第4部分
用户与用户组管理

第4部分　用户与用户组管理

4.1 系统管理概述

4.2 启动与关闭系统

4.3 用户管理

4.4 系统备份

4.1 系统管理概述

系统管理是指针对系统进行的一些日常管理和维护性工作，以保证系统安全、可靠地运行，保证用户能够合理、有效地使用系统资源来完成任务。

4.1.1 系统管理工作的内容

首先介绍基本系统管理，主要包括以下几项内容：

(1)用户管理；

(2)系统备份；

4.1.2 系统管理工具

系统管理员通常使用以下3种方法来管理和维护系统：

(1) **直接编辑系统配置文件和脚本文件**。Linux系统的所有配置文件都是**纯文本文件**，大多数系统配置文件位于/etc和/usr/etc目录下，可以用vi等编辑器直接修改。这是最基本的有时也是唯一可用的手段。

(2) 使用Shell命令。Linux系统提供了丰富的系统管理命令，大多数管理命令位于/sbin和/usr/sbin目录下。这些命令是最安全、最有效，也是最灵活的系统管理工具。

(3) 使用图形化管理工具。Linux的各个发行版都提供了一些图形界面的系统管理工具。这类工具使用起来简单方便，能完成大部分管理工作。

4.1.3 root的权威性与危险性

与Windows系统的Administrator账号相比，Linux赋予root更多的权限。

root几乎可以对系统做任何事情，它拥有对系统内所有用户的管理权，对所有文件和进程的处置权，以及对所有服务的使用权。

由于root账号的权威性，系统管理员要严格保护root口令，防止口令泄露。口令应足够复杂、足够长，并经常更换。

此外，系统管理员还应具有一个普通用户的账号，登录时(尤其是远程登录时)应以普通用户身份进入系统，只在必要时变换成root。

- Linux是一个多用户系统，通常会拥有少至几个多至几百个的可登录用户。为确保系统的安全性和有效性，必须对用户进行妥善的管理和控制，这是系统管理的一项重要工作。

4.2 用户管理

4.2.1 用户管理概述

用户管理就是对用户账号进行管理。

用户账号是用户在系统中的标识，用以鉴别用户身份，限制用户的权限，防止用户非法或越权使用系统资源。

用户管理的工作包括建立、删除用户和用户组，以及管理用户的登录口令等。

◆ 用户

系统中每个用户拥有一个唯一的用户名(login name)和用户标识符(UID)。用户名供用户登录系统使用，而系统则通过UID来识别用户、定义文件和进程的归属关系。系统将用户分为以下3类:

(1) 超级用户：每个系统都有一个超级用户账号，在安装系统时建立。超级用户的用户名为root，UID是0。

(2) 普通用户：普通用户是指除root外的可登录的用户，由root建立。普通用户的UID大于或等于500。

(3) 特殊用户： 特殊用户是系统内部使用的账号，不能登录使用。特殊用户的账号有bin、sys、nobody、daemon等，UID为1～499。通常这些账号只能被系统守护进程使用，用来访问具有特殊UID的文件。

系统中每个用户都对应一个用户账户(保存在passwd文件中)和口令(保存在shadow文件中)。root和普通用户还拥有自己的主目录和邮箱。

用户登录后可以用su(super)命令改变身份，常用于系统管理员在必要时从普通用户改变到root。

su命令

【功能】转变为另一个用户。

【格式】su [-] [用户名]

【说明】不指定用户名时，转换到root；指定“-”选项时，同时变换环境。普通用户执行su时，须输入要转变为的用户口令。

例4.1 转变为root：

```
$ su -                    #转变为root
password:(输入root的口令)
#(转换为root账号，环境也变为root的环境)
...
# exit
$_(回到原来用户账号)
```

- su 的确为管理带来方便，通过切换到root下，能完成所有系统管理工具，只要把root的密码交给任何一个普通用户，他都能切换到root来完成所有的系统管理工作；
- 但是su 工具在多人参与的系统管理中，并不是最好的选择，su只适用于一两个人参与管理的系统，因为su 使普通用户切换到超级权限用户root后，权限是无限制的。

- sudo 授权许可使用的su，也是受限制的su
- 通过sudo，能把某些超级权限有针对性的下放
- 通过修改sudo的配置文件/etc/sudoers来进行授权
- 当前用户通过sudo切换到root（或其它指定切换到的用户），然后以root（或其它指定的切换到的用户）身份执行命令，执行完成后，直接退回到当前用户

◆ 用户组

用户组是可共享文件和其他系统资源的用户集合。分组的原则可以是按工作关系或用户性质来划分。

例如，参与同一个项目的用户可以形成一个组。一个组中可以包含多个用户，同组用户具有相同的组权限。一个用户也可以归属于多个组。每个用户组有一个组账户(保存在group文件中)，用唯一的组名和组标识符GID标识。

4.2.2 用户管理的相关文件

(1) passwd文件

用户账号文件/etc/passwd存放用户账户的基本信息。每个用户账户占一行，每行由7个域组成，用冒号分隔各个域，格式如下：

登录名:密码:用户标识符UID:组标识符GID:用户信息:主目录:登录Shell

passwd文件的属主为root，权限为644，即任何人可读，root可读/写。

```
[root@localhost root]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

```
zhaoyb:x:500:500::/home/zhaoyb:/bin/bash
user:x:1000:1000::/home/user:/bin/bash
zhou:x:1001:1001::/home/zhou:/bin/bash
[root@localhost root]#
```

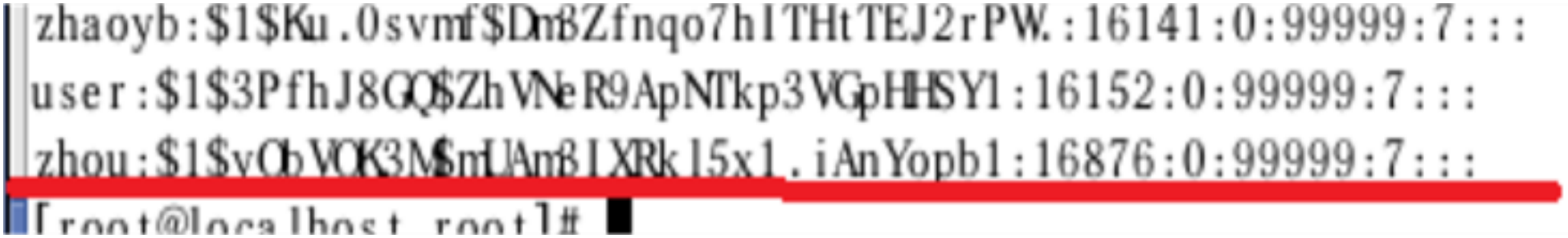
登录名:密码:用户标识符UID:组标识符GID:用户信息:主目录:登录Shell

(2) shadow文件

shadow技术：加密后的口令放在/etc/shadow文件中，而在passwd中相应的位置只放一个“x”。格式如下：

登录名:加密口令:口令上次更改时间:口令不可变更天数:口令需要重新更改的天数:口令失效前警告用户的天数:口令失效距账号被封的天数:账号被封时间:保留字段

shadow文件的属主为root，权限为400。



密码：可以看到**3**类，分别是奇怪的字符串、*和！！其中，**奇怪的字符串就是加密**过的密码文件。星号代表帐号被锁定，双叹号表示这个密码已经过期了。

这里显示的**1 表明是用MD5算法**转变过的

修改日期：这个是表明上一次修改密码的日期与**1970-1-1**相距的天数

密码不可改的天数：假如这个数字是**8**，则**8**天内不可改密码，如果是**0**，则随时可以改。

密码需要修改的期限：如果是**99999**则永远不用改。

(3) group文件

组账号文件/etc/group保存了各个组的账号信息，每个组占一行，每行包括4个域，格式如下：

组名:口令:组描述符GID:用户列表

group文件的属主为root，权限为644。

```
[root@localhost root]# cat /etc/group
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
```

组名:口令:组描述符GID:用户列表

4.2.3 用户管理

◆ 添加用户

添加用户的命令是useradd，它主要完成以下工作：

(1) 向passwd、shadow和group文件写入用户信息。

(2) 在/home目录下建立用户主目录。

(3) 将/etc/skel目录下的文件拷贝到用户主目录下，作为用户的环境初始化脚本。

(4) 在/var/spool/mail目录下建立用户的邮箱。

useradd命令

【功能】添加一个新用户。

【格式】useradd [选项] 用户名

【选项】

-d 目录　　指定用户的主目录，否则使用默认的主目录/home/用户名。

-e 日期 指定用户账号的终止日期，格式为YYYY-MM-DD。

-g 组名 指定用户的用户组，否则默认使用与UID相同的GID。

-s *shell* 指定用户的登录Shell，否则默认使用bash。

例4.2 用useradd命令添加用户：

 # useradd -g faculty zhaoxin #添加新用户zhaoxin，组为faculty

 # useradd -e 2020-12-31 liuliu #添加新用户liuliu，到2020年底终止

◆ 设置用户口令

系统管理员须为新添加的用户设置第一个口令。口令经算法转变后，存在shadow文件中。此后用户可以登录修改自己的口令，root可修改任何用户的口令。

有些时候，root需要对用户的口令设置某些限制。比如，为督促用户定期更换口令，root可以设置用户口令的期限。通过修改/etc/shadow文件中相应行的第5个域来实现的。

有时出于用户本身的原因或系统安全的需要，root需要封锁一个账号。被封锁账号的口令会暂时失效，不能登录直至解禁。封锁用户的方法是在/etc/shadow文件中找到指定的用户，在其口令域的前面插入一个“!”符号。解封时删除这个标记即可。

以上对用户口令的操作都可通过编辑/etc/shadow文件或使用passwd命令来完成。

passwd命令

【功能】设置用户口令。

【格式】passwd [选项] [用户名]

【选项】

-d　　删除用户的口令，使用户登录时不需要口令。

-l　　封锁用户账号，使用户暂无法登录。

-u　　解除封锁用户账号，使用户恢复登录。

-x*n*　　设置口令的有效期限为n天。口令到期后必须重新设置才可登录。

【说明】没有指定用户名时则是修改自己的口令。

例4.3 用passwd命令设置口令：

passwd zhaoxin　　#为新用户zhaoxin设置口令

…(输入口令)

passwd -x10 zhaoxin　　#设置zhaoxin口令的有效期为10天

#

$ passwd　　#用户修改自己的密码

Enter new passed: xxxxx

Re-enter new passed:xxxxx

只有超级用户才能修改其他用户的密码

3. 删除用户

删除用户的命令是userdel，它主要完成以下工作：

(1) 删除passwd和shadow文件中此用户的行。

(2) 修改group文件，如果该用户是组中唯一的成员则删除该组的行，否则从组用户列表中删除该用户。

(3) 若带有-r选项，则删除用户的主目录以及此用户的mail邮箱。

userdel命令

【功能】删除用户

【格式】userdel [-r] 用户名

【选项】该命令只有一个-r选项，表示删除用户的同时删除其主目录及mail邮箱。没有此选项时则保留它们。

例4.4 用userdel命令删除用户：

```
# userdel -r zhaoxin        #删除用户zhaoxin，不保留其主目录和邮箱
```

4.2.4 用户组管理

Linux中如果创建用户时不指定用户组，则系统默认为用户生成一个组，其组名与用户名相同。如果需要分组，则应先建立起用户组，然后向组中添加用户。

建立一个用户组的命令是groupadd，格式是:

groupadd[组名]。

删除一个用户组的命令是groupdel，格式是groupdel [组名]。删除组时，若该组中仍包含有用户，则必须先将这些用户从组中删除(或改变他们的组)，然后才能删除组。

4.3 系统备份

计算机系统在运行过程中不可避免地会发生各种故障，包括**软硬件异常**、**操作失误**和外界环境变化造成的**系统崩溃或文件丢失**。虽然现在的系统都具有一定的容错和安全措施，但都不能替代简单可靠的**备份操作**。

4.3.1 备份策略

备份的方式可以分为以下几种：

(1) **完全备份**：就是指对某一个时间点上的所有数据或应用进行的一个完全拷贝。

实际应用中就是用一盘磁带对整个系统进行完全备份，包括其中的**系统和所有数据**。

完全备份方式最大的好处就是只要用一盘磁带，就可以恢复丢失的数据。因此大大加快了系统或数据的恢复时间。

它的不足之处在于：

- 各个全备份磁带中的备份数据存在大量的重复信息；
- 由于每次需要备份的数据量相当大，因此备份所需时间较长。

(2) 差异备份：备份上一次完全备份后改变的所有数据。差异备份的备份和恢复工作量居中。恢复时需要先恢复上一次的完全备份，再恢复最近一次的差异备份。

差异备份过程中，只备份有标记的那些选中的文件和文件夹。它不清除标记，也即备份后不标记为已备份文件。换言之，不清除存档属性。

- 举例来说，
- 在星期一，网络管理员按惯例进行系统完全备份；
- 在星期二，假设系统内只多了一个资产清单，于是管理员只需将这份资产清单一并备份下来即可；
- 在星期三，系统内又多了一份产品目录，于是管理员不仅要将这份目录，还要连同星期二的那份资产清单一并备份下来。
- 如果在星期四系统内有多了一张工资表，那么星期四需要备份的内容就是：工资表+产品目录+资产清单。

(3) **增量备份**：备份上一次**备份**后改变的所有数据。增量备份工作量小，但恢复较费力，需要从上一次的完全备份开始，逐级恢复随后的各个增量备份。

增量备份是备份自上一次备份（包含完全备份、差异备份、增量备份）之后有变化的数据。增量备份过程中，只备份有标记的选中的文件和文件夹，它清除标记，既：备份后标记文件，换言之，清除存档属性。

- 如果系统在星期四的早晨发生故障，丢失大批数据，那么现在就需要将系统恢复到星期三晚上的状态。
- 这时管理员需要首先找出星期一的那盘完全备份磁带进行系统恢复，然后再找出星期二的磁带来恢复星期二的数据，然后在找出星期三的磁带来恢复星期三的数据。

- 这种备份方式最显著的优点就是：没有重复的备份数据，因此备份的数据量不大，备份所需的时间很短。
- 但增量备份的数据恢复是比较麻烦的。您必须具有上一次全备份和所有增量备份磁带（一旦丢失或损坏其中的一盘磁带，就会造成恢复的失败），并且它们必须沿着从全备份到依次增量备份的时间顺序逐个反推恢复，因此这就极大地延长了恢复时间。

- 系统管理员应根据系统的使用情况制订备份方案并严格执行。
- 常用的方案是每月1～2次完全备份。每周末做一次差异备份，每个工作日做一次增量备份。
- 系统升级前必须进行完全备份。
- 备份的范围也要根据系统的使用情况来决定，原则是对经常改动的文件应该比改动较少的文件备份更频繁一些。

- 例如，对于多用户系统来说，/home目录中的用户文件是经常变化的，对于应用服务器系统来说，/var目录中的系统运行相关数据是经常变化的。这些目录需要每天都备份；
- /etc目录中的配置文件不需要频繁备份，根据配置更改的频繁程度每星期或每月备份一次即可；
- /usr和/opt目录中的程序文件很少发生变化，安装后做一次备份即可。
- 另外，有些目录如/tmp、/mnt等是没有必要备份的；
- 有些目录如/proc、/dev等则是不应该备份的。

4.3.2 备份命令

Linux系统提供了多种图形化的和命令方式的备份工具，用户可以选择使用。

命令方式的备份工具包括归档命令和压缩命令两类。

归档命令的功能是将要备份的文件打包成一个档案文件，写到存档介质上或备份目录下。在需要恢复时，用归档命令可以从档案文件中提取出文件，并写回文件系统中。

常用的归档和压缩命令如表所示

常用的压缩和归档命令

命　令	文件扩展名	功　能
compress	*.Z	压缩和解压文件
zip、unzip	*.zip	压缩和解压文件
gzip	*.gz	压缩和解压文件
tar	*.tar	归档工具，用于归档和提取文件
tar -Z	*.tar.Z	归档和提取文件时，用 compress 压缩和解压文件
tar -z	*.tar.gz	归档和提取文件时，用 gzip 压缩和解压文件
cpio	*.cpio	归档工具，更适合做系统备份
cpio -Z	*.cpio.Z	归档和提取文件时，用 compress 压缩和解压文件

以下仅对最常用的gzip和tar命令做介绍，其他命令见man手册。

(1) gzip命令

gzip命令用于对文件进行压缩和解压缩，其压缩率高于compress和zip命令，且可以和归档命令tar配合使用。

gzip命令

【功能】对文件进行压缩和解压缩。

【格式】gzip [选项] [文件名]

【选项】

-d　　解压缩。

-l　　列出压缩文件的大小和压缩比例等信息。

-r　　压缩子目录。

-v　　显示详细操作信息。

【说明】没有-d和-l选项时执行压缩。

例4.5 gzip命令用法示例：

$ ls

$ gzip -v *.c　　压缩当前目录下的每个.c文件

$ ls

```
[jhm@cilinux book]$ ls
doc1   doc2   include.c   init.c   math.c
[jhm@cilinux book]$
```

```
[jhm@cilinux book]$ gzip -v *.c
include.c:          0.0% -- replaced with include.c.gz
init.c:    0.0% -- replaced with init.c.gz
math.c:    0.0% -- replaced with math.c.gz
```

```
[jhm@cilinux book]$ ls
doc1   doc2   include.c.gz   init.c.gz   math.c.gz
[jhm@cilinux book]$
```

```
$ gzip -l math.c.gz      #显示压缩文件的信息，不解压
$ gzip -dv *.c.gz        #解压缩math.c.gz文件，显示详细信息
    math.c.gz:           45.6% -- replaced with math.c
$
```

(2) tar命令

tar命令用于将一组文件打包成一个文件，称为档案文件(archive)。归档的目的是为了便于对这些文件进行统一处理，如转储、传输、发布和下载等。档案文件比单个文件更节省存储空间，因为它消除了各个文件最后一块内的空闲空间。如果配合压缩命令则会进一步节省存储空间，减少传输时间。

【功能】文件归档工具，可备份整个目录、分区或文件系统。

【格式】tar [选项] [文件/目录列表]

【选项】

-c　　创建档案文件。

-x　　从档案文件中提取并还原文件。

-f 文件名　指定档案文件或归档设备的文件名。

当与-c选项一起使用时，创建的tar文件使用该选项指定的文件名；当与-x选项一起使用时，解除该选项指定的归档。

-p　　归档时保持文件的访问权限。

-r　　向档案文件中添加文件。

-t　　列出档案文件中的内容，显示列表。

-T　　从参数指定的文件中读取要备份的文件列表。

-u　　更新档案文件。

-v　　显示详细操作信息。

-z　　使用gzip的来压缩/解压缩文件。

--exclude 目录/文件　　不备份指定的目录或文件

【参数】参数为文件时，对文件进行操作；参数为目录时，对目录树中的所有文件进行操作；有-T选项时，参数应是文件，其内容被看作是要归档的文件名列表。

```
$ tar -cf  bak/src.tar *.[c,h]  #打包*.c和*.h文件，生成档案文件src.tar

$ tar -tf  src.tar        #显示档案文件内容（注意所在目录）


$ cd newdir
$ tar -xf ~/backup/bak/scr.tar        #在另一目录下解包
```

```
[jhm@cilinux backup]$ ls
bak  print.c  print.h  solo.c  version1
[jhm@cilinux backup]$ tar -cf bak/src.tar *.[c,h]
[jhm@cilinux backup]$ cd bak
[jhm@cilinux bak]$ ls
src.tar
[jhm@cilinux bak]$
```

```
[jhm@cilinux bak]$ tar -tf src.tar
print.c
print.h
solo.c
[jhm@cilinux bak]$
```

```
[jhm@cilinux backup]$ cd newdir
[jhm@cilinux newdir]$ tar -xf ~/backup/bak/scr.tar
[jhm@cilinux newdir]$ ls
print.c  print.h  solo.c
```

用户可以用这种方式打包和保存自己的文件，然后在必要时在某个目录下恢复这些文件。这种方式也常用来复制一个目录树到另一个位置。与用cp -r命令复制目录树的不同之处在于，tar命令可以保持文件的归属权和修改时间等属性不变。在以root身份进行复制操作时，这一点有时尤为有用。

例4.7 把/wenjian目录下包括它的子目录全部做备份文件并进行压缩，备份文件名为usr.tar.gz 。

```
$ tar -czvf usr.tar.gz /wenjian
```

把usr.tar.gz这个备份文件还原并解压缩。

```
$ tar -xzvf usr.tar.gz
```