RHCE 实验手册

目录

第一部分：

Red Hat 系统管理 I (RH124)

登录和使用基本的 Linux 命令
获取对使用命令的帮助
组织目录和文件
搜索和处理文件
标准输入输出和管道
权限的设置和更改
Linux 文件系统
配置 Bash Shell

实验 1：登录和使用基本的 Linux 命令

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。有另外一个无特权用户 student，密码 student 的账户存在。

实验目标：

熟悉基本命令，使用基本的命令执行基本的 linux 任务和命令。

实验背景：

首先，练习使用 Linux 命令来登录、改变口令、转换成超级用户、使用 cat 命令来查看文件、使用 nano 命令来改变文件。

实验要求：

1、登录和使用基本的 linux 命令
2、修改密码，转换成超级用户
3、查看和修改文件

实验详解：

1、按 Ctrl+Alt+F2 切换到虚拟控制台（tty2）：

Red Hat Enterprise Linux Server release 6.0 (Santiago)
Kernel 2.6.32-71.e16.x86_64 on an x86_64

SA2
desktopX login:

2、登录为用户 student：在 login：这个提示下输入 student，然后按回车；在 Password：这个提示后输入 student 的口令 。默认情况下口令为 student：

desktopX login:**student**
Password:
[student@desktopX ~]$
注意：口令在你输入的时候是不会出现在屏幕上的。

3、使用 **passwd** 来设置口令。**passwd** 命令会首先向你询问当前的口令。输入当前密码（和登录时一样，口令在你输入时不会出现在屏幕上）：

[student@desktopX ~]$
Changing password for user student.
Changing password for student
(current) UNIX password:

4、**passwd** 命令会检查你输入的口令的强度以确保它达到一定的难猜程度。试着输入一个坏口令来测试这一功能：把口令设置成你的用户名 student：

New UNIX password:
BAD PASSWORD: it is based on your username
New UNIX password:
注意：口令被拒绝。你会被提示输入一个好一点的口令。

5、再试一次。这次设置一个复杂的口令。混合使用大小写字母、数字和标点。至少使用八个字符。你会被提示把口令再输入一次。如果你选择的口令足够强健，并且两次输入的口令相同，口令就会被成功改变，你就会看到这样的输出：

New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
注意：如果你的口令被拒绝了，就继续尝试，知道成功为止。

6、运行 **exit** 命令来注销：
[student@desktopX ~]$ **exit**

使用你的新口令重新登录
desktopX login: **student**
Password:
[student@desktopX ~]$

7、现在你就登录为用户 student 了，这是一个不具有特权的用户。在本次实验的后半部，你需要超级用户的特权来运行命令。因此，首先使用 **su** 命令编程超级用户，在提示输入的口令的时候输入 redhat：
[student@desktopX ~]$ **su –**
Password:
[root@desktopX ~]#
注意：在使用 su 命令是使用了"-"这个参数。加了这个减号的目的是使环境变量和欲转换的用户相同，不加是取得用户的临时权限。

注意命令提示符的变化：所显示的用户名现在是 root，提示后的最后一个字符是一个#而不是$。这两个外观上的变化表明你现在已有超级用户特权了。从现在起，直到你从超级用户 shell 退出，你所运行的命令都是会带有完全的特权。

8、使用 **passwd** 命令把 student 帐号的口令改为 student：
[root@desktopX ~]# **passwd student**
Changing password for user student.
New UNIX password:
BAD PASSWORD: it is based on a directory word
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
思考：为什么一定要是根用户才能这么做？
9、使用 cat 命令查看/etc/issue：
[root@desktopX ~]# **cat /etc/issue**
Red Hat Enterprise Linux Server release 6.0 (Santigo)
Kernel \r on an \m
[root@desktopX ~]#

/etc/issue 文件在登录提示前显示。默认的内容如上所示（根据版本和配置的不同，内容会略有出入）。常添加的信息是警告未经授权用户的"免入系统"。

10、使用 **nano** 命令编辑/etc/issue：
[root@desktopX ~]# **nano /etc/issue**
*注意：当你开始打字时，文本在闪烁的光标处被输入。在页面底部是一个命令菜单。^ 符号代表你需要同时按住 Ctrl 键。例如，Ctrl+X 会导致 **nano** 退出。*

11、在/etc/issue 顶端新添加一行，使欢迎信息更为有好：
Welcome !

按 Ctrl+X 来保存改变。nano 命令会向你询问是否要保存改变（“保存被修改的缓存”）。输入 y 来保存改变。

nano 命令会建议你保存到/etc/issue 这个文件，这正是我们打算做的。按 Enter 键来确认，保存文件。

12、再查看一下该文件，你会看到：

[root@desktopX ~]# **cat /etc/issue**
Red Hat Enterprise Linux Server release 6.0 (Santigo)
Kernel \r on an \m
[root@desktopX ~]#

现在你就可以查看所做改变的效果。但是对/etc/issue 文件的改变直到下次登陆时才会生效。强迫登陆提示重置的最快方法是转换到 Ctrl+Alt+F2 到 Ctrl+Alt+F6 这些控制台，然后在每个提示后逐一按 Ctrl+D，这会导致登陆提示终止并重新启动，重新读取/etc/issue 文件，显示其中的新内容。

13、清除：

按 Ctrl+Alt+F2 返回刚才登陆的虚拟控制台。

输入 **exit** 来推出超级用户。

注意：提示符的变化。

再输入 **exit** 来注销。登录界面就会返回，包括你新改变的欢迎信息。

按 Ctrl+Alt+F7 来返回图形化界面。

本次实验到此结束。

实验 2：获取对使用命令的帮助

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。有另外一个无特权用户 student，密码 student 的账户存在。

实验目标：

熟悉手头可用的资源，以便回答有关红帽企业版 Linux 命令的问题。

实验背景：

你想再次修改/etc/issue 文件（参见上一次实验）。这一次，你想查看一下有没有能够显示系统主机名的转义符。

实验要求：

1、使用帮助工具

2、使用 man 命令来解决问题

实验详解：

1、使用口令 student 登录为用户 student。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、首先参考 **man** 说明书中的/etc/issus 项目：

[student@desktopX ~]$ **man issue**

注意：其中说明转义符要根据 mingetty 而定。还请注意 See Also 部分让你参考 mingetty(8)。

3、跳到有关转义符(escape)的部分，输入：

/escape

你的光标应该位于标题 ISSUE ESCAPE 上。如果不是，按 **n** 来把光标移动到这个标题上。

4、切换到 root 用户，在 **nano** 中打开/etc/issue 这个文件：

[student@desktopX ~]$ **su -**

[root@desktopX ~]# **nano /etc/issue**

5、修改你前面添加的欢迎行，这一次使用转义符来包括主机名：

/etc/issue 文件现在应该类似：

Welcome to \n!

Red Hat Enterprise Linux Server release 6.0 (Santigo)

Kernel \r on an \m

6、按 Ctrl+x 来保存文件，退出 **nano**：

Kernel \r on an \m

[root@desktopX ~]#

7、使用 **exit** 命令关闭 shell 并注销。现在你就应该能够看到新修改的登录提示了！访问其他虚拟终端来查看这些内容的变化。记住，你需要按 Enter 键才能看到更新的内容。

本次实验到此结束。

✧ man 主要命令参考手册

参数	功能
-c *KEYWORD*	显示使用 cat 命令的手册信息
-f *KEYWORD*	显示在关键字数据库中仅与作为最终参数给定的命令名相关的项
-k *KEYWORD*	显示关键字数据库中包含与作为最终参数给定的字符匹配的标题的字符串的每一行
-K *KEYWORD*	显示关键字数据库中包含与作为最终参数给定的字符匹配的标题的字符串的每一行，然后询问你是否想阅读

实验 3：搜索和处理文件

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。有另外一个无特权用户 student，密码 student 的账户存在。

实验目标：

培养对 find 工具的更深理解，掌握如何使用 find 工具搜索并处理文件。

实验背景：

你在使用 linux 时遇到了一些问题，这时候你可能需要参考 find 命令的说明书页。

当 find 命令试图进入你没有读取权限的目录时，你会看到许多”Permission denied（权限不够）”的消息，此时你不必在乎这些消息，在 find 命令后面添加 2> /dev/null 会隐藏它们。

实验要求：

使用 find 获取帮助

实验详解：

1、使用口令 student 登录为用户 student。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、列举所有/var/lib 目录下的所属能用户为 rpm 的文件：

[student@desktopX ~]$ **find /var/lib -user rpm 2> /dev/null**

3、列举所有/var 目录下的所属用户为 root、所属组群为 mail 的文件：

[student@desktopX ~]$ **find /var/ -user -group mail 2> /dev/null**

4、生成一个 ls –l 格式的列表，列举系统上所属用户不是 root、bin 或 student 的所有文件：

[student@desktopX ~]$ **find / -not -user root -not -user bin -not -user student -ls 2> /dev/null**

或者

[student@desktopX ~]$ **find / ! -user root ! -user bin ! -user student -ls 2 > /dev/null**

注意：这会花很长的一段时间，因此建议大家在一个终端上运行这条命令，在另一个终端上做下一道题目。

5、生成一个 ls –l 格式的列表，列举/usr/bin 目录下所有大于 50KB 的文件：

[student@desktopX ~]$ **find /usr/bin -size +50k -ls 2> /dev/null**

6、对/etc/mail 目录下的所有文件执行 file 命令：

[student@desktopX ~]$ **find /etc/mail -exec file {} \ ; 2> /dev/null**

7、生成一个 ls -l 格式的列表，列举/etc 目录下的所有符号链接：

[student@desktopX ~]$ **find /etc/ -type | -ls 2> /dev/null**

8、生成一个 ls -l 格式的列表，列举/tmp 目录下的所属用户为 student、修改时间早于 120 分钟前的所有“常规”文件：

[student@desktopX ~]$ **find /tmp -user student -mmin +120 -type f -ls 2> /dev/null**

9、列举/bin 和/usr/bin 目录下所有启用了 SetUID 权限位的文件：
[student@desktopX ~]$ **find /bin /usr/bin -perm -4000 2> /dev/null**
或者
[student@desktopX ~]$ **find /bin /usr/bin -perm -u+s 2> /dev/null**

本次实验到此结束。

实验 4：组织目录和文件

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。有另外一个无特权用户 student，密码 student 的账户存在。

实验目标：

熟悉几个基本的操作系统文件和目录的命令的功能、语法和用法，整理出一个更有条理的主目录，每个文件都位于恰当的子目录。

实验背景：

你的主目录中已经积压了一些文件，你决定开始规整它们。你打算新建几个子目录，然后复制或转移文件来适应这个新方案。此外，你还有一些不需要的文件，它们必须被删除。

实验要求：

1、使用 ls 查看文件及其属性
2、使用 cd 切换路径
3、使用 touch、mkdir 创建相应的文件及文件夹
4、使用 rm、rmdir 删除文件及文件夹

实验详解：

1、使用口令 student 登录为用户 student。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、登录到系统之后，你应该就位于你的主目录中。使用”pwd”（打印工作目录）命令来校验：

```
[student@desktopX ~]$ pwd
/home/student
```

3、使用一下每个命令来检查你的主目录中的文件：

ls
ls –a
ls -al

观察返回的文件数量有何不同。

4、现在，你将使用 **touch** 命令来创建该练习序列所欲的文件，输入：

```
[student@desktopX ~]$ touch {report,graph}_{jan,feb,mar}
```

5、使用 **ls** 命令来检查前一个命令执行的结果。你应该看到你的主目录中新建了一下六个空文件：

```
[student@desktopX ~]$ ls
graph_feb  graph_jan      graph_mar      report_feb      report_jan      report_mar
```

6、为了组织文件，你必须首先新建一些目录。使用 mkdir 命令来新建目录。在更改目录时，请确定当前工作目录和预料中一样：

```
[student@desktopX ~]$ mkdir Projects
[student@desktopX ~]$ mkdir Projects/graphs
```

[student@desktopX ~]$ **cd Projects**
[student@desktopX Projects]$ **cd Projects**
[student@desktopX Projects]$ **mkdir reports**
[student@desktopX Projects]$ **cd reports**
[student@desktopX reports]$ **mkdir ../Backups**

使用 **ls** 命令来检查你的工作结果：
[student@desktopX reports]$ **cd**
[student@desktopX ~]$ **ls -l**
totol 4

-rw-rw-r--	1	student student	0	Sep	30	21:08	graph_feb
-rw-rw-r--	1	student student	0	Sep	30	21:08	graph_jan
-rw-rw-r--	1	student student	0	Sep	30	21:08	graph_mar
drw-rw-r-x	5	student student	4096	Sep	30	21:08	Projects
-rw-rw-r--	1	student student	0	Sep	30	21:08	report_feb
-rw-rw-r--	1	student student	0	Sep	30	21:08	report_jan
-rw-rw-r--	1	student student	0	Sep	30	21:08	report_mar

[student@desktopX ~]$ **ls Projects**
Backups graphs reports

7、首先，把所有文件名中带有 graph 的文件都转移到 Projects 目录中的 graphs 子目录中。分两个步骤来完成：
第一步：转移一个文件
第二步：转移两个文件

[student@desktopX ~]$ **mv graph_jan Projects/graphs**
[student@desktopX ~]$ **mv graph_feb graph_mar Projects/graphs**
[student@desktopX ~]$ **ls -l Projects/graphs**
totol 3

-rw-rw-r--	1	student student	0	Sep	30	21:08	graph_feb
-rw-rw-r--	1	student student	0	Sep	30	21:08	graph_jan
-rw-rw-r--	1	student student	0	Sep	30	21:08	graph_mar

8、下一步，把两个”report”文件移动到 Projects 目录中的 reports 子目录中。使用这些命令来转移这些文件：
[student@desktopX ~]$ **mv report_jan report_feb Projects/reports**
[student@desktopX ~]$ **ls –l Projects/reports**
totol 2

-rw-rw-r--	1	student student	0	Sep	30	21:08	report_feb
-rw-rw-r--	1	student student	0	Sep	30	21:08	report_jan

9、删除剩下的 report 文件：
[student@desktopX ~]$ **rm report_mar**
[student@desktopX ~]$ **ls**
Projects

10、转换到 Backups 的目录，把 January 文件复制到这个目录中。一个文件使用绝对路径名复制，另一个文件使用相对路径名复制：
[student@desktopX ~]$ **cd Projects/Backups**
[student@desktopX Backups]$ **pwd**
/home/student/Projects/Backups
[student@desktopX Backups]$ **cp ../reports/report_jan .**
[student@desktopX Backups]$ **cp /home/student/Projects/graphs/graph .**

//'.'表示当前工作的目录

```
[student@desktopX Backups]$ ls –l
total 2
-rw-rw-r--  1  student student  0  Sep  30  21:08  graph_jan
-rw-rw-r--  1  student student  0  Sep  30  21:08  report_jan
```

11、注销，或运行 **exit** 命令来关闭图形化终端。

本次实验到此结束。

实验 5：标准输入输出和管道

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。有另外一个无特权用户 student，密码 student 的账户存在。

实验目标：

熟悉 Red Hat Linux 中的标准输入输出和管道和常用编译器；使用输入输出流创建和修改文本文件；了解管道技术的使用。

实验背景：

在日常使用中，每当我们配置服务时，往往要对系统的配置文件进行修改。在虚拟控制台或者或者服务器中，我们是没有图形化界面的，也没有图形化软件工具可供选择。那怎么办？其实在这种情况下我们可以选择使用 linux 标准的输入输出来解决这个问题。

实验要求：

1、使用 cat 输入输出

2、使用 tr 进行替代输入

3、使用'I'管道

实验详解：

1、使用口令 student 登录为用户 student。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、使用 **cat** 命令在用户主目录下创建两个文本文件，分别命名为 packages1.txt 和 packages2.txt：

```
[student@desktopX ~]$ cat > packages1.txt
amanda
galleon
metacity
mozilla
postgresql
procinfo
rpmfind
squid
^d(Ctrl+D)
[student@desktopX ~]$ cat > packages2.txt
anaconda
openssh
gnome-core
samba
sendmail
xscreensaver
^d(Ctrl+D)
```

注意：如果 cat 后没有参数，则它会等待标准的输入，所以当你输入 cat 命令后，再回车，然后什么也没有显示。输后，cat 命令会监视标准输入，等待输入的到达。如果这个时候输入一些文本，再按回车，cat 就会把输入的内容当的输入，然后输出到标准的输出——显示器上，结束 cat 的命令为按下 ctrl-d，这是结束输入的标志。

$ cat

输入一些文字，然后按回车。

^d (就是按 ctrl-d)

3、cat 工具是最简单的 linux 过滤器，它会默认把跟在后面的参数当作文件名，并把这个文件作为输入：

[student@desktopX ~]$ **cat packages1.txt**
amanda
galleon
metacity
mozilla
postgresql
procinfo
rpmfind
squid

4、使用'**>**'命令 package1.txt 的内容复制到 packages1.catfile，然后用 diff 和 ls 确认原文：

[student@desktopX ~]$ **cat packages1.txt > packages1.catfile**
[student@desktopX ~]$ **cat packages1.catfile**
[student@desktopX ~]$ **diff packages1.txt packages1.catfile**
[student@desktopX ~]$ **ls –l packages1***
……

5、使用'**>>**'来重定向会把输出，把 packages2.txt 文件中的内容附加到 packages1.catfile 已存在的文件的末尾：

[student@desktopX ~]$ **cat packages2.txt >> packages1.catfile**
[student@desktopX ~]$ **cat packages1.catfile**
amanda
galleon
metacity
mozilla
postgresql
procinfo
rpmfind
squid
anaconda
openssh
gnome-core
samba
sendmail
xscreensaver

6、使用 tr 创建一个新文件，并替换其中部分的内容：

[student@desktopX ~]$ **tr 'aeiou' 'AEIOU' > trfile.txt**
This time, when text is typed at the keyboard
^d
[student@desktopX ~]$ **cat trfile.txt**
ThIs tImE, whEn tExt Is typEd At thE kEybOArd

7、使用 **set -o** 命令，查看 noclobber(文件复写)选项：

[student@desktopX ~]$ **set -o**
……
noclobber off
……

[student@desktopX ~]$ **echo “new contect” > trfile**
[student@desktopX ~]$ **cat trifle**
new contect

8、使用 set 命令更改 noclobber 选项，关闭文件复写功能：
[student@desktopX ~]$ **set –o noclobber**
[student@desktopX ~]$ **echo “new contents” > trfile.txt**
bash: trfile.txt：cannot overwrite existing file
[student@desktopX ~]$

9、使用管道技术提取试验中有关 cat 的命令，并保存于名为 cat1 的文件中：
[student@desktopX ~]$ **history**
cat > package1.txt
cat > package2.txt
……
[student@desktopX ~]$ **history | grep cat > cat1**
[student@desktopX ~]$ **cat cat1**
cat > package1.txt
cat > package2.txt
……

本次实验到此结束。

实验 6：权限的设置和更改

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。有另外一个无特权用户 student，密码 student 的账户存在，还有一个无特权的用户 visitor，密码 password 的账户存在。

实验目标：

了解 linux 的安全模型，熟悉用户账号的组群帐号，设置并修改文件权限。

实验背景：

在 Linux 系统中，各个用户（除超级用户外）的空间之间是隔离的，一个用户是不能进入其他用户的注册空间的。通常普通用户可以通过超级用户在/usr 目录下建立共享的用户目录，然后将该共享目录的属主权移交给普通用户。普通用户可将该目录可设置成允许同组用户访问，也可以设置成允许所有用户访问。并在该目录中提供允许共享的文件。

实验要求：

1、转换用户帐号，查看 id 值
2、使用 chown 修改文件属主
3、使用 chmod 采用两种不同方法修改文件权限
4、使用 umask 修改默认的权限

实验详解：

1、使用口令 password 登录为用户 visitor。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、登录到系统之后，你应该就位于你的主目录中。使用”id”命令来查看你的相关信息：

```
[student@desktopX ~]$ id
uid=501(visitor) gid=501(visitor) group=501(visitor)
……
[student@desktopX ~]$ pwd
/home/visitor
```

3、运行 **su – student** 命令来转换成用户 student，在运行上一步：

```
[student@desktopX ~]$ id
uid=500(visitor) gid=500(visitor) group=500(visitor)
……
[student@desktopX ~]$ pwd
/home/student
```

思考：为什么我们要用 su -而不直接用 su 。

4、现在切换到 root 用户，在/usr 中创建两个文件夹，分别为 student 和 visitor：

```
[student@desktopX ~]$ su -
Password:
[root@desktopX ~]# cd /usr
[root@desktopX usr]# mkdir student visitor
[root@desktopX usr]# ls –ld student visitor
drwxrw-r-- 2    root    root    4096    Sep    30    21:08    student
drwxrw-r-- 2    root    root    4096    Sep    30    21:08    visitor
```

5、使用 **chown** 命令来修改这两个文件夹的属主：

[root@desktopX usr]# **chown student:student student**

[root@desktopX usr]# **chown visitor:visitor visitor**

6、回到 student 文件夹，用 student 账户创建两个文本文件，分别为 tf1 和 tf2，并赋予不同的权限：

[student@desktopX student]$ **cat >tf1**

This file is for all user

[student@desktopX student]$ **cat >tf2**

This file is for the users in the same groups

[student@desktopX student]$ **ls –l tf1 tf2**

-rw-rw-r-- 1 student student 0 Sep 30 21:08 tf1

-rw-rw-r-- 1 student student 0 Sep 30 21:08 tf2

[student@desktopX student]$ **chmod 750 tf2**

[student@desktopX student]$ **ls –l tf1 tf2**

-rw-rw-r-- 1 student student 0 Sep 30 21:08 tf1

-rwxrw---- 1 student student 0 Sep 30 21:08 tf2

7、切换到 visitor 用户，分别访问刚才创建的两个文件 tf1 和 tf2：

[student@desktopX student]$ **su visitor**

Password:

[visitor@desktopX student]$ **cat tf1**

This file is for all user

[visitor@desktopX student]$ **cat tf2**

cat: tf2: Permission denied

8、切换到 root 用户，使用 **chmod** 命令增加 visitor 的访问权限：

[visitor@desktopX student]$ **su**

[root@ desktopX student]# **chmod o+r tf2**

9、回到 visitor 用户，继续访问 tf2：

[visitor@desktopX student]$ **cat tf2**

This file is for the users in the same groups

10、使用 umask 修改文件的默认权限：

[student@desktopX student]$ **umask**

0002

[student@desktopX student]$ **mkdir dir1**

[student@desktopX student]$ **umask –p 007**

[student@desktopX student]$ **mkdir dir2**

[student@desktopX student]$ **ls –ld dir1 dir2**

drwxrwxr-- 2 student student 4096 Sep 30 21:08 dir1

drwxrwx--- 2 student student 4096 Sep 30 21:08 dir2

这时候如果切换到 visitor 用户访问 dir2 将会提示权限不够

[visitor@desktopX student]$ **cd dir2**

dir2: .: Permission denied

本次实验到此结束。

实验 7：Linux 文件系统

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。有另外一个无特权用户 student，密码 student 的账户存在。

实验目标：

更好的理解 Linux 文件系统基础，包括创建和使用链接；使用 locate 和 find 命令查找文件；归档和压缩文件。

实验背景：

每次启动的时候，你的系统的主硬盘驱动器都开始发出讨厌的噪音。你怀疑硬盘可能要寿终正了，其中的人数据也要跟着陪葬了。由于你之前没有进行过数据备份的操作，所以你决定手工备份几个至关重要的文件。假设/tmp 目录所在的分区位于另一个驱动器，因此你决定暂时把备份存放在那里。

实验要求：

1、使用 ln 分别创建软连接、硬链接

2、使用 dh 查看磁盘用量

3、使用 tar、gzip、bzip2 备份配置文件

实验详解：

1、使用口令 student 登录为用户 student。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、使用 **cp** 命令把 usr/share/dict/word 文件复制到你的主目录中：

[student@desktopX ~]$ **cp /usr/share/dict/words .**

注意：此处'.'表示当前目录。

3、查看/usr/share/dict/words 的相关信息：

[student@desktopX ~]$ **ls /usr/share/dict/words**

-rw-r--r-- 1 root root 409305 Sep 30 21:08 linux.words

lrwxrwxrwx1 root root 11 Sep 30 21:08 word -> linux.word

这里的文件 word 是一个符号链接：文件模式的第一个字符是代表符号链接的'l'；并且文件名包括了显示链接目标的"-> linux.word"。

4、在主目录中创建一个符号链接和一个硬链接，都指向你的主目录中的 words 文件：

[student@desktopX ~]$ **ln -s word soft**

[student@desktopX ~]$ **ln word hard**

5、测试一下新建的连接是否正确地指向 words 中的数据，我们使用 head 命令显示文件中的前 10 行：

[student@desktopX ~]$ **head hard soft**

==> hard <==

……

==> soft <==

……

我们可以看到，两者输出相同，就说明我们的链接创建正确。

6、详细查看两个文件的相关信息，比较两种链接的区别：

```
[student@desktopX ~]$ ls -il hard soft
84040 -rw-r--r-- 2 student student 4950996 Aug 22 14:43 hard
84021 lrwxrwxrwx 1 student student       5 Aug 22 15:18 soft -> words
[student@desktopX ~]$ stat hard soft
  File: `hard'
  Size: 4950996     Blocks: 9712       IO Block: 4096   regular file
Device: fd01h/64769d    Inode: 84040       Links: 2
Access: (0644/-rw-r--r--)  Uid: (  500/ student)   Gid: (  500/ student)
Access: 2011-08-22 15:22:48.000000000 +0800
Modify: 2011-08-22 14:43:10.000000000 +0800
Change: 2011-08-22 15:17:55.000000000 +0800
  File: `soft' -> `words'
  Size: 5           Blocks: 2          IO Block: 4096   symbolic link
Device: fd01h/64769d    Inode: 84021       Links: 1
Access: (0777/lrwxrwxrwx)  Uid: (  500/ student)   Gid: (  500/ student)
Access: 2011-08-22 15:36:42.000000000 +0800
Modify: 2011-08-22 15:18:35.000000000 +0800
Change: 2011-08-22 15:18:35.000000000 +0800
```

7、使用 df 命令来判断每个文件系统上的空余空间总量：

```
[student@desktopX ~]$ dh
……
[student@desktopX ~]$ dh -h
……
[student@desktopX ~]$ dh -H
……
```

比较这三者输出的差别。

8、使用 **tar** 命令把/etc 的内容打包，保存在/tmp 中：

```
[student@desktopX ~]$ su
[student@desktopX ~]$ tar -cvf /tmp/confbackup.tar /etc
……
```

9、查看压缩文件的属性，特别注意 tar 包的大小：

```
[student@desktopX ~]$ ls -lh /tmp/confbackup.tar
……
```

10、使用 **gzip** 命令来压缩归档文件，注意这个新文件的大小：

```
[student@desktopX ~]$ cd /tmp
[student@desktopX tmp]$ gzip -v confbackup.tar
[student@desktopX tmp]$ ls -lh confbackup.tar.gz
……
```

11、给文件解压，用 **bzip2** 重新压缩，比较压缩文件的大小：

```
[student@desktopX tmp]$ gunzip confbackup.tar.gz
[student@desktopX tmp]$ bzip2 -v confbackup.tar
[student@desktopX tmp]$ ls -lh confbackup.tar.bz2
```

12、注销，清除。

本次实验到此结束。

实验 8：配置 Bash Shell

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。有另外一个无特权用户 student，密码 student 的账户存在。

实验目标：

逐渐掌握更多定制 Bash shell 的操作的技能，包括创建等值的别名，修改 bash 提示以及简单的 shell 编程。

实验背景：

一般我们在 linux 下直接用命令行直接进行操作的。这种方法很快捷很方便，一旦大批量的操作时单纯的命令行操作就显得不足。我们可以采用批处理的方法帮助我们搞定这样的难题。

实验要求：

1、使用 alias 创建别名
2、使用 PS1 修改 bash 的提示符
3、编写简单的 shell 小程序

实验详解：

1、使用口令 student 登录为用户 student。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、使用 **alias** 命令创建一个别名，当你输入 c 后系统会运行 clear 命令来清除屏幕：
[student@desktopX ~]$ **alias c='clear'**
[student@desktopX ~]$ **alias**
[student@desktopX ~]$ **c**
注意： 我们刚才创建的别名'c'并没有保存，注销后就别名就会丢失。要使这个别名在用户 student 每次登录都能使用，需要保存。

3、修改.bashrc，保存创建的别名：
[student@desktopX ~]$ **vim .bashrc**
……
找到包含以下文本的那一行
#User specific aliases and functions
把你的别名直接添加到那一行下面
alias c='clear'
保存文件退出
：wq

4、注销后重新登录终端，输入以下命令测试别名是否保存：
[student@desktopX ~]$ **alias**
[student@desktopX ~]$ **c**

观察屏幕是否被清除。

5、使用 PS1 命令改变 bash 提示：
[student@desktopX ~]$ **PS1='Red Hat -> '**

Red Hat -> **ls**

6、修改 bash 恢复为主机名和传统的美元符号：
Red Hat -> **PS1='\h $'**
desktopX $

7、在 bash 中添加历史提示：
desktopX $ **PS1='[\u@\h \w (\!)]$ '**
[student@desktopX ~ (21)]$

8、编写一个 shell 脚本，当你输入 yes 时它会返回 no：
[student@desktopX ~ (21)]$ **cat >test_script**

```
#! /bin/bash
#if you enter yes it will echo no ,
#if you enter no it will echo yes .

echo "You want YES or NO： "
read ANSWER

if [ "$ANSWER" = "YES" ] || [ "$ANSWER" = "yes" ]
then
            echo "Your idea is NO";
 elif [ "$ANSWER" = "NO" ] || [ "$ANSWER" = "no" ]; then
            echo "Your idea is YES";
 else
          echo "you are wrong" ;
 fi
```

9、运行 shell 脚本 test_script：
[student@desktopX ~ (21)]$ **./test_script**
You want YES or NO：
YES
Your idea is NO

10、注销，清除。

本次实验到此结束。

第二部分：

Red Hat 系统管理 II (RH135)

RHEL6 启动管理
安装和管理软件
配置 Linux 内核
Linux 进程管理
用户和组管理
管理网络设定
计划任务 Crontab
Linux 文件系统管理
配置 LVM 逻辑卷管理器
独立磁盘冗余阵列 RAID
搭建目录服务器
磁盘配额 QUOTA

实验 9：RHEL6 启动管理

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

熟悉 Red Hat Enterprise Linux 启动过程，了解 grub 配置及运行级别。

实验背景：

你的台式机原来是用于家用的，开机时默认采用图形化界面的登录方式。现在你想把它的模式改为用于服务器的完全多用户级别。这时候我们就需要修改运行级别。

实验要求：

1、查看 grub，熟悉启动项相关事宜

2、修改运行级别

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、使用 **vim** 编辑器查看自己的 grub 配置文件：

```
[root@desktopX ~]$ vim /boot/grub/grub.conf
……
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.18-164.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-164.el5 ro root=/dev/vol0/root rhgb quiet
    initrd /initrd-2.6.18-164.el5.img
title Red Hat Enterprise Linux Server (2.6.18-164.el5xen)
    root (hd0,0)
    kernel /xen.gz-2.6.18-164.el5
    module /vmlinuz-2.6.18-164.el5xen ro root=/dev/vol0/root rhgb quiet
    module /initrd-2.6.18-164.el5xen.img
title Install Red Hat Enterprise Linux 5
    root (hd0,0)
    kernel /vmlinuz-5 ks=http://192.168.0.254/workstation.cfg ksdevice=eth0 noipv6
    initrd /initrd-5
    password --md5 $1$FSUEU/$uhUUc8USBK5QAXc.BfW4m.
```

在 grub.conf 中我们可以看到启动菜单的停留时间 timeout=5，以及启动项顺序。你可以根据自身情况修改启动项的顺序。

注意：上述 grub.conf 的信息与各位的电脑的分区和系统有关，每个人并非完全相同。

3、更改运行级别，设置一个默认引导到运行级别 3 的系统：

```
[root@desktopX ~]$ vim /etc/inittab
……
# Default runlevel. The runlevels used by RHS are:
```

```
#      0 - halt (Do NOT set initdefault to this)
#      1 - Single user mode
#      2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#      3 - Full multiuser mode
#      4 - unused
#      5 - X11
#      6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
……
```

找到这一行 **id:5:initdefault:**
将 5 改成 3 **id:3:initdefault:**

重启系统，启动后系统的运行级别就成为完全多用户模式。

4、将默认的运行级别恢复成 5：
修改方法与上述相同，这里不再重复。

本次实验到此结束。

实验 10：安装和管理软件

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

挂载光盘，配置 yum 源，并通过 yum 安装更新软件。

实验背景：

你想要将你的系统连接到专用 yum 库来安装和升级软件。

实验要求：

1、配置 yun
2、尝试使用 yum 的基本语法
3、使用 yum 安装 wireshark

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、将 RHEL6 中的 Server 文件夹拷贝至/yum 下：

[root@desktopX ~]# **mount /dev/cdrom /mnt**
mount: block device /dev/cdrom is write-protected, mountinf read-only
[root@desktopX ~]# **cp -rv /mnt/Server /yum**
'/mnt/Server/' -> '/yum/Server'
'/mnt/Server/' -> '/yum/Server'
……

注意：请保证你的/yum 目录有足够的空间，本实验约占用 2.6G。

3、编辑/etc/yum.repo.d 下的文件：

[root@desktopX ~]# **cd /etc/yum.repo.d**
[root@desktopX yum.repo.d]# **ls**
rhel-debuginfo.repo
[root@desktopX yum.repo.d]# **cp rhel* local.repo**
[root@desktopX yum.repo.d]# **vim lo***

将源文件内容做如下修改

[Server]
name = RHEL local Server
baseurl = file:///yum/Server
enable = 1
gpgcheck = 1
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

注意：如果需要检查软件签名，请设置 gpgcheak=1。关于 baseurl，如果 yum 源在本地，则使用 file://本地地址；如果 yum 源在远端，则应该使用 http://或 ftp://。

4、进入 yum 目录，安装 createrepo：

[root@desktopX ~]# **cd /yum/Server**
[root@desktopX Server]# **rpm -ivh createrepo***
warning: ……

注意：createrepo 是用来创建 yum 数据库的软件包。

5、使用 **createrepo** 命令生成 yum 仓库依赖关系：

[root@desktopX Server]# **createrepo -g /yum/Server**
……

6、清空当前系统保存的 yum 信息：

[root@desktopX ~]# **cd /etc/yum.repo.d**
[root@desktopX yum.repo.d]# **yum clean all**
Loaded plugins: rhnplugin, security
……

注意：每次修改 yum 数据库的内容，就需要清除客户机缓存信息，否则会出现不可预料的错误。

7、刷新 yum 源列表：

[root@desktopX yum.repo.d]# **yum list**
Loaded plugins: rhnplugin, security
This system is not registered with RHM.
RHM support will be disabled.
……

8、使用基本的 **yum** 命令搜索软件 wireshark：

[root@desktopX yum.repo.d]# **yum search wireshark**

Loaded plugins: rhnplugin, security

This system is not registered with RHN.

RHN support will be disabled.

========================= Matcheded: wireshark ==============================

wireshark.i386 :Network traffic analyzer

wireshark-gnome.i386 :Gnome desktop integration for wireshark and wireshark-usermode

9、查看 wireshark 的详细信息：

[root@desktopX yum.repo.d]# **yum info wireshark**

Loaded plugins: rhnplugin, security

This system is not registered with RHN.

RHN support will be disabled.

Available Package

Name : wireshark

Arch : i386

Version : 1.0.8

Release : 1.el5_3.1

10、使用 **yum** 命令安装 wireshark：

[root@desktopX yum.repo.d]# **yum install wireshark**

Loaded plugins: rhnplugin, security

This system is not registered with RHN.

RHN support will be disabled.

Setting up Insrall Process

Resolving Dependencices

--> Running transaction check

---> Package wireshark.i386 0:1.0.8-1.e15_3.1 set to be updated

--> Processing Dependency: libsmi.so.2 for package: wireshark

--> Running libsmi.i386 0:8.4.5-2.e15 set to be updated

yum 在安装过程中会自动处理依赖关系，遇到缺少的软件包会自动下载并安装。

当屏幕出现 Complete 时，表示安装完成。

本次试验到此结束。

✧ yum 主要命令参考手册

参数	功能
help	显示使用信息
list	显示已安装和可用的软件包
search *KEYWORD*	按关键字列出软件包
info *PACKAGENAME*	列出软件包的详细信息
install *PACKAGENAME*	获取并安装包，包括所有依赖项
remove *PACKAGENAME*	删除安装的软件包，包括所有受支持的包
update *PACKAGENAME*	获取并安装更新版本的软件包，包括所有依赖项
provide *PATHNAME*	可显示与指定的路径名相匹配的软件包（通常包括通配符）

实验 11：配置 Linux 内核

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证，能使用 DHCP 联网的系统。

实验目标：

提高调整/proc 文件系统的技能，获取一些配置设备特殊文件和模块的经验，并且实用工具来检测硬件资源。

实验背景：

你想尽量不公开关键系统，打算用 ICMP ECHO 请求将它“藏”到不易被发现的地方。同时，你想确定一下在你的系统上正在运行的进程有哪些，哪些硬件可以用，以及还剩下多少 RAM。

实验要求：

1、使系统对 ping 不响应

2、查看内存使用情况及网卡配置

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、检查/proc/sys/net/ipv4/icmp_echo_ignore_all 的当前值：

[root@desktop2 ~]# **cat /proc/sys/net/ipv4/icmp_echo_ignore_all**

0

就是说你的系统将通常回应 **ping** 的命令。

3、将/proc/sys/net/ipv4/icmp_echo_ignore_all 的值改为 1：

[root@desktop2 ~]# **echo 1> /proc/sys/net/ipv4/icmp_echo_ignore_all**

[root@desktop2 ~]# **cat /proc/sys/net/ipv4/icmp_echo_ignore_all**

1

现在就可以阻止其他主机 ping 你的主机了，但不会影响你 ping 他们。

现在测试 ping 同一网段内的工作站，终端显示一些统计的数据。说明你可以连接服务器了，按 Ctrl+C 会终止 ping 命令。

接下来让同一网段内的工作站 ping 你的工作站，它们不会收到任何从你系统发出的响应。另外，自己 ping 自己的网络地址，也没有反应。

注意：我们对/proc 文件系统的改动是暂时的，如果你想这些改动在重新引导后还有效的话，你需要修改配置文件/ect/sysctl.conf。

4、修改/etc/sysctl.conf，使刚才的改动永久有效：

[root@desktop2 ~]# **vim /etc/sysctl.conf**

……

在末尾添加以下内容

net.ipv4.icmp_echo_ignore_all=1

5、激活修改的配置：

[root@desktop2 ~]# **systcl -p**

再次查看/proc 中的值。如果没有变成 1，继续检查前面两步。然后重新引导你的系统，并在次检查/proc 中的值。

6、清除改动，恢复 ping 响应：
[root@desktop2 ~]# **vim /etc/sysctl.conf**

在配置文件中删除或者标出（语句前加#）**net.ipv4.icmp_echo_ignore_all=1**，然后重新引导系统。

7、查看内存状况：
[root@desktop2 ~]# **top**
……

输入'**M**'将进程按 CPU 消耗量的降序排列
输入'**P**'将进程按内存用量的将序排列
注意：top 命令可以显示一张当前系统中的进程信息，每 5 秒钟更新一次。使用 top 命令还可以轻松的改变进程的排列顺序、中止进程(kill)以及重设 nice 值(renice)。

8、使用 vmstart 命令，每隔 5 秒获取一个内存快照：
[root@desktop2 ~]# **vmstat 5**

同时启动一个需要大量内存空间的应用程序，并观察结果
[root@desktop2 ~]# **cat /dev/had > /dev/null**

9、确定当前连接到你的网络的网卡类型（品牌和/或者型号）：
[root@desktop2 ~]# **lspci**

另外 hal-device-manager 也将显示有关网络接口卡的更多信息。

本次实验到此结束。

实验 12：Linux 进程管理

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证的系统。

实验目标：

进一步学习 Linux 下的进程管理，更加有利于对操作系统的认识和了解。

实验背景：

你想查看你的工作站上运行有点缓慢，因此你想查看以下你的工作站进行检测，查看你的系统上正在运行的哪些进程，又有哪些是不必要的，需要关掉的进程。

实验要求：

1、完成简单的进程查看、件事、调度和控制

2、找出系统中使用最多 CPU 时间的进程

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、使用 **PS** 命令来查看进程信息：

```
[root@desktop2 ~]# ps
PID          TTY          TIME         CMD
3572         tty1         00:00:00     bash
3658         tty1         00:00:00     ps
```

为了查看更完整的进程信息，可以使用参数命令

```
[root@desktop2 ~]# ps axuf
[root@desktop2 ~]# ps axo pid,comm
```

3、搜索并列出使用 tty1 的进程：

```
[root@desktop2 ~]# ps axo comm,tty | grep tty1
bash                    tty1
ps                tty1
grep                    tty1
```

使用管道符将 ps 命令的输出结果发送给 grep 命令进行筛选，这是最灵活的处理方法。

4、查看并调整进程优先级：

```
[root@desktop2 ~]# ps ao comm,nice
COMMAND        NI
mingetty            0
mingetty            0
mingetty            0
mingetty            0
mingetty            0
Xorg                    0
bash                    0
```

ps 0

注意：进程优先级决定了 CPU 对进程调度顺序。在 Linux 中，进程优先级受 nice 值 (nice valve)的影响，nice 值得范围是-20~19，默认为 0。nice 值越小，则进程对 CPU 的使用权越具有优先性。

5、使用 **nice** 命令改变调度优先级：

[root@desktop2 ~]# **nice -n 5 vim**

Ctrl+z

[root@desktop2 ~]# **ps ao comm,nice,pid**

COMMAND	NI	PID
mingetty	0	3089
mingetty	0	3092
mingetty	0	3095
mingetty	0	3096
mingetty	0	3111
Xorg	0	3207
bash	0	3572
vim	0	15396
ps	0	15402

这是在进程启动时使用的命令，对于已经启动的进程，需使用 **renice** 命令

[root@desktop2 ~]# **renice 8 15396**

注意：启动后的进程需要使用 renice 命令结合 pid 来改变优先级。

6、在命令行后面添加'**&**'符使之在后台运行：

[root@desktop2 ~]# **vi &**

[2] 15471

[root@desktop2 ~]# **jobs** //列举后台作业的号码和名称

[1]- Stopped nice -n 5 vim

[2]+ Stopped vi

[root@desktop2 ~]# **fg 2** //使后台进程在前台恢复，需跟随作业号码

Ctrl+z

[root@desktop2 ~]# **cat /dev/urandom > /dev/null**

//为了实验后台恢复运行具有持续性的命令

Ctrl+z

[root@desktop2 ~]# **jobs**

[1] Stopped nice -n 5 vim

[2]- Stopped vi

[3]+ Stopped cat /dev/urandom > /dev/null

[root@desktop2 ~]# **bg 3** //在后台恢复进程的运行，需跟随作业的号码

[root@desktop2 ~]# **jobs**

[1]- Stopped nice -n 5 vim

[2]+ Stopped vi

[3] Running cat /dev/urandom > /dev/null

7、终止之前后台所有的进程：

[root@desktop2 ~]# **killall cat**

[root@desktop2 ~]# **kill -9 %1**

[root@desktop2 ~]# **kill -9 %1**

注意：kill 命令用来给进程发送控制信号，kill 命令后跟 PID 号或者有'%'前缀的 job 作业号。killall 与 kill 命令有着同样的作用。

8、查找系统中占用 CPU 时间最多的进程：

[root@desktop2 ~]# **ps axo pid,comm,pcpu**

//查看进程的 PID、名称以及 CPU 占用率

[root@desktop2 ~]# **ps axo pid,comm,pcpu -sort=pcpu**

// sort 参数以 pcpu 为对象对输出的内容进行整理

[root@desktop2 ~]# **ps axo pid,comm,pcpu -sort=pcpu | tail -n1**

//将结果输出到 tail 命令筛选出最后一条

11695 makewhatis 3.5

本次实验到此结束。

✧ ps 主要命令参考手册

参数	功能
a	包括所有终端中的进程
x	包括不连接终端的进程
u	显示进程所有者的信息
f	显示进程的父进程
o	属性定制：pid、comm、%cpu、%mem、state、tty、euser、ruser

✧ 进程状态(State)

状态名	含义
Running	运行：进程正在运行
Sleeping	休眠：在内存中，但什么也不做
Uninterruptable	等待：正在等待资源，处于不可中断的休眠状态
Zombie	僵尸：已经终止了但尚未从进程列表中清除

实验 13：用户和组管理

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

建立用户和组管理技巧，建立可被多个组和用户访问的目录。

实验背景：

你需要为公司的不同部门设定组，并且需要为员工设定用户帐户。对于每个部门而言你还需要一个共享文件夹，可以让同一部门的用户共享文档，但可以阻止其它部门更改，甚至不允许他们查看这些文档。而有些数据需要几个组都能访问，使用 ACL 来实现这个功能。

实验要求：

1、创建用户和组，并将用户添加到响应的组

2、设定共享的文件目录

3、添加访问控制列表(ACL)

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、确定所有新创建的用户可以生成有组写入权限的文件，在文件 /etc/skel/.bashrc 中添加 umask 条目，以便生成有组写入权限的文件(002)：

[root@desktop2 ~]# **echo 'umask 002' >> /etc/skel/.bushrc**

3、为一下七个用户在系统中添加帐户，他们是 joshua、alex、dax、byran、zak、ed 和 manager，初始密码统一为 password：

[root@desktop2 ~]# **useradd joshua**
[root@desktop2 ~]# **passwd joshua**
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@desktop2 ~]# **useradd alex**
[root@desktop2 ~]# **passwd alex**
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
……

其余帐户添加方法相同，此处省略

注意：因为你需要添加多个帐户，那么使用 for-loop 可以提高效率，你可以在命令行中输入或者放在文件中以 shell 脚本的方式来运行：

```
for USER in Joshua alex dax bryan zak ed manager
do
    useradd $USER
    echo password | psswd --stdin $USER
done
```

4、在系统中添加下列用户组：

sales (GID:200)

hr (GID:201)

web (GID:202)

[root@desktop2 ~]# groupadd -g 200 sales

[root@desktop2 ~]# groupadd -g 201 hr

[root@desktop2 ~]# groupadd -g 202 web

注意：新用户的主要组的 GID 讲师大于 500 的最小可用值。大多数管理员想将附加组的 GID 控制在一定的范围内，以方便管理。所以此处以这种手动的方式设定 GID 而不知系统默认设定 GID。

5、将 joshua 和 alex 添加 sales 辅助组，dax 和 bryan 添加到 hr 辅助组，zak 和 ed 添加到 web 辅助组，manager 添加到所有辅助组：

[root@desktop2 ~]# **usermod -G sales joshua**

[root@desktop2 ~]# **usermod -G sales alex**

[root@desktop2 ~]# **usermod -G hr dax**

[root@desktop2 ~]# **usermod -G hr byran**

[root@desktop2 ~]# **usermod -G web zak**

[root@desktop2 ~]# **usermod -G web ed**

[root@desktop2 ~]# **usermod -G sales,hr,web manager**

6、将 web 组添加到 dax 的辅助组中：

[root@desktop2 ~]# **usermod -G web,hr dax**

或者

[root@desktop2 ~]# **usermod -a -G web dax**

注意：如果你运行了 usermod -G web dax，那么你可能将 dax 从 hr 组和其它拥有 dax 的辅助组中删除了。

7、你可以以每个用户的身份登录，使用 id 命令来验证他们是否属于正确的组：

你可以使用 **su -user**，并运行 **id** 命令，或者简单的使用用户名作为 id 命令的参数

```
for USER in joshua alex dax byran zak ed manager
do
   id $USER
done
```

8、创建一个名为/depts 的目录，其中有 sales、hr 和 web 的子目录：

[root@desktop2 ~]# **mkdir -p /depts/{sales,hr,web}**

9、将每个目录的组拥有者设定为与组名匹配的组：

[root@desktop2 ~]# **chgrp sales /depts/sales**

[root@desktop2 ~]# **chgrp hr /depts/hr**

[root@desktop2 ~]# **chgrp web /depts/web**

10、检查/depts 目录的权限，以确定任何人都可以访问该目录，但不可以在该目录中执行写操作。改变每个子文件夹的权限，为组成员提供完全权限(rwx)，但拒绝其他组的成员访问：

[root@desktop2 ~]# **ls -ld /depts**

drwxr-xr-x 5 student student 4096 Sep 30 21:08 /depts

[root@desktop2 ~]# **chmod 770 /depts/***

11、在那些目录中创建的文件应该属于适当的组，设定合适的权限：

[root@desktop2 ~]# **chmod g+s /depts/***

12、通过以各个用户的身份登录进行测试，并生成或者更改每个目录中的文件。只有manager 应该能够进入所有目录：

[root@desktop2 ~]# **su - joshua**
[joshua@desktop2 ~]$ **touch /depts/sales/text**
[joshua@desktop2 ~]$ **exit**

[root@desktop2 ~]# **su - alex**
[alex@desktop2 ~]$ **touch /depts/sales/text**
[alex@desktop2 ~]$ **exit**

[root@desktop2 ~]# **su - dex**
[dex@desktop2 ~]$ **touch /depts/hr/text**
[dex@desktop2 ~]$ **exit**

[root@desktop2 ~]# **su - bryan**
[bryan@desktop2 ~]$ **touch /depts/hr/text**
[bryan@desktop2 ~]$ **exit**

[root@desktop2 ~]# **su - zak**
[zak@desktop2 ~]$ **touch /depts/web/test**
[zak@desktop2 ~]$ **exit**

[root@desktop2 ~]# **su - ed**
[ed@desktop2 ~]$ **touch /depts/web/test**
[ed@desktop2 ~]$ **exit**
[root@desktop2 ~]# **su - manager**
[manager@desktop2 ~]$ **touch /depts/sales/test**
[manager@desktop2 ~]$ **touch /depts/hr/test**
[manager@desktop2 ~]$ **touch /web/sales/test**
[manager@desktop2 ~]$ **exit**

13、在/dept 中创建一个名为 tech 的新目录。更改权限，使拥有者为根用户，并属于组 hr。赋予该用户和组完全访问权限，并不允许其他任何人访问，并设置 SGID 位：

[root@desktop ~]# **mkdir /dept/tech/**
[root@desktop ~]# **chown root:hr /dept/tech/**
[root@desktop ~]# **chmod 2770 /dept/tech/**

14、使用 acl 赋予 web 组对于/dept/tech/的完全权限：

[root@desktop ~]# **setfacl -m g:web:rwx /depts/tech**

15、赋予 alex 对/depts/tech 目录的读/执行权限，为 alex 设定对该目录的默认 ACL 读/写权限：

[root@desktop ~]# **setfacl -m u:alex:rx /depts/tech**
[root@desktop ~]# **setfacl -m d:u:alex:rw /depts/tech**

16、在/dept/tech/中创建一些文件作为一些用户和验证访问：

[root@desktop2 ~]# **su - joshua**
[joshua@desktop2 ~]$ **touch /depts/tech/joshua**
touch：cannot touch '/depts/tech/joshua': Permission denied

[joshua@desktop2 ~]# **exit**
[root@desktop2 ~]# **su - manager**
[manager@desktop2 ~]$ **touch /depts/tech/manager**

[manager@desktop2 ~]$ **getfacl /depts/tech/manager**
getfacl: Removeing leading '/' from absolute path names
#file: depts/tech/manager
#owner: manager
#group: hr
user::rw-
user:alex:rw-
group：rwx #effective：rw-
mask：rw-
other：---
[manager@desktop2 ~]$ **exit**

[root@desktop2 ~]# **su - alex**
[alex@desktop2 ~]$ **touch /depts/tech/manager**
[alex@desktop2 ~]$ **touch /depts/tech/alex**
touch：cannot touch '/depts/tech/alex': Permission denied
[alex@desktop2 ~]$ **exit**

本次实验到此结束。

实验 14：管理网络设定

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证，能使用 DHCP 联网的系统。局域网内存在一台设置了 IP 为 192.168.0.2 的工作站。

实验目标：

联系手动配置联网的技能，配置成使用静态网络设置而不是动态网络设置进行操作的系统。

实验背景：

在你的系统上安装网络服务，手动配置了 IP 地址，并且你决定为这项服务使用虚拟 IP 地址，这对今后的使用提供了更多灵活的服务。你的同事也照样创建了一个虚拟的 IP 地址，给地址不经过你的网关，但你需要获取访问权。

实验要求：

1、设定一个静态的 IP 地址

2、为工作站添加虚拟 IP

3、与邻居建立静态路由

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、查看当前工作站的网络配置信息(IP 地址、子网掩码、默认网关和 DNS 服务器)：

IP 地址、子网掩码：

[root@desktop2 ~]# **ip addr show**

默认网关：

[root@desktop2 ~]# **ip route show**

DNS 服务器：

[root@desktop2 ~]# **cat /etc/resolv.conf**

3、关闭你的第一以太网口：

[root@desktop2 ~]# **ifdown eth0**

4、使用文本编辑器打开/etc/sysconfig/network-scripts/ifcfg-eth0 文件，手动添加 IP 地址子网掩码和默认网关：

DEVICE=eth0
BOOTPROTO= none
ONBOOT=yes
IPADDR=192.168.0.1
NETMASK=255.255.255.0
GATEWAY=192.168.0.254

注意：其他行的内容不用管。

5、使用你的新配置的接口，并确定你的设置可行：

[root@desktop2 ~]# **ifup eth0**

然后可以 ping 一下同一网段中的工作站。

6、使用名为 eth0:1 的新子接口来配置你的系统。子接口的 IP 地址为 10.0.1.1，子网掩码 255.255.255.0。用文本编辑器创建文件/etc/sysconfig/network-scripts/ifcfg-eth0:1：

[root@desktop2 ~]# **vim /etc/sysconfig/network-scripts/ifcfg-eth0:1**
DEVICE=eth0:1
BOOTPROTO= none
ONBOOT=yes
IPADDR=10.0.1.1
NETMASK=255.255.255.0

7、启用子接口并验证连通性：

[root@desktop2 ~]# **ifup eth0:1**
[root@desktop2 ~]# **ping 10.0.1.1**

8、为邻居(192.168.0.2)添加子接口并启用：

[root@desktop2 ~]# **vim /etc/sysconfig/network-scripts/ifcfg-eth0:1**
DEVICE=eth0:1
BOOTPROTO= none
ONBOOT=yes
IPADDR=10.0.2.1
NETMASK=255.255.255.0
[root@desktop2 ~]# **ifup eth0:1**

9、根据邻居的常规 IP 地址在你的工作站上创建一个连接到你邻居子接口的静态路由：

网络 10.0.2.0/255.255.255.0
通过网关 192.168.0.2
[root@desktop2 ~]#**ip route add 10.0.2.0/24 via 192.168.0.2**

10、查看当前的路由表，确定添加了上述路由：

[root@desktop2 ~]# **ip route**
10.0.2.0/24 via 192.168.0.2 dev eth0

11、通过访问你的邻居的别名设备来访问远程网络：

[root@desktop2 ~]# **ping -c 4 10.0.2.1**

12、固定这个路由，创建/etc/sysconfig/network-scripts/route-eth0 的文件：

[root@desktop2 ~]# **vim /etc/sysconfig/network-scripts/route-eth0**
10.0.2.1/24 via 192.168.0.2

13、重新联网，并测试：

[root@desktop2 ~]# **service network restart**
[root@desktop2 ~]# **ping -c 4 10.0.2.1**

本次实验到此结束。

实验 15：计划任务 Crontab

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

掌握 crotab 基本语法结构，启动 crond 服务，设置相应的计划任务。

实验背景：

你在早上 8 点到下午 5 点之间，每隔 10 分钟运行一次/usr/bin/free 和/bin/ps，并将这些任务的结果发送到其他电子邮件地址。如果说每次都手动进行的话实在是太繁琐了，于是你打算设置计划任务帮助你完成这些工作。

实验要求：

1、学会使用计划任务

2、会使用 crond 服务的黑名单功能

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、编辑 cron 文件：

[root@desktopX ~]# **crontab -e**

在文件中输入下列行

*/10 8-17 * * * /usr/bin/free;/bin/ps|mail rhce@localhost

3、禁止 rhce 账户使用 crontab：

[root@desktopX ~]# **vim /etc/cron.deny**

……

在文件上写入 rhce，然后保存并退出

4、重新启动 crond 服务：

[root@desktopX ~]# **/etc/init.d/crond restart**

注意：若以 root 身份使用 crontab -e -u rhce,依然可以让 rhce 使用 crond,但在 rhce 账号底下则不能使用 crontab -e 命令。

✧ 特别提醒：

新手可能分不清用 crontab -e 编辑和直接编辑/etc/crontab 有什么不一样？以下是需要注意的地方：

1、/etc/crontab 这个里面的计划任务是系统中的计划任务。

2、crontab -e 是用来编辑某个用户的计划任务。

3、每条命令执行完毕之后，系统会自动将输出发送邮件给当前系统用户。日积月累，非常的多，甚至会撑爆整个系统。所以每条命令命令后面进行重定向处理是非常必要的：>>/dev/null 2>&1。

4、每次编辑完某个用户的 cron 设置后，cron 自动在/var/spool/cron 下生成一个与此用户同名的文件，此用户的 cron 信息都记录在这个文件中，这个文件是不可以直接编辑的，只可以用 crontab -e 来编辑。

5、cron 启动后每过一份钟都会读取/var/spool/cron/username 和/etc/crontab 这两个文件，检查是否要执行里面的命令。因此此文件修改后不需要重新启动 cron 服务。

6、在默认情况下，如果 cron 每执行一次指令后，都会向用户的本地信箱中发送邮件，时间长了则产生数量很多的邮件。那么如何禁止掉呢？可能通过下面的方法来实现

将执行命令的信息重定向到

cron_command >/dev/null 2>&1

在 crontab 设置文件中指定发送到得邮件

MAILTO="rhce@localhost"

本次实验到此结束。

实验 16：Linux 文件系统管理

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

增加有关分区和文件系统的技能和知识，掌握常见磁盘分区的相关操作及一般步骤。

实验背景：

无论是 Windows 还是 Linux，我们日常操作与使用几乎都是围绕文件系统而展开的。有一天你突然发现你的现有的硬盘空间不够用了，巧的是你还有一些的空余空间。遇事你打算把这些空余的空间开辟出来以便使用。

实验要求：

按要求新建磁盘分区：

	存储空间	文件系统
	1024MB	ext2 Block size:4K Laber:Music
	512MB	ext3
	512MB	vfat

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、使用 **fdisk** 命令查看当前磁盘使用情况：

[root@desktop2 ~]# **fdisk -l**

Disk /dev/sda: 10.4 GB, 21474836480 bytes

255 heads, 63 sectors/track, 2610 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	ID System
/dev/sda1	*	……其余项省略			
/dev/sda2		……其余项省略			
/dev/sda3		……其余项省略			
/dev/sda4		……其余项省略			

3、使用 **fdisk** 命令按照实验要求创建 3 个新分区：

[root@desktop2 ~]# **fdisk /dev/sda**

This number of cylinder for this disk is set to 2610.

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

1) software that runs at boot time (e.g., old versions of LILO)

2) booting and partitioning software from other OSs (e.g., DOS FDISK, OS/2 FDISK)

在 Command 命令处输入'**n**'，开始新建磁盘分区

Command(m for help): **n**

选择新建磁盘的属性（扩展 extend/主盘 primary），这里我们选择主盘 primary
Command action
e extended
p primary partition (1-4)
p

然后按要求划分新建磁盘的空间
First cylinder (942 - 2610, default 942):
Using default value 942
Last cylinder or +size or +sizeM or +sizeK (942-2610, default 2610): **+1024M**
注意：fdisk 命令会开始分区引导
首先要求输入分区开始的煮面编号，此处可以自动义输入，也可以直接回车，是新建的分区开始的柱面紧邻前一分区的结束柱面；
其次输入新建分区的结束柱面，默认键入回车会将所有剩余空间划分至新的分区，如果需要，请自定义需要新建的分区空间大小，默认形式会定义该分区结束的柱面，若以'+'为前缀并以"容量+单位"为形式输入则表示分区容量的大小。

/dev/sda6 与/dev/sda7 的创建方式相同，这里不再重复

查看当前磁盘上的分区状态，对上述分区操作的正确性进行验证
Command(m for help): **p**

Device	Boot	Start	End	Blocks	ID	System
/dev/sda1	*	……其余项省略				
/dev/sda2		……其余项省略				
/dev/sda3		……其余项省略				
/dev/sda4		……其余项省略				
/dev/sda5		…	…	1004031	83	Linux
/dev/sda6		…	…	522081	83	Linux
/dev/sda7		…	…	522081	83	Linux

保存并退出 fdisk 工具
Command(m for help): **w**
The partition table has been altered !

此时分区信息将被写入，新分区将被创建。
注意：新建完分区后可能会有 warning，那是由于内核仍在使用旧的磁盘分区表，新的磁盘分区表将在计算机重新启动后生效。

4、使用 **partprobe** 命令使内核立刻接受新的分区表信息：
[root@desktop2 ~]# **partprobe /dev/sda**

5、使用 **mke2fs** 命令在/dev/sda5 上创建 ext2 格式的文件系统：
[root@desktop2 ~]# **mke2fs -L music -m 0 -b 4096 -c /dev/sda5**
……
Filesystem label = music
OS type: Linux
Block size = 4096 (log=2)
Fragement size = 4096 (log=2)

……

6、使用 **dump2fs** 命令验证新建的/dev/sda5 是否正确：
[root@desktop2 ~]# **dump2fs /dev/sda5 | more**
Filesystem volume name: music
……

7、为/dev/sda6 建立 ext3 格式的文件系统：
[root@desktop2 ~]# **mke2fs -j -c /dev/sda6**
或者使用 mkfs.ext3 命令
[root@desktop2 ~]# **mkfs.ext3 -c /dev/sda6**

8、使用 **dump2fs** 命令验证新建的/dev/sde6 是否正确：
[root@desktop2 ~]# **dump2fs /dev/sda6 | more**
……
Journal backup: inode blocks
Journal size: 4114k

可以看到，由于有 Journal（日志）的相关信息，可以判定文件系统格式为 ext3。

8、为/dev/sda7 建立 vfat 格式的文件系统并检验：
[root@desktop2 ~]# **mkfs.vfat /dev/sda7**
[root@desktop2 ~]# **dump2fs /dev/sda7 | more**

9、使用 parted 工具对本实验进行验证：
[root@desktop2 ~]# **parted /dev/sda**
……
(parted) p
……
可以看到，新创建的文件系统均符合本实验的要求

本次实验到此结束。

✧ Linux 支持的文件系统

文件系统	说明
msdos	DOS 和 Windows 使用的 FAT 文件系统。不支持许多高级的特性，文件名最多包含 8 个字符和一个 3 个字符的后缀
ext2	非常有特色的出自于 Linux 的文件系统，虚拟文件系统曾围绕 ext2 文件系统而设计。它具有 Linux 文件系统要求具有的所有特性，并且被设计成易于升级的，这样文件系统代码的最新版本就不需要重建已存在的文件系统
ext3	红帽企业版 Linux 的默认文件系统。它是 ext2 的扩成版本，可以支持日志功能，除此之外在特性上与 ext2 完全一样
vfat	msdos 文件系统的扩充版本，允许使用长文件名
nfs	一个网络文件系统，允许在许多计算机之间共享一个文件系统，易于这些计算机对文件的访问，支持并发
smbfs	用来在 Linux 和 Windows 之间共享目录的文件系统
iso9660	标准的 CD-ROM 文件系统，自动支持允许更长文件名的 Rock Ridge 扩展文件系统
proc	仅存在与内核中的一种虚拟文件系统

✧ mke2fs 命令参数

参数	功能
-b block size	指定文件系统块大小，可用值为 1024、2048 或 4096
-c	建立文件系统时检查坏损块

-L label	指定卷标
-m n	将后备块百分比设为 n%
-j	建立文件系统日志

实验 17：配置 LVM 逻辑卷管理器

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

了解磁盘分区的相关知识，掌握如何创建 LVM。

实验背景：

LVM 是逻辑盘卷管理（Logical Volume Manager）的简称，它是 Linux 环境下对磁盘分区进行管理的一种机制，LVM 是建立在硬盘和分区之上的一个逻辑层，来提高磁盘分区管理的灵活性。现在你需要新建两块 LVM 的分区，实现在多个物理设备之间重新组织文件系统，在多个物理设备之间的重新设定和删除逻辑卷，还可以使用自定义名称来命令逻辑卷。

实验要求：

1、创建两个 LVM 逻辑卷，大小为 100MB

2、将创建好的 LVM 卷进行扩展，容量扩展至 200MB

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、使用 **fdisk** 命令创建新的磁盘分区，并将其磁盘类型标记(Hex Code)设置为 8e(type:Linux LVM)：

```
[root@desktop2 ~]# fdisk /dev/sda
Command(m for help): n
Command action
   e      extended
   p      primary partition (1-4)
p
Command(m for help): p
Device          Boot        Start        End        Blocks        ID
        System
/dev/sda1  *            ……其余项省略
/dev/sda2               ……其余项省略
/dev/sda3               ……其余项省略
/dev/sda4               ……其余项省略
/dev/sda5               …            …          1004031       83
        Linux
Command(m for help): t
primary partition (1-4): 5
Hex Code (type L to list codes): 8e
```

注意：对磁盘类型标记值不清楚的话可以输入 L 查看磁盘格式类型值列表。

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

/dev/sda6 的创建与上述相同，这里就不在重复。

3、使用 **pvcreate** 命令创建物理卷：

```
[root@desktop2 ~]# pvcreate /dev/sda5 /dev/sda6
Physical volume “/dev/sda5” successfully created
```

Physical volume “/dev/sda6” successfully created

4、使用 **vgcreate** 命令创建卷组(volume group)，命名为 vg1：
[root@desktop2 ~]# **vgcreate vg1 /dev/sda5 /dev/sda6**
/dev/hdc: open failed: No medium found
Volume group “vg1” successfully created

5、创建完成后，可以使用 **vgdisplay -v** 查看创建的情况：
[root@desktop2 ~]# **vgdisplay -v**
Finding all volume groups
Finding volume group “vg1”
--- Volume group ---
VG Name vg1
System ID
Format lvm2
Metadata Areas 2
Metadata Sequence No 1
……

6、使用 lvcreate 命令从 vg1 创建名为 lvm1 的逻辑卷(logical volumn)，其大小为 100M：
[root@desktop2 ~]# **lvcreate -L 100M -n lvm vg1**
Logical volume “lvm1” created

7、使用 lvdisplay -v 查看 lvm 的创建情况：
[root@desktop2 ~]# **lvdisplay -v**
Finding all logical volumes
- - - Logical volume - - -
LV Name /dev/vg1/lvm1
VG Name vg1
……

8、使用 mkfs.ext3 命令对创建好的 lv 进行格式化：
[root@desktop2 ~]# **mkfs.ext3 /dev/vg1/lvm1**

9、将创建好的 lv 挂载至/mnt 目录：
[root@desktop2 ~]# **mount /dev/vg1/lvm1 /mnt**

至此，创建 lvm 逻辑卷的步骤到此结束。

10、扩展 lvm 的容量至 200MB：
[root@desktop2 ~]# **lvextend -L +100M /dev/vg1/lvm1**
Extending logical volume lvm1 to 200.00 MB
Logical volume lvm1 successfully resized

11、再次使用 lvdisplay -v 查看磁盘中现有的 lvm 的情况：
[root@desktop2 ~]# **lvdisplay -v**

12、使用 resize2fs 命令重新设置文件系统大小：
[root@desktop2 ~]# **resize2fs -p /dev/vg1/lvm1**
……

本次实验到此结束。

✧ vgcreate 命令参数

文件系统	说明
i	卷组最大能包含的逻辑参数
p	卷组最大能包含的物理卷数
e	卷组中每个物理卷最多能包括的物理单元(PE)数
s	卷组中物理单元(PE)的尺寸大小

✧ lvcreate 命令参数

参数	功能
L	以 MB 为单元表示逻辑卷的大小
l	以逻辑单元的数目来表示逻辑卷的大小
n	逻辑卷的名称；缺省为 lvol1、lvol2 等
C	用邻近的存储空间来创建逻辑卷
I	当逻辑卷使用一个以上的磁盘时，设定穿过每个物理卷得条带尺寸，须与’i’一同使用

实验 18：独立磁盘冗余阵列 RAID

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

了解磁盘分区的相关知识，学会使用 Linux 的软件磁盘冗余阵列 RAID 功能。

实验背景：

你的工作站已经使用了一段日子，里面存放了许多重要的数据。你想对磁盘做个备份以防万一。这时候就可以使用 RAID 1 完成软件磁盘冗余阵列从创建到替换坏区的完整操作。

实验要求：

1、学会使用 mdadm 管理 RAID

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、使用 **fdisk** 命令创建三个新的磁盘分区，并将其磁盘类型标记(Hex Code)设置为 fd：

```
[root@desktopX ~]# fdisk /dev/sda
Command(m for help): n
Command action
   e      extended
   p      primary partition (1-4)
p
Command(m for help): p
Device          Boot        Start        End        Blocks        ID
        System
/dev/sda1  *            ……其余项省略
/dev/sda2               ……其余项省略
/dev/sda3               ……其余项省略
/dev/sda4               ……其余项省略
/dev/sda5               …            …          1004031       83
        Linux
Command(m for help): t
primary partition (1-4): 5
Hex Code (type L to list codes): fd
```

注意：对磁盘类型标记值不清楚的话可以输入 L 查看磁盘格式类型值列表。

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

/dev/sda6、/dev/sda7 的创建与上述相同，这里就不在重复。

3、使用 **mdadm** 工具来配置 RAID 磁盘阵列：

```
[root@desktopX ~]# mdadm -C /dev/md0 -l 1 2 /dev/sda5 /dev/sda6 -a yes
```

4、使用 **mkfs.ext4** 命令在新创建的 RAID 磁盘阵列上创建文件系统：

```
[root@desktopX ~]# mkfs.ext4 /dev/md                    //挂载 RAID
[root@desktopX ~]# mount /dev/md0 /mnt
[root@desktopX ~]# df -TH
```

5、模拟/dev/sda5 损坏，把/dev/sda7 替换进去的过程

模拟损坏

[root@desktopX ~]# **mdadm --fail /dev/md0 /dev/sda5** //删除坏区

[root@desktopX ~]# **mdadm --remove /dev/md0 /dev/sda5** //新增分区

[root@desktopX ~]# **mdadm -a /dev/md0 /dev/sda7**

实验 19：使用目录服务器

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证，能使用 DHCP 联网的系统。局域网内存在一台设置了 IP 为 192.168.0.254 的 ldap 服务器。

实验目标：

掌握 ldap 客户端的使用方法，并能配合使用 autofs，能在客户端以 ldap 的账号登录

实验背景：

你的位置是将新用户的帐户信息迁移至 LDAP 目录服务，系统所在的环境的 ldap 服务器的 dc=example，dc=com，ldap 账号是 ldapuser1，家目录为/home/ldap/ldapuser1。

实验要求：

1、能够使用 ldap 的客户端

2、能够使用 autofs 自动挂载家目录

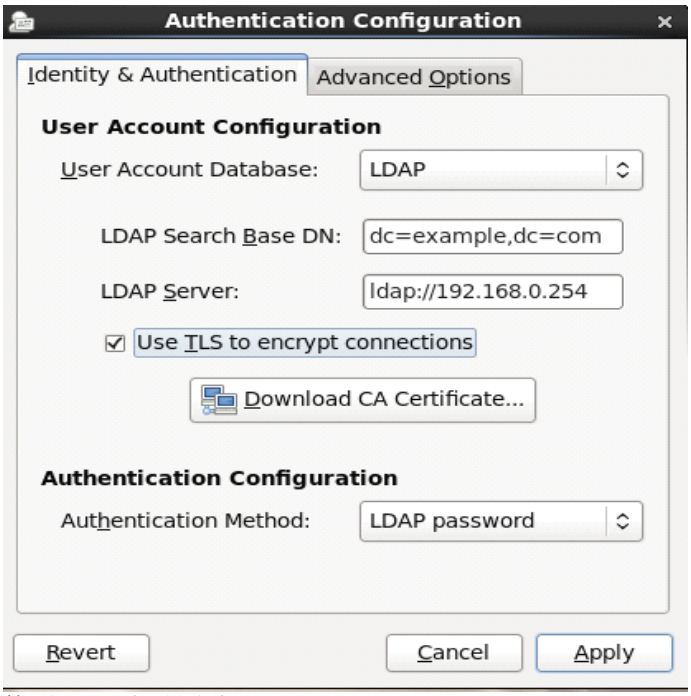
实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、启用 ldap 客户端：

[root@desktopX ~]# **system-config-authentication**

3、配置 ldap 客户端：



4、下面就可以使用 ldap 登录测试了：

[root@desktopX ~]**#su - ldapuser1**

base **$**

注意：此时没有家目录。

5、配置 autofs，自动挂载家目录：
[root@desktopX ~]# **vim /etc/auto.master**
……
在其中增加一行
/home/ldap /etc/auto.ldap
保存后退出

[root@desktopX ~]# **vim /etc/auto.ldap**
*** 192.168.0.254:/home/ldap/&**

7、重启 ldap 服务：
[root@desktopX ~]# **/etc/init.d/autofs stop**
[root@desktopX ~]# **/etc/init.d/autofs start**

8、此时使用 ldapuser1 登录就能进入自己的家目录了：
[root@desktopX ~]# **su - ldapuser1**

本次实验到此结束。

实验 20：磁盘配额 QUOTA

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

掌握磁盘配额的技能和知识，会使用 Linux 的磁盘配额 quota 功能。

实验背景：

你的工作站已经使用了一段日子了。在你的工作站中，你划分了不同的文件系统，每个文件系统中也存放了许多不同的数据。时间长了你也不记得你的工作站当初是如何划分文件系统的，并且你想对磁盘上新建的文件大小做个限制。这时候你就可以借助磁盘配额来帮助你得到你想要的信息。

实验要求：

1、添加磁盘配额的支持

2、使用 quotacheck 扫瞄磁盘

3、使用磁盘配额设定可使用空间

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、编辑 /etc/fstab，使得准备要开放的 quota 磁盘可以支持 quota：

[root@desktopX ~]# **vim /etc/fstab**

……

把下面这一行

/dev/sda2 /home ext4 defaults 1 1

修改成如下形式

/dev/sda2 /home ext4 defaults,usrquota 1 1

3、预扫瞄磁盘的使用情况，使用 **quotacheck** 命令来扫瞄一下我们要使用的磁盘：

[root@desktopX ~]# **umount /dev/sda2**

[root@desktopX ~]# **mount -a**

[root@desktopX ~]# **quotacheck -avu**

4、建立使用者的 quota，使用 **edquota** 命令来编辑每个使用者或群组的可使用空间：

[root@desktopX ~]# **edquota -u test**

……

找到以下配置

Disk quotas for user test (uid 501):

Filesystem blocks soft hard inodes soft hard

/dev/sda2 32 0 0

8 0 0

修改数值如下所述

Disk quotas for user test (uid 501):

Filesystem blocks soft hard inodes soft hard

/dev/sda2 32 40000 50000 8 0 0

4、设定宽限时间：

[root@desktopX ~]# **edquota -t**

Grace period before enforcing soft limits for users:

Time units may be: days, hours, minutes, or seconds

Filesystem	Block grace period	Inode grace period
/dev/sda2	1days	1days

5、测试

在 test 账号下在 home 下无法创建超过 50000KB 的文件

本次实验到此结束。

第三部分：

Red Hat 系统管理 III (RH254)

磁盘加密
访问 iSCSI 存储
实现文件传输协议(FTP)服务
配置 Samba 服务器
实施网络共享(NFS)服务
配置 DNS 服务器
配置邮件服务器
实施 Web(HTTP)服务

实验 21：磁盘加密

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

了解磁盘分区及磁盘加密的相关知识，掌握新建磁盘并对其进行加密处理，实现加密磁盘的开机自动挂载。

实验背景：

你的工作站已经使用了一段日子，里面存放了许多重要的数据。你打算对这个分区进行加密已确保数据的安全性。

实验要求：

1、使用 cryptsetup 对磁盘进行加密

2、设置加密磁盘开机自动挂载

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、使用 **fdisk** 命令创建一块新磁盘：

```
[root@desktopX ~]# fdisk /dev/sda
Command(m for help): n
Command action
   e      extended
   p      primary partition (1-4)
p
Command(m for help): p
Device          Boot        Start        End        Blocks        ID
          System
/dev/sda1  *              ……其余项省略
/dev/sda2                 ……其余项省略
/dev/sda3                 ……其余项省略
/dev/sda4                 ……其余项省略
/dev/sda5                 …            …          1004031       83
          Linux
Command(m for help): t
primary partition (1-4): 5
……
Command(m for help): w
```

3、对改磁盘进行加密和格式化操作

```
[root@desktopX ~]# cryptsetup luksFormat /dev/sda5
```

注意：这里会出现警告，本操作将损坏/dev/sda5 里面的数据，我们输入大写的 YES，切记是大写，然后输入两遍加密密码。

4、设置加密磁盘为开机自动挂载：

如果希望这个映射能在开机的时候自动挂载，那么要写入/etc/fstab 里面去，注意要使用的是 UUID 号

[root@desktopX ~]# **blkid /dev/sda5**
[root@desktopX ~]# **vim /etc/fstab**
……
UUID=XXXXXXXXXXXXXXXX /mnt ext4 defaults 0 0
……

但这个时候系统在启动的时候，会等待你输入密码

如果希望系统启动的时候我们不需要输入密码的话，我们可以先把密码写到一个文本文件里去，系统再启动的时候自动读取这个文件即可

[root@desktopX ~]# **echo -n "redhat">/root/rhce_pw**
[root@desktopX ~]# **cryptsetup luksAddKey /dev/sda5 /root/rhce_pw**

编辑/etc/crypttab
[root@desktopX ~]# **vim /etc/crypttab**
……
写入以下内容
rhce /dev/sdb1 /root/rhce_pw

实验 22：访问 iSCSI 存储

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

了解 isusi 的相关原理，学会使用 iscsi 客户端挂载，掌握如何搭建 isscsi 服务端。

实验背景：

你想在你的工作站上搭建一种基于 TCP/IP 的协议，用来建立和管理 IP 存储设备、主机和客户机等之间的相互连接，并创建存储区域网络的服务。

实验要求：

1、搭建 iscsi 服务器

2、设置 iscsi 服务器开机自动挂载

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、安装 iscsi 的软件包：

[root@desktopX ~]# **yum install scsi-target-utils**

3、搭建 iscsi 服务器端，启动服务：

[root@desktopX ~]# **/etc/init.d/tgtd start**

4、下面就是配置服务端了：

[root@desktopX ~]# **tgtadm --lld iscsi --op new --mode target --tid 1 -T iqn.2011-08.com.example:mayue**

[root@desktopX ~]# **tgtadm --lld iscsi --op new --mode logicalunit --tid 1 --lun 1 -b /dev/sdb1**

[root@desktopX ~]# **tgtadm --lld iscsi --op show --mode target**

[root@desktopX ~]# **tgtadm --lld iscsi --op bind --mode target --tid 1 -I 127.0.0.1**

5、本机作为客户端测试：

[root@desktopX ~]# **iscsiadm -m discovery -t st -p 127.0.0.1**

[root@desktopX ~]# **iscsiadm -m node -T iqn.2011-08.com.example:mayue -p 127.0.0.1:3260 -l**

此时，fdisk 下就会多出/dev/sdc，说明挂载成功

6、格式化 iscsi 磁盘

[root@desktopX ~]# **fdisk /dev/sdc**

7、设置成开机自动挂载：

[root@desktopX ~]# **blkid /dev/sdc1**

……

获得/dev/sdc1 的 UUID 号

……

[root@desktopX ~]# **vim /etc/fstab**

……
把 UUID 添加进去
UUID=”XXXX” /mnt/iscsi ext4 _netdev 0 0

实验 23：实现文件传输协议(FTP)服务

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

掌握 ftp 的使用方法，使用 vsftpd 搭建主机与用户均可正常访问的 FTP 服务器。

实验背景：

LVM 是逻辑盘卷管理（Logical Volume Manager）的简称，它是 Linux 环境下对磁盘分区进行管理的一种机制，LVM 是建立在硬盘和分区之上的一个逻辑层，来提高磁盘分区管理的灵活性。现在你需要新建两块 LVM 的分区，实现在多个物理设备之间重新组织文件系统，在多个物理设备之间的重新设定和删除逻辑卷，还可以使用自定义名称来命令逻辑卷。

实验要求：

1、创建两个 LVM 逻辑卷，大小为 100MB

2、将创建好的 LVM 卷进行扩展，容量扩展至 200MB

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、安装 vsftpd 的软件包：

[root@desktopX ~]# **yum -y install vsftpd***

3、启动 vsftped 服务并设置为开机自动运行：

[root@desktopX ~]# **service vsftpd restart**

[root@desktopX ~]# **chkconfig vsftpd on** //设置开机自动运行服务

注意：vsftpd 主要配置文件：

/etc/sbin/vsftpd 服务文件

/etc/vsftpd/vsftpd.conf 配置文件

/etc/vsftpd.ftpusers 不能用于 ftp 登录的用户

/var/ftp 默认的匿名用户登录的目录

5、使用 vsftpd 创建一个 FTP 站点，允许匿名上传：

[root@desktopX ~]# **vim /etc/vsftpd/vsftpd.conf**

……

anonymous_enable = **YES** //允许匿名用户访问

local_enable = **YES** //允许本地用户访问

write_enable = **YES** //具有写权限

local_umask = **022** //本地用户创建文件戒目录的掩码

connect_from_port_20 = **YES** //开启 20 端口，关闭则是 NO

……

保存配置，重新启动服务

[root@desktop2 ~]# **service vsftpd restart**

登录你创建的 FTP 站点，查看是否能正常访问。

5、完善 FTP 服务器的功能，允许匿名用户具有写权限（上传/创建目录）：

[root@desktopX ~]# **vim /etc/vsftpd/vsftped.conf**

……

添加以下内容

anon_upload_enable = **YES**

anon_mkdir_write_enable **= YES**

anon_world_readable_only **= NO** //允许匿名帐号写目录

……

保存配置，重新启动服务

登录你创建的 FTP 站点，查看是否能正常访问。

6、完善 FTP 服务器的功能，屏蔽本地所有用户浏览其他目录的权限（除了家目录，匿名用户本身只能访问家目录）：

[root@desktopX ~]# **vim /etc/vsftpd/vsftped.conf**

……

修改以下内容

chroot_local_user = **YES**

……

保存配置，重新启动服务

登录你创建的 FTP 站点，查看是否能正常访问。

7、完善 FTP 服务器的功能，屏蔽部分本地用户浏览其他目录的权限：

[root@desktopX ~]# **vim /etc/vsftpd.chroot_list** //添加需要屏蔽的用户并保存

……

[root@desktopX ~]# **vim /etc/vsftpd/vsftpd.conf**

……

修改以下内容

chroot_local_user **= NO**

chroot_list_enable **= YES**

chroot_list_file = **/etc/ vsftpd.chroot_list**

8、更改 FTP 服务器的性能选项：

[root@desktopX ~]# **vim /etc/vsftped.conf**

……

idle_session_timeout = **600**

data_connection_timeout = **120**

local_max_rate = **50000** //本地用户的最高速率

anon_max_rate = **30000** //匿名用户的最高速率

……

[root@desktopX ~]# vim /etc/xinetd.d/vsftpd

……

修改以下内容

only_from = **192.168.1.1|192.168.1.0/24** //只接收来至某 ip 戒网段

no_access = **192.168.3.2|192.168.3.0/24** //拒绝接收来至某 ip 戒网段

access_times = **8:00-17:00** //设置访问时间
instances = **200** //设置最大连接数
per_source = **5** //设置每个 ip 可有几个连接

登录你创建的 FTP 站点，查看是否能正常访问。

本次实验到此结束。

实验 24：配置 Samba 服务器

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

掌握 samba 的配置流程，实现 Linux 与 Windows 的互访。

实验背景：

你的工作站上运行的是 linux 的操作系统，你有个使用 windows 的同事想通过局域网共享的方式与你的工作站进行互联，共享资料。你发现传统的共享方式无法建立 windows 与 linux 的连接关系，现在你需要借助 samba 服务器建立 windows 与 linux 的通道。

实验要求：

1、安装 samba 服务器

2、将创建好的 LVM 卷进行扩展，容量扩展至 200MB

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、安装 samba 服务：

运行 samba 需要以下软件包

samba-common-3.0.33-3.14.el5

samba-3.0.33-3.14.el5

system-config-samba-1.2.41-5.el5

samba-swat-3.0.33-3.14.el5 （以 web 页面管理 samba）

samba-client-3.0.33-3.14.el5

我们使用 yum 来安装 samba

[root@desktop2 ~]# **yum install samba***

2、启动 samba 服务

[root@desktop2 ~]# **/etc/init.d/smb start**

3、查看/etc/samba/smb.comf

[root@ desktop2 ~]# **cat /etc/samba/smb.comf**

#=================Global Settings=================

[global] //Global 配置区域如下，此区域配置对所有的共享文档生效

……

#===============Share Definitions=================

//共享文档配置区域

[homes] //家目录共享配置区域

……

[printers] //打印机共享配置区域

……

注意：在配置之前我们最好先备份一下主配置文件，以免以后文件损坏后好恢复。

*[root@ desktop2 samba]# **cp -a smb.conf smb.conf.bak***

4、创建一个共享目录 share，并赋予其他用户写入的权限：

[root@ desktop2 ~]# **cd /**

[root@ desktop2 ~]# **mkdir share**

[root@ desktop2 ~]# **chmod 777 share**

[root@ desktop2 ~]# **ll | grep share**

drwxrwxrwx 2 root root 4096 Sep 30 21:08 share

[root@ desktop2 ~]# **chmod 1777 share** //设置冒险位

You have new mail in /var/spool/mail/root

[root@ desktop2 ~]# **ll | grep share**

drwxrwxrwt 2 root root 4096 Sep 30 21:08 share

注意：设置冒险位的目的是为了防止用户误删除其他用户的文件。

5、搭建一个基本的 samba 服务器，允许匿名访问：

[root@ desktop2 ~]# **vim /etc/samba/smb.comf**

#================Global Settings================

……

workgroup = jiancelinux.com //配置所属的工作组

server string = samba for jiancelinux.com //配置服务器的描述

; netbios name = MYSERVER

; interfaces = lo eth0 192.168.12.2/24 192.168.13.2/24

; hosts allow = 127. 192.168.12. 192.168.13. //配置允许访问的网段，默认是允许所有

……

#=============Standalone Server Options=============

……

security = share //默认为 user，需要系统帐户访问

……

[share] //服务器共享文档名称

comment = share of jiancelinux //服务器说明

path = /share //共享文档路径

public = yes //允许匿名访问

writable = yes //允许写入，目录要有其他用户写入权限

printable = no //共享打印服务

……

6、在 share 目录里创建一个文件，测试 samba 服务：

[root@desktop2 ~]# **touch /share/test**

[root@desktop2 ~]# **ll /share/test**

-rw-r--r-- 2 root root 4096 Sep 30 21:08 /share/test

[root@desktop2 ~]# **service smb restart** //重新启动 samba 服务

……

[root@desktop2 ~]# **service smb reload** //重新载入 smb.conf

……

接下来就可以使用 Windows 系统来访问了。

7、搭建一个基于用户配置文件的 samba 服务器：

[root@ desktop2 ~]# **vim /etc/samba/smb.comf**

#================Global Settings================

……

workgroup = jiancelinux.com
include = /etc/samba/%.smb.conf //读取用户的单独配置文件
server string = samba for jiancelinux.com
……
#=============Standalone Server Options=============
……
security = user //设置为需要系统帐户访问

8、添加名为 jiance 的用户和用户组：
[root@desktop2 samba]# **groupadd jiance**
[root@desktop2 samba]# **useradd -G jiance jiance**

9、建立 samba 用户，并添加密码（输入的密码不显示）：
[root@desktop2 samba]# **smbpasswd -a jiance** //为用户 jiance 加上 samba 密码
New SMB password：
Retype new SMB password：
Added user jiance.

10、复制用户单独配置文件，直接复制 smb.conf 为用户配置文件即可：
[root@desktop2 samba]# **cp -a smb.conf.bak jiance.smb.conf**
[root@desktop2 samba]# **vim jiance.smb.conf**
……
[jiance]
comment = jiance
path = /jiance
public = no
writable =yes
valid user = jiance //只允许建策来访问本目录
……

11、在根目录下建立经理的共享目录并把属主和属组都设为 jiance：
[root@desktop2 /]# **mkdir jiance**
[root@desktop2 samba]# **chown jiance:jiance jiance**

12、重启 samba 服务，检查是否能正常使用：
[root@desktop2 ~]# **service smb restart** //重新启动 samba 服务
……
[root@desktop2 ~]# **service smb reload** //重新载入 smb.conf

接下来就可以使用客户端来访问了(访问之前要清楚共享访问记录)。

本次实验到此结束。

实验 25：实施网络共享(NFS)服务

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

掌握 nfs 的使用方法，建立网络共享。

实验背景：

公司组建了一个小型局域网，这个局域网中的用户都是用 linux 的操作系统。现在公司想让这个局域网中的工作站实现网络共享，这时候你需要使用 NFS 来实现网络共享的功能。

实验要求：

1、安装 nfs 服务

2、创建 nfs 共享目录

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、安装 nfs 的软件包：

[root@desktopX ~]# **yum -y install nfs-utils**

注意：nfs-utils 软件包提供了 nfs 和 nslookup 服务。

3、设置 nfs 开机自动运行：

[root@desktopX ~]# **chkconfig nfs on**

[root@desktopX ~]# **chkconfig nfslock on**

4、查看 RPC 服务是否正常运行：

[root@desktopX ~]# **rpcinfo -p**

[root@desktopX ~]# **showmount -e localhost**

5、创建一个用户并且配置 NFS 共享他的主目录，共享给 example.com 读写权限：

[root@desktopX ~]# **useradd nfstest**

[root@desktopX ~]# **vim /etc/exports**

……

应该有类似一下的行

/home/nfstest　*.example.com(ro,sync)

6、开启 nfs 服务和 nsflock 服务：

[root@desktopX ~]# **service nfs start**

[root@desktopX ~]# **service nfslock start**

7、观察哪个 RPC 服务正在运行，查看是否在导出的/home/nfstest 目录中:

[root@desktopX ~]# **rpcinfo -p**

[root@desktopX ~]# **showmount -e localhost**

8、再找一台工作站，挂载彼此的共享。尝试用 root 和 nfstest 向读取共享中的内容，并在其中写如内容：

[root@desktopX ~]# **mkdir /home/remote**

[root@desktopX ~]# **mount desktopY:/home/nfstest /home/remote**

注意：如果 nfstest 用户的 UID 和 GID 与你使用的工作站匹配，就应该能看到 NFS 共享的读写权限。若不匹配，就不能访问目录。因为设置了默认的 root_squash 选项，所以 root 用户应该不能访问 NFS 共享。

实验 26：配置 DNS 服务器

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。本次实验需要关闭防火墙及 SELinux。

实验目标：

掌握 DNS 的相关原理，熟悉 BIND 的配置，能够监听网络，满足访问控制要求。

实验背景：

公司使用了一个员工信息系统。以前公司员工都是使用直接使用 IP 地址来访问的。但是大家都觉得记忆 IP 相对比较麻烦。现在公司管理层让你为你们公司架设一台 DNS 服务器为这个系统绑定一个内部的域名，添加正向，反向解析条目，使系统可以通过域名来访问。

实验要求：

1、安装 DNS 服务器

2、创建 DNS 主要域名服务器

2、添加正、反向查找的解析规则

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、安装 DNS 服务：

运行 DNS 需要以下软件包

bind-*

我们使用 yum 来安装 DNS

[root@desktopX ~]# **yum -y install bind***

2、启动 DNS 服务：

[root@desktopX ~]# **service named start**

[root@desktopX ~]# **chkconfig named on** //可以设置为开机自动运行服务

注意：DNS 主要配置文件：

/etc/named.conf DNS 服务器的全局配置文件

/etc/named.rfc1912.zones DNS 服务器的区域配置文件

/var/named/named.local 用于本地回环地址解析的反向解析文件

/var/named/localhost.zone 用于本地回环地址解析的正向解析文件

/var/named/domainname.zone 用户建立的本地主机区域数据库文件

3、创建 DNS 主要域名服务器：

[root@desktopX ~]# **cd /var/named/chroot/etc**

[root@desktopX etc]# **mv named.caching-nameserver.conf named.conf**

//named.caching-nameserver.conf 可以看成是一份 named.conf 的模板，真正起作用的还是 name.conf

[root@desktopX etc]# **vim named.conf**

options {

```
        listen-on port 53                   {any;};
        listen-on-v6 port 53    {::1; };
        directory                           "/var/named";
        dump-file                           "/var/named/data/cache_dump.db";
        statistics-file             "/var/named/data/named_stats.txt";
        memstatistics-file          "/var/named/data/named_mem_stats.txt";
                allow-query             {any;};
                allow-query-cache       {any;};
};
logging {
        channel default_debug {
        file                                "data/named.run";
        severity dynamic;
        };
};
view localhost_resolver {
        match-clients                               {any;};
        match-destinations              {any;};
        recursion yes;
        include                             "/etc/named.rfc1912.zones";
};
```

上面的改动实际上就是把对本机(127.0.0.1)的设置放开到对任何机器(any)，修改完毕后保存退出。

注意：named.caching-nameserver.conf 是 DNS 缓冲的配置文件，而 named.conf 是 DNS 的主配置文件。一般的一个 DNS 解析过程遵循如下的顺序，首先查找本地主机 host 文件，如果没有查找缓冲是否有，如果缓冲没有，是用网卡设定的 DNS 进行转发或者递归或迭代来查询。查询的结果会存放缓冲一份，以便于下次查询可以快速响应。

4、添加正向查找、反向查找及多域查找规则：

在文件的末尾添加正反解析部分，即指定当进行正向域名解析时，查找的用户自定义正向解析文件是”zheng”，当进行反向域名解析时，查找的用户自定义反向解析文件是”fan”。

```
[root@desktopX etc]# vim named.rfc1912.zones
zone “jiance.com” IN {
   type master;
   file “zheng”;
}
zone “0.168.192.in-addr.arpa” IN {
   type master;
   file “fan”;
}
[root@desktopX etc]# cd /var/named/chroot/var/named
[root@desktopX etc]# ll
total 72
drwxrwx--- 2        named  named  4096    Jul      1        16:19 data
-rw-r-----          1      root   named   198             Jul      30              2011
   localdomain.zone
-rw-r-----          1      root           named   195             Jul      30      2011
   localhost.zone
-rw-r-----          1      root           named   427             Jul      30      2011
   named.broadcast
```

```
-rw-r-----  1  root  named  1892  Jul  30  2011  named.ca
-rw-r-----  1  root  named  424   Jul  30  2011  named.ip6.local
-rw-r-----  1  root  named  426   Jul  30  2011  named.local
-rw-r-----  1  root  named  427   Jul  30  2011  named.zero
drwxrwx---  2  named named  4096  Jul  27  2011  slaves
[root@desktopX etc]# cp localhost.zone zheng
[root@desktopX etc]# cp named.local fan
[root@desktopX etc]# cd /var/named
[root@desktopX named]# ln –s /var/named/chroot/var/named/zheng zheng
[root@desktopX named]# ln –s /var/named/chroot/var/named/fan fan
[root@desktopX named]# chown named:named *
```

上述操作分别添加正、反向解析文件，并修改文件的所有者和所有组为 named 的系统用户。在/var/named 下创建接正、反向解析文件的链接到/var/named/chroot/var/named/。

5、修改正、反向解析文件：

```
[root@desktopX named]# vim /var/named/chroot/var/named/zheng
$TTL    86400
@               IN      SOA     jiance.com.             root (
                                        42              ; serial (d. adams)
                                        3H              ; refresh
                                        15M                     ; retry
                                        1W              ; expiry
                                        1D )                    ; minimum

                IN NS                   njjsxy.com.
                IN A                            127.0.0.1
                IN AAAA                 ::1
dns                             IN A                            192.168.0.100
www             IN A                            192.168.0.200
[root@desktopX named]# vim /var/named/chroot/var/named/fan
$TTL    86400
@               IN      SOA             jiance.com.     root.jiance.com. (
                                        1997022700              ; Serial
                                        28800                   ; Refresh
                                        14400                   ; Retry
                                        3600000                 ; Expire
                                        86400 )                 ; Minimum
                IN  NS          jiance.com.
100                             IN  PTR             dns.jiance.com.
200                             IN  PTR             www.jiance.com.
```

6、重启 DNS 服务：

```
[root@desktopX named]# service named configtest
[root@desktopX named]# service named restart
```

在客户端的/etc/resolv.conf 中指向 DNS 服务器地址就可以完成域名解析。

本次实验到此结束。

实验 27：配置邮件服务器

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证，能使用 DHCP 联网的系统，能连接到 Internet。

实验目标：

使用 postfix 邮件服务，能够接收和发送邮件，会使用 mail 命令，通过命令行发送邮件。

实验背景：

在你的系统上安装 postfix，有能连接到外网的网络环境，最好使用 dhcp 获得网络信息。这里假设邮件服务器的 IP 是 192.168.2.14。

实验要求：

1、搭建一个 postfix 邮件服务器

2、能使用 mail 命令发送邮件

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、安装 postfix 邮件服务器：

[root@desktopX ~]# **yum install postfix -y**

3、修改配置文件：

[root@desktopX ~]# **vim /etc/postfix/main.cf**

……

把 **inet_interfaces =** 改为 **inet_interfaces = all**

其他行的内容不用去管

4、启动 postfix 服务：

[root@desktopX ~]# **/etc/init.d/postfix start**

5、查看本机网络是否监听 25 端口：

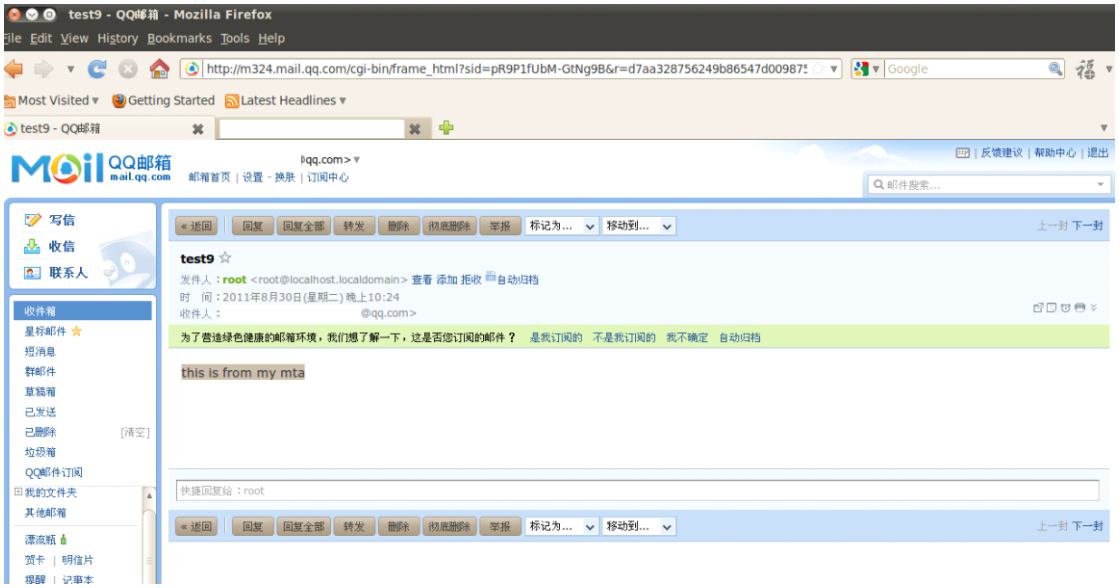
[root@desktopX ~]# **netstat -tupln|grep 25**

6、使用 mail 命令来发送邮件至你的邮箱：

[root@desktopX ~]# **mail -s 'test9' xxxxxxxxx@qq.com**

this is from my mta

7、登录你的邮箱可以查看到如下情况（只要能联网都能发送成功）：


test9 - QQ邮箱 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Most Visited
Getting Started
Latest Headlines
test9 - QQ邮箱
QQ邮箱
mail.qq.com
邮箱首页 | 设置 - 换肤 | 订阅中心
反馈建议 | 帮助中心 | 退出
写信
收信
联系人
收件箱
星标邮件
短消息
群邮件
草稿箱
已发送
已删除
垃圾箱
QQ邮件订阅
我的文件夹
其他邮箱
漂流瓶
贺卡 | 明信片
« 返回
回复
回复全部
转发
删除
彻底删除
举报
标记为...
移动到...
上一封 下一封
test9
发件人：root <root@localhost.localdomain> 查看 添加 拒收 自动归档
时 间：2011年8月30日(星期二) 晚上10:24
收件人：
@qq.com>
为了营造绿色健康的邮箱环境，我们想了解一下，这是否您订阅的邮件？ 是我订阅的 不是我订阅的 我不确定 自动归档
this is from my mta
快捷回复给：root

本次实验到此结束。

实验 28：实施 Web(HTTP)服务

实验环境：

安装了 Red Hat Enterprise Linux 6.0 可运行系统，并且是成功验证系统。

实验目标：

掌握 web 的相关服务内容，学会搭建 WEB 服务器，并能够对其进行配置。

实验背景：

公司需要部署了一个员工信息系统。公司管理人员要求你搭建一个 Web 服务器，是这个系统可以正常访问，同时要求能够使用虚拟主机且虚拟主机里只能由特定用户访问。

实验要求：

1、搭建 WEB 服务器

2、配置 WEB 服务器

实验详解：

1、以 root 用户的身份登录系统。如果你使用的是图形化环境，点击[应用程序(Applications)]->[附件(System Tools)]->[终端(Terminal)]来打开终端：

2、安装 httpd 的软件包：

[root@desktopX ~]# **yun install httpd -y**

3、修改 hosts 文件,建立简单的域名解析：

[root@desktopX ~]# **vim /etc/hosts**

……

添加以下内容

127.0.0.1 www.jiance.net

127.0.0.1 www2.jiance.net

4、创建简单的 WEB 页面：

[root@desktopX ~]# **vim /var/www/html/index.html**

this is www.jiance.net

[root@desktopX ~]# **vim /var/www/html2/index.html**

this is www2.jiance.net

5、修改 WEB 配置文件：

[root@desktopX ~]# **vim /etc/httpd/conf/httpd.conf**

……

按以下内容修改配置文件

```
NameVirtualHost *:80
<Virtualhost *:80>
    ServerName www2.jiance.net
    DocumentRoot /var/www/html2
</Virtualhost>
<Virtualhost *:80>
    ServerName www2.jiance.net
    DocumentRoot /var/www/html2
<Directory /var/www/html/www13>
AuthName server-password
```

AuthType basic
AuthUserfile /etc/httpd/conf/.htpasswd
Require valid-user
</Directory>
</Virtualhost>

6、创建指定用户和密码：
[root@desktopX ~]# **htpasswd -mc /etc/httpd/conf/.htpasswd tom**
//把所有账号和密码加入到.htpasswd 文件中去
cut -d: -f1-2 /etc/shadow > /etc/httpd/conf/.htpasswd

7、重启 web 服务：
[root@desktopX ~]# /etc/init.d/httpd restart
[root@desktopX ~]# **elinks www2.jiance.net**
……
现在访问就需要账号和密码

[root@desktopX ~]# **elinks www.jiance.net**
访问 www.jiance.net 则不需要密码

本次实验到此结束。

附录：Vim 常用命令表

命令模式	光标移动
h 或　向左方向键	光标向左移动一个字符
j 或　向下方向键	光标向下移动一个字符
k 或　向上方向键	光标向上移动一个字符
l 或　向右方向键	光标向右移动一个字符
Ctrl+f	屏幕向前翻一页（常用）
Ctrl+b	屏幕向后翻一页（常用）
Ctrl+d	屏幕向前翻半页
Ctrl+u	屏幕向前翻半页
+	光标移动到非空格符的下一列
-	光标移动到非空格符的上一列
n<space>	按下数字后再按空格键，光标会向右移动这一行的 n 个字符。例如 20<space>,则光标会向右移动 20 个字符
0(HOME)	(是数字 0)动到这一行的第一个字符处（常用）
$(END)	移动到这一行的最后一个字符处（常用）
H	光标移动到这个屏幕最上方的那一行
M	光标移动到这个屏幕中央的那一行
L	光标移动到这个屏幕最下方的那一行
G	光标移动到文件的最后一行
nG	移动到这个文件的第 n 行。例如 20G，则会移动到这个文件的第 20 行（可配合:set　nu）
n<Enter>	光标向下移动 n 行(常用)
命令模式	**查找与替换**
/word	在光标之后查找一个名为 word 的字符串（常用）
?word	在光标之前查找一个名为 word 的字符串
:n1,n2s/word1/word2/g	在第 n1 与 n2 行之间查找 word1 这个字符串，并将该字符串替换为 word2（常用）
:1,$s/　word1/word2/g	在第一行与最后一行之间查找 word1 这个字符串，并将该字符串替换为 word2（常用）
:1,$s/　word1/word2/gc	在第一行与最后一行之间查找 word1 这个字符串，并将该字符串替换为 word2，且在替换前显示提示符让用户确认(conform)（常用）
一般模式	**删除、复制与粘贴**
x,X	X 为向后删除一个字符，X 为向前删除一个字符（常用）
Nx	向后删除 n 个字符
Dd	删除光标所在的那一整行（常用）
Ndd	删除光标所在列的向下 n 列，例如，20dd 则事删除 20 列（常用）
d1G	删除光标所在行到第一行的所有数据
dG	删除光标所在列到最后一行的所有数据
Yy	复制光标所在行（常用）
Nyy	复制光标所在列的向下 n 列，例如，20yy 则是复制 20 列（常用）
y1G	复制光标所在列到第一列的所有数据
yG	复制光标所在列到最后一列的所有数据
p,P	p 为复制的数据粘贴在光标下一列，P 则为粘贴在光标上一列(常用)
J	将光标所在列与下一列的数据结合成一列
U	恢复前一个动作（undo）
编辑模式	

i,I	插入：在当前光标所在处插入输入的文字，已存在
a,A	添加：由当前光标所在处的下一个字符开始输入，已存在的字符会向后退（常用）
o,O	插入新的一行：从光标所在行的下一行行首开始输入字符（常用）
r,R	替换：r 会替换光标所指的那一个字符；R 会一直替换光标所指的文字，直到按下 Esc 为止（常用）
Esc	退出编辑模式，回到一般模式（常用）
命令行模式	
:w	将编辑的数据写入硬盘文件中（常用）
:w!	若文件属性为只读，强制写入该文件
:q	退出 vi（常用），快捷方式为 SHIFT+ZZ
:q!	若曾修改过文件，又不想保存，使用！为强制退出不保存文件，快捷方式为 SHIFT+ZQ
:wq	保存后退出，若为:wq!，则为强制保存后退出（常用）
:w[filename]	将编辑数据保存为另一个文件（类似另存新文档）
:r[filename]	在编辑的数据中，读入另一个文件的数据。即将 filename 这个文件内容加到光标所在行的后面
:set nu	显示行号，设定之后，会在每一行的前面显示该行的行号
:set nonu	与 ser nu 相反，为取消行号
:set nohlsearch	可取消高亮,可编辑/etc/vimrc 来编辑取消所有高亮
n1,n2 w[filename]	将 n1 到 n2 的内容保存为 filename 这个文件