项] XXX.zip 将要压缩的内容 (功能描述: 压缩文件和目录的命令)

unzip [选项] XXX.zip (功能描述: 解压缩文件)

✓ zip 常用选项

-r: 递归压缩, 即压缩目录

✓ unzip 的常用选项

-d<目录> : 指定解压后文件的存放目录

✓ 应用实例

案例 1: 将 /home 下的 所有文件/文件夹进行压缩成 myhome.zip

```
zip -r myhome.zip /home/ [将 home 目录及其包含的文件和子文件夹都压缩]
```

案例 2: 将 myhome.zip 解压到 /opt/tmp 目录下

```
mkdir /opt/tmp
```

```
unzip -d /opt/tmp /home/myhome.zip
```

9.7.3 tar 指令

tar 指令 是打包指令, 最后打包后的文件是 .tar.gz 的文件。

✓ 基本语法

tar [选项] XXX.tar.gz 打包的内容 (功能描述: 打包目录, 压缩后的文件格式.tar.gz)

✓ 选项说明

选项	功能
-c	产生.tar打包文件
-v	显示详细信息
-f	指定压缩后的文件名
-z	打包同时压缩
-x	解包.tar文件

✓ 应用实例

案例 1: 压缩多个文件, 将 /home/pig.txt 和 /home/cat.txt 压缩成 pc.tar.gz

```
tar -zcvf pc.tar.gz /home/pig.txt /home/cat.txt
```

案例 2: 将/home 的文件夹 压缩成 myhome.tar.gz

```
tar -zcvf myhome.tar.gz /home/
```

案例 3: 将 pc.tar.gz 解压到当前目录

```
tar -zxvf pc.tar.gz
```

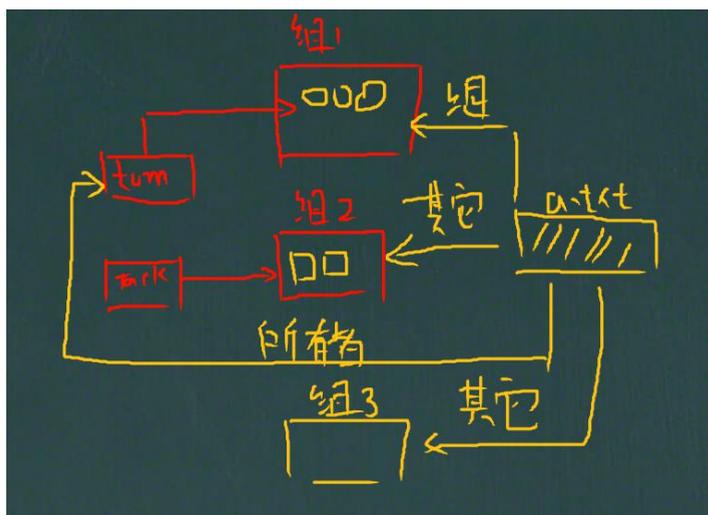
案例 4: 将 myhome.tar.gz 解压到 /opt/tmp2 目录下 (1) mkdir /opt/tmp2 (2) tar -zxvf /home/myhome.tar.gz -C /opt/tmp2

第 10 章 Linux 实操篇-组管理和权限管理

10.1 Linux 组基本介绍

在 linux 中的每个用户必须属于一个组，不能独立于组外。在 linux 中每个文件有所有者、所在组、其它组的概念。

- 1) 所有者
- 2) 所在组
- 3) 其它组
- 4) 改变用户所在的组



10.2 文件/目录 所有者

一般为文件的创建者,谁创建了该文件，就自然的成为该文件的所有者。

10.2.1 查看文件的所有者

指令: `ls -ahl`

应用实例

```
-rw-r--r--. 1 root root 38 11月 3 20:44 hello.txt
```

10.2.2 修改文件所有者

指令: `chown 用户名 文件名`

应用案例

要求: 使用 `root` 创建一个文件 `apple.txt` , 然后将其所有者修改成 `tom`

```
chown tom apple.txt
```

10.3 组的创建

10.3.1 基本指令

```
groupadd 组名
```

10.3.2 应用实例

- ✓ 创建一个组, `monster`

```
groupadd monster
```

- ✓ 创建一个用户 `fox` , 并放入到 `monster` 组中

```
useradd -g monster fox
```

10.4 文件/目录 所在组

当某个用户创建了一个文件后, 这个文件的所在组就是该用户所在的组(默认)。

10.4.1 查看文件/目录所在组

基本指令

```
ls -ahl
```

应用实例, 使用 `fox` 来创建一个文件, 看看该文件属于哪个组?

```
-rw-r--r--. 1 fox monster 0 11 月 5 12:50 ok.txt
```

10.4.2 修改文件/目录所在的组

✓ 基本指令

```
chgrp 组名 文件名
```

✓ 应用实例

使用 root 用户创建文件 orange.txt,看看当前这个文件属于哪个组,然后将这个文件所在组,修改到 fruit 组。

1. groupadd fruit

2. touch orange.txt

3. 看看当前这个文件属于哪个组 -> root 组

4. chgrp fruit orange.txt

10.5 其它组

除文件的所有者和所在组的用户外,系统的其它用户都是文件的其它组

10.6 改变用户所在组

在添加用户时,可以指定将该用户添加到哪个组中,同样的用 root 的管理权限可以改变某个用户所在的组。

10.6.1 改变用户所在组

```
usermod -g 新组名 用户名
```

```
usermod -d 目录名 用户名 改变该用户登陆的初始目录。特别说明: 用户需要有进入到新目录的权限。
```

10.6.2 应用实例

将 zwj 这个用户从原来所在组,修改到 wudang 组

```
usermod -g wudang zwj
```

10.7 权限的基本介绍

ls -l 中显示的内容如下:

```
-rwxrw-r-- 1 root root 1213 Feb 2 09:39 abc
```

0-9 位说明

第 0 位确定文件类型(d, -, l, c, b)

l 是链接, 相当于 windows 的快捷方式

d 是目录, 相当于 windows 的文件夹

c 是字符设备文件, 鼠标, 键盘

b 是块设备, 比如硬盘

第 1-3 位确定所有者(该文件的所有者)拥有该文件的权限。---User

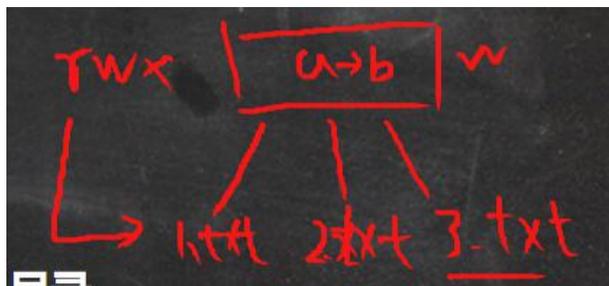
第 4-6 位确定所属组(同用户组的)拥有该文件的权限, ---Group

第 7-9 位确定其他用户拥有该文件的权限 ---Other

10.8 rwx 权限详解, 难点

10.8.1 rwx 作用到文件

- 1) [r]代表可读(read): 可以读取,查看
- 2) [w]代表可写(write): 可以修改,但是不代表可以删除该文件,删除一个文件的前提条件是对该文件所在的目录有写权限, 才能删除该文件.
- 3) [x]代表可执行(execute):可以被执行



10.8.2 rwx 作用到目录

- 1) [r]代表可读(read): 可以读取, ls 查看目录内容
- 2) [w]代表可写(write): 可以修改, 对目录内创建+删除+重命名目录
- 3) [x]代表可执行(execute): 可以进入该目录

10.9 文件及目录权限实际案例

10.9.1 ls -l 中显示的内容如下:

```
-rwxrw-r-- 1 root root 1213 Feb 2 09:39 abc
```

- ✓ 10 个字符确定不同用户能对文件干什么
 - 第一个字符代表文件类型: -l d c b
 - 其余字符每 3 个一组(rwx) 读(r) 写(w) 执行(x)
 - 第一组 rwx: 文件拥有者的权限是读、写和执行
 - 第二组 rw-: 与文件拥有者同一组的用户的权限是读、写但不能执行
 - 第三组 r--: 不与文件拥有者同组的其他用户的权限是读不能写和执行
- ✓ 可用数字表示为: r=4,w=2,x=1 因此 rwx=4+2+1=7, 数字可以进行组合
- ✓ 其它说明
 - 1 文件: 硬连接数或 目录: 子目录数
 - root 用户
 - root 组

1213	文件大小(字节), 如果是文件夹, 显示 4096 字节
Feb 2 09:39	最后修改日期
abc	文件名

10.10 修改权限-chmod

10.10.1 基本说明:

通过 chmod 指令, 可以修改文件或者目录的权限。

10.10.2 第一种方式: +、-、= 变更权限

✓ u:所有者 g:所有组 o:其他人 a:所有人(u、g、o 的总和)

1) chmod u=rwx,g=rx,o=x 文件/目录名

2) chmod o+w 文件/目录名

3) chmod a-x 文件/目录名

✓ 案例演示

1) 给 abc 文件 的所有者读写执行的权限, 给所在组读执行权限, 给其它组读执行权限。

```
chmod u=rwx,g=rx,o=rx abc
```

2) 给 abc 文件的所有者除去执行的权限, 增加组写的权限

```
chmod u-x,g+w abc
```

3) 给 abc 文件的所有用户添加读的权限

```
chmod a+r abc
```

10.10.3 第二种方式: 通过数字变更权限

r=4 w=2 x=1 rwx=4+2+1=7

```
chmod u=rwx,g=rx,o=x 文件目录名
```

相当于 chmod 751 文件/目录名

✓ 案例演示

要求: 将 /home/abc.txt 文件的权限修改成 rwxr-xr-x, 使用给数字的方式实现:

```
chmod 755 /home/abc.txt
```

10.11 修改文件所有者-chown

10.11.1 基本介绍

chown newowner 文件/目录 改变所有者

chown newowner:newgroup 文件/目录 改变所有者和所在组

-R 如果是目录 则使其下所有子文件或目录递归生效

✓ 案例演示

- 1) 请将 /home/abc.txt 文件的所有者修改成 tom

```
chown tom /home/abc.txt
```

- 2) 请将 /home/test 目录下所有的文件和目录的所有者都修改成 tom

```
chown -R tom /home/test
```

10.12 修改文件/目录所在组-chgrp

10.12.1 基本介绍

chgrp newgroup 文件/目录 【改变所在组】

10.12.2 案例演示

请将 /home/abc.txt 文件的所在组修改成 shaolin (少林)

```
groupadd shaolin
```

```
chgrp shaolin /home/abc.txt
```

请将 /home/test 目录下所有的文件和目录的所在组都修改成 shaolin(少林)

```
chgrp -R shaolin /home/test
```

10.13最佳实践-警察和土匪游戏

police , bandit

jack, jerry: 警察

xh, xq: 土匪

1) 创建组 `groupadd police ; groupadd bandit`

2) 创建用户

`useradd -g police jack ; useradd -g police jerry`

`useradd -g bandit xh; useradd -g bandit xq`

3) jack 创建一个文件，自己可以读 r 写 w，本组人可以读，其它组没人任何权限

首先 jack 登录； `vim jack.txt ; chmod 640 jack.txt`

4) jack 修改该文件，让其它组人可以读，本组人可以读写

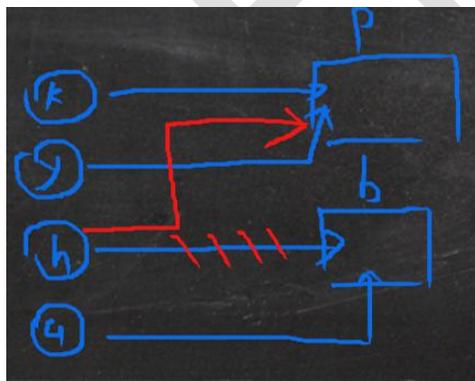
`chmod o=r,g=r jack.txt`

5) xh 投靠 警察，看看是否可以读写。

`usermod -g police xh`

6) 测试，看看 xh 是否可以读写，xq 是否可以，小结论，就是如果要对目录内的文件进行操作，需要要有对该目录的相应权限

7) 示意图



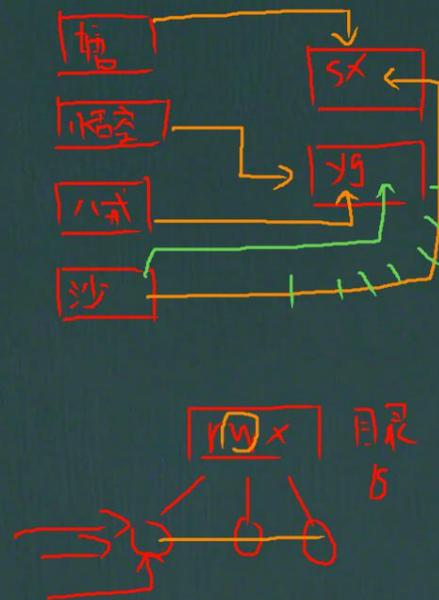
10.14课后练习

练习文件权限管理[课堂练习], 完成如下操作.

- 1) 建立两个组 (神仙(sx),妖怪(yg))
- 2) 建立四个用户(唐僧,悟空, 八戒, 沙僧)
- 3) 设置密码
- 4) 把悟空, 八戒放入妖怪 唐僧 沙僧 在神仙
- 5) 用悟空建立一个文件 (monkey.java 该文件要输出 i am monkey)
- 6) 给八戒一个可以 r w 的权限
- 7) 八戒修改 monkey.java 加入一句话(i am pig)
- 8) 唐僧 沙僧 对该文件没有权限
- 9) 把 沙僧 放入妖怪组
- 10) 让沙僧 修改 该文件 monkey, 加入一句话 ("我是沙僧, 我是妖怪!");
- 11) 对文件夹 rwx 的细节讨论和测试!!!
 - x: 表示可以进入到该目录, 比如 cd
 - r: 表示可以 ls, 将目录的内容显示
 - w: 表示可以在该目录, 删除或者创建文件
- 12) 示意图

● 练习文件权限管理[课堂练习], 完成如下操作.

1. 建立两个组 (神仙(sx),妖怪(yg))
2. 建立四个用户(唐僧,悟空,八戒,沙僧)
3. 设置密码
4. 把悟空,八戒放入妖怪 唐僧 沙僧 在神仙
5. 用悟空建立一个文件 (monkey.java 该文件要输出 i am monkey)
6. 给八戒一个可以r w 的权限
7. 八戒修改monkey.java 加入一句话(i am pig)
8. 唐僧 沙僧 对该文件没有权限
9. 把 沙僧 放入妖怪组
10. 让沙僧 修改 该文件 monkey, 加入一句话 ("我是沙僧,我是妖怪!");
11. 对文件夹rwx 的细节讨论和测试!!!



10.15课堂练习 2, 完成如下操作

- 1) 用 root 登录, 建立用户 mycentos,自己设定密码
- 2) 用 mycentos 登录, 在主目录下建立目录 test/t11/t1
- 3) 在 t1 中建立一个文本文件 aa,用 vi 编辑其内容为 ls -al
- 4) 改变 aa 的权限为可执行文件[可以将当前日期追加到一个文件],运行该文件./aa
- 5) 删除新建立的目录 test/t11/t1
- 6) 删除用户 mycentos 及其主目录中的内容
- 7) 将 linux 设置成进入到图形界面的
- 8) 重新启动 linux 或关机

第 11 章 Linux 实操篇-定时任务调度

11.1 crond 任务调度

crontab 进行 定时任务的设置

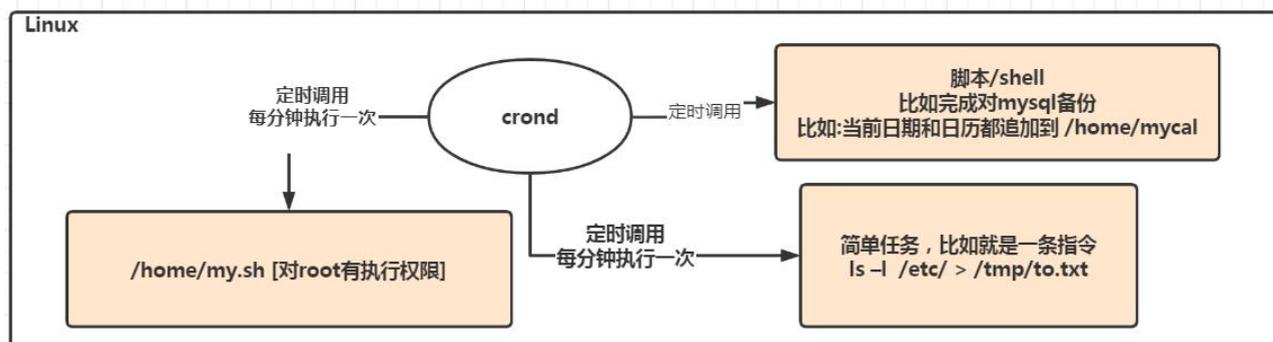
11.1.1 概述

任务调度：是指系统在某个时间执行的特定的命令或程序。

任务调度分类：1.系统工作：有些重要的工作必须周而复始地执行。如病毒扫描等

个别用户工作：个别用户可能希望执行某些程序，比如对 mysql 数据库的备份。

示意图



11.1.2 基本语法

crontab [选项]

11.1.3 常用选项

-e	编辑crontab定时任务
-l	查询crontab任务
-r	删除当前用户所有的crontab任务

11.1.4 快速入门

设置任务调度文件：/etc/crontab

设置个人任务调度。执行 `crontab -e` 命令。

接着输入任务到调度文件

如：`*/* * * * * ls -l /etc/ > /tmp/to.txt`

意思说每小时的每分钟执行 `ls -l /etc/ > /tmp/to.txt` 命令

✓ 参数细节说明

5 个占位符的说明

项目	含义	范围
第一个 “*”	一小时当中的第几分钟	0-59
第二个 “*”	一天当中的第几小时	0-23
第三个 “*”	一个月当中的第几天	1-31
第四个 “*”	一年当中的第几月	1-12
第五个 “*”	一周当中的星期几	0-7 (0和7都代表星期日)

✓ 特殊符号的说明

特殊符号	含义
*	代表任何时间。比如第一个 “*” 就代表一小时中每分钟都执行一次的意思。
,	代表不连续的时间。比如 “0 8,12,16 * * * * 命令”，就代表在每天的8点0分，12点0分，16点0分都执行一次命令
-	代表连续的时间范围。比如 “0 5 * * 1-6 命令”，代表在周一到周六的凌晨5点0分执行命令
/n	代表每隔多久执行一次。比如 “/10 * * * * 命令”，代表每隔10分钟就执行一遍命令

✓ 特殊时间执行案例

时间	含义
45 22 * * * 命令	在22点45分执行命令
0 17 * * 1 命令	每周1 的17点0分执行命令
0 5 1,15 * * 命令	每月1号和15号的凌晨5点0分执行命令
40 4 * * 1-5 命令	每周一到周五的凌晨4点40分执行命令
*/10 4 * * * 命令	每天的凌晨4点, 每隔10分钟执行一次命令
0 0 1,15 * 1 命令	每月1号和15号, 每周1的0点0分都会执行命令。注意: 星期几和几号最好不要同时出现, 因为他们定义的都是天。非常容易让管理员混乱。

11.1.5 应用实例

- ✓ 案例 1: 每隔 1 分钟, 就将当前的日期信息, 追加到 /tmp/mydate 文件中
*/1 * * * * date >> /tmp/mydate
- ✓ 案例 2: 每隔 1 分钟, 将当前日期和日历都追加到 /home/mycal 文件中
步骤:
 (1) vim /home/my.sh 写入内容 date >> /home/mycal 和 cal >> /home/mycal
 (2) 给 my.sh 增加执行权限, chmod u+x /home/my.sh
 (3) crontab -e 增加 */1 * * * * /home/my.sh
- ✓ 案例 3: 每天凌晨 2:00 将 mysql 数据库 testdb , 备份到文件中。提示: 指令为
mysqldump -u root -p 密码 数据库 > /home/db.bak
步骤(1) crontab -e
步骤(2) 0 2 * * * mysqldump -u root -p 密码 testdb > /home/db.bak

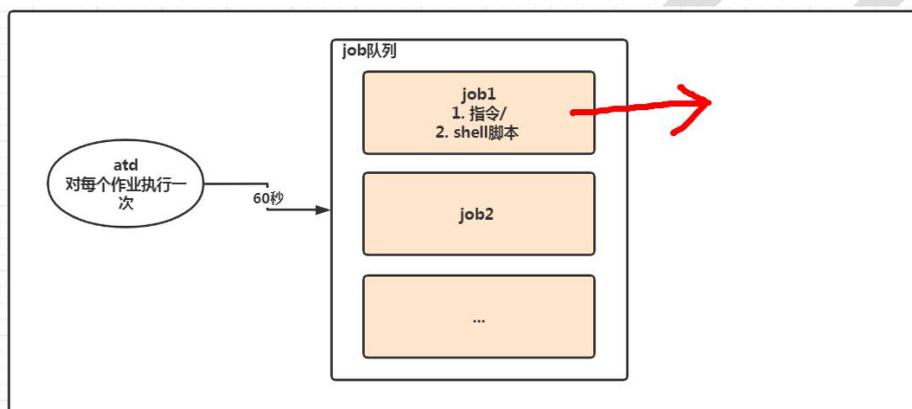
11.1.6 crond 相关指令

- conrtab -r: 终止任务调度。
- crontab -l: 列出当前有那些任务调度
- service crond restart [重启任务调度]

11.2 at 定时任务

11.2.1 基本介绍

- 1) at 命令是一次性定时计划任务，at 的守护进程 atd 会以后台模式运行，检查作业队列来运行。
- 2) 默认情况下，atd 守护进程每 60 秒检查作业队列，有作业时，会检查作业运行时间，如果时间与当前时间匹配，则运行此作业。
- 3) at 命令是一次性定时计划任务，执行完一个任务后不再执行此任务了
- 4) 在使用 at 命令的时候，一定要保证 atd 进程的启动，可以使用相关指令来查看
`ps -ef | grep atd` //可以检测 atd 是否在运行
- 5) 画一个示意图



11.2.2 at 命令格式

at [选项] [时间]

Ctrl + D 结束 at 命令的输入，输出两次

11.2.3 at 命令选项

选项	含义
-m	当指定的任务被完成后，将给用户发送邮件，即使没有标准输出
-l	atq的别名
-d	atrm的别名
-v	显示任务将被执行的时间
-c	打印任务的内容到标准输出
-V	显示版本信息
-q <队列>	使用指定的队列
-f <文件>	从指定文件读入任务而不是从标准输入读入
-t <时间参数>	以时间参数的形式提交要运行的任务

11.2.4 at 时间定义

✓ at 指定时间的方法：

- 1) 接受在当天的 hh:mm（小时:分钟）式的时间指定。假如该时间已过去，那么就放在第二天执行。例如：04:00
- 2) 使用 midnight（深夜），noon（中午），teatime（饮茶时间，一般是下午 4 点）等比较模糊的词语来指定时间。
- 3) 采用 12 小时计时制，即在时间后面加上 AM（上午）或 PM（下午）来说明是上午还是下午。例如：12pm
- 4) 指定命令执行的具体日期，指定格式为 month day（月 日）或 mm/dd/yy（月/日/年）或 dd.mm.yy（日.月.年），指定的日期必须跟在指定时间的后面。例如：04:00 2021-03-1
- 5) 使用相对计时法。指定格式为: now + count time-units，now 就是当前时间，time-units 是时间单位，这里能够是 minutes（分钟）、hours（小时）、days（天）、weeks（星期）。count 是时间的数量，几天，几小时。例如：now + 5 minutes
- 6) 直接使用 today（今天）、tomorrow（明天）来指定完成命令的时间。

11.2.5 应用实例

案例 1: 2 天后的下午 5 点执行 /bin/ls /home

```
[root@hspEdu01 ~]# at 5pm + 2 days
at> /bin/ls /home<EOT>
job 5 at Sat Nov 7 17:00:00 2020
```

案例 2: atq 命令来查看系统中没有执行的工作任务

案例 3: 明天 17 点钟, 输出时间到指定文件内 比如 /root/date100.log

```
[root@hspEdu01 ~]# at 5pm tomorrow
at> date > /root/date100.log<EOT>
job 6 at Fri Nov  6 17:00:00 2020
[root@hspEdu01 ~]# atq
5      Sat Nov  7 17:00:00 2020 a root
6      Fri Nov  6 17:00:00 2020 a root
```

案例 4: 2 分钟后, 输出时间到指定文件内 比如 /root/date200.log

```
[root@hspEdu01 ~]# at now + 2 minutes
at> date > /root/date200.log<EOT>
job 7 at Thu Nov  5 22:15:00 2020
```

案例 5: 删除已经设置的任务 ,atrm 编号

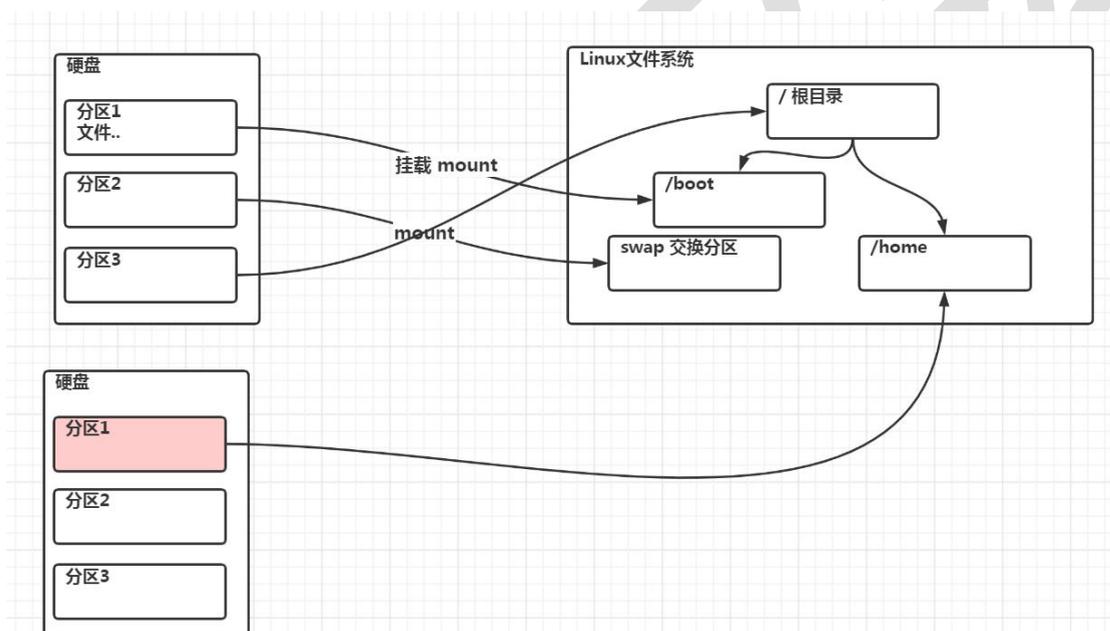
atrm 4 //表示将 job 队列, 编号为 4 的 job 删除.

第 12 章 Linux 实操篇-Linux 磁盘分区、挂载

12.1 Linux 分区

12.1.1 原理介绍

- 1) Linux 来说无论有几个分区，分给哪一目录使用，它归根结底就只有一个根目录，一个独立且唯一的文件结构，Linux 中每个分区都是用来组成整个文件系统的一部分。
- 2) Linux 采用了一种叫“载入”的处理方法，它的整个文件系统中包含了一整套的文件和目录，且将一个分区和一个目录联系起来。这时要载入的一个分区将使它的存储空间在一个目录下获得。
- 3) 示意图



12.1.2 硬盘说明

- 1) Linux 硬盘分 IDE 硬盘和 SCSI 硬盘，目前基本上是 SCSI 硬盘
- 2) 对于 IDE 硬盘，驱动器标识符为“hdx~”，其中“hd”表明分区所在设备的类型，这里是指 IDE 硬盘了。“x”为盘号（a 为基本盘，b 为基本从属盘，c 为辅助主盘，d 为辅助从属盘），“~”代表分区，前四个分区用数字 1 到 4 表示，它们是

主分区或扩展分区，从 5 开始就是逻辑分区。例，hda3 表示为第一个 IDE 硬盘上的第三个主分区或扩展分区，hdb2 表示为第二个 IDE 硬盘上的第二个主分区或扩展分区。

- 3) 对于 SCSI 硬盘则标识为“sdx~”，SCSI 硬盘是用“sd”来表示分区所在设备的类型的，其余则和 IDE 硬盘的表示方法一样

```
[root@hspEdu01 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda   8:0    0  20G  0 disk
├─sda1 8:1    0   1G  0 part /boot
├─sda2 8:2    0   2G  0 part [SWAP]
└─sda3 8:3    0  17G  0 part /
sr0   11:0   1  4.3G  0 rom
```

12.1.3 查看所有设备挂载情况

- ✓ 命令：lsblk 或者 lsblk -f

```
[root@hspedu01 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda   8:0    0  20G  0 disk
├─sda1 8:1    0   1G  0 part /boot
├─sda2 8:2    0   2G  0 part [SWAP]
└─sda3 8:3    0  17G  0 part /
sr0   11:0   1 1024M  0 rom
```

```
[root@hspEdu01 ~]# lsblk -f
NAME FSTYPE LABEL UUID MOUNTPOINT
sda   ext4 分区情况 df4d64bd-981e-41eb-8f73-16ac20f0371e /boot
├─sda1 ext4 494c7f94-5656-45a3-b990-008c2b0e39b7 [SWAP]
├─sda2 swap 12ae1cc1-09e0-42ce-b659-063df3e7c941 /
└─sda3 ext4
sr0   iso9660 CentOS 7 x86_64 2018-11-25-23-54-16-00
```

分区情况 → MOUNTPOINT 挂

12.2 挂载的经典案例

12.2.1 说明：

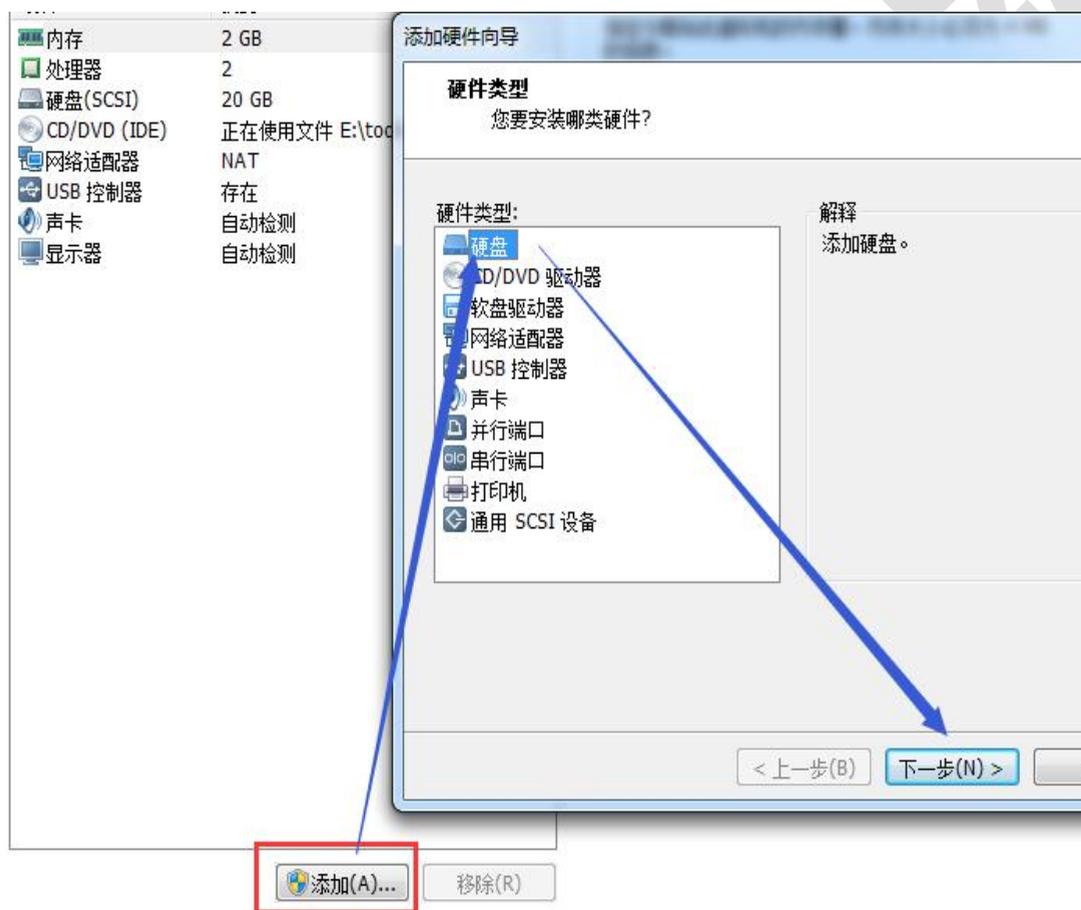
下面我们以增加一块硬盘为例来熟悉下磁盘的相关指令和深入理解磁盘分区、挂载、卸载的概念。

12.2.2 如何增加一块硬盘

- 1) 虚拟机添加硬盘
- 2) 分区
- 3) 格式化
- 4) 挂载
- 5) 设置可以自动挂载

12.2.3 虚拟机增加硬盘步骤 1

在【虚拟机】菜单中，选择【设置】，然后在设备列表里添加硬盘，然后一路【下一步】，中间只有选择磁盘大小的地方需要修改，直到完成。然后重启系统（才能识别）！



12.2.4 虚拟机增加硬盘步骤 2

✓ 分区命令 `fdisk /dev/sdb`

开始对/sdb 分区

m 显示命令列表

p 显示磁盘分区 同 `fdisk -l`

n 新增分区

d 删除分区

w 写入并退出

✓ 说明：开始分区后输入 n，新增分区，然后选择 p，分区类型为主分区。两次回车默认剩余全部空间。最后输入 w 写入分区并退出，若不保存退出输入 q。

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Selected partition 4
First cylinder (2233-2610, default 2233): 回车, 默认全部剩余空间
Using default value 2233
Last cylinder, +cylinders or +size{K,M,G} (2233-2610, default 2610): 同上
Using default value 2610
```

12.2.5 虚拟机增加硬盘步骤 3

✓ 格式化磁盘

分区命令:`mkfs -t ext4 /dev/sdb1`

其中 ext4 是分区类型

12.2.6 虚拟机增加硬盘步骤 4

✓ 挂载: 将一个分区与一个目录联系起来,

`mount 设备名称 挂载目录`

例如: `mount /dev/sdb1 /newdisk`

✓ `umount 设备名称 或者 挂载目录`

例如: `umount /dev/sdb1` 或者 `umount /newdisk`

- ✓ 老师注意: 用命令行挂载,重启后会失效

12.2.7 虚拟机增加硬盘步骤 5

永久挂载: 通过修改/etc/fstab 实现挂载

添加完成后 执行 `mount -a` 即刻生效

```
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/sdb1 /newdisk ext4 defaults 0 0
UUID=22565489-2961-4079-a831-9d8e940f2615 / ext4 defaults 0 0
UUID=27f8ef79-d6e0-42b2-95ca-20a340b46606 /boot ext4 defaults 1 2
UUID=506af6cc-fb37-4279-af1c-72b2a7adc992 swap swap defaults 0 0
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
~
~
```

12.3 磁盘情况查询

12.3.1 查询系统整体磁盘使用情况

- ✓ 基本语法

`df -h`

- ✓ 应用实例

查询系统整体磁盘使用情况

文件系统	容量	已用	可用	已用%	挂载点
/dev/sda3	17G	5.4G	11G	34%	/
devtmpfs	975M	0	975M	0%	/dev
tmpfs	991M	0	991M	0%	/dev/shm
tmpfs	991M	11M	981M	2%	/run
tmpfs	991M	0	991M	0%	/sys/fs/cgroup
/dev/sdb1	991M	2.6M	922M	1%	/newdisk
/dev/sda1	976M	134M	776M	15%	/boot
.host:/	119G	65G	54G	55%	/mnt/hgfs
tmpfs	199M	12K	199M	1%	/run/user/42
tmpfs	199M	0	199M	0%	/run/user/0

12.3.2 查询指定目录的磁盘占用情况

✓ 基本语法

`du -h`

✓ 查询指定目录的磁盘占用情况，默认为当前目录

`-s` 指定目录占用大小汇总

`-h` 带计量单位

`-a` 含文件

`--max-depth=1` 子目录深度

`-c` 列出明细的同时，增加汇总值

✓ 应用实例

查询 `/opt` 目录的磁盘占用情况，深度为 1

```
[root@hspEdu01 opt]# du -hac --max-depth=1 /opt
248K    /opt/tmp2
860K    /opt/Chrysanthemum.jpg
3.5M    /opt/杂文.txt
4.0K    /opt/home
54M     /opt/VMwareTools-10.3.10-13959562.tar.gz
4.0K    /opt/rh
228K    /opt/tmp
163M    /opt/vmware-tools-distrib
221M    /opt
221M    总用量
```

12.4 磁盘情况-工作实用指令

1) 统计/opt 文件夹下文件的个数

```
ls -l /opt | grep "^-" | wc -l
```

2) 统计/opt 文件夹下目录的个数

```
ls -l /opt | grep "^d" | wc -l
```

3) 统计/opt 文件夹下文件的个数，包括子文件夹里的

```
ls -lR /opt | grep "^-" | wc -l
```

- 4) 统计/opt 文件夹下目录的个数，包括子文件夹里的

```
ls -lR /opt | grep "^d" | wc -l
```

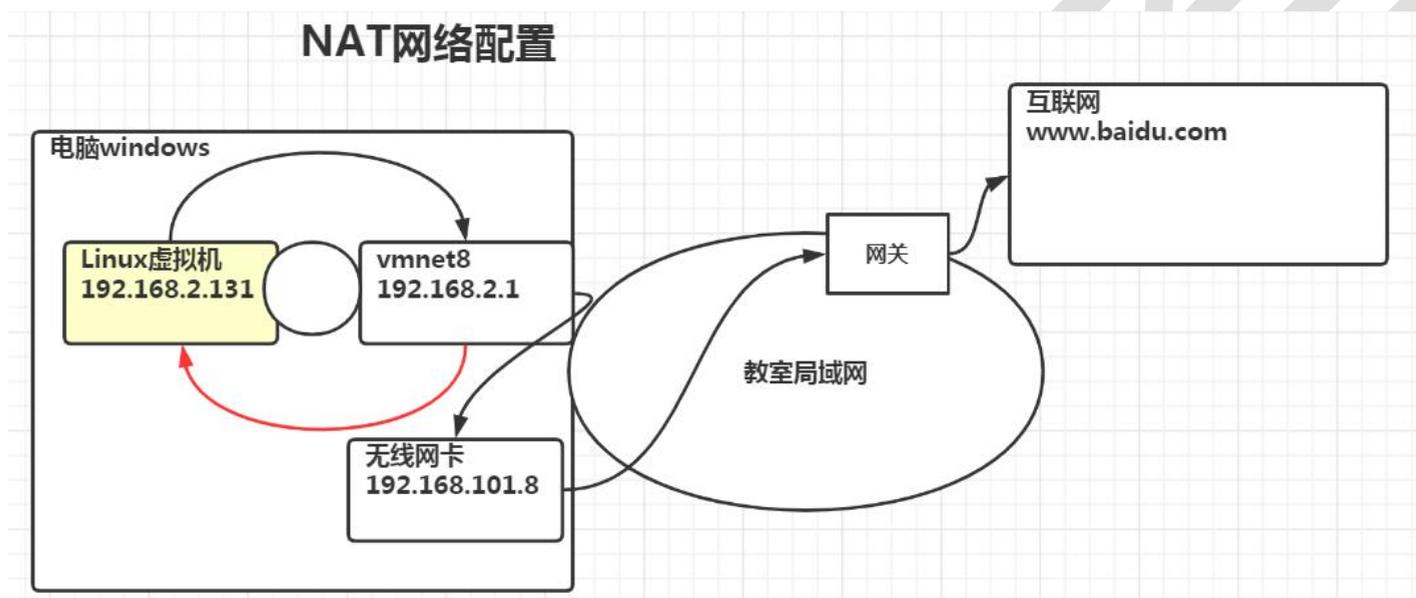
- 5) 以树状显示目录结构 tree 目录，注意，如果没有 tree,则使用 `yum install tree` 安装

```
[root@hsp aaa]# tree /home/
/home/
├── aaa
│   ├── abc.txt
│   └── bbb
│       └── ttt.txt
├── myroot -> /root/
└── newdisk
```

第 13 章 Linux 实操篇-网络配置

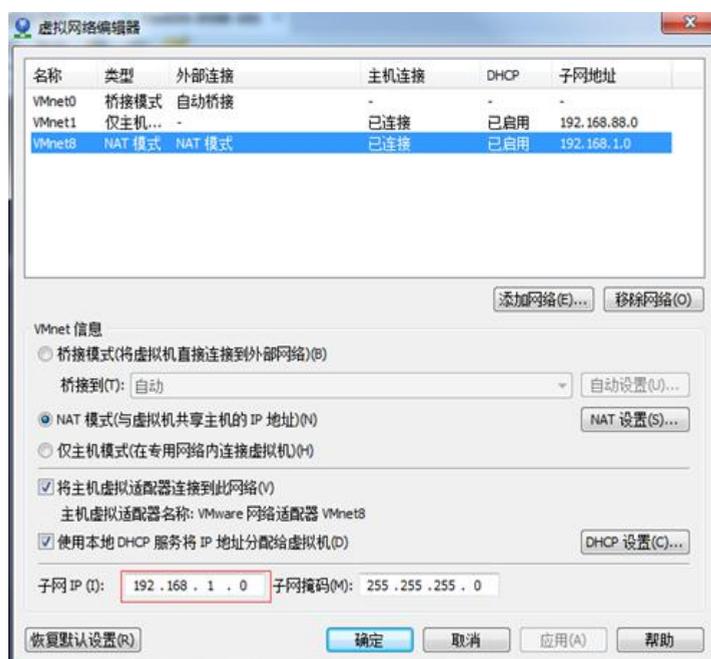
13.1 Linux 网络配置原理图

13.1.1 画出原理图，帮助理解

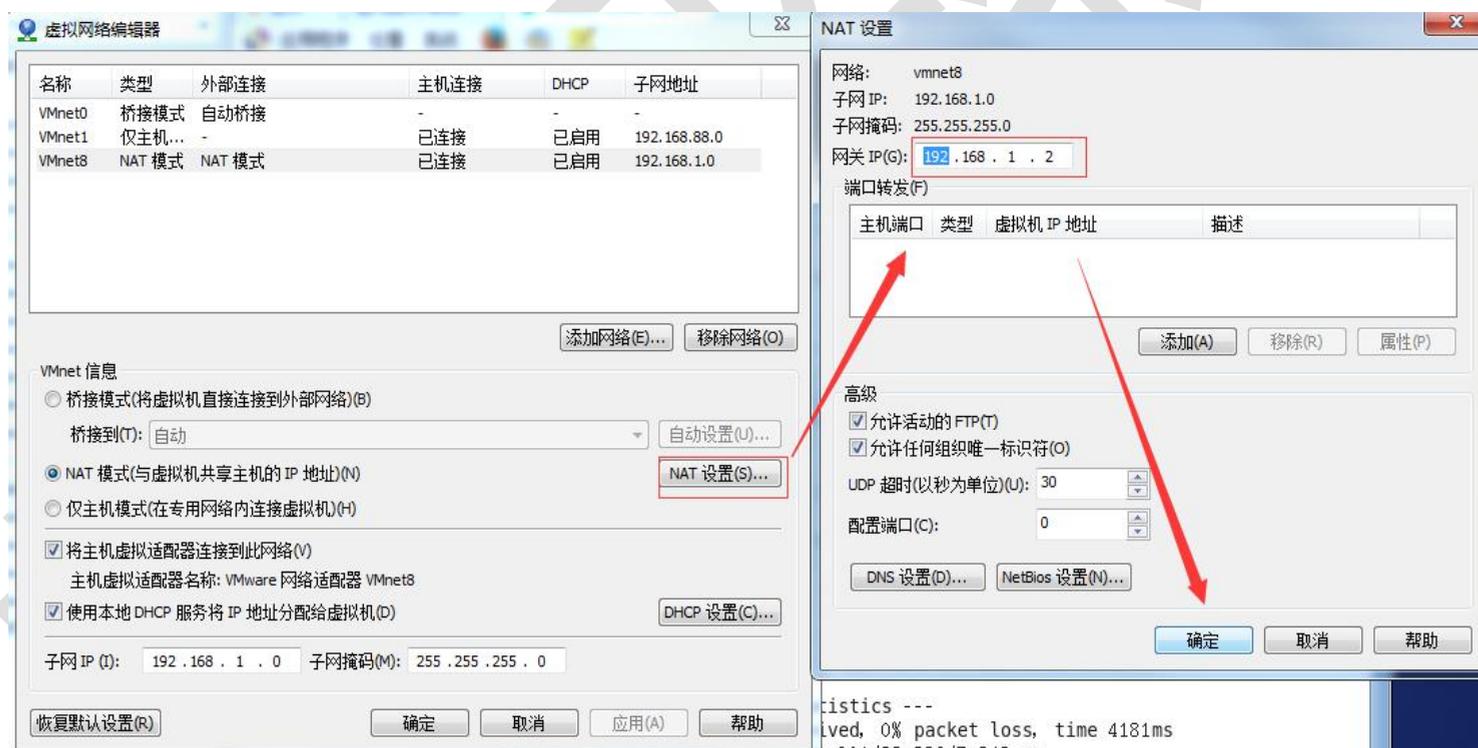


13.2 查看网络 IP 和网关

13.2.1 查看虚拟网络编辑器和修改 IP 地址



13.2.2 查看网关



13.3 查看 windows 环境中 VMnet8 网络配置 (ipconfig 指令)

```
以太网适配器 VMware Network Adapter VMnet8:
  连接特定的 DNS 后缀 . . . . . :
  本地连接 IPv6 地址 . . . . . : fe80::cc2d:b64b:b402:2a15%26
  IPv4 地址 . . . . . : 192.168.200.1
  子网掩码 . . . . . : 255.255.255.0
  默认网关 . . . . . :
```

13.4 查看 linux 的网络配置 ifconfig

```
[root@hspEdu100 ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.200.130 netmask 255.255.255.0 broadcast 192.168.200.255
  inet6 fe80::9a60:e2ac:588c:81bb prefixlen 64 scopeid 0x20<link>
  ether 00:0c:29:c2:8e:1a txqueuelen 1000 (Ethernet)
  RX packets 686 bytes 59334 (57.9 KiB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 595 bytes 72578 (70.8 KiB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

13.5 ping 测试主机之间网络连通性

13.5.1 基本语法

ping 目的主机 （功能描述：测试当前服务器是否可以连接目的主机）

13.5.2 应用实例

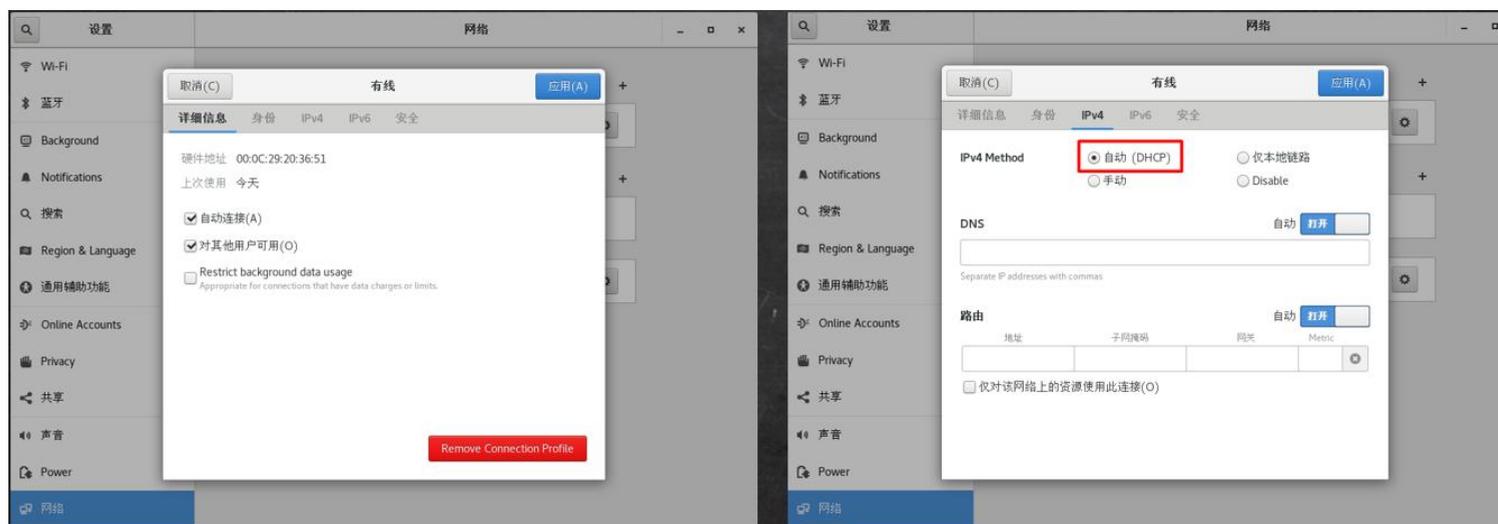
测试当前服务器是否可以连接百度

ping www.baidu.com

13.6 linux 网络环境配置

13.6.1 第一种方法(自动获取):

说明：登陆后，通过界面的来设置自动获取 ip，特点：linux 启动后会自动获取 IP,缺点是每次自动获取的 ip 地址可能不一样



13.6.2 第二种方法(指定 ip)

✓ 说明

直接修改配置文件来指定 IP,并可以连接到外网(程序员推荐)

✓ 编辑 `vi /etc/sysconfig/network-scripts/ifcfg-ens33`

要求: 将 ip 地址配置的静态的, 比如: ip 地址为 192.168.200.130

ifcfg-ens33 文件说明

DEVICE=eth0 #接口名 (设备,网卡)

HWADDR=00:0C:2x:6x:0x:xx #MAC 地址

TYPE=Ethernet #网络类型 (通常是 Ethemet)

UUID=926a57ba-92c6-4231-bacb-f27e5e6a9f44 #随机 id

#系统启动的时候网络接口是否有效 (yes/no)

ONBOOT=yes

IP 的配置方法[none|static|bootp|dhcp] (引导时不使用协议|静态分配 IP|BOOTP 协议|DHCP 协议)

BOOTPROTO=static

#IP 地址

IPADDR=192.168.200.130

#网关

GATEWAY=192.168.200.2

#域名解析器

DNS1=192.168.200.2

- ✓ 重启网络服务或者重启系统生效
service network restart 、reboot

13.7 设置主机名和 hosts 映射

13.7.1 设置主机名

- 1) 为了方便记忆，可以给 **linux** 系统设置主机名，也可以根据需要修改主机名
- 2) 指令 hostname : 查看主机名
- 3) 修改文件在 **/etc/hostname** 指定
- 4) 修改后，重启生效

13.7.2 设置 hosts 映射

思考：如何通过 主机名能够找到(比如 ping) 某个 linux 系统？

- ✓ windows

在 **C:\Windows\System32\drivers\etc\hosts** 文件指定即可

案例: 192.168.200.130 hspedu100

- ✓ linux

在 **/etc/hosts** 文件 指定

案例: 192.168.200.1 ThinkPad-PC

13.8 主机名解析过程分析(Hosts、DNS)

13.8.1 Hosts 是什么

一个文本文件，用来记录 IP 和 Hostname(主机名)的映射关系

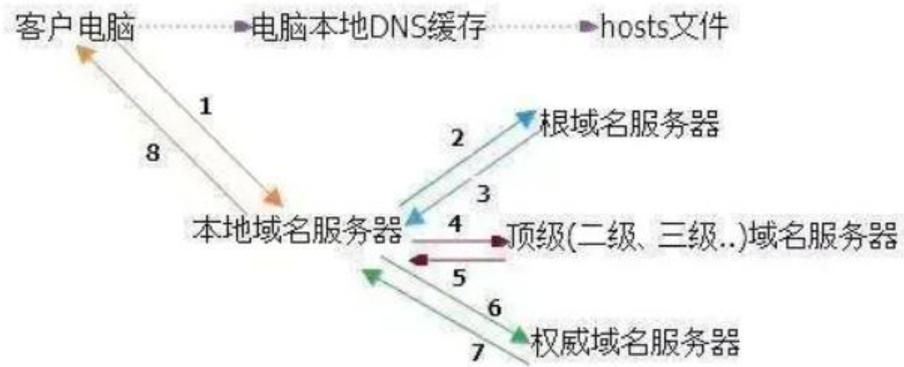
13.8.2 DNS

DNS，就是 **Domain Name System** 的缩写，翻译过来就是域名系统是互联网上作为域名和 IP 地址相互映射的一个分布式数据库

13.8.3 应用实例: 用户在浏览器输入了 www.baidu.com

- 1) 浏览器先检查浏览器缓存中有没有该域名解析 IP 地址，有就先调用这个 IP 完成解析；如果没有，就检查 DNS 解析器缓存，如果有直接返回 IP 完成解析。这两个缓存，可以理解为 本地解析器缓存
- 2) 一般来说，当电脑第一次成功访问某一网站后，在一定时间内，浏览器或操作系统会缓存他的 IP 地址（DNS 解析记录）。如在 cmd 窗口中输入

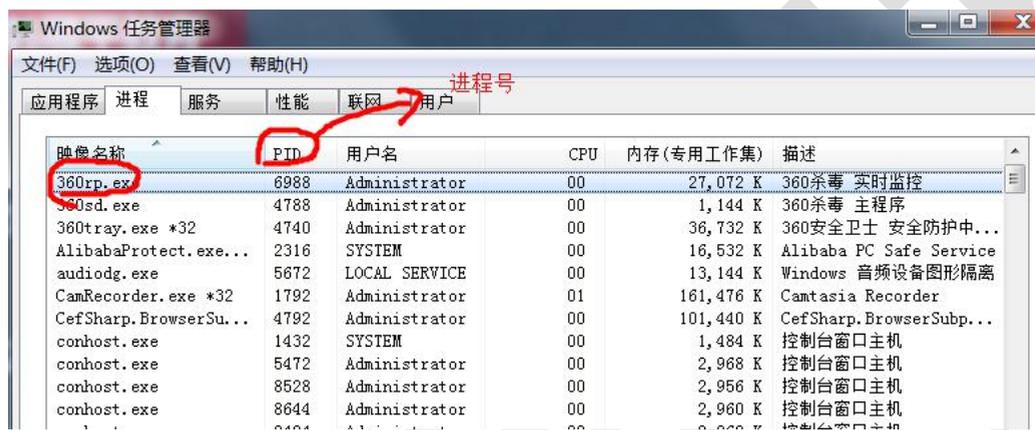
```
ipconfig /displaydns //DNS 域名解析缓存  
ipconfig /flushdns //手动清理 dns 缓存
```
- 3) 如果本地解析器缓存没有找到对应映射，检查系统中 hosts 文件中有没有配置对应的域名 IP 映射，如果有，则完成解析并返回。
- 4) 如果 本地 DNS 解析器缓存 和 hosts 文件 中均没有找到对应的 IP，则到域名服务 DNS 进行解析域
- 5) 示意图



第 14 章 Linux 实操篇-进程管理(重点)

14.1 基本介绍

- 1) 在 LINUX 中，每个执行的程序都称为一个进程。每一个进程都分配一个 ID 号(pid,进程号)。=>windows => linux
- 2) 每个进程都可能以两种方式存在的。前台与后台，所谓前台进程就是用户目前的屏幕上可以进行操作的。后台进程则是实际在操作，但由于屏幕上无法看到的进程，通常使用后台方式执行。
- 3) 一般系统的服务都是以后台进程的方式存在，而且都会常驻在系统中。直到关机才才结束。
- 4) 示意图



14.2 显示系统执行的进程

14.2.1 基本介绍

ps 命令是用来查看目前系统中，有哪些正在执行，以及它们执行的状况。可以不加任何参数。

ps 显示的信息选项:

字段	说明
PID	进程识别号
TTY	终端机号
TIME	此进程所消 CPU 时间
CMD	正在执行的进程

ps -a: 显示当前终端的所有进程信息
ps -u: 以用户的格式显示进程信息
ps -x: 显示后台进程运行的参数

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.2	0.3	128424	7052	?	Ss	10:01	0:05	/usr/lib/systemd/systemd --switched-
root	2	0.0	0.0	0	0	?	S	10:01	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	10:01	0:01	[ksoftirqd/0]
root	5	0.0	0.0	0	0	?	S<	10:01	0:00	[kworker/0:0H]
root	7	0.0	0.0	0	0	?	S	10:01	0:00	[migration/0]
root	8	0.0	0.0	0	0	?	S	10:01	0:00	[rcu_bh]
root	9	0.1	0.0	0	0	?	S	10:01	0:02	[rcu_sched]
root	10	0.0	0.0	0	0	?	S<	10:01	0:00	[lru-add-drain]
root	11	0.0	0.0	0	0	?	S	10:01	0:00	[watchdog/0]
root	12	0.0	0.0	0	0	?	S	10:01	0:00	[watchdog/1]
root	13	0.0	0.0	0	0	?	S	10:01	0:00	[migration/1]
root	14	0.0	0.0	0	0	?	S	10:01	0:00	[ksoftirqd/1]
root	16	0.0	0.0	0	0	?	S<	10:01	0:00	[kworker/1:0H]
root	17	0.0	0.0	0	0	?	S	10:01	0:00	[watchdog/2]
root	18	0.0	0.0	0	0	?	S	10:01	0:00	[migration/2]
root	19	0.0	0.0	0	0	?	S	10:01	0:00	[ksoftirqd/2]

14.2.2 ps 详解

- 1) 指令: `ps -aux|grep xxx` , 比如我看看有没有 sshd 服务
- 2) 指令说明
 - ✓ System V 展示风格
 - ✓ USER: 用户名称
 - ✓ PID: 进程号
 - ✓ %CPU: 进程占用 CPU 的百分比
 - ✓ %MEM: 进程占用物理内存的百分比
 - ✓ VSZ: 进程占用的虚拟内存大小 (单位: KB)
 - ✓ RSS: 进程占用的物理内存大小 (单位: KB)
 - ✓ TT: 终端名称, 缩写
 - ✓ STAT: 进程状态, 其中 S-睡眠, s-表示该进程是会话的先导进程, N-表示进程拥有比普通优先级更低的优先级, R-正在运行, D-短期等待, Z-僵死进程, T-被跟踪或者被停止等等
 - ✓ STARTED: 进程的启动时间
 - ✓ TIME: CPU 时间, 即进程使用 CPU 的总时间
 - ✓ COMMAND: 启动进程所用的命令和参数, 如果过长会被截断显示

14.2.3 应用实例

- ✓ 要求：以全格式显示当前所有的进程，查看进程的父进程。查看 `sshd` 的父进程信息

`ps -ef` 是以全格式显示当前所有的进程

`-e` 显示所有进程。`-f` 全格式

```
ps -ef|grep sshd
```

- ✓ 是 BSD 风格

UID: 用户 ID

PID: 进程 ID

PPID: 父进程 ID

C: CPU 用于计算执行优先级的因子。数值越大，表明进程是 CPU 密集型运算，执行优先级会降低；数值越小，表明进程是 I/O 密集型运算，执行优先级会提高

STIME: 进程启动的时间

TTY: 完整的终端名称

TIME: CPU 时间

CMD: 启动进程所用的命令和参数

14.3 终止进程 `kill` 和 `killall`

14.3.1 介绍:

若是某个进程执行一半需要停止时，或是已消了很大的系统资源时，此时可以考虑停止该进程。使用 `kill` 命令来完成此项任务。

14.3.2 基本语法

`kill` [选项] 进程号 (功能描述: 通过进程号杀死/终止进程)

`killall` 进程名称 (功能描述: 通过进程名称杀死进程，也支持通配符，这在系统因负载过大而变得很慢时很有用)

14.3.3 常用选项

-9 :表示强迫进程立即停止

14.3.4 最佳实践

- 1) 案例 1: 踢掉某个非法登录用户
kill 进程号 , 比如 kill 11421
- 2) 案例 2: 终止远程登录服务 sshd, 在适当时候再次重启 sshd 服务
kill sshd 对应的进程号; /bin/systemctl start sshd.service
- 3) 案例 3: 终止多个 gedit , 演示 killall gedit
- 4) 案例 4: 强制杀掉一个终端, 指令 kill -9 bash 对应的进程号

14.4 查看进程树 pstree

14.4.1 基本语法

pstree [选项], 可以更加直观的来看进程信息

14.4.2 常用选项

- p :显示进程的 PID
- u :显示进程的所属用户

14.4.3 应用实例:

- ✓ 案例 1: 请你树状的形式显示进程的 pid
pstree -p
- ✓ 案例 2: 请你树状的形式进程的用户
pstree -u

```
systemd(1)─┬─ModemManager(6811)─┬─{ModemManager}(6911)
              │                 │─{ModemManager}(6913)
              └─NetworkManager(7068)─┬─{NetworkManager}(7079)
                                      └─{NetworkManager}(7087)
─VGAAuthService(6813)
─abrt-watch-log(6831)
─abrt-watch-log(6837)
─abrt-d(6812)
─accounts-daemon(6810)─{accounts-daemon}(6812)
```

14.5 服务(service)管理

14.5.1 介绍:

服务(service) 本质就是进程,但是是运行在后台的,通常都会监听某个端口,等待其它程序的请求,比如(mysql, sshd 防火墙等), 因此我们又称为守护进程, 是 Linux 中非常重要的知识点。【原理图】

14.5.2 service 管理指令

- 1) service 服务名 [start | stop | restart | reload | status]
- 2) 在 CentOS7.0 后 很多服务不再使用 service ,而是 systemctl (后面专门讲)
- 3) service 指令管理的服务在 /etc/init.d 查看

```
[root@hspedu01 ~]# ls -l /etc/init.d/
总用量 88
-rw-r--r--. 1 root root 18281 8月 24 2018 functions
-rwxr-xr-x. 1 root root 4569 8月 24 2018 netconsole
-rwxr-xr-x. 1 root root 7923 8月 24 2018 network
-rw-r--r--. 1 root root 1160 10月 31 2018 README
```

14.5.3 service 管理指令案例

请使用 service 指令, 查看, 关闭, 启动 network [注意: 在虚拟系统演示, 因为网络连接会关闭]

指令:

```
service network status
```

```
service network stop
```

```
service network start
```

14.5.4 查看服务名:

- ✓ 方式 1: 使用 `setup -> 系统服务` 就可以看到全部。

setup



- ✓ 方式 2: `/etc/init.d` 看到 `service` 指令管理的服务

```
ls -l /etc/init.d
```

14.5.5 服务的运行级别(runlevel):

- ✓ Linux 系统有 7 种运行级别(runlevel): 常用的是级别 3 和 5

运行级别 0: 系统停机状态, 系统默认运行级别不能设为 0, 否则不能正常启动

运行级别 1: 单用户工作状态, root 权限, 用于系统维护, 禁止远程登陆

运行级别 2: 多用户状态(没有 NFS), 不支持网络

运行级别 3: 完全的多用户状态(有 NFS), 无界面, 登陆后进入控制台命令行模式

运行级别 4: 系统未使用, 保留

运行级别 5: X11 控制台, 登陆后进入图形 GUI 模式

运行级别 6: 系统正常关闭并重启, 默认运行级别不能设为 6, 否则不能正常启动

- ✓ 开机的流程说明:



14.5.6 CentOS7 后运行级别说明

在 `/etc/initab`

进行了简化，如下：

multi-user.target: analogous to runlevel 3

graphical.target: analogous to runlevel 5

init 0

To view current default target, run:

```
systemctl get-default
```

To set a default target, run:

```
systemctl set-default TARGET.target
```

14.5.7 chkconfig 指令

✓ 介绍

通过 `chkconfig` 命令可以给服务的各个运行级别设置自启动/关闭

`chkconfig` 指令管理的服务在 `/etc/init.d` 查看

注意: Centos7.0 后，很多服务使用 `systemctl` 管理 (后面马上讲)



✓ chkconfig 基本语法

- 1) 查看服务 `chkconfig --list [| grep xxx]`
- 2) `chkconfig 服务名 --list`
- 3) `chkconfig --level 5 服务名 on/off`

✓ 案例演示：对 network 服务 进行各种操作, 把 network 在 3 运行级别, 关闭自启动

```
chkconfig --level 3 network off
```

```
chkconfig --level 3 network on
```

✓ 使用细节

chkconfig 重新设置服务后自启动或关闭, 需要重启机器 reboot 生效.

14.5.8 systemctl 管理指令

基本语法: `systemctl [start | stop | restart | status] 服务名`

systemctl 指令管理的服务在 `/usr/lib/systemd/system` 查看

14.5.9 systemctl 设置服务的自启动状态

`systemctl list-unit-files [| grep 服务名]` (查看服务开机启动状态, grep 可以进行过滤)

`systemctl enable 服务名` (设置服务开机启动)

`systemctl disable 服务名` (关闭服务开机启动)

`systemctl is-enabled 服务名` (查询某个服务是否是自启动的)

14.5.10 应用案例:

查看当前防火墙的状况, 关闭防火墙和重启防火墙。=> `firewalld.service`

```
systemctl status firewalld; systemctl stop firewalld; systemctl start firewalld
```

14.5.11 细节讨论:

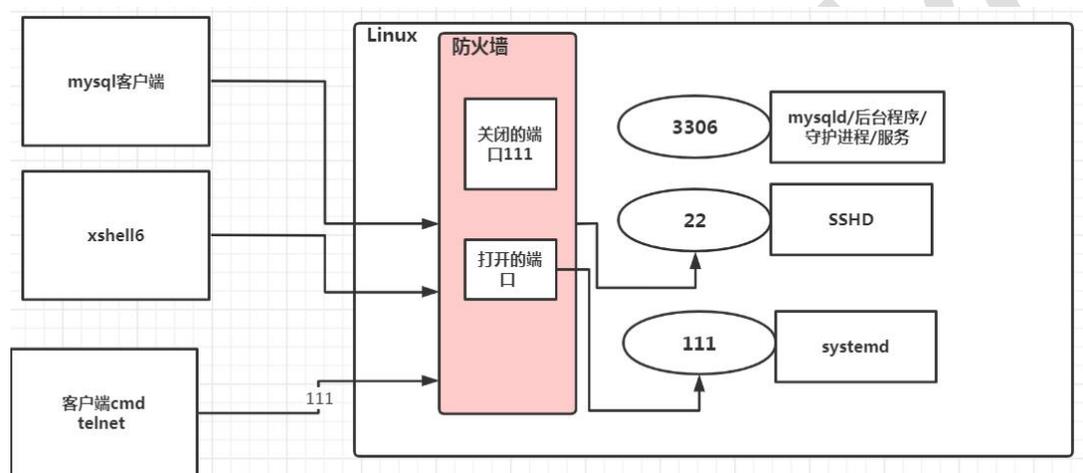
关闭或者启用防火墙后，立即生效。[telnet 测试 某个端口即可]

这种方式只是临时生效，当重启系统后，还是回归以前对服务的设置。

如果希望设置某个服务自启动或关闭永久生效，要使用 `systemctl [enable|disable] 服务名`。[演示]

14.5.12 打开或者关闭指定端口

在真正的生产环境，往往需要将防火墙打开，但问题来了，如果我们把防火墙打开，那么外部请求数据包就不能跟服务器监听端口通讯。这时，需要打开指定的端口。比如 80、22、8080 等，这个又怎么做呢？老韩给大家讲一讲。[示意图]



14.5.13 firewall 指令

- 1) 打开端口: `firewall-cmd --permanent --add-port=端口号/协议`
- 2) 关闭端口: `firewall-cmd --permanent --remove-port=端口号/协议`
- 3) 重新载入,才能生效 : `firewall-cmd --reload`
- 4) 查询端口是否开放: `firewall-cmd --query-port=端口/协议`

14.5.14 应用案例:

- 1) 启用防火墙，测试 111 端口是否能 telnet，不行
- 2) 开放 111 端口
`firewall-cmd --permanent --add-port=111/tcp` ; 需要 `firewall-cmd --reload`
- 3) 再次关闭 111 端口

firewall-cmd --permanent --remove-port=111/tcp ; 需要 firewall-cmd --reload

14.6 动态监控进程

14.6.1 介绍:

top 与 ps 命令很相似。它们都用来显示正在执行的进程。Top 与 ps 最大的不同之处，在于 top 在执行一段时间可以更新正在运行的的进程。

14.6.2 基本语法

top [选项]

```

top - 20:11:12 up 4:47, 2 users, load average: 0.00, 0.01, 0.05
Tasks: 185 total, 1 running, 184 sleeping, 0 stopped, 0 zombie
%cpu(s): 0.0 us, 0.1 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2028116 total, 1550944 free, 191720 used, 285452 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 1631776 avail Mem

  PID USER  PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
    61 root   20   0     0     0   0   S  0.3   0.0   0:16.04 kworker/0:1
   6923 root   20   0 246484 6852 5368  S  0.3   0.3   0:29.16 vmtoolsd
  13467 root   20   0 162012 2332 1596  R  0.3   0.1   0:00.10 top
    1 root   20   0 193972 7124 4220  S  0.0   0.4   0:05.02 systemd
    2 root   20   0     0     0   0   S  0.0   0.0   0:00.00 kthreadd
    3 root   20   0     0     0   0   S  0.0   0.0   0:00.40 ksoftirqd/0
  
```

14.6.3 选项说明:

选项	功能
-d 秒数	指定top命令每隔几秒更新。默认是3秒
-i	使top不显示任何闲置或者僵死进程。
-p	通过指定监控进程ID来仅仅监控某个进程的状态。

14.6.4 交互操作说明:

操作	功能
P	以CPU使用率排序，默认就是此项
M	以内存的使用率排序
N	以PID排序
q	退出top

14.6.5 应用实例

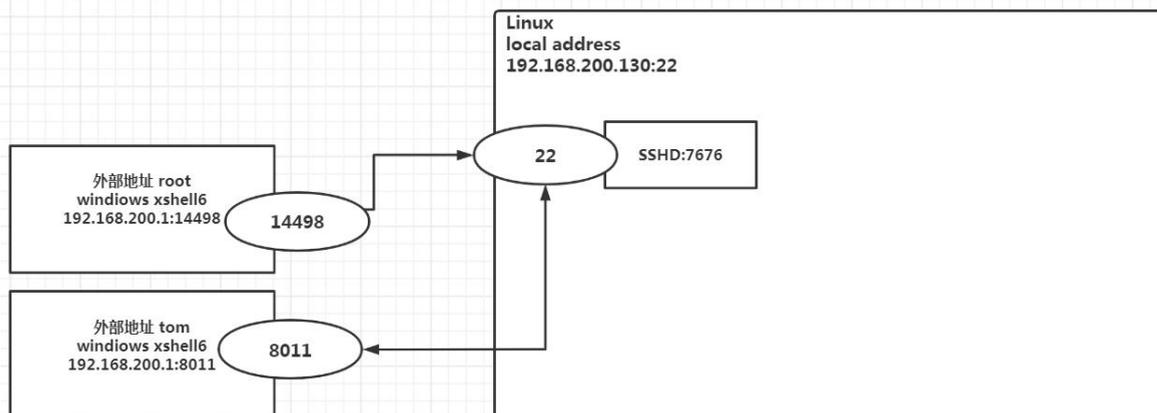
- ✓ 案例 1: 监视特定用户，比如我们监控 tom 用户
top: 输入此命令，按回车键，查看执行的进程。
u: 然后输入“u”回车，再输入用户名，即可，
- ✓ 案例 2: 终止指定的进程，比如我们要结束 tom 登录
top: 输入此命令，按回车键，查看执行的进程。
k: 然后输入“k”回车，再输入要结束的进程 ID 号
- ✓ 案例 3: 指定系统状态更新的时间(每隔 10 秒自动更新)，默认是 3 秒
top -d 10

14.7 监控网络状态

14.7.1 查看系统网络情况 netstat

- ✓ 基本语法
netstat [选项]
- ✓ 选项说明
 - an 按一定顺序排列输出
 - p 显示哪个进程在调用
- ✓ 应用案例
请查看服务名为 sshd 的服务的信息。
netstat -anp | grep sshd

```
[root@hspEdu100 ~]# netstat -anp | grep sshd
tcp        0      0 0.0.0.0:22          0.0.0.0:*          LISTEN     7676/sshd
tcp        0      0 127.0.0.1:6011     0.0.0.0:*          LISTEN     11671/sshd: root@pt
tcp        0      0 192.168.200.130:22 192.168.200.1:14498 ESTABLISHED 11671/sshd: root@pt
tcp6       0      0 :::22             :::*               LISTEN     7676/sshd
```



14.7.2 检测主机连接命令 ping:

是一种网络检测工具，它主要是用检测远程主机是否正常，或是两部主机间的网线或网卡故障。

如: ping 对方 ip 地址

第 15 章 Linux 实操篇-RPM 与 YUM

15.1 rpm 包的管理

15.1.1 介绍

rpm 用于互联网下载包的打包及安装工具，它包含在某些 Linux 分发版中。它生成具有.RPM 扩展名的文件。RPM 是 RedHat Package Manager（RedHat 软件包管理工具）的缩写，类似 windows 的 setup.exe，这一文件格式名称虽然打上了 RedHat 的标志，但理念是通用的。

Linux 的分发版本都有采用（suse,redhat, centos 等等），可以算是公认的行业标准了。

15.1.2 rpm 包的简单查询指令

查询已安装的 rpm 列表 `rpm -qa|grep xx`

举例：看看当前系统，是否安装了 firefox

指令：`rpm -qa | grep firefox`

15.1.3 rpm 包名基本格式

一个 rpm 包名：`firefox-60.2.2-1.el7.centos.x86_64`

名称:firefox

版本号: 60.2.2-1

适用操作系统: el7.centos.x86_64

表示 centos7.x 的 64 位系统

如果是 i686、i386 表示 32 位系统，noarch 表示通用

15.1.4 rpm 包的其它查询指令：

`rpm -qa` :查询所安装的所有 rpm 软件包

`rpm -qa | more`

`rpm -qa | grep X [rpm -qa | grep firefox]`

rpm -q 软件包名 :查询软件包是否安装

案例: rpm -q firefox

rpm -qi 软件包名 : 查询软件包信息

案例: rpm -qi firefox

rpm -ql 软件包名 :查询软件包中的文件

比如: rpm -ql firefox

rpm -qf 文件全路径名 查询文件所属的软件包

rpm -qf /etc/passwd

rpm -qf /root/install.log

15.1.5 卸载 rpm 包:

✓ 基本语法

rpm -e RPM 包的名称 //erase

✓ 应用案例

删除 firefox 软件包

rpm -e firefox

✓ 细节讨论

1) 如果其它软件包依赖于您要卸载的软件包, 卸载时则会产生错误信息。

如: \$ rpm -e foo

removing these packages would break dependencies:foo is needed by bar-1.0-1

2) 如果我们就是要删除 foo 这个 rpm 包, 可以增加参数 --nodeps ,就可以强制删除, 但是一般不推荐这样做, 因为依赖于该软件包的程序可能无法运行

如: \$ rpm -e --nodeps foo

15.1.6 安装 rpm 包

✓ 基本语法

`rpm -ivh` RPM 包全路径名称

✓ 参数说明

`i=install` 安装

`v=verbose` 提示

`h=hash` 进度条

✓ 应用实例

演示卸载和安装 firefox 浏览器

`rpm -e firefox`

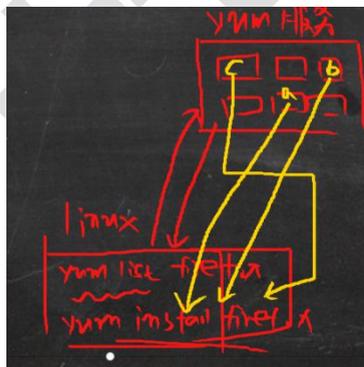
`rpm -ivh firefox`

15.2 yum

15.2.1 介绍:

Yum 是一个 Shell 前端软件包管理器。基于 RPM 包管理，能够从指定的服务器自动下载 RPM 包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软件包。

示意图



15.2.2 yum 的基本指令

查询 yum 服务器是否有需要安装的软件

```
yum list|grep xx 软件列表
```

15.2.3 安装指定的 yum 包

```
yum install xxx 下载安装
```

15.2.4 yum 应用实例：

案例：请使用 yum 的方式来安装 firefox

```
rpm -e firefox
```

```
yum list | grep firefox
```

```
yum install firefox
```

第 16 章 Linux 之 JavaEE 定制篇-搭建 JavaEE 环境

16.1 概述

如果需要在 Linux 下进行 JavaEE 的开发，我们需要安装如下软件



16.2 安装 JDK

16.2.1 安装步骤

- 1) mkdir /opt/jdk
- 2) 通过 xftp6 上传到 /opt/jdk 下
- 3) cd /opt/jdk
- 4) 解压 tar -zxvf jdk-8u261-linux-x64.tar.gz
- 5) mkdir /usr/local/java
- 6) mv /opt/jdk/jdk1.8.0_261 /usr/local/java
- 7) 配置环境变量的配置文件 vim /etc/profile
- 8) export JAVA_HOME=/usr/local/java/jdk1.8.0_261
- 9) export PATH=\$JAVA_HOME/bin:\$PATH
- 10) source /etc/profile [让新的环境变量生效]

16.2.2 测试是否安装成功

编写一个简单的 Hello.java 输出"hello,world!"

16.3 tomcat 的安装

16.3.1 步骤：

- 1) 上传安装文件，并解压缩到/opt/tomcat
- 2) 进入解压目录/bin，启动 tomcat `./startup.sh`
- 3) 开放端口 8080，回顾 `firewall-cmd`

16.3.2 测试是否安装成功：

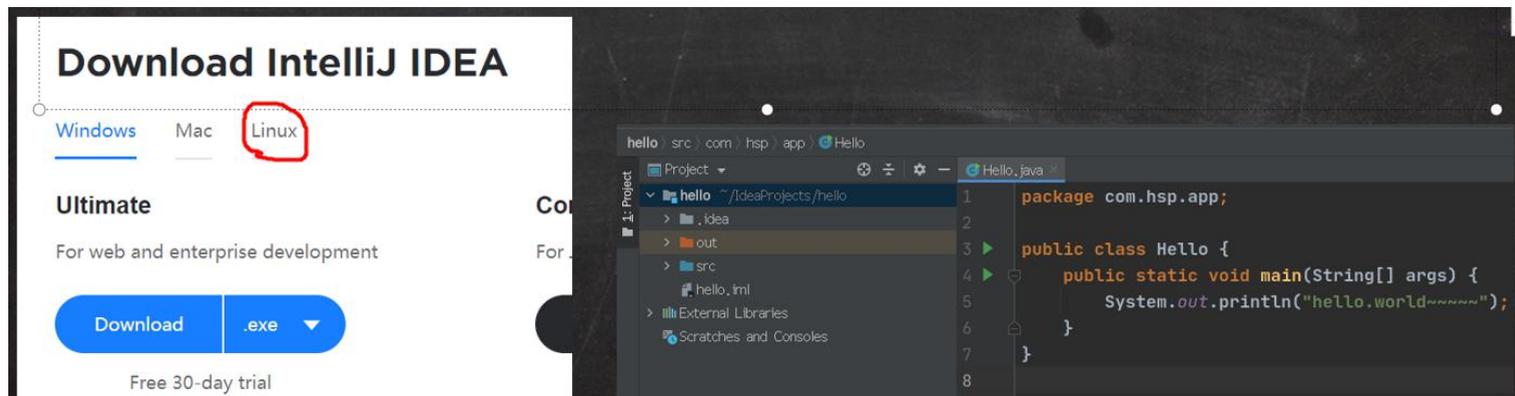
在 windows、Linux 下 访问 <http://linuxip:8080>



16.4 idea2020 的安装

16.4.1 步骤

- 1) 下载地址: <https://www.jetbrains.com/idea/download/#section=windows>
- 2) 解压缩到/opt/idea
- 3) 启动 idea bin 目录下 `./idea.sh`，配置 jdk
- 4) 编写 Hello world 程序并测试成功！



16.5 mysql5.7 的安装 (!!)

- 1) 安装的步骤和文档
- 2) 相关的安装软件, 已经准备好了
- 3) 安装看老师演示-文档

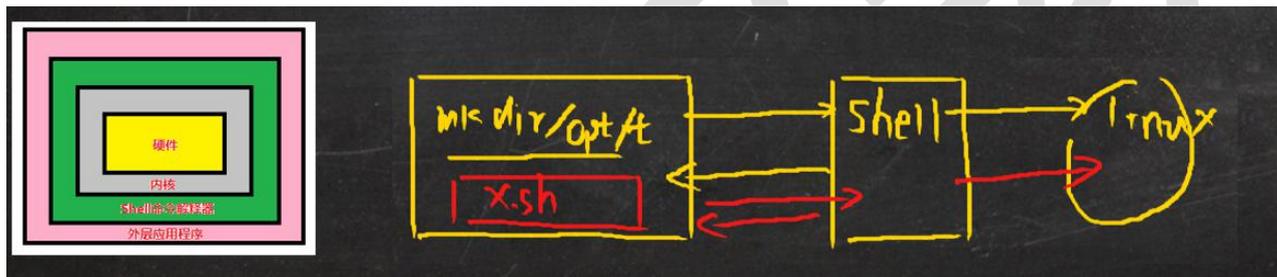
第 17 章大数据定制篇-Shell 编程

17.1 为什么要学习 Shell 编程

- 1) Linux 运维工程师在进行服务器集群管理时，需要编写 Shell 程序来进行服务器管理。
- 2) 对于 JavaEE 和 Python 程序员来说，工作的需要，你的老大会要求你编写一些 Shell 脚本进行程序或者是服务器的维护，比如编写一个定时备份数据库的脚本。
- 3) 对于大数据程序员来说，需要编写 Shell 程序来管理集群

17.2 Shell 是什么

Shell 是一个命令行解释器，它为用户提供了一个向 Linux 内核发送请求以便运行程序的界面系统级程序，用户可以用 Shell 来启动、挂起、停止甚至是编写一些程序。看一个示意图



17.3 Shell 脚本的执行方式

17.3.1 脚本格式要求

- 1) 脚本以#!/bin/bash 开头
- 2) 脚本需要有可执行权限

17.3.2 编写第一个 Shell 脚本

- ✓ 需求说明：创建一个 Shell 脚本，输出 hello world!

```
vim hello.sh
```

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

```
echo "hello,world~"
```

17.3.3 脚本的常用执行方式

✓ 方式 1(输入脚本的绝对路径或相对路径)

说明：首先要赋予 helloworld.sh 脚本的+x 权限，再执行脚本

比如 ./hello.sh 或者使用绝对路径 /root/shcode/hello.sh

✓ 方式 2(sh+脚本)

说明：不用赋予脚本+x 权限，直接执行即可。

比如 sh hello.sh，也可以使用绝对路径

17.4 Shell 的变量

17.4.1 Shell 变量介绍

- 1) Linux Shell 中的变量分为，系统变量和用户自定义变量。
- 2) 系统变量：\$HOME、\$PWD、\$SHELL、\$USER 等等，比如：echo \$HOME 等等..
- 3) 显示当前 shell 中所有变量：set

17.4.2 shell 变量的定义

✓ 基本语法

- 1) 定义变量：变量名=值
- 2) 撤销变量：unset 变量
- 3) 声明静态变量：readonly 变量，注意：不能 unset

✓ 快速入门

- 1) 案例 1：定义变量 A
- 2) 案例 2：撤销变量 A
- 3) 案例 3：声明静态的变量 B=2，不能 unset

```
#!/bin/bash
#案例 1: 定义变量 A
A=100
#输出变量需要加上$
echo A=$A
echo "A=$A"
#案例 2: 撤销变量 A
unset A
echo "A=$A"
#案例 3: 声明静态的变量 B=2, 不能 unset
readonly B=2
echo "B=$B"
#unset B
#将指令返回的结果赋给变量
:<<!
C=`date`
D=$(date)
echo "C=$C"
echo "D=$D"
!
#使用环境变量 TOMCAT_HOME
echo "tomcat_home=$TOMCAT_HOME"
```

4) 案例 4: 可把变量提升为全局环境变量, 可供其他 shell 程序使用[该案例后面讲]

17.4.3 shell 变量的定义

✓ 定义变量的规则

- 1) 变量名称可以由字母、数字和下划线组成，但是不能以数字开头。5A=200(×)
- 2) 等号两侧不能有空格
- 3) 变量名称一般习惯为大写，这是一个规范，我们遵守即可

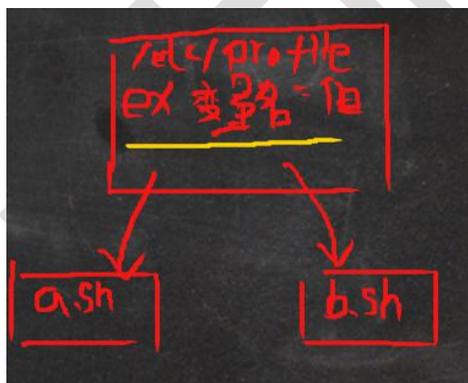
✓ 将命令的返回值赋给变量

- 1) A=`date`反引号，运行里面的命令，并把结果返回给变量 A
- 2) A=\$(date) 等价于反引号

17.5 设置环境变量

17.5.1 基本语法

- 1) export 变量名=变量值 （功能描述：将 shell 变量输出为环境变量/全局变量）
- 2) source 配置文件 （功能描述：让修改后的配置信息立即生效）
- 3) echo \$变量名 （功能描述：查询环境变量的值）
- 4) 示意



17.5.2 快速入门

- 1) 在/etc/profile 文件中定义 TOMCAT_HOME 环境变量
- 2) 查看环境变量 TOMCAT_HOME 的值
- 3) 在另外一个 shell 程序中使用 TOMCAT_HOME

注意：在输出 TOMCAT_HOME 环境变量前，需要让其生效

```
source /etc/profile
```

```
export PATH=$JAVA_HOME/bin:$PATH
#定义一个环境变量
export TOMCAT_HOME=/opt/tomcat
```

shell 脚本的多行注释

```
:<<! 内容 !
```

17.6 位置参数变量

17.6.1 介绍

当我们执行一个 shell 脚本时，如果希望获取到命令行的参数信息，就可以使用到位置参数变量

比如：`./myshell.sh 100 200`，这个就是一个执行 shell 的命令，可以在 myshell 脚本中获取到参数信息

17.6.2 基本语法

`$n`（功能描述：`n` 为数字，`$0` 代表命令本身，`$1-$9` 代表第一到第九个参数，十以上的参数，十以上的参数需要用大括号包含，如 `${10}`）

`$*`（功能描述：这个变量代表命令行中所有的参数，`$*`把所有的参数看成一个整体）

`$@`（功能描述：这个变量也代表命令行中所有的参数，不过 `$@`把每个参数区分对待）

`$#`（功能描述：这个变量代表命令行中所有参数的个数）

17.6.3 位置参数变量

案例：编写一个 shell 脚本 position.sh ， 在脚本中获取到命令行的各个参数信息。

```
#!/bin/bash
echo "0=$0 1=$1 2=$2"
echo "所有的参数=$*"
echo "$@"
echo "参数的个数=$#"
```

17.7 预定义变量

17.7.1 基本介绍

就是 shell 设计者事先已经定义好的变量，可以直接在 shell 脚本中使用

17.7.2 基本语法

- 1) \$\$ （功能描述：当前进程的进程号（PID））
- 2) \$! （功能描述：后台运行的最后一个进程的进程号（PID））
- 3) \$? （功能描述：最后一次执行的命令的返回状态。如果这个变量的值为 0，证明上一个命令正确执行；如果这个变量的值为非 0（具体是哪个数，由命令自己来决定），则证明上一个命令执行不正确了。）

17.7.3 应用实例

在一个 shell 脚本中简单使用一下预定义变量

preVar.sh

```
#!/bin/bash
echo "当前执行的进程 id=$$"
#以后台的方式运行一个脚本，并获取他的进程号
/root/shcode/myshell.sh &
echo "最后一个后台方式运行的进程 id=$!"
echo "执行的结果是=$?"
```

17.8 运算符

17.8.1 基本介绍

学习如何在 shell 中进行各种运算操作。

17.8.2 基本语法

- 1) “`$(运算式)`”或“``${运算式}``”或者 `expr m + n //expression` 表达式
- 2) 注意 `expr` 运算符间要有空格, 如果希望将 `expr` 的结果赋给某个变量, 使用 ``
- 3) `expr m - n`
- 4) `expr *,/,%` 乘, 除, 取余

17.8.3 应用实例 oper.sh

案例 1: 计算 $(2+3) \times 4$ 的值

案例 2: 请求出命令行的两个参数[整数]的和 20 50

```
#!/bin/bash
#案例 1: 计算 (2+3) X4 的值
#使用第一种方式
RES1=$((2+3)*4)
echo "res1=$RES1"
#使用第二种方式, 推荐使用
RES2=$((2+3)*4)
echo "res2=$RES2"
#使用第三种方式 expr
TEMP=`expr 2 + 3`
RES4=`expr $TEMP \* 4`
echo "temp=$TEMP"
```

```
echo "res4=$RES4"
#案例 2: 请求出命令行的两个参数[整数]的和 20 50
SUM=${$1+$2}
echo "sum=$SUM"
```

17.9 条件判断

17.9.1 判断语句

✓ 基本语法

[condition] (注意 condition 前后要有空格)

#非空返回 true, 可使用 \$? 验证 (0 为 true, >1 为 false)

✓ 应用实例

[hspEdu] 返回 true

[] 返回 false

[condition] && echo OK || echo notok 条件满足, 执行后面的语句

✓ 判断语句

常用判断条件

1) = 字符串比较

2) 两个整数的比较

-lt 小于

-le 小于等于 little equal

-eq 等于

-gt 大于

-ge 大于等于

-ne 不等于

3) 按照文件权限进行判断

-r 有读的权限

-w 有写的权限

-x 有执行的权限

4) 按照文件类型进行判断

-f 文件存在并且是一个常规的文件

-e 文件存在

-d 文件存在并是一个目录

✓ 应用实例

案例 1: "ok"是否等于"ok"

判断语句: 使用 =

案例 2: 23 是否大于等于 22

判断语句: 使用 -ge

案例 3: /root/shcode/aaa.txt 目录中的文件是否存在

判断语句: 使用 -f

代码如下:

```
#!/bin/bash
#案例1: "ok"是否等于"ok"
#判断语句: 使用 =
if [ "ok" = "ok" ]
then
    echo "equal"
fi
#案例2: 23是否大于等于22
#判断语句: 使用 -ge
if [ 23 -ge 22 ]
then
    echo "大于"
fi
#案例3: /root/shcode/aaa.txt 目录中的文件是否存在
#判断语句: 使用 -f
if [ -f /root/shcode/aaa.txt ]
then
    echo "存在"
fi
#看几个案例
if [ hspedu ]
then
    echo "hello,hspedu"
fi
```

17.10 流程控制

17.10.1 if 判断

✓ 基本语法

```
if[ 条件判断式 ]
```

```
then
```

```
代码
```

```
fi
```

或者，多分支

```
if[ 条件判断式 ]
```

```
then
```

```
代码
```

```
elif[条件判断式]
```

```
then
```

```
代码
```

```
fi
```

- ✓ 注意事项: [条件判断式], 中括号和条件判断式之间必须有空格
- ✓ 应用实例 ifCase.sh

案例: 请编写一个 shell 程序, 如果输入的参数, 大于等于 60, 则输出 "及格了", 如果小于 60, 则输出 "不及格"

```
#!/bin/bash
#案例: 请编写一个shell程序, 如果输入的参数, 大于等于60,
if [ $1 -ge 60 ]
then
    echo "及格了"
elif [ $1 -lt 60 ]
then
    echo "不及格"
fi
```

17.10.2 case 语句

- ✓ 基本语法

```
case $变量名 in
```

```
"值 1")
```

```
如果变量的值等于值 1, 则执行程序 1
```

```
::
```

```
"值 2")
```

```
如果变量的值等于值 2, 则执行程序 2
```

```
::
```

```
...省略其他分支...
```

*)

如果变量的值都不是以上的值，则执行此程序

::

esac

✓ 应用实例 testCase.sh

案例 1：当命令行参数是 1 时，输出 "周一"，是 2 时，就输出"周二"，其它情况输出 "other"

```
#!/bin/bash
#案例1：当命令行参数是 1 时，输出
case $1 in
"1")
echo "周一"
;;
"2")
echo "周二"
;;
*)
echo "other..."
;;
esac
```

17.10.3 for 循环

✓ 基本语法 1

for 变量 in 值 1 值 2 值 3...

do

程序/代码

done

应用实例 testFor1.sh

案例 1：打印命令行输入的参数 [这里可以看出\$* 和 @\$ 的区别]

✓ 基本语法 2

for ((初始值;循环控制条件;变量变化))

do

程序/代码

done

✓ 应用实例 testFor2.sh

案例 1：从 1 加到 100 的值输出显示

```
#!/bin/bash
#案例1：打印命令行输入的参数 [这里可以看出$* 和 $@ 的区别]
#注意 $* 是把输入的参数，当做一个整体，所以，只会输出一句
for i in "$*"
do
    echo "num is $i"
done
#使用 $@来获取输入的参数，注意，这时是分别对待，所以有几个参数，就输出几
echo "======"
for j in "$@"
do
    echo "num is $j"
done
```

```
#!/bin/bash
#案例1：从1加到100的值输出显示，如何把 100做成一个变量
#定义一个变量 SUM
SUM=0
for ( i=1; i<=100; i++)
do
#写上你的业务代码
    SUM=$((SUM+i))
done
echo "总和SUM=$SUM"
```

17.10.4 while 循环

✓ 基本语法 1

while [条件判断式]

do

程序 /代码

done

注意：while 和 [有空格，条件判断式和 [也有空格

✓ 应用实例 testWhile.sh

案例 1：从命令行输入一个数 n，统计从 1+..+n 的值是多少？

```
#!/bin/bash
#案例 1：从命令行输入一个数 n，统计从 1+..+n 的值是多少？
SUM=0
i=0
while [ $i -le $1 ]
do
    SUM=$((SUM+$i))
    #i 自增
    i=$((i+1))
done
echo "执行结果=$SUM"
```

17.11 read 读取控制台输入

17.11.1 基本语法

read(选项)(参数)

选项：

-p: 指定读取值时的提示符；

-t: 指定读取值时等待的时间（秒），如果没有在指定的时间内输入，就不再等待了。。

参数

变量：指定读取值的变量名

17.11.2 应用实例 testRead.sh

案例 1: 读取控制台输入一个 NUM1 值

案例 2: 读取控制台输入一个 NUM2 值, 在 10 秒内输入。

代码:

```
#!/bin/bash
#案例 1: 读取控制台输入一个 NUM1 值
read -p "请输入一个数 NUM1=" NUM1
echo "你输入的 NUM1=$NUM1"
#案例 2: 读取控制台输入一个 NUM2 值, 在 10 秒内输入。
read -t 10 -p "请输入一个数 NUM2=" NUM2
echo "你输入的 NUM2=$NUM2"
```

17.12 函数

17.12.1 函数介绍

shell 编程和其它编程语言一样, 有系统函数, 也可以自定义函数。系统函数中, 我们这里就介绍两个。

17.12.2 系统函数

✓ basename 基本语法

功能: 返回完整路径最后 / 的部分, 常用于获取文件名

basename [pathname] [suffix]

basename [string] [suffix] (功能描述: basename 命令会删掉所有的前缀包括最后一个 ('/') 字符, 然后将字符串显示出来。

选项:

suffix 为后缀, 如果 suffix 被指定了, basename 会将 pathname 或 string 中的 suffix 去掉。

✓ 应用实例

案例 1: 请返回 /home/aaa/test.txt 的 "test.txt" 部分

```
basename /home/aaa/test.txt
```

✓ dirname 基本语法

功能: 返回完整路径最后 / 的前面的部分, 常用于返回路径部分

dirname 文件绝对路径 (功能描述: 从给定的包含绝对路径的文件名中去除文件名(非目录的部分), 然后返回剩下的路径(目录的部分))

✓ 应用实例

案例 1: 请返回 /home/aaa/test.txt 的 /home/aaa

```
dirname /home/aaa/test.txt
```

17.12.3 自定义函数

✓ 基本语法

```
[ function ] funname[()]  
{  
    Action;  
    [return int;]  
}
```

调用直接写函数名: funname [值]

✓ 应用实例

案例 1: 计算输入两个参数的和(动态的获取), getSum

代码

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

```
#案例 1: 计算输入两个参数的和(动态的获取), getSum
```

```
#定义函数 getSum
function getSum() {

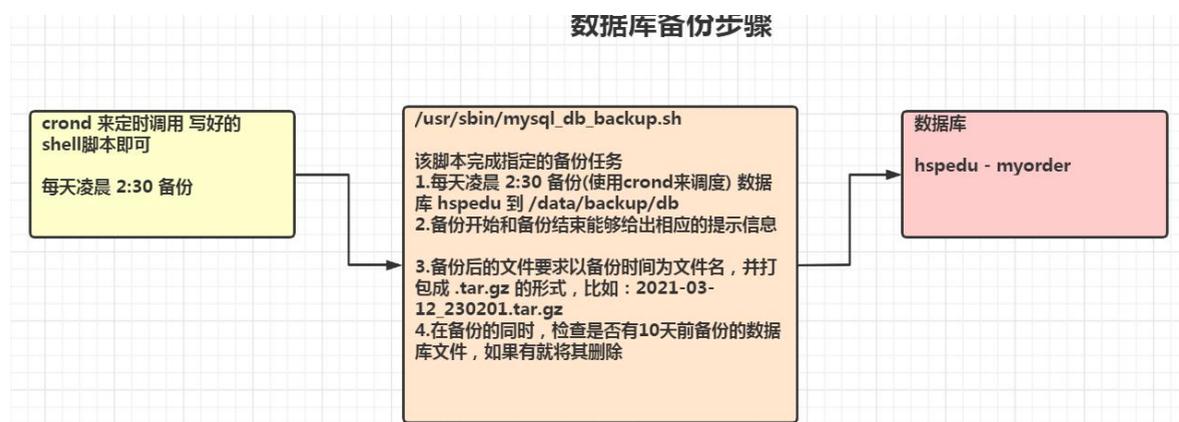
    SUM=${$n1+$n2}
    echo "和是=$SUM"
}

#输入两个值
read -p "请输入一个数 n1=" n1
read -p "请输入一个数 n2=" n2
#调用自定义函数
getSum $n1 $n2
```

17.13 Shell 编程综合案例

17.13.1 需求分析

- 1) 每天凌晨 2:30 备份 数据库 hspedu 到 /data/backup/db
- 2) 备份开始和备份结束能够给出相应的提示信息
- 3) 备份后的文件要求以备份时间为文件名，并打包成 .tar.gz 的形式，比如：2021-03-12_230201.tar.gz
- 4) 在备份的同时，检查是否有 10 天前备份的数据库文件，如果有就将其删除。
- 5) 画一个思路分析图



17.13.2 代码 /usr/sbin/mysql_db.backup.sh

```

#备份目录
BACKUP=/data/backup/db

#当前时间
DATETIME=$(date +%Y-%m-%d_%H%M%S)

echo $DATETIME

#数据库的地址
HOST=localhost

#数据库用户名
DB_USER=root

#数据库密码
DB_PW=hspedu100

#备份的数据库名
DATABASE=hspedu

#创建备份目录, 如果不存在, 就创建
[ ! -d "${BACKUP}/${DATETIME}" ] && mkdir -p "${BACKUP}/${DATETIME}"
  
```

```
#备份数据库
mysqldump -u${DB_USER} -p${DB_PW} --host=${HOST} -q -R --databases ${DATABASE} | gzip >
${BACKUP}/${DATETIME}/${DATETIME}.sql.gz

#将文件处理成 tar.gz
cd ${BACKUP}
tar -zcvf ${DATETIME}.tar.gz ${DATETIME}
#删除对应的备份目录
rm -rf ${BACKUP}/${DATETIME}

#删除 10 天前的备份文件
find ${BACKUP} -atime +10 -name "*.tar.gz" -exec rm -rf {} \;
echo "备份数据库${DATABASE} 成功~"
```

第 18 章 Linux 之 Python 定制篇-Python 开发平台 Ubuntu

18.1 Ubuntu 介绍

Ubuntu (友帮拓、优般图、乌班图) 是一个以桌面应用为主的开源 GNU/Linux 操作系统, Ubuntu 是基于 GNU/Linux, 支持 x86、amd64 (即 x64) 和 ppc 架构, 由全球化的专业开发团队 (Canonical Ltd) 打造的。

专业的 Python 开发者一般会选择 Ubuntu 这款 Linux 系统作为生产平台。

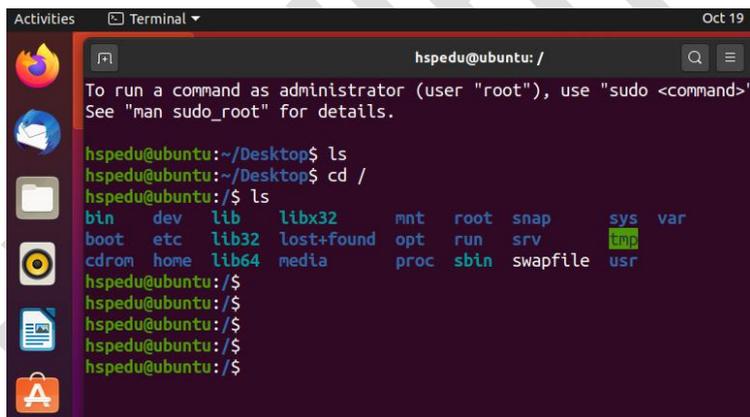
温馨提示:

Ubuntu 和 Centos 都是基于 GNU/Linux 内核的, 因此基本使用和 Centos 是几乎一样的, 它们的各种指令可以通用, 同学们在学习和使用 Ubuntu 的过程中, 会发现各种操作指令在前面学习 CentOS 都使用过。只是界面和预安装的软件有所差别。

Ubuntu 下载地址: <http://cn.ubuntu.com/download/>



18.2 安装的步骤

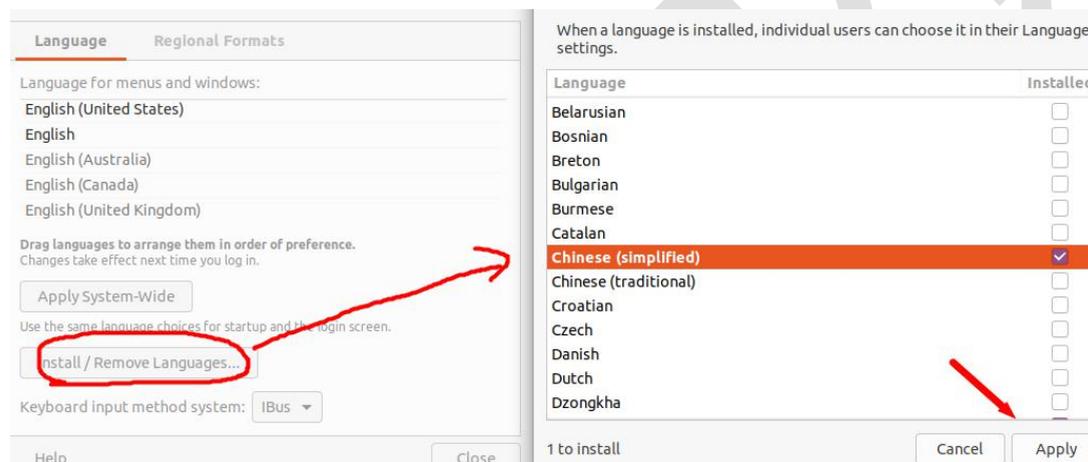


✓ 看老师笔记即可。

18.3 设置 Ubuntu 支持中文

默认安装的 ubuntu 中只有英文语言，因此是不能显示汉字的。要正确显示汉字，需要安装中文语言包。安装中文支持步骤：

- 1) 单击左侧图标栏打开 Language Support 菜单，点击打开 Language Support（语言支持）选项卡。
- 2) 点击 Install / Remove Languages，在弹出的选项卡中下拉找到 Chinese(Simplified)，即中文简体，在后面的选项框中打勾。然后点击 Apply Changes 提交，系统会自动联网下载中文语言包。（保证 ubuntu 是联网的）。
- 3) 这时“汉语（中国）”在最后一位因为当前第一位是”English”，所以默认显示都是英文。我们如果希望默认显示用中文，则应该将“汉语（中国）”设置为第一位。设置方法是拖动，鼠标单击“汉语（中国）”，当底色变化（表示选中了）后，按住鼠标左键不松手，向上拖动放置到第一位。
- 4) 设置后不会即刻生效，需要下一次登录时才会生效



18.4 Ubuntu 的 root 用户

18.4.1 介绍

安装 ubuntu 成功后，都是普通用户权限，并没有最高 root 权限，如果需要使用 root 权限的时候，通常都会在命令前面加上 `sudo`。有的时候感觉很麻烦。(演示)

我们一般使用 `su` 命令来直接切换到 root 用户的，但是如果没有给 root 设置初始密码，就会抛出 `su: Authentication failure` 这样的问题。所以，我们只要给 root 用户设置一个初始密码就好了。

18.4.2 给 root 用户设置密码并使用

- 1) 输入 `sudo passwd` 命令，设定 root 用户密码。
- 2) 设定 root 密码成功后，输入 `su` 命令，并输入刚才设定的 root 密码，就可以切换到 root 了。提示符 `$` 代表一般用户，提示符 `#` 代表 root 用户。
- 3) 以后就可以使用 root 用户了
- 4) 输入 `exit` 命令，退出 root 并返回一般用户

18.5 Ubuntu 下开发 Python

18.5.1 说明

安装好 Ubuntu 后，默认就已经安装好 Python 的开发环境。

```
hspedu@ubuntu:~/Desktop$ python3
Python 3.8.2 (default, Jul 16 2020, 14:00:26)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello,python")
hello,python
>>> quit()
```

18.5.2 在 Ubuntu 下开发一个 Python 程序

```
vi hello.py      [编写 hello.py]
```

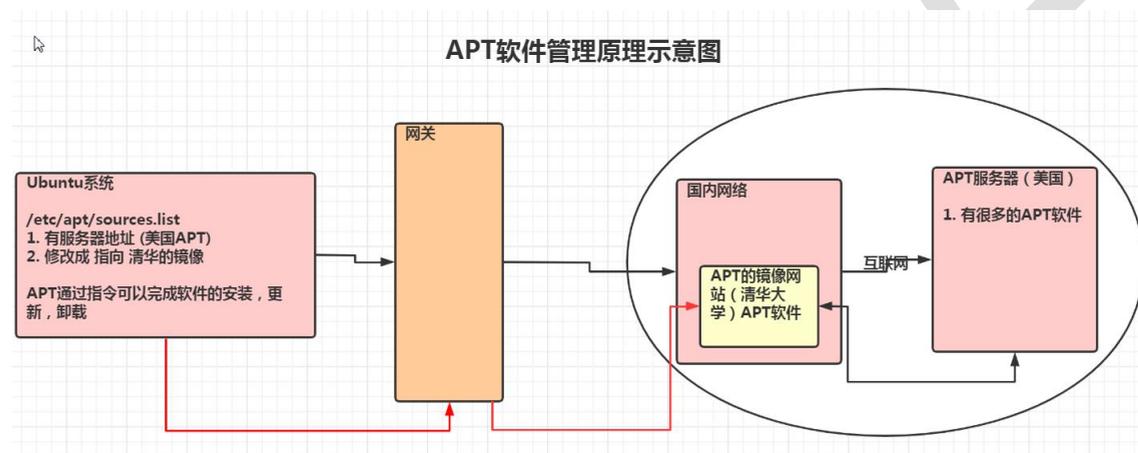
```
python3 hello.py [运行 hello.py]
```

第 19 章 Linux 之 Python 定制篇-APT 软件管理和远程登录

19.1 apt 介绍

apt 是 Advanced Packaging Tool 的简称，是一款安装包管理工具。在 Ubuntu 下，我们可以使用 apt 命令进行软件包的安装、删除、清理等，类似于 Windows 中的软件管理工具。

ubuntu 软件管理的原理示意图：



19.2 Ubuntu 软件操作的相关命令

`sudo apt-get update` 更新源

`sudo apt-get install package` 安装包

`sudo apt-get remove package` 删除包

`sudo apt-cache search package` 搜索软件包

`sudo apt-cache show package` 获取包的相关信息，如说明、大小、版本等

`sudo apt-get install package --reinstall` 重新安装包

`sudo apt-get -f install` 修复安装

`sudo apt-get remove package --purge` 删除包，包括配置文件等

`sudo apt-get build-dep package` 安装相关的编译环境

`sudo apt-get upgrade` 更新已安装的包

`sudo apt-get dist-upgrade` 升级系统

`sudo apt-cache depends package` 了解使用该包依赖那些包

`sudo apt-cache rdepends package` 查看该包被哪些包依赖

`sudo apt-get source package` 下载该包的源代码

19.3 更新 Ubuntu 软件下载地址

19.3.1 原理介绍

(画出示意图)

19.3.2 寻找国内镜像源

<https://mirrors.tuna.tsinghua.edu.cn/>

所谓的镜像源：可以理解为提供下载软件的地方，比如 Android 手机上可以下载软件的安卓市场；iOS 手机上可以下载软件的 AppStore



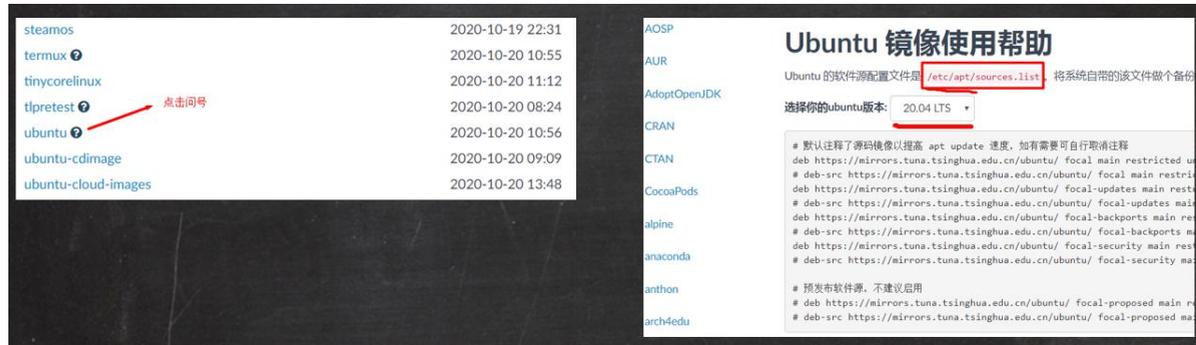
清华大学开源软件镜像站

HOMI

镜像列表

Name	Last Update
AOSP ?	2020-10-20 11:10
AUR ?	0000-00-00 00:00
AdoptOpenJDK ?	2020-10-20 09:02
CPAN	2020-10-20 13:34
CRAN ?	2020-10-20 12:23
CTAN ?	2020-10-20 08:11

19.3.3 寻找国内镜像源



19.3.4 备份 Ubuntu 默认的源地址

```
sudo cp /etc/apt/sources.list /etc/apt/sources.list.backup
```

```
hspedu@ubuntu: /etc/apt
hspedu@ubuntu:~/Desktop$ sudo cp /etc/apt/sources.list /etc/apt/sources.list.backup
[sudo] password for hspedu:
hspedu@ubuntu:~/Desktop$ cd /etc/apt/
hspedu@ubuntu:/etc/apt$ ls
apt.conf.d  preferences.d  sources.list  sources.list.d
auth.conf.d  sources.bak  sources.list.backup  trusted.gpg.d
hspedu@ubuntu:/etc/apt$ cat sources.list
#deb cdrom:[Ubuntu 20.04.1 LTS _Focal Fossa_ - Release amd64 (20200731)]/ focal main r
restricted

# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://us.archive.ubuntu.com/ubuntu/ focal main restricted
# deb-src http://us.archive.ubuntu.com/ubuntu/ focal main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://us.archive.ubuntu.com/ubuntu/ focal-updates main restricted
# deb-src http://us.archive.ubuntu.com/ubuntu/ focal-updates main restricted

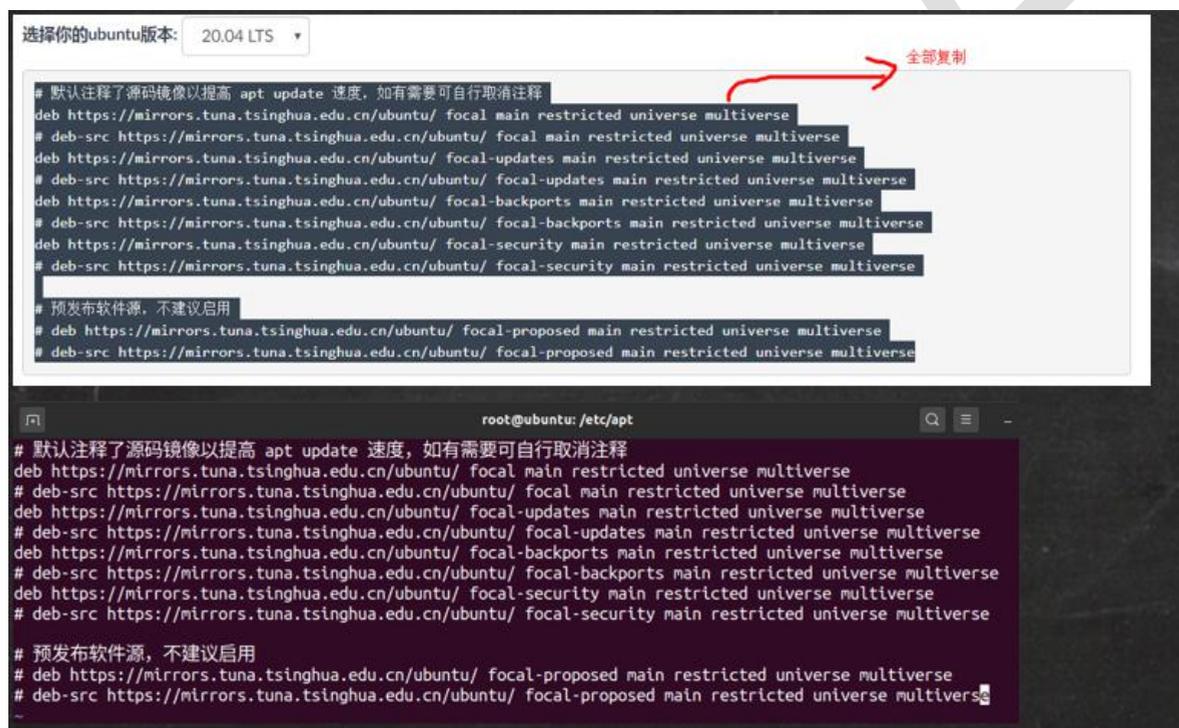
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://us.archive.ubuntu.com/ubuntu/ focal universe
# deb-src http://us.archive.ubuntu.com/ubuntu/ focal universe
deb http://us.archive.ubuntu.com/ubuntu/ focal-updates universe
# deb-src http://us.archive.ubuntu.com/ubuntu/ focal-updates universe
```

19.3.5 更新源服务器列表

先清空 sources.list 文件复制镜像网站的地址

```
hspedu@ubuntu:/etc/apt$ su
Password:
root@ubuntu:/etc/apt# echo '' > sources.list
root@ubuntu:/etc/apt#
```

复制镜像网站的地址， 拷贝到 sources.list 文件



```
选择你的ubuntu版本: 20.04 LTS
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe multiverse

# 预发布软件源，不建议启用
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-proposed main restricted universe multiverse

root@ubuntu: /etc/apt
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe multiverse

# 预发布软件源，不建议启用
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-proposed main restricted universe multiverse
```

19.3.6 更新源

更新源地址: `sudo apt-get update`

```
root@ubuntu:/etc/apt#  
root@ubuntu:/etc/apt# sudo apt-get update  
Get:1 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal InRelease [265 kB]  
Get:2 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates InRelease [111 kB]  
Get:3 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-backports InRelease [98.3 kB]  
Get:4 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-security InRelease [107 kB]  
Get:5 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal/main i386 Packages [718 kB]  
Get:6 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal/main amd64 Packages [970 kB]  
Get:7 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal/main Translation-en [506 kB]  
Get:8 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal/main amd64 DEP-11 Metadata [494 kB]  
Get:9 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal/main DEP-11 48x48 Icons [98.4 kB]  
Get:10 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal/main DEP-11 64x64 Icons [163 kB]  
Get:11 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal/main DEP-11 64x64@2 Icons [15.8 kB]  
Get:12 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal/main amd64 c-n-f Metadata [29.5 kB]  
Get:13 https://mirrors.tuna.tsinghua.edu.cn/ubuntu focal/restricted amd64 Packages [22.0 kB]
```

19.4 Ubuntu 软件安装，卸载的最佳实践

案例说明：使用 apt 完成安装和卸载 vim 软件，并查询 vim 软件的信息：（因为使用了镜像网站，速度很快）

```
sudo apt-get remove vim //删除
```

```
sudo apt-get install vim //安装
```

```
sudo apt-cache show vim //获取软件信息
```

19.5 远程登录 Ubuntu

19.5.1 ssh 介绍

SSH 为 Secure Shell 的缩写，由 IETF 的网络工作小组（Network Working Group）所制定；SSH 为建立在应用层和传输层基础上的安全协议。

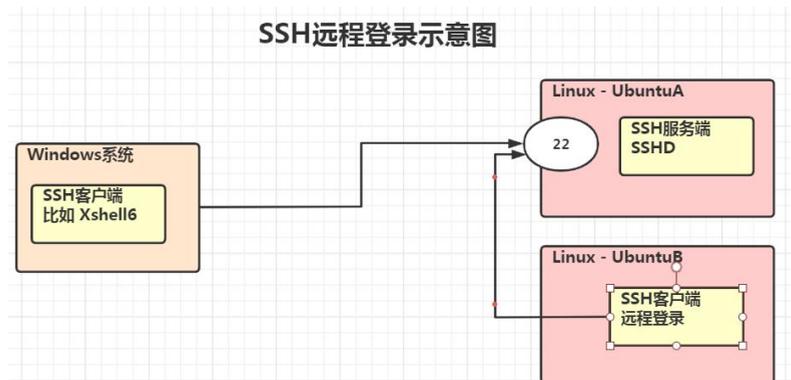
SSH 是目前较可靠，专为远程登录会话和其他网络服务提供安全性的协议。常用于远程登录。几乎所有 UNIX/Linux 平台都可运行 SSH。

使用 SSH 服务，需要安装相应的服务器和客户端。客户端和服务器的关系：如果，A 机器想被 B 机器远程控制，那么，A 机器需要安装 SSH 服务器，B 机器需要安装 SSH 客户端。

和 CentOS 不一样，Ubuntu 默认没有安装 SSHD 服务(使用 netstat 指令查看: apt install net-tools)，因此，我们不能进

行远程登录。

19.5.2 原理示意图



19.5.3 安装 SSH 和启用

```
sudo apt-get install openssh-server
```

执行上面指令后，在当前这台 Linux 上就安装了 SSH 服务端和客户端。

```
service sshd restart
```

执行上面的指令，就启动了 sshd 服务。会监听端口 22

19.5.4 在 Windows 使用 XShell6/XFTP6 登录 Ubuntu

前面我们已经安装了 XShell6，直接使用即可。

注意：使用 hspEdu 用户登录，需要的时候再 su - 切换到 root 用户

19.5.5 从一台 linux 系统远程登陆另外一台 linux 系统

在创建服务器集群时，会使用到该技术

✓ 基本语法：

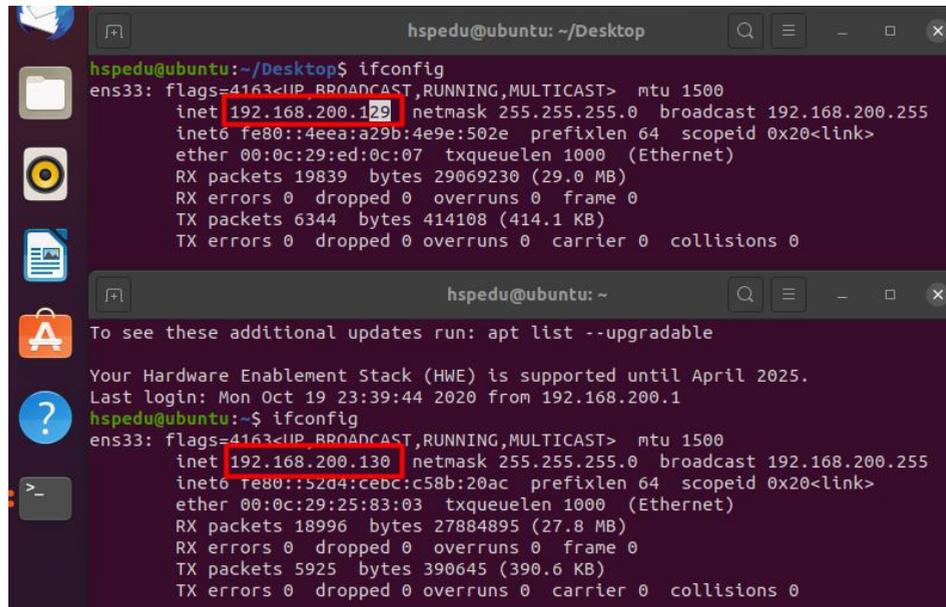
```
ssh 用户名@IP
```

例如：ssh hspedu@192.168.200.130

使用 ssh 访问，如访问出现错误。可查看是否有该文件 `~/.ssh/known_ssh` 尝试删除该文件解决，一般不会有问题

✓ 登出

登出命令: exit 或者 logout



```
hspedu@ubuntu: ~/Desktop
hspedu@ubuntu:~/Desktop$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.200.129 netmask 255.255.255.0 broadcast 192.168.200.255
    inet6 fe80::4eea:a29b:4e9e:502e prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ed:0c:07 txqueuelen 1000 (Ethernet)
    RX packets 19839 bytes 29069230 (29.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6344 bytes 414108 (414.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

hspedu@ubuntu: ~
To see these additional updates run: apt list --upgradable
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Mon Oct 19 23:39:44 2020 from 192.168.200.1
hspedu@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.200.130 netmask 255.255.255.0 broadcast 192.168.200.255
    inet6 fe80::52d4:ceb:c58b:20ac prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:25:83:03 txqueuelen 1000 (Ethernet)
    RX packets 18996 bytes 27884895 (27.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5925 bytes 390645 (390.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

第 20 章 CentOS8.1 的使用

20.1 安装 Centos8.1/8.2

20.1.1 Centos 下载地址

CentOS-8.1.1911-x86_64-dvd1.iso CentOS 8.1/8.2 DVD 版 8G (未来的主流.)

https://mirrors.aliyun.com/centos/8.1.1911/isos/x86_64/

Index of /centos/8.1.1911/isos/x86_64/

../		
CHECKSUM	03-Jan-2020 21:50	319
CHECKSUM.asc	14-Jan-2020 14:22	1179
CentOS-8.1.1911-x86_64-boot.iso	03-Jan-2020 21:30	625999872
CentOS-8.1.1911-x86_64-boot.iso.manifest	03-Jan-2020 21:30	626
CentOS-8.1.1911-x86_64-boot.torrent	14-Jan-2020 15:32	24447
CentOS-8.1.1911-x86_64-dvd1.iso	03-Jan-2020 21:47	7554990080
CentOS-8.1.1911-x86_64-dvd1.iso.manifest	03-Jan-2020 21:47	417936
CentOS-8.1.1911-x86_64-dvd1.torrent	14-Jan-2020 15:32	288772

20.1.2 CentOS8.1 安装的步骤

创建虚拟机 [演示]

开始安装系统(CentOS8.1)的步骤 [演示]

具体看老师演示和文档即可

20.2 CentOS8.0 VS CentOS7.0 比较 (了解)

功能	CentOS 8.0	CentOS 7.0
内核版本	4.18.0-x	3.10.0-x
文件系统	XFS	XFS
时间同步	只支持Chronyd	支持NTP和Chronyd
文件大小	8EB	500TB
文件系统大小	1PB	500TB
包管理工具	DNF (YUM v4)	YUM (YUM v3)
最大内存	24TB	12TB
防火墙	nftables取代iptables	firewalld底层使用iptables
支持架构	支持64-bit ARM	不支持64-bit ARM
网络管理	默认安装Cockpit	无默认安装
CPU日志	CPU日志记录在journald进程中	CPU相关日志位于/var/log/cups目录
发布时间	2019-09-24	2014-07-07
完整更新	2024-05-01	2020-08-06
支持IPv6	支持	Ready Logo Phase 2
Git	2.18	1.7
GCC	8.2.1	4.8.5
Python	3.6.6	2.7.5
Glibc	2.28	2.17
Boost	1.66	1.53
LLVM/Clang	6.0	5.0

第 21 章 Linux 高级篇-日志管理

21.1 基本介绍

- 1) 日志文件是重要的系统信息文件，其中记录了许多重要的系统事件，包括用户的登录信息、系统的启动信息、系统的安全信息、邮件相关信息、各种服务相关信息等。
- 2) 日志对于安全来说也很重要，它记录了系统每天发生的各种事情，通过日志来检查错误发生的原因，或者受到攻击时攻击者留下的痕迹。
- 3) 可以这样理解 日志是用来记录重大事件的工具

21.2 系统常用的日志

- ✓ /var/log/ 目录就是系统日志文件的保存位置，看张图

```
[root@hspedu100 ~]# ls /var/log/
anaconda          chrony             hspsshd            ppp                 sssd                vmware-network.6.log  Xorg.0.log
audit             cron               lastlog            qemu-ga             swtpm               vmware-network.7.log  Xorg.0.log.old
boot.log          cron-20201018     libvirt            rhsm                 tallylog            vmware-network.8.log  Xorg.9.log
boot.log-20201016 cups              maillog            sa                   tuned                vmware-network.9.log  yum.log
boot.log-20201018 dmesg             maillog-20201018  samba                vmware-install.log   vmware-network.log
boot.log-20201019 dmesg.old         messages           secure                vmware-network.1.log vmware-vgauthsvc.log.0
boot.log-20201020 firewallld        messages-20201018 secure-20201018     vmware-network.2.log vmware-vmvc.log
boot.log-20201021 gdm               mysqlld.log        speech-dispatcher  vmware-network.3.log vmware-vmusr.log
boot.log-20201022 glusterfs         ntpstats           spooler              vmware-network.4.log wpa_supplicant.log
btmtp             grubby_prune_debug pluto               spooler-20201018  vmware-network.5.log wttmp
```

- ✓ 系统常用的日志

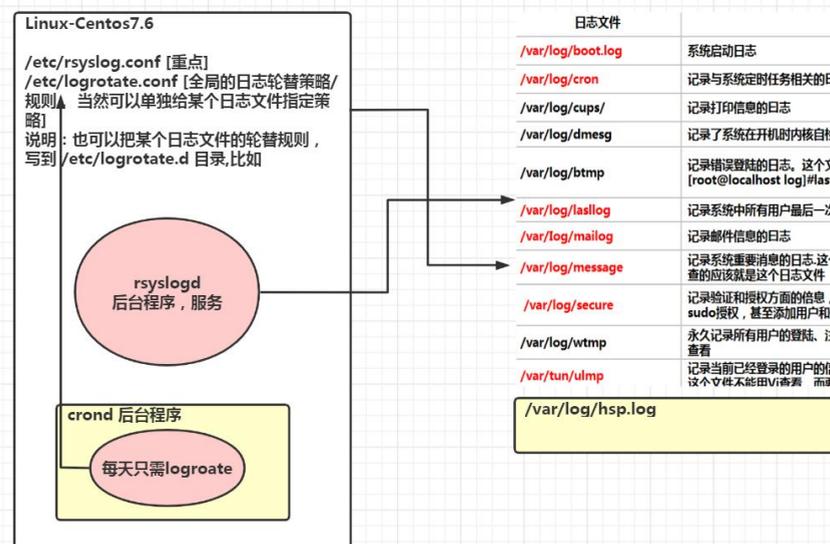
日志文件	说明
/var/log/boot.log	系统启动日志
/var/log/cron	记录与系统定时任务相关的日志
/var/log/cups/	记录打印信息的日志
/var/log/dmesg	记录了系统在开机时内核自检的信息。也可以使用dmesg命令直接查看内核自检信息
/var/log/btmp	记录错误登陆的日志。这个文件是二进制文件，不能直接用Vi查看，而要使用lastb命令查看。命令如下： [root@localhost log]#lastb
/var/log/lastlog	记录系统中所有用户最后一次的登录时间的日志。这个文件也是二进制文件.要使用lastlog命令查看
/var/log/maillog	记录邮件信息的日志
/var/log/message	记录系统重要消息的日志.这个日志文件中会记录Linux系统的绝大多数重要信息。如果系统出现问题，首先要检查的应该就是这个日志文件
/var/log/secure	记录验证和授权方面的信息，只要涉及账户和密码的程序都会记录，比如系统的登录、ssh的登录、su切换用户，sudo授权，甚至添加用户和修改用户密码都会记录在这个日志文件中
/var/log/wtmp	永久记录所有用户的登陆、注销信息，同时记录系统的后动、重启、关机事件。是二进制文件.而要使用last命令查看
/var/tun/ulmp	记录当前已经登录的用户的信息。这个文件会随着用户的登录和注销而不断变化，只记录当前登录用户的信息。这个文件不能用Vi查看，而要使用w、who、users等命令查看

✓ 应用案例

使用 root 用户通过 xshell6 登陆，第一次使用错误的密码，第二次使用正确的密码登录成功
看看在日志文件/var/log/secure 里有没有记录相关信息

21.3 日志管理服务 rsyslogd

CentOS7.6 日志服务是 rsyslogd ， CentOS6.x 日志服务是 syslogd 。rsyslogd 功能更强大。rsyslogd 的使用、日志文件的格式，和 syslogd 服务兼容的。原理示意图



✓ 查询 Linux 中的 rsyslogd 服务是否启动

```
ps aux | grep "rsyslog" | grep -v "grep"
```

✓ 查询 rsyslogd 服务的自启动状态

```
systemctl list-unit-files | grep rsyslog
```

✓ 配置文件：/etc/rsyslog.conf

编辑文件时的格式为： *.* 存放日志文件

其中第一个*代表日志类型，第二个*代表日志级别

1) 日志类型分为：

auth ##pam 产生的日志

authpriv ##ssh、ftp 等登录信息的验证信息
corn ##时间任务相关
kern ##内核
lpr ##打印
mail ##邮件
mark(syslog)-rsyslog##服务内部的信息，时间标识
news ##新闻组
user ##用户程序产生的相关信息
uucp ##unix to nuix copy 主机之间相关的通信
local 1-7 ##自定义的日志设备

2) 日志级别分为:

debug ##有调试信息的，日志通信最多
info ##一般信息日志，最常用
notice ##最具有重要性的普通条件的信息
warning ##警告级别
err ##错误级别，阻止某个功能或者模块不能正常工作的信息
crit ##严重级别，阻止整个系统或者整个软件不能正常工作的信息
alert ##需要立刻修改的信息
emerg ##内核崩溃等重要信息
none ##什么都不记录

注意：从上到下，级别从低到高，记录信息越来越少

✓ 由日志服务 rsyslogd 记录的日志文件，日志文件的格式包含以下 4 列：

- 1) 事件产生的时间
- 2) 产生事件的服务器的主机名
- 3) 产生事件的服务名或程序名
- 4) 事件的具体信息

✓ 日志如何查看实例

查看一下 `/var/log/secure` 日志，这个日志中记录的是用户验证和授权方面的信息 来分析如何查看

```
Nov 12 12:18:26 hspEdu100 unix_chkpwd[10480]: password check failed for user (root)
Nov 12 12:18:26 hspEdu100 sssd[10478]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0
rhost=thinkpad-pc user=root
Nov 12 12:18:26 hspEdu100 sshd[10478]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met by us
Nov 12 12:18:27 hspEdu100 sshd[10478]: Failed password for root from 192.168.200.1 port 8967 ssh2
Nov 12 12:18:33 hspEdu100 unix_chkpwd[10481]: password check failed for user (root)
Nov 12 12:18:33 hspEdu100 sshd[10478]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met by us
Nov 12 12:18:34 hspEdu100 sshd[10478]: Failed password for root from 192.168.200.1 port 8967 ssh2
Nov 12 12:18:37 hspEdu100 sshd[10478]: Accepted password for root from 192.168.200.1 port 8967 ssh2
Nov 12 12:18:37 hspEdu100 sshd[10478]: pam_unix(sshd:session): session opened for user root by (uid=0)
```

✓ 日志管理服务应用实例

在`/etc/rsyslog.conf` 中添加一个日志文件`/var/log/hsp.log`,当有事件发送时(比如 `sshd` 服务相关事件), 该文件会接收到信息并保存. 给小伙伴演示 重启, 登录 的情况, 看看是否有日志保存

```
root@hspedu100 ~]# cat /var/log/hsp.log | grep sshd
Oct 24 11:15:42 hspedu100 sshd[8252]: Server listening on 0.0.0.0 port 22.
Oct 24 11:15:42 hspedu100 sshd[8252]: Server listening on :: port 22.
Oct 24 11:16:45 hspedu100 sshd[9370]: Accepted password for root from 192.168.200.1 port 56395 ssh2
Oct 24 11:16:45 hspedu100 sshd[9370]: pam_unix(sshd:session): session opened for user root by (uid=0)
root@hspedu100 ~]# cat /var/log/hsp.log | grep sshd
Oct 24 11:15:42 hspedu100 sshd[8252]: Server listening on 0.0.0.0 port 22.
Oct 24 11:15:42 hspedu100 sshd[8252]: Server listening on :: port 22.
Oct 24 11:16:45 hspedu100 sshd[9370]: Accepted password for root from 192.168.200.1 port 56395 ssh2
Oct 24 11:16:45 hspedu100 sshd[9370]: pam_unix(sshd:session): session opened for user root by (uid=0)
Oct 24 11:20:09 hspedu100 sshd[9477]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0
Oct 24 11:20:09 hspedu100 sshd[9477]: pam_succeed_if(sshd:auth): requirement "uid >= 1000" not met by
Oct 24 11:20:11 hspedu100 sshd[9477]: Failed password for root from 192.168.200.1 port 56454 ssh2
Oct 24 11:20:14 hspedu100 sshd[9477]: Accepted password for root from 192.168.200.1 port 56454 ssh2
Oct 24 11:20:14 hspedu100 sshd[9477]: pam_unix(sshd:session): session opened for user root by (uid=0)
```

21.4 日志轮替

21.4.1 基本介绍

日志轮替就是把旧的日志文件移动并改名, 同时建立新的空日志文件, 当旧日志文件超出保存的范围之后, 就会进行删除

21.4.2 日志轮替文件命名

- 1) centos7 使用 `logrotate` 进行日志轮替管理, 要想改变日志轮替文件名字, 通过 `/etc/logrotate.conf` 配置文件中“`dateext`”参数:

- 2) 如果配置文件中有“dateext”参数，那么日志会用日期来作为日志文件的后缀，例如“secure-20201010”。这样日志文件名不会重叠，也就不需要日志文件的改名，只需要指定保存日志个数，删除多余的日志文件即可。
- 3) 如果配置文件中没有“dateext”参数，日志文件就需要进行改名了。当第一次进行日志轮替时，当前的“secure”日志会自动改名为“secure.1”，然后新建“secure”日志，用来保存新的日志。当第二次进行日志轮替时，“secure.1”会自动改名为“secure.2”，当前的“secure”日志会自动改名为“secure.1”，然后也会新建“secure”日志，用来保存新的日志，以此类推。

21.4.3 logrotate 配置文件

✓ /etc/logrotate.conf 为 logrotate 的全局配置文件

```
# rotate log files weekly, 每周对日志文件进行一次轮替
```

```
weekly
```

```
# keep 4 weeks worth of backlogs, 共保存 4 份日志文件，当建立新的日志文件时，旧的将会被删除
```

```
rotate 4
```

```
# create new (empty) log files after rotating old ones, 创建新的空的日志文件，在日志轮替后
```

```
create
```

```
# use date as a suffix of the rotated file, 使用日期作为日志轮替文件的后缀
```

```
dateext
```

```
# uncomment this if you want your log files compressed, 日志文件是否压缩。如果取消注释，则日志会在转储的同时进行压缩
```

```
#compress
```

```
#RPM packages drop log rotation information into this directory
```

```
include /etc/logrotate.d
```

```
# 包含 /etc/logrotate.d/ 目录中所有的子配置文件。也就是说会把这个目录中所有子配置文件读取进来，
```

```
#下面是单独设置，优先级更高。
```

```
# no packages own wtmp and btmp -- we'll rotate them here
```

```
/var/log/wtmp {
```

```
    monthly # 每月对日志文件进行一次轮替
```

```

create 0664 root utmp # 建立的新日志文件，权限是 0664 ，所有者是 root ，所属组是 utmp 组
minsize 1M # 日志文件最小轮替大小是 1MB 。也就是日志一定要超过 1MB 才会轮替，否则就算时间达到
一个月，也不进行日志转储
rotate 1 # 仅保留一个日志备份。也就是只有 wtmp 和 wtmp.1 日志保留而已
}
/var/log/btmp {
missingok # 如果日志不存在，则忽略该日志的警告信息
monthly
create 0600 root utmp
rotate 1
}

```

✓ 参数说明

参 数	参 数 说 明
daily	日志的轮替周期是每天
weekly	日志的轮替周期是每周
monthly	日志的轮替周期是每月
rotate 数字	保留的日志文件的个数。0 指没有备份
compress	日志轮替时，旧的日志进行压缩
create mode owner group	建立新日志，同时指定新日志的权限与所有者和所属组。
mail address	当日志轮替时，输出内容通过邮件发送到指定的邮件地址。
missingok	如果日志不存在，则忽略该日志的警告信息
notifempty	如果日志为空文件，则不进行日志轮替
minsize 大小	日志轮替的最小值。也就是日志一定要达到这个最小值才会轮替，否则就算时间达到也不轮替
size 大小	日志只有大于指定大小才进行日志轮替，而不是按照时间轮替。
dateext	使用日期作为日志轮替文件的后缀。

sharedscripts 在此关键字之后的脚本只执行一次。

prerotate/endscript 在日志轮替之前执行脚本命令。

postrotate/endscript 在日志轮替之后执行脚本命令。

21.4.4 把自己的日志加入日志轮替

- 1) 第一种方法是直接在/etc/logrotate.conf 配置文件中写入该日志的轮替策略
- 2) 第二种方法是在/etc/logrotate.d/目录中新建立该日志的轮替文件，在该轮替文件中写入正确的轮替策略，因为该目录中的文件都会被“include”到主配置文件中，所以也可以把日志加入轮替。
- 3) **推荐使用第二种方法**，因为系统中需要轮替的日志非常多，如果全都直接写入/etc/logrotate.conf 配置文件，那么这个文件的可管理性就会非常差，不利于此文件的维护。
- 4) 在/etc/logrotate.d/ 配置轮替文件一览

```
[root@hspedu100 logrotate.d]# ls
bootlog  dnf      iscsiuiolog  mysql  psacct  syslog
chrony   glusterfs libvirtd     numad  samba   wpa_supplicant
cups     hsplog   libvirtd.qemu  ppp    sssd    yum
[root@hspedu100 logrotate.d]#
```

21.4.5 应用实例

看一个案例，在/etc/logrotate.conf 进行配置，或者直接在 /etc/logrotate.d/ 下创建文件 hsplog 编写如下内容，具体轮替的效果可以参考 /var/log 下的 boot.log 情况。

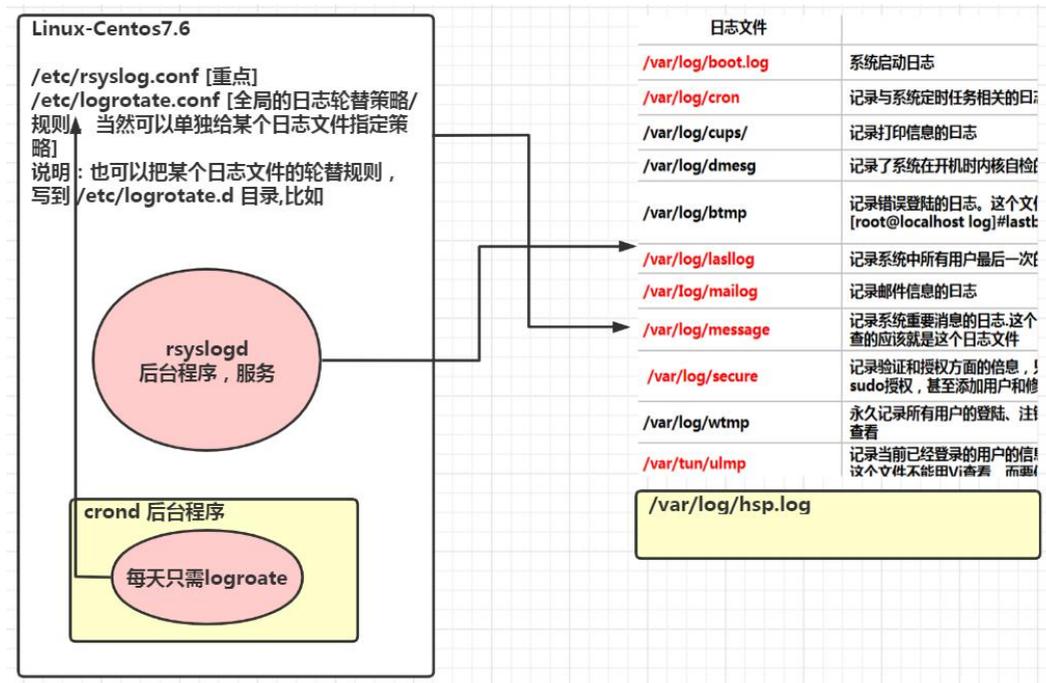
```
/var/log/hsp.log
{
    missingok
    daily
    copytruncate
    rotate 7
    notifempty
}
boot.log
boot.log-20201018
boot.log-20201019
boot.log-20201020
boot.log-20201021
boot.log-20201022
boot.log-20201023
boot.log-20201024
```

21.5 日志轮替机制原理

日志轮替之所以可以在指定的时间备份日志，是依赖系统定时任务。在 /etc/cron.daily/目录，就会发现这个目录中是

有 logrotate 文件(可执行), logrotate 通过这个文件依赖定时任务执行的。

```
[root@hspedu100 logrotate.d]# ls -l /etc/cron.daily/
总用量 12
-rwx-----. 1 root root 219 10月 31 2018 logrotate
-rwxr-xr-x. 1 root root 618 10月 30 2018 man-db.cron
-rwx-----. 1 root root 208 4月 11 2018 mlocate
[root@hspedu100 logrotate.d]#
```



21.6 查看内存日志

`journalctl` 可以查看内存日志, 这里我们看看常用的指令

`journalctl` ##查看全部

`journalctl -n 3` ##查看最新 3 条

`journalctl --since 19:00 --until 19:10:10` #查看起始时间到结束时间的日志可加日期

`journalctl -p err` ##报错日志

`journalctl -o verbose` ##日志详细内容

`journalctl _PID=1245 _COMM=sshd` ##查看包含这些参数的日志 (在详细日志查看)

或者 `journalctl | grep sshd`

注意: journalctl 查看的是内存日志, 重启清空

演示案例:

使用 journalctl | grep sshd 来看看用户登录清空, 重启系统, 再次查询, 看看日志有什么变化没有

第 22 章 Linux 高级篇-定制自己的 Linux 系统

22.1 基本介绍

通过裁剪现有 Linux 系统(CentOS7.6), 创建属于自己的 min Linux 小系统, 可以加深我们对 linux 的理解。

老韩利用 centos7.6, 搭建一个小小 linux 系统, 很有趣。

22.2 基本原理

启动流程介绍:

制作 Linux 小系统之前, 再了解一下 Linux 的启动流程:

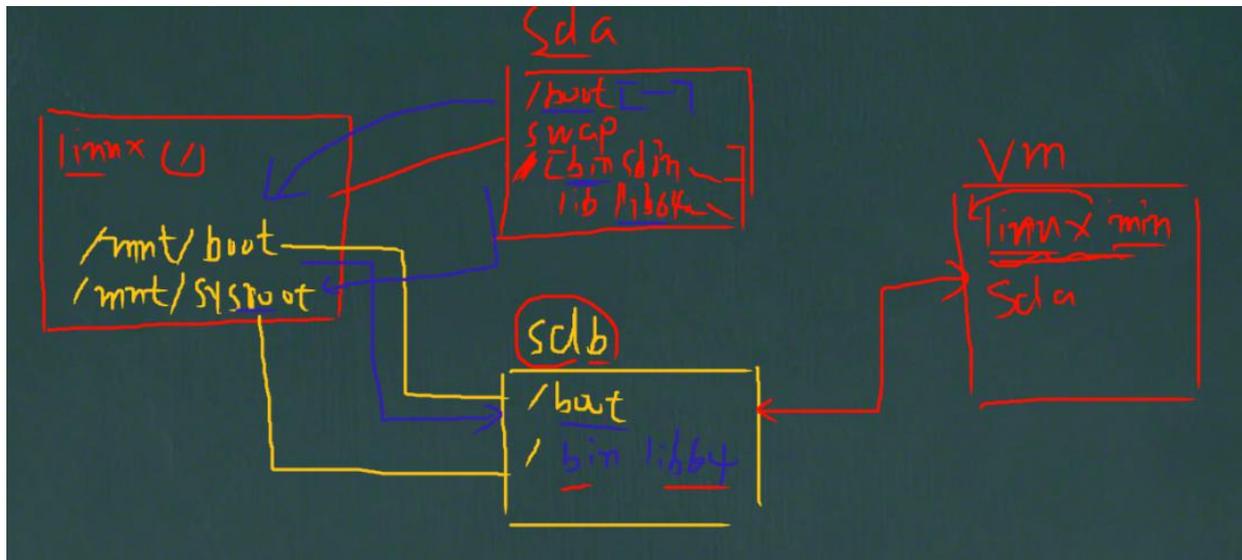
- 1、首先 Linux 要通过自检, 检查硬件设备有没有故障
- 2、如果有多块启动盘的话, 需要在 BIOS 中选择启动磁盘
- 3、启动 MBR 中的 bootloder 引导程序
- 4、加载内核文件
- 5、执行所有进程的父进程、老祖宗 systemd
- 6、欢迎界面

在 Linux 的启动流程中, 加载内核文件时关键文件:

- 1) kernel 文件: `vmlinuz-3.10.0-957.el7.x86_64`
- 2) initrd 文件: `initramfs-3.10.0-957.el7.x86_64.img`

22.3 制作 min linux 思路分析

- 1) 在现有的 Linux 系统(centos7.6)上加一块硬盘/dev/sdb, 在硬盘上分两个分区, 一个是/boot, 一个是/, 并将其格式化。
需要明确的是, 现在加的这个硬盘在现有的 Linux 系统中是/dev/sdb, 但是, 当我们把东西全部设置好时, 要把这个硬盘拔除, 放在新系统上, 此时, 就是/dev/sda
- 2) 在/dev/sdb 硬盘上, 将其打造成独立的 Linux 系统, 里面的所有文件是需要拷贝进去的
- 3) 作为能独立运行的 Linux 系统, 内核是一定不能少, 要把内核文件和 initramfs 文件也一起拷到/dev/sdb 上
- 4) 以上步骤完成, 我们的自制 Linux 就完成, 创建一个新的 linux 虚拟机, 将其硬盘指向我们创建的硬盘, 启动即可
- 5) 示意图



22.4 操作步骤

操作步骤

具体看文档+老师演示

第 23 章 Linux 内核源码介绍&内核升级

23.1 为什么要阅读 linux 内核?

- 1) 爱好, 就是喜欢 linux(黑客精神)
 - 2) 想深入理解 linux 底层运行机制, 对操作系统有深入理解
 - 3) 阅读 Linux 内核, 你会对整个计算机体系有一个更深刻的认识。作为开发者, 不管你从事的是驱动开发, 应用开发还是后台开发, 你都需要了解操作系统内核的运行机制, 这样才能写出更好的代码。
 - 4) 作为开发人员不应该只局限在自己的领域, 你设计的模块看起来小, 但是你不了解进程的调用机制, 你不知道进程为什么会阻塞、就绪、执行几个状态。那么很难写出优质的代码。
 - 5) 找工作面试的需要
- ✓ 老韩忠告, 作为有追求的程序员, 还是应该深入的了解一个操作系统的底层机制,(比如 linux/unix) 最好是源码级别的, 这样你写多线程高并发程序, 包括架构, 优化, 算法等, 高度不一样的, 当然老韩也不是要求小伙伴儿把一个非常庞大的 Linux 内核每一行都读懂。我觉得。你至少能看几个核心的模块。

23.2 linux0.01 内核源码

23.2.1 基本介绍

Linux 的内核源代码可以从网上下载, 解压缩后文件一般也都位于 linux 目录下。内核源代码有很多版本, 可以从 linux0.01 内核入手, 总共的代码 1w 行左右, 最新版本 5.9.8 总共代码超过 700w 行, 非常庞大。

内核地址: <https://www.kernel.org/>



很多人害怕读 Linux 内核，Linux 内核这样大而复杂的系统代码，阅读起来确实有很多困难，但是也不象想象的那么高不可攀。老韩建议可以从 linux0.01 入手。

23.2.2 linux0.01 内核源码目录&阅读

✓ 老韩提示阅读内核源码技巧

- 1) linux0.01 的阅读需要懂 c 语言
- 2) 阅读源码前，应知道 Linux 内核源码的整体分布情况。现代的操作系统一般由进程管理、内存管理、文件系统、驱动程序和网络等组成。Linux 内核源码的各个目录大致与此相对应。
- 3) 在阅读方法或顺序上，有纵向与横向之分。所谓纵向就是顺着程序的执行顺序逐步进行；所谓横向，就是按模块进行。它们经常结合在一起进行。
- 4) 对于 Linux 启动的代码可顺着 Linux 的启动顺序一步步来阅读；对于像内存管理部分，可以单独拿出来进行分析。实际上这是一个反复的过程，不可能读一遍就理解

✓ linux 内核源码阅读&目录介绍&main.c 说明

```

总用量 36
drwxr-xr-x. 2 root root 4096 11月 13 12:10 boot
drwxr-xr-x. 2 root root 4096 11月 13 12:10 fs
drwxr-xr-x. 5 root root 4096 11月 13 12:10 include
drwxr-xr-x. 2 root root 4096 11月 13 12:10 init
drwxr-xr-x. 2 root root 4096 11月 13 12:10 kernel
drwxr-xr-x. 2 root root 4096 11月 13 12:10 lib
-rw-r--r-- 1 root root 2186 11月 13 12:10 Makefile
drwxr-xr-x. 2 root root 4096 11月 13 12:10 mm
drwxr-xr-x. 2 root root 4096 11月 13 12:10 tools

```

Annotations:

- boot → 和系统引导相关的代码
- include → 存放linux支持的文件系统代码
- kernel → 存放linux核心需要的头文件，比如 asm, linux, sys
- lib → 和系统内核相关的源码
- mm → 存放库代码
- tools → 和内存管理相关的代码

```

void main(void) /* This really IS void, no error here. */
{
/*
/* The startup routine assumes (well, ...) this */
/*
/* Interrupts are still disabled. Do necessary setups, then
/* enable them
*/
time_init();           初始化运行时间
tty_init();           tty初始化
trap_init();          陷阱门(硬件中断向量)初始化
sched_init();         调度程序初始化
buffer_init();        缓冲管理初始化
hd_init();            硬盘初始化
sti();
move_to_user_mode();  进入用户模式
if (!fork()) { /* we count on this going ok */
    init();
}
/*
/* NOTE!! For any other task 'pause()' would mean we have to get a
/* signal to awaken, but task0 is the sole exception (see 'schedule()')
/* as task0 gets activated at every idle moment (when no other tasks
/* can run). For task0 'pause()' just means we go check if some other

```

23.3 linux 内核最新版和内核升级

23.3.1 内核地址：<https://www.kernel.org/> 查看



The Linux Kernel Archives

About Contact us FAQ Releases Signatures Site news

Protocol Location

HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

Latest Release
5.9.1 ↓

mainline:	5.9	2020-10-11	[tarball]	[pggp]	[patch]	[view diff]	[browse]
stable:	5.9.1	2020-10-17	[tarball]	[pggp]	[patch]	[view diff]	[browse] [changelog]
stable:	5.8.16	2020-10-17	[tarball]	[pggp]	[patch] [inc. patch]	[view diff]	[browse] [changelog]
longterm:	5.4.72	2020-10-17	[tarball]	[pggp]	[patch] [inc. patch]	[view diff]	[browse] [changelog]
longterm:	4.19.152	2020-10-17	[tarball]	[pggp]	[patch] [inc. patch]	[view diff]	[browse] [changelog]
longterm:	4.14.202	2020-10-17	[tarball]	[pggp]	[patch] [inc. patch]	[view diff]	[browse] [changelog]
longterm:	4.9.240	2020-10-17	[tarball]	[pggp]	[patch] [inc. patch]	[view diff]	[browse] [changelog]
longterm:	4.4.240	2020-10-17	[tarball]	[pggp]	[patch] [inc. patch]	[view diff]	[browse] [changelog]
linux-next:	next-20201023	2020-10-23					[browse]

23.3.2 下载&解压最新版

```
wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.8.16.tar.gz
```

```
tar -zxvf linux-5.8.16.tar.gz
```

23.3.3 linux 内核升级应用实例

将 Centos 系统从 7.6 内核升级到 7.8 版本内核(兼容性问题)

7.5-1804	x86-64	7.5	3.10.0-862	2018-05-10 ^[123]	2018-04-10 ^{[124][103]}	31
7.6-1810	x86-64	7.6	3.10.0-957	2018-12-03 ^[125]	2018-10-30 ^{[126][103]}	34
7.7-1908	x86-64	7.7	3.10.0-1062	2019-09-17 ^[127]	2019-08-06 ^{[128][103]}	42
7.8-2003	x86-64	7.8	3.10.0-1127	2020-04-27 ^[129]	2020-03-30 ^{[130][103]}	28
7.9-xxxx	x86-64	7.9	3.10.0-xxxx	TBA	TBA	?

23.3.4 具体步骤，看老师演示

`uname -a` // 查看当前的内核版本

`yum info kernel -q` //检测内核版本，显示可以升级的内核

`yum update kernel` //升级内核

`yum list kernel -q` //查看已经安装的内核

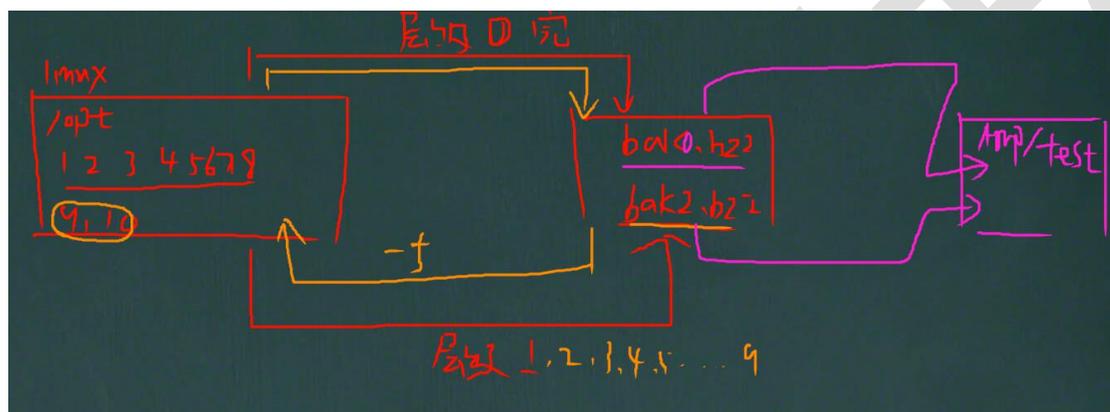
第 24 章 linux 系统-备份与恢复

24.1 基本介绍

实体机无法做快照，如果系统出现异常或者数据损坏，后果严重，要重做系统，还会造成数据丢失。所以我们可以使用备份和恢复技术

linux 的备份和恢复很简单，有两种方式：

- 1) 把需要的文件(或者分区)用 TAR 打包就行，下次需要恢复的时候，再解压开覆盖即可
- 2) 使用 dump 和 restore 命令
- 3) 示意图



24.2 安装 dump 和 restore

如果 linux 上没有 dump 和 restore 指令，需要先按照

```
yum -y install dump
```

```
yum -y install restore
```

24.3 使用 dump 完成备份

24.3.1 基本介绍

dump 支持分卷和增量备份（所谓增量备份是指备份上次备份后 修改/增加过的文件，也称差异备份）。

24.3.2 dump 语法说明

```
dump [-cu] [-123456789] [-f <备份后文件名>] [-T <日期>] [目录或文件系统]
```

```
dump [-wW]
```

-c : 创建新的归档文件, 并将由一个或多个文件参数所指定的内容写入归档文件的开头。

-0123456789: 备份的层级。0 为最完整备份, 会备份所有文件。若指定 0 以上的层级, 则备份至上一次备份以来修改或新增的文件, 到 9 后, 可以再次轮替。

-f <备份后文件名>: 指定备份后文件名

-j : 调用 bzip2 库压缩备份文件, 也就是将备份后的文件压缩成 bz2 格式, 让文件更小

-T <日期>: 指定开始备份的时间与日期

-u : 备份完毕后, 在/etc/dumpdates 中记录备份的文件系统, 层级, 日期与时间等。

-t : 指定文件名, 若该文件已存在备份文件中, 则列出名称

-W : 显示需要备份的文件及其最后一次备份的层级, 时间, 日期。

-w : 与-W 类似, 但仅显示需要备份的文件。

24.3.3 dump 应用案例 1

将/boot 分区所有内容备份到/opt/boot.bak0.bz2 文件中, 备份层级为“0”

```
dump -0uj -f /opt/boot.bak0.bz2 /boot
```

24.3.4 dump 应用案例 2

在/boot 目录下增加新文件, 备份层级为“1”(只备份上次使用层次“0”备份后发生过改变的数据), 注意比较看看这次生成的备份文件 boot1.bak 有多大

```
dump -1uj -f /opt/boot.bak1.bz2 /boot
```

老韩提醒: 通过 dump 命令在配合 crontab 可以实现无人值守备份

24.3.5 dump -W

显示需要备份的文件及其最后一次备份的层级，时间，日期

```
Last dump(s) done (Dump '>' file systems):
> /dev/sda3 ( / ) Last dump: never
/dev/sda1 ( /boot) Last dump: Level 1, Date Sat Oct 24 15:09:06 2020
```

24.3.6 查看备份时间文件

```
cat /etc/dumpdates
```

```
[root@hspedu100 opt]# cat /etc/dumpdates
/dev/sda1 0 Sat Oct 24 15:08:09 2020 +0800
/dev/sda1 1 Sat Oct 24 15:09:06 2020 +0800
```

24.3.7 dump 备份文件或者目录

前面我们在备份分区时，是可以支持增量备份的，如果备份文件或者目录，不再支持增量备份，即只能使用 0 级别备份

案例，使用 dump 备份 /etc 整个目录

```
dump -0j -f /opt/etc.bak.bz2 /etc/
```

#下面这条语句会报错，提示 DUMP: Only level 0 dumps are allowed on a subdirectory

```
dump -1j -f /opt/etc.bak.bz2 /etc/
```

24.3.8 老韩提醒

如果是重要的备份文件，比如数据区，建议将文件上传到其它服务器保存，不要将鸡蛋放在同一个篮子。

24.4 使用 restore 完成恢复

24.4.1 基本介绍

restore 命令用来恢复已备份的文件，可以从 dump 生成的备份文件中恢复原文件

24.4.2 restore 基本语法

```
restore [模式选项] [选项]
```

说明下面四个模式，不能混用，在一次命令中，只能指定一种。

- C : 使用对比模式，将备份的文件与已存在的文件相互对比。
- i: 使用交互模式，在进行还原操作时，restors 指令将依序询问用户
- r: 进行还原模式
- t: 查看模式，看备份文件有哪些文件

选项

-f <备份设备>: 从指定的文件中读取备份数据，进行还原操作

24.4.3 应用案例 1

restore 命令比较模式，比较备份文件和原文件的区别

测试

```
mv /boot/hello.java /boot/hello100.java
```

```
restore -C -f boot.bak1.bz2 //注意和 最新的文件比较
```

```
[root@hspEdu100 opt]# restore -C -f boot.bak1.bz2
Dump tape is compressed.
Dump date: Fri Nov 13 23:11:11 2020
Dumped from: Fri Nov 13 22:08:18 2020
Level 1 dump of /boot on hspEdu100:/dev/sda1
Label: none
filesys = /boot
restore: unable to stat ./hello.java: No such file or directory
Some files were modified! 1 compare errors
```

```
mv /boot/hello100.java /boot/hello.java
```

```
restore -C -f boot.bak1.bz2
```

```
[root@hspEdu100 opt]# restore -C -f boot.bak1.bz2
Dump tape is compressed.
Dump date: Fri Nov 13 23:11:11 2020
Dumped from: Fri Nov 13 22:08:18 2020
Level 1 dump of /boot on hspEdu100:/dev/sda1
Label: none
filesys = /boot
[root@hspEdu100 opt]#
```

24.4.4 应用案例 2

restore 命令查看模式，看备份文件有哪些数据/文件

测试

```
restore -t -f boot.bak0.bz2
```

24.4.5 应用案例 3

restore 命令还原模式, 注意细节: 如果你有增量备份, 需要把增量备份文件也进行恢复, 有几个增量备份文件, 就要恢复几个, 按顺序来恢复即可。

测试

```
mkdir /opt/boottmp
```

```
cd /opt/boottmp
```

```
restore -r -f /opt/boot.bak0.bz2 //恢复到第 1 次完全备份状态
```

```
restore -r -f /opt/boot.bak1.bz2 //恢复到第 2 次增量备份状态
```

```
[root@hspedu100 boottmp]# restore -r -f /opt/boot.bak.bz2
Dump tape is compressed.
[root@hspedu100 boottmp]# ls
config-3.10.0-957.el7.x86_64      initramfs-3.10.0-957.el7.x86_64.img
efi                               initramfs-3.10.0-957.el7.x86_64kdump.img
grub                              lost+found
grub2                             restoresymtable
```

24.4.6 应用案例 4

restore 命令恢复备份的文件, 或者整个目录的文件

基本语法: restore -r -f 备份好的文件

测试

```
[root@hspedu100 opt]# mkdir etctmp
```

```
[root@hspedu100 opt]# cd etctmp/
```

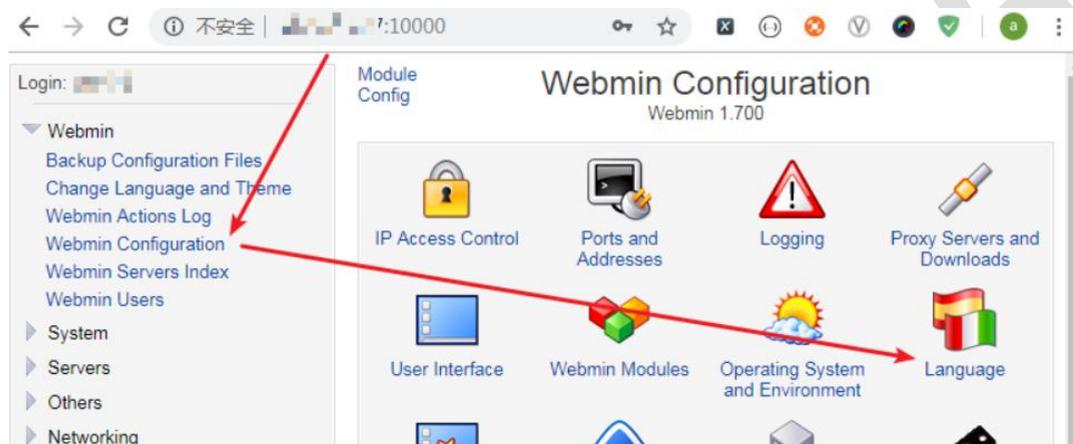
```
[root@hspedu100 etctmp]# restore -r -f /opt/etc.bak0.bz2
```

第 25 章 Linux 可视化管理-webmin 和 bt 运维工具

25.1 webmin

25.1.1 基本介绍

Webmin 是功能强大的基于 Web 的 Unix/linux 系统管理工具。管理员通过浏览器访问 Webmin 的各种管理功能并完成相应的管理操作。除了各版本的 linux 以外还可用于：AIX、HPUX、Solaris、Unixware、Irix 和 FreeBSD 等系统



25.1.2 安装 webmin&配置

1) 下载地址：<http://download.webmin.com/download/yum/>，用下载工具下载即可

Index of /download/yum

Name	Last modified	Size	Description
Parent Directory		-	
headers/	2018-07-16 23:48	-	
make-repo.sh	2018-11-16 16:28	67	
mirrorlist	2018-03-06 19:19	40	
repdata/	2020-10-19 19:36	-	
usermin-1.600-1.noar.>	2014-05-21 06:10	9.1M	

也可以使用 `wget http://download.webmin.com/download/yum/webmin-1.700-1.noarch.rpm`

2) 安装：`rpm -ivh webmin-1.700-1.noarch.rpm`

3) 重置密码 `/usr/libexec/webmin/changepass.pl /etc/webmin root test`

root 是 webmin 的用户名，不是 OS 的，这里就是把 webmin 的 root 用户密码改成了 test

4) 修改 webmin 服务的端口号（默认是 10000 出于安全目的）

```
vim /etc/webmin/miniserv.conf # 修改端口
```

5) 将 port=10000 修改为其他端口号，如 port=6666

6) 重启 webmin

```
/etc/webmin/restart # 重启
```

```
/etc/webmin/start # 启动
```

```
/etc/webmin/stop # 停止
```

7) 防火墙放开 6666 端口

```
firewall-cmd --zone=public --add-port=6666/tcp --permanent # 配置防火墙开放 6666 端口
```

```
firewall-cmd --reload # 更新防火墙配置
```

```
firewall-cmd --zone=public --list-ports # 查看已经开放的端口号
```

8) 登录 webmin

`http://ip:6666` 可以访问了

用 root 账号和重置的新密码 test

25.1.3 简单使用演示

比如修改 语言设置，IP 访问控制，查看进程，修改密码，任务调度，mysql 等。



25.2 bt(宝塔)

25.2.1 基本介绍

bt 宝塔 Linux 面板是提升运维效率的服务器管理软件，支持一键 LAMP/LNMP/集群/监控/网站/FTP/数据库/JAVA 等多项服务器管理功能。

25.2.2 安装和使用

- 1) 安装 : `yum install -y wget && wget -O install.sh http://download.bt.cn/install/install_6.0.sh && sh install.sh`
- 2) 安装成功后控制台会显示登录地址，账户密码，复制浏览器打开登录，

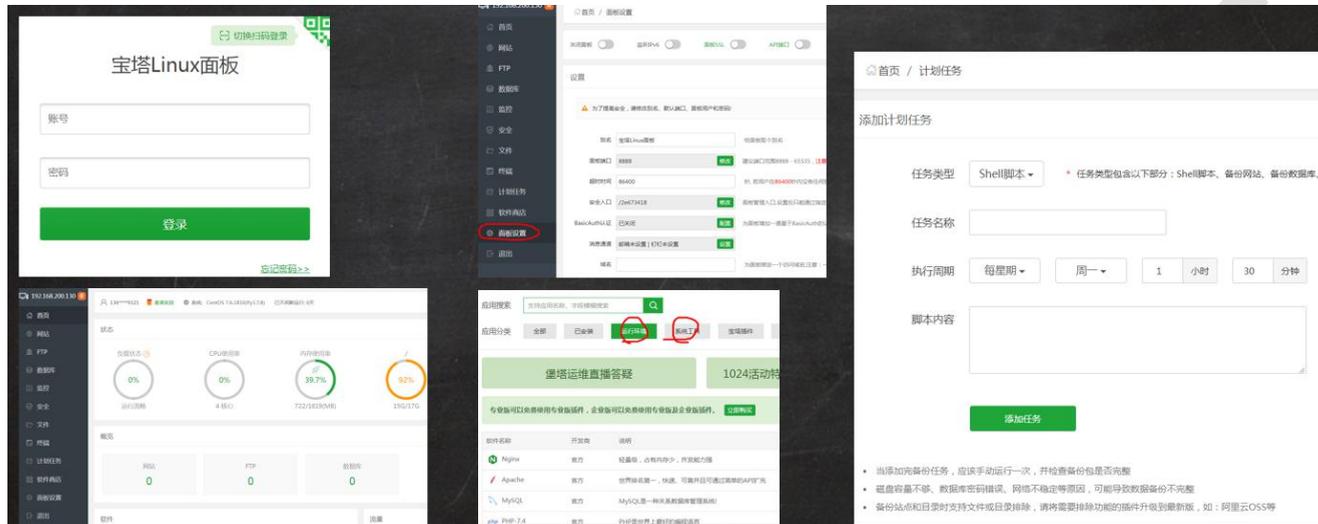
```

success
=====
Congratulations! Installed successfully!
=====
外网面板地址: http://111.197.235.27:8888/2e673418
内网面板地址: http://192.168.200.130:8888/2e673418
username: bkdf4rhh
password: 78a18d2b
If you cannot access the panel,
release the following panel port [8888] in the security group
若无法访问面板，请检查防火墙/安全组是否有放行面板[8888]端口
=====

```

25.2.3 使用介绍，比如可以登录终端，配置，快捷安装运行环境和系统工具，添加计划任务脚本

<http://192.168.200.130:8888/2e673418/>



25.2.4 如果 bt 的用户名，密码忘记了，使用 bt default 可以查看

```
[root@hspedu100 opt]# bt default
=====
BT-Panel default info!
=====
外网面板地址: http://111.197.235.27:8888/2e673418
内网面板地址: http://192.168.200.130:8888/2e673418
username: bkdf4rhh
password: 78a18d2b
If you cannot access the panel,
release the following panel port [8888] in the security
若无法访问面板, 请检查防火墙/安全组是否有放行面板 [8888]
=====
```

第 26 章 Linux 面试题-(腾讯,百度,美团,滴滴)

26.1 分析日志 t.log(访问量), 将各个 ip 地址截取, 并统计出现次数, 并按从大到小排序(腾讯)

```
http://192.168.200.10/index1.html
http://192.168.200.10/index2.html
http://192.168.200.20/index1.html
http://192.168.200.30/index1.html
http://192.168.200.40/index1.html
http://192.168.200.30/order.html
http://192.168.200.10/order.html
```

答案: `cat t.txt | cut -d '/' -f 3 | sort | uniq -c | sort -nr`

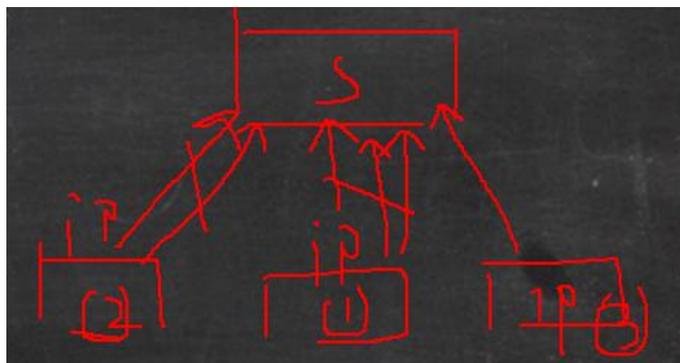
26.2 统计连接到服务器的各个 ip 情况, 并按连接数从大到小排序 (腾讯)

`netstat -an | grep ESTABLISHED | awk -F " " '{print $5}' | cut -d ":" -f 1 | sort | uniq -c | sort -nr`

```

recv_internet_connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN
tcp        0      0 192.168.122.1:53       0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:6010         0.0.0.0:*               LISTEN
tcp        0      0 192.168.200.130:22     192.168.200.1:57854    ESTABLISHED
tcp6       0      0 :::3306                 :::*                    LISTEN
tcp6       0      0 :::111                  :::*                    LISTEN

```



26.3 问题：如忘记了 mysql5.7 数据库的 ROOT 用户的密码，如何找回? (滴滴)

26.4 写出指令：统计 ip 访问情况，要求分析 nginx 访问日志(access.log)，找出访问页面数量在前 2 位的 ip(美团)

```
cat access.log | awk -F " " '{print $1}' | sort | uniq -c | sort -nr | head -2
```

26.5 使用 tcpdump 监听本机，将来自 ip 192.168.200.1，tcp 端口为 22 的数据，保存输出到 tcpdump.log，用做将来数据分析(美团) >>

```
root@hspedu100 ~]# cat /home/tcpdump.log
7:27:42.589871 IP ThinkPad-PC.6990 > 192.168.200.130.ssh: Flags [P.], seq 189889345:189889381, ack 2170440642, win
7:27:42.589979 IP 192.168.200.130.ssh > ThinkPad-PC.6990: Flags [.], ack 36, win 257, length 0
7:27:42.591818 IP 192.168.200.130.ssh > ThinkPad-PC.6990: Flags [P.], seq 1:37, ack 36, win 257, length 36
7:27:42.603496 IP 192.168.200.130.ssh > ThinkPad-PC.6990: Flags [P.], seq 37:545, ack 36, win 257, length 508
7:27:42.604624 IP ThinkPad-PC.6990 > 192.168.200.130.ssh: Flags [.], ack 545, win 256, length 0
7:27:51.708473 IP ThinkPad-PC.6990 > 192.168.200.130.ssh: Flags [P.], seq 36:72, ack 545, win 256, length 36
7:27:51.709376 IP 192.168.200.130.ssh > ThinkPad-PC.6990: Flags [P.], seq 545:581, ack 72, win 257, length 36
7:27:51.908038 IP ThinkPad-PC.6990 > 192.168.200.130.ssh: Flags [.], ack 581, win 256, length 0
7:27:52.171578 IP ThinkPad-PC.6990 > 192.168.200.130.ssh: Flags [P.], seq 72:108, ack 581, win 256, length 36
7:27:52.173472 IP 192.168.200.130.ssh > ThinkPad-PC.6990: Flags [P.], seq 581:617, ack 108, win 257, length 36
7:27:52.380071 IP ThinkPad-PC.6990 > 192.168.200.130.ssh: Flags [.], ack 617, win 256, length 0
7:27:53.609440 IP ThinkPad-PC.6990 > 192.168.200.130.ssh: Flaas [P.], seq 108:144, ack 617, win 256, lenath 36
```

```
tcpdump -i ens33 host 192.168.200.1 and port 22 >> /home/tcpdump.log
```

26.6 常用的 Nginx 模块，用来做什么(头条)

rewrite 模块，实现重写功能

access 模块：来源控制

ssl 模块：安全加密

ngx_http_gzip_module: 网络传输压缩模块

ngx_http_proxy_module 模块实现代理

ngx_http_upstream_module 模块实现定义后端服务器列表

ngx_cache_purge 实现缓存清除功能

26.7 如果你是系统管理员，在进行 Linux 系统权限划分时,应考虑哪些因素? (腾讯)

1) 首先阐述 Linux 权限的主要对象

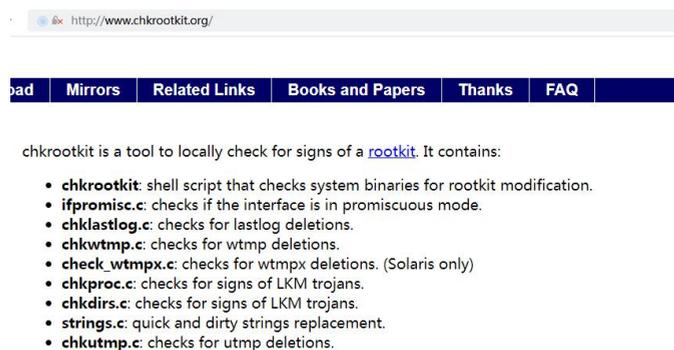
```
[root@hspedu100 etc]# ls -l
总用量 1956
drwxr-xr-x. 3 root root 4096 10月 15 19:46 abrt
-rw-r--r--. 1 root root 16 10月 15 19:53 adjtime
-rw-r--r--. 1 root root 1518 6月 7 2013 aliases
-rw-r--r--. 1 root root 12288 10月 15 19:55 aliases.db
drwxr-xr-x. 3 root root 4096 10月 15 19:48 alsa
drwxr-xr-x. 2 root root 4096 10月 19 17:40 alternatives
```



2) 根据自己实际经验谈考虑因素

- ✓ 注意权限分离，比如：工作中，Linux 系统权限和数据库权限不要在同一部门
- ✓ 权限最小原则(即:在满足使用的情况下最少优先)
- ✓ 减少使用 root 用户，尽量用普通用户+sudo 提权进行日常操作。
- ✓ 重要的系统文件，比如/etc/passwd, /etc/shadow etc/fstab, /etc/sudoers 等,日常建议使用 chattr(change attribute)锁定,需要操作时再打开。【演示 比如: 锁定 /etc/passwd 让任何用户都不能随意 useradd,除非解除锁定】

- ✓ 使用 SUID, SGID, Sticky 设置特殊权限。
- ✓ 可以利用工具，比如 `chkrootkit`/`rootkit hunter` 检测 `rootkit` 脚本（`rootkit` 是入侵者使用工具,在不察觉的建立了入侵系统途径） [演示使用 `wget ftp://ftp.pangeia.com.br/pub/seg/pac/chkrootkit.tar.gz`]
- ✓ 利用工具 `Tripwire` 检测文件系统完整性



26.8 权限操作思考题

- 1) 用户 tom 对目录 `/home/test` 有执行 `x` 和读 `r` 写 `w` 权限, `/home/test/hello.java` 是只读文件, 问 tom 对 `hello.java` 文件能读吗(ok)? 能修改吗(no)? 能删除吗(ok)
- 2) 用户 tom 对目录 `/home/test` 只有读写权限, `/home/test/hello.java` 是只读文件, 问 tom 对 `hello.java` 文件能读吗(no)? 能修改吗(no)? 能删除吗(no)?
- 3) 用户 tom 对目录 `/home/test` 只有执行权限 `x`, `/home/test/hello.java` 是只读文件, 问 tom 对 `hello.java` 文件能读吗(ok)? 能修改吗(no)? 能删除吗(no)?

4) 用户 tom 对目录 /home/test 只有执行和写权限, /home/test/hello.java 是只读文件, 问 tom 对 hello.java 文件能读吗(ok)? 能修改吗(no)? 能删除吗(ok)?

26.9 说明 Centos7 启动流程, 并说明和 CentOS6 相同和不同的地方(腾讯)

26.10问题: 列举 Linux 高级命令, 至少 6 个(百度)

```
netstat //网络状态监控 top //系统运行状态 lsblk //查看硬盘分区 find  
ps -aux //查看运行进程 chkconfig //查看服务启动状态 systemctl //管理系统服务器
```

26.11问题: Linux 查看内存、io 读写、磁盘存储、端口占用、进程查看命令是什么?(瓜子)

```
top, iotop, df -lh , netstat -tunlp , ps -aux | grep 关心的进程
```

26.12使用 Linux 命令计算 t2.txt 第二列的和并输出 (美团)

```
张三 40  
李四 50  
王五 60  
cat t2.txt | awk -F " " '{sum+=$2} END {print sum}'
```

26.13Shell 脚本里如何检查一个文件是否存在? 并给出提示(百度)

```
if [ -f 文件名 ] then echo “存在” else echo “不存在” fi
```

26.14用 shell 写一个脚本, 对文本 t3.txt 中无序的一系列数字排序, 并将总和输出(百度)

```
9  
8  
7  
6
```

5
4
3
2
10

```
sort -nr t3.txt | awk '{sum+=$0; print $0} END {print "和="sum}'
```

26.15 请用指令写出查找当前文件夹 (/home) 下所有的文本文件内容中包含有字符 “cat” 的文件名称(金山)

```
grep -r "cat" /home | cut -d ":" -f 1
```

26.16 请写出统计/home 目录下所有文件个数和所有文件总行数的指令(在金山面试题扩展)

```
find /home/test -name "*" | wc -l
```

```
find /home/test -name "*" | xargs wc -l
```

26.17 列出你了解的 web 服务器负载架构(滴滴)

Nginx

Haproxy

Keepalived

LVS

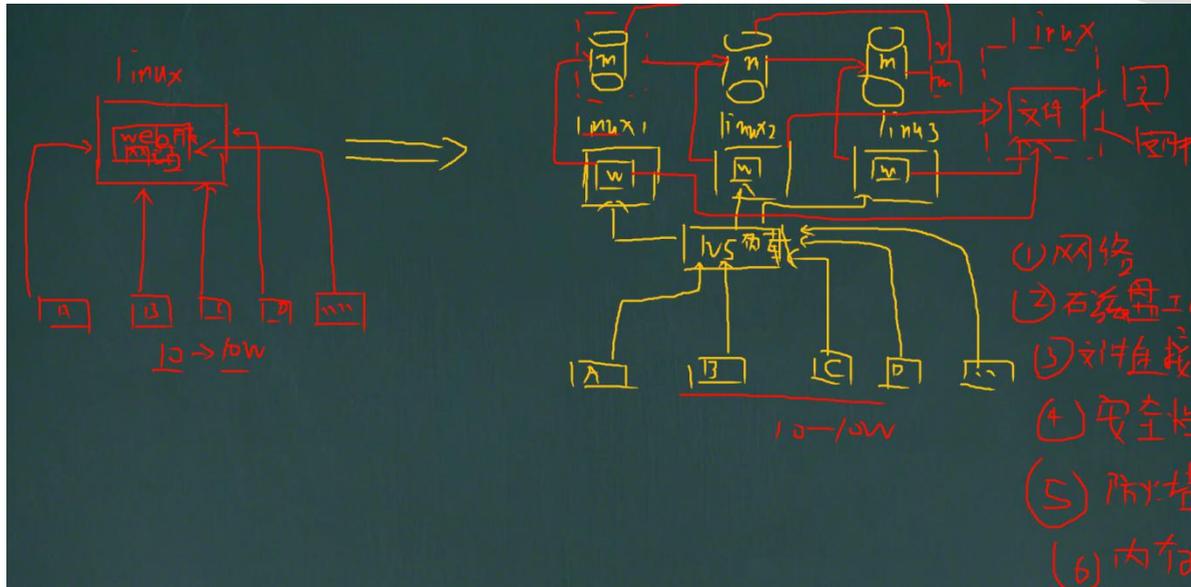
26.18 每天晚上 10 点 30 分, 打包站点目录/var/spool/mail 备份到/home 目录下(每次备份按时间生成不同的备份包 比如按照 年月日时分秒)(滴滴)

```
#!/bin/bash  
cd /var/spool/ && /bin/tar zcf /home/mail-`date +%Y-%m-%d_%H%M%S`.tar.gz mail/
```

```
[root@hspEdu100 ~]# crontab -l  
30 2 * * * /usr/sbin/mysql_db_backu  
30 20 * * * /root/mail.sh
```

26.19如何优化 Linux 系统， 说出你的方法 (瓜子)

1) 对 Linux 的架构的优化, 和原则分析(示意图)



2) 对 linux 系统本身的优化-规则

- (1) 不用 root ,使用 sudo 提示权限
- (2) 定时的自动更新服务时间,使用 `nptdate npt1.aliyun.com` , 让 croud 定时更新
- (3) 配置 yum 源, 指向国内镜像(清华, 163)
- (4) 配置合理的防火墙策略,打开必要的端口, 关闭不必要的端口
- (5) 打开最大文件数(调整文件的描述的数量) `vim /etc/profile ulimit -SHn 65535`
- (6) 配置合理的监控策略
- (7) 配置合理的系统重要文件的备份策略
- (8) 对安装的软件进行优化, 比如 nginx ,apache
- (9) 内核参数进行优化 `/etc/sysctl.conf`
- (10) 锁定一些重要的系统文件 `chattr /etc/passwd /ect/shadow /etc/inittab`
- (11) 禁用不必要的服务 `setup , ntsysv`

26.20Linux 面试 100 道题



自己先做，
答案课后领取