

NSD ADMIN DAY07

1. [案例1：新建一个逻辑卷](#)
2. [案例2：调整现有磁盘的分区](#)
3. [案例3：扩展逻辑卷的大小](#)
4. [案例4：查找并处理文件](#)
5. [案例5：Linux管理员 综合测试](#)

1 案例1：新建一个逻辑卷

1.1 问题

本例要求沿用前一天案例，使用分区 /dev/vdb1 构建 LVM 存储，相关要求如下：

1. 新建一个名为 systemvg 的卷组
2. 在此卷组中创建一个名为 vo 的逻辑卷，大小为180MiB
3. 将逻辑卷 vo 格式化为 EXT4 文件系统
4. 将逻辑卷 vo 挂载到 /vo 目录，并在此目录下建立一个测试文件 votest.txt，内容为“I AM KING.”

1.2 方案

LVM创建工具的基本用法：

01. vgcreate 卷组名 物理设备...
02. lvcreate -L 大小 -n 逻辑卷名 卷组名

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：创建卷组

- 1) 新建名为systemvg的卷组

01. [root@server0 ~]# vgcreate systemvg /dev/vdb1
02. Physical volume "/dev/vdb1" successfully created
03. Volume group "systemvg" successfully created

- 2) 确认结果

01. [root@server0 ~]# vgscan
02. Reading all physical volumes. This may take a while...
03. Found volume group "systemvg" using metadata type lvm2

[Top](#)

步骤二：创建逻辑卷

1) 新建名为vo的逻辑卷

```
01. [root@server0 ~]# lvcreate -L 180MiB -n vo systemvg
02. Logical volume "vo" created
```

2) 确认结果

```
01. [root@server0 ~]# lvscan
02. ACTIVE          '/dev/systemvg/vo' [180.00 MiB] inherit
```

步骤三：格式化及挂载使用

1) 格式化逻辑卷/dev/systemvg/vo

```
01. [root@server0 ~]# mkfs.ext4 /dev/systemvg/vo
02. .. ..
03. Allocating group tables: done
04. Writing inode tables: done
05. Creating journal (4096 blocks): done
06. Writing superblocks and filesystem accounting information: done
```

2) 挂载逻辑卷/dev/systemvg/vo

```
01. [root@server0 ~]# mkdir /vo //创建挂载点
02. [root@server0 ~]# mount /dev/systemvg/vo /vo //挂载
03. [root@server0 ~]# df -hT /vo/ //检查结果
04. Filesystem      Type  Size  Used Avail Use% Mounted on
05. /dev/mapper/systemvg-vo ext4 171M 1.6M 157M 1% /vo
```

3) 访问逻辑卷/dev/systemvg/vo

```
01. [root@server0 ~]# cat /vo/votest.txt
02. I AM KING.
```

[Top](#)

2 案例2：调整现有磁盘的分区

2.1 问题

本例要求沿用前一天案例，对磁盘/dev/vdb的分区表进行调整，要求如下：不更改原有分区，利用剩余空间新增三个分区，大小依次为：500MiB、2000MiB、512MiB

然后再基于刚建立的 2000MiB 分区构建新的 LVM 存储：

1. 新的逻辑卷命名为 database，大小为50个物理扩展单元（Physical Extent），属于 datastore 卷组
2. 在 datastore 卷组中的所有逻辑卷，其物理扩展单元（Physical Extent）的大小为16MiB
3. 使用 EXT3 文件系统对逻辑卷 database 格式化，此逻辑卷应该在开机时自动挂载到 /mnt/database 目录

2.2 方案

创建卷组时，可以通过-s选项指定PE的大小。

在给新建的逻辑卷分配空间时，空间大小只能是PE大小的倍数。

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：调整现有磁盘分区

- 1) 新建扩展分区（使用剩余可用空间）

```
01. [root@server0 ~]# fdisk /dev/vdb
02.
03. Command (m for help): p                //确认原有分区表
04. .. ..
05.   Device Boot      Start         End      Blocks    Id System
06.   /dev/vdb1            2048        411647        204800    8e  Linux LVM
07.   /dev/vdb2           411648        4507647       2048000    83   Linux
08.   /dev/vdb3           4507648        6555647       1024000    83   Linux
09.
10. Command (m for help): n                //新建分区
11. Partition type:
12.   p   primary (3 primary, 0 extended, 1 free)
13.   e   extended
14. Select (default e): e                    //类型指定为e（扩展分区）
15. Selected partition 4                      //只有一个可用编号，自动选取
16. First sector (6555648-20971519, default 6555648):      //起始位置默认
17. Using default value 6555648
18. Last sector, +sectors or +size{K,M,G} (6555648-20971519, default 20971519):
19. Using default value 20971519              //结束位置默认
20. Partition 4 of type Extended and of size 6.9 GiB is set
21.
22. Command (m for help): p
```

[Top](#)

```

23.  .. ..
24.      Device Boot      Start         End      Blocks   Id  System
25.      /dev/vdb1           2048       411647       204800    8e  Linux LVM
26.      /dev/vdb2       411648       4507647      2048000    83  Linux
27.      /dev/vdb3       4507648       6555647      1024000    83  Linux
28.      /dev/vdb4       6555648      20971519      7207936     5  Extended

```

2) 在扩展分区中新建3个逻辑分区

创建第1个逻辑卷（由于主分区编号已用完，分区类型自动选l逻辑分区）：

```

01.  Command (m for help): n
02.  All primary partitions are in use
03.  Adding logical partition 5                //分区编号5
04.  First sector (6557696-20971519, default 6557696):      //起始位置默认
05.  Using default value 6557696
06.  Last sector, +sectors or +size{K,M,G} (6557696-20971519, default 20971519): +500
07.                                          //结束位置默认
08.  Partition 5 of type Linux and of size 500 MiB is set

```

创建第2个逻辑卷：

```

01.  Command (m for help): n
02.  All primary partitions are in use
03.  Adding logical partition 6                //分区编号6
04.  First sector (7583744-20971519, default 7583744):      //起始位置默认
05.  Using default value 7583744
06.  Last sector, +sectors or +size{K,M,G} (7583744-20971519, default 20971519): +200
07.                                          //结束位置默认
08.  Partition 6 of type Linux and of size 2 GiB is set

```

创建第3个逻辑卷：

```

01.  Command (m for help): n
02.  All primary partitions are in use
03.  Adding logical partition 7                //分区编号7
04.  First sector (11681792-20971519, default 11681792):    //起始位置默认

```

[Top](#)

05. Using default value 11681792
06. Last sector, +sectors or +size{K,M,G} (11681792-20971519, default 20971519): +5:
07. //结束位置默认
08. Partition 7 of type Linux and of size 512 MiB is set

根据预计的用途调整分区类型（可选）：

01. Command (m for help): t //修改
02. Partition number (1-7, default 7): 5 //第5个分区
03. Hex code (type L to list all codes): 8e //类型为8e (LVM)
04. Changed type of partition 'Linux' to 'Linux LVM'
- 05.
06. Command (m for help): t //修改
07. Partition number (1-7, default 7): 6 //第6个分区
08. Hex code (type L to list all codes): 8e //类型为8e (LVM)
09. Changed type of partition 'Linux' to 'Linux LVM'
- 10.
11. Command (m for help): t //修改
12. Partition number (1-7, default 7): 7 //第7个分区
13. Hex code (type L to list all codes): 82 //类型为82 (交换分区)
14. Changed type of partition 'Linux' to 'Linux swap / Solaris'

确认分区结果并保存：

01. Command (m for help): p
02.
03. Device Boot Start End Blocks Id System
04. /dev/vdb1 2048 411647 204800 8e Linux LVM
05. /dev/vdb2 411648 4507647 2048000 83 Linux
06. /dev/vdb3 4507648 6555647 1024000 83 Linux
07. /dev/vdb4 6555648 20971519 7207936 5 Extended
08. /dev/vdb5 6557696 7581695 512000 8e Linux LVM
09. /dev/vdb6 7583744 11679743 2048000 8e Linux LVM
10. /dev/vdb7 11681792 12730367 524288 82 Linux swap / Solaris
- 11.
12. Command (m for help): w //保存退出
13. The partition table has been altered!
- 14.

[Top](#)

15. Calling `ioctl()` to re-read partition table.
- 16.
17. WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
18. The kernel still uses the old table. The new table will be used at
19. the next reboot or after you run `partprobe(8)` or `kpartx(8)`
20. Syncing disks. //提示重启

3) 刷新分区表

01. `[root@server0 ~]# partprobe /dev/vdb`
02. `[root@server0 ~]# reboot`

步骤二：新建卷组、逻辑卷

1) 新建卷组datastore，指定PE大小为16MiB

01. `[root@server0 ~]# vgcreate -s 16MiB datastore /dev/vdb6`
02. Volume group "datastore" successfully created
03. `[root@server0 ~]# vgscan` //确认新建的卷组
04. Reading all physical volumes. This may take a while...
05. Found volume group "systemvg" using metadata type lvm2
06. Found volume group "datastore" using metadata type lvm2

2) 新建逻辑卷database，大小设置为50个PE

01. `[root@server0 ~]# lvcreate -l 50 -n database datastore`
02. Logical volume "database" created
03. `[root@server0 ~]# lvscan` //确认新建的逻辑卷
04. ACTIVE `'/dev/systemvg/vo' [180.00 MiB] inherit`
05. ACTIVE `'/dev/datastore/database' [800.00 MiB] inherit`

步骤三：格式化及使用逻辑卷

1) 格式化逻辑卷/dev/datastore/database

01. `[root@server0 ~]# mkfs.ext3 /dev/datastore/database`
02. ...
03. Allocating group tables: done

[Top](#)

04. Writing inode tables: done
05. Creating journal (4096 blocks): done
06. Writing superblocks and filesystem accounting information: done

2) 配置开机挂载

01. [root@server0 ~]# mkdir /mnt/database //创建挂载点
02. [root@server0 ~]# vim /etc/fstab
03.
04. /dev/datastore/database /mnt/database ext3 defaults 0 0

3) 验证挂载配置

01. [root@server0 ~]# mount -a
02. [root@server0 ~]# df -hT /mnt/database/ //确认挂载点设备
03. Filesystem Type Size Used Avail Use% Mounted on
04. /dev/mapper/datastore-database ext3 772M 828K 715M 1% /mnt/database

3 案例3：扩展逻辑卷的大小

3.1 问题

本例要求沿用练习一，将逻辑卷 vo 的大小调整为 300MiB，要求如下：

1. 原文件系统中的内容必须保持完整
2. 必要时可使用之前准备的分区 /dev/vdb5 来补充空间
3. 注意：分区大小很少能完全符合要求的大小，所以大小在270MiB和300MiB之间都是可以接受的

3.2 方案

对于已经格式化好的逻辑卷，在扩展大小以后，必须通知内核新大小。

如果此逻辑卷上的文件系统是EXT3/EXT4类型，需要使用resize2fs工具；

如果此逻辑卷上的文件系统是XFS类型，需要使用xfs_growfs。

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：确认逻辑卷vo的信息

- 1) 找出逻辑卷所在卷组

[Top](#)

01. [root@server0 ~]# lvscan

```
02.    ACTIVE      '/dev/systemvg/vo' [180.00 MiB] inherit
03.    ACTIVE      '/dev/datastore/database' [800.00 MiB] inherit
```

2) 查看该卷组的剩余空间是否可满足扩展需要

```
01.    [root@server0 ~]# vgdisplay systemvg
02.    --- Volume group ---
03.    VG Name          systemvg
04.    System ID
05.    Format            lvm2
06.    Metadata Areas    1
07.    Metadata Sequence No 2
08.    VG Access         read/write
09.    VG Status         resizable
10.    MAX LV            0
11.    Cur LV            1
12.    Open LV           0
13.    Max PV            0
14.    Cur PV            1
15.    Act PV            1
16.    VG Size           196.00 MiB           //卷组总大小
17.    PE Size           4.00 MiB
18.    Total PE          49
19.    Alloc PE / Size   45 / 180.00 MiB
20.    Free PE / Size   4 / 16.00 MiB       //剩余空间大小
21.    VG UUID           czp8IJ-jihS-Ddoh-ny38-j521-5X8J-gqQfUN
```

此例中卷组systemvg的总大小都不够300MiB、剩余空间才16MiB，因此必须先扩展卷组。只有剩余空间足够，才可以直接扩展逻辑卷大小。

步骤二：扩展卷组

1) 将提前准备的分区/dev/vdb5添加到卷组systemvg

```
01.    [root@server0 ~]# vgextend systemvg /dev/vdb5
02.    Physical volume "/dev/vdb5" successfully created
03.    Volume group "systemvg" successfully extended
```

[Top](#)

2) 确认卷组新的大小


```

01. [root@server0 ~]# vgdisplay systemvg
02. --- Volume group ---
03. VG Name          systemvg
04. .. ..
05. VG Size          692.00 MiB           //总大小已变大
06. PE Size         4.00 MiB
07. Total PE        173
08. Alloc PE / Size  45 / 180.00 MiB
09. Free PE / Size  128 / 512.00 MiB     //剩余空间已达512MiB
10. VG UUID          czp8IJ-jihS-Ddoh-ny38-j521-5X8J-gqQfUN

```

步骤三：扩展逻辑卷大小

1) 将逻辑卷/dev/systemvg/vo的大小调整为300MiB

```

01. [root@server0 ~]# lvextend -L 300MiB /dev/systemvg/vo
02. Extending logical volume vo to 300.00 MiB
03. Logical volume vo successfully resized

```

2) 确认调整结果

```

01. [root@server0 ~]# lvscan
02. ACTIVE          '/dev/systemvg/vo' [300.00 MiB] inherit
03. ACTIVE          '/dev/datastore/database' [800.00 MiB] inherit

```

3) 刷新文件系统大小

确认逻辑卷vo上的文件系统类型：

```

01. [root@server0 ~]# blkid /dev/systemvg/vo
02. /dev/systemvg/vo: UUID="d4038749-74c3-4963-a267-94675082a48a" TYPE="ext4"

```

选择合适的工具刷新大小：

```

01. [root@server0 ~]# resize2fs /dev/systemvg/vo
02. resize2fs 1.42.9 (28-Dec-2013)
03. Resizing the filesystem on /dev/systemvg/vo to 307200 (1k) blocks.

```

[Top](#)

04. The filesystem on `/dev/systemvg/vo` is now **307200** blocks long.

确认新大小（约等于300MiB）：

```
01. [root@server0 ~]# mount /dev/systemvg/vo /vo/
02. [root@server0 ~]# df -hT /vo
03. Filesystem      Type  Size  Used Avail Use% Mounted on
04. /dev/mapper/systemvg-vo ext4 287M 2.1M 266M 1% /vo
```

4 案例4：查找并处理文件

4.1 问题

本例要求采用不少于两种方法完成以下任务：

1. 找出所有用户 `student` 拥有的文件
2. 把它们拷贝到 `/root/findfiles/` 文件夹中

4.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：确认能找到指定的文件

- 1) 确认新版内核的下载地址

```
01. [root@server0 ~]# find / -user student -type f
02. find: '/proc/1853/task/1853/fdinfo/6': 没有那个文件或目录
03. find: '/proc/1853/fdinfo/6': 没有那个文件或目录
04. /var/spool/mail/student
05. /home/student/.bash_logout
06. /home/student/.bash_profile
07. /home/student/.bashrc
08. /home/student/.ssh/authorized_keys
09. /home/student/.config/gnome-initial-setup-done
10. /home/student/.config/monitors.xml
```

对于上述操作中出现的 `/proc` 信息忽略即可。

步骤二：处理找到的文件

- 1) 创建目标文件夹

[Top](#)

```
01. [root@server0 ~]# mkdir /root/findfiles
```

2) 拷贝找到的文件到目标文件夹

以下两种方法任选一种：

```
01. [root@server0 ~]# find / -user student -type f -exec cp -p {} /root/findfiles/ \;  
02. ...  
03. 或者  
04. [root@server0 ~]# \cp -p $(find / -user student -type f) /root/findfiles/  
05. ...
```

3) 确认拷贝结果

```
01. [root@server0 ~]# ls -lhA /root/findfiles/  
02. 总用量 24K  
03. -rw-----, 1 student student 1.7K 7月 11 2014 authorized_keys  
04. -rw-r--r--, 1 student student 18 1月 29 2014 .bash_logout  
05. -rw-r--r--, 1 student student 193 1月 29 2014 .bash_profile  
06. -rw-r--r--, 1 student student 231 1月 29 2014 .bashrc  
07. -rw-r--r--, 1 student student 4 7月 11 2014 gnome-initial-setup-done  
08. -rw-r--r--, 1 student student 1.5K 7月 11 2014 monitors.xml  
09. -rw-rw----, 1 student mail 0 7月 11 2014 student
```

5 案例5：Linux管理员 综合测试

5.1 问题

根据本文提供的练习步骤完成所有练习案例。

5.2 方案

开始练习之前，先依次重置虚拟机环境。

```
01. [root@room9pc13 ~]# rht-vmctl reset classroom  
02. [root@room9pc13 ~]# rht-vmctl reset server
```

5.3 步骤

实现此案例需要按照如下步骤进行。

[Top](#)

步骤01：配置一个用户

案例概述：

创建一个名为alex的用户，用户ID是 3456。密码是flectrag

```
01. [root@server0 ~]# useradd -u 3456 alex
02. [root@server0 ~]# echo flectrag | passwd --stdin alex
```

步骤02：创建用户账号和组

案例概述：

创建下列用户、组以及和组的成员关系：

- 一个名为adminuser的组
- 一个名为natasha的用户，其属于adminuser，这个组是该用户的从属组
- 一个名为harry的用户，属于adminuser，这个组是该用户的从属组
- 一个名为sarah的用户，其在系统中没有可交互的shell，并且不是adminuser组的成员用户
- natasha、harry、和sarah的密码都要设置为flectrag

```
01. [root@server0 ~]# groupadd adminuser //添加组
02.
03. [root@server0 ~]# useradd -G adminuser natasha //添加用户
04. [root@server0 ~]# useradd -G adminuser harry
05. [root@server0 ~]# useradd -s /sbin/nologin sarah
06.
07. [root@server0 ~]# echo flectrag | passwd --stdin natasha //设置密码
08. [root@server0 ~]# echo flectrag | passwd --stdin harry
09. [root@server0 ~]# echo flectrag | passwd --stdin sarah
```

步骤03：配置文件 /var/tmp/fstab 的权限

案例概述：

拷贝文件/etc/fstab到/var/tmp/fstab，配置文件/var/tmp/fstab的权限：

- 文件/var/tmp/fstab的拥有者是root用户
- 文件/var/tmp/fstab属于root组
- 文件/var/tmp/fstab对任何人都不可执行
- 用户natasha 能够对文件/var/tmp/fstab执行读和写操作
- 用户harry 对文件/var/tmp/fstab既不能读，也不能写
- 所有其他用户（当前的和将来的）能够对文件/var/tmp/fstab进行读操作

```
01. [root@server0 ~]# cp /etc/fstab /var/tmp/fstab //复制文件
02. [root@server0 ~]# setfacl -m u:natasha:rw /var/tmp/fstab //添加个别用户权限
03. [root@server0 ~]# setfacl -m u:harry:- /var/tmp/fstab
```

[Top](#)

步骤04：配置一个 cron 任务

案例概述：

为用户natasha配置一个定时任务，每天在本地时间14:23时执行以下命令：

/bin/echo hiya

解题参考：

```
01. [root@server0 ~]# systemctl restart crond
02. [root@server0 ~]# systemctl enable crond
03.
04. [root@server0 ~]# crontab -e -u natasha
05. 23 14 * * * /bin/echo hiya
```

步骤05：创建一个共享目录

案例概述：

创建一个共享目录/home/admins，特性如下：

- /home/admins目录的组所有权是adminuser
- adminuser组的成员对目录有读写和执行的权限。除此之外的其他所有用户没有任何权限（root用户能够访问系统中的所有文件和目录）
- 在/home/admins目录中创建的文件，其组所有权会自动设置为属于adminuser组
- [注]此处所谓的共享目录并不是指网络共享，只是某个组成员共用

解题参考：

```
01. [root@server0 ~]# mkdir /home/admins
02. [root@server0 ~]# chown :adminuser /home/admins
03. [root@server0 ~]# chmod ug+rw,o-rwx /home/admins //调整权限
04. [root@server0 ~]# chmod g+s /home/admins //设置Set UID权限
```

步骤06：安装内核的升级

案例概述：

新版内核文件从以下地址获取：

http://classroom.example.com/content/rhel7.0/x86_64/errata/Packages/

- 升级你的系统的内核版本，同时要满足下列要求：
- 当系统重新启动之后升级的内核要作为默认的内核
- 原来的内核要被保留，并且仍然可以正常启动

解题参考：

[Top](#)

```
01. [root@server0 ~]# firefox \
02. http://classroom.example.com/content/rhel7.0/x86_64/errata/Packages/
```

```
03.                                     //根据所给地址找到内核文件，复制其下载地址
04. [root@server0 ~]# wget \
05.     http://classroom.example.com/content/rhel7.0/x86_64/errata/Packages/kernel-3.
06. [root@server0 ~]# rpm -ivh kernel-3.10*.rpm           //安装新内核（耐心等待...）
07. [root@server0 ~]# reboot                             //重启以使新内核生效
08. [root@server0 ~]# uname -r
09. 3.10.0-123.1.2.el7.x86_64                          //确认新内核版本
```

步骤07：绑定到外部验证服务

案例概述：

系统 classroom.example.com 提供了一个 LDAP 验证服务。您的系统需要按照以下要求绑定到这个服务上：

- 验证服务器的基本 DN 是：dc=example,dc=com
- 帐户信息和验证信息都是由 LDAP 提供的
- 连接要使用证书进行加密，证书可以在下面的链接中下载：
- <http://classroom.example.com/pub/example-ca.crt>
- 当正确完成配置后，用户 ldapuser0 应该能够登录到您的系统中，但是没有主目录。当您完成 autofs 的题目之后，才能生成主目录
- 用户 ldapuser0 的密码是 password

解题参考：

```
01. [root@server0 ~]# yum -y install sssd
02. [root@server0 ~]# authconfig-tui           //使用简易配置工具
```

根据提示完成用户和认证方式设置 ——

User Information : [*] Use LDAP

Authentication Method : [*] Use LDAP Authentication

根据提示选中 [*] Use TLS，并设置下列参数 ——

Server : classroom.example.com

Base DN : dc=example,dc=com

提示下载证书到 /etc/openldap/cacerts 目录时，另开一终端执行：

```
01. [root@server0 ~]# cd /etc/openldap/cacerts/           //进入CA机构证书目录
02. [root@server0 ~]# wget http://classroom.example.com/pub/example-ca.crt
```

然后回到 authconfig-tui 工具确认，稍等片刻即可。

[Top](#)

```
01. [root@server0 ~]# systemctl restart sssd
```

```
02. [root@server0 ~]# systemctl enable sssd
03. [root@server0 ~]# id ldapuser0 //验证LDAP用户可用
04. uid=1700(ldapuser0) gid=1700(ldapuser0) groups=1700(ldapuser0)
```

步骤08：家目录漫游

案例概述：

按照下述要求配置手动挂载 LDAP 用户的主目录：

- classroom.example.com (172.25.254.254) 通过 NFS 输出 /home/guests 目录到您的系统，这个文件系统包含了用户ldapuser0的主目录，并且已经预先配置好了
- ldapuser0用户的主目录是 classroom.example.com:/home/guests/ldapuser0
- ldapuser0的主目录应该挂载到本地的/home/guests/ldapuser0 目录下
- 用户对其主目录必须是可写的
- ldapuser0用户的密码是password

解题参考：

```
01. [root@server0 ~]# mkdir /home/guest/ldapuser0
02. [root@server0 ~]# mount classroom.example.com:/home/guests/ldapuser0 /home/
03.
04. [root@server0 ~]# su - ldapuser0 -c 'pwd' //验证结果
05. /home/guests/ldapuser0
```

步骤09：配置NTP网络时间客户端

案例概述：

配置您的系统，让其作为一个 classroom.example.com 的 NTP 客户端

解题参考：

```
01. [root@server0 ~]# yum -y install chrony
02. [root@server0 ~]# vim /etc/chrony.conf
03. #server 0.rhel.pool.ntp.org iburst //注释掉不可用server配置，
04. #server 1.rhel.pool.ntp.org iburst
05. #server 2.rhel.pool.ntp.org iburst
06. #server 3.rhel.pool.ntp.org iburst
07. server classroom.example.com iburst //添加新的配置
08. ...
09. [root@server0 ~]# systemctl restart chronyd
10. [root@server0 ~]# systemctl enable chronyd
```

[Top](#)

步骤10：查找文件

案例概述：

找出所有用户student拥有的文件，并且把它们拷贝到/root/findfiles 目录中

解题参考：

```
01. [root@server0 ~]# mkdir /root/findfiles
02. [root@server0 ~]# find / -user student -type f -exec cp -p {} /root/findfiles/ \;
```

步骤11：查找一个字符串

案例概述：

在文件/usr/share/dict/words中查找到所有包含字符串seismic的行：

- 将找出的行按照原文的先后顺序拷贝到/root/wordlist文件中
- /root/wordlist文件不要包含空行，并且其中的所有行的内容都必须是 /usr/share/dict/words文件中原始行的准确副本

解题参考：

```
01. [root@server0 ~]# grep 'seismic' /usr/share/dict/words > /root/wordlist
```

步骤12：创建一个归档

案例概述：

创建一个名为 /root/backup.tar.bz2 的归档文件，其中包含 /usr/local 目录中的内容，tar 归档必须使用 bzip2 进行压缩

解题参考：

```
01. [root@server0 ~]# tar -jcf /root/backup.tar.bz2 /usr/local/
```