\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.1.1 ls

命令名称：ls 选项[-ald] [文件或目录]

功能描述：显示目录文件

-a 显示所有文件，包括隐藏文件

-l 详细信息显示

-d 查看目录属性

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.1.2

目录处理命令：cd

$ cd .. 回到上一级目录目录处理命令：pwd

命令名称：pwd

功能描述：显示当前目录

范例：$ pwd

命令名称：rmdir

语法：rmdir [目录名]

功能描述： 删除空目录

命令名称：mkdir

功能描述：创建新目录

-p 递归创建

命令名称：cp

语法：cp -rp [原文件或目录] [目标目录]

-r 复制目录

-p 保留文件属性

功能描述：复制文件或目录目录处理命令：cp

范例：

$ cp -r /tmp/Japan/cangjing /root

将目录/tmp/Japan/cangjing复制到目录/root下

$ cp -rp /tmp/Japan/boduo /tmp/Japan/longze /root

将/tmp/Japan目录下的boduo和longze目录复制到

/root下，保持目录属性

命令名称：mv

语法：mv [原文件或目录] [目标目录]

命令名称：rm

-r 删除目录

-f 强制执行

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.1.3

命令名称：cat

-n 显示行号

命令名称：tac

功能描述：显示文件内容（反向列示）

命令名称：more

语法：more [文件名]

(空格) 或f 翻页

(Enter) 换行

q或Q 退出

命令名称：touch

功能描述：创建空文件

命令名称：less

功能描述：分页显示文件内容（可向上翻页）

命令名称：head

功能描述：显示文件前面几行

-n 指定行数

命令名称：tail

功能描述：显示文件后面几行

-n 指定行数

-f 动态显示文件末尾内容

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.1.4

命令名称：ln

语法：ln -s [原文件] [目标文件]

-s 创建软链接

硬链接特征：

1、拷贝cp -p + 同步更新

echo "this is a test" >> /etc/motd

2、可通过i节点识别

3、不能跨分区

4、不能针对目录使用

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.2.1

命令名称：chmod

语法：chmod [{ugoa}{+-=}{rwx}] [文件或目录]

[mode=421 ] [文件或目录]

-R 递归修改

功能描述：改变文件或目录权限

范例：

$ chmod g+w testfile

赋予文件testfile所属组写权限

$ chmod -R 777 testdir

修改目录testfile及其目录下文件为所有用户具

有全部权限

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.2.2权限管理命令：chgrp

功能描述：改变文件或目录的所属组

命令名称：chown

语法：chown [用户] [文件或目录]

功能描述：改变文件或目录的所有者

命令名称：umask

语法：umask [-S]

-S 以rwx形式显示新建文件缺省权限

功能描述：显示、设置文件的缺省权限

范例： $ umask -S

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.3.1

命令名称：find

命令所在路径：/bin/find

执行权限：所有用户

语法：find [搜索范围] [匹配条件]

功能描述：文件搜索 -type 根据文件类型查找

f 文件 d 目录 l 软链接文件

-inum 根据i节点查找

$ find /etc -size +163840 -a -size -204800

在/etc下查找大于80MB小于100MB的文件

-a 两个条件同时满足

-o 两个条件满足任意一个即可

$ find /etc -name inittab -exec ls -l {} \;

在/etc下查找inittab文件并显示其详细信息

-exec/-ok 命令 {} \; 对搜索结果执行操作

$ find /etc -cmin -5

在/etc下查找5分钟内被修改过属性的文件和

目录

-amin 访问时间 access

-cmin 文件属性 change

-mmin 文件内容 modify

$ find /etc -name init

在目录/etc中查找文件init

-iname 不区分大小写

$ find / -size +204800

在根目录下查找大于100MB的文件

+n 大于 -n 小于 n 等于

$ find /home -user shenchao

在根目录下查找所有者为shenchao的文件

-group 根据所属组查找

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.3.2

文件搜索命令：whereis

功能描述：搜索命令所在目录及帮助文档路径

文件搜索命令：which

功能描述：搜索命令所在目录及别名信息

文件搜索命令：locate

功能描述：在文件资料库中查找文件

文件搜索命令：grep

功能描述：在文件中搜寻字串匹配的行并输出

-i 不区分大小写

-v 排除指定字串

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.4

帮助命令：man [命令或配置文件]

功能描述：获得帮助信息

命令名称：help

功能描述：获得Shell内置命令的帮助信息

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.5

用户管理命令：who

功能描述：查看登录用户信息

用户管理命令：passwd 用户名

功能描述：设置用户密码

用户管理命令：useradd 用户名

功能描述：添加新用户

用户管理命令：w

功能描述：查看登录用户详细信息

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.6

压缩解压命令：gunzip 【文件名】

功能描述：解压缩.gz的压缩文件

压缩解压命令：tar

语法：tar 选项[-zcf] [压缩后文件名] [目录]

-c 打包

-v 显示详细信息

-f 指定文件名

-z 打包同时压缩

功能描述：打包目录

压缩后文件格式：.tar.gz 压缩解压命令：tar

范例：

$ tar -zcf Japan.tar.gz Japan

压缩解压命令：gzip

功能描述：压缩文件

压缩解压命令：tar

tar命令解压缩语法：

-x 解包

-v 显示详细信息

-f 指定解压文件

-z 解压缩

范例：$ tar -zxvf Japan.tar.gz

压缩解压命令：zip

语法：

zip 选项[-r] [压缩后文件名] [文件或目录]

-r 压缩目录

压缩后文件格式：.zip

范例：

$ zip buduo.zip boduo

压缩文件

$ zip -r Japan.zip Japan

压缩目录

压缩解压命令：unzip

命令名称：unzip

功能描述：解压.zip的压缩文件

范例：$ unzip test.zip

压缩解压命令：bunzip2

语法： bunzip2 选项 [-k] [压缩文件]

-k 解压缩后保留原文件

功能描述：解压缩

范例：$ bunzip2 -k boduo.bz2

$ tar -xjf Japan.tar.bz2

压缩解压命令：bzip2

语法： bzip2 选项 [-k] [文件]

-k 产生压缩文件后保留原文件

功能描述：压缩文件

压缩后文件格式：.bz2

范例：$ bzip2 -k boduo

$ tar -cjf Japan.tar.bz2 Japan

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.7

网络命令：wall

指令名称：wall

命令英文原意：write all

语法：wall [message]

功能描述：发广播信息

范例： # wall ShenChao is a honest man!

网络命令：ping

语法：ping 选项 IP地址

-c 指定发送次数

功能描述：测试网络连通性

网络命令：write

语法：write <用户名>

功能描述：给用户发信息，以Ctrl+D保存结束

范例： # write linzhiling

网络命令：mail

功能描述：查看发送电子邮件

范例：# mail root

网络命令：ifconfig

执行权限：root

语法：ifconfig 网卡名称 IP地址

功能描述：查看和设置网卡信息

范例：# ifconfig eth0 192.168.8.250

网络命令：lastlog

功能描述：检查某特定用户上次登录的时间

范例：# lastlog

# lastlog -u 502

网络命令：last

功能描述：列出目前与过去登入系统的用户信息

网络命令：traceroute

功能描述：显示数据包到主机间的路径

网络命令：netstat

功能描述：显示网络相关信息

选项：

-t ： TCP协议

-u ： UDP协议

-l ： 监听

-r ： 路由

-n ： 显示IP地址和端口号

范例：

# netstat -tlun 查看本机监听的端口

# netstat -an 查看本机所有的网络连接

# netstat -rn 查看本机路由表

网络命令：setup

执行权限：root

功能描述：配置网络

命令名称：mount

命令语法：mount [-t 文件系统] 设备文件名 挂载点

范例：# mount -t iso9660 /dev/sr0 /mnt/cdrom

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 4.7

1、shutdown命令

[root@localhost ~]# shutdown [选项] 时间

选项：

-c： 取消前一个关机命令

-h： 关机

-r： 重启

2、其他关机命令

[root@localhost ~]# halt

[root@localhost ~]# poweroff

[root@localhost ~]# init 0

3、其他重启命令

[root@localhost ~]# reboot

[root@localhost ~]# init 6

4、系统运行级别

0 关机

1 单用户

2 不完全多用户，不含NFS服务

3 完全多用户

4 未分配

5 图形界面

6 重启

[root@localhost ~]# cat /etc/inittab

#修改系统默认运行级别

id:3:initdefault:

[root@localhost ~]# runlevel

#查询系统运行级别

5、退出登录命令

[root@localhost ~]# logout

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 5.1

Vim

定位命令

$ 移至行尾

0 移至行首

: set nu 设置行号

: set nonu 取消行号

gg 到第一行

G 到最后一行

: n 到第n行

删除命令

:n1,n2d 删除指定范围的行

D 删除光标所在处到行尾内容

dG 删除光标所在行到文件末尾内容

dd 删除光标所在行，ndd删除n行

nx 删除光标所在处后n个字符

x 删除光标所在处字符

复制和剪切命令

yy 复制当前行

nyy 复制当前行以下n行

dd 剪切当前行

ndd 剪切当前行以下n行

p、P 粘贴在当前光标所在行下或行上

替换和取消命令

命令 作用

r 取代光标所在处字符

R 从光标所在处开始替换字

符，按Esc结束

u 取消上一步操作

搜索和搜索替换命令

/string 搜索指定字符串

搜索时忽略大小写 :set ic

n 搜索指定字符串的下一个出现位置

:%s/old/new/g 全

文替换指定字符串

:n1,n2s/old/new/g 在一定范围内替换指定字符串

保存和退出命令

:wq! 保存修改并退出（文件所有者及root可使用）

:q! 不保存修改退出

ZZ 快捷键，保存修改并退出

:wq 保存修改并退出

:w new\_filename 另存为指定文件

:w 保存修改

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 5.2

导入命令执行结果 :r !命令

定义快捷键 :map 快捷键 触发命令

范例： : map ^P I#<ESC>

: map ^B 0x

连续行注释 :n1,n2s/^/#/g

:n1,n2s/^#//g

:n1,n2s/^/\/\//g

替换 :ab mymail samlee@lampbrother.net

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 6.2.1

模块依赖查询网站

www.rpmfind.net

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

6.2.2

RPM安装

rpm –ivh 包全名

选项：

-i（install） 安装

-v（verbose） 显示详细信息

-h（hash） 显示进度

--nodeps 不检测依赖性

RPM包升级

rpm -Uvh 包全名

选项：

-U（upgrade） 升级

卸载

rpm -e 包名

选项：

-e（erase） 卸载

--nodeps 不检查依赖性

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 6.3

查询是否安装

[root@localhost ~]# rpm -q 包名

#查询包是否安装

选项：

-q 查询（query）

[root@localhost ~]# rpm –qa

#查询所有已经安装的RPM包

选项：

-a 所有（all）

查询软件包详细信息

[root@localhost ~]# rpm –qi 包名

选项：

-i 查询软件信息（information）

-p 查询未安装包信息（package）

查询包中文件安装位置

[root@localhost ~]# rpm –ql 包名

选项：

-l 列表（list）

-p 查询未安装包信息（package）

查询系统文件属于哪个RPM包

[root@localhost ~]# rpm –qf 系统文件名

选项：

-f 查询系统文件属于哪个软件包（file）

查询软件包的依赖性

[root@localhost ~]# rpm –qR 包名

选项：

-R 查询软件包的依赖性（requires）

-p 查询未安装包信息（package）

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 6.4

RPM包校验

[root@localhost ~]# rpm –V 已安装的包名

选项：

-V 校验指定RPM包中的文件（verify）

验证内容中的8个信息的具体内容如下：

S 文件大小是否改变

M 文件的类型或文件的权限（rwx）是否被改变

5 文件MD5校验和是否改变（可以看成文件内容是否

改变）

D 设备的中，从代码是否改变

L 文件路径是否改变

U 文件的属主（所有者）是否改变

G 文件的属组是否改变

T 文件的修改时间是否改变

文件类型

c 配置文件（config file）

d 普通文档（documentation）

g “鬼”文件（ghost file），很少见，就是该文件不

应该被这个RPM包包含

l 授权文件（license file）

r 描述文件（read me）

RPM包中文件提取

[root@localhost ~]# rpm2cpio 包全名 | \

cpio -idv .文件绝对路径

rpm2cpio

#将rpm包转换为cpio格式的命令

cpio

#是一个标准工具，它用于创建软件档案文件和从档案文件中提取文件

[root@localhost ~]# cpio 选项 < [文件|设备]

选项：

-i：copy-in模式，还原

-d：还原时自动新建目录

-v：显示还原过程

[root@localhost ~]# rpm -qf /bin/ls

#查询ls命令属于哪个软件包

[root@localhost ~]# mv /bin/ls /tmp/

#造成ls命令误删除假象

[root@localhost ~]# rpm2cpio /mnt/cdrom/Packages/coreutils-

8.4-19.el6.i686.rpm | cpio -idv ./bin/ls

#提取RPM包中ls命令到当前目录的/bin/ls下

[root@localhost ~]# cp /root/bin/ls /bin/

#把ls命令复制会/bin/目录，修复文件丢失

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 6.3.1

yum在线管理 ip地址配置 网络yum源

IP地址配置

[root@localhost ~]# setup

#使用setup工具

[root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth0

把ONBOOT=“no” 改为

ONBOOT=“yes“

#启动网卡

[root@localhost ~]# service network restart

#重启网络服务

网络yum源

[root@localhost yum.repos.d]# vi /etc/yum.repos.d/CentOS-Base.repo

 [base] 容器名称，一定要放在[]中

 name 容器说明，可以自己随便写

 mirrorlist 镜像站点，这个可以注释掉

 baseurl 我们的yum源服务器的地址。默认是CentOS官方的yum源服务

器，是可以使用的，如果你觉得慢可以改成你喜欢的yum源地

址

 enabled 此容器是否生效，如果不写或写成enable=1都是生效，写成

enable=0就是不生效

 gpgcheck 如果是1是指RPM的数字证书生效，如果是0则不生效

 gpgkey 数字证书的公钥文件保存位置。不用修改

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 6.3.2

yum在线安装 光盘yum源

1）查询

[root@localhost yum.repos.d]# yum list

#查询所有可用软件包列表

[root@localhost yum.repos.d]# yum search 关键字

#搜索服务器上所有和关键字相关的包

2）安装

[root@localhost yum.repos.d]# yum –y install 包名

选项：

install 安装

-y 自动回答yes

3）升级

[root@localhost yum.repos.d]# yum -y update 包名

选项：

update 升级

[root@localhost yum.repos.d]# yum -y update

全部升级，包括内核升级。

-y 自动回答yes

4）卸载

[root@localhost yum.repos.d]# yum -y remove 包名

选项：

remove 卸载

-y 自动回答yes

YUM软件组管理命令

[root@localhost ~]# yum grouplist

#列出所有可用的软件组列表

[root@localhost ~]# yum groupinstall 软件组名

#安装指定软件组，组名可以由grouplist查询出来

[root@localhost ~]# yum groupremove 软件组名

#卸载指定软件组

光盘yum源搭建步骤

1）挂载光盘

[root@localhost ~]# mount /dev/cdrom /mnt/cdrom/

2）让网络yum源文件失效

[root@localhost ~]# cd /etc/yum.repos.d/

[root@localhost yum.repos.d]# mv CentOS-Base.repo \

CentOS-Base.repo.bak

[root@localhost yum.repos.d]# mv CentOS-Debuginfo.repo \

CentOS-Debuginfo.repo.bak

[root@localhost yum.repos.d]# mv CentOS-Vault.repo \

CentOS-Vault.repo.bak

3）修改光盘yum源文件

[root@localhost yum.repos.d]# vim CentOS-Media.repo

[c6-media]

name=CentOS-$releasever - Media

baseurl=file:///mnt/cdrom

#地址为你自己的光盘挂载地址

# file:///media/cdrom/

# file:///media/cdrecorder/

#注释这两个不存在的地址

gpgcheck=1

enabled=1

#把enabled=0改为enabled=1，让这个yum源配置文件生效

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 6.4.1

源码包与RPM包的区别

2、 RPM包安装位置

是安装在默认位置中

RPM包默认安装路径

/etc/ 配置文件安装目录

/usr/bin/ 可执行的命令安装目录

/usr/lib/ 程序所使用的函数库保存位置

/usr/share/doc/ 基本的软件使用手册保存位置

/usr/share/man/ 帮助文件保存位置

3、源码包安装位置

安装在指定位置当中，一般是

/usr/local/软件名/

4、安装位置不同带来的影响

 RPM包安装的服务可以使用系统服务管

理命令（ service）来管理，例如RPM包安

装的apache的启动方法是：

 /etc/rc.d/init.d/httpd start

 service httpd start //redhat centos 专有命令

而源码包安装的服务则不能被服务管理命

令管理，因为没有安装到默认路径中。所

以只能用绝对路径进行服务的管理，如：

 /usr/local/apache2/bin/apachectl start

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

6.4.2 源码包安装过程

安装注意事项

 源代码保存位置： /usr/local/src/

 软件安装位置： /usr/local/

 如何确定安装过程报错：

 安装过程停止

 并出现error、 warning或no的提示

3、源码包安装过程

 下载源码包

 解压缩下载的源码包

tar -zxvf httpd-2.2.9.tar.gz

du -sh httd-2.2.9 //查看解压后文件大小

 进入解压缩目录

 ./configure 软件配置与检查

 定义需要的功能选项。

 检测系统环境是否符合安装要求。

 把定义好的功能选项和检测系统环境的信

都写入Makefile文件，用于后续的编辑。

./configure --prefix=/usr/local/apache2

 make 编译

 make clean

 make install 编译安装

运行

/usr/local/apache2/bin/apchectl start

4、源码包的卸载

 不需要卸载命令，直接删除安装目录即可。

不会遗留任何垃圾文件。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

6.5 脚本安装包

1、脚本安装包

脚本安装包并不是独立的软件包类型，常见

安装的是源码包。

是人为把安装过程写成了自动安装的脚本，

只要执行脚本，定义简单的参数，就可以完

成安装。

2、 Webmin的作用

 Webmin 是一个基于 Web 的 Linux 系统管

理界面。您就可以通过图形化的方式设置

用户帐号、 Apache、 DNS、文件共享等服

务。

非常类似于Windows下软件的安装方式。

3、 Webmin安装过程

 下载软件

 http://sourceforge.net/projects/webadmin/files/

webmin/

 解压缩，并进入加压缩目录

 执行安装脚本

./setup.sh ctrl+退格键 进行删除错误的字符

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7.1.1 第七章 用户和用户组管理

1、用户管理简介

 所以越是对服务器安全性要求高的服务器

，越需要建立合理的用户权限等级制度和

服务器操作规范。

 在Linux中主要是通过用户配置文件来查

看和修改用户信息

2、 /etc/passwd

 第1字段： 用户名称

 第2字段：密码标志

 第3字段： UID（用户ID）

 0： 超级用户

 1-499： 系统用户（伪用户）

 500-65535： 普通用户

第4字段： GID（用户初始组ID）

第5字段：用户说明

第6字段：家目录

 普通用户： /home/用户名/

 超级用户： /root/

第7字段：登录之后的Shell

3、初始组和附加组

 初始组： 就是指用户一登录就立刻拥有这

个用户组的相关权限，每个用户的初始组

只能有一个，一般就是和这个用户的用户

名相同的组名作为这个用户的初始组。

 附加组： 指用户可以加入多个其他的用户

组，并拥有这些组的权限， 附加组可以有

多个。

4、 Shell是什么

 Shell就是Linux的命令解释器。

 在/etc/passwd当中，除了标准Shell是

/bin/bash之外，还可以写如/sbin/nologin，

/usr/bin/passwd等。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7.1.2 影子文件/etc/shadow

1、影子文件/etc/shadow

 第1字段：用户名

 第2字段：加密密码

 加密算法升级为SHA512散列加密算法

 如果密码位是“ !!”或“ \*”代表没有密码，不能登

录

 第3字段：密码最后一次修改日期

 使用1970年1月1日作为标准时间，每过一天时间戳

加1

第4字段：两次密码的修改间隔时间（和

第3字段相比）

第5字段：密码有效期（和第3字段相比）

第6字段：密码修改到期前的警告天数（

和第5字段相比）

第7字段： 密码过期后的宽限天数（和第5

字段相比）

 0： 代表密码过期后立即失效

 -1： 则代表密码永远不会失效。

第8字段： 账号失效时间

 要用时间戳表示

第9字段： 保留

2、时间戳换算

 把时间戳换算为日期

 date -d "1970-01-01 16066 days"

 把日期换算为时间戳

 echo $(($(date --date="2014/01/06" +%s)/86400+1))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7.1.3 组信息文件/etc/group和组密码文件

/etc/gshadow

1、组信息文件/etc/group

 第一字段：组名

 第二字段：组密码标志

 第三字段： GID

 第四字段：组中附加用户

2、组密码文件/etc/gshadow

第一字段：组名

第二字段：组密码

第三字段：组管理员用户名

第四字段：组中附加用户

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7.2、用户管理相关文件

1、用户的家目录

 普通用户： /home/用户名/，所有者和所属

组都是此用户，权限是700

 超级用户： /root/，所有者和所属组都是

root用户，权限是550

2、用户的邮箱

 /var/spool/mail/用户名/

3、 用户模板目录

 /etc/skel/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7.3.1 用户添加命令useradd

1、 useradd命令格式

[root@localhost ~]#useradd [选项] 用户名

选项：

-u UID： 手工指定用户的UID号

-d 家目录： 手工指定用户的家目录

-c 用户说明： 手工指定用户的说明

-g 组名： 手工指定用户的初始组

-G 组名： 指定用户的附加组

-s shell： 手工指定用户的登录shell。默认是/bin/bash

2、添加默认用户

 [root@localhost ~]# useradd lamp

 [root@localhost ~]# grep "lamp" /etc/passwd

 [root@localhost ~]# grep "lamp" /etc/shadow

 [root@localhost ~]# grep "lamp" /etc/group

 [root@localhost ~]# grep "lamp" /etc/gshadow

 [root@localhost ~]# ll -d /home/lamp/

 [root@localhost ~]# ll /var/spool/mail/lamp

3、指定选项添加用户

 groupadd lamp1

 useradd -u 550 -g lamp1 -G root -d /home/lamp1 \

-c "test user" -s /bin/bash lamp1

4、用户默认值文件

 /etc/default/useradd

 GROUP=100 #用户默认组

 HOME=/home #用户家目录

 INACTIVE=-1 #密码过期宽限天数（ 7）

 EXPIRE= #密码失效时间（ 8）

 SHELL=/bin/bash #默认shell

 SKEL=/etc/skel #模板目录

 CREATE\_MAIL\_SPOOL=yes #是否建立邮箱

 /etc/login.defs

 PASS\_MAX\_DAYS 99999 #密码有效期（ 5）

 PASS\_MIN\_DAYS 0 #密码修改间隔（ 4）

 PASS\_MIN\_LEN 5 #密码最小5位（ PAM）

 PASS\_WARN\_AGE 7 #密码到期警告（ 6）

 UID\_MIN 500 #最小和最大UID范围

 GID\_MAX 60000

 ENCRYPT\_METHOD SHA512

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7.3.2 修改用户密码passwd

SecureCRT修正乱码：

Appearnce->

Current color scheme =>Tranditional

Font=>微软雅黑 字符集 GB2312

Character encoding =>UTF-8

用户切换命令su

1、 passwd命令格式

[root@localhost ~]#passwd [选项] 用户名

选项：

-S 查询用户密码的密码状态。仅root用户

可用。

-l 暂时锁定用户。仅root用户可用

-u 解锁用户。仅root用户可用

--stdin 可以通过管道符输出的数据作为用户

的密码。

2、查看密码状态

 [root@localhost ~]# passwd -S lamp

lamp PS 2013-01-06 0 99999 7 -1

#用户名密码设定时间（ 2013-01-06） 密码修改间隔时间（ 0）

#密码有效期（ 99999 ） 警告时间（ 7） 密码不失效（ -1）

3、锁定用户和解锁用户

 [root@localhost ~]# passwd -l lamp

 [root@localhost ~]# passwd -u lamp

4、 使用字符串作为用户的密码

[root@localhost ~]# echo "123" | passwd --stdin lamp

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7.3.3 修改用户信息usermod

修改用户密码状态chage

1、修改用户信息usermod

[root@localhost ~]#usermod [选项] 用户名

选项：

-u UID： 修改用户的UID号

-c 用户说明： 修改用户的说明信息

-G 组名： 修改用户的附加组

-L： 临时锁定用户（Lock）

-U： 解锁用户锁定（Unlock）

u[root@localhost ~]# usermod -c "test user" lamp

#修改用户的说明

u[root@localhost ~]# usermod -G root lamp

#把lamp用户加入root组

u[root@localhost ~]# usermod -L lamp

#锁定用户

u[root@localhost ~]# usermod -U lamp

#解锁用户

2、修改用户密码状态chage

[root@localhost ~]#chage [选项] 用户名

选项：

-l： 列出用户的详细密码状态

-d 日期： 修改密码最后一次更改日期（shadow3字段）-m 天数： 两次密码修改间隔（4字段）

-M 天数： 密码有效期（5字段）

-W 天数： 密码过期前警告天数（6字段）

-I 天数： 密码过后宽限天数（7字段）

-E 日期： 账号失效时间（8字段）

u[root@localhost ~]# chage -d 0 lamp

#这个命令其实是把密码修改日期归0了（shadow第3字段）#这样用户一登陆就要修改密码

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7.3.4 删除用户userdel

用户切换命令su

1、删除用户userdel

[root@localhost ~]# userdel [-r] 用户名

选项：

-r 删除用户的同时删除用户家目录

手工删除用户

 [root@localhost ~]# vi /etc/passwd

 [root@localhost ~]# vi /etc/shadow

 [root@localhost ~]# vi /etc/group

 [root@localhost ~]# vi /etc/gshadow

 [root@localhost ~]# rm -rf /var/spool/mail/lamp

 [root@localhost ~]# rm -rf /home/lamp/

2、查看用户ID

[root@localhost ~]# id 用户名

3、 切换用户身份su

[root@localhost ~]# su [选项] 用户名

选项：

- ： 选项只使用“ -”代表连带用户的环境

变量一起切换

-c 命令： 仅执行一次命令，而不切换用户身份

 [lamp@localhost ~]$ su – root

#切换成root

 [lamp@localhost ~]$ su - root -c "useradd user3"

#不切换成root，但是执行useradd命令添加user1用户

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7.4 用户组管理命令

1、 添加用户组

[root@localhost ~]#groupadd [选项] 组名

选项：

-g GID： 指定组 ID

2、 修改用户组

[root@localhost ~]#groupmod [选项] 组名

选项：

-g GID： 指定组 ID

-n 新组名: 修改组名

3、 删除用户组

[root@localhost ~]#groupdel 组名

4、把用户添加入组或从组中删除

[root@localhost ~]#gpasswd 选项 组名

选项：

-a 用户名: 把用户加入组

-d 用户名: 把用户从组中删除

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

第八章 权限管理

8.1 ACL权限

1、 ACL权限简介

df -h 查看分区使用状况

2、查看分区ACL权限是否开启

[root@localhost ~]# dumpe2fs -h /dev/sda3

#dumpe2fs命令是查询指定分区详细文件系统信息的命令

选项：

-h 仅显示超级块中信息，而不显示磁盘块组的详细信

息

3、临时开启分区ACL权限

[root@localhost ~]# mount -o remount,acl /

#重新挂载根分区，并挂载加入acl权限

4、永久开启分区ACL权限

[root@localhost ~]# vi /etc/fstab

UUID=c2ca6f57-b15c-43ea-bca0-f239083d8bd2 / ext4 defaults,acl 1 1

#加入acl

[root@localhost ~]# mount -o remount /

#重新挂载文件系统或重启动系统，使修改生效

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8.1.2 查看与设定ACL权限

1、查看ACL命令

[root@localhost ~]# getfacle 文件名

#查看acl权限

2、设定ACL权限的命令

[root@localhost ~]# setfacl 选项 文件名

选项：

-m 设定ACL权限

-x 删除指定的ACL权限

-b 删除所有的ACL权限

-d 设定默认ACL权限。

-k 删除默认ACL权限

-R 递归设定ACL权限。

3、给用户设定ACL权限

[root@localhost ~]# useradd zhangsan

[root@localhost ~]# useradd lisi

[root@localhost ~]# useradd st

[root@localhost ~]# groupadd tgroup

[root@localhost ~]# mkdir /project

[root@localhost ~]# chown root:tgroup /project/

[root@localhost ~]# chmod 770 /project/

[root@localhost ~]# setfacl -m u:st:rx /project/

#给用户st赋予r-x权限，使用“ u:用户名:权限”格式

4、给用户组设定ACL权限

[root@localhost /]# groupadd tgroup2

[root@localhost /]# setfacl -m g:tgroup2:rwx project/

#为组tgroup2分配ACl权限。使用“ g:组名:权限”格式

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8.1.3 最大有效权限与删除ACL权限

1、最大有效权限mask

 mask是用来指定最大有效权限的。如果

我给用户赋予了ACL权限，是需要和

mask的权限“相与”才能得到用户的真

正权限

A B and

r r r

r - -

- r -

- - -

修改最大有效权限

[root@localhost /]# setfacl -m m:rx 文件名

#设定mask权限为r-x。使用“ m:权限”格式

2、删除ACL权限

[root@localhost /]# setfacl -x u:用户名 文件名

#删除指定用户的ACL权限

[root@localhost /]# setfacl -x g:组名 文件名

#删除指定用户组的ACL权限

[root@localhost /]# setfacl -b 文件名

#会删除文件的所有的ACL�

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8.1.4 默认ACL权限和递归ACL权限

1、递归ACL权限

 递归是父目录在设定ACL权限时，所有的子文

件和子目录也会拥有相同的ACL权限。

 setfacl -m u:用户名:权限 -R 文件名

2、默认ACL权限

 默认ACL权限的作用是如果给父目录设定

了默认ACL权限，那么父目录中所有新建

的子文件都会继承父目录的ACL权限。

 setfacl -m d:u:用户名:权限 文件名

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8.2.1 SetUID

1、 SetUID的功能

 只有可以执行的二进制程序才能设定SUID权限

 命令执行者要对该程序拥有x（执行）权限

 命令执行者在执行该程序时获得该程序文件属

主的身份（在执行程序的过程中灵魂附体为文

件的属主）

 SetUID权限只在该程序执行过程中有效，也就

是说身份改变只在程序执行过程中有效

 passwd命令拥有SetUID权限，所以普通可以修

改自己的密码

[root@localhost ~]# ll /usr/bin/passwd

-rwsr-xr-x. 1 root root 25980 2月 22 2012 /usr/bin/passwd

 cat命令没有SetUID权限，所以普通用户不能查

看/etc/shadow文件内容

[root@localhost ~]# ll /bin/cat

-rwxr-xr-x 1 root root 47976 6月 22 2012 /bin/cat

2、设定SetUID的方法

 4代表SUID

 chmod 4755 文件名

 chmod u+s 文件名

3、取消SetUID的方法

 chmod 755 文件名

 chmod u-s 文件名

4、危险的SetUID

 关键目录应严格控制写权限。比如“ /”

、“ /usr” 等

 用户的密码设置要严格遵守密码三原则

 对系统中默认应该具有SetUID权限的文件

作一列表，定时检查有没有这之外的文件

被设置了

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8.2.2 SetGID

1、 SetGID针对文件的作用

 只有可执行的二进制程序才能设置SGID权限

 命令执行者要对该程序拥有x（执行）权限

 命令执行在执行程序的时候，组身份升级为该

程序文件的属组

 SetGID权限同样只在该程序执行过程中有效，

也就是说组身份改变只在程序执行过程中有效

[root@localhost ~]# ll /usr/bin/locate

-rwx--s--x 1 root slocate 35612 8月 24 2010 /usr/bin/locate

[root@localhost ~]# ll /var/lib/mlocate/mlocate.db

-rw-r----- 1 root slocate 1838850 1月 20 04:29 /var/lib/mlocate/mlocate.db

 /usr/bin/locate是可执行二进制程序，可以赋予

SGID

 执行用户lamp对/usr/bin/locate命令拥有执行权限

 执行/usr/bin/locate命令时，组身份会升级为slocate

组，而slocate组对/var/lib/mlocate/mlocate.db数据

库拥有r权限，所以普通用户可以使用locate命令

查询mlocate.db数据库

 命令结束， lamp用户的组身份返回为lamp组

2、 SetGID针对目录的作用

 普通用户必须对此目录拥有r和x权限，才

能进入此目录

 普通用户在此目录中的有效组会变成此目

录的属组

 若普通用户对此目录拥有w权限时，新建

的文件的默认属组是这个目录的属组

[root@localhost ~]# cd /tmp/

[root@localhost tmp]# mkdir dtest

[root@localhost tmp]# chmod g+s dtest

[root@localhost tmp]# ll -d dtest/

[root@localhost tmp]# chmod 777 dtest/

[root@localhost tmp]# su – lamp

[lamp@localhost ~]$ cd /tmp/dtest/

[lamp@localhost dtest]$ touch abc

[lamp@localhost dtest]$ ll

3、设定SetGID

 2代表SGID

 chmod 2755 文件名

 chmod g+s 文件名

4、取消SetGID

 chmod 755 文件名



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8.2.3 Sticky BIT

1、 SBIT粘着位作用

 粘着位目前只对目录有效

 普通用户对该目录拥有w和x权限，即普通用户

可以在此目录拥有写入权限

 如果没有粘着位，因为普通用户拥有w权限，

所以可以删除此目录下所有文件，包括其他用

户建立的文件。一但赋予了粘着位，除了root可

以删除所有文件，普通用户就算拥有w权限，

也只能删除自己建立的文件，但是不能删除其

他用户建立的文件

[root@localhost ~]# ll -d /tmp/

drwxrwxrwt. 3 root root 4096 12月 13 11:22 /tmp/

2、设置与取消粘着位

 设置粘着位

 chmod 1755 目录名

 chmod o+t 目录名

 取消粘着位

 chmod 777 目录名

 chmod o-t

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8.3 文件系统属性chattr权限

1、 chattr命令格式

[root@localhost ~]# chattr [+-=] [选项] 文件或目录名

+: 增加权限

-: 删除权限

=: 等于某权限

选项

 i： 如果对文件设置i属性，那么不允许对文件进行

删除、改名，也不能添加和修改数据；如果对目录

设置i属性，那么只能修改目录下文件的数据，但

不允许建立和删除文件。

 a： 如果对文件设置a属性，那么只能在文件中增加

数据，但是不能删除也不能修改数据；如果对目录

设置a属性，那么只允许在目录中建立和修改文件

，但是不允许删除

2、查看文件系统属性

[root@localhost ~]# lsattr 选项 文件名

选项：

-a 显示所有文件和目录

-d 若目标是目录，仅列出目录本身的属

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8.4 系统命令sudo权限

1、 sudo权限

 root把本来只能超级用户执行的命令赋予

普通用户执行。

 sudo的操作对象是系统命令

2、 sudo使用

[root@localhost ~]# visudo

#实际修改的是/etc/sudoers文件

root ALL=(ALL) ALL

#用户名 被管理主机的地址=（可使用的身份） 授权命令（绝对路径）

# %wheel ALL=(ALL) ALL

#%组名 被管理主机的地址=（可使用的身份） 授权命令（绝对路径）

3、授权sc用户可以重启服务器

[root@localhost ~]# visudo

sc ALL= /sbin/shutdown –r now

4、普通用户执行sudo赋予的命令

[root@localhost ~]# su – sc

[sc@localhost ~]$ sudo -l

#查看可用的sudo命令

[lamp@localhost ~]$ sudo /sbin/shutdown -r now

#普通用户执行sudo赋予的命令

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

第九讲 文件系统管理

9.1 回顾分区和文件系统

1、分区类型

 主分区：总共最多只能分四个

 扩展分区：只能有一个，也算作主分区的一种

，也就是说主分区加扩展分区最多有四个。但

是扩展分区不能存储数据和格式化，必须再划

分成逻辑分区才能使用。

 逻辑分区：逻辑分区是在扩展分区中划分的，

如果是IDE硬盘， Linux最多支持59个逻辑分区

，如果是SCSI硬盘Linux最多支持11个逻辑分区

2、分区表示方法

分区的设备文件名

主分区1 /dev/sda1

主分区2 /dev/sda2

主分区3 /dev/sda3

扩展分区 /dev/sda4

逻辑分区1 /dev/sda5

逻辑分区2 /dev/sda6

逻辑分区3 /dev/sda7

分区的设备文件名

主分区1 /dev/sda1

扩展分区 /dev/sda2

逻辑分区1 /dev/sda5

逻辑分区2 /dev/sda6

逻辑分区3 /dev/sda7

2、文件系统

 ext2： 是ext文件系统的升级版本， Red

Hat Linux7.2版本以前的系统默认都是ext2

文件系统。 1993年发布，最大支持16TB

的分区和最大2TB的文件（

1TB=1024GB=1024\*1024KB）

 ext3： ext3文件系统是ext2文件系统的升

级版本，最大的区别就是带日志功能，以

在系统突然停止时提高文件系统的可靠性

。支持最大16TB的分区和最大2TB的文件

 ext4： 它是ext3文件系统的升级版。 ext4 在性能

、伸缩性和可靠性方面进行了大量改进。 EXT4

的变化可以说是翻天覆地的，比如向下兼容

EXT3、最大1EB文件系统和16TB文件、无限数

量子目录、 Extents连续数据块概念、多块分配

、延迟分配、持久预分配、快速FSCK、日志校

验、无日志模式、在线碎片整理、 inode增强、

默认启用barrier等。是

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9.2 文件系统常用命令

9.2.1 文件系统管理-文件系统常用命令-df、du、fsck、dumpe2fs

1、文件系统查看命令df

[root@localhost ~]# df [选项] [挂载点]

选项：

-a 显示所有的文件系统信息，包括特殊文件系统，如

/proc、 /sysfs

-h 使用习惯单位显示容量，如KB， MB或GB等

-T 显示文件系统类型

-m 以MB为单位显示容量

-k 以KB为单位显示容量。默认就是以KB为单位

2、统计目录或文件大小

[root@localhost ~]# du [选项] [目录或文件名]

选项：

-a 显示每个子文件的磁盘占用量。默认只统计

子目录的磁盘占用量

-h 使用习惯单位显示磁盘占用量，如KB， MB

或GB等

-s 统计总占用量，而不列出子目录和子文件的

占用量

du命令和df命令的区别

 df命令是从文件系统考虑的， 不光要考虑

文件占用的空间，还要统计被命令或程序

占用的空间（最常见的就是文件已经删除

，但是程序并没有释放空间）

 du命令是面向文件的，只会计算文件或目

录占用的空间

3、文件系统修复命令fsck

[root@localhost ~]# fsck [选项] 分区设备文件名

选项：

-a： 不用显示用户提示，自动修复文件系统

-y： 自动修复。和-a作用一致，不过有些文件系统只支

持-y

4、 显示磁盘状态命令dumpe2fs [-h]

[root@localhost ~]# dumpe2fs 分区设备文件名

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9.2.2 挂载命令

1、查询与自动挂载

[root@localhost ~]# mount [-l]

#查询系统中已经挂载的设备， -l会显示卷标名称

[root@localhost ~]# mount –a

#依据配置文件/etc/fstab的内容，自动挂载

2、挂载命令格式

[root@localhost ~]# mount [-t 文件系统] [-L 卷标名] [-o 特殊选项] 设备文件名 挂载点

选项：

-t 文件系统：加入文件系统类型来指定挂载的类型，可

以ext3、 ext4 、 iso9660等文件系统

-L 卷标名： 挂载指定卷标的分区，而不是安装设

备文件名挂载

-o 特殊选项：可以指定挂载的额外选项

参数 说明

atime/noatime 更新访问时间/不更新访问时间。 访问分区文件时， 是否更新文件

的访问时间， 默认为更新

async/sync 异步/同步， 默认为异步

auto/noauto 自动/手动， mount –a命令执行时， 是否会自动安装/etc/fstab文件

内容挂载， 默认为自动

defaults 定义默认值， 相当于rw,suid,dev,exec,auto,nouser,async这七个选项

exec/noexec 执行/不执行， 设定是否允许在文件系统中执行可执行文件， 默认

是exec允许

remount 重新挂载已经挂载的文件系统， 一般用于指定修改特殊权限

rw/ro 读写/只读， 文件系统挂载时， 是否具有读写权限， 默认是rw

suid/nosuid 具有/不具有SUID权限， 设定文件系统是否具有SUID和SGID的权

限， 默认是具有

user/nouser 允许/不允许普通用户挂载， 设定文件系统是否允许普通用户挂载，

默认是不允许， 只有root可以挂载分区

usrquota 写入代表文件系统支持用户磁盘配额， 默认不支持

grpquota 写入代表文件系统支持组磁盘配额， 默认不支持

[root@localhost ~]# mount -o remount,noexec /home

#重新挂载/boot分区，并使用noexec权限

[root@localhost sh]# cd /home

[root@localhost boot]# vi hello.sh

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9.2.3 挂载光盘与U盘

1、挂载光盘

[root@localhost ~]# mkdir /mnt/cdrom/

#建立挂载点

[root@localhost ~]# mount -t iso9660 /dev/cdrom /mnt/cdrom/

#挂载光盘

[root@localhost ~]# mount /dev/sr0 /mnt/cdrom/

2、卸载命令

[root@localhost ~]# umount 设备文件名或挂载点

[root@localhost ~]# umount /mnt/cdrom

3、挂载U盘

[root@localhost ~]# fdisk –l

#查看U盘设备文件名

[root@localhost ~]# mount -t vfat /dev/sdb1 /mnt/usb/

注意： Linux默认是不支持NTFS文件系统的

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9.2.4 支持NTFS文件系统

1、下载NTFS-3G插件

http://www.tuxera.com/community/ntfs-3gdownload/

2、安装NTFS-3G

[root@localhost ~]# tar -zxvf ntfs-3g\_ntfsprogs-2013.1.13.tgz

#解压

[root@localhost ~]# cd ntfs-3g\_ntfsprogs-2013.1.13

#进入解压目录

[root@localhost ntfs-3g\_ntfsprogs-2013.1.13]# ./configure

#编译器准备。没有指定安装目录，安装到默认位置中

[root@localhost ntfs-3g\_ntfsprogs-2013.1.13]# make

#编译

[root@localhost ntfs-3g\_ntfsprogs-2013.1.13]# make install

#编译安装

3、使用

[root@localhost ~]# mount -t ntfs-3g 分区设备文件名 挂载点

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9.3.1 fdisk命令分区过程

1、添加新硬盘

2、查看新硬盘

[root@localhost ~]# fdisk -l

3、使用fdisk命令分区

[root@localhost ~]# fdisk /dev/sdb

fdisk交互指令说明

命令 说明

a 设置可引导标记

b 编辑bsd磁盘标签

c 设置DOS操作系统兼容标记

d 删除一个分区

l 显示已知的文件系统类型。 82为Linux swap分区， 83为Linux分区

m 显示帮助菜单

n 新建分区

o 建立空白DOS分区表

p 显示分区列表

q 不保存退出

s 新建空白SUN磁盘标签

t 改变一个分区的系统ID

u 改变显示记录单位

v 验证分区表

w 保存退出

x 附加功能（ 仅专家）

4、重新读取分区表信息

[root@localhost ~]# partprobe

5、格式化分区

[root@localhost ~]# mkfs -t ext4 /dev/sdb1

6、建立挂载点并挂载

[root@localhost ~]# mkdir /disk1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9.3.2 分区自动挂载与fstab文件修复

1、 /etc/fstab文件

 第一字段：分区设备文件名或UUID（硬盘通用唯一识别

码）

 第二字段：挂载点

 第三字段：文件系统名称

 第四字段：挂载参数

 第五字段：指定分区是否被dump备份， 0代表不备份， 1

代表每天备份， 2代表不定期备份

 第六字段：指定分区是否被fsck检测， 0代表不检测，其

他数字代表检测的优先级，那么当然1的优先级比2高

2、分区自动挂载

[root@localhost ~]# vi /etc/fstab

…省略部分输出…

/dev/sdb5 /disk5 ext4 defaults 1 2

[root@localhost ~]# mount -a

# 依据配置文件/etc/fstab 的内容，自动挂载

3、 /etc/fstab文件修复

[root@localhost ~]# mount -o remount,rw /

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9.4 分配swap分区

1、新建swap分区

[root@localhost ~]# fdisk /dev/sdb

别忘记把分区ID改为82

2、格式化

[root@localhost ~]# mkswap /dev/sdb1

3、加入swap分区

[root@localhost ~]# swapon /dev/sdb1

#加入swap分区

[root@localhost ~]# swapoff /dev/sdb1

#取消swap分区

4、 swap分区开机自动挂载

[root@localhost ~]# vi /etc/fstab

/dev/sdb1 swap swap defaults 0 0

5、 free命令

[root@localhost ~]# free

#查看内存与swap分区使用状况

 cached（缓存） ： 是指把读取出来的数据保存在内存当

中，当再次读取时，不用读取硬盘而直接从内存当中读

取，加速了数据的读取过程

 buffer（缓冲） ： 是指在写入数据时，先把分散的写入操

作保存到内存当中，当达到一定程度再集中写入硬盘，

减少了磁盘碎片和硬盘的反复寻道，加速了数据的写入

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

第十讲 Shell基础

10.1 Shell概述

1、 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。

 Shell还是一个功能相当强大的编程语言，

易编写，易调试，灵活性较强。 Shell是解

释执行的脚本语言，在Shell中可以直接调

用Linux系统命令。

2、 Shell的分类

 Bourne Shell： 从1979起Unix就开始使用

Bourne Shell， Bourne Shell的主文件名为

sh。

 C Shell： C Shell主要在BSD版的Unix系

统中使用， 其语法和C语言相类似而得名

 Shell的两种主要语法类型有Bourne和C，

这两种语法彼此不兼容。 Bourne家族主要

包括sh、 ksh、 Bash、 psh、 zsh； C家族主

要包括： csh、 tcsh

 Bash： Bash与sh兼容， 现在使用的Linux

就是使用Bash作为用户的基本Shell。

3、 Linux支持的Shell

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.2 Shell脚本的执行方式

1、 echo输出命令

[root@localhost ~]# echo [选项] [输出内容]

选项：

-e： 支持反斜线控制的字符转换

控制字符 作 用

\\ 输出\本身

\a 输出警告音

\b 退格键， 也就是向左删除键

\c 取消输出行末的换行符。 和“ -n” 选项一致

\e ESCAPE键

\f 换页符

\n 换行符

\r 回车键

\t 制表符， 也就是Tab键

\v 垂直制表符

\0nnn 按照八进制ASCII码表输出字符。 其中0为数字零， nnn是三位八进

制数

\xhh 按照十六进制ASCII码表输出字符。 其中hh是两位十六进制数

[root@localhost ~]# echo -e "ab\bc"

#删除左侧字符

[root@localhost ~]# echo -e "a\tb\tc\nd\te\tf"

#制表符与换行符

[root@localhost ~]# echo -e \

"\x61\t\x62\t\x63\n\x64\t\x65\t\x66"

#按照十六进制ASCII码也同样可以输出

[root@localhost ~]# echo -e "\e[1;31m abcd \e[0m"

#输出颜色

#30m=黑色， 31m=红色， 32m=绿色， 33m=黄色

#34m=蓝色， 35m=洋红， 36m=青色， 37m=白色

2、第一个脚本

[root@localhost sh]# vi hello.sh

#!/bin/Bash

#The first program

# Author: shenchao （ E-mail: shenchao@lampbrother.net）

echo -e "Mr. Shen Chao is the most honest man in

LampBrother"

3、脚本执行

 赋予执行权限，直接运行

 chmod 755 hello.sh

 ./hello.sh

 通过Bash调用执行脚本

Bash hello.sh

[root@localhost sh]# cat -A hello.sh 显示隐藏字符

[root@localhost sh]# dos2unix Teris.sh windows文件转换为linux格式的文件

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.3.1 历史命令与命令补全

1、历史命令

[root@localhost ~]# history [选项] [历史命令保存文件]

选项：

-c： 清空历史命令

-w： 把缓存中的历史命令写入历史命令保存文件

~/.bash\_history

历史命令默认会保存1000条,可以在环境

变量配置文件/etc/profile中进行修改

历史命令的调用

使用上、下箭头调用以前的历史命令

使用“ !n”重复执行第n条历史命令

使用“ !!”重复执行上一条命令

使用“ !字串”重复执行最后一条以该字

串开头的命令

2、命令与文件补全

 在Bash中，命令与文件补全是非常方便与

常用的功能，我们只要在输入命令或文件

时，按“ Tab”键就会自动进行补全

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.3.2 命令别名与常用快捷键

1、命令别名

[root@localhost ~]# alias 别名='原命令'

#设定命令别名

[root@localhost ~]# alias

#查询命令别名

命令执行时顺序

1 第一顺位执行用绝对路径或相对路径执行

的命令。

2 第二顺位执行别名。

3 第三顺位执行Bash的内部命令。

4 第四顺位执行按照$PATH环境变量定义的

目录查找顺序找到的第一个命令。

让别名永久生效

[root@localhost ~]# vi /root/.bashrc

删除别名

[root@localhost ~]# unalias 别名

2、 Bash常用快捷键

快捷键 作 用

ctrl+A 把光标移动到命令行开头。 如果我们输入的命令过长， 想要把光标移

动到命令行开头时使用。

ctrl+E 把光标移动到命令行结尾。

ctrl+C 强制终止当前的命令。

ctrl+L 清屏， 相当于clear命令。

ctrl+U 删除或剪切光标之前的命令。 我输入了一行很长的命令， 不用使用退

格键一个一个字符的删除， 使用这个快捷键会更加方便

ctrl+K 删除或剪切光标之后的内容。

ctrl+Y 粘贴ctrl+U或ctrl+K剪切的内容。

ctrl+R 在历史命令中搜索， 按下ctrl+R之后， 就会出现搜索界面， 只要输入

搜索内容， 就会从历史命令中搜索。

ctrl+D 退出当前终端。

ctrl+Z 暂停， 并放入后台。 这个快捷键牵扯工作管理的内容， 我们在系统管

理章节详细介绍。

ctrl+S 暂停屏幕输出。

ctrl+Q 恢复屏幕输出。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.3.3 输入输出重定向

1、标准输入输出

设备 设备文件名 文件描述符 类型

键盘 /dev/stdin 0 标准输入

显示器 /dev/sdtout 1 标准输出

显示器 /dev/sdterr 2 标准错误输

出

2、输出重定向

类 型 符 号 作用

标准输出重定向

命令 > 文件

以覆盖的方式， 把命令的正确输出输出到指定的文件或设备当中。

命令 >> 文件

以追加的方式， 把命令的正确输出输出到指定的文件或设备当中。标准错误输出重定向

错误命令 2>文件

以覆盖的方式， 把命令的错误输出输出到指定的文件或设备当中。

错误命令 2>>文件以追加的方式， 把命令的错误输出输出到指定的文件或设备当中。正确输出和错误输出同时保存

命令 > 文件 2>&1

以覆盖的方式， 把正确输

出和错误输出都保存到同

一个文件当中。

命令 >> 文件 2>&1

以追加的方式， 把正确输

出和错误输出都保存到同

一个文件当中。

命令 &>文件

以覆盖的方式， 把正确输

出和错误输出都保存到同

一个文件当中。

命令 &>>文件

以追加的方式， 把正确输

出和错误输出都保存到同

一个文件当中。

命令>>文件1 2>>文件2

把正确的输出追加到文件

1中， 把错误的输出追加

到文件2中。

[root@localhost ~]# ls &>dev/null 直接丢弃结果，不保存

3、输入重定向

[root@localhost ~]# wc [选项] [文件名]

选项：

-c 统计字节数

-w 统计单词数

-l 统计行数

命令<文件 把文件作为命令的输入

命令<< 标识符

…

标识符 把标识符之间内容作为

命令的输入

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.3.4 多命令顺序执行与管道符

1、多命令顺序执行

多命令执行

符 格式 作 用

； 命令1 ；命令2 多个命令顺序执行， 命令之间没有任何逻辑联

系

&& 命令1 && 命令2

逻辑与

当命令1正确执行， 则命令2才会执行

当命令1执行不正确， 则命令2不会执行

|| 命令1 || 命令2

逻辑或

当命令1 执行不正确， 则命令2才会执行

当命令1正确执行， 则命令2不会执行

例子：

[root@localhost ~]# ls ; date ; cd /user ; pwd

文件拷贝或者磁盘拷贝

[root@localhost ~]# dd if=输入文件 of=输出文件 bs=字节数 count=个数

选项：

if=输入文件 指定源文件或源设备

of=输出文件 指定目标文件或目标设备

bs=字节数 指定一次输入/输出多少字节，即把这些字节看做

一个数据块

count=个数 指定输入/输出多少个数据块

例子：

/dev/zero 表示0

[root@localhost ~]# date ; dd if=/dev/zero of=/root/testfile bs=1k count=100000 ;

date 创建100M的测试文件，内容为全部为0

[root@localhost ~]# ls anaconda-ks.cfg && echo yes

[root@localhost ~]# ls /root/test || echo "no

[root@localhost ~]# 命令 && echo yes || echo no

2、管道符

命令格式：

[root@localhost ~]# 命令1 | 命令2

#命令1的正确输出作为命令2的操作对象

颜色显示

例子：

[root@localhost ~]# ll -a /etc/ | more

[root@localhost ~]# netstat -an | grep "ESTABLISHED"

[root@localhost ~]# grep [选项] "搜索内容"

文件名

选项：

-i： 忽略大小写

-n： 输出行号

-v： 反向查找

--color=auto 搜索出的关键字用颜色显示

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.3.5 通配符与其他特殊符号

1、通配符

通配符 作 用

? 匹配一个任意字符

\* 匹配0个或任意多个任意字符， 也就是可以匹配任何内容

[]

匹配中括号中任意一个字符。 例如： [abc]代表一定匹配

一个字符， 或者是a， 或者是b， 或者是c。

[-]

匹配中括号中任意一个字符， -代表一个范围。 例如： [a-z]

代表匹配一个小写字母。

[^]

逻辑非， 表示匹配不是中括号内的一个字符。 例如： [^0-

9]代表匹配一个不是数字的字符。

[root@localhost ~]# cd /tmp/

[root@localhost tmp]# rm -rf \*

[root@localhost tmp]# touch abc

[root@localhost tmp]# touch abcd

[root@localhost tmp]# touch 012

[root@localhost tmp]# touch 0abc

[root@localhost tmp]# ls ?abc

[root@localhost tmp]# ls [0-9]\*

[root@localhost tmp]# ls [^0-9]\*

2、 Bash中其他特殊符号

符 号 作 用

'' 单引号。 在单引号中所有的特殊符号， 如“ $” 和“ `” (反引号)都

没有特殊含义。

""

双引号。 在双引号中特殊符号都没有特殊含义， 但是“ $” 、 “ `”

和“ \” 是例外， 拥有“ 调用变量的值” 、 “ 引用命令” 和“ 转义

符” 的特殊含义。

`` 反引号。 反引号括起来的内容是系统命令， 在Bash中会先执行它。

和$()作用一样， 不过推荐使用$()， 因为反引号非常容易看错。

$() 和反引号作用一样， 用来引用系统命令。

# 在Shell脚本中， #开头的行代表注释。

$ 用于调用变量的值， 如需要调用变量name的值时， 需要用$name

的方式得到变量的值。

\

转义符， 跟在\之后的特殊符号将失去特殊含义， 变为普通字符。

如\$将输出“ $” 符号， 而不当做是变量引用。

反引号与$()

[root@localhost ~]# echo `ls`

[root@localhost ~]# echo $(date)

单引号与双引号

[root@localhost ~]# name=sc

[root@localhost ~]# echo '$name'

[root@localhost ~]# echo "$name"

[root@localhost ~]# echo ‘$(date)'

[root@localhost ~]# echo “$(date)"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.4.1 用户自定义变量

1、什么是变量

 变量是计算机内存的单元，其中存放的值

可以改变。当Shell脚本需要保存一些信息

时，如一个文件名或是一个数字，就把它

存放在一个变量中。每个变量有一个名字

，所以很容易引用它。 使用变量可以保存

有用信息，使系统获知用户相关设置，变

量也可以用于保存暂时信息。

2、变量设置规则

 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。如果变量名是

“ 2name”则是错误的。

 在Bash中，变量的默认类型都是字符串型

，如果要进行数值运算，则必修指定变量

类型为数值型。

变量用等号连接值，等号左右两侧不能有

空格。

变量的值如果有空格，需要使用单引号或

双引号包括。

在变量的值中，可以使用“ \”转义符。

如果需要增加变量的值，那么可以进行变

量值的叠加。不过变量需要用双引号包含

“$变量名”或用${变量名}包含。

如果是把命令的结果作为变量值赋予变量

，则需要使用反引号或$()包含命令。

环境变量名建议大写，便于区分。

3、变量分类

 用户自定义变量

 环境变量： 这种变量中主要保存的是和系统操

作环境相关的数据。

 位置参数变量：这种变量主要是用来向脚本当

中传递参数或数据的，变量名不能自定义，变

量作用是固定的。

 预定义变量：是Bash中已经定义好的变量，变

量名不能自定义，变量作用也是固定的。

4、本地变量

 变量定义

[root@localhost ~]# name="shen chao"

 变量叠加

[root@localhost ~]# aa=123

[root@localhost ~]# aa="$aa"456

[root@localhost ~]# aa=${aa}789

变量调用

[root@localhost ~]# echo $name

变量查看

[root@localhost ~]# set

变量删除

[root@localhost ~]# unset name

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.4.2 环境变量

1、环境变量是什么

 用户自定义变量只在当前的Shell中生效，

而环境变量会在当前Shell和这个Shell的所

有子Shell当中生效。如果把环境变量写入

相应的配置文件，那么这个环境变量就会

在所有的Shell中生效

[root@localhost ~]# pstree 确定进程树

2、设置环境变量

export 变量名=变量值

#申明变量

env #查询环境变量

set #查询所有变量

unset 变量名

#删除变量

3、系统常见环境变量

 PATH：系统查找命令的路径

[root@localhost ~]# echo $PATH

/usr/lib/qt-3.3/bin:/usr/local/sbin:/usr/local/bin:

/sbin:/bin:/usr/sbin:/usr/bin:/root/bin

 PATH="$PATH":/root/sh

#PATH变量叠加

 PS1：定义系统提示符的变量

\d：显示日期，格式为“星期 月 日”

\h：显示简写主机名。如默认主机名“ localhost”

\t：显示24小时制时间，格式为“ HH:MM:SS”

\T：显示12小时制时间，格式为“ HH:MM:SS”

\A：显示24小时制时间，格式为“ HH:MM”

\u：显示当前用户名

\w：显示当前所在目录的完整名称

\W：显示当前所在目录的最后一个目录

\#：执行的第几个命令

\$：提示符。如果是root用户会显示提示符为“ #”，如果是普通用户

会显示提示符为“ $”

举例：

 [root@localhost ~]# PS1='[\u@\t \w]\$ '

 [root@04:50:08 /usr/local/src]#PS1='[\u@\@ \h \# \W]\$‘

[root@04:53 上午 localhost 31 src]#PS1='[\u@\h \W]\$ '

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.4.3 位置参数变量

1、位置参数变量

位置参数变量 作 用

$n n为数字， $0代表命令本身， $1-$9代表第一

到第九个参数， 十以上的参数需要用大括号

包含， 如${10}.

$\* 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

$@ 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

$# 这个变量代表命令行中所有参数的个数

例子1：

#!/bin/bash

num1=$1

num2=$2

sum=$(( $num1 + $num2))

#变量sum的和是num1加num2

echo $sum

#打印变量sum的值

例子2：

#!/bin/bash

echo "A total of $# parameters"

#使用$#代表所有参数的个数

echo "The parameters is: $\*"

#使用$\*代表所有的参数

echo "The parameters is: $@"

#使用$@也代表所有参数

例子3： $\*与$@的区别

#!/bin/bash

for i in "$\*"

#$\*中的所有参数看成是一个整体，所以这个for循环只会循环一次

do

echo "The parameters is: $i"

done

x=1

for y in "$@"

#$@中的每个参数都看成是独立的，所以“ $@”中有几个参数，就会循环几次

do

echo "The parameter$x is: $y"

x=$(( $x +1 ))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.4.4 预定义变量

1、预定义变量

预定义变量 作 用

$？ 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。

$$ 当前进程的进程号（ PID）

$! 后台运行的最后一个进程的进程号（ PID）

#!/bin/bash

# Author: shenchao （ E-mail: shenchao@lampbrother.net）

echo "The current process is $$"

#输出当前进程的PID。

#这个PID就是variable.sh这个脚本执行时，生成的进程的PID

find /root -name hello.sh &

#使用find命令在root目录下查找hello.sh文件

#符号&的意思是把命令放入后台执行，工作管理我们在系统管理章节

会详细介绍

echo "The last one Daemon process is $!"

2、接收键盘输入

[root@localhost ~]# read [选项] [变量名]

选项：

-p “提示信息”：在等待read输入时，输出提示信息

-t 秒数： read命令会一直等待用户输入，使用

此选项可以指定等待时间

-n 字符数： read命令只接受指定的字符数，就会

执行

-s： 隐藏输入的数据，适用于机密信息的

输入

#!/bin/bash

# Author: shenchao （ E-mail: shenchao@lampbrother.net）

read -t 30 -p "Please input your name: " name

#提示“请输入姓名”并等待30秒，把用户的输入保存入变量name中

echo "Name is $name "

read -s -t 30 -p "Please enter your age: " age

#年龄是隐私，所以我们用“ -s”选项隐藏输入

echo -e "\n"

echo "Age is $age "

read -n 1 -t 30 -p "Please select your gender[M/F]: " gender

#使用“ -n 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.5.1 数值运算与运算符

1、 declare声明变量类型

[root@localhost ~]# declare [+/-][选项] 变量名

选项：

-： 给变量设定类型属性

+： 取消变量的类型属性

-i： 将变量声明为整数型（ integer）

-x： 将变量声明为环境变量

-p： 显示指定变量的被声明的类型

2、数值运算—方法1

[root@localhost ~]# aa=11

[root@localhost ~]# bb=22

#给变量aa和bb赋值

[root@localhost ~]# declare -i cc=$aa+$bb

方法2： expr或let数值运算工具

[root@localhost ~]# aa=11

[root@localhost ~]# bb=22

#给变量aa和变量bb赋值

[root@localhost ~]# dd=$(expr $aa + $bb)

#dd的值是aa和bb的和。注意“ +”号左右两

侧必须有空格

方法3： “ $((运算式))”或“ $[运算式]”

[root@localhost ~]# aa=11

[root@localhost ~]# bb=22

[root@localhost ~]# ff=$(( $aa+$bb ))

[root@localhost ~]# gg=$[ $aa+$bb ]

3、运算符

优先级 运算符 说明

13 -, + 单目负、 单目正

12 !, ~ 逻辑非、 按位取反或补码

11 \* , / , % 乘、 除、 取模

10 +, - 加、 减

9 << , >> 按位左移、 按位右移

8 < =, > =, < , > 小于或等于、 大于或等于、 小于、 大于

7 == , != 等于、 不等于

6 & 按位与

5 ^ 按位异或

4 | 按位或

3 && 逻辑与

2 || 逻辑或

1

=,+=,-

=,\*=,/=,%=,&=, ^=,

|=, <<=, >>=

赋值、 运算且赋值

[root@localhost ~]# aa=$(( (11+3)\*3/2 ))

#虽然乘和除的优先级高于加，但是通过小括号可以调整运算优先级

[root@localhost ~]# bb=$(( 14%3 ))

#14不能被3整除，余数是2

[root@localhost ~]# cc=$(( 1 && 0 ))

#逻辑与运算只有想与的两边都是1，与的结果才是1，

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.5.2 变量测试与内容替换



例子1：测试x=${y-新值}

[root@localhost ~]# unset y

#删除变量y

[root@localhost ~]# x=${y-new}

#进行测试

[root@localhost ~]# echo $x

new

#因为变量y不存在，所以x=new

[root@localhost ~]# y=""

#给变量y赋值为空

[root@localhost ~]# x=${y-new}

#进行测试

[root@localhost ~]# echo $x

[root@localhost ~]# y=old

#给变量y赋值

[root@localhost ~]# x=${y-new}

#进行测试

[root@localhost ~]# echo $x

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.6.1 环境变量配置文件简介

10.6.2 环境变量配置文件作用

10.6.3 其他配置文件和登录信息

1、 source命令

[root@localhost ~]# source 配置文件

或

[root@localhost ~]# . 配置文件

2、环境变量配置文件简介

 环境变量配置文件中主要是定义对系统的

操作环境生效的系统默认环境变量，比如

PATH、 HISTSIZE、 PS1、 HOSTNAME等

默认环境变量。

 /etc/profile

 /etc/profile.d/\*.sh

 ~/.bash\_profile

 ~/.bashrc

 /etc/bashrc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.6.2 环境变量配置文件作用

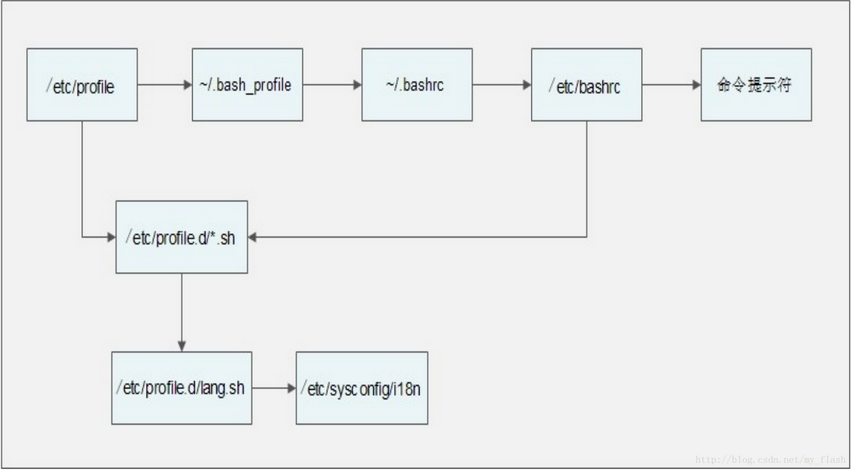
 /etc/profile

 /etc/profile.d/\*.sh

 ~/.bash\_profile

 ~/.bashrc

 /etc/bashrc



/etc/profile的作用：

 USER变量：

 LOGNAME变量：

 MAIL变量：

 PATH变量：

 HOSTNAME变量：

 HISTSIZE变量：

 umask：

 调用/etc/profile.d/\*.sh文件

~/.bash\_profile的作用

调用了~/.bashrc文件。

在PATH变量后面加入了“ :$HOME/bin”

这个目录

~/.bashrc的作用

定义默认别名

调用/etc/bashrc

/etc/bashrc的作用

 PS1变量

 umask

 PATH变量

调用/etc/profile.d/\*.sh文件

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.6.3 其他配置文件和登录信息

1、 注销时生效的环境变量配置文件

 ~/.bash\_logout

2、 其他配置文件

 ~/bash\_history

3、 Shell登录信息

 本地终端欢迎信息： /etc/issue

转义符 作 用

\d 显示当前系统日期

\s 显示操作系统名称

\l 显示登录的终端号， 这个比较常用。

\m 显示硬件体系结构， 如i386、 i686等

\n 显示主机名

\o 显示域名

\r 显示内核版本

\t 显示当前系统时间

\u 显示当前登录用户的序列号

远程终端欢迎信息： /etc/issue.net

 转义符在/etc/issue.net文件中不能使用

 是否显示此欢迎信息，由ssh的配置文件

/etc/ssh/sshd\_config决定，加入“ Banner

/etc/issue.net”行才能显示（记得重启SSH服

务）

登陆后欢迎信息： /etc/motd

不管是本地登录，还是远程登录，都可以显

示此欢迎信息

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

第十一讲 Shell编程

11.1 基础正则表达式

1、正则表达式与通配符

 正则表达式用来在文件中匹配符合条件的

字符串， 正则是包含匹配。 grep、 awk、

sed等命令可以支持正则表达式。

 通配符用来匹配符合条件的文件名，通配

符是完全匹配。 ls、 find、 cp这些命令不

支持正则表达式，所以只能使用shell自己

的通配符来进行匹配了。

2、基础正则表达式

元字符 作 用



“ \*”前一个字符匹配0次，或任意多次

 grep "a\*" test\_rule.txt

#匹配所有内容，包括空白行

 grep "aa\*" test\_rule.txt

#匹配至少包含有一个a的行

 grep "aaa\*" test\_rule.txt

匹配最少包含两个连续a的字符串

 grep "aaaaa\*" test\_rule.txt

#则会匹配最少包含四个个连续a的字符串

“ .” 匹配除了换行符外任意一个字符

 grep "s..d" test\_rule.txt

#“ s..d”会匹配在s和d这两个字母之间一定有两个字符的单词

 grep "s.\*d" test\_rule.txt

#匹配在s和d字母之间有任意字符

 grep ".\*" test\_rule.txt

#匹配所有内容

“ ^”匹配行首，“ $”匹配行尾

 grep "^M" test\_rule.txt

#匹配以大写“ M”开头的行

 grep "n$" test\_rule.txt

#匹配以小写“ n”结尾的行

 grep -n "^$" test\_rule.txt

#会匹配空白行

“ []” 匹配中括号中指定的任意一个

字符，只匹配一个字符

 grep "s[ao]id" test\_rule.txt

#匹配s和i字母中，要不是a、要不是o

 grep "[0-9]" test\_rule.txt

#匹配任意一个数字

 grep "^[a-z]" test\_rule.txt

#匹配用小写字母开头的行

“ [^]” 匹配除中括号的字符以外的任意一个字符

 grep "^[^a-z]" test\_rule.txt

#匹配不用小写字母开头的行

 grep "^[^a-zA-Z]" test\_rule.txt

#匹配不用字母开头的行

“ \” 转义符

 grep "\.$" test\_rule.txt

#匹配使用“ .”结尾的行

“ \{n\}”表示其前面的字符恰好出现n次

 grep "a\{3\}" test\_rule.txt

#匹配a字母连续出现三次的字符串

 grep "[0-9]\{3\}" test\_rule.txt

#匹配包含连续的三个数字的字符串

“ \{n,\}”表示其前面的字符出现不小于n次

 grep "^[0-9]\{3,\}[a-z]" test\_rule.txt

#匹配最少用连续三个数字开头的行

“ \{n,m\}”匹配其前面的字符至少出现n次，最多出现m

 grep "sa\{1,3\}i" test\_rule.txt

#匹配在字母s和字母i之间有最少一个a，最多三个a

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11.2 字符截取命令

[root@localhost ~]# cut [选项] 文件名

选项：

-f 列号： 提取第几列

-d 分隔符： 按照指定分隔符分割列

[root@localhost ~]# vi student.txt

ID Name gender Mark

1 Liming M 86

2 Sc M 90

3 Gao M 83

[root@localhost ~]# cut -f 2 student.txt

[root@localhost ~]# cut -f 2,3 student.txt

[root@localhost ~]# cut -d ":" -f 1,3

/etc/passwd

cut命令的局限

[root@localhost ~]# df -h | cut -d " " -f 1,3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11.2.2 printf命令

printf ’输出类型输出格式’ 输出内容

输出类型：

%ns： 输出字符串。 n是数字指代输出几个字符

%ni： 输出整数。 n是数字指代输出几个数字

%m.nf： 输出浮点数。 m和n是数字， 指代输出的整数

位数和小数位数。 如%8.2f代表共输出8位数，

其中2位是小数， 6位是整数。

输出格式：

\a: 输出警告声音

\b: 输出退格键， 也就是Backspace键

\f: 清除屏幕

\n: 换行

\r: 回车， 也就是Enter键

\t: 水平输出退格键， 也就是Tab键

\v: 垂直输出退格键， 也就是Tab键

[root@localhost ~]# printf %s 1 2 3 4 5 6

[root@localhost ~]# printf %s %s %s 1 2 3 4 5 6

[root@localhost ~]# printf '%s %s %s' 1 2 3 4 5 6

[root@localhost ~]# printf '%s %s %s\n' 1 2 3 4 5 6

[root@localhost ~]# vi student.txt

ID Name PHP Linux MySQL Average

1 Liming 82 95 86 87.66

2 Sc 74 96 87 85.66

3 Gao 99 83 93 91.66

printf '%s' $(cat student.txt)

#不调整输出格式

printf '%s\t %s\t %s\t %s\t %s\t %s\t \n' $(cat student.txt)

#调整格式输出

在awk命令的输出中支持print和printf命令

 print： print会在每个输出之后自动加入一

个换行符（ Linux默认没有print命令）

 printf： printf是标准格式输出命令，并不

会自动加入换行符，如果需要换行，需要

手工加入换行符

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11.2.3 awk命令

# awk ‘条件1{动作1} 条件2{动作2}…’ 文件名

条件（ Pattern）：

一般使用关系表达式作为条件

x > 10 判断变量 x是否大于10

x>=10 大于等于

x<=10 小于等于

动作（ Action）：

格式化输出

流程控制语句

[root@localhost ~]# vi student.txt

ID Name PHP Linux MySQL Average

1 Liming 82 95 86 87.66

2 Sc 74 96 87 85.66

3 Gao 99 83 93 91.66

# awk '{printf $2 "\t" $6 "\n"}' student.txt

# df -h | awk '{print $1 "\t" $3}'

BEGIN

# awk 'BEGIN{printf "This is a transcript \n" }

{printf $2 "\t" $6 "\n"}' student.txt

END

# awk 'END{printf "The End \n" }

{printf $2 "\t" $6 "\n"}' student.txt

FS内置变量

# cat /etc/passwd | grep "/bin/bash" | \

awk 'BEGIN {FS=":"} {printf $1 "\t" $3 "\n"}'

关系运算符

# cat student.txt | grep -v Name | \

awk '$6 >= 87 {printf $2 "\n" }'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11.2.4 sed命令

sed命令

 sed 是一种几乎包括在所有 UNIX 平台（

包括 Linux）的轻量级流编辑器。 sed主要

是用来将数据进行选取、替换、删除、新

增的命令。

[root@localhost ~]# sed [选项] ‘[动作]’ 文件名

选项：

-n： 一般sed命令会把所有数据都输出到屏幕 ，

如果加入此选择，则只会把经过sed命令处

理的行输出到屏幕。

-e： 允许对输入数据应用多条sed命令编辑

-i： 用sed的修改结果直接修改读取数据的文件，

而不是由屏幕输出

动作：

a \： 追加，在当前行后添加一行或多行。添加多行时，除最后 一行

外，每行末尾需要用“ \”代表数据未完结。

c \： 行替换，用c后面的字符串替换原数据行，替换多行时，除最

后一行外，每行末尾需用“ \”代表数据未完结。

i \： 插入，在当期行前插入一行或多行。插入多行时，除最后 一行

外，每行末尾需要用“ \”代表数据未完结。

d： 删除，删除指定的行。

p： 打印，输出指定的行。

s： 字串替换，用一个字符串替换另外一个字符串。格式为“行范

围s/旧字串/新字串/g”（和vim中的替换格式类似） 。

学生成绩表

[root@localhost ~]# vi student.txt

ID Name PHP Linux MySQL Average

1 Liming 82 95 86 87.66

2 Sc 74 96 87 85.66

3 Gao 99 83 93 91.66

行数据操作

[root@localhost ~]# sed '2p' student.txt

#查看文件的第二行

[root@localhost ~]# sed -n '2p' student.txt

[root@localhost ~]# sed '2,4d' student.txt

#删除第二行到第四行的数据，但不修改文

件本身

[root@localhost ~]# sed '2a hello' student.txt

#在第二行后追加hello

[root@localhost ~]# sed '2i hello \

world' student.txt

#在第二行前插入两行数据

# sed '2c No such person‘ student.txt

#数据替换

字符串替换

# sed ‘s/旧字串/新字串/g’ 文件名

# sed '3s/74/99/g' student.txt

#在第三行中，把74换成99

#sed -i '3s/74/99/g' student.txt

#sed操作的数据直接写入文件

# sed -e 's/Liming//g ; s/Gao//g' student.txt

#同时把“ Liming”和“ Gao”替换为空

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11.3 字符处理命令

1、排序命令sort

[root@localhost ~]# sort [选项] 文件名

选项：

-f： 忽略大小写

-n： 以数值型进行排序，默认使用字符串型排序

-r： 反向排序

-t： 指定分隔符，默认是分隔符是制表符

-k n[,m]： 按照指定的字段范围排序。从第n字段开始，

m字段结束（默认到行尾）

[root@localhost ~]# sort /etc/passwd

#排序用户信息文件

[root@localhost ~]# sort -r /etc/passwd

#反向排序

[root@localhost ~]# sort -t ":" -k 3,3 /etc/passwd

#指定分隔符是“：”，用第三字段开头，第三字段结尾排

序，就是只用第三字段排序

[root@localhost ~]# sort -n -t ":" -k 3,3 /etc/passwd

2、统计命令wc

[root@localhost ~]# wc [选项] 文件名

选项：

-l： 只统计行数

-w： 只统计单词数

-m： 只统计字符数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11.4 条件判断

测试选项 作 用

-b 文件 判断该文件是否存在， 并且是否为块设备文件（ 是块设备文件

为真）

-c文件 判断该文件是否存在， 并且是否为字符设备文件（ 是字符设备

文件为真）

-d 文件 判断该文件是否存在， 并且是否为目录文件（ 是目录为真）

-e 文件 判断该文件是否存在（ 存在为真）

-f 文件 判断该文件是否存在， 并且是否为普通文件（ 是普通文件为真）

-L 文件 判断该文件是否存在， 并且是否为符号链接文件（ 是符号链接

文件为真）

-p 文件 判断该文件是否存在， 并且是否为管道文件（ 是管道文件为真）

-s 文件 判断该文件是否存在， 并且是否为非空（ 非空为真）

-S 文件 判断该文件是否存在， 并且是否为套接字文件（ 是套接字文件

为真）

1、 按照文件类型进行判断

两种判断格式

[root@localhost ~]# test -e /root/install.log

[root@localhost ~]# [ -e /root/install.log ]

[ -d /root ] && echo "yes" || echo "no"

#第一个判断命令如果正确执行，则打印“ yes”，否则打

印“ no”

2、 按照文件权限进行判断

测试选项 作 用

-r 文件 判断该文件是否存在， 并且是否该文件拥有读权限（ 有读

权限为真）

-w文件 判断该文件是否存在， 并且是否该文件拥有写权限（ 有写

权限为真）

-x 文件 判断该文件是否存在， 并且是否该文件拥有执行权限（ 有

执行权限为真）

-u 文件 判断该文件是否存在， 并且是否该文件拥有SUID权限（ 有

SUID权限为真）

-g 文件 判断该文件是否存在， 并且是否该文件拥有SGID权限（ 有

SGID权限为真）

-k 文件 判断该文件是否存在， 并且是否该文件拥有SBit权限（ 有

SBit权限为真）

[ -w student.txt ] && echo "yes" || echo "no"

#判断文件是拥有写权限的

3、 两个文件之间进行比较

测试选项

作 用

文件1 -nt 文件2

判断文件1的修改时间是否比文件2的新（ 如果新则为真）

文件1 -ot 文件2

判断文件1的修改时间是否比文件2的旧（ 如果旧则为真）

文件1 -ef 文件2

判断文件1是否和文件2的Inode号一致， 可以理解为两个

文件是否为同一个文件。 这个判断用于判断硬链接是很

好的方法

ln /root/student.txt /tmp/stu.txt

#创建个硬链接吧

[ /root/student.txt -ef /tmp/stu.txt ] && echo "yes" || echo "no"

yes

#用test测试下，果然很有用

4、 两个整数之间比较

测试选项 作 用

整数1 -eq 整数2 判断整数1是否和整数2相等（ 相等为真）

整数1 -ne 整数2 判断整数1是否和整数2不相等（ 不相等位置）

整数1 -gt 整数2 判断整数1是否大于整数2（ 大于为真）

整数1 -lt 整数2 判断整数1是否小于整数2（ 小于位置）

整数1 -ge 整数2 判断整数1是否大于等于整数2（ 大于等于为真）

整数1 -le 整数2 判断整数1是否小于等于整数2（ 小于等于为真）

[ 23 -ge 22 ] && echo "yes" || echo "no"

yes

#判断23是否大于等于22，当然是了

[ 23 -le 22 ] && echo "yes" || echo "no"

no

#判断23是否小于等于22，当然不是了

5、 字符串的判断

测试选项

作 用

-z 字符串

判断字符串是否为空（ 为空返回真）

-n 字符串

判断字符串是否为非空（ 非空返回真）

字串1 ==字串2

判断字符串1是否和字符串2相等（ 相等返回真）

字串1 != 字串2

判断字符串1是否和字符串2不相等（ 不相等返回真）

name=sc

#给name变量赋值

[ -z "$name" ] && echo "yes" || echo "no"

no

#判断name变量是否为空，因为不为空，所

以返回no

aa=11

bb=22

#给变量aa和变量bb赋值

[ "$aa" == "bb" ] && echo "yes" || echo "no"

no

#判断两个变量的值是否相等，明显不相等

，所以返回no

6、 多重条件判断

测试选项

作 用

判断1 -a 判断2

逻辑与， 判断1和判断2都成立， 最终的结果才为真

判断1 -o 判断2

逻辑或， 判断1和判断2有一个成立， 最终的结果就为

真

！判断

逻辑非， 使原始的判断式取反

aa=11

[ -n "$aa" -a "$aa" -gt 23 ] && echo "yes" || echo "no"

no

#判断变量aa是否有值，同时判断变量aa的是否大于23

#因为变量aa的值不大于23，所以虽然第一个判断值为真，

返回的结果也是假

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11.5.1 if语句

1、 单分支if条件语句

if [ 条件判断式 ];then

程序

fi

或者

if [ 条件判断式 ]

then

程序

fi

单分支条件语句需要注意几个点

 if语句使用fi结尾，和一般语言使用大括

号结尾不同

 [ 条件判断式 ]就是使用test命令判断，所

以中括号和条件判断式之间必须有空格

 then后面跟符合条件之后执行的程序，可

以放在[]之后，用“；”分割。也可以换

行写入，就不需要“；”了

例子：判断分区使用率

#!/bin/bash

#统计根分区使用率

# Author: shenchao （ E-mail: shenchao@lampbrother.net）

rate=$(df -h | grep "/dev/sda3" | awk '{print $5}' | cut -d "%" -

f1)

#把根分区使用率作为变量值赋予变量rate

if [ $rate -ge 80 ]

then

echo "Warning! /dev/sda3 is full!!"

fi

2、 双分支if条件语句

if [ 条件判断式 ]

then

条件成立时， 执行的程序

else

条件不成立时， 执行的另一个程序

fi

例子1：备份mysql数据库



#!/bin/bash

#备份mysql数据库。

# Author: shenchao （ E-mail: shenchao@lampbrother.net）

ntpdate asia.pool.ntp.org &>/dev/null

#同步系统时间

date=$(date +%y%m%d)

#把当前系统时间按照“年月日” 格式赋予变量date

size=$(du -sh /var/lib/mysql)

#统计mysql数据库的大小， 并把大小赋予size变量

if [ -d /tmp/dbbak ]

then

echo "Date : $date!" > /tmp/dbbak/dbinfo.txt

echo "Data size : $size" >> /tmp/dbbak/dbinfo.txt

cd /tmp/dbbak

tar -zcf mysql-lib-$date.tar.gz /var/lib/mysql dbinfo.txt

&>/dev/null

rm -rf /tmp/dbbak/dbinfo.txt

else

mkdir /tmp/dbbak

echo "Date : $date!" > /tmp/dbbak/dbinfo.txt

echo "Data size : $size" >> /tmp/dbbak/dbinfo.txt

cd /tmp/dbbak

tar -zcf mysql-lib-$date.tar.gz /var/lib/mysql dbinfo.txt

&>/dev/null

rm -rf /tmp/dbbak/dbinfo.txt

fi

例子2：判断apache是否启动



#!/bin/bash

# Author: shenchao （ E-mail: shenchao@lampbrother.net）

port=$(nmap -sT 192.168.1.156 | grep tcp | grep http | awk '{print

$2}')

#使用nmap命令扫描服务器， 并截取apache服务的状态， 赋予变量port

if [ "$port" == "open" ]

then

echo “ $(date) httpd is ok!” >> /tmp/autostart-acc.log

else

/etc/rc.d/init.d/httpd start &>/dev/null

echo "$(date) restart httpd !!" >> /tmp/autostart-err.log

fi

ps aux | grep httpd 查看服务是否存在，不论是否启动

netstat -tlun 查看端口是否监听

3、 多分支if条件语句

if [ 条件判断式1 ]

then

当条件判断式1成立时， 执行程序1

elif [ 条件判断式2 ]

then

当条件判断式2成立时， 执行程序2

„省略更多条件…

else

当所有条件都不成立时， 最后执行此程序

fi



#!/bin/bash

#判断用户输入的是什么文件

# Author: shenchao （ E-mail: shenchao@lampbrother.net）

read -p "Please input a filename: " file

#接收键盘的输入， 并赋予变量file

if [ -z "$file" ]

#判断file变量是否为空

then

echo "Error,please input a filename"

exit 1

elif [ ! -e "$file" ]

#判断file的值是否存在

then

echo "Your input is not a file!"

exit 2

elif [ -f "$file" ]

#判断file的值是否为普通文件

then

echo "$file is a regulare file!"

elif [ -d "$file" ]

#判断file的值是否为目录文件

then

echo "$file is a directory!"

else

echo "$file is an other file!"

fi

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11.5.2 case语句

多分支case条件语句

 case语句和if…elif…else语句一样都是多

分支条件语句，不过和if多分支条件语句

不同的是， case语句只能判断一种条件关

系，而if语句可以判断多种条件关系。

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



#!/bin/bash

#判断用户输入

# Author: shenchao （ E-mail: shenchao@lampbrother.net）

read -p "Please choose yes/no: " -t 30 cho

case $cho in

"yes")

echo "Your choose is yes!"

;;

"no")

echo "Your choose is no!"

;;

\*)

echo "Your choose is error!"

;;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11.5.3 for循环

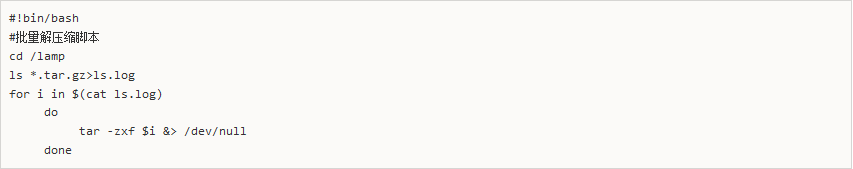
语法一：

for 变量 in 值1 值2 值3

do

程序

done



#!bin/bash

#批量解压缩脚本

cd /lamp

ls \*.tar.gz>ls.log

for i in $(cat ls.log)

do

tar -zxf $i &> /dev/null

done

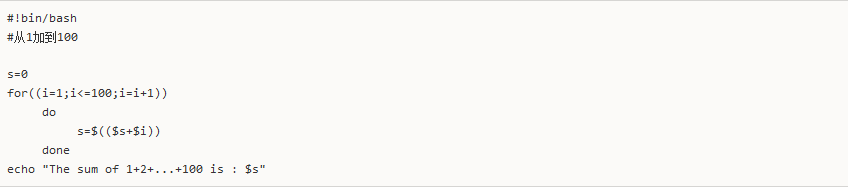
语法二：

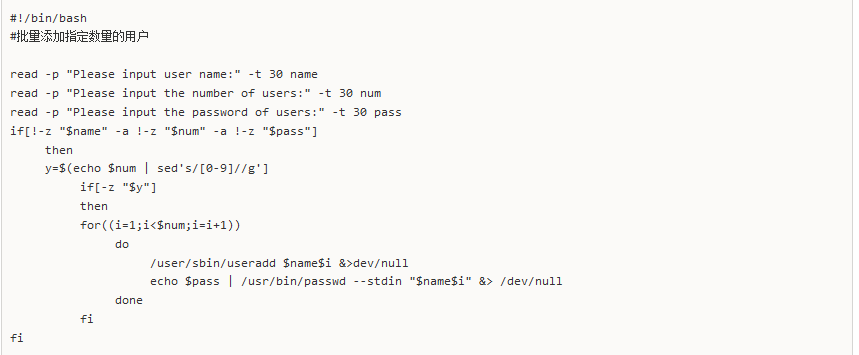
for((初始值;循环控制条件;变量变化))

do

程序

done





#!bin/bash

#从1加到100

s=0

for((i=1;i<=100;i=i+1))

do

s=$(($s+$i))

done

echo "The sum of 1+2+...+100 is : $s"

#!/bin/bash

#批量添加指定数量的用户

read -p "Please input user name:" -t 30 name

read -p "Please input the number of users:" -t 30 num

read -p "Please input the password of users:" -t 30 pass

if[!-z "$name" -a !-z "$num" -a !-z "$pass"]

then

y=$(echo $num | sed's/[0-9]//g']

if[-z "$y"]

then

for((i=1;i<$num;i=i+1))

do

/user/sbin/useradd $name$i &>dev/null

echo $pass | /usr/bin/passwd --stdin "$name$i" &> /dev/null

done

fi

fi

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11.5.4 while循环与until循环

1、 while循环

 while循环是不定循环，也称作条件循环

。只要条件判断式成立，循环就会一直继

续，直到条件判断式不成立，循环才会停

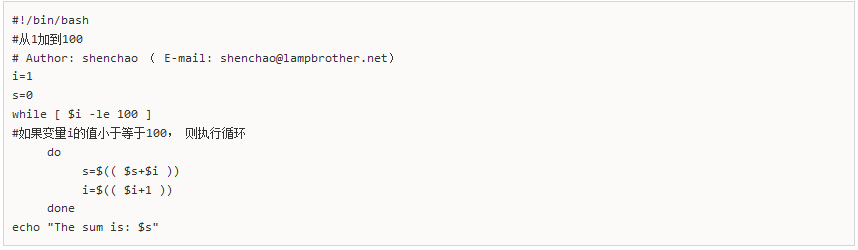
止。 这就和for的固定循环不太一样了。

while [ 条件判断式 ]

do

程序

done



#!/bin/bash

#从1加到100

# Author: shenchao （ E-mail: shenchao@lampbrother.net）

i=1

s=0

while [ $i -le 100 ]

#如果变量i的值小于等于100， 则执行循环

do

s=$(( $s+$i ))

i=$(( $i+1 ))

done

echo "The sum is: $s"

2、 until循环

 until循环，和while循环相反， until循环时

只要条件判断式不成立则进行循环，并执

行循环程序。一旦循环条件成立，则终止

循环。

until [ 条件判断式 ]

do

程序

done



#!/bin/bash

#从1加到100

# Author: shenchao （ E-mail: shenchao@lampbrother.net）

i=1

s=0

until [ $i -gt 100 ]

#循环直到变量i的值大于100， 就停止循环

do

s=$(( $s+$i ))

i=$(( $i+1 ))

done

s=$(( $s+$i ))

i=$(( $i+1 ))

done

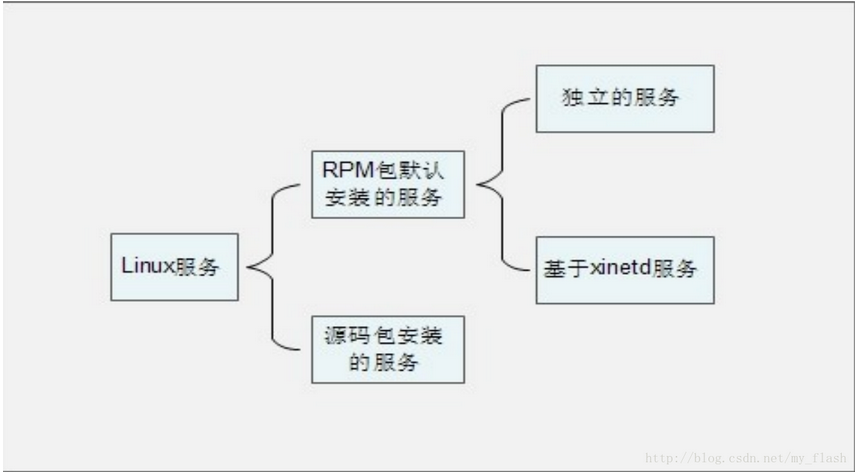
echo "The sum is: $s"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

第十二讲 Linux服务管理

12.1 服务简介与分类

1、服务的分类



启动与自启动

服务启动： 就是在当前系统中让服务运行

，并提供功能。

服务自启动： 自启动是指让服务在系统开

机或重启动之后，随着系统的启动而自动

启动服务。

查询已安装的服务

 RPM包安装的服务

 chkconfig --list

#查看服务自启动状态，可以看到所有RPM包安装的服务

ps aux 查看所有正在运行的服务

netstat -tlun 查看网络端口占用情况

 源码包安装的服务

 查看服务安装位置，一般是/usr/local/下

RPM安装服务和源码包安装服务的区别

 RPM安装服务和源码包安装服务的区别

就是安装位置的不同

 源码包安装在指定位置，一般是/usr/local/

 RPM包安装在默认位置中

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

12.2.1 独立服务的管理

1、 RPM包安装服务的位置

 RPM安装服务和源码包安装服务的区别

就是安装位置的不同

 源码包安装在指定位置，一般是/usr/local/

 RPM包安装在默认位置中



/etc/init.d/ 启动脚本位置 /etc/rc.d/init.d 是启动脚本的软连接

/etc/sysconfig/ 初始化环境配置文件位置

/etc/ 配置文件位置

/etc/xinetd.conf xinetd配置文件

/etc/xinetd.d/ 基于xinetd服务的启动脚本

/var/lib/ 服务产生的数据放在这里

/var/log/ 日志

2、独立服务的启动

 /etc/init.d/独立服务名 start|stop|status|restart|

 service 独立服务名 start|stop|restart||status //redhat系列专用

service --status-all //列出所有rpm包的状态

3、独立服务的自启动

 chkconfig [--level 运行级别] [独立服务名] [on|off]

chkconfig --level 2345 httpd on

或者chkconfig httpd on

 修改/etc/rc.d/rc.local文件 //etc/rc.local 是一个软连接

 使用ntsysv命令管理自启动 //redhat系列专用

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

12.2.2 基于xinetd服务的管理

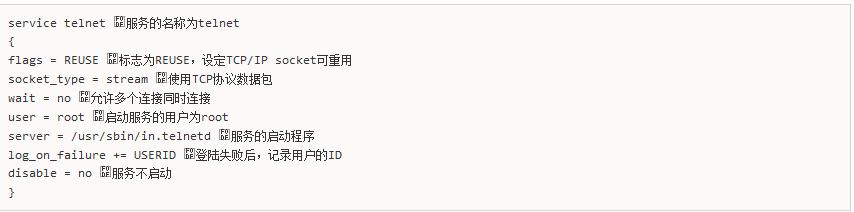
1、安装xinetd与telnet

[root@localhost ~]# yum -y install xinetd

[root@localhost ~]# yum -y install telnet-server

2、 xinetd服务的启动

[root@localhost ~]# vi /etc/xinetd.d/telnet



service telnet 服务的名称为telnet

{

flags = REUSE 标志为REUSE，设定TCP/IP socket可重用

socket\_type = stream 使用TCP协议数据包

wait = no 允许多个连接同时连接

user = root 启动服务的用户为root

server = /usr/sbin/in.telnetd 服务的启动程序

log\_on\_failure += USERID 登陆失败后，记录用户的ID

disable = no 服务不启动

}

重启xinetd服务

[root@localhost ~]# service xinetd restart

3、 xinetd服务的自启动

 [root@localhost ~]# chkconfig telnet on

 ntsysv

xinetd服务 的启动和自启动是关联的，一个设置后，另一个自动设置。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

12.3 源码包安装服务的管理

1、源码包安装服务的启动

 使用绝对路径，调用启动脚本来启动。不

同的源码包的启动脚本不同。可以查看源

码包的安装说明，查看启动脚本的方法。

/usr/local/apache2/bin/apachectl start|stop

2、源码包服务的自启动

[root@localhost ~]# vi /etc/rc.d/rc.local

加入

/usr/local/apache2/bin/apachectl start

3、让源码包服务被服务管理命令识别

 让源码包的apache服务能被service命令管理

启动

ln -s /usr/local/apache2/bin/apachectl /etc/init.d/apache

让源码包的apache服务能被chkconfig与

ntsysv命令管理自启动

vi /etc/init.d/apache

# chkconfig: 35 86 76

#指定httpd脚本可以被chkconfig命令管理。 格式是：

chkconfig： 运行级别 启动顺序 关闭顺序

# description: source package apache

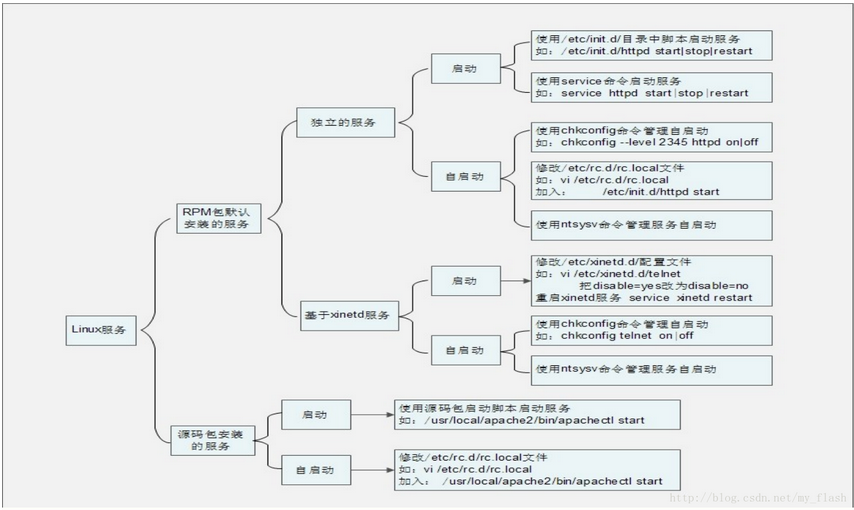
#说明，内容随意

[root@localhost ~]# chkconfig --add apache

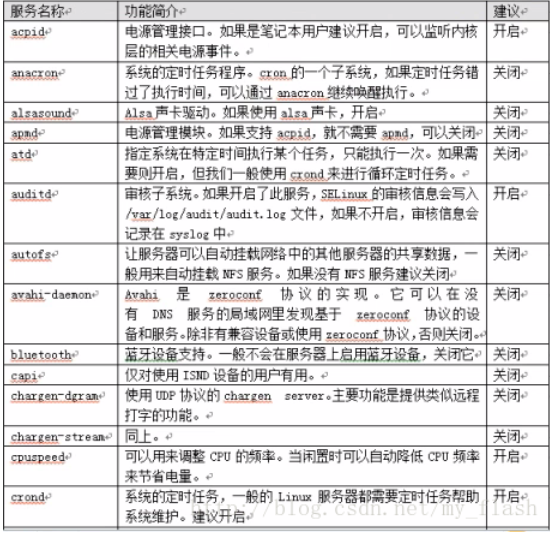
#把源码包apache加入chkconfig

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

12.4 服务管理总结



常见服务的作用



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

第十三讲 Linux系统管理

13.1 进程管理

13.1.1 进程查看

1、进程简介

 进程是正在执行的一个程序或命令，每一

个进程都是一个运行的实体，都有自己的

地址空间，并占用一定的系统资源。

2、进程管理的作用

 判断服务器健康状态

 查看系统中所有进程

 杀死进程

3、查看系统中所有进程

[root@localhost ~]# ps aux

#查看系统中所有进程，使用BSD操作系统格式

[root@localhost ~]# ps -le

#查看系统中所有进程，使用Linux标准命令格式。

 USER：该进程是由哪个用户产生的；

 PID：进程的ID号；

 %CPU：该进程占用CPU资源的百分比， 占用越高， 进程

越耗费资源；

 %MEM：该进程占用物理内存的百分比， 占用越高， 进程

越耗费资源；

 VSZ：该进程占用虚拟内存的大小， 单位KB；

 RSS：该进程占用实际物理内存的大小， 单位KB；

 TTY：该进程是在哪个终端中运行的。 其中tty1-tty7代表

本地控制台终端， tty1-tty6是本地的字符界面终端， tty7

是图形终端。 pts/0-255代表虚拟终端。

 STAT：进程状态。 常见的状态有： R： 运行、 S： 睡眠

、 T：停止状态、 s：包含子进程、 +：位于后台

 START：该进程的启动时间

 TIME：该进程占用CPU的运算时间， 注意不是系统时间

 COMMAND：产生此进程的命令名

4、查看系统健康状态

[root@localhost ~]# top [选项]

选项：

-d 秒数： 指定top命令每隔几秒更新。 默认是3秒

在top命令的交互模式当中可以执行的命令：

？ 或h： 显示交互模式的帮助

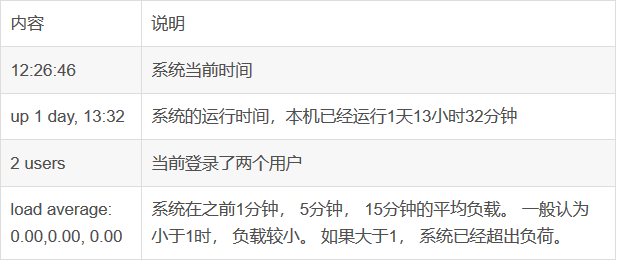
P： 以CPU使用率排序， 默认就是此项

M： 以内存的使用率排序

N： 以PID排序

q： 退出top

第一行信息为任务队列信息



内容 说明

12:26:46 系统当前时间

up 1 day, 13:32 系统的运行时间，本机已经运行1天13小时32分钟

2 users 当前登录了两个用户

load average: 0.00,0.00, 0.00 系统在之前1分钟， 5分钟， 15分钟的平均负载。 一般认为小于1时， 负载较小。 如果大于1， 系统已经超出负荷。

第二行为进程信息



内容 说明

Tasks: 95 total 系统中的进程总数

1 running 正在运行的进程数

94 sleeping 睡眠的进程

0 stopped 正在停止的进程

0 zombie 僵尸进程。 如果不是0， 需要手工检查僵尸进程

第三行为CPU信息



内容 说明

Cpu(s): 0.1%us 用户模式占用的CPU百分比

0.1%sy 系统模式占用的CPU百分比

0.0%ni 改变过优先级的用户进程占用的CPU百分比

99.7%id 空闲CPU的CPU百分比

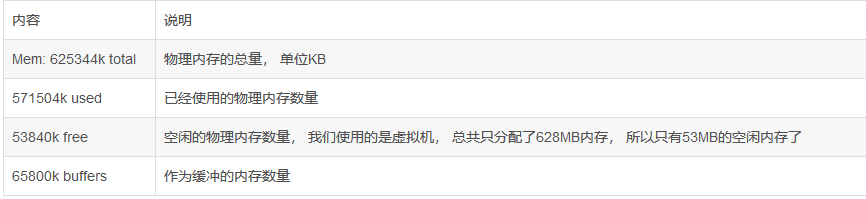
0.1%wa 等待输入/输出的进程的占用CPU百分比

0.0%hi 硬中断请求服务占用的CPU百分比

0.1%si 软中断请求服务占用的CPU百分比

0.0%st st（ Steal time） 虚拟时间百分比。 就是当有虚拟机时， 虚拟CPU等待实际CPU的时间百分比。

第四行为物理内存信息



内容 说明

Mem: 625344k total 物理内存的总量， 单位KB

571504k used 已经使用的物理内存数量

53840k free 空闲的物理内存数量， 我们使用的是虚拟机， 总共只分配了628MB内存， 所以只有53MB的空闲内存了

65800k buffers 作为缓冲的内存数量

第五行为交换分区（ swap）信息



内容 说明

Swap: 524280k total 交换分区（ 虚拟内存） 的总大小

0k used 已经使用的交互分区的大小

524280k free 空闲交换分区的大小

409280k cached 作为缓存的交互分区的大小

5、查看进程树

[root@localhost ~]# pstree [选项]

选项：

-p： 显示进程的PID

-u： 显示进程的所属用户

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

13.1 进程管理

13.1.2 终止进程

1、 kill命令

[root@localhost ~]# kill –l

#查看可用的进程信号



信号代号 信号名称 说明

1 SIGHUP 该信号让进程立即关闭， 然后重新读取配置文件之后重启。

2 SIGINT 程序终止信号， 用于终止前台进程。 相当于输出ctrl+c快捷键。

8 SIGFPE 在发生致命的算术运算错误时发出. 不仅包括浮点运算错误,还包括溢出及除数为0等其它所有的算术的错误。

9 SIGKILL 用来立即结束程序的运行. 本信号不能被阻塞、 处理和忽略。一般用于强制终止进程。

14 SIGALRM 时钟定时信号, 计算的是实际的时间或时钟时间. alarm函数使用该信号。

15 SIGTERM 正常结束进程的信号， kill命令的默认信号。 有时如果进程已经发生问题， 这个信号是无法正常终止进程的， 我们才会尝试SIGKILL信号， 也就是信号9。

18 SIGCONT 该信号可以让暂停的进程恢复执行， 本信号不能被阻断。

19 SIGSTOP 该信号可以暂停前台进程， 相当于输入ctrl+z快捷键。 本信号不能被阻断。

[root@localhost ~]# kill -1 22354

#重启进程

[root@localhost ~]# kill -9 22368

#强制杀死进程

2、 killall命令

[root@localhost ~]# killall [选项][信号] 进程名

#按照进程名杀死进程

选项：

-i： 交互式， 询问是否要杀死某个进程

-I： 忽略进程名的大小写

3、 pkill命令

[root@localhost ~]# pkill [选项] [信号] 进程名

#按照进程名终止进程

选项：

-t 终端号： 按照终端号踢出用户

按照终端号踢出用户

[root@localhost ~]# w

#使用w命令查询本机已经登录的用户

[root@localhost ~]# pkill -t -9 pts/1

#强制杀死从pts/1虚拟终端登录的进程

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

13.2 工作管理

1、把进程放入后台

tar -zcf etc.tar.gz /etc &

[root@localhost ~]#top

#在top命令执行的过程中，按下ctrl+z快捷键 在后台会暂停运行

2、查看后台的工作

[root@localhost ~]#jobs [-1]

选项：

-l：显示工作的PID

注：“+”号代表最近一个放入后台的工作，也是工作恢复时，默认恢复的工作，

“-”代表倒数第二个放入后台的工作

3、将后台暂停的工作恢复到前台执行

[root@localhost ~]#fg %工作号

参数：

%工作号： %号可以省略，但是注意工作号和PID 的区别

4、将后台暂停的工作恢复到后台执行

[root@localhost ~]#bg %工作号

注意： 后台恢复执行的命令，是不能和前台有交互的，否则不能恢复到后台执行

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

13.3 系统资源查看

1、 vmstat命令监控系统资源

[root@localhost ~]# vmstat [刷新延时 刷新次数]

例如：

[root@localhost proc]# vmstat 1 3

2、 dmesg开机时内核检测信息

[root@localhost ~]# dmesg

[root@localhost ~]# dmesg | grep CPU

3、 free命令查看内存使用状态

[root@localhost ~]# free [-b|-k|-m|-g]

选项：

-b： 以字节为单位显示

-k： 以KB为单位显示，默认就是以

KB为单位显示

-m： 以MB为单位显示

-g： 以GB为单位显示

缓存和缓冲的区别

简单来说缓存（ cache）是用来加速数据

从硬盘中“读取”的，而缓冲（ buffer）

是用来加速数据“写入”硬盘的。

4、 查看CPU信息

[root@localhost ~]# cat /proc/cpuinfo

5、 uptime命令

[root@localhost ~]# uptime

#显示系统的启动时间和平均负载，也就是top命令

的第一行。 w命令也可以看到这个数据。

6、 查看系统与内核相关信息

[root@localhost ~]# uname [选项]

选项：

-a： 查看系统所有相关信息；

-r： 查看内核版本；

-s： 查看内核名称。

判断当前系统的位数

[root@localhost ~]# file /bin/ls

查询当前Linux系统的发行版本

[root@localhost ~]# lsb\_release -a

7、 列出进程打开或使用的文件信息

[root@localhost ~]# lsof [选项]

#列出进程调用或打开的文件的信息

选项：

-c 字符串： 只列出以字符串开头的进程打开

的文件

-u 用户名： 只列出某个用户的进程打开的文

件

-p pid： 列出某个PID进程打开的文件

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

13.4 系统定时任务

1、 crond服务管理与访问控制

[root@localhost ~]# service crond restart

[root@localhost ~]# chkconfig crond on

2、用户的crontab设置

[root@localhost ~]# crontab [选项]

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

[root@localhost ~]# crontab -e

#进入crontab编辑界面。会打开vim编辑你的

工作。



\* \* \* \* \* 执行的任务



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月31日

第四个“ \*” 一年当中的第几月 1月12日

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 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



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。

举例

\*/5 \* \* \* \* /bin/echo ”11” >> /tmp/test

5 5 \* \* 2 /sbin/shutdown -r now

0 5 1,10,15 \* \* /root/sh/autobak.sh

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

第十四讲 日志管理

14.1 日志管理简介

1、日志服务

 在CentOS 6.x中日志服务已经由rsyslogd取

代了原先的syslogd服务。 rsyslogd日志服

务更加先进，功能更多。但是不论该服务

的使用，还是日志文件的格式其实都是和

syslogd服务相兼容的，所以学习起来基本

和syslogd服务一致。

 rsyslogd的新特点：

 基于TCP网络协议传输日志信息；

 更安全的网络传输方式；

 有日志消息的及时分析框架；

 后台数据库；

 配置文件中可以写简单的逻辑判断；

 与syslog配置文件相兼容。

确定服务启动

[root@localhost ~]# ps aux | grep rsyslogd

#查看服务是否启动

chkconfig --list | grep rsyslog

#查看服务是否自启动

2、常见日志的作用



日志文件 说 明

/var/log/cron 记录了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

记录错误登录的日志。 这个文件是二进制文件， 不能直接vi查看， 而要使用lastb命令查看， 命令如下：

[root@localhost log]# lastb

root tty1 Tue Jun 4 22:38 - 22:38 (00:00)

#有人在6月4日22:38使用root用户， 在本地终端1登录错误

/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， 不能直接vi， 而要使用lastlog命令查看。

/var/log/mailog 记录邮件信息。

/var/log/message 记录系统重要信息的日志。这个日志文件中会记录Linux系统的绝大多数重要信息， 如果系统出现问题时， 首先要检查的就应该是这个日志文件。

/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。比如说系统的登录， 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录、 注销信息， 同时记录系统的启动、 重启、关机事件。 同样这个文件也是一个二进制文件， 不能直接vi， 而需要使用last命令来查看。

/var/run/utmp 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。 同样这个文件不能直接vi， 而要使用w， who， users等命令来查询。

除了系统默认的日志之外，采用RPM方式安装的系统服务也会默认把日志记录在

/var/log/目录中（源码包安装的服务日志是在源码包指定目录中）。不过这些日志

不是由rsyslogd服务来记录和管理的，而是各个服务使用自己的日志管理文档来记录自身日志。



日志文件 说 明

/var/log/httpd/ RPM包安装的apache服务的默认日志目录

/var/log/mail/ RPM包安装的邮件服务的额外日志目录

/var/log/samba/ RPM包安装的samba服务的日志目录

/var/log/sssd/ 守护进程安全服务目录

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

14.2 rsyslogd日志服务

1、日志文件格式

 基本日志格式包含以下四列：

 事件产生的时间；

 发生事件的服务器的主机名；

 产生事件的服务名或程序名；

 事件的具体信息。

2、 /etc/rsyslog.conf配置文件

authpriv.\* /var/log/secure

#服务名称[连接符号]日志等级 日志记录位置

#认证相关服务.所有日志等级 记录在/var/log/secure日志中

服务名称



服务名称 说 明

auth 安全和认证相关消息（ 不推荐使用authpriv替代）

authpriv 安全和认证相关消息（ 私有的）

cron 系统定时任务cront和at产生的日志

daemon 和各个守护进程相关的日志

ftp ftp守护进程产生的日志

kern 内核产生的日志（ 不是用户进程产生的）

local0-local7 为本地使用预留的服务

lpr 打印产生的日志

mail 邮件收发信息

news 与新闻服务器相关的日志

syslog 有syslogd服务产生的日志信息（ 虽然服务名称已经改为rsyslogd， 但是很多配置都还是沿用了syslogd的， 这里并没有修改服务名） 。

user 用户等级类别的日志信息

uucp uucp子系统的日志信息， uucp是早期linux系统进行数据传递的协议， 后来也常用在新闻组服务中。

连接符号

连接符号可以识别为：

 “ \*”代表所有日志等级，比如：“ authpriv.\*”代表authpriv认证信息服务产生

的日志，所有的日志等级都记录

 “ .”代表只要比后面的等级高的（包含该等级）日志都记录下来。

比如：“ cron.info”代表cron服务产生的日志，只要日志等级大于等于info级别，就记录

 “ .=”代表只记录所需等级的日志，其他等级的都不记录。

比如：“ \*.=emerg”代表人和日志服务产生的日志，只要等级是emerg等级就记录。这种用法及少见，了解就好

 “ .!”代表不等于，也就是除了该等级的日志外，其他等级的日志都记录。

日志等级



等级名称 说 明

debug 一般的调试信息说明

info 基本的通知信息

notice 普通信息， 但是有一定的重要性

warning 警告信息， 但是还不回影响到服务或系统的运行

err 错误信息， 一般达到err等级的信息以及可以影响到服务或系统的运行了。

crit 临界状况信息， 比err等级还要严重

alert 警告状态信息， 比crit还要严重。 必须立即采取行动

emerg 疼痛等级信息， 系统已经无法使用了

日志记录位置

 日志文件的绝对路径， 如“ /var/log/secure”

 系统设备文件， 如“ /dev/lp0”

 转发给远程主机，如“ @192.168.0.210:514”

 用户名， 如“ root”

 忽略或丢弃日志，如“ ~”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

14.3 日志轮替

1、 日志文件的命名规则

 如果配置文件中拥有“ dateext”参数，那么日志会用日期来作为日志文件的后缀，

例如“ secure-20130605”。这样的话日志文件名不会重叠，所以也就不需要日志文件的改名，只需要保存指定的日志个数，删除多余的日志文件即可。

 如果配置文件中没有“ dateext”参数，那么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。

2、 logrotate配置文件



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 ownergroup 建立新日志， 同时指定新日志的权限与所有者和所属组。 如create 0600 root utmp

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。 如mail shenc@lamp.net

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

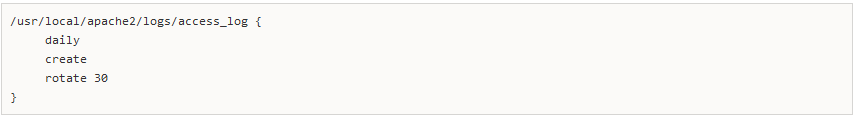
minsize 大小 日志轮替的最小值。 也就是日志一定要达到这个最小值才会轮替， 否则就算时间达到也不轮替

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。 如size 100k

dateext 使用日期作为日志轮替文件的后缀。 如secure-20130605

3、把apache日志加入轮替

[root@localhost ~]# vi /etc/logrotate.conf



/usr/local/apache2/logs/access\_log {

daily

create

rotate 30

}

4、 logrotate命令

[root@localhost ~]# logrotate [选项] 配置文件名

选项：

如果此命令没有选项，则会按照配置文件中的条件进行日志轮替

-v： 显示日志轮替过程。加了-v选项，会显示日志的轮替的过程

-f： 强制进行日志轮替。不管日志轮替的条件是否已经符合，强制配置文件中所有的日志进行轮替

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

第十五讲 启动管理

15.1 CentOS 6.x启动管理

15.1.1 系统运行级别

1、运行级别



运行级别 含 义

0 关机

1 单用户模式， 可以想象为windows的安全模式， 主要用于系统修复

2 不完全的命令行模式， 不含NFS服务

3 完全的命令行模式， 就是标准字符界面

4 系统保留

5 图形模式

6 重启动

2、运行级别命令

[root@localhost ~]# runlevel

#查看运行级别命令

[root@localhost ~]# init 运行级别

#改变运行级别命令

3、系统默认运行级别

[root@localhost ~]# vim /etc/inittab

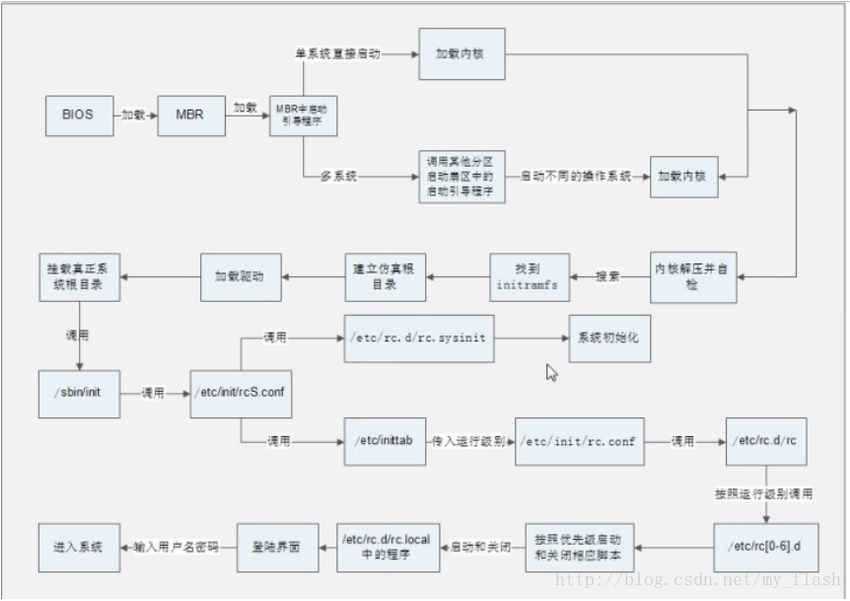
id:3:initdefault:

#系统开机后直接进入哪个运行级别

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

15.1.2 系统启动过程

启动流程图



initramfs内存文件系统

 CentOS 6.x中使用initramfs内存文件系统取代了CentOS 5.x中的initrd RAM Disk。他们的作用类似，可以通过启动引导程序

加载到内存中，然后加载启动过程中所需要的内核模块，比如USB、 SATA、 SCSI硬盘的驱动和LVM、 RAID文件系统的驱动

mkdir /tmp/initramfs

#建立测试目录

cp /boot/initramfs-2.6.32-279.el6.i686.img /tmp/initramfs/

#复制initramfs文件

cd /tmp/initramfs/

file initramfs-2.6.32-279.el6.i686.img

mv initramfs-2.6.32-279.el6.i686.img initramfs-2.6.32-279.el6.i686.img.gz

#修改文件的后缀名为.gz

gunzip initramfs-2.6.32-279.el6.i686.img.gz

#解压缩

file initramfs-2.6.32-279.el6.i686.img

cpio -ivcdu < initramfs-2.6.32-279.el6.i686.img

#解压缩

调用/etc/init/rcS.conf配置文件

主要功能是两个：

 先调用/etc/rc.d/rc.sysinit，然后由/etc/rc.d/rc.sysinit配置文件进行Linux系统初始化。

 然后再调用/etc/inittab，然后由/etc/inittab配置文件确定系统的默认运行级别。

由/etc/rc.d/rc.sysinit初始化

1、获得网络环境

2、挂载设备

3、开机启动画面Plymouth（取替了过往的 RHGB）

4、判断是否启用SELinux

5、显示于开机过程中的欢迎画面

6、初始化硬件

7、用户自定义模块的加载

8、配置内核的参数

9、设置主机名

10、同步存储器

11、设备映射器及相关的初始化

12、初始化软件磁盘阵列（ RAID）

13、初始化 LVM 的文件系统功能

14、检验磁盘文件系统（ fsck）

15、设置磁盘配额(quota)

16、重新以可读写模式挂载系统磁盘

17、更新quota（非必要）

18、启动系统虚拟随机数生成器

19、配置机器（非必要）

20、清除开机过程当中的临时文件

21、创建ICE目录

22、启动交换分区（ swap）

23、将开机信息写入/var/log/dmesg文件中

调用/etc/rc.d/rc文件

运行级别参数传入/etc/rc.d/rc这个脚本之后，由这个脚本文件按照不同的运行级别启动/etc/rc[0-6].d/目录中的相应的程序

 /etc/rc3.d/k??开头的文件（ ??是数字），会按照数字顺序依次关闭

 /etc/rc3.d/S??开头的文件（ ??是数字），会按照数字顺序依次启动

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

15.2 启动引导程序grub

15.2.1 Grub配置文件

1、 grub中分区表示



硬盘 分区 Linux中设备文件名 Grub中设备文件名

第一块SCSI硬盘 第一个主分区 /dev/sda1 hd(0,0)

第二个主分区 /dev/sda2 hd(0,1)

扩展分区 /dev/sda3 hd(0,2)

第一个逻辑分区 /dev/sda5 hd(0,4)

第二块SCSI硬盘 第一个主分区 /dev/sdb1 hd(1,0)

第二个主分区 /dev/sdb2 hd(1,1)

扩展分区 /dev/sdb3 hd(1,2)

第一个逻辑 /dev/sdb5 hd(1,4)

2、 grub配置文件

vi /boot/grub/grub.conf /etc/grub.conf 是/boot/grub/grub.conf的软连接



default=0 #默认启动第一个系统

timeout=5 #等待时间，默认是5秒

splashimage=(hd0,0)/grub/splash.xpm.gz

#这里是指定grub启动时的背景图像文件的保存位置的

hiddenmenu #隐藏菜单

title CentOS (2.6.32-279.el6.i686)

#title就是标题的意思

root (hd0,0)

#是指启动程序的保存分区

kernel /vmlinuz-2.6.32-279.el6.i686 ro root=UUID=b9a7a1a8-767f-4a87-8a2b-a535edb362c9 rd\_NO\_LUKS KEYBOARDTYPE=pc KEYTABLE=us rd\_NO\_MD crashkernel=auto LANG=zh\_CN.UTF-8 rd\_NO\_LVM rd\_NO\_DM rhgb quiet

#定义内核加载时的选项

initrd /initramfs-2.6.32-279.el6.i686.img

#指定了initramfs内存文件系统镜像文件的所在位置

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

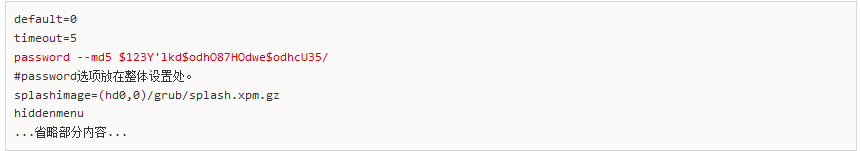
15.2.2 Grub加密与字符界面分辨率调整

1、grub加密

[root@localhost ~]#grub-md5-crypt

#生成加密密码串

[root@localhost ~]#vi /boot/grub/grub.conf



default=0

timeout=5

password --md5 $123Y'lkd$odhO87HOdwe$odhcU35/

#password选项放在整体设置处。

splashimage=(hd0,0)/grub/splash.xpm.gz

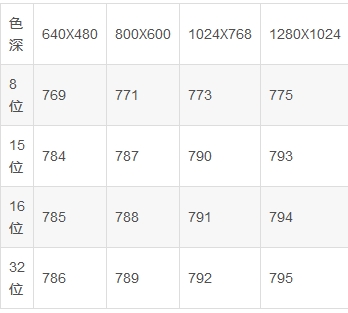
hiddenmenu

...省略部分内容...

2、纯字符界面分辨率调整

grep "CONFIG\_FRAMEBUFFER\_CONSOLE" /boot/config-2.6.32-279.el6.i686

#查询内核是否支持分辨率修改



色深 640X480 800X600 1024X768 1280X1024

8位 769 771 773 775

15位 784 787 790 793

16位 785 788 791 794

32位 786 789 792 795

vi /boot/grub/grub.conf

kernel /vmlinuz-2.6.32-279.el6.i686 ro root=UUID=b9a7a1a8-767f-4a87-8a2b-a535edb362c9 rd\_NO\_LUKS KEYBOARDTYPE=pc KEYTABLE=us rd\_NO\_MD crashkernel=auto LANG=zh\_CN.UTF-8 rd\_NO\_LVM rd\_NO\_DM rhgb quiet vga=791

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

15.3 系统修复模式

1、单用户模式



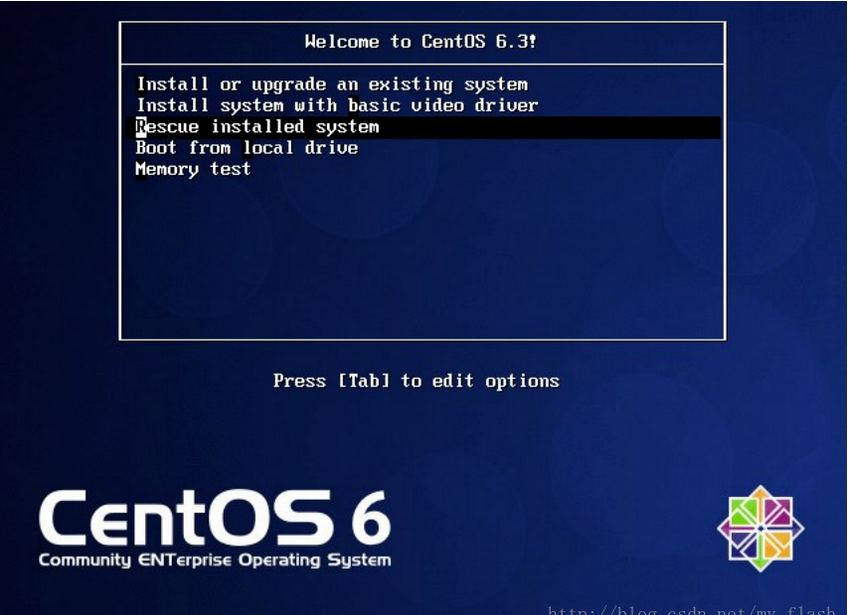
单用户模式常见的错误修复

遗忘root密码

kernel /vmlinuz-2.6.32-279.el6.i686 ro root=UUID=b9a7a1a8-767f-4a87-8a2b-a535edb362c9 rd\_NO\_LUKS KEYBOARDTYPE=pc KEYTABLE=us rd\_NO\_MD crashkernel=auto LANG=zh\_CN.UTF-8 rd\_NO\_LVM rd\_NO\_DM rhgb quiet 1

修改系统默认运行级别

2、 光盘修复模式



重要系统文件丢失，导致系统无法启动

bash-4.1# chroot /mnt/sysimage

#改变主目录

sh-4.1# cd /root

sh-4.1# rpm -qf /etc/inittab

#查询下/etc/inittab文件属于哪个包。

sh-4.1# mkdir /mnt/cdrom

#建立挂载点

sh-4.1# mount /dev/sr0 /mnt/cdrom

#挂载光盘

sh-4.1# rpm2cpio \

/mnt/cdrom/Packages/initscripts-8.45.3-1.i386.rpm \

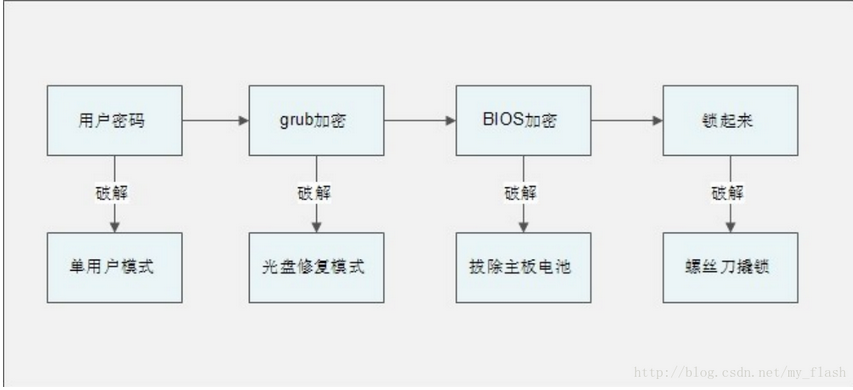
| cpio -idv ./etc/inittab

#提取inittab文件到当前目录

sh-4.1# cp etc/inittab /etc/inittab

#复制inittab文件到指定位置

3、 Linux的安全性



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

第十六讲 备份与恢复

16.1 备份概述

1、 Linux系统需要备份的数据

 /root/目录：

 /home/目录：

 /var/spool/mail/目录：

 /etc/目录：

 其他目录：

安装服务的数据

 apache需要备份的数据

 配置文件

 网页主目录

 日志文件

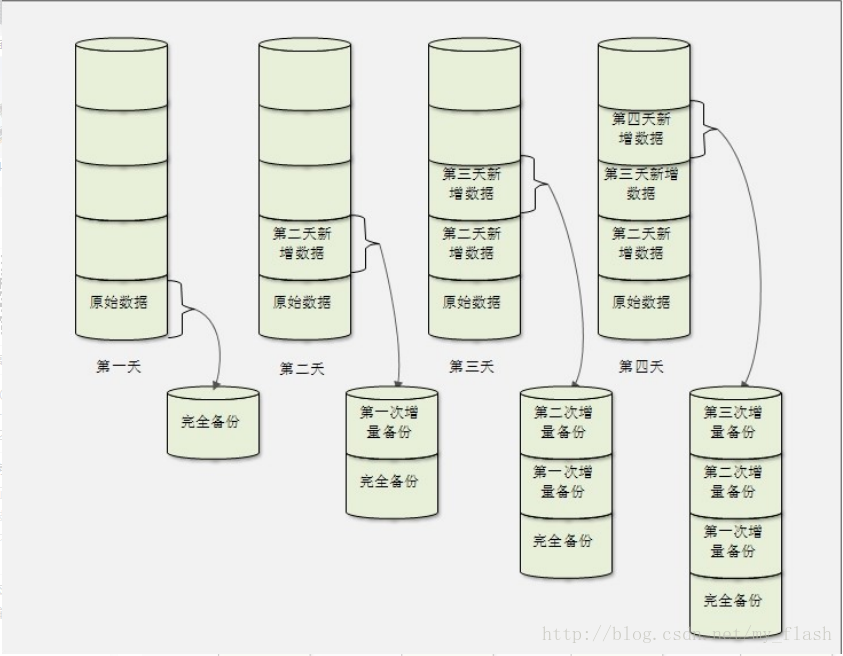
 mysql需要备份的数据

 源码包安装的mysql： /usr/local/mysql/data/

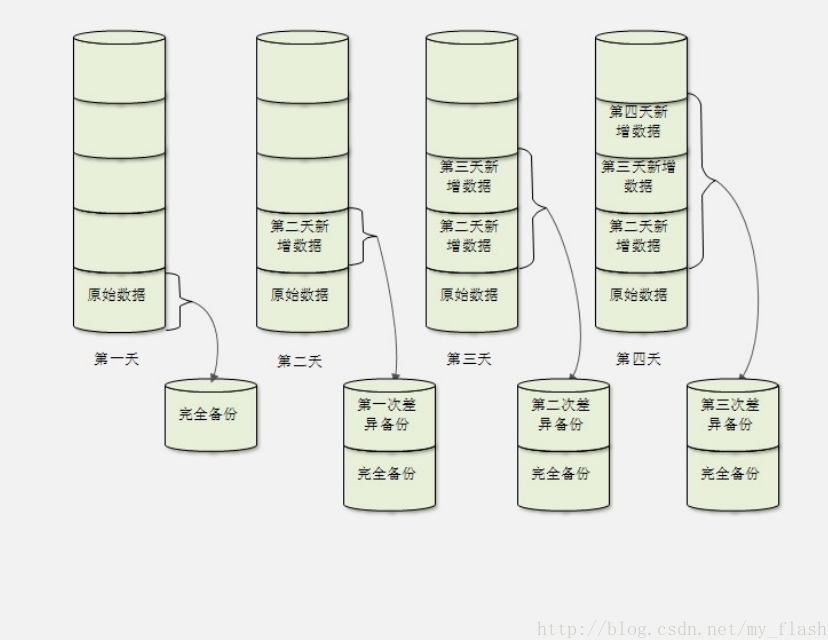
 RPM包安装的mysql： /var/lib/mysql/

2、备份策略

 完全备份： 完全备份就是指把所有需要备份的数据全部备份，当然完全备份可以备份整块硬盘，整个分区或某个具体的目录增量备份



差异备份



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

第十六讲 备份与恢复

16.2 dump和restore命令

1、 dump命令

[root@localhost ~]# dump [选项] 备份之后的文件名 原文件或目录

选项：

-level： 就是我们说的0-9十个备份级别

-f 文件名： 指定备份之后的文件名

-u： 备份成功之后，把备份时间记录在/etc/dumpdates文件

-v： 显示备份过程中更多的输出信息

-j： 调用bzlib库压缩备份文件，其实就是把备份文件压缩

为.bz2格式， 默认压缩等级是2

-W： 显示允许被dump的分区的备份等级及备份时间

备份分区

dump -0uj -f /root/boot.bak.bz2 /boot/

#备份命令。先执行一次完全备份，并压缩和更新备份时间

cat /etc/dumpdates

#查看备份时间文件

cp install.log /boot/

#复制日志文件到/boot分区

dump -1uj -f /root/boot.bak1.bz2 /boot/

#增量备份/boot分区，并压缩

dump –W

#查询分区的备份时间及备份级别的

备份文件或目录

dump -0j -f /root/etc.dump.bz2 /etc/

#完全备份/etc/目录， 只能使用0级别进行完全备份

，而不再支持增量备份

2、 restore命令

[root@localhost ~]# restore [模式选项] [选项]

模式选项： restore命令常用的模式有以下四种，这四个模式不能混用。

-C：比较备份数据和实际数据的变化

-i： 进入交互模式，手工选择需要恢复的文件。

-t： 查看模式，用于查看备份文件中拥有哪些数据。

-r： 还原模式，用于数据还原。

选项：

-f： 指定备份文件的文件名

比较备份数据和实际数据的变化

mv /boot/vmlinuz-2.6.32-279.el6.i686 /boot/vmlinuz-2.6.32-279.el6.i686.bak

#把/boot目录中内核镜像文件改个名字

restore -C -f /root/boot.bak.bz2

#restore发现内核镜像文件丢失

查看模式

restore -t -f boot.bak.bz2

还原模式

#还原boot.bak.bz2分区备份

#先还原完全备份的数据

mkdir boot.test

cd boot.test/

restore -r -f /root/boot.bak.bz2

#解压缩

restore -r -f /root/boot.bak1.bz2

#恢复增量备份数据

#还原/etc/目录的备份etc.dump.bz2

restore -r -f etc.dump.bz2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

---------------------

作者：my\_flash

来源：CSDN

原文：https://blog.csdn.net/my\_flash/article/details/79348390

版权声明：本文为博主原创文章，转载请附上博文链接！

https://blog.csdn.net/my\_flash/article/details/79348390