项目 3　管理 Linux 操作系统

3.1　任 务 描 述

季目开关制造公司运营过程中，由于公司各部门文件权限没有严格限制，造成公司的研发资料以及行政部门的行政管理资料在公司内部随意传播访问，这就需要网络运维管理员对文件服务器上的文档进行权限规划和限制，并对各部分用户的权限进行设置。用户权限的任务要求：

（1）研发部开发人员 David 和 Peter 属于组 A。

（2）行政部人员 Jack 和 Mike 属于组 B。

（3）建立目录/project_a，该目录中的文件只能由研发部开发人员读取、增加、删除、修改以及执行，其他用户不能对该中进行任何访问操作。

（4）建立目录/project_b，该目录中的文件只能由行政部人员读取、增加、删除、修改以及执行，其他用户不能对该目录进行任何访问操作。

（5）建立目录/project，该目录中的文件可由研发部、行政部人员读取、增加、删除、修改以及执行，其他部门用户只可以对该目录进行只读的访问操作。

3.2　任 务 分 析

（1）理解用户与账户的概念。

（2）熟悉账户配置文件。

（3）学会设置基本操作权限。

（4）了解特殊权限的使用。

3.3　知 识 储 备

3.3.1　Linux 系统管理概述

Linux 操作系统是多用户的操作系统，允许多个用户同时登录到系统上，使用系统资源。当多个用户能同时使用系统时，为了使所有用户的工作都能顺利进行，保护每个用户的文件和进程，也为了系统自身的安全和稳定，必须建立一种秩序，使每个用户的权限都得到规范。

3.3.2 用户和组管理

1. 用户的概念

Linux是真正意义上的多用户操作系统,可以在Linux系统中建若干用户。例如,其他人想用我的计算机,但我不想让他用我的用户名登录,因为我的用户名下有不想让别人看到的资料和信息(也就是隐私内容)。这时我就可以给他建一个新的用户名,这从计算机安全角度来说是符合操作规则的;当然用户的概念理解还不仅仅于此,在Linux系统中还有一些用户是用来完成特定任务的,比如nobody和ftp等,我们访问一个Linux服务器的网页程序,就是nobody用户;我们匿名访问ftp时,会用到用户ftp或nobody。Linux系统的一些账号请查看 /etc/passwd。

2. 用户的类型

(1) root用户:系统唯一,是真实的,可以登录系统,可以操作系统任何文件和命令,拥有最高权限。

(2) 普通用户:这类用户能登录系统,但只能操作自己目录的内容;权限有限;这类用户都是系统管理员自行添加的。

3. 用户组的概念

用户组就是具有相同特征的用户的集合体。例如,有时要让多个用户具有相同的权限,如查看、修改某一文件或执行某个命令,这时需要用户组,把用户都定义到同一用户组,通过修改文件或目录的权限,让用户组具有一定的操作权限,这样用户组下的用户对该文件或目录都具有相同的权限,这是通过定义组和修改文件的权限来实现的。例如,为了让一些用户有权限查看某一文档,如是一个时间表,而编写时间表的人要具有读写执行的权限,想让一些用户知道这个时间表的内容,而不让他们修改,所以可以把这些用户都划到一个组,然后来修改这个文件的权限,让用户组可读,这样用户组下面的每个用户都是可读的。

用户和用户组的对应关系是:一对一、多对一、一对多或多对多。

一对一:某个用户可以是某个组的唯一成员。

多对一:多个用户可以是某个唯一的组的成员,不归属其他用户组,如beinan和linuxsir两个用户只归属于beinan用户组。

一对多:某个用户可以是多个用户组的成员,如beinan可以是root组成员,也可以是linuxsir用户组成员,还可以是adm用户组成员。

多对多:多个用户对应多个用户组,并且几个用户可以是归属相同的组。

4. 用户和组管理常用命令

1) 添加系统用户:useradd

格式:

```
useradd [选项] <用户名>
useradd -d                  /* 指定用户的宿主目录 */
useradd -e                  /* 指定用户的账号失效时间,可使用 YYYY-MM-DD 的日期格式 */
useradd -g                  /* 指定用户的基本组名,也可以使用 GID */
useradd -G                  /* 指定用户的公共组名,也可以使用 GID */
useradd -M                  /* 不为用户建立并初始化宿主目录 */
useradd -s                  /* 指定用户的登录 shell 环境 */
```

```
useradd -u                    /* 指定用户的 UID 号 */
```

2）设置系统用户密码 passwd

格式：

```
passwd [选项] <用户名>
passwd -d                     /* 清空指定用户密码 */
passwd -l                     /* 锁定指定用户账户 */
passwd -S                     /* 查看指定用户状态 */
passwd -u                     /* 解锁指定用户账户 */
```

3）修改指定用户账户信息 usermod

格式：

```
usermod [选项] <用户名>
```

4）删除指定用户账户 userdel

格式：

```
userdel [-r] <用户名>
userdel -r                    /* 删除用户后,也将该用户的宿主目录一并删除 */
```

5）添加一个系统用户组 groupadd

格式：

```
groupadd [-g] <组名>
groupadd -g                   /* 为新建的组指定 GID 组标记 */
```

6）删除一个系统用户组 groupdel

格式：

```
groupdel <组名>
```

7）输出指定用户的身份标记信息 id

格式：

```
id [选项] <用户名>
id -u                         /* 只显示有效用户信息 */
id -g                         /* 只显示有效组信息 */
id -n                         /* 只输出用户名称 */
```

8）查看登录到当前主机中的用户 users

格式：

```
users/who
```

9）切换为另一个用户身份 su

格式：

```
su [-l] [目标用户名]
su -l                         /* 使用目标用户的登录 shell 环境,该选项可简写为"-" */
```

5. 账号系统文件

完成用户管理的工作有多种方法,但是每种方法实际上都是对有关的系统文件进行修改。与用户和用户组相关的信息都存放在一些系统文件中,这些文件包括/etc/passwd,/etc/shadow,/etc/group 等。下面分别介绍这些文件的内容。

1) /etc/passwd 文件

这个文件是用户管理工作涉及的最重要的一个文件。Linux 系统中的每个用户都在/etc/passwd 文件中有一个对应的记录行,它记录了这个用户的一些基本属性。这个文件对所有用户都是可读的。例如,用 cat /etc/passwd 可以查看它的详细信息:

```
root:x:0:0:Superuser:/:
daemon:x:1:1:System daemons:/etc:
bin:x:2:2:Owner of system commands:/bin:
sys:x:3:3:Owner of system files:/usr/sys:
adm:x:4:4:System accounting:/usr/adm:
uucp:x:5:5:UUCP administrator:/usr/lib/uucp:
auth:x:7:21:Authentication administrator:/tcb/files/auth:
cron:x:Array:16:Cron daemon:/usr/spool/cron:
listen:x:37:4:Network daemon:/usr/net/nls:
lp:x:71:18:Printer administrator:/usr/spool/lp:
sam:x:200:50:Sam san:/usr/sam:/bin/sh
```

可以看到,/etc/passwd 中一行记录对应着一个用户,每行记录又被冒号分隔为 7 个字段,其格式和具体含义如下:

```
用户名:口令:用户标识号:组标识号:注释性描述:主目录:登录 Shell
```

(1)“用户名”是代表用户账号的字符串。通常长度不超过 8 个字符,并且由大小写字母和/或数字组成。登录名中不能有冒号,因为冒号在这里是分隔符。为了兼容起见,登录名中最好不要包含点字符(.),并且不使用连字符(-)和加号(+)打头。

(2)“口令”:一些系统中,存放着加密后的用户口令字。虽然这个字段存放的只是用户口令的加密串,不是明文,但是由于/etc/passwd 文件对所有用户都可读,所以这仍是一个安全隐患。因此,现在许多 Linux 系统(如 SVR4)都使用了 shadow 技术,把真正的加密后的用户口令字存放到/etc/shadow 文件中,而在/etc/passwd 文件的口令字段中只存放一个特殊的字符,如 x 或 * 。

(3)“用户标识号”是一个整数,系统内部用它来标识用户。一般情况下,它与用户名是一一对应的。如果几个用户名对应的用户标识号是一样的,系统内部将把它们视为同一个用户,但是它们可以有不同的口令、不同的主目录以及不同的登录 Shell 等。通常用户标识号的取值范围是 0~65 535。0 是超级用户 root 的标识号,1~Array,Array 由系统保留,作为管理账号,普通用户的标识号从 100 开始。在 Linux 系统中,这个界限是 500。

(4)“组标识号”字段记录的是用户所属的用户组。它对应着/etc/group 文件中的一条记录。

(5)“注释性描述”字段记录着用户的一些个人情况,如用户的真实姓名、电话、地址等,这个字段并没有什么实际的用途。在不同的 Linux 系统中,这个字段的格式并没有统一。在许多 Linux 系统中,这个字段存放的是一段任意的注释性描述文字,用作 finger 命令的输出。

(6)“主目录”,就是用户的起始工作目录,是用户在登录到系统之后所处的目录。在大多数系统中,各用户的主目录都被组织在同一个特定的目录下,而用户主目录的名称就是该用户的登录名。各用户对自己的主目录有读、写、执行(搜索)权限,其他用户对此目录的访问权限则根据具体情况设置。

(7) 用户登录后,要启动一个进程,负责将用户的操作传给内核,这个进程是用户登录到系统后运行的命令解释器或某个特定的程序,即 Shell。Shell 是用户与 Linux 系统之间的接口。Linux 的 Shell 有许多种,每种都有不同的特点。常用的有 sh(Bourne Shell)、csh(C Shell)、ksh(Korn Shell)、tcsh(TENEX/TOPS-20 type C Shell)、bash(Bourne Again Shell)等。系统管理员可以根据系统情况和用户习惯为用户指定某个 Shell。如果不指定 Shell,那么系统使用 sh 为默认的登录 Shell,即这个字段的值为/bin/sh。用户的登录 Shell 也可以指定为某个特定的程序(此程序不是一个命令解释器)。利用这一特点,可以限制用户只能运行指定的应用程序,在该应用程序运行结束后,用户就自动退出了系统。有些 Linux 系统要求只有那些在系统中登记了的程序才能出现在这个字段中。

系统中有一类用户称为伪用户(Psuedo Users),这些用户在/etc/passwd 文件中也占有一条记录,但是不能登录,因为它们的登录 Shell 为空。它们的存在主要是方便系统管理,满足相应的系统进程对文件属主的要求。常见的伪用户如表 3-1 所示。

表 3-1　系统伪用户列表

伪　用　户	含　　义
bin	拥有可执行的用户命令文件
sys	拥有系统文件
adm	拥有账户文件
uucp	UUCP 使用
lp	lp 或 lpd 子系统使用
nobody	NFS 使用拥有账户文件

除了上面列出的伪用户外,还有许多标准的伪用户,如 audit、cron、mail、usenet 等,它们也都各自为相关的进程和文件所需要。

由于/etc/passwd 文件是所有用户都可读的,如果用户的密码太简单或规律比较明显的话,一台普通的计算机就能够很容易地将它破解,因此对安全性要求较高的 Linux 系统都把加密后的口令字分离出来,单独存放在一个文件中,这个文件是/etc/shadow 文件。只有超级用户才拥有该文件读权限,这就保证了用户密码的安全性。

2) /etc/shadow 文件

这个文件中的记录行与/etc/passwd 中的一一对应,由 pwconv 命令根据/etc/passwd 中的数据自动产生。它的文件格式与/etc/passwd 类似,如用命令 cat /etc/shadow 打开此文件,结果显示如下:

```
root:Dnakfw28zf38w:8764:0:168:7:::
daemon:*::0:0::::
bin:*::0:0::::
sys:*::0:0::::
adm:*::0:0::::
```

```
uucp:*::0:0::::
nuucp:*::0:0::::
auth:*::0:0::::
cron:*::0:0::::
listen:*::0:0::::
lp:*::0:0::::
sam:EkdiSECLWPdSa:Array740:0:0::::
```

这些字段如下：

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

(1)“登录名”是与/etc/passwd文件中的登录名相一致的用户账号。

(2)“口令”字段存放的是加密后的用户口令字,长度为13个字符。如果为空,则对应用户没有口令,登录时不需要口令；如果含有不属于集合{./0-ArrayA-Za-z}中的字符,则对应的用户不能登录。

(3)“最后一次修改时间”表示的是从某个时刻起,到用户最后一次修改口令时的天数。时间起点对不同的系统可能不一样。例如,在SCO Linux中,这个时间起点是1970年1月1日。

(4)“最小时间间隔”指的是两次修改口令之间所需的最小天数。

(5)“最大时间间隔”指的是口令保持有效的最大天数。

(6)“警告时间”字段表示的是从系统开始警告用户到用户密码正式失效之间的天数。

(7)“不活动时间”表示的是用户没有登录活动但账号仍能保持有效的最大天数。

(8)“失效时间”字段给出的是一个绝对的天数,如果使用了这个字段,那么就给出相应账号的生存期。期满后,该账号就不再是一个合法的账号,也就不能再用来登录了。

3)用户组的所有信息都存放在/etc/group文件中

将用户分组是Linux系统中对用户进行管理及控制访问权限的一种手段。每个用户都属于某个用户组；一个组中可以有多个用户,一个用户也可以属于不同的组。当一个用户同时是多个组中的成员时,在/etc/passwd文件中记录的是用户所属的主组,也就是登录时所属的默认组,而其他组称为附加组。用户要访问属于附加组的文件时,必须首先使用newgrp命令使自己成为所要访问的组中的成员。用户组的所有信息都存放在/etc/group文件中。用命令cat /etc/group打开此文件的详细信息,结果如下：

```
root::0:root
bin::2:root,bin
sys::3:root,uucp
adm::4:root,adm
daemon::5:root,daemon
lp::7:root,lp
users::20:root,sam
```

这些字段如下：

组名:口令:组标识号:组内用户列表

(1)“组名”是用户组的名称,由字母或数字构成。与/etc/passwd中的登录名一样,组

名不应重复。

(2)“口令”字段存放的是用户组加密后的口令字。一般 Linux 系统的用户组都没有口令,即这个字段一般为空,或是 * 。

(3)“组标识号”与用户标识号类似,也是一个整数,被系统内部用来标识组。

(4)“组内用户列表”是属于这个组的所有用户的列表/b,不同用户之间用逗号(,)分隔。这个用户组可能是用户的主组,也可能是附加组。

3.3.3 软件包管理

常见软件包的种类有 * .rpm、* .Z、* .bz2、* .tar.gz、* .tar.bz2。

1. RPM 软件包管理器

在 Red Hat Linux 下,标准的软件包是通过 RPM 来进行管理的。RPM 的全名是 Red Hat Package Manager,是由 Red Hat 公司开发的软件包管理系统。使用 RPM 软件包管理系统有如下优点:

(1) 安装、升级与删除软件包都很容易。

(2) 查询非常简单。

(3) 能够进行软件包的验证。

(4) 支持源代码形式的软件包。

传统的 Linux 软件包大多是 tar.gz 文件格式,软件包下载后必须经过解压缩和编译操作后才能进行安装,对于一般用户或初级管理员就不是很方便了。

RPM 软件包通常是以 xxx.rpm 的格式命名的。一般,一个标准的 RPM 软件包的名字能够提示一些信息。例如,rhviewer-3.10a-13.i386.rpm,从这样一个名字的 RPM 软件包,可以知道,软件的名称是 rhviewer,版本是 3.10a,次版本是 13,运行的平台是 i386。

RPM 通常有 5 种方式来管理 RPM 软件包:安装、删除、升级、查询和验证。

1) 安装 rpm 包

格式:

```
[root@localhost root]# rpm -ivh rhviewer-3.10a-13.i386.rpm
```

其中,使用到的参数 ivh 说明如下:

i:使用 RPM 的安装模式。

v:在安装的过程中显示安装的信息。

h:在安装的过程中输出 # 号。

另外,RPM 还能够通过 FTP 来进行远程安装,形式其实和本地安装差不多,只要在文件名的前面加上适当的路径就可以了:

```
# rpm -ivh ftp://xxxx/rhviewer-3.10a-13.i386.rpm
```

在安装过程中,可能会经常遇到以下几种情况:

(1) 重复安装软件包。

如果要安装的软件之前已经安装过,就会在安装过程中出现以下错误信息:

```
# rpm -ivh rhviewer-3.10a-13.i386.rpm
```

```
package rhviewer - 3.10a - 13 is already installed
```

如果确定重新安装一次,可以加上--replacepkgs 参数:

```
# rpm - ivh -- replacepkgs rhviewer - 3.10a - 13.i386.rpm
```

(2) 软件包中用到的某个文件已经被其他软件包安装。

这种情况可能最常出现,多个软件包都包含某个或某些文件,当安装了第一个软件包,再安装其他软件包的时候,就会出现以下错误:

```
# rpm - ivh rhviewer - 3.10a - 13.i386.rpm
rhviewer /usr/bin/rhviewer conflicts with file from msviewer - 1.10b - 01
error: rhviewer - 3.10a - 13.i386.RPM cannot be installed
```

此时,可以用--replacefiles 参数:

```
# rpm - ivh -- replacefiles rhviewer - 3.10a - 13.i386.rpm
```

(3) 软件包之间的相关性。

有时,一个软件包的作用要基于另外一个软件包,如果安装该软件包时没有安装需要的另外一个软件包,就会有错误信息:

```
# rpm - ivh rhviewer - 3.10a - 13.i386.rpm
failed dependencies: rhviewer is needed by rhpainter - 2.24 - 20
```

此时,建议先安装这个需要的软件包。不过,如果愿意尝试是否不安装这个需要的软件包也能够正常使用真正要安装的软件,可以加上--nodeps 参数:

```
# rpm - ivh -- nodeps rhviewer - 3.10a - 13.i386.rpm
```

2) 删除 RPM 包

格式:

```
[root@localhost root]# rpm - e rhviewer
```

注意:这里接的不是安装软件包时的名字 rhviewer-3.10a-13.i386.rpm,而只要用 rhviewer 或 rhviewer-3.10a-13 就可以了。建议的方式是先用 RPM 查询要删除的软件,然后用该命令删除。

这里,常出现错误提示,当要删除的软件包被其他软件包关联的时候,就会出现错误提示:

```
# rpm - e rhviewer
removing these packages would break dependencies: rhviewer is neededby rhpainter - 2.24 - 20
```

3) 升级 RPM 包

更新软件包的版本到最新版本,也是经常用到的。

格式:

```
[root@localhost root]# rpm - Uvh rhviewer - 3.10a - 13.i386.rpm
```

升级软件的模式其实是先删除旧软件包,然后再安装新软件包。而且,还可以选择用这种升级的模式安装软件包,因为没有旧软件包的情况下,此升级方式仍然可正常运行。

如果系统中有旧版本存在，可以看到以下信息：

```
# rpm -Uvh rhviewer-3.10a-13.i386.rpm
saving /etc/rhviewer.conf as /etc/rhviewer.conf.rpmsave
```

如果要降低当前版本到更老的版本，一个办法就是删除该版本，然后再重新安装旧的版本，也可以用“--oldpackage”参数来进行“升级”：

```
# rpm -Uvh --oldpackage rhviewer-3.10a-13.i386.rpm
```

补充说明：

有一种升级的安装方式为“更新”。

```
# rpm -Fvh rhviewer-3.10a-13.i386.rpm
```

更新和普通升级的方式是，当系统中没有旧版本时，普通的升级安装仍然会安装该软件，而更新的模式就不会安装。

4）查询 RPM 包

格式：

```
[root@localhost root]# rpm -q rhviewer
rhviewer-3.10a-13
```

如果忘记了要查询的软件名，可以用 rpm -qa 来显示所有已经安装的软件。更详细的软件信息，可以用“rpm -qi”来查询。

2. YUM：RPM 的前端程序，解决包依赖性，可以在各个库中定位软件包

1）YUM 命令的使用

YUM 命令的使用如下：

（1）yum list 查看 YUM 源软件包列表。

（2）yum install [-y] package 安装软件包。

（3）yum remove package 卸载软件包。

（4）yum update 升级安装的软件包。

（5）yum clean all 清除 YUM 产生的临时文件、记录等。

2）配置额外 YUM 库

方法一：在/etc/yum.repos.d 目录下新建.repo 结尾的文件，内容格式如图 3-1 所示。

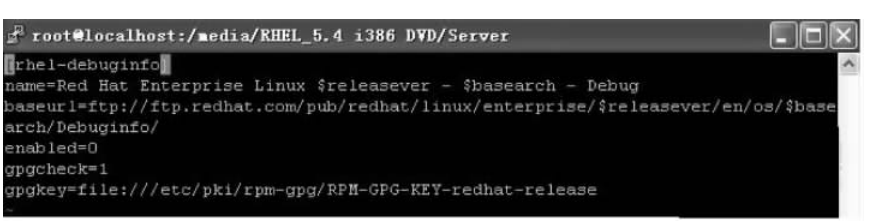

图 3-1　/etc/yum.repos.d 的.repo 文件内容

[repo-name] YUM 源的名字，可以自定义。

Name：yum 源的名字可以随便写，要求和上面括号中的名字相同。

“baseurl：=http：//”为 YUM 源的地址，支持“ftp：//”“http：//”和“file：//”。

Enable=1 启用这个配置文件。

gpgcheck=1 校验密钥。

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release 是指定公钥的位置的,可选;如果不写这句,要运行 rpm -import/etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release。

方法二:直接修改/etc/yum.conf,格式按照上面的来。使用光盘搭建自己的 YUM 源。

3. *.rpm 包的安装

命令 rpm 可以安装软件包,查看已安装包的信息,还可以实现包的卸载命令如下:

(1) 安装包:

```
#rpm -ivh *.rpm
```

(2) 查包的信息:

```
#rpm -pqi *.rpm
```

(3) 查包会向系统何处写入文件:

```
#rpm -pql *.rpm
```

(4) 查系统中所有包:

```
#rpm -qa *.rpm
```

(5) 卸载包:

```
#rpm -e *.rpm
```

4. *.Z 包的安装

```
#compress -d *.Z
```

5. *.bz2 包的安装

```
#bzip2 -d *.bz2
```

6. *.gz 包的安装

```
#gzip -d *.gz
```

7. *tar.gz 包的安装

```
#tar -xzvf *tar.gz          //安装包
#tar -czvf a.tar.gz aaa bbb  //将 aaa 与 bbb 打包成 a.tar.gz
#tar -cjvf b.tar.bz2 aaa bbb //将 aaa 与 bbb 打包成 b.tar.bz2
```

3.3.4 Linux 中的设备文件

Linux 把所有外部设备按其数据交换的特性分为三类,无论哪个类型的设备,Linux 都把它统一作为文件处理。

(1) 字符设备是以字符为单位进行输入输出的设备,如打印机、显示终端。

(2) 块设备是以数据块为单位进行输入输出的设备,如磁带、光盘等。

(3) 网络设备是以数据包为单位进行数据交换的设备,如以太网卡。

把设备看成文件具有以下含义:

(1) 每个设备具有一个文件名称,应用程序可以通过设备的文件名来访问具体的设备,同时要受到文件系统访问权限控制机制的保护。

(2) 设备在内核中应该对应有一个索引节点。

(3) 设备应该可以以文件的方式进行操作。

3.4 任务实施

3.4.1 用户与组账号管理

1. 文本模式查看用户的账号文件

用 cat 命令可以查看账号文件,格式如下:

```
cat /etc/passwd
```

2. 创建用户 david 和 peter

用 useradd 命令创建用户 david 和 peter,并用 passwd 命令为 david 设置密码,格式如下:

```
[root@rhe15 ~]# useradd david
[root@rhe15 ~]# useradd peter
[root@rhe15 ~]# passwd david(设置密码)
```

3. 设置用户账户属性

用 usermod 命令将用户 david 改名为 tom,格式如下:

```
[root@rhe15 ~]# usermod -l tom david
```

4. 将 tom 账户锁定

用 usermod 命令将 tom 用户锁定,参数为 L,格式如下:

```
[root@rhe15D ~]# usermod -L tom
```

5. 删除账户 tom

删除 tom,用命令 userdel,格式如下:

```
[root@rhe15 ~]# userdel -r tom
```

6. 切换用户身份 peter

用 su 命令将 root 用户模式切换到 peter 用户模式下:

```
[root@rhe15~]# su peter
```

7. 查看用户账号信息

用 finger 命令查看用户 peter 信息:

```
[root@rhe15 ~]# finger peter
```

8. 增加一个新的用户组 groupA,groupB

用 groupudd 命令创建不同用户组,格式如下:

```
[root@rhe15 ~]# groupadd groupA
[root@rhe15 ~]# groupadd groupB
```

9. 将 peter 用户添加到群组 groupA

用 useradd 命令加参数 G,实现 Peter 用户加入到组 groupA 中去。

```
[root@rhe15 ~]# useradd -G group peter
```

10. 删除群组 groupA

用 groupdel 命令删除 groupA,格式如下:

```
[root@rhe15 ~]# groupdel groupA
```

3.4.2 设备管理

1. 磁盘限额步骤

(1) 启动 vi 来编辑/etc/fstab 文件。

(2) 把/etc/fstab 文件中的 home 分区添加用户和组的磁盘限额。

(3) 用 quotacheck 命令创建 aquota.user 和 aquota.group 文件 quotacheck -guva。

(4) 给用户 user01 设置磁盘限额功能 edquota -u user01。

(5) 将其 blocks 的 soft 设置为 4000,hard 设置为 5000,inodes 设置为 4000,hard 设置为 5000。

(6) 编辑完成后保存并退出,重新启动系统。

(7) 用 quotaon 命令启用 quota 功能 quotaon -ugva。

(8) 切换到用户 user01 查看自己的磁盘限额及使用情况。

2. U 盘挂载

用 mount 命令挂载 U 盘/dev/sdb 并挂载到/mnt 下,卸载用 umount 命令。

(1) 挂载 U 盘。

```
#mount /dev/sdb /mnt
```

(2) 卸载 U 盘。

```
#umount /mnt
```

3. 光盘挂载

/dev/cdrom 代表光盘,/u01 代表挂载点。

用 mount 命令挂载光盘/dev/cdrom 并挂载到/u01 下:

(1) 光盘挂载。

```
#mount /dev/cdrom /u01
```

(2) 光盘卸载。

```
#umount /u01
```

3.4.3 系统信息命令的使用

1. 系统信息查看

(1) 查看内核/操作系统/CPU 信息。

```
# uname -a
```

(2) 查看操作系统版本。

```
# head -n 1 /etc/issue
```

(3) 查看 CPU 信息。

```
# cat /proc/cpuinfo
```

(4) 查看计算机名。

```
# hostname
```

(5) 列出所有 PCI 设备。

```
# lspci -tv
```

(6) 列出所有 USB 设备。

```
# lsusb -tv
```

(7) 列出加载的内核模。

```
# lsmod 块
```

(8) 查看环境变量。

```
# env
```

2. 资源信息的查看

(1) 查看内存使用量和交换区使用量。

```
# free -m
```

(2) 查看各分区使用情况。

```
# df -h
```

(3) 查看指定目录的大小。

```
# du -sh <目录名>
```

(4) 查看内存总量。

```
# grep MemTotal /proc/meminfo
```

(5) 查看空闲内存量。

```
# grep MemFree /proc/meminfo
```

(6) 查看系统运行时间、用户数、负载。

```
# uptime
```

(7) 查看系统负载。

```
# cat /proc/loadavg
```

3. 磁盘和分区信息的查看

(1) 查看挂接的分区状态。

```
# mount|column -t
```

(2) 查看所有分区。

```
# fdisk -l
```

(3) 查看所有交换分区。

```
# swapon -s
```

(4) 查看磁盘参数(仅适用于 IDE 设备)。

```
# hdparm -i /dev/hda
```

(5) 查看启动时 IDE 设备检测状况。

```
# dmesg|grep IDE
```

4. 用户信息的查看

(1) 查看活动用户。

```
# w
```

(2) 查看指定用户信息。

```
# id <用户名>
```

(3) 查看用户登录日志。

```
# last
```

(4) 查看系统所有用户。

```
# cut -d: -f1 /etc/passwd
```

(5) 查看系统所有组。

```
# cut -d: -f1 /etc/group
```

(6) 查看当前用户的计划任务。

```
# crontab -l
```

3.4.4 软件包管理

1. 创建 TAR 包

将/etc 下所有文件打包，并形成 etc.tar 归档文件，如下：

```
[root@rhel5～]#tar -cvf etc.tar /etc
```

2. 创建压缩 TAR 包

将/etc 下所有文件打包并压缩，并形成 etc.tar.gz 文件，格式如下：

```
[root@rhel5～]#tar -zcvf etc.tar.gz /etc
```

3. 查询 TAR 包中的文件列表

用参数 tvf 可以查询包的文件列表，格式如下：

```
[root@rhel5～]#tar -tvf etc.tar [root@rhel4 ～]#tar -tvf etc.tar.gz
```

4. 还原 TAR 包

用 tar 命令加参数 xvf 可以解开打包文件：

```
[root@rhel5～]#tar -xvf etc.tar
[root@rhel5～]#tar -zvf etc.tar.gz
```

gzip 也可以解压文件，格式如下：

```
[root@rhel5～]#gzip -d etc.tar.gz
```

5. 查看所有安装的软件包

用 rpm 命令可以查询所安装的软件包，格式如下：

```
[root@rhel5～]# rpm -qa
```

3.5 习题与实训

3.5.1 思考与习题

简答题

(1) Linux 系统是如何标识用户和组的？

(2) 举例说明如何创建一个用户账户。

3.5.2 实训

1. 实训目的

(1) 掌握为 root 用户修改密码的方法。

(2) 掌握创建新用户的方法。

(3) 掌握用户组的管理方法。

(4) 掌握为用户授权的方法。

2. 实训任务背景

季目开关制造公司现有行政人员 6 人,每人配备一台电脑用于日常办公,网络环境中存在多种操作系统。运营过程中,由于公司各部门文件权限没有严格限制,造成公司的研发资料以及行政部门的行政管理资料在公司内部随意传播访问,并且人力资源的员工名单及联系方式还被传给了猎头公司,造成公司业务及人员的大量流失,公司受到严重威胁。现对公司的用户与组的权限进行重新设计,规划如下:

研发部开发人员 David 和 Peter 属于组 A;

行政部人员 Jack 和 Mike 属于组 B;

建立共享目录"/project_a",该目录里面的文件只能由研发部开发人员读取、增加、删除、修改以及执行,其他用户不能对该目录进行任何的访问操作;

建立目录"/project_b",该目录里面的文件只能由行政部人员读取、增加、删除、修改以及执行,其他用户不能对该目录进行任何的访问操作;

建立目录"/project",该目录里面的文件可由研发部、行政部人员读取、增加、删除、修改以及执行,其他部门用户只可以对该目录进行只读的访问操作。

3. 实训内容

(1) 创建用户与组。

创建组

```
groupadd a
groupadd b
```

(2) 创建用户,并加入到指定组。

```
useradd -g  a david
useradd -g  a peter
useradd -g  b jack
useradd -g  b mike
```

(3) 为用户设置密码。

```
passwd david
passwd peter
passwd jack
passwd mike
```

(4) 创建文件夹,并设计权限。

① 创建文件夹。

```
mkdir  /project_a
mkdir  /project_b
```

② 创建相应文件。

```
touch /project_a/file_a
touch /project_b/file_b
```

③ 修改文件夹的属组。

```
chown  -R  :a  /project_a
chown  -R  :b  /project_b
```

④ 修改文件夹的权限。

```
chmod  -R  770  /project_a
chmod  -R  770  /project_b
```

4. 实训总结

提交实训报告。