

Linux系统基础培训

计算 决定未来

目录

第一章：Linux操作系统简介

第二章：Linux操作系统安装和基本配置

第三章：Linux操作系统的文件系统结构

第四章：Linux操作系统常用命令详解

第一章：Linux操作系统简介

- 1.1 Linux操作系统起源
- 1.2 Linux操作系统特性
- 1.3 Linux操作系统结构
- 1.4 Linux操作系统内核版本
- 1.5 Linux操作系统发行版本
- 1.6 Linux操作系统特点

第二章：Linux操作系统安装和基本配置

第三章：Linux操作系统的文件系统结构

第四章：Linux操作系统常用命令详解

第一章：Linux操作系统简介

- Linux 以它的高效性和灵活性著称。具有多任务、多用户的能力。
- Linux 之所以受到广大计算机爱好者的喜爱，其主要原因有两个：
 - 一是它属于自由软件，用户不用支付任何费用就可以获得它和它的源代码，并且可以根据自己的需要对它进行必要的修改。
 - 另一个原因是，它具有 Unix 的全部功能，任何使用 Unix 操作系统或想要学习 Unix 操作系统的人都可以从 Linux 中获益。

1.1 Linux操作系统起源



Linux是一个诞生于网络、成长于网络且成熟于网络的操作系统。

1991年，芬兰大学生Linus Torvalds萌发了开发一个自由的UNIX操作系统的想法，当年Linux诞生，为了不让这个羽毛未丰的操作系统夭折，Linus将自己的作品Linux通过Internet发布。从此一大批知名的、不知名的电脑黑客、编程人员加入到开发过程中来，Linux逐渐成长起来。

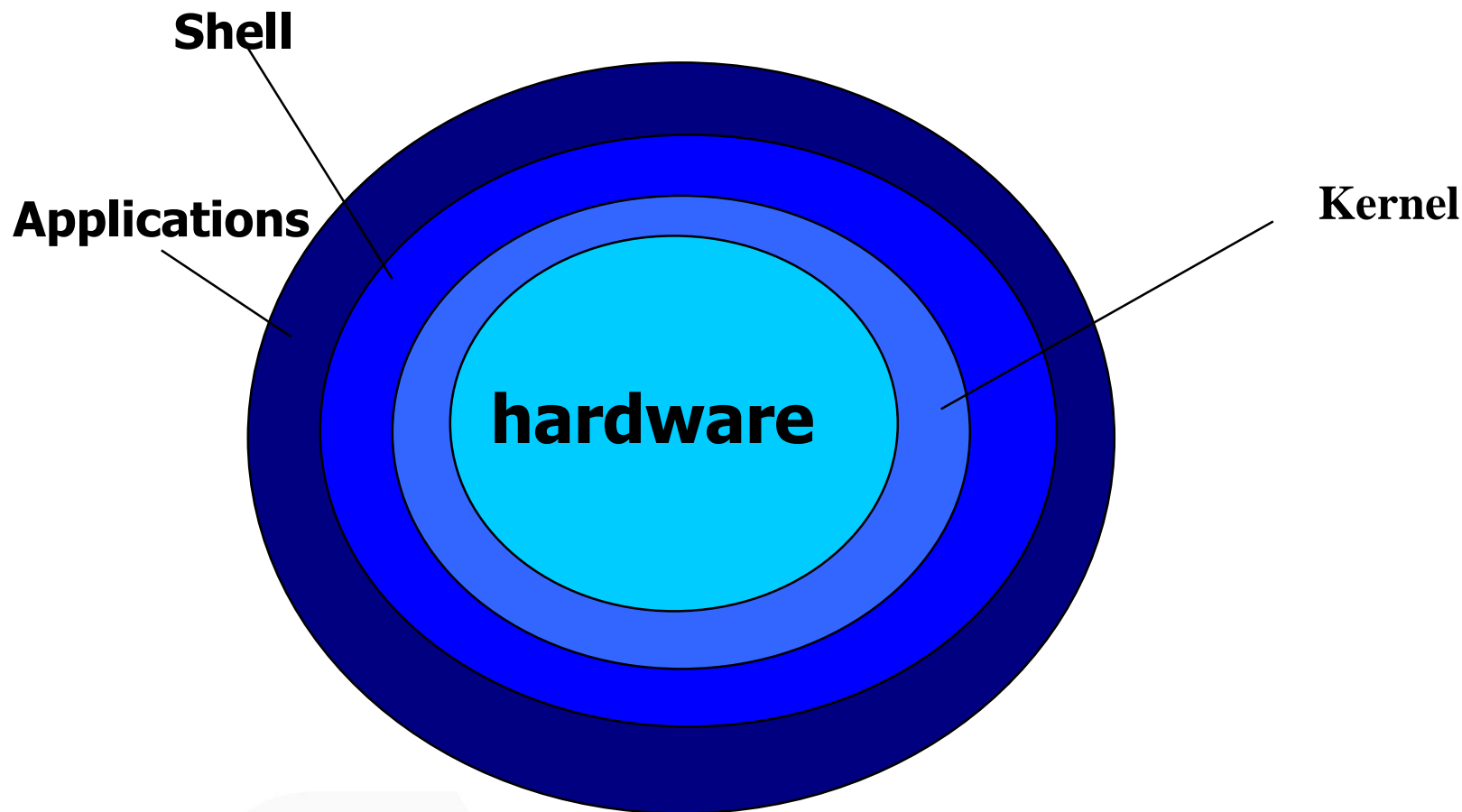
现在，Linux凭借优秀的设计，不凡的性能，加上IBM、INTEL、CA、CORE、ORACLE等国际知名企业的大力支持，市场份额逐步扩大，逐渐成为主流操作系统之一。

Linux只是内核，即操作系统中允许用户的软件与硬件通信的那部分

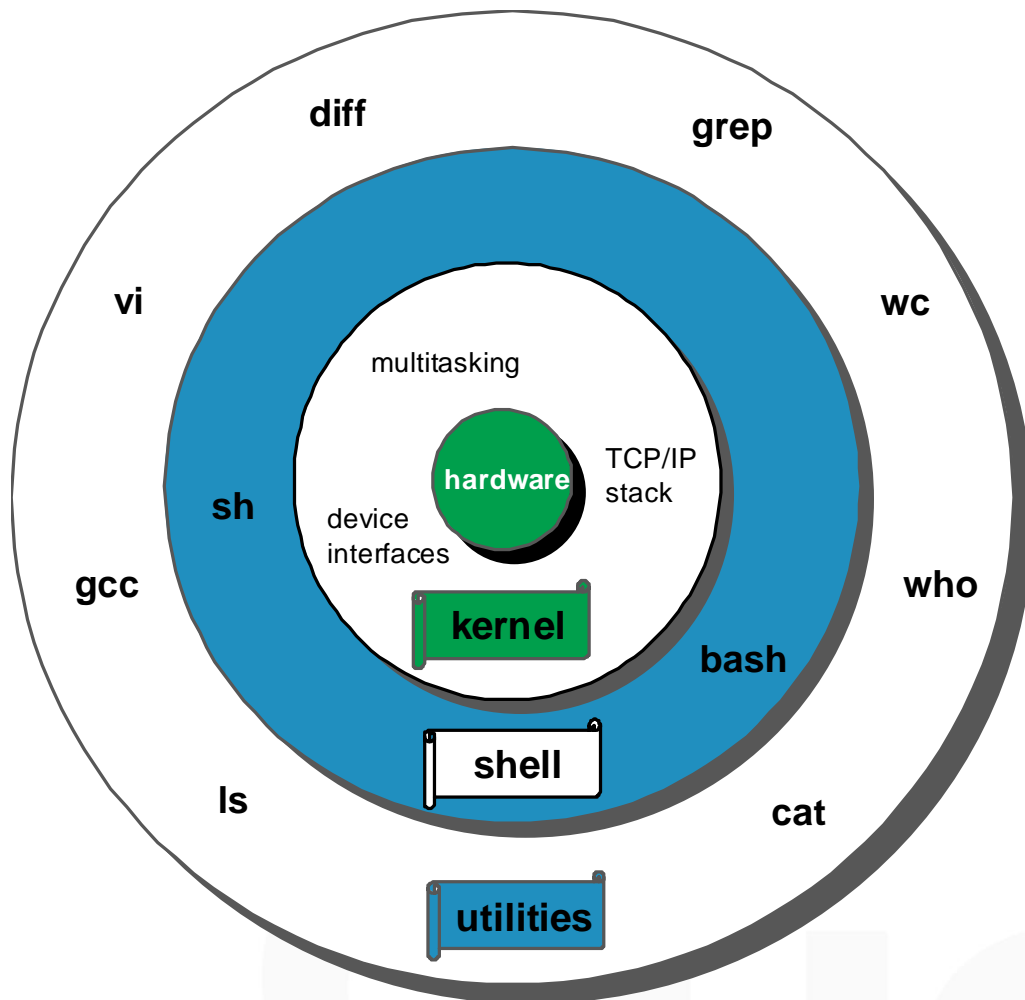
1.2 Linux操作系统特点

- **开放性**：指系统遵循世界标准规范，特别是遵循开放系统互连（OSI）国际标准。
- **多用户**：是指系统资源可以被不同用户使用，每个用户对自己的资源（例如：文件、设备）有特定的权限，互不影响。
- **多任务**：它是指计算机同时执行多个程序，而且各个程序的运行互相独立。
- **良好的用户界面**：Linux向用户提供了两种界面：用户界面和系统调用。Linux还为用户提供了图形用户界面。它利用鼠标、菜单、窗口、滚动条等设施，给用户呈现一个直观、易操作、交互性强的友好的图形化界面
- **设备独立性**：是指操作系统把所有外部设备统一当作成文件来看待，只要安装它们的驱动程序，任何用户都可以象使用文件一样，操纵、使用这些设备，而不必知道它们的具体存在形式。Linux是具有设备独立性的操作系统，它的内核具有高度适应能力
- **提供了丰富的网络功能**：完善的内置网络是Linux一大特点。
- **可靠的安全系统**：Linux采取了许多安全技术措施，包括对读、写控制、带保护的子系统、审计跟踪、核心授权等，这为网络多用户环境中的用户提供了必要的安全保障。
- **良好的可移植性**：是指将操作系统从一个平台转移到另一个平台使它仍然能按其自身的方式运行的能力。Linux是一种可移植的操作系统，能够在从微型计算机到大型计算机的任何环境中和任何平台上运行。

1.3 Linux操作系统结构



1.3 Linux操作系统结构



- Kernel
 - 系统启动时将内核装入内存
 - 管理系统各种资源
- Shell
 - 用户界面，提供用户与内核交互处理接口
 - 是命令解释器，提供强大的编程环境
 - bash,ash,pdksh,tcs h,ksh,sh,csh,zsh....
- Utility
 - 提供各种管理工具，应用程序

1.4 Linux操作系统内核版本

版本号码

稳定版本 2.4.32

开发版本 2.5.47-6

主版本号.次版本号（偶：稳定版/奇：开发版）.发布号-patch号

<http://www.kernel.org/pub/linux/kernel/>

1.5 Linux操作系统发行版本



第一章：Linux操作系统简介

第二章：Linux操作系统安装和基本配置

- 2.1 Linux系统安装前的准备
- 2.2 Linux系统安装方式的选择
- 2.3 Linux系统安装内容的选择
- 2.4 服务器磁盘控制器类型
- 2.5 Linux系统磁盘分区
- 2.6 Linux系统分区优点
- 2.7 Linux系统分区识别
- 2.8 Linux系统主要分区介绍
- 2.9 Linux 操作系统用户登录
- 2.10 Linux系统基本配置

第三章：Linux操作系统的文件系统结构

第四章：Linux操作系统常用命令详解

2.1 Linux系统安装前的准备

■ Linux安装前须知

安装前要了解所安装服务器的硬件配置方面的信息，主要包括磁盘控制器型号、网卡型号、内存大小及CPU类型等，根据相关信息和用户使用情况安装合适的Linux操作系统。在安装Linux前要了解Linux的一些基础知识。

■ 服务器硬盘分区要求

根据需要，硬盘的分区可以自己定义，建议最少安装三个分区：/根分区、/swap 交换分区、/boot引导分区。

2.2 Linux系统安装方式的选择

- 通过光盘介质引导安装

CD、DVD

- 通过网络引导安装

NFS, HTTP, FTP

- 通过软盘 (DOS) 引导安装

需将操作系统拷贝到硬盘某个分区

- 通过网络传输镜像文件安装

2.3 Linux系统安装内容的选择

- 根据实际需要来选择将要安装的软件包
 - 作为网络服务器专用，提供用户服务
 - 作为办公专用，处理日常事务
 - 作为企业工作站专用，提供内部资源管理
 - 如果将来有别的需要，可以很方便的安装相应软件，实现各种所需功能

2.4 服务器磁盘控制器类型

■ 磁盘类型: SATA, SAS, SSD

■ RAID卡型号:

Adaptec: www.adaptec.com

LSI : www.lsilogic.com

2.5 Linux系统磁盘分区

■ 磁盘分区：

- 主分区：最多4个主分区，其中一个主分区必须被标志为是“活动的”，而且包含一个引导装入程序
- 扩展分区：将主分区之一转换成一个扩展分区，然后将扩展分区再细分成任意个逻辑分区，不可将目录安装在扩展分区上
- 逻辑分区：每个物理驱动器上最多只能有11个逻辑分区

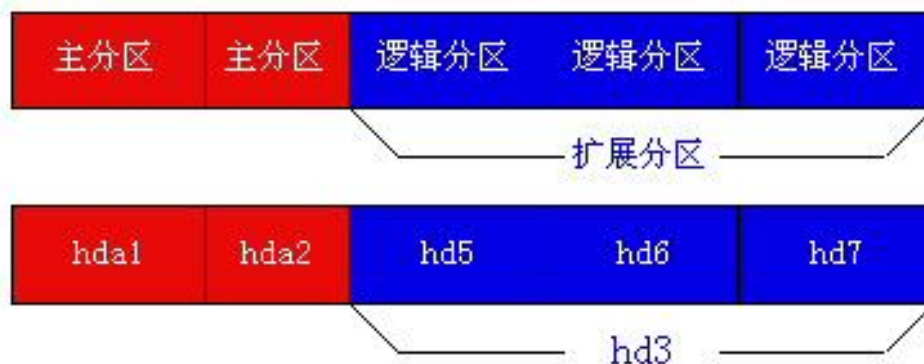
2.6 Linux系统分区优点

- 增强可用性：
如果系统一个分区由于故障而不能使用，其余好的分区仍然可以使用；
- 减少停机维护时间：
如果系统故障只影响一部分分区，那么只有这部分分区需要修复，故能比整个分区修复花的时间更少；
- 维护轻松：
如果需要修复分区，单独修复每个分区比修复整个大分区要轻松得多；
- 区对用户透明：
最终用户感觉不到分区的存在。

2.7 Linux系统分区识别

第一IDE的主盘: /dev/hda 第一SCSI/SAS/SATA的主盘: /dev/sda
第二IDE的从盘: /dev/hdb 第二SCSI/SAS/SATA的从盘: /dev/sdb
第三IDE的主盘: /dev/hdc 第三SCSI/SAS/SATA的主盘: /dev/sdc

图例:



主分区（或扩展分区）被命名为hda1-hda4，如果没有，就跳过。而扩展分区中的逻辑分区则从hda5开始编号，以此类推。

2.8 Linux系统主要分区介绍—/boot

■ /boot引导分区：

该目录放置系统内核及驱动模块引导程序，需要独立分区：

1. 独立有利于避免文件系统损坏造成的无法启动问题，如果独立出来，而/根文件系统因为别的原因损坏了，可以正常引导进入linux内核，并有很大几率进入一个可用的shell来修复系统，如果不独立，则很可能/文件系统损坏导致找不到/boot下的内核镜像，从而无法引导内核。
2. 文件系统支持问题。若安装多操作系统，/boot分区独立可以很方便的管理多系统引导

■ /boot修复

若由于操作或别的原因导致系统引导损坏，可以通过以下方法修复：

进入grub模式

grub> find /boot/grub/stage1 注：找到/boot分区位置

返回 (hd0,0) 注：说明/boot分区在第一块硬盘的第一个分区

grub>root (hd0,0) 注：这是/boot所在的分区；

grub>setup (hd0) 注：把GRUB写到MBR上；

2.8 Linux系统主要分区介绍—/swap

■ Swap空间的作用

- 当系统的物理内存不够用的时候，就需要将物理内存中的一部分空间释放出来，以供当前运行的程序使用。那些被释放的空间可能来自一些很长时间没有什么操作的程序，这些被释放的空间被临时保存到Swap空间中，等到那些程序要运行时，再从Swap中恢复保存的数据到内存中。这样，系统总是在物理内存不够时，才进行Swap交换。
- 有一点要声明的是，并不是所有从物理内存中交换出来的数据都会被放到Swap中（如果这样的话，Swap会不堪重负），有相当一部分的数据直接交换到文件系统

■ 分区规则：

- Swap space 比内存要慢5-6个数量级
- 在安装时，系统会尝试将交换分区安装到磁盘外端
- 当有多个磁盘控制器时，在每个磁盘上都建立交换分区
- 尽量将交换分区安装在访问最频繁的数据区附近

2.8 Linux系统主要分区介绍—/swap

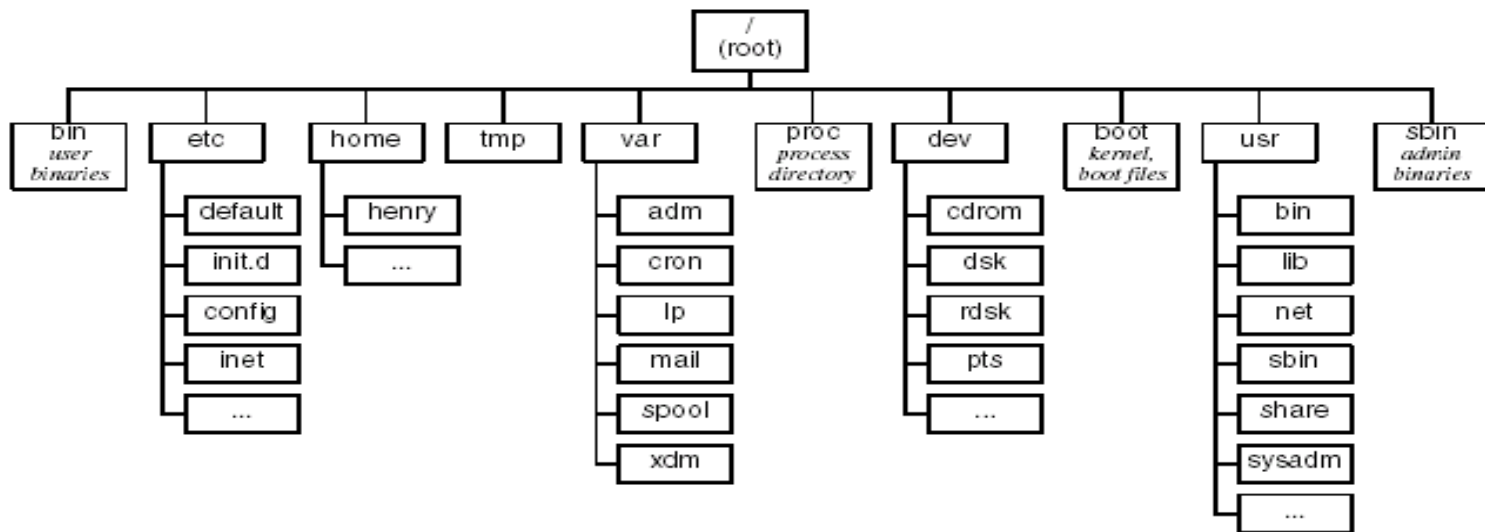
■ 创建Swap文件

1. 以超级用户 (root) 身份登陆
dd if=/dev/zero of=swapfile bs=1024 count=32000
创建一个有连续空间的交换文件。大小为1024*32000=32G
2. 激活Swap文件
#/usr/sbin/swapon swapfile
swapfile指的是上一步创建的交换文件。
3. 在/etc/fstab
/path/swapfile none Swap default 0 0
4. 检验Swap文件是否加上
/usr/sbin/swapon -s

■ 删除多余的Swap空间。

1. 成为超级用户
2. 使用Swapoff命令收回Swap空间。
#/usr/sbin/swapoff swapfile
3. 编辑/etc/fstab文件, 去掉此Swap文件的实体。
4. 从文件系统中回收此文件。
#rm swapfile

2.8 Linux系统主要分区介绍一 /



/—根目录

/root —超级用户主目录

/boot - kernel 和boot配置文件

/usr - 用户程序

/home - 用户目录

/tmp - 临时文件

/var - 可变信息区 (file spool,logs,requests,mail,etc.)

/proc - 进程信息

/sbin - 系统管理员执行程序

/bin - 基本命令

/etc - 各种配置文件

/opt--- 附加的应用软件包

/mnt--- 设备/文件系统挂载点

/dev - 设备

/lib--- 基本的共享库和核心模块

2.9 用户登录

- 超级用户root (password 在安装时已设定)
 - root为系统管理员，具有完全的系统控制权限；
建议除了必要的系统维护之外，平常尽量避免用root登录，以免误操作；
 - login: root
password: *****
[root@localhost root]#
- 普通用户同样需要输入name & password
[xyz@localhost xyz]\$

2.9 用户登录（远程）

■ Putty

- 一款免费的Telnet和SSH客户端实现,可使用在Win32平台下。

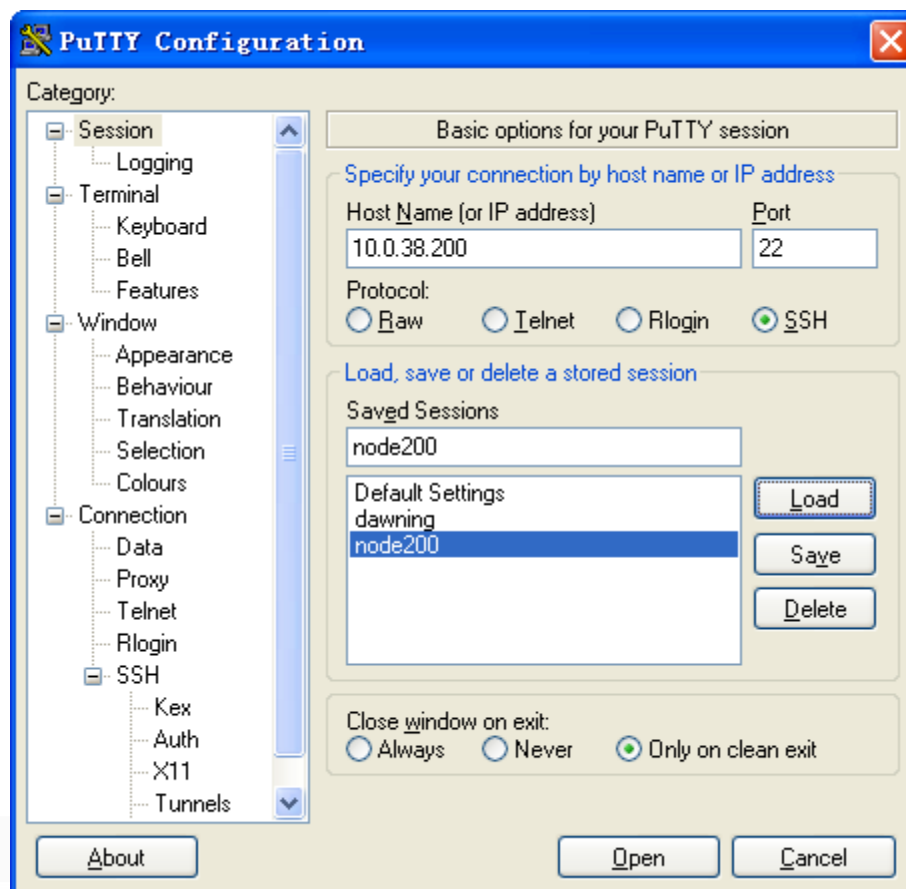
■ Xmanager

- 一个简单易用的运行在Windows平台上的X Server软件。它能把远端Unix/Linux的桌面带到你的Windows上。

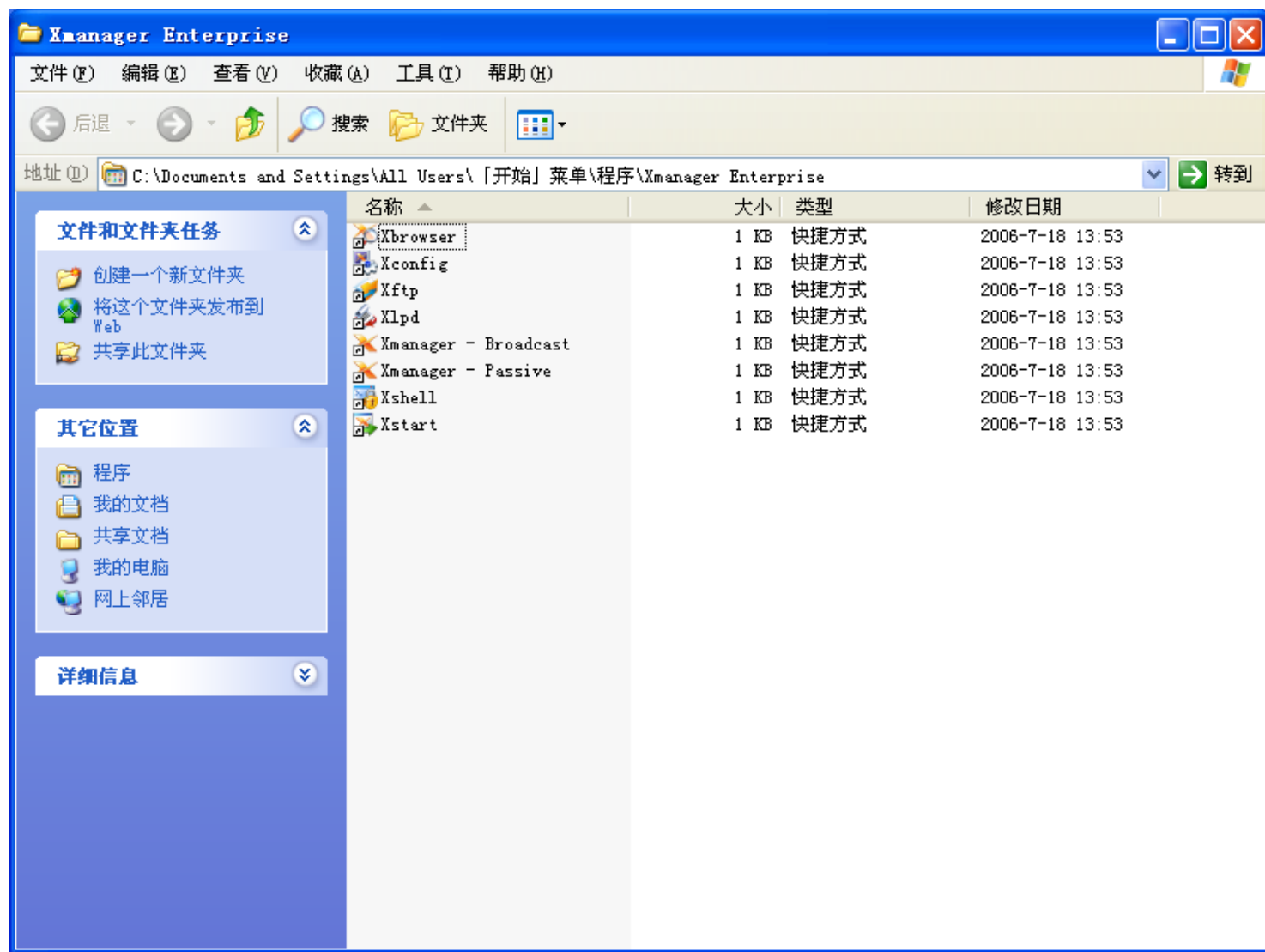
Putty(1)

- ❑ 完全免费;
- ❑ 在Windows 9x/NT/2000下运行的都非常好;
- ❑ 全面支持ssh1和ssh2;
- ❑ 绿色软件, 无需安装, 下载后在桌面建个快捷方式即可使用;
- ❑ 体积很小, 仅364KB (0.54 beta版本);
- ❑ 操作简单, 所有的操作都在一个控制面板中实现。

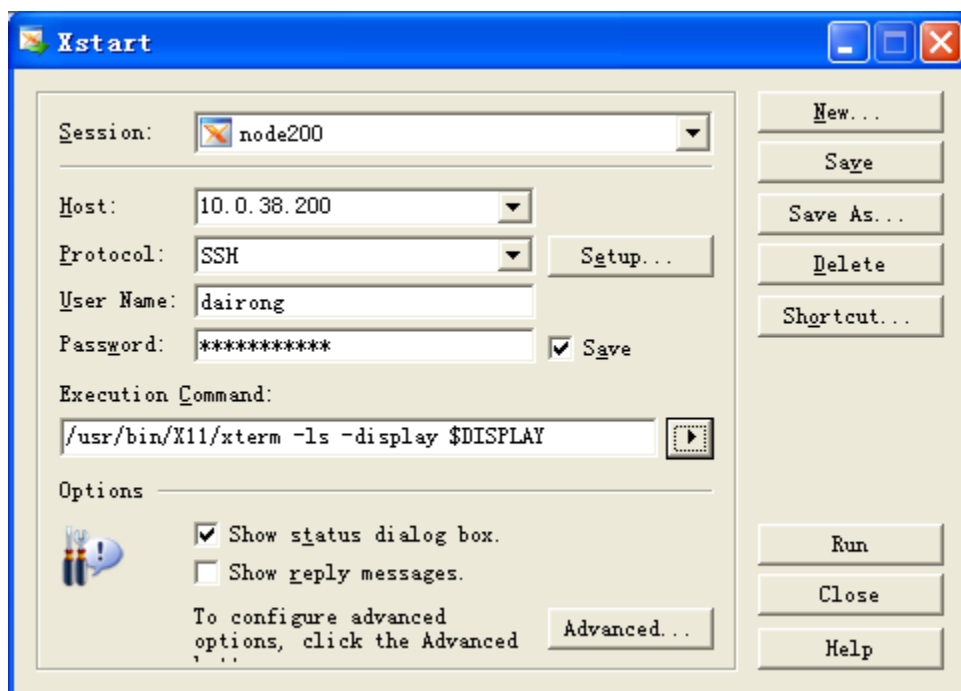
Putty(2)



Xmanager(1)



Xmanager(2)



2.10 Linux系统基本配置-网络配置

■ 网络的配置

#cd /etc/sysconfig/network-scripts

#vi ifcfg-eth0 写入网络IP等

#vi ifcfg-eth1 写入网络IP等

■ 设置完IP后，重启网络系统

/etc/init.d/network restart

■ 配置网关

在ifcfg-eth0、 ifcfg-eth1 添加GATEWAY=网关IP

■ 配置DNS

在ifcfg-eth0、 ifcfg-eth1 添加NAMESERVER=DNSIP

2.10 Linux系统基本配置-网络文件

■ /etc/hosts

- ❑ 如果系统的 IP 不是动态获取，简单的主机名解析
- ❑ 文件格式: ip地址 主机名
- ❑ cat /etc/hosts : 127.0.0.1 localhost.localdomain localhost

■ /etc/services

- ❑ Internet网络服务文件，将网络服务名转换为端口号 / 协议。由 inetd、telnet、tcpdump 和一些其它程序读取。文件中的每一行对应一种服务(它由4个字段组成，中间用TAB或空格分隔)，分别表示“服务名称”、“使用端口”、“协议名称”以及“别名”。
- ❑ 文件格式: 服务 端口/端口类型 别名

■ /etc/sysconfig/network(redhat linux)、/etc/hostname(suse)

- ❑ 主机名配置文件,记录着本机的主机名

2.10 Linux系统基本配置-服务的管理

- 服务开机启动

`chkconfig 服务名称 on`

- 服务开机不启动

`chkconfig 服务名称 off`

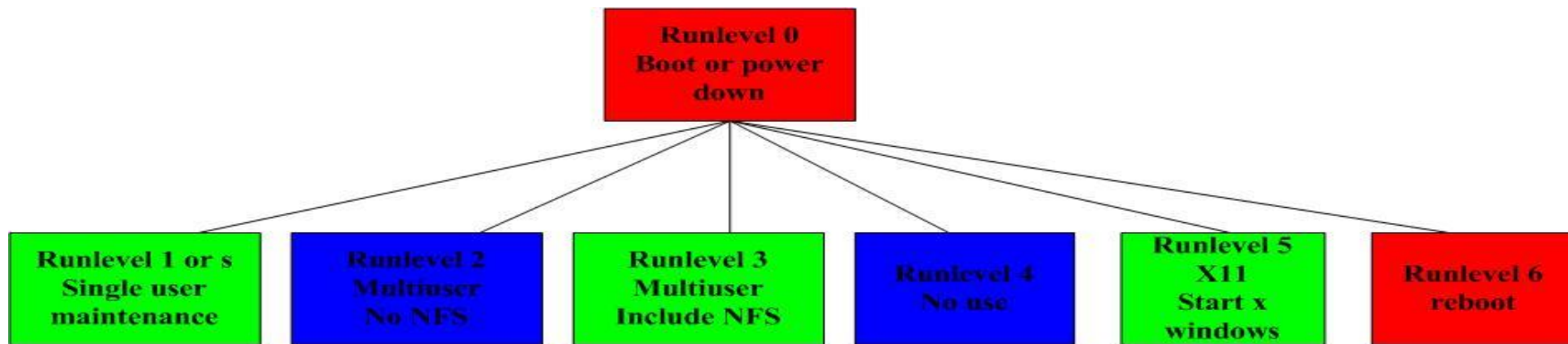
- 服务关于运行级别的定义

`chkconfig --level 35 服务名称 on/off`

- 服务的察看

`chkconfig --list |grep 服务名称`

2.10 Linux系统基本配置-运行级别



/etc/inittab文件:

id:3:initdefault # 缺省的运行级:

0 - 停机 (千万不要把initdefault设置为0)

1 - 单用户模式

2 - 多用户, 但是没有NFS

3 - 完全多用户模式

4 - 没有用到

5 - X11

6 - 重新启动 (千万不要把initdefault设置为6)

第一章：Linux操作系统简介

第二章：Linux操作系统安装和基本配置

第三章：Linux操作系统的文件系统结构

- 3.1 文件与目录的基本概念
- 3.2 文件系统基本概念
- 3.3 Linux文件系统类型
- 3.4 常用文件系统比较
- 3.5 文件系统的创建
- 3.6 Linux系统的文件结构

第四章：Linux操作系统常用命令详解

3.1 文件与目录的基本概念

- 文件：是用来存储信息的基本结构，它是被命名（文件名）的存储在某种介质（磁盘，光盘，磁带等）上的一组信息的集合。
- 从技术上讲，文件不能存贮任何数据，它只是一个用来指向它们相应的索引节点(inode)的名字，索引节点包含了文件的真正信息。
- 文件名：是文件的标识，由字母，数字，下划线，圆点组成。
- 扩展名：为了便于识别和管理，用扩展名作为文件名的一部分，中间用. 隔开
- 目录文件：用来管理和组织大量的文件，常称为目录

3.2 文件系统基本概念

- 文件系统是包括在一个磁盘（包括光盘、软盘、闪盘及其它存储设备）或分区的目录结构；一个可应用的磁盘设备可以包含一个或多个文件系统；如果您想进入一个文件系统，首先您要做的是挂载（mount）文件系统；为了挂载（mount）文件系统，您必须指定一个挂载点；
- 文件系统是在一个磁盘（包括光盘、软盘、闪盘及其它存储设备）或分区组织文件的方法，如NTFS或FAT；
- 文件系统是文件的数据结构或组织方法。在Linux中，文件系统涉及两个非常独特的事情，目录树或在磁盘或分区上文件的排列；文件系统是基于操作系统的，建立在磁盘媒质上的可见体系结构，例如这种结构对于一个Linux用户来说可以用ls 或其它工具可以看到；
- 文件系统是基于被划分的存储设备上的逻辑上单位上的一种定义文件的命名、存储、组织及取出的方法；
- 在计算机业，一个文件系统是有组织存储文件或数据的方法，目的是易于查询和存取。文件系统是基于一个存储设备，比如硬盘或光盘，并且包含文件文件物理位置的维护；也可以说文件系统也是虚拟数据或网络数据存储的方法，比如NFS。

3.3 Linux文件系统类型

■ ext2 文件系统

- ext2文件系统应该说是Linux正宗的文件系统，早期的Linux都是用ext2，但随着技术的发展，大多Linux的发行版本目前并不用这个文件系统了；比如Redhat和Fedora 大多都建议用ext3，ext3文件系统是由ext2发展而来的。ext2支持undelete（反删除），如果您误删除文件，有时是可以恢复的，但操作上比较麻烦；

■ ext3 文件系统：是由ext2文件系统发展而来

- ext3 is a Journalizing file system for Linux（ext3是一个用于Linux的日志文件系统），ext3支持大文件；但不支持反删除（undelete）操作；Redhat和Fedora都力挺ext3；

■ reiserfs 文件系统

- reiserfs 文件系统是一款优秀的文件系统，支持大文件，支持反删除（undelete）；操作反删除比较容易；reiserfs 支持大文件；

3.4 常用文件系统比较

Filesystem	File Size Limit	Filesystem Size Limit
ext2/ext3 with 1 KiB blocksize	16448 MiB (~ 16 GiB)	2048 GiB (= 2 TiB)
ext2/3 with 2 KiB blocksize	256 GiB	8192 GiB (= 8 TiB)
ext2/3 with 4 KiB blocksize	2048 GiB (= 2 TiB)	8192 GiB (= 8 TiB)
ext2/3 with 8 KiB blocksize (Systems with 8 KiB pages like Alpha only)	65568 GiB (~ 64 TiB)	32768 GiB (= 32 TiB)
ReiserFS 3.5	2 GiB	16384 GiB (= 16 TiB)
ReiserFS 3.6 (as in Linux 2.4)	1 EiB	16384 GiB (= 16 TiB)
XFS	8 EiB	8 EiB
JFS with 512 Bytes blocksize	8 EiB	512 TiB
JFS with 4KiB blocksize	8 EiB	4 PiB
NFSv2 (client side)	2 GiB	8 EiB
NFSv3 (client side)	8 EiB	8 EiB

1024 Bytes = 1 KB; 1024 KB = 1 MB; 1024 MB = 1 GB; 1024 GB = 1 TB; 1024 TB = 1 PB; 1024 PB = 1 EiB

3.5 文件系统的创建

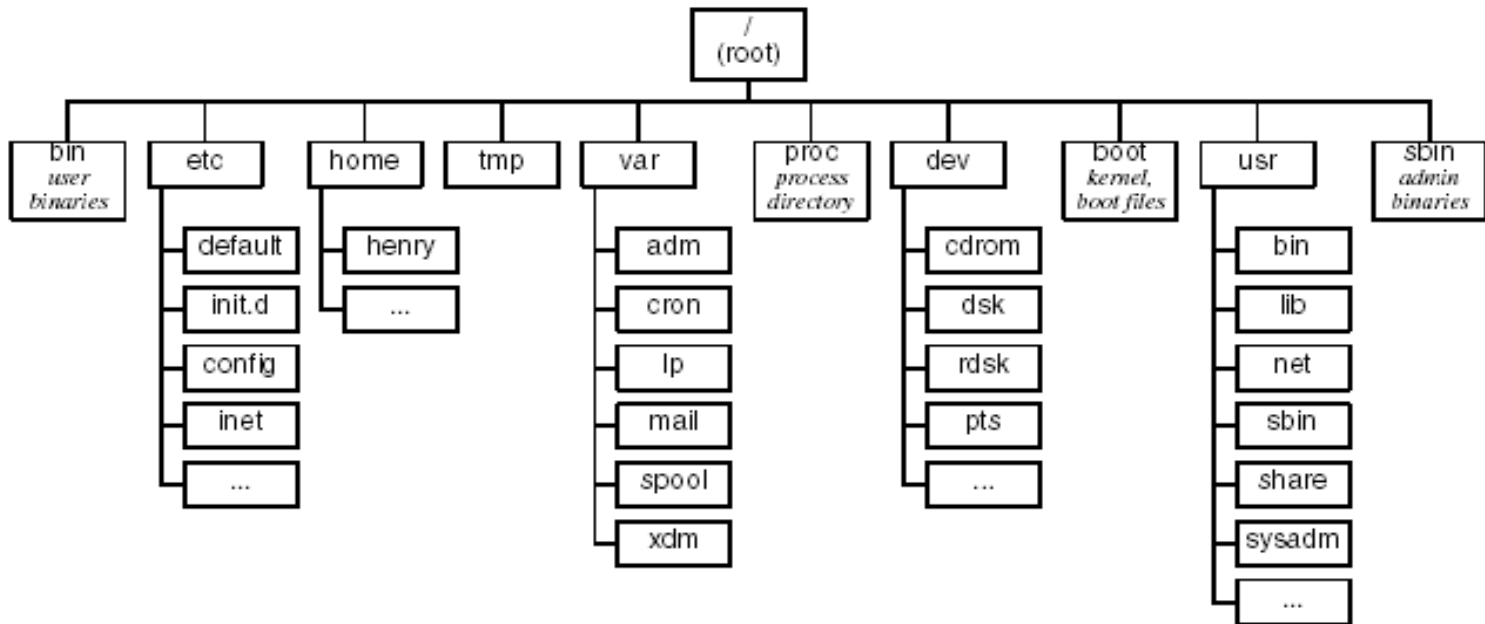
■ 文件系统的创建

- 这个过程是存储设备建立文件系统的过程，一般也被称为格式化或初始化，通过一些初始化工具来进行。一般的情况下每个类型的操作系统都有这方面的工具。在Linux中有mkfs系列工具

■ 创建方法

- 以系统第二块硬盘为例：/dev/sdb
- 分区：#fdisk /dev/sdb
- 分区完毕后，硬盘识别为/dev/sdb1
- 文件系统创建：#mkfs -t 文件系统类型 /dev/sdb1

3.6 Linux系统的文件结构



3.6.1 Linux文件类型的定义

■ 普通文件

- 文本文件：ASCII码形式存储

-开头，如：-rw-r--r-- 1 root root 39599 Mar 8 12:15 x

- 二进制文件：以二进制形式存储在计算机中，不可直接读，要通过相应的软件读取

-开头，如：-rwxrwxrwx 1 root root 46888960 Dec 9 2005 x.sh

■ 目录文件：d字母开头

如：drwxr-xr-x 2 root root 4096 Aug 2 2006 bin

■ 设备文件

- 块设备文件：b字母开头

如：brw-rw---- 1 root disk 3, 1 Jan 30 2003 hda1

- 字符设备文件：c字母开头

如：crw----- 1 root root 4, 1 Jul 31 13:49 tty1

3.6.2 Linux文件属性的定义

```
[root@localhost ~]# ls -lih
```

总计 104K

```
2408830 drwxr-xr-x 2 root root 4.0K 04-21 12:46 mkuml-2004.07.17
2408260 drwxr-xr-x 2 root root 4.0K 04-21 22:15 mydir
2408258 lrwxrwxrwx 1 root root 7 04-21 22:16 sun001.txt -> sun.txt
2408263 -rw-r--r-- 2 root root 39K 04-20 14:17 sun.txt
```

第一字段：inode；

inode：索引节点。每个存储设备或存储设备的分区（存储设备是硬盘、软盘、U盘）被格式化为文件系统后，应该有两部份，一部份是inode，另一部份是Block，Block是用来存储数据用的。而inode是用来存储这些数据的信息，这些信息包括文件大小、属主、归属的用户组、读写权限等。inode为每个文件进行信息索引，所以就有了inode的数值。操作系统根据指令，能通过inode值最快的找到相对应的文件。

第二字段：文件种类和权限；

第三字段：硬链接个数；

第四字段：属主；

第五字段：所归属的组；

第六字段：文件或目录的大小；

第七字段和第八字段：最后访问或修改时间；

第九字段：文件名或目录名

3.6.3 Linux文件权限的定义

```
[root@zhz home]# ls -l
```

```
total 32
```

drwxrwxrwx	38	down	root	4096	Jul	5	19:09	down
drwx-----	9	glh	glh	4096	Mar	26	19:08	glh
drwx-----	4	lei	lei	4096	Mar	21	08:40	lei
drwxr-xr-x	5	root	root	4096	Apr	12	2006	software

- Linux文件或目录的权限位由 9 个权限位来控制，每三位为一组，它们分别是：
 - 文件属主(Ower)的读r、写w、执行x
 - 用户组(Group)的读r、写w、执行x
 - 其它用户(Other) 的读r、写w、执行x;
 - 如果权限位不可读、不可写、不可执行，是用-来表示。

属组权限位

rwXr-xr-x

属主权限位 其它用户权限位

3.6.3 系统目录内容介绍

/	Linux系统根目录，包含所有目录
/bin	Binary的缩写，存放用户的可执行程序，例如ls,cp,也包含其它的SHELL如：bash等
/boot	包含vmlinuz,initrd.img等启动文件，随便改动可能无法正常开机
/dev	接口设备文件目录，如你的硬盘：sda
/etc	Passwd等系统设置与管理的文件
/etc/x11	X Windows System的设置目录
/home	一般用户的主目录
/lib (/lib64)	包含执行/bin和/sbin目录的二进制文件时所需的共享函数库library
/mnt	各项装置的文件系统加载点，例如：/mnt/cdrom是光驱的加载点
/opt	提供空间，较大的且固定的应用程序存储文件之用
/proc	PS命令查询的信息与这里的相同，都是系统内核与程序执行的信息
/root	管理员的主目录

3.6.3 系统目录内容介绍

/sbin	系统启动时所需的二进制程序
/tmp	Temporary,存放暂存盘的目录
/usr	存放用户使用系统命令和应用程序等信息
/usr/bin	存放用户可执行程序, 如grep,mdir等
/usr/doc	存放各式程序文件的目录
/usr/include	保存提供C语言加载的header文件
/usr/include/X11	保存提供X Windows程序加载的header文件
/usr/info	GNU程序文件目录
/usr/lib (/lib64)	函数库
/usr/lib(/lib64)/X11	函数库
/usr/local	提供自行安装的应用程序位置
/usr/man	存放在线说明文件目录
/usr/sbin	存放经常使用的程序, 如showmount
/usr/src	保存系统的源码文件
/usr/X11R6/bin	存放X Windows System的执行程序
/var	Variable,具有变动性质的相关程序目录, 如log

第一章：Linux操作系统简介

第二章：Linux操作系统安装和基本配置

第三章：Linux操作系统的文件系统结构

第四章：Linux操作系统常用命令详解

- 4.1 文件目录类命令
- 4.2 系统信息类命令
- 4.3 网络通讯类命令
- 4.4 磁盘管理类命令

4.1 文件目录类命令

- 浏览目录命令: `cd` `dir` `ls`
- 浏览文件命令: `cat` `more` `less`
- 目录操作命令: `mkdir` `rmdir`
- 文件操作命令: `touch` `vi` `rm` `cp` `mv` `ln` `tar` `gzip`
`gunzip` `whereis` `whatismfile`
- 查找用法帮助: `command -h/--help`
`man command`

4.1.1 列出文件列表的ls命令

- ls(list)是一个非常有用的命令，用来显示当前目录下的内容。配合参数的使用，能以不同的方式显示目录内容。下面是一些常用的范例。
- 格式：ls[选项] [路径或文件名]

```
[root@linux root]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@linux root]# ls -a
.                .bash_logout .cshrc          install.log.syslog .Xresources
..               .bash_profile .gtkrc         .kde
anaconda-ks.cfg  .bashrc     install.log     .tcshrc
[root@linux root]# ls -l
total 32
-rw-r--r--    1 root    root        3811 Mar 10  2005 anaconda-ks.cfg
-rw-r--r--    1 root    root       24636 Mar 10  2005 install.log
-rw-r--r--    1 root    root         0 Mar  9  2005 install.log.syslog
[root@linux root]# _
```

4.1.2 切换目录的cd命令

- cd (change directory)命令让用户切换当前目录。例如：

[test@linux home]\$ cd test 切换到当前目录下的test子目录

[test@linux test]\$ cd .. 切换到上一层目录

[test@linux home]\$ cd / 切换到系统根目录

[test@linux /]\$ cd 切换到用户自家目录（或执行cd~）

[test@linux test]\$ cd /usr/bin 切换到/usr/bin目录

4.1.3 mkdir、rmdir命令

- mkdir(make directory)命令可用来创建子目录。

- 格式：mkdir [参数] <目录名>

- [test@linux test]\$ mkdir dir
在当前目录下建立dir目录

- [test@linux test]\$ mkdir -p dir1/dir2
在当前目录下创建dir1目录，并在dir1目录下创建dir2目录，也就是连续创建两个目录（dir1/和dir1/dir2）

- 格式：rmdir [参数]<目录名>

- rmdir(remove directory)命令可用来删除“空”的子目录：

- [test@linux test]\$ rmdir dir
删除“空”的子目录dir

- [test@linux test]\$ rmdir -p dir1/dir2
删除dir1下的dir2目录，若dir1目录为空也删除它

4.1.4 复制文件的cp命令

- cp(copy)命令可以将文件从一处复制到另一处。一般在使用cp命令时将一个文件复制成另一个文件或复制到某目录时，需要指定源文件名与目标文件名或目录。

- 格式：cp[参数]<源文件路径> <目标文件路径>

- [test@linux test]\$ cp test1.txt test2.txt

将test1.txt复制成test2.txt

- [test@linux test]\$ cp test3.txt /tmp

将test3.txt复制到/tmp目录中

- [test@linux test]\$ cp -r test1(目录) test2(目录)

加-r参数，拷贝目录

4.1.5 删除文件或目录rm命令

■ 功能：删除文件或目录

- 格式：rm[参数] <目标文件路径>

- [test@linux test]\$ rm myfiles
删除一个文件

- [test@linux test]\$ rm *
删除当前目录下的所有文件

- -f参数：强迫删除文件
[test@linux test]\$ rm -f *.txt
强迫删除所有以后缀名为txt文件

- -r参数：删除目录，等同于rmdir命令

4.1.5 rm命令参数-i使用

■ -i参数：删除文件时询问

□ [test@linux test]\$ rm -i *

□ 删除当前目录下的所有文件

□ rm:backup: is a directory 遇到目录会略过

□ rm : remove 'myfiles.txt' ? Y

□ 删除文件时会询问,可按Y或N键表示允许或拒绝删除文件

□ 如果用户确定要删除这些文件，则可以使用参数-f来避免询问。

4.1.6 cat命令

- 用于显示文件的内容,也可以将数个文件合并成一个文件。
 - 格式: `cat[参数]<文件名>`
 - `[test@linux test]$cat test.txt`
显示test.txt文件内容
 - `[test@linux test]$cat test.txt | more`
逐页显示test.txt文件中的内容
 - `[test@linux test]$cat test.txt >>test1.txt`
将test.txt的内容附加到test1.txt文件之后
 - `[test@linux test]$cat test.txt test2.txt >readme.txt`
将test.txt和test2.txt文件合并成readme.txt文件

4.1.7 more命令

- more命令是一般用于要显示的内容会超过一个画面长度的情况。为了避免画面显示时瞬间就闪过去，用户可以使用more命令，让画面在显示满一页时暂停，此时可按空格键继续显示下一个画面，或按Q键停止显示。

- [test@linux test]\$ more /etc/named.conf

显示 etc/named.conf文本文件的内容

- 当用ls命令查看文件列表时，如果文件太多，则可配合more命令使用：

[test@linux etc]\$ ls -al | more

以长格形式显示etc目录下的文件列表，显示满一个画面便暂停，可按空格键继续显示下一画面，或按Q键跳离

4.1.8 less命令

- less命令的用法与more命令类似，也可以用来浏览超过一页的文件。所不同的是less命令除了可以按空格键向下显示文件外，还可以利用上下键来卷动文件。当要结束浏览时，只要在less命令的提示符 “:” 下按Q键即可。

- [test@linux etc]\$less named.conf
显示/etc/named.conf的文本文件内容

- [test@linux etc]\$ls -al | less
以长格形式列出/etc目录中所有的内容。用户可按上下键浏览或按Q键跳离

4.1.9 显示命令head、tail

■ head功能：用于显示文件前几行的内容

□ 格式：head[参数]<文件名>

□ [root@linux root]# head -10 /etc/passwd

显示/etc/passwd/文件的前10行内容

■ tail功能：用于显示文件后几行的内容

□ 格式：tail[参数]<文件名>

□ [root@linux root]# tail -10 /etc/passwd

显示/etc/passwd/文件的倒数10行内容

□ [root@linux root]# tail +10 /etc/passwd

显示/etc/passwd/文件的从第10行开始到末尾的内容

4.1.10 移动或更改文件名称的mv

- 功能：可以将文件及目录移到另一目录下，或更改文件及目录的名称
 - 格式：[参数] <源文件或目录> <目标文件或目录>
 - [test@linux dir1]\$ mv a.txt ../
将a.txt文件移动上层目录
 - [test@linux dir1]\$ mv a.txt b.txt
将a.txt改名为b.txt
 - [test@linux dir1]\$ mv dir2 ../
将dir2目录上移一层

4.1.11 显示当前目录的pwd命令

- 功能：显示用户正在工作或当前所在的目录

- 格式：pwd

- [test@linux test]\$ pwd

- /home/test

- 显示用户test所在的当前目录是/home/test

4.1.12 find命令

■ Find功能：用来寻找文件或目录

□ 格式：find [<路径>] [匹配条件]

□ [root@linux root]# find / -name httpd.conf

搜索系统根目录下名为httpd.conf的文件

/etc/httpd/httpd.conf 显示搜索结果

□ [root@linux root]# find /etc -name httpd.conf

搜索/etc目录下名为httpd.conf的文件，并显示结果

/etc/httpd/httpd.conf 显示搜索结果

4.1.13 grep命令

- 功能：在文件中搜索匹配的字符并进行输出
 - 格式：grep[参数] <要找的字符串> <要寻找字符串的源文件>
 - [root@linux root]# grep linux test.txt
搜索test.txt文件中字符串linux并输出
 - [root@linux root]# rpm -qa | grep httpd
搜索rpm包中含有httpd包的文件名
 - 例：who | grep tty1

4.1.14 vi命令详解

■ 进入vi的命令

- `vi filename` : 打开或新建文件，并将光标置于第一行首
- `vi +n filename` : 打开文件，并将光标置于第n行首
- `vi + filename` : 打开文件，并将光标置于最后一行首
- `vi +/pattern filename` : 打开文件，并将光标置于第一个与pattern匹配的串处
- `vi -r filename` : 在上次正用vi编辑时发生系统崩溃，恢复filename
- `vi filename...filename` : 打开多个文件，依次进行编辑

■ 移动光标类命令

- `w`或`W` : 光标右移一个字至字首
- `b`或`B` : 光标左移一个字至字首
- `e`或`E` : 光标右移一个字至字尾
- `)` : 光标移至句尾

4.1.14 vi命令详解

■ 屏幕翻滚类命令

Ctrl+u: 向文件首翻半屏

Ctrl+d: 向文件尾翻半屏

Ctrl+f: 向文件尾翻一屏

Ctrl + b; 向文件首翻一屏

nz: 将第n行滚至屏幕顶部, 不指定n时将当前行滚至屏幕顶部。

■ 插入文本类命令

i: 在光标前

I: 在当前行首

a: 光标后

A: 在当前行尾

o: 在当前行之下新开一行

O: 在当前行之上新开一行

r: 替换当前字符

R: 替换当前字符及其后的字符, 直至按ESC键

s: 从当前光标位置处开始, 以输入的文本替代指定数目的字符

S: 删除指定数目的行, 并以所输入文本代替之

ncw或nCw: 修改指定数目的字

nCC: 修改指定数目的行

4.1.14 vi命令详解

■ vi常用操作命令

: n1,n2 co n3: 将n1行到n2行之间的内容拷贝到第n3行下

: n1,n2 m n3: 将n1行到n2行之间的内容移至到第n3行下

: n1,n2 d : 将n1行到n2行之间的内容删除

: w : 保存当前文件

: e filename: 打开文件filename进行编辑

: x: 保存当前文件并退出

: q: 退出vi

: q!: 不保存文件并退出vi

: !command: 执行shell命令command

: r!command: 将命令command的输出结果放到当前行

: n1,n2 w!command: 将文件中n1行至n2行的内容作为command的输入并执行; 若不指定n1, n2, 则表示将整个文件内容作为command的输入

4.1.14 vi命令详解

■ 删除命令

ndw或ndW: 删除光标处开始及其后的n-1个字

do: 删至行首

d\$: 删至行尾

ndd: 删除当前行及其后n-1行

x或X: 删除一个字符, x删除光标后的, 而X删除光标前的

Ctrl+u: 删除输入方式下所输入的文本

■ 搜索及替换命令

/pattern: 从光标开始处向文件尾搜索pattern

?pattern: 从光标开始处向文件首搜索pattern

n: 在同一方向重复上一次搜索命令

N: 在反方向上重复上一次搜索命令

: s/p1/p2/g: 将当前行中所有p1均用p2替代

: n1,n2s/p1/p2/g: 将第n1至n2行中所有p1均用p2替代

: g/p1/s//p2/g: 将文件中所有p1均用p2替换

4.1.14 vi命令详解

■ 寄存器操作

?nyy: 将当前行及其下n行的内容保存到寄存器? 中, 其中?为一个字母, n为一个数字

?nyw: 将当前行及其下n个字保存到寄存器? 中, 其中?为一个字母, n为一个数字

?nyl: 将当前行及其下n个字符保存到寄存器? 中, 其中?为一个字母, n为一个数字

?p: 取出寄存器? 中的内容并将其放到光标位置处。这里? 可以是一个字母, 也可以是一个数字

ndd: 将当前行及其下共n行文本删除, 并将所删内容放到1号删除寄存器中。

4.1.15 touch命令

- 功能：生成一个空文件，或修改文件的存取/修改的时间记录值。

- 格式：touch [参数] <文件名>

- [test@linux test]\$ touch *
将当前下的文件时间修改为系统的当前时间

- [test@linux test]\$ touch -d 20070806 test
将test文件的日期改为20070806

- [test@linux test]\$ touch abc
若abc文件存在，则修改为系统的当前时间；若不存在，则生成一个为当前时间的空文件

4.1.16 who或w命令

■ 功能：查看当前系统中有哪些用户登录

□ 格式：who/w[参数]

□ [root@linux root]# who

root tty1 1个本地用户登录

test pts/0 1个远程登录用户

```
[root@linux root]# w
 5:42pm up 36 min, 2 users, load average: 0.08, 0.02, 0.06
USER      TTY      FROM              LOGIN@   IDLE   JCPU   PCPU   WHAT
root      tty1     -                 5:16pm  0.00s  0.17s  0.01s  w
pp        tty2     -                 5:42pm  3.00s  0.04s  0.04s  -bash
[root@linux root]# _
```

4.1.17 ln命令

■ 功能：在文件和目录之间建立链接

□ 格式：ln [参数] <源文件或目录> <目标文件或目录>

□ 链接分“软链接”和“硬链接”

□ 软链接:

```
[root@linux test]# ln -s /usr/share/doc doc
```

创建一个链接文件doc，并指向目录/usr/share/doc

□ 硬链接:

```
[root@linux test]# ln /usr/share/test hard
```

创建一个硬链接文件hard，这时对于test文件对应的存储区域来说，又多了一个文件指向它。

4.1.18 硬链接与软链接

- 文件名称用箭头指到另一个文件此为文件链接数
- 简单地说，软链接就是为同一个文件或目录创建2、3个名字，如同在Windows下创建的快捷方式

```
[root@linux root]# ll
total 64
-rw-r--r--    1 root    root      3811 Mar 10  2005 anaconda-ks.cfg
lrwxrwxrwx    1 root    root         4 Mar  9 17:17 file -> test
-rw-r--r--    2 root    root     24636 Mar 10  2005 hard
-rw-r--r--    2 root    root     24636 Mar 10  2005 install.log
-rw-r--r--    1 root    root         0 Mar  9  2005 install.log.syslog
-rw-r--r--    1 root    root         46 Mar  9 17:16 test
[root@linux root]# _
```

4. 1. 19 软件包管理命令 (RPM)

■ RPM的全名是Red Hat Package Manager,它是由Red Hat 公司提供的的一个开放性软件包管理系统。

- 易于安装、升级与删除软件包
- 有强大的查询功能
- 能够进行软件包验证
- 支持代码形式的软件包

4. 1. 19 软件包的安装

- 使用RPM命令的安装模式，可以将软件包内所有的组件安装到系统中的正确路径；
- 安装软件包的命令是：
 - `[root@linux root]#rpm -ivh vsftpd-2.6.2-x86-64.rpm`
 - i: rpm的安装模式
 - v: 校验文件信息
 - h: 以 # 号显示安装进度

4. 1. 19 软件包的删除

- 删除模式会将指定软件包的内容全部删除，但并不包括已更改过的配置文件；
- 删除RPM软件包的命令如下：
 - `[root@linux /]# rpm -e vsftpd`
注意：这里必须使用软件名“vsftpd”或“vsftpd-2.6.2-x86-64”，而不是使用当初安装时的软件包名vsftpd-2.6.2-x86-64.rpm

4. 1. 19 软件包升级

- 升级模式会安装用户所指定的更新版本，并删除已安装在系统中的相同软件包，升级软件包命令如下：

- `[root@linux /]# rpm -Uvh vsftpd-2.6.2-x86-64.rpm`

4. 1. 19 软件包更新

- 更新模式下，rpm命令会检查在命令行中所指定的软件包是否比系统中原有的软件包更新。
- 如果情况属实，rpm命令会自动更新指定的软件包。
- 反之，若系统中并没有指定软件包的较旧版本，rpm命令并不会安装此软件包。
- 在升级模式下，不管系统中是否有较旧的版本，rpm命令都会安装指定的软件包。
- `[root@linux /]# rpm -Fvh vsftpd-2.6.2-x86-64.rpm`

4. 1. 19 软件包查询

- 若要获取RPM软件包的相关信息，可以使用查询模式。
- 使用-q参数可查询一个已安装的软件包的内容。
 - [root@linux /]# rpm -q vsftpd
vsftpd-2.6.2-x86-64 显示软件包的名称、版本及次版本
- 查询软件包所安装的文件：安装某个软件包之后，常常困扰用户是，不知道该软件包究竟安装到哪里；
- 执行rpm -ql package-name 命令可列出安装的文件。
 - [root@linux /]# rpm -ql xv (l参数：显示文件列表)
/etc/x11/wmconfig/xv
/usr/x11R6/bin/bggen
/usr/X11R6/bin/vdcomp

4. 1. 19 常用的rpm查询命令

- `rpm -qa |grep vsftpd`

查询某个程序是否安装，及其具体版本；

- `rpm -ql vsftpd-3.4.1`

列出已安装程序的所有相关文件；

- `rpm -qpl vsftpd-3.4.1.rpm`

列出RPM软件包中的所有文件；

- `rpm -qf /etc/vsftpd.conf`

检查某个文件属于哪一个软件包；

4. 1. 20 打包命令tar

- tar命令位于/bin目录下，它能够将用户所指定的文件或目录打包成一个文件，但不做压缩。
- 一般Unix上常用的压缩方式是选用tar将许多文件打包成一个文件，再以gzip压缩命令压缩成xxx.tar.gz (或称为xxx.tgz) 的文件。
- tar不仅可以打包文件，也可以将硬盘数据备份
- 常用参数：
 - c: 创建一个新tar文件
 - v: 显示运行过程的信息
 - f: 指定文件名
 - z: 调用gzip压缩命令进行压缩
 - t: 查看压缩文件的内容
 - x: 解开tar文件

4. 1. 20 tar命令范例

- ❑ [root@linux test]# tar -cvf test.tar *
将所有文件打包成test.tar,扩展名.tar需自行加上
- ❑ [root@linux test]# tar -zcvf test.tar.gz *
将所有文件打包成test.tar,再用gzip命令压缩
- ❑ [root@linux ljr]# tar -tf test.tar
查看test.tar文件中包括了哪些文件
- ❑ [root@linux test]# tar -xvf test.tar
将test.tar解开
- ❑ [root@linux test]# tar -zxvf foo.tar.gz
将foo.tar.gz解压缩

4.1.20 gzip和gunzip

- 除了.zip文件的压缩格式外，在Linux系统下更常见的是.gz文件的压缩格式，这种文件一般是由gzip命令所产生。
- zip命令具有将许多文件压缩成一个文件的功能，但gzip却不能，所以gzip一般会与tar一起使用。
- 目前，大部分或见到的压缩文件都是用tar将所有文件打包成一个文件，再用gzip进行压缩，所以我们所看到的扩展名为.tar.gz或.tgz的文件，大多数就是这种类型的文件。

4.1.20 gzip和gunzip

❑ [root@linux test]# gzip test.txt

压缩文件，采用默认压缩比率，不需要其它参数

❑ [root@linux test]# gzip -l test.txt.gz

显示压缩比率

❑ [root@linux test]# gunzip test.txt.gz

文件解压缩

常见的几种压缩文件

□ .tar

- 解包: `tar xvf FileName.tar`
- 打包: `tar cvf FileName.tar DirName`
- (注: tar是打包, 不是压缩!)

□ .gz

- 解压1: `gunzip FileName.gz`
- 解压2: `gzip -d FileName.gz`
- 压缩: `gzip FileName`

□ .tar.gz 和 .tgz

- 解压: `tar zxvf FileName.tar.gz`
- 压缩: `tar zcvf FileName.tar.gz DirName`

常见的几种压缩文件

□ .bz2

- 解压1: `bzip2 -d FileName.bz2`
- 解压2: `bunzip2 FileName.bz2`
- 压缩: `bzip2 -z FileName`

□ .tar.bz2

- 解压: `tar jxvf FileName.tar.bz2`
- 压缩: `tar jcvf FileName.tar.bz2 DirName`

□ .bz

- 解压1: `bzip2 -d FileName.bz`
- 解压2: `bunzip2 FileName.bz`
- 压缩: 未知

□ .tar.bz

- 解压: `tar jxvf FileName.tar.bz`
- 压缩: 未知

常见的几种压缩文件

□ .Z

- 解压: `uncompress FileName.Z`
- 压缩: `compress FileName`

□ .tar.Z

- 解压: `tar Zxvf FileName.tar.Z`
- 压缩: `tar Zcvf FileName.tar.Z DirName`

□ .zip

- 解压: `unzip FileName.zip`
- 压缩: `zip FileName.zip DirName`

□ .rar

- 解压: `rar a FileName.rar`
- 压缩: `rar e FileName.rar`

常见的几种压缩文件

□ .lha

- 解压: `lha -e FileName.lha`
- 压缩: `lha -a FileName.lha FileName`
- lha下载: <http://www.infor.kanazawa-it.ac.jp/~ishii/lhaunix/>

□ .rpm

- 解包: `rpm2cpio FileName.rpm | cpio -div`

□ .deb

- 解包: `ar p FileName.deb data.tar.gz | tar xzf -`

4.1.21 date、cal、clock

- date命令可以显示/修改当前的日期时间
- [root@linux root]# date 121010232004
将时间更改为12月10日10点23分2004年 [MMDDhhmmYY]
- [root@linux root]# cal 显示日历
- [root@linux root]# clock 显示日期时间

```
[root@linux root]# date
Wed Mar  9 17:35:55 CST 2005
[root@linux root]# cal
      March 2005
Su Mo Tu We Th Fr Sa
    1  2  3  4  5
  6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

[root@linux root]# clock
Thu 10 Mar 2005 04:38:59 PM CST  0.832619 seconds
[root@linux root]# _
```

4. 1. 21 显示日历或年历的cal命令

■ cal(calendar)

□ 功能：显示一个日历

□ 格式：cal [参数] 月 年

□ [root@linux root]# cal
显示当月的日历

□ [root@linux root]# cal 4 2004
显示2004年4月的日历

□ [root@linux root]# cal - y 2003
显示2003年的日历

4.2 系统信息类命令

- dmesg命令
- df命令
- du命令
- free命令
- w命令

4.2.1 dmesg命令

- 功能：显示系统诊断信息、操作系统版本号、物理内存的大小以及其它信息

```
[root@host ~]# dmesg | more
Linux version 2.4.18-14 (bhcompile@stripples.devel.redhat.com) (gcc version 3.2
20020903 (Red Hat Linux 8.0 3.2-7)) #1 Wed Sep 4 13:35:50 EDT 2002
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
 BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
 BIOS-e820: 00000000000ca000 - 00000000000cc000 (reserved)
 BIOS-e820: 00000000000dc000 - 00000000000e0000 (reserved)
 BIOS-e820: 00000000000e4000 - 0000000000010000 (reserved)
 BIOS-e820: 0000000000010000 - 000000000007ef0000 (usable)
 BIOS-e820: 000000000007ef0000 - 000000000007eff000 (ACPI data)
 BIOS-e820: 000000000007eff000 - 000000000007f00000 (ACPI NVS)
 BIOS-e820: 000000000007f00000 - 000000000008000000 (usable)
 BIOS-e820: 00000000000fec0000 - 00000000000fec1000 (reserved)
 BIOS-e820: 00000000000fee0000 - 00000000000fee01000 (reserved)
 BIOS-e820: 00000000000ffe0000 - 00000000000100000000 (reserved)
0MB HIGHMEM available.
128MB LOWMEM available.
On node 0 totalpages: 32768
zone(0): 4096 pages.
zone(1): 28672 pages.
zone(2): 0 pages.
Kernel command line: auto BOOT_IMAGE=linux ro BOOT_FILE=/boot/vmlinuz-2.4.18-14
root=LABEL=/  

```


4.2.2 df命令

- 功能：用于查看文件系统的各个分区的占用情况

```
[root@linux root]# df
Filesystem      1k-blocks      Used Available Use% Mounted on
/dev/sda1        806368      156924     608480   21% /
/dev/sda3        600864       16468     553876    3% /home
none             30736         0       30736    0% /dev/shm
/dev/sda2       2419288     1713288     583104   75% /usr
[root@linux root]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       787M  154M  594M   21% /
/dev/sda3       587M   17M  540M    3% /home
none            30M    0    30M    0% /dev/shm
/dev/sda2       2.3G  1.7G  569M   75% /usr
[root@linux root]# _
```

4.2.3 du命令

- 功能：查看某个目录中各级子目录所使用的硬盘空间数
- 格式：du [参数] <目录名>

```
[root@linux root]# du
4      ../kde/share/config
4      ../kde/share/servicetypes
4      ../kde/share/mimelnk
4      ../kde/share/applnk
4      ../kde/share/services
24     ../kde/share
152    ../kde/tmp-localhost.localdomain
180    ../kde
252    .
[root@linux root]# du -h
4.0k   ../kde/share/config
4.0k   ../kde/share/servicetypes
4.0k   ../kde/share/mimelnk
4.0k   ../kde/share/applnk
4.0k   ../kde/share/services
24k    ../kde/share
152k   ../kde/tmp-localhost.localdomain
180k   ../kde
252k   .
[root@linux root]# _
```

4.2.4 free命令

- 功能：用于查看系统内存，虚拟内存（交换空间）的大小占用情况

```
[root@linux root]# free
              total        used        free      shared    buffers     cached
Mem:          61476       56264        5212          0       2528      29476
-/+ buffers/cache:      24260       37216
Swap:        128480         288      128192
[root@linux root]# free -m
              total        used        free      shared    buffers     cached
Mem:           60          54           5          0           2          28
-/+ buffers/cache:         23          36
Swap:          125           0          125
[root@linux root]# _
```

4.3 网络通讯类命令

- ftp 传输文件
- telnet 登录到远程计算机上
- netstat 查看网络的状况
- finger 查询某个使用者的信息
- ping 查询某个机器是否在工作
- route 设置系统网络路由

4.3.1 ftp传输文件

■ ftp登录后执行命令：

- ❑ ls 列出远程机的当前目录
- ❑ cd 在远程机上改变工作目录
- ❑ lcd 在本地机上改变工作目录
- ❑ ascii 设置文件传输方式为ASCII模式
- ❑ binary 设置文件传输方式为二进制模式
- ❑ close终止当前的ftp会话
- ❑ hash 每次传输完数据缓冲区中的数据后就显示一个#号
- ❑ get (mget) 从远程机传送指定文件到本地机
- ❑ put (mput) 从本地机传送指定文件到远程机
- ❑ open 连接远程ftp站点
- ❑ quit断开与远程机的连接并退出ftp
- ❑ ? 显示本地帮助信息
- ❑ ! 转到Shell中
- ❑ passive: 进入被动传输方式
- ❑ prompt: 设置多个文件传输时的交互提示

4.3.2 telnet 登录到远程计算机上

- telnet命令的一般形式

telnet 主机名/IP

其中“主机名/IP”是要连接的远程机的主机名或IP地址

- telnet只为普通终端提供终端仿真，而不支持 X Window等图形环境

4.3.3 netstat 查看网络的状况

■ netstat

- 作用：检查整个Linux网络状态。
- 格式：netstat [-acCeFghilMnNoprstuvVwx][-A][--ip]
- 主要参数
 - a --all：显示所有连线中的Socket。
 - c --continuous：持续列出网络状态。
 - e --extend：显示网络其它相关信息。
 - g --groups：显示多重广播功能群组组员名单。
 - i --interfaces：显示网络界面信息表单。
 - l --listening：显示监控中的服务器的Socket。
 - n --numeric：直接使用IP地址，而不通过域名服务器。
 - r --route：显示Routing Table。
 - s --statistic：显示网络工作信息统计表。
 - t --tcp：显示TCP传输协议的连线状况。
 - u --udp：显示UDP传输协议的连线状况。
 - v --verbose：显示指令执行过程。
 - w --raw：显示RAW传输协议的连线状况。

4.3.4 finger查询某个使用者的信息

- finger命令是显示有关运行 Finger 服务或 Daemon 的指定远程计算机上用户的详细信息。
- 该远程计算机指定显示用户信息的格式和输出。
- 命令中各选项的含义如下：
 - -s 显示用户的注册名、实际姓名、终端名称、写状态、停滞时间、登录时间等信息。
 - -l 除了用-s选项显示的信息外，还显示用户主目录、登录shell、邮件状态等信息，以及用户主目录下的.plan、.project和.forward文件的内容。
 - -p 除了不显示.plan文件和.project文件以外，与-l选项相同。

4.3.5 ping查询某个节点是否在线

- ping [选项] 主机名/IP地址
- 命令中各选项的含义如下：
 - -c 数目 在发送指定数目的包后停止
 - -f 大量且快速地送网络封包给一台机器，看它的回应
 - -l 秒数 设定间隔几秒送一个网络封包给一台机器，预设值是一秒送一次
 - -I 次数 在指定次数内，以最快的方式送封包数据到指定机器（只有超级用户可以使用此选项）
 - -q 不显示任何传送封包的信息，只显示最后的结果
 - -r 不经由网关而直接送封包到一台机器，通常是查看本机的网络接口是否有问题
 - -s 字节数 指定发送的数据字节数，预设值是56，加上8字节的ICMP头，一共是64ICMP数据字节

4.3.6 route 设置系统网络路由

- 在本地 IP 路由表中显示和修改条目
- `route [-f] [-p] [Command] [Destination] [mask Netmask] [metric Metric] [Gateway] dev [Interface]`
 - `-f` 清除所有网关入口的路由表;
 - `-p` 与 `add` 命令一起使用时使路由具有永久性;
 - `Command` 指定您想运行的命令 (Add/Change/Delete/Print);
 - `Destination` 指定该路由的网络目标;
 - `mask Netmask` 指定与网络目标相关的网络掩码 (也被称作子网掩码);
 - `metric Metric` 为路由指定一个整数成本值标 (从 1 至 9999), 当在路由表(与转发的数据包目标地址最匹配)的多个路由中进行选择时可以使用;
 - `Gateway` 指定网络目标定义的地址集和子网掩码可以到达的前进或下一跃点 IP 地址。

4.3.6 route路由设置

- [root@linux /]# route

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
127.0.0.0	127.0.0.1	255.0.0.0	U 0	0	0	0	lo
192.168.1.0	192.168.1.5	255.255.255.0	U 0	0	0	0	eth0

- 第一项是到 localhost 的回送路由，它是在配置 lo 时自动创建的。
- 第二项是通过接口 eth0 到网络 192.168.1.0 的路由。地址 192.168.1.5 不是远程网关地址。它是分配给 eth0 的地址。
- 注意每项的标志。它们都设置了 U（启动）标志，这表示准备使用它们，但它们都未设置 G（网关）标志。不设置 G 标志是因为这两个路由都是通过本地接口，而不是通过外部网关的直接路由。
- 为了规范，在/etc/sysconfig/下建立static-routes文件：

eth1	net 192.168.0.0/24	gw 192.168.0.254
any	net 192.168.0.0/24	gw 192.168.0.6

4.4 磁盘管理类命令-fdisk

■ fdisk

Disk /dev/sda: 73.5 GB, 73543163904 bytes

255 heads, 63 sectors/track, 8941 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	535	4192965	82	Linux swap
/dev/sda3		536	8941	67521195	83	Linux

■ 硬盘容量及分区大小的算法:

其中 heads 是磁盘面; sectors 是扇区; cylinders 是柱面; 每个扇区大小是512byte, 也就是0.5M;

■ 通过上面的例子, 我们发现此硬盘有 255个磁盘面, 有63个扇区, 有8941个柱面; 所以整个硬盘体积换算公式应该是:

磁面个数 x 扇区个数 x 每个扇区的大小512 x 柱面个数 = 硬盘体积 (bytes)

4.4 磁盘管理类命令-fdisk

■ fdisk 分区管理

```
[root@node1 ~]# fdisk /dev/sdb
```

```
Command (m for help): m
```

```
Command action
```

- a toggle a read only flag
- b edit bsd disklabel
- c toggle the mountable flag
- d delete a partition
- l list known partition types
- m print this menu
- n add a new partition
- o create a new empty DOS partition table
- p print the partition table
- q quit without saving changes
- s create a new empty Sun disklabel
- t change a partition's system id
- u change display/entry units
- v verify the partition table
- w write table to disk and exit
- x extra functionality (experts only)

4.4 磁盘管理类命令-fdisk

```
[root@node1 ~]# fdisk /dev/sda
```

The number of cylinders for this disk is set to 8941.

There is nothing wrong with that, but this is larger than 1024, and could in certain setups cause problems with:

- 1) software that runs at boot time (e.g., old versions of LILO)
- 2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/sda: 73.5 GB, 73543163904 bytes

255 heads, 63 sectors/track, 8941 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	535	4192965	82	Linux swap
/dev/sda3		536	8941	67521195	83	Linux

4.4 磁盘管理类命令-fdisk

```
[root@node1 ~]# fdisk /dev/sda
```

The number of cylinders for this disk is set to 8941.

There is nothing wrong with that, but this is larger than 1024, and could in certain setups cause problems with:

- 1) software that runs at boot time (e.g., old versions of LILO)
- 2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): n

Command action

e extended

p primary partition (1-4)

输入p创建主分区，然后回车

Command (m for help):w

写入硬盘分区表



计算 决定未来



THANKS