

# NSD SERVICES DAY01

1. [案例1：补充应用技巧](#)
2. [案例2：软连接与硬连接](#)
3. [案例3：man手册、zip备份](#)
4. [案例4：自定义yum软件仓库](#)
5. [案例5：发布及测试yum仓库](#)
6. [案例6：vim效率操作](#)
7. [案例7：编译安装软件包](#)
8. [案例8：使用systemctl工具](#)

## 1 案例1：补充应用技巧

### 1.1 问题

本例要求掌握在运维中比较常用的一些扩展命令技巧的使用，完成下列小技巧操作：

1. 采用数值形式将目录/root的权限调整为 rwx-----
2. 将记录的历史命令条数更改为 200 条
3. 统计 /boot、/etc/pki 目录占用的空间大小
4. 以格式 “yyyy-mm-dd HH:MM” 显示系统时间

### 1.2 方案

date日期时间工具：

- 显示日期时间：date、date + "%Y-%m-%d %H:%M:%S"
- 调整日期时间：date -s "yyyy-mm-dd HH:MM:SS"
- 恢复为硬件时间：hwclock -s

### 1.3 步骤

[Top](#)

实现此案例需要按照如下步骤进行。

## 步骤一：采用数值形式将目录/root的权限调整为 rwx-----

### 1) 查看原来的权限

```
01. [root@svr7 ~]# ls -ld /root/
02. dr-xr-x---. 22 root root 4096 3月 26 14:59 /root/
```

### 2) 修改为新权限

```
01. [root@svr7 ~]# chmod 700 /root/
```

### 3) 确认权限设置结果

```
01. [root@svr7 ~]# ls -ld /root/
02. drwx-----. 22 root root 4096 3月 26 14:59 /root/
```

## 步骤二：将记录的历史命令条数更改为 200 条

### 1) 调整记录条数

修改配置文件/etc/profile，找到HISTSIZE行，将此变量的值修改为200：

```
01. [root@svr7 ~]# vim /etc/profile
02. .. ..
03. HISTSIZE = 200
```

[Top](#)

## 2) 确认设置结果

所有用户重新登录以后即可生效：

```
01. [root@svr7 ~]# su - root
02. [root@svr7 ~]# echo $HISTSIZE
03. 200
```

## 步骤三：统计 /boot、/etc/pki 目录占用的空间大小

### 1) 分别统计结果

```
01. [root@svr7 ~]# du -sh /boot/ /etc/pki/
02. 130M /boot/
03. 1.5M /etc/pki/
```

### 2) 比较du与ls查看文件大小的差异（默认块大小4096字节）：

```
01. [root@svr7 ~]# ls -lh /etc/inittab //数据大小511字节
02. -rw-r--r--. 1 root root 511 Sep 16 2015 /etc/inittab
03. [root@svr7 ~]# du -sh /etc/inittab //实际占用4KB磁盘空间
04. 4.0K /etc/inittab
```

## 步骤四：以格式“yyyy-mm-dd HH:MM”显示系统时间

[Top](#)

```
01. [root@svr7 ~]# date +"%F %R"  
02. 2016-12-26 16:23
```

## 2 案例2：软连接与硬连接

### 2.1 问题

本例要求理解软连接与硬连接的基本差异，完成下列操作：

1. 新建文件 file1，内容为 AAAA
2. 为 file1 建立软连接 file1-s，对比两文件内容
3. 为 file1 建立硬连接 file1-h，对比两文件内容
4. 对比上述 3 个文件的 i 节点编号
5. 删除文件 file1，再查看文件 file1-s、file1-h 内容

### 2.2 方案

软连接与硬连接：

- 软连接：指向原始文件的路径，若原始文件被删除，连接文件将失效；原始文件可以是目录；原始文件与连接文件可以在不同的分区/文件系统
- 硬连接：指向原始文件的i节点档案，若原始文件被删除，连接文件仍然有效；原始文件不能是目录；原始文件与连接文件必须在同一个分区/文件系统

### 2.3 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：使用ln命令为文档/目录建立连接

1) 新建一个测试文件

```
01. [root@svr7 ~]# vim file1  
02. AAAA
```

[Top](#)

## 2 ) 为文件file1建立软连接file1-s并测试

```
01. [root@svr7 ~]# ln -s file1 file1-s
02. [root@svr7 ~]# cat file1-s
03. linux.tedu.cn
```

## 3 ) 为文件file1建立硬连接file1-h并测试

```
01. [root@svr7 ~]# ln file1 file1-h
02. [root@svr7 ~]# cat file1-h
03. linux.tedu.cn
```

## 4 ) 对比原始文件、软连接、硬连接的属性

可以发现软连接只是一个快捷方式，而硬连接与原始文件的i节点编号相同，其实对应同一块磁盘存储：

```
01. [root@svr7 ~]# ls -li /root/f0*.txt
02. 204645793 -rw-r--r--. 2 root root 14 Jan 6 12:14 file1-h
03. 201628464 lrwxrwxrwx. 1 root root 12 Jan 6 12:16 file1-s -> file1
04. 204645793 -rw-r--r--. 2 root root 14 Jan 6 12:14 file1
```

## 步骤二：原始文件删除测试

[Top](#)

### 1 ) 当原始文件被删除时，软连接将会失效，而硬连接仍然可访问文件数据

```
01. [root@svr7 ~]# rm -rf file1
02. [root@svr7 ~]# cat file1-s
03. cat: file1-s: No such file or directory
04. [root@svr7 ~]# cat file1-h
05. linux.tedu.cn
```

2) 如果已知原始文件和硬连接的路径, 当原始文件丢失时, 可以快速重建

```
01. [root@svr7 ~]# ln file1-h file1
02. [root@svr7 ~]# ls -li /root/f0*.txt
03. 204645793 -rw-r--r-- 2 root root 14 Jan 6 12:14 file1-h
04. 201628464 lrwxrwxrwx. 1 root root 12 Jan 6 12:16 file1-s -> file1
05. 204645793 -rw-r--r-- 2 root root 14 Jan 6 12:14 file1
```

3) 不支持为目录创建硬连接, 但可以为目录建立软连接

```
01. [root@svr7 ~]# ln /etc/sysconfig/network-scripts/ /etc/network
02. ln: '/etc/sysconfig/network-scripts/': hard link not allowed for directory
03. [root@svr7 ~]# ln -s /etc/sysconfig/network-scripts/ /etc/interface
04. [root@svr7 ~]# ls -l /etc/interface
05. lrwxrwxrwx. 1 root root 31 Jan 6 12:28 /etc/interface -> /etc/sysconfig/network-scripts/
```

## 3 案例3: man手册、zip备份

### 3.1 问题

[Top](#)

本例要求掌握man帮助手册的使用，以及zip压缩/解压缩的操作，完成下列任务：

1. 查阅passwd命令、/etc/passwd配置文件的手册页
2. 使用zip打包/usr/share/doc/qemu-kvm/目录

## 3.2 方案

zip/unzip压缩与解压缩：

- 制作zip压缩包：zip [-r] 备份文件.zip 被归档的文档...
- 释放zip压缩包：unzip 备份文件.zip [-d 目标文件夹]

## 3.3 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：使用man手册页获取帮助

1 ) 查看passwd命令的手册页

```
01. [root@svr7 ~]# man passwd
02. PASSWD(1)                User utilities          PASSWD(1)
03.
04. NAME
05.     passwd - update user's authentication tokens
06.
07. SYNOPSIS
08.     passwd [-k] [-l] [-u [-f]] [-d] [-e] [-n minday s] [-x maxday s] [-w
09.     warnday s] [-i inactiveday s] [-S] [--stdin] [username]
10.
11. DESCRIPTION
12.     The passwd utility is used to update user's authentication token(s).
13.
```

[Top](#)

14. This task is achieved through calls to the Linux- PAM and Libuser API.
15. Essentially , it initializes itself as a "passwd" service with Linux-
16. PAM and utilizes configured password modules to authenticate and then
17. update a user's password.
18. . . .

## 2 ) 查看/etc/passwd配置文件的手册页

01. [ root@svr7 ~] # man 2 passwd
02. PASSWD( 5)                      Linux Programmer's Manual                      PASSWD( 5)
- 03.
04. NAME
05.        passwd - password file
- 06.
07. DESCRIPTION
08.        The /etc/passwd file is a text file that describes user login
09.        accounts for the system. It should have read permission allowed for
10.        all users ( many utilities, like ls( 1) use it to map user IDs to user-
11.        names) , but write access only for the superuser.
- 12.
13.        In the good old days there was no great problem with this general
14.        read permission. Everybody could read the encrypted passwords, but
15.        the hardware was too slow to crack a well- chosen password, and more-
16.        over the basic assumption used to be that of a friendly user- commu-
17.        nity . These days many people run some version of the shadow password
18.        suite, where /etc/passwd has an 'x' character in the password
19.        . . .

[Top](#)



## 步骤二：使用zip命令制作压缩包

1 ) 将目录/usr/share/doc/qemu-kvm/备份为/root/qemu-kvm.zip

```
01. [ root@svr7 ~] # zip -r /root/qemu-kvm.zip /usr/share/doc/qemu-kvm/
02.   adding: usr/share/doc/qemu-kvm/ ( stored 0%)
03.   adding: usr/share/doc/qemu-kvm/COPYING ( deflated 62%)
04.   adding: usr/share/doc/qemu-kvm/COPYING.LIB ( deflated 65%)
05.   adding: usr/share/doc/qemu-kvm/Changelog ( deflated 61%)
06.   adding: usr/share/doc/qemu-kvm/LICENSE ( deflated 45%)
07.   adding: usr/share/doc/qemu-kvm/README ( deflated 4%)
```

2 ) 恢复测试

删除目标文件夹并确认结果：

```
01. [ root@svr7 ~] # rm -rf /usr/share/doc/qemu-kvm/
02. [ root@svr7 ~] # ls /usr/share/doc/qemu-kvm/
03. ls: cannot access /usr/share/doc/qemu-kvm/: No such file or directory
```

恢复目标文件夹并确认结果：

```
01. [ root@svr7 ~] # unzip /root/qemu-kvm.zip -d /
02. Archive: /root/qemu-kvm.zip
03.   creating: /usr/share/doc/qemu-kvm/
```

[Top](#)

```
04.   inflating: /usr/share/doc/qemu-kvm/COPYING
05.   inflating: /usr/share/doc/qemu-kvm/COPYING.LIB
06.   inflating: /usr/share/doc/qemu-kvm/Changelog
07.   ...
08.   [root@svr7 ~]# ls /usr/share/doc/qemu-kvm/
09.   COPYING      README          qemu-tech.html
10.   COPYING.LIB  README.rhel6  gpxe-source  qmp-commands.txt
11.   Changelog    README.systemtap  qmp-events.txt
12.   LICENSE     qemu-doc.html    qmp-spec.txt
```

## 4 案例4：自定义yum软件仓库

### 4.1 问题

本例要求在CentOS真机上利用RHEL7的光盘镜像文件准备一个软件仓库目录，完成下列任务：

1. 创建目录 /var/www/html/rh7dvd
2. 挂载 rhel-server-7.2-x86\_64-dvd.iso 到上述目录

另外，利用收集的一些第三方RPM软件包文件，配置为可发布的yum仓库目录，相关任务如下：

1. 挂载RHEL7光盘镜像文件到 /var/www/html/ 目录
2. 下载 LibreOffice 办公软件的 rpm 集合版文件
3. 将其中的内容释放到 /opt/libreoffice/ 目录
4. 为 /opt/libreoffice/ 目录建立仓库档案

### 4.2 方案

作为yum软件源的目录需要准备的内容：

- 大量的 .rpm 软件安装包文件
- 针对这些软件包的 repodata/ 仓库档案

[Top](#)

repodata/ 仓库档案提供的数据：

- filelists.xml.gz : 提供所有软件包的文件安装清单
- primary.xml.gz : 提供所有软件包的基本/主要信息
- other.xml.gz : 提供所有软件包的其他信息
- repomd.xml : 提供上述档案数据文件.xml.gz的下载和校验信息

## 4.3 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：准备 /var/www/html/rh7dvd 仓库目录

1 ) 创建目录 /var/www/html/rh7dvd

```
01. [root@room9pc13 ~]# mkdir /var/www/html/rh7dvd
```

2 ) 挂载 rhel-server-7.2-x86\_64-dvd.iso 到上述目录

```
01. [root@room9pc13 ~]# vim /etc/fstab
02. ...
03. /ISO/rhel-server-7.2-x86_64-dvd.iso /var/www/html/rh7dvd iso9660 loop,ro 0 0
04. [root@room9pc13 ~]# mount -a
```

3 ) 确认部署结果

```
01. [root@room9pc13 ~]# ls /var/www/html/rh7dvd/
02. addons images Packages RPM GPG KEY redhat release
03. EFI isolinux release-notes TRANS.TBL
04. EULA LiveOS repodata
```

[Top](#)

05. GPL media.repo RPM GPG KEY redhat beta

## 步骤二：准备 /opt/libreoffice/ 仓库目录

1) 将获取的LibreOffice软件集合包释放到指定目录

```
01. [root@room9pc13 ~]# ls LibreOffice_5.1.6.2_Linux_x86_64_rpm.zip
02. LibreOffice_5.1.6.2_Linux_x86_64_rpm.zip
03. [root@room9pc13 ~]# unzip LibreOffice_5.1*.zip -d /opt/libreoffice
04. Archive: LibreOffice_5.1.6.2_Linux_x86_64_rpm.zip
05.   inflating: /opt/libreoffice/install
06.   creating: /opt/libreoffice/langpack_zh-CN/
07.   creating: /opt/libreoffice/langpack_zh-CN/RPMS/
08. ... .. //释放到 /opt/libreoffice 目录
```

2) 使用createrepo建立档案

```
01. [root@room9pc13 ~]# createrepo /opt/libreoffice/
02. Spawning worker 0 with 53 pkgs
03. Workers Finished
04. Saving Primary metadata
05. Saving file lists metadata
06. Saving other metadata
07. Generating sqlite DBs
08. Sqlite DBs complete
```

[Top](#)

### 3 ) 确认repodata/档案资料

```
01. [ root@room9pc13 ~] # ls /opt/libreoffice/repodata
02. 1a5d8311268f33ad2cbf91382110e1ef9875aeea366897253a5d27fd42f9e317- other.xml.gz
03. 2cd176f0b00492c0c13e0a659eda7dedeb1ab526dec5fd7c9bac7758558770d2- filelists.xml.gz
04. 6ecab3585a93a917202e177b9569046238332af449a6492fcace96ea79374668- filelists.sqlite.bz2
05. b09a1f838262e0b67a5ab0d6be516bd921a914afd89864e09650f95662a20371- primary.sqlite.bz2
06. b6fcf4a24de5dc08585bf52bd34be50d7df376d5fbcf50903bfd3c1dfdf160bf- other.sqlite.bz2
07. f57724cf309cc102b2ee25596bf8fb39db3c23880404209ac1e379f7b9fd5c49- primary.xml.gz
08. repomd.xml
```

## 5 案例5：发布及测试yum仓库

### 5.1 问题

沿用案例5，本例要求掌握发布及测试yum仓库的方法，方便在网络内提供集中的yum源服务器，主要完成下列任务：

1. 在CentOS真机上发布yum源，包括：rhel7 系统的光盘目录仓库、LibreOffice 的rpm软件包仓库
2. 在主机 pc207 上使用上述yum源

### 5.2 方案

通过网络发布yum软件源时，只需要配置HTTP或FTP资源服务器，然后将提前准备好的yum仓库目录部署到可访问的资源位置即可。

在访问网络yum软件源时，注意客户端的baseurl地址必须与资源提供方式一致：

- baseurl = http://服务器地址/目录名 ==》 /var/www/html/目录名
- baseurl = ftp://服务器地址/目录位置 ==》 /var/ftp/目录名

### 5.3 步骤

[Top](#)

实现此案例需要按照如下步骤进行。

## 步骤一：在CentOS真机上发布yum仓库

### 1) 快速搭建httpd服务器（若已建好，此步可跳过）

```
01. [root@room9pc13 ~]# yum -y install httpd //装包
02. [root@room9pc13 ~]# systemctl restart httpd //起服务
03. [root@room9pc13 ~]# systemctl enable httpd //设置开机自启
```

### 2) 确认前一步已经部署到Web网站目录的RHEL7光盘数据：

```
01. [root@room9pc13 ~]# du -sh /var/www/html/rh7dvd/ //检查部署结果
02. 3.9G /var/www/html/rh7dvd/
```

### 3) 将准备好的LibreOffice仓库目录部署到Web网页目录

```
01. [root@room9pc13 ~]# mv /opt/libreoffice/ /var/www/html/
02. [root@room9pc13 ~]# du -sh /var/www/html/libreoffice/ //检查部署结果
03. 234M /var/www/html/libreoffice
```

## 步骤二：在pc207上访问yum仓库

### 1) 添加新的yum仓库设置

```
01. [root@pc207 ~]# vim /etc/yum.repos.d/new.repo
02. [rh7dvd]
```

[Top](#)

```

03.  name = RHEL 7.2 Server
04.  baseurl = http://192.168.4.254/rh7dvd
05.  gpgcheck = 0
06.
07.  [ libreoffice ]
08.  name = LibreOffice 5
09.  baseurl=http://192.168.4.254/libreoffice
10.  gpgcheck = 0

```

## 2 ) 测试新的yum仓库

```

01.  [ root@pc207 ~ ] # yum repolist
02.  ...
03.  repo id      repo name      status
04.  libreoffice  LibreOffice 5  53
05.  rh7dvd       RHEL 7.2 Server 4620
06.  ...

```

## 6 案例6 : vim效率操作

### 6.1 问题

本例要求掌握使用vim文本编辑器时能够提高操作效率的一些常用技巧和方法，完成下列任务：

1. 将文件 /etc/passwd 复制为 /opt/nsd.txt，然后打开 /opt/nsd.txt 文件，练习命令模式下的切换/复制/删除/查找操作
2. 将文件 /etc/man\_db.conf 复制到 /opt 目录下，然后打开 /opt/man\_db.conf 文件，将第50~100行内的 “man” 替换为 “MAN”，在vim中设置显示行号查看效果

### 6.2 方案

命令模式常用操作：

- 1G 或 gg ，跳转到文件的首行
- G ，跳转到文件的末尾行
- yy、#yy ，复制光标处的一行、#行
- p、P ，粘贴到光标处之后、之前
- x 或 Delete键 ，删除光标处的单个字符
- dd、#dd ，删除光标处的一行、#行
- d^、d\$ ，从光标处之前删除至行首/行尾
- /word 向后查找字符串 “word” ，再按n/N跳至后/前一个结果
- u ，撤销最近的一次操作
- U ，撤销对当前行的所有修改
- Ctrl + r 取消前一次撤销操作
- ZZ 保存修改并退出

末行模式常用操作：

- :s/old/new ，替换当前行第一个 “old”
- :s/old/new/g ，替换当前行所有的 “old”
- :n,m s/old/new/g ，替换第n-m行所有的 “old”
- :% s/old/new/g ，替换文件内所有的 “old”
- :w /root/newfile ，另存为其它文件
- :r /etc/filesystems ，读入其他文件内容
- :set nu|nonu ，显示/不显示行号
- :set ai|noai ，启用/关闭自动缩进

## 6.3 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：vim命令模式下的切换/复制/删除/查找

1) 建立练习文件

将文件 /etc/passwd 复制为 /opt/nsd.txt：

[Top](#)



```
01 [root@svr7 ~]# cp /etc/passwd /opt/nsd.txt
```

## 2) 使用vim打开练习文件，默认处于命令模式

```
01 [root@svr7 ~]# vim /opt/nsd.txt
```

```
02. ....
```

## 3) 在命令模式下完成下列操作

切换操作：G 最后一行，5G 第5行，gg 第一行。

复制操作：按2yy复制2行，7G移动到第7行，p 粘贴。

删除操作：25G 移动到第25行，200dd 从此行开始删除200行（不够就剩下全删）。

查找操作：gg 第一行，/adm 查找关键词adm，n 跳转到下一个结果。

## 4) 保存并退出编辑器

ZZ 保存退出。

## 步骤二：vim末行模式下的替换/设置操作

### 1) 建立练习文件

将文件 /etc/man\_db.conf 复制到 /opt/ 目录下：

```
01 [root@svr7 ~]# cp /etc/man_db.conf /opt/
```

### 2) 使用vim打开练习文件，输入:切换到末行模式

[Top](#)

```
01. [root@svr7 ~]# vim /opt/man_db.conf
02. ...
03. :
```

3) 在末行模式下完成下列操作

输入 :set nu ，确认后显示行号。

输入 :50,100 s/man/MAN/g ，确认将第50~100行内的 “man” 替换为 “MAN” 。

4) 保存并退出编辑器

输入 :wq ，确认后保存并退出编辑器。

## 7 案例7：编译安装软件包

### 7.1 问题

本例要求掌握常规源代码应用的安装过程，通过编译的方式安装inotify-tools 软件工具，完成下列任务：

1. 释放 inotify-tools-3.13.tar.gz 源码包
2. 配置 ./configure
3. 编译 make、安装 make install
4. 测试inotifywait监控工具的用法及用途

### 7.2 方案

对于标准源码发布的C/C++ 软件包，编译安装一般包括以下过程：

1. 解包：使用tar命令，将下载的源代码释放至指定目录
2. 配置：执行源码目录内的 ./configure 脚本，指定安装目录/功能模块等选项
3. 编译：在源码目录下执行 make 操作，根据配置清单Makefile生成可执行的二进制程序文件
4. 安装：在源码目录下执行make install 操作，将编译好的程序及相关文件复制到安装目录

[Top](#)

### 7.3 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：确认已配置好编译环境

```
01. [root@svr7 ~]# yum -y install gcc gcc-c++ make
02. ...
03. [root@svr7 ~]# gcc --version
04. gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-4)
05. Copyright (C) 2015 Free Software Foundation, Inc.
06. This is free software; see the source for copying conditions. There is NO
07. warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

### 步骤二：编译安装inotify-tools软件包

#### 1) 解包inotify-tools-3.13.tar.gz文件

```
01. [root@svr7 ~]# ls inotify-tools-3.13.tar.gz
02. inotify-tools-3.13.tar.gz
03. [root@svr7 ~]# tar xf inotify-tools-3.13.tar.gz -C /usr/src/
```

#### 2) 配置 ./configure , 安装目录默认 ( /usr/local/\*/ )

```
01. [root@svr7 ~]# cd /usr/src/inotify-tools-3.13/ //进入源码目录
02. [root@svr7 inotify-tools-3.13]# ./configure //配置操作
03. checking for a BSD-compatible install... /usr/bin/install -c
04. checking whether build environment is sane... yes
```

[Top](#)

```
05.  checking for gawk... gawk
06.  ...
07.  configure: creating ./config.status
08.  config.status: creating Makefile
09.  ...
10.
11.  [ root@svr7 inotify- tools- 3.13] # ls  Makefile           //检查配置结果
12.  Makefile
```

### 3 ) 编译 make

```
01.  [ root@svr7 inotify- tools- 3.13] # make
02.  ...
03.  Making all in src
04.  make[ 2]: Entering directory `/usr/src/inotify- tools- 3.13/src'
05.  make[ 3]: Entering directory `/usr/src/inotify- tools- 3.13'
06.  make[ 3]: Leaving directory `/usr/src/inotify- tools- 3.13'
07.  ...
```

### 4 ) 安装 make install

```
01.  [ root@svr7 inotify- tools- 3.13] # make  install
02.  ...
03.  /usr/bin/install -c .libs/inotifywait /usr/local/bin/inotifywait
04.  /bin/sh ../libtool --mode=install /usr/bin/install -c 'inotifywatch' '/usr/local/bin/inotifywatch'
05.  ...
```

[Top](#)

```
06.
07. [ root@svr7 inotify- tools- 3.13] # find /usr/local/ - name "inotify*"
08. /usr/local/bin/inotifywait //确认安装结果
09. /usr/local/bin/inotifywatch
10. /usr/local/include/inotifytools
11. /usr/local/include/inotifytools/inotifytools.h
```

### 步骤三：测试inotify-tools软件程序

软件包inotify-tools提供了一个主要程序inotifywait，可以用来监控指定目录或文档的变化，并及时给出通知。

#### 1) 开启对/opt目录的事件监控

```
01. [ root@svr7 ~] # inotifywait -mrq /opt & //开启监控
02. [ 1] 15568
```

#### 2) 修改/opt/目录内容，观察屏幕输出信息

```
01. [ root@svr7 ~] # touch /opt/a.txt //新建文件a.txt
02. /opt/ CREATE a.txt
03. /opt/ OPEN a.txt
04. /opt/ ATTRIB a.txt
05. /opt/ CLOSE_WRITE, CLOSE a.txt
06.
07. [ root@svr7 ~] # mv /opt/a.txt /opt/b.txt //将文件改名
08. /opt/ MOVED_FROM a.txt
09. /opt/ MOVED_TO b.txt
```

[Top](#)

### 3) 结束inotifywait监控

杀死当前用户的第一个后台任务：

```
01. [ root@svr7 ~] # kill -9 %1
02. [ 1] + Killed          inotifywait - mrq /opt
```

## 8 案例8：使用systemctl工具

### 8.1 问题

本例要求掌握systemctl控制工具的基本操作，完成下列任务：

1. 重启 httpd、crond、bluetooth 服务，查看状态
2. 禁止 bluetooth 服务开机自启，并停用此服务
3. 设置默认级别为 multi-user.target 并确认

### 8.2 方案

systemd是一个更高效的系统&服务管理器，其相关特性如下：

- 开机服务并行启动，各系统服务间的精确依赖
- 配置目录：/etc/systemd/system/
- 服务目录：/lib/systemd/system/

systemctl是systemd的管理工具，将相关资源组织为unit配置单元进行管理。

不同的unit决定了一组相关的启动任务，service和target是最常用的配置单元：

- service：后台独立服务
- target：一套配置单元的组合，类似于传统“运行级别”

[Top](#)

### 8.3 步骤

实现此案例需要按照如下步骤进行。

## 步骤一：重启 httpd、crond、bluetooth 服务，查看状态

### 1 ) 重启系统服务httpd、crond、bluetooth

```
01. [root@svr7 ~]# systemctl restart httpd crond bluetooth
```

### 2 ) 查看上述服务的状态

```
01. [root@svr7 ~]# systemctl status httpd crond bluetooth
02. * httpd.service - The Apache HTTP Server
03.    Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
04.    Active: active (running) since Fri 2017-01-06 18:18:20 CST; 18s ago
05.    ...
06.
07. * crond.service - Command Scheduler
08.    Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)
09.    Active: active (running) since Fri 2017-01-06 18:18:19 CST; 19s ago
10.    ...
11.
12. * bluetooth.service - Bluetooth service
13.    Loaded: loaded (/usr/lib/systemd/system/bluetooth.service; enabled; vendor preset: enabled)
14.    Active: active (running) since Fri 2017-01-06 18:18:19 CST; 19s ago
15.    ...
```

[Top](#)

## 步骤二：禁止 bluetooth 服务开机自启，并停用此服务

## 1 ) 停用bluetooth服务

01. [ root@svr7 ~] # systemctl stop bluetooth

## 2 ) 禁止bluetooth服务开机自启

01. [ root@svr7 ~] # systemctl disable bluetooth  
02. Removed symlink /etc/systemd/system/dbus-org.bluez.service.  
03. Removed symlink /etc/systemd/system/bluetooth.target.wants/bluetooth.service.  
04.  
05. [ root@svr7 ~] # systemctl is-enabled Bluetooth //检查结果  
06. disabled

## 步骤三：设置默认级别为 multi-user.target 并确认

### 1 ) 查看默认运行级别

01. [ root@svr7 ~] # systemctl get-default  
02. graphical.target

### 2 ) 将默认运行级别设置为multi-user.target

01. [ root@svr7 ~] # systemctl set-default multi-user.target  
02. Removed symlink /etc/systemd/system/default.target.

[Top](#)



03. Created symlink from `/etc/systemd/system/default.target` to `/usr/lib/systemd/system/multi-user.target`.

### 3) 确认配置结果

01. `[root@svr7 ~]# systemctl get-default`

02. `multi-user.target`

根据此处的设置，重启此虚拟机后图形桌面将不再可用。

[Top](#)