

linux 视频教程第 0 讲.开山篇

为什么学习 linux

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，目前很多中型，大型甚至是巨型项目都在使用 linux

linux 内核: redhat 、红旗 linux、ubuntu、suse、fedora，它们的内核都是一样的（Note: linux 其实是一个统称，就比如面条是一个统称，可以有哨子面、阳春面、打卤面等）

linux for 工作

- linux 系统管理员
 - linux 系统的维护、配置等
- linux 程序员
 - 需 c/c++、java, php、jsp...
 - linux 软件工程师（PC）
 - linux 嵌入式开发（单片机、芯片）

如何学习 linux

第一阶段: linux 平台上的开发，包括 vi, gcc, gdb, make, jdk, tomcat, mysql..和 linux 基本操作

第二阶段: 加厚 c 语言功底《c 专家编程》或是 java 语言

第三阶段: 学习 unix 环境高级编程《unix 环境高级编程》

第四阶段: linux 应用系统开发/linux 嵌入式开发

内容讲解

基础部分

- linux 基础知识
- linux 常用命令 80 个
- linux 分区/vi/权限...

实用部分

- Samba 安装与配置

- linux 网络环境配置
- crontab 使用
- jdk/apache/mysql/ssh/rpm 安装与配置
- linux 下 java 网络编程
- shell 初步介绍

推荐书籍

- 《鸟哥的 Linux 私房菜 基础学习篇》 鸟哥、许伟、林彩娥等编著
- 《Linux 编程从入门到精通》 宫虎波编著
- 《Linux 内核完全剖析》 赵炯编著

linux 视频教程第 1 讲.基础介绍

linux 的初步介绍

linux 的特点

- 免费的/开源
- 支持多线程/多用户
- 安全性好
- 对内存和文件管理优越

linux 的缺点

- 操作相对困难

linux 的历史

- 1960 时期左右，MIT，即麻省理工学院有一台电脑，使用分时操作系统，只能同时允许 30 个人通过终端登录
- 1965 年，MIT、GE、Bell 实验室，决定将 30→300 个人分时系统，multis 计划，即火星计划
- 1969 年，火星计划失败。但 Bell 的 Ken Thompson 开发了一个 file server system 文件系统，在 Bell 实验室很受欢迎
- 在 Dennis Ritchie 的加入下，1973 年，unix 诞生，开源，源码内核共享
 - IBM: AIX
 - Sun: Solaris

- HP: HP unix
- 伯克利分校: BSD
- minix 系统出现
- Linus Torvalds, 芬兰读书, 拥有 PC 386, 1991 年计划把 minix 移植到 pc 上, 1994 发布 linux 1.0 版 linux is not unix, 完全没有桌面
 - redhat 红帽子
 - s.u.s.e
 - 红旗 linux

linux 的第一次接触

关机命令

shutdown -h now 立即进行关机

shutdown -r now 现在重新启动计算机

reboot 现在重新启动计算机

进入桌面

startx

用户登录

登录时尽量少用 **root** 账户登录, 因为它是系统管理员, 最大的权限, 难免操作失误。可以利用普通用户登录, 登录后再用 “**su -**” 命令来切换成系统管理员身份

用户注销

在提示符下输入 **logout** 即可

linux 视频教程第 2 讲. vi 编辑器的使用

什么是 vi 编辑器

vi 编辑器是 linux 下最有名的编辑器, 也是我们学习 linux 必须掌握的工具, 在 linux 下也可使用 vi 进行程序的开发, 如 java 程序, c 程序

如何使用 vi 进行开发？

在 linux 下使用 vi 开发一个简单的 java 程序 Hello.java，并且在 linux 下运行成功

- 开发步骤
 - java 程序
 - **vi** Hello.java
 - 输入 **i**，进入到插入模式
 - 输入 Esc 键，进入命令模式
 - 输入冒号:[wq 表示退出保存，q!表示退出不保存]
 - 编译 **javac** Hello.java
 - 运行 **java** Hello
 - c 程序
 - **gcc** **-o** Hello Hello.cpp[参数 o 表示可自定义生成的 out 文件名，否则默认为 a.out]
 - **./**Hello

linux 视频教程第 3 讲.用户管理.目录结构

概述

简单介绍

linux 的文件系统是采用层级式的树状目录结构，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

深刻理解 linux 文件目录是非常重要的

- /
 - root，存放 root 用户的相关文件
 - home，存放普通用户的相关文件
 - bin，存放常用命令的目录，如 vi，su
 - sbin，要具有一定权限才可以使用命令
 - mnt，默认挂载光驱和软驱的目录
 - etc，存放配置的相关文件
 - var，存放经常变化的文件，如网络连接的 sock 文件
 - boot，存放引导系统启动的相关文件
 - usr，安装一个软件的默认目录，相当于 windows 下的 program files

常用命令介绍

- **pwd**, 显示当前在哪个路径下

linux 的用户管理

- **useradd** 用户名, 添加用户
【案例】**useradd xiaoming**
- **passwd** 用户名, 为新用户设密码
【案例】**passwd xiaoming**, 修改小明的密码
- **userdel** 用户名, 删除用户
【案例】**userdel xiaoming**, 删除用户但保存用户主目录
【案例】**userdel - xiaoming**, 删除用户以及用户主目录
- **logout**, 当前用户推出
- **who am i**, 当前用户是谁

linux 视频教程第 4 讲.常用命令

linux 的常用命令

- **init** [0123456], 指定系统运行级别, 类似 windows 的正常运行模式或安全模式
 - 0: 关机
 - 1: 单用户
 - 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, 要修改默认的运行级别可改文件 `/etc/inittab` 的 `id:5:initdefault:` 这一行中的数字

FAQ: 不小心设置了 6, 导致系统启动-重启-启动循环, 怎么办?

- 在进入 **grub** 引导界面时, 在数秒的时候, 请输入 **e**
- 然后选中第二行, 输入 **e**
- 在出现的界面里, 输入 **1** 【1 表示单用户级别】, 1 的前面需要加一个空格, 单用户模式既可以修改模式, 又可以修改密码, **Enter**
- 返回后, 按 **b**
- **pwd**, 显示当前工作目录
- **cd**, 改变目录

- **ls**, 列出文件和目录
 - **ls -a**, 显示目录下的所有文件, 包括隐藏文件
 - **ls -l**, 显示长列表格式
- **mkdir**, 建立目录
- **rmdir**, 删除空目录
- **touch**, 建立空文件
- **cp**, 复制命令

【案例】**cp -r dir1 dir2**, 递归复制命令 (复制子目录信息)

- **mv**, 移动文件和改文件名
- **rm**, 删除文件和目录
 - **rm -rf ***, 删除所有内容, 包含目录和文件, **r** 表示递归, **f** 表示强制
- **ln**, 建立符号连接, 类似于建立某个文件的快捷方式
 - **ln -s** 源目标

【案例】**ln -s /etc/inittab inittab**, **inittab** 指向实际文件 **/etc/inittab**

- **more**, 显示文件内容带分页, **ctrl + page up** 上翻, 长空格下翻
- **less**, 显示文件内容带分页
- **grep**, 在文本中查询内容

【案例】假设存在某个文件 **aaa.java** 中含有 **shunping** 关键字, 此时可以使用 **grep** 命令来查找, **grep n "shunping" aaa.java**, **n** 表示在第 **n** 行出现

- **|**, 管道命令, 在 **linux** 和 **unix** 系统
 - 怎么理解? 把上一个命令的结果交给 **|** 后面的命令处理

【案例】**ls -l /etc/ | more**

- **man [command]**, 帮助, 类似于 **dos** 中的 **help**
- **find**, 搜索文件及目录。在 **linux** 中, 因为文件系统是以级别式的结构来组成的, 所以要在整个系统中找到特定的文件和目录并不是件容易的事。而 “**find**” 命令可以解决上述问题
 - 在特定的目录下搜索并显示指定名称的文件和目录

【案例】**find / -name man**: 意思是说从根目录开始搜索名称为 **man** 的文件或目录

- 搜索一段时间内被存取/变更的文件或目录

【案例】**find /home -amin -10**: 十分钟内存取的文件或目录

【案例】**find /home -atime -10**: 十小时内存取的文件或目录

【案例】**find /home -cmin -10**: 十分钟内更改过的文件或目录

【案例】**find /home -ctime +10**: 十小时前更改过的文件或目录

- 搜索指定大小的文件

【案例】**find /home -size +10k**: 意思是说查找 **/home** 目录下大小为 **10k** 的文件

- 重定向命令
 - `ls -l > a.txt`, 列表的内容写入文件 a.txt 中（覆盖写）
 - `ls -al >> aa.txt`, 列表的内容追加到文件 aa.txt 的末尾
 - 从文件中输入信息: `database_program < database_data`
- **en**, 查看环境变量
- 压缩和解压
 - 以 **zip** 和 **unzip** 处理.zip 文件
 - zip 命令的基本使用方法
 - `zip file.zip *`: zip 后接压缩后的文件名, 在它的后面输入要压缩的文件即可
 - 压缩后, 自动删除原文件
 - 【案例】`zip m file.zip to.txt`: 把 to.txt 文件压缩成 file.zip 文件, to.txt 会自动删除的
 - 将子目录一起压缩
 - 【案例】`zip -r file.zip *`: 将当前目录下的子目录一起压缩
 - 忽略子目录的内容
 - 【案例】`zip -j file.zip *`
 - 将已压缩的或没有必要压缩的文件去掉
 - 【案例】`zip -n .mpg: .jpg: .gif`: 第一种文件中间要用 “:” 分开
 - 压缩某一日之后的文件
 - 【案例】`zip -t 102002 file.zip`: 将当前目录下在 2002 年 10 月 20 日之后文件压缩
 - 不压缩链接文件的原文件
 - 【案例】`zip -y file.zip *`
 - 压缩率问题, -1~-9, 其中-9 的压缩率最高
 - 【案例】`zip -9 file.zip *`
 - 将不需要压缩的文件排除在外
 - 【案例】`zip file.zip * -x file2.txt`: 在压缩时, 将当前目录内的 file2.txt 文件排除在外
 - 以 **unzip** 命令进行.zip 文件的解压缩
 - 直接解压缩文件
 - 【案例】`unzip file.zip`
 - 排除不需要解压缩的文件
 - 【案例】`unzip file.zip -x file2`: 除了 file2 文件外, 其他的文件都解压缩
 - 查看压缩包的内容
 - 【案例】`unzip -Z file.zip`: 查看 file.zip 压缩包的内容, 也可以使用 “-l” “-v”来查看压缩包的内容
 - 以 **gzip** 和 **gunzip** 处理.gz 文件

linux 视频教程第 5-6 讲.文件权限.用户组

用户组

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

- 所有者
- 所在组
- 其它组
- 改变用户所在的组

所有者

一般为文件的创建者,谁创建了该文件,就天然的成为该文件的所有者

用 `ls -ahl` 命令可以看到文件的所有者

也可以使用 `chown 用户名 文件名`来修改文件的所有者

文件所在组

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

用 `ls -ahl` 命令可以看到文件的所有组

也可以使用 `chgrp 组名 文件名`来修改文件所在的组

其它组

除开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


文件权限

`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 (-)、目录 (d), 链接 (l)
 - 其余字符每 3 个一组 (rwx), 读 (r)、写 (w)、执行 (x)
 - 第一组 rwx: 文件所有者的权限是读、写和执行
 - 第二组 rw-: 与文件所有者同一组的用户的权限是读、写但不能执行
 - 第三组 r--: 不与文件所有者同组的其他用户的权限是读不能写和执行

也可用数字表示为: `r=4, w=2, x=1` 因此 `rwX=4+2+1=7`

- 1 表示连接的文件数
- root 表示用户
- root 表示用户所在的组
- 1213 表示文件大小（字节）
- Feb 2 09:39 表示最后修改日期
- abc 表示文件名 

改变权限的命令

chmod 改变文件或目录的权限

chmod 755 abc: 赋予 abc 权限 **rwxr-xr-x**

chmod u=rwx, g=rx, o=rx abc: 同上 **u**=用户权限, **g**=组权限, **o**=不同组其他用户权限

chmod u-x, g+w abc: 给 abc 去除用户执行的权限, 增加组写的权限

chmod a+r abc: 给所有用户添加读的权限

改变所有者 (**chown**) 和用户组 (**chgrp**) 命令

chown xiaoming abc: 改变 abc 的所有者为 xiaoming

chgrp root abc: 改变 abc 所属的组为 root

chown root ./abc: 改变 abc 这个目录的所有者是 root

chown -R root ./abc: 改变 abc 这个目录及其下面所有的文件和目录的所有者是 root

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, 同样用 **root** 的管理权限可以改变某个用户所在的组

- **usermod -g 组名 用户名**

你可以用

- **usermod -d 目录名 用户名**, 改变该用户登录的初始目录

【综合案例】

【题 1.1】建立两个用户组 **group1** 和 **group2**, 以及三个用户 **dennis**、**daniel**、**abigale**, 并且将前 2 个用户分配在 **group1** 用户组下, 后一个分配在 **group2** 用户组下

```
[root@localhost root]# groupadd group1
[root@localhost root]# groupadd group2
[root@localhost root]# useradd -g group1 Dennis
useradd: invalid user name 'Dennis'
[root@localhost root]# useradd -g group1 dennis
[root@localhost root]# useradd -g group1 daniel
[root@localhost root]# useradd -g group2 abigale
[root@localhost root]# passwd dennis
Changing password for user dennis.
New password:
BAD PASSWORD: it is too simplistic/systematic
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost root]# passwd daniel
```

【题 1.2】以 dennis 用户登录，创建一个 Hello.java 文件

【题 1.3】以 daniel 用户登录，观察是否可以访问/home/dennis 目录以及读或写其创建的 Hello.java 文件

【题 1.4】以 dennis 用户登录，修改目录/home/dennis 及 Hello.java 文件的读写权限（更正：修改目录权限的时候，应该使用 770，而不是 760，否则权限不足）

```
[dennis@localhost home]$ ls -l
total 12
drwx----- 2 abigale group2 4096 Jan 11 13:15 abigale
drwx----- 2 daniel group1 4096 Jan 11 13:14 daniel
drwx----- 2 dennis group1 4096 Jan 11 13:22 dennis
[dennis@localhost home]$ chmod 760 dennis
[dennis@localhost home]$ ls -l
total 12
drwx----- 2 abigale group2 4096 Jan 11 13:15 abigale
drwx----- 2 daniel group1 4096 Jan 11 13:14 daniel
drwxrw---- 2 dennis group1 4096 Jan 11 13:22 dennis
[dennis@localhost home]$ cd /dennis
-bash: cd: /dennis: No such file or directory
[dennis@localhost home]$ cd dennis/
[dennis@localhost dennis]$ ls -l
total 4
-rw-r--r-- 1 dennis group1 51 Jan 11 13:20 Hello.java
[dennis@localhost dennis]$ chmod 770 Hello.java
[dennis@localhost dennis]$ ls -l
total 4
-rwxrwx--- 1 dennis group1 51 Jan 11 13:20 Hello.java
[dennis@localhost dennis]$ logout
```

【题 1.5】重复【题 1.3】

【题 1.6】改变 abigale 的用户组由 group2 变为 group1

```
[root@localhost home]# usermod -g group1 abigale
```

然后，可以使用 cat /etc/passwd 查看并确定

```
dennis:x:500:500::/home/dennis:/bin/bash
daniel:x:501:500::/home/daniel:/bin/bash
abigale:x:502:500::/home/abigale:/bin/bash
```

【参考】

- groupadd 组名，在 linux 中添加组

- `vi /etc/group`, 查看 linux 中所有组信息, 可以看可以编辑
- `cat /etc/group`, 查看 linux 中所有组信息, 只可以看不可以编辑
- `useradd -g 组名 用户名`, 创建用户的同时指定将该用户分配到哪个组下
- `vi /etc/passwd`, 查看 linux 中所有用户信息, 可以看可以编辑
- `cat /etc/passwd`, 查看 linux 中所有用户信息, 只可以看不可以编辑

linux 视频教程第 7 讲.J2EE 环境配置

J2EE 环境搭建

jdk 安装步骤

- 把 mypackage.iso 挂载到 linux 操作系统上
 - 在 vm 做好配置
 - `mount /mnt/cdrom`, 挂载光驱
 - `unmount /mnt/cdrom`, 卸载光驱
- 把安装文件拷贝到/home
 - `cp 文件 /home`
- 安装
 - `./j2sdk-1_4_2_19-linux-i586.bin`
- 查看一个文件 `/etc/profile` [环境配置文件]
- 配置先前安装的 jdk

```
unset i
JAVA_HOME=/home/j2sdk1.4.2_19
PATH=$PATH:/home/j2sdk1.4.2_19/bin
CLASSPATH=.:/home/j2sdk1.4.2_19/jre/lib/rt.jar
export JAVA_HOME PATH CLASSPATH
```

eclipse 安装步骤

- 挂载共享文件
- 把安装文件拷贝到/home
 - `cp 文件 /home`
- 安装
 - `tar -zxvf eclipse-SDK-3.2.1-linux-gtk.tar.gz`
- 进入图形界面, 运行 eclipse 需要桌面支持
 - `startx`
- 启动 eclipse
 - `./eclipse`

MyEclipse 安装步骤

- 挂载共享文件
- 把安装文件拷贝到/home
 - cp 文件 /home
- 安装
 - ./ MyEclipseEnterpriseWorkbenchInstaller_5_1_0GA_E3_2_1.bin
- 注意点
 - 进入图形界面安装支持，否则报错
 - 选择已安装的 eclipse 的主目录
- 重新启动 eclipse
 - ./eclipse &
 - 这时会发现，菜单栏上多了一个 MyEclipse 选项

tomcat 安装步骤

我们知道 java ee 的服务器有 tomcat、jboss、weblogic、websphere、resin...这些都可以安装到 linux 下，我们给人家安装 tomcat，安装步骤如下：

- 挂载共享文件
- 把安装文件拷贝到/home
 - cp 文件 /home
- 安装
 - tar -zxvf jakarta-tomcat-5.0.30.tar.gz
- 测试
 - 编写一个简单的 jsp 页面
 - 配置 tomcat 和 jdk

linux 视频教程第 8 讲. linux 分区详解

概述

硬盘的分区主要分为基本分区（Primary Portion）和扩展分区（Extension Portion）两种。只是针对一个硬盘来讲，基本分区和扩展分区的数目之和不能大于 4 个，且基本分区可以马上被使用但不能再分区。扩展分区必须再进行分区后才能使用，也就是说它必须还要进行二次分区。那么有扩展分区再分下去的是什么呢？它就是逻辑分区（Logical Portion），而且逻辑分区没有数量上限制

对 windows 用户来说，有几个分区就有几个驱动器，并且每个分区都会获得一个字母标识符，然后就可以选用这个字母来指定在这个分区上的文件和目录。它们的文件结构都是

独立的，非常好理解。但对这些用户初上手 Redhat Linux，可就有点恼人了。因为对 Redhat Linux 用户来说无论有几个分区，分给哪一个目录使用，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。Redhat Linux 中每个分区都是用来组成整个文件系统的一部分。因为它采用了一种叫“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，并将一个分区和一个目录联系起来。这时要载入的那个分区将使它的存储空间在这个目录下获得

硬盘

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。例如：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

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

几个重要命令

挂载命令

mount [-parameters] [设备名称] [挂载点]

特别说明：在挂载光驱时，可直接使用 `mount /mnt/cdrom`

【案例】`mount /dev/sda1 /test/`

卸载命令

umount [挂载点]

【案例】`umount /test/`

查看磁盘使用情况

df [-parameters]

- `df -h`
- `df -l`
- `df [目录全路径]`，查看某个目录是在哪个分区

查看 linux 系统分区具体情况

fdisk -l

linux 视频教程第 9 讲.linux 安装演示

linux 视频教程第 10 讲.shell 介绍

概述

每个人在成功登陆 linux 后，系统会出现不同的提示符号，例如\$、~、#等，然后你就可以开始输入需要的命令，若是命令正确，系统就会依据命令的要求来执行，直到注销系统为止；在登录到注销期间，输入的每个命令都会经过解释及执行。而这个负责的机制就是 shell

shell 编程

其实作为命令语言交互式地解释和执行用户输入的命令只是 shell 功能的一个方面。shell 还可以用来进行程序设计。它提供了定义变量和参数的手段以及丰富的程序控制结构。使用 shell 编程类似于 DOS 中批处理文件，称为 shell script，又叫 shell 程序或 shell 命令文件

shell 的分类

Shell 名称	开发者	命令名称
Bourne	S.R.Bourne	/bin/sh
C	Bill Joy	/bin/csh
Korn	David	/bin/ksh

shell 的使用

命令历史和互动：用上下箭头键可以重复以前所输入的命令

命令完成功能：用 tab 键能自动完成相关命令，再次按 tab 可得到清单

shell 脚本文件：

- 是一个文本文件
- 命令的集合
- 有执行的权限
- 执行方式（./文件名）

用户登录后自动执行的 shell 脚本文件

- `.bashrc` 位于主目录下，它之前执行系统的脚本/etc/bashrc 主要是基本配置数据
 - 配置.bashrc 文件可以指定某些程序在用户登录的时候就自动启动
- `.bash_profile` 位于主目录下，它之前执行系统的脚本/etc/profile 主要是配置环境变量

用 **export** 可以临时加入一个系统路径，如 `export PATH=$PATH:$HOME/bin:/root/test/t1`，输出环境 `PATH`，引用原来的值`$PATH`，`$HOME` 表示工作主目录，`:`是路径分隔符

- 已经定义好的环境变量
 - `SHELL`: 默认 shell
 - `PATH`: 路径
 - `USER`: 当前登录用户的用户名
- 显示变量内容
 - `echo $SHELL`
 - `echo $USER`
 - `echo $PATH`

shell 通配符

- `*`代表多个字母或数字
- `?`代表一个字母或数字

【案例】`ls a* ls a? ls f080[1-6].tif`

- 转义字符\

【案例】`ls /mnt/win1/My\Documents`

引号

【案例】`export NAME=Michael`

`echo Welcome $NAME, the date is date`

- 单引号：不处理任何变量和命令

【案例】`echo 'Welcome $NAME, the date is date '`

- 双引号：处理变量但不处理命令

【案例】`echo "Welcome $NAME, the date is date "`

- 反引号：把引号中的每个单词作为一个命令，如果是变量则先求值然后作为一个命令处理

【案例】`echo "Welcome $NAME, the date is `date` "`

别名

- 命令: **alias** 显示系统当前定义的所有 alias

【案例】alias cp='cp -i'

【案例】alias li='ls -l --color=tty'

shell 的修改

chsh -s 输入新的 shell

查阅历史记录

- **history**, 查看使用过的命令的历史记录
- **history 5**, 此项说明会显示最近使用的 5 个命令
- **!5**, 此项说明执行历史编号为 5 的命令
- **!!s**, 此项说明执行最后一次以 “ls” 开头的命令

linux 视频教程第 11 讲.tcp.ip 基础

概述

TCP/IP 是 unix/linux 世界的网络基础, 在某种意义上, unix 网络就是 TCP/IP, 而且 TCP/IP 就是网络互联的标准。它不是一个独立的协议, 而是一组协议 (TCP、IP、UDP、ARP 等协议) 每个 Internet 上的主机和路由器都有一个 IP 地址, 它包括网络号和主机号, 现在所用的 IP 地址都是 32 位的。IP 地址按照国际标准划分为 A、B、C、D、E 五种类型

linux 视频教程第 12 讲.网络环境配置

第一种方法

- 用 root 身份登录, 运行 **setup** 命令进入到 text mode setup utility 对网络进行配置, 这里可以进行 IP、子网掩码、默认网关、DNS 的配置
- 这时网卡的配置没有生效, 运行 **/etc/rc.d/init.d/network restart** 命令我们刚才做的设置才生效
- **ifconfig**

第二种方法

- **ifconfig eth0 x.x.x.x** 对网卡进行设置

- `ifconfig eth0 network x.x.x.x` 对子网掩码设置
- 对广播地址和 DNS 使用默认的

Note: 这样配置网络将会立即生效，但是是临时生效

第三种方法

- 修改`/etc/sysconfig/network-scripts/ifcfg-eth0` 这个文件里各个属性可以修改，包括 IP、子网掩码、广播地址、默认网关等
- 这时网卡的配置没有生效，运行`/etc/rc.d/init.d/network restart` 命令我们刚才做的设置才生效

Note:

- 这种方法是**最底层**的修改方法
- 在 linux 中，所有设备都是文件

linux 视频教程第 13 讲.rpm 包.samba 配置

RPM 包

概述

一种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。这一文件格式虽然打上了 Redhat 的标志，但是其原始设计理念是开放式的，现在包括 OpenLinux、S.u.S.E.以及 Turbo Linux 等 Linux 的分发版本都有采用。可以算是工人的行业标准了

RPM 包的名称格式

`apache-1.3.23-11.i386.rpm`

- “apache”: 软件名称
- “1.3.23-11”: 软件的版本号，主版本和此版本
- “i386”: 是软件所运行的硬件平台
- “rpm”: 文件扩展名，代表 RPM 包

RPM 常用命令

- `rpm -qa`: 查询所安装的所有 rpm 软件包
- `rpm -qa | more`

- rpm -qa | grep X
- rpm -q 软件包名: 查询软件包是否安装
 - rpm -q xinetd
 - rpm -q foo
- rpm -qi 软件包名: 查询软件包信息
 - rpm -qi file
- rpm -ql 软件包名: 查询软件包中的文件
 - rpm -ql file
 - rpm -ql jdk
- 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
- rpm -qp 包文件名: 查询包的信息对这个软件包的介绍
 - rpm -qp jdk-1_5_0-linux-i586.rpm
 - rpm -qpi jdk-1_5_0-linux-i586.rpm
 - rpm -qpl jdk-1_5_0-linux-i586.rpm

安装 RPM 包

rpm -ivh RPM 包全路径名称: 安装包到当前系统

- i=install, 安装
- v=verbose, 提示, 即有提示信息
- h=hash, 进度条

删除 RPM 包

rpm -e RPM 包的名称

【案例】rpm -e jdk

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:

【案例】rpm -e foo

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

若让 RPM 忽略这个错误继续卸载, 请使用--nodeps 命令行选项

【案例】rpm -e --nodeps foo

升级 RPM 包

rpm -U RPM 包全路径名

【案例】rpm -U cvs-1.11.2-10.i386.rpm

samba 配置

什么是 samba

这些年来，windows 与 linux 操作系统各自拥有自己的用户群和市场。然而在一般公司或学校里，可能同时有 windows 和 linux 主机，windows 主机彼此之间可以利用“网上邻居”来访问共享资源。NFS 也能使 linux 主机之间实现资源访问。而 samba 服务软件能够使 windows 与 linux 之间实现资源共享

SMB 通信协议采用的是 C/S 结构，所以 SAMBA 软件可分阶段客户端及服务端两部分。通过执行 samba 客户端程序，linux 主机使可使用网络上的 windows 主机所共享的资源。而在 linux 主机上安装 samba 服务器，则可以使 windows 主机访问 samba 服务器共享的资源

samba 安装

samba 的安装步骤

- 看看是否已经安装了 samba
 - rpm -q samba
- 如果有的话，就先卸载
 - rpm -e --nodeps samba
- 把安装文件挂载到 linux 下
 - samba-common-2.2.7a-7.9.0.i386.rpm
 - samba-client-2.2.7a-7.9.0.i386.rpm
 - samba-2.2.7a-7.9.0.i386.rpm
- 拷贝 samba 的 rpm 包到/home，准备安装
- 开始安装
 - rpm -ivh samba-common-2.2.7a-7.9.0.i386.rpm
- 创建一个用户 youyou
 - useradd youyou
 - passwd youyou
- 给 youyou 设置 samba 密码
 - cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd
 - smbpasswd youyou，设置密码
- 启动 samba 服务器，测试
 - service smb start，启动
 - service smb stop，停止
 - service smb restart，重启

samba 配置

共享资源的基本配置 [/etc/samba/smb.conf](#)

- **comment**: 针对共享资源所做的说明文字。默认值为空字符串
【案例】comment=dir for todayhero: 共享这个目录是为了 todayhero 这个用户
- **path**: 若共享的资源是目录，是指定该目录的位置
【案例】path=/tmp: 共享 tmp 这个目录
- **guest ok**: 是否允许用户不使用账号和密码访问此资源
【案例】guest ok=yes: 允许用户不使用账号和密码访问此资源
【案例】guest ok=no: 不允许用户不使用账号和密码访问此资源
- **hosts allow**: 设置连接主机的地址
【案例】hosts allow=192.168.2.1 server.abc.com: 允许来自 192.168.2.1 或 server.abc.com
- **hosts deny**: 设置禁止连接的主机地址
【案例】hosts deny=192.168.2.1: 不允许 192.168.2.1 的主机访问 samba 服务器的资源
- **read only**: 用于设置共享的资源是否为可读
【案例】read only=yes: 允许只读
【案例】read only=no: 不仅仅只读，也就是说可以写入

linux 视频教程第 14 讲.crontab 详解

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：

- 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

任务调度命令

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

【案例】`5 * * * * ls -l /etc/ > /tmp/to.txt`，意思说每小时的第五分钟执行 `ls` 命令

调度文件的规则

字段名称	说明	范围
分钟	每小时中的第几分钟执行	0-59

小时	每天的第几个小时执行	0-23
日期	每月的第几天执行	1-31
月历	每年的第几个月执行	1-12
星期	每周的第几天执行	0-6

使用任务调度

- 设置任务
 - `crontab -e`
- 每隔一定时间去执行 `date > /home/mydate2`
 - 希望每天凌晨 2:00 去执行 `date >> /home/mydate2`, 可以在 `crontab -e` 中加入: `0 2 * * * date >> /home/mydate2`
 - 希望每分钟去执行: 在 `crontab -e` 中加入: `* * * * * date >> /home/mydate2`
- 怎样去调度多个任务
 - 在 `crontab -e` 中直接写多个命令 (不推荐)
 - 可以把所有的任务, 写入到一个可执行文件 (shell 编程)
- 终止任务调度
 - `crontab -r`: 终止任务调度
 - `crontab -l`: 列出当前有哪些任务调度

linux 视频教程第 15 讲.进程的介绍和管理

概述

- 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
- 每一个进程, 都会对应一个父进程, 而这个父进程可以复制多个子进程, 例如 `www` 服务器
- 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
- 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, 直到关机才结束
- 进程与线程
 - 进程: 就是正在执行的程序
 - 线程

- 轻量级的进程
- 进程有独立的地址空间，线程没有
- 线程不能独立存在，它是由进程创建
- 相对讲，线程耗费的 CPU 和内存要小于进程

进程的管理

ps 命令是用来查看目前系统中，有哪些正在执行，以及它们执行的情况，可以不加任何参数，显示详细的进程信息

- 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

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	正在执行的命令或进程名

终止进程 kill/killall

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

终止某个进程: **kill 进程号**

【案例】kill 16251: 终止进程号为 16251 的进程

【案例】kill -9 16251: 因为有些进程会捕捉某些信号，如果直接不能结束进程可以用“-9”传送信息

killall: 杀死同名的所有进程

动态监控进程

top 命令与 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进程

- 监视特定用户
 - **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
- 终止指定的用户
 - **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
- 指定系统状态更新的时间
 - **top -d 10**: 指定系统更新进程的时间为 10 秒

top 显示选项解释

```
03:32:58 up 1:52, 1 user, load average: 0.00, 0.00, 0.00
38 processes: 37 sleeping, 1 running, 0 zombie, 0 stopped
CPU0 states: 0.1% user 0.1% system 0.0% nice 0.0% iowait 99.3% idle
CPU1 states: 0.0% user 0.0% system 0.0% nice 0.0% iowait 100.0% idle
Mem: 254664k av, 189856k used, 64808k free, 0k shrd, 53016k buff
Swap: 257032k av, 0k used, 257032k free 44276k cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	CPU	COMMAND
2819	root	15	0	1036	1036	856	R	0.2	0.4	0:00	0	top
1	root	15	0	476	476	424	S	0.0	0.1	0:04	1	init
2	root	RT	0	0	0	0	SW	0.0	0.0	0:00	0	migration/0

- 1:52, 表示系统启动了多久
- 1 user, 用户数
- load average: 0.00 0.00 0.00, 当前系统负载情况, 一般来说, 参数越小, 系统运行的越轻松, 当平均数>0.6 时, 系统就很紧张了
- 38 processes, 进程数
- 0 zombie, 僵尸进程数, 相当于这个进程没有用了, 还占用资源, 比如父进程来不及收回子进程
- CPU states: 99.3% idle, 闲置的 CPU
- Mem, 内存
- Swap, 类似于虚拟内存

设置系统时间

- **date 命令**: 显示系统的时间, 可以在直接输入“date”命令来查看系统的时间
- 利用 date 命令来更改系统的时间
 - **date MMDDHHMMCCYY.SS**: 月月日日时时分分年年.秒秒
- 查看月历
 - **cal 3 2002**: 查看 2002 年 3 月的月历
- 查看年历
 - **cal 2008**: 查看 2008 的年历

linux 视频教程第 16 讲.监控网络状态

几个监控命令

显示网络统计信息的命令 **netstat**

此命令用来显示整个系统目前的网络情况。例如目前的连接、数据包传递数据、或是路由表内容，此命令直接输入即可使用

- **netstat -anp**
 - **an**，按一定顺序排列输出
 - **p**，表示显示哪个进程在调用

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，或是两部主机间的介质是否为断、网线是否脱落或网卡故障

- **ping 对方 ip 地址**

显示数据包经过历程命令 **traceroute**

此命令可以直接输入使用，用来检测数据包在网络上传输的过程，从本机到远程的主机完整路径，帮助管理员解决问题

显示路由表 **route**

所谓路由是指将数据由来源网络送往目的网络的操作。在大型网络中，路由是非常复杂的，因为数据包在抵目的地时，可能经过的节点有很多，路由表是存储在路由器或一些其他链接设置上的窗体。其中记录着了到指定目的的网络路径，以及这些路径的相关数值

此命令可以直接输入使用，来查看本机路由的情况

linux 视频教程第 17 讲.mysql 安装.配置.使用

概述

mysql 数据库在 linux 下可以充分发挥威力，mysql 数据库越来越受到软件公司的青睐，为什么呢？

免费、跨平台、轻、支持多并发

在北京很多软件公司属于创业型的中、小公司，从节约成本的角度考虑，mysql 特别适合中、小项目

mysql 安装

- 创建 mysql 组
 - useradd mysql
- 创建 mysql 用户，并放入到 mysql 组中
 - useradd -g mysql mysql
- 进入到 mysql 文件夹
 - 初始化数据库
 - scripts/mysql_install_db -user=mysql
 - 修改文件的所有者
 - chown -R root .
 - 修改 data 文件夹的所有者
 - chown -R mysql data
 - 改变用户组
 - chgrp -R mysql .
- 启动 mysql
 - bin/mysqld_safe --user=mysql &
 - &表示以后台的方式启动
 - 检查一下进程，netstat -anp，查看监听端口是 3306 的是不是打开了
- 如何进入 mysql
 - cd bin
 - ./mysql -u root -p 回车

Notes: 如果希望在任何一个目录下都可以进入 mysql，则需在用户变量/root/.bash_profile 中添加路径

- 测试 mysql 数据库是否可以在 linux 下正确使用
 - 建立数据库和表
 - 加入部分数据
 - 编写一个 ShowUser.java 文件，在控制台显示用户

Note: 特别注意 mysql 的驱动要存放的位置，放在 jdk 的主目录下的/jre/lib/ext/

备份与恢复

备份: mysqldump -u root -p 密码 数据库名 > data.bak

恢复: mysql -u root -p 密码 数据库名 < data.bak

Note: -p 和密码之间没有空格

linux 视频教程第 18 讲.ssh 安装.配置.使用

概述

ssh (secure shell) 是一款集远程操作 linux 和进行文件上传和下载的软件，在软件公司几乎所有的 linux 程序员都会使用 ssh。安全、方便是它最大的特点

linux 上默认安装 ssh 服务，且默认是启动的 sshd，监听的端口是 22。在 windows 系统上安装 SSH 客户端，集成了 secureCRT 与 FTP 的作用

linux 视频教程第 19 讲.补充 linux 重要内容

linux 视频教程第 20 讲.linux 启动过程分析

runlevel 命令，可以查看当前的运行级别

linux 启动过程

- BIOS 自检
- 启动 GRUB/LILO
- 运行 linux 内核并检测硬件
- 运行系统的第一个进程 init
- init 读取系统引导配置文件/etc/inittab 中的信息进行初始化
- /etc/rc.d/rc.sysinit 系统初始化脚本
- /etc/rc.d/rcX.d/[KS] * -根据运行级别 X 配置服务
 - 终止以 “K” 开头的服务
 - 启动以 “S” 开头的服务
- /etc/rc.d/rc.local 执行本地特殊配置
- 其他特殊服务

linux 视频教程第 21 讲.java 网络编程

linux 下网络编程是 linux 最让程序员着迷的地方,我们看看如何在 linux 进行网络编程。
最终大家可以在这个基础上扩展为 my QQ 的程序,并且会使用到 mysql 数据库

你将学习到:

- 如何使用 java 进行 socket 编程
- 如何在 java 中对 mysql 数据库操作
- windows 和 linux 网络通讯
- 了解什么是网络服务这个晦涩的概念

linux 系统作为服务端,代码如下:

```
import java.io.*;
import java.net.*;

public class MyServerTest {

    public static void main(String[] args) {

        try {
            // 创建ServerSocket
            ServerSocket ss = new ServerSocket(8888);
            System.out.println("I am Listening");
            // 监听,直到某个客户端来连接自己,一旦连接成功,就会得到Socket
            // 就好像是在客户端和服务端之间建立了一个数据管道,可以进行数据传输
            Socket s = ss.accept();
            // 可以读取从客户端发送的信息
            InputStreamReader isr = new InputStreamReader(s.getInputStream());
            BufferedReader br = new BufferedReader(isr);
            String message = br.readLine();
            System.out.println("client sent::" + message);
            // 从服务端,回一个消息
            OutputStreamWriter osw = new OutputStreamWriter(s.getOutputStream());
            PrintWriter pw = new PrintWriter(osw, true);
            pw.println("I get your message :" + message);
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

windows 作为客户端,代码如下:

```

import java.io.*;
import java.net.*;

public class MyClient {

    public static void main(String[] args) {
        try {
            // 连接ip和port
            Socket s = new Socket("192.168.220.254", 8888);
            // 通过文件流写入发送消息
            OutputStreamWriter osw = new OutputStreamWriter(s.getOutputStream());
            BufferedWriter bw = new BufferedWriter(osw);
            PrintWriter pw = new PrintWriter(bw, true);
            // 发送
            pw.println("hello ,world");
            // 读取从服务器回送的信息
            InputStreamReader isr = new InputStreamReader(s.getInputStream());
            BufferedReader br = new BufferedReader(isr);
            String response = br.readLine();
            System.out.println("Server response is :::" + response);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

后记

VM 上的 Redhat Linux 9.0 共享文件夹

步骤如下：

- 启动虚拟机 LINUX 操作系统，打开 VMware→工具栏→VM→Install VMware Tools，出现对话框，选择 Install，这时在 mnt 目录下的 cdrom 目录就可以看到我们要装的软件：
 - VMwareTools-5.5.0-13124.i386.rpm
 - VMwareTools-5.5.0-13124.tar.gz
- 打开超级终端，输入命令：cd /mnt/cdrom 进入到光驱的目录下
- cp VMwareTools-5.5.0-13124.tar.gz /tmp 把这个文件拷贝到 tmp 下
- cd /tmp 进入 tmp 目录，输入 ls 查看刚才的文件是否在这个目录下
- tar zxvf VMwareTools-5.5.0-13124.tar.gz 解压这个文件
- cd vmware-tools-distrib 进入 tmp 目录下的 vmware-tools-distrib 目录
- ./vmware-install.pl 执行这个文件，所有提示都按“Enter”键。安装结束后重启
- 打开 VMware→工具栏→VM→Settings→出现对话框，选择工具栏 Options→选左边 Shared Folders→选右边的 Add→下一步→在出现的对话框的里点 Browse(Host folder)，来选择所要共享的目录→下一步→选择 Enable this share，单击“完成”。在/mnt/hgfs 下就有你共享的文件夹